机器人运动指令

机器人在空间中运动主要有关节运动（MoveJ）、线性运动（MoveL）、圆弧运动（MoveC）和绝对位置运动（MoveAbsJ）四种方式。

1、绝对位置运动

用途：

MoveAbsJ（绝对关节移动）用来把机器人或者外部轴移动到一个绝对位置，该位置在轴定位中定义。

MoveAbsJ指令中机器人的最终位置，既不受工具或者工作对象的影响，也不受激活程序更换的影响。机器人要用到这些数据来计算负载、TCP速度和转角点。相同的工具可以被用在相邻的运动指令中。

机器人和外部轴沿着一个非直线的路径移动到目标位置。所有轴在同一时间运动到目标位置。

该指令只能被用在主任务T_ROB1中，或者在多运动系统中的运动任务中。

例1：MoveAbsJ p50，v1000，z50，tool2；

机器人将携带工具tool2沿着一个非线性路径到绝对轴位置p50，以速度数据v1000和zone数据z50。

例2：MoveAbsJ *，v1000\T:=5，fine，grip3；

机器人将携带工具grip3沿着一个非线性路径到一个停止点，该停止点在指令中作为一个绝对轴位置存储（用*标示）。整个运动需要5秒钟。

MoveAbsJ[\Conc] ToJointPos[\ID][\NoEOffs]Speed[\V]|[\T] Zone [\Inpos] Tool [\Wonj]

[\Conc]:

并发事件
数据类型： switch
当机器人正在移动的时候执行的后续指令。该项目通常不使用，但是当和外部设备通讯、不需要同步的时候可以用来缩短循环周期。
当使用项目\Conc的时候，连续运动指令的数量限制为5.在包含StorePath-RestoPath的程序段中不允许包含项目\Conc的运动指令。
如果该项目忽略并且ToJointPos不是一个停止点，在机器人到达程序zone之前一段时间后续指令开始执行了。
该项目不能用在多运动系统的坐标同步运动中。

MoveAbsJ[\Conc] ToJointPos[\ID][\NoEOffs]Speed[\V]|[\T] Zone [\Inpos] Tool [\Wonj]

ToJointPos：

到达的关节位置。
数据类型：jointtarget
机器人和外部轴的绝对目标轴位置。它被定义为一个命名的位置或者直接存储在指令中（在指令中用*标示）

[ID]：

同步ID
数据类型：identno
该项目必须使用在多运动系统中，如果并列了同步运动，则不允许在其他任何情况下使用。
指令的ID号在所有协同的程序任务中必须相同。该ID号保证在rountine中运动不会混乱。

MoveAbsJ[\Conc] ToJointPos[\ID][\NoEOffs]Speed[\V]|[\T] Zone [\Inpos] Tool [\Wonj]

[\NoEOffs]：

没有外部偏移量
数据类型：switch
如果项目\NoEOffs设为1，MoveAbsJ运动将不受外部轴的激活偏移量的影响。

Speed：

数据类型：speeddata
运动所用的速度数据。速度数据定义了TCP、工具再定位和外部轴的速度。

MoveAbsJ[\Conc] ToJointPos[\ID][\NoEOffs]Speed[\V]|[\T] Zone [\Inpos] Tool [\Wonj]

[\V]:

速度
数据类型：num
该项目用来在指令中直接指定TCP的速度，单位mm/s，它替代在速度数据中指定的相应的速度。

[\T]:

时间
数据类型：num
该项目用来指定机器人运动的总时间，单位秒。它替代相应的速度数据。

MoveAbsJ[\Conc] ToJointPos[\ID][\NoEOffs]Speed[\V]|[\T] Zone [\Inpos] Tool [\Wonj]

Zone：

数据类型：zonedata
运动的zone数据。Zone数据描述了产生的转角路径的大小。

[\z]:

Zone
数据类型：num
该项目用来在指令中直接指定机器人TCP的位置精度。转角路径的长度用毫米给出，替代zone数据中指定的相应数据。

MoveAbsJ[\Conc] ToJointPos[\ID][\NoEOffs]Speed[\V]|[\T] Zone [\Inpos] Tool [\Wonj]

