

正运动机械手指令说明书

V2.6

版 权 说 明

Zmotion®

本手册版权归深圳市正运动技术有限公司所有，未经本公司授权，任何人不得翻印、抄袭本手册中的任何内容。

本手册中的信息资料仅供参考。由于改进设计和功能等原因，正运动公司保留对本资料的最终解释权！内容如有更改，恕不另行通知！



调试机器要注意安全！请务必在机器中设计有效的安全保护装置，并在软件中加入出错处理程序，否则所造成的损失，正运动技术公司没有义务或责任对此负责。

前言

感谢选用正运动的运动控制器。在使用本产品之前，请务必仔细阅读本手册以及了解必要的安全信息，注意事项，以及了解硬件接口、使用方法及函数接口，充分理解正运动控制器使用功能。

（一）安全警告



调试机器过程中，人员保持安全的距离。



请务必在机器中设计有效的安全保护装置。



请务必在软件中加入出错处理程序。



调试过程中，测试的速度在合理范围。

（二）常规安全守则

请查看下列安全防范措施以避免受伤害并防止对本产品或任何与其相连接的产品造成损伤。为避免潜在的危险，请仅按详细说明来使用本产品。

使用正确的电源线:请使用满足国家标准的电源线。

不要在潮湿的环境下操作,不要在爆炸性的空气中操作,保持产品表面清洁和干燥。

防止静电损伤:静电释放(ESD)可能会对运动控制卡及其附件中的元件造成损伤。为了防止ESD,请小心处理控制卡元件,不要触摸控制卡上元器件。不要将控制卡放置在可能产生静电的表面。在防护静电的袋子或容器内运输和储存控制卡。

（三）关于保证

保修时间:在指定的地点购买的产品的保修期为 1 年。

保修范围:如果在上述质保期内由于本公司责任发生了故障，本公司提供无偿修理。以下范围不在保修范围内：

▲对于说明书及其它手册记录的不适当环境或不适当使用引起的故障。

▲用户的装置、控制软件等引起本产品意外故障。

▲由客户对本产品的改造引起的故障。

▲火灾、地震及其它自然灾害等外部主要原因引起的故障。

产品的应用范围:本产品设计制造用于普通工业应用，超出预料的用途并对人的生命或财产造成重大的影响不在产品服务范围。

（四）技术支持和售后服务

您可以通过以下途径获得我们的技术支持和售后服务

地址：深圳市宝安区西乡钟屋一路 70 号大广发综合楼 5 楼

公司网站：<http://www.zmotion.com.cn/>

电话：0755-32976042

传真：0755-26066955

服务热线：4000-8989-66

邮箱：tech@zmotion.com.cn

技术支持值班 qq：3002647338

第一章 机械手相关基本知识

一、工业机械手定义

所谓工业机械手，根据 ISO（国际标准化机构）的技术报告定义为“自动控制、可以重复程序、多目的，具备多个有自由度的机械机能的机械”。换言之，现在的工业机械手是“与人的手腕和手掌的动作机能相类似，具备多样动作机能的自动机械”。

工业机械手拥有能够操作多种动作的动作特性，将从前困难的多品种少量生产的自动化变为可能。

二、空间中的自由度

一个物体，在空间上完全没有约束，那么它可以在 3 个正交方向上平动(X, Y, Z)，还可以以三个正交方向为轴进行转动(RX, RY, RZ)，那么就有 6 个自由度。而当把物体限定在 XY 平面上时，自由度减少为 3 个-可以在 X, Y 方向平动，可绕 Z 轴转动。

以标准 scara 为例，末端的工作点可以在 xy 方向移动(2 轴)，可以绕 z 旋转(3 轴)，可以 z 方向上下(4 轴)，最多有 4 个自由度。而 6 自由度机械手的末端工作点可在控件中任意运动，所以有 6 个自由度。

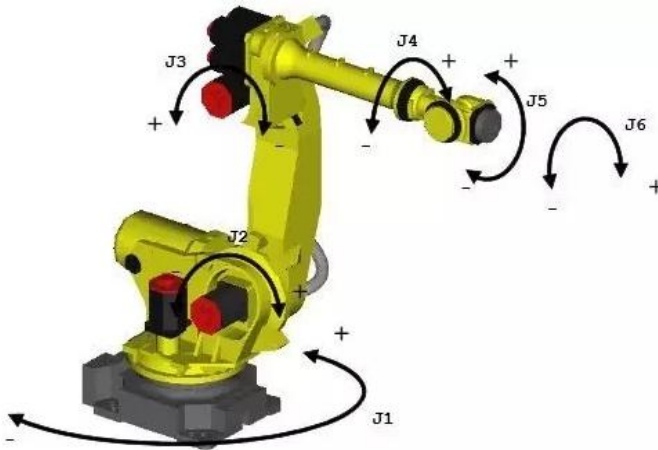
三、坐标系

1. 关节坐标系

包含机械手所有关节，各关节之间相互独立，坐标单位为角度，一般简写作 J。

操作其中一个关节时不影响其他关节坐标。

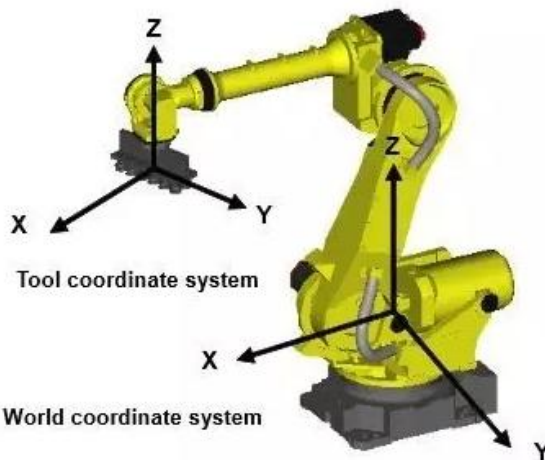
J1:0° J2:0° J3:0° J4:0° J5:0° J6:0°



2. 世界坐标系

世界坐标系是被固定在空间上的标准直角坐标系，其位置根据机械手类型确定。

虚拟轴操作时就是根据世界坐标系运动，此时各关节会自动解算需要旋转的角度。



四、关节轴与虚拟轴

关节轴是指实际机械结构中的旋转关节，在程序中一般显示旋转角度（某些结构也是平移轴）。

由于电机与旋转关节会存在减速比，所以设置 `units` 时要按照实际关节旋转一圈来设置，同时 `table` 中填写结构参数时也要按照旋转关节中心计算，而不是按照电机轴中心计算。

虚拟轴不是实际存在的，抽象为世界坐标系的 6 个自由度，依次为 X、Y、Z、RX、RY、RZ。可以理解为直角坐标系的三个直线轴和三个旋转轴，用来确定机

械手末端工作点的加工轨迹与坐标。

五、正解运动与逆解运动

通过操作关节坐标，再根据机械结构参数可以计算出末端位置在直角坐标系的空间位置，这个过程称为正解运动，此时操作的是实际关节轴，虚拟轴自动计算坐标。

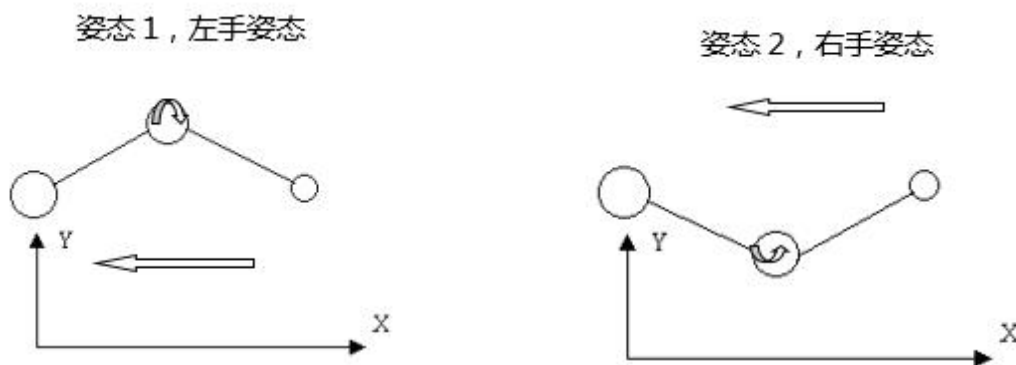
控制器使用 CONNREFRAME 指令建立正解模式，此指令作用在虚拟轴上。此时只能操作关节轴，关节轴也可以做各种运动，但实际运动的轨迹不是直线圆弧，这种模式一般用于手动调整关节位置或上电点位回零。

给定一个直角坐标系中的空间位置，反推出各关节轴坐标，这个过程称为逆解运动，此时操作的是虚拟轴，实际关节轴自动解算坐标并运动。

控制器使用 CONNFRAME 指令建立逆解模式，此指令作用在关节轴上。此时只能操作虚拟轴，虚拟轴可以在笛卡尔坐标系中做直线，圆弧，空间圆弧等运动，关节轴在 CONNFRAME 的作用下会自动运动到逆解后的位置。

六、机械手姿态

机械手姿态在数学上来说，是同一组虚拟轴数值有多组关节轴的解。即机械手在笛卡尔坐标系中运动到某一坐标点 A，可以有多种运动轨迹，这些运动轨迹就对应着不同姿态，如下图 scara 的两种姿态，在 X 方向运动，关节轴可以有两种运动方式。



七、奇异点

机械手运动到某一特定位置时，会失去某个自由度，这个位置就叫做奇异点，实际使用时要避免。如 scara 机械手完全伸直时，此时无法在 X 方向平动，操作往 x 负向运动时，结构无法判断使用哪种姿态运动，此时结构会卡住。为避免出现此种问题，需要正解时调整好姿态，逆解时规划好轨迹。

八、零点位置

机械手算法建立时，需要有个零点位置作为参考，同时需要确定好电机转向。各结构的零点位置与电机转向查看下文第 6 章各类型具体说明确认。

九、正运动机械手模式

机械手有几个电机称为几关节机械手，控制实际机械关节运动的电机轴称为机械手的关节轴，所有的关节轴构成关节坐标系，关节轴在此坐标系中按角度旋转。

控制机械手末端工作点在世界坐标系下运动，世界坐标系的坐标轴假想为虚拟轴，按距离单位移动。

关节轴和虚拟轴通过 CONNREFRAME 或 CONNFRAME 指令来指明，只要轴数足够，控制器支持多机械手。

程序可以通过运动指令(所有的运动指令都可以使用)控制关节轴或虚拟轴运动，但同一时刻只能操作虚拟轴或者关节轴。

操作关节轴运动时，虚拟轴需要位于 CONNREFRAME 模式，从而自动指向当前的空间坐标；操作虚拟轴运动时，关节轴需要位于 CONNFRAME 模式，从而自动指向当前的关节轴坐标。

机械手建立是通过 CONNREFRAME（正解）和 CONNFRAME（逆解）指令，CONNREFRAME 时虚拟轴 MTYPE（运动类型）值为 34，CONNFRAME 时关节轴 MTYPE 值为 33，可以通过 MTYPE 来查看特定轴是否位于对应的模式。

通过 CANCEL 或 RAPIDSTOP 指令可以取消机械手模式。


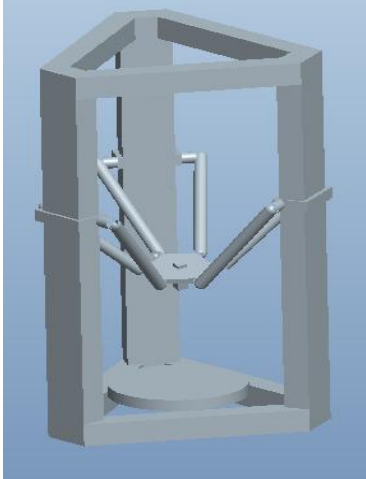
第二章 硬件选配

一、支持的机械手类型

并联机械手类

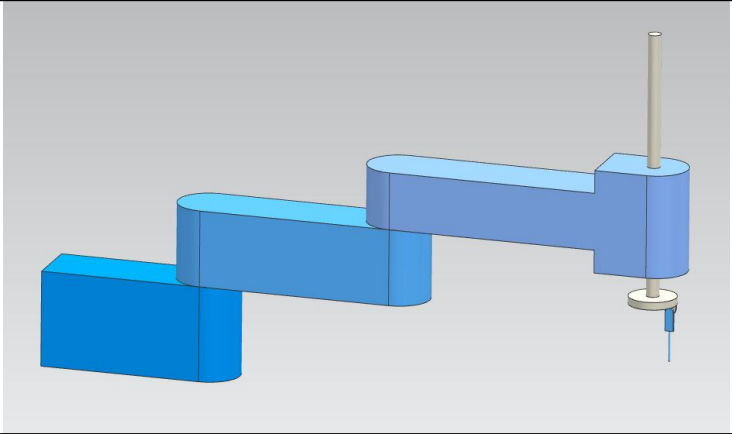
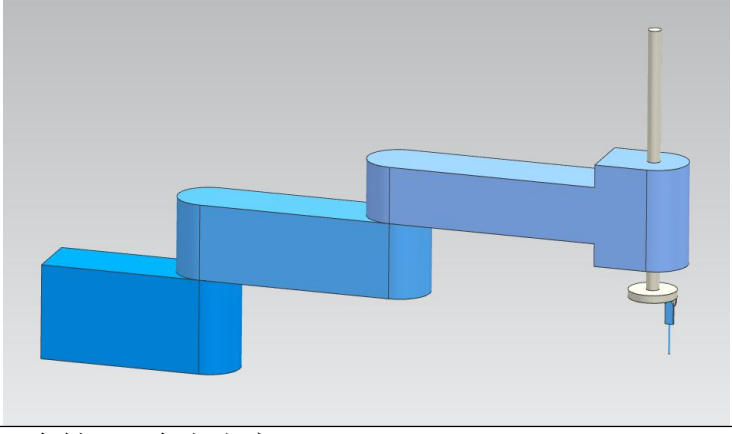

类型	


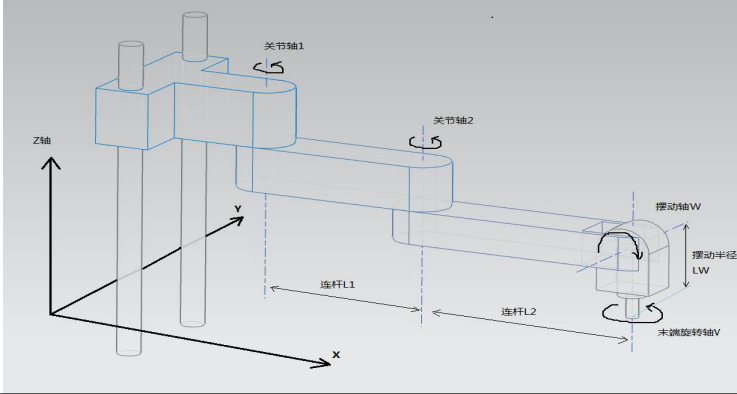
Frame2	图纸	
	信息	Delta 基本类型。 3 个轴，末端点 3 个自由度。
	区别	
Frame12	图纸	
	信息	4 个轴，末端点 4 个自由度。
	区别	与 frame2 比较，增加了末端的旋转轴。
Frame14	图纸	
	信息	4 个轴，4 个自由度。
	区别	与 frame12 比较，末端摆动轴使用虚拟轴控制。

Frame102	图纸	
	信息 区别	2 个轴，末端点 2 个自由度。 与 frame2 比较，两轴的 delta。
Frame13	图纸	
	信息 区别	3 个轴

水平关节类

类型		
Frame1	图纸	
	信息	Scara 基本类型。 4 个轴，末端点 4 个自由度。

	区别	
Frame101	图纸	
	信息区别	5 个轴，末端点 5 个自由度。 与 frame1 比较，增加了末端的摆动轴。
Frame105	图纸	
	信息区别	5 个轴，5 个自由度。 与 frame101 比较，末端摆动轴使用虚拟轴控制。
Frame106	图纸	
	信息区别	4 个轴，末端点 4 个自由度。 与 frame1 比较，大关节与小关节是关联的。

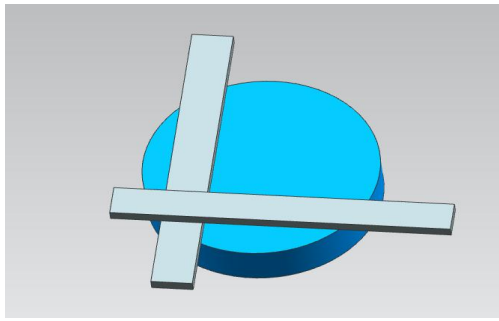
Frame107	图纸	
	信息	4 个轴，末端点 4 个自由度。
Frame108	区别	与 frame1 相比较，小关节与末端旋转轴是关联的。
	图纸	
Frame108	信息	5 个轴，5 个自由度。
	区别	与 frame105 相比较，末端摆动轴与末端旋转轴的链接机构不同。

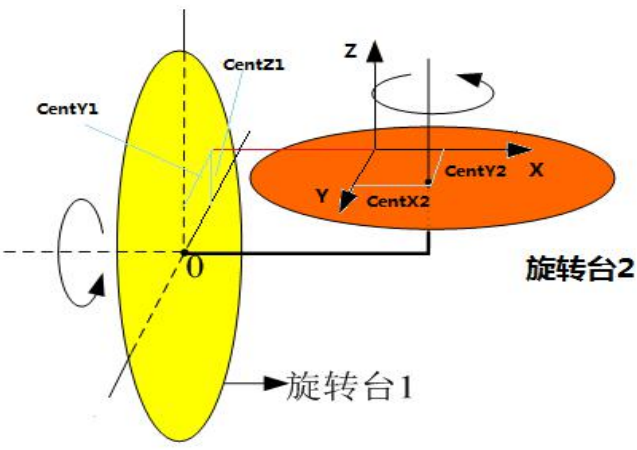
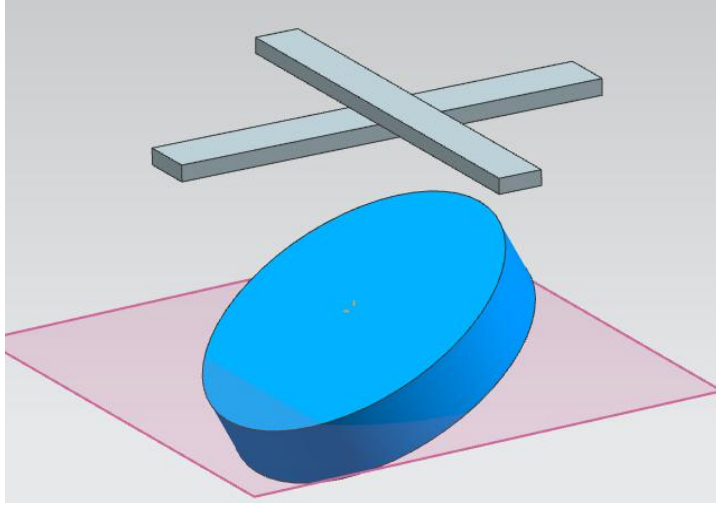
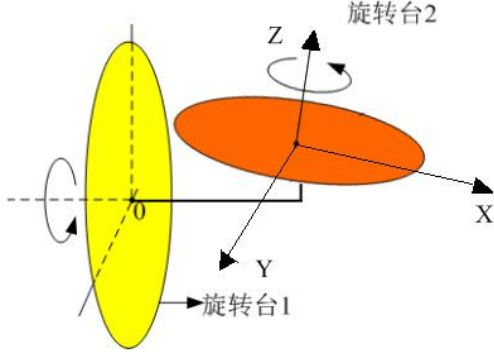
6 自由度类

类型		
Frame6	图纸	
	信息	6 个轴，6 个自由度。

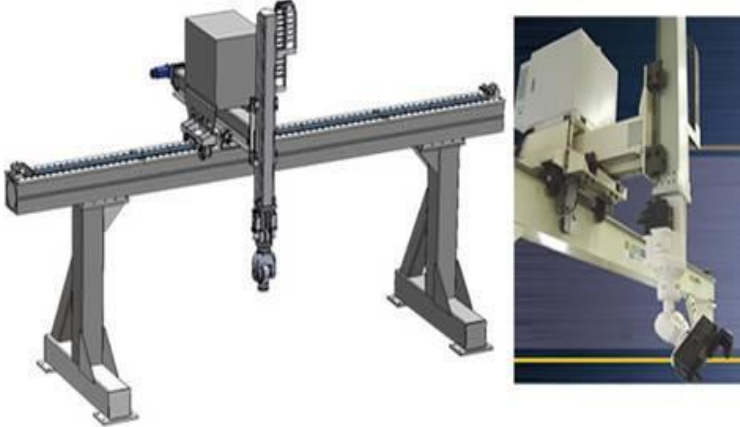
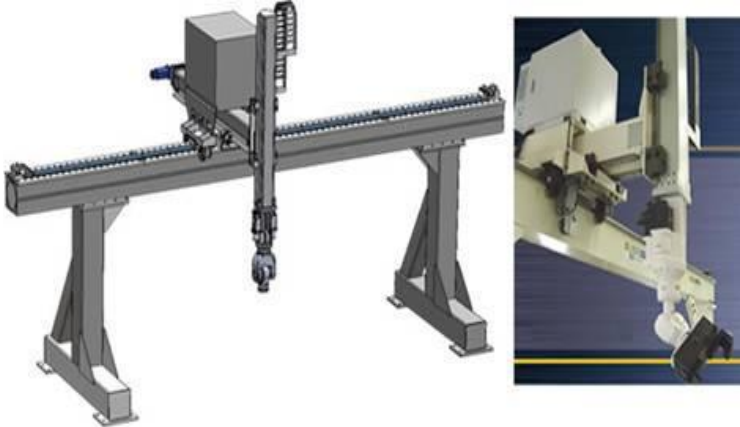
	区别	
Frame26	图纸	
	信息	6 个轴, 6 个自由度。
	区别	与 frame6 相比较, 第 2 关节与第 3 关节关联。
Frame36	图纸	
	信息	6 个轴, 6 个自由度。
	区别	与 frame6 相比较, 第 4 关节轴心 Y 方向的偏移参数。
Frame16	图纸	
	信息	6 个轴, 6 个自由度。
	区别	XYZ+3 个旋转。

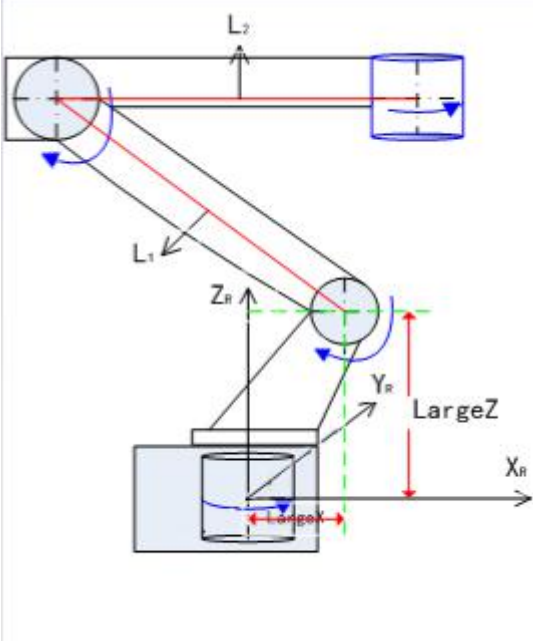
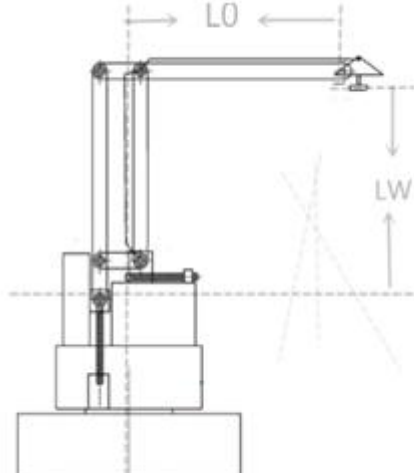
旋转台类

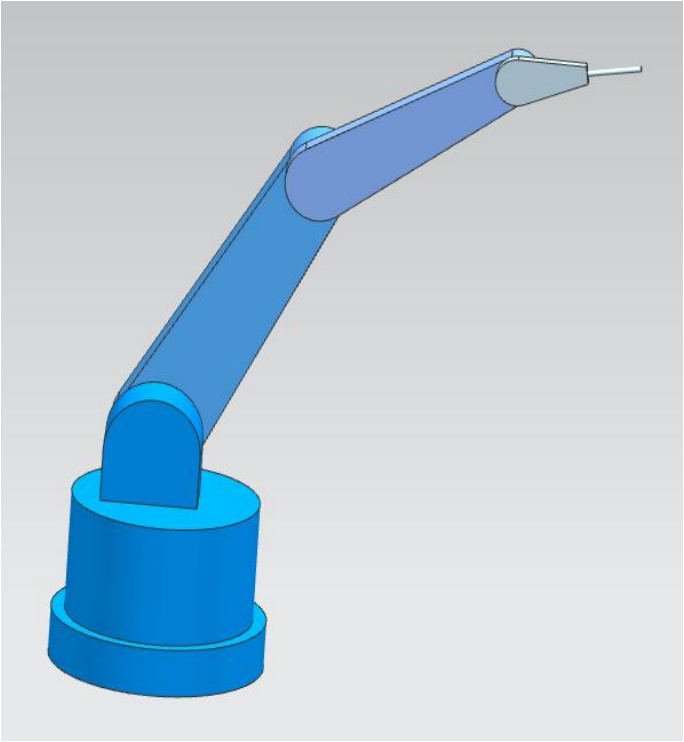
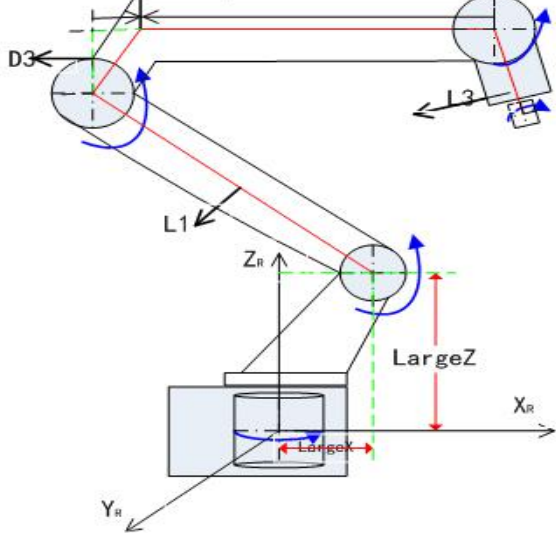
类型		
Frame11	图纸	
	信息	3 个轴

<p>Frame17</p>	<p>区别 图纸</p>	
	<p>信息</p>	<p>5 个轴</p>
<p>Frame18</p>	<p>区别 图纸</p>	
	<p>信息</p>	<p>4 个轴</p>
<p>Frame19</p>	<p>区别 图纸</p>	
	<p>信息</p>	<p>5 个轴</p>
	<p>区别</p>	<p>与 frame17 比较，旋转台 2 相对 xy 平面，支持旋转台任意角度偏转。</p>

直角坐标系类

类型		
Frame100	图纸	
	信息区别	5 个轴。
Frame104	图纸	
	信息区别	5 个轴。 与 frame100 比较，末端旋转轴与末端摆动轴使用虚拟轴进行控制。

类型		
Frame3	图纸	
	信息	<p>码垛基本类型。</p> <p>4 个轴，末端点 4 个自由度。</p> <p>机械原因，末端旋转轴保持水平。</p>
	区别	
Frame4	图纸	
	信息	<p>机械码垛。</p> <p>4 个轴，末端点 4 个自由度。</p> <p>机械原因，末端旋转轴保持水平。</p>
	区别	<p>与 frame3 相比较，少了大小关节的旋转轴，控制在 xyz 平面运动是通过水平伸缩轴与垂直伸缩轴。</p>

<p>Frame103</p>	<p>图纸</p>	
	<p>信息</p>	<p>4 个轴 大旋转台+scara</p>
	<p>区别</p>	
<p>Frame25</p>	<p>图纸</p>	
	<p>信息</p>	<p>5 个轴。</p>
	<p>区别</p>	<p>比 6 关节少一个中间转动轴,比 frame103 结构多一个末端旋转轴。</p>

类型		
Frame15	图纸	
	信息	CORE XY 结构
	区别	
Frame5	图纸	
	信息	3 个轴，3 个自由度。
	区别	
Frame33	图纸	
	信息	UVW 对位平台

二、机械手控制器型号一览

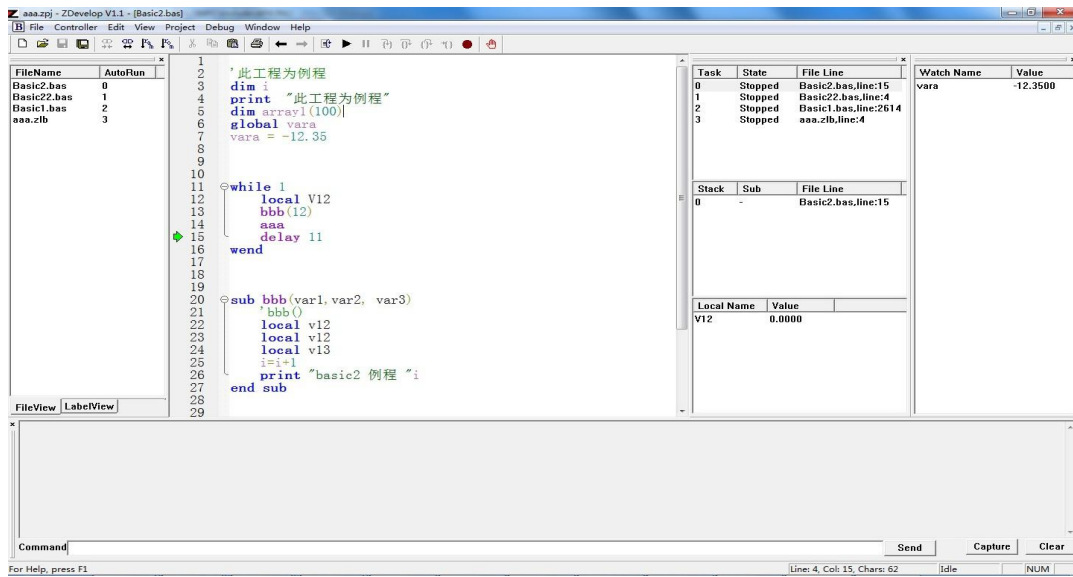
控制器系列	支持机械手功能的控制器型号
ZMC0XX	无
ZMC1XX	ZMC104S / ZMC106S /
ZMC2XX	ZMC204 / ZMC206 / ZMC212 / ZMC212-8
ZMC3XX	ZMC303 / ZMC304 / ZMC306M / ZMC306 / ZMC306X / ZMC308 / ZMC316
ZMC4XX	ZMC406 / ZMC412 / ZMC432 / ZMC464 / ZMC432N
ECI1000 系列	无
ECI2000 系列	ECI2608
ECI3000 系列	ECI3608 / ECI3808

特殊的机械手类型	
机械手类型	支持的控制器型号
基本类型	支持机械手功能的控制器
R 特殊类型	支持机械手功能的控制器，且带后缀—R
	ZMC3XX ZMC303-R/ZMC304-R/ZMC306M-R/ ZMC306-R/ZMC306X-R/ZMC308-R/ ZMC316-R
	ZMC4XX（推荐） ZMC406-R/ZMC412-R/ZMC432-R/ ZMC464-R / ZMC432N-R

第三章 软件工具使用

一、Zdevelop 程序开发环境

编写和调试 ZBASIC 程序需要 ZDevelop 开发软件，ZMC 运动控制器（或仿真器）。Zdevelop 软件的路径“ZMC 光盘资料\5.编程软件\zdevelop2.70”，使用方法可以参看“ZMC 光盘资料\12.视频资料”中的网址获取。



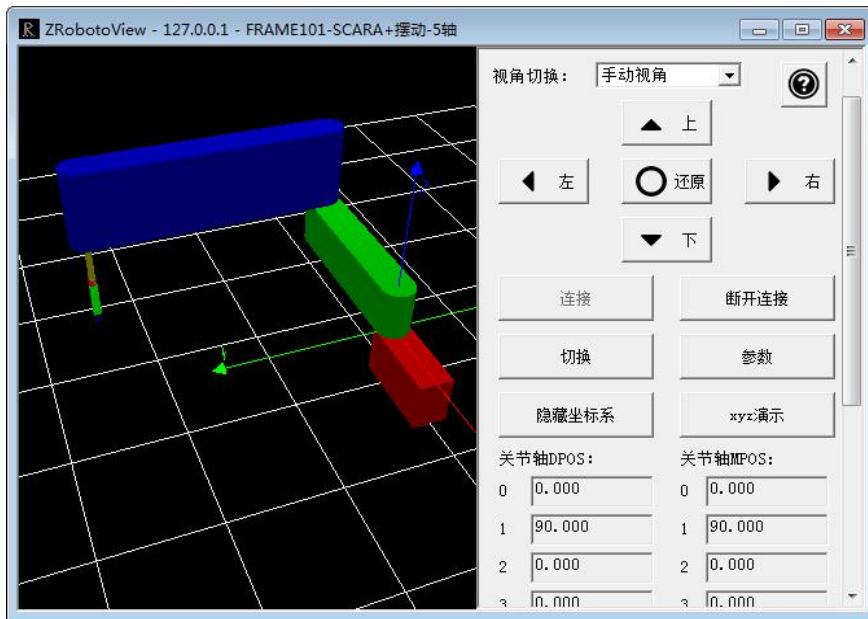
同时，在“ZMC 光盘资料\7.控制器例程\入门程序\机械手程序\机械手基本类型建立程序”中提供了所有机械手类型的基本测试程序，可直接打开 zpj 项目后选择对应 bas 文件设置自动运行使用。



二、ZRobotView 机械手模型仿真工具

ZRobotView 主要是用来模拟机械手的实际运动轨迹，检查机械手程序及实际机械手在运行过程中的速度，轨迹，姿态位置等，防止事故的发生。

工具软件的路径“ZMC 光盘资料\6.工具软件\ZRobotView 机械手仿真工具\ZRobotView.exe”。



三、示波器参数抓取使用

示波器可用于抓取运动中轴的速度，坐标等参数曲线，用以判断运动过程的轨迹准确性，机台抖动大小等性能。在 zdevelop 软件-视图-示波器打开。



四、PC 上位机使用

PC 上位机使用时，根据不同的开发环境，在路径“ZMC 光盘资料\8.PC 编程相关”中选择对应的 dll 链接库及相关文件。然后使用 ZAux_Direct_Connframe 和 ZAux_Direct_Connreframe 指令建立连接。

第四章 机械手相关指令使用

CONFRAME--建立机械手逆解模式

类型	同步运动指令，建立逆解连接
描述	<p>将当前关节坐标系的目标位置与虚拟坐标系的位置关联。 CLUTCH_RATE=0 时关节坐标系的运动速度和加速度受关节轴 SPEED 等参数的限制。</p> <p>当关节轴告警等出错时，此运动会被 CANCEL。</p> <p>不要在虚拟轴高速运动中 CANCEL 此运动，会引起轴在高速运动中停止。</p> <p>此命令 LOAD(加载)时会自动修改虚拟轴的坐标，使得与关节轴坐标正确对应，因此调用后需要 WAIT LOADED 后才开始运动。</p> <p>连接过程中不要修改虚拟轴的坐标，或是调用 DATUM 等可能改坐标的运动，这样会导致关节轴快速运动到新的虚拟位置。</p> <p>CONNFRAME 生效后，关节轴的 MTYPE 为 33，此时无法直接运动关节轴，必须通过运行虚拟轴来间接运动关节轴，当要直接移动关节轴时，先调用 CANCEL 取消 CONNFRAME，再运动关节轴。</p> <p>当虚拟轴和实际轴都是旋转轴时，例如末端旋转轴，虚拟轴和实际轴的脉冲当量注意要一致。</p>
语法	<p>CONNFRAME (frame, tablenum, viraxis0, viraxis1,[...])</p> <p>参数：</p> <p>Frame: 坐标系类型； 1- scara。</p> <p>Tablenum: 机械手参数的 TABLE 起始位置，具体填写哪些参数根据类型确认。</p> <p>viraxis0: 虚拟坐标系第一个轴。</p> <p>viraxis1: 虚拟坐标系第二个轴。</p> <p>[...] : 虚拟坐标系第 N 个轴，可以是实际轴，具体由机械手类型确定。</p>
适用控制器	2 系列以上控制器

例程	<p>L1=10 '第一个关节轴长度 L2=10 '第二个关节轴长度 BASE(0,1) '假设关节轴号是 0/1 UNITS=10,10 '这里以度为单位 DPOS=0,0 '设置关节轴的位置，此处要根据实际情况来修改 BASE(6,7) ATYPE=0,0 '设置为虚拟轴 TABLE(0,L1,L2,3600,3600) '参数存储在 TABLE0 开始的位置，电 '机一圈 360 个脉冲 UNITS=1000,1000 '此脉冲当量要提前设置，中途不能变化 BASE(0,1) '选择关节轴号 CONNFRAME(1,0,6,7) '第 6/7 轴作为虚拟的 XY 轴，启动逆解 '连接 WAIT LOADED '等待运动加载 '第三步：运动虚拟轴 BASE(6,7) MOVEABS(90,0) '虚拟轴的坐标变为 90,0</p>
相关指令	CONNREFRAME

CONNREFRAME--建立机械手正解模式

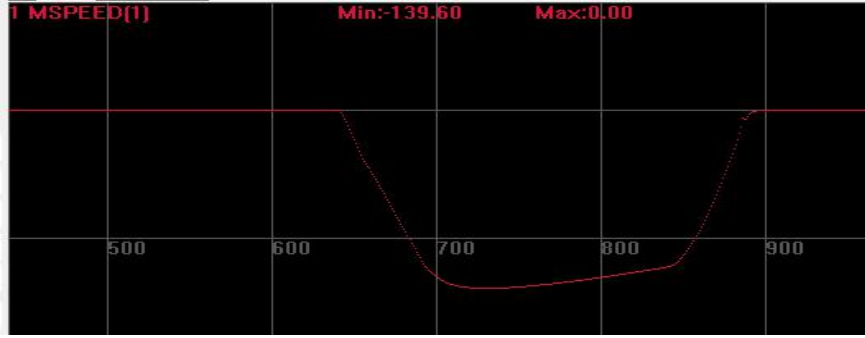

类型	同步运动指令，建立正解连接
描述	<p>将虚拟轴的坐标与关节轴的坐标关联，关节轴运动后，虚拟轴自动走到相应的位置。 此指令为 CONNFRAME 的反运动指令。</p> <p>当虚拟轴 CONNREFRAME 运动 LOAD 时，关节轴的 CONNFRAME 运动会自动被 CANCEL。</p> <p>当关节轴的 CONNFRAME 运动 LOAD 时，虚拟轴 CONNREFRAME 运动会自动被 CANCEL。</p>
语法	<p>CONNREFRAME (frame, tablenum, axis0, axis1,...)</p> <p>参数： 参见 CONNFRAME 的参数，BASE 轴与参数里面的轴的位置相反，当有参考轴时，参考轴列表要加在参数后面(FRAME100 等)。</p>
适用控制器	2 系列以上控制器
例程	<p>L1=10 '第一个关节轴长度 L2=10 '第二个关节轴长度</p>

	<p>BASE(0,1) '假设关节轴号是 0/1 UNITS=10,10 '这里以度为单位 DPOS=0,0 '设置关节轴的位置，此处要根据实际情况来修改 BASE(6,7) ATYPE=0,0 '设置为虚拟轴 TABLE(0,L1,L2,3600,3600) '参数存储在 TABLE0 开始的位置， '电机一圈 360 个脉冲 UNITS=1000,1000 '此脉冲当量要提前设置，中途不能变化 CONNREFRAME(1,0,0,1) '6/7 的坐标根据 0/1 来计算 WAIT LOADED '等待运动加载 '第三步：运动关节轴 BASE(0,1) MOVEABS(90,0) '虚拟轴的坐标变为 0,20</p>
相关指令	CONFRAME

CORNER_MODE--运动参数设置

类型	轴参数																																	
描述	拐角减速等模式设置。																																	
语法	<p>CORNER_MODE = mode mode: 不同的位代表不同的意义，位可以同时使用。</p> <table border="1"> <thead> <tr> <th>位</th> <th>值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> <td>预留</td> </tr> <tr> <td>1</td> <td>2</td> <td>自动拐角减速。 具体说明查看 Zbasic 帮助。</td> </tr> <tr> <td>2</td> <td>4</td> <td>预留</td> </tr> <tr> <td>3</td> <td>8</td> <td>自动小圆限速，半径小于设置值时限速，大于限制值时不限速。 具体说明查看 Zbasic 帮助。</td> </tr> <tr> <td>4</td> <td>16</td> <td>预留</td> </tr> <tr> <td>5</td> <td>32</td> <td>自动倒角设置。 具体说明查看 Zbasic 帮助。</td> </tr> <tr> <td>6</td> <td>64</td> <td>插补使用每个轴的速度限。 MOVER 指令自动使用，其它指令预留。</td> </tr> <tr> <td>7</td> <td>128</td> <td>预留</td> </tr> <tr> <td>8</td> <td>256</td> <td>MOVER 指令运动使用 SP 功能。</td> </tr> <tr> <td>9</td> <td>512</td> <td>MOVER 指令运动使用虚拟轴的速度来限制。</td> </tr> </tbody> </table>	位	值	描述	0	1	预留	1	2	自动拐角减速。 具体说明查看 Zbasic 帮助。	2	4	预留	3	8	自动小圆限速，半径小于设置值时限速，大于限制值时不限速。 具体说明查看 Zbasic 帮助。	4	16	预留	5	32	自动倒角设置。 具体说明查看 Zbasic 帮助。	6	64	插补使用每个轴的速度限。 MOVER 指令自动使用，其它指令预留。	7	128	预留	8	256	MOVER 指令运动使用 SP 功能。	9	512	MOVER 指令运动使用虚拟轴的速度来限制。
位	值	描述																																
0	1	预留																																
1	2	自动拐角减速。 具体说明查看 Zbasic 帮助。																																
2	4	预留																																
3	8	自动小圆限速，半径小于设置值时限速，大于限制值时不限速。 具体说明查看 Zbasic 帮助。																																
4	16	预留																																
5	32	自动倒角设置。 具体说明查看 Zbasic 帮助。																																
6	64	插补使用每个轴的速度限。 MOVER 指令自动使用，其它指令预留。																																
7	128	预留																																
8	256	MOVER 指令运动使用 SP 功能。																																
9	512	MOVER 指令运动使用虚拟轴的速度来限制。																																
适用控制器	通用																																	

CLUTCH_RATE--连接速度设置

类型	轴参数
描述	<p>connect 连接的速度，ratio/秒，默认值 1000000。</p> <p>当设置为 0 时，根据轴的速度/加速度参数来跟踪连接，比较适合手轮运动。</p> <p>当速度不够高时可能导致机械手要持续运动一段时间才能结束，此时运动的最终位置是准确的，但轨迹是变形的。</p>
语法	clutch_rate= value
适用控制器	通用
例程	<p>clutch_rate = 0 ‘根据轴的速度/加速度参数来跟踪连接建立机械手连接后，相同的速度加速度参数下：设为 0（平滑模式），速度曲线如下</p>  <p>设为非 0（强制模式），速度曲线如下</p>  <p>对关节轴有用，当配置为 0 时会使用关节轴的速度加速度做限制（平滑模式），这样运行更平滑，步进电机也可以不丢步。配置为非 0 时使用虚拟轴的速度加速度做限制（强制模式），运动轨迹更精确，但易抖动。</p>

FRAME_STATUS--查询机械手当前姿态

类型	轴状态
描述	<p>指明当前的机械手姿态。 不是机械手状态时返回-1， FRAME_TRANS2 指令会用到这个姿态。</p> <p>只对 SCARA、类 SCARA 结构和 6 自由度结构有多种姿态。 SCARA 左手姿态值为 0，右手姿态值为 1。</p>
语法	VAR1=FRAME_STATUS (AXIS)
适用控制器	通用
例程	<p>在线命令输入?frame_status，打印出当前姿态。</p> <p>>>?frame_status</p>

FRAME_TRANS2--正逆解坐标转换

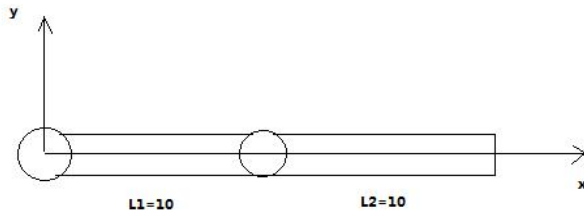
类型	机械手计算指令
描述	<p>坐标转换函数。 使用时必须 base 正确的轴号。逆解时，base 虚拟轴；正解时，base 关节轴。 按照正确的顺序填入相应个数的数据。填入的个数与数据，与? *frame 一致。</p>
语法	<p>FRAME_TRANS2(tablein, tableout, dir)</p> <p>tablein: Table 数组中的索引号，从这个索引开始连续存储数据，正解时输入关节坐标，逆解时输入虚拟轴坐标列表，最后附加姿态。</p> <p>Tableout: Table 这个索引开始存储数据，正解时输出虚拟坐标，最后附加姿态，逆解时输出关节坐标列表。</p> <p>Dir: 0- 逆解，虚拟轴到关节轴，无姿态，自动使用当前的姿态。 1- 正解， 关节轴到虚拟轴，无姿态输出。 2- 逆解， 输入虚拟轴坐标，最后加上姿态。 3- 正解， 输出虚拟轴坐标，输出最后一个位置填姿态。</p>

适用控制器	通用
-------	----

以 scara 结构为例。

第一关节轴 L1=10，第二关节轴 L2=10。Table(100)作为输入坐标的存放位置，table(200)作为输出坐标的存放位置。

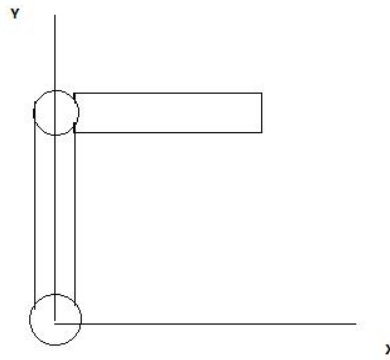
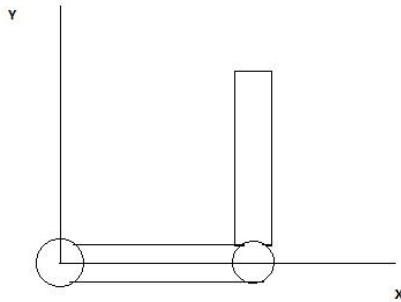
建立连接后，关节轴原点坐标 (0,0)，此时虚拟轴坐标为 (20,0)，如下图。



虚拟轴坐标 (10,10) 时，有两种姿态

关节轴 (0,90)

关节轴 (90,-90)



坐标转换表

状态	Base 轴	输入 X、Y、姿态	使用指令模式	输出关节坐标
逆解	虚拟轴	Table(100,20,0,0)	FRAME_TRANS2(100,200,0)	Table(200,0,0)
		Table(100,10,10,1)		Table(200,0,90)
		Table(100,10,10,1)	FRAME_TRANS2(100,200,2)	Table(200,90,-90)
		输入关节坐标	使用指令模式	输出 X、Y、姿态
		Table(100,0,0)	FRAME_TRANS2(100,200,1)	Table(200,20,0,0)
		Table(100,0,90)		Table(200,10,10,0)
		Table(100,90,-90)		Table(200,10,10,0)
		Table(100,90,-90)	FRAME_TRANS2(100,200,3)	Table(200,10,10,1)

FRAME_ROTATE--工件坐标系变换

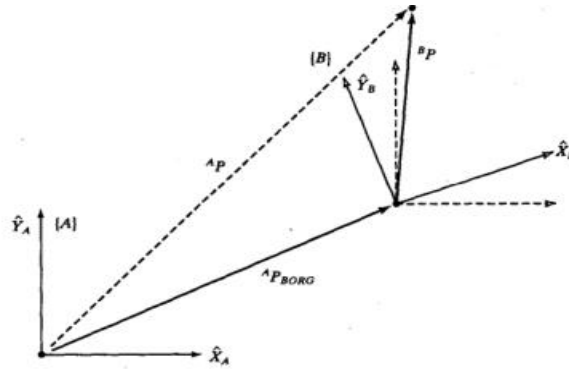
类型	机械手计算指令
描述	<p>用于对工件坐标系进行平移和旋转。</p> <p>目前只对 FRAME6 同时旋转姿态，其他具有 XYZ 虚拟轴的可以支持 XYZ3 个轴旋转，姿态轴不能旋转。</p> <p>旋转后，虚拟轴 WORLD_DPOS 表示世界坐标不会改变，虚拟轴 DPOS 表示工件坐标会改变。</p> <p>使用时需控制器当前存在机械链接。</p> <p>base 轴，可以是虚拟轴与关节轴任意一个；base 轴无机械手链接，会报 1025 的错误。</p> <p>多种机械手叠加的情况，根据 base 轴来识别是哪个机械手模式；如果 base (axis_1,axis_2) 中的 axis_1 是模式 1 的机械手轴，axis_2 是模式 2 的机械手轴，那么结果是模式 1 的机械手进行坐标计算，即以 base 轴的顺序来计算。</p>
语法	<p>FRAME_ROTATE(X,Y,Z,RX,RY,RZ)</p> <p>参数描述：</p> <p>X: 坐标系 B 沿，X^{\wedge} 的平移距离。</p> <p>Y: 坐标系 B 沿，Y^{\wedge} 的平移距离。</p> <p>Z: 坐标系 B 沿，Z^{\wedge} 的平移距离。</p> <p>RX:坐标系 B 沿，X^{\wedge} 的旋转的角度。</p> <p>RY:坐标系 B 沿，Y^{\wedge} 的旋转的角度。</p> <p>RZ:坐标系 B 沿，Z^{\wedge} 的旋转的角度。</p> <div data-bbox="475 1451 1114 1859" data-label="Diagram"> </div> <p>坐标系旋转： 旋转的方法是：x_y_z 固定角坐标系。</p>

