

机器人原理

轨迹生成

中山大学人工智能学院



轨迹生成

- 内容：路径描述和路径生成的介绍、关节空间的规划方法、笛卡儿空间规划方法、笛卡儿路径的几何问题、路径的实时生成
- 重点：轨迹的插值和优化
- 难点：轨迹中的奇异点和多解



7.1 引言



7.1 引言

- 本章讨论计算轨迹的方法，该轨迹描述了操作臂在多维空间中的期望运动。
- 这个问题包括如何通过人机交互指定通过空间的一条轨迹或路径、规划好的轨迹如何在计算机中进行描述、还要研究通过内部表达式计算轨迹的问题，即轨迹生成问题。
- 大多数情况下，在轨迹生成的运行时间内需要计算位置、速度和加速度。这些轨迹由计算机计算，因此轨迹点是以某种速率被计算的，叫作路径更新速率。
- 在典型的操作臂系统中，路径更新速率在 60 Hz--2000 Hz 之间。



7.2 关于路径描述和路径生成的综述

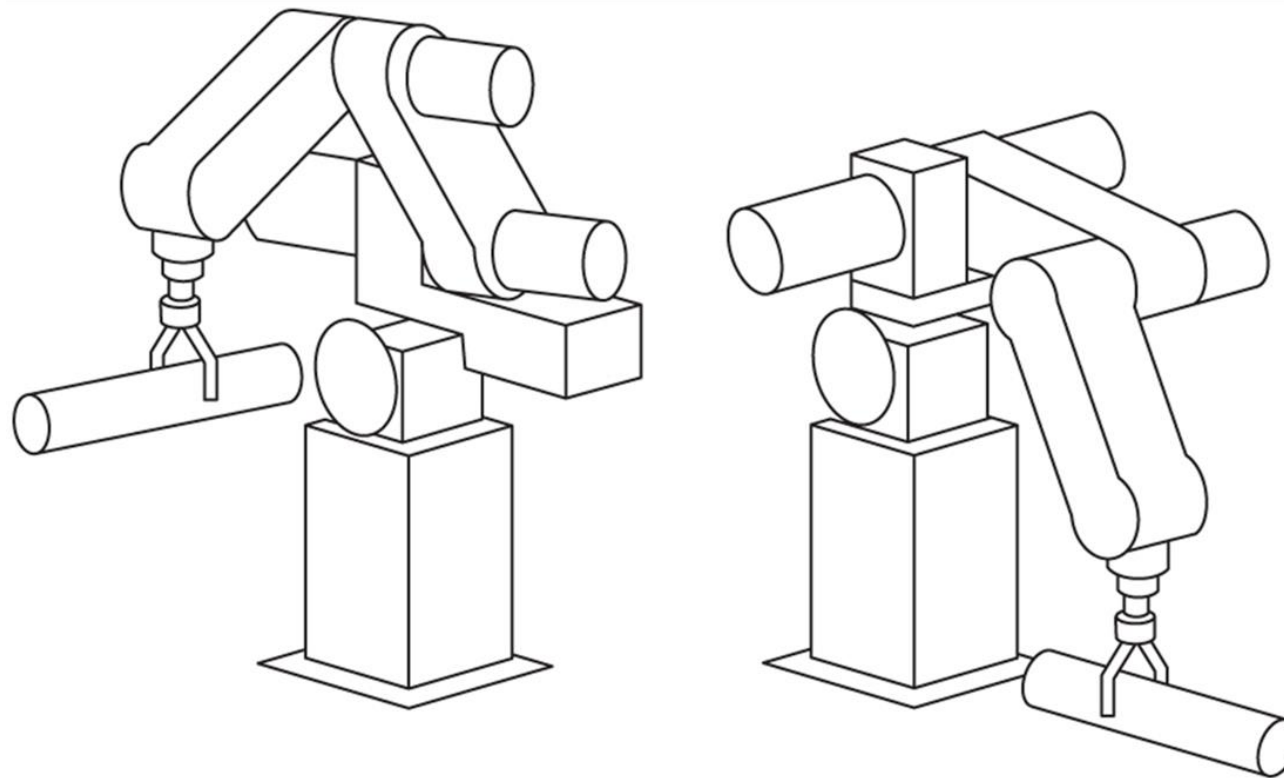


7.2 关于路径描述和路径生成的综述

- 基本问题是将操作臂从初始位置移动到某个最终期望位置——也就是将工具坐标系从当前值 $\{T_{\text{initial}}\}$ 到最终期望值 $\{T_{\text{final}}\}$ 。
- 一般情况下，在路径描述中需要给出一系列的期望中间点。因此，为了完成这个运动，工具坐标系必须经过中间点所描述的一系列过渡位置与姿态。
- 因此，路径点包括了所有的中间点以及初始点和最终点。



- 通常，我们期望操作臂的运动是平滑的。为此，要定义一个连续的且具有连续一阶导数的平滑函数。有时还希望二阶导数也是连续的。因此，为了保证路径平滑，必须在各中间点之间对路径的空间和时间特性添加一些限制条件。



7.3 关节空间的规划方法



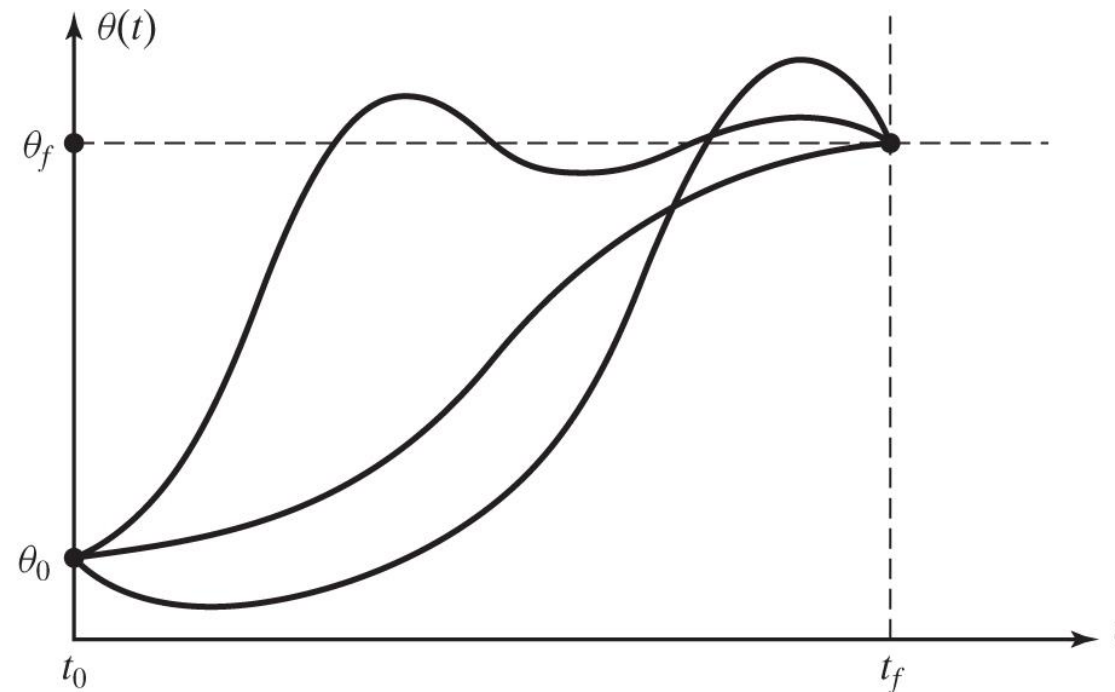
7.3 关节空间的规划方法

- 本节将研究以关节转角的函数来描述轨迹（在空间和时间）的轨迹生成方法。
- 每个路径点通常是用工具坐标系 $\{T\}$ 相对于固定坐标系 $\{S\}$ 的期望位置和姿态来确定的。应用逆运动学理论，求解出轨迹中每点对应的期望关节角。这样，就得到了经过各中间点并终止于目标点的 n 个关节中每个关节的平滑函数。
- 对于每个关节而言，由于各路径段所需要的时间是相同的，所有关节将同时到达各中间点，从而得到在每个中间点上的期望的笛卡儿位置。
- 因此，应用关节空间规划方法可以获得各中间点的期望位置和姿态。



三次多项式

- 现在需要确定每个关节的运动函数。在 t_0 时刻的值为该关节的初始位置，在 t_f 时刻的值为该关节的期望目标位置。如图 7-2 所示，有多种平滑函数 θ_t 均可用于对关节角进行插值。



三次多项式

- 为了获得一条确定的光滑运动曲线，显然至少需要对 θ_t 施加 4 个约束条件。
- 由初始值和最终值可得到对函数值的两个约束条件：

$$\theta(0) = \theta_0, \quad \theta(t_f) = \theta_f \quad (7.1)$$

- 另外，需要保证关节速度连续，即在初始时刻和终止时刻关节速度为零：

$$\dot{\theta}(0) = 0, \quad \dot{\theta}(t_f) = 0 \quad (7.2)$$



三次多项式

- 次数至少为 3 的多项式才能满足这 4 个约束条件。这些约束条件唯一确定了一个三次多项式。该三次多项式具有如下形式：

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \quad (7.3)$$

- 所以对应于该路径的关节速度和加速度为：

$$\dot{\theta}(t) = a_1 + 2a_2 t + 3a_3 t^2 \quad (7.4)$$

$$\ddot{\theta}(t) = 2a_2 + 6a_3 t$$



三次多项式

- 把这 4 个约束条件代入可以得到含有 4 个未知量的 4 个方程：

$$\begin{aligned}\theta_0 &= a_0 \\ \theta_f &= a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 \\ 0 &= a_1 \\ 0 &= a_1 + 2a_2 t_f + 3a_3 t_f^2\end{aligned}\tag{7.5}$$

- 解出方程中的 a_i ,得到:

$$a_0 = \theta_0, \quad a_1 = 0, \quad a_2 = \frac{3}{t_f^2}(\theta_f - \theta_0), \quad a_3 = -\frac{2}{t_f^3}(\theta_f - \theta_0) \tag{7.6}$$

