

机器人学基础：运动学与动力学

Bruno Siciliano, Oussama Khatib (Eds.)

2016

1 简介

本文译自《Springer Handbook of Robotics - 2016》的第二章和第三章。

术语与缩写表：

术语	说明	术语	说明
3-D	三维	6-D	六维
C	圆柱形关节 (cylindrical joint)	H	螺旋形关节 (helical joint)
P	棱柱形关节 (prismatic joint)	R	旋转关节 (revolute joint)
U	万向节 (universal joint)	6-DOF	六自由度
ABA	articulated-body algorithm 铰接体算法	CRBA	composite-rigid-body algorithm 复合刚体算法
DOF	degree of freedom 自由度	JPL	Jet Propulsion Laboratory 喷气推进实验室
JSIM	joint-space inertia matrix 关节空间惯量矩阵		Lagrangian formulation 拉格朗日公式
OSIM	operational-space inertia matrix 操作空间惯量矩阵	RNEA	recursive Newton-Euler algorithm 递归牛顿-欧拉算法
position	位置	velocity	速度
acceleration	加速度	forces	力
moment	力矩	torques	扭矩
momentum	动量	impulse	冲量
dynamics	动力学	kinematics	运动学
screw	旋量	twist	运动旋量
wrench	动力旋量	energy	能量

2 运动学

Kenneth J. Waldron, James Schmiedeler

运动学涉及到机器人机构中机体的运动，而不考虑引起运动的力/扭矩。由于机器人机构本质上是为运动而设计的，因此运动学是机器人设计、分析、控制和模拟的最基本方面。机器人学界一直致力于有效地应用位置和方向的不同表示及其相对于时间的导数来解决基本运动学问题。

本章将介绍机体在空间中的位置和方向的最有用的表示，机器人机构中最常见的关节运动学，以及表示机器人机构几何图形的方便约定。这些代表性工具将用于计算机器人机构的工作空间 (*workspace*)、正向与逆向运动学 (*forward and inverse kinematics*)、正向与逆向瞬时运动学 (*forward and inverse instantaneous kinematics*) 以及静态动力旋量变换 (*static wrench transmission*)。为简洁起见，重点介绍适用于开链机构的算法。

本章的目的是向读者提供表格形式的通用工具，以及更广泛的算法概述，这些算法可以一起应用于解决与特定机器人机构相关的运动学问题。

2.1 概述

除非另有明确说明，否则机器人机构是由关节连接的刚体系统。刚体在空间中的位置和方向统称为位姿 (*pose*)。因此，机器人运动学描述位姿、速度、加速度，以及组成机构的机体的所有的更高阶导数。由于运动学不涉及诱导运动的力/扭矩，因此本章重点描述位姿和速度。这些描述是动力学 (第 3 章)、运动规划 (第 7 章) 和运动控制 (第 8 章) 算法的基本要素。

在机体系统可以连接的许多可能的拓扑结构中，有两种在机器人学中特别重要，即串联链和完全并联机构。串联链是一个刚体系统，其中每个成员都与其他两个成员相连，除了第一个和最后一个成员只与一个其他成员相连。完全并联机构是指有两个成员通过其它成员和关节的多个链条连接在一起。在实践中，这些链中的每一个成员本身往往就是一个串联链。本章几乎只关注适用于串联链的算法。并行机制将在第 18 章中更详细地论述。另一个重要的拓扑结构是树状结构，它与串联链类似，没有闭环，但与串联链不同的是，每个成员可能有多个成员与之相连，形成多个分支 (*branches*)。串联链实际上只是一个没有分支的树形结构的特例。第 3 章将更深入地讨论树形结构。

2.2 位置与方向的表示

空间的刚体运动学可以看作是对代表一个机体的位姿的一种比较研究不同的方式。平移和旋转，组合起来称为刚体位移，也可以用这些表示形式表达。没有一种方法适合所有目的，但是可以适当地利用每种方法的优点以促进不同问题的解决。

在欧几里得空间中定位机体所需的最小坐标数为 6。空间姿态的许多表示都采用了超多坐标的集合，其中坐标之间存在辅助关系。独立辅助关系数是集合中坐标数量与 6 坐标之间的差值。

本章及其后续章节经常使用坐标帧 (*coordinate frames*) 或简称帧 (*frames*)。坐标帧 i 包含一个原点，标志为 O_i ，以及一个相互正交的三元基向量，标志为 $\begin{pmatrix} \hat{x}_i & \hat{y}_i & \hat{z}_i \end{pmatrix}$ ，都固定在一个特定的机体中。一个机体的位姿总是相对于另一个机体来表达，所以它可以表达为一个坐标帧相对于另一个坐标帧的位姿。类似地，刚体位移可以表达为两个坐标帧之间的位移，其中一个坐标帧可以称为移动的 (*moving*)，而另一个坐标帧可以称为固定的 (*fixed*)。这表明观测者定位于固定帧中的静止位置，而不存在绝对固定帧。

2.2.1 位置与位移

坐标帧 i 相对于坐标帧 j 的原点位置可用 3×1 向量标志

$${}^j\mathbf{p}_i = \begin{pmatrix} {}^j p_i^x \\ {}^j p_i^y \\ {}^j p_i^z \end{pmatrix}.$$

该向量的分量是在 j 帧中 O_i 的笛卡尔坐标, 即向量 ${}^j\mathbf{p}_i$ 在相应轴上的投影。向量分量也可以表达为在 j 帧中 O_i 的球坐标或柱坐标。这种表示方法对于机器人机构包括球形和圆柱形关节的分析具有优势。

平移是一种位移, 在这种位移中, 刚体中没有一个点保持其初始位置, 并且刚体中的所有直线都保持与其初始方向平行。(点和线不一定包含在有限刚体的边界内, 相反, 空间中的任何点或线都可以视为刚性固定在机体中。) 一个机体在空间中的平移可以通过平移之前和之后的位置组合来表示。反过来说, 一个机体的位置可以表示为将机体从起始位置 (其中固定到机体的坐标帧与固定坐标帧重合) 变换到当前位置 (其中前面两个帧不再重合) 的平移。因此, 任意位置的表示都可以用于创建位移的表示, 反之亦然。

2.2.2 方向与旋转

方向表征的广度显著大于位置表征的广度。本节不包括详尽的总结, 但重点介绍最常用于机器人机构的表示。

旋转是一种位移, 其中刚体中至少有一个点保持在其初始位置, 并且机体中并非所有直线都保持与其初始方向平行。例如, 一个在圆形轨道中的机体绕轴旋转, 该轴穿过其圆形路径的中心, 旋转轴上的每个点都是机体中保持初始位置的点。与位置和平移的情况一样, 任意方向表示都可以用于创建旋转表示, 反之亦然。

旋转矩阵

坐标帧 i 相对于坐标帧 j 的方向可以通过将基向量 $(\hat{\mathbf{x}}_i \hat{\mathbf{y}}_i \hat{\mathbf{z}}_i)$ 表达为基向量 $(\hat{\mathbf{x}}_j \hat{\mathbf{y}}_j \hat{\mathbf{z}}_j)$ 的项来标志。这产生 $({}^j\hat{\mathbf{x}}^i {}^j\hat{\mathbf{y}}^i {}^j\hat{\mathbf{z}}^i)$, 当将其写为 3×3 矩阵时, 称为旋转矩阵。 ${}^j\mathbf{R}_i$ 的分量是两个坐标帧的基向量的点积。

$${}^j\mathbf{R}_i = \begin{pmatrix} \hat{\mathbf{x}}_i \cdot \hat{\mathbf{x}}_j & \hat{\mathbf{y}}_i \cdot \hat{\mathbf{x}}_j & \hat{\mathbf{z}}_i \cdot \hat{\mathbf{x}}_j \\ \hat{\mathbf{x}}_i \cdot \hat{\mathbf{y}}_j & \hat{\mathbf{y}}_i \cdot \hat{\mathbf{y}}_j & \hat{\mathbf{z}}_i \cdot \hat{\mathbf{y}}_j \\ \hat{\mathbf{x}}_i \cdot \hat{\mathbf{z}}_j & \hat{\mathbf{y}}_i \cdot \hat{\mathbf{z}}_j & \hat{\mathbf{z}}_i \cdot \hat{\mathbf{z}}_j \end{pmatrix}. \quad (2.1)$$

因为基向量是单位向量, 并且任意两个单位向量的点积是它们之间的夹角的余弦, 所以这些分量通常被称为方向余弦。

帧 i 围绕 $\hat{\mathbf{z}}_j$ 轴通过一个角度 θ 的基本旋转为

$$\mathbf{R}_Z(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (2.2)$$

而围绕 $\hat{\mathbf{y}}_j$ 轴的相同旋转为

表 2.1: 对于各种方向表示的等效旋转矩阵, 其中的缩写为 $c_\theta := \cos \theta$, $s_\theta := \sin \theta$, 以及 $v_\theta := 1 - \cos \theta$

$Z - Y - X$ 欧拉角 $(\alpha, \beta, \gamma)^T$	${}^j\mathbf{R}_i = \begin{pmatrix} c_\alpha c_\beta & c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma & c_\alpha s_\beta c_\gamma + s_\alpha s_\gamma \\ s_\alpha c_\beta & s_\alpha s_\beta s_\gamma + c_\alpha c_\gamma & s_\alpha s_\beta c_\gamma - c_\alpha s_\gamma \\ -s_\beta & c_\beta s_\gamma & c_\beta c_\gamma \end{pmatrix}$
$X - Y - Z$ 固定角 $(\psi, \theta, \phi)^T$	${}^j\mathbf{R}_i = \begin{pmatrix} c_\phi c_\theta & c_\phi s_\theta s_\psi - s_\phi c_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ s_\phi c_\theta & s_\phi s_\theta s_\psi + c_\phi c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi \\ -s_\theta & c_\theta s_\psi & c_\theta c_\psi \end{pmatrix}$
角-轴 $\theta \hat{\mathbf{w}}$	${}^j\mathbf{R}_i = \begin{pmatrix} w_x^2 v_\theta + c_\theta & w_x w_y v_\theta - w_z s_\theta & w_x w_z v_\theta + w_y s_\theta \\ w_x w_y v_\theta + w_z s_\theta & w_y^2 v_\theta + c_\theta & w_y w_z v_\theta - w_x s_\theta \\ w_x w_z v_\theta - w_y s_\theta & w_y w_z v_\theta + w_x s_\theta & w_z^2 v_\theta + c_\theta \end{pmatrix}$
单位四元数 $(\epsilon_0 \epsilon_1 \epsilon_2 \epsilon_3)^T$	${}^j\mathbf{R}_i = \begin{pmatrix} 1 - 2(\epsilon_2^2 + \epsilon_3^2) & 2(\epsilon_1 \epsilon_2 - \epsilon_0 \epsilon_3) & 2(\epsilon_1 \epsilon_3 + \epsilon_0 \epsilon_2) \\ 2(\epsilon_1 \epsilon_2 + \epsilon_0 \epsilon_3) & 1 - 2(\epsilon_1^2 + \epsilon_3^2) & 2(\epsilon_2 \epsilon_3 - \epsilon_0 \epsilon_1) \\ 2(\epsilon_1 \epsilon_3 - \epsilon_0 \epsilon_2) & 2(\epsilon_2 \epsilon_3 + \epsilon_0 \epsilon_1) & 1 - 2(\epsilon_1^2 + \epsilon_2^2) \end{pmatrix}$

$$\mathbf{R}_Y(\theta) = \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix}, \quad (2.3)$$

并且围绕 $\hat{\mathbf{x}}_j$ 轴的旋转为

$$\mathbf{R}_X(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{pmatrix} \quad (2.4)$$

通过简单的矩阵乘法组合旋转矩阵, 使得帧 i 相对于帧 k 的方向可以表达为

$${}^k\mathbf{R}_i = {}^k\mathbf{R}_j {}^j\mathbf{R}_i.$$

旋转矩阵 ${}^j\mathbf{R}_i$ 包含九个元素, 而定义机体在空间中的方向只需要三个参数。因此, 矩阵元素之间存在六种辅助关系。因为坐标帧 i 的基向量是相互正交的, 坐标帧 j 的基向量也是如此, 因此由这些向量的点积形成的 ${}^j\mathbf{R}_i$ 的列也是相互正交的。一个由相互正交向量构成的矩阵称为正交矩阵, 其特性是它的逆矩阵仅为其转置矩阵。此特性提供六个辅助关系。三个要求列向量具有单位长度, 三个要求列向量相互正交。另外, 旋转矩阵的正交性可以通过在逆序中考虑这些帧来看到。坐标帧 j 相对于坐标帧 i 的方向是旋转矩阵 ${}^i\mathbf{R}_j$, 其行显然是矩阵 ${}^j\mathbf{R}_i$ 的列。

综上所述, ${}^j\mathbf{R}_i$ 是旋转矩阵, 其将在坐标帧 i 中表达的向量变换为在坐标帧 j 中表达的向量。它提供了坐标帧 i 相对于 j 的方向表示, 因此可以作为从坐标帧 i 到坐标帧 j 的旋转表示。表 2.1 列出了本节中列出的其它方向表示的等效旋转矩阵。表 2.2 包含从一个已知旋转矩阵到这些其它表示的转换。

欧拉角

对于一个最小表示, 坐标帧 i 相对于坐标帧 j 的方向可以标志为三个角 $(\alpha, \beta, \gamma)^T$ 的向量。当每个角表示围绕移动坐标帧的轴旋转时, 这些角称为欧拉角。这样, 每次连续旋转的轴的位置取决

表 2.2: 从旋转矩阵到各种方向表示的转换

旋转矩阵: ${}^j\mathbf{R}_i = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}$
$Z - Y - X$ 欧拉角 $(\alpha, \beta, \gamma)^T$: $\beta = \text{Atan2} \left(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2} \right)$ $\alpha = \text{Atan2} \left(\frac{r_{21}}{\cos \beta}, \frac{r_{11}}{\cos \beta} \right)$ $\gamma = \text{Atan2} \left(\frac{r_{32}}{\cos \beta}, \frac{r_{33}}{\cos \beta} \right)$
$X - Y - Z$ 固定角 $(\psi, \theta, \phi)^T$: $\theta = \text{Atan2} \left(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2} \right)$ $\psi = \text{Atan2} \left(\frac{r_{21}}{\cos \theta}, \frac{r_{11}}{\cos \theta} \right)$ $\phi = \text{Atan2} \left(\frac{r_{32}}{\cos \theta}, \frac{r_{33}}{\cos \theta} \right)$
角-轴 $\theta \hat{\mathbf{w}}$: $\theta = \cos^{-1} \left(\frac{r_{11} + r_{22} + r_{33} - 1}{2} \right)$ $\hat{\mathbf{w}} = \frac{1}{2 \sin \theta} \begin{pmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{pmatrix}$
单位四元数 $(\epsilon_0 \epsilon_1 \epsilon_2 \epsilon_3)^T$: $\epsilon_0 = \frac{1}{2} \sqrt{1 + r_{11} + r_{22} + r_{33}}$ $\epsilon_1 = \frac{r_{32} - r_{23}}{4\epsilon_0}$ $\epsilon_2 = \frac{r_{13} - r_{31}}{4\epsilon_0}$ $\epsilon_3 = \frac{r_{21} - r_{12}}{4\epsilon_0}$

于之前的旋转，因此旋转顺序必须伴随三个角度来定义方向。例如，本手册中使用符号 $(\alpha, \beta, \gamma)^T$ 表示 $Z-Y-X$ 欧拉角。取移动帧 i 与固定帧 j 初始重合， α 为绕帧 i 的 \hat{z} 轴的旋转， β 为绕帧 i 的 \hat{y} 轴的旋转，并且最后， γ 为绕帧 i 的 \hat{x} 轴的两次旋转，等效旋转矩阵 ${}^j\mathbf{R}_i$ 在表 2.1 给出。 $Z-Y-Z$ 和 $Z-X-Z$ 欧拉角是 12 种不同的可能旋转顺序中比其它种更常用的约定。

无论旋转的顺序如何，当第一次旋转和最后一次旋转都围绕同一轴发生时，方向的欧拉角表示总是表现出一个奇异点。这可以在表 2.2 的第二部分中很容易看出，其中当 $\beta = \pm 90^\circ$ 时，角度 α 和 γ 未定义。(对于 $Z-Y-Z$ 和 $Z-X-Z$ 欧拉角当第二次旋转为 0° 或 180° 时，出现奇异点。) 这就产生了机体的角速度向量与欧拉角的时间导数相关的问题，这在一定程度上限制了它们在机器人系统建模中的实用性。对于 $Z-Y-X$ 欧拉角的这种速度关系为

$$\begin{pmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{pmatrix} = \frac{1}{\cos \beta} \begin{pmatrix} 0 & \sin \gamma & \cos \gamma \\ 0 & \cos \gamma \cos \beta & -\sin \gamma \cos \beta \\ \cos \beta & \sin \gamma \sin \beta & \cos \gamma \sin \beta \end{pmatrix} \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix}. \quad (2.5)$$

其中 $(\omega_x, \omega_y, \omega_z)^T = {}^i\boldsymbol{\omega}_i$ 在移动帧 i 中给出。在某些情况下，可能需要此关系的逆运算。

$$\begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} = \begin{pmatrix} -\sin \beta & 0 & 1 \\ \cos \beta \sin \gamma & \cos \gamma & 0 \\ \cos \beta \cos \gamma & -\sin \gamma & 0 \end{pmatrix} \begin{pmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{pmatrix}. \quad (2.6)$$

固定角

当每个角度表示围绕固定帧的一个轴的旋转时，一个有三个角度的向量还可以标志坐标帧 i 相对于坐标帧 j 的方向。适当地，这些角度被称为固定角度，并且旋转的顺序必须再次伴随这些角度来定义方向。 $X-Y-Z$ 固定角，在这里标志为 $(\psi, \theta, \phi)^T$ ，是 12 个不同的可能旋转顺序中的常见约定。假设移动帧 i 和固定帧 j 初始重合， ψ 是围绕固定 \hat{x}_j 轴的偏航旋转， θ 是围绕固定 \hat{y}_j 轴的旋距旋转，并且 ϕ 是围绕固定 \hat{z}_j 轴的滚动旋转。通过比较表 2.1 中相应的等效旋转矩阵和表 2.2 中相应的转换，可以看出：一组 $X-Y-Z$ 固定角完全等价于同一组 $Z-Y-X$ 欧拉角 ($\alpha = \phi, \beta = \theta$ ，并且 $\gamma = \psi$)。该结果通常适用于这样的情况，即围绕固定帧的三个轴的三次旋转定义了与围绕移动帧的三个轴以相反顺序进行的三次旋转的方向的相同。同样，所有方向的固定角表示都会受到欧拉角奇异性的影响。另外，固定角的时间导数与角速度向量之间的关系类似于欧拉角的关系。

角-轴

一个单一的角度 θ 与一个单位向量 $\hat{\mathbf{w}}$ 相结合，也可以标志坐标帧 i 相对于坐标帧 j 的方向。在这种情况下，坐标帧 i 围绕一个轴旋转了一个角度 θ ，该轴由向量 $\hat{\mathbf{w}} = (w_x w_y w_z)^T$ 相对于坐标系帧 j 定义。向量 $\hat{\mathbf{w}}$ 有时被称为一个有限旋转的等效轴。角-轴的表示，通常写成 $\theta \hat{\mathbf{w}}$ 或 $(\theta w_x \theta w_y \theta w_z)^T$ ，因为它包含四个参数，所以是一种超参数的表示。解决这个问题的辅助关系是向量 $\hat{\mathbf{w}}$ 的单位幅值。即使有这种辅助关系，角轴表示也不是唯一的，因为通过 $-\theta$ 的角度围绕 $-\hat{\mathbf{w}}$ 的旋转，相当于通过 θ 围绕 $\hat{\mathbf{w}}$ 的旋转。表 2.3 包含了从角-轴表示到单位四元数的转换，反之亦然。通过使用表 2.2 中的转换和表 2.1 中的等效旋转矩阵，可以很容易地找到从这两种表示到欧拉角或固定角的转换。使用密切相关的四元数表示更容易处理速度关系。

表 2.3: 从角-轴到方向的单位四元数表示的转换, 反之亦然。

角-轴 $\theta\hat{w}$ 到单位四元数 $\begin{pmatrix} \epsilon_0 & \epsilon_1 & \epsilon_2 & \epsilon_3 \end{pmatrix}^T$:
$\epsilon_0 = \cos \frac{\theta}{2}$
$\epsilon_1 = w_x \sin \frac{\theta}{2}$
$\epsilon_2 = w_y \sin \frac{\theta}{2}$
$\epsilon_3 = w_z \sin \frac{\theta}{2}$
单位四元数 $\begin{pmatrix} \epsilon_0 & \epsilon_1 & \epsilon_2 & \epsilon_3 \end{pmatrix}^T$ 到角-轴 $\theta\hat{w}$:
$\theta = 2 \cos^{-1} \epsilon_0$
$w_x = \frac{\epsilon_1}{\sin \frac{\theta}{2}}$
$w_y = \frac{\epsilon_2}{\sin \frac{\theta}{2}}$
$w_z = \frac{\epsilon_3}{\sin \frac{\theta}{2}}$

四元数

由 *Hamilton* [2.1] 给出的方向的四元数表示, 虽然在很大程度上被 *Gibbs* [2.2] 和 *Graßmann* [2.3] 给出的更简单的向量表示所取代, 但对于机器人学中导致向量/矩阵表示法的表示奇异性的问题, 它是非常有用的 [2.4]。四元数不像欧拉角和固定角那样有奇点。

一个四元数 ϵ 被定义为具有以下形式

$$\epsilon = \epsilon_0 + \epsilon_1 i + \epsilon_2 j + \epsilon_3 k,$$

其中, 分量 $\epsilon_0, \epsilon_1, \epsilon_2$ 和 ϵ_3 是标量, 有时称为 Euler 参数, i, j 和 k 是算子。这些算子被定义为满足以下组合规则

$$ii = jj = kk = -1,$$

$$ij = k, jk = i, ki = j,$$

$$ji = -k, kj = -i, ik = -j.$$

两个四元数的相加是通过各自的分量分别相加来实现的, 所以算子起到了分隔符的作用。对于加法的空元素是四元数 $\mathbf{0} = 0 + 0i + 0j + 0k$, 并且四元数加和是结合的、交换的和分配的。对于乘法的空元素是 $\mathbf{I} = 1 + 0i + 0j + 0k$, 对于任意四元数, 可以看出 $\mathbf{I}\epsilon = \epsilon$ 。四元数乘积是结合的和分配的, 但不是交换的, 并且按照算子和加法的约定, 具有以下形式

$$\begin{aligned}
 \mathbf{ab} = & a_0 b_0 - a_1 b_1 - a_2 b_2 - a_3 b_3 \\
 & + (a_0 b_1 + a_1 b_0 + a_2 b_3 - a_3 b_2) i \\
 & + (a_0 b_2 + a_2 b_0 + a_3 b_1 - a_1 b_3) j \\
 & + (a_0 b_3 + a_3 b_0 + a_1 b_2 - a_2 b_1) k.
 \end{aligned} \tag{2.7}$$

定义一个四元数的共轭是很方便的

$$\tilde{\epsilon} = \epsilon_0 - \epsilon_1 i - \epsilon_2 j - \epsilon_3 k,$$

所以

$$\epsilon \tilde{\epsilon} = \tilde{\epsilon} \epsilon = \epsilon_0^2 + \epsilon_1^2 + \epsilon_2^2 + \epsilon_3^2.$$

然后可以定义一个单位四元数, 使得 $\epsilon\tilde{\epsilon} = 1$ 。通常, ϵ_0 被称为四元数的标量部分, 而 $(\epsilon_1\epsilon_2\epsilon_3)^T$ 被称为向量部分。单位四元数用于描述方向, 单位幅值提供辅助关系, 以解决超 (四) 参数坐标的使用问题。一个向量在四元数表示中被定义为 $\epsilon_0 = 0$ 的四元数, 因此, 一个向量 $\mathbf{p} = (p_x p_y p_z)^T$ 可以表达为一个四元数 $\mathbf{p} = p_x i + p_y j + p_z k$ 。对于任意单位四元数 ϵ , 运算 $\epsilon \mathbf{p} \tilde{\epsilon}$ 执行向量 \mathbf{p} 围绕方向 $(\epsilon_1\epsilon_2\epsilon_3)^T$ 的旋转。通过展开运算 $\epsilon \mathbf{p} \tilde{\epsilon}$ 并将结果与表 2.1 中列出的等效旋转矩阵进行比较, 可以清楚地看出这一点。此外, 如表 2.3 所示, 单位四元数与角-轴表示密切相关。 ϵ_0 对应 (但不等于) 旋转角度, 而 ϵ_1, ϵ_2 和 ϵ_3 定义旋转轴。

对于速度分析, 四元数的时间导数可与角速度向量相关, 为

$$\begin{pmatrix} \dot{\epsilon}_0 \\ \dot{\epsilon}_1 \\ \dot{\epsilon}_2 \\ \dot{\epsilon}_3 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} -\epsilon_1 & -\epsilon_2 & -\epsilon_3 \\ \epsilon_0 & \epsilon_3 & -\epsilon_2 \\ -\epsilon_3 & \epsilon_0 & \epsilon_1 \\ \epsilon_2 & -\epsilon_1 & \epsilon_0 \end{pmatrix} \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix}, \quad (2.8)$$

其中 $(\omega_x, \omega_y, \omega_z)^T = {}^j\boldsymbol{\omega}_i$ 在固定帧 j 中给出。定义 $\boldsymbol{\epsilon}_{1:3} = (\epsilon_1, \epsilon_2, \epsilon_3)^T$, 很容易验证

$$\begin{aligned} \dot{\epsilon}_0 &= -\frac{1}{2} {}^j\boldsymbol{\omega}_i^T \boldsymbol{\epsilon}_{1:3} \\ \dot{\boldsymbol{\epsilon}}_{1:3} &= \frac{1}{2} (\epsilon_0 {}^j\boldsymbol{\omega}_i - \boldsymbol{\epsilon}_{1:3} \times {}^j\boldsymbol{\omega}_i). \end{aligned}$$

在角速度在移动坐标中给定的情况下, 这些速率方程采用类似的形式。

$$\begin{aligned} \dot{\epsilon}_0 &= -\frac{1}{2} {}^i\boldsymbol{\omega}_i^T \boldsymbol{\epsilon}_{1:3} \\ \dot{\boldsymbol{\epsilon}}_{1:3} &= \frac{1}{2} (\epsilon_0 {}^i\boldsymbol{\omega}_i + \boldsymbol{\epsilon}_{1:3} \times {}^i\boldsymbol{\omega}_i) \end{aligned}$$

在这种情况下, 相应的速率矩阵与方程 (2.8) 相同, 只是底部 3×3 子矩阵中的非对角线元素具有相反的符号。

虽然单位四元数仅表示机体的方向, 但四元数可以对偶化 [2.5-7], 以创建一个代数, 该代数提供一个机体在空间中的位置和方向的描述。其它组合表示将在后续章节中讨论。

2.2.3 齐次变换

前面几节分别讨论了位置和方向的表示。通过齐次变换, 位置向量和旋转矩阵以一种紧凑的表示法组合在一起。如果 i 坐标帧相对于 j 坐标帧的位置和方向已知, 则相对于 i 坐标帧表达的任意向量 ${}^i\mathbf{r}$ 可以相对于 j 坐标帧表达。使用第 2.2.1 节的表示法, 坐标帧 i 相对于坐标帧 j 的原点位置可用向量 ${}^j\mathbf{p}_i = \begin{pmatrix} {}^j p_i^x & {}^j p_i^y & {}^j p_i^z \end{pmatrix}^T$ 标志, 使用第 2.2.2 节的表示法, 帧 i 相对于帧 j 的方向可以用旋转矩阵 ${}^j\mathbf{R}_i$ 来标志。因此,

$${}^j\mathbf{r} = {}^j\mathbf{R}_i {}^i\mathbf{r} + {}^j\mathbf{p}_i. \quad (2.9)$$

该方程可被写为

$$\begin{pmatrix} {}^j\mathbf{r} \\ 1 \end{pmatrix} = \begin{pmatrix} {}^j\mathbf{R}_i & {}^j\mathbf{p}_i \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} {}^i\mathbf{r} \\ 1 \end{pmatrix}, \quad (2.10)$$

其中

$${}^j\mathbf{T}_i = \begin{pmatrix} {}^j\mathbf{R}_i & {}^j\mathbf{p}_i \\ \mathbf{0}^T & 1 \end{pmatrix} \quad (2.11)$$

是 4×4 齐次变换矩阵，并且 $\begin{pmatrix} {}^j\mathbf{r} & 1 \end{pmatrix}^T$ 和 $\begin{pmatrix} {}^i\mathbf{r} & 1 \end{pmatrix}^T$ 是位置向量 ${}^j\mathbf{r}$ 和 ${}^i\mathbf{r}$ 的齐次表示。该矩阵 ${}^j\mathbf{T}_i$ 将坐标帧 i 中的向量变换为坐标帧 j ，其逆 ${}^j\mathbf{T}_i^{-1}$ 将向量从坐标帧 j 变换到坐标帧 i 。

$${}^j\mathbf{T}_i^{-1} = {}^i\mathbf{T}_j = \begin{pmatrix} {}^j\mathbf{R}_i^T & -{}^j\mathbf{R}_i^T {}^j\mathbf{p}_i \\ \mathbf{0}^T & 1 \end{pmatrix}. \quad (2.12)$$

4×4 齐次变换矩阵的合成通过简单的矩阵乘法来完成，就像 3×3 旋转矩阵的情况一样。因此， ${}^k\mathbf{T}_i = {}^k\mathbf{T}_j {}^j\mathbf{T}_i$ 。由于矩阵乘法不可交换，所以顺序或序列很重要。

一个简单的围绕一个轴旋转的齐次变换有时标志为 **Rot**，因此围绕 $\hat{\mathbf{z}}$ 轴旋转一个 θ 的表示为

$$\mathbf{Rot}(\hat{\mathbf{z}}, \theta) = \begin{pmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (2.13)$$

类似地，一个简单的沿轴平移的齐次变换有时标志为 **Trans**，因此沿轴 $\hat{\mathbf{x}}$ 平移一个 d 的表示为

$$\mathbf{Trans}(\hat{\mathbf{x}}, d) = \begin{pmatrix} 1 & 0 & 0 & d \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (2.14)$$

当需要紧凑的表示法和/或编程的易用性是最重要的考虑因素时，齐次变换特别有吸引力。然而，这并不是一种计算效率很高的表示法，因为它引入了大量额外的 1 和 0 乘法。虽然齐次变换矩阵技术上包含 16 个元素，但有四个元素定义为 0 或 1，其余元素由旋转矩阵和位置向量组成。因此，唯一真正超参数的坐标来自旋转矩阵分量，因此相关的辅助关系是那些与旋转矩阵相关的辅助关系。

2.2.4 旋量变换

方程 (2.9) 中的变换可被视为由坐标帧 i 和 j 之间的旋转和这些坐标帧之间的单独位移组成。要从帧 i 到帧 j ，人们可以先执行旋转，然后再执行位移，反之亦然。另外，除了纯平移的情况外，这些帧之间的空间位移可以表达为围绕一条唯一的直线的旋转以及平行于该直线的平移。

Chasles 定理

Chasles 定理，以 Chirikjian 和 Kyatkin[2.8] 所述的形式，有两个部分。第一部分指出：

一个机体在空间中的任意位移都可以通过将指定的一点从其初始位置平移至其最终位置，然后整个机体围绕该点旋转以使其进入其最终方向来实现。

第二部分指出：

一个机体在空间中的任意位移都可以通过机体围绕空间中一条唯一的直线旋转，并伴随着机体平行于该线的平移来实现。

这样一条直线被称为旋量 (screw) 轴，并且正是这第二个结果，其通常被认为是 Chasles 定理。

该定理的第一部分几乎是公理化的。在欧几里得空间的任意地方，在机体中一个指定的点可以从给定的初始位置移动到给定的最终位置。通过进一步要求机体中所有的点穿过相同的位移，机体平移使得指定点从其初始位置移动到其最终位置。然后，机体可以围绕该点旋转到任意给定的最终方向。

定理的第二部分依赖于空间位移的表示，需要一个更复杂的论证。由欧拉角产生的一个初步定理允许对机体的旋转有更大的特殊性：**一个机体的任意位移，其中一个点保持固定，就相当于该机体围绕一条通过该点的唯一轴线的旋转。**几何上，在移动的机体中嵌入三个点，并让其中一个点作为旋转发生的固定点，另外两个点分别具有初始位置和最终位置。在每种情况下，连接初始和最终位置的直线的右平分面都必须包含固定点。平分线平面中的任意直线都可以是旋转轴，旋转轴将对对应点从其初始位置带到其最终位置。因此，两个平分面共用的唯一直线是这样的：围绕它的旋转将使机体中的任意点从其初始位置移动到其最终位置。刚性条件要求机体中包含该线的所有平面旋转相同角度。

对于由旋转矩阵 ${}^j\mathbf{R}_i$ 描述的刚体的任意旋转，欧拉定理指出存在唯一的特征向量 $\hat{\mathbf{w}}$ ，使得

$${}^j\mathbf{R}_i\hat{\mathbf{w}} = \hat{\mathbf{w}}, \quad (2.15)$$

其中 $\hat{\mathbf{w}}$ 是一个平行于旋转轴的单位向量。该表达式要求 ${}^j\mathbf{R}_i$ 的一个单位特征值对应于特征向量 $\hat{\mathbf{w}}$ 。剩下的两个特征值是 $\cos\theta \pm i\sin\theta$ ，其中 i 是复数算子， θ 是机体围绕该轴旋转的角度。

将 Chasles 定理的第一部分与 Euler 定理相结合，一般的空间位移可以表示为一个点从其初始位置到其最终位置的平移，然后围绕通过该点的唯一轴线进行唯一旋转，将机体从其初始方向带到其最终方向。将平移分解为沿着旋转轴方向和正交于该轴的分量，机体中的每个点在该轴方向上具有相同的位移分量，因为围绕它旋转不会影响该分量。投影到一个垂直于旋转轴的平面上，位移的运动学几何与平面运动的运动学几何相同。正如在平面上有一个唯一的点，机体可以围绕它在两个给定的位置之间旋转一样，在投影平面上也有一个唯一的点。如果旋转轴移动通过该点，空间位移可以通过围绕该轴旋转并沿其平移来实现，正如定理所述。

围绕旋转发生的直线称为位移的旋量轴。线性位移 d 与旋转角 θ 的比率称为旋量轴的旋距 h [2.4]。因此，

$$d = h\theta. \quad (2.16)$$

纯平移的旋量轴不是唯一的。任意平行于平移的直线都可以被视为旋量轴，由于旋转 θ 为零，因此称平移轴具有无限旋距。

一个旋量轴在任意坐标帧中最方便地表示为平行于它的单位向量 $\hat{\mathbf{w}}$ 和位于其上的任意点的位置向量 $\boldsymbol{\rho}$ 。旋距 h 和转角度度 θ 的附加规范旋完全定义第二坐标帧相对于第一坐标帧的位置。因此，总共有八个坐标定义了一个旋量变换，该变换由两个坐标组成。向量 $\hat{\mathbf{w}}$ 的单位大小提供了一个辅助关系，但一般不存在第二个辅助关系，因为同一旋量轴由其上的所有点定义，也就是说，向量 $\boldsymbol{\rho}$ 包含一个自由坐标。

代数上，旋量位移表示为

表 2.4: 从旋量变换到齐次变换的转换, 反之亦然, 缩写为 $c_\theta := \cos \theta$, $s_\theta := \sin \theta$, 以及 $v_\theta := 1 - \cos \theta$

旋量变换到齐次变换
${}^j\mathbf{R}_i = \begin{pmatrix} w_x^2 v_\theta + c_\theta & w_x w_y v_\theta - w_z s_\theta & w_x w_z v_\theta + w_y s_\theta \\ w_x w_y v_\theta + w_z s_\theta & w_y^2 v_\theta + c_\theta & w_y w_z v_\theta - w_x s_\theta \\ w_x w_z v_\theta - w_y s_\theta & w_y w_z v_\theta + w_x s_\theta & w_z^2 v_\theta + c_\theta \end{pmatrix}$ ${}^j\mathbf{p}_i = (\mathbf{1}_{3 \times 3} - {}^j\mathbf{R}_i) \boldsymbol{\rho} + h\theta \hat{\mathbf{w}}$
齐次变换到旋量变换
$\mathbf{l} = \begin{pmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{pmatrix}^T$ $\theta = \text{sign}(\mathbf{l}^T {}^j\mathbf{p}_i) \left \cos^{-1} \left(\frac{r_{11} + r_{22} + r_{33} - 1}{2} \right) \right $ $h = \frac{\mathbf{l}^T {}^j\mathbf{p}_i}{2\theta \sin \theta}$ $\boldsymbol{\rho} = \frac{(\mathbf{1}_{3 \times 3} - \mathbf{R}_i^T) {}^j\mathbf{p}_i}{2(1 - \cos \theta)}$ $\hat{\mathbf{w}} = \frac{\mathbf{l}}{2 \sin \theta}$

$${}^j\mathbf{r} = {}^j\mathbf{R}_i ({}^i\mathbf{r} - \boldsymbol{\rho}) + d\hat{\mathbf{w}} + \boldsymbol{\rho}. \quad (2.17)$$

将此表达式与方程 (2.9) 进行比较得到

$${}^j\mathbf{p}_i = d\hat{\mathbf{w}} + (\mathbf{1}_{3 \times 3} - {}^j\mathbf{R}_i) \boldsymbol{\rho}. \quad (2.18)$$

$\mathbf{1}_{3 \times 3}$ 标志 3×3 单位矩阵。对于 d 的表达式, 通过取方程两边的内积 $\hat{\mathbf{w}}$, 可以很容易地得到。

$$d = \hat{\mathbf{w}}^T {}^j\mathbf{p}_i. \quad (2.19)$$

矩阵 $\mathbf{1}_{3 \times 3} - {}^j\mathbf{R}_i$ 是奇异的, 因此方程 (2.18) 不能通过求解得到 $\boldsymbol{\rho}$ 的唯一值, 但由于 $\boldsymbol{\rho}$ 可以表示旋量轴上的任意点, 因此这并不合适。可以任意选择 $\boldsymbol{\rho}$ 的一个分量, 然后求解任意两个分量方程, 找到 $\boldsymbol{\rho}$ 的另外两个分量。然后旋量轴上的所有其它点由 $\boldsymbol{\rho} + k\hat{\mathbf{w}}$ 给出, 其中 k 可以取任意值。

