

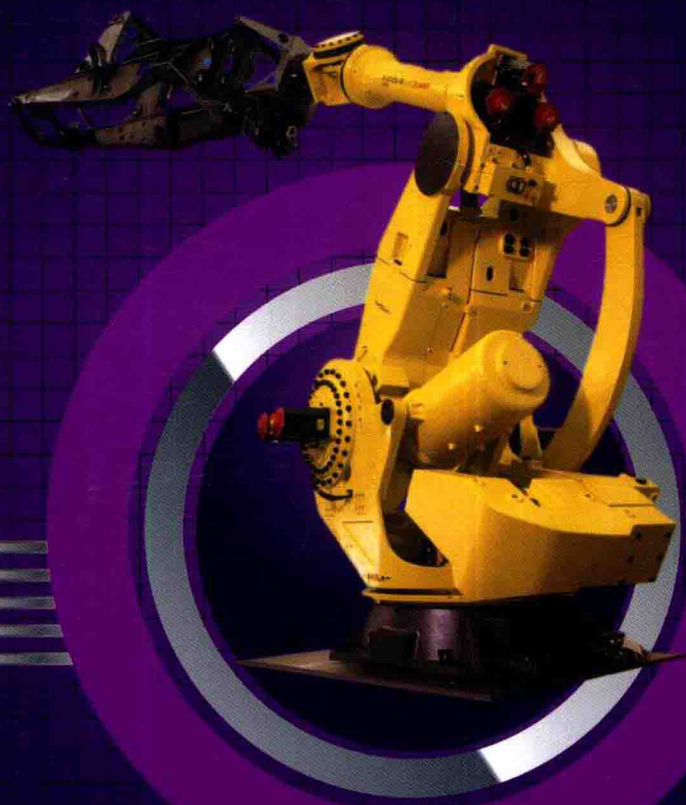


INDUSTRIAL ROBOT

工业机器人

离线编程与仿真

韩鸿鸾 张云强 主编



化学工业出版社



工业机器人 离线编程与仿真

韩鸿鸾 张云强 主编



化学工业出版社

· 北京 ·

图书在版编目 (CIP) 数据

工业机器人离线编程与仿真/韩鸿鸾, 张云强主编.
北京: 化学工业出版社, 2018. 1
ISBN 978-7-122-30917-4

I. ①工… II. ①韩… ②张… III. ①工业机器人-
程序设计②工业机器人-计算机仿真 IV. ①TP242.2

中国版本图书馆 CIP 数据核字 (2017) 第 267712 号

责任编辑: 王 焯
责任校对: 宋 夏

文字编辑: 陈 喆
装帧设计: 刘丽华

出版发行: 化学工业出版社 (北京市东城区青年湖南街 13 号 邮政编码 100011)
印 装: 高教社 (天津) 印务有限公司
787mm×1092mm 1/16 印张 13 字数 350 千字 2018 年 1 月北京第 1 版第 1 次印刷

购书咨询: 010-64518888 (传真: 010-64519686) 售后服务: 010-64518899
网 址: <http://www.cip.com.cn>
凡购买本书, 如有缺损质量问题, 本社销售中心负责调换。

定 价: 59.00 元

版权所有 违者必究

近年来,我国机器人行业在国家政策的支持下,顺势而为,发展迅速,保持着35%的高增长率,远高于德国的9%、韩国的8%和日本的6%。我国已连续两年成为世界第一大工业机器人市场。

我国工业机器人市场之所以能有如此迅速的增长,主要源于以下三点。

(1) 劳动力的供需矛盾。主要体现在劳动力成本的上升和劳动力供给的下降。在很多产业,尤其在中低端工业产业,劳动力的供需矛盾非常突出,这对实施“机器换人”计划提出了迫切需求。

(2) 企业转型升级的迫切需求。随着全球制造业转移的持续深入,先进制造业回流,我国的低端制造业面临产业转移的风险,迫切需要转变传统的制造模式,降低企业运行成本,提升企业发展效率,提升工厂的自动化、智能化程度。而工业机器人的大量应用,是提升企业产能和产品质量的重要手段。

(3) 国家战略需求。工业机器人作为高端制造装备的重要组成部分,技术附加值高,应用范围广,是我国先进制造业的重要支撑技术和信息化社会的重要生产装备,对工业生产、社会发展以及增强军事国防实力都具有十分重要的意义。

