# 第 5 章　机器人规划与人机交互

本章主要介绍任务规划、运动规划、路径规划和人机交互等内容。

机器人得到的任务有时是复杂的，无法一步解决，而任务规划的目的就是将某些比较复杂的问题分解为一些比较小的问题，并进一步分解为机器人可执行的动作。本章介绍了任务规划的作用、问题分解途径、规划域的预测、规划的修正以及机器人规划系统的任务与方法。

在实际应用中，机器人工作空间大多存在障碍物，因此必须考虑机器人执行任务和操作作业时的避碰问题，这称为运动规划。对于运动规划问题，机器人机械手和移动机器人都可采用位形空间的概念，采用有效方式进行公式化表示；其求解技术本质上是算数技术，包括确定性方法、随机性方法和启发式方法。本章第 5.2 节介绍了典型的机器人运动规划方法。

机器人轨迹规划属于机器人底层规划，将介绍在关节空间和工作空间中机器人运动轨迹和轨迹生成方法。所谓轨迹，是指机械手在运动过程中的位移、速度和加速度等项目。例如，在机器人搬运材料的任务中，操作人员只需要指定抓取和放下目标的位置（点对点运动）就可以了，而在加工任务中，末端执行器必须遵循一条期望轨迹（路径运动）。轨迹规划的目的是从对期望运动的简明描述出发，生成相关变量（关节或末端执行器）的时间律，作为运动控制系统的参考输入，以确保机械手完成规划的轨迹。

在机器人规划过程中，任务大多是由人进行下达的。人对机器人下达任务则涉及人与机器人交互的问题。人与机器人的交互（HRI）是和人与计算机的交互（HCI）的发展分不开的。本章 5.2 节将对现阶段人与机器人的交互方式进行介绍，包括基于人机界面交互的示教盒示教系统以及快速发展的语音交互、视觉交互、脑机接口等新型交互方式。

## 5.1　机器人任务规划

任务规划是一种重要的问题求解技术，它从某个特定的问题状态出发，寻求一系列的机器人行为动作，并建立一个操作序列，直到求得目标状态为止。机器人规划（Robot Planning）是机器人学的一个重要研究领域；智能化程度越高，规划的层数越多，用户操作越简单。一般的工业机器人，以轨迹规划为主，高层的规划由人工示教或者智能控制算法完成。服务机器人轨迹规划和高层规划都需要自己完成。

### 5.1.1　任务规划的作用与问题分解

#### 1. 任务规划的概念及作用

在日常生活中，机器人任务规划决定了需要在行动之前决定行动的进程，规划指的是机器人在执行一个任务指令或问题求解程序中的任何一步之前，设计该程序相关步骤的过程。一般一个任务规划是一个行动过程的描述，规划意味着在行动之前决定行动的过程，可用来

监控问题的求解过程，并能够在造成较大的危害之前发现错误。

**2. 问题分解途径及方法**

把某些复杂的问题分解为一些较小的子问题的思想，使应用规划方法求解问题在实际上成为可能。有以下两种实现分解的途径：

(1) 当从一个问题状态移动到下一个状态时，无须计算整个新的状态，而只要考虑状态中可能变化了的部分。例如，一个家庭服务机器人从卧室走动到客厅或厨房，这并不改变两个房屋内门窗的位置。当环境或路径等状态的复杂程度提高时，研究如何决定哪些事物是变化的以及哪些是不变的问题，就显得越来越重要。

(2) 把单一的困难问题分割为若干个较为容易解决的子问题，这种分解能够使困难问题的求解变得容易。但有时候，这种分解是不可行的。替代的方法是，把许多问题看成是待分解的问题，即被分割为只有少量互作用的子问题。

**3. 域的预测和规划的修正**

任务规划的成功取决于问题的另一个方面，即问题论域的可预测性。如果通过在实际上执行某个操作序列以寻找问题的解答，那么在这个过程的任何一步都能够确信该步的结果。但对于不可预测的论域，如果只是通过模拟的方法求解过程，那么就无法知道求解步骤的结果。

最好能考虑可能结果的集合，这些结果很可能按照它们出现的可能性以某个次序排列，然后产生一个规划，并试图去执行这个规划。但对真实世界的任何方面进行完全的预测几乎是不可能的。因此必须随时准备面对规划的失败，并对其进行修改。

### 5.1.2 机器人任务规划基本要素

机器人任务规划的基本要素包括状态空间(State)、时间(Time)、操作状态的动作序列(Actions)、初始和目标状态(Initial and goal states)、准据(A criterion)和运动计划(A plan)。

**1. 机器人状态空间**

(1) 设计问题包括所有可能发生的机器人状态空间。例如，机器人的位置和方向、飞行机器人的位置和速度等。

(2) 离散的和连续的机器人状态空间都是被允许的；应该可以被简洁地用一个计划算法描述。在大多数应用里，状态空间的大小(数目和复杂度)应该尽可能地被简洁描述。

(3) 机器人状态空间是设计问题中最基本的，也是最重要的，应该仔细设计及分析。

**2. 时间**

(1) 所有的设计问题都包括在时间范围内的一系列决策。

(2) 时间模型一定要设计的简洁，能简单反映事实，以便动作执行。

(3) 特殊的时间设计是不必要的，但简洁的时间序列也要保证正确的动作序列。

(4) 一个简洁的时间设计的例子是 Piano Mover's Problem：解决方案是移动钢琴使其到另一个模拟状态，但是特殊的速度在方案中不被专属对待。

**3. 机器人操作状态的动作序列**

(1) 一个机器人计划产生一系列可以改变机器人状态的动作。机器人动作这个术语在

这里可以理解为人工智能中通用的 operators，在控制理论和机器人理论中的对应术语为 inputs 和 controls。

(2) 在设计规范中，当机器人动作序列执行时，状态如何改变是需要进行细致描述的。这就需要一个状态返回函数来处理离散的时间变化或者可微分的连续时间上的变化。

(3) 对应机器人绝大多数动作设计问题，关于时间的函数设计，需要避免直接在状态空间相邻位置连续变换。

**4. 机器人初始和目标状态**

(1) 一个计划问题通常包含机器人初始化的状态和目标状态，过程通过一系列设计的机器人中间状态及动作序列组成。

(2) 初始状态是状态空间的一个特殊点，也是动作序列未发生时的全局状态。

(3) 目标状态是设计的一系列动作执行后，决策者期待经过一系列状态变化后的最终状态。

**5. 准据**

准据是将一个关于机器人状态和动作的计划的输出结果进行编码，形成可执行格式。根据准据的类型，大致有两类方法。

① 可行性：不考虑其效率，只考虑可行性。

② 最佳性：可行的且最优效能，在一些特定条件下，可达到目标状态。

**6. 机器人运动计划**

(1) 总体来说，一个计划利用一个特殊的策略或者行为施加于决策者。

(2) 机器人的运动计划应该使得其动作序列容易被执行。

(3) 预测未来的机器人状态是困难的，因此更多地关注状态转移的实现。

(4) 若不考虑未来的状态，当前机器人状态的最优方案是可以被设计的。

(5) 当前的方法中，利用反馈和活动计划是有效且广泛被采纳的；但是这样的情况下，有些状态不被精确测量。这将被定义成为信息状态，在此之上计划的动作有条件被执行。

### 5.1.3　机器人规划系统的任务与方法

在规划系统中，必须执行下列各项任务：

(1) 根据最有效的启发信息，选择应用于机器人下一步的最优规则。

(2) 应用所选取的规则，计算机器人应用于该规则而生成的新状态。

(3) 对所求得的解答进行检验。

(4) 检验空端，以便舍弃它们，使系统的求解工作向着更有效的方向进行。

(5) 检验正确的解答，并运用具体技术使之完全正确。

规划系统必须具有执行以上步骤任务的方法，方法包含如下所述：

**1. 选择和应用规则**

选择合适的机器人应用规则，最广泛采用的技术是：首先查出机器人的期望目标状态与现有状态之间的差别集合，然后辨识出与减少这些差别有关的规则。如果同时有几种可以使用，则可运用各种启发信息对这些规则加以挑选。

**2. 检验解答与空端**

当规划系统找到一个能够把机器人的初始问题状态变换为目标状态的操作序列时，此系统就成功地求得问题的一个解答。但如何知道求得了一个解答？对于简单问题的处理比较容易，但是对于一些复杂的问题，例如，如果整个状态不是显示地由一个相关特征集合进行描述，则确定解答要困难得多。

解决上述问题的关键是，问题的表示和描述方法。原则上，问题的任何表示方法及它们的组合都可用来描述系统的状态。

**3. 修正殆正确解**

一个求解殆可分解问题的办法是：当执行与所提出的解答相对应的操作序列时，检查求得的状态，并把它们与期望目标比较。

修正一个殆正确的解答的较好办法是，注意有关的出错信息，然后直接加以修正。修正一个殆正确的解答的更好办法是，不对解答进行全面修正，而是不完全确定地让它们保留到最后的可能时刻；然后，当由尽可能多的信息可供利用时，再用一种不产生矛盾的方法完成对解答的详细说明。

### 5.1.4 机器人任务规划的发展现状

机器人任务规划问题(Task Planning)是机器人学科一个重要的研究领域，也是机器人学与人工智能学的一个闪亮结合点。机器人任务规划问题与生活中的规划有类似之处，通常指对机器人任务执行或其他行动的过程进行设计，其目的为制定机器人未来整套行动的一个方案，以达到某种程度的优化。

机器人任务规划是机器人技术研究的重点方向之一，对于提高机器人的智能化水平具有重要意义。机器人领域内的任务规划有两大主要的挑战，第一，机器人工作在非结构化的环境中，任务复杂而多变，难以灵活且有效地对所处的动态环境进行表示和处理。第二，用于描述机器人动作执行序列的构建是任务自主规划的另一个难题。为解决机器人任务规划这两个主要的问题，国内外学者对此展开了广泛而深入的研究。

针对第一个问题，有如下研究。机器人自身具备一定的知识储备，才能更好地提升其任务决策能力。一方面，机器人需要具备语义本体知识。早期，有学者提出基于反应行为的机器人，通过全面预先定义各种反应机制，使没有全局环境知识的机器人做出动作规划；另一方面，机器人需要具备所处空间的环境知识。环境表征的一个主要应用是在本体中通过人工智能推断，处理未知或隐含信息的情况。

针对第二个问题，有如下研究。机器人执行动作序列的构建是决定机器人任务规划是否完整的关键。分层任务规划算法将动作从高层到底层逐层划分，分别使用 STRIPS 结构和 ABSTRIPS 问题解析结构遍历动作序列树生成有效的任务规划。分层任务规划算法因其在构建动作序列方面的高效性和整洁性，成为一直以来高层次任务规划的流行方法。

目前机器人任务规划动作序列的生成大多是通过硬件编码或者预先定义好环境场景，这种方法在灵活性能上具有局限性，因为它需要手动修改源代码，以便对每个任务的动作顺序进行重新编程。总体而言，机器人任务规划目前还基本处于研究阶段，目前机器人规划还没有其他的实用方法，如 STRIPS 是 19 世纪 80 年代的研究成果，但并没有得到实际应用。

### 5.1.5　典型的机器人任务规划系统

典型的机器人任务规划系统有 STRIPS 规划系统、具有学习能力的规划系统和基于专家系统的机器人规划等。

**1. STRIPS 规划系统**

STRIPS 是由 Fikes、Hart 和 Nilsson 三人在 1981—1982 年研究成功的，它是夏凯(Shakey)机器人程序控制系统的一个组成部分。

1) STRIPS 系统组成

(1) 世界模型：为一阶谓词演算公式；

(2) 操作符(F 规则)：包括先决条件、删除表和添加表；

(3) 操作方法：应用状态空间表示和中间—结局分析。

对于一个复杂问题，首先做出规划，即制订求解计划步骤，然后再执行。在执行过程中如出现问题，需要对没有执行的部分计划进行修改，再继续执行。

2) STRIPS 系统的规划过程

每个 STRIPS 问题的解答，就是为某个实现目标的操作符序列，也即达到目标的规划。

**2. 具有学习能力的规划系统**

PULP-Ⅰ机器人规划系统：PULP-Ⅰ机器人规划系统是一种具有学习能力的系统，它采用管理式学习，其作用原理是建立在类比的基础上。

STRIPS 的弱点：需要极其大量的控制器内存和时间等。应用具有学习能力的规划系统能够克服这一缺点。PULP-Ⅰ系统的总体结构如图 5.1 所示。

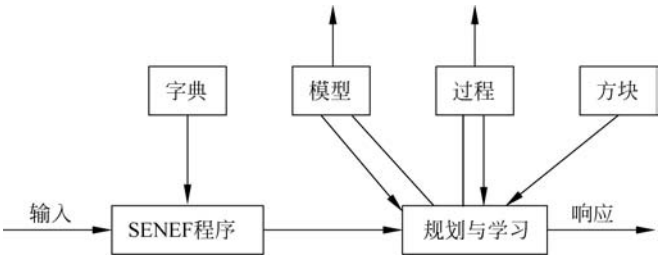



图 5.1　PULP-Ⅰ系统的总体结构

“字典”是英语词汇的集合。

“模型”部分包括模型世界和物体现有状态的事实。

“过程”集中了预先准备好的过程知识。

“方块”集中了 LISP 程序，它配合“规划”对“模型”进行搜索和修正。

PULP-Ⅰ系统的操作方式

PULP-Ⅰ系统具有两种操作方式：学习方式和规则方式。

在学习方式下，输入至系统的知识是由操作人员或者所谓“教师”提供的，如图 5.2 所示。

当某个命令句子送入系统时，PULP-Ⅰ就进入规划方式，如图 5.3 所示。

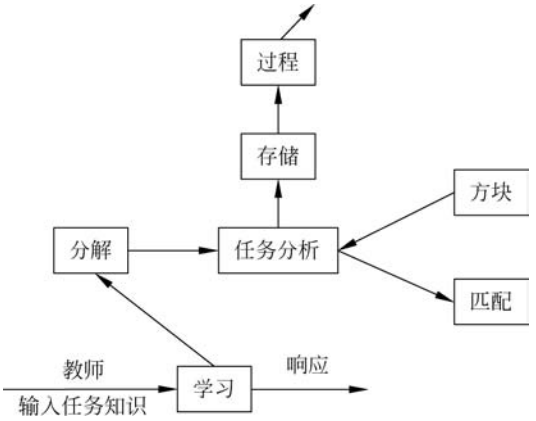



图 5.2 学习方式下 PULP-Ⅰ系统的结构

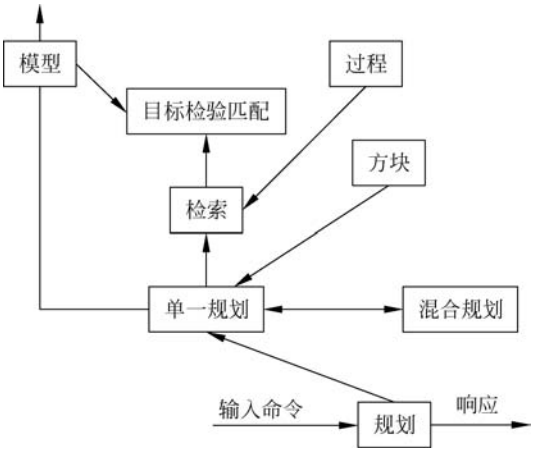



图 5.3 规划方式下 PULP-Ⅰ系统的结构

**3. 基于专家系统的机器人规划**

机器人规划专家系统就是用专家系统的结构和技术建立起来的机器人规划系统。

管理式学习能力的机器人规划系统的不足：

(1) 表达子句的语义网络结构过于复杂。

(2) 与复杂的系统内部数据结构有关的是，PULP-Ⅰ系统具有许多子系统，而且需要花费大量时间编写程序。

(3) 尽管 PULP-Ⅰ系统的执行速度要比 STRIPS 系统快得多，然而它仍然不够快。

系统结构及规划机理如图 5.4 所示。

基于规划的机器人规划专家系统由 5 个部分组成：知识库、控制策略、推理机、知识获取、解释与说明。

基于规划的专家系统的目标就是要通过逐条执行规划及其有关操作，以逐步改变总数据库的状况，直到得到一个可接受的数据库(称为目标数据库)为止。

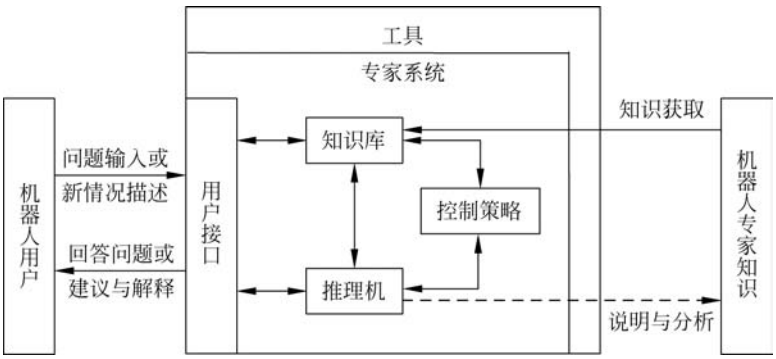



图 5.4　机器人规划专家系统的结构

### 5.1.6　机器人规划系统的任务举例

为说明机器人规划的概念，我们举下面的两个例子：

**例 5.1**　在一些老龄化比较严重的国家，开发了各种各样的机器人专门用于伺候老人，这些机器人有不少是采用声控的方式。例如，主人用声音命令机器人“给我倒一杯开水”，我们先不考虑机器人是如何识别人的自然语言，而是着重分析一下机器人在得到这样一个命令后，如何来完成主人交给的任务。

首先，机器人应该把任务进行分解，把主人交代的任务分解成为“取一个杯子”“找到水壶”“打开瓶塞”“把水倒入杯中”“把水送给主人”等一系列子任务（见图 5.5）。这一层次的规划称为任务规划，它完成总体任务的分解。

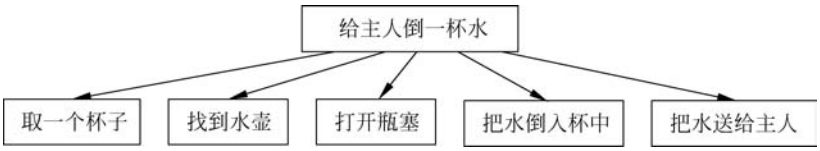



图 5.5　智能机器人的任务分解

然后再针对每一个子任务进行进一步的规划。以“把水倒入杯中”这一子任务为例，可以进一步分解成为一系列动作，这一层次的规划称为动作规划，它把实现每一个子任务的过程分解为一系列具体的动作（见图 5.6）。

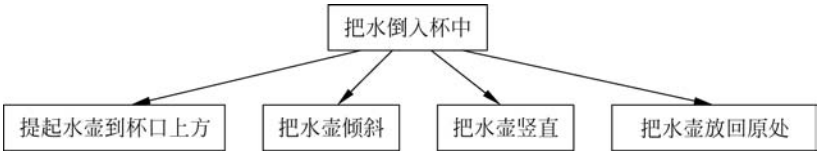



图 5.6　智能机器人的进一步规划

为了实现每一个动作，需要对手部的运动轨迹进行必要的规定，这是手部轨迹规划。

为了使手部实现预定的运动，就要知道各关节的运动规律，这是关节轨迹规划，最后才是关节的运动控制（见图 5.7）。

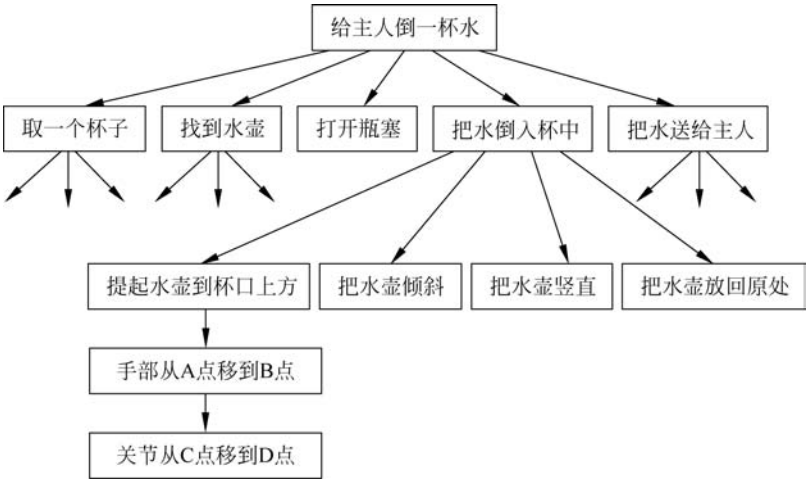



图 5.7 智能机器人的规划层次

机器人的工作过程，就是通过规划将要求的任务变为期望的运动和力，由控制环节根据期望的运动和力的信号产生相应的控制作用，以使机器人输出实际的运动和力，从而完成期望的任务，如图 5.8 所示。机器人实际运动的情况通常还要反馈给规划级和控制级，以便对规划和控制的结果做出适当的修正。

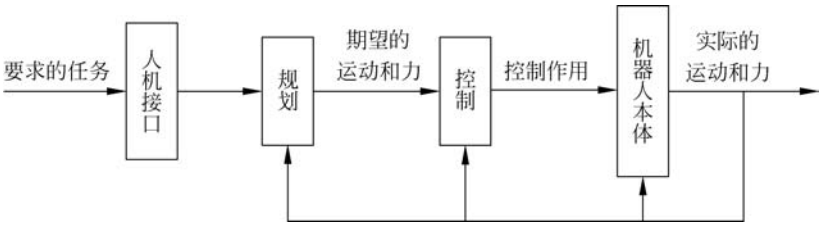



图 5.8 机器人的工作原理

要求的任务由操作人员输入给机器人，为了使机器人操作方便、使用简单，操作人员应该给出尽量简单的描述。

期望的运动和力是进行机器人控制所必需的输入量，它们是机械手末端在每一个时刻的位姿和速度，对于绝大多数情况，还要求给出每一时刻期望的关节位移和速度，有些控制方法还要求给出期望的加速度等。

**例 5.2** 考虑 STRIPS 系统一个比较简单的情况，即要求机器人到邻室去取回一个箱子。机器人的初始状态和目标状态的世界模型如图 5.9 所示。

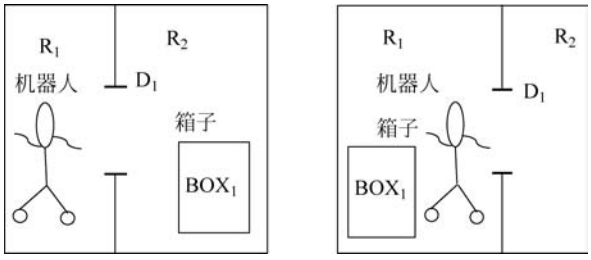



图 5.9 机器人的初始状态和目标状态的世界模型

设有两个操作符，即 gothru 和 pushthru（“走过”和“推过”）。

OP1：gothru(d,$r_1$,$r_2$)