- 因此，可以求出从任何起始关节角位置到终止位置的三次多项式。但是该解仅适用于起始关节角速度与终止关节角速度均为零的情况。



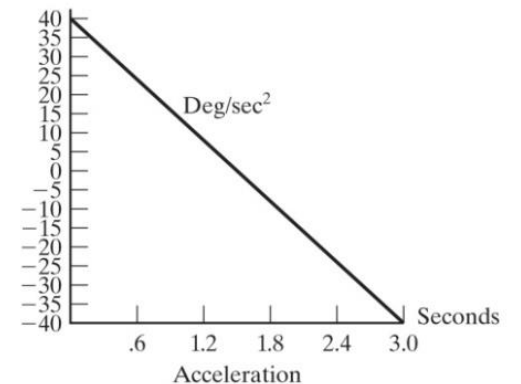
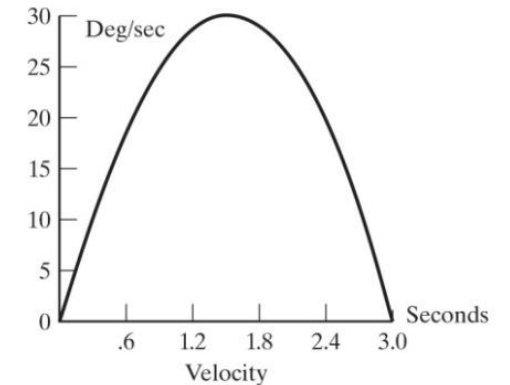
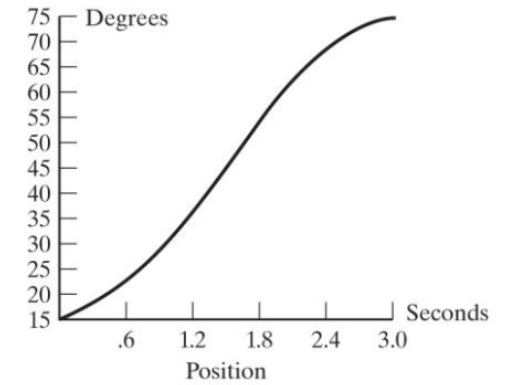
具有中间点的路径的三次多项式

- 这时在每个终止点的速度限制条件不再为零，而是已知的速度。于是，式(7.3)的限制条件变成：

$$\dot{\theta}(0) = \theta_0, \quad \dot{\theta}(t_f) = \theta_f \quad (7.9)$$

- 描述这个一般三次多项式的 4 个方程为：

$$\begin{aligned} \theta_0 &= a_0 \\ \theta_f &= a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 \\ \dot{\theta}_0 &= a_1 \\ \dot{\theta}_f &= a_1 + 2a_2 t_f + 3a_3 t_f^2 \end{aligned} \quad (7.10)$$



用于具有中间点的路径的三次多项式

- 解出方程中的 a_i ,得到:

$$\begin{aligned}a_0 &= \theta_0 \\a_1 &= \dot{\theta}_0 \\a_2 &= \frac{3}{t_f^2} (\theta_f - \theta_0) - \frac{2}{t_f} \dot{\theta}_0 - \frac{1}{t_f} \dot{\theta}_f \\a_3 &= -\frac{2}{t_f^3} (\theta_f - \theta_0) + \frac{1}{t_f^2} (\dot{\theta}_0 + \dot{\theta}_f)\end{aligned}\tag{7.11}$$



用于具有中间点的路径的三次多项式

确定中间点处的期望关节速度可以使用以下几种方法：

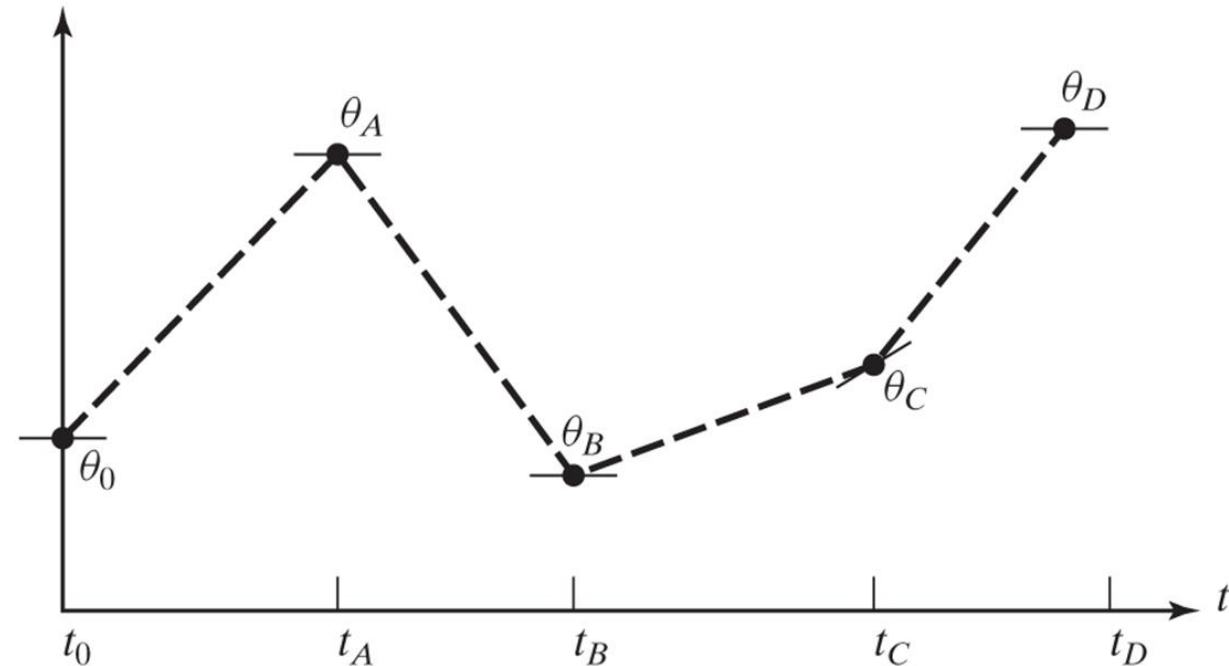
- 1) 用户给出每个瞬时工具坐标系的笛卡儿线速度和角速度，从而确定每个中间点的期望速度。
- 2) 在笛卡儿空间或关节空间中使用适当的启发算法，系统自动选取中间点的速度。
- 3) 系统自动选取中间点的速度，使得中间点处的加速度连续。



用于具有中间点的路径的三次多项式

第二种方法，系统使用一些启发算法来自动地选择合理的过渡速度。考虑在图 7-4 中所示由中间点确定的某一关节的路径。

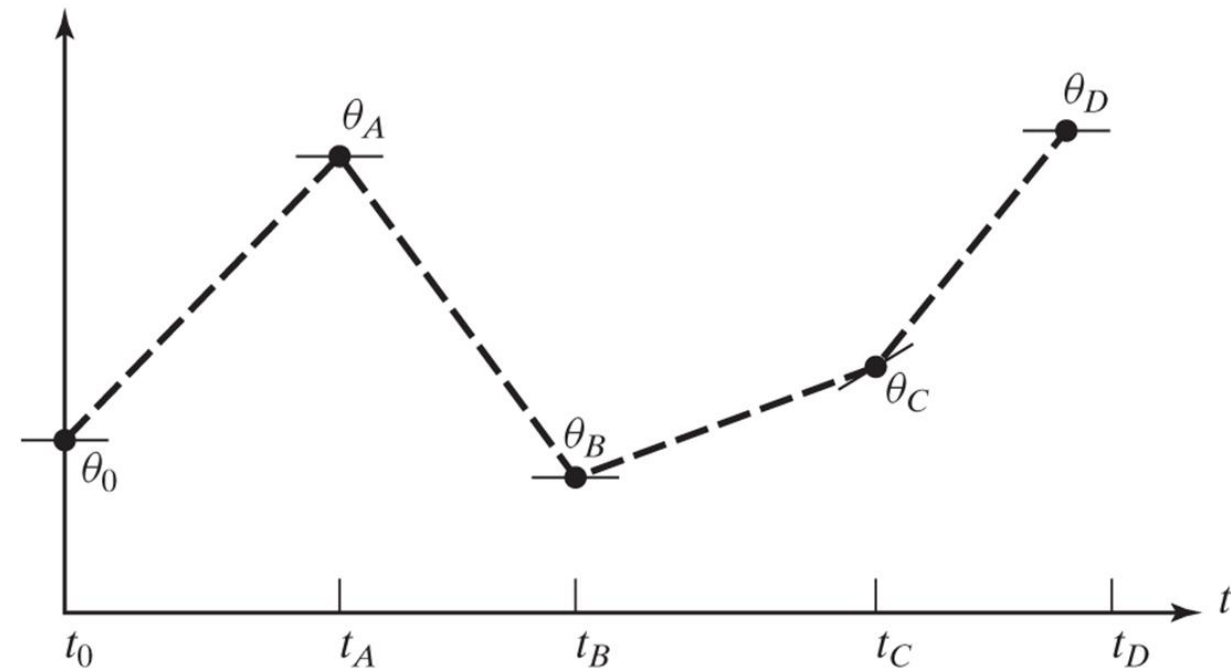
在图 7-4 中，合理选取了各中间点上的关节速度，并用小的直线段来表示，这些直线段即为曲线在每个中间点处的切线。



用于具有中间点的路径的三次多项式

假设用直线段把中间点连接起来。
如果这些直线的斜率在中间点处
改变符号，则把速度选定为零；
如果这些直线的斜率没有改变符号，
则选取中间点两侧线段
斜率的平均值作为该点的速度。

按照此法，系统可以只根据规定的
期望中间点来自动选取每个中
间点的速度。



例 7.2

- 试求解两个三次曲线的系数使得两曲线连成的样条曲线在中间点处具有连续的加速度。假设起始角为 θ_0 ，中间点为 θ_v ，终止点为 θ_g 。
- 第一个三次曲线为
$$\theta(t) = a_{10} + a_{11}t + a_{12}t^2 + a_{13}t^3$$
- 第二个三次曲线为
$$\theta(t) = a_{20} + a_{21}t + a_{22}t^2 + a_{23}t^3$$
- 在一个时间段内，每个三次曲线的起始时刻为 $t = 0$ ，终止时刻 $t = t_{fi}$ ，其中 $i = 1$ 或 $i = 2$ 。