[\Inpos]：

到位
数据类型：stoppointdata（停止点数据）
该项目用来指定机器人TCP在停止点位置的收敛性判别标准。该停止点数据代替在zone参数中指定的zone

Tool：

数据类型：tooldata
运动过程中所携带的工具。
TCP的位置和工具的负载在工具数据中定义。TCP位置用来计算运动的速度和转角路径

MoveAbsJ[\Conc] ToJointPos[\ID][\NoEOffs]Speed[\V]|[\T] Zone [\Inpos] Tool [\Wonj]

[\Wobj]：

工作对象
数据类型：wobjdata
在运动过程中使用的工作对象。
如果机器人抓着工具的时候，该项目可以忽略。但是，如果机器人抓着工作对象，也就是说工具是静止的，或者带有外部轴，那么该项目必须指定。
在有并列工具或者有并列外部轴的情况下，系统使用该数据计算运动的速度和转角路径，该数据在工作对象中定义。

程序执行

MoveAbsJ运动不会受激活的程序转移的影响，并且如果使用了可选项目\NoEOffs，将没有外部轴的偏移。如果不使用\NoEOffs，外部轴的目标位置将会受到激活的外部轴偏移的影响。工具安装轴角度插补移动到绝对轴目标位置。这就是说每一个轴都安装固定的速度运动，并且所有轴都在同一时间到达目标位置，这样就形成一个非线性的路径。

总的来说，TCP大约按照编程的速度运动。在TCP运动的同时，工具重新定向，并且外部轴也在运动。如果重新定向的或者外部轴的程序要求的速度不能达到，TCP的速度将被减小。

当转换到路径的下一段的时候通常会产生转角路径。如果停止点在Zone数据中指定，只有在机器人和外部轴到达合适的轴位置的时候程序才能继续执行。

范例

例1： MoveAbsJ *， v2000\V:=2200， z40\Z:=45， grip3；

Grip3沿着一个非线性路径运动到一个存储在指令中的一个绝对轴位置。执行的运动数据为v2000和z40.

TCP的速度大小是2200m/s，zone的大小是45mm。

例2： MoveAbsJ p5，v2000，fine\Inpos:=inpos50，grip3；

工具grip3沿非线性路径运动至绝对接头位置p5。当满足关于停止点fine的50%的位置条件和50%的速度条件时，机械臂认为该工具位于点内

语法

MoveAbsJ [‘\’ Conc ‘,’] [ToJointPos’ :=’] <关节目标表达式（IN）>

[‘\’ ID ‘:=’ <identno 类型的表达式(IN)>] [‘\’ NoEOffs] ‘,’

[Speed ‘:=’] <speeddata 类型的表达式(IN)>

[‘\’ V ‘:=’ <num 类型的表达式（IN）>]

| [‘\’ T’ :=’ <num 类型的表达式(IN)>] ‘,’

[‘\’ Z ‘:=”] <num 类型的表达式(IN)>

[‘\’ Inpos’ :=’ <stoppointdata 类型的表达式(IN)>] ‘,’

[Tool ‘:=’]<tooldata 类型的恒量(PERS)>

[‘\’Wobj’ :=’ wobjdata 类型的恒量(PRS)>] ‘;’

2、关节运动指令

MoveJ [\Conc], ToPoint,Speed[\V]| [\T],Zone [\Z] [\Inpos],Tool[\Wobj];

[\Conc] :	协作运动开关。	（switch）
ToPoint :	目标点，默认为*	（robtarget)）
Speed：	运行速度数据。	（speeddata）
[\V]：	特殊运行速度mm/s	（num）
[\T]：	运行时间控制s	（num）
Zone :	运行转角数据。	（zonedata）
[\Z] :	特殊运行转角mm。	（num）
[\Inpos] :	运行停止点数据。	（stoppointdata）
Tool :	工具中心点（TCP）	（tooldata）
[\Wobj] :	工件坐标系	（wobjdata）