	<p>首先将做坐标系{B}和一个已知参考坐标系{A}重合。先将{A}绕 X^a 旋转 RX 角，在绕 Y^a 旋转 RY 角，最后绕 Z^a 旋转 RZ 角。</p>
适用控制器	通用
例程	<p>以 frame=2, detla 为例；对其绕 x 轴旋转 90 度。</p> <pre> base(0,1,2) rapidstop atype = 1,1,1 units=3600/360,3600/360,3600/360 dpos=0,0,0 base(6,7,8) atype = 0,0,0 TABLE(0,40,10,32,85,3600,3600,3600, 0, 0, 0) UNITS = 100,100,100 BASE(0,1,2) CONNFRAME(2,0,6,7,8) WAIT LOADED base(6,7,8) FRAME_ROTATE(0,0,0,pi/2,0,0) ?"dpos(7)-world_dpos(7)=",dpos(7)-world_dpos(7) ?"dpos(8)-world_dpos(8)=",dpos(8)-world_dpos(8) 输出行输出>> dpos(7)-world_dpos(7)= -58.1400 dpos(8)-world_dpos(8)= 58.1400 </pre> <p>以 f=1, scara 为例；对其相对于 z 轴旋转 90 度。</p> <pre> base(0,1,2,3) rapidstop atype = 1,1,1,1 units=3600/360,3600/360,3600/360,1000 dpos=0,0,0,0 base(6,7,8,9) atype = 0,0,0,0 TABLE(0,100,100,3600,3600,3600) UNITS = 100,100,3600/360,1000 BASE(0,1,2,3) CONNFRAME(1,0,6,7,8,9) WAIT LOADED base(6,7,8,9) FRAME_ROTATE(0,0,0,0,0,pi/2) ?"dpos(7)-world_dpos(7)=",dpos(7)-world_dpos(7) </pre>

	<pre>?"dpos(8)-world_dpos(8)=",dpos(8)-world_dpos(8)</pre> <p>输出行输出>></p> <pre>dpos(7)-world_dpos(7)= -200 dpos(8)-world_dpos(8)= 0</pre>
相关指令	FRAME_ROTATE2

FRAME_ROTATE2--坐标系变换计算

类型	机械手计算指令
描述	<p>手动计算坐标系旋转后的坐标值。 使用时需控制器当前存在机械链接。</p> <p>base 轴，可以是虚拟轴与关节轴任意一个；base 轴无机械手链接，会报 1025 的错误。</p> <p>多种机械手叠加的情况，根据 base 轴来识别是哪个机械手模式。如果 base (axis_1,axis_2) 中的 axis_1 是模式 1 的机械手轴，axis_2 是模式 2 的机械手轴，那么结果是模式 1 的机械手进行坐标计算，即以 base 轴的顺序来计算。</p>
语法	<pre>FRAME_ROTATE2(tablein, tableout, dir[, x,y,z[, rx,ry,rz]])</pre> <pre>Ret = FRAME_ROTATE2(tablein, tableout, dir[, x,y,z[, rx,ry,rz]])</pre> <p>参数描述：</p> <p>Tablein: 转换前，填写的坐标存储 TABLE 位置</p> <p>Tableout: 转换后，输出的坐标存储 TABLE 位置</p> <p>Dir: 0- DPOS 到 WORLD_DPOS 1- WORLD_DPOS 到 DPOS</p> <p>X: 坐标系 B 沿，X[^]的平移距离</p> <p>Y: 坐标系 B 沿，Y[^]的平移距离</p> <p>Z: 坐标系 B 沿，Z[^]的平移距离</p> <p>RX:坐标系 B 沿，X[^]的旋转的角度</p> <p>RY:坐标系 B 沿，Y[^]的旋转的角度</p> <p>RZ:坐标系 B 沿，Z[^]的旋转的角度</p> <p>[, x,y,z[, rx,ry,rz]]不填时使用当前的缺省 rotate 参数</p> <p>Ret: 返回成功与否. -1: 成功; 0: 未成功</p>



适用控制器

通用

例程

以 f=2, delta 为例；对其绕 x 轴旋转 90 度。

```

base(0,1,2)
rapidstop
atype = 1,1,1
units=3600/360,3600/360,3600/360
dpos=0,0,0
base(6,7,8)
atype = 0,0,0
TABLE(0,40,10,32,85,3600,3600,3600, 0, 0, 0 )
UNITS = 100,100,100
BASE(0,1,2)
CONNFRAME(2,0,6,7,8)
WAIT LOADED
for i=0 to 2
table(100+i)=dpos(i+6)
next
base(6,7,8)
FRAME_ROTATE(0,0,0,pi/2,0,0)
base(6,7,8)
FRAME_ROTATE(0,0,0,pi/2,0,0)
wait loaded
ret=frame_rotate2(100,200,1,0,0,0,pi/2,0,0)
if ret=-1 then
?"计算值"
?"dpos(6)=",table(200)
?"dpos(7)=",table(201)
?"dpos(8)=",table(202)
?"比较值"
?"dpos(6)比较",table(200)-dpos(6)
?"dpos(7)比较",table(201)-dpos(7)
?"dpos(8)比较",table(202)-dpos(8)

```

```
Endif
输出行输出>>
计算值
dpos(6)= 0
dpos(7)= -58.1400
dpos(8)= 0.0000
比较值
dpos(6)比较 0
dpos(7)比较 0
dpos(8)比较 0.0000

以 f=1, scara 为例；对其相对于 z 轴旋转 90 度。
base(0,1,2,3)
rapidstop
atype = 1,1,1,1
units=3600/360,3600/360,3600/360,1000
dpos=0,0,0,0
base(6,7,8,9)
atype = 0,0,0,0
TABLE(0,100,100,3600,3600,3600)
UNITS = 100,100,3600/360,1000
BASE(0,1,2,3)
CONNFRAME(1,0,6,7,8,9)
WAIT LOADED
for i=0 to 3
table(100+i)=dpos(i+6)
next
base(6,7,8,9)
FRAME_ROTATE(0,0,0,0,0,pi/2)
wait loaded
ret=frame_rotate2(100,200,1,0,0,0,0,0,pi/2)
if ret=-1 then
    ?"计算值"
    ?"dpos(6)=",table(200)
    ?"dpos(7)=",table(201)
    ?"dpos(8)=",table(202)
    ?"dpos(9)=",table(203)
    ?"比较值"
    ?"dpos(6)比较",table(200)-dpos(6)
    ?"dpos(7)比较",table(201)-dpos(7)
    ?"dpos(8)比较",table(202)-dpos(8)
    ?"dpos(9)比较",table(203)-dpos(9)
Endif
输出行输出>>
```


	计算值 dpos(6)= -0.0000 dpos(7)= -200 dpos(8)= 0 dpos(9)= 0 比较值 dpos(6)比较 -0.0000 dpos(7)比较 0 dpos(8)比较 0
相关指令	FRAME_ROTATE

WORLD_DPOS--世界坐标系

类型	轴状态
描述	虚拟轴参照世界坐标系的坐标值，没有旋转时与 DPOS 相同。
语法	var1=world_dpos(axis)
适用控制器	通用
例程	在线命令打印 >>?*world_dpos

UNTIS--脉冲当量

类型	轴参数
描述	<p>脉冲当量，指定每单位发送的脉冲数，支持 5 位小数精度。</p> <p>一个机械手的所有虚拟轴和关节轴的长度单位要求统一，一般都是 mm 单位，虚拟轴的一个 mm 的脉冲数一般建议设置为 1000。</p> <p>旋转轴单位一般以度为单位，此单位没有要求（特殊机械手可能要求虚拟旋转轴与对应的关节旋转轴的 units 一样）。</p>
语法	可读：VAR1 = UNITS 或 VR1=UNITS(轴号) 可写：UNITS= expression 或 UNITS(轴号)= expression
适用控制器	通用
例程	查看 Zbasic 帮助

MOVER_L/MOVER_LABS--关节轴直线插补

类型	运动指令
描述	<p>关节轴直线插补。</p> <p>机械手关节插补运动，机械手末端直线运动到指定坐标。</p> <p>此指令正解模式下使用，直接操作关节轴时会有改变姿态的可能，所以要确保起点结束点姿态要保持一致，否则会报错。</p>
语法	<p>MOVER_L(distance1 [,distance2 [,distance3 [,distance4...]]])</p> <p>参数：</p> <p>distance1 第一个轴运动距离</p> <p>distance2 下一个轴运动距离</p>
适用控制器	4 系列 20170511 以上固件支持
例程	<pre> BASE(0,1) dpos=0,0 base(6,7) atype = 0,0 '设置为虚拟轴 units=1000,1000 TABLE(0,L1,L2, 100*360, 100*360, 360) CONNREFRAME(1,0,0,1) '第 6/7 轴作为虚拟的 XY 轴，启动连接 WAIT LOADED '关节运动 Base(0,1) speed=400 SRAMP=100 ACCEL=1000 DECEL=1000 MERGE = 1 CORNER_MODE=32 '启动倒角 ZSMOOTH=2 moveabs(45,90) '关节运动，此时运动关节角度 MOVER_LABS(90, 0) '末端直线运动 Wait idle '等待运动停止 Print *dpos </pre>
相关指令	MOVER_C ， MOVER_C3

MOVER_C / MOVER_CABS--关节轴平面圆弧

类型	运动指令										
描述	<p>关节轴直接走圆弧插补运动。</p> <p>此指令正解模式下使用。 BASE 虚拟的 XYZ 轴，否则无法确定 XYZ，此时的参数也是虚拟轴的距离。</p>										
语法	<p>参数：</p> <p>end1 第 1 个轴运动距离参数 1 end2 第 2 个轴运动距离参数 1 centre1 第 1 个轴运动距离参数 2 centre2 第 2 个轴运动距离参数 2 mode</p> <table border="1"> <thead> <tr> <th>值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>当前点，中间点，终点三点定圆弧。 距离参数 1 为终点距离，距离参数 2 为中间点距离。</td> </tr> <tr> <td>1</td> <td>当前点，圆心，终点定圆弧。 走最短的圆弧。 距离参数 1 为终点距离，距离参数 2 为圆心的距离。</td> </tr> <tr> <td>2</td> <td>当前点，中间点，终点三点定圆。 距离参数 1 为终点距离，距离参数 2 为中间点距离。</td> </tr> <tr> <td>3</td> <td>当前点，圆心，终点定圆。 先走最短的圆弧，再继续走完整圆。 距离参数 1 为终点距离，距离参数 2 为圆心的距离。</td> </tr> </tbody> </table> <p>dis1- disn 螺旋轴的距离。</p>	值	描述	0	当前点，中间点，终点三点定圆弧。 距离参数 1 为终点距离，距离参数 2 为中间点距离。	1	当前点，圆心，终点定圆弧。 走最短的圆弧。 距离参数 1 为终点距离，距离参数 2 为圆心的距离。	2	当前点，中间点，终点三点定圆。 距离参数 1 为终点距离，距离参数 2 为中间点距离。	3	当前点，圆心，终点定圆。 先走最短的圆弧，再继续走完整圆。 距离参数 1 为终点距离，距离参数 2 为圆心的距离。
值	描述										
0	当前点，中间点，终点三点定圆弧。 距离参数 1 为终点距离，距离参数 2 为中间点距离。										
1	当前点，圆心，终点定圆弧。 走最短的圆弧。 距离参数 1 为终点距离，距离参数 2 为圆心的距离。										
2	当前点，中间点，终点三点定圆。 距离参数 1 为终点距离，距离参数 2 为中间点距离。										
3	当前点，圆心，终点定圆。 先走最短的圆弧，再继续走完整圆。 距离参数 1 为终点距离，距离参数 2 为圆心的距离。										
适用控制器	4 系列 20170511 以上固件支持										
例程	<pre>L1 = 500 L2 = 500 TABLE(0,L1,L2, 100*360, 100*360, 360) '参数存储在 TABLE0 开始的位置，电机一圈 360 个脉冲。 base(6,7) CONNREFRAME(1,0,0,1) '第 6/7 轴作为虚拟的 XY 轴，启动连接。 WAIT LOADED '等待运动加载 BASE(6,7) 'REFRAME 直接 MOVER 运动虚拟轴，会自动转换到关节轴上。 MOVER_LABS(500)</pre>										

	MOVER_C(500,0, 250,250, 0)
相关指令	MOVER_L , MOVER_C3

MOVER_C3 / MOVER_C3ABS--关节轴空间圆弧

类型	运动指令										
描述	<p>关节轴直接走空间圆弧插补运动。</p> <p>此指令正解模式下使用。 BASE 虚拟的 XYZ 轴，否则无法确定 XYZ，此时的参数也是虚拟轴的距离。</p>										
语法	<p>MOVER_C3 (endx, endy, endz, midx, midy, midz, mode[, dis1][,dis2][,dis3])</p> <p>参数:</p> <p>end1 第 1 个轴运动距离参数 1 end2 第 2 个轴运动距离参数 1 end3 第 3 个轴运动距离参数 1 centre1 第 1 个轴运动距离参数 2 centre2 第 2 个轴运动距离参数 2 centre3 第 3 个轴运动距离参数 2 mode</p> <table border="1"> <thead> <tr> <th>值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>当前点，中间点，终点三点定圆弧。 距离参数 1 为终点距离，距离参数 2 为中间点距离。</td> </tr> <tr> <td>1</td> <td>当前点，圆心，终点定圆弧。 走最短的圆弧。 距离参数 1 为终点距离，距离参数 2 为圆心的距离。</td> </tr> <tr> <td>2</td> <td>当前点，中间点，终点三点定圆。 距离参数 1 为终点距离，距离参数 2 为中间点距离。</td> </tr> <tr> <td>3</td> <td>当前点，圆心，终点定圆。 先走最短的圆弧，再继续走完整圆。 距离参数 1 为终点距离，距离参数 2 为圆心的距离。</td> </tr> </tbody> </table> <p>dis1- disn 螺旋轴的距离。</p>	值	描述	0	当前点，中间点，终点三点定圆弧。 距离参数 1 为终点距离，距离参数 2 为中间点距离。	1	当前点，圆心，终点定圆弧。 走最短的圆弧。 距离参数 1 为终点距离，距离参数 2 为圆心的距离。	2	当前点，中间点，终点三点定圆。 距离参数 1 为终点距离，距离参数 2 为中间点距离。	3	当前点，圆心，终点定圆。 先走最短的圆弧，再继续走完整圆。 距离参数 1 为终点距离，距离参数 2 为圆心的距离。
值	描述										
0	当前点，中间点，终点三点定圆弧。 距离参数 1 为终点距离，距离参数 2 为中间点距离。										
1	当前点，圆心，终点定圆弧。 走最短的圆弧。 距离参数 1 为终点距离，距离参数 2 为圆心的距离。										
2	当前点，中间点，终点三点定圆。 距离参数 1 为终点距离，距离参数 2 为中间点距离。										
3	当前点，圆心，终点定圆。 先走最短的圆弧，再继续走完整圆。 距离参数 1 为终点距离，距离参数 2 为圆心的距离。										
适用控制器	4 系列 20170511 以上固件支持										
例程	L1 = 500 L2 = 500										

	<p>TABLE(0,L1,L2, 100*360, 100*360, 360) '参数存储在 TABLE0 开始的位置，电机一圈 360 个脉冲。</p> <p>base(6,7)</p> <p>CONNREFRAME(1,0,0,1) '第 6/7 轴作为虚拟的 XY 轴，启动连接。</p> <p>WAIT LOADED '等待运动加载</p> <p>BASE(6,7,8) 'REFRAME 直接 MOVER 运动虚拟轴，会自动转换到关节轴上。</p> <p>MOVER_LABS(400)</p> <p>MOVER_C3ABS(200,0,0,600,400,0, 0)</p>
相关指令	MOVER_L , MOVER_C

FRAME_CAL--参数矫正

类型	机械手计算指令
描述	<p>根据机械手关节示教的坐标和特点自动校正当前的机械手参数。</p> <p>Tablein 中存储的关节轴坐标获取，用当前的零点位置与机械手参数建立机械手链接前提下，控制机械手末端点运动到校正点，取校正点的关节轴坐标。</p> <p>校正当前零点位置与理论零点的偏差，计算出在理论零点的关节轴坐标。</p> <p>校正机械手参数的值（校正部分参数），计算出理论的机械参数的值。</p> <p>FRAME_CAL 只做计算，指令返回值为-1 计算成功，返回 0 计算失败。</p> <p>FRAME_CAL 的 BASE 轴必须是在 FRAME 状态的轴。</p>
语法	<p>FRAME_CAL(tablein,space,groups,tableaux, zeroout, [tableout2])</p> <p>参数：</p> <p>tablein 关节坐标存储的 table 起始编号，每个点的关节坐标按顺序存储，多个点之间间隔 space。</p> <p>space 每两个点之间间隔的 TABLE 元素。</p> <p>Groups 点数。</p> <p>Tableaux 辅助参数的 table 编号，部分 FRAME 需要。</p> <p>Zeroout 计算出来理论零点时关节轴的绝对坐标的 table 编号。</p>

	Tableout2 计算出来的机械手参数存储的位置，不填时直接存储原参数的位置。
适用控制器	通用
例程	查看 机械手参数校正 章节

第五章 机械手相关功能操作

一、工件坐标系转换

工件坐标系是固定于工件上的笛卡尔坐标系，工件在坐标系相对与世界坐标系存在转换。机械手可以拥有若干工件坐标系，或者表示不同的工件，或者表示同一工件在不同的位置。

虚拟轴满足 xyz 三轴的机械手类型支持此功能。

1. 相关指令

Frame_rotate	虚拟轴坐标系旋转。
Frame_rotate2	坐标系旋转的计算指令。
World_dpos	虚拟轴参照世界坐标系的坐标值。

2. 确定工件坐标系

工件坐标系用世界坐标系进行描述，需原点的平移描述与坐标系姿态的描述。原点的平移描述用工件坐标系的原点在世界坐标系的坐标 (X,Y,Z) 表示。坐标系姿态描述，使用 X_Y_Z 固定角坐标系。

X_Y_Z 固定角坐标系

首先将做坐标系 {B} 和一个已知参考坐标系 {A} 重合。先将 {B} 绕 {A} 的 X+ 旋转 RX 角，在绕 {A} 的 Y+ 旋转 RY 角，最后绕 {A} 的 Z+ 旋转 RZ 角。每个旋转都是绕着固定参考坐标系 {A} 的轴。

每个旋转都是绕着固定参考轴坐标系 {A} 的轴。我们规定这种坐标系的旋转方法为 X-Y-Z 固定角坐标系，也是控制器采用的旋转方法。

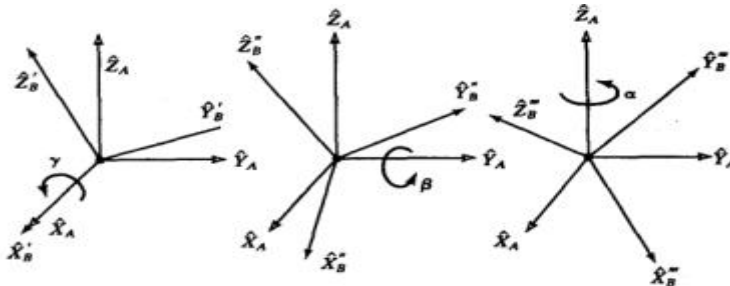


图 3.1 X_Y_Z 固定角坐标系，按照 RX,RY,RZ 的顺序进行旋转

3. 控制器坐标系旋转实现

```
Base(axisx,axisy,axisz)
frame_rotate(x,y,z,rx,ry,rz)
```

(axisx,axisy,axisz) ，填写虚拟轴 xyz 轴。

(x,y,z,rx,ry,rz)，按照计算方法填写正确的参数。

执行后，控制虚拟轴的 xyz 轴，对应的是工件坐标系的 xyz 轴。

需读取点在世界坐标系的位置，可读取 world_dpos 的值。

4. 坐标系转换完成

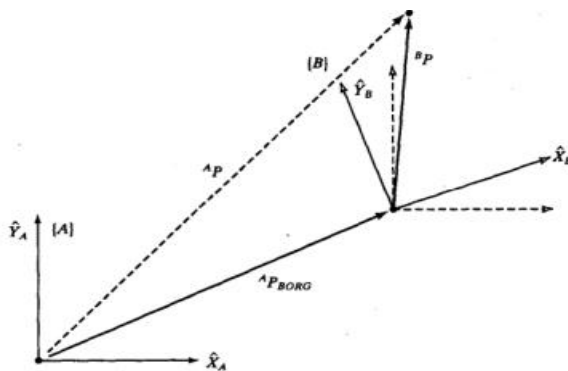


图 3.2 经平移与旋转的坐标系 B

假设 {A} 世界坐标系，{B} 是工件坐标系，{B} 用 {A} 表示为 (X1, Y1, Z1, RX1, RY1, RZ1)

进行坐标系转换前：

末端工作点向虚拟轴 X Y Z 方向移动，沿坐标系 {A} 的 X 方向，Y 方向，Z 方向运动。虚拟轴的 dpos 表示的是在 {A} 的位置。WORLD_DPOS 表示的是在 {A} 的位置。

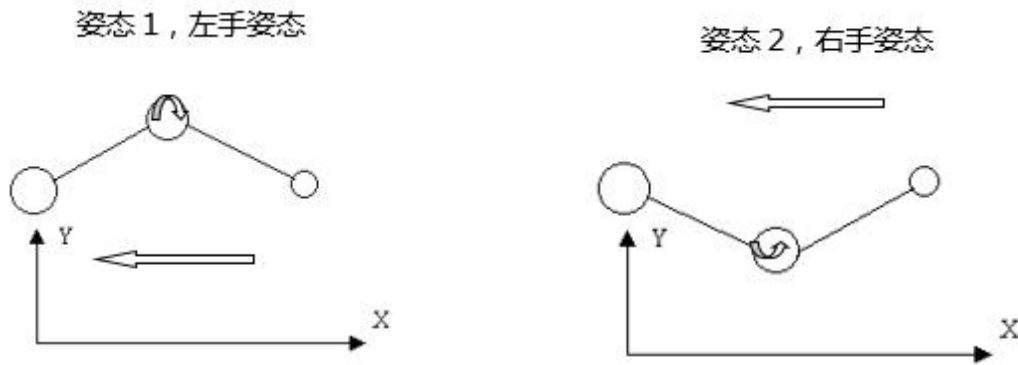
进行坐标系转换后：

移动虚拟轴 X Y Z，末端工作点是沿坐标系 {B} 的 X 方向，Y 方向，Z 方向运动。虚拟轴的 dpos 表示的是在 {B} 的位置。WORLD_DPOS 表示的是在 {A} 的位置。

二、姿态调整

在逆解模式下运行时，由于机械结构的问题，会产生无法运动到某一位置或刚体干涉的问题，此时就需要进行姿态的调整。

机械手姿态只能在正解模式下操作关节轴进行调整，如 FRAME1 类型 scara 机械手，可按下图操作关节轴调整。



三、奇异点规避

机械手结构建立时，需要将机械手先运动到给定的零点位置，某些结构类型的零点位置就是奇异点位置，如 scara 机械手。此时如果直接进行逆解操作的话会导致机台卡死，乱跑等问题产生。

解决方法为先建立正解模式连接，手动调整位置使其处于某个确定姿态，再切换为逆解模式进行操作。

四、机械手参数校正

由于机械手整体配件在设计和装配时都会有误差，所以如果按照理论值填写参数必然会产生坐标的偏差，此时就需要使用参数校正功能。

参数校正是机械手关节示教的坐标和特点自动校正零点位置与机械参数，部分机械手类型支持。

1. 相关指令

[FRAME_CAL](#)

2. 操作步骤

- 1) 操作各关节轴运动到类型规定零点位置。
- 2) 按理论机械手参数建立机械手连接。
- 3) 控制机械手末端点运动到校正点。
- 4) 记录关节轴坐标并填入 Table 数组。
- 5) 通过 `FRAME_CAL` 指令实现参数校正。
- 6) 校正出实际零点时的关节轴绝对坐标与机械参数，输出到指令指定的 Table 位置。
- 7) `Rapidstop` 指令取消机械手模式。
- 8) 手动运动各关节轴到纠偏输出对应的绝对位置。
- 9) 手动清 0 关节轴 `dpos`。
- 10) 最后使用纠偏输出的 Table 参数重新建立机械手。

3. 各类型校正参数及过程

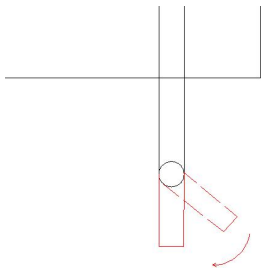
FRAME1scara 及其他变种

校正内容

零点时，小关节轴相对大关节轴的角度偏差。

校正要求

- 1) J1, J2 的长度是正确的。
- 2) 如果有末端旋转摆动，示教时必须保证末端在正下方，如下图。



校正方法

根据 `FRAME_CAL` 指令的 `tableaux` 参数填写值，有两种情况：

- 1) `Tableaux` 对应的 `TABLE` 填入 0，表示左右手法。左右两个关节姿态下运行到世界坐标系中同一个位置，分别记录左手姿态时的关节轴坐标和右手姿态时的关节轴坐标。
- 2) `Tableaux` 对应的 `TABLE` 填入不为 0，表示单手法，此时必须示教世界坐标系中的两个位置点，记录两点对应的关节坐标（同一姿态）和两点直线距离。

此时 Tableaux 填写值为两点直线距离。必须两个关节坐标都有变动，建议沿 X 示教固定长度的两点。

校正前填写 table 数据

点 1 的关节轴坐标填入与 FRAME_CAL 指令参数 tablein 一致的 table。

点 2 的关节轴坐标填入 tablein+space 的 table。

根据选择的校正方式，辅助点 Tableaux 填写 0 或者非 0，非 0 值为两点间的距离。

校正后输出 table 数据

校正后理论零点时关节轴的绝对坐标存储在指令参数 Zerooutz 指定的 table。

校正后实际的机械手参数存储在指令参数 Tableout2 指定的 table。

参考例程

使用单手法校正参数。

```

global const axis_x = 0
global const axis_y = 1

global const axis_virx = 6
global const axis_viry = 7

global L1
global L2
global RET

base(axis_x,axis_y)
atype = 1,1
units = 100,100
dpos=0,0

L1 = 300
L2 = 300

base(axis_virx,axis_viry)
atype = 0,0,0          ' 设置为虚拟轴

TABLE(0,L1,L2,36000,36000,36000)      ' 参数存储在 TABLE0 开始的位置，电机一圈 360 个脉冲。

UNITS(6)=1000          ' 此脉冲当量要提前设置，中途不能变化
UNITS(7)=1000          ' 此脉冲当量要提前设置，中途不能变化
units(8)=100           ' 虚拟旋转轴与关节轴 R 一致

```

```
base(axis_virx,axis_viry)
CONNREFRAME(1,0,axis_x,axis_y) '启动正解连接。
wait loaded

'假定已经按照校正操作前3步骤,记录到了关节坐标
table(300,30,-70) '左手姿态关节轴位置填入 table
table(305,-30,50) '右手姿态关节轴位置填入 table
table(200) = 0 '辅助点,选择左右手方式校正

RET = FRAME_CAL(300,5,2,200,100,150) '开始校正
'FRAME_CAL(tablein,space,groups,tableaux,zeroout,[tableout2])
'校正的实际0点时关节轴绝对坐标存放在 table100,机械手参数存放在 table150

if not ret then
    print "FRAMECAL failed"
Else
    for i = 0 to 1
        print "table",100+i,"=",table(100 + i)
    next

    for i = 0 to 4
        print "table",150+i,"=",table(150 + i)
    next

    RAPIDSTOP(2) '取消机械手连接
    wait IDLE

    base(axis_X,axis_y)
    moveabs(table(100),table(101)) '移动到校正后的0点位置
    wait IDLE

    dpos=0,0 '关节轴当前位置设为0点

    base(axis_virx,axis_viry)
    CONNREFRAME(1,150,axis_x,axis_y) '用校正后的机械手参数重新建立连接。
    wait loaded

end if
end
```

FRAME11

校正内容

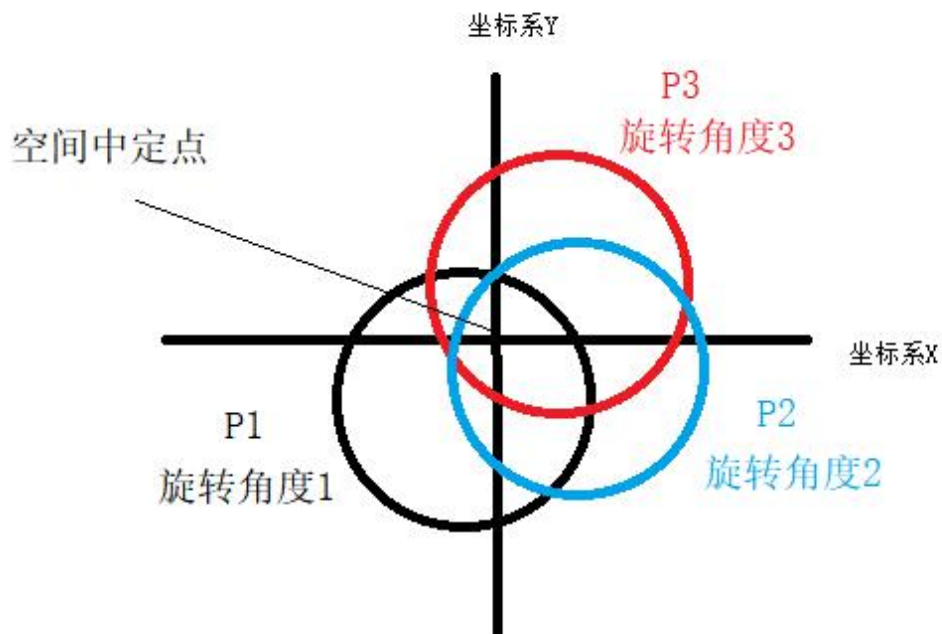
旋转中心实际偏移 CENTERX, CENTERY。

校正要求

- 1) 旋转轴 UNITS 设置为正确的一圈脉冲数。
- 2) 旋转正向与机械手连接设置方向一致。

校正方法

此时 Table (Tableaux) =0, 在世界坐标系中同一个位置, 对应旋转轴不同的角度, 角度范围大一些。至少记录 3 个关节坐标位置, 如下图所示。



校正前填写 table 数据

点 1 关节轴坐标填入编号为 [FRAME_CAL](#) 指令参数 tablein 一致的 table。

点 2 关节轴坐标填入编号为 tablein+space*1 的 table。

点 3 关节轴坐标填入编号为 tablein+space*2 的 table。

编号为 Tableaux 的 table, 填 0。

校正后输出 table 数据

校正后理论零点时关节轴的绝对坐标存储在指令参数 Zerooutz 指定的 table。
校正后实际的机械手参数存储在指令参数 Tableout2 指定的 table。

例程

```

dim axisx,axisy,axisR    '实际关节轴定义
dim axisx,axisy,axisR    '实际关节轴定义
dim viraxisx,viraxisy' 虚拟轴定义
axisx=0
axisy=1    ' 实际关节轴编号
axisR=2
viraxisx=3    ' 虚拟轴编号
viraxisy=4

dim CENTERX    ' x 方向偏移
dim CENTERY    ' y 方向偏移
dim IfAntiClock    ' 是否为逆时针旋转，逆时针旋转遵循右手法则
CENTERX=10    ' 旋转中心偏移
CENTERY=10
IfAntiClock=1 ' 此时为逆时针关系

dim u_mx    ' x 电机旋转一圈的脉冲
dim u_my    ' y 电机旋转一圈的脉冲
dim u_mR    ' 旋转电机旋转一圈的脉冲
u_mx=10000
u_my=10000
u_mR=10000    ' 脉冲量赋值

dim i_x    ' x 电机传动比
dim i_y    ' y 电机传动比
dim i_R    ' 旋转电机传动比
i_x=1
i_y=1
i_R=1    ' 传动比赋值

dim u_jx ' x 关节实际一圈脉冲数
dim u_jy ' y 关节实际一圈脉冲数
dim u_jR ' 旋转关节实际一圈脉冲数
u_jx=u_mx*i_x
u_jy=u_my*i_y
u_jR=u_mR*i_R

dim p_x    ' x 螺距

```

```

dim p_y      ' y 螺距
p_x=2
p_y=2

global RET

'''''''''' 定义关节轴
base(axisx,axisy,axisR)
rapidstop
wait idle      ' 确保所有轴停止运动
units=u_jx/p_x,u_jy/p_y,u_jR/360 ' xy 设为 1mm 脉冲数，旋转轴设为 1° 脉冲数
atype = 1,1,1      ' 轴功能设置
dpos=0,0,0
creep=10,10,10      ' 轴回零速度
speed =300,300,300      ' 移动速度
accel = 1000,1000,1000 ' 加速度
decel = 500,500,500      ' 减速度

'''''''''' 虚拟轴定义
base(viraxisx,viraxisy)
atype = 0,0      ' 定义虚拟轴
TABLE(0,CENTERX,CENTERY,u_jR,IfAntiClock ) ' table 表参数
speed=200,200      ' 虚拟轴运动速度
accel = 300,300      ' 虚拟轴运动加速度
units=10000,10000

base(axisx,axisy,axisR)
CONNREFRAME(11,0,viraxisx, viraxisy,axisR) ' 启动正解连接。
wait loaded

' 假定已经按照校正操作前 3 步骤，记录到了关节坐标
table(300,0,0,0)      ' 第一个点
table(303,-6,10,-45)      ' 第二个点
table(306,5.2,10.8,45)      ' 第三个点
table(200) = 0      ' 辅助点

RET = FRAME_CAL(300,3,3,200,100,150)      ' 开始校正
' FRAME_CAL (tablein, space, groups, tableaux, zeroout, [tableout2])
' 校正的实际 0 点时关节轴绝对坐标存放在 table100，机械手参数存放在 table150

if not ret then
    print "FRAMECAL failed"
Else
    for i = 0 to 2

```

```
        print "table",100+i,"= ",table(100 + i)
    next

    for i = 0 to 3
        print "table",150+i,"= ",table(150 + i)
    next

    RAPIDSTOP(2) '取消机械手连接
    wait IDLE

    base(axisx,axisy,axisR)

    moveabs(table(100),table(101),table(102)) '移动到校正后的0点位置
    wait IDLE

    dpos=0,0 '关节轴当前位置设为0点

    base(viraxisx,viraxisy,axisR)
    CONNREFRAME(11,150,axisx,axisy,axisR) '用校正后的机械手参数重新建立连接。
    wait loaded

end if
end
```

FRAME17/27

校正内容

大旋转中心实际偏移 CentY1,CentZ1, 小旋转中心实际偏移 CentX2,CentY2, 大旋转轴零点时的位置。

校正要求

- 1) 旋转轴 UNITS 设置为正确的一圈脉冲数。
- 2) 旋转正向与机械手连接设置方向一致。
- 3) 大旋转中心线与 x 水平,小旋转中心线与 z 水平。

校正方法

此时 Table (Tableaux) =0, 在世界坐标系中同一个位置, 对应旋转轴不同的角度, 角度范围大一些。

至少记录 6 个关节坐标, 其中大旋转轴相同角度, 小旋转不一样角度下至少 3 点; 小旋转轴相同角度, 大旋转不一样角度下至少 3 点。

校正前填写 table 数据

- 点 1 关节轴坐标填入编号为 FRAME_CAL 指令参数 tablein 一致的 table。
- 点 2 关节轴坐标填入编号为 tablein+space*1 的 table。
- 点 3 关节轴坐标填入编号为 tablein+space*2 的 table。
- 点 4 关节轴坐标填入编号为 tablein+space*3 的 table。
- 点 5 关节轴坐标填入编号为 tablein+space*4 的 table。
- 点 6 关节轴坐标填入编号为 tablein+space*5 的 table。

编号为 Tableaux 的 table，填 0。

校正后输出 table 数据

校正后理论零点时关节轴的绝对坐标存储在指令参数 Zerooutz 指定的 table。

校正后实际的机械手参数存储在指令参数 Tableout2 指定的 table。

例程

```

dim axisx,axisy,axisz,axisr1,axisr2    ''定义实际轴
dim viraxisx,viraxisy,viraxisz        ''定义虚拟轴
axisx=0    '''实际轴编号
axisy=1
axisz=2
axisr1=6    '''旋转轴编号
axisr2=7

viraxisx=3    '''虚拟轴编号
viraxisy=4
viraxisz=5

dim centy1,centz1,centx2,centy2        '旋转中心定义
dim ifanticlock1,ifanticlock2          '是否逆时针
centy1=10    '偏移
centz1=10
centx2=10
centy2=10
ifanticlock1=1    '旋转台 1 逆时针
ifanticlock2=1    '旋转台 2 逆时针

dim u_mx,u_my,u_mz    '轴旋转一圈的脉冲
dim u_mr1,u_mr2
u_mx=10000    '旋转轴脉冲参数赋值
u_my=10000
u_mz=10000
u_mr1=10000
u_mr2=10000

```

```

dim i_x,i_y,i_z      '电机传动比
dim i_r1,i_r2
i_x=2
i_y=2
i_z=2
i_r1=1
i_r2=1

dim u_jx      'x 关节实际一圈脉冲数
dim u_jy      'y 关节实际一圈脉冲数
dim u_jz      'z 关节实际一圈脉冲数
dim u_jr1     '旋转 1 实际一圈脉冲数
dim u_jr2     '旋转 2 实际一圈脉冲数
u_jx=u_mx*i_x
u_jy=u_my*i_y
u_jz=u_mz*i_z
u_jr1=u_mr1*i_r1
u_jr2=u_mr2*i_r2

dim p_x      'x 螺距
dim p_y      'y 螺距
dim p_z      'z 螺距
p_x=2
p_y=2
p_z=10

dim RET

''''''''''定义关节轴
BASE(axisx,axisy,axisz)      '实际轴定义
ATYPE=1,1,1                  '轴运动类型
rapidstop
wait idle                    '确保所有轴停止运动
UNITS=u_jx/p_x,u_jy/p_y,u_jz/p_z      '设置为 1mm 的脉冲数
Dpos=0,0,0
speed=500,500,500
CREEP=20,20,20
ACCEL=1000,1000,1000
DECEL=500,500,500

base(axisr1,axisr2)          '旋转轴设置
ATYPE=1,1                    '轴运动类型
units =u_jr1/360,u_jr1/360    '设置为 1° 的脉冲数
Dpos=0,0

```

```

speed=200, 200          ' 末端旋转轴参数设置
accel=300, 300

'''''''''' 虚拟轴定义
base(viraxisx, viraxisy, viraxisz)    ' 虚拟轴定义
ATYPE=0, 0, 0
TABLE(0, CentY1, CentZ1, u_jr1, IfAntiClock1, CentX2, CentY2, u_jr2, IfAntiClock2)    '''' table
表参数填写
UNITS=1000, 1000, 1000    ''' 虚拟轴运动参数赋值
SPEED=300, 300, 300
ACCEL=500, 500, 500

BASE(viraxisx, viraxisy, viraxisz, axisr1, axisr2)          ' 选择虚拟轴号
CONNREFRAME(17, 0, axisx, axisy, axisz, axisr1, axisr2)    ' 启动正解连接。
WAIT LOADED          ' 等待运动加载。
,
'BASE(axisx, axisy, axisz, axisr1, axisr2)                  ' 选择关节轴号
'CONNFRAME(17, 0, viraxisx, viraxisy, viraxisz, axisr1, axisr2)    ' 启动逆解连接。
'WAIT LOADED          ' 等待运动加载，此时会自动调整虚拟轴的位置。

' 假定已经按照校正操作前 3 步骤，记录到了关节坐标
table(300, 22.929, -15.355, 0, 0, 45)
table(305, 50, -10, 0, 0, 90)
table(310, 60, 40, 0, 0, 180)
table(315, 0, 10, -4.142, 45, 0)
table(320, 0, 20, 0, 90, 0)
table(325, 0, 0, 0, 0, 0)

table(200) = 0          ' 辅助点

RET = FRAME_CAL(300, 5, 6, 200, 100, 150)
'FRAME_CAL(tablein, space, groups, tableaux, zeroout, [tableout2])
'校正的实际 0 点时关节轴绝对坐标存放在 table100，机械手参数存放在 table150 位置

if not ret then
    print "FRAMECAL failed"
Else
    for i = 0 to 4
        print "table", 100+i, "= ", table(100 + i)
    next

```

```
for i = 0 to 7
    print "table",150+i,"= ",table(150 + i)
next

RAPIDSTOP(2) '取消机械手连接
wait IDLE

BASE(axisx,axisy,axisz,axisr1,axisr2)
moveabs(table(100),table(101),table(102),table(103),table(104)) '移动到校正后的 0 点位置
wait IDLE

dpos=0,0 '关节轴当前位置设为 0 点

BASE(viraxisx,viraxisy,viraxisz,axisr1,axisr2) '选择虚拟轴号
CONNREFRAME(17,150,axisx,axisy,axisz,axisr1,axisr2) '启动正解连接。
WAIT LOADED
end if
end
```

FRAME101,105

校正内容

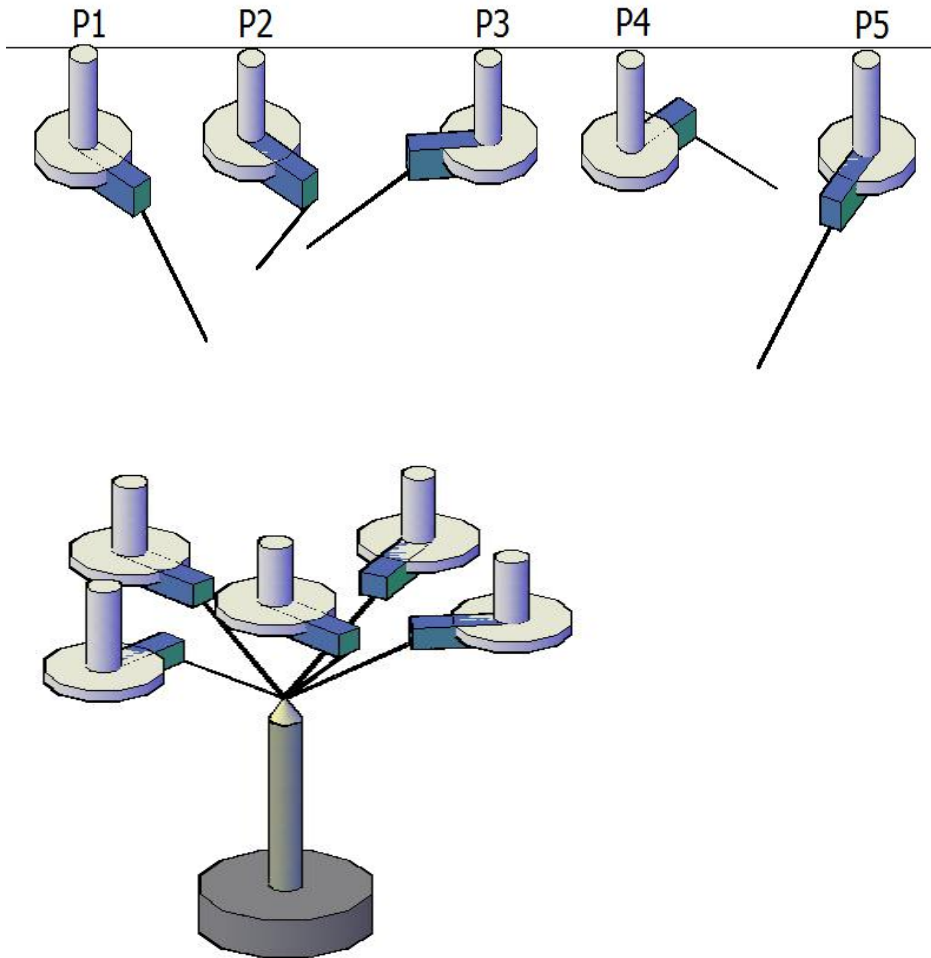
LW 长度，LVWX, LVWY 偏移，零点时 vw 的偏差。
大小关节的零位偏移需要先设置为 FRAME1 校正。

校正要求

- 1) 大小关节的零位已校正。
- 2) J1, J2 的长度是正确的。

校正方法

此时 Table (Tableaux) =0，通过不同的关节坐标，运动到世界坐标系中同一个位置。至少记录 5 个关节坐标位置，如下图所示。



P1~P3 :在世界坐标系中 $x y z$ 位置相同，末端旋转轴 v 角度相同，摆动轴 w 角度不相同的点，至少记录三个。

P4~P5 :在世界坐标系中 $x y z$ 位置相同，摆动轴 w 角度相同（不能为 0），末端旋转轴角度不相同的点，至少记录两个。

校正前填写 table 数据

- 点 1 关节轴坐标填入编号为 FRAME_CAL 指令参数 tablein 一致的 table。
 - 点 2 关节轴坐标填入编号为 tablein+space*1 的 table。
 - 点 3 关节轴坐标填入编号为 tablein+space*2 的 table。
 - 点 4 关节轴坐标填入编号为 tablein+space*3 的 table。
 - 点 5 关节轴坐标填入编号为 tablein+space*4 的 table。
- 编号为 Tableaux 的 table，填 0。

校正后输出 table 数据

- 校正后理论零点时关节轴的绝对坐标存储在指令参数 Zerooutz 指定的 table。
- 校正后实际的机械手参数存储在指令参数 Tableout2 指定的 table。

例程

```
'''' 电机、机械手参数定义
```

```
dim L1      '大臂长度
dim L2      '小臂长度
dim LW      '摆动半径
dim Lvwx    '末端 x 偏移
dim Lvwy    '末端 y 偏移
L1=50
L2=80
LW=10
Lvwx=5
Lvwy=5

dim u_m1    '电机 1 一圈脉冲数
dim u_m2    '电机 2 一圈脉冲数
dim u_mz    '电机 z 一圈脉冲数
dim u_mv    '旋转电机 v 一圈脉冲数
dim u_mw    '摆动电机 w 一圈脉冲数
u_m1=3600
u_m2=3600
u_mz=3600
u_mv=3600
u_mw=3600

dim i_1     '关节 1 传动比
dim i_2     '关节 2 传动比
dim i_z     '关节 z 传动比
dim i_v     '关节 v 传动比
dim i_w     '关节 w 传动比
i_1=2
i_2=2
i_z=2
i_v=2
i_w=2

dim u_j1    '关节 1 实际一圈脉冲数
dim u_j2    '关节 2 实际一圈脉冲数
dim u_jz    '关节 z 实际一圈脉冲数
dim u_jv    '关节 v 实际一圈脉冲数
dim u_jw    '关节 w 实际一圈脉冲数
u_j1=u_m1*i_1
u_j2=u_m2*i_2
u_jz=u_mz*i_z
u_jv=u_mv*i_v
u_jw=u_mw*i_w
```

```

dim p_z      ' z 轴螺距
p_z=10

'''''''' 关节轴设置
base(0, 1, 2, 3, 4)      ' 选择关节轴 0, 1 实际轴
rapidstop              ' 停止原来可能的运动
atype = 1, 1, 1, 1, 1    ' 轴类型为脉冲数
units=u_j1/360,u_j2/360,u_jz/p_z,u_jv/360,u_jw/360  ' z 轴把 units 设为每 mm 的脉冲数, 其他轴设
为每° 脉冲数
speed =300, 300, 300, 300, 300      ' 速度参数设置
accel = 1000, 1000, 1000, 1000, 1000 ' 设置关节轴的加速度限制
decel = 500, 500, 500, 500, 500     ' 设置关节轴的减速度
clutch_rate =0, 0, 0, 0, 0          ' 使用关节轴的速度和加速度限制
DPOS=0, 0, 0, 0, 0                 ' 先手动调整位置, 再坐标清零

'''''''' 虚拟轴设置
base(6, 7, 8, 9)
atype = 0, 0, 0, 0, 0      ' 设置为虚拟轴
TABLE(0, L1, L2, u_j1, u_j2, u_jv, Lw, u_jw, Lvwx, Lvwy)      ' 填写参数
UNITS = 1000, 1000, 1000, u_jv/360          ' 提前设置, 中途不能变化, 末端两个旋转轴保证 units 与
关节轴一致
merge = on                          ' 打开连续插补
CORNER_MODE =2                       ' 启动拐角减速
DECEL_ANGLE = 15 * (PI/180) ' 开始减速的角度 15 度
STOP_ANGLE = 45 * (PI/180) ' 降到最低速度的角度 45 度

''' 建立机械手连接
    BASE(6, 7, 8, 9, 4)
    CONNREFRAME(101, 0, 0, 1, 2, 3, 4)      ' 启动正解连接
    wait loaded
    ?"正解模式"

    ' 假定已经按照校正操作前 3 步骤, 记录到了关节坐标

    table(300, -14.35, 89.25, 4.3958, -64.85, -20)
    table(305, -13.6, 87.35, 3.6597, -63.8, -30)
    table(310, -16.35, 94, 4.9611, -67.6, 5)
    table(315, -14.45, 93.65, 4.9611, -89.15, 5)
    table(320, -17.45, 96.7, 4.398, -69.25, 20)

    table(200) = 0                      ' 辅助点
    RET = FRAME_CAL(300, 5, 5, 200, 100, 150)
    ' FRAME_CAL(tablein, space, groups, tableaux, zeroout, [tableout2])