例 7.2

- 施加的约束条件为

$$\theta_0 = a_{10}$$

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

$$\theta_v = a_{20}$$

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

$$0 = a_{11}$$

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

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

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

这些约束条件确定了一个具有8个方程和8个未知数的线性方程组



例 7.2

- 当 $t_f = t_{f1} = t_{f2}$ 时可以得到

$$a_{10} = \theta_0$$

$$a_{11} = 0$$

- $$a_{12} = \frac{12\theta_v - 3\theta_g - 9\theta_0}{4t_f^2}$$

$$a_{13} = \frac{-8\theta_v + 3\theta_g + 5\theta_0}{4t_f^3}$$

$$a_{20} = \theta_v$$

$$a_{21} = \frac{3\theta_g - 3\theta_0}{4t_f}$$

- $$a_{22} = \frac{-12\theta_v + 6\theta_g + 6\theta_0}{4t_f^2}$$

$$a_{23} = \frac{8\theta_v - 5\theta_g - 3\theta_0}{4t_f^3}$$



高次多项式

有时用高次多项式作为路径曲线段。例如，如果要确定在路径曲线段的起始点和终止点的位置、速度和加速度，则需要用一个五次多项式进行插值，即：

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \quad (7.16)$$

其约束条件为：

$$\begin{aligned} \theta_0 &= a_0 \\ \theta_f &= a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 + a_4 t_f^4 + a_5 t_f^5 \\ \dot{\theta}_0 &= a_1 \\ \dot{\theta}_f &= a_1 + 2a_2 t_f + 3a_3 t_f^2 + 4a_4 t_f^3 + 5a_5 t_f^4 \\ \ddot{\theta}_0 &= 2a_2 \\ \ddot{\theta}_f &= 2a_2 + 6a_3 t_f + 12a_4 t_f^2 + 20a_5 t_f^3 \end{aligned} \quad (7.17)$$



高次多项式

这些约束条件确定了一个具有 6 个方程和 6 个未知数的线性方程组，其解为：

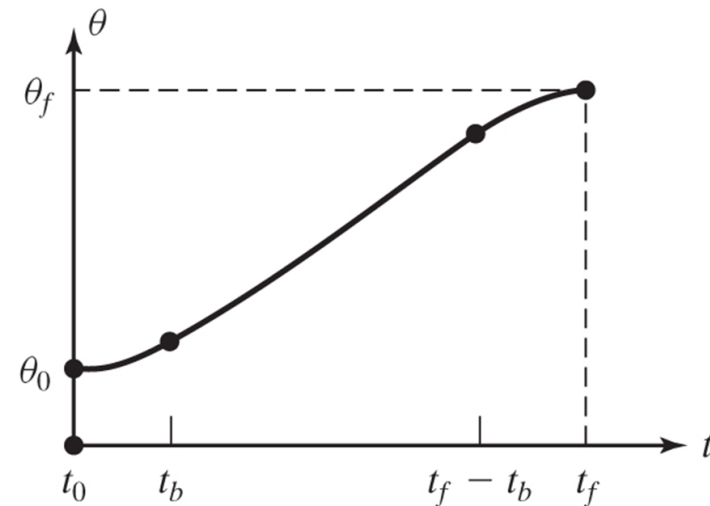
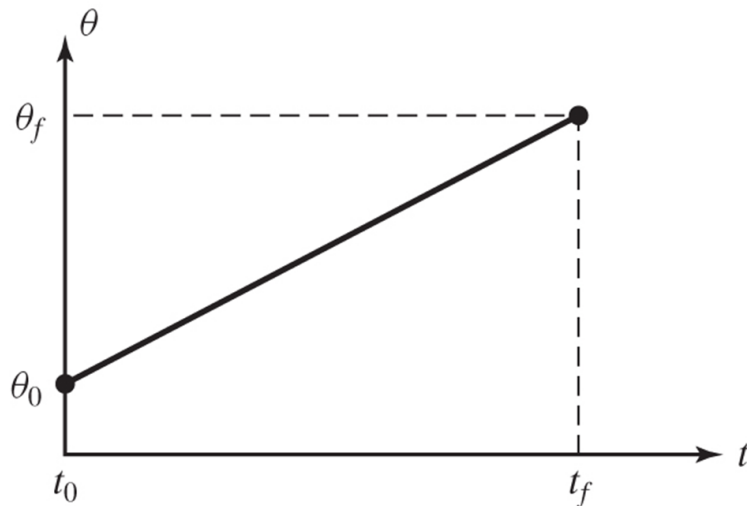
$$\begin{aligned} a &= \theta_0 \\ a_1 &= \dot{\theta}_0 \\ a_2 &= \frac{\ddot{\theta}_0}{2} \\ a_3 &= \frac{20\theta_f - 20\theta_0 - (8\dot{\theta}_f + 12\dot{\theta}_0)t_f - (3\ddot{\theta}_0 - \ddot{\theta}_f)t_f^2}{2t_f^3} \\ a_4 &= \frac{30\theta_0 - 30\theta_f + (14\dot{\theta}_f + 16\dot{\theta}_0)t_f + (3\ddot{\theta}_0 - 2\ddot{\theta}_f)t_f^2}{2t_f^4} \\ a_5 &= \frac{12\theta_f - 12\theta_0 - (6\dot{\theta}_f + 6\dot{\theta}_0)t_f - (\ddot{\theta}_0 - \ddot{\theta}_f)t_f^2}{2t_f^5} \end{aligned} \tag{7.18}$$



带有抛物线过渡的线性函数

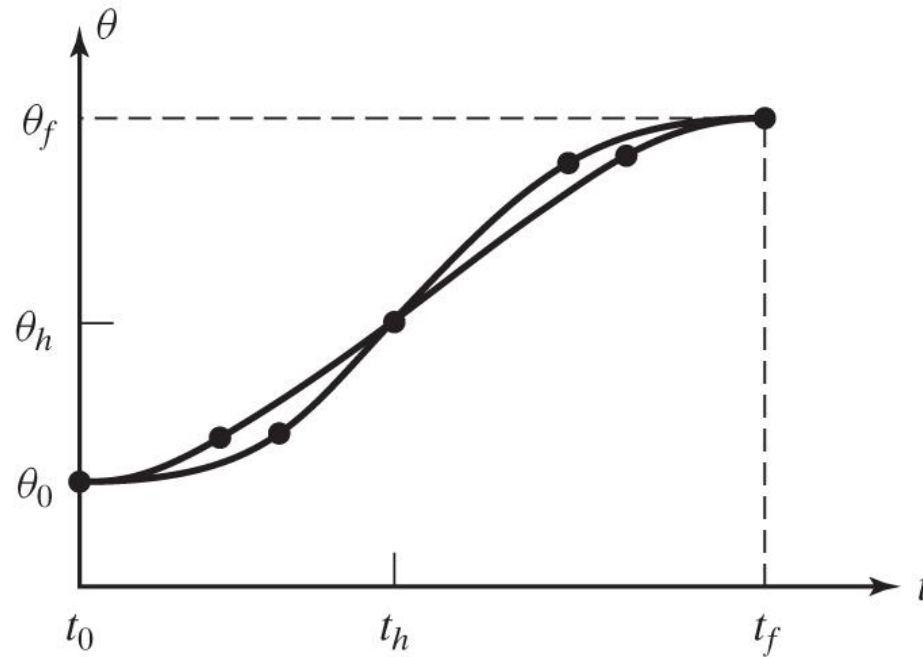
另外一种可选的路径曲线是直线，如图 7-5 所示。然而，单纯线性插值将导致在起始点和终止点的关节运动速度不连续。

为了生成一条位置和速度都连续的平滑运动轨迹，在使用线性函数进行插值时，应该在每个路径点的邻域内增加一段抛物线作为过渡，如图 7-6。



带有抛物线过渡的线性函数

为了构造这样的路径段，假设两端的抛物线区段具有相同的持续时间；因此在这两个过渡区段中采用相同的恒定加速度。如图 7-7 所示，这里存在有多个解，但是每个结果都对称于时间中点 t_h 和位置中点 θ_h 取。



带有抛物线过渡的线性函数

由于过渡区段终点的速度必须等于直线部分的速度，所以有：

$$\ddot{\theta}t_b = \frac{\theta_h - \theta_b}{t_h - t_b} \quad (7.19)$$

其中， θ_b 是过渡区段终点的 θ 值，而 $\ddot{\theta}$ 是过渡区段的加速度。 θ_b 的值由下式给出：

$$\theta_b = \theta_0 + \frac{1}{2}\ddot{\theta}t_b^2 \quad (7.20)$$



带有抛物线过渡的线性函数

联立式(7.19)和式(7.20)，且 $t = 2t_b$ ，可以得到：

$$\ddot{\theta}t_b^2 - \ddot{\theta}tt_b + (\theta_f - \theta_0) = 0 \quad (7.21)$$

其中， t 是期望的运动时间。对于任意给定的 θ_f 、 θ_0 和 t ，可通过选取满足式(7.21)的 $\ddot{\theta}$ 和 t_b 来获得任一条路径。

通常，选择好加速度 $\ddot{\theta}$ ，再计算式(7.21)，求解出相应的 t_b 。选择的加速度必须足够高，否则解将不存在。使用加速度和其他已知参数计算式(7.21)，求解 t_b ：

$$t_b = \frac{t}{2} - \frac{\sqrt{\ddot{\theta}^2 t^2 - 4\ddot{\theta}(\theta_f - \theta_0)}}{2\ddot{\theta}} \quad (7.22)$$



带有抛物线过渡的线性函数

在过渡区段使用的加速度的限制条件为：

$$\ddot{\theta} \geq \frac{4(\theta_f - \theta_0)}{t^2} \quad (7.23)$$

当式(7.23)的等号成立时，直线部分的长度缩减为零，整个路径由两个过渡区段组成，且衔接处的斜率相等。如果加速度的取值越来越大，则过渡区段的长度将随之越来越短。当处于极限状态时，即加速度无限大，路径又回到简单的直线情况。