应用：

机器人以最快捷的方式运动至目标点，机器人运动状态不完全可控，但运动路径保持唯一，常用于机器人在空间大范围移动。

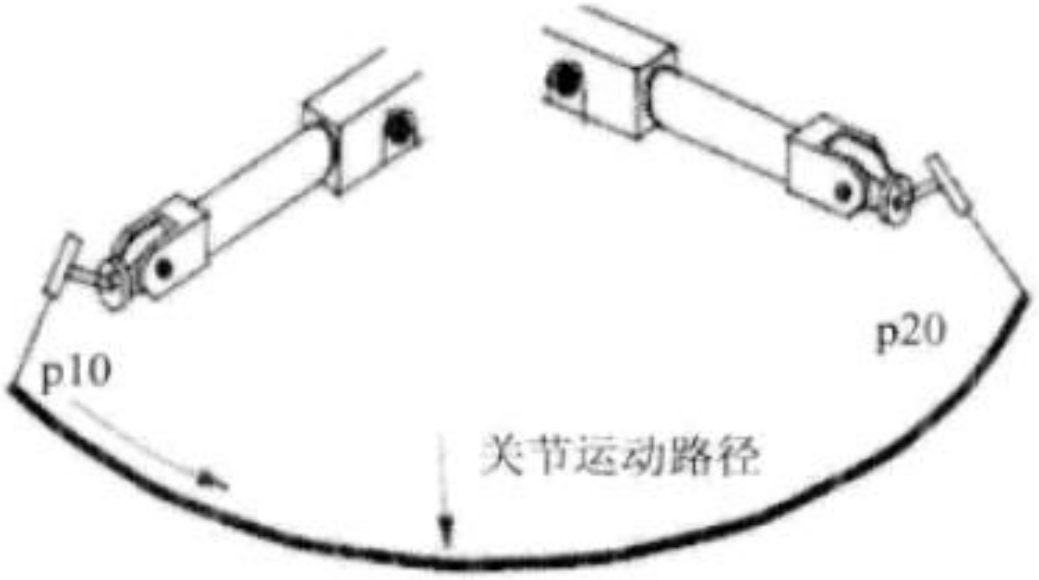
实例：

```
MoveJ p1,v2000,fine,grip1;
MoveJ\Conc,p1,v2000,fine,grip1;
MoveJ p1,v2000\V:=2200,z40\z:45,grip1;
MoveJ p1,v2000,z40,grip1\Wobj:=wobjTable;
MoveJ p1,v2000,fine\Inpos:=inpos50,grip1;
```

关节运动指令

```
MoveJ p10，v1000，z50，tool1\wobj:=wobj1；
MoveJ p20，v1000，z50，tool1\wobj:=wobj1；
```

关节运动指令是在对路径精度要求不高的情况下，机器人的工具中心点TCP从一个位置移动另一个位置，两个位置之间的路径不一定是直线。



3、线性运动指令

MoveL [\Conc,] ToPoint,Speed[\V]| [\T],Zone [\Z] [\Inpos],Tool[\Wobj] [\Corr];

[\Conc] :	协作运动开关。	（switch）
ToPoint :	目标点，默认为*	（robtarget)）
Speed：	运行速度数据。	（speeddata）
[\V]：	特殊运行速度mm/s	（num）
[\T]：	运行时间控制s	（num）
Zone :	运行转角数据。	（zonedata）
[\Z] :	特殊运行转角mm。	（num）
[\Inpos] :	运行停止点数据。	（stoppointdata）
Tool :	工具中心点（TCP）	（tooldata）
[\Wobj] :	工件坐标系	（wobjdata）
[\Corr] :	修正目标点开关	（switch）

应用：

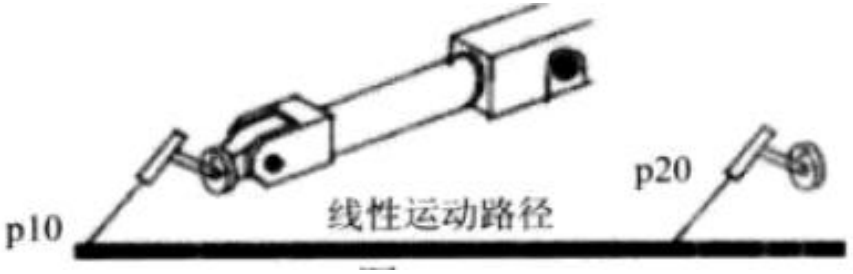
机器人以线性方式运动至目标点，当前点与目标点两点决定一条直线，机器人运动状态可控，运动路径保持唯一，可能出现死点，常用于机器人在工作状态移动。