```

```

'校正的实际 0 点时关节轴绝对坐标存放在 table100, 机械手参数存放在 table150

if not ret then

    print "FRAMECAL failed"
else
    for i = 0 to 10
        print "table",100+i,"= ",table(100 + i) '打印零点时关节轴绝对坐标
    next
    for i = 0 to 10
        print "table",150+i,"= ",table(150 + i) '打印实际机械手参数
    next

    RAPIDSTOP(2) '取消机械手连接
    wait idle

    BASE(6, 7, 8, 9, 4)
    moveabs(table(100), table(101), table(102), table(103), table(104)) '移动到校正后的 0
点位置
    wait IDLE

    dpos=0, 0, 0, 0, 0 '关节轴当前位置设为 0 点

    base(6, 7, 8, 9, 4)
    CONNREFRAME(101, 150, 0, 1, 2, 3, 4) '用校正后的机械手参数重新建立连接。
    wait loaded
end if

end

```

FRAME100,104

分为四轴模式与五轴模式两种情况。

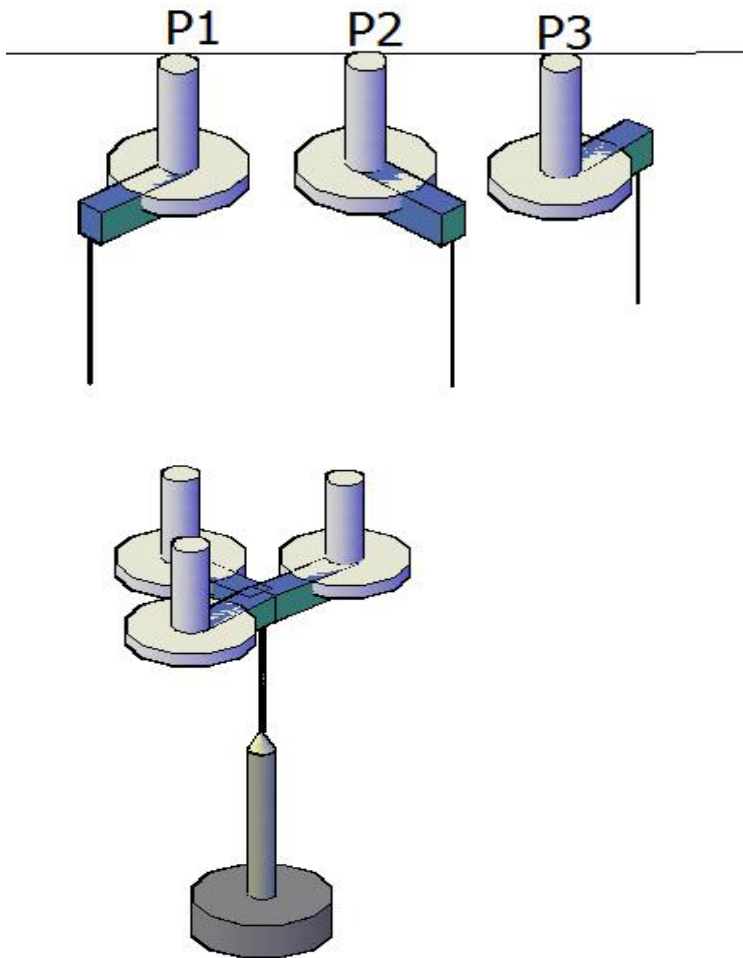
四轴模式

校正内容

LVWX, LVWY 的偏移。

校正方法

此时 Table (Tableaux) =0, 通过不同的关节坐标, 运动到世界坐标系中同一个位置。至少记录 3 个关节坐标位置, 如下图所示。



P1~P3: 在世界坐标系中 $x y z$ 位置相同, 末端旋转轴 v 角度不相同的点, 至少记录三个。

校正前填写 table 数据

- 点 1 关节轴坐标填入编号为 FRAME_CAL 指令参数 tablein 一致的 table。
- 点 2 关节轴坐标填入编号为 tablein+space*1 的 table。
- 点 3 关节轴坐标填入编号为 tablein+space*2 的 table。
- 编号为 Tableaux 的 table, 填 0。

校正后输出 table 数据

- 校正后理论零点时关节轴的绝对坐标存储在指令参数 Zerooutz 指定的 table。
- 校正后实际的机械手参数存储在指令参数 Tableout2 指定的 table。

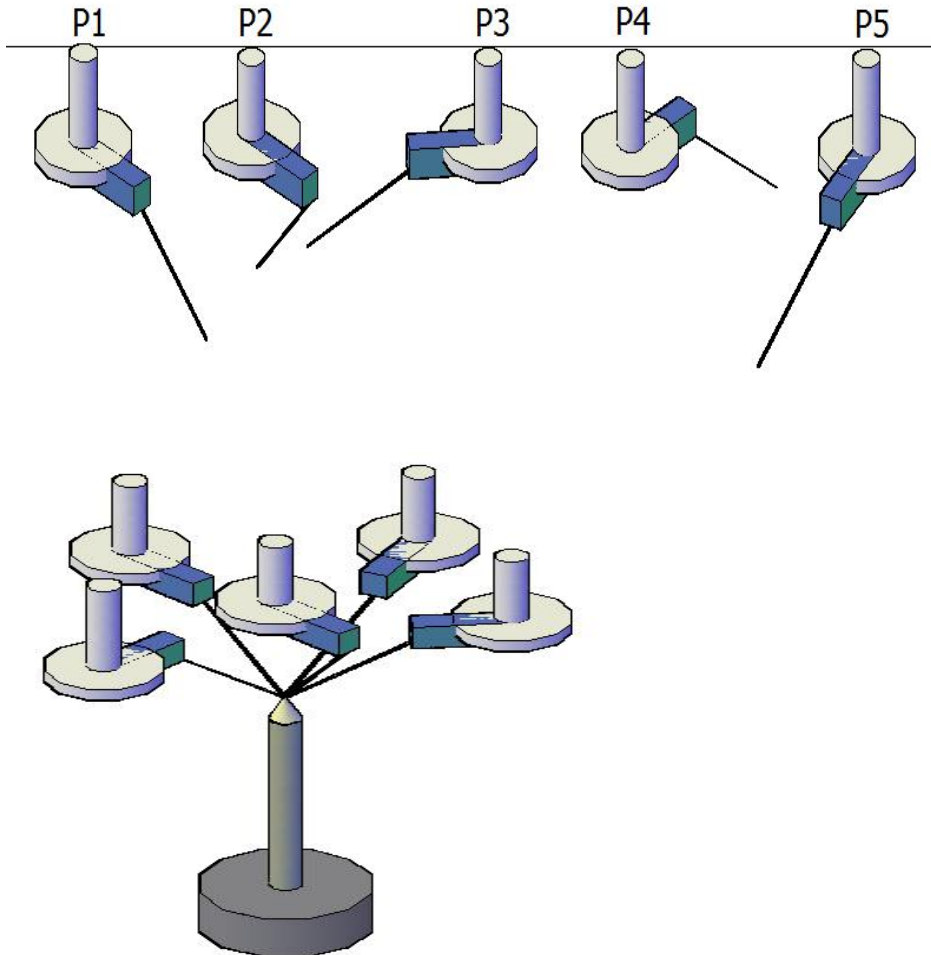
五轴模式

校正内容

- LW 长度, LVWX, LVWY 偏移, 零点时 vw 的偏差。

校正方法

此时 $\text{Table}(\text{Tableaux})=0$ ，通过不同的关节坐标，运动到世界坐标系中同一个位置。至少记录 5 个关节坐标位置，如下图所示。



P1~P3 :在世界坐标系中 $x y z$ 位置相同，末端旋转轴 v 角度相同，摆动轴 w 角度不相同的点，至少记录三个。

P4~P5 :在世界坐标系中 $x y z$ 位置相同，摆动轴 w 角度相同（不能为 0），末端旋转轴角度不相同的点，至少记录两个。

校正前填写 table 数据

点 1 关节轴坐标填入编号为 `FRAME_CAL` 指令参数 `tablein` 一致的 `table`。

点 2 关节轴坐标填入编号为 `tablein+space*1` 的 `table`。

点 3 关节轴坐标填入编号为 `tablein+space*2` 的 `table`。

点 4 关节轴坐标填入编号为 `tablein+space*3` 的 `table`。

点 5 关节轴坐标填入编号为 `tablein+space*4` 的 `table`。

编号为 `Tableaux` 的 `table`，填 0。

校正后输出 table 数据

校正后理论零点时关节轴的绝对坐标存储在指令参数 Zerooutz 指定的 table。
校正后实际的机械手参数存储在指令参数 Tableout2 指定的 table。

例程

```

global const axis_x = 0
global const axis_y = 1
global const axis_z = 2
global const axis_v = 3
global const axis_w = 4

global const axis_virx = 6
global const axis_viry = 7
global const axis_virz = 8
global const axis_virv = 9

    local lw
    local lwx
    local lwy

    lw = 20
    lwx = 10
    lwy = 10

    for i = 0 to 400
    table(i) = 0
    next

    base(axis_x, axis_y, axis_z, axis_v, axis_w)
    atype = 1, 1, 1, 1, 1
    units = 100, 100, 100, 100, 100

    base(axis_virx, axis_viry, axis_virz)
    atype = 0, 0, 0                ' 设置为虚拟轴

    TABLE(0, lw, 3600, 3600, lwx, lwy)    ' 参数存储在 TABLE0 开始的位置, 电机一圈 360 个脉冲。

    UNITS(6)=1000                    ' 此脉冲当量要提前设置, 中途不能变化
    UNITS(7)=1000                    ' 此脉冲当量要提前设置, 中途不能变化
    units(8)=1000                    ' 虚拟旋转轴与关节轴 R 一致

    base(axis_virx, axis_viry, axis_virz, axis_v, axis_w)
    CONNREFRAME(100, 0, axis_x, axis_y, axis_z, axis_v, axis_w) ' 启动正解连接
    wait loaded

```

'假定已经按照校正操作前 3 步骤，记录到了关节坐标

```
table(300,21.5800, 1.8800, 0.0000 ,10.0000 ,0.0000 )
table(305,0, 0, 0.0000, 0, 0.0000 )
table(310,13.6600, -3.6600, 0.0000, 6.0000, 0.0000)
table(315,-15.3200, 0.0000 , -7.1400, 0.0000, 5.0000)
table(320,-19.6900,0.0000, -23.4700, 0.0000, 10.0000)
```

```
table(200) = 0          '辅助点
RET = FRAME_CAL(300,5,5,200,100,150)
'FRAME_CAL (tablein, space, groups, tableaux, zeroout, [tableout2])
if not ret then
    print "FRAMECAL failed"
```

else

```
    for i = 0 to 10
        print "table",100+i,"= ",table(100 + i)
    next
```

```
    for i = 0 to 10
        print "table",150+i,"= ",table(150 + i)
    next
```

```
RAPIDSTOP(2)          '取消机械手连接
wait idle
```

```
BASE(6,7,8,9,4)
moveabs(table(100),table(101),table(102),table(103),table(104)) '移动到校正后的 0
```

点位置

```
wait IDLE
```

```
dpos=0,0,0,0          '关节轴当前位置设为 0 点
```

```
base(axis_virx,axis_viry,axis_virz,axis_v,axis_w)
CONNREFRAME(100,150,axis_x,axis_y,axis_z,axis_v,axis_w) '用校正后的机械手参数
```

重新建立连接。

```
wait loaded
```

```
end if
```

end

FRAME6

校正内容

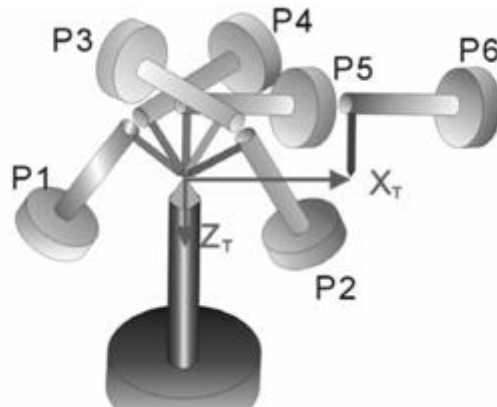
根据校正点的不同，校正不同参数。

4 点法--P1~P4，只计算末端的 SmallZ Small X, Small Y 偏移。

5 点法--P1~P5，P5 的末端认为是垂直的，以此计算 InitRx, InitRy, InitRz。

6 点法--P1~P6，P5 的末端认为是垂直的，P5~P6 之间认为是 X 方向，以此计算 InitRx, InitRy, InitRz。

校正方法



校正前填写 table 数据

点 1 关节轴坐标填入编号为 FRAME_CAL 指令参数 tablein 一致的 table。

点 2 关节轴坐标填入编号为 tablein+space*1 的 table。

点 3 关节轴坐标填入编号为 tablein+space*2 的 table。

点 4 关节轴坐标填入编号为 tablein+space*3 的 table。

点 5 关节轴坐标填入编号为 tablein+space*4 的 table。

点 6 关节轴坐标填入编号为 tablein+space*5 的 table。

编号为 Tableaux 的 table，填 0。

校正后输出 table 数据

校正后理论零点时关节轴的绝对坐标存储在指令参数 Zerooutz 指定的 table。

校正后实际的机械手参数存储在指令参数 Tableout2 指定的 table。

例程

```
rapidstop(2)
dim p1(6)
dim p2(6)
dim p3(6)
dim p4(6)
dim p5(6)
```

```

dim p6(6)
dim j
dim i
""电机、机械手参数定义
dim L1      '大臂长度
dim L2      '小臂长度
dim L3      'X 方向偏移
dim ZDis    '旋转轴一圈，z 轴移动距离
L1=100
L2=100
L3=0
ZDis=0
""电机、机械手参数定义
dim LargeZ  '基座的垂直高度
dim L1      '1 轴到 2 轴的 X 偏移；转盘中心到大摆臂中心的偏移。
dim L2      '大摆臂长度
dim L3      '3 轴中心到 4 轴中心距离
dim L4      '4 轴到 5 轴的距离。
dim D5      '5 转一圈，6 转动的圈数，0 表示不关联。
dim PulesVROneCircle '虚拟姿态轴一圈脉冲数
dim SmallZ  '末端到 5 轴的垂直距离
dim SmallX, SmallY  '末端到转盘中心的 XY 偏移。 0 位置的时候
dim InitRx, InitRy, InitRz '初始的姿态，焊枪针头的方向。 0, 0, 0 指向 z
正向
LargeZ=50
L1=0
L2=100
L3=0
L4=60
D5=0
SmallZ=10
SmallX=0
SmallY=0
InitRx=0
InitRy=0
InitRz=0
PulesVROneCircle=360*10000

dim u_m1    '电机 1 一圈脉冲数
dim u_m2    '电机 2 一圈脉冲数
dim u_m3    '电机 3 一圈脉冲数
dim u_m4    '电机 4 一圈脉冲数
dim u_m5    '电机 5 一圈脉冲数
dim u_m6    '电机 6 一圈脉冲数

```

```

u_m1=360000
u_m2=360000
u_m3=360000
u_m4=360000
u_m5=360000
u_m6=360000

```

```

dim i_1      '关节 1 传动比
dim i_2      '关节 2 传动比
dim i_3      '关节 3 传动比
dim i_4      '关节 4 传动比
dim i_5      '关节 5 传动比
dim i_6      '关节 6 传动比
i_1=1
i_2=1
i_3=1
i_4=1
i_5=1
i_6=1

```

```

dim u_j1     '关节 1 实际一圈脉冲数
dim u_j2     '关节 2 实际一圈脉冲数
dim u_j3     '关节 3 实际一圈脉冲数
dim u_j4     '关节 4 实际一圈脉冲数
dim u_j5     '关节 5 实际一圈脉冲数
dim u_j6     '关节 6 实际一圈脉冲数
u_j1=u_m1*i_1
u_j2=u_m2*i_2
u_j3=u_m3*i_3
u_j4=u_m4*i_4
u_j5=u_m5*i_5
u_j6=u_m6*i_6

```

```

""关节轴设置
BASE(0,1,2,3,4,5)           '选择关节轴号 0、1、2、3、4、5
atype=1,1,1,1,1,1         '轴类型设为脉冲轴
UNITS = u_j1/360,u_j2/360,u_j3/360,u_j4/360,u_j5/360 ,u_j6/360  '把 units
设成每°脉冲数
DPOS=0,0,0,0,0,0         '设置关节轴的位置，此处要根据实际
情况来修改。
speed=100,100,100,100,100,100      '速度参数设置
accel=1000,1000,1000,1000,1000,1000
decel=1000,1000,1000,1000,1000,1000

```

```

CLUTCH_RATE=0,0,0,0,0,0      '使用关节轴的速度和加速度限制

merge=on                      '开启连续插补
corner_mode = 2               '启动拐角减速
decel_angle = 15 * (PI/180)   '开始减速的角度 15 度
stop_angle = 45 * (PI/180)    '降到最低速度的角度 45 度

"*****虚拟轴设置*****"
BASE(6,7,8,9,10,11)
ATYPE=0,0,0,0,0,0            '设置为虚拟轴
TABLE(0,LargeZ,L1,L2,L3,L4,D5,u_j1,u_j2,u_j3,u_j4,u_j5,u_j6,PulesVROneCircle,SmallX,SmallY,SmallZ,InitRx,InitRy,InitRz) '根据手册说明填写参数
UNITS=10000,10000,10000,10000,10000,10000 '运动精度,要提前设置,中途不能变化

"*****建立机械手连接*****"
funtorobt

"*****正确遇到并取点信息*****"
'p1 点
base(6,7,8,9,10,11)
moveabs(50,20,100,0,-60,-30)
wait idle
savejdpos

'p2 点
moveabs(50,20,100,0,60,0)
wait idle
savejdpos

'p3 点
moveabs(50,20,100,0,-60,0)
wait idle
savejdpos

'p4 点
moveabs(50,20,100,0,60,30)
wait idle
savejdpos

'p5 点
moveabs(50,20,100,0,180,0)
wait idle
savejdpos

```



```
'p6 点
moveabs(60,20,100,0,180,0)
wait idle
savejdpos

for i=0 to 5
    print "table(",350+i,")="table(350+i)
    print "table(",360+i,")="table(360+i)
    print "table(",370+i,")="table(370+i)
    print "table(",380+i,")="table(380+i)
    print "table(",390+i,")="table(390+i)
    print "table(",400+i,")="table(400+i)
next

table(300)=0
base(0)
Frame_cal(350,10,6,300,420,450)
for k=0 to 19
    print "table(",450+k,")="table(450+k)
next k
for k=0 to 19
    print "table(",420+k,")="table(420+k)
next k
end

global sub funtorobt()
    BASE(0,1,2,3,4,5)                '选择关节轴号
    CONNFRAME(6,0,6,7,8,9,10,11)    '启动逆解连接。
    WAIT LOADED                      '等待运动加载，此时会自动调整虚拟轴的位置。
    ?"逆解模式"
end sub

global sub savejdpos()
for i=0 to 5
    table(350+i+10*j)=dpos(i)
next
j=j+1
delay(10)
end sub
```

五、关节轴插补

关节插补运动是机械手在正解模式下，控制末端点走直线、圆弧等插补运动。

相关指令：

[MOVER_L](#) 直线插补

[MOVER_C](#) 平面圆弧插补

[MOVER_C3](#) 空间圆弧插补

正解模式下直接用 MOVE 指令操作关节轴，即使角度计算正确，末端工作点走出的轨迹也不能保证为直线。而 MOVER_L 指令操作关节轴，可以保证运动的轨迹为准确的直线。

使用 MOVER_L 等指令不支持运动中脉冲叠加。

六、逆解模式速度参考设置

逆解运动模式是指机械手逆解运动时，参考关节轴速度参数，还是参考虚拟轴的速度参数。

逆解运动模式通过设置关节轴指令 clutch_rate 实现，控制器默认 clutch_rate 为 1000000。

关节轴的 clutch_rate	运动模式描述：
0	平滑模式： 此模式下关节轴使用自己的速度加速度做速度规划，速度平滑，但只能保证运动终点位置正确，中间运动轨迹会有变形。
非 0	强制模式： 此模式下关节轴完全按照虚拟轴的速度加速度进行规划，保证运动轨迹准确，但高速时可能会产生抖动。

机械手保证结束点的位置准确的前提下，根据实际应用对运动过程的轨迹准确与速度平滑之间做取舍。

举例说明

```

""电机、机械手参数定义
dim L1      '大臂长度
dim L2      '小臂长度
dim L3      'X 方向偏移
dim ZDis    '旋转轴一圈，z 轴移动距离
L1=100
L2=100

```

```

L3=0
ZDis=0

dim u_m1 '电机 1 一圈脉冲数
dim u_m2 '电机 2 一圈脉冲数
dim u_mz '电机 z 一圈脉冲数
dim u_mv '电机 v 一圈脉冲数
u_m1=3600
u_m2=3600
u_mz=3600
u_mv=3600

dim i_1 '关节 1 传动比
dim i_2 '关节 2 传动比
dim i_z '关节 z 传动比
dim i_v '关节 v 传动比
i_1=2
i_2=2
i_z=2
i_v=2

dim u_j1 '关节 1 实际一圈脉冲数
dim u_j2 '关节 2 实际一圈脉冲数
dim u_jz '关节 z 实际一圈脉冲数
dim u_jv '关节 v 实际一圈脉冲数
u_j1=u_m1*i_1
u_j2=u_m2*i_2
u_jz=u_mz*i_z
u_jv=u_mv*i_v

dim p_z 'z 轴螺距
p_z=1.5

""关节轴设置
BASE(0,1,2,3) '选择关节轴号
atype=1,1,1,1 '轴类型设为脉冲轴
UNITS=u_j1/360,u_j2/360,u_jv/360,u_jz/p_z '把 z 轴 units 设成 1mm 的脉冲数
DPOS=0,0,0,0 '设置关节轴的位置，此处要根据实际情况来修改。
speed=100,100,500,1000 '速度参数设置
accel=2000,2000,5000,5000
decel=2000,2000,5000,5000
CLUTCH_RATE=0'使用关节轴的速度和加速度限制
merge=on

```

```

corner_mode=2
DECEL_ANGLE = 15 * (PI/180)      '设置开始减速角度
STOP_ANGLE = 45 * (PI/180)      '设置结束减速角度
FORCE_SPEED=10

""虚拟轴设置
BASE(6,7,8,9)
ATYPE=0,0,0,0                    '设置为虚拟轴
TABLE(0,L1,L2,u_j1,u_j2,u_jv, L3,ZDis) '根据手册说明填写参数
UNITS=1000,1000 ,u_jv/360,1000    '运动精度，要提前设置，中途不能变化
speed=1000,1000,500,1000        '速度参数设置
accel=5000,5000,5000,5000
decel=5000,5000,500,5000
merge=on
corner_mode=2
DECEL_ANGLE = 15 * (PI/180)      '设置开始减速角度
STOP_ANGLE = 45 * (PI/180)      '设置结束减速角度
FORCE_SPEED=50

BASE(0,1,2,3)                    '选择关节轴号
CONNFRAME(1,0,6,7,8,9)          '第 6/7 轴作为虚拟的 XY 轴，启动逆解连接。
WAIT LOADED                      '等待运动加载，此时会自动调整虚拟轴的位置。

TRIGGER
delay(10)
base(6,7)
moveabs(50,150)
wait loaded

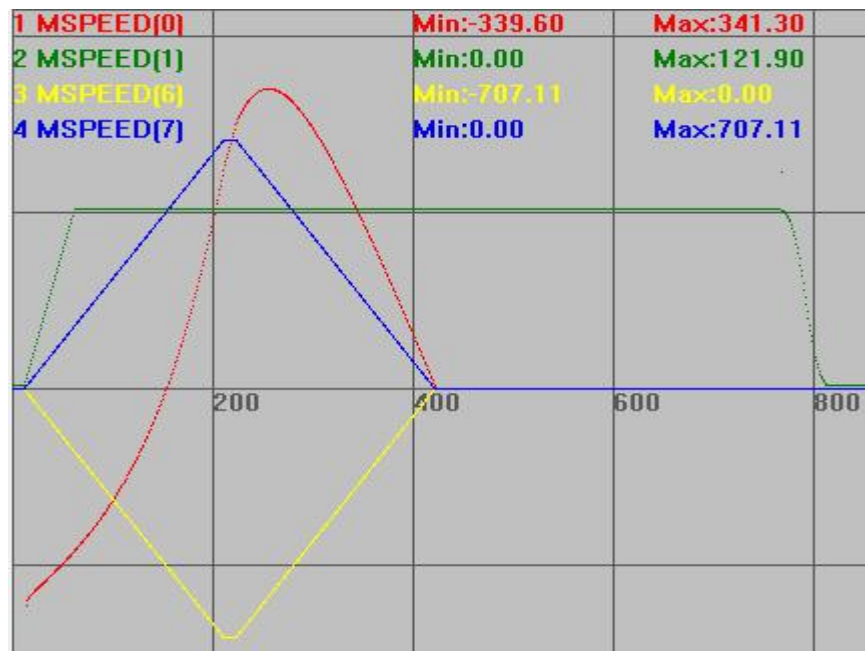
end

```

CLUTCH_RATE=0 情况：
关节轴以自己的加速度做规划。



CLUTCH_RATE=1000000 情况：
 关节轴完全按照虚拟轴的速度自动进行规划。



比较两种情况 mspeed(0)、mspeed(1)的变化曲线：

CLUTCH_RATE=0，其关节轴的最大加减速，不超过设置的加减速速度；
 CLUTCH_RATE<>0，关节轴的加减速超过设置的加减速速度，当前的情况关节轴的最大加减速速度默认为是无穷大。

七、拐角处理

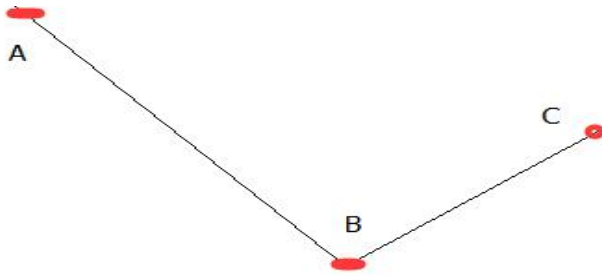
如果运动轨迹中间有拐角，在高速运动时，其拐角处会有速度的突变，机械会有抖动。所以需要进行拐角处理（在对轨迹要求不高的情况下）。

拐角处理：

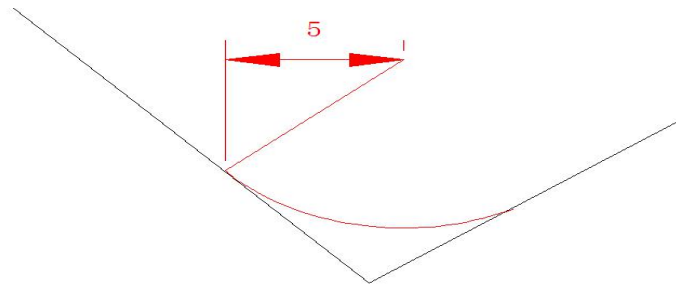
使用 [CORNER_MODE](#) 指令设置拐角减速+自动小圆限速+自动倒角。

```
Base(axis)
CORNER_MODE=2+8+32          '启动拐角减速
DECEL_ANGLE = 15 * (PI/180) '设置开始减速角度
STOP_ANGLE = 45 * (PI/180) '设置结束减速角度
FORCE_SPEED=20              '小圆限速
FULL_SP_RADIUS=5            '限速半径 60
CORNER_MODE=32             '启动倒角
ZSMOOTH = 5                 '倒角参考半径
```

拐角处理前轨迹



拐角处理后轨迹



第六章 各类型机械手使用具体说明

控制器使用步骤说明

各类型机械手使用操作的步骤都是相同的，参考以下步骤。每个步骤对应查看各类型中的对应小节。

- 1) 确认电机转向是否正确。>>查看**电机方向说明**
- 2) 确认机械手各关节轴对应到控制器指令参数的轴次序。>>查看**轴顺序**
- 3) Table 中设置好机械结构相关参数。>>查看**结构参数**
- 4) 设置关节轴参数及虚拟轴参数。>>查看**电机参数**
- 5) 移动各关节轴到规定的零点位置。>>查看**零点位置**
- 6) 使用 CONNREFARME 指令建立正解模式。>>查看**建立连接**
- 7) 操作关节轴调整机械手姿态,确认在运动中不会发生干涉(某些结构只有一个姿态)。>>查看**姿态说明**
- 8) 使用 CONNFARME 指令切换为逆解模式。
- 9) 选择虚拟轴，发送运动指令使用。

机械手建立完整程序查看**使用例程**，仿真时可用 [ZRobotView 机械手模型仿真工具](#)查看实时运动轨迹。

Scara 水平关节类

FRAME1--标准 SCARA

结构示意图

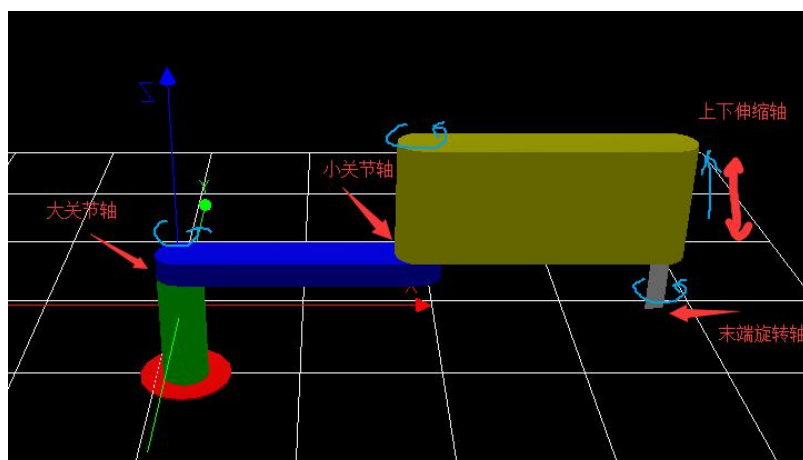
Scara 是水平多关节机械手臂（Selective Compliance Assembly Robot Arm），是一种特殊组态的机械手臂，如下图所示。



支持 2-4 轴，大关节轴+小关节轴 [+Z 轴] [+末端旋转轴]。

电机方向及角度范围

定义各关节电机正向如下图蓝色箭头所示。



	角度或移动范围
大关节轴	$(-2\pi, 2\pi)$
小关节轴	$(-2\pi, 2\pi)$

末端旋转轴	无限制
上下伸缩轴	机械结构限制

轴顺序

实际机械手关节轴:	定义简写:
大关节轴电机	Axis_a
小关节轴电机	Axis_b
末端旋转轴电机	Axis_c
上下伸缩轴电机	Axis_d

直角坐标系虚拟轴:	定义简写:
平移轴 X	Viraxis_x
平移轴 Y	Viraxis_y
旋转轴 RZ	Viraxis_v
平移轴 Z	Viraxis_z

指令规定轴号填写次序如下:

逆解时:

```
BASE(Axis_a,Axis_b[, Axis_c][,Axis_d])
CONNFRAME(1,tablenum, Viraxis_x, Viraxis_y, [Viraxis_v] [,Viraxis_z])
```

正解时:

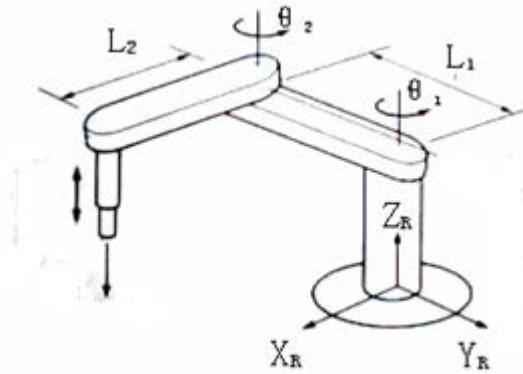
```
BASE(Viraxis_x, Viraxis_y, [Viraxis_v] [,Viraxis_z])
CONNREFRAME(1,tablenum, Axis_a,Axis_b[, Axis_c][,Axis_d])
```

结构参数设置

建立机械手连接时, 需要将机械结构参数依次填写到 table 数组中, 参数说明如下。

TABLE(tablenum, L1,L2, Pules1OneCircle, Pules2OneCircle,[Pules3OneCircle,L3] [,ZDis])	
Tablenum	存储转换参数的 TABLE 地址
L1	大关节轴旋转中心到小关节轴旋转中心的距离
L2	小关节轴旋转中心到末端旋转轴的旋转中心距离
Pules1OneCircle	大关节旋转一圈的脉冲数
Pules2OneCircle	小关节旋转一圈的脉冲数

Pules3OneCircle	末端旋转轴旋转一圈的脉冲数
L3	末端旋转轴旋转中心到末端工作点的距离
ZDis	末端旋转轴旋转一圈，伸缩轴移动的距离



电机参数设置

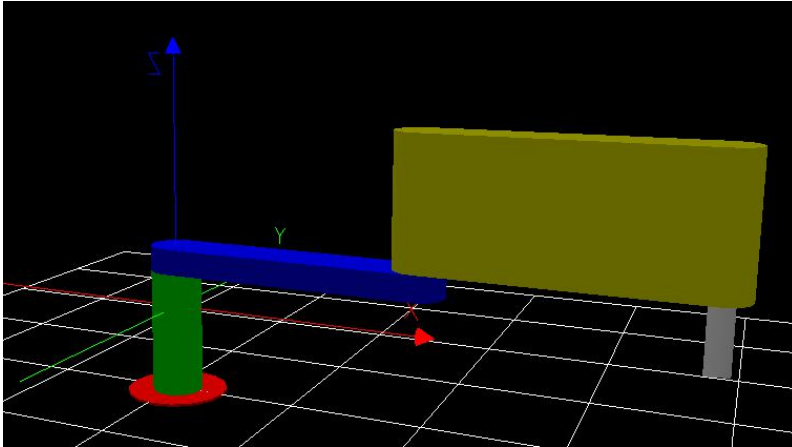
各轴的轴类型和脉冲当量要设置正确。

一个机械手的所有虚拟轴和关节轴的长度单位要求统一，一般都是mm单位，虚拟轴的一个mm的脉冲数一般建议设置为1000，表示精度为小数点后3位。

	Atype	Units
Axis_a	1 或 4 或 7 或 65 或 50	Pules1OneCircle/360
Axis_b		Pules2OneCircle/360
Axis_c		Pules3OneCircle/360
Axis_d		伸缩轴移动 1MM 的脉冲数
Viraxis_x	0	1000
Viraxis_y		1000
Viraxis_v		Pules3OneCircle/360
Viraxis_z		1000

零点位置

两个关节轴的零点时成一条直线，此时指向虚拟 X 轴的正向。
当关节轴为零点的位置时，虚拟轴零点的坐标为 $(L1+L2,0)$ 。
上下伸缩轴零点位置无特殊要求。



建立连接

正解模式

```
BASE(Viraxis_x, Viraxis_y, [Viraxis_v] [,Viraxis_z])
CONNREFRAME(1,tablenum, Axis_a,Axis_b[, Axis_c][,Axis_d])
WAIT LOADED
```

建立成功后，虚拟轴 MTYPE 将显示为 34，IDLE 显示为 0。

此时只能操作关节轴在关节坐标系中运动，虚拟轴会自动计算末端工作点位于直角坐标系中的位置。

逆解模式

```
BASE(Axis_a,Axis_b[, Axis_c][,Axis_d])
CONNFRAME(1,tablenum, Viraxis_x, Viraxis_y, [Viraxis_v] [,Viraxis_z])
WAIT LOADED
```

建立成功后，关节轴 MTYPE 将显示为 33，IDLE 显示为 0。

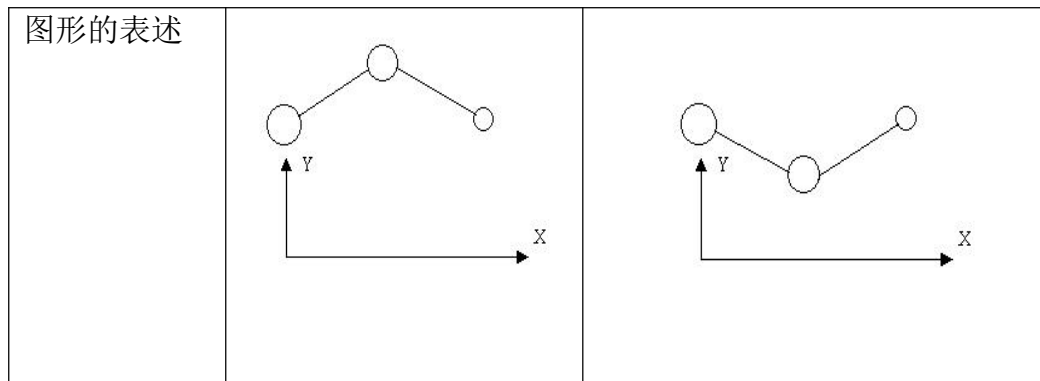
此时只能操作虚拟轴在直角坐标系中运动，关节轴会自动计算在关节坐标系中如何联合运动。

姿态说明

Scara 机械手有两个姿态，只可在正解模式下移动关节轴选择。

当前姿态通过指令 [FRAME_STATUS](#) 查询。

姿态	左手姿态	右手姿态
Frame_status	0	1



使用例程

```

'''电机、机械手参数定义
dim L1      '大臂长度
dim L2      '小臂长度
dim L3      'X 方向偏移
dim ZDis    '旋转轴一圈，z 轴移动距离
L1=10
L2=10
L3=0
ZDis=0

dim u_m1    '电机 1 一圈脉冲数
dim u_m2    '电机 2 一圈脉冲数
dim u_mz    '电机 z 一圈脉冲数
dim u_mv    '电机 v 一圈脉冲数
u_m1=10000
u_m2=10000
u_mz=10000
u_mv=10000

dim i_1     '关节 1 传动比
dim i_2     '关节 2 传动比
dim i_z     '关节 z 传动比
dim i_v     '关节 v 传动比
i_1=2
i_2=2
i_z=2
i_v=2

dim u_j1    '关节 1 实际一圈脉冲数

```

```

dim u_j2    '关节 2 实际一圈脉冲数
dim u_jz    '关节 z 实际一圈脉冲数
dim u_jv    '关节 v 实际一圈脉冲数
u_j1=u_m1*i_1
u_j2=u_m2*i_2
u_jz=u_mz*i_z
u_jv=u_mv*i_v

dim p_z     'z 轴螺距
p_z=1.5

""关节轴设置
BASE(0,1,2,3)      '选择关节轴号
atype=1,1,1,1     '轴类型设为脉冲轴
UNITS=u_j1/360,u_j2/360,u_jv/360,u_jz/p_z '把 z 轴 units 设成 1mm 的脉冲数
                                     '其余轴设成 1°的脉冲数
DPOS=0,0,0,0     '设置关节轴位置，确认移动到零点位置再清 0
speed=100,100,100,100 '速度参数设置
accel=1000,1000,1000,1000
decel=1000,1000,1000,1000
CLUTCH_RATE=0,0,0,0 '使用关节轴的速度和加速度限制

""虚拟轴设置
BASE(6,7,8,9)
ATYPE=0,0,0,0     '设置为虚拟轴
TABLE(0,L1,L2,u_j1,u_j2,u_jv,L3,ZDis) '根据手册说明填写参数
UNITS=1000,1000,u_jv/360,1000 '运动精度，要提前设置，中途不能变化
speed=200,200,200,200 '速度参数设置
accel=1000,1000,1000,1000
decel=1000,1000,1000,1000

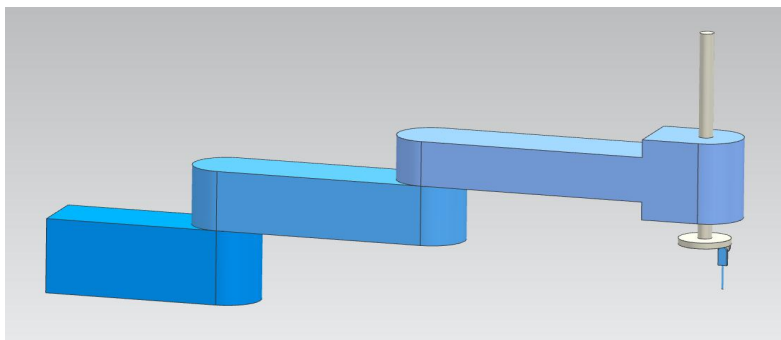
""建立机械手连接
while 1
  if scan_event(in(0))>0 then '输入 0 上升沿触发
    BASE(0,1,2,3)            '选择关节轴号
    CONNFRAME(1,0,6,7,8,9) '第 6/7 轴作为虚拟的 XY 轴，启动逆解连接
    WAIT LOADED             '等待运动加载，此时会自动调整虚拟轴的位置
    ?"逆解模式：此时可以发送运动指令操作虚拟轴"
  elseif scan_event(in(0))<0 then '输入 0 下降沿触发
    BASE(6,7,8,9)           '选择虚拟轴号
    CONNREFRAME(1,0,0,1,2,3) '第 0/1 轴作为关节轴，启动正解连接。
    WAIT LOADED             '等待运动加载
    ?"正解模式：此时可发送运动指令操作关节轴"
  endif
endif

```

wend

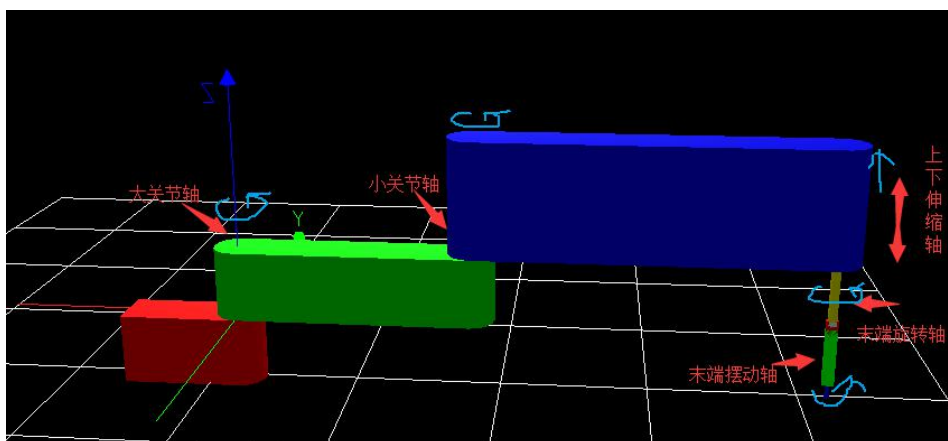
FRAME101/105--标准 SCARA+末端摆动轴

结构示意图



电机方向及角度范围

定义各关节电机正向如下图蓝色箭头所示。



	角度或移动范围
大关节轴	$(-2\pi, 2\pi)$
小关节轴	$(-2\pi, 2\pi)$
末端旋转轴	无限制
上下伸缩轴	无限制
末端摆动轴	无限制

轴顺序

实际机械手关节轴:	定义简写:
大关节轴电机	Axis_a

小关节轴电机	Axis_b
末端旋转轴电机	Axis_c
上下伸缩轴电机	Axis_d
末端摆动轴电机	Axis_e

直角坐标系虚拟轴:	定义简写:
平移轴 X	Viraxis_x
平移轴 Y	Viraxis_y
旋转轴 RZ	Viraxis_v
平移轴 Z	Viraxis_z
摆动轴 w	Axis_e (FRAME101)
	Viraxis_w (FRAME105)

注意末端摆动轴的操作，FRAME101 类型无论处于正解还是逆解，始终用关节轴号控制；FRAME105 类型正解时关节轴号控制，逆解时虚拟轴号控制。

指令规定轴号填写次序如下：

逆解时：

FRAME101 类型

BASE(Axis_a,Axis_b, Axis_c, Axis_d, Axis_e)

CONNFRAME(101,tablenum,Viraxis_x,Viraxis_y,Viraxis_v,Viraxis_z, Axis_e)

FRAME105 类型

BASE(Axis_a,Axis_b, Axis_c, Axis_d, Axis_e)

CONNFRAME(105,tablenum,Viraxis_x,Viraxis_y,Viraxis_v,Viraxis_z, Viraxis_w)

正解时：

FRAME101 类型

BASE(Viraxis_x, Viraxis_y, Viraxis_v,Viraxis_z,Axis_e)

CONNREFRAME(101,tablenum, Axis_a,Axis_b, Axis_c,Axis_d,Axis_e)

FRAME105 类型

BASE(Viraxis_x, Viraxis_y, Viraxis_v,Viraxis_z,Viraxis_w)

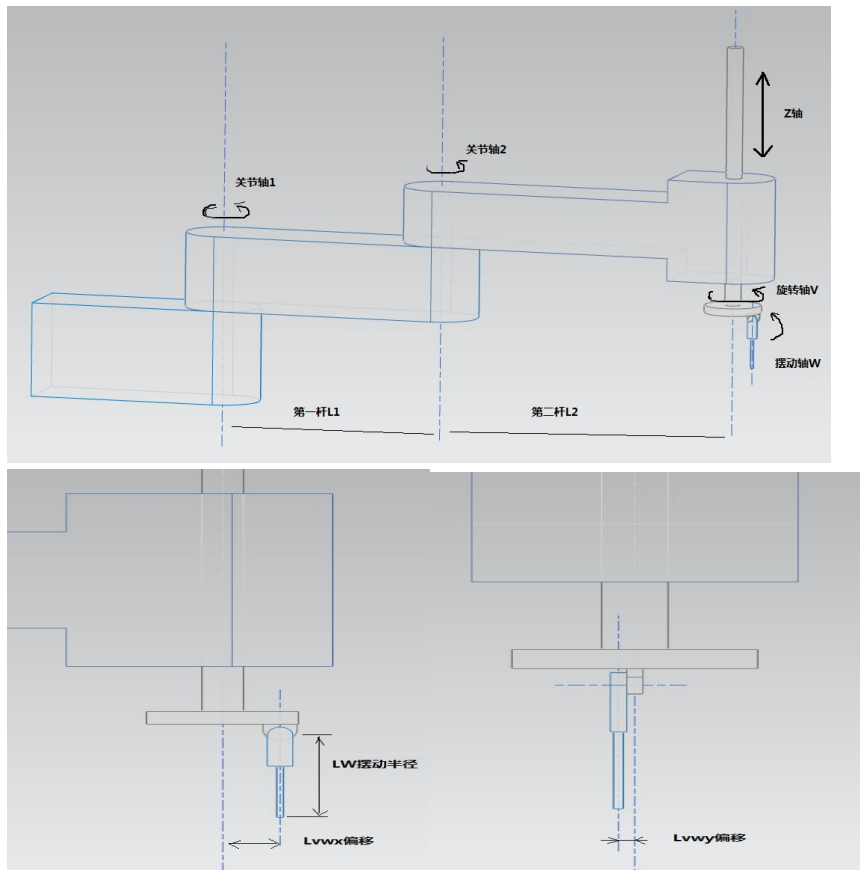
CONNREFRAME(105,tablenum, Axis_a,Axis_b, Axis_c,Axis_d,Axis_e)

结构参数设置

建立机械手连接时，需要将机械结构参数依次填写到 table 数组中，参数说明如下。

TABLE(tablenum,L1,L2,Pules1OneCircle,Pules2OneCircle,PulesvOneCircle,Lw,Pule

sWOneCircle, Lvwx, Lvwy)	
Tablenum	存储转换参数的 TABLE 位置
L1	Xy 平面内, 第一个关节轴的旋转中心到第二个关节轴的旋转中心的距离
L2	Xy 平面内, 第二个关节轴的旋转中心到末端旋转轴的旋转中心的距离
Pules1OneCircle	第一个关节轴旋转一圈的脉冲数
Pules2OneCircle	第二个关节轴旋转一圈的脉冲数
PulesvOneCircle	末端旋转轴旋转一圈的脉冲数
Lw	摆动半径
PuleswOneCircle	末端摆动轴旋转一圈的脉冲数
Lvwx	Xy 平面内, 末端工作点与末端旋转轴旋转中心的 x 方向距离
Lvwy	Xy 平面内, 末端工作点与末端旋转轴旋转中心的 y 方向距离