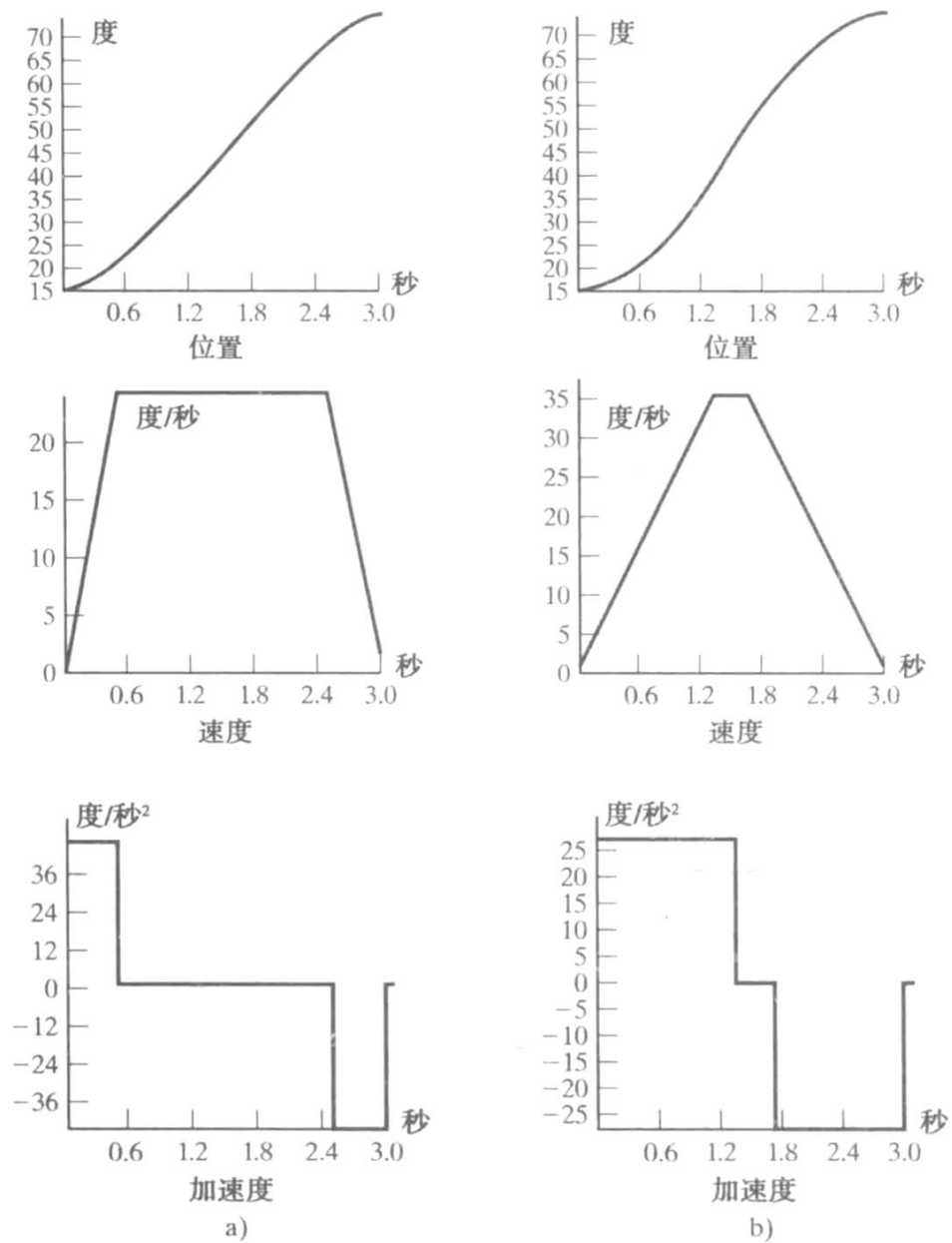


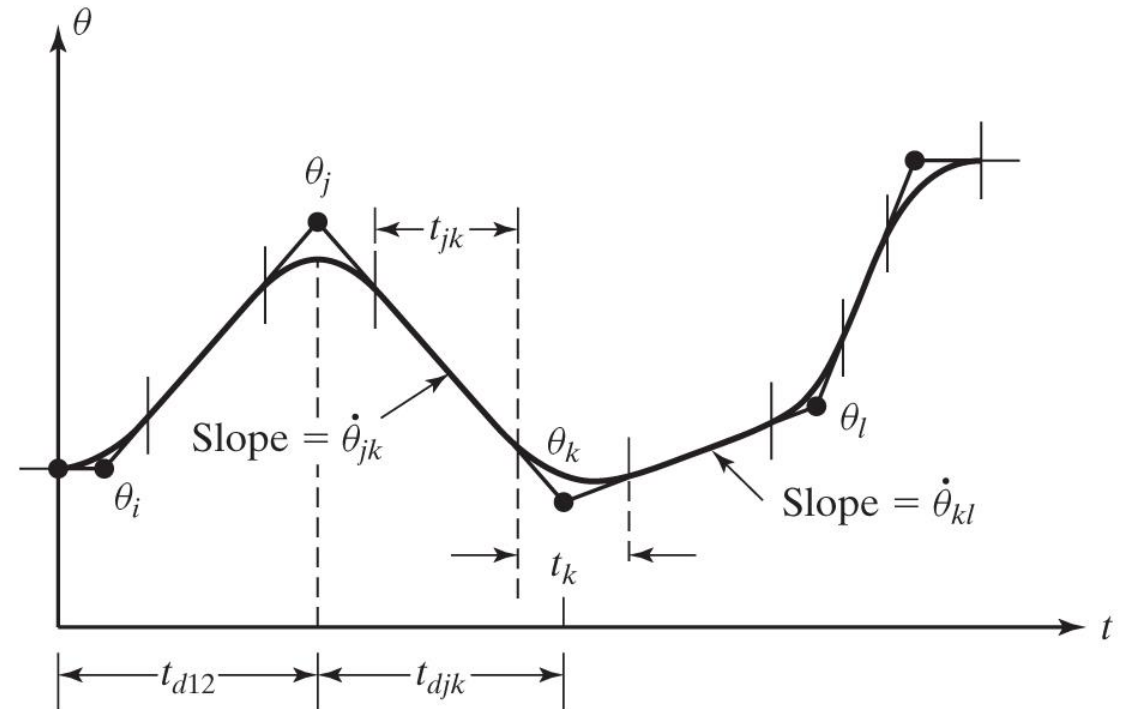
图 7-8 带有抛物线过渡的直线插值的位置、速度和加速度曲线。左边的曲线在过渡区段处的加速度高于右边的曲线



带有抛物线过渡的线性函数 用于经过中间点的路径

现在考虑带有抛物线过渡的直线路径，路径上指定了任意数量的中间点。

如图 7-9 所示，在关节空间中为某个关节 θ 的运动指定了一组中间点。每两个中间点之间使用线性函数相连，而各中间点附近使用抛物线过渡。



在这里将使用以下符号：

用 j 、 k 和 l 表示三个相邻的路径点。

位于路径点 k 处的过渡区段的时间间隔为 t_k 。

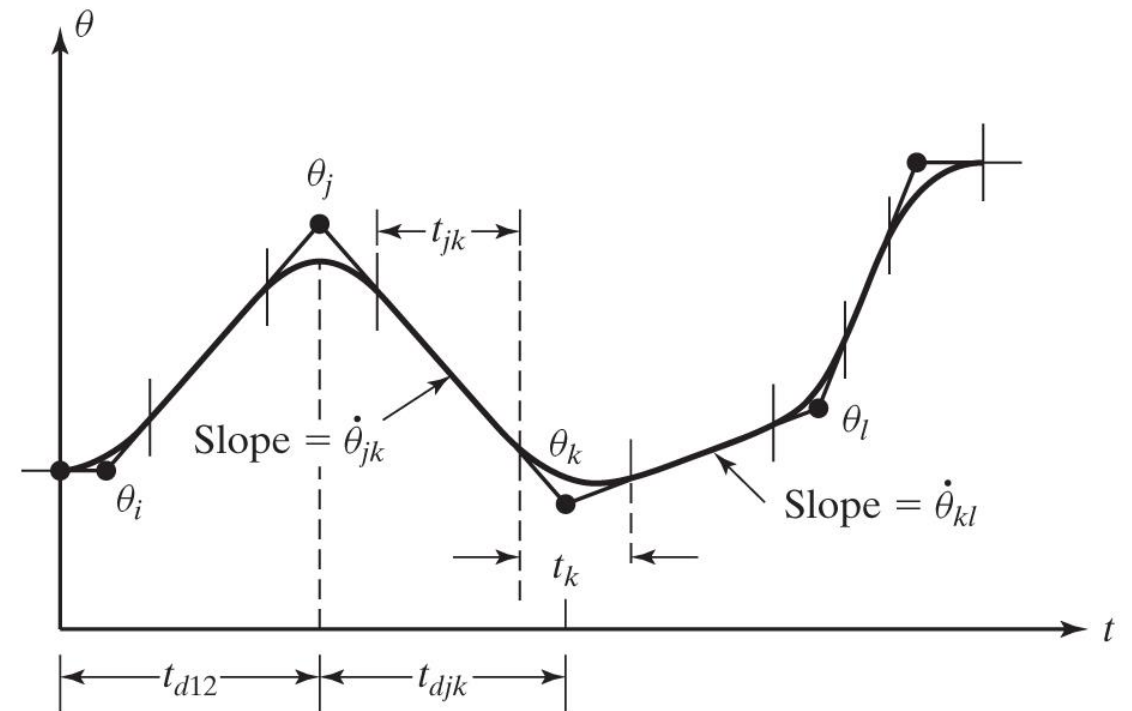
位于点 j 和 k 之间的直线部分的时间间隔为 t_{jk} 。

点 j 和 k 之间总的的时间间隔为 t_{djk} 。

直线部分的速度为 $\dot{\theta}_{jk}$ ，

而在点 j 处过渡区段的加速度为 $\ddot{\theta}_j$ ，

如图 7-9所示。



与单段路径的情形相似，存在有许多可能解，这取决于每个过渡区段的加速度值。

已知所有的路径点 θ_k 、期望的时间区间 t_{djk} 以及每个路径点处加速度的模值 $|\ddot{\theta}_k|$ ，则可计算出过渡区段的时间间隔 t_k 。对于那些内部的路径点，可直接使用下列公式计算：

$$\begin{aligned}\dot{\theta}_{jk} &= \frac{\theta_k - \theta_j}{t_{djk}} \\ \ddot{\theta}_k &= SGN(\dot{\theta}_{kl} - \dot{\theta}_{jk}) |\ddot{\theta}_k| \\ t_k &= \frac{\dot{\theta}_{kl} - \dot{\theta}_{jk}}{\ddot{\theta}_k} \\ t_{jk} &= t_{djk} - \frac{1}{2}t_j - \frac{1}{2}t_k\end{aligned}\tag{7.24}$$