机器人通过房间 $r_1$ 和房间 $r_2$ 之间的 d，即机器人从房间 $r_1$ 走过门 d 而进入房间 $r_2$。

先决条件：INROOM(ROBOT,$r_1$)∧CONNECTS(d,$r_1$,$r_2$)；即机器人在房间 $r_1$ 内，而且门 d 连接 $r_1$ 和 $r_2$ 两个房间。

删除表：INROOM(ROBOT,$S$)；对于任何 $S$ 值。

添加表：INROOM(ROBOT,$r_2$)。

OP2：pushthru(b,d,$r_1$,$r_2$)

机器人把物体 b 从房间 $r_1$ 经过门 d 推到房间 $r_2$。

先决条件：INROOM(b,$r_1$)∧INROOM(ROBOT,$r_1$)∧CONNECTS(d,$r_1$,$r_2$)。

删除表：INROOM(ROBOT,$S$),INROOM(b,$S$)；对于任何 $S$ 值。

添加表：INROOM(ROBOT,$r_2$),INROOM(b,$r_2$)。

这个问题的差别如表 5.1 所示。

**表 5.1　针对差别，须采用标有“×”的操作**

| 差　　别 | 操　作　符 | |
|---|---|---|
| | gothru | pushthru |
| 机器人和物体不在同一房间内 | × | |
| 物体不在目标房间内 | | × |
| 机器人不在目标房间内 | × | |
| 机器人和物体在一房间内，但不是目标房间 | | × |

假定这个问题的初始状态 $M_0$ 和目标 $G_0$ 如下：

M0：INROOM(ROBOT,$R_1$)∧INROOM(BOX1,$R_2$)∧CONNECTS($D_1$,$R_1$,$R_2$)

G0：INROOM(ROBOT,$R_1$)∧INROOM(BOX1,$R_1$)∧CONNECTS($D_1$,$R_1$,$R_2$)

采用中间—结局分析方法，逐步求解这个机器人规划：

(1) do GPS 的主循环迭代，until $M_0$ 与 $G_0$ 匹配为止。

(2) begin。

(3) $G_0$ 不能满足 $M_0$，找出 $M_0$ 与 $G_0$ 的差别。尽管这个问题不能马上得到解决，但是如果初始数据库含有语句 INROOM($BOX_1$,$R_1$)，那么这个问题的求解过程就可以得到继续。GPS 找到它们的差别 $d_1$：INROOM($BOX_1$,$R_1$)，即要把箱子（物体）放到目标房间 $R_1$ 内。

(4) 选取操作符：一个与减少差别 $d_1$ 有关的操作符。根据差别表，STRIPS 选取操作为：

$OP_2$：pushthru($BOX_1$,d,$r_1$,$R_1$)

(5) 消去差别 $d_1$，为 $OP_2$ 设置先决条件 $G_1$ 为：

INROOM($BOX_1$,$r_1$)∧INROOM(ROBOT,$r_1$)∧CONNECTS(d,$r_1$,$R_1$)

这个先决条件被设定为子目标。STRIP 发现：

若 $r_1=R_2$，$d=D_1$，当前数据库含有 INROOM(ROBOT,$R_1$)

那么此过程能够继续进行。现在新的子目标 $G_1$ 为：

$G_1$：INROOM($BOX_1$,$R_2$)∧INROOM(ROBOT,$R_2$)∧CONNECTS($D_1$,$R_2$,$R_1$)

(6) GPS(p)；重复第 3～第 5 步骤，迭代调用，以求解此问题。

步骤 3：$G_1$ 和 $M_0$ 的差别 $d_2$ 为 INROOM(ROBOT,$R_2$)，即要求机器人移到房间 $R_2$。

步骤 4：根据差别表，对应于 $d_2$ 的相关操作符为 $OP_1$：gothru(d，$r_1$，$R_2$)。

步骤 5：$OP_1$ 的先决条件为 $G_2$：INROOM(ROBOT，$R_1$)∧CONNECTS(d，$r_1$，$R_2$)。

步骤 6：应用置换式 $r_1=R_1$ 和 $d=D_1$，STRIPS 系统能够达到 $G_2$。

(7) 把操作符 gothru($D_1$，$R_1$，$R_2$)作用于 $M_0$

删除表：INROOM(ROBOT，$R_1$)

添加表：INROOM(ROBOT，$R_2$)

求出中间状态 $M_1$：INROOM(ROBOT，$R_2$)

INROOM($BOX_1$，$R_2$)

CONNECTS($D_1$，$R_1$，$R_2$)

把操作符 pushthru 应用中间状态 $M_1$

删除表：INROOM(ROBOT，$R_2$)，INROOM($BOX_1$，$R_2$)

添加表：INROOM(ROBOT，$R_1$)，INROOM($BOX_1$，$R_1$)

得到另一中间状态 $M_2$ 为INROOM(ROBOT，$R_1$)

INROOM($BOX_1$，$R_1$)

CONNECTS($D_1$，$R_1$，$R_2$)

$M_2=G_0$

(8) end。

求解过程中，用到的 STRIPS 规则为操作符 $OP_1$ 和 $OP_2$，即 gothru($D_1$，$R_1$，$R_2$)和 pushthru($BOX_1$，$D_1$，$R_2$，$R_1$)。

中间—结局分析法逐步求解机器人规划如图 5.10 所示。

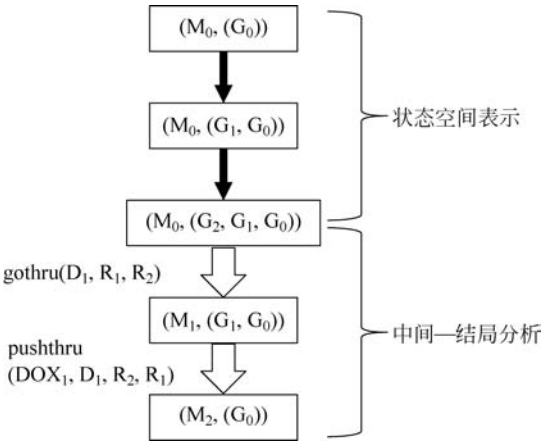



图 5.10 中间—结局分析法逐步求解机器人规划

## 5.2 机器人运动规划方法

在实际应用中，机器人的工作环境大多是分布着障碍物的。安装在工厂生产线上的机器人系统的工作区域经常会有其他的物体，这些物体的存在对于机器人的运动构成了阻碍。例如，在生产线上一个工位中工作的机械手必须避免与可能靠近的运动物体发生碰撞，包含

其他机械手、物流机器人等，同时它也必须避免与其自身结构发生碰撞。在服务机器人领域，例如餐厅中的送餐机器人必须在动态的环境中进行自主导航，避免与固定障碍物（建筑构件、桌椅、各种摆设等）或移动障碍物（工作人员、食客等）发生碰撞。

因此，机器人在完成任务规划后，需要在其工作区域内规划出一条无碰撞的运动路径，以顺利完成其被分配的任务。本节将介绍机器人运动规划的相关内容，包含位形空间、障碍等，并着重介绍多种机器人运动规划方法的原理。

## 5.2.1　机器人运动规划的概念与发展现状

### 1. 机器人运动规划的概念

运动规划是决定一条从初始位姿到最终位姿的路径，使机器人能沿这条路径无碰撞地完成作业任务。这需要为机器人赋予自主规划能力，需要从用户提供的任务级高层描述和工作空间的几何特征出发进行自主规划。

根据工作空间是否已知，运动规划分为离线规划和在线规划。离线规划是在工作空间为完全预先已知的条件下进行的运动规划，在线规划则要求机器人在运动过程中借助自身搭载的传感器对工作空间进行实时感知，并以此进行运动规划。

在空间中寻找一条无碰撞的路径对人类来说是一件很简单的事情，但对机器人来说是一项十分艰巨的任务。其主要原因在于，很难将人们本能地安全穿行于障碍中所依赖的空间感进行复制并转化为机器人可以执行的算法。时至今日，机器人运动规划仍然是一个非常活跃的研究方向，并得益于如相关的算法和理论、计算几何和自动控制等不同领域的研究成果。

无论机器人运动规划属于哪种类别，采用何种规划算法，基本上都遵循以下步骤：

（1）建立环境模型，即对机器人根据所在的现实环境进行抽象后建立相关模型；

（2）搜索无碰路径，是在某个模型的空间中找到符合条件路径的搜索算法。

移动机器人的运动规划问题，可以形式化地描述为以下的推理机：

$$\langle X, x_{\text{init}}, X_{\text{goal}}, U, f, X_{\text{obst}} \rangle$$

其中，$X$ 是搜索空间；$x_{\text{init}} \in X$，表示初始位置（包括姿态、状态等）；$x_{\text{goal}} \in X$，表示目标区域，即目标位置的集合；对于每一个状态 $x \in X$，$U(x)$ 是在状态 $X$ 下所有备选控制的输入集合；状态对控制输入响应由状态转换方程 $f$ 确定。一般来说，当时间是离散的，状态转换方程可以用函数 $f: X \times U \rightarrow X$ 表示；当时间是连续的，状态转换方程可以用一个偏微分方程 $\mathrm{d}x/\mathrm{d}t = f(x, u)$ 表示；$X_{\text{obst}} \subseteq X$ 定义，由一些不允许通行的非法状态组成的集合，也就是 $C$ 空间中的障碍集合。

为定义规划问题的解决方案，考虑一组行为控制序列 $u_1, u_2, \cdots, u_k$，由此序列导出一个状态序列：

$$x_1 = x_{\text{init}},$$
$$x_i = f(x_{i-1}, u_{i-1}) \quad (i = 1, 2, \cdots, k)。$$

如果 $x_{k+1} \in X_{\text{goal}}$ 且 $\{x_1, x_2, \cdots, x_{k+1}\} \bigcap X_{\text{obst}} = \varnothing$，那么称序列 $u_1, u_2, \cdots, u_k$ 为规划问题的一个解决方案。机器人运动规划的目的就是要找到一组满足一定准则的行为控制序列。

**2. 运动规划的发展及现状**

运动规划的研究最早集中在移动机器人领域，移动机器人的运动规划一般称为路径规划(Path Planning)。目前对于移动机器人路径规划技术的研究已经取得了大量的成果，一系列的方法被提出，例如可视图法、自由空间法、栅格法、最优控制法、拓扑法、人工势场法、蚁群算法、遗传算法、模糊逻辑算法和神经网络法等。许多问题获得了比较满意的答案。

根据掌握环境信息的完整程度可以分为环境信息完全已知的全局路径规划、环境信息完全未知或部分未知的局部路径规划。全局路径规划的目的是尽量使规划的效果达到最优。对于全局路径规划已经有了许多成熟的方法，包括可视图法、切线图法、Voronoi 图法、拓扑法、惩罚函数法、栅格法等。局部路径规划由于对环境信息未知，因此以提高机器人的避障能力为主，代表性算法有人工势场法、模糊逻辑算法、遗传算法、人工神经网络、模拟退火算法、蚁群优化算法、粒子群算法和启发式搜索方法等。

机器人运动规划方法经历了几十年的发展，20 世纪 70 年代，科学界开始了对移动机器人路径规划技术的广泛应用与深入研究，怎样为机器人提供一种高效的工作路径是当时的研究核心。早期的机器人路径规划技术主要面向全局静态已知的环境，路径规划技术的最早研究方法是基于直角坐标空间的假设和检验法，该方法由 Pieper 在 1968 年提出，可以起到躲避碰撞的作用；同年，Nilsson 提出了用可视图方法为移动机器人寻找一条无碰撞路径的方法，并描述了具有运动规划能力的机器人。20 世纪 80 年代初，基于 C 空间的自由空间法被麻省理工学院人工智能实验室的 Lozano-Perez 提出，得到了众多研究学者的认可，之后由其拓展的各种几何法和拓扑法也得到了广泛应用。到 20 世纪 80 年代末，Fujimura 等提出相对动态的人工势场法，其将时间参量引入到路径规划模型中，在新的模型中动态障碍物是静态表示的，这样动态路径规划就可以用静态的路径规划算法实现。进入 21 世纪后，机器人运动规划算法取得了快速的发展，在 2000 年，Ge 等人利用优化的人工势场法解决移动机器人的路径规划问题，明确了吸引势场和排斥势场函数及虚拟合力的方向，该方法可用于足球移动机器人；2009 年，Perez 等使用了基于速度场的模糊路径算法解决路径规划问题，该方法处理环境信息不完全已知的情况具有很大优越性。近年来，有许多学者采取模糊逻辑、人工神经网络以及遗传算法等创新的技术解决机器人路径规划的难题，并得到显著的效果。

但对于机械臂的运动规划而言，其规划维度更高。规划维度的增加导致了计算量增加，障碍物更是无法在构型空间(Configuration Space)中进行描述。基于随机采样的规划算法目前是机械臂运动规划领域中最为主流的算法之一，其中以 RRTs 算法最具代表性，RRTs 算法由美国爱荷华州立大学的 Steven M. LaValle 教授在 1998 年提出，RRTs 算法不考虑障碍物在构型空间中的分布情况，它仅仅通过对位形空间中的采样点进行碰撞检测以获取障碍物信息，并在此基础上进行运动规划；RRTs 对同一个规划问题的表现可能时好时坏，连续出现完全相同的规划结果的概率很低。要判断算法对于某一规划问题的效果，往往需要多次反复的试验，因此对 RRTs 的改进以及与其他算法的融合，仍是当前的研究热点方向。很多用于移动机器人路径规划的算法无法应用于机械臂，目前还没有能够兼顾实时性和最优性的算法，因此，运动规划目前是机器人领域的研究热点之一。

机器人运动规划算法主要应用到移动机器人领域，随着移动机器人应用范围的扩大，移动机器人路径规划对规划技术的要求也越来越高，单个运动规划方法有时不能很好地解决

某些规划问题，所以新的发展趋势为将多种方法相结合，如机器人组协同工作的路径搜索算法、静态全局路径搜索算法与动态局部搜索算法相结合、传统规划方法与新的智能方法之间的结合等新方向。

### 5.2.2　位形空间

解决运动规划问题一个非常有效的策略是在适当的空间中将机器人表示成移动质点，并在其中标明工作区和障碍。机器人上各个位置的一个完整规范被称为位形(Configuration)，而由所有可能位形组成的集合被称为位形空间，用 $C$ 表示位形空间。

以下给出几个位形空间的例子。

(1) 对于一个多边形移动机器人，用在固定参考系下本体上一个代表点(如顶点)的位置和多边形朝向描述其位形空间，由此，$C=IR^2\times SO(2)$，其维数为 3。

(2) 对于一个多面体移动机器人，$C=IR^3\times SO(3)$，其维数为 6。

(3) 对于一个单连杆回转机械臂，它的位形空间只是连杆的姿态角，因此，其位形空间 $C=S^1$，$S^1$ 表示单位圆，也可以表示为 $C=SO(2)$，实际上 $S^1$ 和 $SO(2)$的选择不是特别重要，因为这两种是等价的表示方法。不管哪种情况下，我们都可以用参数对 $C$ 进行参数化。对于双连杆平面机械臂，有 $C=S^1\times S^1=T^2$，$T^2$ 表示环面(torus)，这样可以用关节变量向量 $q=(\theta_1,\theta_2)$表示一个位形。直角坐标型机械臂有 $C=R^3$，用 $q=(d_1,d_2,d_3)$表示其中一个位形。

(4) 对于一个有固定基座的 $n$ 个旋转关节平面机械手，位形空间 $C$ 是$(IR^2\times SO(2))^n$的一个子集，维数等于$(IR^2\times SO(2))^n$ 的维数减去因关节带来的约束个数，即 $3n-2n=n$。实际上，一个平面运动链中，每个关节对其后的部分施加了两个非完整约束。

(5) 对于一个有固定基座的 $n$ 个旋转关节空间机械手，位形空间 $C$ 是$(IR^3\times SO(3))^n$的一个子集。维数等于$(IR^3\times SO(3))^n$ 的维数减去因关节带来的约束个数，即 $6n-5n=n$。此时每个关节对其后的部分施加了 5 个约束。

如果位形空间 $C$ 的维数为 $n$，则其中的一个位形可用向量 $q\in IR^n$ 表示。但这样的表示只是局部有效的，这是因为位形空间 $C$ 的几何结构通常比欧几里得空间更加复杂。

### 5.2.3　障碍

位形空间 $C$ 用来描述无碰撞路径规划的代表性方法，障碍是运动物体在位形空间相应的运动禁区。假设操作空间障碍 $O=\{O_i,i=1,2,\cdots,p\}$是封闭的(包含边界)，但一定有界，每一个障碍物 $O_i$ 在位形空间 $C$ 中定义为：

$$CO_i=\{q\in C:B(q)\cap O_i\neq\varnothing\} \tag{5.1}$$

即 $CO_i$ 表示一个在工作空间中会导致机器人 B 和障碍 $O_i$ 发生碰撞(包括接触)的一个位形空间子集。位形空间中障碍物 $O_i$ 膨胀为 $CO_i$。

$CO$ 为上述集合的并集：

$$CO=\bigcup_{i=1}^{p}CO_i \tag{5.2}$$

空间 $C_{\text{free}}$ 为 $CO$ 的补集：

$$C_{\text{free}}=C-CO=\left\{q\in C:B(q)\cap\left(\bigcup_{i-1}^{p}O_i\right)=\varnothing\right\} \tag{5.3}$$

亦即使机器人不会与障碍碰撞的机器人位形空间子集。如果一条位形空间中的路径完全包含在 $C_{free}$ 中，则称为自由路径。

虽然 $C$ 空间本身是连通的(即给定两形位 $q_1$ 与 $q_2$，存在一条路径将它们连接起来)，但因障碍引起的影响，自由位形空间 $C_{free}$ 并不是总是连通的。如果一条线路完全在 $C_{free}$ 中，则这是一条自由(可行)线路。

**障碍举例**

下面将介绍一些典型情况的“$C$ 障碍”产生过程。为简便起见，假设中的障碍具有多边形或多边体外形。

**例 5.3** 如图 5.11 所示，平面中运动的圆形机器人，其位置用中心原点表示，机器人对障碍物的检测不会因为机器人自身的转动而受到影响，因此位形空间 $C$ 就是此平面，位形空间 $C$ 则由障碍物以机器人的半径做等距生长新生成的包络线构成。

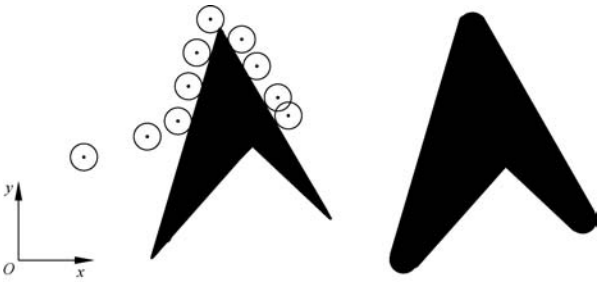



图 5.11 平面中运动的圆形机器人及障碍物

**例 5.4** 对一个在 $IR^N$ 中既可以旋转和平移的多面体机器人，因为需对其方向自由度进行描述，所以其位形空间的维数相对少一些。考虑一个在 $IR^2$ 中平移和旋转的多边形，其位形可以用一个在参考系中描述机器人朝向的方向角 $\theta$ 和一个代表点(如一个顶点)的笛卡儿坐标表示。相应的位形空间是 $IR^2 \times SO(2)$，可以用 $IR^3$ 表示局部。确定一个障碍 $O_i$ 在位形空间 $C$ 中的像，原则上需要对机器人每个可能的朝向角度 $\theta$ 重复如图 5.12 中的过程。“$C$ 障碍”$CO_i$ 是由所有恒值方向时得到的阴影切片“堆垒”(在 $\theta$ 轴方向上)而成的空间体。

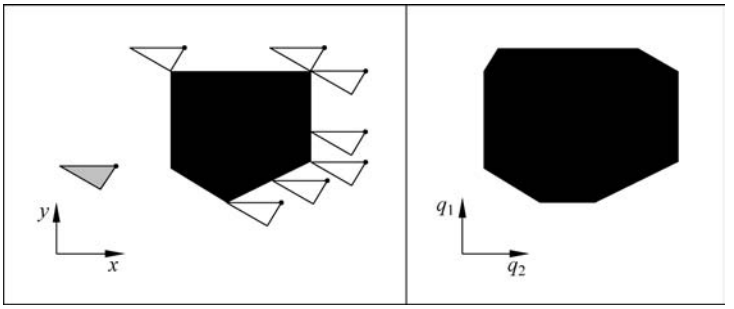



图 5.12 $IR^2$ 中一个平移运动多边形机器人的“$C$ 障碍”

(左图：机器人 B，障碍 $O_i$，以及建立“$C$ 障碍”的生长过程；右图：位形空间 $C$ 和“$C$ 障碍”$CO_i$)

### 5.2.4　图形搜索类规划算法

图形搜索类规划算法的基本思想是按照一些原则将机器人所在的空间用图形的形式加以表示，在给定起始点与目标点后在图形中进行搜索得到连接起始点与目标点的无碰撞路径。根据搜索出的路径是否是随机的，可将图形搜索类算法分为确定性图形搜索和随机图形搜索。图形搜索类规划算法主要有两个步骤：

第一步是图形构建，节点放在何处以及用边将其连接。图形的构建有多种方法。典型的有可视性图法、沃罗诺伊(Voronoi)图法、单元分解法。在可视性图的情况下，道路尽可能地靠近障碍物，且最终最优的路径是极小长度解。在沃罗诺伊(Voronoi)图情况下，路径尽可能地远离障碍物。而单元分解方法的概念是区分自由和被占几何面积的差别，精确的单元分解是无损失的分解，而近似的单元分解代表对原始地图的一个近似。然后，通过单元间的特殊连接关系，形成了图形。第二步是图形搜索，是在以构建的图形中进行(最优)解计算。

而对于随机图形搜索算法，图形的构建与搜索往往是同时进行的，典型的代表有 PRM 算法与 RRT 算法。下面将分别介绍沃罗诺伊图法和单元分解法等算法。

**1. 可视性图**

可视图法通常以障碍物多边形为环境表示，将起始点 $S$、目标点 $G$ 和多边形障碍物的各顶点(设 $V_O$ 是所有障碍物的顶点构成的集合)进行组合连接，要求起始点和障碍物各顶点之间、目标点和障碍物各顶点之间以及各障碍物顶点与顶点之间的连线，均不能穿越障碍物，即直线是“可视的”。给图中的边赋权值，构造可见图 $G(V,E)$，点集 $V=V_O\cup\{S,G\}$，$E$ 为所有弧段即可见边的集合。然后采用某种算法搜索从起始点 $S$ 到目标点 $G$ 的最优路径，则规划最优路径的问题转化为从起始点至目标点经过这些可视直线的最短距离问题。