随着机器人技术及智能化水平的提高,工业机器人已在众多领域得到了广泛的应用。其中,汽车、电子产品、冶金、化工、塑料、橡胶是我国使用机器人最多的几个行业。未来几年,随着行业需要和劳动力成本的不断提高,我国机器人市场增长潜力巨大。尽管我国将成为当今世界最大的机器人市场,但每万名制造业工人拥有的机器人数量却远低于发达国家水平和国际平均水平。工信部组织制订了我国机器人技术路线图及机器人产业“十三五”规划,到2020年,工业机器人密度达到每万名员工使用100台以上。我国工业机器人市场将高倍速增长,未来十年,工业机器人是看不到“天花板”的行业。

虽然多种因素推动着我国工业机器人行业不断发展,但应用人才严重缺失的问题清晰地摆在我们面前,这是我国推行工业机器人技术的最大瓶颈。中国机械工业联合会的统计数据表明,我国当前机器人应用人才缺口20万,并且以每年20%~30%的速度持续递增。

工业机器人作为一种高科技集成装备,对专业人才有着多层次的需求,主要分为研发工程师、系统设计与应用工程师、调试工程师和操作及维护人员四个层次。其中,需求量最大的是基础的操作及维护人员以及掌握基本工业机器人应用技术的调试工程师和更高层次的应用工程师,工业机器人专业人才的培养,要更加着力于应用型人才的培养。

为了适应机器人行业发展的形势,满足从业人员学习机器人技术相关知识的需求,我们从生产实际出发,组织业内专家编写了本书,全面讲解了工业机器人编程的基础、构建基本仿真工业机器人工作站、仿真软件RobotStudio中的建模功能、机器人离线轨迹编辑、RobotArt离线编程软件的基本操作与工作站系统的构建、离线编程的应用等内容,以期给从业人员和高等院校相关专业师生提供实用性指导与帮助。

本书由韩鸿鸾、张云强主编,陶建海、王小方、李永杉副主编,阮洪涛、刘曙光、马灵芝、范维进、刘兵、程宝鑫、彭红学、张艳红、马岩、姜海军、张瑞社、张青、宁爽、董海

萍、戚建爱、胡春蕾参加了本书的编写。在本书编写过程中得到了山东省、河南省、河北省、江苏省、上海市等技能鉴定部门的大力支持，此外，青岛利博尔电子有限公司、青岛时代焊接设备有限公司、山东鲁南机床有限公司、山东山推工程机械有限公司、西安乐博士机器人有限公司、诺博泰智能科技有限公司等企业为本书的编写提供了大量帮助，在此深表谢意。

在本书编写过程中，参考了《工业机器人装调维修工》《工业机器人操作调整工》职业技能标准的要求，以备读者考取技能等级；同时还借鉴了全国及多省工业机器人大赛的相关要求，为读者参加相应的大赛提供参考。

由于水平所限，书中不足之处在所难免，恳请广大读者给予批评指正。

编者

第 1 章 工业机器人编程的基础 / 1

1.1 机器人编程	1
1.1.1 机器人编程系统及方式	1
1.1.2 对机器人的编程要求	3
1.1.3 机器人编程语言的类型	8
1.1.4 动作级语言	12
1.1.5 对象级语言	12
1.2 工业机器人的离线编程技术	13
1.2.1 离线编程及其特点	13
1.2.2 离线编程系统的软件架构	15
1.2.3 离线编程的基本步骤	17

第 2 章 构建基本仿真工业机器人工作站 / 25

2.1 布局工业机器人基本工作站	25
2.1.1 工业机器人工作站的建立	25
2.1.2 加载物件	37
2.1.3 保存机器人基本工作站	40
2.2 建立工业机器人系统与手动操作	41
2.2.1 建立工业机器人系统操作	41
2.2.2 机器人的位置移动	45
2.2.3 工业机器人的手动操作	46
2.2.4 回机械原点	51
2.3 创建工业机器人工件坐标系与轨迹程序	51
2.3.1 建立工业机器人工件坐标	51
2.3.2 创建工业机器人运动轨迹程序	54
2.4 机器人仿真运行	59
2.4.1 仿真运行机器人轨迹	59
2.4.2 机器人的仿真制成视频	62

第 3 章 仿真软件 RobotStudio 中的建模功能 / 65

3.1 建模功能的使用	65
3.1.1 RobotStudio 建模	65
3.1.2 对 3D 模型进行相关设置	65

3.2 测量工具的使用	69
3.2.1 测量矩形体的边长	69
3.2.2 测量锥体的角度	71
3.2.3 测量圆柱体的直径	73
3.2.4 测量两个物体间的最短距离	74
3.3 创建机器人用工具	76
3.3.1 设定工具的本地末端点	76
3.3.2 创建工具坐标系框架	84
3.3.3 创建工具	87