对于第一个路径，令线性区段速度的两个表达式相等来求解 t_1 ：

$$\frac{\theta_2 - \theta_1}{t_{d12} - \frac{1}{2}t_1} = \ddot{\theta}_1 t_1 \quad (7.25)$$

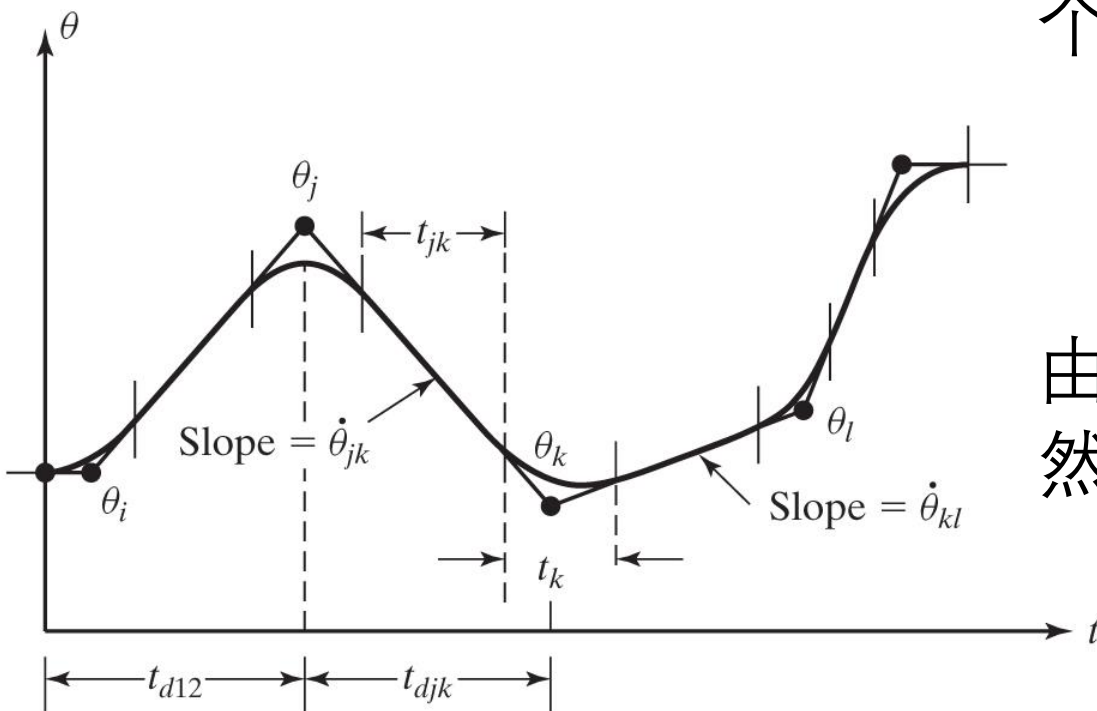
由此可解出在起始点处的过渡时间 t_1 ，然后即可解出 $\dot{\theta}_{12}$ 和 t_{12} ：

$$\ddot{\theta}_1 = \text{SGN}(\theta_2 - \theta_1) |\ddot{\theta}_1|$$

$$t_1 = t_{d12} - \sqrt{t_{d12}^2 - \frac{2(\theta_2 - \theta_1)}{\ddot{\theta}_1}} \quad (7.26)$$

$$\dot{\theta}_{12} = \frac{\theta_2 - \theta_1}{t_{d12} - \frac{1}{2}t_1}$$

$$t_{12} = t_{d12} - t_1 - \frac{1}{2}t_2$$



同样，对于最后一个路径段（连接点n-1到n），有：

$$\frac{\theta_n - \theta_{n-1}}{t_{d(n-1)n} - \frac{1}{2}t_n} = \ddot{\theta}_n t_n \quad (7.27)$$

根据上式可求出：

$$\begin{aligned} \ddot{\theta}_n &= \text{SGN}(\theta_n - \theta_{n-1}) |\ddot{\theta}_n| \\ t_n &= t_{d(n-1)n} - \sqrt{t_{d(n-1)n}^2 + \frac{2(\theta_n - \theta_{n-1})}{\ddot{\theta}_n}} \\ \dot{\theta}_{(n-1)n} &= \frac{\theta_n - \theta_{n-1}}{t_{d(n-1)n} - \frac{1}{2}t_n} \\ t_{(n-1)n} &= t_{d(n-1)n} - t_n - \frac{1}{2}t_{n-1} \end{aligned} \quad (7.28)$$



穿越点和中间点

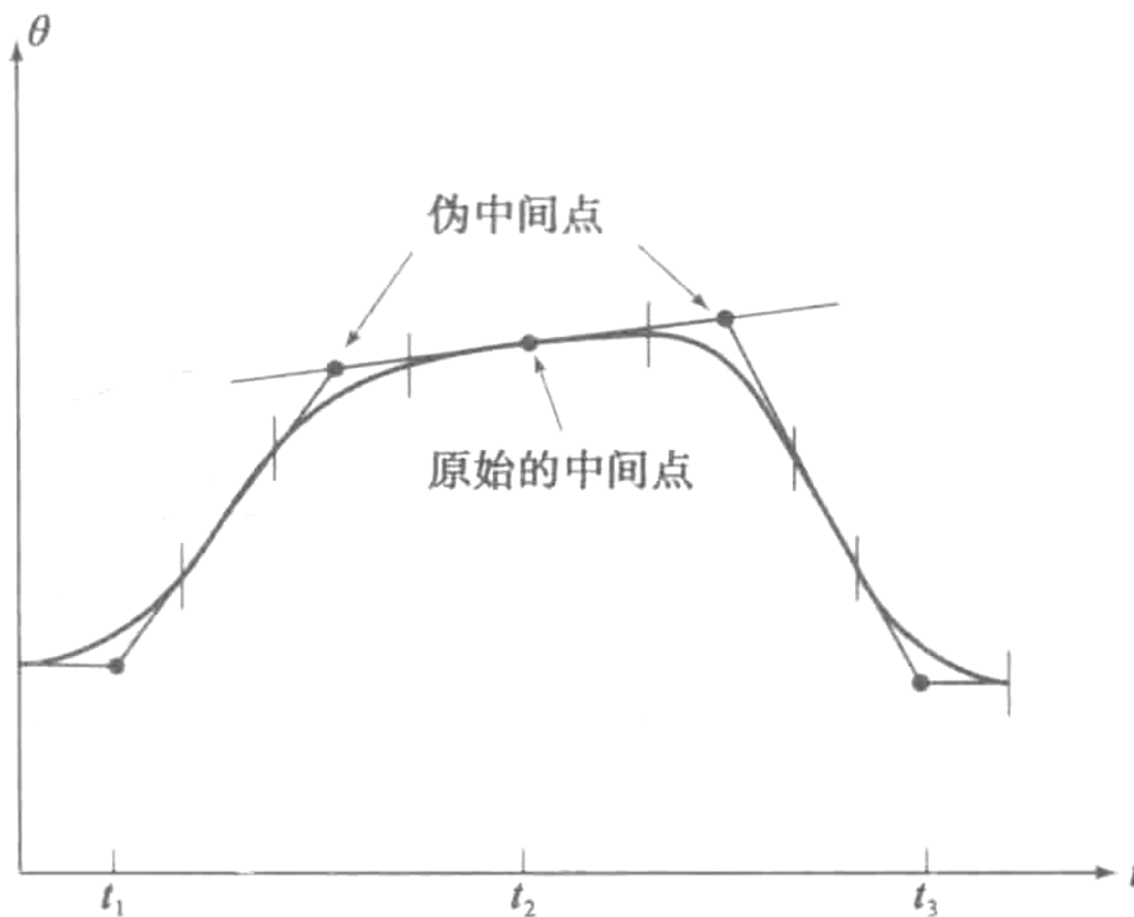


图 7-10 用伪中间点来产生一个“穿越点”



7.4 笛卡儿空间规划方法



笛卡儿直线运动

- 如果在一条直线上密集地指定许多分离的中间点，那么不管在中间点之间使用何种平滑函数进行连接，工具末端都走直线。假设能让工具在相隔较远的中间点之间走直线，这种定义和执行路径的模式被称作笛卡儿直线运动。
- 在规划和生成笛卡儿直线路径时，最好使用带有抛物线过渡的直线样条函数。在每段的直线部分，位置的三个分量都以线性的方式发生变化，所以末端执行器会沿着直线在空间运动。



笛卡儿直线运动

- 然而，如果在每个中间点将姿态定义成旋转矩阵，则无法对其分量进行线性插值，因为这样做不一定总得到有效的旋转矩阵。一个旋转矩阵必须是由正交列向量组成，而在两个正确的矩阵之间对矩阵元素进行线性插值并不能保证满足这个条件。因此，将使用另一种姿态的表示方法。



笛卡儿直线运动

- 如第 2 章所述，使用角度-轴线的表示方法可以只需三个数来定义姿态。如果把这种姿态的表示方法与 3×1 的笛卡儿位置表示方法相结合，就可得到 6×1 的笛卡儿位置与姿态的表示方法。
- 考虑一个中间点，其相对于固定坐标系的定义为 ${}^S_A T$ 。坐标系 $\{A\}$ 定义了一个中间点，在该点处末端执行器的位置由 ${}^S P_{AORG}$ 给定，姿态 ${}^S_A R$ 由给定。该旋转矩阵可被转换成“角度-轴线”的表示方法： $ROT({}^S \hat{K}_A, \theta_{SA})$ 或简写成 ${}^S K_A$ 。
- 使用符号 χ 代表该 6×1 的笛卡儿位置与姿态矢量，得到：

$${}^S \chi_A = \begin{bmatrix} {}^S \mathbf{p}_{AORG} \\ {}^S \mathbf{k}_A \end{bmatrix} \quad (7.43)$$