电机参数设置

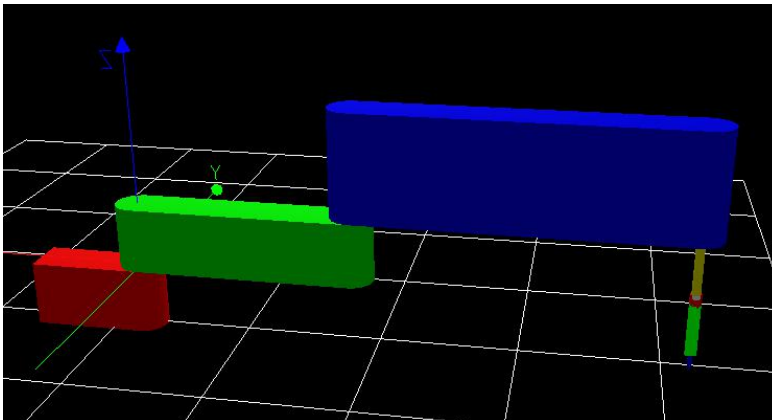
各轴的轴类型和脉冲当量要设置正确。

一个机械手的所有虚拟轴和关节轴的长度单位要求统一, 一般都是 mm 单位, 虚拟轴的一个 mm 的脉冲数一般建议设置为 1000, 表示精度为小数点后 3 位。

	Atype	Units
Axis_a	1 或 4 或 7 或 65 或 50	Pules1OneCircle/360
Axis_b		Pules2OneCircle/360
Axis_c		PulesvOneCircle/360
Axis_d		伸缩轴移动 1MM 的脉冲数
Axis_e		PuleswOneCircle
Viraxis_x	0	1000
Viraxis_y		1000
Viraxis_v		PulesvOneCircle/360
Viraxis_z		1000
Viraxis_w		PuleswOneCircle (FRAME105)

零点位置

两个关节轴的零点时成一条直线，此时指向虚拟 X 轴的正向。
当关节轴为零点的位置时，虚拟轴零点的坐标为 $(L1+L2,0)$ 。
末端摆动轴竖直指向下方。



建立连接

正解模式

FRAME101 类型

```
BASE(Viraxis_x, Viraxis_y, Viraxis_v, Viraxis_z, Axis_e)
CONNREFRAME(101, tablenum, Axis_a, Axis_b, Axis_c, Axis_d, Axis_e)
WAIT LOADED
```

FRAME105 类型

```
BASE(Viraxis_x, Viraxis_y, Viraxis_v, Viraxis_z, Viraxis_w)
CONNREFRAME(105, tablenum, Axis_a, Axis_b, Axis_c, Axis_d, Axis_e)
```

WAIT LOADED

建立成功后，虚拟轴 MTYPE 将显示为 34，IDLE 显示为 0。

此时只能操作关节轴在关节坐标系中运动，虚拟轴会自动计算末端工作点位于直角坐标系中的位置。

逆解模式

FRAME101 类型

BASE(Axis_a,Axis_b, Axis_c, Axis_d, Axis_e)

CONNFRAME(101,tablenum,Viraxis_x,Viraxis_y,Viraxis_v,Viraxis_z, Axis_e)

WAIT LOADED

FRAME105 类型

BASE(Axis_a,Axis_b, Axis_c, Axis_d, Axis_e)

CONNFRAME(105,tablenum,Viraxis_x,Viraxis_y,Viraxis_v,Viraxis_z, Viraxis_w)

WAIT LOADED

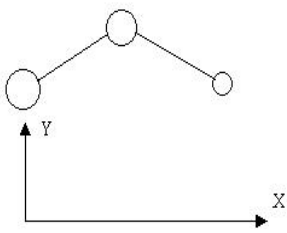
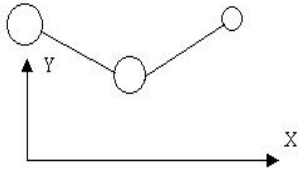
建立成功后，关节轴 MTYPE 将显示为 33，IDLE 显示为 0。

此时只能操作虚拟轴在直角坐标系中运动，关节轴会自动计算在关节坐标系中如何联合运动。

姿态说明

Scara 机械手有两个姿态，只可在正解模式下移动关节轴选择。

当前姿态通过指令 [FRAME_STATUS](#) 查询。

姿态	左手姿态	右手姿态
Frame_status	0	1
图形的表述		

使用例程

以下例程为 FRAME101 类型，若使用 FRAME105，则逆解时注意用虚拟轴

操作末端摆动轴。

```
""电机、机械手参数定义
dim L1      '大臂长度
dim L2      '小臂长度
dim Lw      '摆动半径
dim Lvwx    '末端 x 偏移
dim Lvwy    '末端 y 偏移
L1=20
L2=30
Lw=10
Lvwx=0
Lvwy=0

dim u_m1    '电机 1 一圈脉冲数
dim u_m2    '电机 2 一圈脉冲数
dim u_mz    '电机 z 一圈脉冲数
dim u_mv    '旋转电机 v 一圈脉冲数
dim u_mw    '摆动电机 w 一圈脉冲数
u_m1=10000
u_m2=10000
u_mz=10000
u_mv=10000
u_mw=10000

dim i_1     '关节 1 传动比
dim i_2     '关节 2 传动比
dim i_z     '关节 z 传动比
dim i_v     '关节 v 传动比
dim i_w     '关节 w 传动比
i_1=2
i_2=2
i_z=2
i_v=2
i_w=2

dim u_j1    '关节 1 实际一圈脉冲数
dim u_j2    '关节 2 实际一圈脉冲数
dim u_jz    '关节 z 实际一圈脉冲数
dim u_jv    '关节 v 实际一圈脉冲数
dim u_jw    '关节 w 实际一圈脉冲数
u_j1=u_m1*i_1
```

```

u_j2=u_m2*i_2
u_jz=u_mz*i_z
u_jv=u_mv*i_v
u_jw=u_mw*i_w

dim p_z      'z 轴螺距
p_z=10

""""关节轴设置
base(0,1,2,3,4)      '选择关节轴 0,1 实际轴
rapidstop            '停止原来可能的运动
atype = 1,1,1,1,1    '轴类型为脉冲数
units=u_j1/360,u_j2/360,u_jz/p_z,u_jv/360,u_jw/360  'z 轴把 units 设为每 mm 的
脉冲数，其他轴设为每°脉冲数
speed =300,300,300,300,300      '速度参数设置
accel = 1000,1000,1000,1000,1000  '设置关节轴的加速度限制
decel = 500,500,500,500,500      '设置关节轴的减速度
clutch_rate =0,0,0,0,0          '使用关节轴的速度和加速度限制
DPOS=0,0,0,0,0                  '先手动调整位置，再坐标清零

""""虚拟轴设置
base(6,7,8,9)
atype = 0,0,0,0,0      '设置为虚拟轴
TABLE(0,L1,L2,u_j1,u_j2,u_jv,Lw,u_jw,Lvwx,Lvwy)      '填写参数
UNITS = 1000,1000,1000,u_jv/360      '提前设置，中途不能变化,末端两个
旋转轴保证 units 与关节轴一致
merge = on                '打开连续插补
CORNER_MODE =2            '启动拐角减速
DECEL_ANGLE = 15 * (PI/180) '开始减速的角度 15 度
STOP_ANGLE = 45 * (PI/180) '降到最低速度的角度 45 度

""""建立机械手连接
while 1
  if scan_event(in(0))>0 then      '输入 0 上升沿触发
    BASE(0,1,2,3,4)                '配置关节轴
    CONNFRAME(101,0,6,7,8,9,4)     '启动逆解连接。
    wait loaded                    '等待运动加载，此时会自动调整虚拟轴的
位置'运动控制
    ?"逆解模式"
  elseif scan_event(in(0))<0 then  '输入 0 下降沿触发
    base(6,7,8,9,4)                '选择虚拟轴号
    CONNREFRAME(101,0,0,1,2,3,4)   '启动正解连接
    wait loaded
    ?"正解模式"

```

```
endif  
wend
```

FRAME106--特殊 SCARA



此类型结构与 FRAME1 一致，区别只是大关节轴和小关节轴是关联的，大关节旋转会带动小关节转动，结构参数如下，其他参考 [FRAME1](#)。

结构参数设置

TABLE(tablenum,L1,L2,L1toL2,Pules1OneCircle,Pules2OneCircle,[Pules3OneCircle,L3] [,ZDis])	
Tablenum	存储转换参数的 TABLE 位置
L1	Xy 平面内，第一个关节轴的旋转中心到第二个关节轴的旋转中心的距离
L2	Xy 平面内，第二个关节轴的旋转中心到末端旋转轴的旋转中心的距离
L1toL2	大关节转动一圈，小关节变动的圈数
Pules1OneCircle	第一个关节轴旋转一圈的脉冲数
Pules2OneCircle	第二个关节轴旋转一圈的脉冲数
Pules3OneCircle	末端旋转轴旋转一圈的脉冲数
L3	Xy 平面内，末端旋转轴旋转中心到末端工作点的距离
ZDis	末端旋转轴旋转一圈，伸缩轴移动的距离

使用例程

```

""电机、机械手参数定义
dim L1      '大臂长度
dim L2      '小臂长度
dim L3      'X 方向偏移
dim L1to2   '大关节转动一圈，小关节变动的圈数
dim ZDis    '旋转轴一圈，z 轴移动距离

```

```
L1=10
L2=10
L3=0
L1to2=0.5
ZDis=0

dim u_m1 '电机 1 一圈脉冲数
dim u_m2 '电机 2 一圈脉冲数
dim u_mz '电机 z 一圈脉冲数
dim u_mv '电机 v 一圈脉冲数
u_m1=100
u_m2=10000
u_mz=10000
u_mv=10000

dim i_1 '关节 1 传动比
dim i_2 '关节 2 传动比

dim i_z '关节 z 传动比
dim i_v '关节 v 传动比
i_1=2
i_2=2
i_z=2
i_v=2

dim u_j1 '关节 1 实际一圈脉冲数
dim u_j2 '关节 2 实际一圈脉冲数
dim u_jz '关节 z 实际一圈脉冲数
dim u_jv '关节 v 实际一圈脉冲数
u_j1=u_m1*i_1
u_j2=u_m2*i_2
u_jz=u_mz*i_z
u_jv=u_mv*i_v

dim p_z 'z 轴螺距
p_z=1.5

""关节轴设置
BASE(0,1,2,3) '选择关节轴号
atype=1,1,1,1 '轴类型设为脉冲轴
UNITS=u_j1/360,u_j2/360,u_jv/360,u_jz/p_z '把 z 轴 units 设成 1mm 的脉冲
```


数

```

                                '其余轴设成 1°的脉冲数
DPOS=0,0,0,0                    '设置关节轴的位置，此处要根据实际情况来
修改。
speed=100,100,100,100          '速度参数设置
accel=1000,1000,1000,1000
decel=1000,1000,1000,1000
CLUTCH_RATE=0,0,0,0           '使用关节轴的速度和加速度限制

""虚拟轴设置
BASE(6,7,8,9)
ATYPE=0,0,0,0                  '设置为虚拟轴
TABLE(0,L1,L2,L1to2,u_j1,u_j2,u_jv,L3,ZDis)  '根据手册说明填写参数
UNITS=1000,1000 ,u_jv/360,1000      '运动精度，要提前设置，中
途不能变化
speed=200,200,200,200          '速度参数设置
accel=1000,1000,1000,1000
decel=1000,1000,1000,1000
""建立机械手连接
while 1
  if scan_event(in(0))>0 then    '输入 0 上升沿触发
    BASE(0,1,2,3)                '选择关节轴号
    CONNFRAME(106,0,6,7,8,9)     '第 6/7 轴作为虚拟的 XY 轴，
启动逆解连接。
    WAIT LOADED                  '等待运动加载，此时会自动调整虚拟
轴的位置。
    ?"逆解模式"
  elseif scan_event(in(0))<0 then '输入 0 下降沿触发
    BASE(6,7,8,9)                '选择虚拟轴号
    CONNREFRAME(106,0,0,1,2,3)   '第 0/1 轴作为关节轴，启动正
解连接。
    WAIT LOADED                  '等待运动加载。
    ?"正解模式"
  endif
Wend

```

FRAME107--特殊 SCARA



此类型结构与 FRAME1 一致，区别只是小关节轴和末端旋转轴是关联的，小关节旋转会带动末端旋转轴转动，结构参数如下，其他参考 [FRAME1](#)。

结构参数设置

TABLE(tablenum,L1,L2,L2toLV,Pules1OneCircle,Pules2OneCircle,[Pules3OneCircle,L3] [,ZDis])	
Tablenum	存储转换参数的 TABLE 位置
L1	Xy 平面内，第一个关节轴的旋转中心到第二个关节轴的旋转中心的距离
L2	Xy 平面内，第二个关节轴的旋转中心到末端旋转轴的旋转中心的距离
L2toLV	小关节转动一圈，旋转关节变动的圈数
Pules1OneCircle	第一个关节轴旋转一圈的脉冲数
Pules2OneCircle	第二个关节轴旋转一圈的脉冲数
Pules3OneCircle	末端旋转轴旋转一圈的脉冲数
L3	Xy 平面内，末端旋转轴旋转中心到末端工作点的距离
ZDis	末端旋转轴旋转一圈，伸缩轴移动的距离

使用例程

""电机、机械手参数定义

dim L1 '大臂长度

dim L2 '小臂长度

dim L3 'X 方向偏移

dim L2toV '小关节转动一圈，旋转关节变动的圈数

dim ZDis '旋转轴一圈，z 轴移动距离

```
L1=10
L2=10
L3=0
L2toV=1
ZDis=0
```

```
dim u_m1 '电机 1 一圈脉冲数
dim u_m2 '电机 2 一圈脉冲数
dim u_mz '电机 z 一圈脉冲数
dim u_mv '电机 v 一圈脉冲数
u_m1=10000
u_m2=10000
u_mz=10000
u_mv=10000
```

```
dim i_1 '关节 1 传动比
dim i_2 '关节 2 传动比
dim i_z '关节 z 传动比
dim i_v '关节 v 传动比
i_1=2
i_2=2
i_z=2
i_v=2
```

```
dim u_j1 '关节 1 实际一圈脉冲数
dim u_j2 '关节 2 实际一圈脉冲数
dim u_jz '关节 z 实际一圈脉冲数
dim u_jv '关节 v 实际一圈脉冲数
u_j1=u_m1*i_1
u_j2=u_m2*i_2
u_jz=u_mz*i_z
u_jv=u_mv*i_v
```

```
dim p_z 'z 轴螺距
p_z=1.5
```

```
""关节轴设置
BASE(0,1,2,3) '选择关节轴号
atype=1,1,1,1 '轴类型设为脉冲轴
UNITS=u_j1/360,u_j2/360,u_jv/360,u_jz/p_z '把 z 轴 units 设成 1mm 的脉冲
数
DPOS=0,0,0,0 '其余轴设成 1°的脉冲数
'设置关节轴的位置，此处要根据实际情况来
```

修改。

```
speed=100,100,100,100      '速度参数设置
accel=1000,1000,1000,1000
decel=1000,1000,1000,1000
CLUTCH_RATE=0,0,0,0      '使用关节轴的速度和加速度限制
```

""虚拟轴设置

```
BASE(6,7,8,9)
```

```
ATYPE=0,0,0,0          '设置为虚拟轴
```

```
TABLE(0,L1,L2,L2toV,u_j1,u_j2,u_jv,L3,ZDis)  '根据手册说明填写参数
```

```
UNITS=1000,1000 ,u_jv/360,1000          '运动精度，要提前设置，中途不能变化
```

```
speed=200,200,200,200      '速度参数设置
```

```
accel=1000,1000,1000,1000
```

```
decel=1000,1000,1000,1000
```

""建立机械手连接

```
while 1
```

```
  if scan_event(in(0))>0 then  '输入 0 上升沿触发
```

```
    BASE(0,1,2,3)              '选择关节轴号
```

```
    CONNFRAME(107,0,6,7,8,9)   '第 6/7 轴作为虚拟的 XY 轴，
```

启动逆解连接。

```
    WAIT LOADED                '等待运动加载，此时会自动调整虚拟轴的位置。
```

```
    ?"逆解模式"
```

```
  elseif scan_event(in(0))<0 then  '输入 0 下降沿触发
```

```
    BASE(6,7,8,9)              '选择虚拟轴号
```

```
    CONNREFRAME(107,0,0,1,2,3)  '第 0/1 轴作为关节轴，启动正
```

解连接。

```
    WAIT LOADED                '等待运动加载。
```

```
    ?"正解模式"
```

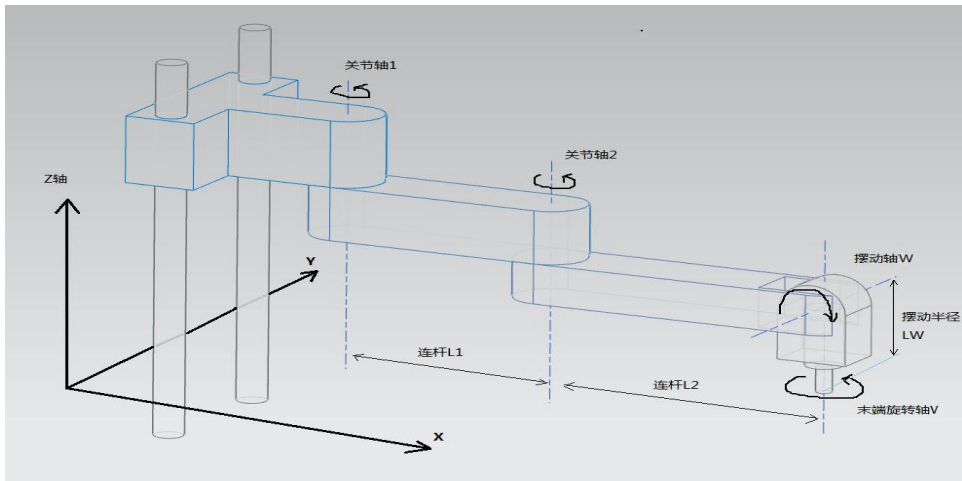
```
  endif
```

```
wend
```

FRAME108--特殊 SCARA

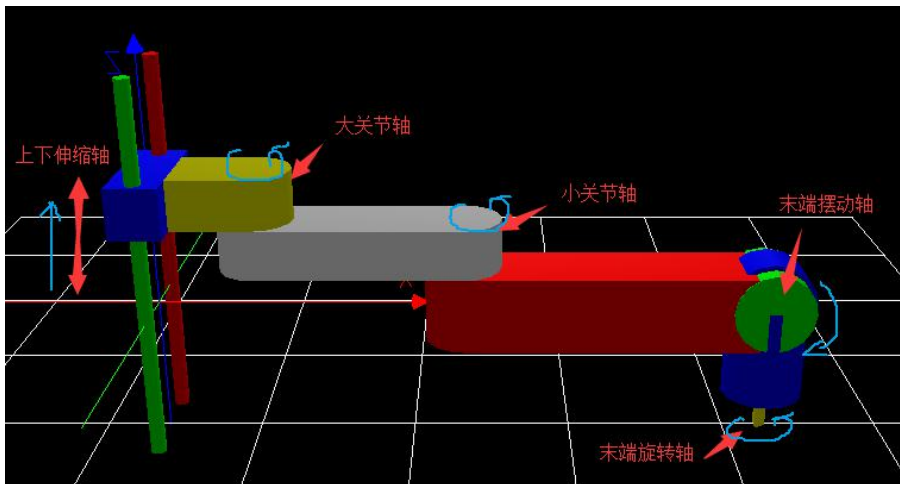
结构示意图

FRAME108 为特殊 SCARA，此结构 V 是手腕转动，W 是手腕摆动。



电机方向及角度范围

定义各关节电机正向如下图所示蓝色箭头所示。



	角度或移动范围
大关节轴	$(-2\pi, 2\pi)$
小关节轴	$(-2\pi, 2\pi)$
末端旋转轴	无限制
上下伸缩轴	机械结构限制
末端摆动轴	机械结构限制

轴顺序

实际机械手关节轴:	定义简写:
大关节轴电机	Axis_a
小关节轴电机	Axis_b
末端旋转轴电机	Axis_c
上下伸缩轴电机	Axis_d
末端摆动轴电机	Axis_e

直角坐标系虚拟轴:	定义简写:
平移轴 X	Viraxis_x
平移轴 Y	Viraxis_y
旋转轴 RZ	Viraxis_v
平移轴 Z	Viraxis_z
摆动轴 w	Viraxis_w

指令规定轴号填写次序如下:

逆解时:

```
BASE(Axis_a,Axis_b, Axis_d, Axis_c, Axis_e)
CONNFRAME(108,tablenum,Viraxis_x,Viraxis_y,Viraxis_z,Viraxis_v, Viraxis_w)
```

正解时:

```
BASE(Viraxis_x, Viraxis_y, Viraxis_z,Viraxis_v,Viraxis_w)
CONNREFRAME(108,tablenum, Axis_a,Axis_b, Axis_d,Axis_c,Axis_e)
```

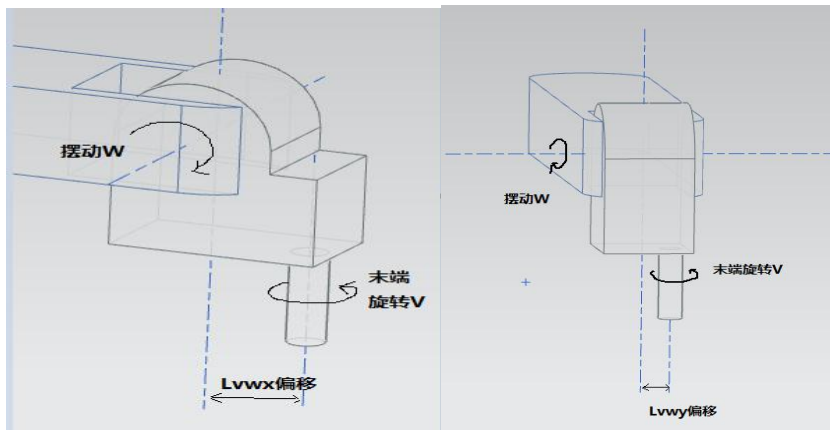
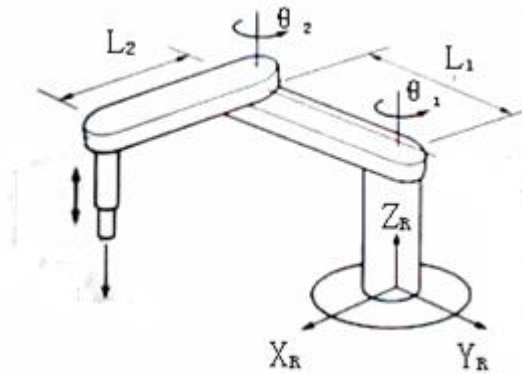
注意此类型中 Z 轴和旋转轴的次序进行了互换，而不是其他 SCARA 类型中先旋转轴后 Z 轴。

结构参数设置

建立机械手连接时，需要将机械结构参数依次填写到 table 数组中，参数说明如下。

TABLE(tablenum, L1, L2, Pules1OneCircle, Pules2OneCircle, PulesvOneCircle, Lw, PulesWOneCircle, Lvwx, Lvwy)	
Tablenum	存储转换参数的 TABLE 位置
L1	Xy 平面内，第一个关节轴的旋转中心到第二个关节轴的旋转中心的距离
L2	Xy 平面内，第二个关节轴的旋转中心到末端旋转轴的旋转中心的距离
Pules1OneCircle	第一个关节轴旋转一圈的脉冲数
Pules2OneCircle	第二个关节轴旋转一圈的脉冲数
PulesvOneCircle	末端旋转轴旋转一圈的脉冲数

Lw	摆动半径
PuleswOneCircle	末端摆动轴旋转一圈的脉冲数
LvwX	Xy 平面内，末端工作点与末端旋转轴旋转中心的 x 方向距离。LvwX 固定为 0
Lvwy	Xy 平面内，末端工作点与末端旋转轴旋转中心的 y 方向距离



电机参数设置

各轴的轴类型和脉冲当量要设置正确。

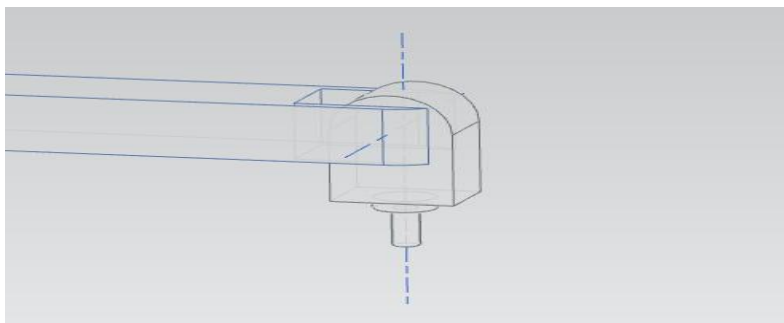
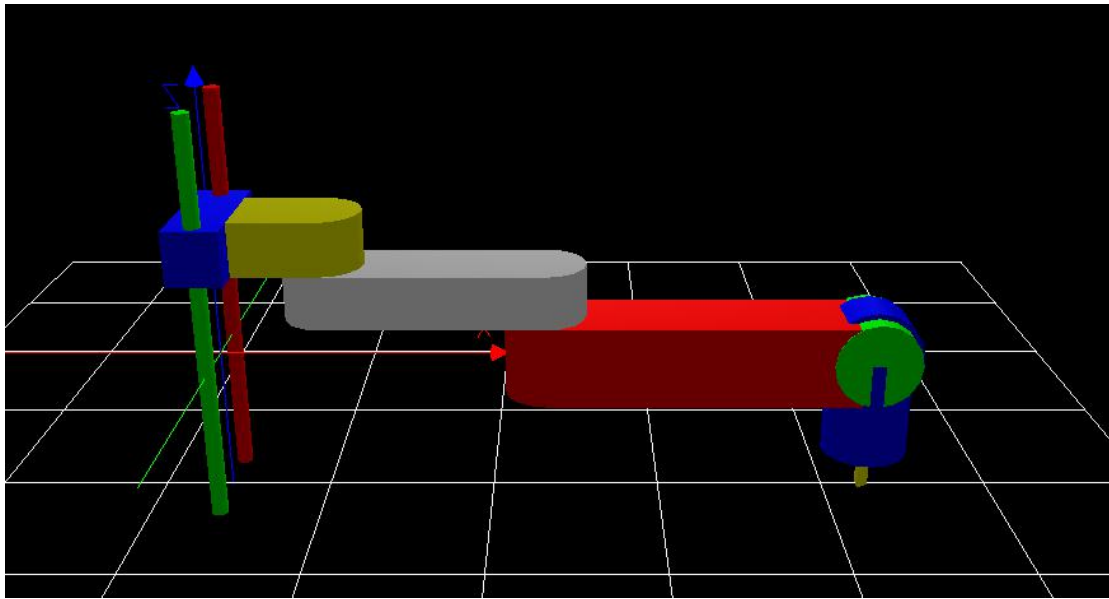
一个机械手的所有虚拟轴和关节轴的长度单位要求统一，一般都是 mm 单位，虚拟轴的一个 mm 的脉冲数一般建议设置为 1000，表示精度为小数点后 3 位。

	Atype	Units
Axis_a	1 或 4 或 7 或 65 或 50	Pules1OneCircle/360
Axis_b		Pules2OneCircle/360
Axis_c		PulesvOneCircle/360
Axis_d		伸缩轴移动 1MM 的脉冲数
Axis_e		PuleswOneCircle
Viraxis_x	0	1000
Viraxis_y		1000

Viraxis_v		PulesvOneCircle/360
Viraxis_z		1000
Viraxis_w		PuleswOneCircle

零点位置

两个关节轴的零点时成一条直线，此时指向虚拟 X 轴的正向。
当关节轴为零点的位置时，虚拟轴零点的坐标为 (L1+L2,0)。
手腕摆动轴竖直指向下方。



建立连接

正解模式

```
BASE(Viraxis_x, Viraxis_y, Viraxis_z, Viraxis_v, Viraxis_w)
CONNREFRAME(108, tablenum, Axis_a, Axis_b, Axis_d, Axis_c, Axis_e)
WAIT LOADED
```


建立成功后，虚拟轴 MTYPE 将显示为 34，IDLE 显示为 0。

此时只能操作关节轴在关节坐标系中运动，虚拟轴会自动计算末端工作点位于直角坐标系中的位置。

逆解模式

```
BASE(Axis_a,Axis_b, Axis_d, Axis_c, Axis_e)
CONNFRAME(108,tablenum,Viraxis_x,Viraxis_y,Viraxis_z,Viraxis_v, Viraxis_w)
WAIT LOADED
```

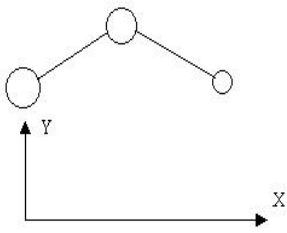
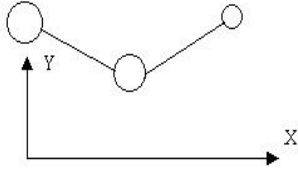
建立成功后，关节轴 MTYPE 将显示为 33，IDLE 显示为 0。

此时只能操作虚拟轴在直角坐标系中运动，关节轴会自动计算在关节坐标系中如何联合运动。

姿态说明

Scara 机械手有两个姿态，只可在正解模式下移动关节轴选择。

当前姿态通过指令 [FRAME_STATUS](#) 查询。

姿态	左手姿态	右手姿态
Frame_status	0	1
图形的表述		

使用例程

```
""电机、机械手参数定义
dim L1    '大臂长度
dim L2    '小臂长度
dim Lw    '摆动半径
dim Lvwx  '末端 x 偏移
dim Lvwy  '末端 y 偏移
L1=20
```

```
L2=30
Lw=20
Lvwx=0
Lvwy=0

dim u_m1 '电机 1 一圈脉冲数
dim u_m2 '电机 2 一圈脉冲数
dim u_mz '电机 z 一圈脉冲数
dim u_mv '旋转电机 v 一圈脉冲数
dim u_mw '摆动电机 w 一圈脉冲数
u_m1=10000
u_m2=10000
u_mz=10000
u_mv=10000
u_mw=10000

dim i_1 '关节 1 传动比
dim i_2 '关节 2 传动比
dim i_z '关节 z 传动比
dim i_v '关节 v 传动比
dim i_w '关节 w 传动比
i_1=2
i_2=2
i_z=2
i_v=2
i_w=2

dim u_j1 '关节 1 实际一圈脉冲数
dim u_j2 '关节 2 实际一圈脉冲数
dim u_jz '关节 z 实际一圈脉冲数
dim u_jv '关节 v 实际一圈脉冲数
dim u_jw '关节 w 实际一圈脉冲数
u_j1=u_m1*i_1
u_j2=u_m2*i_2
u_jz=u_mz*i_z
u_jv=u_mv*i_v
u_jw=u_mw*i_w

dim p_z 'z 轴螺距
p_z=10

""""关节轴设置
base(0,1,2,3,4) '选择关节轴 0,1 实际轴
```

```

rapidstop          '停止原来可能的运动
atype = 1,1,1,1,1  '轴类型为脉冲数
units=u_j1/360,u_j2/360,u_jz/p_z,u_jv/360,u_jw/360  'z轴把 units 设为每 mm 的
脉冲数，其他轴设为每°脉冲数
speed =300,300,300,300,300          '速度参数设置
accel = 1000,1000,1000,1000,1000    '设置关节轴的加速度限制
decel = 500,500,500,500,500         '设置关节轴的减速度
clutch_rate =0,0,0,0,0              '使用关节轴的速度和加速度限制
DPOS=0,0,0,0,0                      '先手动调整位置，再坐标清零

""""虚拟轴设置
base(6,7,8,9,10)
atype = 0,0,0,0,0    '设置为虚拟轴
TABLE(0,L1,L2,u_j1,u_j2,u_jv,Lw,u_jw,Lvwx,Lvwy)  '填写参数
UNITS = 1000,1000,1000,u_jv/360,u_jw/360        '提前设置，中途不能变
化,末端两个旋转轴保证 units 与关节轴一致

merge = on                      '打开连续插补
CORNER_MODE =2                  '启动拐角减速
DECEL_ANGLE = 15 * (PI/180)    '开始减速的角度 15 度
STOP_ANGLE = 45 * (PI/180)    '降到最低速度的角度 45 度

""""建立机械手连接
while 1
  if scan_event(in(0))>0 then    '输入 0 上升沿触发
    BASE(0,1,2,3,4)              '配置关节轴
    CONNFRAME(108,0,6,7,8,9,10) '启动逆解连接。
    wait loaded                  '等待运动加载，此时会自动调整虚拟轴的
位置'运动控制
    ?"逆解模式"
  elseif scan_event(in(0))<0 then '输入 0 下降沿触发
    base(6,7,8,9,10)            '选择虚拟轴号
    CONNREFRAME(108,0,0,1,2,3,4) '启动正解连接
    wait loaded
    ?"正解模式"
  endif
wend

```

并联机械手类

FRAME2/12/14--标准 DELTA

结构示意图



FRAME2 为 3 轴模式，没有末端旋转轴。

FRAME12 为 4 轴模式，有末端旋转轴，使用关节轴号操作旋转轴。

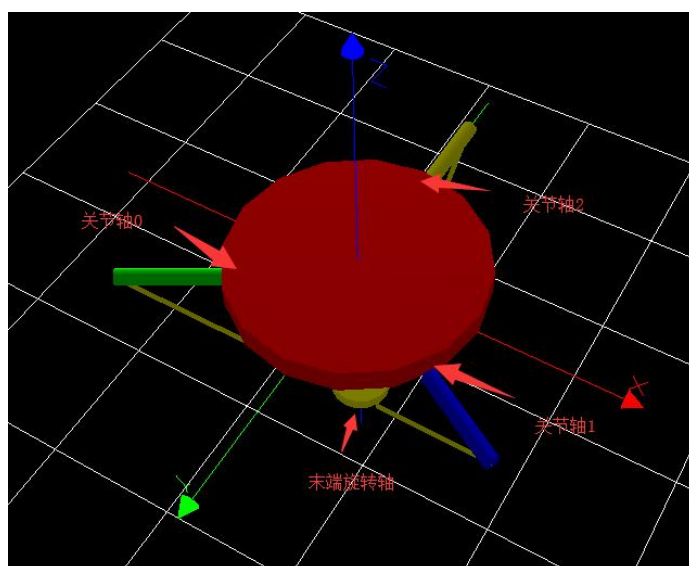
FRAME14 为 4 轴模式，有末端旋转轴，使用虚拟轴号操作旋转轴。

带“R”后缀的控制器支持。

电机方向及角度范围

定义 3 个关节轴向下旋转时为正向。

末端旋转轴逆时针旋转为正（俯视）。



	角度或移动范围
关节轴 0	机械结构限制
关节轴 1	
关节轴 2	
末端旋转轴	无限制

轴顺序

实际机械手关节轴:	定义简写:
关节轴 0 电机	Axis_a
关节轴 1 电机	Axis_b
关节轴 2 电机	Axis_c
末端旋转轴电机	Axis_d (FRAME12/14)

直角坐标系虚拟轴:	定义简写:
平移轴 X	Viraxis_x
平移轴 Y	Viraxis_y
平移轴 Z	Viraxis_z
旋转轴 RZ	Viraxis_v (FRAME14)

指令规定轴号填写次序如下:

逆解时:

<p>FRAME2/12 类型 BASE(Axis_a,Axis_b, Axis_c[,Axis_d]) CONNFRAME(2/12,tablenum, Viraxis_x, Viraxis_y, Viraxis_z[,Axis_d])</p> <p>FRAME14 类型 BASE(Axis_a,Axis_b, Axis_c,Axis_d) CONNFRAME(14,tablenum, Viraxis_x, Viraxis_y, Viraxis_z, Viraxis_v)</p>

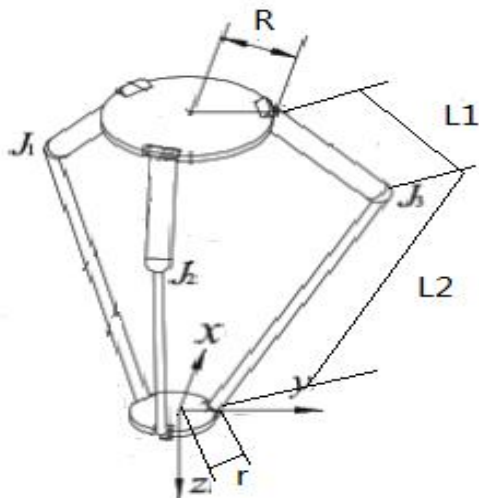
正解时:

<p>FRAME2/12 类型 BASE(Viraxis_x, Viraxis_y, Viraxis_z[,Axis_d]) CONNREFRAME(2/12,tablenum, Axis_a,Axis_b, Axis_c[,Axis_d])</p> <p>FRAME14 类型 BASE(Viraxis_x, Viraxis_y, Viraxis_z, Viraxis_v) CONNREFRAME(14,tablenum, Axis_a,Axis_b, Axis_c,Axis_d)</p>

结构参数设置

建立机械手连接时，需要将机械结构参数依次填写到 table 数组中，参数说明如下。

TABLE(tablenum,R,r,L1,L2,Pules1OneCircle,Pules2OneCircle,Pules3OneCircle,Lx,Ly,Lz,[Pules4OneCircle])	
Tablenum	存储转换参数的 TABLE 位置
R	上间距半径
r	下间距半径
L1	上杆长度
L2	下杆长度
Pules1OneCircle	第一个关节轴旋转一圈的脉冲数
Pules2OneCircle	第二个关节轴旋转一圈的脉冲数
Pules3OneCircle	第三个关节轴旋转一圈的脉冲数
Lx	末端与下面中心点的 X 偏移
Ly	末端与下面中心点的 Y 偏移
Lz	末端与下面中心点的 Z 偏移
Pules4OneCircle	末端旋转轴旋转一圈的脉冲数



电机参数设置

各轴的轴类型和脉冲当量要设置正确。

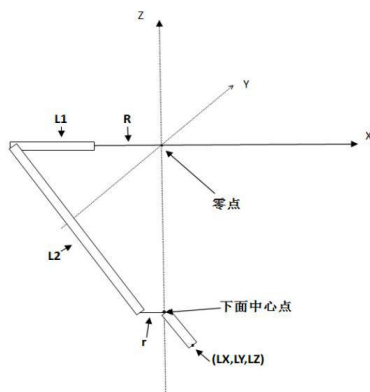
一个机械手的所有虚拟轴和关节轴的长度单位要求统一，一般都是 mm 单位。虚拟轴的一个 mm 的脉冲数一般建议设置为 1000，表示精度为小数点后 3 位。

	Atype	Units
--	-------	-------

Axis_a	1 或 4 或 7 或 65 或 50	Pules1OneCircle/360
Axis_b		Pules2OneCircle/360
Axis_c		Pules3OneCircle/360
Axis_d		Pules4OneCircle/360
Viraxis_x	0	1000
Viraxis_y		1000
Viraxis_z		1000
Viraxis_v		Pules4OneCircle/360

零点位置

- 1) 各关节轴的连杆 L1 都处于水平位置时，认为是关节零点位置。
- 2) 此时轴 0 和轴 1 连线方向为直角坐标系 X 方向，直角坐标系零点位于连杆 L1 水平面的中心。



直角坐标系

建立连接

正解模式

FRAME2/12 类型

BASE(Viraxis_x, Viraxis_y, Viraxis_z[,Axis_d])

CONNREFRAME(2/12,tablenum, Axis_a,Axis_b, Axis_c[,Axis_d])

WAIT LOADED

FRAME14 类型

BASE(Viraxis_x, Viraxis_y, Viraxis_z, Viraxis_v)

CONNREFRAME(14,tablenum, Axis_a,Axis_b, Axis_c,Axis_d)

WAIT LOADED

建立成功后，虚拟轴 MTYPE 将显示为 34，IDLE 显示为 0。

此时只能操作关节轴在关节坐标系中运动，虚拟轴会自动计算末端工作点位于直角坐标系中的位置。

逆解模式

FRAME2/12 类型

```
BASE(Axis_a,Axis_b,Axis_c[,Axis_d])
```

```
CONNFRAME(2/12,tablenum, Viraxis_x, Viraxis_y, Viraxis_z[,Axis_d])
```

```
WAIT LOADED
```

FRAME14 类型

```
BASE(Axis_a,Axis_b,Axis_c,Axis_d)
```

```
CONNFRAME(14,tablenum, Viraxis_x, Viraxis_y, Viraxis_z, Viraxis_v)
```

```
WAIT LOADED
```

建立成功后，关节轴 MTYPE 将显示为 33，IDLE 显示为 0。

此时只能操作虚拟轴在直角坐标系中运动，关节轴会自动计算在关节坐标系中如何联合运动。

使用例程

此例程使用 FRAME12 类型，若实际为 FRAME2 类型，需要去掉末端旋转轴；若实际为 FRAME14 类型，需要将旋转轴号用虚拟轴替换。

```

''''''''''电机、机械手参数定义
global R1      '上间距半径
global r2      '下间距半径
global L1      '上杆长度
global L2      '下杆长度
R1 = 40
r2 = 10
L1 = 32
L2 = 85

global u_m1    '电机 1 一圈脉冲数
global u_m2    '电机 2 一圈脉冲数
global u_m3    '电机 3 一圈脉冲数
global u_m4    '电机 4 一圈脉冲数
u_m1=10000
u_m2=10000
u_m3=10000
u_m4=10000

global i_1     '关节 1 传动比

```



```
while 1
  if scan_event(in(0))>0 then      '输入 0 上升沿触发
    BASE(0,1,2,3)                  '配置关节轴
    CONNFRAME(12,0,6,7,8,3) '第 6/7/8/9 轴为虚拟的 XYZW 轴，启动逆解连接。
  wait loaded                      '等待运动加载，此时会自动调整虚拟轴的位置
  ?"逆解模式"
  elseif scan_event(in(0))<0 then  '输入 0 下降沿触发
    base(6,7,8,3)                  '选择虚拟轴号
    CONNREFRAME(12,0,0,1,2,3)      '第 0/1/2/3 轴为关节轴，启动正解连接
  wait loaded
  ?"正解模式"
  endif
wend
```

FRAME102--2 轴 DELTA

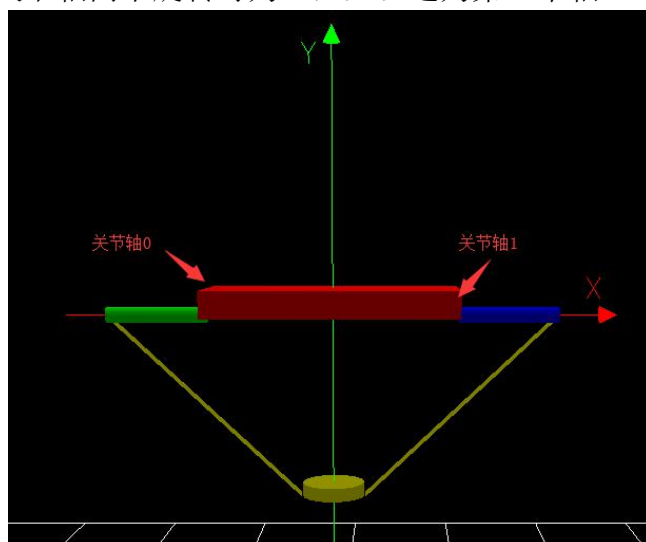
结构示意图



带“R”后缀的控制器支持。

电机方向及角度范围

定义两关节轴向下旋转时为正，且左边为第一个轴。



	角度或移动范围
关节轴 0	机械结构限制
关节轴 1	

轴顺序

实际机械手关节轴:	定义简写:
关节轴 0 电机	Axis_a
关节轴 1 电机	Axis_b

直角坐标系虚拟轴:	定义简写:
平移轴 X	Viraxis_x
平移轴 Y	Viraxis_y

指令规定轴号填写次序如下:

逆解时:

```
BASE(Axis_a,Axis_b)
CONNFRAME(102,tablenum, Viraxis_x, Viraxis_y)
```

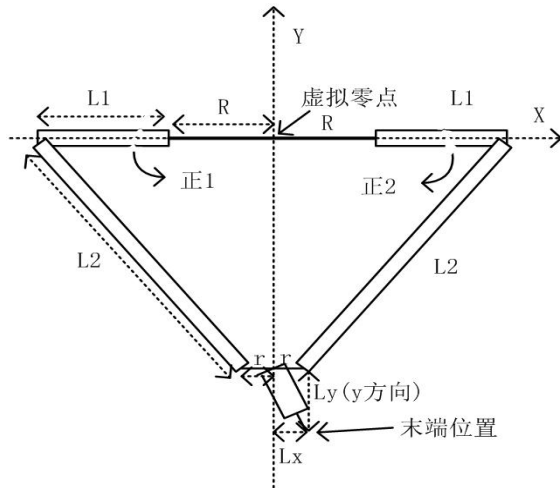
正解时:

```
BASE(Viraxis_x, Viraxis_y)
CONNREFRAME(102,tablenum, Axis_a,Axis_b)
```

结构参数设置

建立机械手连接时，需要将机械结构参数依次填写到 table 数组中，参数说明如下。

TABLE(tablenum, R, r, L1,L2, Pules1OneCircle,Pules2OneCircle, Lx, Ly))	
Tablenum	存储转换参数的 TABLE 位置
R	上间距半径
r	下间距半径
L1	上杆长度
L2	下杆长度
Pules1OneCircle	第一个关节轴旋转一圈的脉冲数
Pules2OneCircle	第二个关节轴旋转一圈的脉冲数
Lx	末端与下面中心点的 X 偏移
Ly	末端与下面中心点的 Y 偏移



电机参数设置

各轴的轴类型和脉冲当量要设置正确。

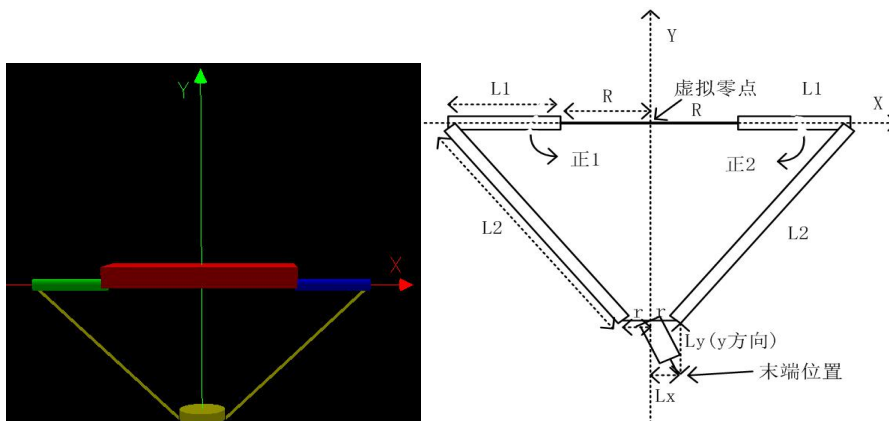
一个机械手的所有虚拟轴和关节轴的长度单位要求统一，一般都是 mm 单位。虚拟轴的一个 mm 的脉冲数一般建议设置为 1000，表示精度为小数点后 3 位。

	Atype	Units
Axis_a	1 或 4 或 7 或 65 或 50	Pules1OneCircle/360
Axis_b		Pules2OneCircle/360
Viraxis_x	0	1000
Viraxis_y		1000

零点位置

两关节轴连杆 L1 水平时为零点，向下转为正。

此时直角坐标系如下图所示。



建立连接

正解模式

```
BASE(Viraxis_x, Viraxis_y)
CONNREFRAME(102,tablenum, Axis_a,Axis_b)
WAIT LOADED
```

建立成功后，虚拟轴 MTYPE 将显示为 34，IDLE 显示为 0。

此时只能操作关节轴在关节坐标系中运动，虚拟轴会自动计算末端工作点位于直角坐标系中的位置。

逆解模式

```
BASE(Axis_a,Axis_b)
CONNFRAME(102,tablenum, Viraxis_x, Viraxis_y)
WAIT LOADED
```