第4章 机器人离线轨迹编辑 / 91

4.1 创建机器人离线轨迹曲线及路径	91
4.1.1 RobotStudio 离线编程软件的自动路径功能实现步骤	91
4.1.2 RobotArt 离线编程软件的自动路径功能实现步骤	96
4.2 机器人目标点调整及轴配置参数	103
4.2.1 RobotStudio 离线编程软件的轨迹调整	103
4.2.2 RobotArt 离线编程软件的轨迹调整	114

第5章 RobotArt 离线编程软件的基本操作与工作站系统的构建/120

5.1 离线编程软件开发环境介绍	120
5.1.1 RobotArt 离线编程软件界面	120
5.1.2 RobotArt 软件界面各部分详细介绍	120
5.1.3 三维球仿真软件基本操作	126
5.1.4 机器人 TCP 校准方式	133
5.2 工业机器人工作站系统构建	136
5.2.1 准备机器人	136
5.2.2 准备工具	139
5.2.3 准备工件	147
5.3 工业机器人系统工作轨迹生成	150
5.3.1 导入轨迹	150
5.3.2 生成轨迹	150
5.3.3 轨迹选项	157
5.3.4 轨迹操作命令	158
5.3.5 轨迹调整	160
5.3.6 合并前一个轨迹	163
5.3.7 轨迹点操作命令	166

第6章 离线编程的应用 / 172

6.1 激光切割	172
6.1.1 环境搭建	172
6.1.2 轨迹设计	174
6.1.3 仿真	181

6.1.4 后置	181
6.2 去毛刺	182
6.2.1 环境搭建	182
6.2.2 轨迹设计	185
6.2.3 仿真	193
6.2.4 后置	193

附录 工业机器人词汇 / 194

参考文献 / 199

第 1 章

工业机器人编程的基础

1.1 机器人编程

1.1.1 机器人编程系统及方式

机器人编程 (robot programming) 为使机器人完成某种任务而设置的动作顺序的描述。机器人运动和作业的指令都是由程序进行控制的, 常见的编程方法有两种: 示教编程 (图 1-1 为 ABB 工业机器人的示教器) 和离线编程 (图 1-2 为离线编程开始界面)。

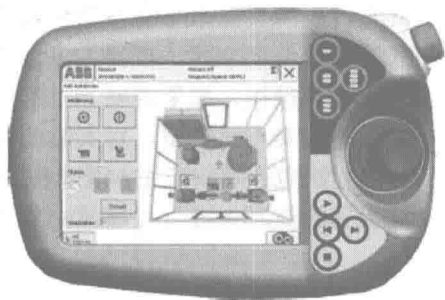


图 1-1 ABB 工业机器人的示教器



图 1-2 离线编程软件

(1) 示教编程

目前大多数机器人还是采用示教方式编程。示教方式是一项成熟的技术，易于被熟悉工作任务的人员所掌握，而且用简单的设备和控制装置即可进行。示教过程进行得很快，示教过后马上即可应用。在对机器人进行示教时，将机器人的轨迹和各种操作存入其控制系统的存储器。如果需要，过程还可以重复多次。在某些系统中，还可以用与示教时不同的速度再现。

① 示教编程法分为三个步骤：

- a. 示教。即机器人学习的过程，在这个过程中，操作者要手把手教会机器人做某些动作。
- b. 存储。机器人的控制系统以程序的形式将示教的动作记忆下来。
- c. 再现。机器人按照示教时记忆下来的程序展现这些动作，就是“再现”过程。

② 示教编程可分为在线示教方式和离线示教方式。

a. 在线示教。即在现场直接对操作对象进行的一种编程方法，常用的有：人工引导示教，由有经验的操作人员移动机器人的末端执行器，计算机记忆各自由度的运动过程；辅助装置示教，对一些人工难以牵动的机器人，例如一些大功率或高减速比机器人，可以用特别的辅助装置帮助示教。也可以用示教盒进行示教，为了方便现场示教，一般工业机器人都配有示教盒，它相当于键盘，有回零、示教方式、数字、输入、编辑、启动、停止等键。

b. 离线示教。由于离线示教不便于现场操作，而且工作量大、精度低，故不建议采用。离线示教方法包括：解析示教，将计算机辅助设计的数据直接用于示教，并利用传感技术进行必要的修正；任务示教，指定任务以及操作对象的位置、形状，由控制系统自动规划运动路径。任务示教是一种发展方向，具有较高的智能水平，目前仍处于研究中。

