

绪 论

由于机器人、飞行器和复杂液压系统等大规模机械系统的数学模型具有显著的非线性和复杂性，因此应用控制理论研究这类系统是极为困难的。本书研究的对象是工业机器人系统，为达到位置的高精度要求和提高运动速度，不可避免地要应用控制理论，并且需要探索新的控制算法。到目前为止，适用于局部简化的，或甚至在线运行的精确数学模型的某些控制方法已达到相当先进的水平。然而对机器人的在线控制来说，对每一个采样周期都要计算模型方程，因为大多数机械操作手的共振频率约为 10Hz，所以采样频率一般不应低于 50Hz。然而，用现代微型计算机在 20ms 时间内，实时地解决操作手动力学方程问题则是相当困难和复杂的。

本书中将提出解决上述问题的新方法，在阐述这些新方法之前，让我们先叙述几个在机器人建模方面目前所得到的结果。根据不同的准则可把这些结果系统化。因此，可按建立运动方程的力学规律对所用方法进行分类，以此作为准则，我们就可按拉格朗日方程、牛顿-欧拉方程、Appel 方程和其它方程来划分所用方法，也可把所用方法能否求解动力学正问题和逆问题的性质作为另外的分类准则。所谓动力学的正问题指的是，已知驱动力(转矩)，确定机器人的运动；而动力学的逆问题指的是，给定操作手的运动，确定驱动力。很明显，能给出动力学两种问题解的方法具有特别重要的意义。建立模型所需的浮点乘法/加法的运算次数也可作为比较所用方

法的另一个准则。毋庸置疑,从在线应用观点来看,这是最重要的分类准则。

Uicker 得到了拉格朗日法这类方法的第一个结果,这种方法既能求解动力学的正问题,又能求解动力学的逆问题。由于 Uicker 的方法主要涉及某类闭环机构,因此 Kahn 研究出一种开运动链的建模算法。Woo 和 Freudenstein 通过引入螺旋计算修改并重新阐述了这种方法。为减少这种方法的数值运算的次数,Mahil 和 Renaud 提出了对这种方法的改进方案。这些算法的共同特点是,它们没有利用递推关系式,而是利用了模型矩阵元素的闭式表示式。然而,这些方法的乘法/加法运算的次数是 n^4 (n 表示机器人的自由度数) 的函数。于是,三个自由度的操作手需要数千次乘法和加法运算,六个自由度的操作手的运算次数比它还要大 10 倍以上。利用现代微型计算机对这些算法进行实时计算是困难的。用最新的 16 位微型计算机(带算术协同处理器或适当的硬件支持),一次乘法和一次加法运算要花费 0.1ms (本书中,将把这样的微型计算机看作是标准微型计算机)。这表明,在 20ms 内只能完成约 200 次浮点乘法和加法运算,远小于上述方法所需的运算次数。

Vukobratović 和 Potkonjak 研究出一种算法,这种算法用递推关系式表示动力学模型的矩阵构造。尽管这种算法的运算次数仅有 Kahn-Uicker 法的几分之一,但因这种算法对标准操作手需要进行 2000 多次浮点运算,所以仍不易实时实现。

由上述讨论可知,基于拉格朗日方程的算法,既能产生动力学正问题也能产生动力学逆问题的解,但由于需要计算机运算次数太多,以致不能用于实时控制。这就是 Waters 和

Hollerbach 研究新算法的原因，他们在 Uicker-Kahn 法的基础上提出了只能求解动力学逆问题的算法。他们的方法允许在给定操作手运动条件下，确定驱动力(转矩)。浮点乘法运算次数简化为 n^2 的函数，甚至只是 n 的函数。节省运算的原因在于，操作手模型的惯性矩阵不是显含计算，也就是关节坐标的二阶导数隐含地计入动力学方程中了。但是，这种算法不能用来求解所有控制问题。此外，尽管不用考虑乘法运算次数是否取决于 n 的函数，但是，这种方法仍难于实时实现。

在 Kane 的论文中可以找到牛顿-欧拉动力学方程在关节联接多刚体系建模中的应用。Kane 导出的动力学方程不只限于机器人系统，而且还适应于适用面稍宽的一类问题，例如包括带有几个关节联接多刚体系的航天器(人造卫星)。Huston 给出了 Kane 法的具体结果，并完成了程序实现。这种算法利用了动力学模型矩阵元素的闭式表示式，因此既能求解动力学的正问题，又能求解动力学的逆问题。然而，和 Kahn-Uicker 法类似，利用这种算法建立模型所需的运算次数仍是很大的。Vukobratović 和 Stepanenko 通过引入递推关系式提出了一种更有效的计算机方法，这种方法既能给出动力学正问题的解，也能给出动力学逆问题的解。可以证明，这种方法所需浮点乘法运算次数与 n^3 成正比。Walker 和 Orin 重新编排了计算步骤，使运算次数减少到只是 n^2 的函数。这样的运算次数仍未能减少到使这种方法满足实时实现的要求。Luh, Walker 和 Paul 处理了牛顿-欧拉法的一种特殊情况，这种情况只能给出动力学逆问题的解。通过只用局部坐标系独立地表示每个连杆的运动方程，他们获得了一种比前述方法更有效的方法。依照这种方法，浮点乘法运算次数简化为只是 n 的函数，于是，对三个自由度的操作手，约

需 400 次运算。尽管这个运算次数比前述方法小得多，但仍差不多是标准微型计算机 20ms 内完成运算次数的两倍。此外，由于这种算法不能给出系统的惯性矩阵，只能给出驱动力和转矩，因此它不能应用于所有控制问题。为实现实时控制，Luh 和 Lin 利用上述算法研究出一种方法，它通过在一台具有几个中央协同处理器部件的计算机上进行平行计算来计算驱动力。我们以由六个协同处理器部件控制的，六个自由度的斯坦福操作手为例，详细说明了这种方法。可以证明，为计算执行机构的驱动力(转矩)，每个协同处理器部件需要进行 20 次浮点乘法和加法运算。

我们看到，通过利用单个协同处理器部件对不超过三个自由度的操作手进行实时控制也是很难实现的。于是可提出如下基本问题，上述方法的浮点运算次数是最少的吗？在回答问题前，让我们先看一看上述所有方法的共同特点。上述方法只适应于动力学建模方法，而与操作手机械设计及刚体系统运动学、动力学的一般规律无关。我们可以断定，对给定操作手，实际的解析模型只能包含相当少的数值运算次数。Aldon、本书作者和 Renaud 给出了这种假定有效性的确切指标。例如，可以证明，构造三个自由度的拟人操作手模型，需要不超过 44 次浮点乘法运算和 23 次浮点加法运算，它比上述任一种方法都少一个数量级以上。对五个自由度的操作手，可以证明，它只需要 352 次乘法运算和同样次数的加法运算。但是，上述方法中还没有一个结果能给出计算机辅助生成适当解析模型问题的解。本书中将详细考虑这个问题。

在第二章中，我们将介绍一种基于闭式动力学方程的，生成机器人动力学模型的面向计算机的新算法。这种算法是非递推的，由于它的紧凑性，并且与物理参数明显相关，因此便

于计算机实现。我们还证明了几个有关线性化和灵敏度的闭式模型构造的定理。

我们利用研究的动力学方程，严格地论证了动力学模型的某些特性。其中几个特性，例如惯性矩阵的对称性和正定性是众所周知的。另外几个特性在论述机器人的书中至今尚未发表，例如，我们要证明动力学矩阵的反对称性，并用它描述哥氏和离心效应。

我们要引入计入机器人动力学方程的变量的一般表示法。这种表示法称为“数值-符号”法，顾名思义，数值-符号法是指对参数进行数值处理，而对关节坐标函数的变量进行符号处理的方法。我们要证明这种表示法的唯一性。还要证明，描述系统动力学的变量可用多项式形式唯一地表示出来。再进一步证明，这些多项式可表示为矩阵形式。正因为这些矩阵描述了多项式结构，我们就把这些矩阵称为“多项式矩阵”。我们将用几个定理来说明这些矩阵的几个基本特性，并做了证明。获得的结果是重要的，因为可用这些结果估计出需要多少计算机存储器来存放各种数值-符号表示式，即存放各种相应的矩阵。

接着，我们研究了多项式矩阵代数。这种代数是以专门引入的多项式为依据，可以证明，这种专门引入的多项式与数值-符号表示式是等价的。后面要证明，机器人系统变量的多项式矩阵的集合构成一个向量空间。为此目的，不但对实常数的加法和乘法等运算下了定义，而且对纯量乘法和向量乘法运算也下了定义。还对相对于给定变量的零元素和逆元素也下了定义。再进一步证明，引入的代数很适合数字计算机的程序实现。

在本书中我们还着重考虑了关于上述多项式的唯一性问

题。可以证明,对给定的模型变量,都存在不定数目的不同的多项式矩阵与之对应,这种非唯一性主要是由确定动力学变量的三角关系式的非唯一性引起的。现在,我们遇到了一个问题,这就是对等价矩阵的集合能否确定一种唯一的表示式?可以证明,这种表示式是存在的,并且它对应最小行数的矩阵。我们研究了一种便于计算机实现的维数最小的有效算法,并详细介绍了几个例子。

用解析形式表示的,机器人的完全非线性动力学模型的生成是基于后面理论上的结果而产生的。利用获得的解析表示式,我们还研究了机器人在数值-符号定义域内的线性化模型。利用多项式的偏导数就可得到线性化模型,我们把这种运算定型为一个定理,可以证明,它容易在计算机上实现。我们还研究了灵敏度模型和近似解析模型。

上述多项式是多变量的。现在,我们说明一个与实现上述理论有关的最重要的问题:怎样用最少的浮点运算次数计算多项式?本书中详细地考虑了这个问题。为计算多项式,我们给出了一种运算优化(最终是次最优)序列的方法。在这种方法的范围内,我们引入了几个概念,例如有序集的次数,有序集之间的包含关系等。可把这种方法分成若干步,用一系列独立的逐次最小量代替全局最小量。在一般情况下,这种方法达不到绝对最小运算次数。另一方面,这种经过简化的算法对实现来说更有吸引力。

上述结果提供了构造“专家程序”的基础,专家程序是用需要的计算机语言生成一种计算机器人解析模型的程序。程序码可在微型计算机上实现,而微型计算机是属于建立模型的机器人控制系统的。

为简述起见,我们仅指出,数值-符号建模不同于前面研

究的数值算法的几个基本思想。

1) 机器人动力学方程的解析结构可在一系列数学优化后预先离线计算。

2) 数值-符号算法不是由各种运动学和动力学规律组成的。相反,它包含很紧凑的解析形式,在数值意义上,它是被优化凝聚了。

最后,我们考虑了机器人机构动力学建模的解析法和数值法这两种方法的进一步发展远景。

就解析法而论,为提高该法的有效性,选择的途径是寻找多项式全局最小量的算法。对六个和更多个自由度的操作机器人来说,这是特别重要的,而六个和更多个自由度的机器人在应用中变得越来越重要。一般认为,对于不多于五个自由度的操作手,为了实际应用和微型计算机实现,使用多项式的逐次最小量法就能给出满意的结果。

用递推方法也能使数值的复杂性进一步简化。本书中介绍了一个例子,并给出了已经得到的结果,在参考文献[68]中,对此做了详细描述。这是一种至今最有效的新的数值计算机算法,对六个或更多个自由度的操作机器人,这种算法比解析法需要更少的数值运算次数,其中解析法指的是以多项式逐次最小量为基础的方法。为建立使浮点乘法/加法运算次数最小的机器人动力学模型,多项式全局优化方案的进一步改进将会提供更有效的解析方法。

对操作机器人进行非自适应,尤其是自适应控制,则是十分有趣和重要的工作。

第一章 机器人计算机辅助建模方法综述

1 1 引 言

我们进一步把建模方法分为数值法和符号法这样两种方法。在过去 10 年间几乎只发展了数值法。在数值法中,数值量与每个变量有关,而每个变量又都包含在模型构造中。另外,在符号法中,变量表示为符号集,用符号描述系统参数和状态(位置和速度)之间的函数关系。在这一章中,我们要详细介绍这两类方法的优缺点。可以证明,因为数值法的复杂性和数值运算次数过多,使得它难于实时实现。反之,也可以证明,为了在线应用,符号法是很有用的方法。遗憾的是,生成符号方程是一个极端复杂的问题。这就是符号法在过去没有得到发展的主要原因。

在本章末尾,我们要阐述一类新方法的基本原理,并在一定条件下称之为数值-符号法,它是数值建模和符号建模这样两种方法的特殊“混杂”。我们将指出,这类方法的研究已经产生了非常有吸引力的结果,尤其是在机器人动力学模型的实时应用领域更是如此。

1.2 基于拉格朗日方程的方法

在这一节中,我们扼要介绍基于二阶拉格朗日方程的机器人系统的建模方法。二阶拉格朗日方程具有形式:

$$\frac{d}{dt} \left(\frac{\partial E_k}{\partial \dot{q}_i} \right) - \frac{\partial E_k}{\partial q_i} + \frac{\partial E_p}{\partial q_i} = P_i, \quad i = 1, \dots, n \quad (1.2.1)$$

式中

n ——系统的自由度数;

E_k ——动能;

E_p ——势能;

P_i ——广义力;

• • •

q_i ——对应第 i 个自由度的广义关节坐标。

动能和势能由对应的机构连杆能量求和得到。在所有介绍的方法中,机器人的机构指的是一种开运动链,它由 n 个关节联接的 n 个刚体组成。广义坐标 q_i 表示第 i 个关节的内坐标。

尽管本节描述的所有方法都是来自拉格朗日方程(1.2.1)的结果,但在下面将会看到,这些方法中的每一种都有若干独特的特点。

1.2.1 Uicker-Kahn 法

Uicker-Kahn 法是属于第一类构造机器人动力学模型方法中的一种。它的原始形式是 J.J. Uicker^[1-3] 在 1965 年研究出来的。由于这种方法主要涉及某类闭环机构,1969 年, M. Kahn^[4,5] 又研究出一种建模开运动链的算法。1971 年 L. Woo 和 F. Freudenstein^[6], A. Yang^[7] 通过引入“螺旋”向量^[6,7] 修改并重新阐述了这种方法。1981 年, N. Orleandea 和 T. Berenui^[10] 为分析机器人动力学,以程序包形式完成了这种方法的程序实现。S. Mahil^[11,12], M. Renaud^[13], M. Tomas 和 D. Tesar^[14], R. Waters^[15], J. Hollerbach^[16,17] 等人都在减少数值运算次数的意义上提出了改进方案。

利用 Uicker-Kahn 法,我们既可求解动力学的正问题,也能求解动力学的逆问题。此外,用这种方法能够计算机器人动力学模型的所有矩阵:惯性矩阵、科里奥利和离心效应矩阵,以及重力向量。利用这种方法导出的, n 个自由度的操作手的动力学方程具有如下形式^[1-5]:

$$P_i = \sum_{j=1}^n \left\{ \sum_{k=1}^j \left[\text{tr} \left(\frac{\partial W_j}{\partial q_i} J_j \frac{\partial W_j}{\partial q_k} \right) \right] \ddot{q}_k \right.$$

$$\begin{aligned}
& + \sum_{k=1}^i \sum_{l=1}^i \left[\text{tr} \left(\frac{\partial W_i}{\partial q_i} J_i \frac{\partial^2 W_i^T}{\partial q_k \partial q_l} \right) \dot{q}_k \dot{q}_l \right] \\
& - m_i \mathbf{g}^T \frac{\partial W_i}{\partial q_i} \tilde{\mathbf{r}}_{i0} \} \quad (1.2.2)
\end{aligned}$$

式中

P_i ——作用在第 i 个关节上的驱动力，如果该关节是滑动关节；或是作用在第 i 个关节上的驱动转矩，如果该关节是旋转关节；

W_i ——第 i 个局部坐标系和参考坐标系之间的转换矩阵；

J_i ——第 i 个连杆对局部坐标系的惯性矩阵；

m_i ——连杆 i 的质量；

$\tilde{\mathbf{r}}_{i0}$ ——用第 i 个连杆的局部坐标系表示的，连杆 i 的质量中心和参考坐标系的原点之间的距离向量；

\mathbf{g} ——重力向量。

矩阵 W_i 可表示为

$$W_i = A_1^0 A_2^1 \cdots A_i^{i-1} \quad (1.2.3)$$

式中 A_k^{k-1} 是一个由属于连杆 k 的坐标系 $o_k x_k y_k z_k$ 到属于连杆 $k-1$ 的坐标系 $o_{k-1} x_{k-1} y_{k-1} z_{k-1}$ 的 4×4 转换矩阵。设

$$\mathbf{v}^k = [1 \ x^k \ y^k \ z^k]$$

是一个表示在连杆 k 的坐标系上的任意向量。对相邻的坐标系，必成立

$$\mathbf{v}^{k-1} = A_k^{k-1} \mathbf{v}^k$$

由矩阵 W_i 的定义式 (1.2.3) 可知，这个矩阵确实表示了由第 i 个坐标系到参考坐标系 $oxyz$ 的转换矩阵。我们看到，Uicker 采用了 4×4 转换矩阵。因此，连杆之间的旋转和平移都能用唯一的方法描述。这就为协调处理包括旋转和

滑动关节的连杆机构提供了保证。Uicker 采用了 Denavit-Hartenberg 符号表示这些矩阵^[18],这是文献中共用的符号(例如,参阅参考文献 [12,13,17])。按照这种符号,转换矩阵 A_k^{k-1} 有如下形式:

$$A_k^{k-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ a_k \cos q_k & \cos q_k & -\sin q_k \cos \alpha_k & \sin q_k \sin \alpha_k \\ a_k \sin q_k & \sin q_k & \cos q_k \cos \alpha_k & -\cos q_k \sin \alpha_k \\ s_k & 0 & \sin \alpha_k & \cos \alpha_k \end{bmatrix} \quad (1.2.4)$$

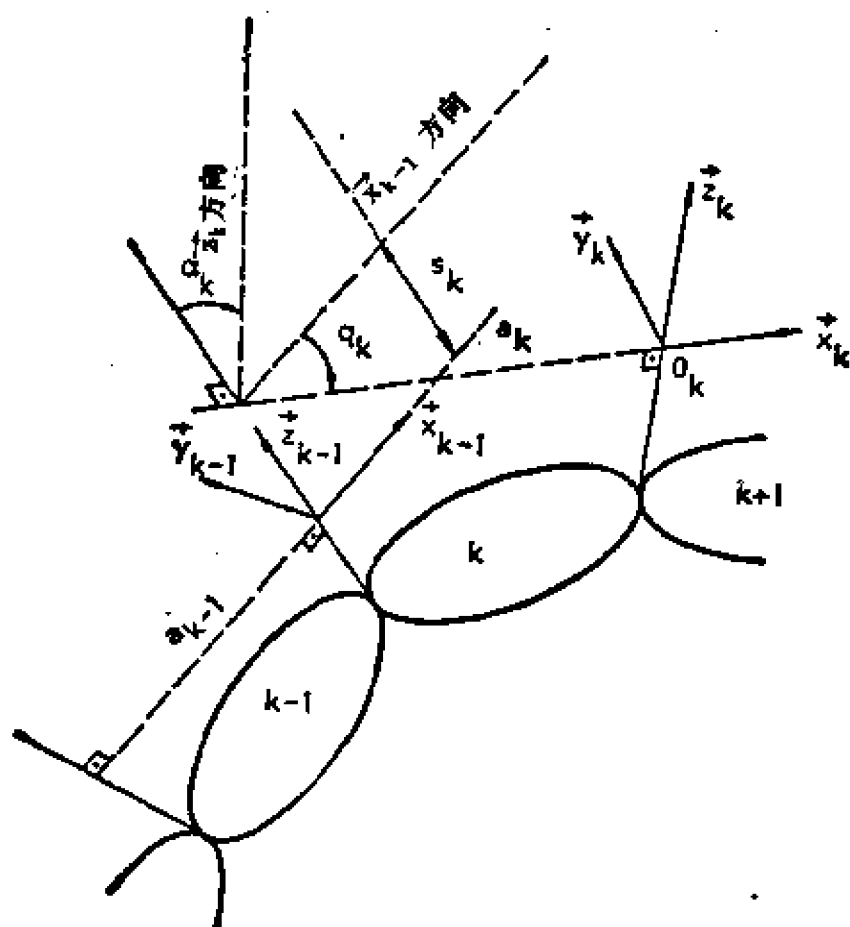


图 1.1 Denavit-Hartenberg 的符号

式中

a_k ——沿 x_k 轴量测的, 连杆 $k-1$ 和连杆 k 的局部坐标系原点之间的距离(图 1.1);

s_k ——沿 z_{k-1} 轴量测的, z_{k-1} 轴和 x_k 轴之间的距离;