实例：

```
MoveL p1,v2000,fine,grip1;
MoveL\Conc,p1,v2000,fine,grip1;
MoveL p1,v2000\V:=2200,z40\z:45,grip1;
MoveL p1,v2000,z40,grip1\Wobj:=wobjTable;
MoveL p1,v2000,fine\Inpos:=inpos50,grip1;
MoveL p1,v2000,fine,grip1\corr;
```

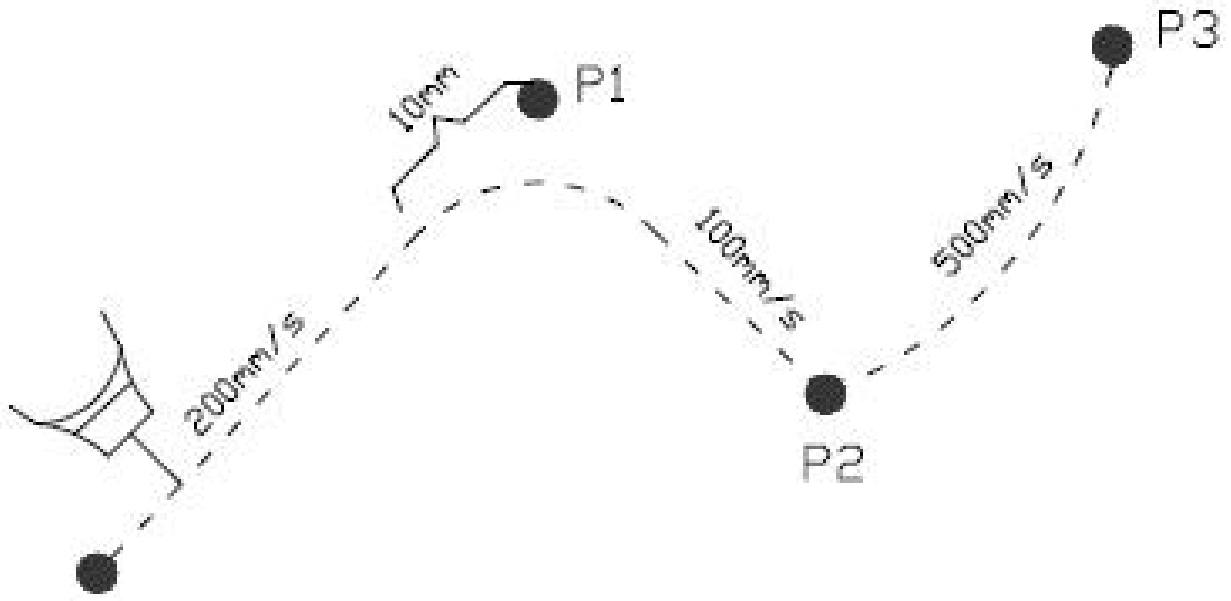
线性运动指令

```
MoveL p10，v1000，z50，tool1\wobj:=wobj1；
MoveL p20，v1000，z50，tool1\wobj:=wobj1；
```