建立成功后，关节轴 MTYPE 将显示为 33，IDLE 显示为 0。

此时只能操作虚拟轴在直角坐标系中运动，关节轴会自动计算在关节坐标系中如何联合运动。

使用例程

```
""""电机，机械手参数定义
dim R1    '上间距半径
dim r2    '下间距半径
dim L1    '上杆长度
dim L2    '上杆长度
dim Lx    '末端 X 偏移
dim Ly    '末端 Y 偏移
R1 = 40
r2 = 10
L1 = 32
L2 = 85
Lx = 0
Ly = 0

dim u_m1  '电机 1 一圈脉冲数
dim u_m2  '电机 2 一圈脉冲数
u_m1=10000
u_m2=10000
```

```

dim i_1      '关节 1 传动比
dim i_2      '关节 2 传动比
i_1=2
i_2=2

dim u_j1     '关节 1 实际一圈脉冲数
dim u_j2     '关节 2 实际一圈脉冲数
u_j1=u_m1*i_1
u_j2=u_m2*i_2

''''''''关节轴设置
base(0,1)    '选择关节轴 0,1 实际轴
rapidstop    '停止原来可能的运动
atype = 1,1  '轴类型为脉冲数
units=u_j1/360,u_j2/360 '把 units 设为每° 的脉冲数
dpos=0,0     '设置关节轴的位置，此处要根据实际情况来修改
creep=10,10  '轴回零时低速度
speed =300,300 '速度参数设置
accel = 1000,1000 '设置关节轴的加速度限制
decel = 500,500 '设置关节轴的减速度
clutch_rate =0,0 '使用关节轴的速度和加速度限制

''''''''虚拟轴设置
base(6,7)
atype = 0,0  '设置为虚拟轴
TABLE(0, R1,r2,L1,L2,u_j1,u_j2,Lx,Ly) '填写参数
UNITS = 1000,1000 '此脉冲当量应提前设置，中途不能变化

merge = on '打开连续插补
CORNER_MODE =2 '启动拐角减速
DECEL_ANGLE = 15 * (PI/180) '开始减速的角度 15 度
STOP_ANGLE = 45 * (PI/180) '降到最低速度的角度 45 度

''''''建立机械手连接
while 1
  if scan_event(in(0))>0 then '输入 0 上升沿触发
    BASE(0,1) '配置关节轴
    CONNFRAME(102,0,6,7) '第 6/7/8/9 轴为虚拟的 XYZW 轴，启动逆解
    连接。
    wait loaded '等待运动加载，此时会自动调整虚拟轴的位置'运动
    控制

```

```
    ?"逆解模式"  
elseif scan_event(in(0))<0 then    '输入 0 下降沿触发  
    base(6,7)                        '选择虚拟轴号  
    CONNREFRAME(102,0,0,1) '第 0/1 轴为关节轴，启动正解连接  
    wait loaded  
    ?"正解模式"  
endif  
wend
```


FRAME13--垂直蜘蛛手

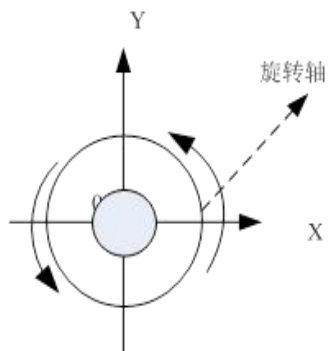
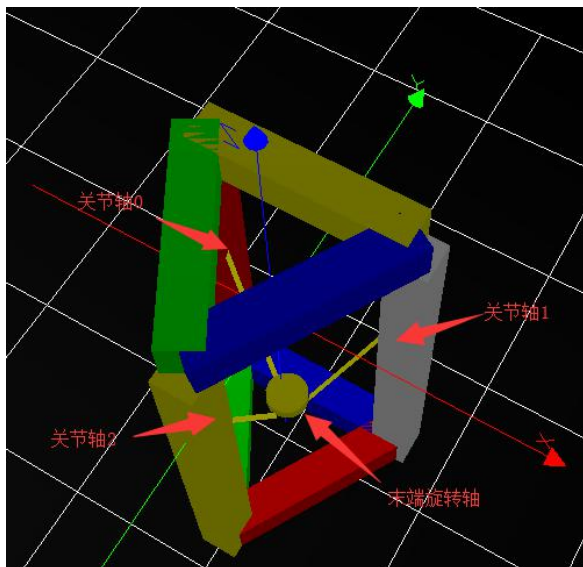
结构示意图



带“R”后缀的控制器支持。

电机方向及角度范围

定义3个关节轴向上运动时为正向。
末端旋转轴逆时针转动为正向（俯视）。



	角度或移动范围
关节轴 0	机械结构限制
关节轴 1	

关节轴 2	
末端旋转轴	无限制

轴顺序

实际机械手关节轴:	定义简写:
关节轴 0 电机	Axis_a
关节轴 1 电机	Axis_b
关节轴 2 电机	Axis_c
末端旋转轴电机	Axis_d

直角坐标系虚拟轴:	定义简写:
平移轴 X	Viraxis_x
平移轴 Y	Viraxis_y
平移轴 Z	Viraxis_z
旋转轴 RZ	Viraxis_v

指令规定轴号填写次序如下:

逆解时:

```
BASE(Axis_a,Axis_b, Axis_c[,Axis_d])
CONNFRAME(13,tablenum, Viraxis_x, Viraxis_y, Viraxis_z[,Viraxis_v])
```

正解时:

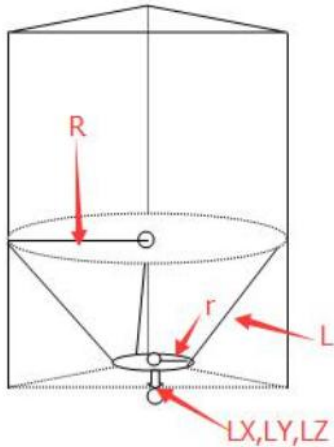
```
BASE(Viraxis_x, Viraxis_y, Viraxis_z[,Viraxis_v])
CONNREFRAME(13,tablenum, Axis_a,Axis_b, Axis_c[,Axis_d])
```

结构参数设置

建立机械手连接时，需要将机械结构参数依次填写到 table 数组中，参数说明如下。

TABLE(tablenum,R,r,L ,LX, LY, LZ, [PULSE4ONECIRCLE])	
Tablenum	存储转换参数的 TABLE 位置
R	上间距半径
r	下间距半径
L	连杆的长度
LX	末端点到小外切圆圆心的 X 的偏移(距离)
LY	末端点到小外切圆圆心的 Y 的偏移(距离)

LZ	末端点到小外切圆圆心的 Z 的偏移(距离)
PULSE4ONECIRCLE	末端旋转轴旋转一圈的脉冲数



电机参数设置

各轴的轴类型和脉冲当量要设置正确。

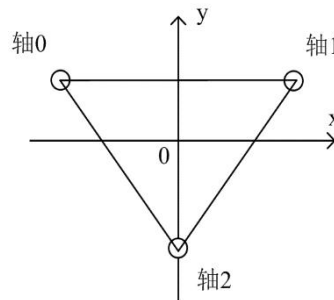
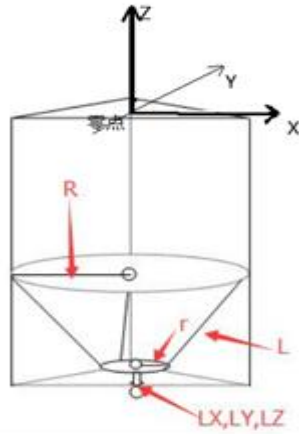
一个机械手的所有虚拟轴和关节轴的长度单位要求统一，一般都是 mm 单位。虚拟轴的一个 mm 的脉冲数一般建议设置为 1000，表示精度为小数点后 3 位。

	Atype	Units
Axis_a	1 或 4 或 7 或 65 或 50	每 mm 脉冲数
Axis_b		每 mm 脉冲数
Axis_c		每 mm 脉冲数
Axis_d		Pules4OneCircle/360
Viraxis_x	0	1000
Viraxis_y		1000
Viraxis_z		1000
Viraxis_v		Pules4OneCircle/360

零点位置

定义三个关节轴移动到顶端时为关节轴零点（此时应处于同一平面）。

此时轴 0 和轴 1 连线方向为直角坐标系 X 方向，直角坐标系零点位于机械结构顶端。



建立连接

正解模式

```
BASE(Viraxis_x, Viraxis_y, Viraxis_z[,Viraxis_v])
CONNREFRAME(13,tablenum, Axis_a,Axis_b, Axis_c[,Axis_d])
WAIT LOADED
```

建立成功后，虚拟轴 MTYPE 将显示为 34，IDLE 显示为 0。

此时只能操作关节轴在关节坐标系中运动，虚拟轴会自动计算末端工作点位于直角坐标系中的位置。

逆解模式

```
BASE(Axis_a,Axis_b, Axis_c[,Axis_d])
CONNFRAME(13,tablenum, Viraxis_x, Viraxis_y, Viraxis_z[,Viraxis_v])
WAIT LOADED
```

建立成功后，关节轴 MTYPE 将显示为 33，IDLE 显示为 0。

此时只能操作虚拟轴在直角坐标系中运动，关节轴会自动计算在关节坐标系中如何联合运动。

使用例程

```
'=====机械参数、电机参数定义=====
dim axis_x      '底座关节轴
dim axis_y      '大臂关节轴
dim axis_z      '小臂关节轴
dim axis_u      '末端旋转轴
```

```

axis_x=0
axis_y=1
axis_z=2
axis_u=3

dim AXIS_VIRX      '虚拟 X 轴
dim AXIS_VIRY      '虚拟 Y 轴
dim AXIS_VIRZ      '虚拟 Z 轴
dim AXIS_VIRU      '虚拟 U 轴
AXIS_VIRX=6
AXIS_VIRY=7
AXIS_VIRZ=8
AXIS_VIRU=9

dim R              '三个关节轴组成的大外切圆的半径
dim r              '三个关节轴组成的小外切圆的半径
dim L              ' 连杆的长度
dim LX            '末端点到小外切圆圆心的 X 轴的偏移(距离)
dim LY            '末端点到小外切圆圆心的 Y 轴的偏移(距离)
dim LZ            '末端点到小外切圆圆心的 Z 轴的偏移(距离)

R=138
r=24.5
L=213
LX=0
LY=0
LZ=0

dim u_m1          '电机每 mm 脉冲数
dim u_m2          '电机每 mm 脉冲数
dim u_m3          '电机每 mm 脉冲数
dim u_m4          '电机 4 一圈脉冲数
u_m1=10000
u_m2=10000
u_m3=10000
u_m4=10000

dim i_1           '关节 1 传动比
dim i_2           '关节 2 传动比
dim i_3           '关节 3 传动比
dim i_4           '关节 4 传动比
i_1=2

```

```

i_2=2
i_3=2
i_4=2

dim u_j1    '关节 1 每 mm 脉冲数
dim u_j2    '关节 2 每 mm 脉冲数
dim u_j3    '关节 3 每 mm 脉冲数
dim u_j4    '关节 4 转一圈的电机脉冲数
u_j1=u_m1*i_1
u_j2=u_m2*i_2
u_j3=u_m3*i_3
u_j4=u_m4*i_4

'=====关节轴参数设置=====

base(axis_x,axis_y,axis_z,axis_u) '定义实际轴
rapidstop                          '停止原来可能的运动
units=u_j1,u_j2,u_j3,u_j4/360     '前三个轴脉冲以 mm 为单位，最后个轴的单
位为度
atype = 1,1,1,1                    '设置轴类型
dpos=0,0,0,0                       '设置关节轴的位置，此处要根据实际情
况来修改
clutch_rate =0,0,0,0               '使用关节轴的速度和加速度限制
creep=10,10,10,10                 '轴回零时低速度
speed =300,300,300,300             '设置关节轴的速度限制
accel = 1000,1000,1000,1000       '设置关节轴的加速度限制
decel = 500,500,500,500           '设置关节轴的减速度

'=====虚拟轴参数设置=====

base(AXIS_VIRX,AXIS_VIRY,AXIS_VIRZ,AXIS_VIRU) '虚拟轴
atype = 0,0,0,0                    '设置轴类型
TABLE(0,R,r,L,LX,LY,LZ,u_j4)      '0,0,0 末端点胶头到 X,Y,Z 偏移的距离
speed = 200,200,200,200
accel = 300,300,300,300
UNITS = 1000,1000,1000,u_j4/360   '此脉冲当量要提前设置，中途不能变化
merge = on                          '打开轴 0 的连续插补状态
CORNER_MODE =2+32                   '启动拐角减速加倒角
ZSMOOTH=10                          ' 倒角半径设置
DECEL_ANGLE = 15 * (PI/180)        '开始减速的角度 15 度
STOP_ANGLE = 45 * (PI/180)        '降到最低速度的角度 45 度
BASE(axis_x,axis_y,axis_z,axis_u) '配置蜘蛛手

```

'=====运动模式连接=====

WHILE 1

IF scan_event(in(0))>0 then '逆解运动

BASE(axis_x,axis_y,axis_z,axis_u)'配置蜘蛛手

dpos=0,0,0,0' 设置关节轴的位置， 此处要根据实际情况来修改

CONNFRAME(13,0,AXIS_VIRX,AXIS_VIRY,AXIS_VIRZ,AXIS_VIRU) '

第 6/7/8/3 轴作为虚拟的 XYZW 轴， 启动连接。

wait loaded '等待运动加载， 此时会自动调整虚拟轴的位置

?"逆解模式"

elseif scan_event(in(0))<0 then '正解运动,手动操作关节轴

base(AXIS_VIRX,AXIS_VIRY,AXIS_VIRZ,AXIS_VIRU) '选择虚拟

轴号

CONNREFRAME(13,0,axis_x,axis_y,axis_z,axis_u)'第 0/1/2/3 轴为关节轴， 启动正解连接

WAIT LOADED

?"正解模式"

END IF

WEND

码垛机械手类

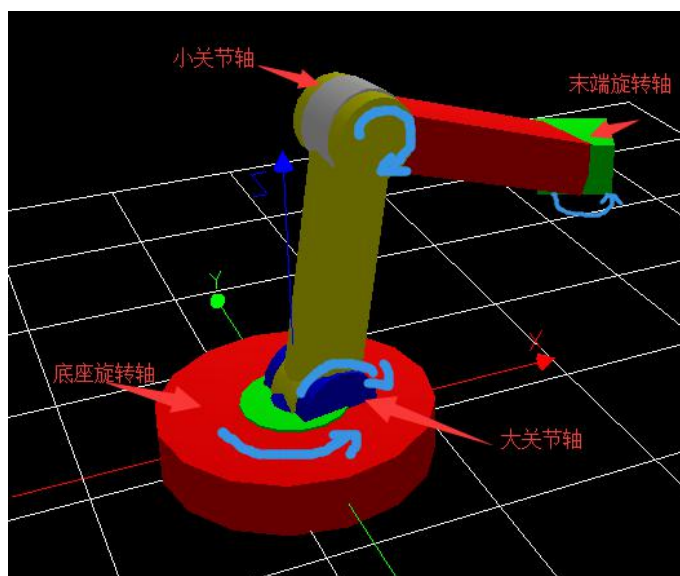
FRAME3--标准码垛

结构示意图



电机方向及角度范围

定义各关节轴的正向如下图蓝色箭头所示。



	角度或移动范围
底座旋转轴	无限制
大关节轴	机械结构限制
小关节轴	
末端旋转轴	无限制

轴顺序

实际机械手关节轴:	定义简写:
底座旋转轴电机	Axis_a
大关节轴电机	Axis_b
小关节轴电机	Axis_c
末端旋转轴电机	Axis_d

直角坐标系虚拟轴:	定义简写:
平移轴 X	Viraxis_x
平移轴 Y	Viraxis_y
平移轴 Z	Viraxis_z
旋转轴 RZ	Viraxis_v

指令规定轴号填写次序如下:

逆解时:

```
BASE(Axis_a,Axis_b, Axis_c[,Axis_d])
CONNFRAME(3,tablenum, Viraxis_x, Viraxis_y, Viraxis_z[,Viraxis_v])
```

正解时:

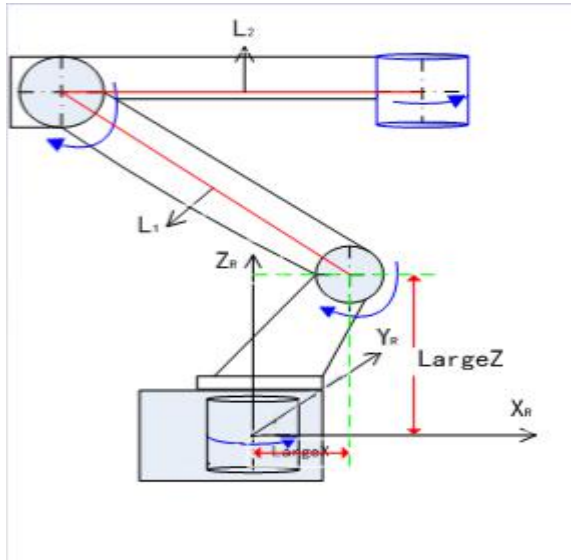
```
BASE(Viraxis_x, Viraxis_y, Viraxis_z[,Viraxis_v])
CONNREFRAME(3,tablenum, Axis_a,Axis_b, Axis_c[,Axis_d])
```

结构参数设置

建立机械手连接时，需要将机械结构参数依次填写到 table 数组中，参数说明如下。

TABLE(tablenum,LargeX,LargeZ,PulesLargeOneCircle,L1,L2,Pules1OneCircle,Pules2OneCircle,SmallX,SmallZ ,[PulesSmallOneCircle])	
Tablenum	存储转换参数的 TABLE 位置
LargeX	零点时，xz 平面内，底座旋转轴的旋转中心到大关节轴的旋转中心 x 方向的距离
LargeZ	零点时，xz 平面内，底座旋转轴的旋转中心到大关节轴的旋转中心 z 方向的距离
PulesLargeOneCircle	底座轴一圈脉冲数
L1	大臂轴长度

L2	小臂轴长度
Pules1OneCircle	大臂轴一圈脉冲数
Pules2OneCircle	小臂轴一圈脉冲数
SmallX	末端的水平偏移
SmallZ	末端的垂直偏移
PulesSmallOneCircle	末端轴的一圈脉冲数



电机参数设置

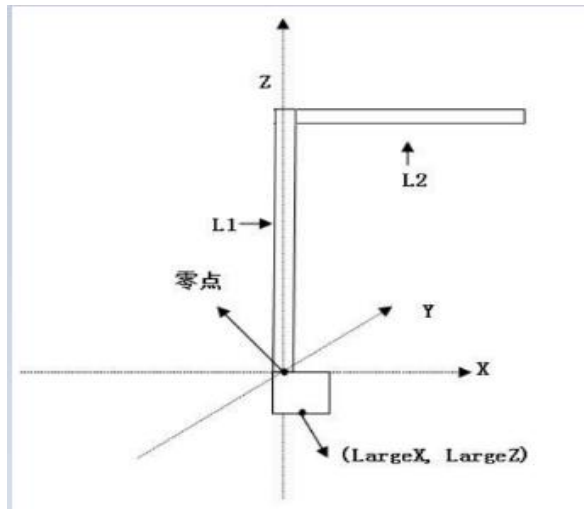
各轴的轴类型和脉冲当量要设置正确。

一个机械手的所有虚拟轴和关节轴的长度单位要求统一，一般都是 mm 单位。虚拟轴的一个 mm 的脉冲数一般建议设置为 1000，表示精度为小数点后 3 位。

	Atype	Units
Axis_a	1 或 4 或 7 或 65 或 50	PulesLargeOneCircle/360
Axis_b		Pules1OneCircle/360
Axis_c		Pules2OneCircle/360
Axis_d		PulesSmallOneCircle/360
Viraxis_x	0	1000
Viraxis_y		1000
Viraxis_z		1000
Viraxis_v		PulesSmallOneCircle/360

零点位置

定义大关节轴 0 点位置为大臂竖直；小关节轴 0 点位置为小臂水平。
直角坐标系零点位置为底座中心。



建立连接

正解模式

```
BASE(Viraxis_x, Viraxis_y, Viraxis_z[,Viraxis_v])
CONNREFRAME(3,tablenum, Axis_a,Axis_b, Axis_c[,Axis_d])
WAIT LOADED
```

建立成功后，虚拟轴 MTYPE 将显示为 34，IDLE 显示为 0。

此时只能操作关节轴在关节坐标系中运动，虚拟轴会自动计算末端工作点位于直角坐标系中的位置。

逆解模式

```
BASE(Axis_a,Axis_b, Axis_c[,Axis_d])
CONNFRAME(3,tablenum, Viraxis_x, Viraxis_y, Viraxis_z[,Viraxis_v])
WAIT LOADED
```

建立成功后，关节轴 MTYPE 将显示为 33，IDLE 显示为 0。

此时只能操作虚拟轴在直角坐标系中运动，关节轴会自动计算在关节坐标系中如何联合运动。

使用例程

```
.....机械手参数定义.....
dim L1          '大臂长度
dim L2          '小臂长度
dim smallX     '末端水平偏移
dim smallZ     '末端垂直偏移
```

```

dim LargeX      '底座水平方向偏移
dim LargeZ      '底座垂直方向偏移
L1=100
L2=100
smallX=10
smallZ=10
LargeX=10
LargeZ=10

dim u_mx        '电机 1 一圈脉冲数
dim u_my        '电机 2 一圈脉冲数
dim u_mz        '电机 3 一圈脉冲数
dim u_mw        '电机 4 一圈脉冲数
u_mx=10000
u_my=10000
u_mz=10000
u_mw=10000

dim i_x         '关节 1 传动比
dim i_y         '关节 2 传动比
dim i_z         '关节 3 传动比
dim i_w         '关节 4 传动比
i_x=1
i_y=1
i_z=1
i_w=1

dim u_jx        '关节 1 实际一圈脉冲数
dim u_jy        '关节 2 实际一圈脉冲数
dim u_jz        '关节 3 实际一圈脉冲数
dim u_jw        '关节 4 实际一圈脉冲数
u_jx=u_mx*i_x
u_jy=u_my*i_y
u_jz=u_mz*i_z
u_jw=u_mw*i_w

"关节轴设置"
base(0,1,2,3)      '关节轴轴号 0, 1, 2, 3
units=u_jx/360,u_jy/360,u_jz/360,u_jw/360      '以°为单位
dpos=0,0,0,0      '关节轴的初始位置 根据实际情况修改
speed=10000,10000,10000,10000 '关节轴的速度
accel=10000,10000,10000,10000 '关节轴的加速度

```

```

"设置启动拐角减速"
merge=on          '开启连续插补
corner_mode = 2   '启动拐角减速
decel_angle = 15 * (PI/180) '开始减速的角度 15 度
stop_angle = 45 * (PI/180) '降到最低速度的角度 45 度

"虚拟轴设置"
base(6,7,8,9)     '虚拟轴轴号 6, 7, 8, 9
atype=0,0,0,0     '设置为虚拟轴
table(0,LargeX,LargeZ,u_jx,L1,L2,u_jy,u_jz,smallX,smallZ,u_jw) '参数存储在
table0 开始的位置, 电机一圈 10000 个脉冲
units=1000,1000,1000,u_jw/360 '脉冲当量提前设置, 中途不能变化
merge = on       '连续缓冲开启
CORNER_MODE = 2+32 '设置启动拐角减速+倒角
ZSMOOTH=10      '倒角半径设置
DECEL_ANGLE = 15 * (PI/180) '开始减速的最小拐角
STOP_ANGLE = 45 * (PI/180) '减速到最低的减速拐角

"建立连接,运动"
while 1
  if scan_event(in(0))>0 then '输入 0 上升沿触发
    BASE(0,1,2,3)             '选择关节轴号
    CONNFRAME(3,0,6,7,8,9)   '启动逆解连接。
    WAIT LOADED              '等待运动加载, 此时会自动调整虚拟
    轴的位置。
    ?"逆解模式"
  elseif scan_event(in(0))<0 then '输入 0 下降沿触发
    BASE(6,7,8,9)           '选择虚拟轴号
    CONNREFRAME(3,0,0,1,2,3) '启动正解连接。
    WAIT LOADED             '等待运动加载。
    ?"正解模式"
  endif
wend

```

FRAME4--特殊码垛

比较 frame3_码垛，少了大小关节的旋转轴，控制在 xyz 平面运动是通过水平伸缩轴与垂直伸缩轴。

结构示意图



电机方向及角度范围

底座旋转轴逆时针转动为正向（俯视）。

大小伸缩轴使小臂抬起为正向。

	角度或移动范围
底座旋转轴	无限制
大关节伸缩轴	机械结构限制
小关节伸缩轴	
末端旋转轴	无限制

轴顺序

实际机械手关节轴：	定义简写：
底座旋转轴电机	Axis_a
大关节伸缩电机	Axis_b
小关节伸缩电机	Axis_c
末端旋转轴电机	Axis_d

直角坐标系虚拟轴：	定义简写：
平移轴 X	Viraxis_x
平移轴 Y	Viraxis_y
平移轴 Z	Viraxis_z

旋转轴 RZ

Viraxis_v

指令规定轴号填写次序如下：

逆解时：

```
BASE(Axis_a,Axis_b, Axis_c[,Axis_d])
CONNFRAME(4,tablenum, Viraxis_x, Viraxis_y, Viraxis_z[,Viraxis_v])
```

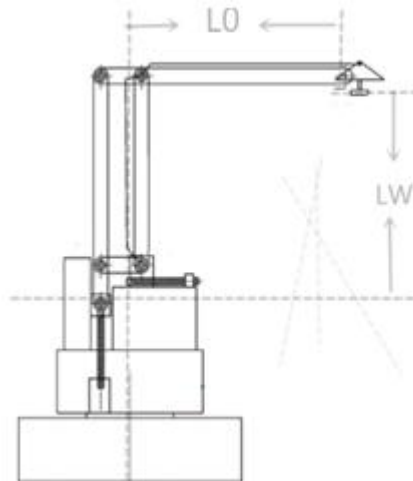
正解时：

```
BASE(Viraxis_x, Viraxis_y, Viraxis_z[,Viraxis_v])
CONNREFRAME(4,tablenum, Axis_a,Axis_b, Axis_c[,Axis_d])
```

结构参数设置

建立机械手连接时，需要将机械结构参数依次填写到 table 数组中，参数说明如下。

TABLE(tablenum, L0, LW, Pules1OneCircle, Pules4OneCircle, Lvx, Lvy)	
Tablenum	存储转换参数的 TABLE 位置
L0	Xy 平面内，零点时，轴 0 的旋转中心到末端旋转轴的旋转中心的 X 方向的距离
LW	xz 平面内，零点时，垂直伸缩轴的零点到末端工作点的 Z 方向的距离
Pules1OneCircle	轴 0 旋转一圈的脉冲数
Pules4OneCircle	轴 3 旋转一圈的脉冲数
Lvx	Xy 平面内，末端旋转轴旋转中心到末端工作点 X 方向的距离
Lvy	Xy 平面内，末端旋转轴旋转中心到末端工作点 y 方向的距离



电机参数设置

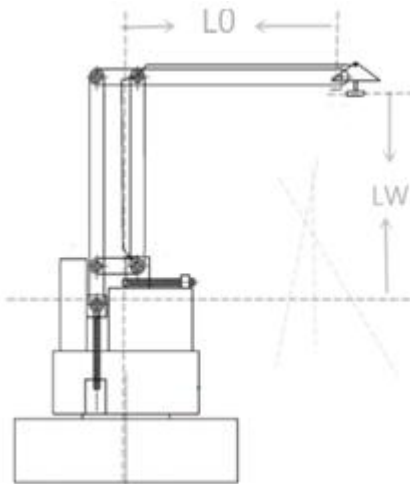
各轴的轴类型和脉冲当量要设置正确。

一个机械手的所有虚拟轴和关节轴的长度单位要求统一，一般都是 mm 单位。虚拟轴的一个 mm 的脉冲数一般建议设置为 1000，表示精度为小数点后 3 位。

	Atype	Units
Axis_a	1 或 4 或 7 或 65 或 50	Pules1OneCircle/360
Axis_b		每 mm 脉冲数
Axis_c		每 mm 脉冲数
Axis_d		Pules4OneCircle/360
Viraxis_x	0	1000
Viraxis_y		1000
Viraxis_z		1000
Viraxis_v		Pules4OneCircle/360

零点位置

大关节伸缩轴 0 点位置为大臂竖直时；小关节伸缩轴 0 点位置为小臂水平时。直角坐标系种，水平臂的指向为 x 的正方向。



建立连接

正解模式

```
BASE(Viraxis_x, Viraxis_y, Viraxis_z[,Viraxis_v])
CONNREFRAME(4,tablenum, Axis_a,Axis_b, Axis_c[,Axis_d])
WAIT LOADED
```

建立成功后，虚拟轴 MTYPE 将显示为 34，IDLE 显示为 0。

此时只能操作关节轴在关节坐标系中运动，虚拟轴会自动计算末端工作点位于直角坐标系中的位置。

逆解模式

```
BASE(Axis_a,Axis_b, Axis_c[,Axis_d])
CONNFRAME(4,tablenum, Viraxis_x, Viraxis_y, Viraxis_z[,Viraxis_v])
WAIT LOADED
```

建立成功后，关节轴 MTYPE 将显示为 33，IDLE 显示为 0。

此时只能操作虚拟轴在直角坐标系中运动，关节轴会自动计算在关节坐标系中如何联合运动。

使用例程

```
dim L0 '水平伸缩轴零点时的长度
dim Lw '垂直伸缩轴零点时的长度
dim Lvx
dim Lvy
L0=100
Lw=10
Lvx=9
Lvy=9

dim u_m1 '关节轴一 一圈脉冲数，大旋转轴
dim u_m2 '关节轴二 一圈脉冲数，伸缩轴
dim u_m3 '末端旋转轴一圈脉冲数
dim u_m4 '垂直伸缩轴一圈脉冲数
u_m1=10000
u_m2=10000
u_m3=10000
u_m4=10000

dim i_1 '关节 1 传动比
dim i_2 '关节 2 传动比
dim i_3 '末端旋转轴传动比
dim i_4 '垂直伸缩轴传动比
i_1=1
i_2=1
i_3=1
i_4=1

dim u_j1 '关节一实际一圈脉冲数
```

```

dim u_j2    '关节二实际一圈的脉冲数
dim u_j3    '末端旋转轴实际一圈脉冲数
dim u_j4    '垂直伸缩轴实际一圈脉冲数
u_j1=u_m1*i_1
u_j2=u_m2*i_2
u_j3=u_m3*i_3
u_j4=u_m4*i_4

dim p_2     '伸缩轴螺距
p_2=2

'=====关节轴设置=====

BASE(0,1,2,3)           '选择关节轴号
atype=1,1,1 ,1         '轴类型设为脉冲轴
UNITS=u_j1/360,u_j2/p_2,u_j4/p_2,u_j3/360      '把 units 设成每°的脉冲数，
其伸缩轴为一个 mm 的脉冲数

DPOS=0,0,0,0           '设置关节轴的位置，此处要根据实际情况来修改。
speed=100,100,100,100  '速度参数设置
accel=1000,1000,1000,1000 '加速度参数设置
decel=1000,1000,1000,1000 '减速度参数设置
CLUTCH_RATE=0,0,0,0    '使用关节轴的速度和加速度限制

'=====虚拟轴设置=====

BASE(6,7,8,9)
ATYPE=0,0,0,0          '设置为虚拟轴
UNITS=1000,1000,1000,u_j3/360 '此脉冲当量要提前设置，中途不能变化
speed=100,100,100,100  '速度参数设置
accel=1000,1000,1000,1000 '加速度参数设置
decel=1000,1000,1000,1000 '减速度参数设置
TABLE(0,L0,Lw,u_j1,u_j3,Lvx,Lvy) '根据手册说明填写参数
merge = on             '打开连续插补状态
CORNER_MODE =2+32     '启动拐角减速和倒角
ZSMOOTH=0             '倒角半径设置
DECEL_ANGLE = 15 * (PI/180) '开始减速的角度 15 度
STOP_ANGLE = 45 * (PI/180) '降到最低速度的角度 45 度
while 1
  if scan_event(in(0))>0 then '输入 0 上升沿触发
    BASE(0,1,2,3)           '选择关节轴号
    CONNFRAME(4,0,6,7,8,9)  '第 6/7 轴作为虚拟的 XY 轴，启动逆解连接。
    WAIT LOADED             '等待运动加载，此时会自动调整虚拟轴的

```

位置。

?"逆解模式"

elseif scan_event(in(0))<0 then '输入 0 下降沿触发

BASE(6,7,8,9) '选择虚拟轴号

CONNREFRAME(4,0,0,1,2,3) '第 0/1 轴作为关节轴，启动正解

连接。

WAIT LOADED '等待运动加载，手动操作实际轴

?"正解模式"

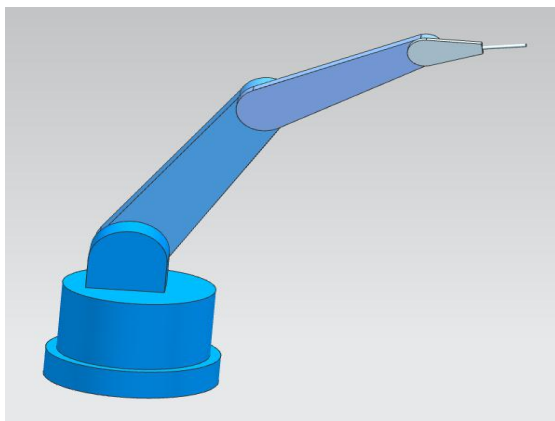
endif

wend

end

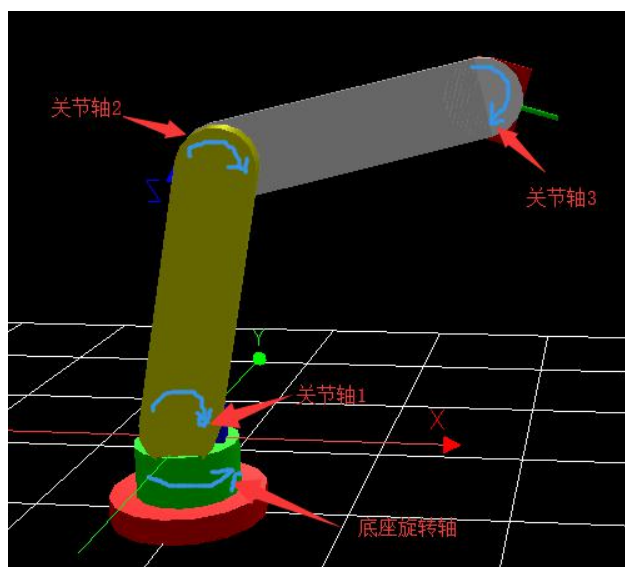
FRAME103--喷涂机械手

结构示意图



电机方向及角度范围

各关节电机正向如下图所示蓝色箭头所示。



	角度或移动范围
底座旋转轴	无限制
关节轴 1	机械结构限制
关节轴 2	
关节轴 3	

轴顺序

实际机械手关节轴:	定义简写:
底座旋转轴电机	Axis_a
关节轴 1 电机	Axis_b
关节轴 2 电机	Axis_c
关节轴 3 电机	Axis_d

直角坐标系虚拟轴:	定义简写:
平移轴 X	Viraxis_x
平移轴 Y	Viraxis_y
平移轴 Z	Viraxis_z
旋转轴 RY	Viraxis_v

指令规定轴号填写次序如下:

逆解时:

```
BASE(Axis_a,Axis_b, Axis_c,Axis_d)
CONNFRAME(103,tablenum, Viraxis_x, Viraxis_y, Viraxis_z,Viraxis_v)
```

正解时:

```
BASE(Viraxis_x, Viraxis_y, Viraxis_z,Viraxis_v)
CONNREFRAME(103,tablenum, Axis_a,Axis_b, Axis_c,Axis_d)
```

结构参数设置

建立机械手连接时，需要将机械结构参数依次填写到 table 数组中，参数说明如下。

TABLE(tablenum, LargeX, LargeZ, PulesLargeOneCircle, L1, L2, L3, Pules1OneCircle, Pules2OneCircle, Pules3OneCircle)	
Tablenum	存储转换参数的 TABLE 位置
LargeX	底座的水平方向偏移
LargeZ	底座的垂直方向偏移
PulesLargeOneCircle	底座旋转轴旋转一圈的脉冲数
L1	yz 平面内，大关节轴的旋转中心到小关节轴的旋转中心的距离
L2	yz 平面内，小关节轴的旋转中心到小小臂关节轴的旋转中心的距离
L3	yz 平面内，小小臂关节轴旋转中心到末端工作点的距离
Pules1OneCircle	大关节轴旋转一圈的脉冲数。

Pules2OneCircle	小关节轴旋转一圈的脉冲数
Pules3OneCircle	小小臂关节轴旋转一圈的脉冲数

电机参数设置

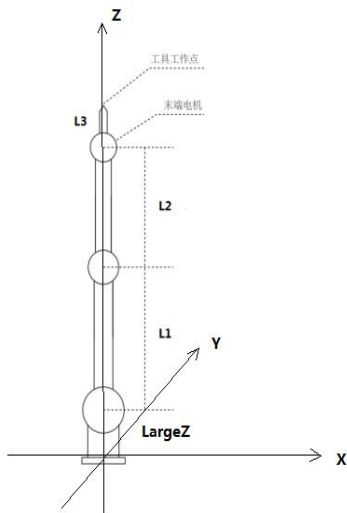
各轴的轴类型和脉冲当量要设置正确。

一个机械手的所有虚拟轴和关节轴的长度单位要求统一，一般都是 mm 单位。虚拟轴的一个 mm 的脉冲数一般建议设置为 1000，表示精度为小数点后 3 位。

	Atype	Units
Axis_a	1 或 4 或 7 或 65 或 50	Pules1OneCircle/360
Axis_b		Pules2OneCircle/360
Axis_c		Pules3OneCircle/360
Axis_d		Pules4OneCircle/360
Viraxis_x	0	1000
Viraxis_y		1000
Viraxis_z		1000
Viraxis_v		Pules4OneCircle/360

零点位置

机构指向正上方时为各关轴零点位置。



建立连接

正解模式

```

BASE(Viraxis_x, Viraxis_y, Viraxis_z, Viraxis_v)
CONNREFRAME(103, tablenum, Axis_a, Axis_b, Axis_c, Axis_d)
WAIT LOADED

```

建立成功后，虚拟轴 MTYPE 将显示为 34，IDLE 显示为 0。

此时只能操作关节轴在关节坐标系中运动，虚拟轴会自动计算末端工作点位于直角坐标系中的位置。

逆解模式

```

BASE(Axis_a, Axis_b, Axis_c, Axis_d)
CONNFRAME(103, tablenum, Viraxis_x, Viraxis_y, Viraxis_z, Viraxis_v)
WAIT LOADED

```

建立成功后，关节轴 MTYPE 将显示为 33，IDLE 显示为 0。

此时只能操作虚拟轴在直角坐标系中运动，关节轴会自动计算在关节坐标系中如何联合运动。

使用例程

""""""机械手参数定义""""""

```

dim L1          '大臂长度
dim L2          '小臂长度
dim L3          '小小臂长度
dim LargeX      '底座水平方向偏移
dim LargeZ      '底座垂直方向偏移
L1=100
L2=100
L3=30
LargeX=0
LargeZ=10

dim u_m1        '电机 1 一圈脉冲数
dim u_m2        '电机 2 一圈脉冲数
dim u_m3        '电机 3 一圈脉冲数
dim u_m4        '电机 4 一圈脉冲数
u_m1=10000
u_m2=10000
u_m3=10000
u_m4=10000

dim i_1         '关节 1 传动比
dim i_2         '关节 2 传动比

```

```

dim i_3      '关节 3 传动比
dim i_4      '关节 4 传动比
i_1=1
i_2=1
i_3=1
i_4=1

dim u_j1     '关节 1 一圈脉冲数
dim u_j2     '关节 2 一圈脉冲数
dim u_j3     '关节 3 一圈脉冲数
dim u_j4     '关节 4 一圈脉冲数
u_j1=u_m1*i_1
u_j2=u_m2*i_2
u_j3=u_m3*i_3
u_j4=u_m4*i_4

""""""关节轴设置""""""
base(0,1,2,3)      '关节轴轴号 0, 1, 2, 3
units=u_j1/360,u_j2/360,u_j3/360,u_j4/360  '以°为单位
dpos=0,0,0,0      '关节轴的初始位置 根据实际情况修改
speed=10000,10000,10000,10000      '关节轴的速度
accel=10000,10000,10000,10000      '关节轴的加速度
""""""虚拟轴设置""""""
base(6,7,8,9)     '虚拟轴轴号 5, 6, 7, 8
atype=0,0,0,0     '设置为虚拟轴
table(0,LargeX,LargeZ,u_j1,L1,L2,L3,u_j2,u_j3,u_j4) '参数存储在 table0 开始的
位置, 电机一圈 3600 个脉冲
units=1000,1000,1000,u_j4/360      '脉冲当量提前设置, 中途不能变化

""""""设置启动拐角减速""""""
merge=on          '开启连续插补
corner_mode = 2   '启动拐角减速
decel_angle = 15 * (PI/180)      '开始减速的角度 15 度
stop_angle = 45 * (PI/180)      '降到最低速度的角度 45 度
""""""

""""""建立连接""""""
while 1
  if scan_event(in(0))>0 then  '输入 0 上升沿触发
    BASE(0,1,2,3)              '选择关节轴号
    CONNFRAME(103,0,6,7,8,9)   '启动逆解连接。
    WAIT LOADED                '等待运动加载, 此时会自动调整虚拟轴的
位置。
    ?"逆解模式"
  elseif scan_event(in(0))<0 then  '输入 0 下降沿触发

```



```
BASE(6,7,8,9)           '选择虚拟轴号
CONNREFRAME(103,0,0,1,2,3) '启动正解连接。
WAIT LOADED             '等待运动加载。
?"正解模式"
endif

wend
```

FRAME25--5 关节机械手

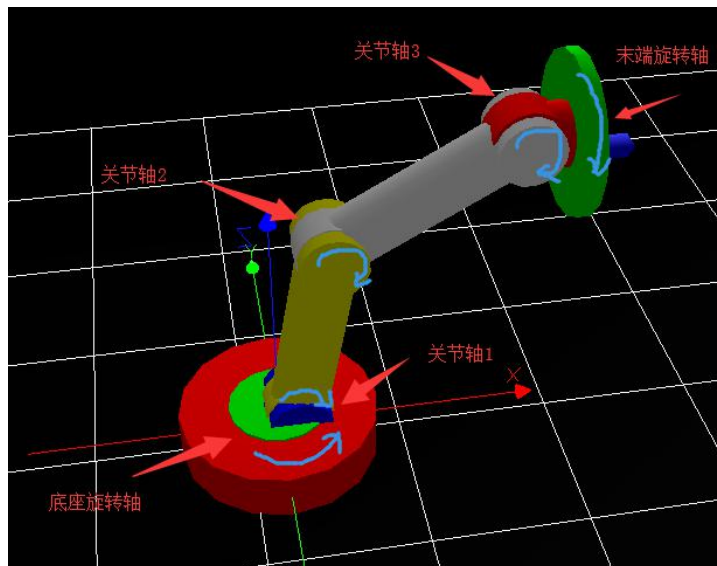
5 关节机械手，比 6 关节少一个中间转动轴,比 103 结构多一个末端旋转轴。

结构示意图



电机方向及角度范围

各关节电机正向如下图蓝色箭头所示。



	角度或移动范围
底座旋转轴	无限制
关节轴 1	机械结构限制
关节轴 2	
关节轴 3	
末端旋转轴	无限制

轴顺序

实际机械手关节轴:	定义简写:
底座旋转轴电机	Axis_a
关节轴 1 电机	Axis_b
关节轴 2 电机	Axis_c
关节轴 3 电机	Axis_d
末端旋转轴	Axis_e

直角坐标系虚拟轴:	定义简写:
平移轴 X	Viraxis_x
平移轴 Y	Viraxis_y
平移轴 Z	Viraxis_z
旋转轴 RY	Viraxis_w
旋转轴 RZ	Viraxis_v

指令规定轴号填写次序如下:

逆解时:

```
BASE(Axis_a,Axis_b,Axis_c,Axis_d,Axis_e)
CONNFRAME(25,tablenum,Viraxis_x,Viraxis_y,Viraxis_z,Viraxis_w,Viraxis_v)
```

正解时:

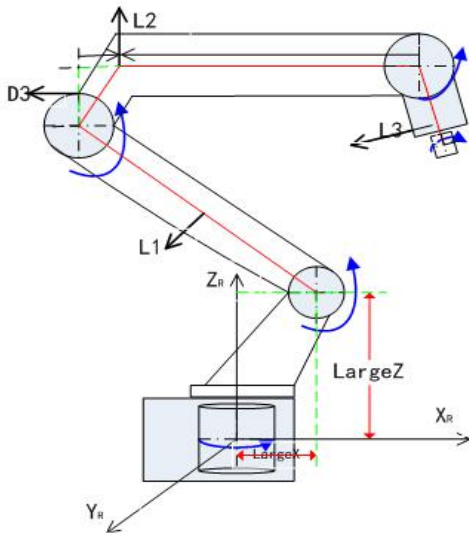
```
BASE(Viraxis_x, Viraxis_y, Viraxis_z, Viraxis_w, Viraxis_v)
CONNREFRAME(25,tablenum, Axis_a,Axis_b, Axis_c,Axis_d,Axis_e)
```

结构参数设置

建立机械手连接时，需要将机械结构参数依次填写到 table 数组中，参数说明如下。

TABLE(tablenum, LargeX, LargeZ, PulesLargeOneCircle, L1, L2, L3, Pules1OneCircle, Pules2OneCircle, Pules3OneCircle, Pules4OneCircle, Lx, Ly, D3)	
Tablenum	存储转换参数的 TABLE 位置
LargeX	LargeX
LargeZ	LargeZ
PulesLargeOneCircle	底座旋转轴旋转一圈的脉冲数
L1	大臂轴长度
L2	小臂轴长度
L3	小小臂轴长度
Pules1OneCircle	大关节轴旋转一圈的脉冲数
Pules2OneCircle	小关节轴旋转一圈的脉冲数

Pules3OneCircle	小关节轴旋转一圈的脉冲数
Pules4OneCircle	末端旋转轴旋转一圈的脉冲数
Lx	末端零点位置时 X 方向偏移
Ly	末端零点位置时 Y 方向偏移
D3	大臂小臂零点位置时的偏移



电机参数设置

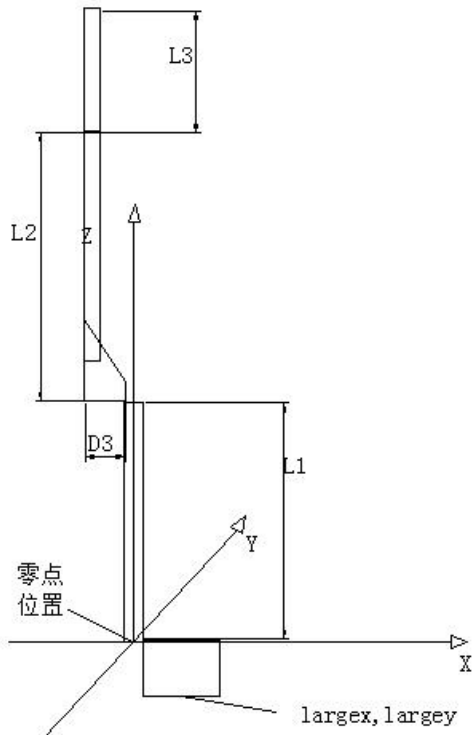
各轴的轴类型和脉冲当量要设置正确。

一个机械手的所有虚拟轴和关节轴的长度单位要求统一，一般都是 mm 单位。虚拟轴的一个 mm 的脉冲数一般建议设置为 1000，表示精度为小数点后 3 位。

	Atype	Units
Axis_a	1 或 4 或 7 或 65 或 50	PulesLargeOneCircle
Axis_b		Pules1OneCircle/360
Axis_c		Pules2OneCircle/360
Axis_d		Pules3OneCircle/360
Axis_e		Pules4OneCircle/360
Viraxis_x	0	1000
Viraxis_y		1000
Viraxis_z		1000
Viraxis_w		Pules3OneCircle/360
Viraxis_v		Pules4OneCircle/360

零点位置

机构指向正上方时为各关节零点位置。



建立连接

正解模式

```
BASE(Viraxis_x, Viraxis_y, Viraxis_z, Viraxis_w, Viraxis_v)
CONNREFRAME(25, tablenum, Axis_a, Axis_b, Axis_c, Axis_d, Axis_e)
WAIT LOADED
```

建立成功后，虚拟轴 MTYPE 将显示为 34，IDLE 显示为 0。

此时只能操作关节轴在关节坐标系中运动，虚拟轴会自动计算末端工作点位于直角坐标系中的位置。

逆解模式

```
BASE(Axis_a, Axis_b, Axis_c, Axis_d, Axis_e)
CONNFRAME(25, tablenum, Viraxis_x, Viraxis_y, Viraxis_z, Viraxis_w, Viraxis_v)
WAIT LOADED
```

建立成功后，关节轴 MTYPE 将显示为 33，IDLE 显示为 0。

此时只能操作虚拟轴在直角坐标系中运动，关节轴会自动计算在关节坐标系中如何联合运动。

姿态说明

当前姿态通过指令 [FRAME STATUS](#) 查询。
4 个，大旋转轴与 scara 两个不同的状态。

使用例程

```

"*****机械手参数定义*****"

dim L1           '大臂长度
dim L2           '小臂长度
dim L3           '小小臂长度
dim smalLX       '末端 X 偏移
dim smalLY       '末端 Y 偏移

dim units_x      '电机 1 一圈脉冲数
dim units_y      '电机 2 一圈脉冲数
dim units_z      '电机 3 一圈脉冲数
dim units_w      '电机 4 一圈脉冲数
dim units_s      '电机 5 一圈脉冲数

dim drive_ratio_x  '关节 1 传动比
dim drive_ratio_y  '关节 2 传动比
dim drive_ratio_z  '关节 3 传动比
dim drive_ratio_w  '关节 4 传动比
dim drive_ratio_s  '关节 5 传动比

dim LargeX        '底座水平方向偏移
dim LargeZ        '底座垂直方向偏移

dim D3            '零点时，大关节轴中心与小关节轴中心，x 正方向
                  的距离
*****

"*****变量赋值*****"
L1=100
L2=100
L3=100
smalLX=10
smalLY=0

```

```

units_x=10000
units_y=10000
units_z=10000
units_w=10000
units_s=10000

drive_ratio_x=1
drive_ratio_y=1
drive_ratio_z=1
drive_ratio_w=1
drive_ratio_s=1

LargeX=10
LargeZ=10

D3=0
.....

""""""关节轴设置""""""
base(0,1,2,3,4)          '关节轴轴号 0, 1, 2, 3, 4
units=units_x/drive_ratio_x/360,units_y/drive_ratio_y/360,units_z/drive_ratio_z/360
0,units_w/drive_ratio_w/360,units_y/drive_ratio_s/360      '以°为单位
dpos=0,0,0,0           '关节轴的初始位置 根据实际情况修改
speed=10000,10000,10000,10000,10000      '关节轴的速度
accel=10000,10000,10000,10000,10000      '关节轴的加速度

      '参数存储在 table0 开始的位置, 电机一圈 3600 个脉冲
.....

""""""虚拟轴设置""""""
base(5,6,7,8,9)          '虚拟轴轴号 5, 6, 7, 8, 9
atype=0,0,0,0           '设置为虚拟轴
table(0,LargeX,LargeZ,units_x,L1,L2,L3,units_y,units_z,units_w,units_s,smallX,smallY,D3)
units=1000,1000,1000,units_w/drive_ratio_w/360, units_y/drive_ratio_s/360      '
脉冲当量提前设置, 中途不能变化
base(0,1,2,3,4)          '关节轴轴号 0, 1, 2, 3, 4
.....

""""""设置启动拐角减速""""""
merge=on                 '开启连续插补

```

```
corner_mode = 2           '启动拐角减速
decel_angle = 15 * (PI/180) '开始减速的角度 15 度
stop_angle = 45 * (PI/180) '降到最低速度的角度 45 度
.....

""""建立连接,运动""""
""建立机械手连接
while 1
    if scan_event(in(0))>0 then '输入 0 上升沿触发
        BASE(0,1,2,3,4)         '选择关节轴号
        CONNFRAME(25,0,5,6,7,8,9) '第 6/7 轴作为虚拟的 XY 轴,启动
逆解连接。
        WAIT LOADED           '等待运动加载,此时会自动调整虚拟轴的
位置。
        ?"逆解模式"
    elseif scan_event(in(0))<0 then '输入 0 下降沿触发
        BASE(5,6,7,8,9)         '选择虚拟轴号
        CONNREFRAME(25,0,0,1,2,3,4) '第 0/1 轴作为关节轴,启动正
解连接。
        WAIT LOADED           '等待运动加载。
        ?"正解模式"
    endif
wend

end
```


6 自由度机械手类

FRAME6--6 自由度机械手

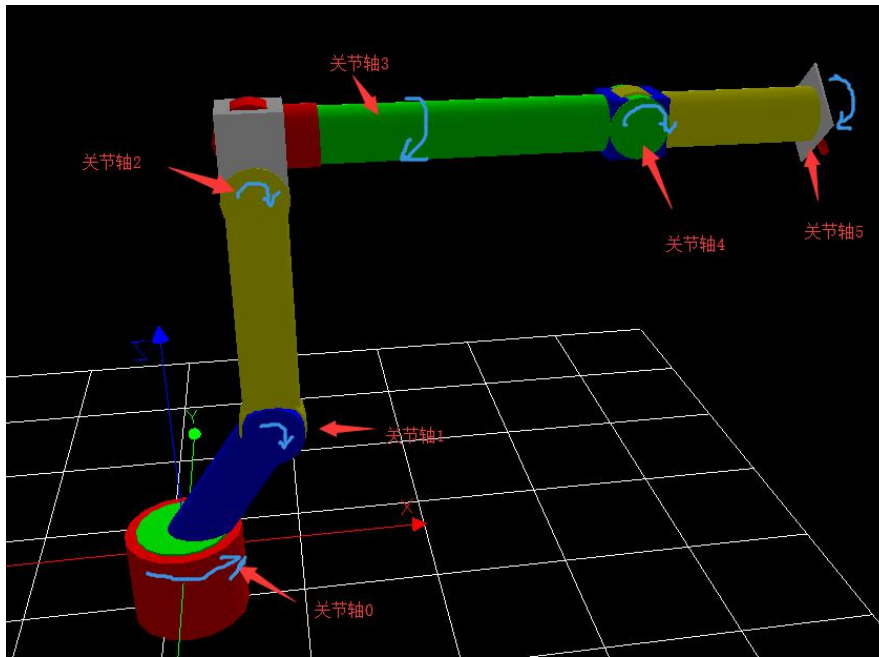
结构示意图



带“R”后缀的控制器支持。

电机方向及角度范围

各关节电机正向如下图蓝色箭头所示。



	角度或移动范围
关节轴 0	$(-\pi, \pi)$
关节轴 1	$(-\pi, \pi)$
关节轴 2	$(-3\pi/2, \pi/2)$
关节轴 3	$(-2\pi, 2\pi)$
关节轴 4	$(-\pi, \pi)$
关节轴 5	$(-2\pi, 2\pi)$

轴顺序

关节轴顺序:

J1- J6: 从大关节到末端旋转轴.