表 2.4 包含旋量变换和齐次变换之间的转换。注意, 旋量变换的等效旋转矩阵与表 2.1 中表示方向的角-轴的等效旋转矩阵具有相同的形式。此外, 表 2.4 中使用向量 $\boldsymbol{\rho}$ 与旋量轴 ($\hat{\mathbf{w}}^T \boldsymbol{\rho} = 0$) 正交的辅助关系, 以提供旋量变换的唯一转换。该逆解, 由 Rodrigues 方程求出给定旋量位移对应的旋转矩阵 ${}^j\mathbf{R}_i$ 和平移向量 ${}^j\mathbf{p}_i$ 。

Rodrigues 方程

给定一个旋量轴, 一个机体围绕它的角位移, 以及该机体沿着它的平移, 可以找到该机体中任意一点的位移。将矩阵变换视为描述机体的位移, 这等价于找到与给定旋量位移的等价矩阵变换。

参考图 2.1, 一个点在旋量位移前后的位置向量在几何学上可以有如下关系

$$\begin{aligned} {}^j\mathbf{r} = & {}^i\mathbf{r} + d\hat{\mathbf{w}} + \sin \theta \hat{\mathbf{w}} \times ({}^i\mathbf{r} - \boldsymbol{\rho}) \\ & - (1 - \cos \theta) ({}^i\mathbf{r} - \boldsymbol{\rho}) - ({}^i\mathbf{r} - \boldsymbol{\rho}) \cdot \hat{\mathbf{w}} \hat{\mathbf{w}}, \end{aligned} \quad (2.20)$$

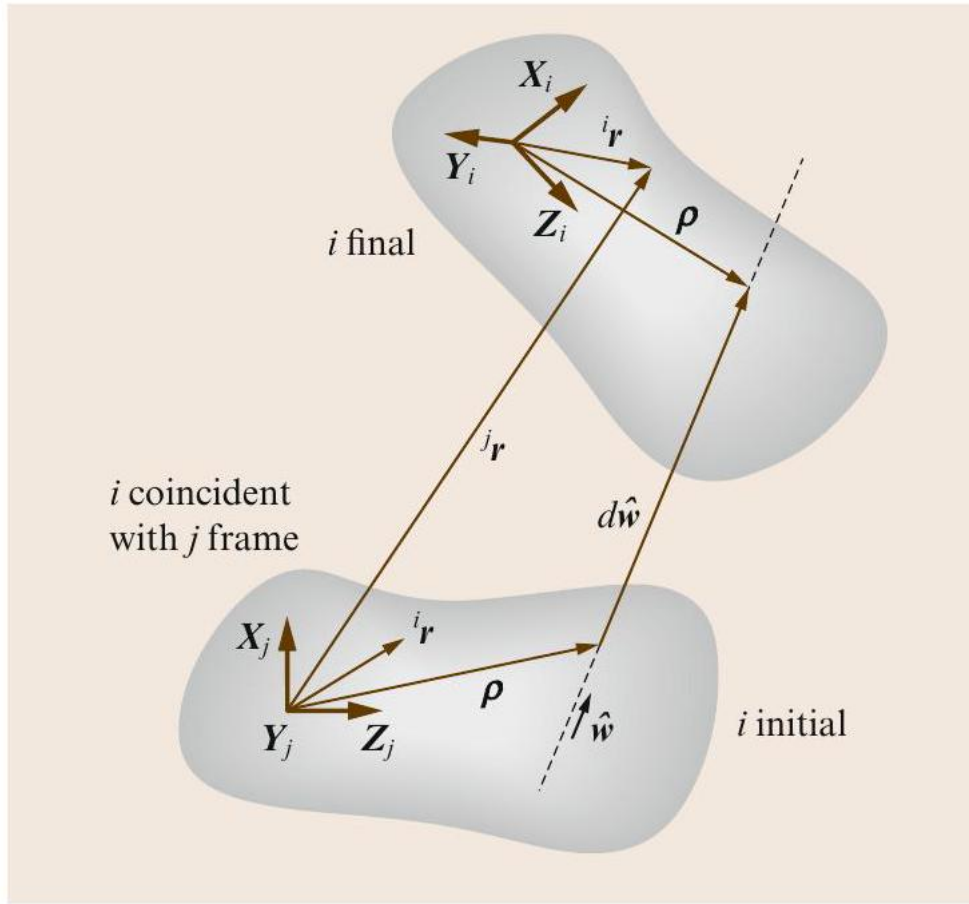


图 2.1: 经历旋量位移的机体中的任意点的初始和最终位置; ${}^i\mathbf{r}$ 是该点相对于移动帧的位置, 其初始位置与固定帧 j 重合; ${}^j\mathbf{r}$ 是移动机体旋量位移后该点相对于固定帧的位置。

其中 ${}^i\mathbf{r}$ 和 ${}^j\mathbf{r}$ 标志该点的初始和最终位置, $\hat{\mathbf{w}}$ 和 ρ 指定旋量轴, 并且 θ 和 d 给出围绕它的位移。该结果通常被称为 Rodrigues 方程 [2.9], 它可以写为一个矩阵变换 [2.10],

$${}^j\mathbf{r} = {}^j\mathbf{R}_i {}^i\mathbf{r} + {}^j\mathbf{p}_i, \quad (2.21)$$

因为, 当展开时, 它给出了 ${}^j\mathbf{r}$ 各分量与 ${}^i\mathbf{r}$ 各分量的三个线性方程。

$${}^j\mathbf{R}_i = \begin{pmatrix} w_x^2 v_\theta + c_\theta & w_x w_y v_\theta - w_z s_\theta & w_x w_z v_\theta + w_y s_\theta \\ w_x w_y v_\theta + w_z s_\theta & w_y^2 v_\theta + c_\theta & w_y w_z v_\theta - w_x s_\theta \\ w_x w_z v_\theta - w_y s_\theta & w_y w_z v_\theta + w_x s_\theta & w_z^2 v_\theta + c_\theta \end{pmatrix}$$

$${}^j\mathbf{p}_i = (\mathbf{1}_{3 \times 3} - {}^j\mathbf{R}_i) \rho + h\theta \hat{\mathbf{w}},$$

其中缩写为 $c_\theta := \cos \theta$, $s_\theta := \sin \theta$ 和 $v_\theta = 1 - \cos \theta$ 。以这种形式表达的旋转矩阵 ${}^j\mathbf{R}_i$ 也被称为旋量矩阵, 这些方程根据旋量参数给出 ${}^j\mathbf{R}_i$ 和 ${}^j\mathbf{p}_i$ 的元素。

一个例外出现在纯平移的情况下, 对于这种情况, $\theta = 0$, 并且 Rodrigues 方程变为

$${}^j\mathbf{r} = {}^i\mathbf{r} + d\hat{\mathbf{w}}. \quad (2.22)$$

代入这种情况, ${}^j\mathbf{R}_i = \mathbf{1}_{3 \times 3}$ 并且 ${}^j\mathbf{p}_i = d\hat{\mathbf{w}}$ 。

有关旋量理论的更多信息, 可在文献 [2.11-15] 中找到。

2.2.5 矩阵指数参数化

一个机体的位置 and 方向也可以用指数映射统一表达。该方法首先被引入到在纯旋转中的应用, 并被推广到刚体运动。有关该方法的更多详细信息, 可以在文献 [2.16] 和 [2.17] 中找到。

对于旋转的指数坐标

行列式为 1 的所有正交矩阵集合, 即所有旋转矩阵 \mathbf{R} 的集合, 是矩阵乘法运算下的一个群 (*group*), 标志为 $SO(3) \subset \mathbb{R}^{3 \times 3}$ [2.18]。这表示特殊正交 (*special orthogonal*), 其中特殊 (*special*) 暗示 $\det \mathbf{R}$ 为 +1 而不是 ± 1 。旋转矩阵集满足群的四个公理:

- 封闭性 (*Closure*): $\mathbf{R}_1\mathbf{R}_2 \in SO(3) \quad \forall \mathbf{R}_1, \mathbf{R}_2 \in SO(3)$
- 幺元 (*Identity*): $\mathbf{1}_{3 \times 3}\mathbf{R} = \mathbf{R}_{3 \times 3} = \mathbf{R} \quad \forall \mathbf{R} \in SO(3)$
- 逆元 (*Inverse*): $\mathbf{R}^T \in SO(3)$ 是 $\mathbf{R} \forall \mathbf{R} \in SO(3)$ 的唯一逆元
- 结合律 (*Associativity*): $(\mathbf{R}_1\mathbf{R}_2)\mathbf{R}_3 = \mathbf{R}_1(\mathbf{R}_2\mathbf{R}_3) \quad \forall \mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3 \in SO(3)$ 。

在第 2.2.2 节提出的角-轴表示中, 方向表达为围绕单位向量 $\hat{\mathbf{w}}$ 定义的轴旋转的角度 θ 。表 2.1 中的等效旋转矩阵可表达为指数映射

$$\begin{aligned} \mathbf{R} &= e^{\mathbf{S}(\hat{\mathbf{w}})\theta} \\ &= \mathbf{1}_{3 \times 3} + \theta \mathbf{S}(\hat{\mathbf{w}}) + \frac{\theta^2}{2!} \mathbf{S}(\hat{\mathbf{w}})^2 \\ &\quad + \frac{\theta^3}{3!} \mathbf{S}(\hat{\mathbf{w}})^3 + \dots, \end{aligned} \quad (2.23)$$

其中 $\mathbf{S}(\hat{\mathbf{w}})$ 是单位斜对称矩阵

$$\mathbf{S}(\hat{\mathbf{w}}) = \begin{pmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{pmatrix} \quad (2.24)$$

因此, 指数映射将对应于旋转轴 $\hat{\mathbf{w}}$ 的斜对称矩阵 $\mathbf{S}(\hat{\mathbf{w}})$ 变换为的一个正交矩阵 \mathbf{R} , 其对应于围绕轴 $\hat{\mathbf{w}}$ 旋转一个角度 θ 。可以证明, 对于 $e^{\mathbf{S}(\hat{\mathbf{w}})\theta}$ 的闭式表达式, 是可有效计算的, 为

$$e^{\mathbf{S}(\hat{\mathbf{w}})\theta} = \mathbf{1}_{3 \times 3} + \mathbf{S}(\hat{\mathbf{w}}) \sin \theta + \mathbf{S}(\hat{\mathbf{w}})^2 (1 - \cos \theta) \quad (2.25)$$

该 $\begin{pmatrix} \theta w_x & \theta w_y & \theta w_z \end{pmatrix}^T$ 的分量, 其与表 2.2 中旋转矩阵 \mathbf{R} 的元素相关, 被称为 \mathbf{R} 的指数坐标 (*exponential coordinates*)。

对于刚体运动的指数坐标

如第 2.2.3 节所示, 一个机体的位置 and 方向可被表达为位置向量 $\mathbf{p} \in \mathbb{R}^3$ 和旋转矩阵 $\mathbf{R} \in SO(3)$ 的组合。 \mathbb{R}^3 与 $SO(3)$ 的乘积空间是称为 $SE(3)$ 的群, 它表示特殊的欧几里得空间 (*special Euclidean*)。

$$\begin{aligned} SE(3) &= \{(\mathbf{p}, \mathbf{R}) : \mathbf{p} \in \mathbb{R}^3, \mathbf{R} \in SO(3)\} \\ &= \mathbb{R}^3 \times SO(3). \end{aligned}$$

齐次变换集满足群四个公理:

- 封闭性 (*Closure*): $\mathbf{T}_1 \mathbf{T}_2 \in SE(3) \quad \forall \mathbf{T}_1, \mathbf{T}_2 \in SE(3)$
- 幺元 (*Identity*): $\mathbf{1}_{4 \times 4} \mathbf{T} = \mathbf{T} \mathbf{1}_{4 \times 4} = \mathbf{T} \quad \forall \mathbf{T} \in SE(3)$
- 逆元 (*Inverse*): $\mathbf{T} \quad \forall \mathbf{T} \in SE(3)$ 的唯一逆元在方程 (2.12) 中给出
- 结合律 (*Associativity*): $(\mathbf{T}_1 \mathbf{T}_2) \mathbf{T}_3 = \mathbf{T}_1 (\mathbf{T}_2 \mathbf{T}_3) \quad \forall \mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3 \in SE(3)$.

在第 2.2.4 节的旋量变换表示中, 位置和方向是通过由单位向量 $\hat{\mathbf{w}}$ 定义的旋量轴, 绕该轴的旋转角度 θ , 在轴上的点 $\boldsymbol{\rho}$ 满足 $\hat{\mathbf{w}}^T \boldsymbol{\rho} = 0$, 以及旋量轴的旋距 h 来表达的。表 2.4 中的等效齐次变换可表达为指数映射

$$\mathbf{T} = e^{\hat{\boldsymbol{\xi}}\theta} = \mathbf{1}_{4 \times 4} + \hat{\boldsymbol{\xi}}\theta + \frac{(\hat{\boldsymbol{\xi}}\theta)^2}{2!} + \frac{(\hat{\boldsymbol{\xi}}\theta)^3}{3!} + \dots, \quad (2.26)$$

其中

$$\hat{\boldsymbol{\xi}} = \begin{pmatrix} \mathbf{S}(\hat{\mathbf{w}}) & \mathbf{v} \\ \mathbf{0}^T & 0 \end{pmatrix} \quad (2.27)$$

是单位斜对称矩阵 $\mathbf{S}(\hat{\mathbf{w}})$ 的推广, 其被称为运动旋量 (*twist*)。 $\hat{\boldsymbol{\xi}}$ 的运动旋量坐标由 $\boldsymbol{\xi} := \begin{pmatrix} \hat{\mathbf{w}}^T & \mathbf{v}^T \end{pmatrix}^T$ 给出。可以证明, 对于 $e^{\hat{\boldsymbol{\xi}}\theta}$ 的闭式表达式为

$$\begin{aligned} &e^{\hat{\boldsymbol{\xi}}\theta} \\ &= \begin{pmatrix} e^{\mathbf{S}(\hat{\mathbf{w}})\theta} & (\mathbf{1}_{3 \times 3} - e^{\mathbf{S}(\hat{\mathbf{w}})\theta})(\hat{\mathbf{w}} \times \mathbf{v}) + \hat{\mathbf{w}}^T \mathbf{v} \theta \hat{\mathbf{w}} \\ \mathbf{0}^T & 1 \end{pmatrix}. \end{aligned} \quad (2.28)$$

此结果与表 2.4 中齐次变换和旋量变换之间的转换进行比较, 得到

$$\mathbf{v} = \boldsymbol{\rho} \times \hat{\mathbf{w}} \quad (2.29)$$

以及

$$h = \hat{\mathbf{w}}^T \mathbf{v}. \quad (2.30)$$

因此, 运动旋量的指数映射将机体的初始位姿变换为其最终位姿。它给出了相对刚体运动。向量 $\boldsymbol{\xi}\theta$ 包含对于刚体变换的指数坐标 (*exponential coordinates*)。

至于旋量变换, 纯平移的情况是唯一的。在这种情况下, $\hat{\mathbf{w}} = \mathbf{0}$, 所以

$$e^{\hat{\boldsymbol{\xi}}\theta} = \begin{pmatrix} \mathbf{1}_{3 \times 3} & \theta \mathbf{v} \\ \mathbf{0}^T & 1 \end{pmatrix}. \quad (2.31)$$

2.2.6 Plücker 坐标

在空间中定义直线最少需要四个坐标。一条直线的 Plücker 坐标形成一个六维 (6-D) 向量，因此它们由两个超参数。它们可以看作是一对三维 (3-D) 向量；一个平行于直线，而另一个是向量关于原点的矩。因此，如果 \mathbf{u} 是平行于直线的任意向量，并且 $\boldsymbol{\rho}$ 是直线上任意点相对于原点的位置，则 Plücker 坐标 (L, M, N, P, Q, R) 给出为：

$$(L, M, N) = \mathbf{u}^T; (P, Q, R) = (\boldsymbol{\rho} \times \mathbf{u})^T. \quad (2.32)$$

对于简单地定义一条直线， \mathbf{u} 的幅值不是唯一的，并且 $\boldsymbol{\rho}$ 平行于 \mathbf{u} 的分量也不是唯一的。施加两个辅助关系将集合简化为四个独立的坐标。一是两个三维向量的标量积等于零。

$$LP + MQ + NR \equiv 0. \quad (2.33)$$

另一个是当所有坐标都乘以相同的比例因子时指定的直线的不变性。

$$(L, M, N, P, Q, R) \equiv (kL, kM, kN, kP, kQ, kR). \quad (2.34)$$

这种关系可以采取约束 \mathbf{u} 具有单位幅值的形式，以便 L, M 和 N 是方向余弦。

在本手册中，用 Plücker 坐标表示速度通常是有用的，其中与直线的定义不同，两个三维向量的幅值不是任意的。这导致了 *von Mises* [2.9,19] 和 *Everett* [2.20] 的电机表示法。对于瞬时重合的坐标帧，一个是固定的，而另一个是嵌入移动机体的，当两者都相对于固定坐标帧表达时， $\boldsymbol{\omega}$ 是机体的角速度，并且 \mathbf{v}_O 是机体固定坐标帧原点 O 的速度。对于机体的空间速度 \mathbf{v} ，这就提供了一个 Plücker 坐标系统。 \mathbf{v} 的 Plücker 坐标就是简单的 $\boldsymbol{\omega}$ 和 \mathbf{v}_O 的笛卡尔坐标，

$$\mathbf{v} = \begin{pmatrix} \boldsymbol{\omega} \\ \mathbf{v}_O \end{pmatrix}. \quad (2.35)$$

从 Plücker 坐标系统 i 到 Plücker 坐标系统 j 的空间速度变换是通过空间变换 ${}^j\mathbf{X}_i$ 实现的。如果 \mathbf{v}_i 和 \mathbf{v}_j 分别标志一个机体相对于 i 和 j 帧的空间速度，并且 ${}^j\mathbf{p}_i$ 和 ${}^j\mathbf{R}_i$ 分别标志机体帧 i 相对于帧 j 的位置和方向，

$$\mathbf{v}_j = {}^j\mathbf{X}_i \mathbf{v}_i, \quad (2.36)$$

其中

$${}^j\mathbf{X}_i = \begin{pmatrix} {}^j\mathbf{R}_i & \mathbf{0}_{3 \times 3} \\ \mathbf{S}({}^j\mathbf{p}_i) {}^j\mathbf{R}_i & {}^j\mathbf{R}_i \end{pmatrix}, \quad (2.37)$$

以使得

$${}^j\mathbf{X}_i^{-1} = {}^i\mathbf{X}_j = \begin{pmatrix} {}^i\mathbf{R}_j & \mathbf{0}_{3 \times 3} \\ -{}^i\mathbf{R}_j \mathbf{S}({}^j\mathbf{p}_i) & {}^i\mathbf{R}_j \end{pmatrix}, \quad (2.38)$$

以及

$${}^k\mathbf{X}_i = {}^k\mathbf{X}_j {}^j\mathbf{X}_i, \quad (2.39)$$

并且 $\mathbf{S}({}^j\mathbf{p}_i)$ 是一个斜对称矩阵

$$\begin{pmatrix} 0 & -{}^j p_i^z & {}^j p_i^y \\ {}^j p_i^z & 0 & -{}^j p_i^x \\ -{}^j p_i^y & {}^j p_i^x & 0 \end{pmatrix}. \quad (2.40)$$

空间向量表示法，包括这里简要提到的空间速度和变换，在第 3.2 节中有更深入的讨论。具体而言，表 3.1 给出了应用空间变换的一个高效计算算法。

2.3 关节运动学

除非另有明确说明，机器人机构的运动学描述通常采用许多理想化方法。构成机器人机构的连杆被假定为在几何上具有位置和形状都是完美的曲面的完美刚体。因此，这些刚体在关节处连接在一起，它们的理想表面处于理想接触状态，彼此之间没有任何间隙。这些接触曲面的各自几何图形决定了两个连杆之间的运动自由度或关节运动学 (*joint kinematics*)。

运动学关节是两个机体之间的连接，它约束着它们的相对运动。相互接触的两个机体形成简单的运动关节。接触的两个机体的表面能够彼此移动，从而允许这两个机体的相对运动。如果在曲面上发生接触，则该简单运动关节被归类为低副关节 [2.21]，而如果仅在点或沿直线发生接触，则该简单运动关节被归类为高副关节。

一个关节模型描述固定在一个关节的一个机体中的帧相对于固定在另一个机体中的帧的运动。运动表达为关节运动变量的函数，并且关节模型的其它元素包括旋转矩阵、位置向量、自由模式和约束模式。关节的自由模式定义允许运动的方向。它们由 $6 \times n_i$ 矩阵 Φ_i 表示，其列为允许运动的 Plücker 坐标。该矩阵将穿过关节的空间速度向量 $\mathbf{v}_{\text{rel},i}$ 与关节速度向量 $\dot{\mathbf{q}}_i$ 相关联，

$$\mathbf{v}_{\text{rel},i} = \Phi_i \dot{\mathbf{q}}_i \quad (2.41)$$

相反，关节的受约束模式定义不允许运动的方向。它们由 $6 \times (6 - n_i)$ 矩阵 Φ_i^c 表示，其互补于矩阵 Φ_i 。表 2.5 和 2.6 包含本节所描述的所有关节的关节模型公式。它们广泛用于第 3 章中介绍的动力学分析。关于关节的更多信息，可在第 4 章中找到。

2.3.1 低副关节

低副关节具有机械吸引力，因为磨损分散到整个表面，润滑剂被困在表面之间的小间隙空间 (在非理想系统中)，从而产生相对良好的润滑。从表面接触的要求可以证明 [2.23]，只有六种可能的低副关节形式：旋转关节、棱柱关节、螺旋关节、圆柱形关节、球形关节和平面关节。

旋转关节

旋转关节的最一般形式，通常缩写为 R ，有时通俗地称为铰链或销关节，是一个由两个相同的旋转面组成的低副。这两个曲面是相同的，只是其中一个是外表面，在旋转轴的任意法线上都是凸的，另一个是内表面，在旋转轴的任意法线上都是凹的。表面可能不仅仅是直角圆柱形式，因为该形式的表面不会对轴向滑动提供任何约束。旋转关节仅允许连接的一个机体相对于另一个机体旋转。一个机体相对于另一个机体的位置可以表达为对关节轴线的两条法线之间的角度，每一个机体上都固定一个。因此，该关节具有一个自由度 (*degree of freedom*, DOF)。当坐标帧 i 的 \hat{z} 轴与旋转关节轴对齐时，表 2.5 中的公式定义了旋转关节模型。

表 2.5: 对于单一自由度低副关节的关节模型公式, 缩写为 $c_{\theta_i} := \cos \theta_i$ 和 $s_{\theta_i} := \sin \theta_i$ (部分适配自文献 [2.22] 中的表 4.1)

关节类型	关节旋转矩阵 ${}^i\mathbf{R}_j$	位置向量 ${}^j\mathbf{p}_i$	自由模式 Φ_i	约束模式 Φ_i^c	位姿 变量	速度变量 $\dot{\mathbf{q}}_i$
旋转关节 R	$\begin{pmatrix} c_{\theta_i} & -s_{\theta_i} & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$	θ_i	$\dot{\theta}_i$
棱柱关节 P	$\mathbf{1}_{3 \times 3}$	$\begin{pmatrix} 0 \\ 0 \\ d_i \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$	d_i	\dot{d}_i
螺旋关节 H (旋距 h)	$\begin{pmatrix} c_{\theta_i} & -s_{\theta_i} & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ h\theta_i \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ h \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -h \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$	θ_i	$\dot{\theta}_i$

棱柱关节

棱柱关节的最一般形式, 通常缩写为 P , 有时通俗地称为滑动关节, 是由两个相同的一般圆柱面形成的低副。这些表面可能不一定是正圆柱形表面。一个通常的圆柱面是通过在一个恒定的方向上挤压任意曲线得到的。同样, 一个表面是内表面, 另一个是外表面。棱形连接仅允许一个连接的构件相对于另一个构件沿拉伸方向滑动。一个机体相对于另一个物体的位置由平行于滑动方向的直线上两点之间的距离确定, 每一个机体都有一个点固定。因此, 该关节也具有一个自由度。当坐标帧 i 的 \hat{z} 轴与棱柱关节轴对齐时, 表 2.5 中的公式定义棱柱关节模型。

螺旋关节

螺旋关节的最一般形式, 通常缩写为 H , 有时通俗地称为螺杆关节, 是由两个螺旋表面形成的低副, 沿着螺旋路径挤压任意曲线形成的。简单的例子是螺栓和螺母, 其中基本生成曲线是一对直线。绕螺旋关节轴旋转的角度 θ 与一个机体相对于另一个机体沿该轴的位移距离 d 直接相关, 即表达式 $d = h\theta$, 其中常数 h 称为螺旋关节的旋距。当坐标帧 i 的 \hat{z} 轴与螺旋关节轴对齐时, 表 2.5 中的公式定义了螺旋关节模型。

圆柱形关节

圆柱形关节, 通常缩写为 C , 是由两个相同的直角圆柱体接触而形成的低副, 一个为内表面, 另一个为外表面。它既允许绕圆柱轴旋转, 也允许平行于圆柱轴滑动。因此, 它是一个具有两个自由度的关节。具有一个以上自由度的低副关节很容易被运动等效复合关节 (第 2.3.3 节) 替代, 该关节是一个自由度的低副的串联链。在这种情况下, 圆柱形关节可以用一个与棱柱形关节串联的旋转

表 2.6: 更高自由度的低副关节、万向节、滚动接触关节和 6-DOF 关节的关节模型公式, 缩写为 $c_{\theta_i} := \cos \theta_i$ 和 $s_{\theta_i} := \sin \theta_i$ (部分适配自文献 [2.22] 中的表 4.1) * 欧拉角 α_i, β_i 和 γ_i 可以被用来代替单位四元数 ϵ_i 以表示方向。

关节类型	关节旋转矩阵 ${}^i\mathbf{R}_j$	位置向量 ${}^j\mathbf{p}_i$	自由模式 Φ_i	约束模式 Φ_i^c	位姿变量	速度变量 $\dot{\mathbf{q}}_i$
圆柱关节 C	$\begin{pmatrix} c_{\theta_i} & -s_{\theta_i} & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ d_i \end{pmatrix}$	$\begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$	θ_i d_i	$\begin{pmatrix} \dot{\theta}_i \\ \dot{d}_i \end{pmatrix}$
球形关节 * S	(Table 2.1)	$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$		
平面关节	$\begin{pmatrix} c_{\theta_i} & -s_{\theta_i} & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} c_{\theta_i} d_{xi} - s_{\theta_i} d_{yi} \\ s_{\theta_i} d_{xi} + c_{\theta_i} d_{yi} \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	θ_i d_{xi} d_{yi}	$\begin{pmatrix} \dot{\theta}_i \\ \dot{d}_{xi} \\ \dot{d}_{yi} \end{pmatrix}$
平面滚动 接触关节 (固定半 径 r)	$\begin{pmatrix} c_{\theta_i} & -s_{\theta_i} & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} r\theta_i c_{\theta_i} - r s_{\theta_i} \\ -r\theta_i s_{\theta_i} - r c_{\theta_i} \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 1 \\ r \\ 0 \\ h \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -r & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$	θ_i	$\dot{\theta}_i$
万向节 U	$\begin{pmatrix} c_{\alpha_i} c_{\beta_i} & -s_{\alpha_i} & c_{\alpha_i} s_{\beta_i} \\ s_{\alpha_i} c_{\beta_i} & c_{\alpha_i} & s_{\alpha_i} s_{\beta_i} \\ -s_{\beta_i} & 0 & c_{\beta_i} \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} -s_{\beta_i} & 0 \\ 0 & 1 \\ c_{\beta_i} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$	$\begin{pmatrix} c_{\beta_i} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ s_{\beta_i} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	α_i β_i	$\begin{pmatrix} \dot{\alpha}_i \\ \dot{\beta}_i \end{pmatrix}$
6-DOF 关节 *	(Table 2.1)	${}^0\mathbf{p}_i$	$\mathbf{1}_{6 \times 6}$		ϵ_i ${}^0\mathbf{p}_i$	$\begin{pmatrix} \boldsymbol{\omega}_i \\ \mathbf{v}_i \end{pmatrix}$

关节代替，其滑动方向与旋转轴平行。虽然使用在第 2.4 节中讨论的几何表示更容易实现，但这种方法对于动力学模拟来说有缺点。将单个圆柱关节建模为棱柱关节和旋转关节的组合，需要在质量和长度均为零的两个关节之间添加虚拟连杆。无质量连杆会产生计算问题。当坐标帧 i 的 \hat{z} 轴与圆柱形关节轴对齐时，表 2.6 中的公式定义了圆柱形关节模型。

球形关节

球形关节，常缩写为 S ，是由两个相同的球面接触而形成的低副。同样，一个是内表面，另一个是外表面。一个球形关节允许围绕穿过球体中心的任意直线旋转。因此，它允许围绕三个不同方向的轴线独立旋转，并且有三个自由度。球形关节可以很容易被一个运动学上等价的复合关节所替代，该复合关节由三个旋转轴组成，这些旋转轴的轴线都相交于一个点，即球形关节的中心。旋转关节轴不需要连续正交，但通常以这种方式实现。通常，这种布置在运动学上等同于球形关节，但当旋转关节轴共面时，它确实会表现出一种奇异性。这与原生球形关节相比，后者从未具有这种奇异性。类似地，如果在模拟中将球形关节建模为三个旋转，则由于必须包含长度为零的无质量虚拟连杆，因此计算困难会再次出现。表 2.6 给出了球形关节的关节模型公式。

平面关节

平面关节由平面接触面形成。与球形关节一样，它是一个具有三个自由度的低副关节。由具有平行轴的三个旋转的串联链组成的运动学等效复合关节可以代替平面关节。与球形关节的情况一样，当旋转轴共面时，复合关节表现出奇异性。当坐标帧 i 的 \hat{z} 轴与接触面的法线对齐时，表 2.6 中的公式定义了平面关节模型。

2.3.2 高副关节

一些高副关节也具有吸引人的特性，特别是滚动副，其中一个机体滚动而不是滑过另一个表面。这在机械上是有吸引力的，因为没有滑动意味着没有磨粒磨损。然而，由于理想接触发生在一个点上或沿着一条直线，在关节上施加载荷可能会导致非常高的局部应力，从而导致其它形式的材料失效，并因此产生磨损。高副关节可用于创建具有特殊几何特性的运动关节，如齿轮副或凸轮和从动件副的情况。

滚动接触关节

滚动接触实际上包含几个不同的几何体。平面运动中的滚动接触允许相对运动的一个自由度，例如滚动轴承。平面滚动接触可以沿直线发生，从而在一定程度上分散载荷和磨损。三维滚动接触允许通过接触点围绕任意轴旋转，这在原则上是唯一的。因此，一个三维滚动接触副允许三个自由度的相对运动。当坐标帧 i 的 \hat{z} 轴与旋转轴对齐并穿过固定半径 r 的滚柱中心时，表 2.6 中的公式定义了平面上滚柱的平面滚动接触关节模型。无论关节是平面的还是三维的，与滚动接触关节相关的无滑动条件要求两个接触体上的点之间的瞬时相对速度为零。如果 P 是机体 i 和 j 之间的滚动接触点，

$$\mathbf{v}_{P_i/P_j} = 0. \quad (2.42)$$

同样，相对加速度是在接触点的两个表面的共同法线方向上。因为与关节相关的约束是以速度表达的，而不能仅以位置表达，所以它是非完整的，如第 2.3.6 节所述。第 24 章对滚动接触的运动学约束进行了更详细的讨论。

2.3.3 复合关节

复合运动关节是由其它构件和简单运动学关节合成链形成的两个机体之间的连接。与简单关节一样的方式，复合关节可以约束连接着的两个机体的相对运动。在这种情况下，这两种关节在运动学上是等效的。

万向节关节

万向节通常缩写为 U ，称为卡丹关节或胡克关节，是一种具有两个自由度的复合关节。它由两个轴线正交相交的旋转链组成。表 2.6 给出了万向节的关节模型，根据欧拉角表示法， α_i 是绕 Z 轴的第一个旋转，并且 β_i 是绕 Y 轴的旋转。对于这种关节，矩阵 Φ_i 和 Φ_i^c 不是常数，因此通常 $\dot{\Phi}_i \neq 0$ 并且 $\dot{\Phi}_i^c \neq 0$ 。如表 2.6 所示，第一个关节轴 (用外侧坐标帧表达) 的方向随 β_i 的变化而变化。

2.3.4 6-DOF 关节

两个未连接在一起的机体的运动可以建模为不引入约束的六自由度关节 (6-DOF joint)。这对于移动机器人特别有用，例如飞机，因为它最多只能与地面进行间歇性接触，因此，相对于固定帧能自由运动的机体被称为浮动基座 (*floating base*)。这样的自由运动关节模型可以用六个关节变量来表示浮动基座在空间中的位置和方向。表 2.6 中包含了 6-DOF 关节模型。

2.3.5 物理实现

在实际的机器人机构中，关节可能具有物理限制，超过该限制的运动是被禁止的。机器人机械臂的工作空间 (第 2.5 节) 通过考虑机构内所有关节的组合极限和运动自由度来确定。旋转关节很容易由旋转电机驱动，因此在机器人系统中非常常见。它们也可能以被动、非驱动关节的形式出现。同样常见的是，虽然比旋转关节少，但棱柱关节相对容易通过线性致动器驱动，如液压缸或气动缸、滚珠丝杠或螺旋千斤顶。它们总是有运动限制，因为原则上单向滑动可以产生无限位移。螺旋关节最常见于机器人机构中，作为线性致动器 (如螺旋千斤顶和滚珠丝杠) 的组成部分，很少用作主要运动关节。具有一个以上自由度的关节通常被动地用于机器人机构中，因为主动关节的每个自由度必须单独致动。被动球形关节经常出现在机器人机构中，而被动平面关节只是偶尔出现。通过使用三个旋转的运动学等效组合并分别驱动，以实现致动球形关节的效果。万向节用于主动和被动形式的机器人机构。

串联链通常由其包含的关节的缩写来标志这些关节在链中的出现顺序。例如，一条 RPR 链包含三个连杆，第一个连杆用旋转轴连接到底部，第二个连杆用棱柱连接，而第二个和第三个连杆用另一个旋转轴连接在一起。如果所有关节都相同，则符号由关节缩写前的关节数组成，例如 6R 六轴串联链式机械臂仅包含旋转关节。

关节是通过硬件实现的，这比第 2.3.1 节和第 2.3.2 节中介绍的理想化的硬件更加复杂。例如，旋转关节可以通过一个滚珠轴承实现，该滚珠轴承由两个轴颈之间的一组轴承球组成。滚珠理想地滚动而不会在轴颈上滑动，从而利用了滚动接触关节的特殊特性。棱柱形关节可以通过滚柱导轨组件实现。

2.3.6 完整与非完整约束

除滚动接触外，与前面章节中讨论的关节相关的所有约束都可以通过仅包含关节位置变量的方程进行数学表达。这些被称为完整约束。方程的数量，也即是约束的数量，为 $6 - n$ ，其中 n 是关节的自由度。约束本质上是轴向关节模型的一部分。

非完整约束是不能单独用位置变量表达的约束，而是包括一个或多个位置变量的时间导数。这些约束方程不能被集成以获得单纯的关节变量之间的关系。机器人系统中最常见的例子来自于使用一个轮子或滚轮，其在另一个构件上滚动而不会打滑。非完整约束，特别是当它们应用于轮式机器人时，将在第 24 章中更详细地讨论。

2.3.7 广义坐标

在一个由 N 个机体组成的机器人机构中，需要 $6N$ 个坐标来指定所有机体相对于坐标帧的位置和方向。由于这些机体中的一部份是连接在一起的，一些约束方程将建立其中一些坐标之间的关系。在这种情况下， $6N$ 个坐标可以表达为较小的坐标集合 \mathbf{q} 的函数，这些坐标都是独立的。该集合中的坐标称为广义坐标，与这些坐标相关的运动符合所有的约束条件。机器人机构的关节变量 \mathbf{q} 形成广义坐标集合 [2.24,25]。

2.4 几何表示

机器人机构的几何结构可以通过将坐标帧附着到每个连杆来方便地定义。虽然这些帧可以任意定位，但对于一致性和计算效率，遵守在连杆上定位帧的约定都是有利的。*Denavit* 和 *Hartenberg* [2.26] 提出了基本约定，并以多种不同方式改编，其中之一是 *Khalil* 和 *Dombre* [2.27] 提出的约定，在本手册中使用。在所有的形式中，该约定只需要四个参数而不是六个参数来相对于另一个坐标帧定位一个坐标帧。这四个参数包括两个连杆参数，连杆长度 a_i 和连杆运动旋量 α_i ，以及两个关节参数，关节偏移 d_i 和关节角度 θ_i 。通过明智地放置坐标帧原点和轴，使得一个坐标帧的 \hat{x} 轴与下一坐标帧的 \hat{z} 轴线相交并垂直，可以实现这种简约。该约定适用于由旋转关节和棱柱关节组成的机器人机构，因此，当存在多自由度关节时，它们被建模为旋转关节和棱柱关节的组合，如第 2.3 节所述。

在机器人机构中定位坐标帧基本上有四种不同形式的约定。通过管理直观表现的权衡，每一种都显示出自己的优势。在最初的 *Denavit* 和 *Hartenberg* [2.26] 约定中，关节 i 位于连杆 i 和 $i + 1$ 之间，因此位于连杆 i 的外侧。此外，关节偏移 d_i 和关节角度 θ_i 是沿着和围绕 $i - 1$ 关节轴测量的，因此关节参数的下标与关节轴的下标不匹配。*Waldron* [2.28] 和 *Paul* [2.29] 修改了原始约定中的轴标记，使得关节 i 位于连杆 $i - 1$ 和 i 之间，以使其与作为构件 0 的串联链的基部构件一致。这将关节 i 放置在连杆 i 的内侧，并且是所有其它修改版本中使用的约定。此外，*Waldron* 和 *Paul* 通过沿 $i + 1$ 关节轴放置 \hat{z}_i 轴，解决了关节参数下标与关节轴之间的不匹配问题。当然，这会将下标不匹配重新定位到坐标帧的关节轴和 \hat{z} 轴之间的对应关系。*Craig* [2.30] 通过沿关节 i 放置 \hat{z}_i 轴消除了所有下标失配，但代价是用带有下标 i 的关节参数和带有下标 $i - 1$ 的连杆参数的混合以形成齐次变换。*Khalil* 和 *Dombre* [2.27] 引入了另一种类似于 *Craig* 的变体，不同之处在于它定义了沿着和围绕 \hat{x}_{i-1} 轴的连杆参数 a_i 和 α_i 。在这种情况下，齐次变换 ${}^{i-1}\mathbf{T}_i$ 仅由下标为 i 的参数形成，下标失配使得 a_i 和 α_i 表示连杆 $i - 1$ 的长度和运动旋量，而不是连杆 i 的，本手册中使用的约定与其它约定相比的优势在于，坐标帧的坐标轴共享关节轴的公共下标，定义从坐标帧 i 到坐标帧 $i - 1$ 的空间变换的四个参数都共享公共下标 i 。