α_k ——用右手系表示的沿 x_k 轴量测的, z_{k-1} 轴和 x_k 轴之间的夹角;

q_k ——用右手系表示的沿 z_{k-1} 轴量测的, z_{k-1} 轴和 x_k 轴之间的夹角, 夹角 q_k 也称为第 k 个关节的“关节坐标”, 如果关节是旋转关节, 当关节是滑动关节时, 则用长度 s_k 表示关节坐标。

连杆 k 的局部坐标系的轴用单位向量 x_k, y_k 和 z_k 定义, 其中

z_k ——表示连杆 k 和 $k+1$ 之间的关节轴的方向, 按定义, 这个关节看作是关节 $k+1$;

x_k ——沿 z_{k-1} 轴和 z_k 轴的垂直线, 并指向 z_k 的一个向量;

y_k ——构成右手向量集 (x_k, y_k, z_k) 的一个单位向量。

可以看出, 坐标系的选择只取决于机构的几何参数。从而, 局部坐标系的轴并不与主惯性轴相吻合, 因此, 我们必须用惯性张量(9项), 而不用3个惯性矩。Uicker 采用了 4×4 维矩阵 J_i ,^[(1-15), (12-17)] 以便使 J_i 既包含惯性张量, 又包含连杆质量^[(1-5), (12-17)]。

$$J_i = \begin{bmatrix} m_i & m_i x_{i0} & m_i y_{i0} & m_i z_{i0} \\ m_i x_{i0} & \frac{1}{2}(-I_{xx} + I_{yy} + I_{zz}) & I_{xy} & I_{xz} \\ m_i y_{i0} & I_{xy} & \frac{1}{2}(I_{xx} - I_{yy} + I_{zz}) & I_{yz} \\ m_i z_{i0} & I_{xz} & I_{yz} & \frac{1}{2}(I_{xx} + I_{yy} - I_{zz}) \end{bmatrix}$$

式中

x_{i0}, y_{i0} 和 z_{i0} —— 连杆 i 的质量中心对参考坐标系的坐标;

I_{xx}, I_{yy}, \dots —— 有关连杆 x_i, y_i 和 z_i 各轴的惯性矩。惯性矩阵 J_i 描述了连杆 i 的质量分布, 同时它又是其质量中心位置的函数, 其中质量中心位置的坐标是对固定坐标系的坐标。

由转换矩阵 A_i^{-1} 可知, 可以解析地确定它对 q_k 的偏导数。对矩阵 W_i , 也有同样的情况。由于这种方法没有使用递推关系式, 因此它属于一类非递推方法。现在让我们更详细地阐述如何求解动力学的正和逆问题。为此目的, 让我们注意显含地计入模型式(1.2.2)中的关节加速度 \ddot{q}_k 和关节速度 \dot{q}_k 。因而, 式(1.2.2)可写为

$$P_i = \sum_{j=1}^n H_{ij}(q) \ddot{q}_j + \sum_{k=1}^n \sum_{l=1}^n C_{kij}^l(q) \dot{q}_k \dot{q}_l + g_i(q) \quad (1.2.5)$$

式中 H_{ij} , C_{kij}^l 和 g_i 是关节坐标向量 $q = [q_1 \cdots q_n]^T$ 的标量函数。为证明这一点, 让我们注意, 按照式(1.2.2), 这些函数是矩阵 W_i 的偏导数的函数。因为矩阵 W_i 只是关节坐标的函数, 那么可以断定, W_i 的偏导数也只是关节坐标的函数。模型式(1.2.5)可以被表示为矩阵形式:

$$P = H(q) \ddot{q} + \dot{q}^T C(q) \dot{q} + g(q) \quad (1.2.6)$$
$$P = [P_1 \cdots P_n]^T$$

式中

P —— 驱动力(转矩)向量;

$H(q)$ —— $n \times n$ 矩阵, 称为系统的惯性矩阵;

$C(q)$ —— $n \times n \times n$ 矩阵, 称为哥氏和离心效应矩阵, 或简称为“C-阵”;

$g(q)$ ——由重力作用而产生的 n 维向量，因而称为重力效应向量。

在模型式(1.2.6)中， $\dot{q}^T C(q) \dot{q}$ 表示 n 维向量：

$$\dot{q}^T C(q) \dot{q} = \begin{bmatrix} \dot{q}^T C^1(q) \dot{q} \\ \vdots \\ \dot{q}^T C^n(q) \dot{q} \end{bmatrix}$$

式中 $C^i(q)$ 是 $n \times n$ 矩阵，它的元素对应式(1.2.5)中的 $C_{ij}^i(q)$ 。通常，可把模型式(1.2.6)写为

$$P = H(q)\ddot{q} + h(q, \dot{q}) \quad (1.2.7)$$

式中

$$h(q, \dot{q}) = \dot{q}^T C(q) \dot{q} + g(q)$$

现在我们能直接求解动力学的逆问题，也就是，在已知运动(q, \dot{q}, \ddot{q})的基础上，确定驱动力和转矩 (P)。为求解动力学的正问题，必须在已知 $P(t), t \geq t_0$ ，初始条件 $q(t_0)$ 和 $\dot{q}(t_0)$ 等条件下，确定 $q(t), t \geq t_0$ 。很明显，用模型式(1.2.7)可得出问题的解，这是因为向量 \ddot{q} 显含地计入式(1.2.7)中了，而惯性矩阵 $H(q)$ 也有逆矩阵(第二章将证明这个性质)。因此，我们得到

$$\ddot{q} = H(q)^{-1}[P - h(q, \dot{q})] \quad (1.2.8)$$

利用任何一种积分方法都可求出 $q(t), t \geq t_0$ 。

最后，让我们考虑通过 Uicker-Kahn 法建立机器人动力学模型所需的数值运算次数。我们立即注意到，数值运算是指实数(浮点数)的乘法和加法(减法)运算。和后面将要介绍的其它方法一样，在 Uicker-Kahn 法中也出现其它的运算，例如，余弦和正弦函数。然而，这些函数不管采用什么方法都必须计算出来。这就是我们以乘法和加法运算次数为依据来比较各种方法的原因，其中加法运算是指实数的加法或减法运算。计入程序实现中的其它运算不在考虑之列(例如，把存

数由一个储存地址转到另一个储存地址, 逻辑函数, 程序调入, 等等), 因为这需要很复杂的分析方法, 另外, 这些运算主要取决于所用程序语言的性质。

在参考文献[17]中指出, 通过 Uicker-Kahn 法建立动力学模型所需的乘法和加法运算次数是 n^3 的函数。于是, 对三个自由度的操作手, 总计需要 5000 多次乘法运算和大致相同的加法运算。而对六个自由度的操作手, 运算次数要增加 10 多倍。这样, 就提出了一个问题, 在现代微型计算机上实时实现的可能性是怎样的? 在文献中一般认为, 如果至少在每 20ms, 即对应 50Hz 内能完成该模型的计算, 那么, 一个模型则被认为是可实时实现的。正如绪论中所述, 一种算法能否实时实现的问题, 与处理该算法的标准计算机密切相关。进一步可知, 利用 16 位现代微型计算机 (具有算术协同处理器或适当的硬件支持) 完成一次乘法和一次加法运算约需要 0.1ms。用这一结果就可估计出为建立模型式(1.2.2)所需的时间。对三个自由度的操作手需要的时间大于 500ms, 而对六个自由度的操作手需时超过 5s。可以看出, 对三个自由度的操作手, 所需时间大于允许时间 (20ms) 的 25 倍, 而对六个自由度的操作手, 则大于 250 倍。这样就提出了一系列改进基本方法的方案, 其目的是减少运算次数。在这一节中要考虑这些算法的基本思想和数值的有效性, 但不给出详细描述。

1.2.2 S. Mahil, S. Megahed 和 M. Renaud 算法

由上述可知, Uicker-Kahn 法中包含了二阶偏导数。在程序有效实时实现的意义上说, 这是 Uicker-Kahn 法的主要缺点。这就是 S. Mahil^[11,12], S. Megahed 和 M. Renaud^[19]重新研究 Uicker-Kahn 法的原因, 其目的是设法取消偏导数

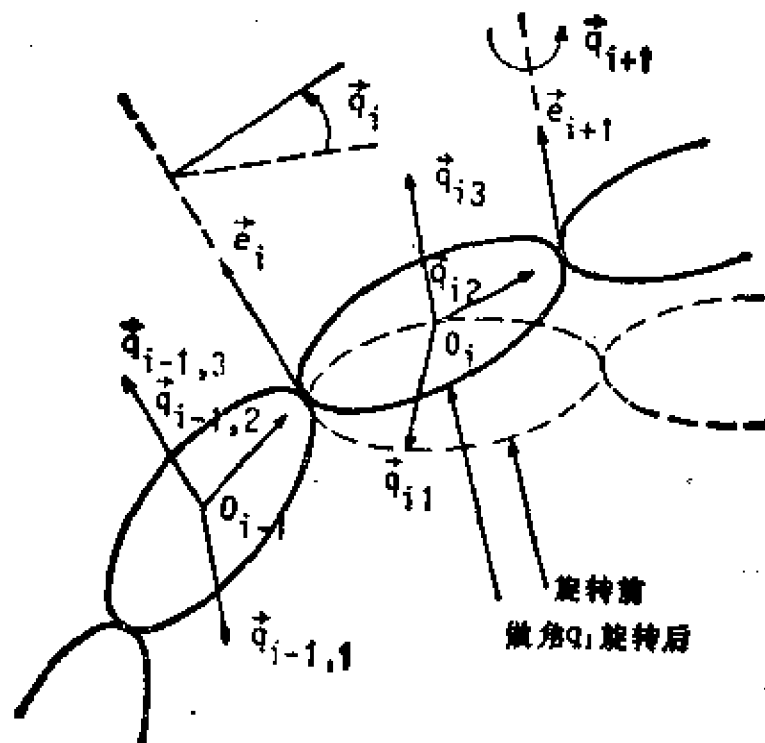


图 1.2 关节 i 的旋转

和减少数值运算的次数。

1979年, S. Mahil 提出用 Rodrigues 有限旋转公式代替 Denavit-Hartenberg 矩阵。这个公式首先于 1973 年在牛顿-欧拉法中使用^[20-23]。在参考文献[23]中详细地描述了 Denavit-Hartenberg 矩阵和 Rodrigues 公式之间的变换关系。虽然在第二章中要详细介绍 Rodrigues 公式,但由于它的重要性,在这里仍做一简要的描述。让我们考虑,第 i 个连杆对第 $i-1$ 个连杆做夹角为 q_i 的旋转(图 1.2)。用 r_i 表示做角 q_i 旋转前,属于第 i 个连杆的一个任意向量。旋转后,按照 Rodrigues 公式,这个向量可表示为

$$r'_i = r_i \cos q_i + (1 - \cos q_i)(e_i \cdot r_i)e_i + e_i \times r_i \sin q_i \quad (1.2.9)$$

式中 e_i 是第 i 个关节轴的单位向量。现在我们利用这个公式确定转换矩阵 A_i^{-1} ，用它描述第 i 个坐标系对第 $i-1$ 个坐标系的旋转。用 q_{ij} , ($j=1,2,3$) 表示第 i 个坐标系的轴对第 $i-1$ 个坐标系的单位向量。那么矩阵 A_i^{-1} 给定为

$$A_i^{-1} = [q_{i1} q_{i2} q_{i3}] \quad (1.2.10)$$

为确定向量 q_{ij} ，我们要利用 Rodrigues 公式

$$q_{ij} = q_{ij}^0 \cos q_i + (1 - \cos q_i)(e_i \cdot q_{ij}^0)e_i + e_i \times q_{ij}^0 \sin q_i \quad (1.2.11)$$

式中 q_{ij}^0 是第 i 个坐标系在旋转前 j 轴的向量。因为对第 $i-1$ 个坐标系的向量 e_i 和 q_{ij}^0 是已知的，所以可用公式 (1.2.10) 直接确定转换矩阵 A_i^{-1} 。

与 Denavit-Hartenberg 运动学大不相同，我们把 Rodrigues 公式用到局部坐标系，不管以什么方式局部坐标系总是属于连杆的。为方便起见我们规定，坐标原点与连杆质量中心吻合，而坐标轴与主惯性轴一致。因为用主惯性转矩代替了惯性张量，所以使动力学方程大大简化了。

让我们再回到 S. Mahil 算法。在这种算法中，包含的惯性矩不多于两个，即纵向和横向惯性矩。然而，这种近似只适用于柱形连杆，这种柱形连杆，其长度远远大于它自身的直径。遗憾的是，这样近似常常是不合理的。

从拉格朗日方程出发，S. Mahil 导出了用矩阵形式表示的动力学方程

$$P = \sum_{i=1}^n (H^i(q)\ddot{q} + D^i(q, \dot{q})\dot{q} + q^i(q)) \quad (1.2.12)$$

式中

$P \in R^{(n)}$ ——驱动力(转矩)向量；

1) $(\cdot) \in R^n$ 是指 (\cdot) 是一个 n 维向量，它的元素是实数。

$H^i(q)$ 和 $D^i(q, \dot{q})$ —— $n \times n$ 矩阵;

$g^i(q)$ —— n 维向量。

式(1.2.12)中矩阵元素可用关节轴向量、局部坐标系和距离向量表示出来。这些向量很容易用 Rodrigues 公式求得。为说明这种方法,我们要介绍若干个模型矩阵元素的特征式。例如,在旋转关节情况下,矩阵 $H^i(q)$ 的第 (j, k) 个元素可表示为

$$\begin{aligned} H_{jk}^i = & [Jn_i + m_i(r_{ji} \cdot r_{ki})](e_j \cdot e_k) \\ & + Js_i(e_j \cdot q_{ii})(e_k \cdot q_{ii}) \\ & - m_i(e_j \cdot r_{ki})(e_k \cdot r_{ji}), j \leq k \leq i \end{aligned} \quad (1.2.13)$$

式中

Jn_i —— 第 i 个连杆的横向惯性矩;

Js_i —— 第 i 个连杆的纵向惯性矩;

m_i —— 第 i 个连杆的质量;

r_{ji} —— 用惯性系表示的,第 j 个关节到第 i 个连杆质量中心的距离向量。

向量 e_i 和 q_{ii} 也用惯性系表示。

矩阵 $D^i(q, \dot{q})$ 的元素可表示为

$$D_{kl}^i = \sum_{j=r+1}^i \frac{\partial H_{kj}^i}{\partial q_j} \dot{q}_j - \frac{1}{2} \sum_{j=1}^q \frac{\partial H_{ij}^i}{\partial q_k} \dot{q}_j \quad (1.2.14)$$

式中 p 和 q 是 k 和 l 的函数^[2]。这个表示式包含用闭式表示的矩阵项的偏导数¹⁾。由于这些表示式相当复杂,并且是标记 i, k, l 和 j 之间关系的函数,为举例说明,在这里,我们只在 $k < j < l \leq i$ 条件下,给出一个用于旋转关节的表示式:

1) 在这里项的闭式是指一种表示式,它的值不能通过一个或多个递推关系式得到,而是通过直接代入自变量的适当数值计算得到。

$$\begin{aligned}
\frac{\partial H'_{ki}}{\partial q_j} = & (J_{n_i} + m_i r_{ki} \cdot r_{li})(e_k \cdot e_j \times e_l) \\
& + m_i [e_k \cdot e_l (R_{ki} \cdot e_j \times r_{li}) \\
& - e_l \cdot r_{ki} (R_{ki} \cdot e_j \times e_l) \\
& - e_k \cdot r_{li} (R_{ki} \cdot e_j \times e_l)] \\
& + J_{s_i} e_l \cdot q_{ik} (e_k \cdot e_j \times q_{il}) \quad (1.2.15)
\end{aligned}$$

式中 R_{ki} 是连杆 i 质量中心到连杆 k 质量中心的距离向量。

表示式(1.2.14)建立了哥氏和离心效应矩阵元素与惯性矩阵元素偏导数之间的关系式。然而，这个表示式在一定程度上是相当复杂和“不适用的”。在第二章中，我们要导出这个表示式的更精巧的形式。我们还要导出任意形式连杆的表示式(1.2.13)和(1.2.15)，这就是在这里不详细论述这些表示式的原因。

由上述可知，S. Mahil 法属于一类基于拉格朗日方程的非递推法。通过与 Uicker-Kahn 法比较可知，这两种方法的数值运算次数具有相同的数量级。因此可以断定，尽管对这种方法进行近似计算，实际上仍不能使它实时实现。这种算法的主要特点是，可直接透彻地了解计入动力学模型矩阵元素中的物理参数。

1982年，Thomas 和 D. Tesar^[10] 提出了一种算法，其基本思想类似于前述算法。这种算法也是从拉格朗日方程出发，导出动力学模型矩阵元素的闭式表示式。其主要区别在于，这种算法用“旋转矩阵”代替了 Rodrigues 公式。旋转矩阵用与 Denavit-Hartenberg 符号对应的局部坐标系说明(图 1.1)。旋转矩阵表示从局部坐标系到另一坐标系的转换矩阵，这个坐标系其原点与局部坐标系原点相同，但坐标轴与参考坐标系的坐标轴平行。与 4×4 Denavit-Hartenberg 矩阵不

同, 旋转矩阵是 3×3 维的。M. Thomas 和 D. Tesar 已经证明, 第 k 个连杆的旋转矩阵可表示为

$$A_k = [\alpha_k^0 | z_{k-1} \times \alpha_k^0 | z_{k-1}] \quad (1.2.16)$$

式中 α_k^0 是一个与 z_{k-1} 和 z_k 轴垂直, 且指向 z_k 的单位向量(图 1.1)。

从这种方法定义的运动学的概念出发, 作者导出了一系列分析机器人动力学的矩阵表示式。但是, 作者没有分析这种算法对 Uicker-Kahn 法的优点。

1981 和 1982 年, S. Megahed 和 M. Renaud^[19] 提出了一种有效算法, 用来计算 Uicker-Kahn 法中的偏导数矩阵。这种算法是建立在 Denavit-Hartenberg 矩阵的如下性质基础上的:

$$\frac{\partial A_k^{k-1}}{\partial q_k} = Q A_k^{k-1} \quad (1.2.17)$$

上式中如果关节 k 是旋转关节, 则

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

如果关节是滑动关节, 则

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

这一性质容易通过对式(1.2.4)取导数加以验证。在这一关系式的基础上, 可以证明, 下列各式成立:

$$\frac{\partial W_j}{\partial q_i} = \left[\begin{array}{c} 0 \\ \vdots \\ \Omega_i \end{array} \right] W_j, \quad i \leq j$$

$$\frac{\partial W_j}{\partial q_i \partial q_k} = \left[\begin{array}{c} 0 \\ \vdots \\ \Omega_i \end{array} \right] \left[\begin{array}{c} 0 \\ \vdots \\ \Omega_k \end{array} \right] W_j, \quad i \leq k \leq j$$
(1.2.18)

式中 Ω_i 是 3×3 矩阵, 它的元素直接从矩阵 W_j 得到^[19].

从上述讨论可知, S. Megahed 和 M. Renaud 算法属于一类基于拉格朗日方程的非递推算法, 但是, 由于数值运算次数仍然很多, 因此这种算法仍不适合实时应用.