障碍物环境示意图如图 5.13 所示，$O_1$、$O_2$ 表示的封闭多边形分别代表两个障碍物，$S$、$G$ 分别表示起始点和目标点。其对应的可视图如图 5.14 所示，由起始点、目标点与各障碍物顶点之间的可视直线构成。

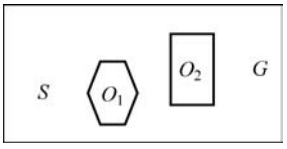



图 5.13　障碍物环境示意图

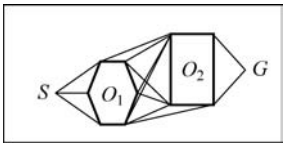



图 5.14　可视图

由此可见，利用可视图法规划避障路径主要在于构建可视图，而构建可视图的关键在于障碍物各顶点之间可见性的判断。判断时主要分为两种情况，同一障碍物各顶点之间可见性的判断以及不同障碍物之间顶点可见性的判断。

(1) 同一障碍物中，相邻顶点可见(通常不考虑凹多边形障碍物中不相邻顶点也有可能可见的情况)，不相邻顶点不可见，权值赋为$\infty$。

(2) 不同障碍物之间顶点可见性的判断则转化为判断顶点连线是否会与其他顶点连线相交的几何问题。如图 5.15 虚线所示，$V_1$、$V_2$ 分别是障碍物 $O_1$、$O_2$

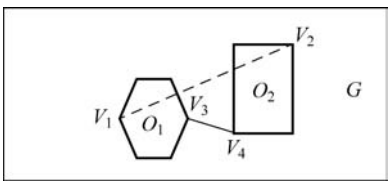



图 5.15　可见性判断示意图

的顶点，但 $V_1$ 与 $V_2$ 连线与障碍物其他顶点连线相交，故 $V_1$、$V_2$ 之间不可见；而实线所示的 $V_3$ 与 $V_4$ 连线不与障碍物其他顶点连线相交，故 $V_3$、$V_4$ 之间可见。

可视图的构建是使用搜索策略的基础，其建立的成功与否影响着最终避障路径的规划结果。但是，可视图中的大部分可视直线在生成避障路径时是用不到的，却又占据着较大的存储空间，对搜索算法的运算效率也会产生极大的考验，且如果障碍物发生变化，可视图还需重新生成。可视性图在移动机器人学的路径规划中比较普遍，部分是因为实现比较简单。特别在连续的或离散的空间中，当环境的表示把环境中的物体描述成多边形时，可视性图可容易地使用障碍物的多边形描述。但可视图法忽略了机器人尺寸，容易造成机器人通过障碍物时与障碍物顶点距离非常近，从而产生摩擦，且该方法搜索时间较长。该方法灵活性较差，一旦机器人的起点和终点发生变化，可视图就需要重新构建，所需的工作量较大。

**2. 沃罗诺伊图**

与可视性图相比，沃罗诺伊图是一种全道路图的方法，它倾向于使图中机器人与障碍物之间的距离最大化。对自由空间中的各点，计算它到最近障碍物的距离。当你离开障碍物而移动时，高度值增加。在离两个或多个障碍物等距离的点上，这种距离图就有陡的山脊，沃罗诺伊图就是由这些陡的山脊点所形成的边缘组成。当方位空间障碍物都是多边形时，沃罗诺伊图仅由直线和抛物线段组成。在沃罗诺伊道路图上寻找路径的算法，像可视性图方法一样，是完备的，因为自由空间中路径的存在意味着在沃罗诺伊图上也存在一条路径（即两种方法都确保完备性）。但是，在总长度的意义上，沃罗诺伊图常常远非最优。

假设在一片林区内设置 $n$ 个火情观察塔 $P_1, P_2, \cdots, P_n$，每个观察塔 $P_i(i=1,2,\cdots,n)$ 负责其附近 $V(P_i)$ 的火情发现及灭火任务。其中 $V(P_i)$ 由距其他 $P_j(j=1,2,\cdots,n, j\neq i)$ 更近的树组成，则 $V(P_i)$ 就是关联于 $P_i$ 的 Voronoi 多边形，由所有的 $P_i(i=1,2,\cdots,n)$ 组成的图就是关于 $n$ 个生成元的沃罗诺伊图。

沃罗诺伊图构建的基本原理为：在一个已确定了尺度的量度平面上，对该平面上分布的离散点集进行区域划分，使划分区域中的点到点集中某一点的距离比其到点集中所有其他点的距离小。具体的数学定义为：

设平面 $B$ 是一个已确定了尺度的量度平面，设 $P$ 为平面上的离散点集，$p_1, p_2, \cdots, p_n \in p$ 且 $p_1, p_2, \cdots, p_n$ 对应的坐标为 $(x_i, y_i)(i=1,2,\cdots,n)$，取 $B$ 平面上任一点 $(x, y)$ 称：

$$\sqrt{(x-x_i)^2+(y-y_i)^2} < \sqrt{(x-x_j)^2+(y-y_j)^2} \tag{5.4}$$

使式(5.4)对于选定点 $p_i$，任取点 $p_i \in p$ 且 $p_i \neq p_j$ 成立的 $(x, y)$ 点形成的轨迹称为离散点 $p_i$ 的沃罗诺伊区域，如图 5.16 所示。

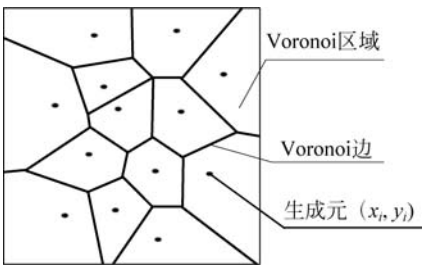


图 5.16 Voronoi 图

经过 Voronoi 区域划分之后，最终把平面 $B$ 划分成 $n$ 个相互邻接的多边形，并且每个多边形中有且只有包含一个离散点。如果点 $(x, y)$ 位于相邻多边形的公共边上，则有任取 $P_k \in P$ 且 $P_i \neq p_j \neq p_k$ 式(5.5)成立：

$$\begin{cases} \sqrt{(x-x_i)^2+(y-y_i)^2} = \sqrt{(x-x_j)^2+(y-y_j)^2} \\ \sqrt{(x-x_j)^2+(y-y_j)^2} < \sqrt{(x-x_k)^2+(y-y_k)^2} \end{cases} \tag{5.5}$$

其中 $P_i$，$P_j$ 为相邻两多边形包含的离散点。我们称由式(5.5)确定出的点$(x,y)$形成的轨迹为离散点集 $P$ 的沃罗诺伊边，$p_1,p_2,\cdots,p_n$ 称为沃罗诺伊图的生成元。

点集沃罗诺伊图的一些性质：

在叙述沃罗诺伊图性质之前，假设在沃罗诺伊图的离散生成元中，没有四个点是共圆的。

性质 1：沃罗诺伊图的每一个顶点恰好是图的三条边的公共交点；

性质 2：在生成元点集$\{p_1,p_2,\cdots,p_n\}$中，$P_i$ 的每一个最邻近点，确定沃罗诺伊多边形 $V(i)$的一条边；

性质 3：沃罗诺伊边是由相邻的两个生成元间的垂直平分线，或是该直线上的线段或射线构成。

**3. 单元分解法**

单元分解法的思想是将机器人所在环境空间切分为多个简单相连的区域，每个区域单元分为自由的和被物体占用的两种。然后找出起点和目标位置所在单元，并在连接图中用搜索算法找到一条连接起点和目标单元的路径。单元分解法又分为精确单元分解(Exact Cell Decomposition)和近似单元分解(Approximate Cell Decomposition)。单元分解法的一个重要评价标准是对环境划分的完备程度，如果环境分解后是无损的，那么这种划分法就是精确单元分解。如果分解形成实际地图的近似，则称为近似单元分解。

1) 精确单元分解

这里，单元的边界建立在几何临界性的基础上。最后得到的单元或各自是完全自由的，或各自被完全占用的。支持这种分解的基本抽象概念是：在自由空间的各单元内，机器人的特殊位置无关紧要，重要的是机器人从各自由空间单元走向其相邻自由单元的能力。精确单元分解法包括梯形图法(见图 5.17)、三角剖分法(见图 5.18)等。

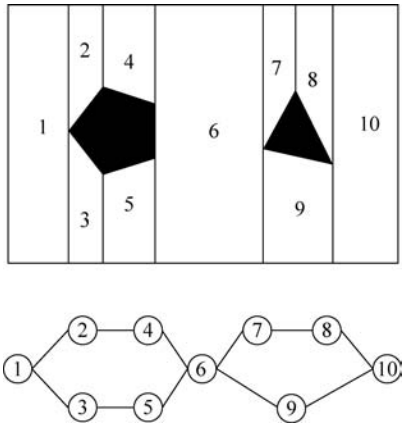



图 5.17　梯形图法

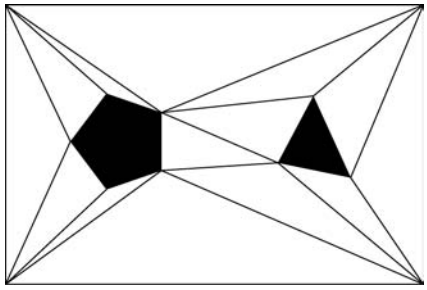


图 5.18　三角剖分法

精确单元分解法的计算效率取决于环境中物体的密度和复杂性，如果在复杂和高密度的环境中，该算法的计算复杂性会增加，而且基于梯形图和三角剖分等算法，很难找到长度最短的路径。

2) 近似单元分解

近似单元分解是移动机器人路径规划中普遍应用的技术之一，这种方法最流行的形式

是栅格法。栅格法是一种最直接的对环境描述的方法。如图 5.16 所示,在用栅格法描述环境时,把一系列的格子“镶嵌”到要描述的环境中去,通过格子的状态描述环境信息。假设在环境中,用黑色表示障碍物,用白色表示可行路径,则落在黑色区域的格子不可行,落在白色区域的格子可行。然后,可以利用搜索算法找到一条由起点到终点的只穿越白色可行格子的路径。

栅格法的最大优点是其把整个环境作为一个整体进行描述,而不是对每一个障碍物进行描述。这样一来,就极大地减少了在环境中区分不同障碍物并获取其几何特性的麻烦,从而使其具有特别强的通用性。但是,此种描述环境的方法也有一个明显的不足之处,即对环境描述的精确程度完全取决于格子的大小。当格子取得越小越密时,对环境的描述就越精确。当格子取得越大越稀疏时,对环境的描述也就越粗糙。但同时,格子的大小、多少又直接决定了存储量的大小和搜索空间的大小,即格子越多越密,存储量越大,搜索空间就越大;格子越少越稀疏,存储量越小,搜索空间就越小;而这些又影响了计算的复杂度和实时性。

为了解决这个矛盾,于是有学者提出了分级递归的思想,也称为“四叉树”法。如图 5.19 所示,即在用栅格法描述环境的时候,对环境进行分级递归的划分。包围自由空间的矩形格子被分解成四个相同的矩形,当矩形格子完全落在“黑”色区域,或者完全落在“白”色区域时就停止对其做进一步的划分;否则,当格子中既包含“黑”色,又包含“白”色时,则继续对其进行划分,直至得到某种预定的解。与精确单元分解法相比,近似的方法可能牺牲完备性,但它很少涉及数学几何运算,所以比较容易实现。

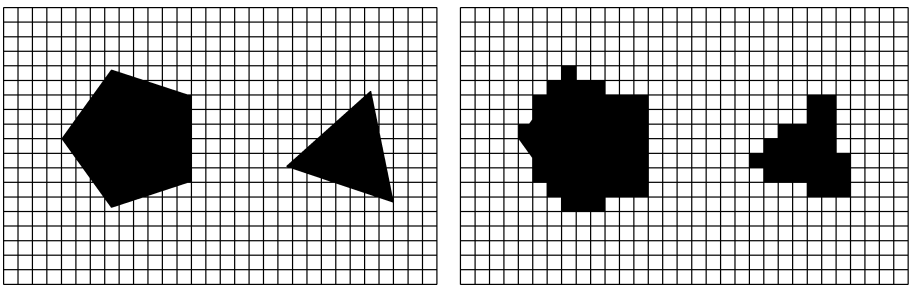


图 5.19 栅格法

**4. 随机图形搜索**

随机图形搜索代表着一类非常高效的规划方法,尤其对高维位形空间中的规划问题更是如此。它们属于基于抽样(Sampling-based)的方法族,其基本思路是:确定一个能充分表示 $C_{\text{free}}$ 连通性的有限避碰位形集合,并利用该位形集合建立用于解决运动规划问题的路径图。

实现该思路的途径是在每步迭代过程中抽取一个位形样本并检查是否会使机器人与工作空间内的障碍发生碰撞。如果结论是肯定的,就丢弃该样本。而对一个不会导致碰撞的位形,则将其加入当前路径图中并与其他已经记录的位形建立可行的连接即可。

1) PRM 方法

PRM 算法将路径规划问题求解分为两个阶段来完成:学习阶段(learning phase)和查询阶段(query phase)。在学习阶段,PRM 算法主要是利用局部规划器,通过对整个位姿空间进行某种方式的采样,构建一张自由空间内的无向路径图 $G(V,E)$,其中 $V$ 是顶点的集

合，$E$ 是 $C_{free}$ 中路径的集合。在查询阶段通过图形搜索算法，在构建的路径图中找到一条从起点到终点的连通路径，该路径包含了一系列通过边连接在一起的采样点。PRM 算法的扩展示意图如图 5.20 所示，其具体执行步骤如下：

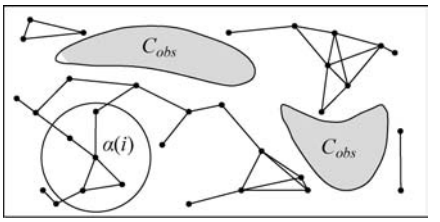


图 5.20　基本 PRM 扩展示意图

(1) 初始化路径图 $G(V,E)$、终止循环数 $K$、最大连接边数 $i$、步长 $\lambda$ 等。在终止循环数 $K$ 范围内循环执行(2)～(4)，对 $G$ 进行扩展；

(2) 通过某种采样机制得到一个随机采样点 $\alpha(i)$；

(3) 对 $\alpha(i)$进行碰撞检测，若发生碰撞，则进入下一循环，未碰撞则继续当前循环；

(4) 将 $\alpha(i)$加入 $V$ 中，找到 $V$ 中与 $\alpha(i)$间距离小于设定步长 $\lambda$ 的点集，对于集合中每个点 $q$ 与 $\alpha(i)$之间进行碰撞，若发生碰撞，则检查下一个点，未发生碰撞则新建边 $\alpha(i)$并加入 $E$ 中；

(5) 在路径图中导入起始点和终止点；

(6) 利用图搜索算法在 $G$ 中搜索连接起点和终点的可行路径，直到找到最短路径或者搜索失败。

基本 PRM 算法构建路径图的伪代码如下：

```
BUILD_ROADMAP
1    G.init(); i←0;
2    while i<N do;
3        if α(i)∈C_free
4            G.add_vertex(); i←i+1;
5            for each q∈neighborhood(α(i),G)
6                if (not G.same_component(α(i).q)) and CONNECT(α(i),q)
7                    G.add_edge(α(i),q);
```

PRM 算法通过碰撞检测判断采样点和边是否处于自由空间 $C_{free}$，而不需要精确计算空间，避免了描述位姿空间时进行大量的计算；此外算法具有概率完备性，即对于任意存在可行解的复杂求解空间，当 PRM 算法的采样数量足够多时，一定能够找到一条可行的路径。由于算法在构建路径图的过程中更倾向于探索未知空间，并且把有效的采样点加入路径图中，因此对于具有多个起点或终点的分支路径问题求解来说，PRM 算法可以将尽可能多的路径解包含在路径图中。

PRM 算法进行路径规划的两个阶段中，学习阶段构建的路径图的质量直接关系到查询阶段获得的路径质量的好坏，前者在计算过程中耗费的时间也远比后者多，因此国内外学者对算法学习阶段的改进进行了大量的研究，主要集中在采样点的生成方面，同时在其他方面也有相关的改进。

2) RRT 方法

快速扩展随机树(Rapidly-exploring Random Tree)最初是由美国科学家 S. M. LaVall 在 1998 年提出，此算法可以解决在高维空间内的非完整性约束的问题。他是基于以下设想提出：将机器人的起始点设置为树的根节点，在书中根据某种原则选择一个已有节点，然后

在这个选择的节点上根据机器人的约束条件做扩展，产生一个新的节点，然后将此新节点添加到树中，如此反复直到找到目标点为止，算法要保证随机采样可使机器人向未被探索过的空间探索，并在探索过程中逐步推进搜索树。

以下为随机数扩张的伪代码：

```
Build RRT(x_init, x_goal)
1 T.init(x_init)
2 for k = 1 to K do
3     x_rand ← Random_state()
4     Extend(Xrand, T)
5       if Reach; break;
6 return T
```

```
Extend(T, x_rand)
1 xnear ← Nearest_Neighbor(Xrand, t);
2 u ← Control_Select(U)
3 x_new = Newstate(u, x_near)
4 T.add_vertex(x_new)
5 T.add_edge(xnear, xnew, unew)
```

基本的 RRT 算法的伪代码如上所示，首先我们要建立一棵树 T，起始点 $x_{\text{init}}$ 为其根节点。在状态空间中随机选取某个状态点 $x_{\text{rand}}$，并执行 Extend 函数向此状态点对数进行扩展。首先我们要找到树 T 中离 $x_{\text{rand}}$ 最近的点 $x_{\text{near}}$。然后选择一个控制输入量 $U$，将其作用在 $x_{\text{near}}$ 上得到新的状态点，其扩展过程如图 5.21 所示。不考虑运动约束的情况下，状态空间和位姿空间相等，一般情况下，对控制变量的选择为对扩张方向的选择，一般在 $x_{\text{near}}$ 和 $x_{\text{rand}}$ 的连线上，前进一个步长得到 $x_{\text{new}}$。在非完整型约束下，控制变量可以随机选取，也可以将控制集合中的所以变量都进行尝试(若 $U$ 是一个连续集，则可以先将其进行离散化)。得到 $x_{\text{new}}$ 点后将 $x_{\text{new}}$ 和 $x_{\text{near}}$ 做链接，RRT 算法流程如图 5.22 所示。Extend 有三种可能的结果：

(1) 若 $\| x_{\text{new}} - x_{\text{goal}} \| < \varepsilon$，$\varepsilon$ 为事先设定好的比较小的常数，则可以将 $x_{\text{new}}$ 和 $x_{\text{goal}}$ 看作是同一点，搜索成功。

(2) 将 $x_{\text{new}}$ 添加到树上，但是 $x_{\text{new}} \neq x_{\text{goal}}$。

(3) 通过碰撞检测，发现并不位于自由空间中，或者从点到点的路径与障碍物发生碰撞。

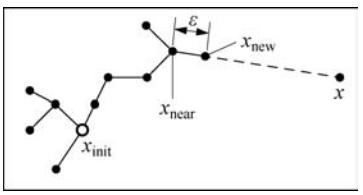



图 5.21 静态环境下树的扩展过程

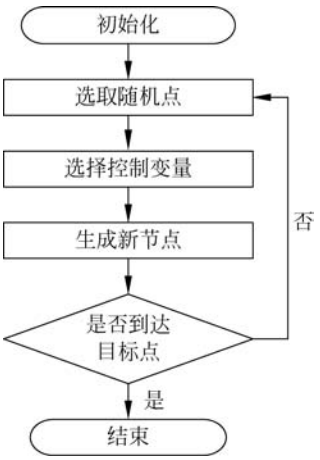



图 5.22 RRT 算法流程

RRT 算法的特点分析：

RRT 作为路径规划的经典算法，主要有以下的特点：

(1) RRT 算法的流程比较简单,只需要进行重复迭代就可以了;

(2) RRT 算法是一种概率完备的算法,即只要时间充足,迭代的次数足够,一定能找到一条合适的路径;

(3) RRT 算法会倾向于像没有被探索过的空间做扩展。

### 5.2.5　基于人工势场的规划方法

人工势场法在机器人导航和路径规划方面有广泛的应用,最初由 Khatib 在 1985 年提出。人工势场是对机器人运行环境的一种抽象描述,它将物理学中场的概念引入到规划环境的表达中。这种方法的基本思想是引入一个称为人工势场的数值函数描述空间结构,通过势场中的力引导机器人到达目标。这种势场分两种:目标产生的吸引势和障碍产生的排斥势。吸引势使机器人接近目标,排斥势使机器人避开障碍,二者的叠加构成机器人运动的虚拟势场。

人工势场法是一种拟物方法,按势函数的不同选取方法,可以分为牛顿型势场(Newton Potential Field)、圆形对称势场(Circular Symmetry Potential Field)、虚拟力场(Virtual Force Field)、超四次方势场和调和势场(Harmonic Potential Field)。这种方法的实质是使障碍物分布情况及其形状等信息反映在环境每一点的势场值中,机器人依次决定行进方向。

人工势场法的优点是机器人的运行环境可以直接对系统路径的生成起到闭环控制的作用,因此可以加强导航系统的动态避障能力,从而提高导航系统对环境的适用性。人工势场法具有一个不可避免的缺陷,也即局部极小点问题。所谓局部极小点,就是在状态空间中的某些区域内,机器人受到了多个势函数的作用,造成了其所受到的斥力与引力的相等的点,也即在该点处,机器人受到的合力为零,进而产生“死锁现象”,使机器人滞留在该局部极小点处,无法成功地到达目标点。通常情况下,障碍物越密集,机器人所受到的势能函数也就越多,产生局部极小点问题的概率也就越大。

**1. 人工势场法的基本原理**

假设在位姿空间中,任意一个位姿用 $q$ 表示,势场用 $U(q)$ 表示,目标状态位姿用 $q_g$ 表示,与目标位姿相关联的吸引势 $U_{att}(q)$ 及和障碍物相关联的排斥势 $U_{rep}(q)$。那么位姿空间中某一位姿的势场可用下面的公式表示:

$$U_q = U_{att}(q) + U_{rep}(q) \tag{5.6}$$

假定每一个位姿 $U(q)$ 都是可微的,那么机器人所受的合力为:

$$\vec{F}(q) = -\nabla U(q) = -\nabla U_{att}(q) - \nabla U_{rep}(q) \tag{5.7}$$

$\nabla U(q)$ 表示势场在 $q$ 点的梯度,该方向是 $q$ 点势场变化率最大的方向。吸引势场和排斥势场采用静电力势场模型进行定义:

$$\begin{aligned} U_{att}(q) &= \frac{1}{2}\xi\rho_g^2(q) \\ U_{rep}(q) &= \begin{cases} \dfrac{1}{2}\eta\left(\dfrac{1}{\rho(q)} - \dfrac{1}{\rho_0}\right), & \rho(q) \leqslant \rho_0 \\ 0, & \rho(q) > \rho_0 \end{cases} \end{aligned} \tag{5.8}$$