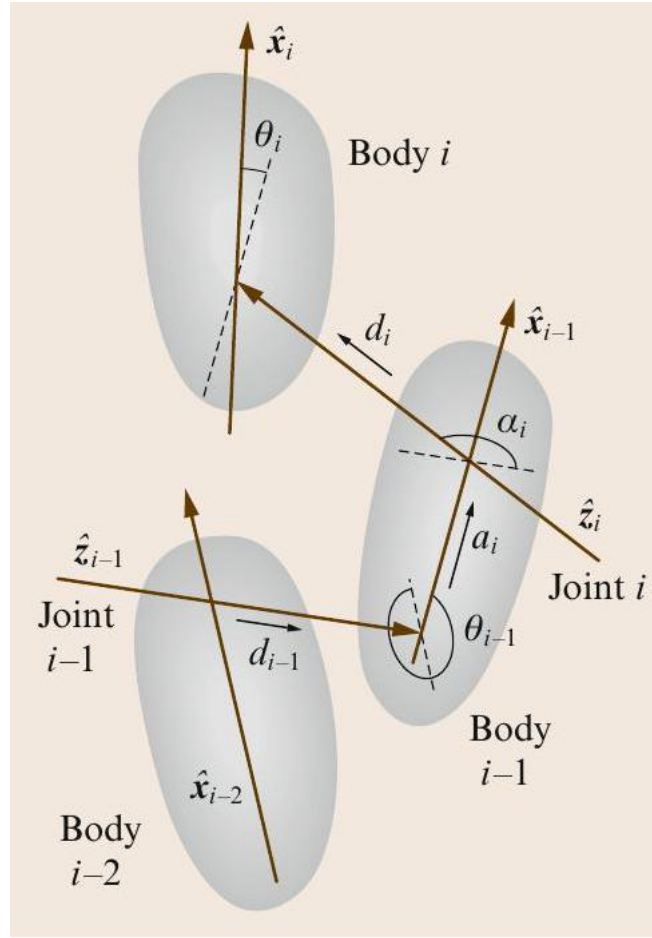


图 2.2: 在机器人机构中机体和关节编号的示意图, 对于坐标帧附着到机体的约定, 以及定位一个帧相对于另一个帧的四个参数 a_i, α_i, d_i 和 θ_i 的定义

在本手册中, 串联链式机构的约定如图 2.2 所示, 总结如下。机体和关节的编号遵循约定:

- 机器人机构的 N 个移动机体从 1 到 N 编号。基座的编号为 0。
- 机器人机构的 N 个关节从 1 到 N 编号, 关节 i 位于构件 $i-1$ 和 i 之间。

使用此编号方案, 坐标帧的附着遵循以下约定:

- \hat{z}_i 轴线沿着关节 i 的轴,
- \hat{x}_{i-1} 轴线位于 \hat{z}_{i-1} 和 \hat{z}_i 轴之间的公共法线上。

使用附着的帧, 将一个帧相对于另一个帧定位的四个参数定义为:

- a_i 是沿着 \hat{x}_{i-1} 从 \hat{z}_{i-1} 到 \hat{z}_i 的距离,
- α_i 是围绕 \hat{x}_{i-1} 从 \hat{z}_{i-1} 到 \hat{z}_i 的角度,
- d_i 是沿着 \hat{z}_i 从 \hat{x}_{i-1} 到 \hat{x}_i 的距离,
- θ_i 是围绕 \hat{z}_i 从 \hat{x}_{i-1} 到 \hat{x}_i 的角度。

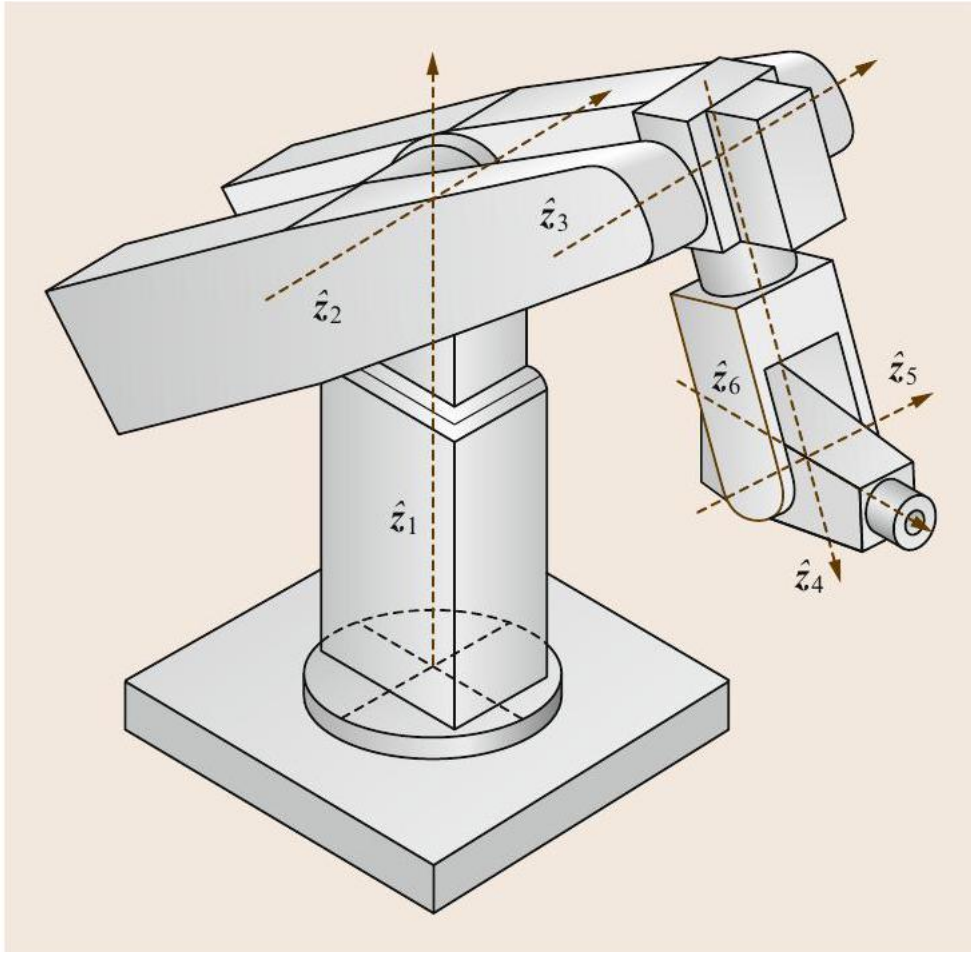


图 2.3: 示例六自由度串联链式机械臂，由无关节偏移的铰接臂和球形腕部组成

图 2.3 所示示例机械臂的几何参数列于表 2.7 中。该机械臂的所有关节都是旋转的，关节 1 具有垂直方向。关节 2 垂直于关节 1 并与其相交。关节 3 平行于关节 2，连杆 2 的长度为 a_3 。关节 4 垂直于关节 3 并与关节 3 相交。关节 5 同样与关节 4 垂直相交，与关节 3 的偏差为 d_4 。最后，关节 6 与关节 5 垂直相交。

根据该约定，坐标帧 i 可以通过以下方式相对于坐标帧 $i-1$ 定位：围绕 \hat{x}_{i-1} 轴旋转一个角度 α_i 、沿着 \hat{x}_{i-1} 平移距离 a_i 、通过围绕 \hat{z}_i 轴旋转一个角度 θ_i ，以及沿着 \hat{z}_i 平移距离 d_i 。对这些单独的变换进行串联，

$$\begin{aligned} & \mathbf{Rot}(\hat{x}_{i-1}, \alpha_i) \mathbf{Trans}(\hat{x}_{i-1}, a_i) \mathbf{Rot}(\hat{z}_i, \theta_i) \\ & \mathbf{Trans}(\hat{z}_i, d_i), \end{aligned} \quad (2.43)$$

表 2.7: 在图 2.3 中示例串联链式机械臂的几何参数

i	α_i	a_i	d_i	θ_i
1	0	0	0	θ_1
2	$-\frac{\pi}{2}$	0	0	θ_2
3	0	a_3	0	θ_3
4	$-\frac{\pi}{2}$	0	d_4	θ_4
5	$\frac{\pi}{2}$	0	0	θ_5
6	$-\frac{\pi}{2}$	0	0	θ_6

等效齐次变换为,

$${}^{i-1}\mathbf{T}_i = \begin{pmatrix} \cos \theta_i & -\sin \theta_i & 0 & a_i \\ \sin \theta_i \cos \alpha_i & \cos \theta_i \cos \alpha_i & -\sin \alpha_i & -\sin \alpha_i d_i \\ \sin \theta_i \sin \alpha_i & \cos \theta_i \sin \alpha_i & \cos \alpha_i & \cos \alpha_i d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (2.44)$$

在第 8 章中讨论了几何参数的识别。

2.5 工作空间

一般来说, 机械臂的工作空间是机械臂执行所有可能的运动时末端执行器扫出的总体积。工作空间由机械臂的几何结构和关节运动的限制确定。更具体的定义是, 可达工作空间是末端执行器可放置的点的总轨迹, 而灵巧工作空间 [2.31] 是末端执行器可放置且具有任意方向的点的子集。灵巧工作空间只存在于某些理想几何体中, 因此具有关节运动限制的实际工业机械臂几乎不具备灵巧工作空间。

许多串联链式机器人被设计为, 使其关节可分为区域结构和方向结构。区域结构中的关节完成末端执行器在空间中的定位, 方向结构中的关节完成末端执行器的方向。通常, 串联链式机械臂的内侧关节包括区域结构, 而外侧关节包括方向结构。另外, 由于棱柱关节不具备旋转能力, 因此通常不在方向结构中使用。

区域工作空间体积可以根据串联链式机械臂的已知几何结构和关节的运动限制来计算。在三个内侧关节构成区域结构的情况下, 首先计算外侧两个关节 (关节 2 和 3) 的工作空间面积, 然后通过对剩余内侧关节 (关节 1) 的关节变量进行积分来计算体积。对于棱柱形关节, 这只需将面积乘以棱柱形关节的总行程长度。在更常见的旋转关节情况下, 它涉及在旋转关节的整个运动范围内围绕关节轴旋转区域 [2.32]。根据 Pappus 定理, 相关体积 V 为

$$V = A\bar{r}\gamma, \quad (2.45)$$

其中, A 是区域, \bar{r} 是从区域中心到轴的距离, 并且 γ 是区域旋转的角度。通过跟踪末端执行器中的参考点的运动来确定区域的边界, 通常是用作方向结构的手腕的旋转中心。从两个关节中的每个关节处于运动极限并且关节 2 锁定开始, 关节 3 移动直到达到其第二运动极限。然后, 关节 3 被锁定, 关节 2 被释放以移动到其第二运动极限。关节 2 再次被锁定, 而关节 3 被释放以移动回其原始

运动极限。最后，关节 3 被锁定，关节 2 被释放以同样地移动到其原始运动极限。这样，参考点的轨迹是一条闭合曲线，其面积和形心可以用数学方法计算。

关于机械臂工作空间的更多细节可以在第 4 章和第 16 章找到。

2.6 正向运动学

串联链式机械臂的正向运动学问题是在给定所有关节位置和所有几何连杆参数值的情况下，找到末端执行器相对于基座的位置和方向。通常，固定在末端执行器中的帧称为工具帧 (*tool frame*)，并且当固定在最终连杆 N 中时，它通常在位置和方向上与帧 N 具有恒定偏移。同样，工位帧 (*station frame*) 通常位于基座中，以确定要执行的任务的位置。此帧通常在其相对于第 0 帧的位姿中具有恒定偏移，该帧也固定在基座中。

正向运动学问题的一个更一般的表达是，给定机器人机构的几何结构和等于机构自由度的若干关节位置数值，找到任意两个指定构件的相对位置和方向。正向运动学问题是开发机械臂协调算法的关键，因为关节位置通常由安装在关节上的传感器测量，并且需要计算关节轴相对于固定帧的位置。

在实践中，通过计算固定在末端执行器中的坐标帧与固定在基座中的另一坐标帧 (即工具帧和工位帧) 之间的变换来解决正向运动学问题。这对于串联链很简单，因为描述末端执行器相对于基座的位置的变换是通过简单地将固定在链中的相邻连杆中的帧之间的变换串联而获得的。第 2.4 节中介绍的机械臂几何表示的约定。将此简化为找到等效的 4×4 齐次变换矩阵，该矩阵将末端执行器坐标帧的空间位移与基座帧相关联。

对于图 2.3 所示的串联链式机械臂的例子，忽略工具帧和工位帧的增加，该变换为

$${}^0\mathbf{T}_6 = {}^0\mathbf{T}_1 {}^1\mathbf{T}_2 {}^2\mathbf{T}_3 {}^3\mathbf{T}_4 {}^4\mathbf{T}_5 {}^5\mathbf{T}_6. \quad (2.46)$$

表 2.8 包含使用表 2.7 和方程 (2.44) 计算的 ${}^0\mathbf{T}_6$ 的元素。

同样，齐次变换提供了紧凑的表示法，但对于解决正向运动学问题来说，计算效率很低。通过分离变换的位置和方向部分以消除矩阵的 0 和 1 元素的所有乘法，可以实现计算量的减少。在第 3 章中，使用在本章第 2.2.6 节中简要介绍的，并在第 3.2 节中详细说明的空间向量表示法进行计算。该方法不采用齐次变换，而是分离旋转矩阵和位置以实现计算效率。表 3.1 提供了详细的公式，其中空间变换的乘积与正向运动学问题特别相关。

运动学树是不包含闭环的机器人机构的一般结构，树状结构的正向运动学问题在第 3 章中讨论。由于存在附加约束，使得闭环链的正向运动学问题更加复杂。在第 18 章中包含了闭环链的求解方法。

2.7 逆向运动学

串联链式机械臂的逆向运动学问题是在给定末端执行器相对于基座的位置和方向以及所有几何连杆参数的情况下，找到关节位置的数值。同样，这是一个仅适用于串联链的简化语句。更一般的说法是：给定机构两个构件的相对位置和方向，找到所有关节位置的数值。这相当于在给定两个感兴趣的构件之间的齐次变换的情况下，找到所有关节位置。

在六自由度串联链式机械臂的一个常见情况下，已知的变换是 ${}^0\mathbf{T}_6$ 。回顾第 2.6 节中对这一变换的表述，很明显，串联链式机械臂的逆向运动学问题需要求解非线性方程组。对于六自由度机械臂，其中三个方程与齐次变换中的位置向量有关，另外三个方程与旋转矩阵有关。在后一种情况下，

表 2.8: 在图 2.3 中示例串联链式机械臂的正向运动学, 缩写为 $c_{\theta_i} := \cos \theta_i$ 和 $s_{\theta_i} := \sin \theta_i$

$${}^0\mathbf{T}_6 = \begin{pmatrix} r_{11} & r_{12} & r_{13} & {}^0p_6^x \\ r_{21} & r_{22} & r_{23} & {}^0p_6^y \\ r_{31} & r_{32} & r_{33} & {}^0p_6^z \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$\begin{aligned} r_{11} &= c_{\theta_1} (s_{\theta_2} s_{\theta_3} - c_{\theta_2} c_{\theta_3}) (s_{\theta_4} s_{\theta_6} - c_{\theta_4} c_{\theta_5} c_{\theta_6}) \\ &\quad - c_{\theta_1} s_{\theta_5} c_{\theta_6} (c_{\theta_2} s_{\theta_3} + s_{\theta_2} c_{\theta_3}) \\ &\quad + s_{\theta_1} (s_{\theta_4} c_{\theta_5} c_{\theta_6} + c_{\theta_4} s_{\theta_6}), \\ r_{21} &= s_{\theta_1} (s_{\theta_2} s_{\theta_3} - c_{\theta_2} c_{\theta_3}) (s_{\theta_4} s_{\theta_6} - c_{\theta_4} c_{\theta_5} c_{\theta_6}) \\ &\quad - s_{\theta_1} s_{\theta_5} c_{\theta_6} (c_{\theta_2} s_{\theta_3} + s_{\theta_2} c_{\theta_3}) \\ &\quad - c_{\theta_1} (s_{\theta_4} c_{\theta_5} c_{\theta_6} + c_{\theta_4} s_{\theta_6}), \\ r_{31} &= (c_{\theta_2} s_{\theta_3} + s_{\theta_2} c_{\theta_3}) (s_{\theta_4} s_{\theta_6} - c_{\theta_4} c_{\theta_5} c_{\theta_6}) \\ &\quad + s_{\theta_5} c_{\theta_6} (s_{\theta_2} s_{\theta_3} - c_{\theta_2} c_{\theta_3}), \\ r_{12} &= c_{\theta_1} (s_{\theta_2} s_{\theta_3} - c_{\theta_2} c_{\theta_3}) (c_{\theta_4} c_{\theta_5} s_{\theta_6} + s_{\theta_4} c_{\theta_6}) \\ &\quad + c_{\theta_1} s_{\theta_5} s_{\theta_6} (c_{\theta_2} s_{\theta_3} + s_{\theta_2} c_{\theta_3}) \\ &\quad + s_{\theta_1} (c_{\theta_4} c_{\theta_6} - s_{\theta_4} c_{\theta_5} s_{\theta_6}), \\ r_{22} &= s_{\theta_1} (s_{\theta_2} s_{\theta_3} - c_{\theta_2} c_{\theta_3}) (c_{\theta_4} c_{\theta_5} s_{\theta_6} + s_{\theta_4} c_{\theta_6}) \\ &\quad + s_{\theta_1} s_{\theta_5} s_{\theta_6} (c_{\theta_2} s_{\theta_3} + s_{\theta_2} c_{\theta_3}) \\ &\quad - c_{\theta_1} (c_{\theta_4} c_{\theta_6} - s_{\theta_4} c_{\theta_5} s_{\theta_6}), \\ r_{32} &= (c_{\theta_2} s_{\theta_3} + s_{\theta_2} c_{\theta_3}) (c_{\theta_4} c_{\theta_5} s_{\theta_6} + s_{\theta_4} c_{\theta_6}) \\ &\quad - s_{\theta_5} s_{\theta_6} (s_{\theta_2} s_{\theta_3} - c_{\theta_2} c_{\theta_3}), \\ r_{13} &= c_{\theta_1} c_{\theta_4} s_{\theta_5} (s_{\theta_2} s_{\theta_3} - c_{\theta_2} c_{\theta_3}) \\ &\quad - c_{\theta_1} c_{\theta_5} (c_{\theta_2} s_{\theta_3} + s_{\theta_2} c_{\theta_3}) \\ &\quad - s_{\theta_1} s_{\theta_4} s_{\theta_5}, \\ r_{23} &= s_{\theta_1} c_{\theta_4} s_{\theta_5} (s_{\theta_2} s_{\theta_3} - c_{\theta_2} c_{\theta_3}) \\ &\quad - s_{\theta_1} c_{\theta_5} (c_{\theta_2} s_{\theta_3} + s_{\theta_2} c_{\theta_3}) + c_{\theta_1} s_{\theta_4} s_{\theta_5}, \\ r_{33} &= c_{\theta_4} s_{\theta_5} (c_{\theta_2} s_{\theta_3} + s_{\theta_2} c_{\theta_3}) \\ &\quad + c_{\theta_5} (s_{\theta_2} s_{\theta_3} - c_{\theta_2} c_{\theta_3}), \\ {}^0p_6^x &= a_3 c_{\theta_1} c_{\theta_2} - d_4 c_{\theta_1} (c_{\theta_2} s_{\theta_3} + s_{\theta_2} c_{\theta_3}), \\ {}^0p_6^y &= a_3 s_{\theta_1} c_{\theta_2} - d_4 s_{\theta_1} (c_{\theta_2} s_{\theta_3} + s_{\theta_2} c_{\theta_3}), \\ {}^0p_6^z &= -a_3 s_{\theta_2} + d_4 (s_{\theta_2} s_{\theta_3} - c_{\theta_2} c_{\theta_3}). \end{aligned}$$

由于旋转矩阵中的相关性，这三个方程不能来自同一行或同一列。对于这些非线性方程，可能不存在解或存在多个解 [2.33]。要存在解，末端执行器的所需位置和方向必须位于机械臂的工作空间中。在存在解的情况下，它们通常不能以封闭形式呈现，因此需要数值方法。

2.7.1 封闭形式的解

封闭形式的解是受欢迎的，因为他们比数值解法更快，而且容易识别所有可能的解。闭式解的缺点是它们不具有普遍性，而是取决于机器人的方案。寻找闭式解的最有效方法是利用特定机构的特定几何特征的特殊技术。通常，只有具有特殊运动学结构的六自由度系统才能得到闭式解，其特点是在第 2.4 节中定义的几何参数大量为零值。大多数工业机械臂都具有这种结构，因为它允许使用更有效的协调软件。六自由度机械臂具有封闭形式的逆向运动学解的充分条件为 [2.34-36]：

1. 三个连续的旋转关节轴相交于一个公共点，如球形腕部。
2. 三个连续的旋转关节轴平行。

闭式求解方法一般分为代数方法和几何方法。

代数方法

代数方法包括识别包含关节变量的重要方程，并将其处理为可解形式。一个常用的策略化简为有单个变量的超越方程，例如，

$$C_1 \cos \theta_i + C_2 \sin \theta_i + C_3 = 0, \quad (2.47)$$

其中 C_1, C_2 和 C_3 是常数。这样一个方程的解是

$$\theta_i = 2 \tan^{-1} \left(\frac{C_2 \pm \sqrt{C_2^2 - C_3^2 + C_1^2}}{C_1 - C_3} \right). \quad (2.48)$$

一个或多个常数为零的特殊情况也很常见。

简化为一对方程，其具有形式为

$$C_1 \cos \theta_i + C_2 \sin \theta_i + C_3 = 0, \quad (2.49)$$

$$C_1 \sin \theta_i - C_2 \cos \theta_i + C_4 = 0, \quad (2.50)$$

这是另一个特别有用的策略，因为其结果只有一个解，

$$\theta_i = \text{Atan2}(-C_1 C_4 - C_2 C_3, C_2 C_4 - C_1 C_3). \quad (2.51)$$

几何方法

几何方法包括识别机械臂上的若干点，相对于这些点，位置和/或方向可以表示为关节变量的简化集合的函数。这通常相当于将空间问题分解为单独的平面问题。所得方程采用代数运算求解。上面列出的六自由度机械臂存在闭式解的两个充分条件使得问题能够分解为逆位置运动学和逆方向运动学。这是第 2.5 节中讨论的区域和方向结构的分解，并通过重写方程 (2.46) 找到其解，

$${}^0\mathbf{T}_6 {}^6\mathbf{T}_5 {}^5\mathbf{T}_4 {}^4\mathbf{T}_3 = {}^0\mathbf{T}_1 {}^1\mathbf{T}_2 {}^2\mathbf{T}_3. \quad (2.52)$$

表 2.9: 在图 2.3 中示例串联链式机械臂内铰接臂的逆位置运动学

$$\begin{aligned}
\theta_1 &= \text{Atan2}({}^0p_6^y, {}^0p_6^x) \\
&\quad \text{or } \text{Atan2}(-{}^0p_6^y, -{}^0p_6^x) \\
\theta_3 &= -\text{Atan2}\left(D, \pm\sqrt{1-D^2}\right), \\
&\quad \text{where } D := \frac{({}^0p_6^x)^2 + ({}^0p_6^y)^2 + ({}^0p_6^z)^2 - a_3^2 - d_4^2}{2a_3d_4}, \\
\theta_2 &= \text{Atan2}\left({}^0p_6^z, \sqrt{({}^0p_6^x)^2 + ({}^0p_6^y)^2}\right) \\
&\quad - \text{Atan2}(d_4 \cos \theta_3, a_3 - d_4 \sin \theta_3)
\end{aligned}$$

表 2.10: 在图 2.3 中示例串联链式机械臂内球形腕部的逆方向运动学，缩写为 $c_{\theta_i} := \cos \theta_i$ 和 $s_{\theta_i} := \sin \theta_i$

$$\begin{aligned}
\theta_5 &= \text{Atan2}\left(\pm\sqrt{1-(r_{13}s_{\theta_1}-r_{23}c_{\theta_1})^2}, r_{13}s_{\theta_1}-r_{23}c_{\theta_1}\right) \\
\theta_4 &= \text{Atan2}\left(\mp(r_{13}c_{\theta_1}+r_{23}s_{\theta_1})s_{(\theta_2+\theta_3)} \mp r_{33}c_{(\theta_2+\theta_3)}, \right. \\
&\quad \left. \pm(r_{13}c_{\theta_1}+r_{23}s_{\theta_1})c_{(\theta_2+\theta_3)} \mp r_{23}s_{(\theta_2+\theta_3)}\right) \\
\theta_6 &= \text{Atan2}\left(\pm(r_{12}s_{\theta_1}+r_{22}c_{\theta_1}), \pm(r_{11}s_{\theta_1}-r_{21}c_{\theta_1})\right), \\
&\quad \text{其中，对于 } \theta_5 \text{ 的 } \pm \text{ 选择决定对于 } \theta_4 \text{ 和 } \theta_6 \text{ 的所有后续的 } \pm \text{ 和 } \mp.
\end{aligned}$$

在图 2.3 中示例机械臂具有这种的结构，其区域结构通常被称为铰接或拟人手臂或肘部机械臂。表 2.9 总结了此类结构的逆位置运动学问题的解。因为 θ_1 有两个解，同样，每个 θ_1 解对应的 θ_2 和 θ_3 也有两个解，因此铰接臂机械臂的逆位置运动学问题共有四个解。方向结构只是一个球形腕部，表 2.10 总结了逆方向运动学问题的相应解。表 2.10 给出了 θ_5 的两个解，但 θ_4 和 θ_6 都仅对应一个解。因此，球面腕关节的逆方向运动学问题有两个解。结合区域和方向结构，在图 2.3 中机械臂的逆方向运动学的解的总数为 8。

2.7.2 数值方法

与用于寻找封闭形式解的代数和几何方法不同，数值方法不依赖于机器人方案，因此可以应用于任意运动学结构。数值方法的缺点是速度较慢，而且在某些情况下，不允许计算所有可能的解。对于一个仅具有旋转关节和棱柱关节的六自由度串联链式机械臂，平移和旋转方程始终可以简化为不大于 16 项的单项变量的多项式 [2.37]。因此，这样的机械臂可以有多达 16 个逆运动学问题的实数解 [2.38]。由于多项式方程的闭式解只有在多项式为四次或更少的情况下才有可能，因此许多机械臂的几何结构不能以封闭形式求解。一般而言，更大数量的非零几何参数对应于归约中的高次多项式。对于此类机械臂结构，最常见的数值方法可分为符号消元法、延拓法和迭代法。

符号消元法

符号消元法涉及分析操作，以从非线性方程组中消除变量，从而将其简化为一组较小的方程。Raghavan 和 Roth [2.39] 使用透析消元法将一般 6R 串联链式机械臂的逆方向运动学问题简化为 16

项多项式，并找到所有可能的解。该根解 (roots) 提供其中一个关节变量的所有解，而其它变量通过求解线性系统来计算。*Manocha* 和 *Canny* [2.40] 通过将问题重新表述为广义特征值问题，改进了该技术的数值性质。另一种消元方法是利用 *Gröbner* 基 [2.41,42]。

延拓法

延拓方法涉及跟踪从已知解的起始系统到目标系统的求解路径，在起始系统转化为目标系统的过程中寻求其解。这些技术已被应用于逆向运动学问题 [2.43]，并且可以利用多项式系统的特殊性来找到所有可能的解 [2.44]。

迭代法

可以采用一些不同的迭代方法来求解逆向运动学问题。大多数算法基于初始猜测收敛到一个单一的解，因此该猜测的质量对求解时间有很大影响。牛顿-拉弗森方法提供了一种基本方法，使用原始方程的一阶近似。*Pieper* [2.34] 是最早将该方法应用于逆向运动学的人之一，其他人也紧随其后 [2.45,46]。优化方法将问题描述为一个非线性优化问题，并使用搜索技术从初始猜测转移到一个解 [2.47,48]。求解运动速率控制是将问题转化为微分方程 [2.49]，并且可以使用改进的预测-校正算法来执行关节速度积分 [2.50]。基于控制理论的方法将微分方程转化为控制问题 [2.51]。区间分析法 [2.52] 可能是最有前途的迭代方法之一，因为它提供了快速收敛到一个解的方法，并可用于寻找所有可能的解。对于复杂机构，阻尼最小二乘法 [2.53] 特别有吸引力，更多的细节在第 10 章中提供。

2.8 正向瞬时运动学

串联链式机械臂的正瞬时运动学问题是：给定链中的所有成员的位置和所有关节的运动速率，找到末端执行器的总速度。这里，关节的运动速率是围绕旋转关节的旋转角速度或沿着棱柱关节滑动的平移速度。一个构件的总速度是固定在其上的坐标帧原点的速度与其角速度的组合。也就是说，总速度有六个独立的分量，因此完全代表了构件的速度场。需要注意的是，这个问题定义包括一个假设，即机构的位姿是完全已知的。在大多数情况下，这意味着在解决正向瞬时运动学问题之前，必须解决正向或逆向位置运动学问题。下面一节讨论的逆向瞬时运动学问题也是如此。为了研究动力学，在进行加速度分析时，正向瞬时运动学问题非常重要。在计算科里奥利和向心加速度分量时，需要各构件的总速度。

2.8.1 雅可比矩阵

正向位置运动学方程相对于时间进行微分，产生一组方程，其形式为

$${}^k\mathbf{v}_N = \mathbf{J}(\mathbf{q}) \dot{\mathbf{q}}, \quad (2.53)$$

其中 ${}^k\mathbf{v}_N$ 是末端执行器的空间速度，在任意帧 k 中表达， $\dot{\mathbf{q}}$ 是由关节速率组成的一个 n 维向量，并且 $\mathbf{J}(\mathbf{q})$ 是一个 $6 \times n$ 矩阵，其元素通常是 \mathbf{q} 的非线性函数。 $\mathbf{J}(\mathbf{q})$ 称为该代数系统的雅可比矩阵，并在与空间速度 ${}^k\mathbf{v}_N$ 相同的坐标帧中表达 [2.54]。或者，方程 (2.53) 可以表达为

$${}^k\mathbf{v}_N = \begin{bmatrix} \mathbf{J}_1 & \mathbf{J}_2 & \cdots & \mathbf{J}_N \end{bmatrix} \dot{\mathbf{q}}, \quad (2.54)$$

其中， N 是关节的数量（每个关节可能超过 1 个自由度），并且 \mathbf{J}_i 提供了 $\mathbf{J}(\mathbf{q})$ 矩阵的列，其对应于 \dot{q}_i 。如果已知关节位置，则方程 (2.53) 在关节速率中产生六个线性代数方程。如果给定关节速率，

则方程 (2.53) 的解是正向瞬时运动学问题的解。注意，只要所有的关节位置都是已知的，则 $\mathbf{J}(\mathbf{q})$ 可被视为对于此用途的已知矩阵。

使用在本章第 2.2.6 节中简要介绍并在第 3 章中详细说明的空间向量表示法，通过关节的自由模式 Φ_i 以及相关空间变换 ${}^j\mathbf{X}_i$ ，雅可比矩阵被轻松计算出来。

$$\mathbf{J}_i = {}^k\mathbf{X}_i \Phi_i. \quad (2.55)$$

为理解这一点的原因，注意 Φ_i 描述了关节 i 在局部坐标中创建的空间速度，而 ${}^k\mathbf{X}_i$ 将在帧 i 中表达的空间速度变换到帧 k 中。可以从 ${}^k\mathbf{X}_N$ 以及连杆到连杆的变换 ${}^i\mathbf{X}_{i-1}$ 中高效地计算 \mathbf{J}_i 的量。算法 2.1 包含一个以这种方式计算雅可比矩阵的列的算法。表 3.1 提供了利用空间变换矩阵结构以执行该算法所需乘法运算的高效方法。注意，使用正向运动学，以及可用简单的关节运动学计算出的每一个 ${}^i\mathbf{X}_{i-1}$ ，可以计算出 ${}^k\mathbf{X}_N$ 的量。因此，在算法 2.1 中所使用的算法将正向瞬时运动学问题简化为一个正向标准运动学问题。关于雅可比矩阵的更多信息可以在第 10 章中找到。

Algorithm 2.1 对于串联链式机构的雅可比矩阵计算算法

input: ${}^k\mathbf{X}_N, {}^N\mathbf{X}_{N-1}, \dots, {}^i\mathbf{X}_{i-1}, \dots, {}^2\mathbf{X}_1$

output: \mathbf{J}

$\mathbf{X} = {}^k\mathbf{X}_N$

for $i = N$ **to** 1 **do**

$\mathbf{J}_i = \mathbf{X} \Phi_i$

if $i > 1$ **then**

$\mathbf{X} = \mathbf{X} {}^i\mathbf{X}_{i-1}$

end if

end for

2.9 逆向瞬时运动学

从机器人协调的角度来看，重要的问题是逆向瞬时运动学问题。更多关于机器人协调的信息可以在第 7 章和第 8 章中找到。串联链式机械臂的逆向瞬时运动学问题是：给定链中所有成员的位置和末端执行器的总速度，找出所有关节的运动速率。当控制一个以点对点模式操作的工业机器人运动时，不仅需要计算由假定所期望的最终手位所需要的最终关节位置。还需要在初始位置和最终位置之间生成平滑的运动轨迹。当然，对于这个目的，有无数条可能的轨迹。然而，最直接、最成功的方法是采用基于逆向瞬时运动学问题的解的算法。这项技术起源于 Whitney [2.55] 和 Pieper [2.34] 的工作。

2.9.1 逆雅可比矩阵

当 \mathbf{v}_N 已知时，为求解通过将方程 (2.53) 分解为其分量方程而得到的关节速率的线性方程组，有必要对雅各布矩阵进行求逆。该方程变为

$$\dot{\mathbf{q}} = \mathbf{J}^{-1}(\mathbf{q}) \mathbf{v}_N. \quad (2.56)$$

由于 \mathbf{J} 是一个 6×6 矩阵，数值求逆不是很有吸引力。矩阵 \mathbf{J} 很可能变得奇异的 ($|\mathbf{J}| = 0$)，在这种情况下，逆矩阵不存在。关于奇异性的更多信息可以在第 4 章和第 18 章中找到。即使雅可比矩

阵没有变成奇异的，它也有可能是病态的，导致机械臂工作空间的大部分性能下降。大多数工业机器人的几何结构都很简单，雅可比矩阵可以通过解析进行求逆，从而得到一组关节速率的显式方程 [2.56-58]。与数值求逆相比，这大大减少了所需的运算次数。但是，对于更复杂的机械臂几何结构，数值求逆是唯一的解决选项。冗余机械臂的雅可比矩阵不是方形矩阵，因此不能求逆。第 10 章讨论了如何在这种情况下使用各种伪逆。

2.10 静态动力旋量变换

一般力系可以被证明为等效于单个力及其作用线附近的力矩。这称为动力旋量 (wrench)。动力旋量轴系统和瞬时旋量轴系统的几何结构之间存在深度等距 [2.59]。机械臂的静态动力旋量分析建立应用于末端执行器的动力旋量与应用于关节的力/扭矩之间的关系。这对于控制机械臂与其环境的交互至关重要。示例包括涉及固定或准固定工件的任务，例如以指定的力将部件插入到位，并将螺母拧紧至规定的扭矩。更多信息可以在第 9 章和第 37 章中找到。通过虚功原理，施加在末端执行器上的动力旋量与施加在关节上的力/扭矩之间的关系可被显示为

$$\boldsymbol{\tau} = \mathbf{J}^T \mathbf{f}, \quad (2.57)$$

其中， $\boldsymbol{\tau}$ 是对于 n 个自由度的机械臂所施加的关节力/扭矩的 n 维向量，并且 \mathbf{f} 是空间力向量

$$\mathbf{f} = \begin{pmatrix} \mathbf{n} \\ \mathbf{f} \end{pmatrix}, \quad (2.58)$$

其中， \mathbf{n} 和 \mathbf{f} 分别是施加到末端执行器的扭矩和力的向量，都是在相对于雅可比矩阵的坐标帧中表达。因此，与雅可比矩阵将关节速率映射到末端执行器的空间速度的方式相同，它的转置矩阵将应用于末端执行器的动力旋量映射到等效关节力/扭矩。与速度的情况一样，当雅可比矩阵不是方形矩阵时，逆矩阵关系不是唯一定义的。

2.11 结论与进一步的阅读

本章概述了运动学的基本原理如何应用于机器人机构。主题包括在空间中刚体的位置和方向的各种表示、运动自由度和伴随的关节数学模型、描述机器人机构的刚体和关节的几何表示、机械臂的工作空间、正向运动学和逆向运动学问题，正向和逆向瞬时运动学问题，包括雅可比矩阵的定义，最后是静态动力旋量变换的问题。本章当然不是对机器人运动学的全面介绍。幸运的是，许多优秀的文章对机器人学提供了广泛的介绍，并对运动学进行了重点关注 [2.17, 27,29,30,51,60-64]。

从历史的角度看，机器人技术从根本上改变了机构运动学领域的性质。在第一次为机器人建立协调方程之前 [2.34,55]，该领域的焦点几乎完全集中在单自由度机构上。这就是为什么继数值计算出现之后，机器人技术导致了机构运动学工作的复兴。更多的细节可以在第 4 章找到。该领域的发展一直在继续，因为它已经从对工业机器人的简单串联链的研究，本章分析的重点，扩展到并行机器人 (第 18 章)、仿人抓握器 (第 19 章)、机器人车辆 (第 17 章和第 24-26 章)，甚至小尺寸机器人 (第 27 章) 等领域。

2.12 References

- 2.1 W. R. Hamilton: On quaternions, or on a new system of imaginaries in algebra, Philos. Mag. 18 (2000)