最后, 让我们说明, M. Renaud 在 1983 年提出了另一种改进方法, 和一系列计算动力学模型矩阵元素的递推关系式一样, 他通过引入张量计算, 改进了前述方法^[2]. 但是, 这一改进方法并没有导致大大减少数值运算次数. 然而, 作者在他的论文中已经证明, 用改进方法得到的解析模型在数值意义上是很有用的. 遗憾的是, 这种建立解析模型的方法不能自动进行, 因此不能在计算机上实现, 而只能用“手算”完成. 这种方法是冗长乏味和复杂的, 并且误差几乎是不可避免的. 所以, 这种方法不能看作是面向计算机的, 而且不能与其它面向计算机的算法进行比较.

1.2.3 R. Waters 和 J. Hollerbach 算法

1979 和 1980 年, R. Waters^[45] 和 J. Hollerbach^[46, 47] 研究了求解动力学逆问题的算法, 这种算法是 Uicker-Kahn 法的一种特殊情况. 这种算法是在已知运动的基础上用来确定驱动力(转矩)的. 由于这种算法隐含地包含了关节坐标的二阶导数, 因此不能用它计算惯性系统矩阵. 这就是这种算法与 Uicker-Kahn 法不能等价的原因, 并且只有考虑动力学的

逆问题时,这种算法的性能才是可比较的。

按照 R. Waters 算法,驱动力和转矩可表示为

$$P_i = \sum_{j=1}^i \left[\text{tr} \left(\frac{\partial W_j}{\partial q_i} J_j \dot{W}_j^T \right) - m_j \mathbf{g}^T \frac{\partial W_j}{\partial q_i} \tilde{\mathbf{r}}_{j0} \right] \quad (1.2.19)$$

对 W_i , \dot{W}_i 和 \ddot{W}_i 有递推关系式

$$\begin{aligned} W_i &= W_{i-1} A_i^{-1} \\ \dot{W}_i &= \dot{W}_{i-1} A_i^{-1} + W_{i-1} \frac{\partial A_i^{-1}}{\partial q_i} \dot{q}_i \quad (1.2.20) \\ \ddot{W}_i &= \ddot{W}_{i-1} A_i^{-1} + 2\dot{W}_{i-1} \frac{\partial A_i^{-1}}{\partial q_i} \dot{q}_i \\ &\quad + W_{i-1} \frac{\partial^2 A_i^{-1}}{\partial q_i^2} \dot{q}_i^2 + W_{i-1} \frac{\partial A_i^{-1}}{\partial q_i} \ddot{q}_i \end{aligned}$$

$$\begin{aligned} D_i &= J_i \ddot{W}_i^T + A_{i+1}^T D_{i+1} \\ C_i &= m_i \tilde{r}_{i0} + A_{i+1}^T C_{i+1} \end{aligned} \quad (1.2.23)$$

像在 Waters 算法中一样,通过前向递推(由 $i = 1$ 到 $i = n$) 计算加速度 \ddot{W}_i^T 。通过后向递推(由 $i = n$ 到 $i = 1$) 计算 D_i 和 C_i 。于是,可以得到,乘法运算次数是

$$n_M = 830n - 592$$

而加法运算次数是

$$n_A = 675n - 464$$

Hollerbach 还断定,通过用 3×3 矩阵代替 4×4 矩阵可进一步减少运算次数。正如我们所看到的那样,由于用 3×3 矩阵就能描述旋转,用位置向量就能描述移动,那么用 4×4 矩阵同时描述旋转和移动时就引入了相当多余的计算。因为这种改进算法没有从根本上导出比上述算法更新的结果,所以我们只给出确定驱动力(转矩)所需的运算次数

$$n_M = 412n - 277$$

$$n_A = 320n - 201$$

但是,可以证明,牛顿-欧拉法可给出更少的运算次数。

1.2.4 Vukobratović-Potkonjak 递推法

在 1.2.1 和 1.2.2 两小节中,我们已经介绍,基于拉格朗日方程的方法既能求解动力学的正问题,也能求解动力学的逆问题。这些方法都使用闭式表示式,而没有用递推关系式,于是,产生了一个问题,为减少运算次数,能否通过递推关系式获得动力学模型矩阵。1979 年, Vukobratović-Potkonjak 解决了这个问题,已在参考文献 [25] 中做了论述。在参考文献 [26] 中也详细地介绍了这种方法。

这种方法采用 Rodrigues 公式描述连杆的空间布局,由

于在 1.2.2 节中曾经论述过这种方法,在此我们只强调这样的事实,因为用主惯性矩代替了惯性张量,所以这种方法更便于机械机构的动力学分析。

我们可把这种方法的运动学部分归纳为下列两个基本关系式:

$$\begin{aligned}\omega_i &= \tilde{N}(i)\dot{q} \\ \tilde{v}_i &= \tilde{M}(i)\dot{q}\end{aligned}\quad (1.2.24)$$

式中

ω_i ——在第 i 个局部坐标系中表示的第 i 个连杆的角速度向量;

\tilde{v}_i ——第 i 个连杆质心对第 i 个局部坐标系的线速度向量;

$\tilde{N}(i), \tilde{M}(i)$ ——通过递推关系式建立的,取决于关节坐标的 $3 \times n$ 矩阵^[25];

\dot{q} —— $n \times 1$ 关节速度向量。

把式(1.2.24)代入整个系统的动能表示式

$$E_k = \sum_{i=1}^n \left(\frac{1}{2} m_i \tilde{v}_i^T \tilde{v}_i + \frac{1}{2} \omega_i^T \tilde{J}_i \omega_i \right) \quad (1.2.25)$$

并利用拉格朗日方程(1.2.1)的矩阵形式

$$\frac{d}{dt} \left(\frac{\partial E_k}{\partial \dot{q}} \right) - \frac{\partial E_k}{\partial q} + \frac{\partial E_p}{\partial q} = P \quad (1.2.26)$$

则可获得用式(1.2.7)形式表示的机器人动力学模型

$$P = H(q)\ddot{q} + h(q, \dot{q})$$

在上列表示式中 m_i 是第 i 个连杆的质量; \tilde{J}_i 为 3×3 对角线矩阵,它的元素是主惯性矩; E_k 和 E_p 为整个系统的动能和势能; P 为驱动力(转矩)向量。

模型矩阵给定为

$$H(q) = \sum_{i=1}^n (m_i \tilde{M}(i)^T \tilde{M}(i) + \tilde{N}(i)^T \tilde{J}_i \tilde{N}(i)) \quad (1.2.27)$$

$$h(q, \dot{q}) = \frac{\partial (E_T - E_P)}{\partial \dot{q}} + \dot{H}(q) \dot{q} \quad (1.2.28)$$

像惯性矩阵 $H(q)$ 的时间导数一样, 后一个关系式的偏导数也可通过递推关系式计算^[25], 因为递推关系式很复杂, 这里不再介绍。

应用这种方法达到的数值运算次数比 Uicker-Kahn 法少几倍。但是, 这样的运算次数仍大于牛顿-欧拉法。

1.3 基于牛顿-欧拉方程的方法

在这一节中, 我们要介绍几种基于牛顿-欧拉方程的机器人建模算法。和拉格朗日法一起, 这些算法都是用于机器人建模的最一般的算法。

牛顿-欧拉方程可用来确定作用在连杆上的惯性力和惯性力矩。如果用 \tilde{F}_i 表示对第 i 个连杆局部坐标系的, 加在第 i 个连杆质量中心上的惯性力, 用 m_i 表示第 i 个连杆的质量, 则按牛顿定律, 得到

$$\tilde{F}_i = m_i \tilde{\omega}_i \quad (1.3.1)$$

式中 $\tilde{\omega}_i$ 是第 i 个连杆质量中心对第 i 个连杆坐标系的加速度。欧拉动力学方程可用来确定加在第 i 个连杆质量中心上的惯性力转矩

$$\tilde{M}_i = \tilde{J}_i \tilde{\varepsilon}_i + \tilde{\omega}_i \times (\tilde{J}_i \tilde{\omega}_i) \quad (1.3.2)$$

式中

\tilde{J}_i —— 3×3 矩阵, 它的对角线元素是主惯性矩;

$\tilde{\varepsilon}_i$ —— 第 i 个连杆的角加速度;

$\dot{\omega}_i$ ——第 i 个连杆的角速度。

在式(1.3.2)中的所有量都是用第 i 个连杆的局部坐标系表示的, 并且都是对质量中心而言的。局部坐标系的坐标轴与主惯性轴吻合。

当用固定的惯性系表示时, 牛顿-欧拉方程也有像式(1.3.1)和(1.3.2)相同的形式, 不过要用 $J_i - A_i \tilde{J}_i A_i^T$ 代替 \tilde{J}_i 。 A_i 是第 i 个连杆局部坐标系到固定坐标系的转换矩阵, J_i 也是 3×3 矩阵, 它表示第 i 个连杆对参考坐标系的惯性矩阵。

下面我们不仅介绍对局部坐标系导出的方法, 而且介绍对固定的惯性坐标系导出的牛顿-欧拉法。

1.3.1 Vukobratović-Stepanenko 法

1973年, Vukobratović-Stepanenko 首先把牛顿-欧拉动力学方程应用于可动机构的建模上, 见参考文献[20]。在这篇文章中, 还介绍了既能求解动力学正问题, 又能求解动力学逆问题的递推性概念。这种方法也称为动静平衡(Kinestatic)法, 它基于 D'Alembert 原理的方法或一般力学定理法^[21,22]。在参考文献[27—29]中详细地介绍了一种基于这种方法的, 更有效和面向计算机的方法。R. Huston 和 F. Kelly^[30]通过引入欧拉参数, 并且基本上以牛顿-欧拉动力学方程和 Kane 算法为基础^[31,32], 导出了动力学建模的非递推法。J. Luh, M. Walker 和 R. Paul 搞出了为给出逆问题的有效解(就是给定操作手运动, 确定驱动力)的另一种牛顿-欧拉法的变型^[33]。A. Leskov 和 B. Medvedov 在“分块矩阵”法中也利用了牛顿-欧拉方程^[34]。在 E. P. Popov 及其同事的专著中介绍了这种方法^[35]。由于在本丛书第一册中详

细描述了相同的方法,因此在这里不再重复。牛顿-欧拉法是研究计算开运动链线性化动力学方程算法^[36-38]和研究计算灵敏度方程算法^[39,40]的基础。

像基于拉格朗日方程的方法一样,牛顿-欧拉法既利用 Rodrigues 公式^[20-22],又利用 Denavit-Hartenberg 矩阵^[33]。在参考文献[23]中第一次介绍了上述两种方法的相似之处,并且说明了这两种方法很适合研究机器人运动学和动力学。换句话说,用 Denavit-Hartenberg 运动符号时,在一般情况下,连杆坐标系的轴与连杆主惯性轴不是吻合的,因此必须进行惯性张量的运算。反之,在 Rodrigues 公式中,局部坐标系的轴与主惯性轴的方向是吻合的,因此在动力学方程中,只用三个主惯性矩就可代替惯性张量。下面我们就从 Rodrigues 公式出发,简要介绍牛顿-欧拉法的基本关系式(第二章将详细描述这种方法)。

假设已给定关节坐标集 q_1, \dots, q_n , 通过应用 Rodrigues 公式,对每个 $i = 1, \dots, n$, 描述连杆空间布局的下列向量已经确定(图 1.3):

$q_{ij}(j = 1, 2, 3)$ ——第 i 个连杆局部坐标系的第 j 个轴的单位向量;

r_{ii} ——第 i 个关节到第 i 个连杆质量中心的距离向量;

$r_{i,i+1}$ ——第 $i + 1$ 个关节到第 i 个连杆质量中心的距离向量;

e_i ——第 i 个关节轴的单位向量。

那么角速度 ω_i , 角加速度 ϵ_i 和线加速度 w_i 用下列方程表示:

$$\omega_i = \omega_{i-1} + \dot{q}_i e_i$$

$$\epsilon_i = \sum_{j=1}^i a_{ij} \ddot{q}_j + a_j^0$$

$$\omega_i = \sum_{j=1}^i \beta_{ij} \dot{q}_j + \beta_i^0 \quad (1.3.3)$$

式中系数 α_{ij}, α_i^0 和 β_i^0 由对应的递推关系式得到^[20-22]。例如,系数 α_i^0 可由关系式

$$\alpha_i^0 = \alpha_{i-1}^0 + \dot{q}_i \omega_{i-1} \times e_i \quad (1.3.4)$$

得到。第二章将推导这些关系式。让我们注意,这些关系式中包括了这种方法两个基本特性:

- 1) 递推性;
- 2) 关节加速度 \dot{q}_i 显含地表示在这些方程中,因此可提

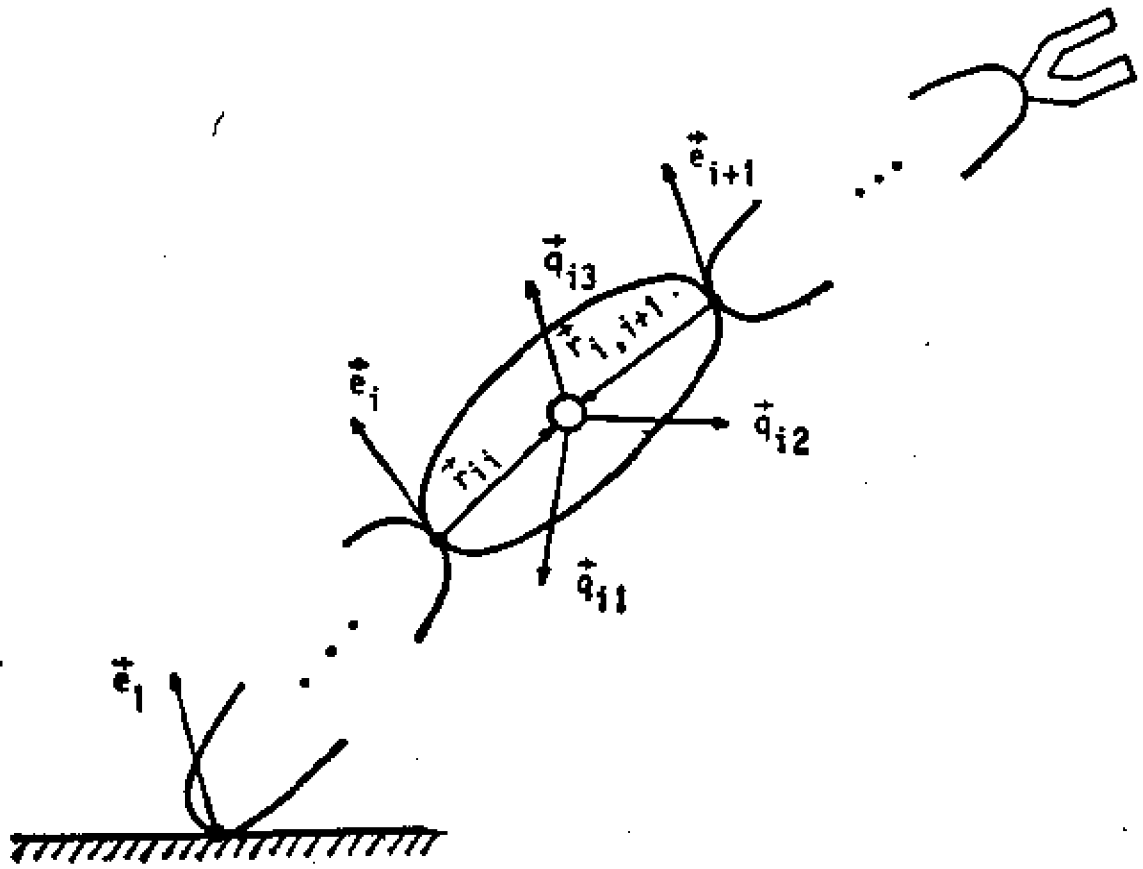


图 1.3 操作手连杆

出惯性矩阵,给出动力学正问题的解。

惯性和转矩可用类似关系式表示为

$$\begin{aligned} F_i &= \sum_{j=1}^i a_{ij} \ddot{q}_j + a_i^0 \\ M_i &= \sum_{j=1}^i b_{ij} \ddot{q}_j + b_i^0 \end{aligned} \quad (1.3.5)$$

式中系数 a_{ij} , a_i^0 , b_{ij} 和 b_i^0 由对应的递推关系式得到。注意,系数 b_{ij} 和 b_i^0 由欧拉动力学方程导出,并且它们只使用主惯性矩,而没有用惯性张量。最后,运动方程由如下矩阵形式得到

$$P = H(q)\ddot{q} + h(q, \dot{q})$$

式中 $H(q)$ 是 $n \times n$ 惯性矩阵,

$$H_{ik} = -e_i \cdot \sum_{j=\max(i,k)}^n (b_{ij} + r_{ij} \times a_{ik}), \quad i, k = 1, \dots, n \quad (1.3.6)$$

而 $h(q, \dot{q})$ 是一个具有 n 个元素的向量

$$h_i = -e_i \cdot \sum_{j=1}^n [r_{ij} \times (a_j^0 + G_j) + b_j^0] \quad (1.3.7)$$

式中 G_j 是第 j 个连杆的重力向量。上述关系式是针对旋转关节而写的,但对滑动关节也容易改写出对应的公式^[22,44]。

计算矩阵 $H(q)$ 和 $h(q, \dot{q})$ 所需的数值运算次数是

$$\begin{aligned} n_M &= \frac{3}{2} n^3 + 28n^2 + \frac{525}{2} n \\ n_A &= \frac{4}{3} n^3 + 20n^2 + \frac{530}{3} n \end{aligned}$$

式中 n_M 表示乘法运算次数, n_A 表示加法运算次数。

只求解动力学逆问题时,牛顿-欧拉法变得相当简

单^[22,41]。这时,运动学方程简化为三个递推关系式:

$$\begin{aligned}\omega_i &= \omega_{i-1} + \dot{q}_i e_i \\ \epsilon_i &= \epsilon_{i-1} + \ddot{q}_i e_i + \dot{q}_i (\omega_{i-1} \times e_i) \\ \omega_i &= \omega_{i-1} - \epsilon_{i-1} \times r_{i-1,i} + \epsilon_i \times r_{ii} - \omega_{i-1} \\ &\quad \times (\omega_{i-1} \times r_{i-1,i}) + \omega_i \times (\omega_i \times r_{ii})\end{aligned}\quad (1.3.8)$$

而牛顿-欧拉法的动力学方程简化为

$$\begin{aligned}F_i &= m_i \omega_i \\ M_i &= J_i \epsilon_i + \omega_i \times (J_i \omega_i)\end{aligned}\quad (1.3.9)$$

式中 $J_i = A_i J_i A_i^T$ 是第 i 个连杆对参考系的 3×3 惯性矩阵, m_i 是第 i 个连杆质量。

让我们用 R_i 和 μ_i 分别表示第 i 个连杆加在第 $i-1$ 个连杆上的力和转矩。对这两个变量,下列递推关系式成立^[22,41]:

$$\begin{aligned}R_i &= R_{i+1} - F_i - G_i \\ \mu_i &= \mu_{i+1} + (r_{ii} - r_{i,i+1}) \times R_{i+1} + r_{ii} \times (F_i + G_i) - M_i\end{aligned}\quad (1.3.10)$$

式中 R_{n+1} 和 μ_{n+1} 是加在第 n 个连杆(机器人抓手)上的力和转矩。由于式(1.3.10)是按次序 $i = n, i = n-1, \dots, i = 1$ 计算,因此它是后向递推关系式。对旋转关节,驱动转矩由

$$P_i = \mu_i \cdot e_i \quad (1.3.11)$$

计算,而对滑动关节,驱动力由

$$F_i = R_i \cdot e_i$$

计算。

计算驱动转矩所需的数值运算次数是

$$n_M = 225n$$

$$n_A = 152n$$

J. Luh, M. Walker 和 R. Paul 通过用连杆坐标系表示所有运动学和动力学变量改善了上述算法^[33]。这就为更有效地计算欧拉动力学方程创造了条件,使数值运算次数减少到^[16,17]

$$n_M = 150n - 48$$

$$n_A = 131n - 48$$

让我们再一次强调,这种算法只能求解动力学的逆问题。

最后,让我们研究在现代微型计算机上,把牛顿-欧拉算法用于实时实现的可能性。首先,让我们考虑既能求解动力学正问题,又能求解动力学逆问题的算法(1.3.3—1.3.7)。基于给定表示式,我们得到,为完成三个自由度的计算,需要超过 900 次浮点乘法运算和相近次数的加法运算。对六个自由度的操作手,乘法运算次数总计超过 2300 次。在介绍 Uicker-Kahn 法时,我们采用了这样的现代微型计算机,它进行一次乘法和一次加法运算约需要 0.1ms。这意味着,对三个自由度的操作手,计算模型矩阵所需的时间超过 90ms。这个时间远大于 20ms,而 20ms 被认为是机器人系统的最大采样周期。所以,我们的结论是,这些方法不适合实时应用。