目前，相当数量的机器人仍采用示教编程方式。机器人示教后可以立即应用，再现时，机器人重复示教时存入存储器的轨迹和各种操作，如果需要，过程可以重复多次。

示教编程法的优点是简单方便，不需要环境模型，对实际的机器人进行示教时，可以修正机械结构带来的误差。当然，示教编程法也存在一定的缺陷，比如功能编辑比较困难，难以使用传感器，难以表现条件分支，对实际的机器人进行示教时，要占用机器人。

(2) 离线编程法

机器人离线编程系统都是利用计算机图形学的成果，建立起机器人及工作环境的几何模型，再利用一些规划算法，通过对图形的控制和操作，在离线的情况下进行轨迹的规划，通过对编程结果进行三维图形的动画仿真，以检验编程的正确性，最后将生成的代码传给机器人控制系统，以控制机器人的运动，完成给定的任务。

机器人的离线编程系统是已被证明的一个有力的工具，可以增加安全性，减少机器人不工作的时间和降低成本。机器人离线编程系统是机器人编程语言的拓展，通过该系统可以建立机器人和 CAD/CAM 之间的联系。

离线编程有以下几个方面的优点：

- ① 编程时可以不使用机器人，以腾出机器人去做其他工作。
- ② 可预先优化操作方案和运行周期。
- ③ 以前完成的过程或子程序可结合到待编的程序中去。
- ④ 可用传感器探测外部信息，从而使机器人作出相应的响应。这种响应使机器人可以工作在自适应的方式下。
- ⑤ 控制功能中可以包含现有的计算机辅助设计 (CAD) 和计算机辅助制造 (CAM) 的信息。
- ⑥ 可以预先运行程序来模拟实际运动，从而不会出现危险。利用图形仿真技术，可以在屏幕上模拟机器人运动来辅助编程。
- ⑦ 对不同的工作目的，只需替换一部分待定的程序。

机器人离线编程技术对机器人的推广应用及其工作效率的提升有着重要意义，离线编程可以大幅度节约制造时间，实现机器人的实时仿真，为机器人的编程和调试提供安全灵活的环境，是机器人开发应用的方向。

1.1.2 对机器人的编程要求

机器人编程系统是机器人编程语言的拓展，通过该系统可以建立机器人和 CAD/CAM 之间的联系。

(1) 设计一个编程系统应具备的知识

- ① 所编程的工作过程的知识；
- ② 机器人和工作环境三维实体模型；
- ③ 机器人几何学、运动学和动力学的知识；
- ④ 基于图形显示的软件系统、可进行机器人运动的图形仿真；
- ⑤ 轨迹规划和检查算法，如检查机器人关节角超限、检测碰撞以及规划机器人在工作空间的运动轨迹等；
- ⑥ 传感器的接口和仿真，以利用传感器的信息进行决策和规划；
- ⑦ 通信功能，以完成离线编程系统所生成的运动代码到各种机器人控制柜的通信；
- ⑧ 用户接口，以提供有效的人机界面，便于人工干预和进行系统的操作。

此外，由于编程系统是基于机器人系统的图形模型来模拟机器人在实际环境中的工作进行编程的，因此为了使编程结果能很好地符合实际情况，系统应能够计算仿真模型和实际模型之间的误差，并尽量减少二者间的误差。

(2) 对机器人语言的编程要求

① 能够建立世界模型 在进行机器人编程时，需要一种描述物体在三维空间内运动的方式。所以需要给机器人及其相关物体建立一个基础坐标系。这个坐标系与大地相连，也称“世界坐标系”。

机器人工作时，为了方便起见，也建立其他坐标系，同时建立这些坐标系与基础坐标系的变换关系。

机器人编程系统应具有在各种坐标系下描述物体位姿的能力和建模能力。

② 能够描述机器人的作业 机器人作业的描述与其环境模型密切相关，编程语言水平决定了描述水平。其中以自然语言输入为最高水平。现有的机器人语言需要给出作业顺序，由语法和词法定义输入语言，并由它描述整个作业。

③ 能够描述机器人的运动 描述机器人需要进行的运动是机器人编程语言的基本功能之一。用户能够运用语言中的运动语句，与路径规划器和发生器连接，允许用户规定路径上的点及目标点，决定是否采用点插补运动或笛卡儿直线运动。用户还可以控制运动速度或运动持续时间。