笛卡儿直线运动

$${}^s\chi_A = \begin{bmatrix} {}^s\mathbf{p}_{AORG} \\ {}^s\mathbf{k}_A \end{bmatrix} \quad (7.43)$$

- 其中, sK_A 为旋转量 θ_{sA} 模与单位矢量 ${}^s\hat{K}_A$ 相乘。
- 如果每个路径点均使用这种方法来表示, 那么就可以选择适当的样条函数, 使这 6 个分量随时间从一个路径点平滑地移动到下一个路径点。如果采用带有抛物线过渡的直线函数, 那么中间点之间的路径则为直线。当经过中间点时, 末端执行器的线速度与角速度将作平滑地变化。

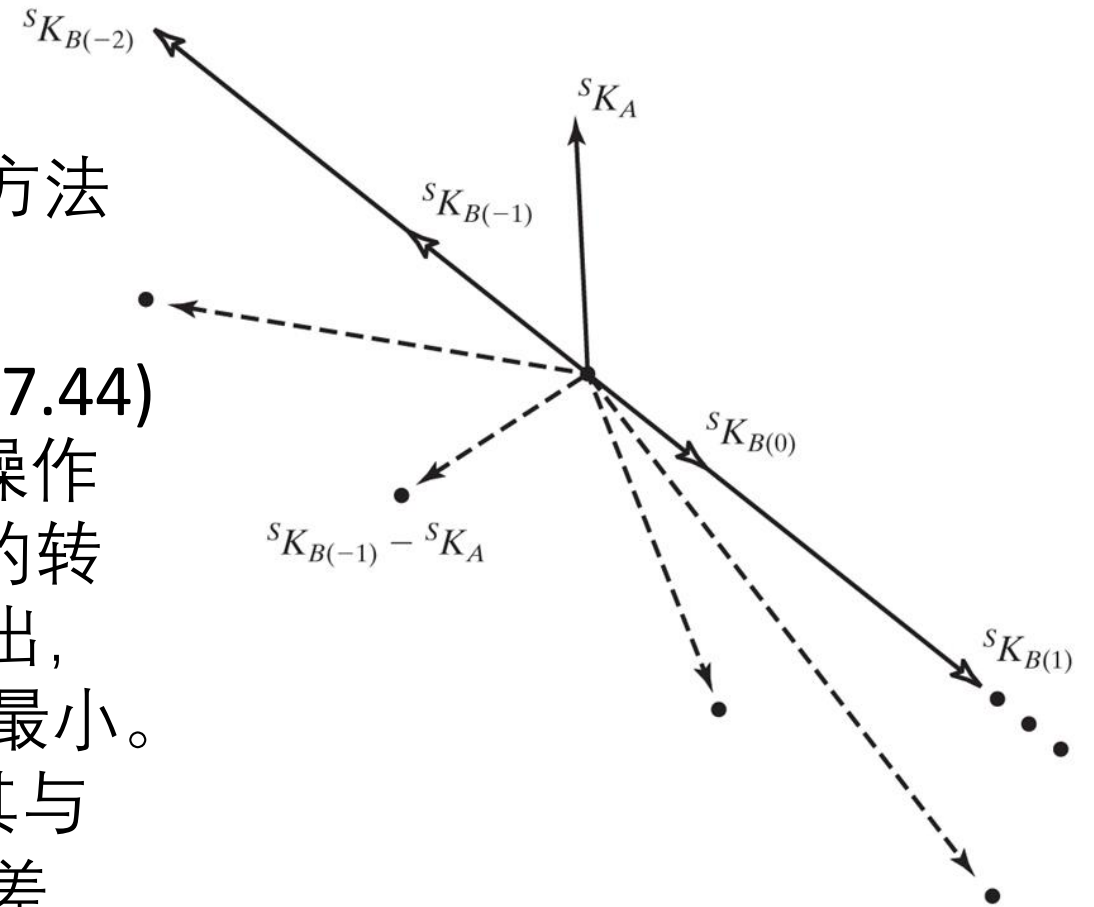


笛卡儿直线运动

- 另外需要说明的是，角度-轴线的表示方法并不唯一，即：

$$({}^s\mathbf{k}_A, \theta_{SA}) = ({}^s\mathbf{k}_A, \theta_{SA} + n360^\circ) \quad (7.44)$$

其中， n 为任意的正整数或负整数。当操作臂从中间点 $\{A\}$ 运动到中间点 $\{B\}$ 时，总的转角应取最小值。如果 $\{A\}$ 的姿态由 ${}^s\mathbf{k}_A$ 给出，则必须选择特定的 ${}^s\mathbf{k}_B$ 使得 $|{}^s\mathbf{k}_B - {}^s\mathbf{k}_A|$ 最小。例如，图 7-11 显示了 4 个可能的 ${}^s\mathbf{k}_B$ 及其与给定的 ${}^s\mathbf{k}_A$ 之间的关系。通过对矢量的差(虚线)进行比较，从而判断哪个 ${}^s\mathbf{k}_B$ 会使转动最小化——在此例中，是 ${}^s\mathbf{k}_{B(-1)}$ 。



笛卡儿直线运动

- 一旦对每个中间点选好了 χ 的6个值，即可使用前面介绍的用直线和抛物线组合的样条函数进行路径规划。只是要附加一个约束条件：每个自由度的过渡区段的时间间隔必须是相同的。这样才能保证各自由度形成的复合运动在空间形成一条直线。
- 因为各自由度的过渡区段的时间间隔相同，所以在过渡区段的加速度便不相同。因此，在指定过渡区段的时间间隔时，应该使用式（7.24）计算所需要的加速度（而不使用其他方法）。可以通过选择恰当的过渡时间以使加速度不超过上限。



7.5 笛卡儿空间规划方法



7.5 笛卡儿空间规划方法

因为笛卡儿空间中描述的路径形状与关节空间中的路径形状有连续的对应关系，所以笛卡儿空间中的路径容易出现与工作空间和奇异点有关的各种问题。

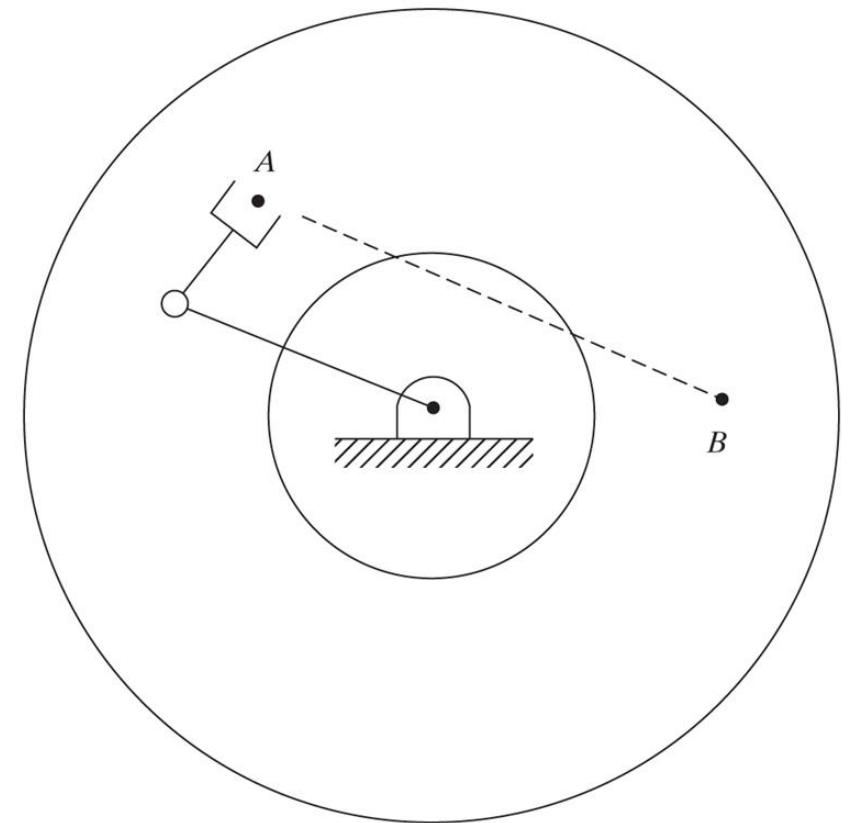
- 问题之一：无法到达中间点
- 问题之二：在奇异点附近关节速度增大
- 问题之三：起始点和终止点有不同的解



问题之一：无法到达中间点

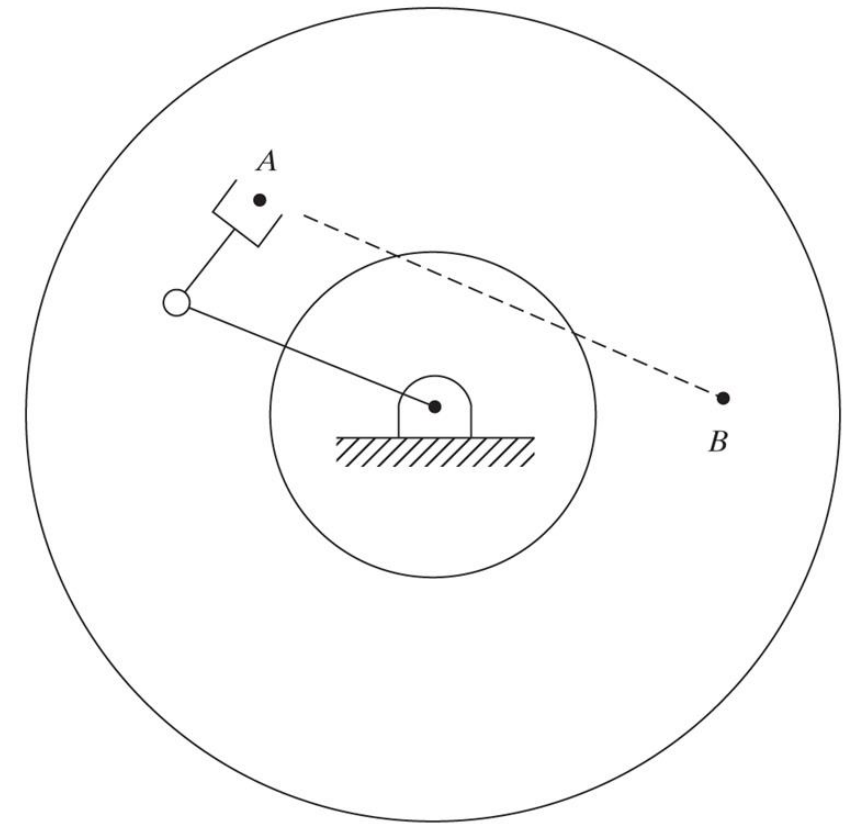
尽管操作臂的起始点和终止点都在其工作空间内部，但是很有可能在连接这两点的直线上有某些点不在工作空间中。