其次,让我们考虑求解动力学逆问题的牛顿-欧拉算法。按照 J. Luh 和其它作者的算法,可以得到,对三个自由度的操作手,乘法运算次数是 $n_M \approx 400$,而对六个自由度的机器人, $n_M \approx 850$ 。通过类似于上述情况的分析,我们得到结论,对三个自由度的操作手,计算驱动力(转矩)需要 40ms。对实时系统运算,这个时间仍然太长。因此可见,使用现代微型计算机,牛顿-欧拉法也不适合实时实现。

1.3.2 Huston-Kane 法

Huston-Kane 算法是 Kane 动力学方程为基础的 特

$$+ 2 \sin \frac{q_i}{2} \cos \frac{q_i}{2} (\mathbf{e}_i \times \mathbf{r}) \quad (1.3.13)$$

式中

\mathbf{r} ——旋转前的向量；

\mathbf{r}' ——对轴 \mathbf{e}_i 经过角 q_i 旋转后的向量。

把

$$\mathbf{e}_i = \mathbf{e}_i \sin \frac{q_i}{2}$$

和

$$\mathbf{e}_{i+1} = \cos \frac{q_i}{2}$$

引入这个公式,得到

$$\mathbf{r}' = \mathbf{r} + 2\mathbf{e}_i \times (\mathbf{e}_i \times \mathbf{r}) + 2\mathbf{e}_{i+1}(\mathbf{e}_i \times \mathbf{r}) \quad (1.3.14)$$

所以,如果已知欧拉参数,则可立即写出 Rodrigues 公式;反过来也是一样。

现在考虑 Huston-Kane 算法运动学问题,它是由欧拉参数表示的连杆的速度和加速度组成的,并可由非递推关系式表示,例如

$$\boldsymbol{\omega}_i = \sum_{k=1}^i \hat{\omega}_k \quad (1.3.15)$$

式中

$\boldsymbol{\omega}_i$ ——第 i 个连杆对参考系的角速度；

$\hat{\omega}_k$ ——第 k 个连杆对第 $k-1$ 个连杆的相对角速度。

我们注意到,这个关系式可写为递推形式:

$$\boldsymbol{\omega}_i = \boldsymbol{\omega}_{i-1} + \hat{\omega}_i \quad (1.3.16)$$

我们知道,由于

$$\boldsymbol{\omega}_i = \mathbf{e}_i \dot{q}_i$$

关系式(1.3.16)与前述的牛顿-欧拉法中的对应公式(1.3.3)是相同的。这样,两种方法的根本区别只在于,为计算所有运动学和动力学变量, Huston-Kane 算法利用了非递推关系式,而不是递推关系式。

Huston-Kane 法的动力学分析是以牛顿-欧拉动力学方程(1.3.9)为基础的。驱动转矩(力)用类似于前节描述的方法确定。

由上述可知, Huston 算法属于一类基于牛顿-欧拉方程的非递推方法。这种算法的优点是,它能简单地处理复杂的关节(直到具有六个自由度),换言之,它能处理具有直到 $6n$ 个自由度 (n 是连杆数)的复杂操作手。但是,把这种算法用来处理简单开式机构,反而使问题复杂化了,而最通用的机器人都属于这类机构。由于这种算法是一种非递推算法,因此为计算模型矩阵所需的数值运算次数远大于上述递推算法。

1.3.3 一种新的递推法

在参考文献[24]中, Renaud 给出了解析-迭代法,以获得操作手动力学数学模型的惯性矩阵,它表示为

$$H(q)\ddot{q} + \dot{q}^T C(q)\dot{q} + G(q) = P \quad (1.3.17)$$

为获得 C 矩阵,他利用了 H 矩阵元素对广义坐标的偏导数,如果目标是建立计算机辅助方法,这种表示法是相当困难的。

通过利用拉格朗日方程, Renaud 获得了 H 矩阵元素,并判明这些元素包括张量 K' ,事实上张量 K' 表示了具有迭代性的广义连杆的动力学参数集合^[24]。Renaud 断定,广义连杆动力学参数是 H 和 C 矩阵元素所共有的。

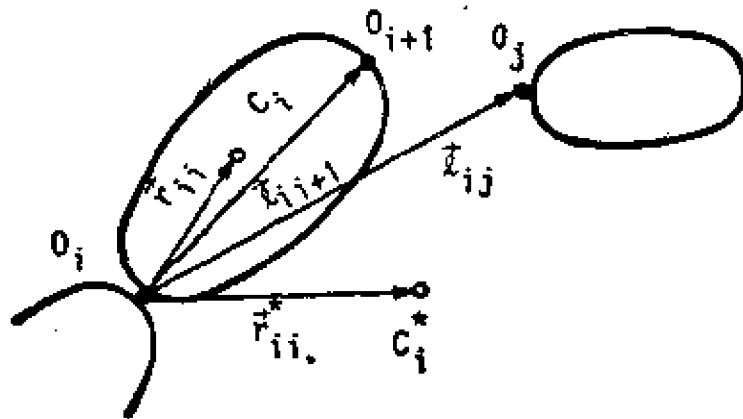


图 1.4 广义连杆 i

现在让我们简明扼要地阐述这种方法。从操作手基底的末端开始, 连杆 $n, n-1, \dots, i$ 组成广义连杆^[24](图 1.4), 引入下列符号:

- m_i^* ——广义连杆 i 的质量;
- r_{ii}^* ——由原点 O_i 到广义连杆 i 质量中心 C_i^* 的向量;
- l_{ii} ——由原点 O_i 到原点 O_i 的向量;
- r_{ij} ——由原点 O_i 到连杆 i 质量中心 C_i 的向量;
- m_i ——连杆 i 的质量;
- J_i ——连杆 i 对原点 O_i 的惯性张量;
- J_i^* ——广义连杆 i 对原点 O_i 的惯性张量;
- A_i ——两个联接连杆之间的转换矩阵。

让我们引入

$$P_i = m_i^* \cdot r_{ii}^* \quad (1.3.18)$$

则有

$$\begin{aligned} (J_{i+1}^*)_{O_i} = & J_{i+1}^* + m_{i+1}^* [(l_{i,i+1}^*)I - l_{i,i+1} \cdot l_{i,i+1}^*] \\ & + (2P_{i+1}^T \cdot l_{i,i+1}^*)I - l_{i,i+1} \cdot P_{i+1}^T \\ & - P_{i+1} \cdot l_{i,i+1}^* \end{aligned}$$

$$\begin{aligned}
m_n^* &= m_n, \quad m_i^* = m_{i+1}^* + m_i, \quad i = n-1, \dots, 1 \\
P_n &= m_n r_{nn}, \quad (r_{nn}^* = r_{nn}) \\
P_i &= P_{i+1} + m_{i+1}^* l_{i,i+1} + m_i r_{ii}, \quad i = n-1, \dots, 1 \\
J_n^* &= J_n \\
J_i^* &= J_{i+1}^* + (2P_{i+1}^T \cdot l_{i,i+1})l - l_{i,i+1} \cdot P_{i+1}^T \\
&\quad - P_{i+1} \cdot l_{i,i+1}^T + J_i + m_{i+1}^* (l_{i,i+1}^T l \\
&\quad - l_{i,i+1} \cdot l_{i,i+1}^T), \quad i = n-1, \dots, 1
\end{aligned} \tag{1.3.19}$$

由 H , C 和 G 矩阵的定义, 我们有

$$\begin{aligned}
H_{ij} &= e_i^T \cdot \frac{\partial P_i}{\partial \dot{q}_j} \\
C_{ik} &= \frac{1}{2} e_i^T \cdot \frac{\partial^2 P_i}{\partial \dot{q}_i \partial \dot{q}_k} \\
G_i &= e_i^T [r_{ii}^* \times (m_i^* g)]
\end{aligned} \tag{1.3.20}$$

根据引入的广义连杆动力学参数, 这些矩阵的元素是

$$\begin{aligned}
H_{ij} &= e_i^T J_i^* e_j + e_i^T [P_i \times (e_j \times l_{ii})] \\
C_{ik} &= e_i^T \left\{ e_i \times \left[J_i^* - \left(\frac{1}{2} \text{tr} J_i^* \right) l \right] e_k \right\} \\
&\quad + e_i^T \{ l_{ii} \times [e_k \times (e_i \times P_i)] \} \\
G_i &= e_i^T [r_{ii}^* \times (m_i^* g)]
\end{aligned} \tag{1.3.21}$$

动力学参数的迭代式适用于连杆坐标系。

让我们引入几个符号:

${}^i J_i$ —— 对应连杆系 i 原点 O_i 的连杆 i 的惯性张量;

${}^i \bar{J}_i$ —— 对应连杆系 i 原点 O_i 的增广连杆 i 的惯性张量。

则有

$$\begin{aligned}
{}^i \bar{J}_i &= {}^i J_i + (l_{i,i+1}^T l - {}^i l_{i,i+1} \cdot {}^i l_{i,i+1}^T) m_{i+1}^* \\
&\quad i = 1, \dots, n-1 \\
{}^n \bar{J}_n &= {}^n J_n
\end{aligned}$$

$$\begin{aligned}
v_i &= \frac{1}{2} \operatorname{tr} J_i^*, \quad i = 1, \dots, n \\
\bar{v}_i &= \frac{1}{2} \operatorname{tr} \bar{J}_i, \quad i = 1, \dots, n \\
{}^i\bar{P}_i &= m_i r_{ii} + m_{i+1}^* l_{i,i+1} \\
b_{ii} &= e_i \times l_{ii}
\end{aligned} \tag{1.3.22}$$

应当注意, ${}^i\bar{J}_i, {}^i\bar{P}_i$ 和 v_i 与操作手的位置是独立的。换句话说, 这些量可预先计算。

计算 H, C 和 G 矩阵所有元素的方法如下:

$$\begin{aligned}
\text{I} \quad & {}^{i+1}e_j = A_{i+1}^T e_j, \quad j = 1, \dots, n-1 \\
& \quad \quad \quad i = j, \dots, n-1 \\
\text{II} \quad & {}^*P_n = {}^*P_n \\
& \quad \quad \quad {}^{i-1}P_i = A_i^T P_i \\
& \quad \quad \quad {}^{i-1}P_{i-1} = {}^{i-1}P_i + {}^{i-1}\bar{P}_{i-1}, \quad i = n, \dots, 2 \\
\text{III} \quad & {}^i b_{ii} = A_i^T ({}^{i-1}b_{i,i-1} + {}^{i-1}e_i \times {}^{i-1}e_{i-1}) \\
& \quad \quad \quad {}^i b_{ij} = 0, \quad j = 1, \dots, n-1 \\
& \quad \quad \quad \quad \quad \quad i = j+1, \dots, n \\
\text{IV} \quad & {}^i J_i^* = A_{i+1} {}^{i+1} J_{i+1}^* A_{i+1}^T + (2 {}^i e_{i,i+1}^T {}^i P_{i+1}) l \\
& \quad \quad \quad - {}^i l_{i,i+1} {}^i P_{i+1}^T - {}^i P_{i+1} {}^i e_{i,i+1}^T + {}^i J_i \\
& \quad \quad \quad {}^* J_i^* = {}^* \bar{J}_i, \quad i = n-1, \dots, 1 \tag{1.3.23} \\
\text{V} \quad & v_i = v_{i+1} + 2 {}^i e_{i,i+1}^T {}^i P_{i+1} + \bar{v}_i, \quad i = n-1, \dots, 1 \\
\text{VI} \quad & H_{ij} = {}^i e_j^T {}^i J_i^* e_i + {}^i e_j^T ({}^i P_i \times {}^i b_{ii}), \quad i = n-1, \dots, 1 \\
& \quad \quad \quad \quad \quad \quad j = 2, \dots, i-1 \\
& \quad \quad \quad H_{ii} = {}^i e_i^T {}^i J_i^* e_i, \quad i = 1, \dots, n \\
\text{VII} \quad & C_{ik} = {}^i e_k \cdot [({}^i J_i^* - v_i I) ({}^i e_i \times {}^i e_i) \\
& \quad \quad \quad + ({}^i e_i \times {}^i P_i) \times {}^i b_{ii}], \quad i = 2, \dots, n \\
& \quad \quad \quad \quad \quad \quad j = 1, \dots, i-1 \\
& \quad \quad \quad \quad \quad \quad k = 1, \dots, i
\end{aligned}$$

VIII

$$\dot{G}_i = {}^i e_i^T ({}^i P_i \times {}^i g)$$

$${}^i g = A_i^T {}^{i-1} g, \quad i = 1, \dots, n$$

当利用 Denavit-Hartenberg 符号时^[69], 则有

$$A_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha_i & -\sin \alpha_i \\ 0 & \sin \alpha_i & \cos \alpha_i \end{bmatrix} \begin{bmatrix} \cos q_i & -\sin q_i & 0 \\ \sin q_i & \cos q_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1.3.24)$$

和

$${}^i e_{i,i+1} = \begin{bmatrix} a_i \\ 0 \\ d_i \end{bmatrix}, \quad {}^i e_i = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad {}^i b_{i,i+1} = \begin{bmatrix} 0 \\ a_i \\ 0 \end{bmatrix} \quad (1.3.25)$$

计算驱动转矩 P_i 以及计算矩阵 H , C 和 G 所需的浮点乘法运算次数 n_M 和加法运算次数 n_A 给定为

$$\begin{aligned} n_M &= \frac{3}{2} n^3 + \frac{35}{2} n^2 + 9n - 16 \\ n_A &= \frac{7}{6} n^3 + \frac{23}{2} n^2 + \frac{64}{3} n - 28 \end{aligned} \quad (1.3.26)$$

式中 n 是自由度数。对具有六个旋转关节的操作手, 我们有 $n_M = 992$ 和 $n_A = 776$ 。这远小于(甚至 3 倍)的其它数值方法, 在参考文献[68]中较详细地论述了这种方法。

1.4 基于 Appel 方程的方法

Appel 方程法最初是由 E. P. Popov 及其同事设想出来的^[67], 后由 M. Vukobratović 和 V. Potkonjak 完成了它的最终形式^[48,26]。这种方法既能求解逆的, 也能求解正的动力学问题。这些方程是以 Gibbs “加速度能量”函数为基础的。

在这种方法中,我们要用与牛顿-欧拉法相同的方法来描述连杆的布局和它们的运动学。

邻近连杆间的相对位置用 Rodrigues 公式描述。第 $i-1$ 个连杆坐标系和第 i 个连杆局部系之间的转换矩阵 A_i^{-1} 则用这种方式获得。角速度 $\tilde{\omega}_i$, 角加速度 $\tilde{\epsilon}_i$ 和线加速度 $\tilde{\omega}_i$ 等运动学变量用第 i 个连杆的局部坐标系确定。确定这些运动学变量的关系式类似于牛顿-欧拉法的关系式 (1.3.8):

$$\begin{aligned}\tilde{\omega}_i &= A_i^{-1}\tilde{\omega}_{i-1} + \dot{q}_i\tilde{e}_i \\ \tilde{\epsilon}_i &= A_i^{-1}\tilde{\epsilon}_{i-1} + \ddot{q}_i\tilde{e}_i + \dot{q}_i(\tilde{\omega}_i \times \tilde{e}_i) \\ \tilde{\omega}_i &= A_i^{-1}[\tilde{\omega}_{i-1} - \tilde{\epsilon}_{i-1} \times \tilde{r}_{i-1,i} - \tilde{\omega}_{i-1} \\ &\quad \times (\tilde{\omega}_{i-1} \times \tilde{r}_{i-1,i})] + \tilde{\epsilon}_i \times \tilde{r}_{ii} \\ &\quad + \tilde{\omega}_i(\tilde{\omega}_i \times \tilde{r}_{ii})\end{aligned}\quad (1.4.1)$$

式中

\tilde{e}_i ——用第 i 个连杆局部坐标系表示的第 i 个关节轴的单位向量;

\tilde{r}_{ii} ——用第 i 个连杆局部坐标系表示的,由第 i 个关节到第 i 个连杆质量中心的距离向量。

机构动力学由如下 Gibbs-Appel 方程描述:

$$\frac{\partial G}{\partial \ddot{q}_i} = Q_i, \quad i = 1, \dots, n \quad (1.4.2)$$

式中

G ——“加速度能量”函数;

Q_i ——用第 i 个关节表示的广义力。

“加速度能量”函数 G 可用和式表示

$$G = \sum_{i=1}^n G_i \quad (1.4.3)$$

式中 G_i 是与第 i 个连杆有关的 Gibbs 函数,由表示式

$$G_i = \frac{1}{2} m_i \tilde{\omega}_i^2 + \frac{1}{2} \tilde{\epsilon}_i \cdot \tilde{J}_i \tilde{\epsilon}_i + 2(\tilde{\omega}_i \times \tilde{J}_i \tilde{\omega}_i) \cdot \tilde{\epsilon}_i \quad (1.4.4)$$

给出。把式(1.4.1)代入式(1.4.4)和(1.4.2)，则得到机器人动力学模型

$$\begin{aligned} H(q)\ddot{q} + h_c(q, \dot{q}) &= Q \\ Q &= [Q_1, \dots, Q_n]^T \end{aligned} \quad (1.4.5)$$

式中

Q ——广义力向量；

$H(q)$ —— $n \times n$ 模型矩阵；

$h_c(q, \dot{q})$ —— $n \times 1$ 模型矩阵。

广义力向量 Q 可表示为驱动力(转矩)向量 P 与重力效应向量 $g(q)$ 的差

$$Q = P - g(q)$$

把上式代入式(1.4.5)，得到模型

$$H(q)\ddot{q} + h(q, \dot{q}) = P$$

式中

$$h(q, \dot{q}) = h_c(q, \dot{q}) + g(q)$$

应当说明，在介绍拉格朗日法和牛顿-欧拉法时，这些向量已经表明清楚了。例如，在 Uicker-Kahn 法中，在模型式(1.2.6)中，向量 $h_c(q, \dot{q})$ 已经表示为 $\dot{q}^T C(q) \dot{q}$ ，它代表科里奥利效应和离心效应向量。

如上所述可以得出结论，Appel 法属于一类既能给出正的，又能给出逆的动力学问题解的方法。应用 Appel 法建立动力学模型所需的数值运算次数是

$$n_M = \frac{7}{3} n^3 + 27n^2 + \frac{722}{3} n + 9$$

$$n_A = \frac{10}{3} n^3 + \frac{43}{2} n^2 + \frac{931}{6} n + 6$$

式中

n_M ——浮点乘法运算次数；

n_A ——浮点加法运算次数。

对具有三个和六个自由度的操作手，我们得到下列数值运算次数

n	n_M	n_A
3	1037	755
6	2929	2431

在前面几节介绍的算法中，我们采用的标准是，一次浮点乘法运算和一次加法运算约花费 0.1ms。我们已经假定，所有算法是在装有浮点协同处理器或适当硬件支持的现代微型计算机上实现的。我们还得到了这样的结果，在 20ms 采样周期内，每个算法允许最多包括 200 次乘法和加法运算。由于基于 Appel 方程的算法需要 5 倍多的规定运算次数（对具有三个旋转自由度的操作手），因此这种算法也不适合于实时实现。