线性运动是机器人的TCP从起点到终点之间的路径始终保持为直线。一般如焊接、涂胶等应用对路径要求高的场合使用此指令。

关节运动指令与线性运动指令



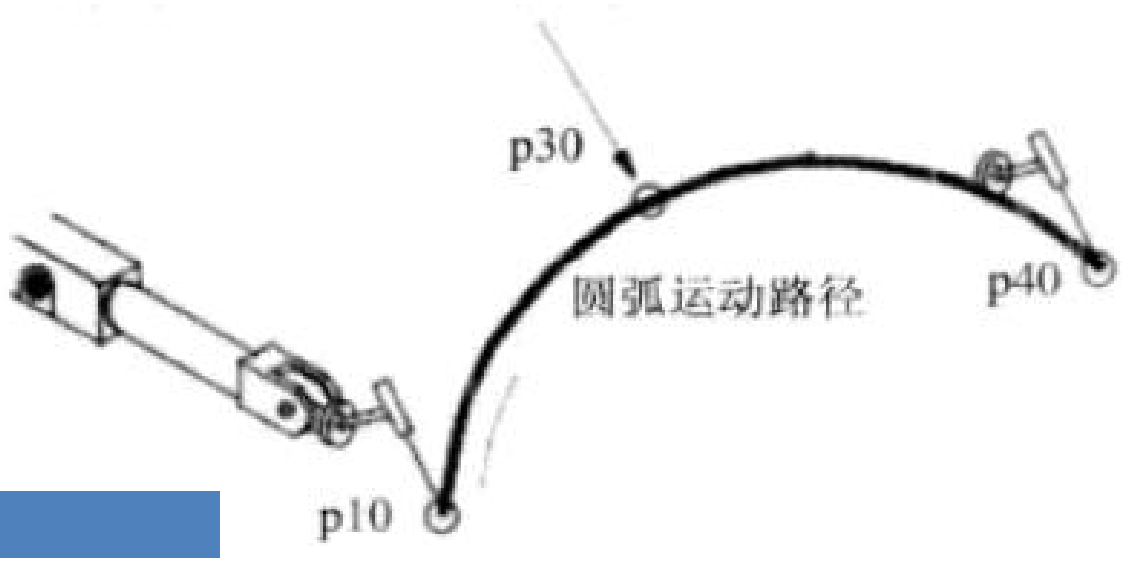

```
MoveL p1,v200,z10,tool1
MoveL p2,v100,fine,tool1
MoveJ p3,v500,fine,tool1
```

4、圆弧运动指令

MoveC [\Conc,] CirPoint,ToPoint,Speed[\V]| [\T],Zone [\Z] [\Inpos],Tool [\Wobj][\Corr];

[\Conc] :	协作运动开关。	（switch）
CirPoint :	中间点，默认为*	（robtarget）
ToPoint :	目标点，默认为*	（robtarget)）
Speed：	运行速度数据。	（speeddata）
[\V]：	特殊运行速度mm/s	（num）
[\T]：	运行时间控制s	（num）
Zone :	运行转角数据。	（zonedata）
[\Z] :	特殊运行转角mm。	（num）
[\Inpos] :	运行停止点数据。	（stoppointdata）
Tool :	工具中心点（TCP）	（tooldata）
[\Wobj] :	工件坐标系	（wobjdata）
[\Corr] :	修正目标点开关	（switch）

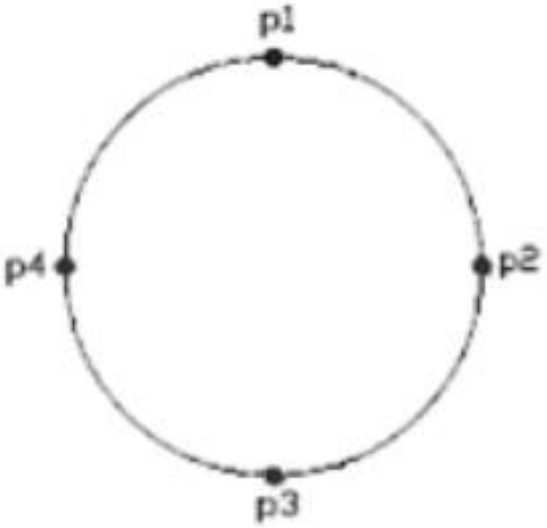
```
MoveL p10，v1000，fine，tool1\wobj:=wobj1;
MoveC p30，p40，v1000，z1，tool1\wobj:=wobj1;
```



参数	含义
p10	圆弧的第一个点
p30	圆弧的第二个点
p40	圆弧的第三个点
fine\z1	转弯区数据

应用：

机器人通过中心点以圆弧方式移动至目标点，当前点、中心点与目标点三点决定一段圆弧，机器人活动状态可控，运动路径保持唯一，常用于机器人在工作状态移动。




```
MoveL p1,v500,fine,tool1
Movec p2,p3,v500,z20,tool1
Movec p4,p1,v500,fine,tool1
```

限制：不可能通过一个MoveC指令完成一个圆。