由式(5.7)和式(5.8)可以得到机器人所受吸引力、排斥力为:

$$\boldsymbol{F}_{att}(q)=-\xi(q-q_g)$$
$$\boldsymbol{F}_{rep}(q)=\frac{\eta}{\rho^2(q)}\left(\frac{1}{\rho(q)}-\frac{1}{\rho_0}\right)\nabla\rho(q) \tag{5.9}$$

继而可以得到控制机器人运动的加速度：

$$a_k=\frac{\boldsymbol{F}(q_k)}{\|\boldsymbol{F}(q_k)\|}a_0 \tag{5.10}$$

设系统对环境的采样周期为 $T_0$，系统实际位姿为 $q(k)=(x_k,y_k,\theta_k)$，经过一个采样周期后，系统的位姿变成 $q(k+1)=(x_{k+1},y_{k+1},\theta_{k+1})$，则有：

$$\begin{cases}x_{k+1}=x_k+a_{xk}T_0^2/2\\ y_{k+1}=y_k+a_{yk}T_0^2/2\end{cases} \tag{5.11}$$

利用上述公式计算环境中每一点的势场，机器人作为一个质点，在势场力的引导下从起点开始移动，直到终点结束，其移动轨迹即为规划路径。

梯度势场算法的描述如下：

(1) 输入：受控机器人初始位姿 $q_i$、目标位姿 $q_g$ 和障碍物信息；

(2) 过程：从 $q_i$ 开始每次计算当前位姿 $q_k$ 的势场力 $\boldsymbol{F}(q_k)$ 并沿其方向前进一个小步长 $\delta_k$，$\delta_k$ 根据当前位姿设置不同的值，$\delta_k$ 必须足够小，以免在从 $q_k$ 到 $q_{k+1}$ 的过程中碰到障碍物；

(3) 重复步骤(2)，一直到找到 $q_g$ 或无路可走时结束；

(4) 输出：一条连接 $q_i$ 和 $q_g$ 的位姿序列或指出该序列不存在。

人工势场算法的流程图如图 5.23 所示。

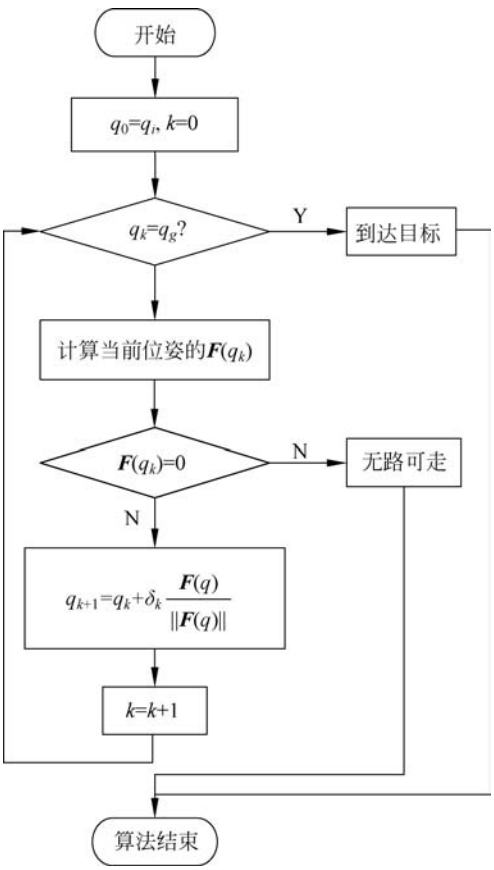


图 5.23 人工势场法路径规划流程图

势场的形式根据实际需要而定，把机器人等效成质点，使用位置相关的势函数解决机器人的避碰问题，以便在实际系统中得以实现。

**2. 引力势函数选取**

机器人无限接近目标点时，其所受势能为零，表示机器人已经到达了目标点，引力势函数可如下表示：

$$U_a=K_g d \tag{5.12}$$

$d$ 为机器人和目标点之间的距离；$K_g$ 为动态参数；引力函数表示为：

$$\boldsymbol{F}_a=K_g \tag{5.13}$$

障碍物斥力分别为 $\boldsymbol{F}_i(i,\cdots,n)$，$n$ 表示障碍物个数，合力 $\boldsymbol{F}_{\text{r\_totle}}$ 为：

$$\boldsymbol{F}_{\text{r\_totle}}=\sum_{i=1}^{n}\boldsymbol{F}_i \tag{5.14}$$

机器人 $R$ 受到合力为：

$$\boldsymbol{F}=\boldsymbol{F}_{\text{a\_totle}}+\boldsymbol{F}_{\text{r\_totle}} \tag{5.15}$$

经上述分析可知，在机器人的路径规划过程中，可以根据 $\boldsymbol{F}$ 的方向决定机器人的运动方向，进一步调用机器人的底层控制模块，实现机器人避障功能。

**3. 斥力势函数选取**

由于在人工势场中障碍物 $OR_i$ 产生的势场对机器人 $R$ 将会产生排斥作用，距离越小，排斥作用越大；反之依然。这种势场与电势场比较类似，与距离成反比，一般可以将斥力函数表达为以下形式：

$$U_r=\begin{cases}K_0/\rho, & \rho\leqslant\rho_m\\ K_0/\rho_m, & \rho>\rho_m\end{cases} \tag{5.16}$$

其中，$\rho_m$ 定义为势场最大范围作用域；$\rho$ 表示为障碍物 $OR_i$ 到机器人 $R$ 的距离；$k_0$ 为加权系数。

机器人所受斥力函数可用下式表示：

$$\boldsymbol{F}_r=-\nabla(U_0)=-\mathrm{d}U_0/\mathrm{d}\rho=\begin{cases}k_0/\rho^2, & \rho\leqslant\rho_m\\ 0, & \rho>\rho_m\end{cases} \tag{5.17}$$

此时，斥力方向为背离障碍物。

(1) 当 $\rho\to\rho_0$ 时，$F_r\to\infty$，同时为使 $\boldsymbol{F}_r$ 连续，并且 $\boldsymbol{F}_r$ 可修改为：

$$\boldsymbol{F}_i=\begin{cases}\dfrac{K_0}{(\rho-\rho_0)^2}-\dfrac{K_0}{(\rho_m-\rho_0)^2}, & \rho\leqslant\rho_m\\ 0, & \rho>\rho_m\end{cases} \tag{5.18}$$

(2) 当 $\rho\to0$ 时，$F_r\to\infty$，也即当机器人与障碍物相碰时，机器人受到的斥力将会变成无穷大；要避免机器人与障碍物相碰，则可以设置一个最小安全距离 $\rho_0$。

$\rho_0$ 和 $\rho_m$ 的取值是由机器人速度、尺寸和实际环境中障碍物的稀疏程度等因素有关。

**4. 局部极点的问题处理**

实际过程中，采用上述的路径规划算法，通常会存在局部极点的问题，即在机器人尚未到达目标点之前的某个运动位置上，所受合力为零，导致机器人停止不前。局部极点问题表现为以下缺陷：

(1) 当目标附近有障碍物时无法到达目标点；

(2) 在狭窄通道中不能发现路径，也即碰撞问题；

(3) 在障碍物前发生振荡；

(4) 存在陷阱区域。

针对局部极值点问题，许多研究者进行了深入的研究和改进。

目前主要有采用随机势场法，采用随机运动逃脱局部的最小点的方法；引入模拟退火

算法和一些启发性知识避免局部极点问题的方法；把人工势场和蚁群算法或神经网络相结合克服局部极点的方法。从本质上讲，上述的克服局部极点的方法的基本思想都是尽量避免机器人在运动途中，产生势场合力为零的情况，或者是在运动到局部极点之后，设计某种规则，使机器人改变运动方向，最终脱离局部极点的位置。

针对局部极点问题提出了如下的解决方案，设某一时刻机器人所处点为 $R(x_i, y_i)$，势力场对机器人的作用点为 $V(x_v, y_v)$，在产生局部极小值情况下，让机器人垂直于受力方向，按逆时针方向移动 $a$ 步长，其算法描述为：

(1) 当机器人处于局部极小值时，机器人的受力方向为：$k=(y_i-y_v)/(x_i-x_v)$；

(2) 机器人的垂直受力方向为：$k'=-(x_i-x_v)/(y_i-y_v)$；

(3) 按逆时针移动后，机器人所在位置：$(x_i+a, k'x_i+k'a+y_i)$。

## 5.3 机器人轨迹规划

机器人轨迹规划意味着在行动之前决定行动的进程；机器人轨迹规划是一个对机器人行动过程的描述。

机器人自动轨迹规划是一种重要的问题求解技术，它从某个特定的问题状态出发，寻求一系列行为动作，并建立一个操作序列，直到求得目标状态为止。与一般问题求解相比，自动轨迹规划更注重于求解过程。此外轨迹规划要解决的问题往往是真实问题，而不是抽象的数学模型问题。机器人轨迹规划是机器人学的一个重要研究领域，研究机器人各种控制求解问题。

### 5.3.1 机器人轨迹规划概述与发展现状

#### 1. 机器人轨迹的概念

机器人轨迹泛指机器人在运动过程中的位移、速度和加速度，也可定义为机器人运动构件的位姿和位姿变化情况。多数是指机器人末端执行器的位姿和位姿变化情况。

机器人运动轨迹的描述一般是对其末端执行器位姿变化的描述。控制轨迹也就是按时间控制手部走过的空间路径。在轨迹规划中，也常用点表示机器人在某一时刻的状态，或某一时刻的轨迹，或用它表示末端执行器的位姿，例如起始点、终止点就分别表示末端执行器的起始位姿及终止位姿。

#### 2. 轨迹规划的一般性问题

机器人在作业空间要完成给定的任务，其手部运动必须按一定的轨迹进行。轨迹规划是根据作业任务的要求，计算出预期的运动轨迹。

机器人轨迹的生成一般是先给定轨迹上的若干个点，将其经运动学反解映射到关节空间，对关节空间中的相应点建立运动方程，然后按这些运动方程对机器人关节进行插值，用于机器人关节运动的控制，从而实现作业空间的运动要求，这一过程通常称为机器人轨迹规划。

(1) 机器人的作业可看作是工具坐标系$\{T\}$相对于工件坐标系$\{S\}$的一系列运动。如

图 5.24 所示，将销插入工件孔中作业，可以借助工具坐标系的一系列位姿 $P_i(i=1,2,\cdots,n)$ 描述。

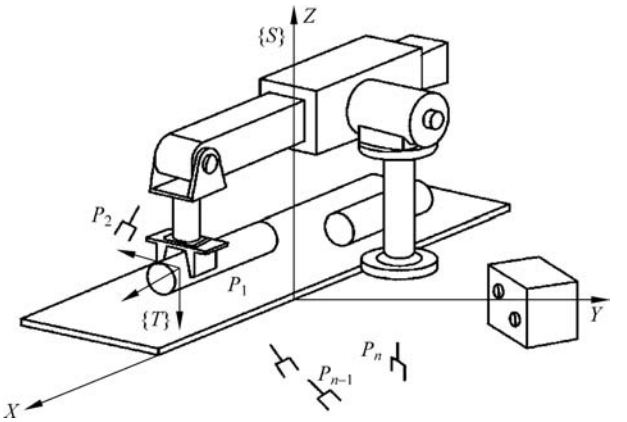



图 5.24　机器人将销插入工件孔中的作业描述

(2) 用工具坐标系相对于工件坐标系的运动描述作业路径是一种通用的作业描述方法。它把作业路径描述与具体的机器人、手爪或工具分离开来，形成了模型化的作业描述方法，从而使这种描述既适用于不同的机器人，也适用于在同一机器人上装夹不同规格的工具。把图 5.25 所示的机器人从初始状态运动到终止状态的作业，看作是工具坐标系从初始位置$\{T_0\}$变化到终止位置$\{T_f\}$的坐标变换。

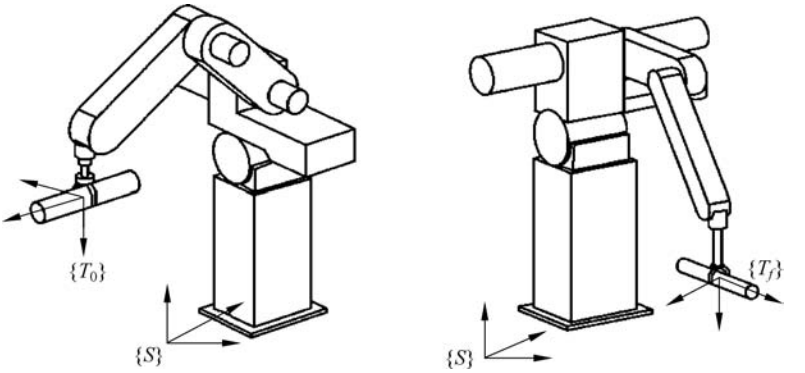



图 5.25　机器人的初始状态和终止状态

(3) 更详细地描述运动时不仅要规定机器人的起始点和终止点，而且要给出介于起始点和终止点之间的中间点，也称路径点。这时，机器人运动轨迹除了位姿约束外，还存在着各路径点之间的时间分配问题。

(4) 机器人的运动应当平稳，不平稳的运动将加剧机械部件的磨损，并导致机器人的振动和冲击。为此，要求所选择的运动轨迹描述函数必须连续，且它的一阶导数(速度)，有时二阶导数(加速度)也应该连续。

(5) 轨迹规划既可以在关节空间中进行，也可以在直角坐标空间中进行。在关节空间中进行轨迹规划是指将所有关节变量表示为时间的函数，用这些关节函数及其一阶、二阶导数描述机器人预期的运动；在直角坐标空间中进行轨迹规划是指将手爪位姿、速度和加速度表示为时间的函数，而相应的关节位置、速度和加速度由手爪信息导出。

**3. 机器人轨迹的生成方式**

(1) 机器人示教—再现运动。这种运动由人手把手示教机器人,定时记录各关节变量,得到沿路径运动时各关节的位移时间函数 $q(t)$;再现时,按内存中记录的各点的值产生序列动作。

(2) 机器人关节空间运动。这种运动直接在关节空间中进行。由于动力学参数及其极限值直接在关节空间里描述,所以用这种方式求最短时间运动很方便。

(3) 机器人空间直线运动。这是一种直角空间里的运动,它便于描述空间操作,计算量小,适宜简单的作业。

(4) 机器人空间曲线运动。这是一种在描述空间中用明确的函数表达的运动,如圆周运动、螺旋运动等。

**4. 机器人轨迹规划涉及的主要问题**

为了描述一个完整的作业,往往需要将上述运动进行组合。这种规划涉及以下问题:

(1) 用示教方法给出轨迹上的若干个节点。

(2) 用一条轨迹通过或逼近节点,此轨迹可按一定的原则优化,如加速度平滑得到直角空间的位移时间函数 $X(t)$或关节空间的位移时间函数 $q(t)$;在节点之间如何进行插补,即根据轨迹表达式在每一个采样周期实时计算轨迹上点的位姿和各关节变量值。

(3) 以上生成的轨迹是机器人位置控制的给定值,可以据此或根据机器人的动态参数设计一定的控制规律。

(4) 规划机器人的运动轨迹时,尚需明确其路径上是否存在障碍约束的组合。一般将机器人的规划与控制方式分为四种情况,如表 5.2 所示。

**表 5.2 机器人的规划与控制方式**

| 状态 | | 障碍约束 | |
|---|---|---|---|
| | | 有 | 无 |
| 路径约束 | 有 | 离线无碰撞路径规划+在线路径跟踪 | 离线路径规划+在线路径跟踪 |
| | 无 | 位置控制+在线障碍探测和避障 | 位置控制 |

**5. 机器人轨迹规划的发展及现状**

随着 21 世纪工业自动化的不断发展,工业生产领域的很多人工作业都被工业机器人所取代,在工业应用中,关节型机器人(Articulated Robot,也称作机械手或机械臂)最为常见,其主要特点是模仿人类身体从腰部到手部的构造,形式上从二自由度到冗余自由度不等。轨迹规划,是机器人设计中一个非常关键的技术模块,轨迹规划的重点是研究满足用户多样化需求的各种轨迹规划算法。工业机器人的轨迹规划在机器人的运动控制中占据着重要的位置,其不但直接控制着机器人末端执行器的工作方式,还对机器人的运动效率、能量消耗、平稳运行和使用寿命有着较大的影响,所以轨迹规划成了机器人学最重要的研究领域之一。

工业机器人的轨迹规划问题经过半个世纪的研究,在基本轨迹规划方面已有了一大批成熟的技术,在最优轨迹规划方面也不乏可以指导工程实践的成果。轨迹规划主要分为笛卡儿空间中的轨迹规划和关节空间中轨迹规划。在众多学者对轨迹规划的研究过程中,关节空间的轨迹规划是大多数学者的研究方向,而笛卡儿空间的轨迹规划并不多见。机器人的轨迹规划大多在关节空间中进行,这种规划方式十分便捷;但是针对某些对空间轨迹要

求严格的工况下，则采用笛卡儿空间规划，这是因为虽然笛卡儿空间轨迹规划有着众多的优点，但是之前并没有用于笛卡儿空间坐标测量机器人末端执行器位置的传感器。

近年来，随着图像处理技术的发展以及多轴传感器定位技术的快速研发，使得笛卡儿空间坐标定位的问题得以解决。与此同时，工业加工作业中对机器人执行精度的要求越来越高，如高精度焊接等，所以对笛卡儿空间轨迹规划进行深入研究具有重要意义，例如目前离线编程技术可直接运用计算机仿真，提取工件的模型，生成作业需求的运动轨迹，这时笛卡儿空间轨迹规划就显得更加直观，易于运用。任务级机器人语言（如发出“抓住螺钉”指令）的发展趋势，也需要机器人拥有自动执行并规划任务的能力，能够从空间任务中直接提取出笛卡儿空间的运动路径，并规划运动状态。随着机器人逆解算法的进步和计算芯片性能的提高，笛卡儿空间规划也会越来越多地发挥其显示直观、实时性高的优点。建立在基本轨迹规划上的最优轨迹规划，则是以后轨迹规划的重要发展方向，找到各种工况下的最优轨迹也是学者们研究的热点，但是目前还没有一种通用性的综合优化方法适用于工业机器人。

目前智能轨迹规划算法是机器人研究的热点之一，基于最优化理论的发展，即搜寻最优轨迹的一种算法，如遗传算法、模糊算法、粒子群算法、模拟退火算法和人工神经网络等，并且很多实际应用与实验仿真也表明这些优化算法应用到机器人轨迹规划中，确实能够达到优化的目的，优化的结果包括降低能耗或者缩短时间，实现最优时间或者最优能量的轨迹优化设计，在保证能稳定准确的运行情况下，提高机器人的工作效率。

工业自动化应用中越来越高的精度要求和越来越复杂的工况，也给机器人轨迹规划提出了新的挑战，轨迹规划必须朝着高精度、高效率、模块化、自动化、智能化的方向进化。综上所述，展望工业机器人轨迹规划的最新发展趋势，包含基于机器视觉（Machine Vision）的实时自动轨迹规划、考虑实际工作模式的高可靠性轨迹规划、基于虚拟现实的机器人轨迹规划系统及轨迹规划算法的集成化、可视化、智能化等方面。

### 5.3.2　机器人关节轨迹的插值

机器人关节空间路径规划，是指给定关节角的约束条件，包括机器人的起点、终点或中间点的位置、速度、加速度等，生成机器人各关节变量的变化曲线的过程。当只给定起点、终点时刻的约束条件时，相应的机器人路径规划称为点到点路径规划；若要求机器人关节变量严格按照指定曲线运动时的规划，称为连续路径规划；当要求经过多个中间点，而对节点间的轨迹没有严格限制时的路径规划，称为多节点路径规划。

实际中，往往是采用介于点到点和连续路径规划之间的机器人多节点路径规划。可以采用三次多项式、过路径点的三次多项式插值、高阶多项式插值、用抛物线过渡的线性插值等插值函数进行关节空间的路径规划。

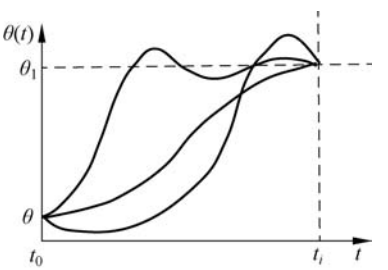


图 5.26　单个关节的不同轨迹曲线

#### 1. 三次多项式插值

考虑机器人末端在一定时间内从初始位置和方位移动到目标位置和方位的问题。利用逆运动学计算，可首先求出一组起始和终点的关节位置。现在的问题是求出一组通过起点和终点的光滑函数，满足这个条件的光滑函数可以有许多条，如图 5.26 所示。

为满足机器人关节运动速度的连续性要求，还有两个约束条件，即在起始点和终止点的关节速度要求。为了满足关节运动速度连续性的要求，起始点和终止点的关节速度可简单地设定为零。

$$\begin{cases}\theta(0)=\theta_0\\ \theta(t_f)=\theta_f\end{cases} \tag{5.19}$$

$$\begin{cases}\dot{\theta}(0)=0\\ \dot{\theta}(t_f)=0\end{cases} \tag{5.20}$$

上述四个边界约束条件式(5.19)和式(5.20)唯一地确定了一个三次多项式：

$$\theta(t)=a_0+a_1t+a_2t^2+a_3t^3 \tag{5.21}$$

机器人运动轨迹上的关节速度和加速度则为：

$$\begin{cases}\dot{\theta}(t)=a_1+2a_2t+3a_3t^2\\ \ddot{\theta}(t)=2a_2+6a_3t\end{cases} \tag{5.22}$$

为求得三次多项式的系数 $a_0$，$a_1$，$a_2$ 和 $a_3$，代以给定的约束条件，有方程组：

$$\begin{cases}\theta_0=a_0\\ \theta_f=a_0+a_1t_f+a_2t_f^2+a_3t_f^3\\ 0=a_1\\ 0=a_1+2a_2t_f+3a_3t_f^2\end{cases} \tag{5.23}$$

求解上述线性方程组可得：

$$\begin{cases}a_0=\theta_0\\ a_1=0\\ a_2=-\dfrac{3}{t_f^3}(\theta_f-\theta_0)\\ a_3=-\dfrac{2}{t_f^3}=(\theta_f-\theta_0)\end{cases} \tag{5.24}$$