实际机械手关节轴:	定义简写:
关节轴 0	Axis_a
关节轴 1	Axis_b
关节轴 2	Axis_c
关节轴 3	Axis_d
关节轴 4	Axis_e
关节轴 5	Axis_f

直角坐标系虚拟轴:	定义简写:
平移轴 X	Viraxis_x
平移轴 Y	Viraxis_y
平移轴 Z	Viraxis_z
旋转轴 RX	Viraxis_rx

旋转轴 RY	Viraxis_ry
旋转轴 RZ	Viraxis_rz

指令规定轴号填写次序如下：

逆解时：

```
BASE(Axis_a,Axis_b,Axis_c,Axis_d,Axis_e,Axis_f)
CONNFRAME(6,tablenum, Viraxis_x, Viraxis_y,
Viraxis_z,Viraxis_rx,Viraxis_ry,Viraxis_rz)
```

正解时：

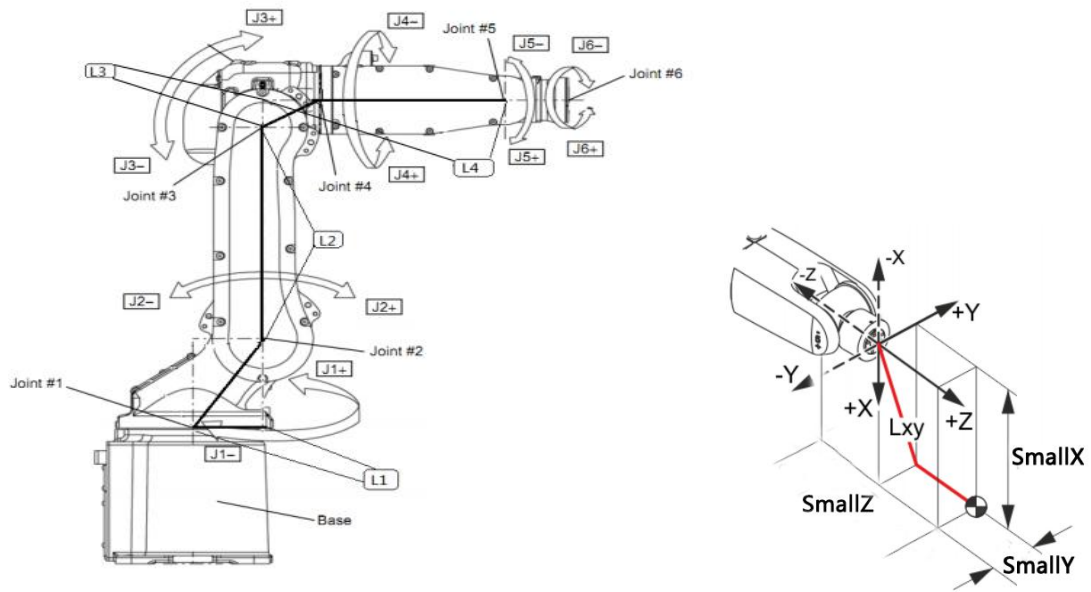
```
BASE(Viraxis_x, Viraxis_y,Viraxis_z,Viraxis_rx,Viraxis_ry,Viraxis_rz)
CONNREFRAME(6,tablenum, Axis_a,Axis_b,
Axis_c,Axis_d,Axis_e,Axis_f)
```

结构参数设置

建立机械手连接时，需要将机械结构参数依次填写到 table 数组中，参数说明如下。

TABLE(Tablenum, LargeZ, L1, L2, L3, L4, D5, Pules1OneCircle, Pules2OneCircle, Pules3OneCircle, Pules4OneCircle, Pules5OneCircle, Pules6OneCircle, PulesVROneCircle, SmalLX, SmalLY, SmalLZ, InitRx, InitRy, InitRz)	
Tablenum	存储转换参数的 TABLE 位置
LargeZ	大盘的垂直高度。虚拟轴坐标系零点到关节轴 2 旋转中心，Z 方向的距离
L1	关节轴 1 到关节轴 2 在 X 方向的偏移。转盘中心到大摆臂中心的偏移
L2	大摆臂的长度，关节轴 2 到关节轴 3 的距离
L3	3 轴中心到 4 轴在 Z 方向的距离
L4	3 轴到 5 轴的水平距离
D5	5 转一圈，6 转动的圈数，0 表示不关联
Pules1OneCircle	关节轴 1 旋转一圈的脉冲数
Pules2OneCircle	关节轴 2 旋转一圈的脉冲数
Pules3OneCircle	关节轴 3 旋转一圈的脉冲数
Pules4OneCircle	关节轴 4 旋转一圈的脉冲数
Pules5OneCircle	关节轴 5 旋转一圈的脉冲数
Pules6OneCircle	关节轴 6 旋转一圈的脉冲数
PulesVROneCircle	虚拟姿态轴旋转一圈的脉冲数
SmalLX	零点时，末端工作点到关节轴 6 在 X 方向的偏移
SmalLY	零点时，末端工作点到关节轴 6 在 Y 方向的偏

	移
SmalLZ	末端旋转轴 6 到轴 5 中心的距离
InitRx	初始的姿态, 弧度单位, 末端指向的方向, 为末端工具坐标系的 Z 方向, 0,0,0,指向 Z 方向
InitRy	初始的姿态, 弧度单位, 末端指向的方向, 为末端工具坐标系的 Z 方向, 0,0,0,指向 Z 方向
InitRz	初始的姿态, 弧度单位, 末端指向的方向, 为末端工具坐标系的 Z 方向, 0,0,0,指向 Z 方向



电机参数设置

各轴的轴类型和脉冲当量要设置正确。

一个机械手的所有虚拟轴和关节轴的长度单位要求统一, 一般都是 mm 单位。虚拟轴的一个 mm 的脉冲数一般建议设置为 1000, 表示精度为小数点后 3 位。

	Atype	Units
Axis_a	1 或 4 或 7 或 65 或 50	Pules1OneCircle/360
Axis_b		Pules2OneCircle/360
Axis_c		Pules3OneCircle/360
Axis_d		Pules4OneCircle/360
Axis_e		Pules5OneCircle/360
Axis_f		Pules6OneCircle/360
Viraxis_x	0	1000
Viraxis_y		1000
Viraxis_z		1000
Viraxis_rx		PulesVROneCircle/360
Viraxis_ry		PulesVROneCircle/360

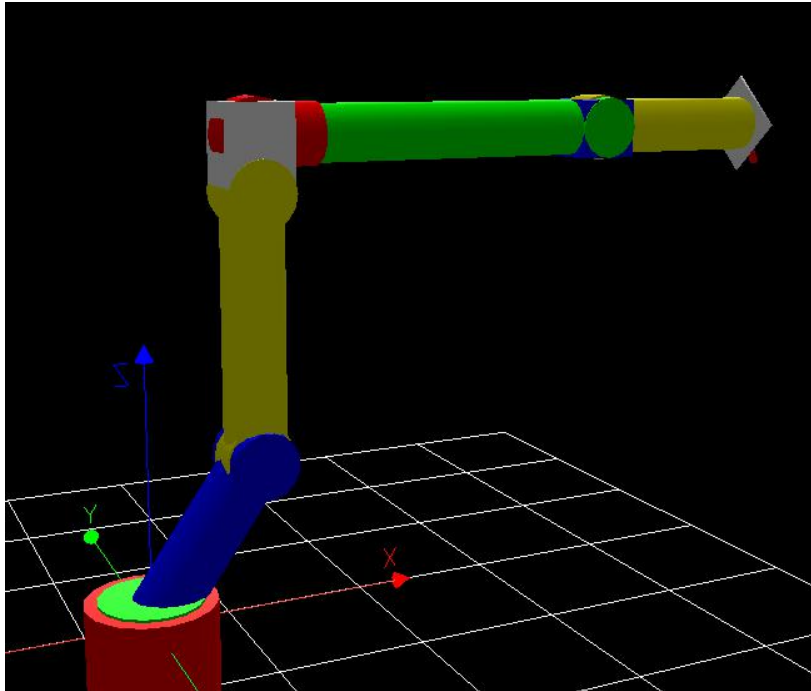
Viraxis_rz

PulesVROneCircle/360

零点位置

关节零点如下图所示。

虚拟轴顺序：X, Y, Z , RX, RY, RZ。RX RY RZ 是先绕 X 旋转，再绕 Y 旋转，再绕 Z 旋转。



建立连接

正解模式

```
BASE(Viraxis_x, Viraxis_y, Viraxis_z, Viraxis_rx, Viraxis_ry, Viraxis_rz)
CONNREFRAME(6, tablenum, Axis_a, Axis_b,
Axis_c, Axis_d, Axis_e, Axis_f)
WAIT LOADED
```

建立成功后，虚拟轴 MTYPE 将显示为 34，IDLE 显示为 0。

此时只能操作关节轴在关节坐标系中运动，虚拟轴会自动计算末端工作点位于直角坐标系中的位置。

逆解模式

```
BASE(Axis_a, Axis_b, Axis_c, Axis_d, Axis_e, Axis_f)
CONNFRAME(6, tablenum, Viraxis_x, Viraxis_y,
Viraxis_z, Viraxis_rx, Viraxis_ry, Viraxis_rz)
WAIT LOADED
```

建立成功后，关节轴 MTYPE 将显示为 33，IDLE 显示为 0。

此时只能操作虚拟轴在直角坐标系中运动，关节轴会自动计算在关节坐标系中如何联合运动。

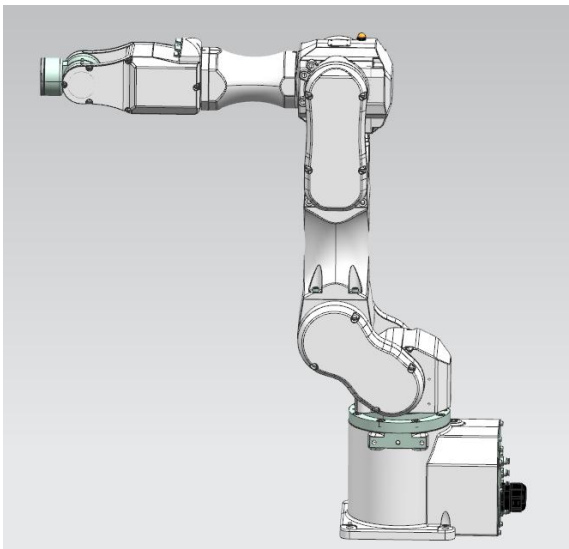
姿态说明

当前姿态通过指令 [FRAME_STATUS](#) 查询。

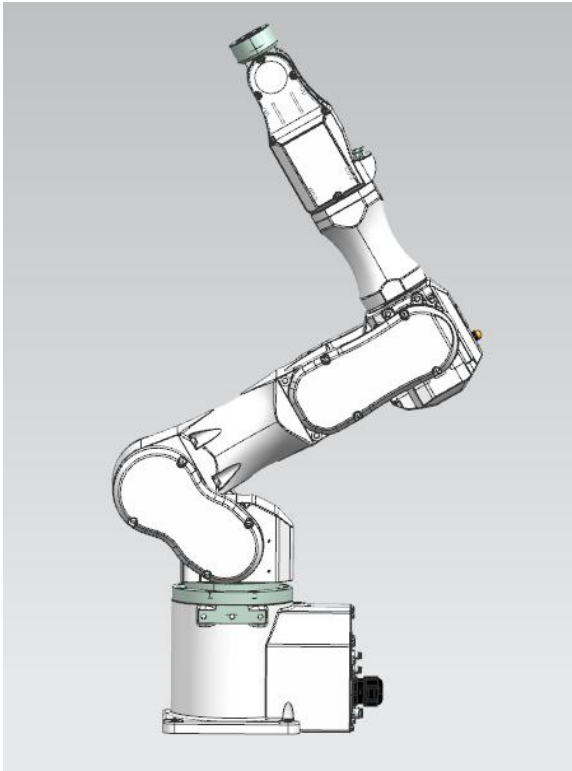
6 自由度机械手姿态是根据关节轴几种基本姿态的组合叠加。

下面列出关节轴基本的姿态说明：

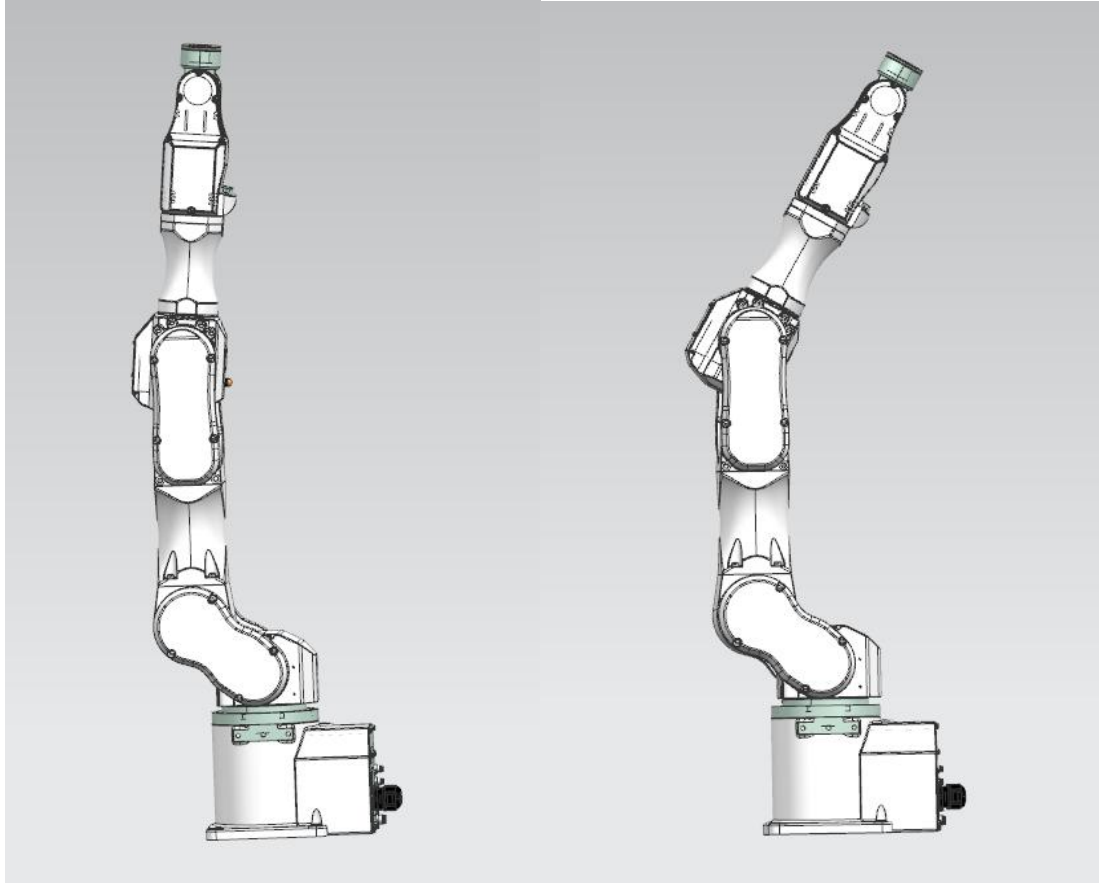
所有轴处于零点位置时，默认为姿态 0。



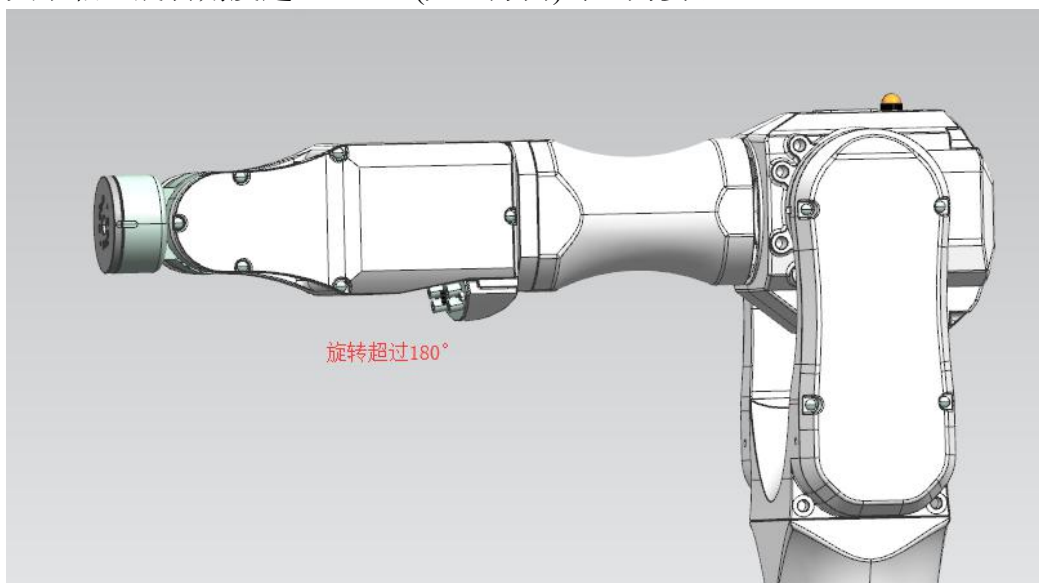
关节轴 1 处于 $(-180^{\circ}, -60^{\circ})$ 之间时，为姿态 1。



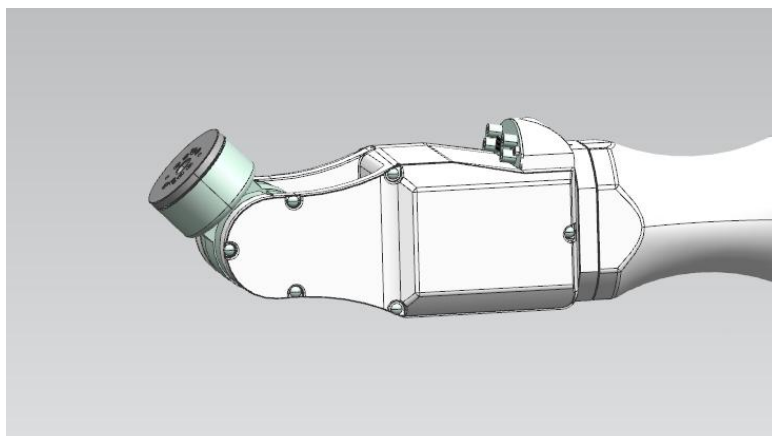
关节轴 2 处于 $(-270^{\circ}, -99^{\circ})$ 之间时, 为姿态 2; 处于 $[-99^{\circ}, -90^{\circ})$ 之间时, 为姿态 3。



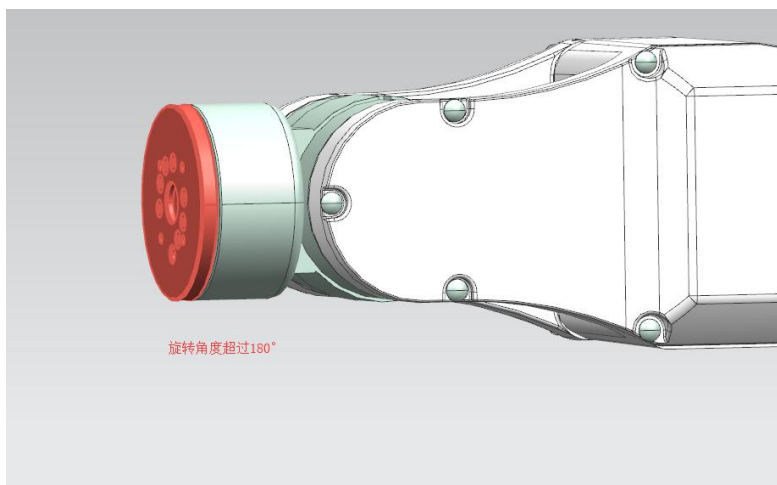
关节轴 3 旋转角度超过 180° (无论方向)时, 为姿态 8。



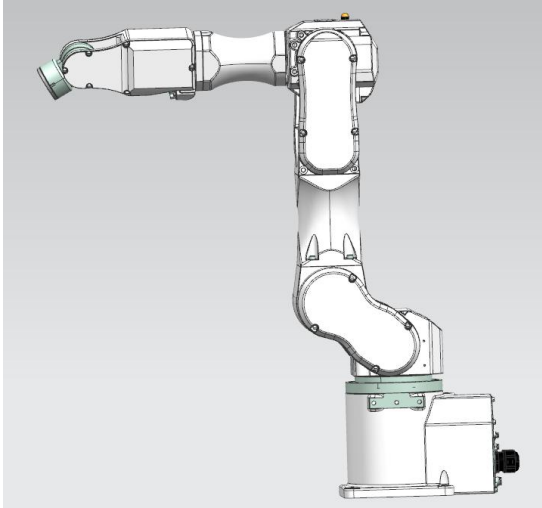
关节轴 4 处于 $(-180^\circ, 0^\circ)$ 之间时, 为姿态 4。



关节轴 5 旋转角度超过 180° (无论方向)时, 为姿态 16。



机械手最终实际姿态由各关节轴旋转角度组合判断，比如关节轴 5 此时 200°，关节轴 4 此时 -30°，关节轴 3 此时 190°，那么当前 Frame_status 姿态值为 0+0+0+8+4+16=28。



实际使用时一般使用以下几种姿态。

右手姿势	左手姿势		
		手腕非反转姿势	手腕反转姿势

定义各姿态简写如下：

L (左手姿势) 或者 R (右手姿势)

A (上肘姿势) 或者 B (下肘姿势)

NF (手腕非反转姿势) 或者 F (手腕反转姿势)

手臂姿势能够进行如下 8 种形式的组合，但根据点的情况，并不能说所有的组合都是能够动作的。

手臂姿势组合			
	/R /A /NF		/R /A /F
	/L /A /NF		/L /A /F

	/R /B /NF		/R /B /F
	/L /B /NF		/L /B /F

使用例程

""电机、机械手参数定义

```

dim LargeZ          '基座的垂直高度
dim L1              '1 轴到 2 轴的 X 偏移；转盘中心到大摆臂中心的偏移。
dim L2              '大摆臂长度
dim L3              '3 轴中心到 4 轴中心距离
dim L4              '4 轴到 5 轴的距离。
dim D5              '5 转一圈，6 转动的圈数，0 表示不关联。
dim PulesVROneCircle '虚拟姿态轴一圈脉冲数
dim SmalLZ          '末端到 5 轴的垂直距离
dim SmalLX, SmalLY '末端到转盘中心的 XY 偏移。 0 位置的时候
dim InitRx, InitRy, InitRz '初始的姿态，弧度单位。 0, 0, 0 指向 z 正向
LargeZ=50
L1=0
L2=100
L3=0
L4=60
D5=0
SmalLZ=10
SmalLX=0
SmalLY=0
InitRx=0
InitRy=0
InitRz=0
PulesVROneCircle=360*1000

dim u_m1           '电机 1 一圈脉冲数
dim u_m2           '电机 2 一圈脉冲数
dim u_m3           '电机 3 一圈脉冲数
dim u_m4           '电机 4 一圈脉冲数
dim u_m5           '电机 5 一圈脉冲数
dim u_m6           '电机 6 一圈脉冲数
u_m1=10000
u_m2=10000
u_m3=10000
u_m4=10000
u_m5=10000
u_m6=10000

```

```

dim i_1      '关节 1 传动比
dim i_2      '关节 2 传动比
dim i_3      '关节 3 传动比
dim i_4      '关节 4 传动比
dim i_5      '关节 5 传动比
dim i_6      '关节 6 传动比
i_1=1
i_2=1
i_3=1
i_4=1
i_5=1
i_6=1

dim u_j1     '关节 1 实际一圈脉冲数
dim u_j2     '关节 2 实际一圈脉冲数
dim u_j3     '关节 3 实际一圈脉冲数
dim u_j4     '关节 4 实际一圈脉冲数
dim u_j5     '关节 5 实际一圈脉冲数
dim u_j6     '关节 6 实际一圈脉冲数
u_j1=u_m1*i_1
u_j2=u_m2*i_2
u_j3=u_m3*i_3
u_j4=u_m4*i_4
u_j5=u_m5*i_5
u_j6=u_m6*i_6

""关节轴设置
BASE(0,1,2,3,4,5)      '选择关节轴号 0、1、2、3、4、5
atype=1,1,1,1,1,1     '轴类型设为脉冲轴
UNITS = u_j1/360,u_j2/360,u_j3/360,u_j4/360,u_j5/360 ,u_j6/360  '把 units 设成
每° 脉冲数
DPOS=0,0,0,0,0,0     '设置关节轴的位置，此处要根据实际情况
来修改。
speed=100,100,100,100,100,100      '速度参数设置
accel=1000,1000,1000,1000,1000,1000
decel=1000,1000,1000,1000,1000,1000
CLUTCH_RATE=0,0,0,0,0,0      '使用关节轴的速度和加速度限制

""""""""""设置启动拐角减速""""""""""
merge=on      '开启连续插补
corner_mode = 2      '启动拐角减速
decel_angle = 15 * (PI/180)      '开始减速的角度 15 度

```

```

stop_angle = 45 * (PI/180)      '降到最低速度的角度 45 度

"虚拟轴设置"
BASE(6,7,8,9,10,11)
ATYPE=0,0,0,0,0,0      '设置为虚拟轴
TABLE(0,LargeZ,L1,L2,L3,L4,D5,u_j1,u_j2,u_j3,u_j4,u_j5,u_j6,PulesVROneCircle,
SmallX,SmallY,SmallZ,InitRx,InitRy,InitRz)      '根据手册说明填写参数
UNITS=1000,1000,1000,1000,1000,1000      '运动精度，要提前设置，中途不能变化

"建立机械手连接"
while 1
  if scan_event(in(0))>0 then      '输入 0 上升沿触发
    BASE(0,1,2,3,4,5)      '选择关节轴号
    CONNFRAME(6,0,6,7,8,9,10,11)      '启动逆解连接。
    WAIT LOADED      '等待运动加载，此时会自动调整虚拟轴的位置。
    ?"逆解模式"
  elseif scan_event(in(0))<0 then      '输入 0 下降沿触发
    BASE(6,7,8,9,10,11)      '选择虚拟轴号
    CONNREFRAME(6,0,0,1,2,3,4,5)      '启动正解连接。
    WAIT LOADED      '等待运动加载。
    ?"正解模式"
  endif
wend

```

FRAME26

带“R”后缀的控制器支持。

结构跟机械手文档中的 FRAME6 是类似的。

区别在于：

轴 2 和轴 3 单方面关联。轴 2 运动--轴 3 跟着运动（比例按参数设置值），轴 3 动的时候--轴 2 不跟着动。

```

TABLE(LargeZ,L1,L2,L3,L4,D5,Pules1OneCircle,Pules2OneCircle,Pules3OneCircle,
Pules4OneCircle,Pules5OneCircle,Pules6OneCircle,PulesROneCircle,SmallX,
SmallY, SmallZ, InitRx, InitRy, InitRz, j2toj3)

```

j2toj3 旋转轴 2 与旋转轴 3 关联，关联的比例设置，不关联设置 0。

FRAME36

带“R”后缀的控制器支持。

增加第 4 关节轴心 Y 方向的偏移参数 DiffY，其它同 FRAME6.

TABLE(LargeZ, L1,L2,L3, L4, D5, Pules1OneCircle, Pules2OneCircle, Pules3OneCircle, Pules4OneCircle, Pules5OneCircle, Pules6OneCircle, PulesVROneCircle, SmalLX, SmalLY, SmalLZ, InitRx, InitRy, InitRz, DiffY)

FRAME46

带“R”后缀的控制器支持。

结构跟机械手文档中的 FRAME6 是类似的。

区别在于：

1. 轴 3 和轴 4 单方面关联。轴 3 运动--轴 4 跟着运动（固定比例 2:1），轴 4 动的时候--轴 3 不跟着动。
2. 轴 1 和轴 2 单方面关联。轴 1 运动--轴 2 跟着运动（比例按参数设置值），轴 2 动的时候--轴 1 不跟着动。

结构示意图



结构参数设置

TABLE(LargeZ,L1,L2,L3,L4,D5,Pules1OneCircle,Pules2OneCircle,Pules3OneC

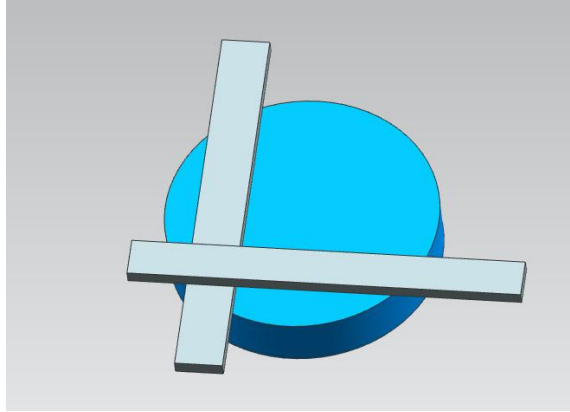
ircle, Pules4OneCircle,Pules5OneCircle,Pules6OneCircle,PulesROneCircle,SmalLX,
SmalLY, SmallLZ, InitRx, InitRy, InitRz, j4toj5)

j4toj5 旋转轴 4 与旋转轴 5 关联, 关联的比例设置, 不关联设置 0。

旋转台类

FRAME11--单旋转台

结构示意图



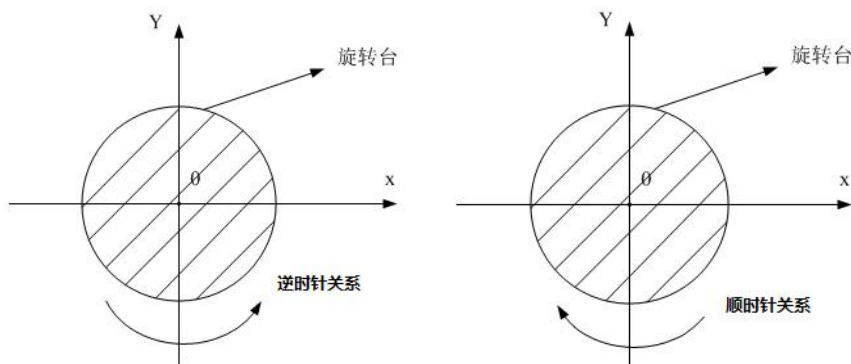
电机方向及角度范围

两个平移轴方向无要求。

旋转台转向从上往下看，根据参数 IfAntiClock 的值确定。

IfAntiClock=1 时，与 XY 为逆时针关系（旋转台从 X 正转向 Y 正）。

IfAntiClock=0 时，与 XY 为顺时针关系（旋转台从 Y 正转向 X 正）。



	角度或移动范围
X 轴	机械结构限制
Y 轴	
旋转轴	无限制

轴顺序

实际机械手关节轴:	定义简写:
平移轴 X	Axis_a
平移轴 Y	Axis_b
旋转轴	Axis_c

直角坐标系虚拟轴:	定义简写:
平移轴 X	Viraxis_x
平移轴 Y	Viraxis_y
旋转轴 RZ	Axis_c

指令规定轴号填写次序如下:

逆解时:

```
BASE(Axis_a,Axis_b, Axis_c)
CONNFRAME(11,tablenum,Viraxis_x,Viraxis_y,Axis_c)
```

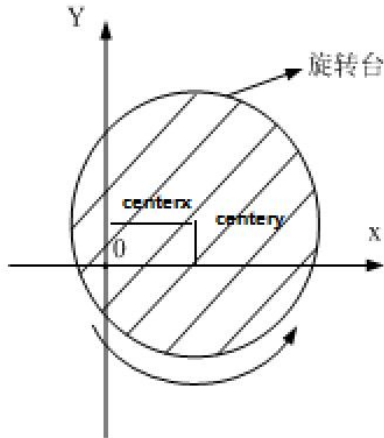
正解时:

```
BASE(Viraxis_x, Viraxis_y, Axis_c)
CONNREFRAME(11,tablenum, Axis_a,Axis_b, Axis_c)
```

结构参数设置

建立机械手连接时，需要将机械结构参数依次填写到 table 数组中，参数说明如下。

TBALE(tablenum, CENTERX, CENTERY, pulseRonecircle, IfAntiClock)	
Tablenum	存储转换参数的 TABLE 位置
CENTERX	X 方向偏移量
CENTERY	Y 方向偏移量
pulseRonecircle	末端旋转轴旋转一圈的脉冲数
IfAntiClock	是否逆时针转动



电机参数设置

各轴的轴类型和脉冲当量要设置正确。

一个机械手的所有虚拟轴和关节轴的长度单位要求统一，一般都是 mm 单位。虚拟轴的一个 mm 的脉冲数一般建议设置为 1000，表示精度为小数点后 3 位。

	Atype	Units
Axis_a	1 或 4 或 7 或 65 或 50	每 mm 脉冲数
Axis_b		每 mm 脉冲数
Axis_c		pulseRonecircle/360
Viraxis_x	0	1000
Viraxis_y		1000

零点位置

无特殊要求，当前位置即可作为 0 点，但是注意此时的圆心偏移参数 CENTERX 和 CENTERY 要设置正确。

建立连接

正解模式

```
BASE(Viraxis_x, Viraxis_y, Axis_c)
CONNREFRAME(11,tablenum, Axis_a,Axis_b, Axis_c)
WAIT LOADED
```

建立成功后，虚拟轴 MTYPE 将显示为 34，IDLE 显示为 0。

此时只能操作关节轴在关节坐标系中运动，虚拟轴会自动计算末端工作点位于直角坐标系中的位置。

逆解模式

```
BASE(Axis_a,Axis_b,Axis_c)
CONNFRAME(11,tablenum,Viraxis_x,Viraxis_y,Axis_c)
WAIT LOADED
```

建立成功后，关节轴 MTYPE 将显示为 33，IDLE 显示为 0。
此时只能操作虚拟轴在直角坐标系中运动，关节轴会自动计算在关节坐标系中如何联合运动。

使用例程

```
""""""机械参数、点击参数定义
dim axisx,axisy,axisR '实际关节轴定义
dim viraxisx,viraxisy'虚拟轴定义
axisx=0
axisy=1 '实际关节轴编号
axisR=2
viraxisx=3 '虚拟轴编号
viraxisy=4

dim CENTERX 'x 方向偏移
dim CENTERY 'y 方向偏移
dim IfAntiClock '是否为逆时针旋转，逆时针旋转遵循右手法则
CENTERX=30 '旋转中心偏移
CENTERY=40
IfAntiClock=1 '此时为逆时针关系

dim u_mx 'x 电机旋转一圈的脉冲
dim u_my 'y 电机旋转一圈的脉冲
dim u_mR '旋转电机旋转一圈的脉冲
u_mx=10000
u_my=10000
u_mR=10000 '脉冲量赋值

dim i_x 'x 电机传动比
dim i_y 'y 电机传动比
dim i_R '旋转电机传动比
i_x=1
i_y=1
i_R=1 '传动比赋值
```

```

dim u_jx    'x 关节实际一圈脉冲数
dim u_jy    'y 关节实际一圈脉冲数
dim u_jR    '旋转关节实际一圈脉冲数
u_jx=u_mx*i_x
u_jy=u_my*i_y
u_jR=u_mR*i_R

dim p_x     'x 螺距
dim p_y     'y 螺距
p_x=2
p_y=2

""""""定义关节轴
base(axisx,axisy,axisR)
rapidstop
wait idle           '确保所有轴停止运动
units=u_jx/p_x,u_jy/p_y,u_jR/360 'xy 设为 1mm 脉冲数， 旋转轴设为 1°脉冲数
atype = 1,1,1      ' 轴功能设置
dpos=0,0,0
creep=10,10,10     '轴回零速度
speed =300,300,300 '移动速度
accel = 1000,1000,1000 '加速度
decel = 500,500,500 '减速度

""""""虚拟轴定义
base(viraxisx,viraxisy)
atype = 0,0        '定义虚拟轴
TABLE(0,CENTERX,CENTERY,u_jR,IfAntiClock) 'table 表参数
speed=200,200     '虚拟轴运动速度
accel = 300,300   '虚拟轴运动加速度
units=10000,10000

""""""运动控制开始
merge = on          '连续缓冲开启
CORNER_MODE =2+32   '设置启动拐角减速+倒角
ZSMOOTH=10          '倒角半径设置
DECEL_ANGLE = 15 * (PI/180) '开始减速的最小拐角
STOP_ANGLE = 45 * (PI/180) '减速到最低的减速拐角

""建立机械手连接
while 1

```

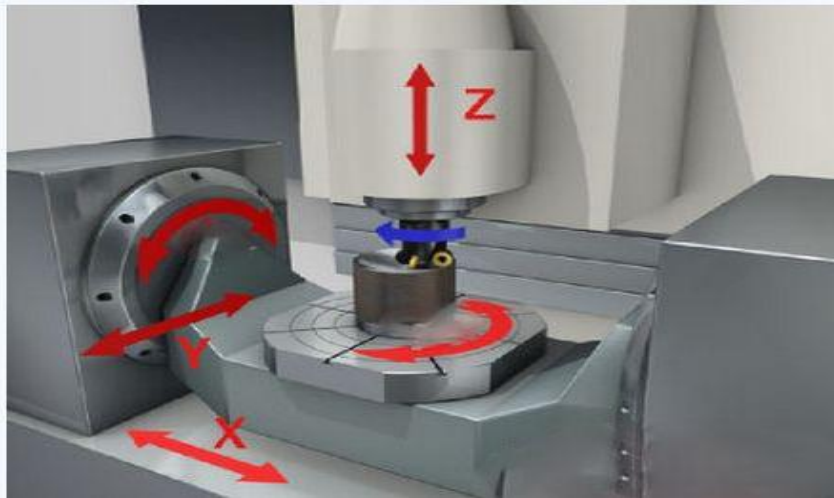
```
if scan_event(in(0))>0 then  '输入 0 上升沿触发
    BASE(axisx,axisy,axisR)      '选择关节轴号
    CONNFRAME(11,0,viraxisx, viraxisy,axisR)  '启动逆解连接。
    WAIT LOADED                  '等待运动加载，此时
会自动调整虚拟轴的位置。
    ?"逆解模式"
elseif scan_event(in(0))<0 then  '输入 0 下降沿触发
    BASE(viraxisx, viraxisy,axisR)  '选择虚拟轴号
    CONNREFRAME(11,0,axisx,axisy,axisR)  '启动正解连接。
    WAIT LOADED                  '等待运动加载。
    ?"正解模式"
endif

wend
```

FRAME17--双选转台

在 YZ 平面有一个旋转台 1，在 XY 平面上有一个旋转台 2，旋转台 2 安装在旋转台 1 上。当旋转轴坐标 0 时，虚拟轴和实际轴坐标一致，虚拟坐标系与实际坐标系也重合。

结构示意图

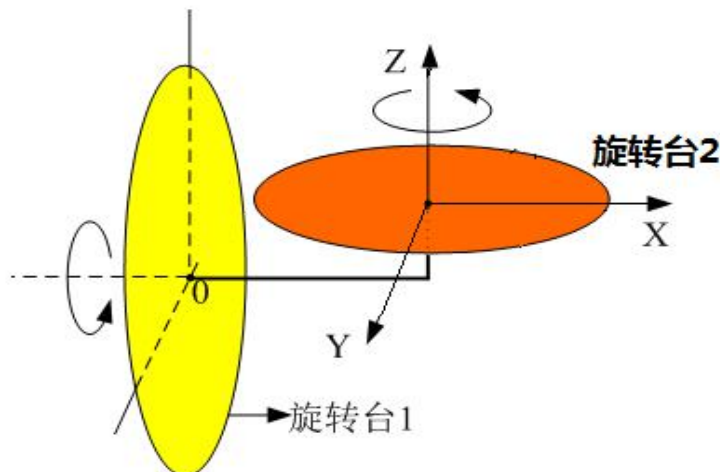


电机方向及角度范围

关节轴 XYZ 方向没有要求。

旋转台 1 方向根据实际情况设置 IfAntiClock1 参数：逆时针值为 1（旋转台从 Y 正转向 Z 正），顺时针值为 0（旋转台从 Z 正转向 Y 正）。

旋转台 2 方向根据实际情况设置 IfAntiClock2 参数：逆时针值为 1（旋转台从 X 正转向 Y 正），顺时针值为 0（旋转台从 Y 正转向 X 正）。



	角度或移动范围
X 轴	机械结构限制
Y 轴	
Z 轴	
RX 旋转轴	无限制
Rz 旋转轴	

轴顺序

实际机械手关节轴:	定义简写:
平移轴 X	Axis_a
平移轴 Y	Axis_b
平移轴 Z	Axis_c
旋转台 1	Axis_d
旋转台 2	Axis_e

直角坐标系虚拟轴:	定义简写:
平移轴 X	Viraxis_x
平移轴 Y	Viraxis_y
平移轴 Z	Viraxis_z
旋转轴 RX	Axis_d
旋转轴 RZ	Axis_e

指令规定轴号填写次序如下:

逆解时:

```
BASE(Axis_a,Axis_b, Axis_c,Axis_d,Axis_e)
CONNFRAME(17,tablenum,Viraxis_x,Viraxis_y,Viraxis_v,Axis_d,Axis_e)
```

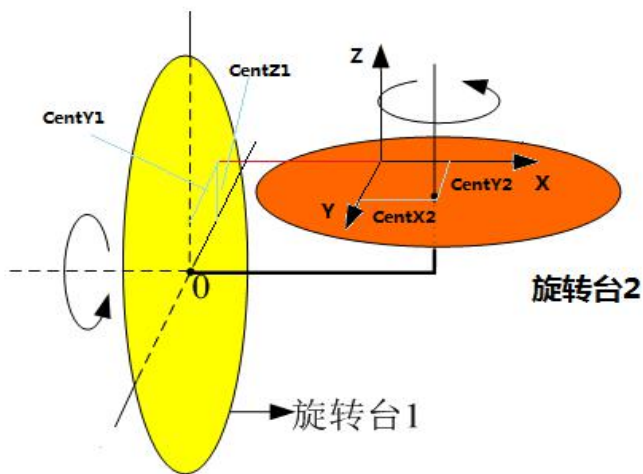
正解时:

```
BASE(Viraxis_x, Viraxis_y, Viraxis_v,Axis_d,Axis_e)
CONNREFRAME(17,tablenum, Axis_a,Axis_b, Axis_c,Axis_d,Axis_e)
```

结构参数设置

建立机械手连接时，需要将机械结构参数依次填写到 table 数组中，参数说明如下。

TABLE(tablenum, CentY1, CentZ1, pulse1onecircle, IfAntiClock1, CentX2, CentY2, pulse2onecircle, IfAntiClock2)	
Tablenum	存储转换参数的 TABLE 位置
CentY1	零点时, yz 平面内, rx 旋转轴的旋转中心与原点 y 方向的偏移
CentZ1	零点时, yz 平面内, rx 旋转轴的旋转中心与原点 Z 方向的偏移
pulse1onecircle	旋转台 1 的一圈脉冲数
IfAntiClock1	旋转台 1 与 YZ 是否逆时针的关系, 逆时针值为 1 (旋转台从 Y 正转向 Z 正), 顺时针值为 0 (旋转台从 Z 正转向 Y 正)
CentX2	零点时, xy 平面内, rz 旋转轴的旋转中心与原点 x 方向的偏移
CentY2	零点时, xy 平面内, rz 旋转轴的旋转中心与原点 y 方向的偏移
pulse2onecircle	旋转台 2 的一圈脉冲数
IfAntiClock2	旋转台 2 与 XY 是否逆时针的关系, 逆时针值为 1, 顺时针值为 0, 逆时针的时候符合右手法则