- 2.2 E.B. Wilson: Vector Analysis (Dover, New York 1960), based upon the lectures of J.W. Gibbs (reprint of the 2nd edn. published by Charles Scribner's Sons, 1909)
- 2.3 H. Graßmann: Die Wissenschaft der extensiven Größe oder die Ausdehnungslehre (Wigand, Leipzig 1844)
- 2.4 J.M. McCarthy: Introduction to Theoretical Kinematics (MIT Press, Cambridge 1990)
- 2.5 W.K. Clifford: Preliminary sketch of bi-quaternions, Proc. Lond. Math. Soc. 4, 381-395 (1873)
- 2.6 A.P. Kotelnikov: Screw calculus and some applications to geometry and mechanics (Annal. Imp. Univ., Kazan 1895)
- 2.7 E. Study: Geometrie der Dynamen (Teubner, Leipzig 1903)
- 2.8 G.S. Chirikjian, A.B. Kyatkin: Engineering Applications of Noncommutative Harmonic Analysis (CRC, Boca Raton 2001)
- 2.9 R. von Mises: Anwendungen der Motorrechnung, Z. Angew. Math. Mech. 4(3), 193-213 (1924)
- 2.10 J.E. Baker, I.A. Parkin: Fundamentals of Screw Motion: Seminal Papers by Michel Chasles and Olinde Rodrigues, School of Information Technologies (University of Sydney, Sydney 2003), translated from O. Rodrigues: Des lois géométriques qui régissent les déplacements d'un système dans l'espace, J. Math. Pures Applicqu. Liouville 5, 380-440 (1840)
- 2.11 R.S. Ball: A Treatise on the Theory of Screws (Cambridge Univ. Press, Cambridge 1998)
- 2.12 J.K. Davidson, K.H. Hunt: Robots and Screw Theory: Applications of Kinematics and Statics to Robotics (Oxford Univ. Press, Oxford 2004)
- 2.13 K.H. Hunt: Kinematic Geometry of Mechanisms (Clarendon, Oxford 1978)
- 2.14 J.R. Phillips: Freedom in Machinery. Vol 1. Introducing Screw Theory (Cambridge Univ. Press, Cambridge 1984)
- 2.15 J.R. Phillips: Freedom in Machinery. Vol 2. Screw Theory Exemplified (Cambridge Univ. Press, Cambridge 1990)
- 2.16 G.S. Chirikjian: Rigid-body kinematics. In: Robotics and Automation Handbook, ed. by T. Kurfess (CRC, Boca Raton 2005), Chap. 2
- 2.17 R.M. Murray, Z. Li, S.S. Sastry: A Mathematical Introduction to Robotic Manipulation (CRC, Boca Raton 1994)
- 2.18 A. Karger, J. Novak: Space Kinematics and Lie Groups (Routledge, New York 1985)
- 2.19 R. von Mises: Motorrechnung, ein neues Hilfsmittel in der Mechanik, Z. Angew. Math. Mech. 2(2), 155-181 (1924)

- 2.20 J.D. Everett: On a new method in statics and kinematics, *Mess. Math.* 45, 36-37 (1875)
- 2.21 F. Reuleaux: *Kinematics of Machinery* (Dover, New York 1963), reprint of *Theoretische Kinematik*, 1875, in German
- 2.22 R. Featherstone: *Rigid Body Dynamics Algorithms* (Kluwer, Boston 2007)
- 2.23 K.J. Waldron: A method of studying joint geometry, *Mechan. Mach. Theory* 7, 347-353 (1972)
- 2.24 T.R. Kane, D.A. Levinson: *Dynamics, Theory and Applications* (McGraw-Hill, New York 1985)
- 2.25 J.L. Lagrange: *Oeuvres de Lagrange* (Gauthier-Villars, Paris 1867)
- 2.26 J. Denavit, R.S. Hartenberg: A kinematic notation for lower-pair mechanisms based on matrices, *J. Appl. Mech.* 22, 215-221 (1955)
- 2.27 W. Khalil, E. Dombre: *Modeling, Identification and Control of Robots* (Taylor Francis, New York 2002)
- 2.28 K.J. Waldron: A study of overconstrained linkage geometry by solution of closure equations, Part I: A method of study, *Mech. Mach. Theory* 8(1), 95-104 (1973)
- 2.29 R. Paul: *Robot Manipulators: Mathematics, Programming and Control* (MIT Press, Cambridge 1982)
- 2.30 J.J. Craig: *Introduction to Robotics: Mechanics and Control* (Addison-Wesley, Reading 1986)
- 2.31 K.J. Waldron, A. Kumar: The dextrous workspace, *ASME Mech. Conf.*, Los Angeles (1980), ASME paper No. 80-DETC-108
- 2.32 R. Vijaykumar, K.J. Waldron, M.J. Tsai: Geometric optimization of manipulator structures for working volume and dexterity, *Int. J. Robotics Res.* 5(2), 91-103 (1986)
- 2.33 J. Duffy: *Analysis of Mechanisms and Robot Manipulators* (Wiley, New York 1980)
- 2.34 D. Pieper: *The Kinematics of Manipulators Under Computer Control*, Ph.D. Thesis (Stanford University, Stanford 1968)
- 2.35 C.S.G. Lee: Robot arm kinematics, dynamics, and control, *Computer* 15(12), 62-80 (1982)
- 2.36 M.T. Mason: *Mechanics of Robotic Manipulation* (MIT Press, Cambridge 2001)
- 2.37 H.Y. Lee, C.G. Liang: A new vector theory for the analysis of spatial mechanisms, *Mech. Mach. Theory* 23(3), 209-217 (1988)
- 2.38 R. Manseur, K.L. Doty: A robot manipulator with 16 real inverse kinematic solutions, *Int. J. Robotics Res.* 8(5), 75-79 (1989)

- 2.39 M. Raghavan, B. Roth: Kinematic analysis of the 6R manipulator of general geometry, 5th Int. Symp. Robotics Res. (1990)
- 2.40 D. Manocha, J. Canny: Real Time Inverse Kinematics for General 6R Manipulators, Tech. Rep. (University of California, Berkeley 1992)
- 2.41 B. Buchberger: Applications of Gröbner bases in non-linear computational geometry, Lect. Notes Comput. Sci. 296, 52-80 (1989)
- 2.42 P. Kovacs: Minimum degree solutions for the inverse kinematics problem by application of the Buchberger algorithm. In: Advances in Robot Kinematics, ed. by S. Stifter, J. Lenarcic (Springer, New York 1991) pp. 326-334
- 2.43 L.W. Tsai, A.P. Morgan: Solving the kinematics of the most general six- and five-degree-of-freedom manipulators by continuation methods, ASME J. Mech. Transm. Autom. Des. 107, 189-195 (1985)
- 2.44 C.W. Wampler, A.P. Morgan, A.J. Sommese: Numerical continuation methods for solving polynomial systems arising in kinematics, ASME J. Mech. Des. 112, 59-68(1990)
- 2.45 R. Manseur, K.L. Doty: Fast inverse kinematics of 5-revolute-axis robot manipulators, Mechan. Mach. Theory 27(5), 587-597 (1992)
- 2.46 S.C.A. Thomopoulos, R.Y.J. Tam: An iterative solution to the inverse kinematics of robotic manipulators, Mechan. Mach. Theory 26(4), 359-373 (1991)
- 2.47 J.J. Uicker Jr., J. Denavit, R.S. Hartenberg: An interactive method for the displacement analysis of spatial mechanisms, J. Appl. Mech. 31, 309-314 (1964)
- 2.48 J. Zhao, N. Badler: Inverse kinematics positioning using nonlinear programming for highly articulated figures, Trans. Comput. Graph. 13(4), 313-336 (1994)
- 2.49 D.E. Whitney: Resolved motion rate control of manipulators and human prostheses, IEEE Trans. Man Mach. Syst. 10, 47-63 (1969)
- 2.50 H. Cheng, K. Gupta: A study of robot inverse kinematics based upon the solution of differential equations, J. Robotic Syst. 8(2), 115-175 (1991)
- 2.51 L. Sciavicco, B. Siciliano: Modeling and Control of Robot Manipulators (Springer, London 2000)
- 2.52 R.S. Rao, A. Asaithambi, S.K. Agrawal: Inverse kinematic solution of robot manipulators using interval analysis, ASME J. Mech. Des. 120(1), 147-150 (1998)
- 2.53 C.W. Wampler: Manipulator inverse kinematic solutions based on vector formulations and damped least squares methods, IEEE Trans. Syst. Man Cybern. 16, 93-101 (1986)
- 2.54 D.E. Orin, W.W. Schrader: Efficient computation of the jacobian for robot manipulators, Int. J. Robotics Res. 3(4), 66-75 (1984)

- 2.55 D.E. Whitney: The mathematics of coordinated control of prosthetic arms and manipulators, J. Dynamic Sys. Meas. Control 122, 303-309 (1972)
- 2.56 R.P. Paul, B.E. Shimano, G. Mayer: Kinematic control equations for simple manipulators, IEEE Trans. Syst. Man Cybern. 11(6), 339-455 (1981)
- 2.57 R.P. Paul, C.N. Stephenson: Kinematics of robot wrists, Int. J. Robotics Res. 20(1), 31-38 (1983)
- 2.58 R.P. Paul, H. Zhang: Computationally efficient kinematics for manipulators with spherical wrists based on the homogeneous transformation representation, Int. J. Robotics Res. 5(2), 32-44 (1986)
- 2.59 K.J. Waldron, K.H. Hunt: Series-parallel dualities in actively coordinated mechanisms, Int. J. Robotics Res. 10, 473-480 (1991)
- 2.60 H. Asada, J.J.E. Slotine: Robot Analysis and Control (Wiley, New York 1986)
- 2.61 F.L. Lewis, C.T. Abdallah, D.M. Dawson: Control of Robot Manipulators (Macmillan, New York 1993)
- 2.62 R.J. Schilling: Fundamentals of Robotics: Analysis and Control (Prentice Hall, Englewood Cliffs 1990)
- 2.63 M.W. Spong, M. Vidyasagar: Robot Dynamics and Control (Wiley, New York 1989)
- 2.64 T. Yoshikawa: Foundations of Robotics (MIT Press, Cambridge 1990)

3 动力学

Roy Featherstone, David E. Orin

运动的动力学方程提供了作用于机器人机构上的驱动力和接触力之间的关系，以及由此产生的加速度和运动轨迹。动力学对于机械设计、控制和模拟非常重要。一些算法在这些应用中十分重要，包括以下计算：逆向动力学 (*inverse dynamics*)、正向动力学 (*forward dynamics*)、关节空间惯量矩阵 (*joint-space inertia matrix*) 以及操作空间惯量矩阵 (*operational-space inertia matrix*)。本章提供了在机器人机构的刚体模型上执行这些计算的有效算法。该算法以最通用的形式给出，适用于具有通用的连接性、几何结构和关节类型的机器人机构。这些机构包括固定基座机器人、移动机器人和并联机器人机构。

除了需要计算效率外，算法还应以一组紧凑的方程式来制定，以便于开发和实现。在这方面，空间表示法的使用非常有效，并用于演示动力学算法。空间向量代数是一种简洁的向量表示法，对于描述刚体的速度、加速度、惯量等等，使用六维 (6-D) 向量和张量。

本章的目的是向读者介绍机器人动力学的主题，并以紧凑的形式向读者提供一组丰富的算法，这些算法可应用于它们特定的机器人机构。这些算法以表格形式呈现，以便随时访问。

3.1 概述

机器人动力学提供驱动力和接触力之间的关系，以及由此产生的加速度和运动轨迹。运动的动力学方程为许多计算算法提供了基础，这些算法在机械设计、控制和模拟中很有用。它们越来越多的应用于移动系统的计算机动画，特别是使用于人类和拟人机器人模型方面。本章介绍了机器人机构的基本动力学关系，以及最常见计算的高效算法。在这些算法中使用了空间向量表示法，这是一种利用 6-D 向量和张量的简洁表示法。

本章介绍四种主要计算过程的高效低阶算法：

1. 逆向动力学，其中所需的关节执行器扭矩/力是根据机器人轨迹 (位置、速度和加速度) 的规范计算出来的，
2. 正向动力学，其中指定了施加于关节传动器的扭矩/力，并确定了关节加速度，
3. 关节空间惯量矩阵，它将关节加速度映射到关节扭矩/力，以及
4. 操作空间惯量矩阵，将任务加速度映射到操作空间或笛卡尔空间中的任务力。

逆向动力学用于前馈控制和轨迹规划。模拟需要正向动力学。关节空间惯量 (质量) 矩阵用于分析、反馈控制以线性化动力学，并且是许多正向动力学公式的一个组成部分。操作空间惯量矩阵用于任务或末端执行器层级的控制。

3.1.1 空间向量表示法

第 3.2 节介绍了空间向量表示法，它用于以清晰简洁的方式表达本章中的算法。它最初由 Featherstone [3.1] 开发，以提供一种简洁的向量符号，对于描述刚体速度、加速度、惯量等等，使用 6-D 向量和张量。第 3.2 节解释了空间向量和算子的含义，并提供了空间的量与标准三维 (3-D) 的量和算子之间的对应关系的详细表格，以便于理解后面章节中的算法。文中还提供了对于空间演算的高

效计算机实现的公式。在空间向量的讨论中, 本文努力区分坐标向量和它们所代表的量。这说明了空间向量的一些重要特征。

3.1.2 正则方程组

运动的动力学方程在第 3.3 节中以两种基本形式提供: 关节空间公式和操作空间公式。在关节空间公式中的项传统上使用拉格朗日方法推导, 在这种方法中, 它们独立于任意参考坐标帧。拉格朗日公式提供了关节传动器的力与机构运动之间的关系的一种描述, 并从根本上对系统的动能和势能进行操作。由此产生的关节空间公式具有许多显著的特性, 已被证明对开发控制算法有用。本节还提供了与关节空间和操作空间公式中的各项的关系方程式, 以及一个冲击模型。

3.1.3 刚体系统动力学模型

在本章中的算法是基于模型的, 需要将机器人机构描述为其输入参数之一的数据结构。第 3.4 节给出了此模型的组件的描述: 连接图、连杆几何参数、连杆惯量参数和一组关节模型。连接性的描述是通用的, 因此它既包括运动学树, 也包括闭环机构。对于闭环机构的运动学树和生成树共享一个共同的表示法。为了描述连杆和关节几何结构, 每个关节都关联两个坐标帧, 每个坐标帧都分别附着到前导 (*predecessor*) 和后继 (*successor*) 连杆上。对于包含单自由度 (degree-of-freedom, DOF) 关节的串联机构, 这些帧定义为与 Craig [3.2] 改进的 Denavit-Hartenberg 约定兼容, 但仍然适用于包含一般多自由度关节的一般刚体系统。本文使用 Roberson 和 Schwertassek [3.3] 的一般关节模型描述了连接的连杆之间的关系。以一个拟人机器人为例, 说明了连杆和关节的编号方案, 以及描述连杆和关节的坐标帧的分配。该示例包括一个浮动基座, 还有旋转关节、万向节和球形关节。

3.1.4 运动学树

在第 3.5 节中介绍的算法, 对于任意机器人机构 (其做为一个运动学树), 计算它们的逆向动力学、正向动力学、关节空间惯量矩阵和操作空间惯量矩阵。本文为逆向动力学提供了一种 $O(n)$ 算法, 其中 n 是机构的自由度。它使用牛顿-欧拉公式来描述问题, 并基于 Luh 等人 [3.4] 非常高效的递归牛顿-欧拉算法 (recursive Newton-Euler algorithm, RNEA)。本文为正向动力学提供了两种算法。第一种是 $O(n)$ 铰接机体算法 (articulated-body algorithm, ABA), 由 Featherstone [3.1] 开发。第二种是 $O(n^2)$ 复合刚体机体算法 (composite-rigid-body algorithm, CRBA), 用于计算关节空间惯量矩阵 (joint-space inertia matrix, JSIM), 由 Walker 和 Orin [3.5] 开发。该矩阵与使用 RNEA 计算出的向量一起, 提供运动方程的系数, 然后可直接求解加速度 [3.5]。操作空间惯量矩阵 (operational-space inertia matrix, OSIM) 是一种铰接机体惯量, 并有两种算法被给出以计算它。第一种算法使用 OSIM 的基本定义, 第二种是一种直接的 $O(n)$ 算法, 其基于正向动力学问题高效解法。对于每一种算法的输入、输出、模型数据和伪代码都汇总在表格中, 以便随时访问。

3.1.5 运动学环路

上述算法仅适用于具有运动学树连接性的机构, 包括不分支的运动学链。对于闭环系统的正向动力学, 包括并联机器人机构, 一个最终的算法在第 3.6 节中提供。对于闭环系统的一个生成树, 该算法利用运动的动力学方程被制造出来, 并用环路闭约束方程对其进行补充。本文概述了三种不同的方法以求解所产生的线性方程组。如果 $n \gg n^c$, 其中 n^c 是由于环路闭合关节而产生的约束数量, 方法 2 特别有用。该方法提供了在生成树上使用 $O(n)$ 算法的机会 [3.6]。由于环路闭约束

方程应用在加速度层级上，因此使用标准 *Baumgarte* 稳定 [3.7]，以防止环路闭约束中的位置和速度误差累积。

本章最后一节给出了结论和建议，以供进一步阅读。机器人动力学领域一直是，并将继续是，一个非常丰富的研究领域。本节概述了在该领域所作的主要贡献和最常被引用的工作。遗憾的是，篇幅不允许我们对这一领域的广泛文献进行全面的回顾。

3.2 空间向量表示法

对于机器人动力学，没有单一的标准表示法。目前使用的表示法包括 3-D 向量、 4×4 矩阵和几种类型的 6-D 向量：旋量 (screw)、矩量 (motor)、李代数元素和空间向量。六维向量表示法通常是最好的，比 3-D 向量更紧凑，比 4×4 矩阵更强大。因此，我们在本章中使用 6-D 向量。特别是，我们将使用在文献 [3.8] 中描述的空间向量代数。本节提供空间向量的简要概述。 4×4 矩阵表示法的描述可在文献 [3.2,9] 中找到，其它 6-D 向量表示法的描述可在文献 [3.10,12] 中找到。

在本手册中，向量通常用粗斜体字母标志 (如 \mathbf{f}, \mathbf{v})。但是，为了避免一些名称冲突，我们将使用竖直的粗体字母来标志空间向量 (例如 \mathbf{f}, \mathbf{v})。注意，这只适用于向量，而不适用于张量。另外，仅在本节中，我们将为坐标向量加下划线，以区分它们与它们所表示的向量 (例如， \underline{v} 和 \underline{v} ，表示 v 和 \mathbf{v})。

3.2.1 运动与力

出于数学原因，区分描述刚体运动的向量和描述作用于其上的力的向量是有用的。因此，我们将运动向量放置在称为 M^6 的向量空间中，并将力向量放置在称为 F^6 的空间中。(上标表示维度。) 运动向量描述速度、加速度、无穷小位移和运动自由度的方向等量；力向量描述力、动量、接触法线，如此等等。

3.2.2 基向量

假设 \mathbf{v} 是一个 3-D 向量，并且 $\underline{v} = (v_x, v_y, v_z)^T$ 是笛卡尔坐标向量，表示在正交基 $\{\hat{x}, \hat{y}, \hat{z}\}$ 中的 \mathbf{v} 。 \mathbf{v} 和 \underline{v} 之间的关系由一下公式给出为

$$\mathbf{v} = \hat{x}v_x + \hat{y}v_y + \hat{z}v_z.$$

同样的想法也适用于空间向量，除了使用 Plücker 坐标代替笛卡尔坐标，并使用 Plücker 基代替正交基。

Plücker 坐标在第 2.2.6 节中介绍，但基向量如图 3.1 所示。总共有 12 个基向量：6 个用于运动向量，6 个用于力。给定笛卡尔坐标帧 O_{xyz} ，Plücker 基向量定义如下：围绕有向直线 Ox, Oy 和 Oz 的三个单位旋转，用 $\mathbf{d}_{Ox}, \mathbf{d}_{Oy}$ 和 \mathbf{d}_{Oz} 标志；在 x, y 和 z 方向上的三个单位平移，用 $\mathbf{d}_x, \mathbf{d}_y$ 和 \mathbf{d}_z 标志；围绕 x, y 和 z 方向的三个单位力偶，用 $\mathbf{e}_x, \mathbf{e}_y$ 和 \mathbf{e}_z 标志；以及沿着直线 Ox, Oy 和 Oz 的三个单位力，用 $\mathbf{e}_{Ox}, \mathbf{e}_{Oy}$ 和 \mathbf{e}_{Oz} 标志。

3.2.3 空间的速度与力

给定任意点 O ，刚体的速度可以用一对 3-D 向量 $\boldsymbol{\omega}$ 和 \mathbf{v}_O 来描述，它们指定机体的角速度，以及当前在 O 点处的机体固定点的线速度。注意， \mathbf{v}_O 不是 O 点本身的速度，而是机体固定点的速度，其在当前时刻恰好与 O 重合。

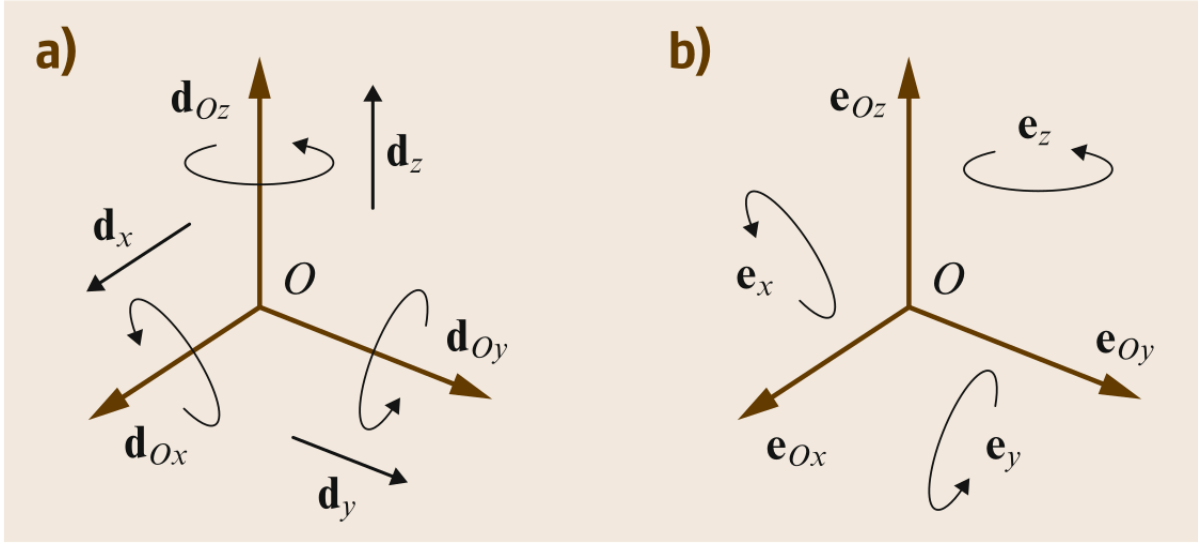


图 3.1: Plücker 基向量用于 (a) 运动, 以及 (b) forces

同一个刚体的速度也可以用单个空间运动向量 $\mathbf{v} \in M^6$ 来描述。为了从 $\boldsymbol{\omega}$ 和 \mathbf{v}_O 中获取 \mathbf{v} , 我们首先引入一个笛卡尔坐标帧 O_{xyz} , 其原点在 O 点处。该坐标帧定义了 $\boldsymbol{\omega}$ 和 \mathbf{v}_O 的笛卡尔坐标帧, 并也是 \mathbf{v} 的 Plücker 坐标帧。给定这些坐标系统, 可以证明

$$\begin{aligned} \mathbf{v} = & \mathbf{d}_{Ox}\omega_x + \mathbf{d}_{Oy}\omega_y + \mathbf{d}_{Oz}\omega_z + \mathbf{d}_x v_{Ox} \\ & + \mathbf{d}_y v_{Oy} + \mathbf{d}_z v_{Oz}, \end{aligned} \quad (3.1)$$

其中 ω_x, \dots, v_{Oz} 是在 O_{xyz} 中 $\boldsymbol{\omega}$ 和 \mathbf{v}_O 的笛卡尔坐标。因此, \mathbf{v} 的 Plücker 坐标是 $\boldsymbol{\omega}$ 和 \mathbf{v}_O 的笛卡尔坐标。在 O_{xyz} 中表示的 \mathbf{v} 的坐标向量可被写为

$$\underline{\mathbf{v}}_O = \begin{pmatrix} \omega_x \\ \vdots \\ v_{Oz} \end{pmatrix} = \begin{pmatrix} \underline{\boldsymbol{\omega}} \\ \underline{\mathbf{v}}_O \end{pmatrix}. \quad (3.2)$$

该方程最右边的表示法只是 Plücker 坐标列表的一个简单缩写。

空间力的定义非常类似。给定任意点 O , 作用在单个刚体上的任意力系都等效于作用在通过点 O 的直线上的单个力 \mathbf{f} , 以及一个纯力偶 \mathbf{n}_O , 它是力系围绕点 O 的力矩。因此, 两个向量 \mathbf{f} 和 \mathbf{n}_O 描述作用在刚体上的力的方式与 $\boldsymbol{\omega}$ 和 \mathbf{v}_O 描述其速度的方式非常相同。同样的力也可以用单个空间力向量 $\mathbf{f} \in F^6$ 来描述。如前所述, 引入帧 O_{xyz} , 可以证明

$$\begin{aligned} \mathbf{f} = & \mathbf{e}_x n_{Ox} + \mathbf{e}_y n_{Oy} + \mathbf{e}_z n_{Oz} + \mathbf{e}_{Ox} f_x \\ & + \mathbf{e}_{Oy} f_y + \mathbf{e}_{Oz} f_z, \end{aligned} \quad (3.3)$$

其中 n_{Ox}, \dots, f_z 是在 O_{xyz} 中 \mathbf{n}_O 和 \mathbf{f} 的笛卡尔坐标。在 O_{xyz} 中表示的 \mathbf{f} 的坐标向量可被写为

$$\underline{\mathbf{f}}_O = \begin{pmatrix} n_{Ox} \\ \vdots \\ f_z \end{pmatrix} = \begin{pmatrix} \underline{\mathbf{n}}_O \\ \underline{\mathbf{f}} \end{pmatrix}. \quad (3.4)$$

同样, 这些是 \mathbf{f} 在 O_{xyz} 中的 Plücker 坐标, 最右边的表示法只是 Plücker 坐标列表的一个简单缩写。

3.2.4 加法与标量乘法

空间向量的行为在加法和标量乘法下很明显。例如，如果 \mathbf{f}_1 和 \mathbf{f}_2 都作用于同一个刚体，则它们的合力为 $\mathbf{f}_1 + \mathbf{f}_2$ ；如果两个不同机体的速度为 \mathbf{v}_1 和 \mathbf{v}_2 ，则第二个机体相对于第一个机体的速度为 $\mathbf{v}_2 - \mathbf{v}_1$ ；并且如果 \mathbf{f} 标志沿空间中特定直线作用的 1 N 的力，则 $\alpha\mathbf{f}$ 标志沿同一直线作用的 αN 的力。

3.2.5 标量积

标量积定义在任意两种空间向量之间提供，其中一个为运动的空间向量，另一个为力的空间向量。给定任意 $\mathbf{m} \in \mathbf{M}^6$ 和 $\mathbf{f} \in \mathbf{F}^6$ ，标量积可以写成 $\mathbf{f} \cdot \mathbf{m}$ 或 $\mathbf{m} \cdot \mathbf{f}$ ，并表达力 \mathbf{f} 作用在具有运动 \mathbf{m} 的机体上所做的功。类似 $\mathbf{f} \cdot \mathbf{f}$ 和 $\mathbf{m} \cdot \mathbf{m}$ 的表达式没有定义。如果 $\underline{\mathbf{m}}$ 和 $\underline{\mathbf{f}}$ 是在同一个坐标系统中表示 \mathbf{m} 和 \mathbf{f} 的坐标向量，则

$$\mathbf{m} \cdot \mathbf{f} = \underline{\mathbf{m}}^T \underline{\mathbf{f}}. \quad (3.5)$$

3.2.6 坐标变换

运动向量和力向量遵循不同的变换规则。设 A 和 B 为两个坐标帧，每个坐标帧定义一个同名坐标系统；并设 $\underline{\mathbf{m}}_A, \underline{\mathbf{m}}_B, \underline{\mathbf{f}}_A$ 和 $\underline{\mathbf{f}}_B$ 是分别表示在 A 和 B 坐标中的空间向量 $\mathbf{m} \in \mathbf{M}^6$ 和 $\mathbf{f} \in \mathbf{F}^6$ 。则其变换规则为

$$\underline{\mathbf{m}}_B = {}^B\mathbf{X}_A \underline{\mathbf{m}}_A \quad (3.6)$$

以及

$$\underline{\mathbf{f}}_B = {}^B\mathbf{X}_A^F \underline{\mathbf{f}}_A, \quad (3.7)$$

其中 ${}^B\mathbf{X}_A$ 和 ${}^B\mathbf{X}_A^F$ 分别是运动和力向量从 A 到 B 的坐标变换矩阵。这些矩阵由恒等式相关联

$${}^B\mathbf{X}_A^F \equiv ({}^B\mathbf{X}_A)^{-T} \equiv ({}^A\mathbf{X}_B)^T. \quad (3.8)$$

假设帧 A 相对于帧 B 的位置和方向由一个位置向量 ${}^B\underline{\mathbf{p}}_A$ 和一个 3×3 旋转矩阵 ${}^B\mathbf{R}_A$ 所描述，如第 2.2 节所述。则对于 ${}^B\mathbf{X}_A$ 的公式为

$$\begin{aligned} {}^B\mathbf{X}_A &= \begin{pmatrix} 1 & 0 \\ \mathbf{S}({}^B\underline{\mathbf{p}}_A) & 1 \end{pmatrix} \begin{pmatrix} {}^B\mathbf{R}_A & 0 \\ 0 & {}^B\mathbf{R}_A \end{pmatrix} \\ &= \begin{pmatrix} {}^B\mathbf{R}_A & 0 \\ \mathbf{S}({}^B\underline{\mathbf{p}}_A) {}^B\mathbf{R}_A & {}^B\mathbf{R}_A \end{pmatrix}, \end{aligned} \quad (3.9)$$

并且其逆矩阵为

$${}^A\mathbf{X}_B = \begin{pmatrix} {}^A\mathbf{R}_B & 0 \\ 0 & {}^A\mathbf{R}_B \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -\mathbf{S}({}^B\underline{\mathbf{p}}_A) & 1 \end{pmatrix}. \quad (3.10)$$

量 $\mathbf{S}(\underline{\mathbf{p}})$ 是对于任意 3-D 向量 $\underline{\mathbf{v}}$ 满足 $\mathbf{S}(\underline{\mathbf{p}})\underline{\mathbf{v}} = \underline{\mathbf{p}} \times \underline{\mathbf{v}}$ 的斜对称矩阵。它由以下方程定义

$$\mathbf{S}(\underline{\mathbf{p}}) = \begin{pmatrix} 0 & -p_z & p_y \\ p_z & 0 & -p_x \\ -p_y & p_x & 0 \end{pmatrix}. \quad (3.11)$$

3.2.7 向量积

在空间向量上定义了两种向量 (交叉) 积。第一种获取两个运动向量参数, 并产生运动向量结果。它由以下公式定义

$$\begin{aligned}\underline{\mathbf{m}}_1 \times \underline{\mathbf{m}}_2 &= \begin{pmatrix} \underline{\mathbf{m}}_1 \\ \underline{\mathbf{m}}_{1O} \end{pmatrix} \times \begin{pmatrix} \underline{\mathbf{m}}_2 \\ \underline{\mathbf{m}}_{2O} \end{pmatrix} \\ &= \begin{pmatrix} \underline{\mathbf{m}}_1 \times \underline{\mathbf{m}}_2 \\ \underline{\mathbf{m}}_1 \times \underline{\mathbf{m}}_{2O} + \underline{\mathbf{m}}_{1O} \times \underline{\mathbf{m}}_2 \end{pmatrix}.\end{aligned}\quad (3.12)$$

第二种将运动向量作为左侧参数, 以及将力向量作为右侧参数, 并产生一个力-向量结果。它由以下公式定义

$$\begin{aligned}\underline{\mathbf{m}} \times \underline{\mathbf{f}} &= \begin{pmatrix} \underline{\mathbf{m}} \\ \underline{\mathbf{m}}_O \end{pmatrix} \times \begin{pmatrix} \underline{\mathbf{f}}_O \\ \underline{\mathbf{f}} \end{pmatrix} \\ &= \begin{pmatrix} \underline{\mathbf{m}} \times \underline{\mathbf{f}}_O + \underline{\mathbf{m}}_O \times \underline{\mathbf{f}} \\ \underline{\mathbf{m}} \times \underline{\mathbf{f}} \end{pmatrix}.\end{aligned}\quad (3.13)$$

该积出现在微分公式中。

可以定义一个空间交积算子, 与方程 (3.11) 相类似, 如下所示

$$\mathbf{S}(\underline{\mathbf{m}}) = \begin{pmatrix} \mathbf{S}(\underline{\mathbf{m}}) & \mathbf{0} \\ \mathbf{S}(\underline{\mathbf{m}}_O) & \mathbf{S}(\underline{\mathbf{m}}) \end{pmatrix}, \quad (3.14)$$

在这种情况下

$$\underline{\mathbf{m}}_1 \times \underline{\mathbf{m}}_2 = \mathbf{S}(\underline{\mathbf{m}}_1) \underline{\mathbf{m}}_2, \quad (3.15)$$

但是

$$\underline{\mathbf{m}} \times \underline{\mathbf{f}} = -\mathbf{S}(\underline{\mathbf{m}})^T \underline{\mathbf{f}}. \quad (3.16)$$

注意, $\mathbf{S}(\underline{\mathbf{m}})$ 将运动向量映射为运动向量, 但是 $\mathbf{S}(\underline{\mathbf{m}})^T$ 将力向量映射为力向量。

3.2.8 微分

空间向量的导数定义为

$$\frac{d}{dx} \mathbf{s}(x) = \lim_{\delta x \rightarrow 0} \frac{\mathbf{s}(x + \delta x) - \mathbf{s}(x)}{\delta x}, \quad (3.17)$$

其中 \mathbf{s} 代表任意空间向量。导数是一个与被微分的空间向量相同类型 (运动或力) 的空间向量。

在移动坐标系中对空间坐标向量进行微分的公式是

$$\left(\frac{d}{dt} \mathbf{s} \right)_A = \frac{d}{dt} \mathbf{s}_A + \mathbf{v}_A \times \mathbf{s}_A, \quad (3.18)$$

其中 \mathbf{s} 为任意空间向量, $d\mathbf{s}/dt$ 为 \mathbf{s} 的时间导数, A 为移动坐标系, $(d\mathbf{s}/dt)_A$ 为在 A 坐标中表示 $d\mathbf{s}/dt$ 的坐标向量, \mathbf{s}_A 为在 A 坐标中表示 \mathbf{s} 的坐标向量, $d\mathbf{s}_A/dt$ 为 \mathbf{s}_A 的时间导数 (其为分量导数, 因为 \mathbf{s}_A 为坐标向量), 并且 \mathbf{v}_A 是 A 坐标帧的速度, 在 A 坐标中表达。

空间向量的时间导数, 仅因其移动而改变, 给出为

$$\frac{d}{dt} \mathbf{s} = \mathbf{v} \times \mathbf{s}, \quad (3.19)$$

其中, \mathbf{v} 是 \mathbf{s} 的速度。此公式对于微分某些量是很有用的, 即那些在自身一侧不发生变化, 但附着到移动刚体 (例如关节轴向量) 的量。

3.2.9 加速度

空间加速度被定义为空间速度的变化率。不幸的是，这意味着空间加速度与教科书中关于刚体加速度的经典定义不同，那里被我们称为经典加速度 (*classical acceleration*)。本质上，差异可以总结如下

$$\underline{\mathbf{a}} = \begin{pmatrix} \dot{\underline{\boldsymbol{\omega}}} \\ \dot{\underline{\mathbf{v}}}_O \end{pmatrix} \quad \text{and} \quad \underline{\mathbf{a}}' = \begin{pmatrix} \dot{\underline{\boldsymbol{\omega}}} \\ \dot{\underline{\mathbf{v}}}'_O \end{pmatrix}, \quad (3.20)$$

其中 $\underline{\mathbf{a}}$ 是空间加速度， $\underline{\mathbf{a}}'$ 是经典加速度， $\dot{\underline{\mathbf{v}}}_O$ 是 $\underline{\mathbf{v}}_O$ 的导数，其取自固定在空间中 O 的分量，并且 $\dot{\underline{\mathbf{v}}}'_O$ 是 $\underline{\mathbf{v}}'_O$ 的导数，其取自固定在机体中 O 的分量。两个加速度之间的关系如下

$$\underline{\mathbf{a}}' = \underline{\mathbf{a}} + \begin{pmatrix} \mathbf{0} \\ \underline{\boldsymbol{\omega}} \times \underline{\mathbf{v}}_O \end{pmatrix}. \quad (3.21)$$

如果 \mathbf{r} 是一个位置向量，给出在点 O 处的机体固定点相对于任意固定点的位置，则

$$\begin{aligned} \mathbf{v}_O &= \dot{\mathbf{r}}, \\ \dot{\mathbf{v}}'_O &= \ddot{\mathbf{r}}, \\ \dot{\mathbf{v}}_O &= \ddot{\mathbf{r}} - \boldsymbol{\omega} \times \mathbf{v}_O. \end{aligned} \quad (3.22)$$

实际区别是空间加速度更容易使用。例如，如果机体 B_1 和 B_2 的速度分别为 \mathbf{v}_1 和 \mathbf{v}_2 ，并且 \mathbf{v}_{rel} 是 B_2 相对于 B_1 的相对速度，则

$$\mathbf{v}_2 = \mathbf{v}_1 + \mathbf{v}_{\text{rel}}.$$

它们的空间加速度之间的关系可以通过对速度公式进行简单的微分获得

$$\frac{d}{dt}(\mathbf{v}_2 = \mathbf{v}_1 + \mathbf{v}_{\text{rel}}) \Rightarrow \underline{\mathbf{a}}_2 = \underline{\mathbf{a}}_1 + \underline{\mathbf{a}}_{\text{rel}}.$$

注意，空间加速度是由加法组成的，与速度完全相同。没有科里奥利或离心项可担心。这是对经典加速度组成公式的一个重大改进，例如在文献 [3.2,13,14] 中的那些公式。

3.2.10 空间动量

假设一个刚体的质量为 m ，质心在 C 处，以及围绕 C 的转动惯量为 $\bar{\mathbf{I}}^{\text{cm}}$ (图 3.2)。如果该机体以 $\underline{\mathbf{v}}_C = \begin{pmatrix} \underline{\boldsymbol{\omega}}^T & \underline{\mathbf{v}}_C^T \end{pmatrix}^T$ 的空间速度移动，则其线动量为 $\underline{\mathbf{h}} = m\underline{\mathbf{v}}_C$ ，并且其固有角动量为 $\underline{\mathbf{h}}_C = \bar{\mathbf{I}}^{\text{cm}}\underline{\boldsymbol{\omega}}$ 。其围绕一般点 O 的动量矩为 $\underline{\mathbf{h}}_O = \underline{\mathbf{h}}_C + \underline{\mathbf{c}} \times \underline{\mathbf{h}}$ ，其中 $\underline{\mathbf{c}} = \overrightarrow{OC}$ 。我们可以将这些向量组合成一个空间动量向量，如下所示

$$\underline{\mathbf{h}}_C = \begin{pmatrix} \underline{\mathbf{h}}_C \\ \underline{\mathbf{h}} \end{pmatrix} = \begin{pmatrix} \bar{\mathbf{I}}^{\text{cm}}\underline{\boldsymbol{\omega}} \\ m\underline{\mathbf{v}}_C \end{pmatrix} \quad (3.23)$$