1.5 符 号 法

前几节，我们介绍了几种机器人机构建模的数值方法。在这些方法中，所有变量都作为实数来处理。这意味着，在程序实现中，把一个存储地址分配给一个变量。变量值由对应存储地址的内容确定。在这一节和下一节中，我们要描述几种机器人建模的解析法。在这些方法中，一个变量用描述变量结构的符号集合表示，它独立于参数、关节坐标和关节变化率。下面我们把解析表示式称为“符号表示式”或“符号型”。

在最近 10 年，对处理符号代数式和超越式的软件包开展

了大量的研究工作。例如，这些研究工作包括函数代数的程序^[49]，计算有理函数的程序^[50,51]，等等。这些软件包都有下列相同的结构：

- 1) 输入符号式及其编辑程序的输入模块；
- 2) 综合分析和辨识表示式的模块；
- 3) 把符号译为机内码的模块；
- 4) 代数操作和化简的模块(它适当模拟人的动作)；
- 5) 输出模块。

很明显，这些软件包是很复杂的，这是因为依靠应用物理学和数学得到的表示式是极不相同的，需要高性能计算机才能实现。在非线性的数学模型中，为生成复杂符号型所需的计算机时间要求特别苛刻。这就是以应用物理学和化学某些领域为目标的软件包也得到发展的原因。例如，在理论地震学的应用中就使用了这类软件包^[52]。但是，由于复杂问题的数学模型通常是很复杂和显著呈现非线性的，因此甚至特定目的的程序也不能用来解决实际工程领域中更复杂的问题。为阐明这一点，让我们举出一类空间机构作为例子，这类空间机构包括机器人、人造卫星、维修装置、电力液压系统等通过关节联接的刚体或弹性体系统。对这类系统，由于惯性、哥氏力、离心力、外力和转矩的作用，各连杆之间必然存在动力学的相互影响(耦合)。相互作用力的大小取决于各连杆的瞬时布局、速度和加速度。这些力作用在所有连杆上，也就是单个连杆的运动通常影响到其余连杆的运动。于是，由此得到描述机器人动力学数学函数的复杂性，随着连杆数，即自由度的增加而迅速增加。因为这类模型的复杂性，所以实际上还没有符号代数的软件包适合处理机器人的动力学建模。S. Dillon 做过某些尝试，进一步分析并开发了开空间机构的

特定软件包^[53]。但是，实际上这种软件包似乎只能用于不超过三个自由度的系统¹⁾。

1.6 数值-符号法

在这一节中，我们要介绍一种新的机器人建模的计算机辅助法的基本原理，我们把这种方法称为数值-符号法。在下一章中，我们将详细地描述这种方法。

在上一节中，我们已经描述了以解析形式，即符号形式，用计算机构造机器人动力学方程方面遇到的困难。这些困难使得有关符号建模的思想在过去 10 年实际上已被抛弃了，除了一些例外，在参考文献中对此也没有予以考虑^[54]。

从另一方面来看，恰好在这过去 10 年中，机器人建模的数值方法得到了充分的发展。通过简要介绍，我们把数值方法分为两类：

1) 产生动力学模型矩阵的方法，这种方法既能给出动力学正问题的解，也能给出动力学逆问题的解；

2) 只能求解逆问题的方法，也就是，给定操作手运动，计算驱动力(转矩)。另外，我们还得到结论，属于第 1 类的方法，因为数值运算次数大，因此很难实时实现。而第 2 类的算法虽然只需相当少的数值运算，但对具有三个或更多个自由度的操作手，这类算法仍不适于实时实现。这些困难导致了对多道处理算法的研究^[55]，因为问题的复杂性和花费较大，因此这种算法仍被精心研制，目前处于仿真水平。应当说明，第 2 类算法只包括一类基于直接计算驱动力的控制算法。这些

1) 参考文献[53]中介绍了一个四个自由度操作手的例子，其中三个关节轴是平行的，结果导出了相当简化的模型。

方法不能用于建立动力学模型矩阵的控制规律和机器人运动的计算机仿真。总之,可以得到结论,数值方法还不能给出令人满意的结果,特别是在现代通用微型计算机上加以实现则更是如此。

通过分析数值方法,我们得到结论,数值运算次数太多是由于数值方法把数学模型结构中的所有运动学和动力学规律混在一起了。例如,牛顿-欧拉法就把 Rodrigues 公式、刚体线速度和角速度的规律、欧拉动力学方程和动静平衡条件混在一起了。但是,在建立模型期间,“通过”这些规律需要很长的处理时间,因而难于实时实现。现在我们提出这样一个问题,计算符号形式的模型矩阵所需的数值运算次数与对应的数值算法(它给出相同的模型)所需的数值运算次数之间有什么关系。因为符号模型不需要“通过”那些力学规律,而是表示为解析表示式的集合来直接计算模型矩阵的元素,所以很明显,符号模型比数值算法需要少得多的数值运算次数。

但是,如前所述,计算机辅助生成符号模型是一个很复杂的问题。这就是至今还没有满意求解方法的原因。为阐述下文将要详细描述和验证的新的建模原理,我们先考虑机器人建模的符号法与数值法之间的基本差别。正如表 1.1 所示¹⁾,两种建模原理都有各自的优点和缺点。现在人们不禁要问,能否把这两种方法结合起来产生一种新原理,而这种新原理又包括这两种方法的优点?下面我们就研究这种新原理,它正好有上述特性,在机器人建模中,我们把它称为“数值-符号”法或“解析”法。下面我们只介绍这种方法的基本思想,在第二章中将描述整个算法。

1) 表中给出的建模特性是相对关系,“极复杂”指的是建立符号模型的算法比数值法极为复杂,依此类推。

表 1.1 符号法和数值法的特性比较

性质 \ 方法	符号法	数值法
建模算法的复杂性	极复杂	简单
模型矩阵计算的数值运算次数	少	多

让我们考虑具有 n 个连杆和 n 个关节的操作手，这种操作手共有 n 个自由度。每个连杆都用参数集合表示它的特征。对已介绍过的 Uicker-Kahn 法，我们已经证明，转换矩阵用 3 个几何参数 (a_k, α_k, s_k) 描述。完整的表征一个连杆不仅需要对质量和惯性张量做出详细说明，而且对质量中心的位置也要做出详细说明。因此，每个连杆都要用 16 个参数描述。这种情况也适合其它方法。例如，利用 Rodrigues 公式的方法要用 12 个几何参数，质量和 3 个主惯性矩描述一个连杆。所以，我们得到结论， n 个自由度的机器人，其数学模型包括 $16n$ 个参数。用符号法建模时，把这 $16n$ 个参数看作为 $16n$ 个符号，加上 n 个关节坐标和 n 个关节速度，我们共得到 $18n$ 个符号。很明显，对有 $18n$ 个变量的函数的处理是极端复杂的。对六个自由度的操作手，在符号表示式中共有 100 多个符号变量，其中包括关节角的正弦和余弦函数。然而，这恰好是处理符号表示式的软件包不能生成机器人模型的难点之一。另一个难点是表示式本身的极端复杂性。这就是我们研究数值-符号法的原因，在数值-符号法中，表示式变得尽可能简单。为此，我们采用下列假定^[56-58]：

- 1) 把参数看作为数值常数，把参数之间的数值运算看作是数值量之间的运算；
- 2) 符号表示式只作为关节坐标的函数。很明显，第 1 个

假定省去了对应系统参数的 $16n$ 个符号变量。为满足第 2 个假定,我们引出机器人模型

$$P_i = \sum_{k=1}^n H_{ik}(q)\dot{q}_k + \sum_{k=1}^n \sum_{l=1}^n C_{il}^i(q)\dot{q}_k\dot{q}_l + h_i(q), \quad i = 1, \dots, n \quad (1.6.1)$$

用 Uicker-Kahn 法和多种其它方法都容易辨识这种模型.方程(1.6.1)可写成矩阵形式:

$$P = H(q)\dot{q} + \dot{q}^T C(q)\dot{q} + h_s(q) \quad (1.6.2)$$

式中

P —— n 维驱动转矩(力)向量;

$q(t)$ —— n 维关节坐标向量;

$H(q)$ —— $n \times n$ 惯性矩阵;

$C(q)$ —— $n \times n \times n$ 哥氏和离心效应矩阵;

$h_s(q)$ —— n 维重力效应向量.

这里, $\dot{q}^T C(q)\dot{q}$ 表示向量

$$\dot{q}^T C(q)\dot{q} = \begin{bmatrix} \dot{q}^T C^1(q)\dot{q} \\ \vdots \\ \dot{q}^T C^n(q)\dot{q} \end{bmatrix} \quad (1.6.3)$$

式中 $C^i(q) = [C_{ij}^i(q)]$ 是 $n \times n$ 矩阵.

我们可以从给定动力学模型的形式来掌握第 2 个假定的意义. 基本思想是符号表示式的操作只是关节坐标的函数. 因为用标准二次型 $\dot{q}^T C^i(q)\dot{q}$ 表示的模型中已含有关节速度,所以在符号模型生成的过程中,就没有理由再把关节速度当作符号了.

我们可以导出如下结论: 矩阵 $H(q)$, $C(q)$ 和向量 $h_s(q)$ 只是关节坐标的函数,而与关节速度无关. 通过确定这些矩阵,我们就完全确定了机器人的数学模型. 因为这些

矩阵只是关节坐标的函数，而这些矩阵的元素又可作为只是关节坐标函数的符号表示式，所以第 2 个假定可以得到满足。下面，我们把这些矩阵称为“动力学模型矩阵”，或简称为“动力学矩阵”。

现在让我们简要考虑建立动力学模型矩阵的问题。回顾数值方法可知，这些矩阵可通过 Uicker-Kahn 法^[18-19]，基于拉格朗日方程的 Mahil^[12] 和 Renaud^[20] 算法，或基于牛顿-欧拉方程的 Huston-Kane 算法^[43-45]等建立起来。但是，这些算法中没有一种算法能以适当形式给出动力学矩阵的元素。例如，在 Uicker-Kahn 法中包含转换矩阵的偏导数和惯性张量。在 Renaud 算法中也只包含惯性矩阵。很明显，用三个主惯性矩代替惯性张量更为方便。S. Mahil 和 R. Huston 算法中给出的动力学矩阵表示式也不够简单。这就是我们在下一章要研究这种表示式的原因，这种表示式尽可能最简单地描述了动力学矩阵的元素。理论推导是以牛顿-欧拉法为基础的^[20-22]。我们要重新阐述，这种方法是递推的，并且不以闭式给出这些矩阵的元素。作为结果，我们将得到动力学矩阵元素极为紧凑的表示式。例如，惯性矩阵第 (i, k) 个元素为¹⁾

$$H_{ik} = \sum_{j=\max(i,k)}^n [m_j(\mathbf{e}_i \times \mathbf{r}_{ij}) \cdot (\mathbf{e}_k \times \mathbf{r}_{jk}) + \mathbf{e}_i J_j \mathbf{e}_k] \quad (1.6.4)$$

式中

m_i ——第 i 个连杆的质量；

J_i ——用固定系表示的惯性矩阵；

\mathbf{e}_i ——第 i 个关节轴用参考系表示的单位向量；

1) 这个表示式适用于具有旋转关节的操作手。存在滑动关节时，表示式更为简单(参阅第二章)。

r_{ij} ——参考坐标系表示的,第 i 个关节到第 j 个连杆质量中心的距离向量。

可以看到,表示式(1.6.4)的结构比表示式(1.3.6)的结构更为简单,式(1.3.6)是源自牛顿-欧拉法的对应表示式。就是说,表示式(1.3.6)中包含一些系数,这些系数由一系列运动学和动力学建模的递推关系式(1.3.1)—(1.3.5)确定。

其余的动力学矩阵元素由类似于表示式(1.6.4)的公式获得。在这些表示式的基础上,容易证明模型矩阵的某些特性,如惯性矩阵的对称性和正定性,矩阵 $C^i(q)$ 的对称性和反对称性,等等。另外,我们不仅容易导出机器人的线性化模型,而且容易导出动力学参数变化的灵敏度模型。在 2.4 节和 2.5 节中将详细介绍这些算法。

现在让我们回到主题,研究建立符号表示式的问题。为简单阐述数值-符号法的基本思想,我们考察一个简单例子。在本书 5.2 节中将导出一个三个自由度的节肢机器人的解析模型。可以证明,这个机构惯性矩阵的元素 $H_{11}(q)$ 有解析形式:

$$\begin{aligned}
 H_{11}(q) = & 4.57 + 112.98\sin^2q_2 + 26.8 \cos q_3\sin^2q_2 \\
 & + 26.8 \cos q_2\sin q_2 \sin q_3 + 2.68\cos^2q_3\sin^2q_1 \\
 & + 5.36 \cos q_2 \cos q_3 \sin q_2 \sin q_3 \\
 & + 2.68\cos^2q_2\sin^2q_3
 \end{aligned} \tag{1.6.5}$$

这个表示式可用数值数据的排列集合:

- 1) 系数向量;
- 2) 指数矩阵;

也就是用排列对

$$\begin{bmatrix} 4.57 \\ 112.98 \\ 26.8 \\ 26.8 \\ 2.68 \\ 5.36 \\ 2.68 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \end{bmatrix} \quad (1.6.6)$$

唯一地表示出来。那么我们容易建立这个数据集合和解析表示式之间的相互关系。系数向量的元素直接对应解析表示式的系数。指数矩阵的一个行对应每一个系数。这一行描述了对应系数组成的解析式内容。因此,行

$$[c_1 \cdots c_n, s_1 \cdots s_n, u_1 \cdots u_n]$$

在解析表示式中对应

$$(\cos q_1)^{c_1} \cdots (\cos q_n)^{c_n} (\sin q_1)^{s_1} \cdots (\sin q_n)^{s_n} q_1^{u_1} \cdots q_n^{u_n}$$

其中 c_i, s_i 和 u_i 是整数指数,符号“ \cdot ”表示乘法。让我们看一个例子,在式(1.6.6)中,系数向量和指数矩阵的第三行是

$$([26.80], [001 \ 020 \ 000])$$

在解析表示式(1.6.5)中,对应上述行的第三相加项是

$$26.8 \cos q_3 \sin^2 q_3$$

很明显,式(1.6.6)中的行数对应解析表示式(1.6.5)中的相加项数。

按照上述定义,我们把数值数据的排列结构称为“多项式矩阵”。参与构造数学模型的任何变量都可用这类多项式矩阵描述。在第二章,我们将证明一个对应任何模型变量指数矩阵的极为有趣的特性:指数矩阵元素只取值 0,1 或 2。对算法的程序实现,这一特性特别重要。很明显,如果我们希望用多项式矩阵代替变量(参与建立模型矩阵的变量),那么就

要研究适合于多项式矩阵的代数。在 3.2 节将要证明，多项式矩阵之间的代数运算可简化为系数向量与指数矩阵之间的简单运算，而这些运算很适合程序设计。我们可把这些结果用来研究机器人建模的数值-符号算法，例如

- 1) 生成机器人动力学模型的算法 (3.4 节);
- 2) 生成线性化动力学模型的算法 (3.5 节);
- 3) 生成参数变化的灵敏度动力学模型的算法 (3.5 节);
- 4) 在给定允许相对误差条件下，推导近似模型的算法 (3.6 节)。

对已得多项式矩阵的相应解析表示式，进一步提出的问题是它的优化计算问题。为清楚起见，让我们回到前述的例子。通过直接枚举运算可知，节肢机器人元素 $H_{11}(q)$ 的解析形式包含 20 次乘法和 6 次加法运算。另一方面，可把它表示为因子分解形式：

$$\begin{aligned}
 H_{11}(q) = \sin q_2 \{ \sin q_1 [112.98 + \cos q_3 (26.80 \\
 + 2.68 \cos q_3)] + \sin q_3 \cos q_1 (26.8 \\
 + 5.36 \cos q_3) \} + 2.68 \sin^2 q_3 \cos^2 q_2 + 4.57
 \end{aligned}
 \tag{1.6.7}$$

在这种形式中，只有 11 次乘法和 6 次加法运算。这表明，乘法运算次数节省了将近 50%。在第四章中，我们将仔细处理数值-符号型的优化因子分解问题(在数值运算次数最小意义上)。可以证明，优化计算结构可用计算图或与它对应的适当矩阵表示。例如，前述表示式 $H_{11}(q)$ 可用图 1.5 所示计算图表示，图中节点对应表示式 $H_{11}(q)$ 的元素，而支路则表示这些元素相乘。图中元素用缩写形式表示为

$$\begin{aligned}
 c_2 &= \cos q_2, \quad c_3 = \cos q_3 \\
 s_2 &= \sin q_2, \quad s_3 = \sin q_3
 \end{aligned}$$

常数表示为

$$K_1 = 4.57, K_2 = 112.98, \dots, K_7 = 2.68$$

(编号与系数向量次序一致)。加法运算通过几个支路汇入一个节点描述。这种计算图可用两个矩阵描述，其中一个矩阵包含节点元素，另一个矩阵描述支路结构。因此，公式化问题可归结为“确定优化计算矩阵表示式”问题。对所有参与构造数学模型的解析表示式，它们的矩阵已经确定时，问题就成为，如何以这些矩阵为基础生成“实时运算程序”。在第四章中将要阐述，计算图的矩阵表示式可用来生成一组递推关系式来计算对应的解析式。建立这种递推关系式的程序称为“专家程序”。专家程序的输出是提供一系列建立模型的程序指令。显然，为完成专家程序，就要把输出程序码写为微型计算机语言，以便在微型计算机上实现模型，或写为高级程序语言。

图 1.6 给出了数值-符号法的一般方块图，它包括实时运算的所有程序生成的步骤。事实上，这个方块图包括软件包

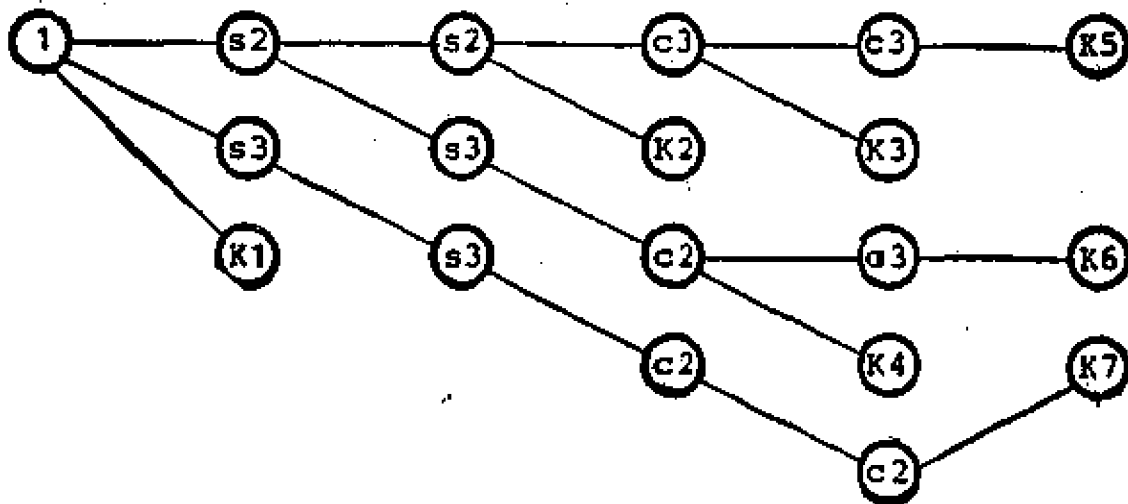


图 1.5 优化计算结构

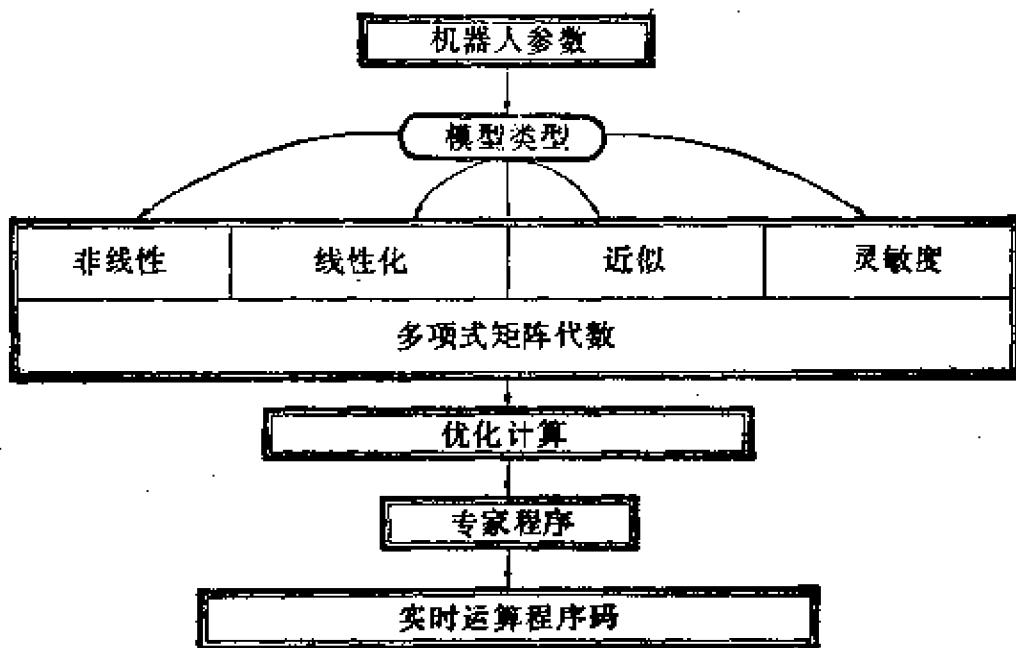


图 1.6 生成为实时运算的机器人模型算法的一般方块图

的所有基本部分,它的输入数据是机器人参数,而它的输出是机器人模型的源程序。软件包由下列几部分组成:

结构的机器人来说明数学模型对操作手结构的依存关系。这些例子都用来说明用数值-符号型建立非线性、线性化和灵敏度模型的方法。在每个例子的末尾都要介绍通过“专家程序”生成程序码。从而导出了建立模型所需的乘法和加法运算次数。

分析所得结果表明,数值-符号法生成模型所需的数值运算次数比数值法的对应次数小几倍。为说明研究的数值-符号法能极大地节省运算次数,在这里我们把一种数值法(1.3.3节中的方法)与这种方法加以比较。表 1.2 中给出了比较结果。选择拟人机器人作为例子,在 5.3 节导出了它的模型(表 5.39)。在 1.3.3 节中介绍了按照递推数值法所需的运算次数表示式

$$n_M = \frac{3}{2} n^3 + \frac{35}{2} n^2 + 9n - 16$$

$$n_A = \frac{7}{6} n^3 + \frac{23}{2} n^2 + \frac{64}{3} n - 28$$

这些表示式给出了建立动力学模型矩阵和驱动转矩向量所需的数值运算次数。用下述方法可获得数值-符号法所需的运算次数数据。在 5.3 节中将要证明,计算拟人机器人动力学模型矩阵式(1.6.2)需要 $n_M = 44$ 次浮点乘法运算和 $n_A = 23$ 次加法运算。为确保公平地比较这些方法,我们增加 21 次乘法和 6 次加法,以便计算向量

$$h(q, \dot{q}) = \dot{q}^T C(q) \dot{q} + g(q)$$

从而,我们获得表 1.2 中数据 $n_M = 65$ 和 $n_A = 29$ 。正如由表 1.2 看到的那样,乘法运算次数减少了 69%,加法运算次数减少了 59%。假设一次乘法运算持续时间是两次加法运算那么长,则我们得到,运算次数约有效地减少了 75%。

表 1.2 计算拟人机器人模型矩阵所需的运算次数

	新递推法	数值-符号法
n_M	209	65
n_A	171	29

在前几节介绍的各种方法中,我们采用的标准是,使用现代微型计算机进行一次乘法和一次加法运算约花费 0.1ms。我们已经知道,对具有三个或更多个旋转自由度的操作手,还没有一种数值方法能够实时实现(这是指,在 20ms 内不能建立起模型)。但是,在表 1.2 中表明,约在 5ms 内,就能建立起拟人操作手数值-符号模型,这只是最大允许时间的四分之一。随后表明,不仅对具有三个,而且对具有四个,有时甚至对具有五个和六个自由度的操作手,都能实时地建立起模型,当然,所有这些结论都是针对用单个处理机的机器人控制系统。利用多道处理机系统实现数值-符号法是一个重要问题,但这已超出了本书的范围。

尽管基于数值-符号法的实时运算程序比经典数值法的程序具有更好的性能,但要指出,这种新方法也有限制条件。如果一个运动学参数发生变化,就必须利用现有软件包来生成实时运算的新程序。经典法就不是这样,可是,如果一个动力学参数发生变化,则只要按照灵敏度模型,在现有程序中改变几个常数就足够了(第三章),而不需要生成新程序码。

第二章 建立机器人闭式动力学模型的计算机辅助法

2.1 引言

在这一章中,我们要介绍一种机器人系统建模的新方法。另外,还要证明动力学模型矩阵的几个基本性质。

为准确推导数学模型,在本章开始,我们先定义几个与机器人机构物理模型有关的基本概念,给出构形、连杆、运动副、运动链、关节、外部坐标和可动机构等词的定义。对机器人的物理模型,我们采用的主要假设可归纳为下列三点:

- 1) 忽略弹性效应,把连杆看作为刚体;
- 2) 机器人操作手的运动链中既无支路,也不闭合;
- 3) 在运动学上,不存在耦合自由度。

对操作机器人,这些假设大部分都能满足,但对移动机器人、复杂液压和其它机械系统,这些假设不成立。这就是为什么在本书中把“机器人”理解为“操作机器人”的原因。

为了对操作手进行数学建模,在 2.2 节中我们将介绍一

学变量, 及其参数联系起来。首先计算第一个连杆的各个变量, 然后利用前面的结果, 再计算第二个连杆的各个变量, 等等。

通过分析牛顿-欧拉法的算法可知(参阅 1.3.1 节), 实际上, 在这种方法中, 一系列的递推关系式是表示了一系列定律, 即表示了一系列互联刚体系统运动学和动力学的定理。既然如此, 我们可以就这类方法的本质提出一个重要问题, 这就是能否得到闭式数学模型, 而不用一系列递推关系式呢? 这意味着, 要以已知几何和动力学参数为基础, 直接生成模型的微分方程, 而不经历那些力学定律。在 2.3 节中将考虑这个问题。在这里要说明, 在准确列写闭式模型构造的问题时, 要满足下列要求:

- 1) 通过建立系统的惯性矩阵, 把惯性效应的作用从模型中分离出来;
- 2) 通过一个矩阵隔离集, 把哥氏和离心效应的作用分离出来, 矩阵隔离集的元素只是关节坐标的函数, 而与关节速度无关;
- 3) 通过一个单独的列矩阵, 把重力效应的作用分离出来;
- 4) 这些矩阵的元素都表示为最简单的表示式, 而不利用递推关系式。

我们要注意, 条件 1), 2) 和 3) 要求所有模型矩阵都只是关节坐标的函数, 也就是说, 在模型中显含地计入了关节速度和加速度。正如在第一章中所阐述的那样, 对于研究机器人建模的解析方法, 这是特别重要的。

在 1.2.1 节中, 我们扼要介绍了基本的牛顿-欧拉法, 现在把它作为开始求解所述问题的基础。在推导闭式模型中, 递

推关系式要转换为非递推形式。因为这种转换过程导致很复杂的表示式，所以引入一系列代数恒等式是必要的。将引入两类恒等式：

1) 为建立物理量之间，也就是运动学和动力学变量之间的关系所需的物理恒等式；

2) 为把多项式的和简化成对称二次型所需的数学恒等式。例如，物理恒等式是把关节轴单位向量与连杆坐标系单位向量联系起来的恒等式。通过数学恒等式获得对称二次型的思想是来自这样的事实，哥氏和离心效应是关节速度的乘积或平方的函数，就是说，科里奥利和离心力可用二次型描述，在一般情况下，它有如下结构

$$F_c = \sum_{k=1}^n \sum_{l=1}^n f_c(k, l) \dot{q}_k \dot{q}_l \quad (2.1.1)$$

式中

\dot{q}_i ——关节速度；

F_c ——总力；

$f_c(k, l)$ ——由乘积 $\dot{q}_k \dot{q}_l$ 引起的力的分量；

n ——自由度数。

这个二次型容易缩写成对称型

$$F_c = \dot{q}^T \phi_c \dot{q} \quad (2.1.2)$$

式中对称阵 ϕ_c 包括元素 $f_c(k, l)$ ， $f_c(k, l)$ 只是关节坐标的函数，而与关节速度无关。在这里可以判明，闭式模型构造满足第2)类恒等式的要求。

因为闭式模型构造方法相当复杂，所以在2.3节中只介绍它的基本思想和最重要的关系式，在附录2.1和2.2中给出其余部分的推导。

在2.3节的末尾，通过详细流程图，对获得的闭式模型加

以总结。可以看出,尽管进行了相当复杂的推导,但获得的模型关系式却是很简单和精致的。可以证明,闭式模型矩阵有下列特性:

1) 模型矩阵的元素只是三个向量变量集合的函数,这三个向量变量是: 关节轴单位向量,连杆坐标系单位向量,关节与连杆质量中心之间的距离向量;

2) 连杆的动力学参数(质量和惯性矩)线性地计入模型矩阵的所有元素中;

3) 如果存在滑动关节,则获得的模型关系式可变得相当简单。

在 2.4 节中,我们将证明另外几个动力学模型矩阵的基本特性。可以看出,这些特性与建模算法无关,而是表示了开链机构数学模型的固有特性:

1) 惯性矩阵是对称的和正定的;

2) 哥氏和离心效应矩阵是对称的。另外,同时考虑各不相同的自由度时,这些矩阵也必然拥有反对称性。

我们将准确地列写这些性质,并通过适当定理给出证明。从而解决完全非线性数学模型构造的问题。

下面,我们介绍构造线性化动力学闭式模型的方法。考虑到导出的数学模型中显含地包含关节速度和加速度,因此确定线性化模型的问题就归结为确定对关节坐标偏导数矩阵的问题。为确定这些矩阵,在 2.5 节中我们陈述和证明了三个定理:

1) 连杆上所附向量的偏导数定理,即,这个向量对相应的连杆坐标系是不变的;

2) 距离向量的偏导数定理;

3) 两个任意向量叉积(向量积)或点积的偏导数定理。

我们把这些定理应用到闭式模型的方程式就可导出机构的线性化模型。最后,通过详细流程图对模型加以总结。

在本章末尾,我们将介绍构造闭式灵敏度模型的算法。我们认为,灵敏度模型要回答如下问题^[39~40]:驱动转矩(力)怎样变化才能补偿连杆动力学参数的改变,以便使关节坐标和速度不出现摄动?我们用动力学参数表示质量和惯性。参数变化通常用最后一个连杆——末端执行器来描述,因为它代表了工件的变量参数。在推导结尾可以证明,计人灵敏度模型构造的矩阵,在数学上和物理上类似于惯性矩阵、哥氏和离心效应矩阵,以及重力向量。而且,对灵敏度模型矩阵来说,对称性、正定性和反对称性等特性也是成立的。然而,基本区别在于,灵敏度模型矩阵与动力学参数是独立的,因此不管动力学参数有多大改变,灵敏度模型矩阵总是成立的。这是非常重要的特性,因为在机器人中经常出现有限的参数变化。在本章最后,通过详细的流程图对介绍的算法加以总结。

2.2 基于牛顿-欧拉方程的模型

让我们考虑图 2.1 所示机器人机构的模型。模型由 n 个刚体组成,各个刚体就是机构运动链中的连杆。这些连杆用棱柱形(滑动)关节或旋转关节联接起来。为准确推导机器人动力学微分方程,让我们引入几个基本定义。

构形

机构的构形用一定规律排列的 n 个接头 (J_1, \dots, J_n) 表示,其中 $i \in N = \{1, \dots, n\}$, $J_i \in \{R, T\}$, R 表示旋转关节, T 表示棱柱形关节。

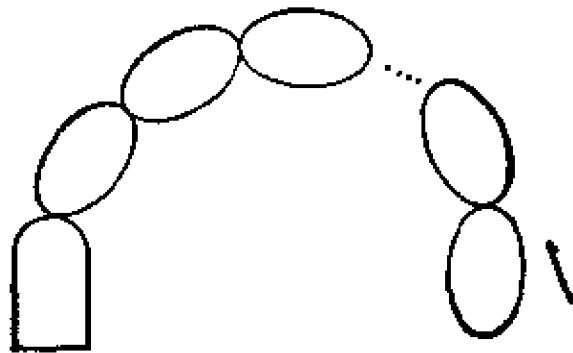


图 2.1 n 个连杆和 n 个关节的机器人机构的模型

举例说明，RTT-RRR 型机器人的构形是具有六个关节的机构，其中第二个和第三个关节是滑动的，其余的关节是旋转的。

连杆

连杆用一定规律排列的参数集合 $C_i(K_i, D_i)$ 定义，其中 K_i 表示运动学参数集合， D_i 表示动力学参数集合。

集合 K_i 和 D_i 可用不同方法定义，和本节研究的方法一样，在基本牛顿-欧拉法中，假设这些集合有下列形式：

$$K_i = (Q_i, \tilde{R}_i, \tilde{E}_i)$$

$$D_i = (m_i, \underline{J}_i)$$

式中

Q_i ——附属于连杆 i 的内部（局部）正交坐标系， $Q_i = (q_{i1}, q_{i2}, q_{i3})$ ；

\tilde{R}_i ——从点 z_{ik} 到坐标系 Q_i 原点的距离向量集合，其中点 z_{ik} 表示第 i 个和第 $k, k \in \{1, \dots, k_i\}$ 个机构连杆联接关节的中心， $\tilde{R}_i = \{\tilde{r}_{ik}\}^0$ ；

\tilde{E}_i ——关节轴单位向量的集合，通过这个关节轴，在点

1) 符号“ \sim ”表示对连杆坐标系 Q_i 规定的投影向量。

z_{ik} 把第 i 个连杆 C_i 与连杆 C_k 联接起来, $\tilde{E}_i = \{\tilde{e}_{ik}\}$;

m_i —— 第 i 个连杆的质量;

J_i —— 按局部系 Q_i 定义的第 i 个连杆的惯性张量。

我们通常认为,坐标系 Q_i 的轴与主惯性轴一致,因此惯性张量 J_i 可简化为三个惯性矩 $J_i = (J_{i1}, J_{i2}, J_{i3})$ 。另外认为,坐标系 Q_i 的原点与连杆 i 的质量中心相吻合。在图 2.2 中,给出了一个机器人连杆的例子。

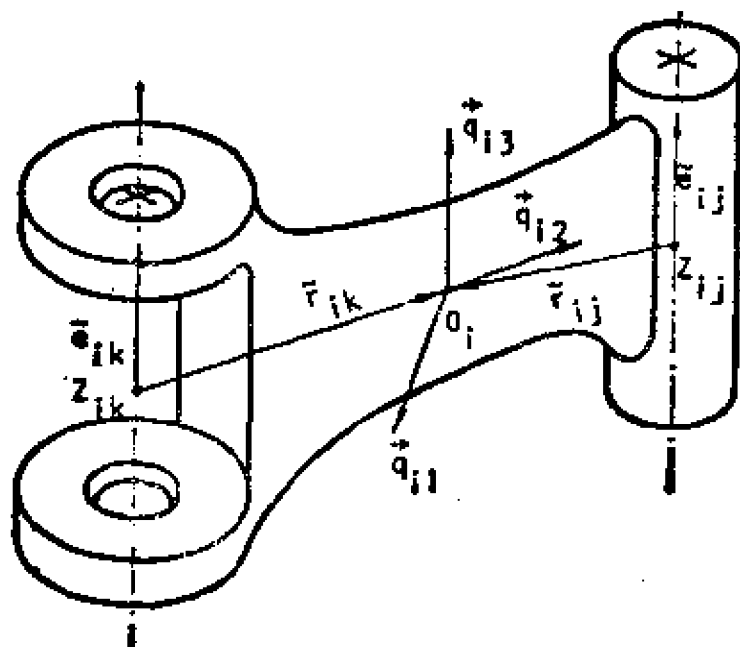


图 2.2 具有两个关节 (z_{ik} 和 z_{ij}), 质量中心为 O_i 的连杆 C_i

运动副

运动副 P_{ik} 表示通过一个关节在点 z_{ik} 互相联接的, 两个邻近连杆的集合 $\{C_i, C_k\}$ 。

以关节联接的类型为基础, 引入运动副的级和子级的概念。第 j ($j = 1, \dots, 5$) 级运动副定义为通过 $n - 6 - j$ 个自由度的关节互相联接的两个相邻连杆的集合。第 j 级和第 l 子级运动副定义为在点 z_{ik} 有 r 个旋转关节和 s 个棱柱形

关节的运动副,满足

$$r = \begin{cases} m - l + 1, & l \leq m + 1 \\ 0, & l > m + 1 \end{cases}$$

$$j = n - r$$

m 表示在 j 级中最大可能的旋转自由度数,例如,1级、2级和3级允许有三个旋转自由度 ($m = 3$),4级允许有两个旋转自由度,而5级只允许有一个旋转自由度。

运动链

运动链 A_n 是 n 个互联运动副的集合

$$A_n = \{P_{ik}\}, i \in N, k \in N$$

按照联接结构,链分为简单链、复杂链、开链和闭链。

在一个链中,没有一个连杆 C_i 引入两个以上的运动副,则该链称为简单运动链。反之,复杂运动链至少包括这样一个连杆 $C_i, i \in N$, 它引入了两个以上的运动副。

开式运动链至少拥有这样一个连杆,它只属于一个运动副。如果一个链中,每个连杆都至少引入了两个运动副,则该链称为闭链。

关节坐标

某些标量以唯一方式确定了运动副 $P_{ik} = \{C_i, C_k\}$ 中连杆的相对位置,则这些标量称为操作手关节坐标 q'_{ik} 。上标 $l \in \{1, \dots, s\}$, 其中 $s = 6 - j$ 是自由度数,而 j 是运动副 P_{ik} 的级。

如果只考虑第5级副,也就是只考虑具有单自由度相对运动的运动副时,关节坐标的定义可以简化。这种假设不丧失一般性,因为任何级的副都可用虚拟5级副的叠加来表示。

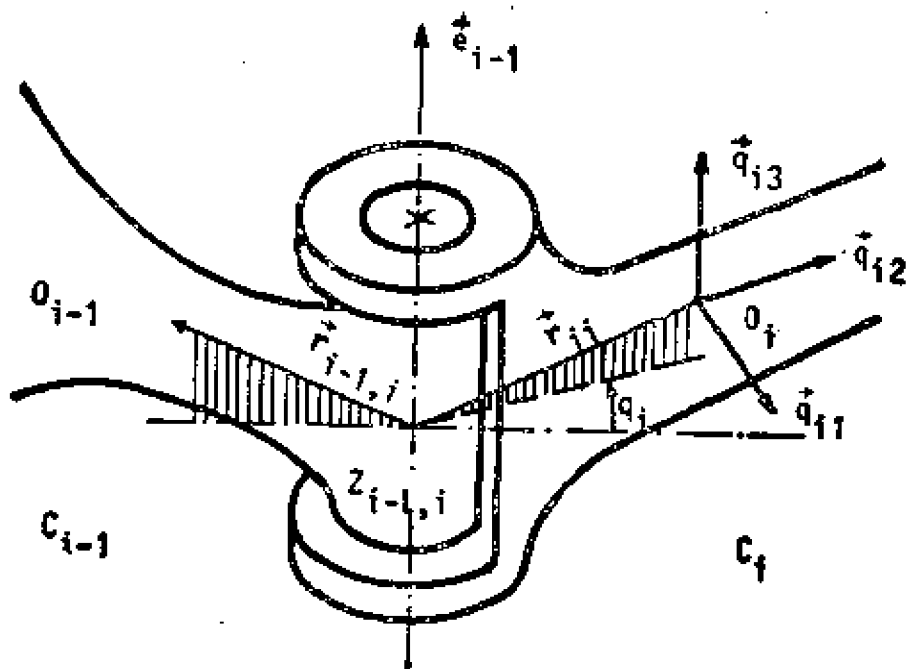


图 2.3 旋转运动副 $\{C_{i-1}, C_i\}$ 的关节坐标

所以，对于这样的副，以唯一方式定义关节坐标具有特殊意义。图 2.3 展示了一个旋转运动副，它的关节角（坐标）用 q_i 表示。图 2.4 展示了一个对应的滑动副。