例如图 7-12 所示的平面两杆机器人及其工作空间。在此例中，连杆 2 比连杆 1 短，所以在工作空间的中间存在一个孔，其半径为两连杆长度之差。起始点 A 和终止点 B 被画在工作空间中。



问题之一：无法到达中间点

在关节空间中规划轨迹从 A 运动到 B 没有问题，但是如果试图在笛卡儿空间中沿直线运动，将无法到达路径上的某些中间点。该例表明了在某些情况下关节空间中的路径容易实现，而笛卡儿空间中的直线路径将无法实现。



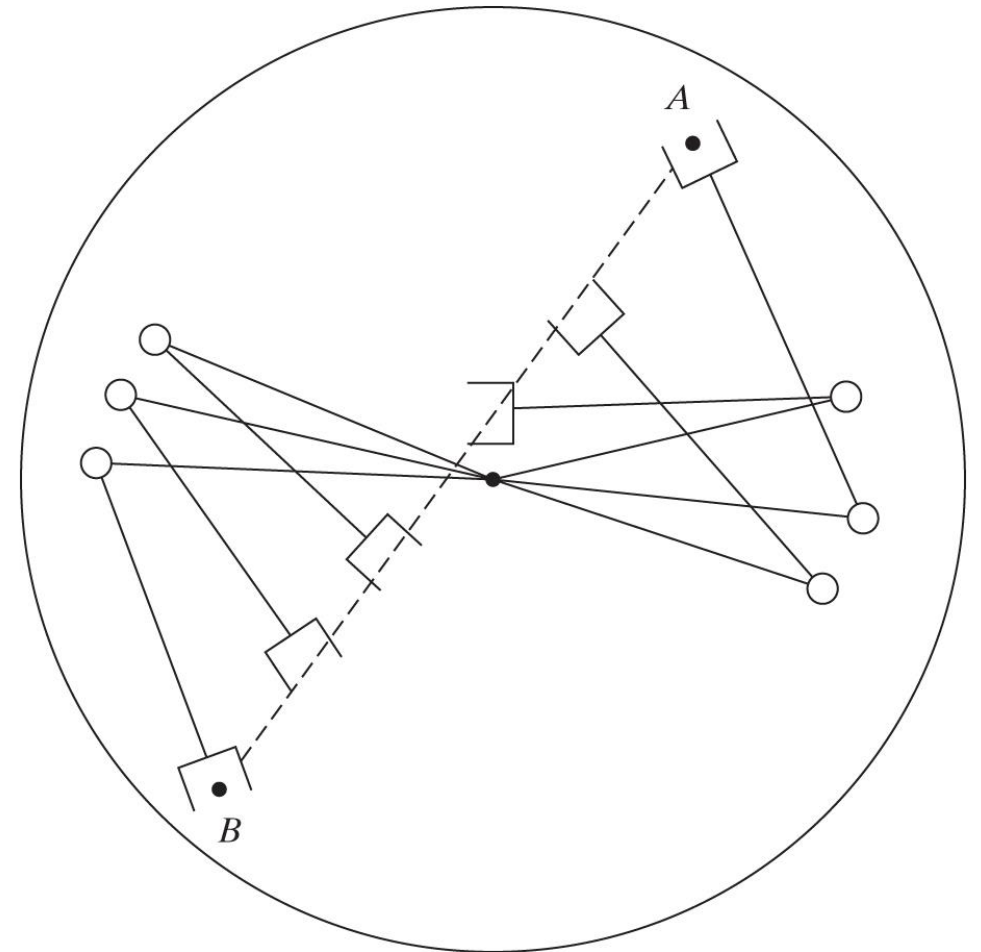
问题之二：在奇异点附近关节速度增大

- 从第 5 章可以看到，在操作臂的工作空间中存在着某些位置，在这些位置处无法用有限的关节速度来实现末端执行器在笛卡儿空间中的期望速度。因此，有某些路径（在笛卡儿空间中描述）是操作臂所无法执行的。
- 例如，如果一个操作臂沿笛卡儿直线路径接近机构的某个奇异位形时，则机器人的一个或多个关节速度可能激增至无穷大。由于机构所容许的轨迹速度是有上限的，因此这通常将导致操作臂偏离期望的路径。



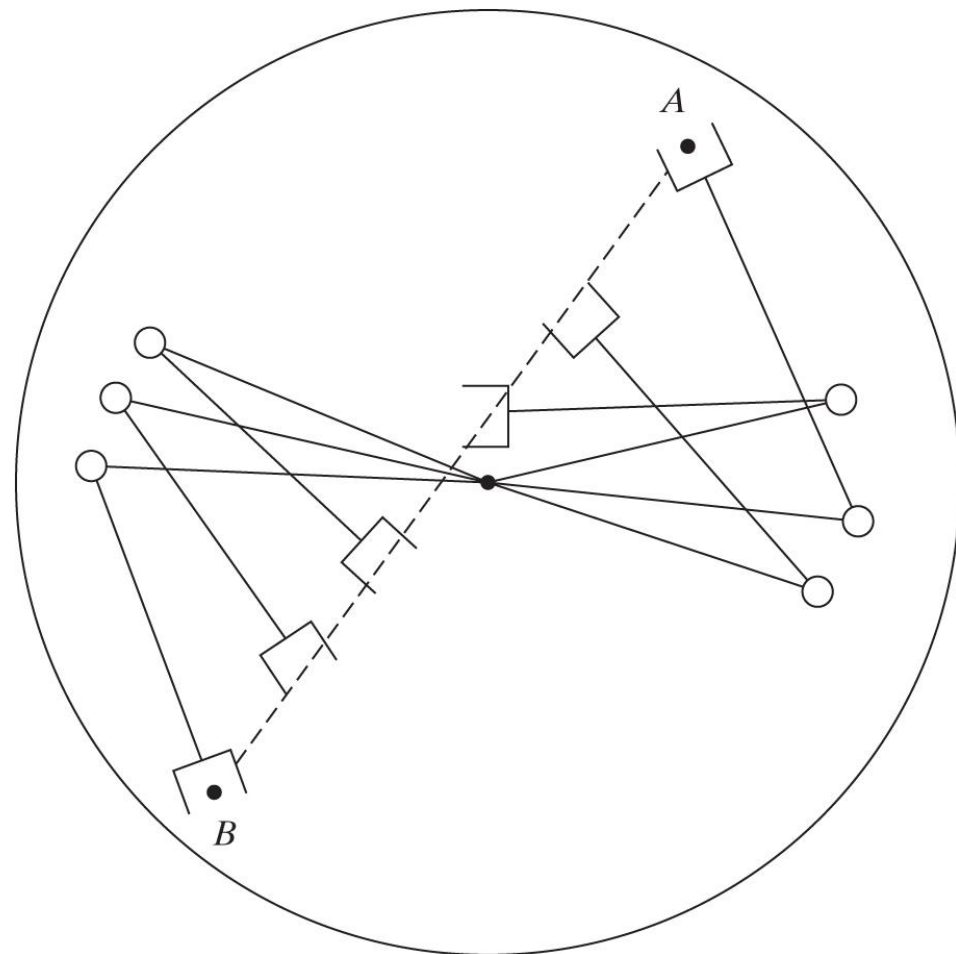
问题之二：在奇异点附近关节速度增大

例如，图 7- 13 给出了一个平面两杆（两杆长相同）机器人，从 A 点沿着路径运动到 B 点。期望的轨迹是使操作臂末端以恒定的速度作直线运动。图中画出了操作臂在运动过程中的多个中间位置以便于观察其运动。可见，路径上的所有点都可以到达。但是当机器人经过路径的中间部分时，关节 1 的速度非常高。路径越接近关节 1 的轴线，关节 1 的速度就变得越大。



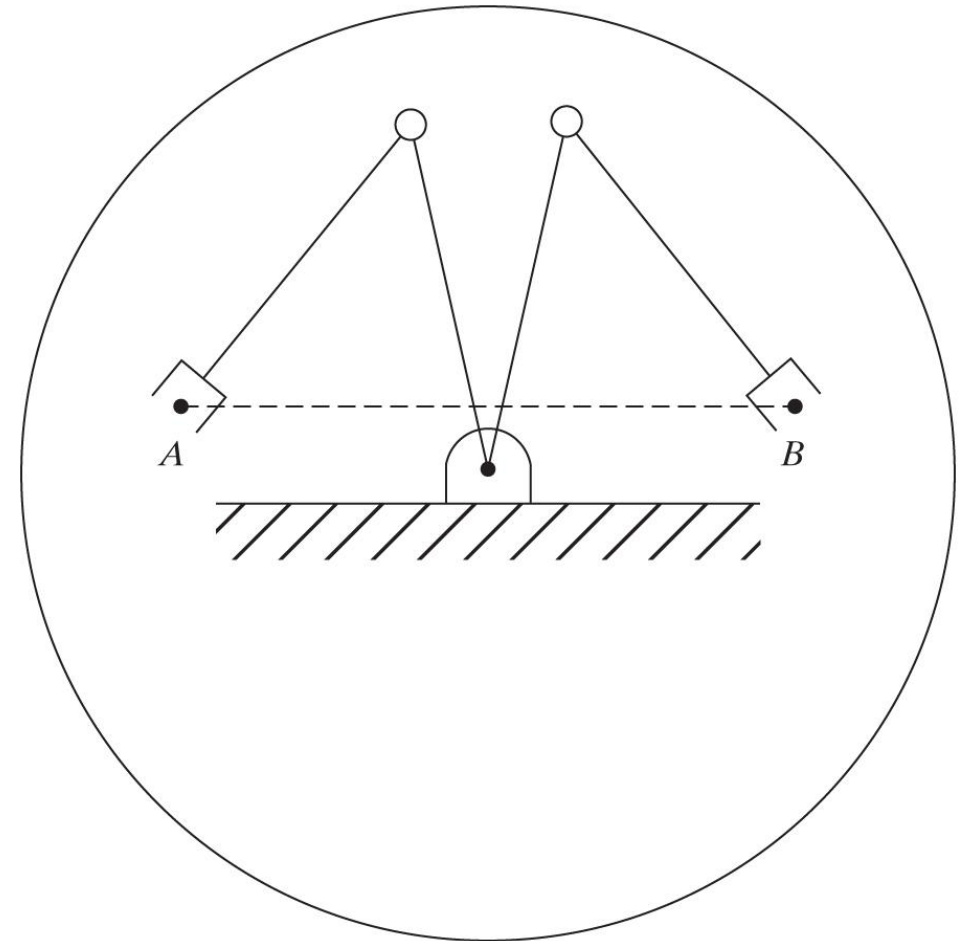
问题之二：在奇异点附近关节速度增大

一个解决办法是：降低路径的整体运动速度，以保证所有关节的速度不超出其容许范围。这样做可能会使路径的运动速度不再是匀速，但是路径仍然保持为直线。



问题之三：起始点和终止点有不同的解

如图 7-14 所示，平面两杆机器人的两个杆长相等，但是关节存在约束，这使机器人到达空间给定点的解的数量减少。尤其是当机器人的终止点不能使用与起始点相同的解到达时，就会出现这个问题。操作臂可以使用某些解到达所有的路径点，但并非任何解都可以到达。为此，操作臂的路径规划系统无须使机器人沿路径运动就应该检测到这种问题，并向用户报错。