对于起始速度及终止速度为零的关节运动，满足连续平稳运动要求的三次多项式插值函数为：

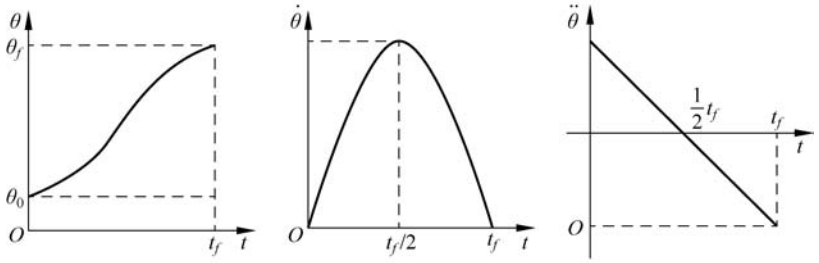
$$\theta(t)=\theta_0+\frac{3}{t_f^2}(\theta_f-\theta_0)t^2-\frac{2}{t_f^3}(\theta_f-\theta_0)t^3 \tag{5.25}$$

机器人关节角速度和角加速度的表达式为：

$$\begin{aligned}\dot{\theta}(t)&=\frac{6}{t_f^2}(\theta_f-\theta_0)t-\frac{6}{t_f^3}(\theta_f-\theta_0)t^2\\ \ddot{\theta}(t)&=\frac{6}{t_f^2}(\theta_f-\theta_0)t-\frac{12}{t_f^3}(\theta_f-\theta_0)t\end{aligned} \tag{5.26}$$

这里再次指出：这组解只适用于机器人关节起始、终止速度为零的运动情况。

三次多项式插值的机器人关节运动轨迹曲线如图 5.27 所示。由图可知，其速度曲线为抛物线，相应的加速度曲线为直线。




(a) 角位移　　(b) 角速度　　(c) 角加速度

图 5.27　三次多项式插值的关节运动轨迹

可以把所有路径点也看作是“起始点”或“终止点”，求解逆运动学，得到相应的关节矢量值。然后确定所要求的三次多项式插值函数，把路径点平滑地连接起来。但是，这些“起始点”和“终止点”的关节运动速度不再是零。

路径点上的关节速度可以根据需要设定，确定三次多项式的方法与前面所述的完全相同，只是速度约束条件变为：

$$\begin{cases}\dot{\theta}(0)=\dot{\theta}_0\\ \dot{\theta}(t_f)=\dot{\theta}_f\end{cases} \tag{5.27}$$

求得三次多项式的系数：

$$\begin{cases}a_0=\theta_0\\ a_1=\dot{\theta}_0\\ a_2=\dfrac{3}{t_f^2}(\theta_f-\theta_0)-\dfrac{2}{t_f}\dot{\theta}_0-\dfrac{1}{t_f}\dot{\theta}_f\\ a_3=-\dfrac{2}{t_f^3}(\theta_f-\theta_0)+\dfrac{1}{t_f}(\dot{\theta}_0+\dot{\theta}_f)\end{cases} \tag{5.28}$$

路径点上的关节速度可由以下三种方法规定：

(1) 在直角坐标空间或关节根据工具坐标系在直角坐标空间中的瞬时线速度和角速度来确定每个路径点的关节速度。

(2) 空间中采用适当的启发式方法，由控制系统自动地选择路径点的速度。

(3) 为了保证每个路径点上加速度的连续，由控制系统按此要求自动地选择路径点的速度。

**2. 过路径点的三次多项式插值**

对于这种情况，假如末端执行器在路径点停留，即各路径点上速度为 0，则轨迹规划可连续直接使用前面介绍的三次多项式插值方法；但若末端执行器只是经过，并不停留，就需要将前述方法推广。

对于机器人作业路径上的所有路径点可以用求解逆运动学的方法，先得到多组对应的关节空间路径点，进行轨迹规划时，把每个关节上相邻的两个路径点分别看作起始点和终止点，再确定相应的三次多项式插值函数，把路径点平滑连接起来。一般情况下，这些起始点和终止点的关节运动速度不再为零。

设路径点上的关节速度已知，在某段路径上，起始点为 $\theta_0$ 和 $\dot{\theta}_0$，终止点为 $\theta_f$ 和 $\dot{\theta}_f$，这时，确定三次多项式的方法与前面所述的全相同，只是速度约束条件变为：

$$\begin{cases}\dot{\theta}(0)=\dot{\theta}_0\\ \dot{\theta}(t_f)=\dot{\theta}_f\end{cases} \tag{5.29}$$

利用约束条件确定三次多项式系数，有下列方程组：

$$\begin{cases}\theta_0=a_0\\ \theta_f=a_0+a_1t_f+a_2t_f^2+a_3t_f^3\\ \dot{\theta}_0=a_1\\ \dot{\theta}_f=a_1+2a_2t_f+3a_3t_f^2\end{cases} \tag{5.30}$$

求解方程组可得：

$$\begin{cases}a_0=\theta_0\\ a_1=\dot{\theta}_0\\ a_2=\dfrac{3}{t_f^2}(\theta_f-\theta_0)-\dfrac{2}{t_f}\dot{\theta}_0-\dfrac{1}{t_f}\dot{\theta}_f\\ a_3=-\dfrac{2}{t_f^3}(\theta_f-\theta_0)+\dfrac{1}{t_f^2}(\dot{\theta}_0+\dot{\theta}_f)\end{cases} \tag{5.31}$$

实际上，由上式确定的三次多项式描述了起始点和终止点具有任意给定位置和速度的运动轨迹。剩下的问题就是如何确定路径点上的关节速度，可由以下 3 种方法规定：

(1) 根据工具坐标系在直角坐标空间中的瞬时线速度和角速度确定每个路径点的关节速度。

该方法利用操作臂在此路径点上的逆雅可比，把该点的直角坐标速度“映射”为所要求的关节速度。当然，如果操作臂的某个路径点是奇异点，这时就不能任意设置速度值。按照该方法生成的轨迹虽然能满足用户设置速度的需要，但是逐点设置速度毕竟要耗费很大的工作量。

(2) 在直角坐标空间或关节空间中采用适当的启发式方法，由控制系统自动地选择路径点的速度。

图 5.28 表示一种启发式选择路径点速度的方式。图中 $\theta_0$ 为起始点；$\theta_D$ 为终止点，$\theta_A$，$\theta_B$ 和 $\theta_C$ 是路径点，用细实线表示过路径点时的关节运动速度。这里所用的启发式信息从概念到计算方法都很简单，即假设用直线段把这些路径点依次连接起来，如果相邻线段的斜率在路径点处改变符号，则把速度选定为零；如果相邻线段不改变符号，则选取路径点两侧的线段斜率的平均值作为该点的速度。因此，根据规定的路径点，系统就能够按此规则自动生成相应的路径点速度。

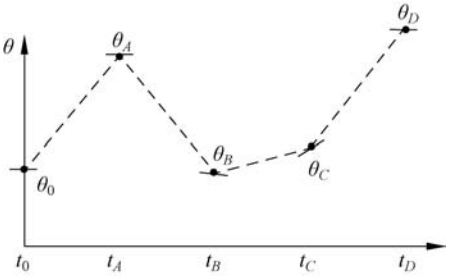



图 5.28 路径点上的速度自动生成

(3) 为了保证每个路径点上的加速度连续，由控制系统按此要求自动地选择路径点的速度。

为了保证路径点处的加速度连续，可设法用两条三次曲线在路径点处按一定规则连接起来，拼凑成所要求的轨迹。其约束条件是：连接处不仅速度连续，而且加速度也连续。

设所经过的路径点处的关节角度为 $\theta_v$，与该点相邻的前后两点的关节角分别为 $\theta_0$ 和 $\theta_g$。设其路径点处的关节加速度连续。如果路径点用三次多项式连接，试确定多项式的所有系数。该机器人路径可分为 $\theta_0 \sim \theta_v$ 段及 $\theta_v \sim \theta_g$ 段两段，可通过由两个三次多项式组成的样条函数连接。设 $\theta_0 \sim \theta_v$ 的三次多项式插值函数为：

$$\theta(t) = a_{10} + a_{11}t + a_{12}t^2 + a_{13}t^3 \tag{5.32}$$

$\theta_v \sim \theta_g$ 的插值三次多项式为：

$$\theta(t) = a_{20} + a_{21}t + a_{22}t^2 + a_{23}t^3 \tag{5.33}$$

上述两个三次多项式的时间区间分别为$[0, t_{f1}]$和$[0, t_{f2}]$。对这两个多项式的约束是：

$$\begin{cases} \theta_0 = a_{10} \\ \theta_v = a_{10} + a_{11}t_{f1} + a_{12}t_{f1}^2 + a_{13}f_{f1}^3 \\ \theta_v = a_{20} \\ \theta_k = a_{20} + a_{21}t_{f2} + a_{22}t_{f2}^2 + a_{23}t_{f2}^3 \\ 0 = a_{11} \\ 0 = a_{21} + 2a_{22}t_{f2} + 3a_{23}t_{f2}^2 \\ a_{11} + 2a_{12}t_{f1} + 3a_{13}t_{f1}^2 = a_{21} \\ 2a_{12} + 6a_{13}t_{f1} = 2a_{22} \end{cases} \tag{5.34}$$

以上约束组成了含有 8 个未知数的 8 个线性方程。对于 $t_{f1} = t_{f2} = t_f$ 的情况，这个方程组的解为：

$$\begin{cases} a_{10} = \theta_0 \\ a_{11} = 0 \\ a_{12} = \dfrac{12\theta_v - 3\theta_g - 9\theta_0}{4t_f^2} \\ a_{13} = \dfrac{-8\theta_v + 3\theta_g + 5\theta_0}{4t_f^2} \\ a_{20} = \theta_v \\ a_{21} = \dfrac{3\theta_g - 3\theta_0}{4t_f} \\ a_{22} = \dfrac{-12\theta_v + 6\theta_g + 6\theta_0}{4t_f^2} \\ a_{23} = \dfrac{8\theta_v - 5\theta_g - 3\theta_0}{4t_f^3} \end{cases} \tag{5.35}$$

一般情况下，一个完整的机器人轨迹由多个三次多项式表示，约束条件(包括路径点处的关节加速度连续)构成的方程组。

**3. 高阶多项式插值**

如果对于运动轨迹的要求更为严格，约束条件增多，那么三次多项式就不能满足需要，

必须用更高阶的多项式对运动轨迹的路径段进行插值。例如，对某段路径的起始点和终止点都规定了关节的位置、速度和加速度，则要用一个五次多项式进行插值，即

$$\theta(t)=a_0+a_1t+a_2t^2+a_3t^3+a_4t^4+a_5t^5 \tag{5.36}$$

多项式的系数 $a_0,a_1,\cdots,a_5$ 必须满足 6 个约束条件：

$$\begin{cases}\theta_0=a_0\\ \theta_f=a_0+a_1t_f+a_2t_f^2+a_3t_f^3+a_4t_f^4+a_5t_f^5\\ \dot{\theta}_0=a_1\\ \dot{\theta}_f=a_1+2a_2t_f+3a_3t_f^2+4a_4t_f^3+5a_5t_f^4\\ \ddot{\theta}_0=2a_2\\ \ddot{\theta}_f=2a_2+6a_3t_f+12a_4t_f^2+20a_5t_f^3\end{cases} \tag{5.37}$$

这个线性方程组含有 6 个未知数和 6 个方程，其解为：

$$\begin{cases}a_0=\theta_0\\ a_1=\dot{\theta}_0\\ a_2=\dfrac{\ddot{\theta}_0}{2}\\ a_3=\dfrac{20\theta_f-20\theta_0-(8\dot{\theta}_f+12\dot{\theta}_0)t_f-(3\ddot{\theta}_0-\ddot{\theta}_f)t_f^2}{2t_f^3}\\ a_4=\dfrac{30\theta_0-30\theta_f+(14\dot{\theta}_f+16\dot{\theta}_0)t_f+(3\ddot{\theta}_0-2\ddot{\theta}_f)t_t^2}{2t_f^4}\\ a_5=\dfrac{12\theta_f-12\theta_0-(6\dot{\theta}_f+6\dot{\theta}_0)t_f+(\ddot{\theta}_0-\ddot{\theta}_f)t_f^2}{2t_f^5}\end{cases} \tag{5.38}$$

**4. 用抛物线过渡的线性插值**

对于给定了起始点和终止点的机器人关节空间轨迹规划，似乎选择线性函数最为简单，但是最纯线性插值往往会导致在起始点和终止点关节运动速度的不连续，并且加速度无限大，在两端点会造成刚性冲击。

为了改进，在线性插值两端点的邻域内设置一段抛物线形缓冲区段，如图 5.29 所示。由于抛物线函数对于时间的二阶导数为常数，因此在端点的速度平滑过渡，从而使整个轨迹上的位置和速度连续，如图 5.30 所示。

对于这种路径规划存在有多个解，其轨迹不唯一，如图 5.31 所示。但是，每条路径都对称于时间中点 $t_h$ 和位置中点 $\theta_h$。

为了保证机器人路径轨迹的连续、光滑，即要求抛物线轨迹的终点速度必须等于线性段的速度，故有下列关系：

$$\dot{\theta}_{t_b}=\frac{\theta_h-\theta_b}{t_h-t_b} \tag{5.39}$$

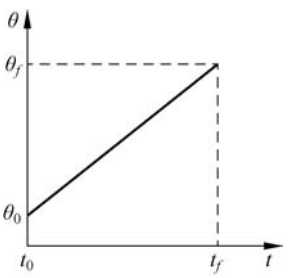



图 5.29 两点间的线性插值轨迹

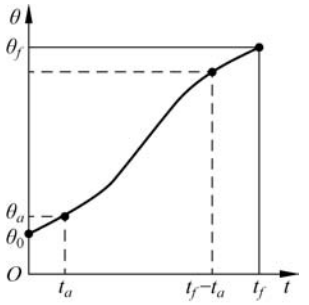



图 5.30　带有抛物线过渡域的线性轨迹

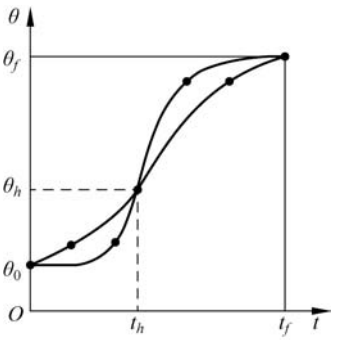



图 5.31　轨迹的多解性与对称性

式中，$\theta_b$ 为过渡域终点 $t_b$ 处的关节角度。用 $\ddot{\theta}$ 表示过渡域内的加速度，$\theta_b$ 的值可按下式解得：

$$\theta_b = \theta_0 + \frac{1}{2}\ddot{\theta}t_b^2 \tag{5.40}$$

令 $t = 2t_h$，可得：

$$\ddot{\theta}t_b^2 - \ddot{\theta}tt_b + (\theta_f - \theta_0) = 0 \tag{5.41}$$

式中，$t$ 为所要求的运动持续时间。

一般情况下，$\theta_0$、$\theta_f$、$t_f$ 是已知条件，这样，根据式(5.41)可以选择相应的 $\ddot{\theta}$ 和 $t_b$，得到相应的轨迹。通常的做法是先选定加速度 $\ddot{\theta}$ 的值，然后按式(5.42)求出相应的 $t_b$：

$$t_b = \frac{t}{2} - \frac{\sqrt{\ddot{\theta}^2 t^2 - 4\ddot{\theta}(\theta_f - \theta_0)}}{2\ddot{\theta}} \tag{5.42}$$

由式(5.42)可知，为保证 $t_b$ 有解，过渡域加速度值 $\ddot{\theta}$ 必须选得足够大，即

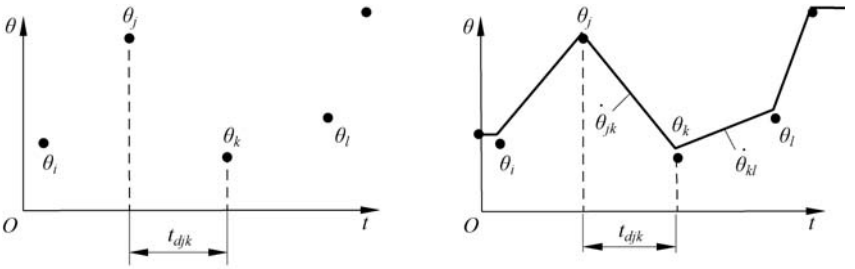
$$\ddot{\theta} \geqslant \frac{4(\theta_f - \theta_0)}{t^2} \tag{5.43}$$

当式(5.37)中的等号成立时，机器人轨迹线性段的长度缩减为零，整个轨迹由两个过渡域组成，这两个过渡域在衔接处的斜率(关节速度)相等；加速度 $\ddot{\theta}$ 的取值越大，过渡域的长度就越短，若加速度趋于无穷大，轨迹又复归到简单的线性插值情况。

**5. 过路径点的用抛物线过渡的线性插值**

简单的线性插值会使得机器人在某一路径点处的关节运行速度产生跳变，导致加速度急剧增加，甚至接近于无穷大，如图 5.32 所示。为了避免这种情况的发生，使用线性插值法时，需要在每个路径点附近加入一截使用抛物线拟合的过渡段，使生成的轨迹具有连续的角度值变化和均匀的速度。至于抛物线对时间的二阶导数，因为它是常数，所以对应的路径段具有恒定的加速度，使得该点处的轨迹运动速度平稳而不引起阶跃变化。通过抛物线的拟合，即可得到位移和速度连续的整个轨迹。

机器人某一个关节的一组关节路径点之间使用线性样条相连，而各线性样条与路径点相连的两端都使用了一段抛物线连接，具体如图 5.33 所示。线性样条和抛物线拟合形成的




(a) 关节路径点　　(b) 线性插值轨迹

图 5.32　机器人某一关节的轨迹

轨迹称为带有抛物线拟合的线性轨迹。在图中，关节路径点多余两个，所以有多条路径段，机器人移动到第一个路径段的终点，还会向下一点移动，下一点可以是目标点或者另一个中间点。如前所述，采用带有抛物线拟合的分段线性运动轨迹生成方法，以避免运动速度的不连续。

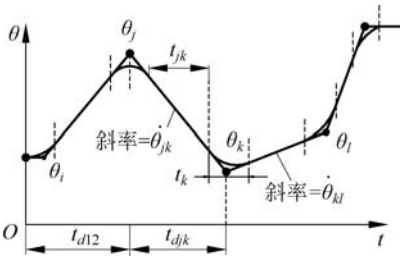



图 5.33　多段带有抛物线过渡的线性插值轨迹

在各路径段进行抛物线拟合时，可利用各个路径点的边界条件得到相应的抛物线拟合函数的系数。例如，如果已知机器人关节某一起始路径点的运动速度，因为下一个中间点处于该点的路径段中，该中间点要有连续的速度和位移，根据这一边界条件，可以计算下一条路径段的系数，以此类推，直到计算出全部路径段并到达目标点。当然，我们还需要根据给定的关节运动速度对各个路径段计算新的时间 $t$，并且还需检查加速度的值是否大于极限值。

在这里将使用以下符号：用 $j$，$k$ 和 $l$ 表示三个临近路径点，它们都表示关节在某一时刻的转动角度值。位于路径点 $k$ 处的拟合区段的运行时间间隔为 $t_k$，位于点 $j$ 和 $k$ 之间的直线段的时间间隔为 $t_{jk}$，此两点间包括拟合段的总时间间隔为 $t_{djk}$。此外，$j$、$k$ 间直线段的速度为 $\dot{\theta}_{jk}$，而在点 $j$ 处拟合区段的加速度为 $\dot{\theta}_j$。

与只包含首尾两个路径点的单一路径段的情形相似，通过多个中间路径点的带有抛物线拟合的线性插值轨迹可能存在许多不同的结果，而不同结果的产生主要由各个拟合区段的加速度值决定。已知任意一个路径点的位置 $\theta_k$、期望的时间间隔 $t_{djk}$ 以及此路径点处的加速度值为 $|\ddot{\theta}_k|$，那么可以计算出该点相应拟合区段的时间间隔。对于那些不是处于轨迹首或尾的路径段（$j,k\neq1,2$；$k\neq n,n-1$），可直接使用下列公式计算：

$$\begin{cases}\dot{\theta}_{jk}=\dfrac{\theta_k-\theta_j}{t_{djk}}\\ \ddot{\theta}_k=\operatorname{sgn}(\dot{\theta}_{kl}-\dot{\theta}_{jk})\,|\,\ddot{\theta}_k\,|\\ t_k=\dfrac{\dot{\theta}_{kl}-\dot{\theta}_{jk}}{\ddot{\theta}_k}\\ t_{jk}=t_{djk}-\dfrac{1}{2}t_j-\dfrac{1}{2}t_k\end{cases}\tag{5.44}$$

因为一条机器人的路径段包括轨迹首尾端部拟合区段的时间，而首尾端部路径段为拟合区段的一部分，其处理情况与中间路径段有所差异。为求取首部路径段的持续时间，可使此处直线区段的速度相等，即

$$\frac{\theta_2-\theta_1}{t_{d12}-\frac{1}{2}t_1}=\ddot{\theta}_1 t_1 \tag{5.45}$$

用式(5.45)算出起始点拟合区段的持续时间 $t_1$ 之后，进而求出 $\dot{\theta}_{12}$ 和 $t_{12}$：

$$\begin{cases}\ddot{\theta}_1=\operatorname{sgn}(\dot{\theta}_2-\dot{\theta}_1)\mid\ddot{\theta}_1\mid\\ t_1=t_{d12}-\sqrt{t_{d12}^2-\dfrac{2(\theta_2-\theta_1)}{\ddot{\theta}_1}}\\ \dot{\theta}_{12}=\dfrac{\theta_2-\theta_1}{t_{d12}-t_1-\dfrac{1}{2}t_2}\\ t_{12}=t_{d12}-t_1-\dfrac{1}{2}t_2\end{cases} \tag{5.46}$$

对于最后一个路径段，路径点 $n-1$ 与终止点 $n$ 之间的参数与第一个路径段相似，即

$$\frac{\theta_{m-1}-\theta_m}{t_{d(n-1)n}-\frac{1}{2}t_n}=\ddot{\theta}_n t_n \tag{5.47}$$

根据式(5.47)便可求出：

$$\begin{cases}\ddot{\theta}_n=\operatorname{sgn}(\dot{\theta}_{n-1}-\dot{\theta}_n)\mid\ddot{\theta}_n\mid\\ t_n=t_{d(n-1)n}-\sqrt{t_{d(n-1)n}^2+\dfrac{2(\theta_m-\theta_{n-1})}{\ddot{\theta}_n}}\\ \dot{\theta}_{(n-1)n}=\dfrac{\theta_n-\theta_{n-1}}{t_{d(n-1)n}-\dfrac{1}{2}t_n}\\ t_{(n-1)n}=t_{d(n-1)_n}-t_n-\dfrac{1}{2}t_{n-1}\end{cases} \tag{5.48}$$