并且

$$\underline{\mathbf{h}}_O = \begin{pmatrix} \underline{\mathbf{h}}_O \\ \underline{\mathbf{h}} \end{pmatrix} = \begin{pmatrix} \mathbf{1} & \mathbf{S}(\underline{\mathbf{c}}) \\ \mathbf{0} & \mathbf{1} \end{pmatrix} \underline{\mathbf{h}}_C. \quad (3.24)$$

空间动量是一个力向量，并进行相应地变换。

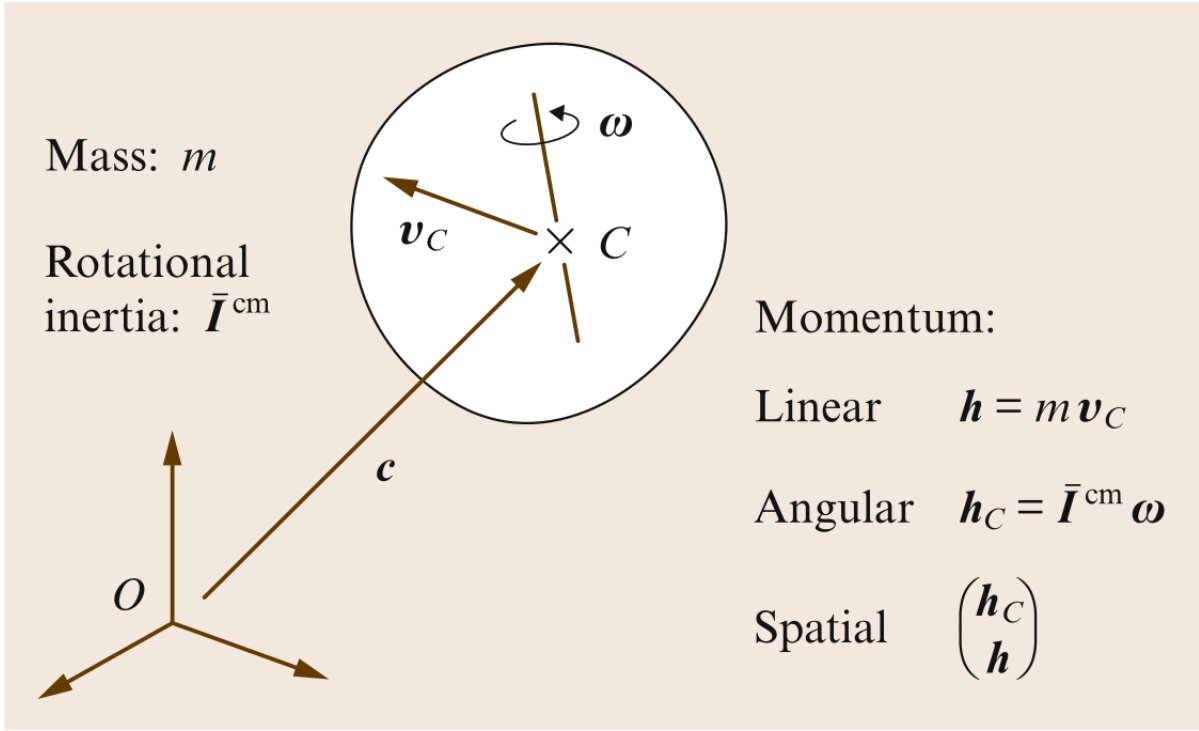


图 3.2: 空间动量

3.2.11 空间惯量

刚体的空间动量是其空间惯量与速度的乘积

$$\mathbf{h} = \mathbf{I} \mathbf{v}, \quad (3.25)$$

其中 \mathbf{I} 是空间惯量。用在 C 处的 Plücker 坐标表达，我们有

$$\mathbf{h}_C = \mathbf{I}_C \mathbf{v}_C, \quad (3.26)$$

这意味着

$$\mathbf{I}_C = \begin{pmatrix} \bar{I}^{\text{cm}} & \mathbf{0} \\ \mathbf{0} & m\mathbf{1} \end{pmatrix}. \quad (3.27)$$

这是在刚体的质心处表达的刚体空间惯量的一般公式。为了在另一点 O 处表达，我们进行如下操作。从方程 (3.24)，(3.26)，和 (3.27)

$$\begin{aligned} \underline{\mathbf{h}}_O &= \begin{pmatrix} \mathbf{1} & \mathbf{S}(\underline{\mathbf{c}}) \\ \mathbf{0} & \mathbf{1} \end{pmatrix} \begin{pmatrix} \bar{I}^{\text{cm}} & \mathbf{0} \\ \mathbf{0} & m\mathbf{1} \end{pmatrix} \underline{\mathbf{v}}_C \\ &= \begin{pmatrix} \mathbf{1} & \mathbf{S}(\underline{\mathbf{c}}) \\ \mathbf{0} & \mathbf{1} \end{pmatrix} \begin{pmatrix} \bar{I}^{\text{cm}} & \mathbf{0} \\ \mathbf{0} & m\mathbf{1} \end{pmatrix} \begin{pmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{S}(\underline{\mathbf{c}})^T & \mathbf{1} \end{pmatrix} \underline{\mathbf{v}}_O \\ &= \begin{pmatrix} \bar{I}^{\text{cm}} + m\mathbf{S}(\underline{\mathbf{c}})\mathbf{S}(\underline{\mathbf{c}})^T & m\mathbf{S}(\underline{\mathbf{c}}) \\ m\mathbf{S}(\underline{\mathbf{c}})^T & m\mathbf{1} \end{pmatrix} \underline{\mathbf{v}}_O; \end{aligned}$$

但我们还有 $\underline{\mathbf{h}}_O = \mathbf{I}_O \underline{\mathbf{v}}_O$ ，所以

$$\mathbf{I}_O = \begin{pmatrix} \bar{I}^{\text{cm}} + m\mathbf{S}(\underline{\mathbf{c}})\mathbf{S}(\underline{\mathbf{c}})^T & m\mathbf{S}(\underline{\mathbf{c}}) \\ m\mathbf{S}(\underline{\mathbf{c}})^T & m\mathbf{1} \end{pmatrix}. \quad (3.28)$$

该方程也可被写为

$$\mathbf{I}_O = \begin{pmatrix} \bar{\mathbf{I}}_O & m\mathbf{S}(\underline{\mathbf{c}}) \\ m\mathbf{S}(\underline{\mathbf{c}})^T & m\mathbf{1} \end{pmatrix}, \quad (3.29)$$

其中

$$\bar{\mathbf{I}}_O = \bar{\mathbf{I}}^{\text{cm}} + m\mathbf{S}(\underline{\mathbf{c}})\mathbf{S}(\underline{\mathbf{c}})^T \quad (3.30)$$

是围绕点 O 的刚体的转动惯量。

空间惯量矩阵是对称的和正定的。在一般情况下，需要 21 个数字来指定一个空间惯量 (例如，对于铰接体或操作空间的惯量)；但刚体惯量只需要 10 个参数：质量、质心坐标以及 $\bar{\mathbf{I}}^{\text{cm}}$ 或 $\bar{\mathbf{I}}_O$ 的六个独立元素。

空间惯量的变换规则为

$$\mathbf{I}_B = {}^B\mathbf{X}_A^F \mathbf{I}_A^A \mathbf{X}_B, \quad (3.31)$$

其中 A 和 B 是任意两个坐标系统。在实践中，我们经常需要在只给定 ${}^B\mathbf{X}_A$ 的情况下，从 \mathbf{I}_B 中计算出 \mathbf{I}_A 。这种变换的公式为

$$\mathbf{I}_A = ({}^B\mathbf{X}_A)^T \mathbf{I}_B {}^B\mathbf{X}_A. \quad (3.32)$$

如果两个机体，具有 \mathbf{I}_1 和 \mathbf{I}_2 的惯量，被刚性连接以形成一个单一的复合机体，则该复合机体的惯量 \mathbf{I}_{tot} 是其各部分的惯量总和

$$\mathbf{I}_{\text{tot}} = \mathbf{I}_1 + \mathbf{I}_2. \quad (3.33)$$

这个单一的方程取代了传统的 3-D 向量方法中的三个方程：一个用于计算复合质量，一个用于计算复合质心，还有一个用于计算复合转动惯量。如果一个具有惯量 \mathbf{I} 的刚体以一个 \mathbf{v} 的速度移动，则其动能为

$$T = \frac{1}{2} \mathbf{v} \cdot \mathbf{I} \mathbf{v}. \quad (3.34)$$

如果一个刚体 B 是一个更大系统的一部分，则可以定义 B 的表观惯量矩阵，该矩阵描述了作用在 B 上的力与其产生的加速度之间的关系，同时考虑了系统中的其它机体的影响。该量被称为铰接体惯量。如果 B 恰好是机器人的末端执行器，则它的表观惯量被称为操作空间惯量。

3.2.12 运动方程

空间运动方程表明作用在刚体上的净力等于其动量变化率

$$\mathbf{f} = \frac{d}{dt} (\mathbf{I} \mathbf{v}) = \mathbf{I} \mathbf{a} + \dot{\mathbf{I}} \mathbf{v}.$$

可以证明，表达式 $\dot{\mathbf{I}} \mathbf{v}$ 的计算结果为 $(\mathbf{v} \times \mathbf{I} \mathbf{v})$ [3.8,15]，因此运动方程可被写为

$$\mathbf{f} = \mathbf{I} \mathbf{a} + \mathbf{v} \times \mathbf{I} \mathbf{v}. \quad (3.35)$$

这个单一的方程同时包含刚体的牛顿和欧拉运动方程。为验证这一点，我们可以按以下方式恢

复它们。在机体的质心处表达方程 (3.35)，使用方程 (3.16)、(3.14) 和 (3.22)，我们有

$$\begin{aligned}
 \begin{pmatrix} \underline{n}_C \\ \underline{f} \end{pmatrix} &= \begin{pmatrix} \bar{\mathbf{I}}^{\text{cm}} & \mathbf{0} \\ \mathbf{0} & m\mathbf{1} \end{pmatrix} \begin{pmatrix} \underline{\dot{\omega}} \\ \underline{\dot{v}}_C \end{pmatrix} - \begin{pmatrix} \mathbf{S}(\underline{\omega})^T & \mathbf{S}(\underline{v}_C)^T \\ \mathbf{0} & \mathbf{S}(\underline{\omega})^T \end{pmatrix} \begin{pmatrix} \bar{\mathbf{I}}^{\text{cm}} \underline{\omega} \\ m\underline{v}_C \end{pmatrix} \\
 &= \begin{pmatrix} \bar{\mathbf{I}}^{\text{cm}} & \mathbf{0} \\ \mathbf{0} & m\mathbf{1} \end{pmatrix} \begin{pmatrix} \underline{\dot{\omega}} \\ \underline{\ddot{c}} - \underline{\omega} \times \underline{v}_C \end{pmatrix} + \begin{pmatrix} \underline{\omega} \times \bar{\mathbf{I}}^{\text{cm}} \underline{\omega} \\ m\underline{\omega} \times \underline{v}_C \end{pmatrix} \\
 &= \begin{pmatrix} \bar{\mathbf{I}}^{\text{cm}} \underline{\dot{\omega}} + \underline{\omega} \times \bar{\mathbf{I}}^{\text{cm}} \underline{\omega} \\ m\underline{\ddot{c}} \end{pmatrix}. \tag{3.36}
 \end{aligned}$$

3.2.13 计算机实现

在计算机上实现空间向量运算的最简单方法是从现有的矩阵运算工具 (如 MATLAB) 开始，并编写 (或从 Web 下载) 例程来执行以下操作：

1. 根据方程 (3.14) 从 \mathbf{m} 计算 $\mathbf{S}(\mathbf{m})$ 。
2. 根据方程 (3.9)，从 \mathbf{R} 和 \mathbf{p} 组合 \mathbf{X} 。
3. 根据方程 (3.28)，从 m, \mathbf{c} 和 $\bar{\mathbf{I}}^{\text{cm}}$ 组合 \mathbf{I} 。

所有其它空间算术运算都可以使用标准矩阵算术例程来执行。然而，一些附加例程可被有效地添加到这个列表中，例如：

- 根据其它各种旋转表示计算 \mathbf{R} 的例程。
- 在空间和 4×4 矩阵量之间转换的例程。

当人类生产力比计算效率更重要时，这是推荐的方法。沿着这个思路的软件包可以在文献 [3.16] 中找到。

如果需要更高的效率，则必须使用更复杂的空间算术库，其中

1. 为每一种空间量定义专用数据结构，并且
2. 提供一套计算例程，每个例程通过有效公式实现空间算术运算。

表 3.1 显示了一些合适的数据结构和有效公式的示例。注意，建议用于刚体惯量和 Plücker 变换的数据结构只包含它们所代表的 6×6 矩阵的三分之一的数字。在本表中所列的高效算术公式所提供的成本节约，与使用一般的 6×6 和 6×1 矩阵运算相比，范围从 1.5 倍因子到 6 倍因子。甚至还更高效的公式可以在文献 [3.17] 中找到。

3.2.14 总结

空间向量是结合刚体运动的线性和角度方面的 6-D 向量，从而产生非常适合描述动力学算法的紧凑表示法。为了避免与 3-D 向量产生一些名称冲突，我们使用了粗体竖直字母来标志空间向量，而张量仍然用斜体标志。在接下来的部分中，将使用竖直字母标志空间向量与其它向量串联在一起的向量，如 $\underline{\mathbf{q}}$ 。

表 3.1 总结了在本节中介绍的空间量和算子，以及根据 3-D 量和算子定义它们的公式。对于空间算术的高效计算机中实现，它还给出了数据结构和公式。此表应与表 2.5 和 2.6 一起阅读，它们显示了如何计算各种关节类型的方向、位置和空间速度。注意，当结合表 3.1 一起阅读时，那些表中的 ${}^j\mathbf{R}_i$ 和 ${}^j\mathbf{p}_i$ 分别对应于此表的 ${}^B\mathbf{R}_A^T$ 和 ${}^A\mathbf{p}_B$ 。

表 3.1: 空间向量表示法的汇总

空间的量:	
\mathbf{v}	刚体的速度
\mathbf{a}	刚体的空间加速度 ($\mathbf{a} = \dot{\mathbf{v}}$)
\mathbf{a}'	刚体加速度的经典描述, 做为 6-D 向量表达
\mathbf{f}	作用在刚体上的力
\mathbf{I}	刚体的惯量
\mathbf{X}	对于运动向量的 Plücker 坐标变换
\mathbf{X}^F	对于力向量的 Plücker 坐标变换 ($\mathbf{X}^F = \mathbf{X}^{-T}$)
${}^B\mathbf{X}_A$	从 A 坐标到 B 坐标的 Plücker 变换
\mathbf{m}	通用的运动向量 (\mathbf{M}^6 的任意元素)
3-D 的量:	
O	坐标系原点
\mathbf{r}	在点 O 处的机体固定点相对于任意固定点的位置
$\boldsymbol{\omega}$	刚体的角速度
\mathbf{v}_O	在 O 处机体固定点的线速度 ($\mathbf{v}_O = \dot{\mathbf{r}}$)
$\dot{\boldsymbol{\omega}}$	刚体的角加速度
$\dot{\mathbf{v}}_O$	将 O 固定在空间中的 \mathbf{v}_O 的导数
$\dot{\mathbf{v}}'_O$	将 O 固定在机体中的 \mathbf{v}_O 的导数; 即在 O 处机体固定点的经典加速度 ($\dot{\mathbf{v}}'_O = \ddot{\mathbf{r}}$)
\mathbf{f}	作用在刚体上的线性力, 或线性力系统的合力
\mathbf{n}_O	线性力或线性力系统围绕 O 的力矩
m	刚体质量
\mathbf{c}	刚体质心的位置, 相对于 O 测量
\mathbf{h}	刚体的第一质量矩, $\mathbf{h} = m\mathbf{c}$; 也可以标志线性动量
$\bar{\mathbf{I}}^{\text{cm}}$	围绕机体质心的转动惯量
$\bar{\mathbf{I}}$	围绕 O 的转动惯量
${}^B\mathbf{R}_A$	从 A 坐标到 B 坐标的正交旋转矩阵变换
${}^A\mathbf{p}_B$	B 坐标原点相对于 A 坐标原点的定位, 在 A 坐标中表达

表 3.1: (continued)

方程组			
$\mathbf{v} = \begin{pmatrix} \boldsymbol{\omega} \\ \mathbf{v}_O \end{pmatrix} \quad \mathbf{a} = \begin{pmatrix} \dot{\boldsymbol{\omega}} \\ \dot{\mathbf{v}}_O \end{pmatrix} = \begin{pmatrix} \dot{\boldsymbol{\omega}} \\ \ddot{\mathbf{r}} - \boldsymbol{\omega} \times \dot{\mathbf{r}} \end{pmatrix}$			
$\mathbf{f} = \begin{pmatrix} \mathbf{n}_O \\ \mathbf{f} \end{pmatrix} \quad \mathbf{a}' = \begin{pmatrix} \dot{\boldsymbol{\omega}} \\ \dot{\mathbf{v}}'_O \end{pmatrix} = \begin{pmatrix} \dot{\boldsymbol{\omega}} \\ \ddot{\mathbf{r}} \end{pmatrix} = \mathbf{a} + \begin{pmatrix} \mathbf{0} \\ \boldsymbol{\omega} \times \mathbf{v}_O \end{pmatrix}$			
$\mathbf{I} = \begin{pmatrix} \bar{\mathbf{I}} & \mathbf{S}(\mathbf{h}) \\ \mathbf{S}(\mathbf{h})^T & m\mathbf{1} \end{pmatrix} = \begin{pmatrix} \bar{\mathbf{I}}^{\text{cm}} + m\mathbf{S}(\mathbf{c})\mathbf{S}(\mathbf{c})^T & m\mathbf{S}(\mathbf{c}) \\ m\mathbf{S}(\mathbf{c})^T & m\mathbf{1} \end{pmatrix}$			
${}^B\mathbf{X}_A = \begin{pmatrix} {}^B\mathbf{R}_A & \mathbf{0} \\ {}^B\mathbf{R}_A\mathbf{S}({}^A\mathbf{p}_B)^T & {}^B\mathbf{R}_A \end{pmatrix} = \begin{pmatrix} {}^B\mathbf{R}_A & \mathbf{0} \\ \mathbf{S}({}^B\mathbf{p}_A){}^B\mathbf{R}_A & {}^B\mathbf{R}_A \end{pmatrix}$			
$\mathbf{v} \cdot \mathbf{f} = \mathbf{f} \cdot \mathbf{v} = \mathbf{v}^T \mathbf{f} = \boldsymbol{\omega} \cdot \mathbf{n}_O + \mathbf{v}_O \cdot \mathbf{f}$			
$\mathbf{v} \times \mathbf{m} = \begin{pmatrix} \boldsymbol{\omega} \times \mathbf{m} \\ \mathbf{v}_O \times \mathbf{m} + \boldsymbol{\omega} \times \mathbf{m}_O \end{pmatrix} = \begin{pmatrix} \mathbf{S}(\boldsymbol{\omega}) & \mathbf{0} \\ \mathbf{S}(\mathbf{v}_O) & \mathbf{S}(\boldsymbol{\omega}) \end{pmatrix} \begin{pmatrix} \mathbf{m} \\ \mathbf{m}_O \end{pmatrix}$			
$\mathbf{v} \times \mathbf{f} = \begin{pmatrix} \boldsymbol{\omega} \times \mathbf{n}_O + \mathbf{v}_O \times \mathbf{f} \\ \boldsymbol{\omega} \times \mathbf{f} \end{pmatrix} = \begin{pmatrix} \mathbf{S}(\boldsymbol{\omega}) & \mathbf{S}(\mathbf{v}_O) \\ \mathbf{0} & \mathbf{S}(\boldsymbol{\omega}) \end{pmatrix} \begin{pmatrix} \mathbf{n}_O \\ \mathbf{f} \end{pmatrix}$			
紧凑的计算机表示			
数学对象	大小	计算机表示	大小
$\begin{pmatrix} \boldsymbol{\omega} \\ \mathbf{v}_O \end{pmatrix}$	6×1	$(\boldsymbol{\omega}; \mathbf{v}_O)$	$3 + 3$
$\begin{pmatrix} \mathbf{n}_O \\ \mathbf{f} \end{pmatrix}$	6×1	$(\mathbf{n}_O; \mathbf{f})$	$3 + 3$
$\begin{pmatrix} \bar{\mathbf{I}} & \mathbf{S}(\mathbf{h}) \\ \mathbf{S}(\mathbf{h})^T & m\mathbf{1} \end{pmatrix}$	6×6	$(m; \mathbf{h}; \bar{\mathbf{I}})$	$1 + 3 + 9$
$\begin{pmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{RS}(\mathbf{p})^T & \mathbf{R} \end{pmatrix}$	6×6	$(\mathbf{R}; \mathbf{p})$	$9 + 3$
高效的空间算术公式			
表达式	计算值		
$\mathbf{X}\mathbf{v}$	$(\mathbf{R}\boldsymbol{\omega}; \mathbf{R}(\mathbf{v}_O - \mathbf{p} \times \boldsymbol{\omega}))$		
$\mathbf{X}^F\mathbf{f}$	$(\mathbf{R}(\mathbf{n}_O - \mathbf{p} \times \mathbf{f}); \mathbf{R}\mathbf{f})$		
\mathbf{X}^{-1}	$(\mathbf{R}^T; -\mathbf{R}\mathbf{p})$		
$\mathbf{X}^{-1}\mathbf{v}$	$(\mathbf{R}^T\boldsymbol{\omega}; \mathbf{R}^T\mathbf{v}_O + \mathbf{p} \times \mathbf{R}^T\boldsymbol{\omega})$		
$(\mathbf{X}^F)^{-1}\mathbf{f}$	$(\mathbf{R}^T\mathbf{n}_O + \mathbf{p} \times \mathbf{R}^T\mathbf{f}; \mathbf{R}^T\mathbf{f})$		
$\mathbf{X}_1\mathbf{X}_2$	$(\mathbf{R}_1\mathbf{R}_2; \mathbf{p}_2 + \mathbf{R}_2^T\mathbf{p}_1)$		
$\mathbf{I}_1 + \mathbf{I}_2$	$(m_1 + m_2; \mathbf{h}_1 + \mathbf{h}_2; \bar{\mathbf{I}}_1 + \bar{\mathbf{I}}_2)$		
$\mathbf{I}\mathbf{v}$	$(\bar{\mathbf{I}}\boldsymbol{\omega} + \mathbf{h} \times \mathbf{v}_O; m\mathbf{v}_O - \mathbf{h} \times \boldsymbol{\omega})$		
$\mathbf{X}^T\mathbf{I}\mathbf{X}$	$(m; \mathbf{R}^T\mathbf{h} + m\mathbf{p}; \mathbf{R}^T\bar{\mathbf{I}}\mathbf{R} - \mathbf{S}(\mathbf{p})\mathbf{S}(\mathbf{R}^T\mathbf{h}) - \mathbf{S}(\mathbf{R}^T\mathbf{h} + m\mathbf{p})\mathbf{S}(\mathbf{p}))$		
对于 $\mathbf{X}^T\mathbf{I}\mathbf{X}$ 的含义, 参见方程 (3.32)			

3.3 正则方程组

机器人机构的运动方程通常以两种规范形式之一呈现：关节空间公式，

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \boldsymbol{\tau}_g(\mathbf{q}) = \boldsymbol{\tau}, \quad (3.37)$$

或操作空间公式，

$$\boldsymbol{\Lambda}(\mathbf{x})\dot{\mathbf{v}} + \boldsymbol{\mu}(\mathbf{x}, \mathbf{v}) + \boldsymbol{\rho}(\mathbf{x}) = \mathbf{f}. \quad (3.38)$$

这些公式明确地显示了函数依赖关系： \mathbf{H} 是 \mathbf{q} 的函数， $\boldsymbol{\Lambda}$ 是 \mathbf{x} 的函数，依此类推。一旦理解了这些依赖关系，通常就会省略它们。在方程 (3.38) 中， \mathbf{x} 是操作空间坐标的向量，而 \mathbf{v} 和 \mathbf{f} 则是表示末端执行器的速度和作用在其上的外力的空间向量。如果机器人是冗余的，则该方程的系数必须定义为 \mathbf{q} 与 $\dot{\mathbf{q}}$ 的函数，而不是 \mathbf{x} 与 \mathbf{v} 的函数。

下面将进一步解释这两个方程，跟着方程 (3.37) 的拉格朗日公式的描述，以及对于冲击的运动冲量方程。

3.3.1 关节空间公式

符号 $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}$ 和 $\boldsymbol{\tau}$ 分别标志关节位置、速度、加速度和力变量的 n 维向量，其中 n 是机器人机构的运动自由度。 \mathbf{H} 是一个 $n \times n$ 对称正定矩阵，称为广义的，或称关节空间的惯量矩阵 (JSIM)。 \mathbf{C} 是 $n \times n$ 矩阵，使得 $\mathbf{C}\dot{\mathbf{q}}$ 是科里奥利项和离心项的向量，统称为速度积 (*velocity product*) 项；并且 $\boldsymbol{\tau}_g$ 是重力项的向量。根据需要，可在该方程中添加更多项，以说明其它动力学效应 (例如粘性摩擦)。施加在末端执行器处的机构上的力 \mathbf{f} 的影响可通过将项 $\mathbf{J}^T \mathbf{f}$ 添加到方程 (3.37) 的右侧来说明，其中 \mathbf{J} 是末端执行器的雅可比矩阵 (第 2.8.1 节)。

\mathbf{q} 指定在机构位形空间中的一个点的坐标。如果该机构是运动学树 (第 3.4 节)，则 \mathbf{q} 包含机构中的每个关节变量，否则仅包含独立子集。 \mathbf{q} 的元素是广义坐标。同样， $\dot{\mathbf{q}}, \ddot{\mathbf{q}}$ 和 $\boldsymbol{\tau}$ 的元素是广义速度、加速度和力。

3.3.2 拉格朗日公式

存在各种方法来推导方程 (3.37) 中的项。两个在机器人学中最常用的方法是牛顿-欧拉公式和拉格朗日公式。前者直接适用于刚体的牛顿方程和欧拉方程，它们被包含在空间运动方程 (3.35) 中。该公式特别适用于开发用于动力学计算的高效递归算法，如第 3.5 和 3.6 节所述。

拉格朗日公式通过机器人机构的拉格朗日算符进行，

$$L = T - U, \quad (3.39)$$

其中， T 和 U 分别是机构的总动能和势能。动能给出为

$$T = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{H} \dot{\mathbf{q}}. \quad (3.40)$$

然后，可以使用每个广义坐标的拉格朗日方程建立运动的动力学方程

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \tau_i. \quad (3.41)$$

其结果方程可以在标量形式中写为

$$\sum_{j=1}^n H_{ij} \ddot{q}_j + \sum_{j=1}^n \sum_{k=1}^n C_{ijk} \dot{q}_j \dot{q}_k + \tau_{gi} = \tau_i, \quad (3.42)$$

其显示了速度-积项的结构。 C_{ijk} 被称为第一种类型的 Christoffel 符号, 并被给出为

$$C_{ijk} = \frac{1}{2} \left(\frac{\partial H_{ij}}{\partial q_k} + \frac{\partial H_{ik}}{\partial q_j} - \frac{\partial H_{jk}}{\partial q_i} \right). \quad (3.43)$$

它们只是位置变量 q_i 的函数。在方程 (3.37) 中的 C 元素可被定义为

$$C_{ij} = \sum_{k=1}^n C_{ijk} \dot{q}_k. \quad (3.44)$$

然而, C 不是唯一的, 并且其它定义也是可能的。

选择在方程 (3.44) 中给出的 C , 可以表示矩阵 N , 给出为

$$N(\mathbf{q}, \dot{\mathbf{q}}) = \dot{H}(\mathbf{q}) - 2C(\mathbf{q}, \dot{\mathbf{q}}), \quad (3.45)$$

这是斜对称的矩阵 [3.18]。因此, 对于任意 $n \times 1$ 向量 α ,

$$\alpha^T N(\mathbf{q}, \dot{\mathbf{q}}) \alpha = 0. \quad (3.46)$$

这个性质在控制中非常有用, 特别是当考虑 $\alpha = \dot{\mathbf{q}}$ 时, 其给出

$$\dot{\mathbf{q}}^T N(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} = 0. \quad (3.47)$$

通过应用能量守恒原理, 可以证明方程 (3.47) 适用于矩阵 C 的任意选择 [3.18, 19]。

3.3.3 操作空间公式

在方程 (3.38) 中, \mathbf{x} 是操作空间坐标的 6-D 向量, 给出机器人末端执行器的位置和方向; \mathbf{v} 是末端执行器的速度; \mathbf{f} 是施加在末端执行器上的力。 \mathbf{x} 通常是笛卡尔坐标、欧拉角或四元数分量的列表, 并通过以下形式的微分方程与 \mathbf{v} 相关联

$$\dot{\mathbf{x}} = E(\mathbf{x}) \mathbf{v}. \quad (3.48)$$

Λ 是操作空间惯量矩阵, 它是末端执行器的表观惯量, 考虑了机器人机构其余部分的影响 (即, 它是铰接机体惯量)。 μ 和 ρ 分别是速度-积和重力项的向量。

操作空间 (也称为任务空间) 是发出和执行高级运动和力命令的空间。操作空间公式因此在运动和力控制系统的背景中特别有用 (第 8.2 和 9.2 节)。方程 (3.38) 可以推广到维度不为 6 的操作空间, 以及包含超过一个末端执行器的运动的操作空间 [3.20]。

方程 (3.37) 和 (3.38) 中的项由以下公式相关联

$$\mathbf{v} = J\dot{\mathbf{q}}, \quad (3.49)$$

$$\dot{\mathbf{v}} = J\ddot{\mathbf{q}} + \dot{J}\dot{\mathbf{q}}, \quad (3.50)$$

$$\boldsymbol{\tau} = J^T \mathbf{f}, \quad (3.51)$$

$$\Lambda = (JH^{-1}J^T)^{-1}, \quad (3.52)$$

$$\mu = \Lambda (JH^{-1}C\dot{\mathbf{q}} - \dot{J}\dot{\mathbf{q}}), \quad (3.53)$$

并且

$$\rho = \Lambda JH^{-1}\boldsymbol{\tau}_g. \quad (3.54)$$

这些方程假定 $m \leq n$ (m 是操作空间坐标的维数), 并且雅可比 J 具有满秩。更多详细信息可在文献 [3.21] 中找到。

3.3.4 冲击模型

如果机器人在其环境中撞击一个刚体，则冲击瞬间产生冲击力，导致机器人速度发生阶跃变化。让我们假设冲击发生在末端执行器和在环境中的一个刚体之间，并且 \mathbf{f}' 的空间冲量作用在末端执行器上。该冲量导致末端执行器速度的 $\Delta \mathbf{v}$ 阶跃变化；并且两者由冲量运动的操作空间方程联系起来 [3.22]，

$$\Lambda \Delta \mathbf{v} = \mathbf{f}'. \quad (3.55)$$

在关节空间中，机器人机构的冲量运动方程为

$$\mathbf{H} \Delta \dot{\mathbf{q}} = \boldsymbol{\tau}', \quad (3.56)$$

其中 $\boldsymbol{\tau}'$ 和 $\Delta \dot{\mathbf{q}}$ 分别标志关节空间冲量和速度变化。在机器人末端执行器发生碰撞的情况下，我们有

$$\boldsymbol{\tau}' = \mathbf{J}^T \mathbf{f}' \quad (3.57)$$

和

$$\Delta \mathbf{v} = \mathbf{J} \Delta \dot{\mathbf{q}}, \quad (3.58)$$

其由方程 (3.51) 和 (3.49) 得出。方程组 (3.55)-(3.57) 意味着

$$\Delta \dot{\mathbf{q}} = \bar{\mathbf{J}} \Delta \mathbf{v}, \quad (3.59)$$

其中 $\bar{\mathbf{J}}$ 是 \mathbf{J} 的惯量加权伪逆，并给出为

$$\bar{\mathbf{J}} = \mathbf{H}^{-1} \mathbf{J}^T \Lambda. \quad (3.60)$$

$\bar{\mathbf{J}}$ 也被称为雅各布矩阵的动力学一致广义逆 [3.21]。注意，该表达式 $\Lambda \mathbf{J} \mathbf{H}^{-1}$ ，其出现在方程 (3.53) 和 (3.54) 中，等于 $\bar{\mathbf{J}}^T$ ，因为 \mathbf{H} 和 Λ 都是对称的。尽管我们在冲量动力学的背景下引入 $\bar{\mathbf{J}}$ ，但它更多地被用于正常（即非冲量）的动力学方程。

3.4 刚体系统动力学模型

机器人机构的基本刚体模型由四部分组成：连接图、连杆和关节几何参数、连杆惯量参数和一组关节模型。在此模型中，可以添加各种产生力的元素，例如弹簧、阻尼器、关节摩擦、执行器和驱动器。特别是执行器和驱动器，其本身可能具有非常复杂的动力学模型。也可以增加额外的运动自由度以建模在关节轴承或连杆中的弹力（第 11 章）。本节介绍基本模型。关于这个主题的更多内容可以在诸如文献 [3.3,8,23] 等书中找到。

3.4.1 连接性

连接图是一个无向图，其中每个节点代表一个刚体，每个圆弧代表一个关节。图形必须连接；只有一个节点代表一个固定的基座或参考帧。如果图形表示移动机器人（即未连接到固定基座的机器人），则有必要在固定基座和移动机器人中的任意一个机体之间引入虚拟的 6-DOF 关节。选定的机体被称为浮动基座 (*floating base*)。如果单个图形表示移动机器人的集合，那么每个机器人都有自己的浮动基座，每个浮动基座都有自己的 6-DOF 关节。注意，6-DOF 关节不会对其连接的两个实体施加任何约束，因此引入 6-DOF 关节会更改图形的连接性，而不会更改它所表示的系统的物理性质。

在图论项中, 一个自身循环 (loop) 是连接节点和自身的圆弧, 而一个回环 (cycle) 是一条闭合路径, 它不会多次穿过任何圆弧。在机器人机构的连接图中, 不允许自身循环, 回环称为运动学循环。包含运动学循环的机构称为闭环机构; 而不这样的机构称为开环机构或运动学树。每个闭环机构都有一个生成树, 它定义了一个开环机构, 不在生成树中的每个关节称为闭环关节。树中的关节称为树关节。

固定基座作为运动学树的根节点, 以及闭环机构上任何生成树的根节点。若至少一个节点至少有两个子节点, 则称运动学树为分支树, 否则称为不分支树。不分支的运动学树也称为运动学链, 分支树可以称为分支运动学链。一个典型的工业机器人手臂, 不带夹具, 是一个运动学链, 而一个典型的模拟机器人是一个带有浮动基座的运动学树。

在包含 N_B 个移动机体和 N_J 个关节的系统中, 其中 N_J 包括上述 6-DOF 关节, 机体和关节的编号如下所示。首先, 固定底座编号为机体 0。然后, 以任意顺序对其它机体从 1 到 N_B 进行编号, 以使每个机体的编号高于其父机体。如果系统包含运动学循环, 则必须首先选择生成树, 并执行该选择, 因为机体父对象的标识由生成树确定。这种类型的编号称为规则编号方案 (*regular numbering scheme*)。

对机体编号后, 我们将树关节编号为 1 到 N_B , 以便关节 i 将机体 i 连接到其父对象。闭环关节 (如有) 按任意顺序从 $N_B + 1$ 到 N_J 编号。每个闭环关节 k 闭合一个独立的运动学循环, 并且我们对循环进行编号, 从 1 到 N_L (其中 $N_L = N_J - N_B$ 是独立循环的数量), 使得循环 l 是由关节 $k = N_B + l$ 闭合的循环。运动学循环 l 是图中唯一的循环, 它穿过关节 k , 但不穿过任何其它循环闭合关节。

对于未分支的运动学树, 这些规则生成一个唯一的编号方案, 其中机体从底部到顶部连续编号, 关节编号使关节 i 连接机体 i 和 $i - 1$ 。在所有其它情况下, 规则数字不是唯一的。

虽然连接图是无向的, 但为了定义关节速度和力, 有必要为每个关节指定方向。这对于树关节和闭环关节都是必要的。具体地说, 关节是指从一个机体连接到另一个机体。我们可以分别称它们为关节 i 的前导 (predecessor) 关节 $p(i)$ 和后继 (successor) 关节 $s(i)$ 。然后将关节速度定义为后继关节相对于前导关节的速度; 并且关节力是指作用于后继关节的力。所有树关节从父节点连接到子节点是标准做法 (但不是必需的)。

运动学树或闭环机构上生成树的连接性由父机体编号的 N_B 个元素数组描述, 其中第 i 个元素 $p(i)$ 是机体 i 的父元素。注意, 机体 i 的父元素 $p(i)$ 也是关节 i 的前导关节 $p(i)$, 并因此是通用表示法。许多算法依赖于属性 $p(i) < i$ 以正确的顺序执行计算。机体 i 的子体的所有机体数的集合, $c(i)$, 在许多递归算法中也是有用的。

运动学循环的连接性数据可以用多种方式描述。便于在递归算法中使用的表示包括以下约定。闭环关节 k 连接机体 $p(k)$ (前导关节) 和 $s(k)$ (后继关节)。当机体 i 是根结点时, 机体 i 的集合 $LR(i)$ 给出了循环数。使用生成树中机体的性质 $p(i) < i$, 循环的根被选择为具有最小数目的机体。另外, 当机体 i 不是根结点时, 机体 i 的集合 $LB(i)$ 给出了其所属的循环数。

在图 3.3 中给出了一个闭环系统的示例。该系统由一个拓扑变化的拟人机器人机构组成, 它与环境 and 机构内的接触形成闭环。该系统具有 $N_B = 16$ 个移动机体和 $N_J = 19$ 个关节, 带有 $N_L = N_J - N_B = 3$ 个循环。主要机体 (1) 被认为是该移动机器人系统的浮动基座。与固定底座 (0) 通过虚拟的 6-DOF 关节 (1) 连接。为完成该示例, 在表 3.2 中给出了闭环关节、机体编号 $p(k)$ 和 $s(k)$ 以及每个循环的根节点机体。基于机体的集合 $c(i)$ 和 $LB(i)$ 在表 3.3 中给出。注意, 对于本例, $LR(0) = \{1, 3\}$ 和 $LR(1) = \{2\}$, 并且所有其它 LR 集合都为空。