对于简单的运动语句，大多数编程语言具有相似的语法。不同语言间在主要运动基元上的差别是比较表面的。

④ 允许用户规定执行流程 同一般的计算机编程语言一样，机器人编程系统允许用户规定执行流程，包括转移、循环、调用子程序以及中断等。

对于许多计算机应用，并行处理对于自动工作站是十分重要的。首先，一个工作站常常运用两台或多台机器人同时工作，以减少过程周期。在单台机器人的情况下，工作站的其他设备也需要机器人控制器以并行方式控制。因此，在机器人编程语言中常常含有信号和等待等基本语句或指令，而且往往提供比较复杂的并行执行结构。

通常首先需要用某种传感器来监控不同的过程。然后，通过中断或登记通信，机器人系统能够反映由传感器检测到的一些事件。有些机器人语言提供规定这种事件的监控器。

⑤ 要有良好的编程环境 如同任何计算机一样，一个好的编程环境有助于提高程序员的工作效率。机械手的程序编制是困难的，其编程趋向于试探对话式。如果用户忙于应付连续重复的编译语言的编辑→编译→执行循环，那么其工作效率必然是低的。因此，现在大多数机器人编程语言含有中断功能，以便能够在程序开发和调试过程中每次只执行一条单独语句。典型的编程支撑和文件系统也是需要的。

根据机器人编程特点，其支撑软件应具有下列功能：在线修改和立即重新启动；传感器的输出和程序追踪；仿真。

⑥ 需要人机接口和综合传感信号 在编程和作业过程中，应便于人与机器人之间进行信息交换，以便在运动出现故障能及时进行处理，确保安全。而且，随着作业环境和作业内容复杂程度的增加，需要有功能强大的人机接口。

机器人语言的一个极其重要的部分是与传感器的相互作用。语言系统应能提供一般的决策结构，以便根据传感器的信息来控制程序的流程。

在机器人编程中，传感器的类型一般分为三类：位置检测；力觉和触觉；视觉。如何对传感器的信息进行综合，各种机器人语言都有自己的句法。

(3) 常用编程指令简介

对编程语言的掌握也是实现机器人编程的基本要求，下面介绍几种常见的机器人编程指令。

① 运动指令。

a. 指令介绍。移动指令包含三条：MOVJ、MOVL、MOVC。

MOVJ：关节移动指令，即在运动过程中以关节的方式运动。

指令格式：

MOVJ 代表指令；LP 表示局部变量；0 表示标号，用于区别使用；VJ 表示速度，最大速度为 100%；PL 为平滑度，范围 0~9。

MOVL：直线运动指令，即在运动过程中以直线的方式运动。

指令格式：

MOVL 代表指令；LP 表示局部变量；1 表示标号，用于区别使用；VL 表示速度，最大速度为 1999；PL 为平滑度，范围 0~9。

MOVC：圆弧运动指令，即在运动过程中以圆弧的方式运动。

指令格式：

MOVC 代表指令；LP 表示局部变量；2 表示标号，用于区别使用；VL 表示速度，最大速度为 1999；PL 为平滑度，范围 0~9。一段圆弧轨迹必须是由三段圆弧指令实现的，三段圆弧指令分别定义了圆弧的起始点、中间点、结束点。

b. 说明。


局部变量 (LP)：在某个程序中所使用的变量和其他程序中的相同变量不冲突。例如在程序一中使用了 LP0，也可以在程序二中使用 LP0，这样是不会产生矛盾的。

全局变量 (GP)：在此系统中我们还设置了全局变量，意思是如果在一个程序中使用了 GP0，而后就不可以在其他的程序中使用 GP0 了，否则程序会出现混乱现象，系统将会默认将第二次设定的值覆盖第一次设定的值。

平滑度 (PL)：简单地说就是过渡的弧度，确定是以直角方式过渡还是以圆弧方式过渡。假如两条直线要连接起来，怎么连接，就需要对此变量进行设置。

② 逻辑指令。

WAIT 指令：条件等待指令。

指令格式：

说明：当所设定的条件满足时，则程序往下执行；当所设定的条件不满足时，则程序一直停在这里，直到满足所设定的条件为止。但是，后面还有一个时间的设定，当条件不满足时，在等待后面的设定时间之后，会继续执行下面的程序。