可使用式(5.44)～式(5.48)的计算轨迹中各拟合段的运行时间以及相应的速度。通常情况下，需要指定的是各个路径点和每个路径段的持续时间，此时，关节轨迹规划将使用机器人的隐式加速度值。有些时候，为了简化计算，可直接使用隐式加速度值求得路径段的持续时间。对于轨迹的各拟合段部分，为了使路径段有较长的直线区段，应选取较大的加速度值以压缩拟合区段。需要注意的是，带有抛物线拟合的线性插值轨迹通常不通过各个中间路径点，除了机器人需要在某个路径点处暂停。如果选定了足够大的加速度值，那么实际生成的轨迹将非常接近提供的已知路径点。如果机器人需要通过一个路径点，可行的方法是将轨迹划分成两个段，该点可以用来作为上一段轨迹的目标点以及下一段轨迹的出发点，而运行的机器人会在此处停止并运行下一段轨迹。

### 5.3.3 笛卡儿空间规划方法

在笛卡儿空间中，应用机器人轨迹规划的插补算法可求得中间点(插补点)的坐标，把这些插补点的位置和姿态转换成对应的机器人关节角，通过机器人运动学逆解运算，然后沿着规划的轨迹运动使得这些关节角控制机器人末端执行器。一个机器人在笛卡儿空间中的轨迹规划位姿控制如图 5.34 所示。

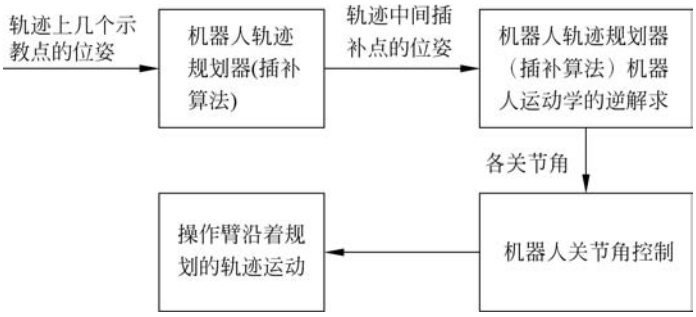



图 5.34 机器人在笛卡儿空间中轨迹规划的控制过程

笛卡儿空间机器人轨迹规划系统中两个最基本的轨迹规划方法，即空间直线和圆弧的轨迹规划，因为空间很多曲线都可以分割为多段直线或圆弧。而在很多情况下会出现空间多段直线连接或者直线与圆弧连接，这就不可避免地会碰到连接处尖角的问题。为能使运动轨迹平滑，本章采用圆弧过渡进行平滑尖角，接下来将介绍空间连续直线和直线—圆弧轨迹规划。

此外，还有一些自由型曲线，只给出一系列路径点，利用一条平滑曲线将这些路径点光滑地连接起来。如果由直线或圆弧分割这些自由型曲线将难以保证精度，而 B 样条曲线能够很好地逼近这些自由型曲线。

**1. 直线轨迹规划**

直线的轨迹规划是已知直线始末两点的位置和姿态，求直线轨迹上的插补点的位置和姿态。各个插补点的位置和姿态可用以下的公式求出：

$$\begin{cases} x = x_1 + \lambda \Delta x \\ y = y_1 + \lambda \Delta y \\ z = z_1 + \lambda \Delta z \\ \alpha = \alpha_1 + \lambda \Delta \alpha \\ \beta = \beta_1 + \lambda \Delta \beta \\ \gamma = \gamma_1 + \lambda \Delta \gamma \end{cases} \tag{5.49}$$

式(5.49)中，$(x_1, y_1, z_1)$、$(\alpha_1, \beta_1, \gamma_1)$分别为开始点的位置 RPY 变换姿态角，$(x_1, y_1, z_1)$、$(\alpha_1, \beta_1, \gamma_1)$分别为插补点的位置和 RPY 变换姿态角，$\lambda$ 为归一化因子，$(\Delta x, \Delta y, \Delta z)$、$(\Delta\alpha, \Delta\beta, \Delta\gamma)$为位置和姿态角的增量，其求解如下：

$$\begin{cases}\Delta x = x_2 - x_1 \\ \Delta y = y_2 - y_1 \\ \Delta z = z_2 - z_1 \\ \Delta\alpha = \alpha_2 - \alpha_1 \\ \Delta\beta = \beta_2 - \beta_1 \\ \Delta\gamma = \gamma_2 - \gamma_1 \end{cases} \tag{5.50}$$

式(5.50)中$(x_2, y_2, z_2)$、$(\alpha_2, \beta_2, \gamma_2)$分别为结束点的位置和姿态角。

采用抛物线过渡的 $\lambda$ 归一化因子线性函数，这种方式不仅简单易解，同时也能保证整条轨迹上的位移和速度的连续性要求。抛物线过渡的线性函数对两点的位姿使用线性插值时，两点的领域内增加一段抛物线的缓冲区段。由于相应区段内有恒定不变的加速度且抛物线时间的二阶导数为常数，轨迹的过渡平滑，于是整条轨迹上的位移和速度都具有连续性。为了能够构造出这段轨迹的运动曲线，假设有相等两端的抛物线运动时间，在这两个过渡区域内采用相同的恒加速度值，于是 $\lambda$ 归一化因子可按如下的方式求得。

设抛物线过渡的线性函数的直线段速度 $v$，抛物线段的加速度为 $a$。那么抛物线段的运动时间和位移分别为：

$$T_b = \frac{v}{a} \tag{5.51}$$

$$L_b = \frac{1}{2} a T_b^2 \tag{5.52}$$

直线运动总的位移和时间分别表示为：

$$\begin{aligned} L &= \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \\ T &= 2T_b + \frac{L - 2L_b}{v} \end{aligned} \tag{5.53}$$

抛物线段位移、时间、加速度分别归一化处理：

$$\begin{aligned} L_{b\lambda} &= \frac{L_b}{L} \\ T_{b\lambda} &= \frac{T_b}{T} \\ \alpha\lambda &= \frac{2L_{b\lambda}}{T_{b\lambda}^2} \end{aligned} \tag{5.54}$$

则，可得到 $\lambda$ 的计算公式：

$$\lambda = \begin{cases} \dfrac{1}{2} a_\lambda t^2 & (0 \leqslant t \leqslant T_{b\lambda}) \\ \dfrac{1}{2} a_\lambda T_{b\lambda}^2 + a_\lambda T_{b\lambda}(t - T_{b\lambda}) & (T_{b\lambda} < t < 1 - T_{b\lambda}) \\ \dfrac{1}{2} a_\lambda T_{b\lambda}^2 + a_\lambda T_{b\lambda}(t - T_{b\lambda}) - \dfrac{1}{2} a_\lambda (t + T_{b\lambda} - 1)^2 & (1 - T_{b\lambda} < t < 1) \end{cases} \tag{5.55}$$

式(5.55)中，$t = (i/N)$，$i = 0, 1, 2, \cdots, N$，$0 \leqslant \lambda \leqslant 1$，当 $\lambda = 0$ 时，对应于起点，当 $\lambda = 1$ 时，对应于终点。$\lambda$ 是分段离散函数关于时间 $t$ 来说，$\lambda$ 和 $t$ 均为无量纲，对称的抛物线线段

$0 \leqslant t \leqslant T_{b\lambda}$ 和 $1-T_{b\lambda} \leqslant t \leqslant 1$。图 5.35 为 $\lambda$ 和 $\lambda'$ 随离散时间 $t$ 变化的图形。

设抛物线过渡的线性函数的直线段速度为 $v$，抛物线段的加速度为 $a$，而不是给出运动的总时间，主要是考虑机器人的运动速度和加速度都有约束，并且不同的机器人约束也不一样，因此这里把速度和加速度作为输入量。

**2. 平面圆弧插补**

平面圆弧为与基坐标系三个平面中的任一个重合的圆弧运动轨迹的平面，如 $XOY$ 内的圆弧：平面内有三个不共线的点 $P_1, P_2, P_3$ 与它们相应的机器人末端的位姿，如图 5.36 所示。假设 $v$ 为沿圆弧运动的速度；$T_s$ 为插补周期。由 $P_1, P_2, P_3$ 的坐标，可求出圆弧的半径 $R$，以及总的圆心角 $\theta$，根据圆的公式可得：

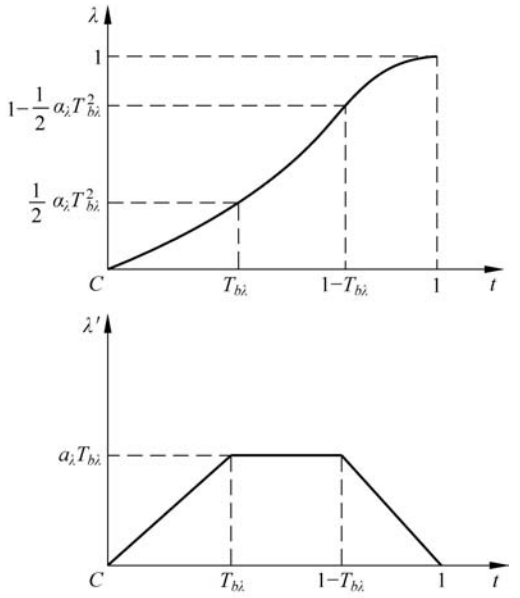



图 5.35 为 $\lambda$ 和 $\lambda'$ 随离散时间 $t$ 变化的图形

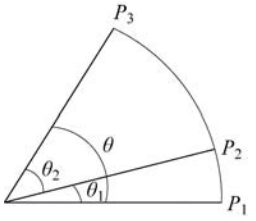



图 5.36 平面圆弧插补

$$\begin{cases} \theta_1 = \arccos \dfrac{[(x_3 - x_2) + (y_3 - y_2) - 2R^2]}{2R^2} \\ \theta_2 = \arccos \dfrac{[(x_2 - x_1) + (y_2 - y_1) - 2R^2]}{2R^2} \end{cases} \tag{5.56}$$

根据圆弧上的数学关系可计算各个插补点的坐标值。同直线插补一样，判断插补过程是匀速、匀加速和匀减速阶段，求出总时间 $t = t_1 + t_2 + t_3$。求解插补过程当中的中间点数：$N = t/T_s$。假设插补过程都是匀速阶段，那么插补周期 $T_s$ 内的角位移量 $\Delta\theta = vT_s/R$ 能计算出圆弧上任意插补点 $P_{i+1}$ 的坐标：

$$\begin{cases} x_{i+1} = R\cos(\theta_i + \Delta\theta) = x_i\cos\theta - y_i\sin\Delta \\ y_{i+1} = R\sin(\theta_i + \Delta\theta) = y_i\cos\theta + x_i\sin\Delta \\ \theta_{i+1} = \theta_i + \Delta\theta \end{cases} \tag{5.57}$$

**3. 空间圆弧插补**

给定空间中不共线的三个点 $p_1(x_1, y_1, z_1)$，$p_2(x_2, y_2, z_2)$ 和 $p_3(x_3, y_3, z_3)$，如

图 5.37 所示。

为了求得圆弧上各点的坐标值，必须先求出其圆心的坐标及半径 $R$。由三个点能够决定唯一的平面，那么 $P_1$，$P_2$，$P_3$ 所描述的平面 $M$ 可表示为：

$$\begin{vmatrix} x-x_3 & y-y_3 & z-z_3 \\ x_1-x_3 & y_1-y_3 & z_1-z_3 \\ x_2-x_3 & y_2-y_3 & z_2-z_3 \end{vmatrix} \tag{5.58}$$

$T$ 是过直线 $P_1P_2$ 的中点并且还垂直 $P_1P_2$ 的平面，那么 $T$ 上的任意直线都垂直 $P_1P_2$，于是平面 $T$ 的方程可表示为：

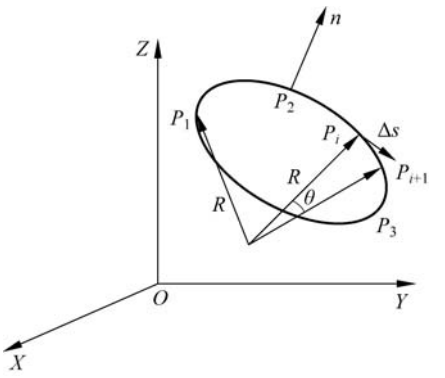


图 5.37　空间圆弧插补算法的原理图

$$\left(x-\frac{x_1+x_2}{2}\right)(x_2-x_1)+\left(y-\frac{y_1+y_2}{2}\right)(y_2-y_1)+\left(z-\frac{z_1+z_2}{2}\right)(z_2-z_1)=0 \tag{5.59}$$

同理，过 $P_2P_3$ 中点并垂直于 $P_2P_3$ 的平面 $S$ 的方程为：

$$\left(x-\frac{x_2+x_3}{2}\right)(x_3-x_2)+\left(y-\frac{y_2+y_3}{2}\right)(y_3-y_2)+\left(z-\frac{z_2+z_3}{2}\right)(z_3-z_2)=0 \tag{5.60}$$

所以，平面 $M$，$T$ 和 $S$ 的交点便是要求的圆心 $O$，确定 $O$ 的坐标后可求圆弧半径：

$$R=\sqrt{(x_1-x_0)^2+(y_1-y_0)^2+(z_1-z_0)^2} \tag{5.61}$$

一个圆弧的走向可以根据设定的起点、终点与一个中间点来确定，可取 $\boldsymbol{n}=\overrightarrow{AB}\times\overrightarrow{BC}$，则从 $\boldsymbol{n}$ 的正向看去，圆弧的走向为逆时针。

设 $\boldsymbol{n}=\overrightarrow{AB}\times\overrightarrow{BC}=u\boldsymbol{i}+v\boldsymbol{j}+w\boldsymbol{k}$，则：

$$\begin{cases} u=(y_2-y_1)(z_3-z_2)-(z_2-z_1)(y_3-y_2) \\ v=(z_2-z_1)(x_3-x_2)-(x_2-x_1)(z_3-z_2) \\ w=(x_2-x_1)(y_3-y_2)-(y_2-y_1)(x_3-x_2) \end{cases} \tag{5.62}$$

如图 5.33 所示，圆弧上的任一点 $p_i(x_i,y_i,z_i)$ 处在沿前进方向的切向量为：

$$m_i\boldsymbol{i}+n_i\boldsymbol{j}+l_i\boldsymbol{k}=\boldsymbol{n}\times\boldsymbol{OP}_i=\begin{vmatrix} \boldsymbol{i} & \boldsymbol{j} & \boldsymbol{k} \\ u & v & w \\ x_i-x_0 & y_i-y_0 & z_i-z_0 \end{vmatrix} \tag{5.63}$$

可得：

$$\begin{cases} m_i=v(z_i-z_0)-w(y_i-y_0) \\ n_i=w(x_i-x_0)-u(z_i-z_0) \\ l_i=u(y_i-y_0)-v(x_i-x_0) \end{cases} \tag{5.64}$$

经过一个插补周期之后，机器人的末端从 $p_i(x_i,y_i,z_i)$ 沿圆弧切向运动距离 $\Delta s$ 到 $p_{i+1}(x_{i+1},y_{i+1},z_{i+1})$，则有：

$$\begin{cases} x'_{i+1}=x_i+\Delta x'_i=x_i+Em_i \\ y'_{i+1}=y_i+\Delta y'_i=y_i+Em_i \\ z'_{i+1}=z_i+\Delta z'_i=z_i+Em_i \end{cases} \tag{5.65}$$

其中，$E=\dfrac{\Delta s}{\sqrt{m_i^2+n_i^2+l_i^2}}$，并能证明 $E=\dfrac{\Delta s}{R\sqrt{u^2+v^2+w^2}}$ 是常量。

从图 5.34 可以看出，圆弧上并没有点 $p'_{i+1}(x'_{i+1},y'_{i+1},z'_{i+1})$，为了让其能落在此空间的圆弧上，要改进式(5.65)。连接 $OP'_{i+1}$ 交圆弧于点并用 $P_{i+1}$ 代替 $P'_{i+1}$，这样就可以实现插补点落在圆弧上。在直角三角形 $OP_iP'_{i+1}$ 中，有 $|OP'_{i+1}|^2=|OP_i|^2+|P_iP'_{i+1}|^2$，即 $(R+\Delta R)^2=R^2+\Delta s^2$，则有：

$$\begin{cases} x_{i+1}=x_0+\dfrac{R(x'_{i+1}-x_0)}{\sqrt{R^2+\Delta s^2}} \\ y_{i+1}=y_0+\dfrac{R(y'_{i+1}-y_0)}{\sqrt{R^2+\Delta s^2}} \\ z_{i+1}=z_0+\dfrac{R(z'_{i+1}-z_0)}{\sqrt{R^2+\Delta s^2}} \end{cases} \tag{5.66}$$

令 $G=\dfrac{R}{\sqrt{R^2+\Delta s^2}}$，并将式(5.65)代入式(5.66)中，可求出插补地推公式为：

$$\begin{cases} x_{i+1}=x_0+G(x_i+Em_i-x_0) \\ y_{i+1}=y_0+G(y_i+Em_i-y_0), \quad (0\leqslant i\leqslant N-1) \\ z_{i+1}=z_0+G(z_i+Em_i-z_0) \end{cases} \tag{5.67}$$

### 5.3.4 机器人轨迹的实时生成

上面所述的计算结果即构成了机器人的轨迹规划。运行中的轨迹实时生成是指由这些数据，以轨迹更新的速率不断产生 $\theta$，$\dot{\theta}$ 和 $\ddot{\theta}$ 所表示的轨迹，并将此信息送至操作臂的控制系统。

**1. 机器人关节空间轨迹的生成**

在使用带抛物线过渡的线性轨迹生成方法时，需要判断当前轨迹位置是处于拟合区段还是直线区段，在直线区段，对每个关节的轨迹计算如下：

$$\begin{cases} \theta=\theta_j+\dot{\theta}_{jk}t \\ \dot{\theta}=\dot{\theta}_{jk} \\ \ddot{\theta}=0 \end{cases} \tag{5.68}$$

式中，$t$ 是自第 $j$ 个路径点算起的时间，$\dot{\theta}_{jk}$ 的值在路径规划时按式(5.68)计算。

在拟合区段：令 $t_{inb}=t-\left(\dfrac{1}{2}t_j+t_{jk}\right)$，则对各关节的轨迹计算如下：

$$\begin{cases} \theta=\theta_j+\dot{\theta}_{jk}(t-t_{inb})+\dfrac{1}{2}\ddot{\theta}_k t_{inb}^2 \\ \dot{\theta}=\dot{\theta}_{jk}+\ddot{\theta}_k t_{inb} \\ \ddot{\theta}=\ddot{\theta}_k \end{cases} \tag{5.69}$$

其中 $\dot{\theta}_{jk}$，$\ddot{\theta}_k$，$t_j$ 和 $t_{jk}$ 在轨迹规划时，已由上式当进入新的直线区段时，重新把 $t$ 置成 $1/2t_k$，

利用该路径段的数据，继续生成轨迹。

综上所述，机器人的关节轨迹联合表达式为：

$$\begin{cases}\theta=\lambda(\theta_j+\dot{\theta}_{jk}t)+(1-\lambda)[\theta_j+\dot{\theta}_{jk}(t-t_{inb})+\dfrac{1}{2}\ddot{\theta}_k t_{inb}^2] \\ \dot{\theta}=\lambda\dot{\theta}_{jk}+(1-\lambda)(\dot{\theta}_{jk}+\ddot{\theta}_k t_{inb}) \\ \ddot{\theta}=(1-\lambda)\ddot{\theta}_k\end{cases} \tag{5.70}$$

其中，当轨迹曲线处于直线区段时，$\lambda$ 取 1；当轨迹曲线处于直线区段时，$\lambda$ 取 0。

**2. 笛卡儿空间轨迹的生成**

在笛卡儿空间中实时地产生运动轨迹，则需要通过规划的方式实现，从而得到轨迹参数。由于是机器人控制器完成轨迹生成，因此实际上第一步是先实时计算出离散的轨迹点，然后再将这些轨迹点按照一定的采样速率经逆运动学计算转换到关节空间中。

在笛卡儿空间中，由 3 个量$(x,y,z)$表示路径点位置，另外 3 个量$(k_x,k_y,k_z)$通过矢量表示姿态。当线性函数插值采用抛物线连接时，则分别可对以上的 6 个变量，通过如式(5.71)～式(5.73)所示的计算公式计算。以其中的一个分量 $x$ 为例，计算公式如下：

$$\begin{cases}x(t)=x_1+\dfrac{1}{2}\ddot{x}_1 t^2 \\ \dot{x}=\ddot{x}_1 t \\ \ddot{x}(t)=\ddot{x}_1\end{cases} \qquad (0\leqslant t\leqslant t_1) \tag{5.71}$$

$$\begin{cases}x(t)=x_1+\dot{x}_{i(i+1)}\left(t+\dfrac{t_i}{2}\right) \\ \dot{x}(t)=\dot{x}_{i(i+1)} \\ \dot{x}(t)=0\end{cases} \qquad (0\leqslant t\leqslant t_{i(i+1)},i=1,2,\cdots,n-1) \tag{5.72}$$

$$\begin{cases}x(t)=x_{i-1}+\dot{x}_{i(i-1)}\left(t+\dfrac{t_{i-1}}{2}+t_{i(i-1)}+\dfrac{1}{2}\ddot{x}_i t_2\right) \\ \dot{x}(t)=\dot{x}_{i(i-1)} \\ \ddot{x}(t)=\ddot{x}_i\end{cases} \qquad (0\leqslant t\leqslant t_i,i=2,3,\cdots,n) \tag{5.73}$$

通过上面相同的公式将其他各分量也进行计算，应该注意到机器人末端的速度或加速度并不与姿态 3 个量的一阶和二阶导数相等。当求得了直角坐标空间的轨迹 $\bar{s}$，$\dot{s}$，$\ddot{s}$ 后，再利用求解逆运动学的方法，求得关节空间中的轨迹 $q$，$\dot{q}$ 和 $\ddot{q}$。由于计算 $\dot{q}$ 和 $\ddot{q}$ 的逆运动学需要较大工作量的计算，因此当进行实际计算的时候，只需要根据 $\bar{s}$ 的逆解计算出 $q$ 值，然后再用数值微分的方法计算出 $\dot{q}$ 和 $\ddot{q}$ 的值即可。

$$\begin{aligned}\dot{q}(t)&=\frac{q(t)-q(t-t)}{t} \\ \ddot{q}(t)&=\frac{\dot{q}(t)-\dot{q}(t-t)}{t}\end{aligned} \tag{5.74}$$