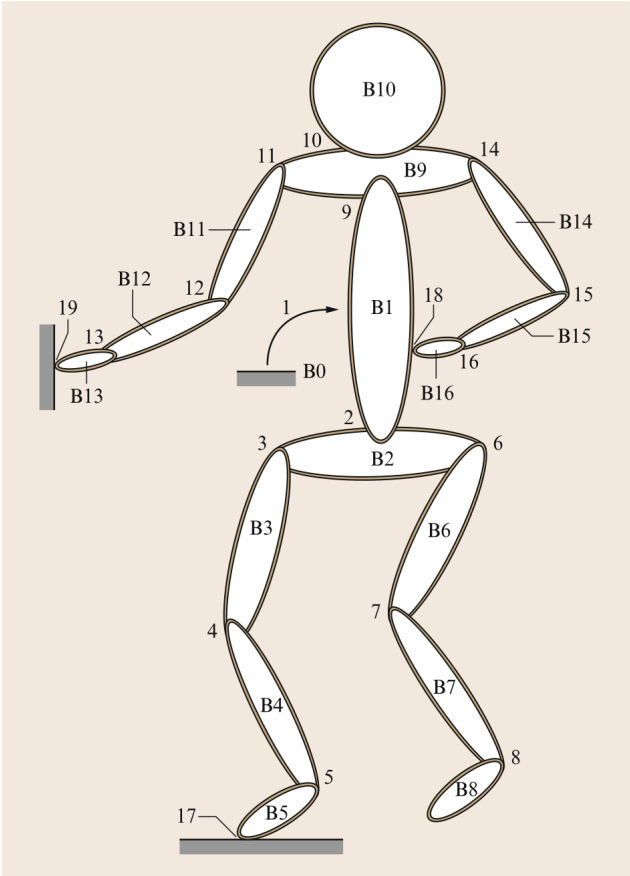


图 3.3: 拟人机器人示例。注意：为了区分图中的机体编号和关节编号，机体编号前面加上一个 **B**

表 3.2: 拟人机器人示例的闭环关节和循环的根节点

循环 l	闭环关节 k	$p(k)$	$s(k)$	根节点
1	17	0	5	0
2	18	16	1	1
3	19	0	13	0

表 3.3: 拟人机器人示例中基于机体的集合

机体 c	$c(i)$	$LB(i)$	机体 i	$c(i)$	$LB(i)$
0	1		9	10, 11, 14	2, 3
1	2, 9	1, 3	10		
2	3, 6	1	11	12	3
3	4	1	12	13	3
4	5	1	13		3
5		1	14	15	2
6	7		15	16	2
7	8		16		2
8					

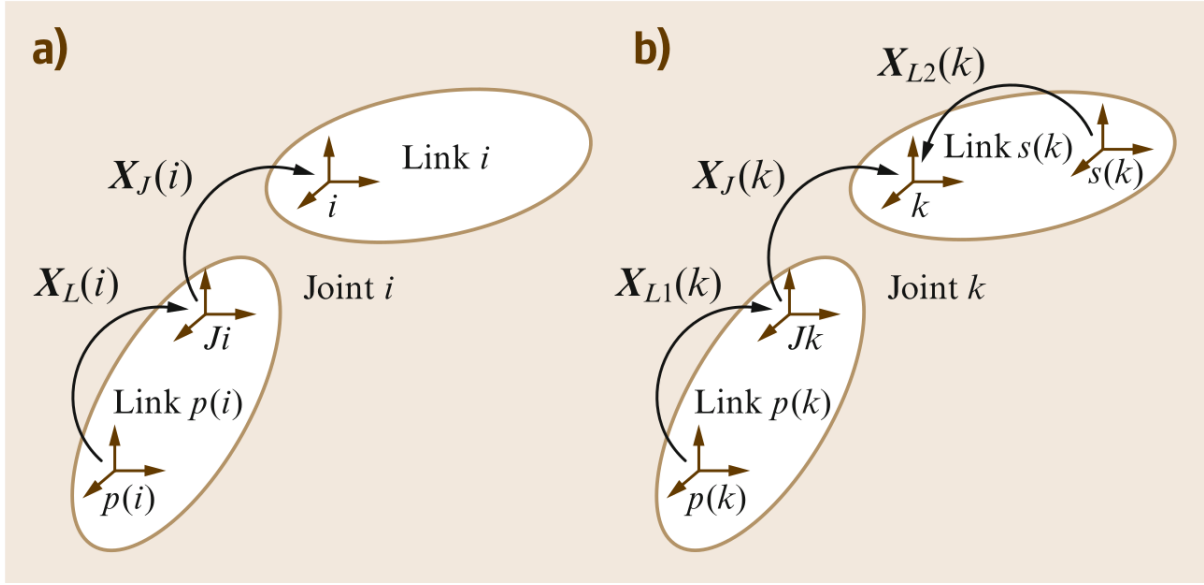


图 3.4: 与坐标帧和坐标变换相关联的 (a) 一个树状关节, 以及 (b) 一个环路闭合关节

3.4.2 连杆几何结构

当两个机体通过关节连接时, 连接的完整描述包括关节本身的描述以及两个坐标帧的位置, 每个坐标帧都附着在两个机体中, 这两个坐标帧指定关节在每个机体中的位置。如果系统中有 N_J 个关节, 则总共有 $2N_J$ 个关节附着帧。其中一半帧用数字 1 至 N_J 标识, 其余帧用标签 $J1$ 至 JN_J 标识。每个关节 i 从帧 Ji 连接到帧 i 。

对于关节 1 至 N_B (即树形关节), 帧 i 刚性连接至机体 i 。对于关节 $N_B + 1$ 至 N_J , 用于闭环关节 k 的帧 k 将刚性连接至机体 $s(k)$ 。对于每个关节 i , 第二个坐标帧 Ji 附着到前导关节 $p(i)$, 无论它是树关节还是闭环关节。坐标帧 Ji 为关节 i 提供了一个基本帧, 因为关节旋转和/或平移是相对于该帧定义的。

图 3.4 显示了与系统中每个关节关联的坐标帧和变换。树关节从帧 $p(i)$ 坐标到帧 i 坐标的整体变换给出为

$${}^i\mathbf{X}_{p(i)} = {}^i\mathbf{X}_{Ji} {}^{Ji}\mathbf{X}_{p(i)} = \mathbf{X}_J(i) \mathbf{X}_L(i). \quad (3.61)$$

变换 $\mathbf{X}_L(i)$ 是固定连杆变换, 它设置关节 i 相对于 $p(i)$ 的基本帧 Ji 。它可用于将空间运动向量从 $p(i)$ 变换为 Ji 坐标。变换 $\mathbf{X}_J(i)$ 是可变关节变换, 完成关节 i 从 Ji 坐标到 i 坐标的变换。

类似地, 闭环关节从帧 $p(k)$ 坐标到帧 k 坐标的整体变换给出为

$${}^k\mathbf{X}_{p(k)} = {}^k\mathbf{X}_{Jk} {}^{Jk}\mathbf{X}_{p(k)} = \mathbf{X}_J(k) \mathbf{X}_{L1}(k). \quad (3.62)$$

一个附加变换 $\mathbf{X}_{L2}(k)$ 被定义为从帧 $s(k)$ 坐标到帧 k 坐标的变换, 给出为

$$\mathbf{X}_{L2}(k) = {}^k\mathbf{X}_{s(k)}. \quad (3.63)$$

可以以多种不同的方式指定连杆和关节几何结构数据。最常用的方法是使用 Denavit-Hartenberg 参数 [3.24]。然而, 标准的 Denavit-Hartenberg 参数不完全通用, 不足以描述分支运动树的几何结构, 也不足以描述包含某些多自由度关节类型的机构。Denavit-Hartenberg 参数 [3.2] 的改进形式用于本手册 (第 2.4 节) 中的单自由度关节。这些参数已扩展到分支运动树 [3.23] 和闭环机构。

3.4.3 连杆惯量

连杆惯量数据包括机构中每个连杆的质量、质心位置和转动惯量。连杆 i 的惯量参数以坐标帧 i 表示，因此为常量。

3.4.4 关节模型

使用 *Roberson* 和 *Schwertassek* [3.3] 的通用关节模型描述连接连杆之间的关系。对于一个运动学树或在一个闭环机构上的生成树，一个 $n_i \times 1$ 向量， $\dot{\mathbf{q}}_i$ ，将连杆 i 的速度与其父连杆 $p(i)$ 的速度相关联，其中 n_i 是连接两个连杆的关节处的自由度数。对于闭环机构中的闭环关节，其速度关系为后继连杆 $s(i)$ 的速度与前导连杆 $p(i)$ 的速度之间的关系。在任何一种情况下，都是坐标帧 i 和 Ji 的速度之间的关系。

设 \mathbf{v}_{rel} 和 \mathbf{a}_{rel} 标志穿过关节 i 的速度和加速度，即连杆 $s(i)$ 相对于 $p(i)$ 的速度和加速度。关节的自由模式由 $6 \times n_i$ 矩阵 Φ_i 表示，因此 \mathbf{v}_{rel} 和 \mathbf{a}_{rel} 给出为

$$\mathbf{v}_{\text{rel}} = \Phi_i \dot{\mathbf{q}}_i \quad (3.64)$$

并且

$$\mathbf{a}_{\text{rel}} = \Phi_i \ddot{\mathbf{q}}_i + \dot{\Phi}_i \dot{\mathbf{q}}_i, \quad (3.65)$$

其中 Φ_i 和 $\dot{\Phi}_i$ 取决于关节类型 [3.3]。矩阵 Φ_i 具有满列秩，因此我们可以定义互补矩阵 Φ_i^c ，使得 6×6 矩阵 $(\Phi_i \Phi_i^c)$ 是可逆的。我们可以将该矩阵的列视为在 M^6 上形成的基，使得第一个 n_i 基向量定义允许的运动方向，而剩余的 $6 - n_i = n_i^c$ 向量定义不允许的运动方向。因此， Φ_i^c 表示关节 i 的约束模式。

通过关节 i 从其前导传递到其继任的力 \mathbf{f}_i 给出为

$$\mathbf{f}_i = (\Psi_i \Psi_i^c) \begin{pmatrix} \boldsymbol{\tau}_i \\ \boldsymbol{\lambda}_i \end{pmatrix}, \quad (3.66)$$

其中 $\boldsymbol{\tau}_i$ 是沿自由模式施加的力的 $n_i \times 1$ 向量， $\boldsymbol{\lambda}_i$ 是约束力的 $(6 - n_i) \times 1$ 向量，并且 Ψ_i 和 Ψ_i^c 被计算为

$$(\Psi_i \Psi_i^c) = (\Phi_i \Phi_i^c)^{-T}. \quad (3.67)$$

对于最常见的关节类型，可以选择 Φ_i 和 Φ_i^c ，以使得矩阵 $(\Phi_i \Phi_i^c)$ 为数值形式正交，因此 $(\Psi_i \Psi_i^c)$ 在数值上与 $(\Phi_i \Phi_i^c)$ 相等。注意方程 (3.67) 意味着以下关系： $(\Psi_i)^T \Phi_i = \mathbf{1}_{n_i \times n_i}$ ， $(\Psi_i)^T \Phi_i^c = \mathbf{0}_{n_i \times (6 - n_i)}$ ， $(\Psi_i^c)^T \Phi_i = \mathbf{0}_{(6 - n_i) \times n_i}$ ，并且 $(\Psi_i^c)^T \Phi_i^c = \mathbf{1}_{(6 - n_i) \times (6 - n_i)}$ 。当应用于方程 (3.66) 时，将产生以下有用的关系

$$\boldsymbol{\tau}_i = \Phi_i^T \mathbf{f}_i. \quad (3.68)$$

在方程 (3.65) 中 $\dot{\Phi}_i$ 的值取决于关节类型。该通用公式为

$$\dot{\Phi}_i = \overset{\circ}{\Phi}_i + \mathbf{v}_i \times \Phi_i, \quad (3.69)$$

其中 \mathbf{v}_i 是连杆 i 的速度， $\overset{\circ}{\Phi}_i$ 是 Φ_i 的表观导数，如与连杆 i 一起移动的观察所见，并给出为

$$\overset{\circ}{\Phi}_i = \frac{\partial \Phi_i}{\partial \mathbf{q}_i} \dot{\mathbf{q}}_i. \quad (3.70)$$

对于最常见的关节类型， $\overset{\circ}{\Phi}_i = \mathbf{0}$ 。

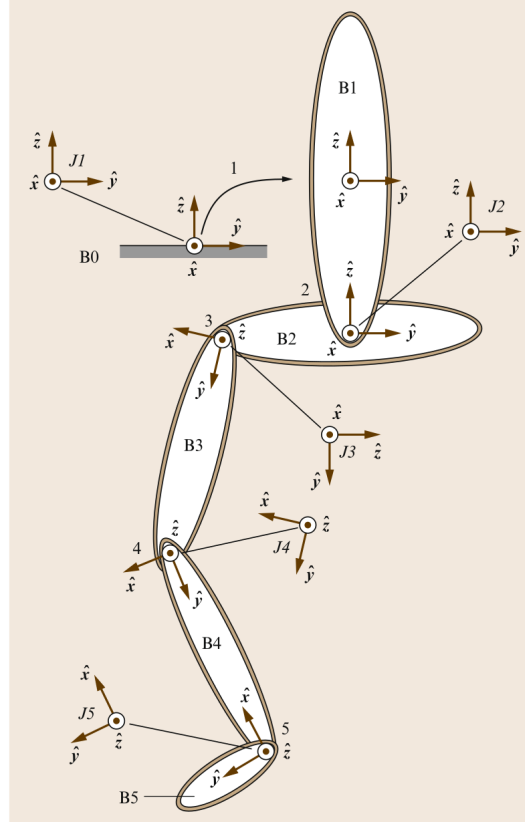


图 3.5: 拟人机器人示例的前五个连杆和关节的坐标帧

当使用 Denavit-Hartenberg 约定时, 使用单自由度关节 ($n_i = 1$) 特别简单。沿 (棱形关节) 或绕 (旋转关节) 的 \hat{z}_i 坐标轴选择运动。在这种情况下, 对于棱形关节, $\Phi_i = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}^T$, 对于旋转关节, $\Phi_i = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}^T$ 。另外, $\dot{\Phi}_i = \mathbf{0}$ 。

对于移动机器人浮动基座的虚拟 6-DOF 关节也相对容易处理。对于这种情况, $\Phi_i = \mathbf{1}$ (6×6 单位矩阵) 并且 $\dot{\Phi}_i = \mathbf{0}$ 。

在下一节的示例中说明旋转关节和浮动基座关节, 以及万向节 ($n_i = 2$) 和球形关节 ($n_i = 3$)。关于关节运动学的其它详细信息, 参见第 2.3 节。

3.4.5 示例系统

为了说明用于连杆和关节模型的约定, 坐标帧连接到拟人机器人的前五个连杆 (机体) 和固定基座, 如图 3.5 所示。注意, 对于前五个关节中的每个关节, 帧 J_i 附着到连杆 $p(i) = i - 1$ 。在本例中, 帧 J_1 的原点设置为与帧 0 的原点重合, 帧 J_2, J_3, J_4 和 J_5 的原点分别设置为与帧 2, 3, 4 和 5 的原点重合。

注意, J_1 可以设置在固定基座 (B_0) 上的任何位置/方向, 以允许浮动基座 (B_1) 相对于固定基座的运动的最方便表示。另外, J_2 的原点可以沿 \hat{z}_2 任意设置。

表 3.4 给出了示例系统每个关节的自由度数量, 以及基本帧 J_i 的固定旋转和位置。旋转 ${}^{J_i}R_{p(i)}$ 将 $p(i)$ 坐标中的 3-D 向量变换为 J_i 坐标。位置 ${}^{p(i)}p_{J_i}$ 是给出原点 O_{J_i} 相对于 $O_{p(i)}$ 的位置的向量, 在 $p(i)$ 坐标中表达。空间变换 $\mathbf{X}_L(i) = {}^{J_i}\mathbf{X}_{p(i)}$ 可以通过在表 3.1 中的 ${}^B\mathbf{X}_A$ 公式由这些 3-D

表 3.4: 对于示例系统的关节 i , 从帧 $p(i)$ 到基本帧 Ji 的自由度数量 (n_i)、固定旋转 (${}^{Ji}\mathbf{R}_{p(i)}$) 和位置 (${}^{(pi)}\mathbf{p}_{Ji}$)。注意, $2l_i$ 是连杆 i 沿其长轴的标称长度

关节	(n_i)	$({}^{Ji}\mathbf{R}_{p(i)})$	$({}^{(pi)}\mathbf{p}_{Ji})$
1	6	$\mathbf{1}_{3 \times 3}$	$\mathbf{0}_{3 \times 3}$
2	1	$\mathbf{1}_{3 \times 3}$	$\begin{pmatrix} 0 \\ 0 \\ -l_1 \end{pmatrix}$
3	3	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ -l_2 \\ 0 \end{pmatrix}$
4	1	$\mathbf{1}_{3 \times 3}$	$\begin{pmatrix} 0 \\ 2l_3 \\ 0 \end{pmatrix}$
5	2	$\begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 2l_4 \\ 0 \end{pmatrix}$

量组成。拟人机器人有一个浮动的基座和躯干, 一个在躯干和骨盆之间的旋转关节 (围绕 $\hat{\mathbf{z}}_2$ 轴旋转), 臀部的球形关节, 膝盖的旋转关节, 脚踝的万向关节。如图 3.5 所示, 腿稍微弯曲, 脚向外侧弯曲 (臀部围绕 $\hat{\mathbf{y}}_3$ 旋转 $\approx 90^\circ$)。

拟人机器人中所有关节类型的自由模式、速度变量和位置变量如表 2.5 和 2.6 所示。对于在这些表中的 ${}^j\mathbf{R}_i$ 和 ${}^j\mathbf{p}_i$ 的表达式分别给出了 ${}^i\mathbf{R}_{Ji}^T$ 和 ${}^{Ji}\mathbf{p}_i$, 通过它们可以组成关节变换 $\mathbf{X}_J(i) = {}^i\mathbf{X}_{Ji}$ 。旋转关节遵循 Denavit-Hartenberg 约定, 围绕 $\hat{\mathbf{z}}_i$ 轴旋转。踝关节围绕 $\hat{\mathbf{z}}_{J5}$ 轴有一个俯仰旋转为 α_5 , 然后围绕 $\hat{\mathbf{y}}_5$ 轴有一个横滚旋转为 β_5 (参见表 2.1 中的 $Z-Y-X$ 欧拉角的定义)。髋关节被建模为球窝球形关节。为了避免与欧拉角相关的奇异性, 可以使用四元数 ϵ_i 来表示臀部的方向。四元数速率 $\dot{\epsilon}_i$ 与相对旋转率 $\omega_{i\text{rel}}$ 之间的关系在本手册的方程 (2.8) 中给出。

浮动基座分别使用躯干的位置 ${}^0\mathbf{p}_1$ 和四元数 ϵ_1 作为其位置和方向状态变量。躯干的位置可以通过对连杆的速度进行积分来计算, 在固定的基坐标中表达: ${}^0\mathbf{v}_1 = {}^0\mathbf{R}_1\mathbf{v}_1$, 其中 \mathbf{v}_1 是躯干在移动坐标中的速度。

注意, 对于除万向节之外的所有关节, $\dot{\Phi}_i = 0$ 。由于连杆 5 坐标中 $\hat{\mathbf{z}}_{J5}$ 的分量随 β_5 而变化, 因此 $\dot{\hat{\mathbf{z}}}_{J5} \neq 0$ 。参见手册的第 2.3 节, 以了解关节运动学的更多详细信息。

3.5 运动学树

运动学树的动力学比闭环机构的动力学更简单、更容易计算。实际上, 许多用于闭环机构的算法首先计算生成树的动力学, 然后使其服从闭环约束。

本节介绍运动学树的以下动力学算法: 递归牛顿-欧拉算法 (RNEA) 用于逆向动力学, 铰接机体算法 (ABA) 用于正向动力学, 复合刚体机体算法 (CRBA) 用于计算关节空间惯量矩阵 (JSIM), 并给出两种算法计算操作空间惯量矩阵 (OSIM)。前三种方法的实现可以在文献 [3.16] 中找到。

3.5.1 递归牛顿-欧拉算法

这是一种 $O(n)$ 算法，用于计算固定基座运动树的逆向动力学，它基于 *Luh* 等人 [3.4] 非常高效的 RNEA 实现。一个浮动基座版本可在文献 [3.8,15] 中找到。给定关节位置和速度变量，该算法计算产生给定关节加速度集合所需的应用关节扭矩/力变量。

首先通过从固定基座到树的叶子节点的连接的一个向外递归，计算连接速度和加速度。在此递归过程中，使用牛顿-欧拉方程 (3.35) 计算每个连杆上所需的力。第二，向内递归使用每个连杆处的力平衡方程来计算每个关节上的空间力以及每个关节扭矩/力变量的值。计算效率的关键步骤是将大部分的量引用到局部连接坐标。同时，通过加速机构底部向上运动，将重力对各连杆的影响有效地包含在方程中。

计算分为四个步骤进行，如下所示，两个递归中的每一个都有两个步骤。

步骤 1

依次计算每个连杆的速度和加速度，从固定基座的已知速度和加速度开始，朝尖端 (即连接图中的叶子节点) 移动。

运动树中每个连杆的速度由递推公式给出为

$$\mathbf{v}_i = \mathbf{v}_{p(i)} + \Phi_i \dot{\mathbf{q}}_i, \quad (\mathbf{v}_0 = \mathbf{0}), \quad (3.71)$$

其中， \mathbf{v}_i 是连杆 i 的速度，对于关节 i ， Φ_i 是运动矩阵，并且 $\dot{\mathbf{q}}_i$ 是关节 i 的关节速度变量向量。

通过对方程 (3.71) 进行微分，获得加速度的等效公式，给出为

$$\mathbf{a}_i = \mathbf{a}_{p(i)} + \Phi_i \ddot{\mathbf{q}}_i + \dot{\Phi}_i \dot{\mathbf{q}}_i, \quad (\mathbf{a}_0 = \mathbf{0}), \quad (3.72)$$

其中， \mathbf{a}_i 是连杆 i 的加速度， $\ddot{\mathbf{q}}_i$ 是关节加速度变量的向量。

通过将 \mathbf{a}_0 初始化为 $-\mathbf{a}_g$ (而不是零) 来模拟均匀重力场对机构的影响，其中 \mathbf{a}_g 是重力加速度向量。在这种情况下， \mathbf{a}_i 不是连杆 i 的真实加速度，而是它的真实加速度与 $-\mathbf{a}_g$ 之和。

步骤 2

计算每个连杆的运动方程。此步骤计算产生在步骤 1 中为计算加速度所需的力。连杆 i 的运动方程为

$$\mathbf{f}_i^a = \mathbf{I}_i \mathbf{a}_i + \mathbf{v}_i \times \mathbf{I}_i \mathbf{v}_i, \quad (3.73)$$

其中， \mathbf{I}_i 是连杆 i 的空间惯量，并且 \mathbf{f}_i^a 是作用在连杆 i 上的净力。

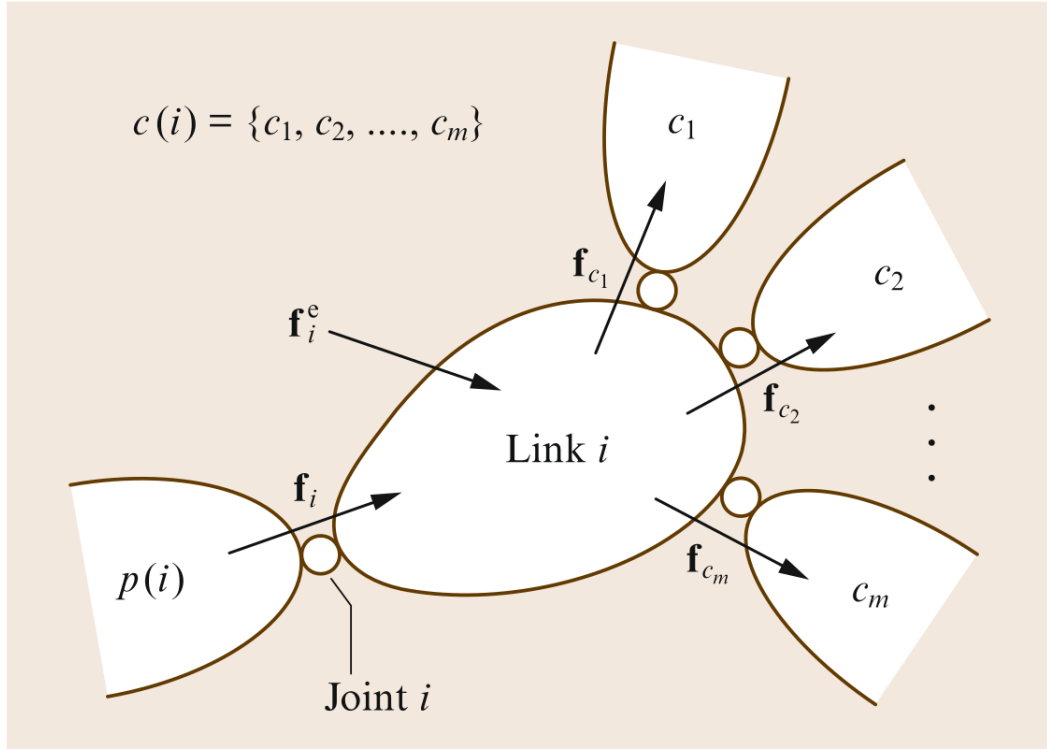
步骤 3

计算穿过每个关节的空间力。参考图 3.6，作用在连杆 i 上的净力为

$$\mathbf{f}_i^a = \mathbf{f}_i^e + \mathbf{f}_i - \sum_{j \in c(i)} \mathbf{f}_j,$$

其中， \mathbf{f}_i 是通过关节 i 传递的力， \mathbf{f}_i^e 是作用在关节 i 上的所有相关外力的总和， $c(i)$ 是关节 i 的子关节集合。为计算关节力，重新排列该方程给出以下的递归公式

$$\mathbf{f}_i = \mathbf{f}_i^a - \mathbf{f}_i^e + \sum_{j \in c(i)} \mathbf{f}_j, \quad (3.74)$$

图 3.6: 作用于连杆 i 上的力

其中, i 从 N_B 迭代到 1。

\mathbf{f}_i^e 可能包括来自弹簧、阻尼器、力场、与环境接触等的贡献, 但其值假定为已知值, 或至少可根据已知量计算。

如果重力没有被虚拟的基座加速度模拟, 则作用在连杆 i 上的重力必须包含在 \mathbf{f}_i^e 中。

步骤 4

计算关节力变量 τ_i 。根据定义, 它们由以下方程给出为

$$\tau_i = \Phi_i^T \mathbf{f}_i. \quad (3.75)$$

坐标自由算法

方程 (3.71)-(3.75) 表明在算法 3.1 中所示的算法, 它是 RNEA 的坐标自由版本。这是最简单的算法形式, 适用于数学分析和相关用途。然而, 它不适用于数值计算, 因为数值版本必须使用坐标向量。

连杆坐标算法

通常, 如果为每个连杆定义了一个坐标系, 则我们说算法是在连杆坐标中实现的, 并且与连杆 i 相关的计算是在与连杆 i 相关联的坐标系中执行的。替代方法是在绝对坐标中实现算法, 在这种情况下, 所有计算都是在单一坐标系中执行的, 通常是基本连杆的坐标系。在实际应用中, 当在连杆坐标中实现时, RNEA 计算效率更高, 并且其它大多数动力学算法也是如此。

要将 RNEA 转换为连杆坐标, 我们首先检查方程组, 查看哪些方程包含来自超过一个连杆的量。方程 (3.73) 和 (3.75) 各自仅涉及与连杆 i 有关的量, 因此无需修改。这些方程被称为连杆 i

Algorithm 3.1 对于逆向动力学，坐标自由的递归牛顿-欧拉算法 (RNEA)

```

 $\mathbf{v}_0 = \mathbf{0}$ 
 $\mathbf{a}_0 = -\mathbf{a}_g$ 
for  $i = 1$  to  $N_B$  do
   $\mathbf{v}_i = \mathbf{v}_{p(i)} + \Phi_i \dot{\mathbf{q}}_i$ 
   $\mathbf{a}_i = \mathbf{a}_{p(i)} + \Phi_i \ddot{\mathbf{q}}_i + \dot{\Phi}_i \dot{\mathbf{q}}_i$ 
   $\mathbf{f}_i = \mathbf{I}_i \mathbf{a}_i + \mathbf{v}_i \times \mathbf{I}_i \mathbf{v}_i - \mathbf{f}_i^e$ 
end for
for  $i = N_B$  to 1 do
   $\boldsymbol{\tau}_i = \Phi_i^T \mathbf{f}_i$ 
  if  $p \neq 0$  then
     $\mathbf{f}_{p(i)} = \mathbf{f}_{p(i)} + \mathbf{f}_i$ 
  end if
end for

```

的局部方程。其余方程均涉及来自超过一个连杆的量，因此需要插入坐标变换矩阵。方程 (3.71)、(3.72) 和 (3.74) 的修改版本为

$$\mathbf{v}_i = {}^i \mathbf{X}_{p(i)} \mathbf{v}_{p(i)} + \Phi_i \dot{\mathbf{q}}_i, \quad (3.76)$$

$$\mathbf{a}_i = {}^i \mathbf{X}_{p(i)} \mathbf{a}_{p(i)} + \Phi_i \ddot{\mathbf{q}}_i + \dot{\Phi}_i \dot{\mathbf{q}}_i, \quad (3.77)$$

和

$$\mathbf{f}_i = \mathbf{f}_i^e - {}^i \mathbf{X}_0^F {}^0 \mathbf{f}_i^e + \sum_{j \in c(i)} {}^i \mathbf{X}_j^F \mathbf{f}_j \quad (3.78)$$

方程 (3.78) 假定外力在绝对坐标 (即连杆 0) 中表达。

完整的算法在算法 3.2 中显示。函数 `jtype` 返回关节 i 的类型代码；函数 `xjcalc` 计算指定类型关节的关节变换矩阵；函数 `pcalc` 和 `pdcalc` 计算 Φ_i 和 $\dot{\Phi}_i$ 。这些函数用于各种关节类型的公式可在表 2.5 和 2.6 中找到，并记住必须使用的旋转矩阵是这些表中列出的转置矩阵。一般情况下，需要 `pcalc` 和 `pdcalc`。然而，对于最常见的关节类型， Φ_i 是连杆坐标中的已知常数，并因此 $\dot{\Phi}_i$ 为零。如果事先知道所有关节都具有此特性，则可以相应地简化算法。量 \mathbf{I}_i 和 $\mathbf{X}_L(i)$ 是连杆坐标中的已知常量，并且是描述机器人机构的数据结构的一部分。

在第一层循环中的最后一个赋值将每个 \mathbf{f}_i 初始化为表达式 $\mathbf{f}_i^e - {}^i \mathbf{X}_0^F {}^0 \mathbf{f}_i^e$ (使用特征 ${}^i \mathbf{X}_0^F = {}^i \mathbf{X}_0^{-T}$)。然后在第二层循环中执行方程 (3.78) 右侧的求和。该算法包含计算 ${}^i \mathbf{X}_0$ 的代码，用于将外力变换为连杆坐标。如果没有外力，则可以省略此代码。如果只有一个外力 (例如，机器人手臂末端执行器处的力)，则这段代码可以用这样的代码代替，即使用 ${}^i \mathbf{X}_{p(i)}$ 将外力向量从一个连杆坐标系统依次变换到下一个连杆坐标系统。

注意：虽然短语连杆坐标 (*link coordinates*) 表明我们使用的是移动坐标帧，但该算法实际上是在静止坐标中实现的，而它恰好与当前瞬间的移动坐标重合。

3-D 向量 RNEA

RNEA 的原始版本的开发和表达使用 3-D 向量 (例如文献 [3.2,4])。算法 3.3 给出了该算法的一个特例，其中假定关节是旋转的，并且假定关节轴与连杆坐标帧的 z 轴重合。(如果没有这些假设，方程会更长。) 它还假设外力为零。

Algorithm 3.2 使用空间向量的递归牛顿-欧拉算法

input: $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, model, {}^0\mathbf{f}_i$
output: $\boldsymbol{\tau}$
model data : $N_B, \text{jtype}, \mathbf{X}_L, \mathbf{I}_i$
 $\mathbf{v}_0 = \mathbf{0}$
 $\mathbf{a}_0 = -\mathbf{a}_g$
for $i = 1$ **to** N_B **do**
 $\mathbf{X}_J(i) = \text{xjcalc}(\text{jtype}(i), \mathbf{q}_i)$
 ${}^i\mathbf{X}_{p(i)} = \mathbf{X}_J(i) \mathbf{X}_L(i)$
if $p \neq 0$ **then**
 ${}^i\mathbf{X}_0 = {}^i\mathbf{X}_{p(i)} {}^{p(i)}\mathbf{X}_0$
end if
 $\Phi_i = \text{pcalc}(\text{jtype}(i), \mathbf{q}_i)$
 $\overset{\circ}{\Phi}_i = \text{pdcalc}(\text{jtype}(i), \mathbf{q}_i, \dot{\mathbf{q}}_i)$
 $\mathbf{v}_i = {}^i\mathbf{X}_{p(i)} \mathbf{v}_{p(i)} + \Phi_i \dot{\mathbf{q}}_i$
 $\zeta_i = \overset{\circ}{\Phi}_i \dot{\mathbf{q}}_i + \mathbf{v}_i \times \Phi_i \dot{\mathbf{q}}_i$
 $\mathbf{a}_i = {}^i\mathbf{X}_{p(i)} \mathbf{a}_{p(i)} + \Phi_i \ddot{\mathbf{q}}_i + \zeta_i$
 $\mathbf{f}_i = \mathbf{I}_i \mathbf{a}_i + \mathbf{v}_i \times \mathbf{I}_i \mathbf{v}_i - {}^i\mathbf{X}_0^{-T} {}^0\mathbf{f}_i$
end for
for $i = N_B$ **to** 1 **do**
 $\boldsymbol{\tau}_i = \Phi_i^T \mathbf{f}_i$
if $p \neq 0$ **then**
 $\mathbf{f}_{p(i)} = \mathbf{f}_{p(i)} + {}^i\mathbf{X}_{p(i)}^T \mathbf{f}_i$
end if
end for

Algorithm 3.3 仅适用于旋转关节的 3-D 向量递归牛顿-欧拉算法

input: $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, model$

output: $\boldsymbol{\tau}$

model data : $N_B, {}^p\mathbf{R}_L \mathbf{p}_i, m_i, \mathbf{c}_i, \bar{\mathbf{I}}_i^{\text{cm}}$

$$\boldsymbol{\omega}_0 = \mathbf{0}$$

$$\dot{\boldsymbol{\omega}}_0 = \mathbf{0}$$

$$\dot{\mathbf{v}}'_0 = -\dot{\mathbf{v}}'_g$$

for $i = 1$ **to** N_B **do**

$${}^i\mathbf{R}_{p(i)} = \text{rotz}(\mathbf{q}_i) \mathbf{R}_L(i)$$

$$\boldsymbol{\omega}_i = {}^i\mathbf{R}_{p(i)} \boldsymbol{\omega}_{p(i)} + \hat{\mathbf{z}}_i \dot{\mathbf{q}}_i$$

$$\dot{\boldsymbol{\omega}}_i = {}^i\mathbf{R}_{p(i)} \dot{\boldsymbol{\omega}}_{p(i)} + ({}^i\mathbf{R}_{p(i)} \boldsymbol{\omega}_{p(i)}) \times \hat{\mathbf{z}}_i \dot{\mathbf{q}}_i + \hat{\mathbf{z}}_i \ddot{\mathbf{q}}_i$$

$$\begin{aligned} \dot{\mathbf{v}}'_i = & {}^i\mathbf{R}_{p(i)} (\dot{\mathbf{v}}'_{p(i)} + \dot{\boldsymbol{\omega}}_{p(i)} \times {}^{p(i)}\mathbf{p}_i \\ & + \boldsymbol{\omega}_{p(i)} \times \boldsymbol{\omega}_{p(i)} \times {}^{p(i)}\mathbf{p}_i) \end{aligned}$$

$$\mathbf{f}_i = m_i (\dot{\mathbf{v}}'_i + \dot{\boldsymbol{\omega}}_i \times \mathbf{c}_i + \boldsymbol{\omega}_i \times \boldsymbol{\omega}_i \times \mathbf{c}_i)$$

$$\mathbf{n}_i = \bar{\mathbf{I}}_i^{\text{cm}} \dot{\boldsymbol{\omega}}_i + \boldsymbol{\omega}_i \times \bar{\mathbf{I}}_i^{\text{cm}} \boldsymbol{\omega}_i + \mathbf{c}_i \times \mathbf{f}_i$$

end for

for $i = N_B$ **to** 1 **do**

$$\tau_i = \hat{\mathbf{z}}_i^T \mathbf{n}_i$$

if $p \neq 0$ **then**

$$\mathbf{f}_{p(i)} = \mathbf{f}_{p(i)} + {}^i\mathbf{R}_{p(i)}^T \mathbf{f}_i$$

$$\mathbf{n}_{p(i)} = \mathbf{n}_{p(i)} + {}^i\mathbf{R}_{p(i)}^T \mathbf{n}_i + {}^{p(i)}\mathbf{p}_i \times {}^i\mathbf{R}_{p(i)}^T \mathbf{f}_i$$

end if

end for

在该算法中, $\dot{\mathbf{v}}'_g$ 是重力引起的线性加速度, 在基座 (连杆 0) 坐标中表达; rotz 计算在方程 (2.2) 中所示矩阵的转置; $\mathbf{R}_L(i)$ 是 $\mathbf{X}_L(i)$ 的旋转分量; ${}^i\mathbf{R}_{p(i)}$ 是 ${}^i\mathbf{X}_{p(i)}$ 的旋转分量; 不使用 pcalc 和 pdcalc , 因为 Φ_i 是已知的常量 $\begin{pmatrix} \hat{\mathbf{z}}^T & \mathbf{0}^T \end{pmatrix}^T$; $\dot{\mathbf{v}}'_i$ 是连杆 i 坐标原点 (O_i) 的线性加速度, 是连杆 i 的经典加速度的线性分量; ${}^{p(i)}\mathbf{p}_i$ 是 O_i 相对于 $O_{p(i)}$ 的位置, 在 $p(i)$ 坐标中表达; 并且 m_i, \mathbf{c}_i 和 $\bar{\mathbf{I}}_i^{\text{cm}}$ 是连杆 i 的惯量参数。(关于将这些 3-D 量与相应空间量相关联的方程组, 参见表 3.1。)

乍一看, 3-D 向量算法与空间向量算法明显不同。然而, 它可以直接从空间向量算法中获得, 只是简单将空间向量展开为它们的 3-D 分量, 将关节类型限制为旋转, 将空间加速度转换为经典加速度 (即, 根据方程 (3.22), 将 $\dot{\mathbf{v}}_i$ 的每个实例替换为 $\dot{\mathbf{v}}'_i - \boldsymbol{\omega}_i \times \mathbf{v}_i$, 并应用一些 3-D 向量特征式, 将方程转换为表中所示的形式。从空间加速度到经典加速度的转换有一个有趣的副作用: \mathbf{v}_i 从运动方程中消除, 因此不需要计算。因此, 该算法的 3-D 版本比空间版本具有略微的速度优势。