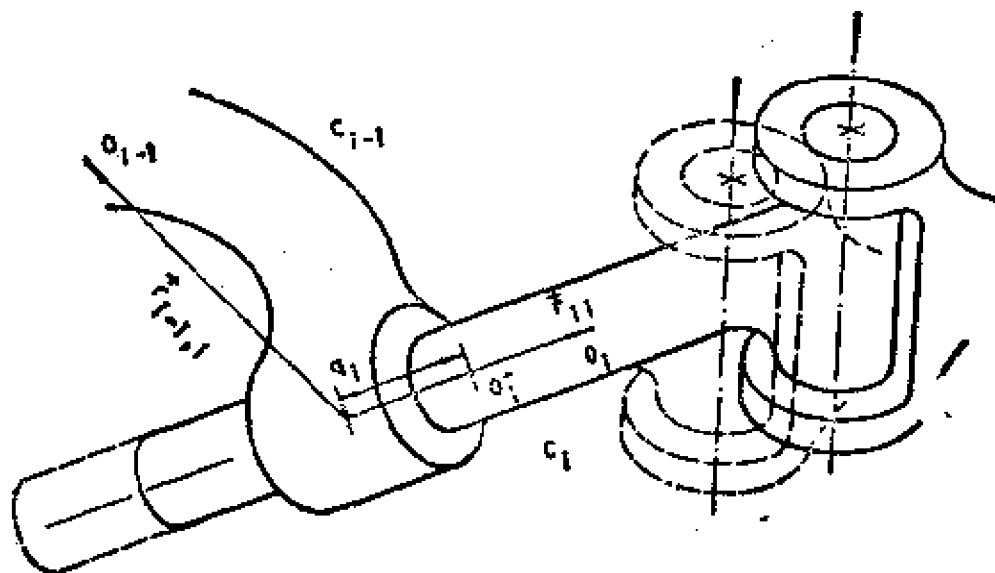


图 2.4 滑动运动副 $\{C_{i-1}, C_i\}$ 的关节坐标

外部坐标

某些标量 $x_{r,k}, k \in \{1, \dots, m\}$ 确定了链 A_r 中第 n 个连杆对参考坐标系的位置和(部分或全部)方位, 则这些标量称为机构的外部坐标。

对 $m \leq 6$, 外部坐标的分类是以操作任务的类型为基础的, 这里暂不考虑。

可动机构

可动机构表示一个系统, 它包括: 1) 机械部件, 可用适当的运动链建模; 2) 执行机构集合, 用它实现驱动转矩(力)。

为建立可动机构的数学模型, 必须

- 1) 辨识运动链和执行机构的参数;
- 2) 建立机构的动力学模型;
- 3) 建立执行机构的模型。

开式可动机构的动力学模型表示为 n 个非线性微分方程组, 它描述在驱动力作用下, 关节坐标空间内的系统运动。

下面介绍建立开式可动机构的动力学模型 我们介绍一种

$$\xi_i = \begin{cases} 0, & \text{第 } i \text{ 对是旋转副} \\ 1, & \text{第 } i \text{ 对是滑动副} \end{cases}$$

和由上述定义引入的符号一样,我们还使用符号:

$$\bar{\xi}_i = 1 - \xi_i$$

第一步: 确定基始位置

假设研究的可动机构用简单开运动链 A_n 描述,而 A_n 由运动副集合 $\{C_{i-1}, C_i\}, i \in N$ 组成. 实际上,运动副 $\{C_0, C_1\}$ 表示机构的第一个连杆,它通过点 z_{10} 与基座 C_0 联接. 现在需要在所有关节坐标等于零, $q_i = 0, i \in N$ 的条件下,确定所有连杆对参考坐标系的位置.

让我们假设已知所有运动学参数

$$K_i^0 = (Q_i^0, R_i^0, E_i^0), i \in (1, \dots, i-1)$$

式中上标“0”指的是 $q_i = 0, i \in N$. 那么确定基始位置的任务就归结为确定

$$K_i^0 = (Q_i^0, R_i^0, E_i^0)$$

对简单运动链情况,集合 R_i 和 E_i 有下列形式:

$$R_i = \{r_{ii}, r_{i,i+1}\}$$

$$E_i = \{e_i, e_{i+1}\}$$

式中 e_i 和 e_{i+1} 代表向量 e_{ii} 和 $e_{i,i+1}$.

以已知 K_{i-1}^0 假设为基础,可以得出, e_i^0 和 $r_{i-1,i}^0$ 也是已知的. 接着我们注意到,在点 $s_{i-1,i}$ 的三个正交单位向量集合 e_i^0, α_i^0 和 $e_i^0 \times \alpha_i^0$, 其中

$$\alpha_i^0 = \text{ort}[e_i^0 \times (r_{i-1,i}^0 \times e_i^0)]^{1)}$$

它们对参考坐标系的分量都是已知的(图 2.5). 另一方面,让

1) $\text{ort}(\cdot)$ 表示 (\cdot) 的单位向量.

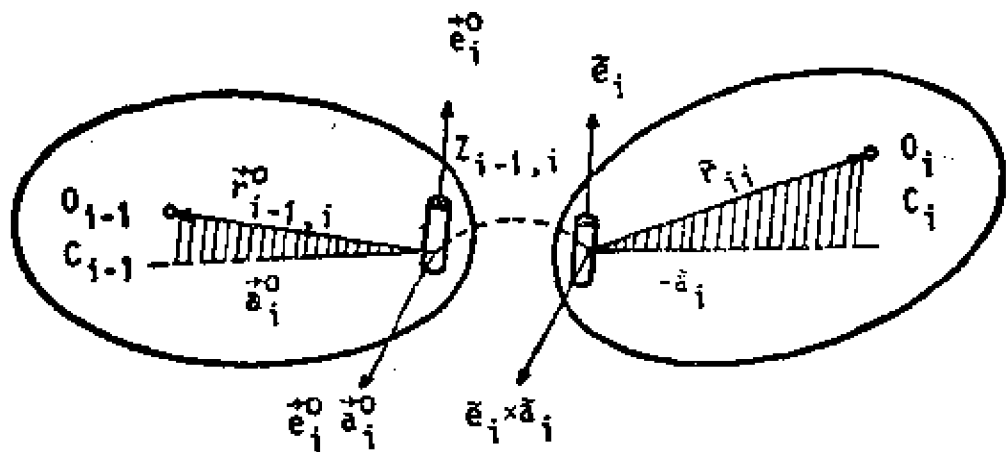


图 2.5 $q_i = 0$ 时, 连杆 C_i 对 C_{i-1} 的相对位置

我们看一看向量集合 \tilde{e}_i, \tilde{a}_i 和 $\tilde{e}_i \times \tilde{a}_i$, 其中

$$\tilde{a}_i = -\text{ort}[\tilde{e}_i \times (\tilde{r}_{ii} \times \tilde{e}_i)]$$

它们对局部坐标系 Q_i 的分量也是已知的。在 $q_i = 0$ 的条件下, 这两个向量集合是吻合的。由于

$$Q_i^0 = [q_i^0, q_i^0, q_i^0]$$

表示第 i 个连杆坐标系到参考系的转换矩阵, 所以得到

$$\begin{aligned} e_i^0 &= Q_i^0 \tilde{e}_i \\ a_i^0 &= Q_i^0 \tilde{a}_i \\ e_i^0 \times a_i^0 &= Q_i^0 (\tilde{e}_i \times \tilde{a}_i) \end{aligned} \quad (2.2.1)$$

这就完全确定了矩阵:

$$Q_i^0 = [e_i^0 \ a_i^0 \ e_i^0 \times a_i^0] [\tilde{e}_i \ \tilde{a}_i \ \tilde{e}_i \times \tilde{a}_i]^T \quad (2.2.2)$$

让我们注意, 由于向量 \tilde{e}_i, \tilde{a}_i 和 $\tilde{e}_i \times \tilde{a}_i$ 的正交性, 矩阵转置可用来代替它的逆。

现在需要确定集合 K_i^0 的其余元素, 即集合 R_i^0 和 E_i^0 。由于转换矩阵 Q_i^0 已经确定, 下列关系式显然成立:

$$\begin{aligned} r_{ii}^0 &= Q_i^0 \tilde{r}_{ii} \\ r_{i,i+1}^0 &= Q_i^0 \tilde{r}_{i,i+1} \\ e_{i+1}^0 &= Q_i^0 \tilde{e}_{i+1} \end{aligned} \quad (2.2.3)$$

至此,我们建立了计算 $Q_i^0, \forall i \in N$, 以及 R_i^0 和 E_i^0 中所有向量所需的递推关系式。

第二步: 确定机构的位置

这一步是确定运动学变量集合 $K_i = (Q_i, R_i, E_i)$ 的元素, 其中用关节坐标 $q_i, i \in N$ 规定机构的位置。

把有限旋转定理 (Rodrigues 公式) 用到旋转关节情况 ($\xi_i = 0$), 或把刚体平移定理用到滑动关节情况 ($\xi_i = 1$), 得到

$$q_{ij} = \begin{bmatrix} e_i(q_i^0 \times e_i)\xi_i \\ (e_i \times q_i^0)\xi_i \\ (e_i \cdot q_i^0)e_i\xi_i + q_i^0\xi_i \end{bmatrix}^T \begin{bmatrix} \cos q_i \\ \sin q_i \\ 1 \end{bmatrix}, \quad j = 1, 2, 3 \quad (2.2.4)$$

由此我们确定了矩阵:

$$Q_i = [q_{i1} \quad q_{i2} \quad q_{i3}]$$

它表示由第 i 个连杆坐标系 S_i 到参考系的转换矩阵。现在, 我们直接得到

$$\begin{aligned} R_i &= \{r_{ii}, r_{i,i+1}\} \\ &= \{Q_i \tilde{r}_{ii} + q_i e_i \xi_i, Q_i \tilde{r}_{i,i+1}\} \\ E_i &= \{e_i, e_{i+1}\} = \{e_i, Q_i \tilde{e}_{i+1}\} \end{aligned} \quad (2.2.5)$$

式中显然成立

$$r_{ij} = r_{i-1,j} - r_{i-1,i} + r_{ii}, \quad j \in I, \quad j \in N$$

在确定了 e_{i+1} 之后, 就可按照式 (2.2.4) 递推计算转换矩阵 Q_i , 进而计算出集合式 (2.2.5) 中的元素。

第三步: 机构运动学

这一步是建立运动学各量的集合

$$\mathcal{L}_i = \{Q_i, W_i\}$$

式中

$$Q_i = \{\omega_i\}$$

$$W_i = \{\epsilon_i, w_i\}$$

式中

ω_i ——第 i 个连杆的角速度；

ϵ_i ——第 i 个连杆的角加速度；

w_i ——第 i 个连杆的线加速度。

这些运动学量是关节坐标速度 \dot{q}_i 和加速度 \ddot{q}_i 的函数，也是上一步建立的变量

$$K_i = (Q_i, R_i, E_i)$$

的函数。这一步考虑的所有关系式都是递推的。应用刚体运动学基本定理，得到

$$\begin{aligned} \omega_i &= \omega_{i-1} + \dot{q}_i e_i \xi_i \\ \epsilon_i &= \epsilon_{i-1} + \dot{q}_i \omega_{i-1} \times e_i \xi_i + \ddot{q}_i e_i \xi_i \\ w_i &= w_{i-1} - \epsilon_{i-1} \times r_{i-1,i} - \omega_{i-1} (\omega_{i-1} \times r_{i-1,i}) \\ &\quad + \epsilon_i \times r_{ii} + \omega_i \times (\omega_i \times r_{ii}) \\ &\quad + (2\dot{q}_i \omega_{i-1} \times e_i + \ddot{q}_i e_i) \xi_i \end{aligned} \tag{2.2.6}$$

由此得到，角和线加速度是关节坐标二次导数 $\ddot{q}_i, j \in I$ 的函数。可是，为建立动力学模型矩阵，必须把 ϵ_i 和 w_i 的关系式重新排列，使得加速度 $\ddot{q}_i, j \in N$ 显含地表示为

$$\begin{aligned} \epsilon_i &= [\alpha_{i1} \cdots \alpha_{in} 0 \cdots 0] \ddot{q} + \alpha_i^0 \\ w_i &= [\beta_{i1} \cdots \beta_{in} 0 \cdots 0] \ddot{q} + \beta_i^0 \end{aligned} \tag{2.2.7}$$

式中

$$\ddot{q} = [\ddot{q}_1 \cdots \ddot{q}_n]$$

于是，运动学量集合 W_i 可归结为

$$W_i = \{\alpha_{ij}, \alpha_i^0, \beta_{ij}, \beta_i^0; j \in I\}$$

由式(2.2.6)和(2.2.7), 我们可获得确定集合 W_i 元素的递推关系式:

$$\begin{aligned} \alpha_{ij} &= e_j \xi_j, \quad j \in I \\ \alpha_i^0 &= \alpha_{i-1,i}^0 + \dot{q}_i (\omega_{i-1} \times e_i) \xi_i \\ \beta_{ij} &= \beta_{i-1,i} + e_j \times R_{i,i-1} \xi_j, \quad j \in \{1, \dots, i-1\} \\ \beta_{ii} &= (e_i \times r_{ii}) \xi_i + e_i \xi_i \\ \beta_i^0 &= \beta_{i-1,i} + \alpha_{i-1,i}^0 \times R_{i,i-1} + \dot{q}_i (\omega_{i-1} \times e_i) \times r_{ii} \xi_i \\ &\quad + 2\dot{q}_i (\omega_{i-1} \times e_i) \xi_i - r_{i-1,i} + r_{ii} \end{aligned} \quad (2.2.8)$$

式中

$$r_{ij} = \omega_i \times (\omega_i \times r_{ij}), \quad j \in I$$

用向量

$$R_{i,i-1} = r_{ii} - r_{i-1,i}$$

表示第 $i-1$ 个和第 i 个连杆之间的距离向量。从而, 确定了为建立动力学模型所需的所有运动学量。

第四步: 机构动力学

这一步是计算动力学各量 $\Delta_i = \{F_i, \mu_i\}$ 的值, 其中

$$F_i = \{F_i\}$$

$$\mu_i = \{M_i\}$$

式中

F_i ——第 i 个连杆质量中心的惯性力;

M_i ——第 i 个连杆的惯性力转矩。

可利用牛顿定律计算惯性力

$$F_i = -m_i \omega_i = [\alpha_{i1} \dots \alpha_{ii} 0 \dots 0] \ddot{q} + \alpha_i^0 \quad (2.2.9)$$

式中 m_i 为第 i 个连杆的质量。

把上式与式(2.2.7)进行比较,得到

$$\begin{aligned} \mathbf{a}_{ij} &= -m_i \beta_{ij}, \quad j \in I \\ \mathbf{a}_i^o &= -m_i \beta_i^o \end{aligned} \quad (2.2.10)$$

可由欧拉动力学方程确定惯性力转矩,并表示为

$$\mathbf{M}_i = [b_{i1} \cdots b_{in} 0 \cdots 0] + \mathbf{b}_i^o \quad (2.2.11)$$

式中

$$\begin{aligned} b_{ij} &= -T_i \alpha_{ij}, \quad j \in I \\ \mathbf{b}_i^o &= -T_i \mathbf{a}_i^o + \lambda_i \end{aligned} \quad (2.2.12)$$

其中

$$\begin{aligned} T_i &= \sum_{j=1}^3 Q_{ij} J_{ij} \\ Q_{ij} &= [q_{ij}^1 q_{ij}^2 q_{ij}^3] \\ \lambda_i &= \begin{bmatrix} (\omega_i \cdot \mathbf{q}_{i2})(\omega_i \cdot \mathbf{q}_{i3})(J_{i2} - J_{i3}) \\ (\omega_i \cdot \mathbf{q}_{i3})(\omega_i \cdot \mathbf{q}_{i1})(J_{i3} - J_{i1}) \\ (\omega_i \cdot \mathbf{q}_{i1})(\omega_i \cdot \mathbf{q}_{i2})(J_{i1} - J_{i2}) \end{bmatrix} \end{aligned} \quad (2.2.13)$$

$q_{ij}^k (j=1, 2, 3)$ 表示向量 \mathbf{q}_{ij} 的第 j 个分量。

除惯性力和转矩外,连杆还受到外部力 $\mathbf{G}_i, i \in N$ 和转矩 $\mathbf{M}_i^e, i \in N$ 的作用。于是,总力和总转矩可表示为

$$\begin{aligned} \mathbf{F}_i^e &= \mathbf{F}_i + \mathbf{G}_i \\ \mathbf{M}_i^e &= \mathbf{M}_i + \mathbf{M}_i^e \end{aligned} \quad (2.2.14)$$

利用刚体动力学一般定理(动量矩变分定理和重力中心运动定理),可获得机构运动微分方程。当第 i 个运动副是旋转副,则用第一个定理;当第 i 个运动副是滑动副,则用第二个定理。因而,对旋转运动副 ($\xi_i = 0$), 得到

$$P_i = -\mathbf{e}_i \cdot \sum_{k=1}^n \sum_{j=\max(1,k)}^n (\mathbf{b}_{ik} + \mathbf{r}_{ij} \times \mathbf{a}_{ik}) \ddot{q}_k$$

$$-e_i \cdot \sum_{j=i}^n [r_{ji} \times (\alpha_j^0 + G_j) + b_j^0] \quad (2.2.15)$$

式中 $P_i \in R$ 是第 i 个执行机构的驱动转矩，对滑动运动副 ($\xi_i = 1$)，得到

$$P_i = -e_i \sum_{k=1}^n \sum_{j=m \times (i,k)}^n a_{jk} \ddot{q}_k - e_i \sum_{k=1}^n \sum_{j=1}^n (\alpha_j^0 + G_j) \quad (2.2.16)$$

可把表示式(2.2.15)和(2.2.16)写为矩阵形式：

$$P = H(q, \theta) \ddot{q} + h(q, \dot{q}, \theta) \quad (2.2.17)$$

式中

P ——驱动转矩(力)向量， $P = [P_1 \cdots P_n]^T$ ；

q ——关节坐标向量， $q = [q_1 \cdots q_n]^T$ ；

θ ——几何和动力学参数向量， $\theta = [\theta_1 \cdots \theta_m]^T$ ；

$H(q, \theta)$ —— $R^n \times R^n \rightarrow R^{n \times n}$ 系统矩阵(向量)；

$h(q, \dot{q}, \theta)$ —— $R^n \times R^n \times R^n \rightarrow R^n$ 系统矩阵(向量)。

按照式(2.2.15)和(2.2.16)，有

$$H(q, \theta) = [H_{ik}]$$

$$= \left[-e_i \sum_{j=m \times (i,k)}^n ((b_{jk} + r_{ji} \times a_{jk}) \xi_j + a_{jk} \xi_i) \right]$$

$$h(q, \dot{q}, \theta) = [h_i]$$

$$= \left[-e_i \cdot \sum_{j=i}^n ((r_{ji} \times (\alpha_j^0 + G_j) + b_j^0) \xi_j + (\alpha_j^0 + G_j) \xi_i) \right]$$

让我们注意，向量 e_i 和 r_{ji} 包含几何参数，而向量 b_{jk} ， b_j^0 ， α_j^0 和 G_j 既包含几何参数，又包含动力学参数（连杆的质量和惯性矩）。由式(2.2.18)和所有上述关系式可以推定，矩阵

函数 $H(q, \theta)$ 和 $h(q, \dot{q}, \theta)$ 是广义坐标 q 和速度 \dot{q} 的复杂函数,也是自变量的非线性函数,这些结果来源于连杆的动力学交互作用。实际上向量 θ 可认为是常数。