电机参数设置

各轴的轴类型和脉冲当量要设置正确。

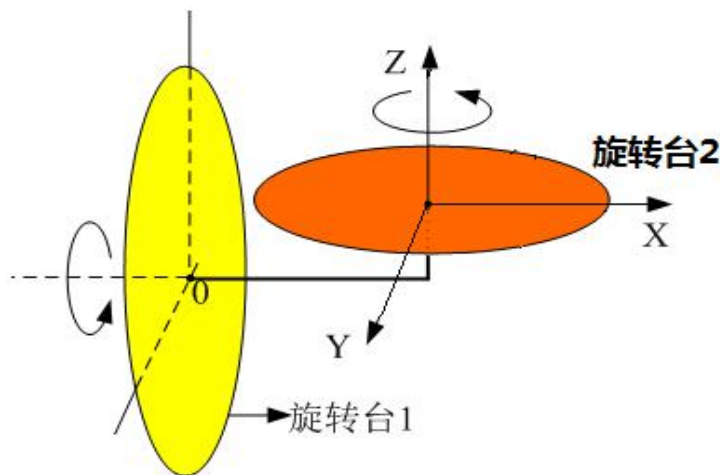
一个机械手的所有虚拟轴和关节轴的长度单位要求统一, 一般都是 mm 单位。虚拟轴的一个 mm 的脉冲数一般建议设置为 1000, 表示精度为小数点后 3 位。

	Atype	Units
Axis_a	1 或 4 或 7 或 65 或 50	每 mm 脉冲数
Axis_b		每 mm 脉冲数
Axis_c		每 mm 脉冲数

Axis_d		Pulse1onecircle/360
Axis_e		Pulse2onecircle/360
Viraxis_x	0	1000
Viraxis_y		1000
Viraxis_z		1000

零点位置

无特殊要求，当前位置即可作为 0 点，但是注意此时旋转台 1 的圆心偏移参数 CENTERZ1 和 CENTERY1，旋转台 2 的圆心偏移参数 CENTERX2 和 CENTERY2 要设置正确。



建立连接

正解模式

```
BASE(Viraxis_x, Viraxis_y, Viraxis_z, Axis_d, Axis_e)
CONNREFRAME(17, tablenum, Axis_a, Axis_b, Axis_cs, Axis_d, Axis_e)
WAIT LOADED
```

建立成功后，虚拟轴 MTYPE 将显示为 34，IDLE 显示为 0。

此时只能操作关节轴在关节坐标系中运动，虚拟轴会自动计算末端工作点位于直角坐标系中的位置。

逆解模式

```
BASE(Axis_a, Axis_b, Axis_c, Axis_d, Axis_e)
CONNFRAME(17, tablenum, Viraxis_x, Viraxis_y, Viraxis_z, Axis_d, Axis_e)
```


WAIT LOADED

建立成功后，关节轴 MTYPE 将显示为 33，IDLE 显示为 0。
此时只能操作虚拟轴在直角坐标系中运动，关节轴会自动计算在关节坐标系中如何联合运动。

使用例程

```
'=====机械参数、电机参数定义=====
dim axisx,axisy,axisz,axisr1,axisr2    "定义实际轴
dim viraxisx,viraxisy,viraxisz        "定义虚拟轴
axisx=0    "'实际轴编号
axisy=1
axisz=2
axisr1=6    "'旋转轴编号
axisr2=7

viraxisx=3    "'虚拟轴编号
viraxisy=4
viraxisz=5

dim centy1,centz1,centx2,centy2        '旋转中心定义
dim ifanticlock1,ifanticlock2          '是否逆时针
centy1=10    '偏移
centz1=10
centx2=30
centy2=20
ifanticlock1=1    '旋转台 1 逆时针
ifanticlock2=1    '旋转台 2 逆时针

dim u_mx,u_my,u_mz    '轴旋转一圈的脉冲
dim u_mr1,u_mr2
u_mx=10000    '旋转轴脉冲参数赋值
u_my=10000
u_mz=10000
u_mr1=10000
u_mr2=10000

dim i_x,i_y,i_z    '电机传动比
dim i_r1,i_r2
i_x=2
i_y=2
```

```

i_z=2
i_r1=1
i_r2=1

dim u_jx 'x 关节实际一圈脉冲数
dim u_jy 'y 关节实际一圈脉冲数
dim u_jz 'z 关节实际一圈脉冲数
dim u_jr1 '旋转 1 实际一圈脉冲数
dim u_jr2 '旋转 2 实际一圈脉冲数
u_jx=u_mx*i_x
u_jy=u_my*i_y
u_jz=u_mz*i_z
u_jr1=u_mr1*i_r1
u_jr2=u_mr2*i_r2

dim p_x 'x 螺距
dim p_y 'y 螺距
dim p_z 'z 螺距
p_x=2
p_y=2
p_z=10

""""""定义关节轴
BASE(axisx,axisy,axisz) '实际轴定义
ATYPE=1,1,1 '轴运动类型
rapidstop
wait idle '确保所有轴停止运动
UNITS=u_jx/p_x,u_jy/p_y,u_jz/p_z '设置为 1mm 的脉冲数
Dpos=0,0,0
speed=500,500,500
CREEP=20,20,20
ACCEL=1000,1000,1000
DECEL=500,500,500

base(axisr1,axisr2) '旋转轴设置
ATYPE=1,1 '轴运动类型
units =u_jr1/360,u_jr2/360 '设置为 1° 的脉冲数
Dpos=0,0
speed=200,200 '末端旋转轴参数设置
accel=300,300

""""""虚拟轴定义

```

```

base(viraxisx,viraxisy,viraxisz) '虚拟轴定义
ATYPE=0,0,0
TABLE(0, CentY1, CentZ1,u_jr1, IfAntiClock1, CentX2, CentY2,u_jr2,
IfAntiClock2) ""table 表参数填写
UNITS=1000,1000,1000 ""虚拟轴运动参数赋值
SPEED=300,300,300
ACCEL=500,500,500

""""""运动控制开始
merge = on '连续缓冲开启
CORNER_MODE =2+32 '设置启动拐角减速+倒角
ZSMOOTH=10 '倒角半径设置
DECEL_ANGLE = 15 * (PI/180) '开始减速的最小拐角
STOP_ANGLE = 45 * (PI/180) '减速到最低的减速拐角

""""建立机械手连接
while 1
  if scan_event(in(0))>0 then '输入 0 上升沿触发
    BASE(axisx,axisy,axisz,axisr1,axisr2) '选择关
    节轴号
    CONNFRAME(17,0, viraxisx,viraxisy,viraxisz,axisr1,axisr2) '启动逆
    解连接。
    WAIT LOADED '等待运动加载，此时会自动调整虚拟轴的位置。
    ?"逆解模式"
  elseif scan_event(in(0))<0 then '输入 0 下降沿触发
    BASE(viraxisx,viraxisy,viraxisz,axisr1,axisr2) '选择虚拟轴号
    CONNREFRAME(17,0,axisx,axisy,axisz,axisr1,axisr2) '启动正解连
    接。
    WAIT LOADED '等待运动加载。
    ?"正解模式"
  endif
wend

```

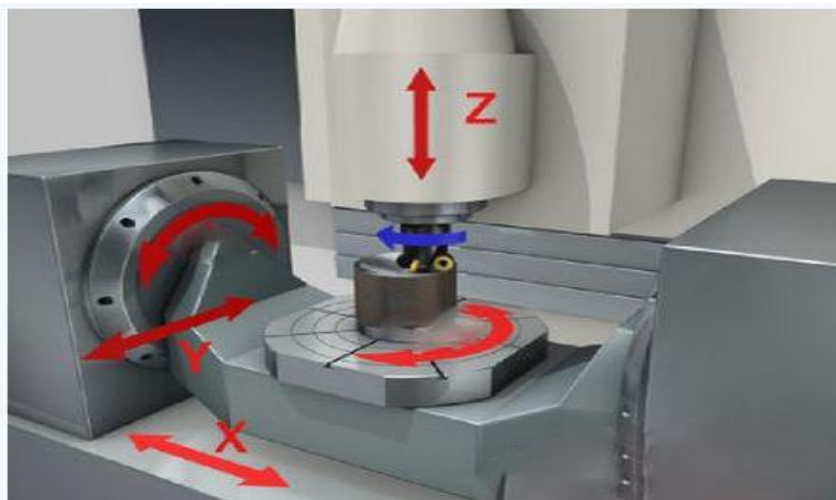
FRAME29--高精度双选转台

在 YZ 平面有一个旋转台 1，在 XY 平面上有一个旋转台 2，旋转台 2 安装在旋转台 1 上。当旋转轴坐标 0 时，虚拟轴和实际轴坐标一致，虚拟坐标系与实际坐标系也重合。

虚拟轴使用工件坐标，以第二个旋转台的旋转中心为 0 点，旋转中心垂直水

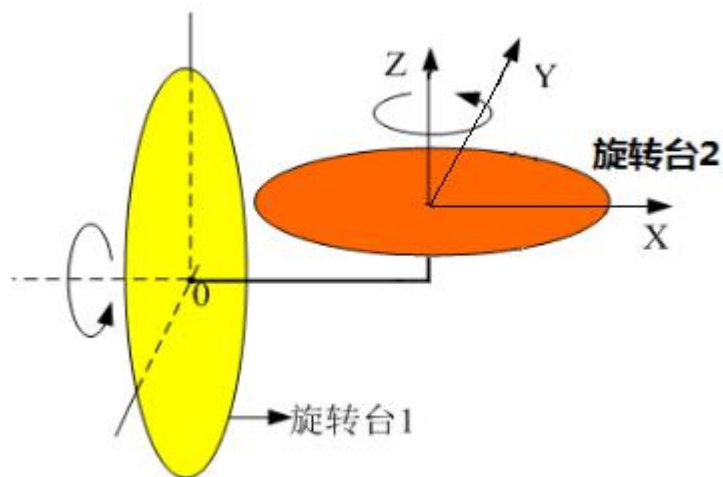
平面为 XY 平面。XYZ 的坐标规律必须满足右手法则。

结构示意图



电机方向及角度范围

关节轴 XYZ 方向没有要求。
旋转台 1 和旋转台 2 转向满足右手法则。



	角度或移动范围
X 轴	机械结构限制
Y 轴	
Z 轴	
RX 旋转轴	无限制
Rz 旋转轴	

轴顺序

实际机械手关节轴:	定义简写:
平移轴 X	Axis_a
平移轴 Y	Axis_b
平移轴 Z	Axis_c
旋转台 1	Axis_d
旋转台 2	Axis_e

直角坐标系虚拟轴:	定义简写:
平移轴 X	Viraxis_x
平移轴 Y	Viraxis_y
平移轴 Z	Viraxis_z
旋转轴 RX	Axis_d
旋转轴 RZ	Axis_e

指令规定轴号填写次序如下:

逆解时:

```
BASE(Axis_a,Axis_b, Axis_c,Axis_d,Axis_e)
CONNFRAME(29,tablenum,Viraxis_x,Viraxis_y,Viraxis_v,Axis_d,Axis_e)
```

正解时:

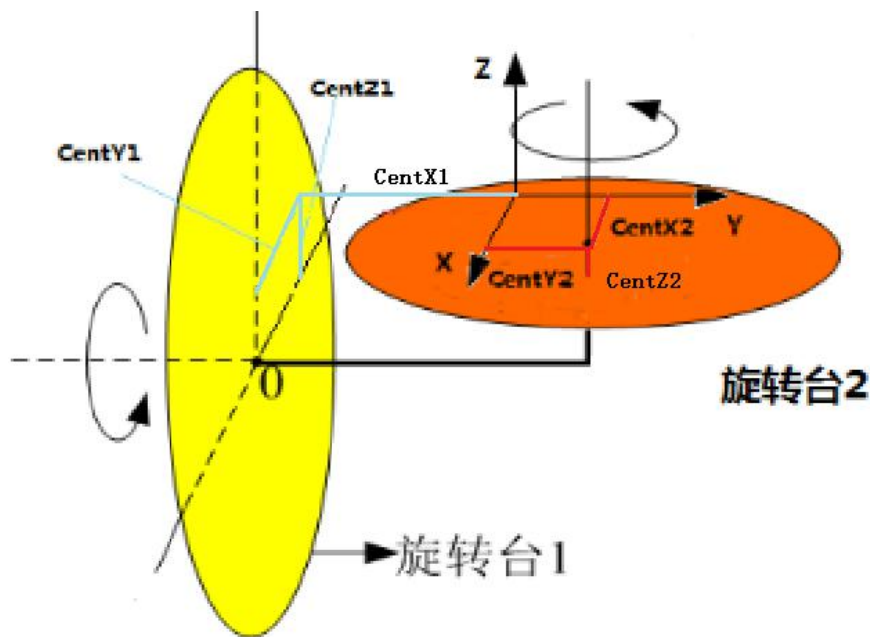
```
BASE(Viraxis_x, Viraxis_y, Viraxis_v,Axis_d,Axis_e)
CONNREFRAME(29,tablenum, Axis_a,Axis_b, Axis_c,Axis_d,Axis_e)
```

结构参数设置

建立机械手连接时, 需要将机械结构参数依次填写到 table 数组中, 参数说明如下。

TABLE(tablenum,CentX1,CentY1,CentZ1,pulse1onecircle,CentX2,CentY2,CentZ2, pulse2onecircle, R1tor2,Reserve1- Reserve6)	
Tablenum	存储转换参数的 TABLE 位置
CentX1	零点时, yz 平面内, rx 旋转轴的旋转中心与原点 x 方向的偏移
CentY1	零点时, yz 平面内, rx 旋转轴的旋转中心与原点 y 方向的偏移
CentZ1	零点时, yz 平面内, rx 旋转轴的旋转中心与原点 Z 方向的偏移
pulse1onecircle	旋转台 1 的一圈脉冲数

CentX2	零点时, xy 平面内, rz 旋转轴的旋转中心与原点 x 方向的偏移
CentY2	零点时, xy 平面内, rz 旋转轴的旋转中心与原点 y 方向的偏移
CentZ2	零点时, xy 平面内, rz 旋转轴的旋转中心与原点 z 方向的偏移
pulse2onecircle	旋转台 2 的一圈脉冲数
R1tor2	旋转台 1 旋转时旋转台 2 的关联比例, 0-不关联
Reserve1- Reserve6	预留其他辅助参数, 缺省 0, 其它值只能自动矫正的时候计算出来



电机参数设置

各轴的轴类型和脉冲当量要设置正确。

一个机械手的所有虚拟轴和关节轴的长度单位要求统一, 一般都是 mm 单位。虚拟轴的一个 mm 的脉冲数一般建议设置为 1000, 表示精度为小数点后 3 位。

	Atype	Units
Axis_a	1 或 4 或 7 或 65 或 50	每 mm 脉冲数
Axis_b		每 mm 脉冲数
Axis_c		每 mm 脉冲数
Axis_d		Pulse1onecircle/360
Axis_e		Pulse2onecircle/360
Viraxis_x	0	1000
Viraxis_y		1000
Viraxis_z		1000

零点位置

以第二个旋转台的旋转中心为 0 点，旋转中心垂直水平面为 XY 平面。
XYZ 的坐标规律必须是右手法则。

建立连接

正解模式

```
BASE(Viraxis_x, Viraxis_y, Viraxis_z, Axis_d, Axis_e)
CONNREFRAME(29, tablenum, Axis_a, Axis_b, Axis_cs, Axis_d, Axis_e)
WAIT LOADED
```

建立成功后，虚拟轴 MTYPE 将显示为 34，IDLE 显示为 0。

此时只能操作关节轴在关节坐标系中运动，虚拟轴会自动计算末端工作点位于直角坐标系中的位置。

逆解模式

```
BASE(Axis_a, Axis_b, Axis_c, Axis_d, Axis_e)
CONNFRAME(29, tablenum, Viraxis_x, Viraxis_y, Viraxis_z, Axis_d, Axis_e)
WAIT LOADED
```

建立成功后，关节轴 MTYPE 将显示为 33，IDLE 显示为 0。

此时只能操作虚拟轴在直角坐标系中运动，关节轴会自动计算在关节坐标系中如何联合运动。

使用例程

```
'=====机械参数、电机参数定义=====
dim axisx,axisy,axisz,axisr1,axisr2    "定义实际轴
dim viraxisx,viraxisy,viraxisz        "定义虚拟轴
axisx=0    ""实际轴编号
axisy=1
axisz=2
axisr1=3   ""旋转轴编号
axisr2=4

viraxisx=6 ""虚拟轴编号
```

```

viraxisy=7
viraxisz=8

dim centx1,centy1,centz1,centx2,centy2,centz2      '旋转中心定义
dim dirx1,diry1,dirz1,dirx2,diry2,dirz2
dim R1tor2

centx1=0
centy1=0      '旋转中心偏移
centz1=0
centx2=0
centy2=0
centz2=0

dirx1=1
diry1=0
dirz1=0

dirx2=0
diry2=0
dirz2=1

R1tor2=0

dim u_mx,u_my,u_mz      '轴旋转一圈的脉冲
dim u_mr1,u_mr2
u_mx=3600      '旋转轴脉冲参数赋值
u_my=3600
u_mz=3600
u_mr1=360
u_mr2=360

dim i_x,i_y,i_z      '电机传动比
dim i_r1,i_r2
i_x=2
i_y=2
i_z=2
i_r1=1
i_r2=1

dim u_jx      'x 关节实际一圈脉冲数
dim u_jy      'y 关节实际一圈脉冲数

```



```

dim u_jz    'z 关节实际一圈脉冲数
dim u_jr1   '旋转 1 实际一圈脉冲数
dim u_jr2   '旋转 2 实际一圈脉冲数
u_jx=u_mx*i_x
u_jy=u_my*i_y
u_jz=u_mz*i_z
u_jr1=u_mr1*i_r1
u_jr2=u_mr2*i_r2

dim p_x     'x 螺距
dim p_y     'y 螺距
dim p_z     'z 螺距
p_x=2
p_y=2
p_z=10

""""""定义关节轴
BASE(axisx,axisy,axisz)    '实际轴定义
ATYPE=1,1,1                '轴运动类型
rapidstop
wait idle                  '确保所有轴停止运动
UNITS=u_jx/p_x,u_jy/p_y,u_jz/p_z    '设置为 1mm 的脉冲数
Dpos=0,0,0
speed=500,500,500
CREEP=20,20,20
ACCEL=1000,1000,1000
DECEL=500,500,500

base(axisr1,axisr2)        '旋转轴设置
ATYPE=1,1                 '轴运动类型
units =u_jr1/360,u_jr2/360    '设置为 1° 的脉冲数
Dpos=0,0
speed=200,200              '末端旋转轴参数设置
accel=300,300

""""""虚拟轴定义
base(viraxisx,viraxisy,viraxisz)    '虚拟轴定义
ATYPE=0,0,0
TABLE(0,centx1,          CentY1,          CentZ1,dirx1,diry1,dirz1,u_jr1,          CentX2,
CentY2,centz2,dirx2,diry2,dirz2,u_jr2,R1tor2,0)    ""table 表参数填写
UNITS=1000,1000,1000    ""虚拟轴运动参数赋值
SPEED=300,300,300

```

```
ACCEL=500,500,500
```

```
""""""运动控制开始
```

```
merge = on           '连续缓冲开启
CORNER_MODE = 2+32   '设置启动拐角减速+倒角
ZSMOOTH=10           '倒角半径设置
DECEL_ANGLE = 15 * (PI/180) '开始减速的最小拐角
STOP_ANGLE = 45 * (PI/180) '减速到最低的减速拐角
```

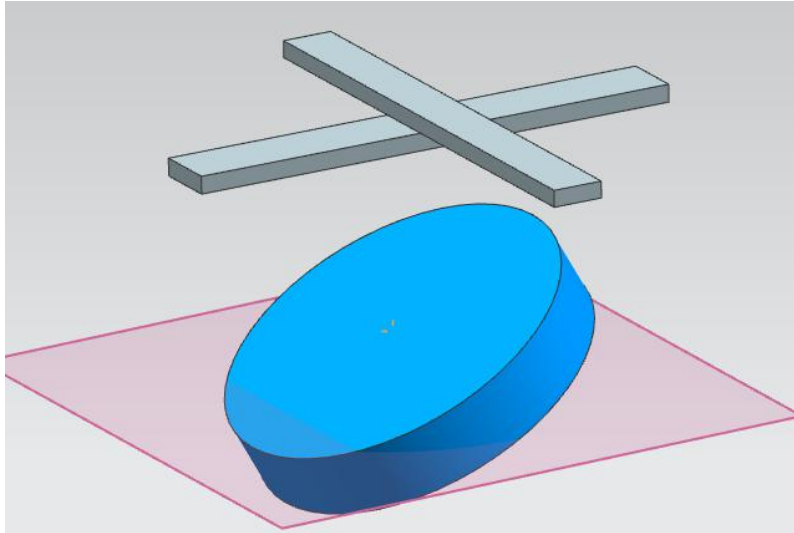
```
""""建立机械手连接
```

```
while 1
  if scan_event(in(0))>0 then '输入 0 上升沿触发
    BASE(axisx,axisy,axisz,axisr1,axisr2) '选择关节轴号
    CONNFRAME(29,0, viraxisx,viraxisy,viraxisz,axisr1,axisr2) '第 6/7 轴
    作为虚拟的 XY 轴，启动逆解连接。
    WAIT LOADED '等待运动加载，此时会自动调整虚拟轴的位置。
    ?"逆解模式"
  elseif scan_event(in(0))<0 then '输入 0 下降沿触发
    BASE(viraxisx,viraxisy,viraxisz,axisr1,axisr2) '选择虚拟轴号
    CONNREFRAME(29,0,axisx,axisy,axisz,axisr1,axisr2) '第 0/1 轴作
    为关节轴，启动正解连接。
    WAIT LOADED '等待运动加载。
    ?"正解模式"
  endif
wend
```

FRAME18--单旋转台+偏转

FRAME18 类型与 FRAME11 一致，只是支持旋转台任意角度偏转。此时可选择是否使用偏转后的坐标系，查看[零点位置](#)说明。

结构示意图



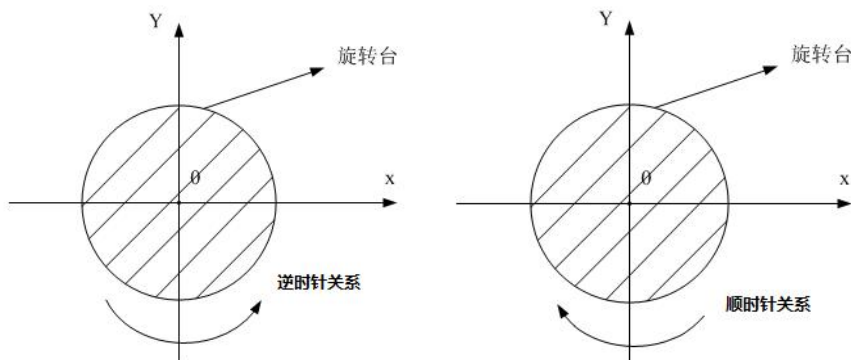
电机方向及角度方向

两个平移轴方向无要求。

旋转台转向从上往下看，根据参数 IfAntiClock 的值确定。

IfAntiClock=1 时，与 XY 为逆时针关系（旋转台从 X 正转向 Y 正）。

IfAntiClock=0 时，与 XY 为顺时针关系（旋转台从 Y 正转向 X 正）。



	角度或移动范围
--	---------

X 轴	机械结构限制
Y 轴	
旋转轴	无限制

轴顺序

实际机械手关节轴：	定义简写：
平移轴 X	Axis_a
平移轴 Y	Axis_b
旋转轴	Axis_c

直角坐标系虚拟轴：	定义简写：
平移轴 X	Viraxis_x
平移轴 Y	Viraxis_y
旋转轴 RZ	Axis_c

指令规定轴号填写次序如下：

逆解时：

```
BASE(Axis_a,Axis_b, Axis_c)
CONNFRAME(18,tablenum,Viraxis_x,Viraxis_y,Axis_c)
```

正解时：

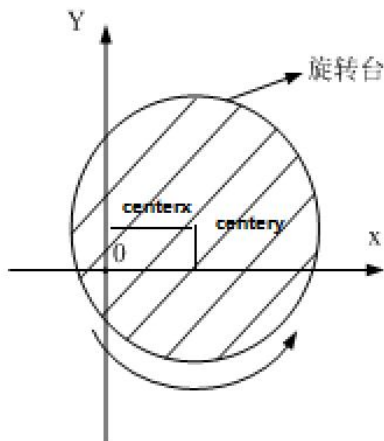
```
BASE(Viraxis_x, Viraxis_y, Axis_c)
CONNREFRAME(18,tablenum, Axis_a,Axis_b, Axis_c)
```

结构参数设置

建立机械手连接时，需要将机械结构参数依次填写到 table 数组中，参数说明如下。

TABLE(Tableunm,CENTERX, CENTERY, pulseonecircle, IFANTICLOCK, centerZ, rx,ry,rz, ifWorkCoor)	
Tablenum	存储转换参数的 TABLE 位置
CENTERX	零点时，xy 平面内，旋转轴的旋转中心与原点 x 方向的偏移
CENTERY	零点时，xy 平面内，旋转轴的旋转中心与原点 y 方向的偏移
pulseRonecircle	旋转轴旋转一圈的脉冲数
IfAntiClock	旋转台与 XY 是否逆时针的关系，逆时针值为 1（旋转台从 X 正转向 Y 正），顺时针值为 0（旋转台从 Y 正转向 Z 正）
centerZ	零点时，yz 平面内，旋转轴的旋转中心与原点 z

	方向的偏移
rx,ry,rz	非标准姿态的旋转台, 相对标准位置旋转的角度,弧度单位
ifWorkCoor	虚拟轴坐标是否使用相对旋转中心的工件坐标, 1 使用, 此时 rx、ry、rz 生效。 0 不使用, 此时 rx、ry、rz 不生效, 与 frame11 标准旋转台相同。



电机参数设置

各轴的轴类型和脉冲当量要设置正确。

一个机械手的所有虚拟轴和关节轴的长度单位要求统一, 一般都是 mm 单位。虚拟轴的一个 mm 的脉冲数一般建议设置为 1000, 表示精度为小数点后 3 位。

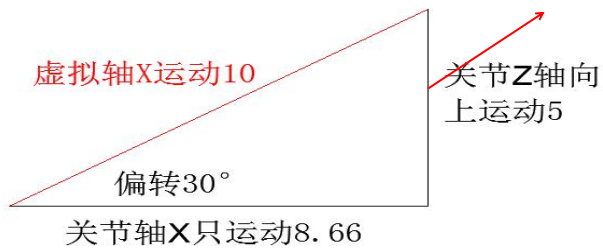
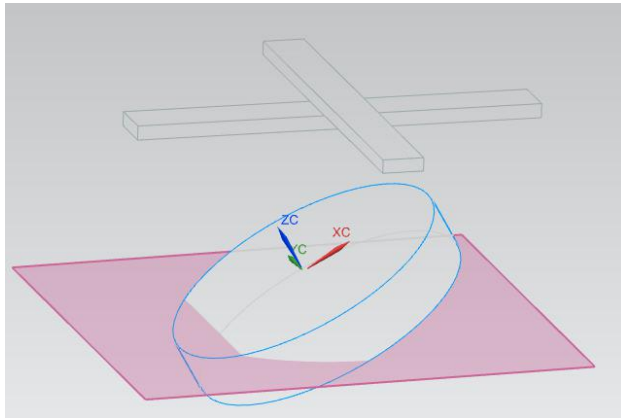
	Atype	Units
Axis_a	1 或 4 或 7 或 65 或 50	每 mm 脉冲数
Axis_b		每 mm 脉冲数
Axis_c		pulseRonecircle/360
Viraxis_x	0	1000
Viraxis_y		1000

零点位置

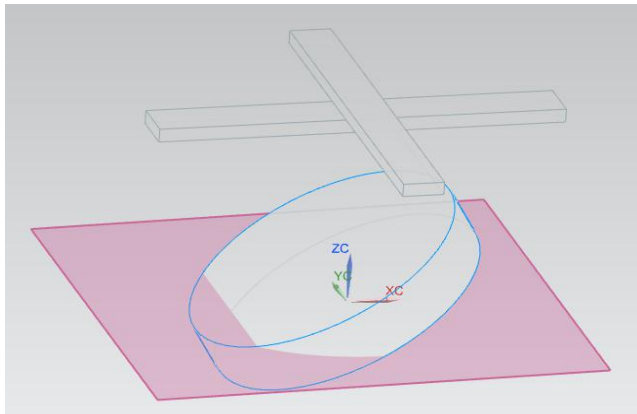
无特殊要求, 当前位置即可作为 0 点, 但是注意此时的圆心偏移参数 CENTERX 和 CENTERY 要设置正确。

假设此时旋转台中心没有偏移, 只绕着 Y 轴偏转 30°。

ifWorkCoor=1, 使用工件坐标系, 此时虚拟轴坐标按工件坐标系, 虚拟坐标系原点为旋转中心。此时操作虚拟轴 X 运动 10, 实际关节轴 X 只运动 8.66, 同时 Z 轴向上运动 5。



ifWorkCoor=0，使用世界坐标系，此时虚拟轴按世界坐标系移动，虚拟坐标系原点为工作点。此时操作虚拟轴 X 运动 10，实际关节轴 X 就运动 10，其他轴不动。



建立连接

正解模式

```
BASE(Viraxis_x, Viraxis_y, Axis_c)
CONNREFRAME(18, tablenum, Axis_a, Axis_b, Axis_c)
WAIT LOADED
```

建立成功后，虚拟轴 MTYPE 将显示为 34，IDLE 显示为 0。

此时只能操作关节轴在关节坐标系中运动，虚拟轴会自动计算末端工作点位于直角坐标系中的位置。

逆解模式

```
BASE(Axis_a,Axis_b,Axis_c)
CONNFRAME(18,tablenum,Viraxis_x,Viraxis_y,Axis_c)
WAIT LOADED
```

建立成功后，关节轴 MTYPE 将显示为 33，IDLE 显示为 0。
此时只能操作虚拟轴在直角坐标系中运动，关节轴会自动计算在关节坐标系中如何联合运动。

使用例程

```
""""""机械参数、点击参数定义
dim axisx,axisy,axisz,axisR '实际关节轴定义
dim viraxisx,viraxisy,viraxisz'虚拟轴定义
axisx=0
axisy=1 '实际关节轴编号
axisz=2
axisR=3
viraxisx=6 '虚拟轴编号
viraxisy=7
viraxisz=8

dim CENTERX 'x 轴旋转中心
dim CENTERY 'y 轴旋转中心
dim CENTERZ '旋转中心 z 坐标
dim IfAntiClock '是否为逆时针旋转，逆时针旋转遵循右手法则
dim ifWorkCoor '旋转台是否使用相对旋转中心的工件坐标，1 使用
dim rx '相对标准位置 x 旋转的角度,弧度单位
dim ry '相对标准位置 y 旋转的角度,弧度单位
dim rz '相对标准位置 z 旋转的角度,弧度单位
CENTERX=0 '旋转中心偏移
CENTERY=0
CENTERZ=0
IfAntiClock=1 '此时为逆时针关系
ifWorkCoor=1 '使用工件坐标
rx=0.5236
ry=0 'RY30°
rz=0

dim u_mx 'x 电机旋转一圈的脉冲
dim u_my 'y 电机旋转一圈的脉冲
```

```

dim u_mz      'z 电机旋转一圈的脉冲
dim u_mR      '旋转电机旋转一圈的脉冲
u_mx=10000
u_my=10000
u_mz=10000
u_mR=10000  '脉冲量赋值

dim i_x      'x 电机传动比
dim i_y      'y 电机传动比
dim i_z      'z 电机传动比
dim i_R      '旋转电机传动比
i_x=1
i_y=1
i_z=1
i_R=1      '传动比赋值

dim u_jx      'x 关节实际一圈脉冲数
dim u_jy      'y 关节实际一圈脉冲数
dim u_jz      'z 关节实际一圈脉冲数
dim u_jR      '旋转关节实际一圈脉冲数
u_jx=u_mx*i_x
u_jy=u_my*i_y
u_jz=u_mz*i_z
u_jR=u_mR*i_R

dim p_x      'x 螺距
dim p_y      'y 螺距
dim p_z      'z 螺距
p_x=2
p_y=2
p_z=2

''''''''定义关节轴
base(axisx,axisy,axisz,axisR)
rapidstop
wait idle      '确保所有轴停止运动
units=u_jx/p_x,u_jy/p_y,u_jz/p_z,u_jR/360'xy 设为 1mm 脉冲数，旋转轴设为 1°
脉冲数
atype = 1,1,1,1      ' 轴功能设置
dpos=0,0,0,0
creep=10,10,10,10      '轴回零速度
speed =300,300,300,300      '移动速度

```



```

accel = 1000,1000,1000,1000  '加速度
decel = 500,500,500,500    '减速度

""""""虚拟轴定义
base(viraxisx,viraxisy,viraxisz)
atype = 0,0,0      '定义虚拟轴
TABLE(0,CENTERX,CENTERY,u_jR,IfAntiClock,CENTERZ,rx,ry,rz,ifWorkCoo
r) 'table 表参数
speed=200,200  '虚拟轴运动速度
accel = 300,300  '虚拟轴运动加速度
units=1000,1000

""""""运动控制开始
merge = on                '连续缓冲开启
CORNER_MODE = 2+32        '设置启动拐角减速+倒角
ZSMOOTH=10                '倒角半径设置
DECEL_ANGLE = 15 * (PI/180)  '开始减速的最小拐角
STOP_ANGLE = 45 * (PI/180)  '减速到最低的减速拐角

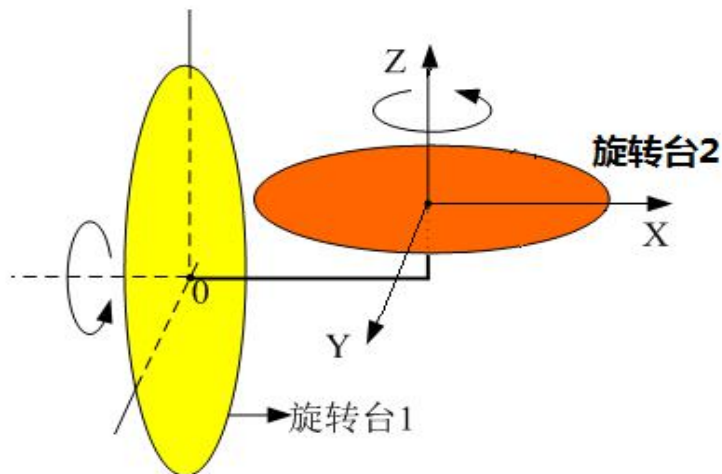
""建立机械手连接
while 1
    if scan_event(in(0))>0 then  '输入 0 上升沿触发
        BASE(axisx,axisy,axisz,axisR)                '选择关节轴号
        CONNFRAME(18,0,viraxisx, viraxisy,viraxisz,axisR)  '启动逆解连
接。
        WAIT LOADED                                     '等待运动加载，此时
会自动调整虚拟轴的位置。
        ?"逆解模式"
    elseif scan_event(in(0))<0 then  '输入 0 下降沿触发
        BASE(viraxisx, viraxisy,viraxisz,axisR)        '选择虚拟轴
号
        CONNREFRAME(18,0,axisx,axisy,axisz,axisR)      '启动正解连
接。
        WAIT LOADED                                     '等待运动加载。
        ?"正解模式"
    endif
wend

```

FRAME19--双旋转台+偏转

机械结构与 FRAME17 类同，主要区别在旋转台 2 可以角度偏转此时可选择是否使用偏转后的坐标系，查看[零点位置](#)说明。

结构示意图

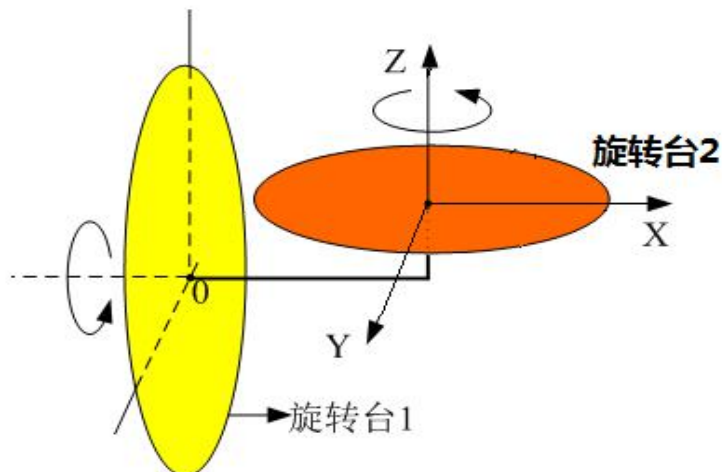


电机方向及角度范围

关节轴 XYZ 方向没有要求。

旋转台 1 方向根据实际情况设置 IfAntiClock1 参数：逆时针值为 1（旋转台从 Y 正转向 Z 正），顺时针值为 0（旋转台从 Z 正转向 Y 正）。

旋转台 2 方向根据实际情况设置 IfAntiClock2 参数：逆时针值为 1（旋转台从 X 正转向 Y 正），顺时针值为 0（旋转台从 Y 正转向 X 正）。



	角度或移动范围
X 轴	机械结构限制
Y 轴	
Z 轴	
RX 旋转轴	无限制
Rz 旋转轴	

轴顺序

实际机械手关节轴：	定义简写：
平移轴 X	Axis_a
平移轴 Y	Axis_b
平移轴 Z	Axis_c
旋转台 1	Axis_d
旋转台 2	Axis_e

直角坐标系虚拟轴：	定义简写：
平移轴 X	Viraxis_x
平移轴 Y	Viraxis_y
平移轴 Z	Viraxis_z
旋转轴 RX	Axis_d
旋转轴 RZ	Axis_e

指令规定轴号填写次序如下：

逆解时：

```
BASE(Axis_a,Axis_b, Axis_c,Axis_d,Axis_e)
CONNFRAME(19,tablenum, Viraxis_x, Viraxis_y, Viraxis_v, Axis_d, Axis_e)
```

正解时：

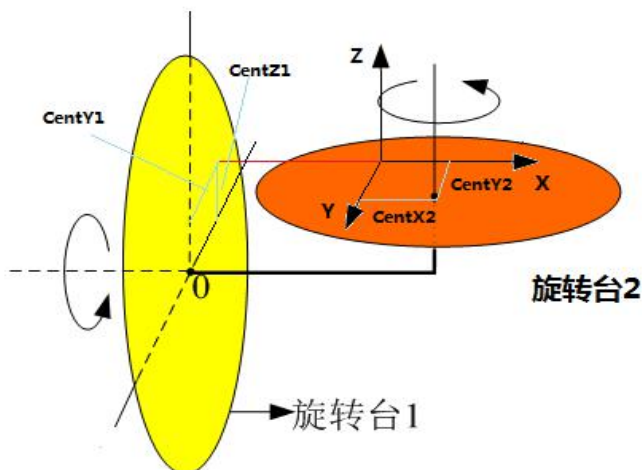
```
BASE(Viraxis_x, Viraxis_y, Viraxis_v, Axis_d, Axis_e)
CONNREFRAME(19,tablenum, Axis_a, Axis_b, Axis_c, Axis_d, Axis_e)
```

结构参数设置

建立机械手连接时，需要将机械结构参数依次填写到 table 数组中，参数说明如下。

```
TABLE(Tablenum ,CentY1, CentZ1, pulse1onecircle, IFANTICLOCK1,
CentX2, CentY2, pulse2onecircle, IFANTICLOCK2, centx1, centz2, rx, ry, rz,
ifWorkCoor )
```

Tablenum	存储转换参数的 TABLE 位置
CentY1	零点时, yz 平面内, rx 旋转轴的旋转中心与原点 y 方向的偏移
CentZ1	零点时, yz 平面内, rx 旋转轴的旋转中心与原点 z 方向的偏移
pulse1onecircle	rx 旋转轴旋转一圈的脉冲数
IfAntiClock1	旋转台 1 与 YZ 是否逆时针的关系, 逆时针值为 1 (旋转台从 Y 正转向 Z 正), 顺时针值为 0 (旋转台从 Z 正转向 Y 正)
CentX2	零点时, xy 平面内, rz 旋转轴的旋转中心与原点 x 方向的偏移
CentY2	零点时, xy 平面内, rz 旋转轴的旋转中心与原点 y 方向的偏移
pulse2onecircle	rz 旋转轴旋转一圈的脉冲数
IfAntiClock2	旋转台 2 与 XY 是否逆时针的关系, 逆时针值为 1, 顺时针值为 0, 逆时针的时候符合右手法则
centx1	rx 旋转轴中心的 x 坐标
centz2	rz 旋转轴中心的 Z 坐标。
rx, ry, rz	非标准姿态的旋转台, 相对标准位置旋转的角度
ifWorkCoor	虚拟轴坐标是否使用相对旋转中心的工件坐标, 参看 FRAME18 1 使用, 此时 rx、ry、rz 生效 0 不使用, 此时 rx、ry、rz 不生效, 与 frame11 标准旋转台相同



电机参数设置

各轴的轴类型和脉冲当量要设置正确。

一个机械手的所有虚拟轴和关节轴的长度单位要求统一, 一般都是 mm 单位。

虚拟轴的一个 mm 的脉冲数一般建议设置为 1000，表示精度为小数点后 3 位。

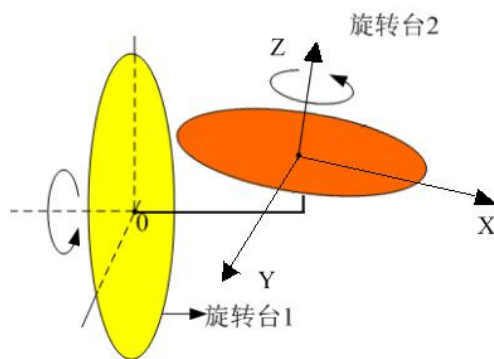
	Atype	Units
Axis_a	1 或 4 或 7 或 65 或 50	每 mm 脉冲数
Axis_b		每 mm 脉冲数
Axis_c		每 mm 脉冲数
Axis_d		Pulse1oncircle/360
Axis_e		Pulse2oncircle/360
Viraxis_x	0	1000
Viraxis_y		1000
Viraxis_z		1000

零点位置

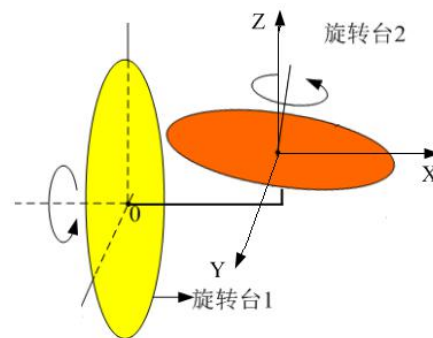
零点位置[查看 FRAME17 相关说明。](#)

坐标系旋转说明[查看 FRAME18 相关说明。](#)

使用工件坐标系



使用世界坐标系



建立连接

正解模式

```
BASE(Viraxis_x, Viraxis_y, Viraxis_z, Axis_d, Axis_e)
CONNREFRAME(19, tablenum, Axis_a, Axis_b, Axis_cs, Axis_d, Axis_e)
WAIT LOADED
```

建立成功后，虚拟轴 MTYPE 将显示为 34，IDLE 显示为 0。

此时只能操作关节轴在关节坐标系中运动，虚拟轴会自动计算末端工作点位于直角坐标系中的位置。

逆解模式

```
BASE(Axis_a, Axis_b, Axis_c, Axis_d, Axis_e)
CONNFRAME(19, tablenum, Viraxis_x, Viraxis_y, Viraxis_z, Axis_d, Axis_e)
```

WAIT LOADED

建立成功后，关节轴 MTYPE 将显示为 33，IDLE 显示为 0。
此时只能操作虚拟轴在直角坐标系中运动，关节轴会自动计算在关节坐标系中如何联合运动。

使用例程

```
'=====机械参数、电机参数定义=====
dim axisx,axisy,axisz,axisr1,axisr2    "定义实际轴
dim viraxisx,viraxisy,viraxisz        "定义虚拟轴
axisx=0    ""实际轴编号
axisy=1
axisz=2
axisr1=3    ""旋转轴编号
axisr2=4

viraxisx=6    ""虚拟轴编号
viraxisy=7
viraxisz=8

dim centy1,centz1,centx2,centy2,centx1,centz2    '旋转中心定义
dim ifanticlock1,ifanticlock2    '是否逆时针
dim rx,ry,rz    '角度偏转，弧度单位
dim ifWorkCoor    '旋转台是否使用相对旋转中心的工件坐标，1 使用
centy1=10    '旋转中心偏移
centz1=10
centx2=30
centy2=20
centx1=0
centz2=0
ifanticlock1=1    '旋转台 1 逆时针
ifanticlock2=1    '旋转台 2 逆时针
rx=0
ry=0.5236    'RY30°
rz=0
ifWorkCoor=1

dim u_mx,u_my,u_mz    '轴旋转一圈的脉冲
dim u_mr1,u_mr2
u_mx=10000    '旋转轴脉冲参数赋值
u_my=10000
```

```

u_mz=10000
u_mr1=10000
u_mr2=10000

dim i_x,i_y,i_z      '电机传动比
dim i_r1,i_r2
i_x=2
i_y=2
i_z=2
i_r1=1
i_r2=1

dim u_jx      'x 关节实际一圈脉冲数
dim u_jy      'y 关节实际一圈脉冲数
dim u_jz      'z 关节实际一圈脉冲数
dim u_jr1     '旋转 1 实际一圈脉冲数
dim u_jr2     '旋转 2 实际一圈脉冲数
u_jx=u_mx*i_x
u_jy=u_my*i_y
u_jz=u_mz*i_z
u_jr1=u_mr1*i_r1
u_jr2=u_mr2*i_r2

dim p_x      'x 螺距
dim p_y      'y 螺距
dim p_z      'z 螺距
p_x=2
p_y=2
p_z=10

''''''''定义关节轴
BASE(axisx,axisy,axisz)      '实际轴定义
ATYPE=1,1,1                  '轴运动类型
rapidstop
wait idle                    '确保所有轴停止运动
UNITS=u_jx/p_x,u_jy/p_y,u_jz/p_z  '设置为 1mm 的脉冲数
Dpos=0,0,0
speed=500,500,500
CREEP=20,20,20
ACCEL=1000,1000,1000
DECEL=500,500,500

base(axisr1,axisr2)          '旋转轴设置

```

```

ATYPE=1,1          '轴运动类型
units =u_jr1/360,u_jr1/360  '设置为 1°的脉冲数
Dpos=0,0
speed=200,200      '末端旋转轴参数设置
accel=300,300

""""""虚拟轴定义
base(viraxisx,viraxisy,viraxisz)  '虚拟轴定义
ATYPE=0,0,0
TABLE(0,CentY1,CentZ1,u_jr1,IfAntiClock1,CentX2,CentY2,u_jr2,IfAntiClock2,
centx1,centz2,rx,ry,rz,ifWorkCoor)  ""table 表参数填写
UNITS=1000,1000,1000  ""虚拟轴运动参数赋值
SPEED=300,300,300
ACCEL=500,500,500

""""""运动控制开始
merge = on          '连续缓冲开启
CORNER_MODE =2+32  '设置启动拐角减速+倒角
ZSMOOTH=10         '倒角半径设置
DECEL_ANGLE = 15 * (PI/180)  '开始减速的最小拐角
STOP_ANGLE = 45 * (PI/180)  '减速到最低的减速拐角

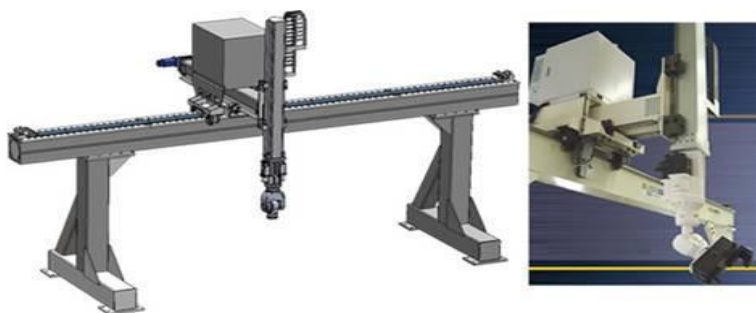
""建立机械手连接
while 1
  if scan_event(in(0))>0 then  '输入 0 上升沿触发
    BASE(axisx,axisy,axisz,axisr1,axisr2)  '选择关节轴号
    CONNFRAME(19,0,viraxisx,viraxisy,viraxisz,axisr1,axisr2)  '启动逆解连接。
    WAIT LOADED  '等待运动加载，此时会自动调整虚拟轴的位置。
    ?"逆解模式"
  elseif scan_event(in(0))<0 then  '输入 0 下降沿触发
    BASE(viraxisx,viraxisy,viraxisz,axisr1,axisr2)  '选择虚拟轴号
    CONNREFRAME(19,0,axisx,axisy,axisz,axisr1,axisr2)  '启动正解连接。
    WAIT LOADED  '等待运动加载。
    ?"正解模式"
  endif
wend