3.5.2 铰接体算法

ABA 是一种 $O(N_B)$ 算法, 用于计算运动学树的正向动力学。然而, 在正常情况下, $O(N_B) = O(n)$, 因此我们将其称为 $O(n)$ 算法。ABA 由 Featherstone [3.1] 开发, 并且是约束传播算法的一个示例。给定关节位置、速度和施加的扭矩/力变量, 该算法计算关节加速度。在确定关节加速度的情况下, 可以使用数值积分来提供机构的模拟。

ABA 中的关键概念如图 3.7 所示。以连杆 i 为根的子树仅通过关节 i 传递的力 \mathbf{f}_i 与运动学树的其余部分交互。假设我们在这一点断开树, 并且只考虑子树受未知力 \mathbf{f}_i 作用于连杆 i 的运动。这可能表明, 连杆 i 的加速度与所施加的力有关, 其方程为

$$\mathbf{f}_i = \mathbf{I}_i^A \mathbf{a}_i + \mathbf{p}_i^A, \quad (3.79)$$

其中 \mathbf{I}_i^A 称为在子树中连杆 i 的铰接体惯量 (我们现在可以称之为铰接体), 并且 \mathbf{p}_i^A 是相关的偏置力 (bias force), 这是在连杆 i 中产生零加速度所需的力。注意, \mathbf{p}_i^A 取决于铰接体中各个物体的速度。方程 (3.79) 考虑了子树的完整动力学。因此, 如果我们碰巧知道 \mathbf{f}_i 的正确值, 则方程 (3.79) 将立即给出连杆 i 的正确加速度。

我们之所以对 \mathbf{I}_i^A 和 \mathbf{p}_i^A 这两个量感兴趣, 是因为它们允许我们从 $\mathbf{a}_{p(i)}$ 计算 $\ddot{\mathbf{q}}_i$, 这反过来允许我们计算 \mathbf{a}_i , 然后允许我们计算出更多关节加速, 如此等等。将方程 (3.79) 与方程 (3.75) 和 (3.72) 结合起来, 给出

$$\boldsymbol{\tau}_i = \Phi_i^T \mathbf{f}_i = \Phi_i^T \left(\mathbf{I}_i^A \left(\mathbf{a}_{p(i)} + \Phi_i \ddot{\mathbf{q}}_i + \dot{\Phi}_i \dot{\mathbf{q}}_i \right) + \mathbf{p}_i^A \right),$$

其可求解 $\ddot{\mathbf{q}}_i$, 给出为

$$\ddot{\mathbf{q}}_i = \mathbf{D}_i (\mathbf{u}_i - \mathbf{U}_i^T \mathbf{a}_{p(i)}), \quad (3.80)$$

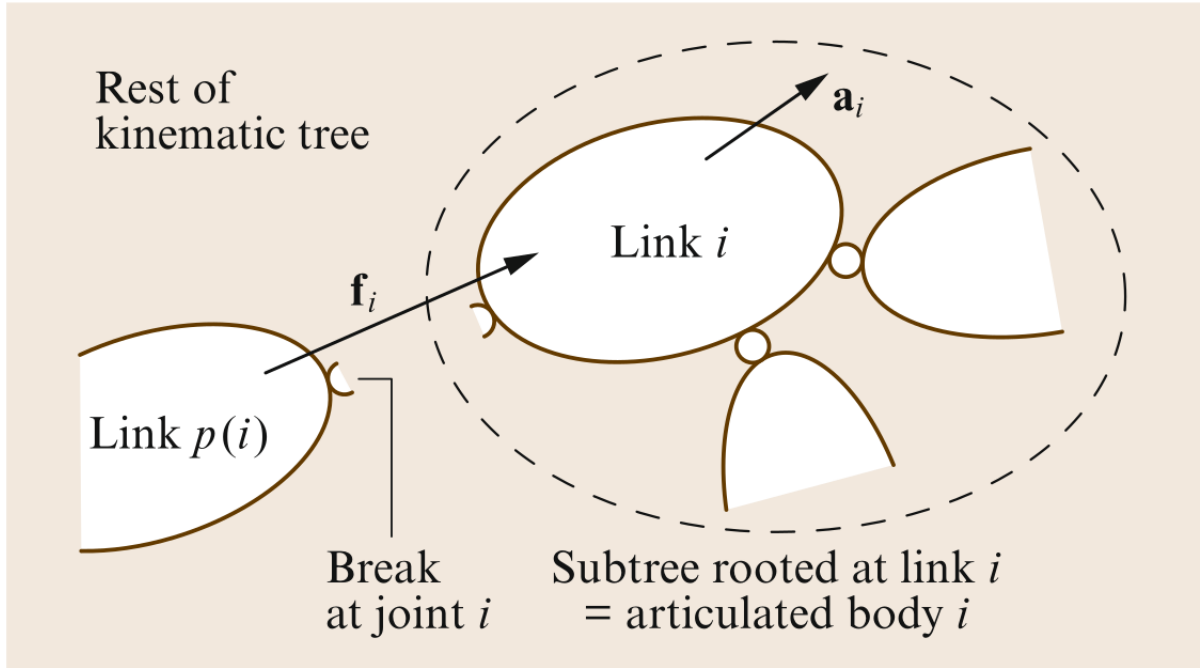
其中

$$\begin{aligned} \mathbf{U}_i &= \mathbf{I}_i^A \Phi_i, \\ \mathbf{D}_i &= (\Phi_i^T \mathbf{U}_i)^{-1} = (\Phi_i^T \mathbf{I}_i^A \Phi_i)^{-1}, \\ \mathbf{u}_i &= \boldsymbol{\tau}_i - \mathbf{U}_i^T \zeta_i - \Phi_i^T \mathbf{p}_i^A \end{aligned}$$

并且

$$\zeta_i = \dot{\Phi}_i \dot{\mathbf{q}}_i = \overset{\circ}{\Phi}_i \dot{\mathbf{q}}_i + \mathbf{v}_i \times \Phi_i \dot{\mathbf{q}}_i.$$

则 \mathbf{a}_i 可通过方程 (3.72) 计算出来。

图 3.7: 一个铰接体 i 的定义

结果表明，铰接体的惯量和偏置力通过递归公式可以有效地计算出来

$$\mathbf{I}_i^A = \mathbf{I}_i + \sum_{j \in c(i)} (\mathbf{I}_j^A - \mathbf{U}_j \mathbf{D}_j \mathbf{U}_j^T) \quad (3.81)$$

并且

$$\mathbf{p}_i^A = \mathbf{p}_i + \sum_{j \in c(i)} (\mathbf{p}_j^A + \mathbf{I}_j^A \boldsymbol{\zeta}_j + \mathbf{U}_j \mathbf{D}_j \mathbf{u}_j), \quad (3.82)$$

其中

$$\mathbf{p}_i = \mathbf{v}_i \times \mathbf{I}_i \mathbf{v}_i - \mathbf{f}_i^e.$$

这些公式是通过调查在图 3.7 中 \mathbf{f}_i 和 \mathbf{a}_i 之间的关系，并假设对于每一个 $j \in c(i)$ ， \mathbf{I}_j^A 和 \mathbf{p}_j^A 都已经知道的，而获得的。关于更多的细节，参见文献 [3.1,8,15,25]。

完整的算法如算法 3.4 所示。

根据在算法 3.2 中的 RNEA，这是在连杆坐标中表达。它总共在运动学树上进行了三遍迭代。第一遍迭代从基座到顶端迭代；它使用方程 (3.76)，速度乘积项 $\boldsymbol{\zeta}_i = \dot{\Phi}_i \dot{\mathbf{q}}_i$ ，并且它将变量 \mathbf{I}_i^A 和 \mathbf{p}_i^A 分别初始化为 \mathbf{I}_i 和 $\mathbf{p}_i (= \mathbf{v}_i \times \mathbf{I}_i \mathbf{v}_i - {}^i \mathbf{X}_0^F {}^0 \mathbf{f}_i^e)$ ，以计算连杆速度。第二遍迭代从顶端重复到基座；它使用方程 (3.81) 和 (3.82) 计算每个连杆的铰接体惯量和偏置力。第三遍迭代从基座到顶端；使用方程 (3.80) 和 (3.77) 计算连杆和关节的加速度。

3.5.3 复合刚体算法

CRBA 是一种计算运动树关节空间惯量矩阵 (JSIM) 的算法。CRBA 最常见的用途是作为正向动力学算法的一部分。它在文献 [3.5] 中首次作为方法 3 出现。

Algorithm 3.4 对于正向动力学的铰接机体算法

input: $\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\tau}$, model, ${}^0\mathbf{f}_i^e$
output: $\ddot{\mathbf{q}}$
model data : $N_B, \text{jtype} \mathbf{X}_L \mathbf{I}_i$
 $\mathbf{v}_0 = \mathbf{0}$
 $\mathbf{a}_0 = -\mathbf{a}_g$
for $i = 1$ **to** N_B **do**
 $\mathbf{X}_J(i) = \text{xjcalc}(\text{jtype}(i), \mathbf{q}_i)$
 ${}^i\mathbf{X}_{p(i)} = \mathbf{X}_J(i) \mathbf{X}_L(i)$
if $p \neq 0$ **then**
 ${}^i\mathbf{X}_0 = {}^i\mathbf{X}_{p(i)} {}^{p(i)}\mathbf{X}_0$
end if
 $\Phi_i = \text{pcalc}(\text{jtype}(i), \mathbf{q}_i)$
 $\overset{\circ}{\Phi}_i = \text{pdcalc}(\text{jtype}(i), \mathbf{q}_i, \dot{\mathbf{q}}_i)$
 $\mathbf{v}_i = {}^i\mathbf{X}_{p(i)} \mathbf{v}_{p(i)} + \Phi_i \dot{\mathbf{q}}_i$
 $\zeta_i = \overset{\circ}{\Phi}_i \dot{\mathbf{q}}_i + \mathbf{v}_i \times \Phi_i \dot{\mathbf{q}}_i$
 $\mathbf{I}_i^A = \mathbf{I}_i$
 $\mathbf{p}_i^A = \mathbf{v}_i \times \mathbf{I}_i \mathbf{v}_i - {}^i\mathbf{X}_0^{-T} {}^0\mathbf{f}_i^e$
end for
for $i = N_B$ **to** 1 **do**
 $\mathbf{U}_i = \mathbf{I}_i^A \Phi_i$
 $\mathbf{D}_i = (\Phi_i^T \mathbf{U}_i)$
 $\mathbf{u}_i = \boldsymbol{\tau}_i - \mathbf{U}_i^T \zeta_i - \Phi_i^T \mathbf{p}_i^A$
if $p \neq 0$ **then**
 $\mathbf{I}_{p(i)}^A = \mathbf{I}_{p(i)}^A + {}^i\mathbf{X}_{p(i)}^T (\mathbf{I}_i^A - \mathbf{U}_i \mathbf{D}_i \mathbf{U}_i^T) {}^i\mathbf{X}_{p(i)}$
 $\mathbf{p}_{p(i)}^A = \mathbf{p}_{p(i)}^A + {}^i\mathbf{X}_{p(i)}^T (\mathbf{p}_i^A + \mathbf{I}_i^A \zeta_i + \mathbf{U}_i \mathbf{D}_i \mathbf{u}_i)$
end if
end for
for $i = 1$ **to** N_B **do**
 $\mathbf{a}_i = {}^i\mathbf{X}_{p(i)} \mathbf{a}_{p(i)}$
 $\ddot{\mathbf{q}}_i = \mathbf{D}_i (\mathbf{u}_i - \mathbf{U}_i^T \mathbf{a}_i)$
 $\mathbf{a}_i = \mathbf{a}_i + \Phi_i \ddot{\mathbf{q}}_i + \zeta_i$
end for

在关节空间中，正向动力学的任务是从 $\mathbf{q}, \dot{\mathbf{q}}$ 和 $\boldsymbol{\tau}$ 中计算 $\ddot{\mathbf{q}}$ 。从方程 (3.37) 开始，最明显的方法是计算 \mathbf{H} 和 $\mathbf{C}\dot{\mathbf{q}} + \boldsymbol{\tau}_g$ ，然后对于 $\ddot{\mathbf{q}}$ 求解线性方程

$$\mathbf{H}\ddot{\mathbf{q}} = \boldsymbol{\tau} - (\mathbf{C}\dot{\mathbf{q}} + \boldsymbol{\tau}_g) \quad (3.83)$$

如果机构是运动树，则 \mathbf{H} 和 $\mathbf{C}\dot{\mathbf{q}} + \boldsymbol{\tau}_g$ 可分别在 $O(n^2)$ 和 $O(n)$ 运算中计算，且方程 (3.83) 可在 $O(n^3)$ 运算中求解。因此，采用这种方法的算法统称为 $O(n^3)$ 算法。然而， $O(n^3)$ 的这一数字应被视为最坏情况下的复杂性，因为实际复杂性取决于树中的分支数量 [3.26]。此外，即使在最坏情况下， n^3 项也具有较小的系数，并且直到大约 $n = 60$ 时才占主导地位。

$\mathbf{C}\dot{\mathbf{q}} + \boldsymbol{\tau}_g$ 可使用逆向动力学算法计算。如果 $ID(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ 是具有参数 $\mathbf{q}, \dot{\mathbf{q}}$ 和 $\ddot{\mathbf{q}}$ 的逆向动力学计算的结果，则

$$ID(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) = \boldsymbol{\tau} = \mathbf{H}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \boldsymbol{\tau}_g,$$

所以

$$\mathbf{C}\dot{\mathbf{q}} + \boldsymbol{\tau}_g = ID(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{0}). \quad (3.84)$$

因此，可以使用具有 $\ddot{\mathbf{q}} = \mathbf{0}$ 的 RNEA 有效地计算运动学树的 $\mathbf{C}\dot{\mathbf{q}} + \boldsymbol{\tau}_g$ 的值。

在 CRBA 中的关键概念是要注意，JSIM 仅取决于关节位置，而不是其速率。CRBA 进行简化假设，即每个关节处的速率为零。通过假设重力为零， $\mathbf{C}\dot{\mathbf{q}} + \boldsymbol{\tau}_g$ 从方程 (3.83) 中消除。此外，对于旋转关节，应用于第 j 个关节的一个关节加速度单位将产生 JSIM 的第 j 列。该方法将机构划分为两个由第 j 个关节连接的复合刚体，并大大简化了动力学过程。这一概念已被推广，因此 CRBA 可应用于运动学树结构中的任意关节类型。

可以证明，对于运动学树，JSIM 的一般形式为

$$\mathbf{H}_{ij} = \begin{cases} \boldsymbol{\Phi}_i^T \mathbf{I}_i^C \boldsymbol{\Phi}_j & \text{if } i \in c^*(j) \\ \boldsymbol{\Phi}_i^T \mathbf{I}_j^C \boldsymbol{\Phi}_j & \text{if } j \in c^*(i) \\ \mathbf{0} & \text{otherwise.} \end{cases} \quad (3.85)$$

其中 $c^*(i)$ 是以连杆 i 为根的子树中的连杆集合，包括连杆 i 本身，以及

$$\mathbf{I}_i^C = \sum_{j \in c^*(i)} \mathbf{I}_j. \quad (3.86)$$

参见文献 [3.8,15]。事实上， \mathbf{I}_i^C 是由在 $c^*(i)$ 中所有的连杆的刚性装配形成的复合刚体的惯量，这就是算法的名称。

方程 (3.85) 和 (3.86) 是在算法 3.5 中所示算法的基础，即在连杆坐标中的 CRBA。

该算法假定矩阵 ${}^i\mathbf{X}_{p(i)}$ 和 $\boldsymbol{\Phi}_i$ 已经被计算出来，例如在计算 $\mathbf{C}\dot{\mathbf{q}} + \boldsymbol{\tau}_g$ 期间计算。如果不是这样，则可以将算法 3.2 中的相关行插入到第一层循环中。矩阵 \mathbf{F} 是局部变量。第一步， $\mathbf{H} = \mathbf{0}$ ，如果树中没有分支，则可以省略。

在计算了 $\mathbf{C}\dot{\mathbf{q}} + \boldsymbol{\tau}_g$ 和 \mathbf{H} 之后，最后一步是对于 $\ddot{\mathbf{q}}$ 求解方程 (3.83)。这可以使用标准的 Cholesky 或 \mathbf{LDL}^T 因子分解来完成。注意， \mathbf{H} 可能是高度病态的 [3.27]，这反映了运动学树本身的潜在病态，因此建议在正向动力学计算中的每一步都使用双精度算法。(该建议也适用于 ABA 算法。)

利用稀疏性

方程 (3.85) 意味着，如果运动学树中有分支， \mathbf{H} 的某些元素将自动为零。这种效果的例子如图 3.8 所示。这可观察到几乎一半的元素为零。使用在文献 [3.26] 中描述的因子分解算法可以利用这种稀疏性。根据树中的分支数量，稀疏算法的运行速度比标准算法快很多倍。

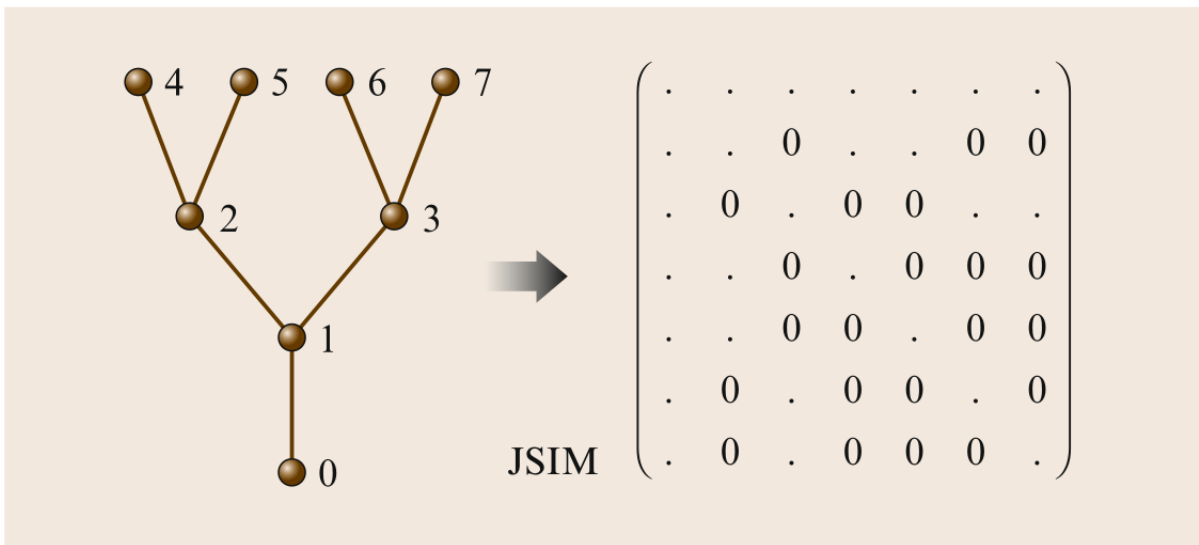
Algorithm 3.5 用于计算 JSIM 的复合刚体算法**input:** $model, RNEA$ partial results**output:** H $model$ data : N_B, pI_i $RNEA$ data : $\Phi_i, {}^iX_{p(i)}$ $H = 0$ **for** $i = 1$ **to** N_B **do** $I_i^C = I_i$ **end for****for** $i = N_B$ **to** 1 **do** $F = I_i^C \Phi_i$ $H_{ii} = \Phi_i^T F$ **if** $p \neq 0$ **then** $I_{p(i)}^C = I_{p(i)}^C + {}^iX_{p(i)}^T I_i^C {}^iX_{p(i)}$ **end if** $j = i$ **while** $p(j) \neq 0$ **do** $F = {}^jX_{p(j)}^T F$ $j = p(j)$ $H_{ij} = F^T \Phi_j$ $H_{ji} = H_{ij}^T$ **end while****end for**

图 3.8: 分支导致的稀疏性: 运动学树中的分支导致在 JSIM 中的某些元素为零

3.5.4 操作空间惯量矩阵

有两种不同的算法被提出来计算 OSIM。第一种是 $O(n^3)$ 算法，它使用的是 OSIM 的基本定义以及 JSIM 的有效因式分解。第二种是基于正向动力学问题有效解的 $O(n)$ 算法。

使用基本定义的算法

如果机器人具有相对较少的自由度 (例如, 6 个), 则计算 OSIM 的最有效方法是通过方程 (3.52)。过程如下:

1. 通过 CRBA 计算 \mathbf{H} 。
2. 将 \mathbf{H} 分解为 $\mathbf{H} = \mathbf{L}\mathbf{L}^T$ (Cholesky 因式分解)。
3. 使用回代以计算 $\mathbf{Y} = \mathbf{L}^{-1}\mathbf{J}^T$ 。
4. $\mathbf{\Lambda}^{-1} = \mathbf{Y}^T\mathbf{Y}$ 。
5. 分解 $\mathbf{\Lambda}^{-1}$ (可选)。

最后一步仅在末端执行器具有完整的 6 个自由度时才可能, 并且仅在应用程序需要 $\mathbf{\Lambda}$ 而不是 $\mathbf{\Lambda}^{-1}$ 时才有必要。在第二步中, 可以使用 \mathbf{LDL}^T 分解代替 Cholesky 因式分解, 或者可以使用在文献 [3.26] 中描述的用于分支运动学树的有效分解之一。

在方程 (3.38) 中的其它项可通过方程 (3.53) 和 (3.54) 计算。特别地, 方程 (3.38) 在形式上可以重写为

$$\dot{\mathbf{v}} + \mathbf{\Lambda}^{-1}(\mathbf{x})[\boldsymbol{\mu}(\mathbf{x}, \mathbf{v}) + \boldsymbol{\rho}(\mathbf{x})] = \mathbf{\Lambda}^{-1}(\mathbf{x})\mathbf{f}, \quad (3.87)$$

并且量 $\mathbf{\Lambda}^{-1}(\boldsymbol{\mu} + \boldsymbol{\rho})$ 可根据公式计算

$$\mathbf{\Lambda}^{-1}(\boldsymbol{\mu} + \boldsymbol{\rho}) = \mathbf{J}\mathbf{H}^{-1}(\mathbf{C}\dot{\mathbf{q}} + \boldsymbol{\tau}_g) - \dot{\mathbf{J}}\dot{\mathbf{q}}. \quad (3.88)$$

该项 $\dot{\mathbf{J}}\dot{\mathbf{q}}$ 是末端执行器方程 (3.50) 的速度-积加速度项。它是通过 RNEA 方程 (3.84) 计算 $\mathbf{C}\dot{\mathbf{q}} + \boldsymbol{\tau}_g$ 的副产品。具体地说, $\dot{\mathbf{J}}\dot{\mathbf{q}} = \mathbf{a}_{ee} - \mathbf{a}_0$, 其中 \mathbf{a}_{ee} 是末端执行器 (end-effector) 的计算加速度 (在与 $\dot{\mathbf{v}}$ 相同的坐标中表达), 并且 \mathbf{a}_0 是基座的加速度 ($-\mathbf{a}_g$)。

$O(n)$ 算法

对于足够大的 n 值, 使用 $O(n)$ 算法变得更有效。在文献 [3.28-30] 中可以找到几种这样的算法。在本节中, 给出了一种更简单的算法, 该算法基于关节空间正向动力学问题的 $O(n)$ 计算, 例如通过 ABA 算法。它是单位力方法 [3.29] 的一种变体, 并计算 OSIM 的逆矩阵。

从方程 (3.87) 开始, 注意 $\mathbf{\Lambda}^{-1}$ 只是位置的函数, 且可以忽略动力学方程中的某些项, 而不影响其值。具体地说, 如果关节速率 $\dot{\mathbf{q}}$ 、关节力 $\boldsymbol{\tau}$ 和重力都设置为零, $\mathbf{\Lambda}$ 的值将保持不变。在这些条件下,

$$\dot{\mathbf{v}} = \mathbf{\Lambda}^{-1}\mathbf{f}. \quad (3.89)$$

让我们将 $\hat{\mathbf{e}}_i$ 定义为一个 6-D 坐标向量, 在第 i 个坐标中为 1, 在其它位置为零。如果我们在方程 (3.89) 中设置 $\mathbf{f} = \hat{\mathbf{e}}_i$, 则 $\dot{\mathbf{v}}$ 将等于 $\mathbf{\Lambda}^{-1}$ 的列 i 。让我们也定义函数 $FD(i, j, \mathbf{q}, \dot{\mathbf{q}}, \mathbf{a}_0, \boldsymbol{\tau}, \mathbf{f})$, 它执行正向动力学计算并返回连杆 i 的真实加速度 (即 $\mathbf{a}_i - \mathbf{a}_0$), 在与 \mathbf{f} 相同的坐标 (通常是基座坐标) 中表达。变元 $\mathbf{q}, \dot{\mathbf{q}}$ 和 $\boldsymbol{\tau}$ 设置关节位置、速度和力变量的值, 而 j 和 \mathbf{f} 指定将 \mathbf{f} 的外力应用于连杆 j 。变元 \mathbf{a}_0 指定包含重力效应的虚拟基座加速度, 并设置为 $\mathbf{0}$ 或 $-\mathbf{a}_g$ 。

根据这些定义，我们有

$$(\Lambda^{-1})^i = FD(ee, ee, \mathbf{q}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \hat{\mathbf{e}}_i) \quad (3.90)$$

和

$$\Lambda^{-1}(\boldsymbol{\mu} + \boldsymbol{\rho}) = -FD(ee, ee, \mathbf{q}, \dot{\mathbf{q}}, -\mathbf{a}_g, \boldsymbol{\tau}, \mathbf{0}), \quad (3.91)$$

其中 $(\Lambda^{-1})^i$ 是 Λ^{-1} 的第 i 列，并且 ee 是末端执行器的机体编号。因此，可以使用在算法 3.6 中所示的算法来计算方程 (3.87) 的系数。该算法是 $O(n)$ 复杂度。

Algorithm 3.6 计算操作空间惯量矩阵和其它项的逆矩阵的算法

for $j = 1$ to 6 do

$$\dot{\mathbf{v}}^j = FD(ee, ee, \mathbf{q}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \hat{\mathbf{e}}_j)$$

end for

$$\Lambda^{-1} = \begin{bmatrix} \dot{\mathbf{v}}^1 & \dot{\mathbf{v}}^2 & \dots & \dot{\mathbf{v}}^6 \end{bmatrix}$$

$$\Lambda^{-1}(\boldsymbol{\mu} + \boldsymbol{\rho}) = -FD(ee, ee, \mathbf{q}, \dot{\mathbf{q}}, -\mathbf{a}_g, \boldsymbol{\tau}, \mathbf{0})$$

在计算 Λ^{-1} 时，注意以下几点可以显著提高算法的效率：(1) 在 ABA 算法中的计算 (算法 3.4)， $\mathbf{v}_i, \boldsymbol{\zeta}_i$ 和 $\boldsymbol{\tau}_i$ 可设置为零，并且 (2) \mathbf{I}_i^A 和依赖于它的量 (\mathbf{U}_i 和 \mathbf{D}_i) 只需计算一次，因为它们不会随所施加的力而变化。另外，请注意，该算法可通过修改 FD 来应用于多个末端执行器，以在其第一个参数中接受末端执行器机体编号列表，并返回包含所有指定机体的加速度的合成向量。然后，算法被封装在一个 for 循环中，该循环控制 FD 的第二个参数，并迭代所有末端执行器机体编号 [3.20]。

然而，对于具有多个末端执行器的分支机构的情况，一些已发表的算法实现了更好的效率，因此应该使用这些算法来替代 [3.20, 29, 31-33]。可实现的最佳复杂度为 $O(n + md + m^2)$ ，其中 m 是末端执行器的数量， d 是系统连接树的深度 [3.33]；但是对于典型的拟人机器人，最快的算法利用了分支诱导的稀疏性 [3.32, 33]。

3.6 运动学环路

在前面各节中的所有算法都是针对运动学树的。本节提供了闭环系统正向动力学的最终算法。该算法用闭环约束方程补充了闭环系统生成树的运动方程。本节给出了求解线性方程组的三种不同方法。给出了一种计算闭环约束的有效算法。

具有闭环运动环路的系统比运动树具有更复杂的动力学特性。例如：

1. 运动树的运动自由度是固定的，而闭环系统的运动自由度是可变的。
2. 瞬时运动自由度与运动树中的有限运动自由度相同，但在闭环系统中可能不同。
3. 运动树中的每个力都可以确定，但闭环系统中的一些力是不确定的。当闭环系统过约束时，就会发生这种情况。

这样现象的两个例子如图 3.9 所示。在图 3.9a 中的机构没有有限的运动自由度，但有两个无穷小的运动自由度。当 $\theta \neq 0$ 时，图 3.9b 中的机构具有一个自由度，但是如果 $\theta = 0$ ，则两个手臂 A 和 B 能够独立移动，并且机构具有两个自由度。此外，在这两个运动区域之间的边界处，机构具有三个无穷小运动自由度。这两种机构都是平面的，并因此是过度约束。因此，关约束力的平面外组件是不确定的。这种不确定性对这些机构的运动没有影响，但使其动力学计算复杂化。

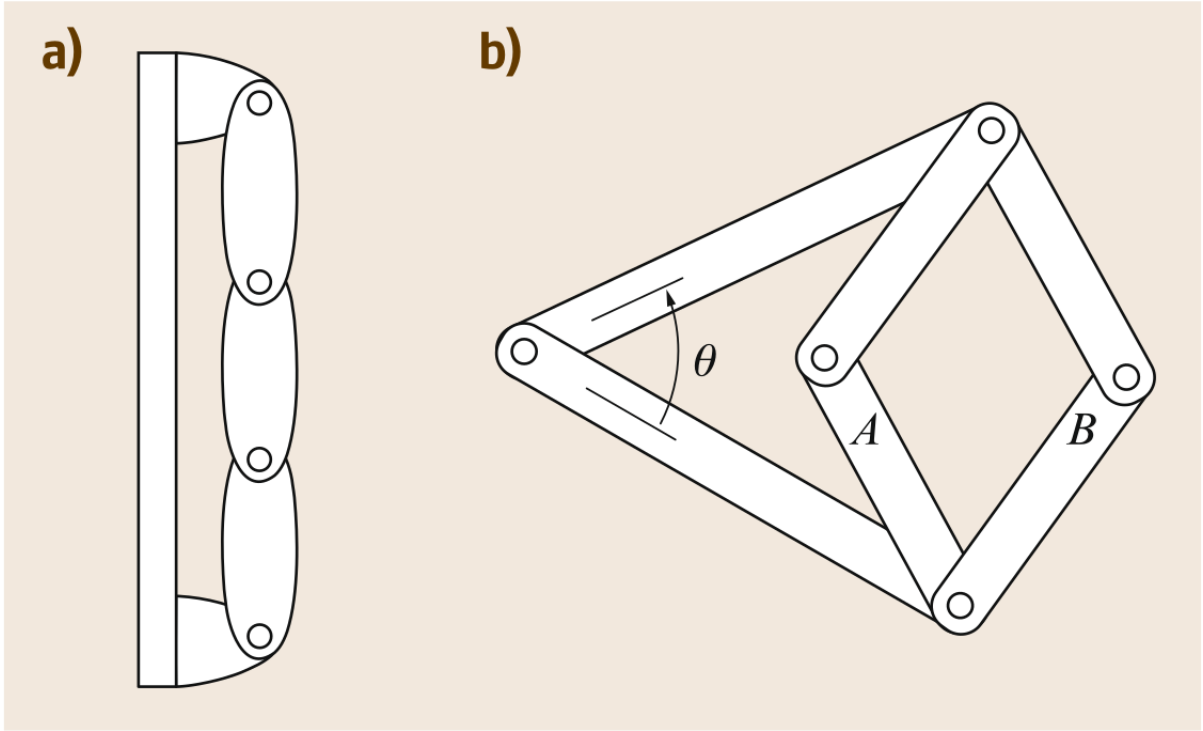


图 3.9: 病态的闭环系统。(a) 改变运动自由度的示例, (b) 具有不同的有限和无穷小运动自由度的示例

3.6.1 闭环算法公式

一个闭环系统可以建模为一个受一组闭环约束的生成树。如果

$$\mathbf{H}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \boldsymbol{\tau}_g = \boldsymbol{\tau}$$

是生成树本身的运动方程, 则闭环系统的运动方程为

$$\mathbf{H}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \boldsymbol{\tau}_g = \boldsymbol{\tau} + \boldsymbol{\tau}^a + \boldsymbol{\tau}^c, \quad (3.92)$$

其中 $\boldsymbol{\tau}^a$ 和 $\boldsymbol{\tau}^c$ 分别是闭环主动力和约束力的向量, 在生成树的广义力坐标中表达。 $\boldsymbol{\tau}^a$ 为已知量, 而 $\boldsymbol{\tau}^c$ 为未知量。 $\boldsymbol{\tau}^a$ 来自作用于闭环关节 (弹簧、阻尼器和执行器) 的力元素。如果不存在此类力元素, 则 $\boldsymbol{\tau}^a = \mathbf{0}$ 。

闭环约束限制了生成树的运动。在加速度层级上, 这些约束可以用线性方程的形式表达,

$$\mathbf{L}\ddot{\mathbf{q}} = \mathbf{1}, \quad (3.93)$$

其中 \mathbf{L} 是 $n^c \times n$ 矩阵。 n^c 是由于闭环关节而产生的约束数量, 并由以下公式给出为

$$n^c = \sum_{k=N_B+1}^{N_J} n_k^c, \quad (3.94)$$

其中 n_k^c 是关节 k 施加的约束数量。如果 $\text{rank}(\mathbf{L}) < n^c$, 则闭环约束是线性相关的, 并且闭环机制是过度约束的。闭环系统的机动性 (即其运动自由度) 由以下公式给出为

$$\text{mobility} = n - \text{rank}(\mathbf{L}). \quad (3.95)$$

给定在方程 (3.93) 中的形式的约束方程, 则约束力可在以下形式中表达为

$$\boldsymbol{\tau}^c = \mathbf{L}^T \boldsymbol{\lambda}, \quad (3.96)$$

其中 $\boldsymbol{\lambda} = (\lambda_{N_B+1}^T \cdots \lambda_{N_J}^T)^T$ 是一个未知约束力变量 (或拉格朗日乘子) 的 $n^c \times 1$ 的向量。如果机构是过约束的, 则 \mathbf{L}^T 将是一个空 (null) 空间, 并且位于该空空间中的 $\boldsymbol{\lambda}$ 的分量将是不确定的。

通常可以提前识别冗余约束。例如, 如果已知运动环路是平面的, 则平面外闭环约束是多余的。在这些情况下, 删除 \mathbf{L} 的相应行以及 $\mathbf{1}$ 和 $\boldsymbol{\lambda}$ 的元素是有利的。删除的 $\boldsymbol{\lambda}$ 元素可以被分配一个零值。

组合方程 (3.92)、(3.93) 和 (3.96) 可生成闭环系统的以下运动方程

$$\begin{pmatrix} \mathbf{H} & \mathbf{L}^T \\ \mathbf{L} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \ddot{\mathbf{q}} \\ -\boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\tau} + \boldsymbol{\tau}^a - (\mathbf{C}\dot{\mathbf{q}} + \boldsymbol{\tau}_g) \\ \mathbf{1} \end{pmatrix}. \quad (3.97)$$

该系统矩阵是对称的, 但不是确定的。若 \mathbf{L} 具有满秩, 则系统矩阵是非奇异的, 否则它将是奇异的, 并且 $\boldsymbol{\lambda}$ 的一个或多个元素将是不确定的。

方程 (3.97) 可用以下任意一方法求解:

1. 直接求解 $\ddot{\mathbf{q}}$ 和 $\boldsymbol{\lambda}$ 。
2. 先求解 $\boldsymbol{\lambda}$, 再求解 $\ddot{\mathbf{q}}$ 。
3. 为 $\ddot{\mathbf{q}}$ 求解方程 (3.93), 将结果代入方程 (3.92), 消除未知的约束力, 并求解剩余的未知数。

方法 1 是最简单的, 但通常也是最不有效的。该方法适用于系统矩阵非奇异的情况。当系统矩阵的大小为 $(n + n^c) \times (n + n^c)$ 时, 该方法复杂度为 $O((n + n^c)^3)$ 。

方法 2 在 $n \gg n^c$ 时特别有用, 并提供了在生成树上使用 $O(n)$ 算法的机会 [3.6]。从方程 (3.97),

$$\mathbf{LH}^{-1}\mathbf{L}^T\boldsymbol{\lambda} = \mathbf{1} - \mathbf{LH}^{-1}[\boldsymbol{\tau} + \boldsymbol{\tau}^a - (\mathbf{C}\dot{\mathbf{q}} + \boldsymbol{\tau}_g)]. \quad (3.98)$$

该方程可通过 $O(n)$ 算法在 $O(n(n^c)^2)$ 运算中制定, 并在 $O((n^c)^3)$ 中求解。一旦 $\boldsymbol{\lambda}$ 已知, 则可通过 $O(nn^c)$ 运算中的方程 (3.96) 计算 $\boldsymbol{\tau}^c$, 并通过 $O(n)$ 算法求解方程 (3.92); 总复杂度为 $O(n(n^c)^2 + (n^c)^3)$ 。若 \mathbf{L} 为秩亏, 则 $\mathbf{LH}^{-1}\mathbf{L}^T$ 是奇异的; 但它仍然是一个半正定矩阵, 并且与在方程 (3.97) 中不确定的系统矩阵的一个奇异实例相比, 它提出了一个稍微简单的因式分解问题。

如果 $n - n^c$ 很小, 或者如果 \mathbf{L} 被期望为秩亏, 则方法 3 是有用的。求解方程 (3.93) 采用一种特殊的高斯消元法 (或类似的消元法), 该消元法配有数值的秩检验, 用于求解欠定系统。该解为以下形式的方程

$$\ddot{\mathbf{q}} = \mathbf{K}\mathbf{y} + \ddot{\mathbf{q}}_0,$$

其中 $\ddot{\mathbf{q}}_0$ 是方程 (3.93) 的任意特解, \mathbf{K} 是一个性质为 $\mathbf{LK} = \mathbf{0}$ 的 $n \times (n - \text{rank}(\mathbf{L}))$ 矩阵, \mathbf{y} 是 $n - \text{rank}(\mathbf{L})$ 的未知向量。(通常, \mathbf{y} 是 $\ddot{\mathbf{q}}$ 元素的线性独立子集。) 为 $\ddot{\mathbf{q}}$ 将此表达式代入方程 (3.92), 并在两侧预乘 \mathbf{K}^T 以消除 $\boldsymbol{\tau}^c$, 产生

$$\mathbf{K}^T\mathbf{H}\mathbf{K}\mathbf{y} = \mathbf{K}^T(\boldsymbol{\tau} + \boldsymbol{\tau}^a - (\mathbf{C}\dot{\mathbf{q}} + \boldsymbol{\tau}_g) - \mathbf{H}\ddot{\mathbf{q}}_0). \quad (3.99)$$