采样计算的周期表示为 $t$。当使用这种方法在直角坐标空间中计算 $\dot{q}$ 和 $\ddot{q}$ 时，可不必计算出各个分量的一阶以及二阶导数，如此可缩短一部分计算的工作量。

当采用圆弧插值时，直接先计算出 $x(t)$、$y(t)$和 $z(t)$，并通过类似的公式计算出 3 个姿态量。然后由 $\bar{s}(t)$逆解计算出 $q(t)$，再利用式(5.74)的数值微分公式算出 $\dot{q}(t)$和 $\ddot{q}(t)$的值。

## 5.4 人机交互

英文中的人机交互有 3 个对应的概念：其一是人与计算机的交互(Human-Computer Interaction，HCI)；其二是人与机器的交互(Human-Machine Interaction，HMI)；其三是人与机器人的交互(Human-Robot Interaction，HRI)。

本节中的人机交互是指人与机器人的交互。这种人与机器人交互也离不开人与计算机之间的交互，其之间互为支撑、互相交叉和融合发展。

### 5.4.1 人与机器人交互(HRI)概念和发展现状

#### 1. 人与机器人交互的概念

人—机器人交互是指人与机器人之间通过某种特定的传感器和接口，在一定的交互技术支撑下，实现相互理解的信息通信。HRI 的研究目的是使机器人与人类和谐共处、自然高效地完成用户交代的任务，并为用户提供及时有效的反馈。HRI 技术的发展不仅有益于提高人类的工作效率，而且有助于满足人类的生活需求。目前 HRI 技术已经成为机器人应用领域的一个重要研究热点。

一般根据人与机器人交流的方式不同，人与机器人的交互可分为近距离交互和远程交互两类。

(1) 近距离交互：人与机器人在同一个地方，交互在近距离发生，常见的工业机器人与服务机器人多为此种情况。

(2) 远程交互：人与机器人不在同一地方，并且在时间或空间上隔开。远程交互移动机器人通常称为遥控或遥操作，远程交互的物理机械手通常称为遥操作系统。

一般情况下，不同的应用需求驱动不同的人—机器人交互方法。下面介绍用来评价 HRI 方法性能的主要指标：

(1) 鲁棒性。指人—机器人交互系统面对干扰因素影响时的稳定性。

(2) 效率。在人—机器人交互过程中，用户控制和机器人运动在时间上最大限度地实现无缝衔接，减少机器人和用户之间因互相等待出现饥饿现象，导致人机交互系统冗余时间累积的问题。

(3) 交互体验。指人和机器人之间的交互方式最大限度地自然化、直观化、简单化、舒适化，提高人—机器人协作系统的友好性。

(4) 智能性。指在人—机器人交互过程中，机器人对用户语言、情绪、手势等的理解程度，以及是否能做出相应的符合用户习惯的回应。

人—机器人交互技术主要经历了四个发展阶段：

(1) 人—机器人交互发展的早期阶段是在结构化环境中，人—机器人基于命令交互和基于图形化界面交互，即用户通过专业的命令行语言或图形化界面对机器人任务和任务场

景进行描述，机器人运动状态也是以命令行语言或图形化界面数据方式反馈给用户。

(2) 人—机器人交互发展的第二个阶段是基于接触式手持传感装置和可佩戴传感装置的交互方式，如数据手套、加速计、陀螺仪、肌电传感器、力反馈器、惯性测量仪、特制抓握工具等。

(3) 人—机器人交互发展的第三个阶段是基于手势的人机交互方式。

(4) 人—机器人交互发展的第四个阶段是智能化人机交互方式，与语音识别等人工智能技术融合发展。

**2. 人与机器人交互的发展现状**

目前，不论是工业机器人还是服务机器人领域均广泛出现人机共融环境：一方面，近年来服务机器人成为机器人学的炙手可热的研究领域，服务机器人的工作特点决定了其必然要与服务对象(人)进行交互；另一方面，工业机器人也不可避免要与人/环境进行直接接触和交互，尤其是新一代人机协作工业机器人，它定位于辅助工业生产，与人共享工作空间，甚至需要与人协作完成工业生产，人机协作机器人已经成为工业机器人产品中的热点。

人机共融的环境中，机器人需要与人交互甚至协作完成复杂任务，称为人机交互，共融机器人、协作机器人已经成为各国机器人领域的研究重点之一。而不论是共融机器人还是人机协作机器人，均需要工作在人的活动空间内，与人进行人机交互、协作，完成不同的任务。因此，人机共融技术、人机交互技术、人机协作技术成为机器人领域的研究热点之一。

机器人人机交互技术的发展历程可分为两个阶段：以机器人为中心的受限方式、以人为中心的非受限方式。

以机器人为中心的受限方式，要求人在交互过程中将自己的意图按照机器人特有的输入方式进行精确分解，因此人需要耗费大量时间学习和记忆交互系统的使用方法，且操作受到较大限制。传统的人机交互方式如命令语言交互、图形交互界面和直接操纵方式等大都属于此类。传统的人机交互方式在交互设备和交互效率上存在着很多局限，例如使用机器人命令语言对操作者的专业知识要求较高；在图形化的人机交互系统中，操作员使用鼠标和键盘作为交互工具，每次只能通过单击和输入数据来控制机器人，交互效率低下，而且交互方式不够自然和方便。

随着传感器技术的兴起和发展，传统的以机器人为中心的受限式人机交互方式正逐渐退出研究的中心，而基于传感器技术和智能感知技术实现的以人为中心的非受限自然人机交互方式，则成了研究的热点和重点。

近年来，在人机交互领域中，许多公司设计开发了很多新型的交互设备，这些设备改变和促进了新型人机交互方式的产生。基于智能感知的机器人交互方式，通过多种模态感知人类的自然行为，增强机器人的自主决策能力，使得人类可以充分使用诸如手势、表情、语音等多种自然的方式实现人机通信，减少人的学习认知压力，提高人机交互的效率和自然性。

在当前的研究中，机器人主要以触觉、视觉和听觉的方式感知人和周边环境，新型交互方式的提出，促进了很多学者对于新型交互方式的探索，因此近年来在人机交互领域出现了很多创新实用的设计，如新型的交互方式语音交互、脑机交互、穿戴式交互、体感交互等。

随着机器人技术和应用领域的迅猛发展，人们对机器人人机交互方式的要求也越来越高。作为人类体能、智能和感知的延伸，人们希望机器人不再局限于一些简单的流水线任务或完全被动地受人控制，而是能够在一些复杂的、非结构化的甚至危险的环境中发挥主导的

作用。因此,研究自然和智能的机器人人机交互技术,显得格外重要和迫切。

智能感知技术是实现机器人人机交互的核心技术之一,重点研究基于生物特征、以自然语言和动态图像的理解为基础、“以人为中心”的智能信息处理和控制技术。利用智能感知技术,机器人能够拥有类似人眼、人耳、鼻子、皮肤等感官的功能,使得机器人和人之间、机器人和环境之间的沟通能够像人和人以及人和环境一样自然。基于智能感知技术的机器人人机交互,将智能感知技术和机器人技术相结合,机器人通过自身携带的或外界辅助的各类传感器感知自身和外界环境信息,机器人基于上述感知的信息,做出相应的运动决策,从而完成特定工作任务。

未来的人工智能技术、人机交互方式,如意念控制、虚拟现实技术等交互方式,将会在人—机器人的交互中大放异彩。

### 5.4.2 人机交互技术应用

#### 1. 虚拟示教

工业机器人示教是指操作者在实际工作环境中,通过下述方法实现:人手引导机器人末端执行器或引导一个机械模拟装置,或用示教盒操作机器人完成作业所需位姿,并记录下各个示教点的位姿数据,利用机器人语言进行在线编程,程序回放时,机器人便执行程序要求的轨迹运动。

目前,随着机器人应用技术的推广以及为了进一步提高生产效率和操作安全性,迫切需要开发新的示教编程技术;而且随着机器人离线编程技术的出现,虚拟现实技术的飞速发展已经为机器人虚拟示教的实现提供了技术支持。

虚拟现实技术是一种对事件的现实性从时间和空间上进行分解后重新组合的技术。这一技术包括三维计算机图形学技术、多功能传感器的交互接口技术以及高清晰度的显示技术。虚拟现实技术应用于遥控机器人和临场感通信,既可以将人们从危险和恶劣的环境中解脱出来,同时还可以解决远程通信时延等问题。它是一个能使人沉浸其中、超越其上、进出自如、交互作用的环境,与以往的传统仿真环境相比,在这种多维信息空间中所进行的仿真和建模,具有更高的逼真度。例如,北京航空航天大学等高校研制成功的 CRAS-BHI 型机器人与计算机辅助脑外科手术系统,已成功用于临床。

目前虚拟现实技术在机器人学主要应用于以下三个方面:

(1) 作为遥操作界面,可应用于半自主式操作;

(2) 作为机器视觉中自动目标识别和三维场景表示的直观表达;

(3) 建立具有真实感的多传感器融合系统仿真平台。

虚拟现实技术在机器人系统编程与仿真系统中的应用研究,主要集中在以下几方面:

(1) 应用虚拟现实技术提供的新型人机交互设备,寻求更好的机器人编程方式。编程方式主要有两种:一种是在虚拟环境中通过机器人示教生成机器人程序;另一种是采用人的操作示范方式,跟踪人的动作序列,自动生成机器人的动作序列。

(2) 研究虚拟环境内目标对象的建模方式,更加注重对物体物理特性的建模,以实现更趋真实的拟实操作。

(3) 研究虚拟环境与真实环境的建模与映射问题,期望实现虚拟环境下生成的控制程序,可以直接在真实环境中运行。

因而，随着虚拟现实技术研究的深入将为机器人操作、维护、教学和培训等应用提供较好的平台。

**例 5.5**　仿人神经体系机器人控制体系及人机交互方式。

根据以上多方面的分析和探讨，结合人机交互系统技术，目前世界主流的机器人示教器尚处于图形用户界面的时代，其特点是：桌面隐喻、WIMP 技术、直接操纵和“所见即所得”，很大程度上依赖于菜单选择和交互。同时，也存在着极大的弊端：图形用户界面需要占用较多的屏幕空间，并且难以表达和支持非空间性的抽象信息的交互。

工业机器人示教器尚处于人机交互技术的早期模式，因此可以断定，随着工业机器人产业和技术的发展，虚拟现实技术必将很快地被引入机器人的人机交互系统之中。仿人神经体系机器人控制体系架构如图 5.38 所示，将感觉器官影射为传感器，而把运动器官影射为机器人的驱动电动机，基于脑、脊髓、神经传导通路映射的机器人控制体系，融合人机交互技术的软硬件技术设备，经过映射和优化，建立新型的人机交互体系。从中可以看到，机器人示教器硬件体系更人性化的一个发展方向，更多的虚拟现实技术将会应用在未来的机器人人机交互系统中。

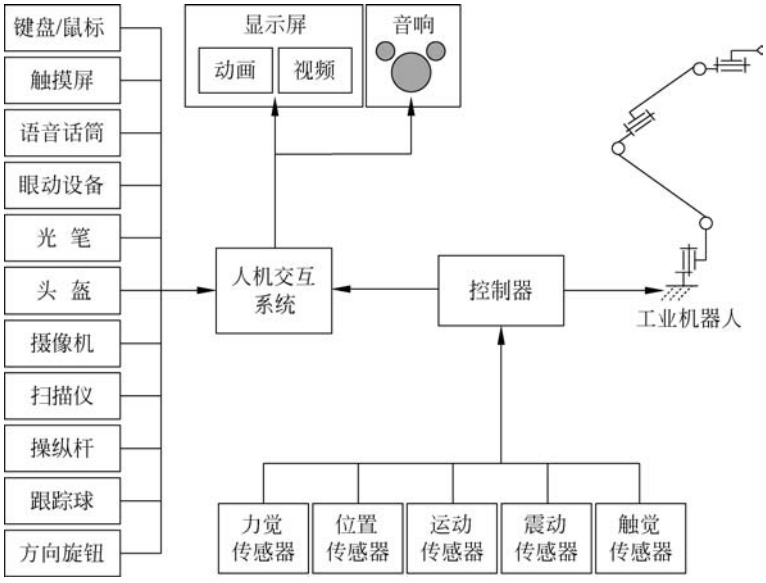



图 5.38　仿人神经体系机器人控制体系

**2. 机器人智能语音交互**

随着界面设计理论和各种人机交互支撑技术的发展，人机交互技术和方式有了很大的改变，图形对话、形式对话和自然语言对话都从自身的简单形式发展到了复杂形式。图形对话不再局限于简单的单窗口界面，还能够使用多窗口和图标等从各个角度模拟真实环境。基于形式对话的人机交互系统设计越发完美，其应用领域也越来越广。从人机交互的角度来讲，系统使用语音交互技术有两个直接的优点：一是语音输入比键盘输入快速方便；二是语音交互可以拓宽用户的输入通道，方便用户更自然地与计算机进行交互。

目前，语音识别技术已十分成熟，有的语音识别系统甚至已能够识别自然语言，这对于智能机器人产业的发展、人机交互的研究都有重大的意义。语音行业不再局限于嵌入式和

固定语法结构识别模式，而是向云端推动，向自然语言识别发展。语音交互的蓬勃发展，使其与智能人机交互机器人的深度整合成为必然。而随着语音识别技术的加入，智能人机交互机器人的想象空间越来越大，应用场景也逐渐增多。

语音交互技术在服务机器人领域获得越来越普遍的应用。

语音识别技术和智能人机交互机器人技术都属于人工智能科学领域，且拥有很高的技术含量，其发展和研究依赖于大量的语料统计和规则累计。对于这两种技术而言，高效的数据积累模型和数据规模尤为重要。

**例 5.6** 日立 Emiew2 机器人。

2014 年，日本日立公司(Hitachi)推出了名为 Emiew2 的机器人，如图 5.39 所示，不仅会讲笑话，还能读懂听众的表情，判断听众对笑话的反应。

图 5.39 Emiew2 机器人

Emiew2 机身红白相间，依靠底部的滑轮移动。其特点是能与人类进行简短的即兴对话。首先，Emiew2 从听到的句子中提取关键词，据此来判断问题在问什么，例如“多少”；然后，它还能在给出答案之前验证自己的判断——通过读取人类的表情。向 EMIEW2 说话时，它能通过麦克风和声音识别系统分辨是命令还是谈话。

**3. 机器人视觉交互**

与人类用眼睛看世界不同，机器人多是通过传感器看到电磁频谱。如今机器人开始大量使用摄像头进行图像捕捉，进而通过算法转化为计算机信号。在识别过程中，系统应先将待识别物体正确地从图像的背景中分割出来，再设法将图像中物体的属性图与假定模型库的属性图之间进行匹配，也就是对比相似的特征。在这种情况下，如果要识别准确，则算法要准确，同时也需要大量的图片库作为支撑。谷歌公司依托每天上传的海量图片，建立了人工神经网络。尽管如此，机器人眼里的世界与人类眼中的世界还是有很大的不同。人机交互机器视觉识别过程如图 5.40 所示。

基于视觉的人机交互方式主要是通过彩色图像或者深度图像实时识别人手手势。根据交互方式的不同，基于视觉的手势交互可以分为两类，一类是通过事先定义好的手势姿态，每种手势姿态对应一种不同的运动控制指令，例如向上、向下、向左、向右等。在交互过程中，机器人通过彩色图像识别到操作者的手势姿态，通过相关算法对该手势姿态与系统手势姿态库进行匹配，进而解析为相关的机器人运动控制指令，最后发送指令远程控制机器人的运动。但这种交互方式只能定性交互，无法进行定量交互，因而较大地限制了机器人的工作

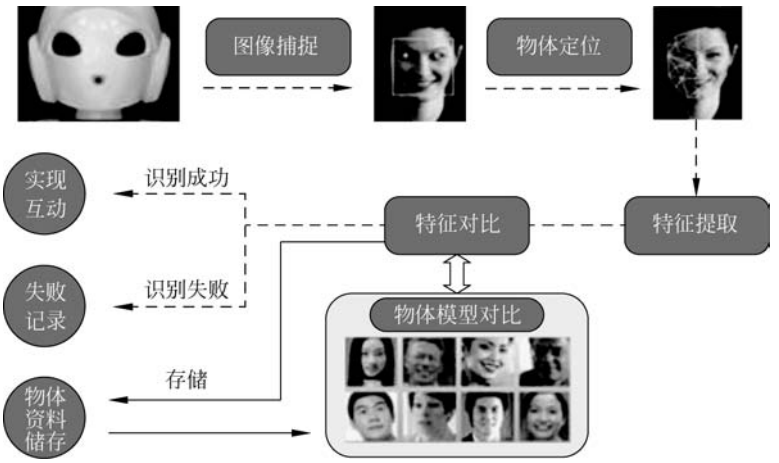



图 5.40　人机交互机器视觉识别过程

效率。另一类是直接操控型，直接通过 Leap Motion、Kinect 等深度传感器，获取操作者手的位置和姿态变化数据，将其转化为机器人的位置和姿态的变化指令，进而控制机器人的运动。这一类交互方式直观便捷，操作者无须记忆复杂烦琐的手势数据库，也无须对操作者进行事前培训，既能定性又能定量地对机器人的运动状态进行控制，因此具有很大的发展前景。

**例 5.7**　人机协作的智能无人机系统。

2017 年 5 月 20 日，中科院在沈阳科研机构公众科学日，中国科学院沈阳自动化所研制“人机协作的智能无人机系统”首次对外亮相。根据已经公开的资料，该系统是国内第一套基于视觉的智能手势控制无人机系统，如图 5.41 所示。


图 5.41　中国科学院沈阳自动化所人机协作的智能无人机系统

基于视觉的智能手势识别技术，使无人机可以根据人的意图完成不同的飞行任务，从而实现了飞行平台上的人机协同作业功能。操控的过程中，操作人员只需要对着摄像头改变相应手势即可。这套系统视觉识别模块，创新性地采用了在线实时特征提取与编程技术，在保证实时性的同时使手势正确识别率达到了 99%以上，同时还具备个性化手势定制功能，用户可以根据自己的习惯与需求，实时采集用户的手势并训练识别系统，实现不同手势功能的灵活定义与切换。

**4. 机器人脑机交互**

脑机接口(Brain-Computer Interface,BCI)能够提供一种非神经肌肉传导的通道,直接把从人头皮上采集到的脑电信号通过预处理、特征提取、选择和分类,最终转换成机器人或其他外部设备的控制指令,从而为有运动功能障碍的残疾人提供一种与外界交流或控制外部设备的方式,脑机接口的提出为实现人类梦寐以求的人脑直接控制外部设备提供了可能。

国内外学者对于脑机接口的研究已有 40 多年的历史,研究的方向和重点各有不同。按照不同的分类标准,可将 BCI 系统按以下三种方式进行分类:

(1) 按照脑电信号采集方式。

(2) 按照脑电信号控制方式。

(3) 按照脑电信号产生方式。

**例 5.8** 机器人意念控制技术。

日本国际电气通信基础技术研究所(ATR)对外展示了一款可用意念驱使的穿戴式机器人,如图 5.42 所示。该机构希望这款机器人未来与家电、轮椅等组合起来,在老年人或残疾人的日常生活中发挥辅助作用。

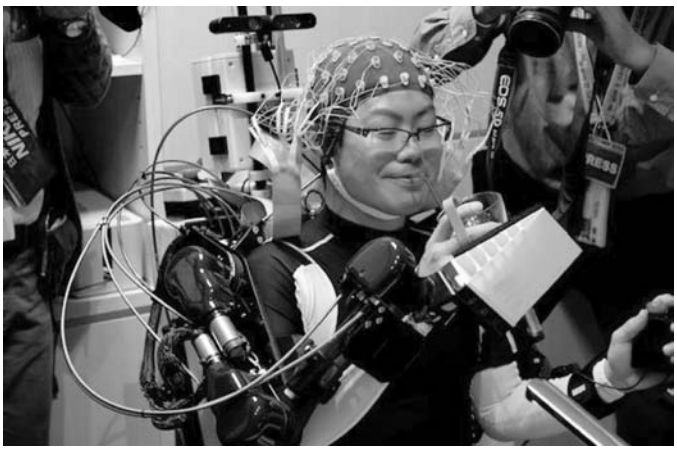


图 5.42 可用意念驱使穿戴式机器人

这种机器人运用了连接大脑与机器的"脑机接口(BMI)"技术,由 ATR、日本 NTT 公司、岛津制作所等共同研发。

当天一名使用者展示了坐在电动轮椅里喝水的情景。他头戴读取脑波的设备,不出声默想约 6s 后,电动轮椅便自动移动到水龙头面前。之后,穿在使用者上半身的机器人转动它的手臂,最终用杯子接水并送到他的嘴边。据研究小组介绍,"想喝水"的脑波不够明确,设备较难解析,因此他们让使用者默念"想活动手",使之成为一连串动作的开关。实验房间内安装了约 3000 个传感器,因此成功把握了水龙头和使用者的位置。该机器人有望帮助有脑梗塞后遗症的患者等人群更好地生活。

**5. 机器人触觉交互**

触觉是机器人获取环境信息的一种仅次于视觉的重要知觉形式,可直接测量对象和环境的多种性质特征。触觉交互主要任务是为获取对象与环境信息和为完成某种作业任务而

对机器人与对象、环境相互作用时的一系列物理特征量进行检测或感知。广义地说，它包括接触觉、压觉、力觉、滑觉、冷热觉等，狭义地说，它是机器人与对象接触面上的力感觉。

触觉传感器技术是机器人触觉功能实现的关键，触觉传感器是机器人与环境直接作用的必要媒介，是模仿人手使之具有接触觉、滑动觉、热觉等感知功能。在机器人领域使用触觉传感器的目的在于获取机械手与工作空间中物体接触的有关信息。

类皮肤型触觉传感器具有以下功能和特性：

(1) 触觉敏感能力，包括接触觉、分布压觉、接触力觉和滑觉；

(2) 柔性接触表面，以避免硬性碰撞和适应不同形状的表面；

(3) 小巧的片状外形，以利于安装在机器人手爪上。

**例 5.9**　自带触觉反馈机器人微创手术。

澳大利亚迪肯大学和哈佛大学联手打造了一款 HeroSurg 机器人，HeroSurg 机器人是为了腹腔镜手术专门打造的，所以它尤其适用于那些需要缝合微小组织的手术中。虽然在患者看来，它的外观可能会有点惊悚，但是研发团队希望它可以让手术更加安全、精准度更高。更重要的是，它可以让外科医生们获得一个重要的感觉——触觉。它可以通过触觉反馈机制，将触觉传递给主刀医生以及 3D 图像处理器，进而，医生可以看到他们的手术刀到底割到了哪里。

HeroSurg 机器人是个主从式的手术系统，如图 5.43 所示，当它处于从属地位时，配置了多重机械臂，上面带有各种手术工具和腹腔镜；而当它处于主导地位的时候，就成了为外科医生提供触觉的操控手柄。

图 5.43　HeroSurg 腔镜手术机器人

有了 HeroSurg 机器人，医生们可以感受到他们在使用器具的过程中，到底给患者施加了多少的力。即当医生通过手术器具抓到了一些东西，或是在切一些身体组织时，他们可以感受到自己对这个组织施加的力气，他们还可以感受到组织的软硬程度，并对组织的性质做出判断。HeroSurg 机器人因为具有触觉反馈能力、避免碰撞的功能以及自动适应患者和床的能力，外科医生根本不用担心在手术过程中，机器人是否会触碰到患者的体内或体外。

**6. 触屏交互**

触屏技术是一种新型的人机交互输入方式，与传统的键盘和鼠标输入方式相比，触摸屏

输入更直观。配合识别软件,触摸屏还可以实现手写输入。触摸屏作为一种最新的计算机、手机等电子类产品输入设备,它是目前最简单、方便、自然的人机交互方式之一。触摸屏主要应用于公共信息的查询、办公、工业控制、军事指挥、电子游戏、点歌点菜、多媒体教学等。

触屏技术在机器人领域主要用于和机器人的信息交互,通过触屏对机器人下达操作指令,机器人收到指令完成相应的工作,并在触屏上反馈工作执行情况。

**例 5.10** 远程呈现技术。

派宝机器人是映博智能科技公司研发的一款远程呈现机器人,如图 5.44 所示。人们可以利用派宝机器人在不同的地方实时呈现出自己的行为,包括话语、表情和动作。人们可以远程控制派宝机器人前进、后退、左转、右转、抬头和低头。派宝机器人创造了一种崭新的通信方式,通过触屏交互的方式,使人可以不受空间限制进行面对面的沟通。

图 5.44 远程呈现机器人

派宝机器人使用平板电脑作为它的大脑,例如 iPad 或安卓平板电脑。平板电脑通过蓝牙 4.0 技术和机器人的底座进行连接。人们只要将派宝 App 安装到手中的手机或平板上,并连入 WiFi 或 4G 网络,即可远程控制派宝机器人。

## 本章小结

本章主要介绍了机器人任务规划、运动规划、轨迹规划和人机交互等内容。第 1 节介绍了机器人任务规划的基础知识,主要包含任务规划问题分解途径、规划域的预测与规划修正。第 2 节介绍了运动规划所要解决的问题和位形空间的概念,重点介绍了一些典型的机器人规划方法,包括确定性图形搜索方法、PRM 和 RRT 为代表的随机图形搜索方法和人工势场法。第 3 节阐述了机器人轨迹规划的内容,着重讨论了机器人关节空间轨迹的差值计算方法,包括三次多项式插值、过路径点的三次多项式插值、高阶多项式插值、用抛物线过渡的线性插值和过路径点的用抛物线过渡的线性插值等方法;随后介绍了笛卡儿空间路径轨迹规划方法。第 4 节介绍了人与机器人交互的基本概念,当前主要的交互方式和发展趋势。

# 参 考 文 献


[1] 布鲁诺·西西里安诺，洛伦索·夏维科，路易吉·维拉尼，等. 机器人学 建模、规划与控制[M]. 张国良，曾静，陈励华，等译. 西安：西安交通大学出版社，2015.