JUMP 指令：跳转指令，包含无条件跳转指令和条件跳转指令两种类型。

格式一：无条件跳转指令。

格式二：条件跳转指令。

说明：在使用此条指令时，要配合使用标号指令。标号就是所要程序跳转到的位置，后面不加条件，只要程序执行到此行，则直接跳到标号所处的位置；后面有条件，当程序执行到该行指令时，程序不一定跳转，只有当后面的条件满足时，程序才跳转到标号所处的位置。

CALL 指令：子程序调用指令，包含有条件调用和无条件调用两种类型。

格式一：无条件调用指令。

格式二：条件调用指令。

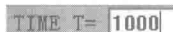
子程序的建立：子程序的建立和主程序的建立唯一的区别就是在编写完所有的程序之后，在程序的末尾加上 RET 指令。

说明：%就是所要调用的程序。后面不加条件，只要程序执行到此行，则直接调用该子程序；后面有条件，当程序执行到该行时，程序不一定调用该子程序，只有当后面的条件满足时，程序才调用该子程序。

TIME 指令：延时指令，以 10ms 为单位。

指令格式：

例：延时 10s。



DOUT 指令：数字量输出。

指令格式：

说明：数字量只有两种形式，因此在使用该指令时只有两种状态，即“ON”和“OFF”两种状态。

AOUT 指令：模拟量输出。

指令格式：

例：使 A0#0 的输出为 2.500。



PAUSE 指令：停止指令，包括无条件停止和有条件停止指令。

格式一：无条件停止。

格式二：有条件停止。

说明：PAUSE 指令后就是所要调用的程序。后面不加条件，只要程序执行到此行，则程序立刻停止；后面有条件，当程序执行到该行时，程序不一定停止，只有当后面的条件满足时，程序才停止。

③ 运算指令。

ADD 指令：加法运算指令。

指令格式：**ADD** **TC** **0** **TC** **1**

说明：执行加法指令时，将前一个变量和后一个变量相加，可以进行的加法指令有：GI、LI、GD、LD、GP、LP、TC、CC 指令。

SUB 指令：减法运算指令。

指令格式：**SUB** **GI** **0** **GI** **1**

说明：执行减法指令时，将前一个变量和后一个变量相减，可以进行的减法指令有：GI、LI、GD、LD、GP、LP、TC、CC 指令。

MUL 指令：乘法运算指令。

指令格式：**MUL** **LI** **0** **LI** **1**

说明：执行乘法指令时，将前一个变量和后一个变量相乘，可以进行的乘法指令有：GI、LI、GD、LD、GP、LP、TC、CC 指令。

DIV 指令：除法运算指令。

指令格式：**DIV** **GD** **0** **LI** **0**

说明：执行除法指令时，将前一个变量除以后一个变量，可以进行的除法指令有：GI、LI、GD、LD、GP、LP、TC、CC 指令。

INC 指令：加 1 运算指令。

指令格式：**INC** **GD** **0**

说明：执行加 1 指令时，将指令后的变量进行加 1，可以进行的加 1 指令有：GI、LI、GD、LD、GP、LP、TC、CC 指令。

DEC 指令：减 1 运算指令。

指令格式：**DEC** **CC** **0**

说明：执行减 1 指令时，将指令后的变量进行减 1，可以进行的减 1 指令有：GI、LI、GD、LD、GP、LP、TC、CC 指令。

SET 指令：置位指令。

指令格式：**SET** **TC** **0** **D** **10.0**

说明：执行置位指令时，将后一个变量的值赋给前一个变量，可以进行的置位指令有：GI、LI、GD、LD、GP、LP、TC、CC 指令。

④ IF 指令：条件判断指令。

IF...END IF 指令

指令格式：**IF** **T** **0** **==** **ON**

说明：判断条件里面的内容是否满足，若条件满足，则执行下面的程序；若条件不满足，则程序不执行 IF...END IF 所包含的内容。若有多个条件进行判断，可以采用 IF...ELSE IF...ELSE...END IF。

例：假如满足条件 X0=ON，就执行 TC#(0) 加 1，若满足条件 X0=OFF，就执行 TC#(0) 减 1，若两个条件都不满足，则将 TC#(0) 里面的值自加。