```


直角坐标类

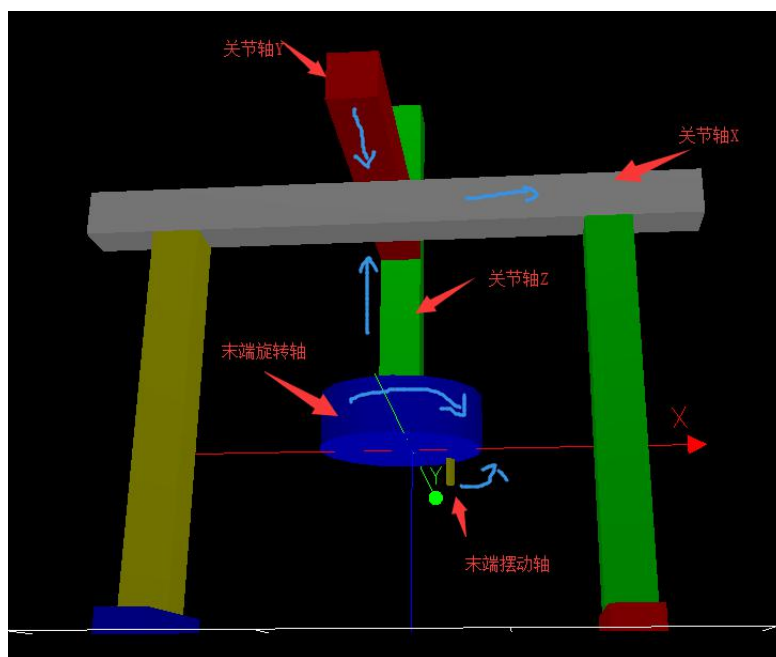
FRAME100/104--XYZ+2 轴手腕

结构示意图



电机方向及角度范围

各关节电机正向如下图所示蓝色箭头所示。



	角度或移动范围
关节轴 x	机械结构限制
关节轴 y	
关节轴 z	
末端旋转轴	无限制
末端摆动轴	机械结构限制

轴顺序

实际机械手关节轴:	定义简写:
关节轴 x	Axis_a
关节轴 y	Axis_b
关节轴 z	Axis_c
末端旋转轴	Axis_d
末端摆动轴	Axis_e

直角坐标系虚拟轴:	定义简写:
平移轴 X	Viraxis_x
平移轴 Y	Viraxis_y
平移轴 Z	Viraxis_z
旋转轴 RY	Viraxis_v (FRAME104)
旋转轴 RZ	Viraxis_w (FRAME104)

指令规定轴号填写次序如下:

逆解时:

<p>FRAME100 类型 BASE(Axis_a,Axis_b, Axis_c,Axis_d,Axis_e) CONNFRAME(100,tablenum,Viraxis_x,Viraxis_y,Viraxis_z,Axis_d,Axis_e)</p> <p>FRAME104 类型 BASE(Axis_a,Axis_b, Axis_c,Axis_d,Axis_e) CONNFRAME(104,tablenum,Viraxis_x,Viraxis_y,Viraxis_z,Viraxis_v,Viraxis_w)</p>

正解时:

<p>FRAME100 类型 BASE(Viraxis_x, Viraxis_y, Viraxis_z,Axis_d,Axis_e) CONNREFRAME(100,tablenum, Axis_a,Axis_b, Axis_c,Axis_d,Axis_e)</p> <p>FRAME104 类型 BASE(Viraxis_x, Viraxis_y, Viraxis_z,Viraxis_v,Viraxis_w) CONNREFRAME(104,tablenum, Axis_a,Axis_b, Axis_c,Axis_d,Axis_e)</p>

结构参数设置

建立机械手连接时，需要将机械结构参数依次填写到 table 数组中，参数说明如下。

TABLE(tablenum, Lw, PulesVOneCircle, PulesWOneCircle, Lvwx, Lvwy)	
Tablenum	存储转换参数的 TABLE 位置
Lw	摆动轴长度
PulesVOneCircle	末端旋转轴旋转一圈的脉冲数
PulesWOneCircle	末端摆动轴旋转一圈的脉冲数
Lvwx	零点时末端与 V 旋转中心的 X 偏移
Lvwy	零点时末端与 V 旋转中心的 y 偏移

电机参数设置

各轴的轴类型和脉冲当量要设置正确。

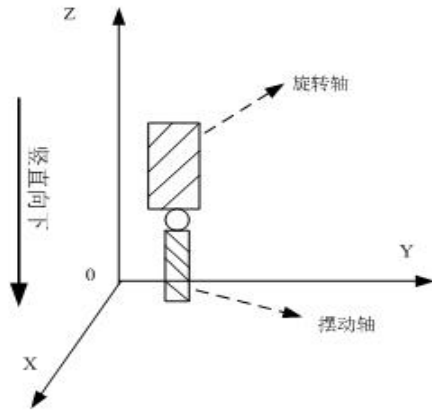
一个机械手的所有虚拟轴和关节轴的长度单位要求统一，一般都是 mm 单位。虚拟轴的一个 mm 的脉冲数一般建议设置为 1000，表示精度为小数点后 3 位。

	Atype	Units
Axis_a	1 或 4 或 7 或 65 或 50	每 mm 脉冲数
Axis_b		每 mm 脉冲数
Axis_c		每 mm 脉冲数
Axis_d		PulesVOneCircle/360
Axis_e		PulesWOneCircle/360
Viraxis_x	0	1000
Viraxis_y		1000
Viraxis_z		1000
Viraxis_w		PulesVOneCircle/360
Viraxis_v		PulesWOneCircle/360

零点位置

对关节轴 XYZ 没有要求，当前位置即为 0 点。

摆动轴竖直向下时为 0 点位置。



建立连接

正解模式

FRAME100 类型

```
BASE(Viraxis_x, Viraxis_y, Viraxis_z, Axis_d, Axis_e)
CONNREFRAME(100, tablenum, Axis_a, Axis_b, Axis_c, Axis_d, Axis_e)
WAIT LOADED
```

FRAME104 类型

```
BASE(Axis_a, Axis_b, Axis_c, Axis_d, Axis_e)
CONNFRAME(104, tablenum, Viraxis_x, Viraxis_y, Viraxis_z, Viraxis_v, Viraxis_w)
WAIT LOADED
```

建立成功后，虚拟轴 MTYPE 将显示为 34，IDLE 显示为 0。

此时只能操作关节轴在关节坐标系中运动，虚拟轴会自动计算末端工作点位于直角坐标系中的位置。

逆解模式

FRAME100 类型

```
BASE(Axis_a, Axis_b, Axis_c, Axis_d, Axis_e)
CONNFRAME(100, tablenum, Viraxis_x, Viraxis_y, Viraxis_z, Axis_d, Axis_e)
WAIT LOADED
```

FRAME104 类型

```
BASE(Axis_a, Axis_b, Axis_c, Axis_d, Axis_e)
CONNFRAME(104, tablenum, Viraxis_x, Viraxis_y, Viraxis_z, Viraxis_v, Viraxis_w)
WAIT LOADED
```

建立成功后，关节轴 MTYPE 将显示为 33，IDLE 显示为 0。

此时只能操作虚拟轴在直角坐标系中运动，关节轴会自动计算在关节坐标系中如何联合运动。

使用例程

此例程为 FRAME100 类型，FRAME104 类型注意把旋转轴和摆动轴用虚拟轴操作。

```

"电机、机械手变量参数定义
dim axisx,axisy,axisz,axisv,axisw      '关节轴定义
dim viraxisx,viraxisy,viraxisz '虚拟轴定义
axisx=0      '关节 x 轴编号
axisy=1      '关节 Y 轴编号
axisz=2      '关节 Z 轴编号
axisv=3      '垂直旋转轴编号
axisw=4      '摆动旋转轴编号
viraxisx=5   '虚拟 X 轴编号
viraxisy=6   '虚拟 Y 轴编号
viraxisz=7   '虚拟 Z 轴编号

dim Lw      '摆动轴长度定义
dim Lvwx    'Lvwx:零点时末端与 V 旋转中心的 X 偏移，反向为负数。
dim Lvwy    'Lvwy:零点时末端与 V 旋转中心的 Y 偏移，反向为负数。
Lw=12      '摆动轴长度
Lvwx=20
Lvwy=20      '末端偏移量赋值

dim u_mr1,  '垂直旋转电机一圈脉冲数。
dim u_mr2  '摆动旋转电机一圈脉冲数。
dim u_mx    'x 轴电机一圈脉冲数
dim u_my    'y 轴电机一圈脉冲数
dim u_mz    'z 轴电机一圈脉冲数
u_mr1=10000
u_mr2=10000
u_mx=10000
u_my=10000
u_mz=10000

dim i_r1    '垂直旋转电机传动比
dim i_r2    '摆动旋转电机传动比
dim i_x     'x 轴电机传动比
dim i_y     'y 轴电机传动比
dim i_z     'z 轴电机传动比
i_r1=2

```

```

i_r2=2
i_x=2
i_y=2
i_z=2

dim u_jx      'x 实际一圈脉冲数
dim u_jy      'y 实际一圈脉冲数
dim u_jz      'z 实际一圈脉冲数
dim u_jr1     'r1 实际一圈脉冲数
dim u_jr2     'r2 实际一圈脉冲数
u_jx=u_mx*i_x
u_jy=u_my*i_y
u_jz=u_mz*i_z
u_jr1=u_mr1*i_r1
u_jr2=u_mr2*i_r2

dim p_x      'x 螺距
dim p_y      'y 螺距
dim p_z      'z 螺距
p_x=1.5
p_y=1.5
p_z=3

"""关节轴设置
base(axisx,axisy,axisz,axisv,axisw)          '定义关节轴
rapidstop
wait idle          '确保所有轴停止运动

units=u_jx/p_x,u_jy/p_y,u_jz/p_z,u_jr1/360,u_jr2/360
      'xyz 设置 1mm 为单位，两个旋转轴设置 1°为单位
atype = 1,1,1,1,1          '轴功能设置
clutch_rate =0,0,0,0,0      '各轴连接比率
creep=10,10,10,10,10      '轴回零速度
speed =300,300,300,300,300  '移动速度
accel = 1000,1000,100,100,100  '加速度
decel = 500,500,500,500,500  '减速度

"""虚拟轴定义
base(viraxisx,viraxisy,viraxisz)
atype = 0,0,0,          '定义虚拟轴
TABLE(0, Lw,u_jr1,u_jr2,Lvwx,Lvwy) 'table 表参数
speed=200,200,200      '虚拟轴运动速度
accel = 300,300,300    '虚拟轴运动加速度

```

```
units=1000,1000,1000

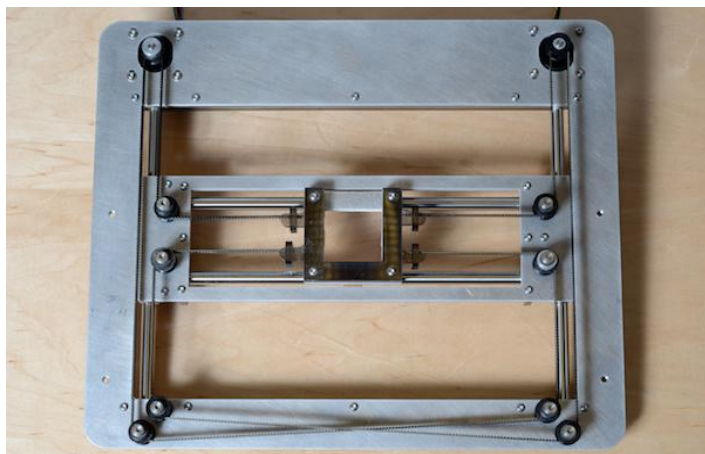
""运动控制开始
merge = on                '连续缓冲开启
CORNER_MODE =2+32        '设置启动拐角减速+倒角
ZSMOOTH=10                '倒角半径设置
DECEL_ANGLE = 15 * (PI/180) '开始减速的最小拐角
STOP_ANGLE = 45 * (PI/180) '减速到最低的减速拐角

while 1
  if scan_event(in(0))>0 then '输入 0 上升沿触发
    BASE(axisx,axisy,axisz,axisv,axisw) '选择关节轴号
    CONNFRAME(100,0,viraxisx,viraxisy,viraxisz,axisv,axisw) '启动逆解连
接。
    WAIT LOADED '等待运动加载，此时
    会自动调整虚拟轴的位置。
    ?"逆解模式"
  elseif scan_event(in(0))<0 then '输入 0 下降沿触发
    BASE(viraxisx,viraxisy,viraxisz,axisv,axisw) '选择虚拟轴号
    CONNREFRAME(100,0,axisx,axisy,axisz,axisv,axisw) '启动正解连
接。
    WAIT LOADED '等待运动加载。
    ?"正解模式"
  endif
wend
```


其余类

FRAME15--CORE XY 滑台

结构示意图



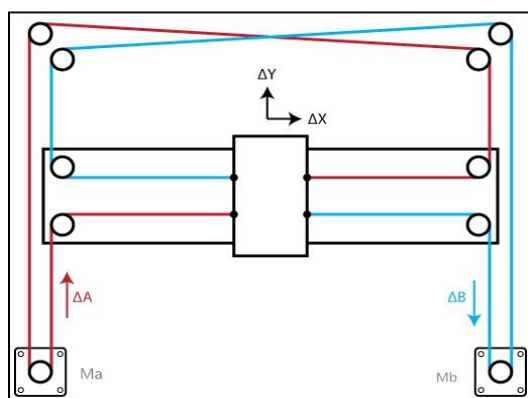
电机方向及角度范围

Ma 逆时针旋转, Mb 顺时针旋转, 向+Y 轴运动;

Ma 顺 Mb 逆, 向-Y 轴运动;

Ma 顺 Mb 顺, 向-X 轴运动;

Ma 逆 Mb 逆, 向+X 轴运动。



	角度或移动范围
Ma 电机	无限制
Mb 电机	

轴顺序

实际机械手关节轴:	定义简写:
关节轴 Ma 电机	Axis_a
关节轴 Mb 电机	Axis_b

直角坐标系虚拟轴:	定义简写:
平移轴 X	Viraxis_x
平移轴 Y	Viraxis_y

指令规定轴号填写次序如下:

逆解时:

```
BASE(Axis_a,Axis_b)
CONNFRAME(15,tablenum, Viraxis_x, Viraxis_y)
```

正解时:

```
BASE(Viraxis_x, Viraxis_y)
CONNREFRAME(15,tablenum, Axis_a,Axis_b)
```

结构参数设置

建立机械手连接时，需要将机械结构参数依次填写到 table 数组中，参数说明如下。

TABLE(tablenum,J1Xdir, J1Ydir, J2Xdir, J2Ydir)	
Tablenum	存储转换参数的 TABLE 位置
J1Xdir	关节轴 1 正向时 X 方向, 1 正向, -1 反向
J1Ydir	关节轴 1 正向时 Y 方向
J2Xdir	关节轴 2 正向时 X 方向
J2Ydir	关节轴 2 正向时 Y 方向

电机参数设置

各轴的轴类型和脉冲当量要设置正确。

一个机械手的所有虚拟轴和关节轴的长度单位要求统一，一般都是 mm 单位。虚拟轴的一个 mm 的脉冲数一般建议设置为 1000，表示精度为小数点后 3 位。

	Atype	Units
Axis_a	1 或 4 或 7 或 65 或 50	每 mm 脉冲数

Axis_b		每 mm 脉冲数
Viraxis_x	0	1000
Viraxis_y		1000

零点位置

此结构对零点位置无要求，当前停止位置作为零点。

建立连接

正解模式

```
BASE(Viraxis_x, Viraxis_y)
CONNREFRAME(15,tablenum, Axis_a,Axis_b)
WAIT LOADED
```

建立成功后，虚拟轴 MTYPE 将显示为 34，IDLE 显示为 0。

此时只能操作关节轴在关节坐标系中运动，虚拟轴会自动计算末端工作点位于直角坐标系中的位置。

逆解模式

```
BASE(Axis_a,Axis_b)
CONNFRAME(15,tablenum, Viraxis_x, Viraxis_y)
WAIT LOADED
```

建立成功后，关节轴 MTYPE 将显示为 33，IDLE 显示为 0。

此时只能操作虚拟轴在直角坐标系中运动，关节轴会自动计算在关节坐标系中如何联合运动。

使用例程

```
""""电机、机械结构参数定义
dim axis1,axis2      '关节轴变量定义
dim viraxisx,viraxisy '虚拟轴变量定义
axis1=0             "关节轴编号
axis2=1
viraxisx=5          "虚拟轴编号
viraxisy=6

dim J1Xdir, J1Ydir, J2Xdir, J2Ydir 'table 参数变量定义
```

```

J1Xdir=1      "table 参数变量赋值(1 或-1)
J1Ydir=1
J2Xdir=-1
J2Ydir=1

dim u_m1      '电机 1 一圈脉冲数
dim u_m2      '电机 2 一圈脉冲数
u_m1=10000
u_m2=10000

dim p_1       '电机 1 一圈螺距
dim p_2       '电机 2 一圈螺距
p_1=2
p_2=2

"关节轴定义
BASE(axis1,axis2)
atype=1,1      '轴类型设置
RAPIDSTOP      '所有轴停止
WAIT IDLE
units=u_m1/p_1,u_m1/p_1  '关节轴脉冲当量
SPEED=100,100  '运动速度
ACCEL=1000,1000 '加速度
DECEL=500,500  '减速度
DPOS=0,0       '先手动调整位置，再坐标清零

"虚拟轴定义
base(viraxisx,viraxisy)
ATYPE=0,0      '轴类型设置
UNITS=1000,1000 '虚拟轴脉冲当量
SPEED=100,100  '运动速度
ACCEL=500,500  '加速度
DECEL=300,300  '减速度
TABLE(0,J1Xdir,J1Ydir,J2Xdir,J2Ydir) 'table 参数配置

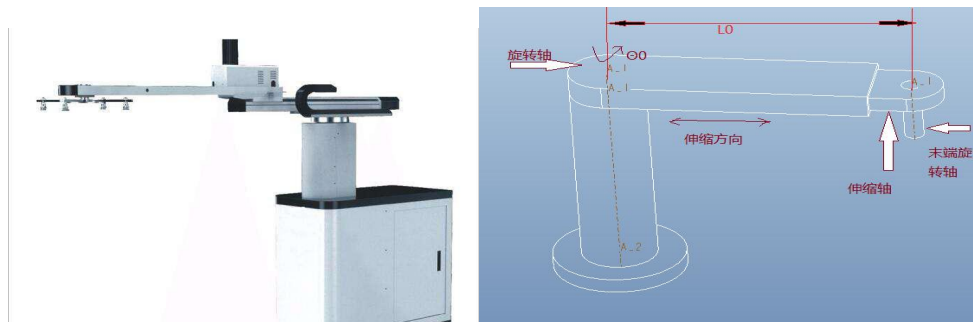
"运动控制开始
while 1
  if scan_event(in(0))>0 then      '输入 0 上升沿触发
    BASE(axis1,axis2)              '配置关节轴
    CONNFRAME(15,0,viraxisx,viraxisy) '启动逆解连接。
    wait loaded                    '等待运动加载，此时会自动调整虚拟轴的位置'运动控制

```

```
? "逆解模式"  
elseif scan_event(in(0)) < 0 then    '输入 0 下降沿触发  
    base(viraxisx, viraxisy)         '选择虚拟轴号  
    CONNREFRAME(15, 0, axis1, axis2) '启动正解连接  
    wait loaded  
    ? "正解模式"  
endif  
wend
```

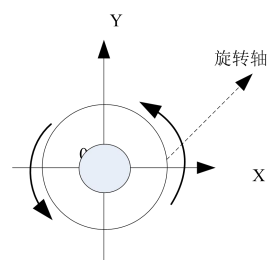
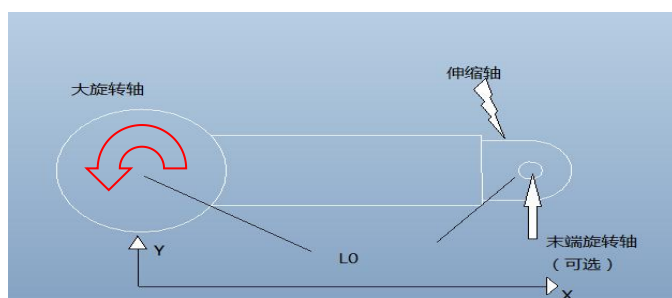
FRAME5--旋转伸缩机械手

结构示意图



电机方向及角度范围

大旋转轴和末端旋转轴逆时针旋转时为正向（俯视）。
伸缩轴伸出时为正向。



	角度或移动范围
大旋转轴	无限制
伸缩轴	机械结构限制
末端旋转轴	无限制

轴顺序

实际机械手关节轴:	定义简写:
大旋转轴	Axis_a
伸缩轴	Axis_b
末端旋转轴	Axis_c

直角坐标系虚拟轴:	定义简写:
平移轴 X	Viraxis_x
平移轴 Y	Viraxis_y
旋转轴 RZ	Viraxis_v

指令规定轴号填写次序如下:

逆解时:

```
BASE(Axis_a,Axis_b[,Axis_c])
CONNFRAME(5,tablenum, Viraxis_x, Viraxis_y[,Viraxis_v])
```

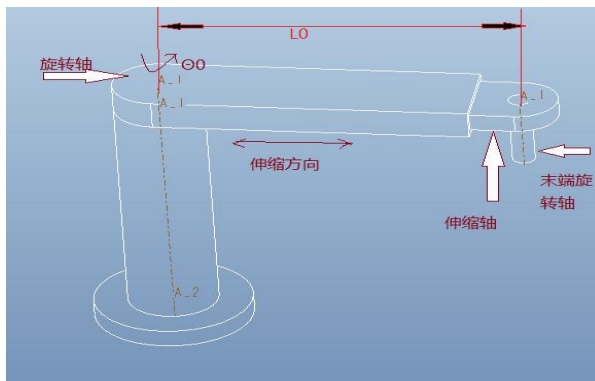
正解时:

```
BASE(Viraxis_x, Viraxis_y[,Viraxis_v])
CONNREFRAME(5,tablenum, Axis_a,Axis_b[,Axis_c])
```

结构参数设置

建立机械手连接时，需要将机械结构参数依次填写到 table 数组中，参数说明如下。

TABLE(tablenum, L0, Pules1OneCircle, Pules2OneUnit, [Pules3OneCircle])	
Tablenum	存储转换参数的 TABLE 位置
L0	水平伸缩轴零点时的长度
Pules1OneCircle	底座旋转轴旋转一圈的脉冲数
Pules2OneUnit	水平伸缩轴走 1mm 的脉冲数
Pules3OneCircle	末端旋转轴旋转一圈的脉冲数



电机参数设置

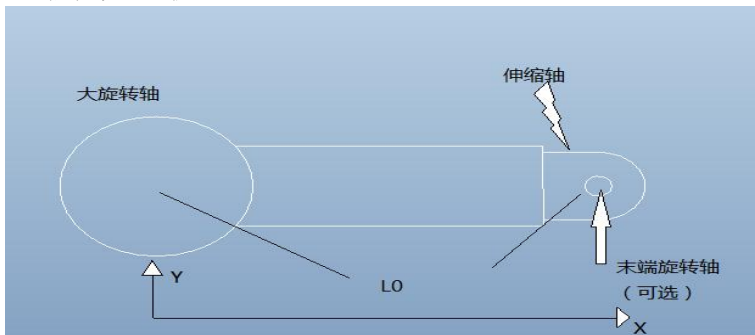
各轴的轴类型和脉冲当量要设置正确。

一个机械手的所有虚拟轴和关节轴的长度单位要求统一，一般都是 mm 单位。虚拟轴的一个 mm 的脉冲数一般建议设置为 1000，表示精度为小数点后 3 位。

	Atype	Units
Axis_a	1 或 4 或 7 或 65 或 50	Pules1OneCircle/360
Axis_b		Pules2OneUnit
Axis_c		Pules3OneCircle/360
Viraxis_x	0	1000
Viraxis_y		1000
Viraxis_v		Pules3OneCircle/360

零点位置

大旋转轴和末端旋转轴无要求，当前位置即为 0 点。
伸缩轴完全收回时为 0 点。



建立连接

正解模式

```
BASE(Viraxis_x, Viraxis_y[,Viraxis_v])
CONNREFRAME(5,tablenum, Axis_a,Axis_b[,Axis_c])
WAIT LOADED
```

建立成功后，虚拟轴 MTYPE 将显示为 34，IDLE 显示为 0。

此时只能操作关节轴在关节坐标系中运动，虚拟轴会自动计算末端工作点位于直角坐标系中的位置。

逆解模式

```
BASE(Axis_a,Axis_b[,Axis_c])
CONNFRAME(5,tablenum, Viraxis_x, Viraxis_y[,Viraxis_v])
WAIT LOADED
```

建立成功后，关节轴 MTYPE 将显示为 33，IDLE 显示为 0。

此时只能操作虚拟轴在直角坐标系中运动，关节轴会自动计算在关节坐标系中如何联合运动。

使用例程

```

'=====电机、机械手参数定义=====

dim L0 '伸缩轴零点时的长度
L0=10

dim u_m1 '关节轴一 一圈脉冲数，大旋转轴
dim u_m2 '关节轴二 一圈脉冲数，伸缩轴
dim u_m3 '末端旋转轴一圈脉冲数
u_m1=10000
u_m2=10000
u_m3=10000

dim i_1 '关节 1 传动比
dim i_2 '关节 2 传动比
dim i_3 '末端旋转轴传动比
i_1=1
i_2=1
i_3=1

dim u_j1 '关节一实际一圈脉冲数
dim u_j2 '关节二实际一圈的脉冲数
dim u_j3 '末端旋转轴实际一圈脉冲数
u_j1=u_m1*i_1
u_j2=u_m2*i_2
u_j3=u_m3*i_3

dim p_2 '伸缩轴螺距
p_2=2

'=====关节轴设置=====

BASE(0,1,2) '选择关节轴号
atype=1,1,1 '轴类型设为脉冲轴
UNITS=u_j1/360,u_j1/p_2,u_j3/360 '把 units 设成每°的脉冲数，其伸缩轴
为一个 mm 的脉冲数

DPOS=0,0,0 '设置关节轴的位置，此处要根据实际情况来修改。
speed=100,100,100 '速度参数设置
accel=1000,1000,1000 '加速度参数设置
decel=1000,1000,1000 '减速度参数设置

```

```

CLUTCH_RATE=0,0,0      '使用关节轴的速度和加速度限制

'=====虚拟轴设置=====

BASE(6,7,8)
ATYPE=0,0,0            '设置为虚拟轴
UNITS=1000,1000,u_j3/360  '此脉冲当量要提前设置，中途不能变化
speed=100,100,100      '速度参数设置
accel=1000,1000,1000   '加速度参数设置
decel=1000,1000,1000   '减速度参数设置
TABLE(0,L0,u_j1,u_j2/p_2,u_j3)  '根据手册说明填写参数
merge = on             '打开连续插补状态
CORNER_MODE =2+32     '启动拐角减速和倒角
ZSMOOTH=10            '倒角半径设置
DECEL_ANGLE = 15 * (PI/180)  '开始减速的角度 15 度
STOP_ANGLE = 45 * (PI/180)  '降到最低速度的角度 45 度

'=====建立机械手连接=====

while 1
  if scan_event(in(0))>0 then  '输入 0 上升沿触发
    BASE(0,1,2)                '选择关节轴号
    CONNFRAME(5,0,6,7,8)      '第 6/7 轴作为虚拟的 XY 轴，启动逆
    解连接。
    WAIT LOADED                '等待运动加载，此时会自动调整虚拟轴的位置。
    ?"逆解模式"

  elseif scan_event(in(0))<0 then  '输入 0 下降沿触发
    BASE(6,7,8)                '选择虚拟轴号
    CONNREFRAME(5,0,0,1,2)     '第 0/1 轴作为关节轴，启动正解连接。
    WAIT LOADED                '等待运动加载，手动操作实际轴
    ?"正解模式"

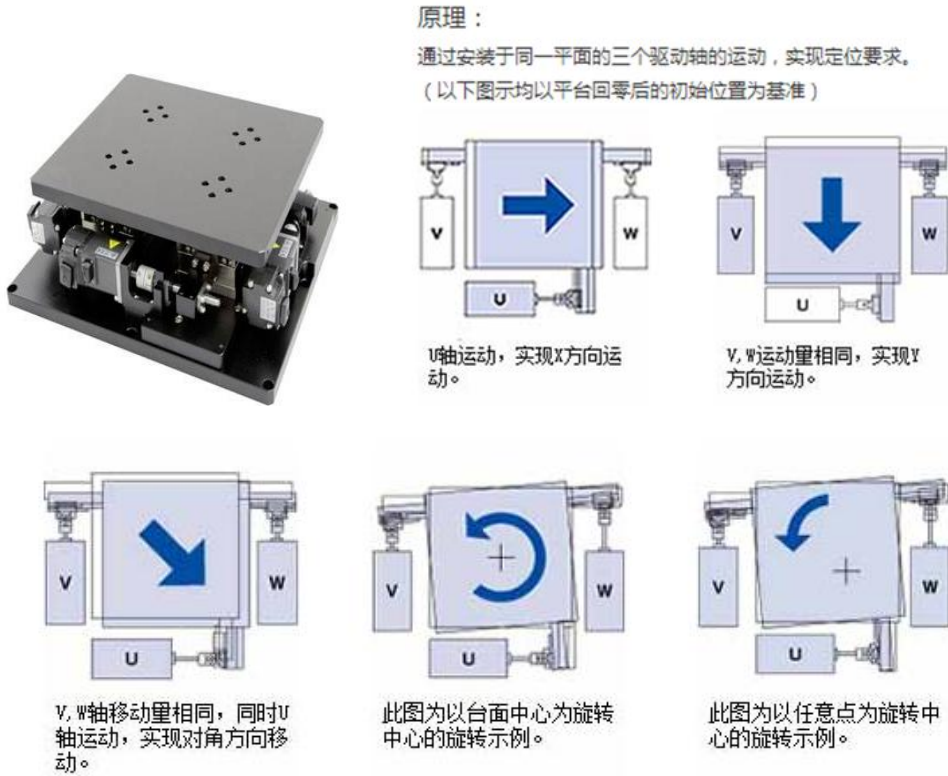
  endif

wend

```

FRAME33--UVW 对位平台

结构示意图

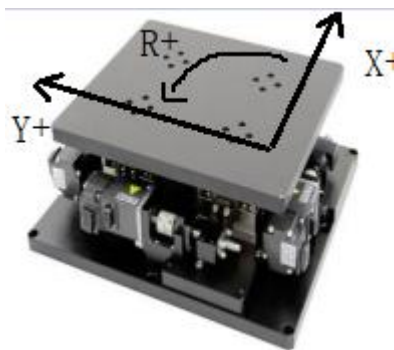


电机方向及角度范围

虚拟的 XY 满足右手法则；

角度正向与虚拟 XY 满足右手法则；

实际电机轴方向无要求，根据实际情况填写结构参数即可。



	角度或移动范围
--	---------

U 轴	机械结构限制
V 轴	
W 轴	

轴顺序

实际机械手关节轴:	定义简写:
U 轴	Axis_a
V 轴	Axis_b
W 轴	Axis_c

直角坐标系虚拟轴:	定义简写:
平移轴 X	Viraxis_x
平移轴 Y	Viraxis_y
旋转轴 RZ	Viraxis_v

指令规定轴号填写次序如下:

逆解时:

```
BASE(Axis_a,Axis_b,Axis_c)
CONNFRAME(33,tablenum, Viraxis_x, Viraxis_y,Viraxis_v)
```

正解时:

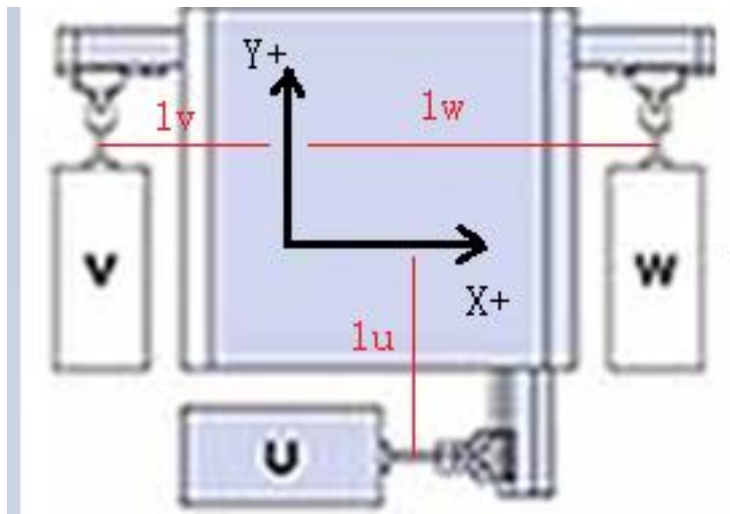
```
BASE(Viraxis_x, Viraxis_y,Viraxis_v)
CONNREFRAME(33,tablenum, Axis_a,Axis_b,Axis_c)
```

结构参数设置

建立机械手连接时，需要将机械结构参数依次填写到 table 数组中，参数说明如下。

TABLE(Tablenum,lu, lv,lw, angleonecircle, dirx, diry1, diry2)	
Tablenum	存储转换参数的 TABLE 位置。
lu	U 轴的距离参数，填写正值，零点到 U 轴的垂直距离。
lv	V 轴的距离参数，填写正值，零点到 V 轴的垂直距离。
lw	W 轴的距离参数，填写正值，零点到 W 轴的垂直距离。
angleonecircle	虚拟旋转轴一圈脉冲数，任意值。
diru	U 轴的方向，0-与虚拟 X 正向一致，1 反向。

dirv	V 轴的方向，0-与虚拟 Y 正向一致，1 反向。
zdirw	W 轴的方向，0-与虚拟 Y 正向一致，1 反向。



电机参数设置

各轴的轴类型和脉冲当量要设置正确。

一个机械手的所有虚拟轴和关节轴的长度单位要求统一，一般都是 mm 单位。虚拟轴的一个 mm 的脉冲数一般建议设置为 1000，表示精度为小数点后 3 位。

	Atype	Units
Axis_a	1 或 4 或 7 或 65 或 50	每 mm 脉冲数
Axis_b		每 mm 脉冲数
Axis_c		每 mm 脉冲数
Viraxis_x	0	1000
Viraxis_y		1000
Viraxis_v		Angleonecircle/360

零点位置

保证 VW 轴是水平的。

平台上任意一点可作为零点，保证结构参数正确即可。

建立连接

正解模式

```
BASE(Viraxis_x, Viraxis_y, Viraxis_v)  
CONNREFRAME(33, tablenum, Axis_a, Axis_b, Axis_c)  
WAIT LOADED
```

建立成功后，虚拟轴 MTYPE 将显示为 34，IDLE 显示为 0。

此时只能操作关节轴在关节坐标系中运动，虚拟轴会自动计算末端工作点位于直角坐标系中的位置。

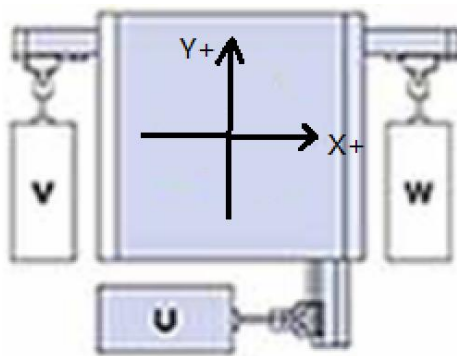
逆解模式

```
BASE(Axis_a, Axis_b, Axis_c)  
CONNFRAME(33, tablenum, Viraxis_x, Viraxis_y, Viraxis_v)  
WAIT LOADED
```

建立成功后，关节轴 MTYPE 将显示为 33，IDLE 显示为 0。

此时只能操作虚拟轴在直角坐标系中运动，关节轴会自动计算在关节坐标系中如何联合运动。

注意事项



常见问题 1：uvw 轴对应虚拟轴 XY 关系错误。

u 轴运行方向应当与虚拟轴 X 平行。

vw 轴分别位于左右，次序相反时旋转方向错误。

常见问题 2：虚拟轴脉冲当量定义错误。

虚拟轴 XY 为长度轴，UNITS 意义要与实际轴相同。

虚拟旋转轴的 UNITS 应当与平台旋转一圈的脉冲数一致。

常见问题 3：虚拟旋转轴正向为顺时针。

缺省虚拟 Y 轴正方向应当指向 u 电机的反方向。

可通过调换 vw 轴次序，同时将 lu 参数修改为负值的方式来改变旋转正向。

使用例程

```
""电机、机械手参数定义
dim lu          '零点到 U 轴的垂直距离
dim lv          '零点到 V 轴的垂直距离
dim lw          '零点到 W 轴的垂直距离
dim angleonecircle '虚拟旋转轴一圈脉冲数
lu=20
lv=10
lw=10
angleonecircle=360000

dim u_mu      '电机 u 一圈脉冲数
dim u_mv      '电机 v 一圈脉冲数
dim u_mw      '电机 w 一圈脉冲数
u_mu=3600
u_mv=3600
u_mw=3600

dim i_u      'u 传动比
dim i_v      'v 传动比
dim i_w      'w 传动比
i_u=2
i_v=2
i_w=2

dim u_ju      '关节 1 实际一圈脉冲数
dim u_jv      '关节 2 实际一圈脉冲数
dim u_jw      '关节 z 实际一圈脉冲数

u_ju=u_mu*i_u
u_jv=u_mv*i_v
u_jw=u_mw*i_w

dim p_u,p_v,p_w '螺距
p_u=1
p_v=1
p_w=1

dim diru,dirv,dirw '关节轴方向
```

```

diru=0      'U 轴的方向, ?0-与虚拟 X 正向一致, 1 反向
dirv=0      'V 轴的方向, ?0-与虚拟 Y 正向一致, 1 反向
dirw=0      'W 轴的方向, ?0-与虚拟 Y 正向一致, 1 反向

""关节轴设置
BASE(0,1,2,3)      '选择关节轴号
atype=1,1,1,1      '轴类型设为脉冲轴
UNITS=u_ju/p_u,u_jv/p_v,u_jw/p_w  '把 z 轴 units 设成 1mm 的脉冲数
                                '其余轴设成 1°的脉冲数
DPOS=0,0,0        '设置关节轴的位置, 此处要根据实际情况来修改。
speed=100,100,100  '速度参数设置
accel=1000,1000,1000
decel=1000,1000,1000
CLUTCH_RATE=10000,10000,10000  '使用关节轴的速度和加速度限制

""虚拟轴设置
BASE(6,7,8)
ATYPE=0,0,0        '设置为虚拟轴
TABLE(0,lu,lv,lw,angleonecircle,diru,dirv,dirw)  '根据手册说明填写参数
UNITS=1000,1000 ,angleonecircle/360  '运动精度, 要提前设置, 中
途不能变化
speed=200,200,200  '速度参数设置
accel=1000,1000,1000
decel=1000,1000,1000
""建立机械手连接
while 1
  if scan_event(in(0))>0 then  '输入 0 上升沿触发
    BASE(0,1,2)                '选择关节轴号
    CONNFRAME(33,0,6,7,8)      '第 6/7 轴作为虚拟的 XY 轴, 启动逆
解连接。
    WAIT LOADED                '等待运动加载, 此时会自动调整虚拟轴的
位置。
    ?"逆解模式"
  elseif scan_event(in(0))<0 then  '输入 0 下降沿触发
    BASE(6,7,8)                '选择虚拟轴号
    CONNREFRAME(33,0,0,1,2)     '第 0/1 轴作为关节轴, 启动正解
连接。
    WAIT LOADED                '等待运动加载。
    ?"正解模式"
  endif
endwhile
wend

```


常见问题

错误码说明

错误情况	可能原因
连接机械手时，控制器报错 3205	控制器型号不支持此类型机械手。
控制器报错 1025	不是 FRAME 状态，没有连接为机械手。
FRAME_ROTATE 偏移、旋转坐标，控制器报错 1005	当前类型机械手不支持 FRAME_ROTATE 功能。

FAQ

Q:机械手运动轨迹，坐标不准确

问题描述：

机械手逆解运动时，实际轨迹并不是按照插补运动指令轨迹运行，或最终运动位置不准确。

分析与解决：

1. 关节轴的 units 设置不准确。

使用 zdevelop 软件内的“视图” - “手动运动”。



在【距离】栏中填写 45，然后运动，若机械结构转动了 45°（平移轴则按距离算）则说明 units 设置正确，否则请重新确认 units 的实际值。

2. 建立连接时未运动到指定的零点位置。

手动操作各关节轴回到 dpos 为 0 时的位置，然后比对此时关节轴的位置是否符合零点位置的描述。

3. 电机的实际方向与定义方向相反。

操作机械手回到正确的零点位置后，手动操作各轴，查看电机转向是否与文档描述相符。

4. 当前姿态无法实现规划的运动轨迹。
手动操作关节轴切换姿态再重新逆解运动。
5. 当前 FRAME 类型旋转与实际机械手结构不符合。
查看文档示意图比对实际机械手确认。

Q:逆解状态下进行逆解链接，程序阻塞

问题描述:

客户在调用了逆解链接的情况，程序总会阻塞在逆解链接地方。

分析与解决:

判断程序阻塞的代码，程序在 wait loaded 阻塞。判断关节轴 ntype 的值，发现关节轴的 ntype=33，而且 mtype=33。

程序阻塞，因为在逆解状态进行逆解链接。

建议客户，在进行逆解链接需判断当前是否在逆解状态。

结论:

进行逆解链接与正解链接，需判断当前是否在正解或逆解状态。

Q:旋转台类机械手，末端点滞后旋转台，速度越高滞后越严重

问题描述:

双机械手模式，frame17+frame100，frame17 的虚拟轴用 connect 链接 frame100 的关节轴，控制 frame100 的虚拟轴+frame17 的旋转轴运动，在此先说明实现情况。客户控制末端点与旋转台做插补运动，预期目标末端点随旋转台一起旋转，并在旋转台上加工出预期轨迹。实际情况低速情况能达到预期目标，提高虚拟轴速度末端工作点会滞后旋转台，速度越高滞后距离越长，达不到预期目标。

解决过程:

现场看客户使用程序，frame17 关节轴的 clutch_rate=0，但是关节轴的 accel 与的

decel 设置的远小于设置的速度。

对 frame17 关节轴的 accel 与 decel 增大，问题解决。

问题原因：

因 clutch_rate=0 模式，关节轴使用自己的速度加速度做速度规划。但关节轴的加减速过小，关节轴在一直加速追赶的过程，追赶的过程虚拟轴与旋转台的速度大于末端点的速度，所以末端点滞后旋转台，随虚拟轴的速度越高滞后的距离越长。

结论：

旋转台类机械手，正确设置关节轴的加减速度。

问题的测试程序参照“例 6.1”

例 6.1

```
'=====机械参数、电机参数定义=====
dim axisx,axisy,axisz,axisr1,axisr2    "定义实际轴
dim viraxisx,viraxisy,viraxisz        "定义虚拟轴
axisx=0    ""实际轴编号
axisy=1
axisz=2
axisr1=3    ""旋转轴编号
axisr2=4

viraxisx=6    ""虚拟轴编号
viraxisy=7
viraxisz=8

dim centy1,centz1,centx2,centy2        '旋转中心定义
dim ifanticlock1,ifanticlock2          '是否逆时针
centy1=0    '旋转中心偏移
centz1=0
centx2=10
centy2=0
ifanticlock1=1    '旋转台 1 逆时针
ifanticlock2=1    '旋转台 2 逆时针

dim u_mx,u_my,u_mz    '轴旋转一圈的脉冲
dim u_mr1,u_mr2
u_mx=3600    '旋转轴脉冲参数赋值
u_my=3600
u_mz=3600
u_mr1=360
u_mr2=360

dim i_x,i_y,i_z    '电机传动比
```

```

dim i_r1,i_r2
i_x=2
i_y=2
i_z=2
i_r1=1
i_r2=1

dim u_jx    'x 关节实际一圈脉冲数
dim u_jy    'y 关节实际一圈脉冲数
dim u_jz    'z 关节实际一圈脉冲数
dim u_jr1   '旋转 1 实际一圈脉冲数
dim u_jr2   '旋转 2 实际一圈脉冲数
u_jx=u_mx*i_x
u_jy=u_my*i_y
u_jz=u_mz*i_z
u_jr1=u_mr1*i_r1
u_jr2=u_mr2*i_r2

dim p_x     'x 螺距
dim p_y     'y 螺距
dim p_z     'z 螺距
p_x=2
p_y=2
p_z=10

""""""定义关节轴
BASE(axisx,axisy,axisz)    '实际轴定义
ATYPE=1,1,1                '轴运动类型
rapidstop
wait idle                  '确保所有轴停止运动
UNITS=u_jx/p_x,u_jy/p_y,u_jz/p_z    '设置为 1mm 的脉冲数
Dpos=0,0,0
speed=500,500,500
CREEP=20,20,20
ACCEL=100,100,100
DECEL=100,100,100
clutch_rate=0,0,0

base(axisr1,axisr2)       '旋转轴设置
ATYPE=1,1                 '轴运动类型
units =u_jr1/360,u_jr1/360 '设置为 1°的脉冲数
Dpos=0,0
speed=200,200             '末端旋转轴参数设置
accel=100,100

```

```

decel=100,100
clutch_rate=0,0

""""""虚拟轴定义
base(viraxisx,viraxisy,viraxisz)    '虚拟轴定义
ATYPE=0,0,0
TABLE(0, CentY1, CentZ1,u_jr1, IfAntiClock1, CentX2, CentY2,u_jr2,
IfAntiClock2)    ""table 表参数填写
UNITS=1000,1000,1000    ""虚拟轴运动参数赋值
SPEED=100,100,100
ACCEL=500,500,500
clutch_rate=0,0

""""""运动控制开始
merge = on                    '连续缓冲开启
CORNER_MODE =2+32            '设置启动拐角减速+倒角
ZSMOOTH=10                   '倒角半径设置
DECEL_ANGLE = 15 * (PI/180)  '开始减速的最小拐角
STOP_ANGLE = 45 * (PI/180)  '减速到最低的减速拐角

if modbus_reg(0)=1 then
    base(0,1,2)
    accel=10,10,10
    decel=10,10,10
elseif modbus_reg(0)=2 then
    base(0,1,2)
    accel=100,100,100
    decel=100,100,100
end if

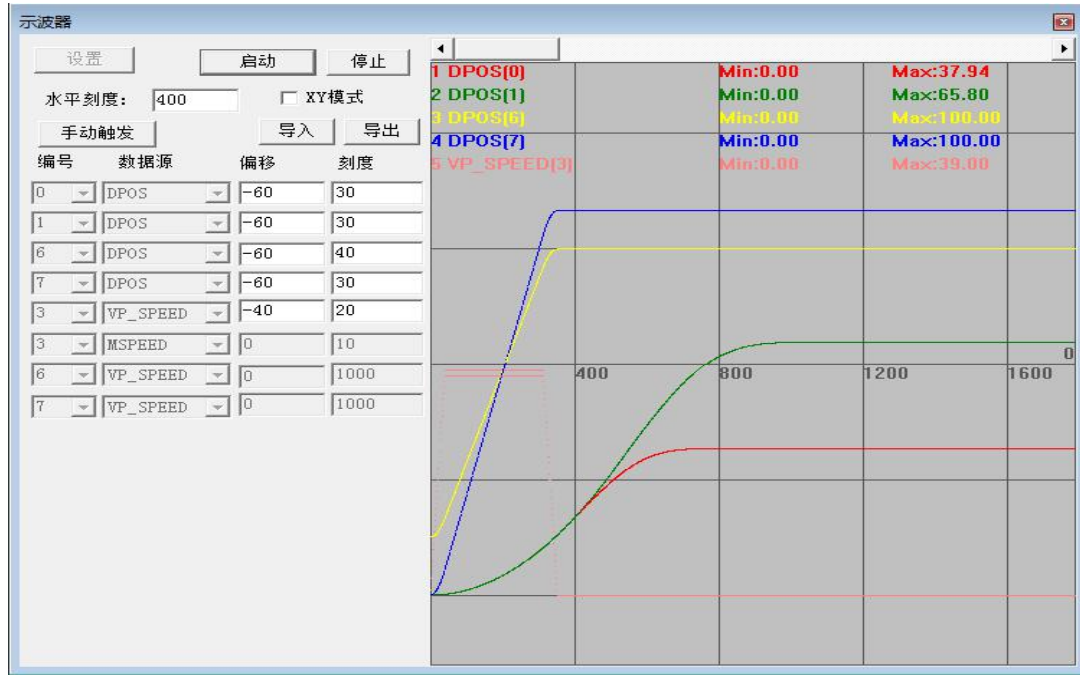
BASE(axisx,axisy,axisz,axisr1,axisr2)    '选择关节轴号
CONNFRAME(17,0, viraxisx,viraxisy,viraxisz,axisr1,axisr2)    '第 6/7 轴作为虚拟
的 XY 轴，启动逆解连接。
WAIT LOADED    '等待运动加载，此时会自动调整虚拟轴的位置。
?"逆解模式"

TRIGGER
base(6,7,8,3,4)
moveabs(100,100,0,60,30)
end

```

错误情况：

关节轴滞后与虚拟轴。



正确情况：
关节轴与虚拟轴同步。