[2] 熊有伦. 机器人技术基础[M]. 武汉：华中科技大学出版社，1996.

[3] Saeed B. Biku 等. 机器人学导论——分析、控制及应用[M]. 孙富春，朱纪洪，刘国栋，等译. 2 版. 北京：电子工业出版社，2013.

[4] 布鲁诺·西西里安诺，欧沙玛·哈提卜. 机器人手册第 2 卷 机器人基础[M].《机器人手册》翻译委员会，译. 北京：机械工业出版社，2016.

[5] 杜广龙，张平. 机器人自然交互理论与方法[M]. 广州：华南理工大学出版社，2017.

[6] J. F. Canny. The Complexity of Robot Motion Planning. Cambridge，MA：MIT Press，1998.

[7] J. Barraquand，J. C. Latombe. Robot motion planning：A distributed representation approach. International Journal of Robotics Research. vol. 10，pp. 628-649，1991.

[8] H. Choset，K. M. Lynch，S. Hutchinson，et al. Principles of Robot Motion：Theory，Algorithms，and Implementations. Cambridge，MA：MIT Press，2005.

[9] L. E. Kavraki，P. Svestka，J. C. Latombe，et al. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE Transactions on Robotics and Automation，vol. 12，pp. 566-580，1996.

[10] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. International Journal of Robotics Research，vol. 5，no. 1，pp. 90-98，1986.

[11] J.-C. Latombe，Robot Motion Planning，Boston，MA：Kluwer，1991.

[12] J.-P. Laumond. Robot Motion Planning and Control，Berlin：Springer-Verlag，1998.

[13] S. M. LaValle. Planning Algorithms，New York：Cambridge University Press，2006.

[14] S. M. LaValle，J. J. Kuffner. Rapidly-exploring random trees：Progress and prospects. New Directions in Algorithmic and Computational Robotics. pp. 293-308，2001.

[15] 蔡自兴，谢斌. 机器人学[M]. 3 版. 北京：清华大学出版社，2015.

[16] Tian-Miao Wang，Yong Tao，Hui Liu. Current Researches and Future Development Trend of Intelligent Robot：A Review. International Journal of Automation and Computing，Vol. 15，Iss5，pp. 525-546，2018.

[17] 袁媛. 基于自然语言的服务机器人任务规划与执行研究[D]. 山东大学，2018.

[18] 梁献霞. 室内环境下移动机器人的路径规划研究[D]. 河北科技大学. 2017.

[19] 李达. 工业机器人轨迹规划控制系统的研究[D]. 哈尔滨工业大学，2011.

[20] 刘翔宇. 基于仿人智能的人机交互技术研究[D]. 华南理工大学，2017.

[21] 安徽大学高级人工智能课程 http://www.docin.com/p-974289323.htm.

[22] 机器人的工程及应用轨迹规划、生成与控制技术 http://www.doc88.com/p-9045496433173.html.

[23] 闫宏阳. 基于限定 Delaunay 三角剖分的移动机器人路径规划[D]. 2008.

[24] 黎波. 机器人作业任务规划研究[D]. 2012.

[25] 机器人的轨迹规划 https://wenku.baidu.com/view/6c2bf8172f60ddccda38a0aa.html.

[26] 机器人轨迹规划 http://www.docin.com/p-534767855.html.

[27] 机器人的轨迹规划 http://www.doc88.com/p-5425418773324.html.

[28] 从人体解剖学分析机器人的人机交互及控制系统 https://wenku.baidu.com/view/2df7d160ddccda38376bafd0.html.

[29] 中国首套视觉智能手势控制无人机系统沈阳造 http://www.sia.cn/xwzx/mtjj/201705/t20170522_4794804.html.

[30] 自带触觉反馈的机器人隔空微创手术 so easy
http://tech.sina.com.cn/q/tech/2016-09-30/doc-ifxwkzyh3922996.shtml.
[31] 2015 仿人服务机器人调研报告
https://wenku.baidu.com/view/db7b6eccb52acfc788ebc98c.html.
[32] 基于限定 Delaunay 三角剖分的移动机器人路径规划[D].北京工业大学,2008.
[33] 刘娅. 基于可视图法的避障路径生成及优化[D]. 昆明理工大学,2012.
[34] 徐联杰. 机电产品中的管路自动布局设计技术研究[D].北京理工大学,2016.
[35] 孔迎盈. 机器人路径规划算法的研究[D]. 复旦大学,2012.
[36] 杨柳. 移动机器人动态路径规划方法研究[D]. 江南大学,2011.
[37] 陈伟华. 工业机器人笛卡儿空间轨迹规划的研究[D]. 华南理工大学,2015.
[38] 周旋. 工业机器人最优轨迹规划问题研究[D].沈阳建筑大学,2016.

# 思考题与练习题

**思考题**

1. 机器人任务规划的要素是什么?
2. 主要有哪几种典型的机器人任务规划系统?
3. 简述机器人轨迹规划与运动规划的区别。
4. 精确单元分解与近似单元分解的异同点是什么?
5. 机器人的人机交互主要应用于哪些方面?
6. 简述你想象中的未来人机交互方式。

**练习题**

1. 设要求某个转动关节路径点为 10、35、25。每一段的持续时间分别为 2s、1s 和 3s。如图 5.45 和图 5.46 所示,并设在抛物线段的默认加速度均为 $50°/s^2$。要求计算各轨迹段的速度,抛物线段的持续时间以及线性段的时间。

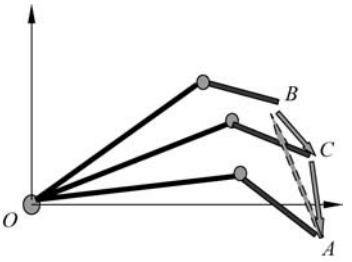



图 5.45 平面机器人动作示意

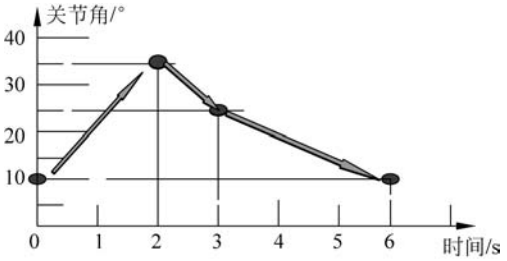



图 5.46 关节角度随时间变化图

2. 要求一个六轴机器人的关节 1 在 5s 内从初始 30°,运动到 75°,且起始点和终止点速度均为零,起始点角加速度和终止点角减速度均为 $5°/s^2$,用五次多项式规划该关节的运动。

3. 设一机器人具有 6 个转动关节,其关节运动均按三次多项式规划,要求经过两个中间路径点后停在一个目标位置。试问要描述该机器人关节的运动,共需要多少个独立的三次多项式?要确定这些三次多项式,需要多少个系数?

4. 单连杆机器人的转动关节,从 $\theta=-5°$静止开始运动,要想在 4s 内使该关节平滑地

运动到 $\theta=+80°$ 的位置停止。试按下述要求确定运动轨迹：

(1) 关节运动依三次多项式插值方式规划。

(2) 关节运动按抛物线过渡的线性插值方式规划。

5. 在从 $t=0$ 到 $t=2$ 的时间区间使用一条三次多项式：$\theta(t)=10+5t+70t^2-45t^3$，求其起始点和终点的位置、速度和加速度。

# 思考题与练习题参考答案

**思考题**

1. 答：机器人任务规划的基本要素包括状态空间(State)、时间(Time)、操作状态的动作序列(Actions)、初始和目标状态(Initial and goal states)、标准(A criterion)和运动计划(A plan)。

2. 答：典型的机器人任务规划系统有 STRIPS 规划系统、具有学习能力的规划系统和基于专家系统的机器人规划。

3. 答：运动规划是决定一条从初始位姿到最终位姿的路径，使机器人能沿这条路无碰撞地完成作业任务。这需要为机器人赋予自主规划能力，自主规划需要从用户提供的任务级高层描述和工作空间的几何特征出发。

机器人运动轨迹一般是对其末端执行器位姿变化的描述。控制轨迹也就是按时间控制手部走过的空间路径。在轨迹规划中，也常用点表示机器人在某一时刻的状态，或某一时刻的轨迹，或用它表示末端执行器的位姿，例如起始点、终止点就分别表示末端执行器的起始位姿及终止位姿。

4. 答：相同点：

单元分解法的思想是将机器人所在环境空间切分为多个简单相连的区域，每个区域单元分为自由的和被物体占用的两种。然后找出起点和目标位置所在单元，并在连接图中用搜索算法找到一条连接起点和目标单元的路径。

不同点：

单元分解法又分为精确单元分解(Exact Cell Decomposition)和近似单元分解(Approximate Cell Decomposition)。单元分解法的一个重要评价标准是对环境划分的完备程度，如果环境分解后是无损的，那么这种划分法就是精确单元分解。如果分解形成实际地图的近似，则称为近似单元分解。

5. 答：主要应用于虚拟示教、语音交互、视觉交互、脑机交互机、触觉交互、触屏交互等方面，未来还有更多的应用。

6. 答：(本题属于开放题)

伴随着信息物理融合系统、物联网等传感网络技术的不断发展，人—机器人交互将在不久的将来形成集视觉、嗅觉、触觉、推理与情感计算于一体的全方位感知智能时代，而物联网、云计算、大数据处理等新一代信息技术的研究将在不远的未来继续带动信息、生物、医疗等交叉学科的建设，并将人—机器人交互推向心理学、脑科学、认知科学、神经科学与计算技术相融合的灵魂智能时代。

**练习题**

1. **解**：$\ddot{\theta}_1=\mathrm{sgn}(\theta_2-\theta_1)\,|\ddot{\theta}_1|=50°/s^2$

$$t_1 = t_{d12} - \sqrt{t_{d12}^2 - \frac{2(\theta_2 - \theta_1)}{\ddot{\theta}_1}} = 2 - \sqrt{4 - \frac{2(35-10)}{50}} = 0.27\text{s}$$

$$\dot{\theta}_{12} = \frac{\theta_2 - \theta_1}{t_{d12} - \frac{1}{2}t_1} = \frac{35-10}{2 - 0.5 \times 0.27} = 13.5^\circ/\text{s}^2$$

$$\dot{\theta}_{23} = \frac{\theta_3 - \theta_2}{t_{d23}} = \frac{25-35}{1} = -10^\circ/\text{s}^2$$

$$\ddot{\theta}_2 = \text{sgn}(\dot{\theta}_{23} - \dot{\theta}_2)\,|\ddot{\theta}_2| = -50^\circ/\text{s}^2$$

$$t_2 = \frac{\dot{\theta}_{23} - \dot{\theta}_{12}}{\ddot{\theta}_2} = \frac{-10 - 13.5}{-50} = 0.47\text{s}$$

$$t_{12} = t_{d12} - t_1 - \frac{1}{2}t_2 = 2 - 0.27 - \frac{1}{2} \times 0.47 = 1.5\text{s}$$

$$\ddot{\theta}_4 = \text{sgn}(\theta_3 - \theta_4)\,|\ddot{\theta}_4| = 50^\circ/\text{s}^2$$

$$t_4 = t_{d34} - \sqrt{t_{d34}^2 - \frac{2(\theta_4 - \theta_3)}{\ddot{\theta}_4}} = 3 - \sqrt{9 + \frac{2(10-25)}{50}} = 0.102\text{s}$$

$$\dot{\theta}_{34} = \frac{\theta_4 - \theta_3}{t_{d34} - \frac{1}{2}t_4} = \frac{10-25}{3 - 0.5 \times 0.102} = -5.1^\circ/\text{s}^2$$

$$\ddot{\theta}_3 = \text{sgn}(\theta_{34} - \theta_{23})\,|\ddot{\theta}_3| = 50^\circ/\text{s}^2$$

$$t_3 = \frac{\dot{\theta}_{34} - \dot{\theta}_{23}}{\ddot{\theta}_3} = \frac{-5.1 - (-10)}{50} = 0.098\text{s}$$

$$t_{23} = t_{d23} - \frac{1}{2}t_2 - \frac{1}{2}t_3 = 1 - \frac{1}{2} \times 0.47 - \frac{1}{2} \times 0.098 = 0.716\text{s}$$

$$t_{34} = t_{d34} - t_4 - \frac{1}{2}t_3 = 3 - 0.102 - \frac{1}{2} \times 0.098 = 2.849\text{s}$$

2. **解**：把 $\theta_0$ 和 $\theta_f$ 的值代入式(5.38)中

$$\begin{cases} a_0 = \theta_0 \\ a_1 = \dot{\theta}_0 \\ a_2 = \dfrac{\ddot{\theta}_0}{2} \\ a_3 = \dfrac{20\theta_f - 20\theta_0 - (8\dot{\theta}_f + 12\dot{\theta}_0)t_f - (3\ddot{\theta}_0 - \ddot{\theta}_f)t_f^2}{2t_f^3} \\ a_4 = \dfrac{30\theta_0 - 30\theta_f + (14\dot{\theta}_f + 16\dot{\theta}_0)t_f + (3\ddot{\theta}_0 - 2\ddot{\theta}_f)t_t^2}{2t_f^4} \\ a_5 = \dfrac{12\theta_f - 12\theta_0 - (6\dot{\theta}_f + 6\dot{\theta}_0)t_f + (\ddot{\theta}_0 - \ddot{\theta}_f)t_f^2}{2t_f^5} \end{cases}$$

可得五次多项式的系数：

$a_0=30.0$，　$a_1=0.0$，　$a_2=2.5$，　$a_3=1.6$，　$a_4=-0.58$，　$a_5=0.0464$

确定操作臂为：

$$\theta(t)=30+2.5t^2+1.6t^3-0.58t^4+0.0464t^5$$

$$\dot{\theta}(t)=5+4.8t^2-2.32t^3+0.232t^4$$

$$\ddot{\theta}(t)=5+9.6t-6.96t^2+0.928t^3$$

3. 答：该机器人路径可分为 $q_0\sim q_r$ 段和 $q_r\sim q_s$ 段及 $q_s\sim q_5$ 段三段。可通过由三个三次多项式组成的样条函数连接。

设从 $q_0\sim q_r$ 的三次多项式插值函数为：$\theta_1(t)=a_{10}+a_{11}t+a_{12}t^2+a_{13}t^3$

从 $q_r\sim q_s$ 的三项多项式插值函数为：$\theta_2(t)=a_{20}+a_{21}t+a_{22}t^2+a_{23}t^3$

从 $q_s\sim q_5$ 的三项多项式插值函数为：$\theta_3(t)=a_{30}+a_{31}t+a_{32}t^2+a_{33}t^3$

上述三个三次多项式的时间区间分别为$[0,t_{f1}]$和$[0,t_{f2}]$及$[0,t_{f3}]$，若要保证路径点处的速度及加速度均连续，即存在下列约束条件：

$$\dot{\theta}_1(t_{f1})=\dot{\theta}_2(0)$$

$$\dot{\theta}_2(t_{f2})=\dot{\theta}_3(0)$$

$$\ddot{\theta}_1(t_{f1})=\dot{\theta}_2(0)$$

$$\ddot{\theta}_2(t_{f2})=\ddot{\theta}_3(0)$$

根据约束条件建立方程组，为：

$$\theta_0=a_{10}$$

$$\theta_r=a_{10}+a_{11}t_{f1}+a_{12}t_{f2}^2+a_{13}t_{f3}^3$$

$$\theta_r=a_{20}$$

$$\theta_s=a_{20}+a_{21}t_{f2}+a_{22}t_{f2}^2+a_{23}t_{f2}^3$$

$$\theta_s=a_{30}$$

$$\theta_5=a_{30}+a_{31}t_{f3}+a_{32}t_{f3}^2+a_{33}t_{f3}^3$$

$$a_{11}=0$$

$$a_{21}+2a_{22}t_{f2}+3a_{23}t_{f2}^2=0$$

$$a_{11}+2a_{12}t_{f1}+3a_{13}t_{f1}^2=a_{21}$$

$$a_{21}+2a_{22}t_{f2}+3a_{23}t_{f2}^2=a_{31}$$

$$2a_{12}+6a_{13}t_{f1}=2a_{22}$$

$$2a_{22}+6a_{23}t_{f2}=2a_{32}$$

显然上述约束条件含有 12 个未知数的 12 个线性方程。

综上所求解知：需要三个独立方程，12 个系数。

4. **解**：过渡域：$[t_0,t_b]$。

由于过渡域$[t_0,t_b]$终点的速度必须等于线性域的速度，所以

$$\dot{\theta}_{tb}=\frac{\theta_h-\theta_b}{t_h-t_b} \tag{1}$$

其中，$\theta_b$ 为过渡域终点 $t_b$ 处的关节角度。用 $\ddot{\theta}$ 表示过渡域内的加速度，$\theta_b$ 的值可按下式求解，$\theta_b=\theta_0+\frac{1}{2}\ddot{\theta}t_b^2$ (2)

且：$t=2t_h$

$$\ddot{\theta}t_b^2-\ddot{\theta}tt_b+(\theta_f-\theta_0)=0 \tag{3}$$

其中 $t$ 为运动持续时间，此处为 4s。

对于给定的 $\theta_f=+80°$，$\theta_0=-5°$，$t=4$s 可根据(3)选择相应的 $\ddot{\theta}$ 和 $t_b$，得到路径曲线。通常的做法是先选择加速度 $\ddot{\theta}$ 的值，然后按(3)算出相应的 $t_b$，

$$t_b=\frac{t}{2}-\frac{\sqrt{\ddot{\theta}^2t^2-4\ddot{\theta}(\theta_f-\theta_0)}}{2\ddot{\theta}} \tag{4}$$

为了保证 $t_b$ 有解，过渡域加速度值 $\ddot{\theta}$ 必须选得足够大，即

$$\ddot{\theta}\geqslant\frac{4(\theta_f-\theta_0)}{t^2} \tag{5}$$

此处不妨取：$\ddot{\theta}=\frac{4(\theta_f-\theta_0)}{t^2}=\frac{4\times[80°-(-5°)]}{4^2}=21.25$。代入(4)可求得：

$$t_b=\frac{t}{2}-\frac{\sqrt{\ddot{\theta}^2t^2-4\ddot{\theta}(\theta_f-\theta_0)}}{2\ddot{\theta}}=\frac{4}{2}-\frac{\sqrt{21.25^2\times4^2-4\times21.25\times[80-(-5)]}}{2\times21.25}=2$$

按照(2)式可求得 $\theta_b$ 得：

$$\theta_b=\theta_0+\frac{1}{2}\ddot{\theta}t_b^2=(-5)+\frac{1}{2}\times21.25\times2^2=37.5°$$

综上求解可得(1)式：

$$\dot{\theta}_{tb}=\frac{\theta_h-\theta_b}{t_h-t_b}=\frac{\theta_h-37.5°}{t_h-2}$$

5. **解**：由已知有下述条件：

时间区域：[0,2]。

$$a_0=10°,\quad a_1=5°,\quad a_2=70°,\quad a_3=-45°$$

由条件可得：

$$a_0=\theta_0=10°$$

$$\theta_f=10+5\times2+70\times2^2-45\times2^3=-60°$$

$$\dot{\theta}(t)\mid_{t=0}=5+140t-135t^2=5$$

$$\ddot{\theta}(t)\mid_{t=0}=140-270t=140$$

$$\dot{\theta}(t)\mid_{t=2}=5+140t-135t^2=-255$$

$$\ddot{\theta}(t)\mid_{t=2}=140-270t=-400$$

问题得解。