该方法也具有立方复杂度, 但若 $n - n^c$ 较小, 它可以是效率最高的。据报道, 它比方法 1 更稳定 [3.34]。

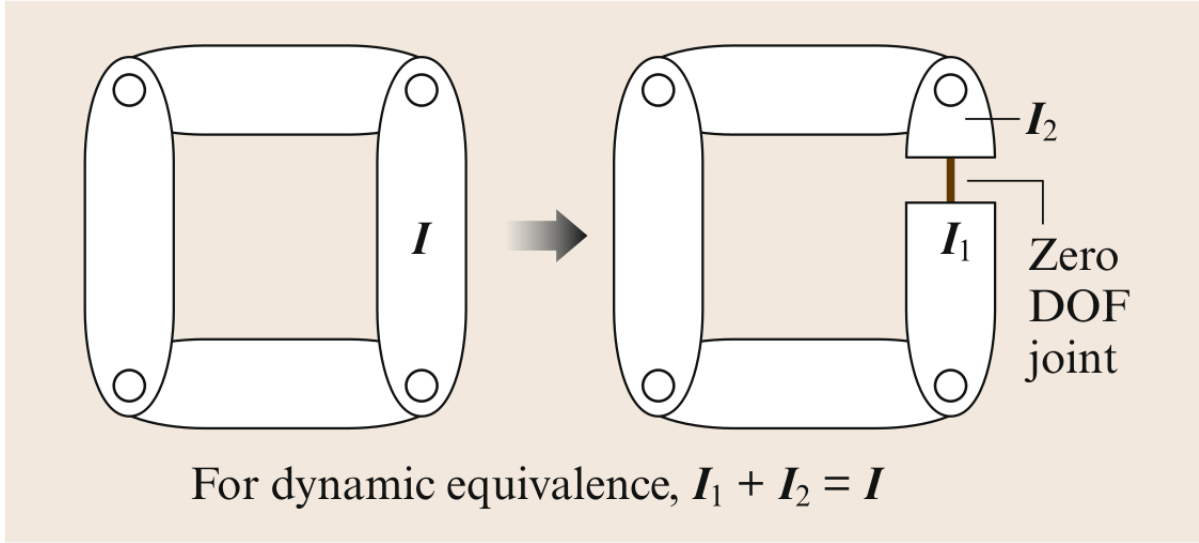


图 3.10: 插入一个零自由度 (Zero-DOF) 关节, 准备在该关节处切断打开环路

3.6.2 闭环算法

计算 \mathbf{H} 和 $\mathbf{C}\dot{\mathbf{q}} + \boldsymbol{\tau}_g$ 的算法分别见第 3.5.3 和 3.5.1 节。现在只剩下如何计算 \mathbf{L}, \mathbf{l} 和 $\boldsymbol{\tau}^a$ 。为简单起见, 我们将假定所有闭环关节都是零自由度关节。

这一假设不失一般性: 只需通过切割连杆而不是关节来断开环路 (图 3.10)。但是, 可能会有一些效率损失。根据这个假设, 我们只需要计算 \mathbf{L} 和 \mathbf{l} , 因为 $\boldsymbol{\tau}^a = \mathbf{0}$ 。

循环约束

在一般情况下, 循环 k 的速度约束方程为

$$(\boldsymbol{\Psi}_k^c)^T (\mathbf{v}_{s(k)} - \mathbf{v}_{p(k)}) = \mathbf{0}, \quad (3.100)$$

并且加速度约束是

$$(\boldsymbol{\Psi}_k^c)^T (\mathbf{a}_{s(k)} - \mathbf{a}_{p(k)}) + (\dot{\boldsymbol{\Psi}}_k^c)^T (\mathbf{v}_{s(k)} - \mathbf{v}_{p(k)}) = \mathbf{0}. \quad (3.101)$$

然而, 如果每个闭环关节都是零自由度, 则这些方程简化为

$$\mathbf{v}_{s(k)} - \mathbf{v}_{p(k)} = \mathbf{0} \quad (3.102)$$

和

$$\mathbf{a}_{s(k)} - \mathbf{a}_{p(k)} = \mathbf{0}. \quad (3.103)$$

让我们定义一个循环雅可比矩阵, \mathbf{J}_k , 有性质为

$$\mathbf{v}_{s(k)} - \mathbf{v}_{p(k)} = \mathbf{J}_k \dot{\mathbf{q}}. \quad (3.104)$$

\mathbf{J}_k 是由公式定义的一个 $6 \times n$ 矩阵

$$\mathbf{J}_k = \begin{pmatrix} e_{1k} \boldsymbol{\Phi}_1 & \cdots & e_{N_B k} \boldsymbol{\Phi}_{N_B} \end{pmatrix}, \quad (3.105)$$

其中

$$e_{ik} = \begin{cases} +1 & \text{if } s(k) \in c^*(i) \text{ and } p(k) \notin c^*(i), \\ -1 & \text{if } p(k) \in c^*(i) \text{ and } s(k) \notin c^*(i), \\ 0 & \text{otherwise.} \end{cases}$$

换言之, 如果关节 i 位于到 $s(k)$ 的路径上而不是到 $p(k)$ 的路径上, 则 $e_{ik} = +1$; 如果关节 i 位于 $p(k)$ 的路径上而不是 $s(k)$ 的路径上, 则 $e_{ik} = -1$; 如果关节 i 位于两条路径上或两条路径都不在, 则为 $e_{ik} = 0$ 。

现在循环加速度约束可被写为

$$\begin{aligned} \mathbf{0} &= \mathbf{a}_{s(k)} - \mathbf{a}_{p(k)} \\ &= \mathbf{J}_k \ddot{\mathbf{q}} + \dot{\mathbf{J}}_k \dot{\mathbf{q}} \\ &= \mathbf{J}_k \ddot{\mathbf{q}} + \mathbf{a}_{s(k)}^{\text{vp}} - \mathbf{a}_{p(k)}^{\text{vp}}, \end{aligned} \quad (3.106)$$

其中 \mathbf{a}_i^{vp} 是连杆 i 的速度-积加速度, 这是如果 $\ddot{\mathbf{q}}$ 为零时的加速度。在方程 (3.84) 计算向量 $\mathbf{C}\dot{\mathbf{q}} + \boldsymbol{\tau}_g$ 的过程中, 计算每个连杆的速度-积加速度。如果 RNEA 被用于计算 $\mathbf{C}\dot{\mathbf{q}} + \boldsymbol{\tau}_g$, 则 \mathbf{a}_i^{vp} 将是由 RNEA 计算的 \mathbf{a}_i 的值, 其加速度参数设置为零。

矩阵 \mathbf{L} 和 \mathbf{l} 现在可以表达为

$$\mathbf{L} = \begin{pmatrix} \mathbf{L}_{N_B+1} \\ \vdots \\ \mathbf{L}_{N_J} \end{pmatrix} \quad \text{and} \quad \mathbf{l} = \begin{pmatrix} \mathbf{l}_{N_B+1} \\ \vdots \\ \mathbf{l}_{N_J} \end{pmatrix}, \quad (3.107)$$

其中

$$\mathbf{L}_k = \mathbf{J}_k \quad (3.108)$$

和

$$\mathbf{l}_k = \mathbf{a}_{p(k)}^{\text{vp}} - \mathbf{a}_{s(k)}^{\text{vp}}. \quad (3.109)$$

约束稳定性

在实际应用中, 有必要稳定闭环约束, 否则在模拟过程中, 由于数值积分误差, 闭环约束将简单地分离。标准技术源于 *Baumgarte* [3.3,7,35], 并包括替换形式的每个约束方程

$$a_e = 0,$$

使用其中一种形式

$$a_e + K_v v_e + K_p p_e = 0,$$

其中, a_e , v_e 和 p_e 分别为加速度、速度和位置误差, K_v 和 K_p 为正常量。通常, 选择一个时间常量, t_c , 根据位置和速度误差衰减的速度。 K_p 和 K_v 由公式 $K_v = 2/t_c$ 和 $K_p = 1/t_c^2$ 给出。然而, 选择 t_c 并没有好的规则, 如果 t_c 太长, 则循环约束误差积累的速度比衰减的速度快; 如果 t_c 太短, 则运动方程变得过于僵硬, 导致数值积分精度损失。大型、慢速工业机器人的合理值为 $t_c = 0.1$, 而小型、快速机器人可能需要 $t_c = 0.01$ 。

为了将稳定项纳入循环约束方程，我们将方程 (3.109) 替换为

$$\mathbf{l}_k = \mathbf{a}_{p(k)}^{\text{vp}} - \mathbf{a}_{s(k)}^{\text{vp}} - K_v (\mathbf{v}_{s(k)} - \mathbf{v}_{p(k)}) - K_p \mathbf{p}_{ek}, \quad (3.110)$$

其中 \mathbf{p}_{ek} 是一个向量，表示在循环 k 中的位置误差。在绝对坐标 (即，连杆 0 坐标) 中， \mathbf{p}_{ek} 给出为

$$\mathbf{p}_{ek} = \text{x_to_vec} \left({}^0\mathbf{X}_{p(k)} \mathbf{X}_{L1}^{-1}(k) \mathbf{X}_{L2}(k)^{s(k)} \mathbf{X}_0 \right), \quad (3.111)$$

其中， $\mathbf{X}_{L1}(k)$ 和 $\mathbf{X}_{L2}(k)$ 变换在方程 (3.62) 和 (3.63) 中定义，并且在图 3.4 中显示，对于关节 k ，以及 $\text{x_to_vec}({}^B\mathbf{X}_A)$ 计算近似于从帧 A 到帧 B 的位移的向量，假设该位移是无穷小的。 x_to_vec 可被定义为

$$\text{x_to_vec}(\mathbf{X}) = \frac{1}{2} \begin{pmatrix} X_{23} - X_{32} \\ X_{31} - X_{13} \\ X_{12} - X_{21} \\ X_{53} - X_{62} \\ X_{61} - X_{43} \\ X_{42} - X_{51} \end{pmatrix}. \quad (3.112)$$

算法

算法 3.7 显示了对于特殊情况的一种计算 \mathbf{L} 和 \mathbf{l} 的算法，当所有闭环关节都具有 0-DOF 时。它通过将公式化闭环约束所需的每一个量变换为单个坐标系统，在本例中为绝对 (连杆 0) 坐标，将简单性与良好性能相结合，因此不需要进一步的转换。

第一遍循环计算从绝对坐标到连杆坐标的变换，并使用它们将 Φ_i 变换为绝对坐标。仅变换在闭环约束中所需的 Φ_i 。

根据方程 (3.105)，第二个遍计算 \mathbf{L} (可以是稀疏的) 的非零元素。内部 `while` 循环终止于循环的根节点，它是连杆 $p(k)$ 和 $s(k)$ 的最高编号的共同祖先。如果它们没有其它共同的祖先，它可能是固定基座。根据方程 (3.110)，第二遍循环以在绝对坐标中 \mathbf{l} 的计算为结束。

3.7 结论与进一步的阅读

本章介绍了应用于机器人机构的刚体动力学基本原理。它涵盖了以下主题：空间向量代数，它为描述和实现动力学方程和算法提供了简洁的符号；机器人学中最常用的正则运动方程组；如何构建机器人的动力学模型；还有用于计算逆向动力学、正向动力学以及关节空间和操作空间惯量矩阵的几种基于模型的高效算法。

在动力学中有许多主题在本章中未提及，但可在本手册的后续章节中找到。第 11 章介绍了具有弹性连杆和关节的机器人的动力学；第 6 章讨论了动力学模型参数的识别问题；第 37 章描述了机器人与环境中物体之间物理接触的动力学；第 55 章描述了带有浮动基座的机器人的动力学。

我们在结束本章时指出，在文献 [3.36] 中可以找到机器人动力学的简要历史，并且在文献 [3.8,10,15,29,37-40] 等书中可以找到更广泛的机器人动力学论述。最后，下面列出了一些进一步阅读的建议。

Algorithm 3.7 计算闭环约束的算法

input: *model, RNEA partial results***output:** L, l *model data* : $N_B, p(i), N_J, p(k), s(k), LB(i), \mathbf{X}_{L1}(k), \mathbf{X}_{L2}(k), K_p, K_v$ *RNEA data* : $\Phi_i, {}^i\mathbf{X}_{p(i)}, \mathbf{v}_{p(k)}, \mathbf{v}_{s(k)}, \mathbf{a}_{p(k)}^{\text{vp}}, \mathbf{a}_{s(k)}^{\text{vp}}$ **for** $i = 1$ **to** N_B **do** **if** $p \neq 0$ **then**

$${}^i\mathbf{X}_0 = {}^i\mathbf{X}_{p(i)} {}^{p(i)}\mathbf{X}_0$$

end if **if** $LB \neq \text{null}$ **then**

$${}^0\Phi_i = {}^i\mathbf{X}_0^{-1} \Phi_i$$

end if**end for** $L = \mathbf{0}$ **for** $k = N_B + 1$ **to** N_J **do** $i = p(k)$ $j = s(k)$ **while** $i \neq j$ **do** **if** $i > j$ **then**

$$L_{k,i} = -{}^0\Phi_i$$

 $i = p(i)$ **else**

$$L_{k,j} = {}^0\Phi_j$$

 $j = p(j)$ **end if** **end while**

$$\mathbf{a}_e = {}^{s(k)}\mathbf{X}_0^{-1} \mathbf{a}_{s(k)}^{\text{vp}} - {}^{p(k)}\mathbf{X}_0^{-1} \mathbf{a}_{p(k)}^{\text{vp}}$$

$$\mathbf{v}_e = {}^{s(k)}\mathbf{X}_0^{-1} \mathbf{v}_{s(k)} - {}^{p(k)}\mathbf{X}_0^{-1} \mathbf{v}_{p(k)}$$

$$\mathbf{p}_e = \text{x_to_vec} \left({}^{p(k)}\mathbf{X}_0^{-1} \mathbf{X}_{L1}^{-1}(k) \mathbf{X}_{L2}(k) {}^{s(k)}\mathbf{X}_0 \right)$$

$$l_k = -\mathbf{a}_e - K_v \mathbf{v}_e - K_p \mathbf{p}_e$$

end for

3.7.1 多体动力学

机器人动力学可以看作是多体动力学更广泛学科的一个子集 (或一个特定应用)。关于多体动力学的书籍包括文献 [3.3, 14,35,41-46]。当然, 多体动力学又是经典力学的一个子集; 这门学科的数学基础可以在任何一本关于经典力学的好书中找到, 比如文献 [3.13]。

3.7.2 替代表示

本章使用空间向量来表达运动方程。使用空间向量有多种选择: 其它类型的 6-D 向量、3-D 向量、 4×4 矩阵和空间算子代数。所有 6-D 向量形式都类似, 但不完全相同。空间向量的主要替代方法有: 旋量 [3.10-12]、矩量 [3.47]、李代数 [3.12,48] 和特殊表示法。(一种特殊的表示法, 其中 3-D 向量分组成对以减少代数的体积。) 三维向量是大多数经典力学和多体文本中使用的形式, 也是 6-D 向量和 4×4 矩阵形式的前身。 4×4 矩阵在机器人学中很流行, 因为它们对运动学非常有用。但是, 它们对于动力学不是很有用。 4×4 动力学的矩阵公式可在文献 [3.37,49,50] 中找到。空间算子代数是 Rodriguez、Jain 等人在喷气推进实验室 (Jet Propulsion Laboratory, JPL) 开发的。它使用 $6N$ 维向量和 $6N \times 6N$ 矩阵, 后者被视为线性算子。该表示法的示例可在文献 [3.38, 51-53] 中找到。

3.7.3 替代公式

本章使用了运动方程的向量公式, 通常称为牛顿公式或牛顿-欧拉公式。主要的替代是拉格朗日公式, 其中的运动方程是通过拉格朗日方程获得。拉格朗日公式的例子可在文献 [3.9, 10,18,54,55] 中找到。*Kane* 方法也已应用于机器人学 [3.56,57]。

3.7.4 效率

由于需要实时实现, 特别是在控制方面, 机器人学界一直关注计算效率问题。对于逆向动力学 (*inverse dynamics*), Luh 等人 [3.4] 的 $O(n)$ 递归牛顿-欧拉算法 (RNEA) 仍然是最重要的算法。在文献 [3.58,59] 中给出了对该算法的进一步改进, 对于正向动力学 (*forward dynamics*), 本章中介绍的两种算法对于计算考虑仍然是最重要的: Featherstone [3.1] 开发的 $O(n)$ 铰接体算法 (ABA) 和基于 Walker 和 Orin [3.5] 的复合刚体算法 (CRBA) 的 $O(n^3)$ 算法。多年来, ABA 算法进行了改进 [3.15,17,25], 使其比基于 CRBA 的算法, 在较小的 n 值时更有效率。然而, 最近将 CRBA 应用于分支运动学树 [3.26] 和具有运动控制附件 [3.60] 的机器人系统, 继续表明该方法的可行性。

对于关节空间惯量矩阵 (*joint-space inertia matrix*), CRBA[3.5] 是最重要的算法。多年来, 为了提高计算效率, 已经进行了大量改进和修改 [3.15,61-63]。对于操作空间惯量矩阵 (*operational-space inertia matrix*), 已经开发了有效的 $O(n)$ 算法 [3.28-30], 并将其应用于日益复杂的系统 [3.20,31,33]。利用分支诱导的稀疏性也产生了有效的算法 [3.32]。

3.7.5 精度

对动力学算法的数值精度、模拟的精度 (即数值积分精度) 或动力学模型的精度都会引起关注。与效率相比, 动力学算法的数值精度受到的关注相对较少。RNEA、CRBA 和 ABA 都已经在大量的刚体系统上进行了精度测试, 但其它大多数算法就不是这样了。刚体系统通常是条件不良的, 也就是说, 施加的力 (或模型参数) 的微小变化会使所产生的加速度发生很大的变化。Featherstone

[3.27] 对这一现象进行了研究, 他发现条件不良会随着机体数量的增加而变得更加严重, 在最坏的情况下, 它可以增长到 $O(n^4)$ 的比例。关于这个主题的其它出版物包括文献 [3.8, 34, 64, 65]。

3.7.6 软件包

为了给多体系统, 特别是机器人系统提供动力学模拟能力, 已经开发了许多软件包。有几个是用 MATLAB 编写的, 以便于与其它分析、控制和模拟程序整合。许多软件包是开放源码的, 有些软件包以相对较低的成本提供给用户。它们的能力有很多不同之处, 包括: 速度、支持的拓扑结构和关节模型、精度、底层的动力学公式和相关的复杂程度、用户界面、图形支持、数值集成程序、与其它代码的集成、应用支持和成本。通常被引用的有 Adams [3.66], Autolev [3.67], Bullet [3.68], DART [3.69], DynaMechs [3.70], Gazebo [3.71], Open Dynamics Engine [3.72], Robotics Studio [3.73], Robotics Toolbox [3.74], Robotran [3.75, 76], SD/FAST [3.77], Simbody [3.78], SimMechanics [3.79], SYMORO [3.80, 81] 和 Webots [3.82]。

3.7.7 符号简化

符号简化技术采用通用的动力学算法, 并将其符号化地应用于特定的动力学模型。其结果是一个赋值语句列表, 详细说明了, 如果该算法被实际执行, 它会做什么。然后检查该列表并精简所有不必要的计算, 剩余部分以计算机源代码的形式输出到文本文件中。这段代码的运行速度比原来的通用算法快十倍, 但它只适用于一个动态模型。Autolev[3.67] 和 SD/FAST[3.77] 均使用此技术。其它关于动力学符号简化的出版物包括文献 [3.76,80,81,83-88]。

3.7.8 并行计算机算法

为了加快常见的动力学计算, 已经为并行和流水线计算机开发了许多算法。对于逆向动力学 (*inverse dynamics*), 早期的工作集中于在多达 n 个处理器上加速 $O(n)$ 的 RNEA[3.89,90], 而后续的工作则产生了 $O(\log_2 n)$ 算法 [3.91, 92]。对于用 $O(n^2)$ 的 CRBA 来计算关节空间惯量矩阵 (*joint-space inertia matrix*), 早期的工作为 n 个处理器产生了 $O(\log_2 n)$ 的算法, 以计算矩阵的复合刚体惯量和对角元素 [3.93,94]。随后的工作为 $O(n^2)$ 个处理器产生了 $O(\log_2 n)$ 的算法以计算整个矩阵 [3.95,96]。对于正向动力学 (*forward dynamics*), 并行/流水线超级计算机上的多机械臂系统获得了加速 [3.97]。用于 n 个处理器的第一个 $O(\log_2 n)$ 算法是为非分支串联链开发的 [3.98]。最近的工作集中在用于更复杂结构的 $O(\log_2 n)$ 算法上 [3.65, 99, 100]。

3.7.9 拓扑变化系统

有许多机器人机构, 由于接触条件的变化, 特别是与环境的接触条件的变化, 其拓扑结构随时间而变化。在腿部机身中, 使用顺应性地面接触模型来计算接触力, 将闭环结构减少为树形结构 [3.101]。然而, 对于接触非常刚性的情况, 可能会产生数值积分问题。在最近的工作中 [3.40,102], 在假设硬接触约束的情况下, 使用了一种有效的方法来减少大量的坐标变量, 而这些变量在通用运动分析系统中可能是必要的 [3.43]。此外, 它们能够在结构变化时自动识别变量, 并且一种计算位形变化后速度边界条件的方法已被开发出来 [3.40, 102]。

3.8 References

- 3.1 R. Featherstone: The calculation of robot dynamics using articulated-body inertias, *Int. J. Robotics Res.* 2(1), 13-30 (1983)
- 3.2 J.J. Craig: *Introduction to Robotics: Mechanics and Control*, 3rd edn. (Prentice Hall, Upper Saddle River 2005)
- 3.3 R.E. Roberson, R. Schwertassek: *Dynamics of Multibody Systems* (Springer, Berlin, Heidelberg 1988)
- 3.4 J.Y.S. Luh, M.W. Walker, R.P.C. Paul: On-line computational scheme for mechanical manipulators, *Trans. ASME J. Dyn. Syst. Meas. Control* 102(2), 69-76 (1980)
- 3.5 M.W. Walker, D.E. Orin: Efficient dynamic computer simulation of robotic mechanisms, *Trans. ASME J. Dyn. Syst. Meas. Control* 104, 205-211 (1982)
- 3.6 D. Baraff: Linear-time dynamics using lagrange multipliers, *Proc. 23rd Annu. Conf. Comp. Graph. Interact. Tech.*, New Orleans (1996) pp. 137-146
- 3.7 J. Baumgarte: Stabilization of constraints and integrals of motion in dynamical systems, *Comput. Methods Appl. Mech. Eng.* 1, 1-16 (1972)
- 3.8 R. Featherstone: *Rigid Body Dynamics Algorithms* (Springer, New York 2008)
- 3.9 R.M. Murray, Z. Li, S.S. Sastry: *A Mathematical Introduction to Robotic Manipulation* (CRC, Boca Raton 1994)
- 3.10 J. Angeles: *Fundamentals of Robotic Mechanical Systems*, 2nd edn. (Springer, New York 2003)
- 3.11 R.S. Ball: *A Treatise on the Theory of Screws* (Cambridge Univ. Press, London 1900), Republished (1998)
- 3.12 J.M. Selig: *Geometrical Methods in Robotics* (Springer, New York 1996)
- 3.13 D.T. Greenwood: *Principles of Dynamics* (PrenticeHall, Englewood Cliffs 1988)
- 3.14 F.C. Moon: *Applied Dynamics* (Wiley, New York 1998)
- 3.15 R. Featherstone: *Robot Dynamics Algorithms* (Kluwer, Boston 1987)
- 3.16 R. Featherstone: *Spatial v2*, <http://royfeatherstone.org/spatial/v2> (2012)
- 3.17 S. McMillan, D.E. Orin: Efficient computation of articulated-body inertias using successive axial screws, *IEEE Trans. Robotics Autom.* 11, 606-611 (1995)
- 3.18 L. Sciavicco, B. Siciliano: *Modeling and Control of Robot Manipulators*, 2nd edn. (Springer, London 2000)
- 3.19 J. Slotine, W. Li: On the adaptive control of robot manipulators, *Int. J. Robotics Res.* 6(3), 49-59 (1987)

- 3.20 K.S. Chang, O. Khatib: Operational space dynamics: Efficient algorithms for modeling and control of branching mechanisms, Proc. IEEE Int. Conf. Robotics Autom., San Francisco (2000) pp. 850-856
- 3.21 O. Khatib: A unified approach to motion and force control of robot manipulators: The operational space formulation, IEEE J. Robotics Autom. 3(1), 4353 (1987)
- 3.22 Y.F. Zheng, H. Hemami: Mathematical modeling of a robot collision with its environment, J. Robotics Syst. 2(3), 289-307 (1985)
- 3.23 W. Khalil, E. Dombre: Modeling, Identification and Control of Robots (Kogan Page Sci., London 2002)
- 3.24 J. Denavit, R.S. Hartenberg: A kinematic notation for lower-pair mechanisms based on matrices, J. Appl. Mech. 22, 215-221 (1955)
- 3.25 H. Brandl, R. Johanni, M. Otter: A very efficient algorithm for the simulation of robots and similar multibody systems without inversion of the mass matrix, Proc. IFAC/IFIP/IMACS Int. Symp. Theory Robots, Vienna (1986)
- 3.26 R. Featherstone: Efficient factorization of the joint space inertia matrix for branched kinematic trees, Int. J. Robotics Res. 24(6), 487-500 (2005)
- 3.27 R. Featherstone: An empirical study of the joint space inertia matrix, Int. J. Robotics Res. 23(9), 859-871 (2004)
- 3.28 K. Kreutz-Delgado, A. Jain, G. Rodriguez: Recursive formulation of operational space control, Proc. IEEE Int. Conf. Robotics Autom., Sacramento (1991) pp. 1750-1753
- 3.29 K.W. Lilly: Efficient Dynamic Simulation of Robotic Mechanisms (Kluwer, Boston 1993)
- 3.30 K.W. Lilly, D.E. Orin: Efficient $O(N)$ recursive computation of the operational space inertia matrix, IEEE Trans. Syst. Man Cybern. 23(5), 1384-1391 (1993)
- 3.31 G. Rodriguez, A. Jain, K. Kreutz-Delgado: Spatial operator algebra for multibody system dynamics, J. Astronaut. Sci. 40(1), 27-50 (1992)
- 3.32 R. Featherstone: Exploiting sparsity in operational space dynamics, Int. J. Robotics Res. 29(10), 1353-1368 (2010)
- 3.33 P. Wensing, R. Featherstone, D.E. Orin: A reduced order recursive algorithm for the computation of the operational-space inertia matrix, Proc. IEEE Int. Conf. Robotics Autom., St. Paul (2012) pp. 4911-4917
- 3.34 R.E. Ellis, S.L. Ricker: Two numerical issues in simulating constrained robot dynamics, IEEE Trans. Syst. Man Cybern. 24(1), 19-27 (1994)
- 3.35 J. Wittenburg: Dynamics of Systems of Rigid Bodies (Teubner, Stuttgart 1977)

- 3.36 R. Featherstone, D.E. Orin: Robot dynamics: Equations and algorithms, Proc. IEEE Int. Conf. Robotics Autom., San Francisco (2000) pp. 826-834
- 3.37 C.A. Balafoutis, R.V. Patel: Dynamic Analysis of Robot Manipulators: A Cartesian Tensor Approach (Kluwer, Boston 1991)
- 3.38 A. Jain: Robot and Multibody Dynamics: Analysis and Algorithms (Springer, New York 2011)
- 3.39 L.W. Tsai: Robot Analysis and Design: The Mechanics of Serial and Parallel Manipulators (Wiley, New York 1999)
- 3.40 K. Yamane: Simulating and Generating Motions of Human Figures (Springer, Berlin, Heidelberg 2004)
- 3.41 F.M.L. Amirouche: Fundamentals of Multibody Dynamics: Theory and Applications (Birkhäuser, Boston 2006)
- 3.42 M.G. Coutinho: Dynamic Simulations of Multibody Systems (Springer, New York 2001)
- 3.43 E.J. Haug: Computer Aided Kinematics and Dynamics of Mechanical Systems (Allyn and Bacon, Boston 1989)
- 3.44 R.L. Huston: Multibody Dynamics (Butterworths, Boston 1990)
- 3.45 A.A. Shabana: Computational Dynamics, 2nd edn. (Wiley, New York 2001)
- 3.46 V. Stejskal, M. Valášek: Kinematics and Dynamics of Machinery (Marcel Dekker, New York 1996)
- 3.47 L. Brand: Vector and Tensor Analysis, 4th edn. (Wiley/Chapman Hall, New York/London 1953)
- 3.48 F.C. Park, J.E. Bobrow, S.R. Ploen: A lie group formulation of robot dynamics, Int. J. Robotics Res. 14(6), 609-618 (1995)
- 3.49 M.E. Kahn, B. Roth: The near minimum-time control of open-loop articulated kinematic chains, J. Dyn. Syst. Meas. Control 93, 164-172 (1971)
- 3.50 J.J. Uicker: Dynamic force analysis of spatial linkages, Trans. ASME J. Appl. Mech. 34, 418-424 (1967)
- 3.51 A. Jain: Unified formulation of dynamics for serial rigid multibody systems, J. Guid. Control Dyn. 14(3), 531-542 (1991)
- 3.52 G. Rodriguez: Kalman filtering, smoothing, and recursive robot arm forward and inverse dynamics, IEEE J. Robotics Autom. RA-3(6), 624-639 (1987)
- 3.53 G. Rodriguez, A. Jain, K. Kreutz-Delgado: A spatial operator algebra for manipulator modelling and control, Int. J. Robotics Res. 10(4), 371-381 (1991)

- 3.54 J.M. Hollerbach: A recursive Lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation complexity, IEEE Trans. Syst. Man Cybern. SMC-10(11), 730-736 (1980)
- 3.55 M.W. Spong, S. Hutchinson, M. Vidyasagar: Robot Modeling and Control (Wiley, Hoboken 2006)
- 3.56 K.W. Buffinton: Kane's Method in Robotics. In: Robotics and Automation Handbook, ed. by T.R. Kurfess (CRC, Boca Raton 2005), 6-1-6-31
- 3.57 T.R. Kane, D.A. Levinson: The use of Kane's dynamical equations in robotics, Int. J. Robotics Res. 2(3), 3-21 (1983)
- 3.58 C.A. Balafoutis, R.V. Patel, P. Misra: Efficient modeling and computation of manipulator dynamics using orthogonal cartesian tensors, IEEE J. Robotics Autom. 4, 665-676 (1988)
- 3.59 X. He, A.A. Goldenberg: An algorithm for efficient computation of dynamics of robotic manipulators, Proc. 4th Int. Conf. Adv. Robotics, Columbus (1989) pp. 175-188
- 3.60 W. Hu, D.W. Marhefka, D.E. Orin: Hybrid kinematic and dynamic simulation of running machines, IEEE Trans. Robotics 21(3), 490-497 (2005)
- 3.61 C.A. Balafoutis, R.V. Patel: Efficient computation of manipulator inertia matrices and the direct dynamics problem, IEEE Trans. Syst. Man Cybern. 19, 1313-1321 (1989)
- 3.62 K.W. Lilly, D.E. Orin: Alternate formulations for the manipulator inertia matrix, Int. J. Robotics Res. 10, 64-74 (1991)
- 3.63 S. McMillan, D.E. Orin: Forward dynamics of multilegged vehicles using the composite rigid body method, Proc. IEEE Int. Conf. Robotics Autom. (1998) pp. 464-470
- 3.64 U.M. Ascher, D.K. Pai, B.P. Cloutier: Forward dynamics: Elimination methods, and formulation stiffness in robot simulation, Int. J. Robotics Res. 16(6), 749-758 (1997)
- 3.65 R. Featherstone: A divide-and-conquer articulated-body algorithm for parallel $O(\log(n))$ calculation of rigid-body dynamics. Part 2: Trees, loops and accuracy, Int. J. Robotics Res. 18(9), 876-892 (1999)
- 3.66 MSC Software Corporation: Adams, <http://www.mscsoftware.com/>
- 3.67 T. Kane, D. Levinson: Autolev user's manual (OnLine Dynamics Inc., Sunnyvale 2005)
- 3.68 Real-Time Physics Simulation: Bullet, <http://bulletphysics.org/wordpress> (2015)
- 3.69 Georgia Tech Graphics Lab and Humanoid Robotics Lab: DART, <http://dartsim.github.io> (2011)
- 3.70 S. McMillan, D.E. Orin, R.B. McGhee: DynaMechs: An object oriented software package for efficient dynamic simulation of underwater robotic vehicles. In: Underwater Robotic Vehicles: Design and Control, ed. by J. Yuh (TSI Press, Albuquerque 1995) pp. 73-98

- 3.71 Open Source Robotics Foundation: Gazebo, <http://gazebo-sim.org> (2002)
- 3.72 R. Smith: Open Dynamics Engine User Guide, <http://opende.sourceforge.net> (2006)
- 3.73 Microsoft Corporation: Robotics Developer Studio, <http://www.microsoft.com/robotics> (2010)
- 3.74 P.I. Corke: A robotics toolbox for MATLAB, IEEE Robotics Autom. Mag. 3(1), 24-32 (1996)
- 3.75 Robotran: <http://www.robotran.be> (Center for Research in Mechatronics, Université catholique de Louvain 2015)
- 3.76 J.C. Samin, P. Fisette: Symbolic Modeling of Multibody Systems (Kluwer, Dordrecht 2003)
- 3.77 M.G. Hollars, D.E. Rosenthal, M.A. Sherman: SD/FAST User's Manual (Symbolic Dynamics Inc., Mountain View 1994)
- 3.78 M. Sherman, P. Eastman: Simbody, <https://simtk.org/home/simbody> (2015)
- 3.79 G.D. Wood, D.C. Kennedy: Simulating Mechanical Systems in Simulink with SimMechanics (MathWorks Inc., Natick 2003)
- 3.80 W. Khalil, D. Creusot: SYMORO+: A system for the symbolic modeling of robots, Robotica 15, 153-161 (1997)
- 3.81 W. Khalil, A. Vijayalingam, B. Khomutenko, I. Mukhanov, P. Lemoine, G. Ecorchard: OpenSYMORO: An open-source software package for symbolic modelling of robots, Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatron. (2014) pp. 126-1211
- 3.82 Cyberbotics Ltd.: Webots User Guide, <http://www.cyberbotics.com> (2015)
- 3.83 I.C. Brown, P.J. Larcombe: A survey of customised computer algebra programs for multibody dynamic modelling. In: Symbolic Methods in Control System Analysis and Design, ed. by N. Munro (Inst. Electr. Eng., London 1999) pp. 53-77
- 3.84 J.J. Murray, C.P. Neuman: ARM: An algebraic robot dynamic modeling program, Proc. IEEE Int. Conf. Robotics Autom., Atlanta (1984) pp. 103-114
- 3.85 J.J. Murray, C.P. Neuman: Organizing customized robot dynamic algorithms for efficient numerical evaluation, IEEE Trans. Syst. Man Cybern. 18(1), 1151-125 (1988)
- 3.86 F.C. Park, J. Choi, S.R. Ploen: Symbolic formulation of closed chain dynamics in independent coordinates, Mech. Mach. Theory 34, 731-751 (1999)
- 3.87 M. Vukobratovic, N. Kircanski: Real-Time Dynamics of Manipulation Robots (Springer, New York 1985)

- 3.88 J. Wittenburg, U. Wolz: Mesa Verde: A symbolic program for nonlinear articulated-rigid-body dynamics, ASME Des. Eng. Div. Conf., Cincinnati (1985) pp. 1-8, ASME Paper No. 85-DET-151
- 3.89 J.Y.S. Luh, C.S. Lin: Scheduling of parallel computation for a computer-controlled mechanical manipulator, IEEE Trans. Syst. Man Cybern. 12(2), 214-234 (1982)
- 3.90 D.E. Orin: Pipelined approach to inverse plant plus jacobian control of robot manipulators, Proc. IEEE Int. Conf. Robotics Autom., Atlanta (1984) pp. 1691-75
- 3.91 R.H. Lathrop: Parallelism in manipulator dynamics, Int. J. Robotics Res. 4(2), 80-102 (1985)
- 3.92 C.S.G. Lee, P.R. Chang: Efficient parallel algorithm for robot inverse dynamics computation, IEEE Trans. Syst. Man Cybern. 16(4), 532-542 (1986)
- 3.93 M. Amin-Javaheri, D.E. Orin: Systolic architectures for the manipulator inertia matrix, IEEE Trans. Syst. Man Cybern. 18(6), 939-951 (1988)
- 3.94 C.S.G. Lee, P.R. Chang: Efficient parallel algorithms for robot forward dynamics computation, IEEE Trans. Syst. Man Cybern. 18(2), 238-251 (1988)
- 3.95 M. Amin-Javaheri, D.E. Orin: Parallel algorithms for computation of the manipulator inertia matrix, Int. J. Robotics Res. 10(2), 162-170 (1991)
- 3.96 A. Fijany, A.K. Bejczy: A class of parallel algorithms for computation of the manipulator inertia matrix, IEEE Trans. Robotics Autom. 5(5), 600-615 (1989)
- 3.97 S. McMillan, P. Sadayappan, D.E. Orin: Parallel dynamic simulation of multiple manipulator systems: Temporal versus spatial methods, IEEE Trans. Syst. Man Cybern. 24(7), 982-990 (1994)
- 3.98 A. Fijany, I. Sharf, G.M.T. D'Eleuterio: Parallel $O(\log N)$ algorithms for computation of manipulator forward dynamics, IEEE Trans. Robotics Autom. 11(3), 389 – 400 (1995)
- 3.99 R. Featherstone: A divide-and-conquer articulated-body algorithm for parallel $O(\log(n))$ calculation of rigid-body dynamics. Part 1 : Basic algorithm, Int. J. Robotics Res. 18(9), 867-875 (1999)
- 3.100 R. Featherstone, A. Fijany: A technique for analyzing constrained rigid-body systems and its application to the constraint force algorithm, IEEE Trans. Robotics Autom. 15(6), 1140-1144 (1999)
- 3.101 P.S. Freeman, D.E. Orin: Efficient dynamic simulation of a quadruped using a decoupled treestructured approach, Int. J. Robotics Res. 10, 619-627 (1991)
- 3.102 Y. Nakamura, K. Yamane: Dynamics computation of structure-varying kinematic chains and its application to human figures, IEEE Trans. Robotics Autom. 16(2), 124-134 (2000)