程序如下：

```
IF    X0 == ON
    INC TC#(0)
ELSE IF X0 == OFF
    DEC TC#(0)
ELSE
```

```

ADD TC#(0) TC#(0)
END IF

```

⑤ WHILE 指令：循环指令。

指令格式：WHILE TC 0 == D 10.0 条件开始。

ENDWHILE 条件结束。

说明：当 WHILE 后的条件满足要求时，即条件为 ON 时，执行 WHILE 里面的程序，直到 WHILE 条件后的指令不满足要求，则退出该循环。

例：当 TC#(1) ≥ 20 时，执行 WHILE 里面的程序 [TC#(1) 的初始值为 50]。

程序如下：

```

SET TC#(1) 50.000
WHILE TC#(1) ≥ 20
{DEC TC#(1) 程序} 循环体
DEC TC#(1)
END WHILE

```

在循环体中，一定要对 TC#(1) 进行设置，否则该程序将会成为死程序，即程序始终在这个地方执行。

⑥ SWITCH 指令：条件选择指令。

指令格式：SWITCH TC 2

CASE 10

DEFAULT

BREAK

ENDSWITCH

例：SWITCH...END SWITCH 指令的应用。

```

SWITCH TC#(2)
CASE 10: AOUT AO#(0)=1.000
CASE 20: AOUT AO#(0)=2.000
CASE 30: AOUT AO#(0)=3.000
CASE 40: AOUT AO#(0)=4.000
CASE 50: AOUT AO#(0)=5.000
CASE 60: AOUT AO#(0)=6.000
CASE 70: AOUT AO#(0)=7.000
CASE 80: AOUT AO#(0)=8.000
CASE 90: AOUT AO#(0)=9.000
DEFAULT: AOUT AO#(0)=10.000
END SWITCH

```

说明：

当 TC#(2) == 10 时，AOUT AO#(0) = 1.000；

当 TC#(2) == 20 时，AOUT AO#(0) = 2.000；

当 TC#(2) == 30 时，AOUT AO#(0) = 3.000；

当 TC#(2) == 40 时，AOUT AO#(0) = 4.000；

当 TC#(2) == 50 时，AOUT AO#(0) = 5.000；

当 TC#(2)=60 时, AOUT AO#(0)=6.000;
 当 TC#(2)=70 时, AOUT AO#(0)=7.000;
 当 TC#(2)=80 时, AOUT AO#(0)=8.000;
 当 TC#(2)=90 时, AOUT AO#(0)=9.000;
 其他情况下, AOUT AO#(0)=10.000。

1.1.3 机器人编程语言的类型

伴随着机器人的发展,机器人语言也得到了发展和完善,机器人语言已经成为机器人技术的一个重要组成部分。机器人的功能除了依靠机器人的硬件支撑以外,相当一部分是靠机器人语言来完成的。早期的机器人由于功能单一,动作简单,可采用固定程序或者示教方式来控制机器人的运动。随着机器人作业动作的多样化和作业环境的复杂化,依靠固定的程序或示教方式已经满足不了要求,必须依靠能适应作业和环境随时变化的机器人语言编程来完成机器人工作。

一般用户接触到的语言都是机器人公司自己开发的针对用户的语言平台,通俗易懂,在这一层次,每一个机器人公司都有自己的语法规则和语言形式,这些都不重要,因为这层是给用户示教编程使用的。在这个语言平台之后是一种基于硬件相关的高级语言平台,如 C 语言、C++ 语言、基于 IEC 61131 标准语言等,这些语言是机器人公司做机器人系统开发时所使用的语言平台,这一层次的语言平台可以编写翻译解释程序,针对用户示教的语言平台编写的程序进行翻译、解释成该层语言所能理解的指令,该层语言平台主要进行运动学和控制方面的编程,再底层就是硬件语言,如基于 Intel 硬件的汇编指令等。

商用机器人公司提供给用户的编程接口一般都是自己开发的简单的示教编程语言系统,如 KUKA、ABB 等,机器人控制系统提供商提供给用户的一般是第二层语言平台,在这一平台层次,控制系统供应商可能提供了机器人运动学算法和核心的多轴联动插补算法,用户可以针对自己设计的产品自由地进行二次开发,该层语言平台具有较好的开放性,但是用户的工作量也相应增加,这一层次的平台主要是针对机器人开发厂商的平台,如欧系一些机器人控制系统供应商就是基于 IEC 61131 标准的编程语言平台。下面来了解一下常见的机器人编程语言。

(1) VAL 语言

VAL 语言是美国 Unimation 公司于 1979 年推出的一种机器人编程语言,主要配置在 PUMA 和 UNIMATION 等型机器人上,是一种专用的动作类描述语言。VAL 语言是在 BASIC 语言的基础上发展起来的,所以与 BASIC 语言的结构很相似。在 VAL 的基础上 Unimation 公司推出了 VAL-II 语言。VAL 语言可应用于上下两级计算机控制的机器人系统。上位机为 LSI-11/23,编程在上位机中进行,上位机进行系统的管理;下位机为 6503 微处理器,主要控制各关节的实时运动。编程时可以 VAL 语言和 6503 汇编语言混合编程。