7.5 笛卡儿空间规划方法

- 为了应对在笛卡儿空间轨迹规划中存在的这些问题。大多数工业机器人的控制系统都具有关节空间和笛卡儿空间两种轨迹生成方法。
- 由于使用笛卡儿空间路径着一定的难度，所以默认使用关节空间路径，只有在必要时，才使用笛卡儿空间路径规划方法。



7.6 路径的实时生成



关节空间路径的生成

按照前面在 7.3 节介绍的某种样条方法生成的路径，其结果都是有关各个路径段的一组数据。这些数据被路径生成器用来实时计算 θ 、 $\dot{\theta}$ 和 $\ddot{\theta}$ 。

对于三次样条，路径生成器只需随 t 的变化不断计算式(7.3)。当到达路径段的终点时，调用新路径段的三次样条系数，重新把 t 置成零，继续生成路径。



关节空间路径的生成

对于带抛物线过渡的直线样条曲线，每次更新轨迹时，应首先检测时间 t 的值，以判断当前是处在路径段的直线区段还是抛物线过渡区段。

在直线区段，对每个关节的轨迹计算如下：

$$\begin{aligned}\theta &= \theta_j + \dot{\theta}_{jk} t \\ \dot{\theta} &= \dot{\theta}_{jk} \\ \ddot{\theta} &= 0\end{aligned}\tag{7.45}$$

其中， t 是自第 j 个中间点算起的时间， $\dot{\theta}_{jk}$ 的值在轨迹规划时按式 (7.24) 计算。



关节空间路径的生成

在过渡区段，对各关节的轨迹计算如下：

$$t_{\text{inb}} = t - \left(\frac{1}{2} t_j + t_{jk} \right)$$
$$\theta = \theta_j + \dot{\theta}_{jk} t + \frac{1}{2} \ddot{\theta}_k t_{\text{inb}}^2 \quad (7.46)$$
$$\dot{\theta} = \dot{\theta}_{jk} + \ddot{\theta}_k t_{\text{inb}}$$
$$\ddot{\theta} = \ddot{\theta}_k$$

其中， t 、 t_{jk} 、 $\dot{\theta}_{jk}$ 、 $\ddot{\theta}_k$ 的值在轨迹规划时根据式(7.24)(7.28)计算。当进入一个新的直线区段时，将 t 重置成 $0.5t_k$ 继续计算，直到计算出所有表示路径段的数据集合。



笛卡儿空间路径的生成

在 7.4 节中已经使用路径生成器生成了带有抛物线过渡的直线样条曲线。但是，计算得到的数值表示的是笛卡儿空间的位置和姿态而不是关节变量值，所以这里使用符号 x 来表示笛卡儿位姿矢量的一个分量，并重写式(7.45)和式(7.46)。

在曲线的直线区段， x 中的每个自由度按下式计算下：

$$\begin{aligned}x &= x_j + \dot{x}_{jk}t \\ \dot{x} &= \dot{x}_{jk} \\ \ddot{x} &= 0\end{aligned}\tag{7.47}$$

其中， t 是自第 j 个中间点算起的时间，而 \dot{x}_{jk} 是在轨迹规划过程中由类似于式 (7.24) 的方程求出的。



笛卡儿空间路径的生成

在过渡区段中，每个自由度的轨迹计算如下：

$$t_{\text{inb}} = t - \left(\frac{1}{2} t_j + t_{jk} \right) \quad (7.48)$$

$$r = x_j + \dot{x}_{jk} t + \frac{1}{2} \ddot{x}_k t_{\text{inb}}^2$$

$$\dot{x} = \dot{x}_{jk} + \ddot{x}_k t_{\text{inb}}$$

$$\ddot{x} = \ddot{x}_k$$

其中， t 、 t_{jk} 、 \dot{x}_{jk} 、 \ddot{x}_k 的值在轨迹规划过程中计算出，与关节空间的情况完全相同。



笛卡儿空间路径的生成

- 最后，这些笛卡儿空间的路径 (x, \dot{x}, \ddot{x}) 必须被等价变换为关节空间的变量。对此，可以通过运动学反解得到关节位移；用逆雅可比矩阵计算关节速度，用逆雅可比矩阵及其导数计算角加速度。



笛卡儿空间路径的生成

在实际中经常使用的简单方法为：根据路径更新的速率，将 x 转化成等价的坐标系表示： ${}^S_G T$ 。然后使用 SOLVE 算法(见 4.8 节)求出所需的关节角矢量 θ 。然后用数值微分计算出 $\dot{\theta}$ 和 $\ddot{\theta}$ 。于是，算法为：

$$\begin{aligned} x &\rightarrow {}^S_G T \\ \theta(t) &= \text{SOLVE}({}^S_G T) \\ \dot{\theta}(t) &= \frac{\theta(t) - \theta(t - \delta t)}{2} \\ \ddot{\theta}(t) &= \frac{\dot{\theta}(t) - \dot{\theta}(t - \delta t)}{t} \end{aligned} \tag{7.49}$$

然后把 θ 、 $\dot{\theta}$ 和 $\ddot{\theta}$ 输入给操作臂的控制系统。



7.7 使用机器人编程语言描述路径



7.7 使用机器人编程语言描述路径

- 在第 12 章, 会更详细地讨论机器人编程语言。
- 在此, 将说明在本章讨论的多种路径如何用机器人编程语言表达。
- 在这些例子中, 将使用 AL 语言, 一种由斯坦福大学开发的机器人编程语言。



7.7 使用机器人编程语言描述路径

符号 A、B、C、D 表示 AL 语言例子中的“位姿”。它们代表了路径点，假定这些路径点已被示教或文本输入进了系统。

假定操作臂从位置 A 开始运动。在关节空间使操作臂沿着带有抛物线过渡的直线路径运动，就写入：

- move ARM to C with duration = 3 * seconds;



以直线移动到相同的位置和姿态，就写入：

- **move ARM to C linearly with duration= 3 * seconds;**

其中，关键词“linearly”表示使用笛卡儿直线运动。

如果时间区间并不重要，用户可忽略此定义，而系统将采用默认的速度，即：

- **move ARM to C ;**

如果加入一个中间点，则可写为：

- **move ARM to C via B;**

或者加入一组中间点：

- **move ARM to C via B, A, D;**



注意到在

- move ARM to C via B with duration= 6 * seconds;

中给出了整个运动的时间区间。系统自行决定如何在整个运动的时间区间中分配直线和抛物线区段的时间。

在 AL 中，可以为某个区段单独指定时间区间，例如：

- move ARM to C via B where duration= 3 * seconds;

这样，走向点 B 的第一个区段将有 3 秒钟。



7.8 使用动力学模型的路径规划



7.8 使用动力学模型的路径规划

- 通常，在规划路径时，在每个过渡点处使用默认的或最大的加速度。实际上，操作臂在任何时刻的最大加速度与其动力学性能和驱动器能力有关。绝大多数驱动器的特征并不是由固定的最大扭矩或加速度来描述的，而是由它的力矩-速度曲线来描述的。
- 在规划路径时，如果假定每个关节或每个自由度具有最大的加速度，则意味着作了过度的简化。实际上，为了避免超出驱动设备的承受能力，必须保守地选取最大加速度值。因而，采用本章介绍的轨迹规划方法不能充分利用操作臂的速度性能。



7.8 使用动力学模型的路径规划

- 存在下列问题：如果已知末端执行器的期望空间路径，如何使操作臂以最少的时间到达目标点（将空间路径的描述转变成轨迹）。
- 这些问题已由数值计算方法解决。这些方法考虑了操作臂动力学和驱动器的速度-力矩约束曲线。



7.9 无碰撞路径规划



7.9 无碰撞路径规划

- 如果能简单地把操作臂的期望目标点告诉机器人系统，让系统自行决定所需中间点的数量和位置以使操作臂到达目标而不碰到任何障碍，这将是一件非常好的事情。
- 为此，必须要有操作臂的模型、工作区域以及区域内所有潜在的障碍。如果在该区域中还有一个操作臂在工作，那么每个机械臂都应该把对方视作障碍。



7.9 无碰撞路径规划

- 无碰撞路径规划系统尚未在工业上获得商业应用。此领域的研究形成了两个主要的相互竞争的技术以及二者的若干变型和综合技术。
- 一种解决问题的方法是做出一个用于描述无碰撞空间的连接图，然后在该图中搜索无碰撞路径。然而，这些技术的复杂性随着操作臂的关节数量增加而按指数幅度增加。



7.9 无碰撞路径规划

- 第二种方法是在障碍的周围设置人工势场，以使操作臂在被拉向在目标点处设置的人工吸引极点的同时避免碰上障碍物。然而，这些方法通常只考虑了局部空间，因而会在人工势场区域的某个局部极小区被“卡”住。



问题？