VAL 语言命令简单、清晰易懂,描述机器人作业动作及与上位机的通信均较方便,实时功能强;可以在离线和在线两种状态下编程,适用于多种计算机控制的机器人;能够迅速地计算出不同坐标系下复杂运动的连续轨迹,能连续生成机器人的控制信号,可以与操作者交互地在线修改程序和生成程序;VAL 语言包含有一些子程序库,通过调用各种不同的子程序可很快组合成复杂操作控制;能与外部存储器进行快速数据传输以保存程序和数据。VAL 语言系统包括文本编辑、系统命令和编程语言三个部分。在文本编辑状态下可以通过键盘输入文本程序,也可通过示教盒在示教方式下输入程序。在输入过程中可修改、编辑、生成程序,最后保存到存储器中。在此状态下也可以调用已存在的程序。系统命令包括位置定义、程序和数据列表、程序和数据存储、系统状态设置和控制、系统开关控制、系统诊断和修改。编程语言把一

条条程序语句转换执行。

为了说明 VAL-II 的一些功能，我们通过下面的程序清单来描述其命令语句：

PROGRAM TEST	程序名
SPEED 30 ALWAYS	设定机器人的速度
height=50	设定沿末端执行器 a 轴方向抬起或落下的距离
MOVES p_1	沿直线运动机器人到点 p_1
MOVE p_2	用关节插补方式运动机器人到第二个点 p_2
REACT 1001	如果端口 1 的输入信号为高电平（关），则立即停止机器人
BREAK	当上述动作完成后停止执行
DELAY 2	延迟 2s 执行
IF SIG(1001) GOTO 100	检测输入端口 1，如果为高电平（关），则转入继续执行第 100 行命令，否则继续执行下一行命令
OPEN	打开手爪
MOVE p_5	运动到点 p_5
SIGNAL 2	打开输出端口 2
APPRO p_6 , height	将机器人沿手爪（工具坐标系）的 a 轴移向 p_6 ，直到离开它一段指定距离 height 的地方，这一点叫抬起点
MOVE p_6	运动到位于 p_6 点的物体
CLOSE	关闭手爪，并等待直至手爪闭合
DEPART height	沿工具坐标系向上移动 height 距离
MOVE p_1	将机器人移到 p_1 点
TYPE "all done"	在显示器上显示 all done

(2) SIGLA 语言

SIGLA 是一种仅用于直角坐标式 SIGMA 装配型机器人运动控制时的一种编程语言，是 20 世纪 70 年代后期由意大利 Olivetti 公司研制的一种简单的非文本语言。这种语言主要用于装配任务的控制，它可以把装配任务划分为一些装配子任务，如取旋具、在螺钉上料器上取螺钉 A、搬运螺钉 A、定位螺钉 A、装入螺钉 A、紧固螺钉等。编程时预先编制子程序，然后用子程序调用的方式来完成。

SIGLA 类语言有多个指令字，它的主要特点是为用户提供定义机器人任务的能力在 SIGMA 型机器人上，装配任务常由若干子任务组成，为了完成对子任务的描述及将子任务进行相应的组合，SIGLA 设计了 32 个指令定义字。要求这些指令定义字能够描述各种子任务；将各子任务组合起来成为可执行的任务。

(3) IML 语言

IML 也是一种着眼于末端执行器的动作级语言，由日本九州大学开发而成。IML 语言的特点是编程简单，能人机对话，适合于现场操作，许多复杂动作可由简单的指令来实现，易被操作者掌握。

IML 用直角坐标系描述机器人和目标物的位置和姿态。坐标系分两种，一种是机座坐标系，一种是固连在机器人作业空间上的工作坐标系。语言以指令形式编程，可以表示机器人的工作点、运动轨迹、目标物的位置及姿态等信息，从而可以直接编程。往返作业可不用循环语句描述，示教的轨迹能定义成指令插到语句中，还能完成某些力的施加。

IML 语言的主要指令有运动指令 MOVE、速度指令 SPEED、停止指令 STOP、手指开合指令 OPEN 及 CLOSE、坐标系定义指令 COORD、轨迹定义命令 TRAJ、位置定义命令