2.3 闭式动力学模型

考虑机器人机构模型,设该机构是由一系列通过旋转或滑动关节互相联接的刚体组成。分析问题时,我们继续使用上一节引入的符号(构形、连杆、运动副和运动链等)。

问题的提出

我们研究建立机器人闭式动力学模型 P 的算法:

$$P = H(q, \theta)\ddot{q} + \dot{q}^T C(q, \theta)\dot{q} + h_g(q, \theta) \quad (2.3.1)$$

式中

$H(q, \theta)$ ——系统的惯性矩阵, $R^n \times R^m \rightarrow R^{n \times n}$;

$C(q, \theta)$ ——系统的“C-矩阵”, $R^n \times R^m \rightarrow R^{n \times n \times n}$;

$h_g(q, \theta)$ ——重力向量, $R^n \times R^m \rightarrow R^n$ 。

其中 C-矩阵考虑了哥氏和离心效应,并表示为 n 矩阵的集合 $[C^1(q, \theta), \dots, C^n(q, \theta)]$, $C^i(q, \theta) \in R^{n \times n}$, 使得下式成立:

$$\dot{q}^T C(q, \theta)\dot{q} = \begin{bmatrix} \dot{q}^T C^1(q, \theta)\dot{q} \\ \vdots \\ \dot{q}^T C^n(q, \theta)\dot{q} \end{bmatrix} \quad (2.3.2)$$

系统表示式(2.3.1)中的所有矩阵都是参数/变量集合

$$K_i = (Q_i, R_i, E_i), \quad i \in N$$

$$D_i = (m_i, J_i), \quad i \in N$$

的显含函数。式中

$$J_i = (J_{i1}, J_{i2}, J_{i3})$$

表示主惯性轴惯性矩的集合。其余参数/变量在前一节已经定义。

我们要导出模型矩阵式(2.3.1)的闭式表示式。

问题的解

在研究算法中,除前一节引入的符号外,我们还要使用下列符号:

$\overline{(i_1, \dots, i_k)} = \max(i_1, \dots, i_k)$ —— i_1, \dots, i_k 的最大数;

$\eta_{ik} = \begin{cases} 1, & i > k \\ 0, & i \leq k \end{cases}$ —— 两个指符的整数函数;

$\eta_{ik} = 1 - \eta_{ik}$;

\bar{R}_{ik} —— 对 $i > k$ 定义的函数;

E_{ik} —— 对 $i \leq k$ 定义的函数。

让我们假设,所有连杆的位置已经确定,也就是,已知集合

$$K_i = (Q_i, R_i, E_i)$$

的所有元素值。我们首先导出惯性矩阵 $H(q, \theta)$ 的非递推形式。为此目的,我们考虑基本牛顿-欧拉法第三步中定义的运动学参数的集合

$$\mathcal{L}_i = \{Q_i, W_i\}$$

让我们注意集合 \mathcal{L}_i 的所有元素,它们涉及到惯性矩阵,也就是元素 α_{ij} 和 β_{ij} 的推导。按照式(2.2.8),对 β_{ij} , $j \in \{1, \dots, i-1\}$ 的递推表示式求和,得到

$$\beta_{ij} = \beta_{ij} + e_j \times \sum_{l=j+1}^i R_{l,l-1} \bar{E}_j \quad (2.3.3)$$

式中求和符号是对 $j+1$ 到 i 范围内的指符集 l 进行的。显

然,下式成立:

$$\sum_{i=j+1}^n k_{i,i-1} = k_{jj}$$

式中 R_{ij} 表示连杆 j 质量中心到连杆 i 质量中心的距离向量。考虑式(2.2.8)中 β_{ij} 的表示式,并把它代入式(2.3.3),得到

$$\beta_{ij} = (\mathbf{e}_j \times \mathbf{r}_{ij}) \xi_j + \mathbf{e}_j \xi_j, j \in I \quad (2.3.4)$$

推导中利用了关系式

$$\mathbf{r}_{ij} + \mathbf{R}_{ij} = \mathbf{r}_{ii}$$

我们引入缩写符号 $\langle \mathbf{e}, \mathbf{r} \rangle_{ij}$ 来代替方程式(2.3.4)中右边的表示式。于是,式(2.3.4)变为

$$\beta_{ij} = \langle \mathbf{e}, \mathbf{r} \rangle_{ij} \quad (2.3.5)$$

让我们考虑牛顿-欧拉法第四步中定义的动力学参数集合:

$$\Delta_i = \{P_i, \mu_i\}$$

按照式(2.2.10),系数 $\mathbf{a}_{ij} \in P_i$ 有如下形式:

$$\mathbf{a}_{ij} = -m_j \langle \mathbf{e}, \mathbf{r} \rangle_{ij}, j \in I \quad (2.3.6)$$

由式(2.2.12)中第一个关系式和式(2.2.13)中前两个关系式给出的系数 $\mathbf{b}_{ij} \in \mu_i$ 可表示为

$$\mathbf{b}_{ij} = -\sum_{\mu=1}^3 (Q_{i\mu} \mathbf{e}_j) J_{i\mu} \xi_j \quad (2.3.7)$$

把式(2.3.6)和(2.3.7)代入式(2.2.18)中的惯性矩阵表示式,并利用恒等式

$$\mathbf{e}_i (Q_{i\mu} \mathbf{e}_k) = (\mathbf{e}_i \cdot \mathbf{q}_{j\mu}) (\mathbf{e}_k \cdot \mathbf{q}_{i\mu}) \quad (2.3.8)$$

得到

$$H_{ik} = \sum_{j=(i,k)}^n [m_j \langle \mathbf{e}, \mathbf{r} \rangle_{ij} \cdot \langle \mathbf{e}, \mathbf{r} \rangle_{jk}]$$

$$+ \sum_{\mu=1}^3 (\mathbf{e}_i \cdot \mathbf{q}_{i\mu})(\mathbf{e}_k \cdot \mathbf{q}_{i\mu}) J_{i\mu} \xi_i \xi_k] \quad (2.3.9)$$

式中 H_{ik} 就是惯性矩阵 $H(q, \theta)$ 第 (i, k) 个元素。

按照问题的提出，我们还要确定由式(2.3.1)和(2.3.2)定义的哥氏和离心效应矩阵 $C(q, \theta)$ 。假设已知参数/变量集合 K_i 和 $D_i, \forall i \in N$ ，让我们把集合

$$\mathcal{L}_i = \{O_i, W_i\}$$

的元素由递推形式变为非递推形式。例如，通过对 $\omega_j \in O_j$ ，指符集为 $j \in I$ 的关系式(2.2.6)求和，得到

$$\omega_i = \sum_{j=1}^i \mathbf{e}_j \xi_j \dot{q}_j \quad (2.3.10)$$

现在让我们考虑元素 $\alpha_i^0 \in W_i$ 和 $\beta_i^0 \in W_i$ 。按照式(2.2.8)，对 $\alpha_i^0, j \in I$ 的关系式求和，得到

$$\alpha_i^0 = \sum_{j=1}^i (\omega_{j-1} \times \mathbf{e}_j) \xi_j \dot{q}_j \quad (2.3.11)$$

把式(2.3.10)代入式(2.3.11)，得到结果

$$\alpha_i^0 = \sum_{k=1}^i \sum_{l=1}^{k-1} \mathbf{e}_{ik} \xi_k \xi_l \dot{q}_k \dot{q}_l$$

式中

$$\mathbf{e}_{ik} = \mathbf{e}_i \times \mathbf{e}_k$$

我们用 P_{ik} 表示向量 $\mathbf{e}_{ik} \xi_k \xi_l \dot{q}_k \dot{q}_l$ 并考虑恒等式

$$\begin{aligned} \sum_{k=1}^i \sum_{l=1}^{k-1} P_{ik} &= \sum_{k=1}^i \sum_{l=1}^i P_{ik} \eta_{kl} \\ &= \sum_{k=1}^i \sum_{l=1}^i \frac{1}{2} (P_{ik} \eta_{kl} + P_{li} \eta_{lk}) \end{aligned}$$

让我们注意，最后双重求和的加数对指符集 k 和 l 是对称的。

考虑到

$$\epsilon_{ik} = -\epsilon_{ki}$$

得到

$$\alpha_i^0 = \sum_{k=1}^i \sum_{l=1}^i \frac{1}{2} \epsilon_{ik} (\eta_{kl} - \eta_{lk}) \xi_k \xi_l \dot{q}_k \dot{q}_l \quad (2.3.12)$$

很明显,式(2.3.12)可表示为

$$\alpha_i^0 = \dot{q}^T C_i^0 \dot{q} \quad (2.3.13)$$

式中 C_i^0 是 $n \times n$ 对称矩阵:

$$C_i^0 = \frac{1}{2} \begin{bmatrix} 0 & \epsilon_{12} \xi_1 \xi_2 & \epsilon_{13} \xi_1 \xi_3 & \cdots & \epsilon_{1i} \xi_1 \xi_i \\ \epsilon_{12} \xi_1 \xi_2 & 0 & \epsilon_{23} \xi_2 \xi_3 & \cdots & \epsilon_{2i} \xi_2 \xi_i \\ \vdots & & & & \\ \epsilon_{1j} \xi_1 \xi_j & \epsilon_{2j} \xi_2 \xi_j & \epsilon_{3j} \xi_3 \xi_j & \cdots & 0 \\ \cdots & & & & \\ 0 & & & & 0 \end{bmatrix}$$

我们看到,表示式(2.3.13)的形式与定义系统 C-矩阵的式(2.3.2)的形式是相同的。

像处理 ω_i 的式(2.3.10)一样,对 β_i^0 的关系式(2.2.8),在域 $j \in I$ 内求和,并考虑到 α_i^0 的表示式(2.3.12),得到

$$\begin{aligned} \beta_i^0 = & \sum_{l=1}^i \left\{ \sum_{k=1}^{l-1} \sum_{l=1}^{k-1} \epsilon_{lk} \times R_{l,j-1} \xi_k \xi_l \dot{q}_k \dot{q}_l \right. \\ & + \sum_{k=1}^{j-1} (\epsilon_{kj} \times r_{jj} \xi_j + 2\epsilon_{kj} \xi_j) \xi_k \dot{q}_j \dot{q}_k \\ & + \sum_{k=1}^i \sum_{l=1}^i \epsilon_k \times (\epsilon_l \times r_{jj}) \xi_k \xi_l \dot{q}_k \dot{q}_l \\ & \left. - \sum_{k=1}^{j-1} \sum_{l=1}^{j-1} \epsilon_k \times (\epsilon_l \times r_{j-1,l}) \xi_k \xi_l \dot{q}_k \dot{q}_l \right\} \end{aligned}$$

因为获得尽量最简单形式的简化表示式是相当复杂的, 附录 2.1 中描述了这种方法, 这里只给出最终结果

$$\beta_i^0 = \sum_{k=1}^i \sum_{l=1}^i (\xi_l \mathbf{e}_l \times \langle \mathbf{e}, \mathbf{r} \rangle_{ik} \eta_{kl} + \xi_k \mathbf{e}_k \times \langle \mathbf{e}, \mathbf{r} \rangle_{il} \eta_{lk}) \dot{q}_k \dot{q}_l \quad (2.3.14)$$

这个表示式也可以写为

$$\beta_i^0 = \dot{q}^T C_i^0 \dot{q} \quad (2.3.15)$$

式中 C_i^0 是 $n \times n$ 对称矩阵, 按照式(2.3.14), C_i^0 矩阵的元素 $(k, l), k \geq l$ 给定为

$$C_i^0(k, l) = (\mathbf{e}_l \times \langle \mathbf{e}, \mathbf{r} \rangle_{ik}) \xi_l \eta_{lk} \quad (2.3.16)$$

在式(2.3.13)和(2.3.15)基础上, 我们得到结论, 运动学系数 α_i^0 和 β_i^0 只参与构造 C -矩阵。但是, 这些系数不能完全确定 C -矩阵, 还要考虑动力学参数/变量集合

$$\Delta_i = \{F_i, \mu_i\}$$

的元素。因此, 按照式(2.2.10)的第二个关系式, 对 $\alpha_i^0 \in F_i$, 得到

$$\alpha_i^0 = \dot{q}^T C_i^1 \dot{q} \quad (2.3.17)$$

式中 $C_i^1 \in R^{n \times n}$ 是对称矩阵, 它的元素 $(k, l), k \geq l$ 由

$$C_i^1(k, l) = -m_i (\mathbf{e}_l \times \langle \mathbf{e}, \mathbf{r} \rangle_{ik}) \xi_l \eta_{lk} \quad (2.3.18)$$

确定。

现在让我们考虑关系式(2.2.11)–(2.2.13), 以确定 $b_i^0 \in \mu_i$ 。把式(2.2.13)中 T_i 的表示式和式(2.3.12)中 α_i^0 的非递推式代入式(2.2.12)中 b_i^0 的表示式, 得到

$$b_i^0 = - \sum_{k=1}^i \sum_{\mu=1}^{k-1} \sum_{l=1}^k Q_{i\mu} \mathbf{e}_{lk} J_{i\mu} \xi_k \xi_l \dot{q}_k \dot{q}_l + \lambda_i \quad (2.3.19)$$

表示式 $Q_{i\mu} \mathbf{e}_{lk}$ 可用向量 $\mathbf{q}_{i\mu}$ 和 \mathbf{e}_{lk} 表示为

$$Q_{i\mu} \mathbf{e}_{lk} = (\mathbf{e}_{lk} \cdot \mathbf{q}_{i\mu}) \mathbf{q}_{i\mu} \quad (2.3.20)$$

式中 $Q_{i\mu}$ 由式(2.2.13)给出。式(2.3.19)中向量 λ_i 由式

(2.2.13)确定, 把式(2.3.10)中 ω_i 的非递推式和转换矩阵

$$Q_i = [q_{i1} q_{i2} q_{i3}]$$

代入式(2.2.13)中 λ_i 的表示式, 得到

$$\lambda_i = \sum_{k=1}^i \sum_{l=1}^i \lambda_{kl}^i q_k q_l \quad (2.3.21)$$

式中

$$\lambda_{kl}^i = \sum_{\mu=1}^3 (e_k \cdot q_{i[\mu+1]})(e_l \cdot q_{i[\mu+1]}) q_{i\mu} (J_{i[\mu+1]} - J_{i[\mu+2]}) \xi_k \xi_l \quad (2.3.22)$$

其中 $[\mu]$ 表示整数变量 $\mu \in \{1, \dots, 5\}$ 的函数, 定义为

$$[\mu] = \mu \bmod 3 = \begin{cases} \mu, & 1 \leq \mu \leq 3 \\ \mu - 3, & 4 \leq \mu \leq 5 \end{cases}$$

通过恒等变换(附录 2.2), 式(2.3.21)和(2.3.22)可简化为

$$\lambda_i = \dot{q}^T C_i^i \dot{q} \quad (2.3.23)$$

式中 C_i^i 是 $n \times n$ 对称阵, 对 $k \geq l$, 它的元素由

$$C_i^i(k, l) = -\frac{1}{2} \sum_{\mu=1}^3 [(e_k \cdot q_{i\mu})(e_l \times q_{i\mu}) + (e_l \cdot q_{i\mu})(e_k \times q_{i\mu})] J_{i\mu} \eta_{ik} \xi_k \xi_l \quad (2.3.24)$$

确定。把式(2.3.23)和(2.3.24)代入式(2.3.19), 得到

$$b_i^0 = \dot{q}^T C_i^i \dot{q} \quad (2.3.25)$$

式中 C_i^i 是 $n \times n$ 对称矩阵, 它的元素 (k, l) , $k \geq l$ 给定为(附录 2.2)

$$C_i^i(k, l) = -\frac{1}{2} \sum_{\mu=1}^3 [(e_{lk} \cdot q_{i\mu}) q_{i\mu} + (e_k \cdot q_{i\mu})(e_l \times q_{i\mu}) + (e_l \cdot q_{i\mu})(e_k \times q_{i\mu})] J_{i\mu} \eta_{lk} \xi_k \xi_l \quad (2.3.26)$$

在上一节已经证明, 利用刚体动力学基本定理, 由式

(2.2.18) 可得到数学模型矩阵。按照式(2.2.18)，向量 $h(q, \dot{q}, \theta)$ 的每个元素 h_i 可表示为

$$h_i = h_i^c + h_i^g \quad (2.3.27)$$

式中

$$h_i^c = -e_i \cdot \sum_{j=1}^n [(r_{ji} \times a_j^c + b_j^c) \xi_j + a_j^c \xi_j] \quad (2.3.28)$$

$$h_i^g = -e_i \cdot \sum_{j=1}^n (r_{ji} \times G_j \xi_j + G_j \xi_j) \quad (2.3.29)$$

把式(2.3.17)和(2.3.25)代入(2.3.28)，得到

$$h_i^c = \dot{q}^T C^i \dot{q} \quad (2.3.30)$$

式中 $C^i \in R^{n \times n}$ 给定为

$$C^i = - \sum_{j=1}^n \{ [(e_i \times r_{ji}) \xi_j + e_i \xi_j] \cdot C_j^i + e_i \cdot C_j^i \xi_j \} \quad (2.3.31)$$

其中向量与矩阵(它的元素也是向量)之间的点积运算可简化为向量与该矩阵每个元素之间的点积运算。很明显，矩阵 C^i 表示 C -矩阵的第 i 个分量(见式(2.3.2))。

现在，我们把式(2.3.18)中 C_j^i 和式(2.3.26)中 C_j^i 代入方程式(2.3.31)右边的表示式。对称矩阵 C^i 的元素 (k, l) ， $k \geq l$ 由

$$\begin{aligned} C_{kl}^i = & \sum_{j=(l, k)}^n \{ m_j \langle e, r \rangle_{ji} \cdot (e_i \times \langle e, r \rangle_{jk}) \\ & + \frac{1}{2} \sum_{\mu=1}^3 [(e_i \cdot q_{j\mu}) e_{lk} + (e_k \cdot q_{j\mu}) e_{il} \\ & + (e_l \cdot q_{j\mu}) e_{ik}] \cdot q_{j\mu} J_{j\mu} \xi_j \xi_k \} \xi_j \quad (2.3.32) \end{aligned}$$

得到。从而确定了由式(2.3.1)和(2.3.2)定义的科里奥利效应

和离心效应矩阵 $C(q, \theta)$ 。

按照问题的提出，尚需确定重力向量。这个向量的元素由表示式(2.3.29)给定，等价于

$$h_i^g = - \sum_{j=1}^n \{ [(\mathbf{e}_i \times \mathbf{r}_{ij}) \xi_j + \mathbf{e}_i \xi_j] \cdot \mathbf{G}_j \}$$

也就是

$$h_i^g = - \sum_{j=1}^n (\langle \mathbf{e}_i, \mathbf{r} \rangle_{ij} \cdot \mathbf{G}_j) \quad (2.3.33)$$

实际上，这是问题的解，因为 h_i^g 可用集合

$$K_i = (Q_i, R_i, E_i)$$

和

$$D_i = (m_i, J_i)$$

的元素表示。向量 \mathbf{G}_j 可表示为

$$\mathbf{G}_j = m_j \mathbf{g}$$

式中 \mathbf{g} 是参考坐标系给出的重力加速度常数向量 ($|\mathbf{g}| = 9.81 \text{ m/s}^2$)。

到此，提出的问题已经解决，也就是，对简单运动链描述其结构的机构，我们已经导出了为建立它的数学模型矩阵所需的所有非递推关系式。我们曾假设，存在两种类型关节：滑动和旋转关节。但是，这不是限制条件，因为任意类型的关节联接都可用一系列滑动和旋转关节以及忽略大小和质量的虚拟连杆来实现。

图 2.6 给出了刚分析过的算法。运动学参数/变量集合

$$K_i = (Q_i, R_i, E_i)$$

利用牛顿-欧拉法的运动学步骤(确定机构基始和实际位置的步骤)确定。为构造操作手运动学模型，代之以其它方法也是

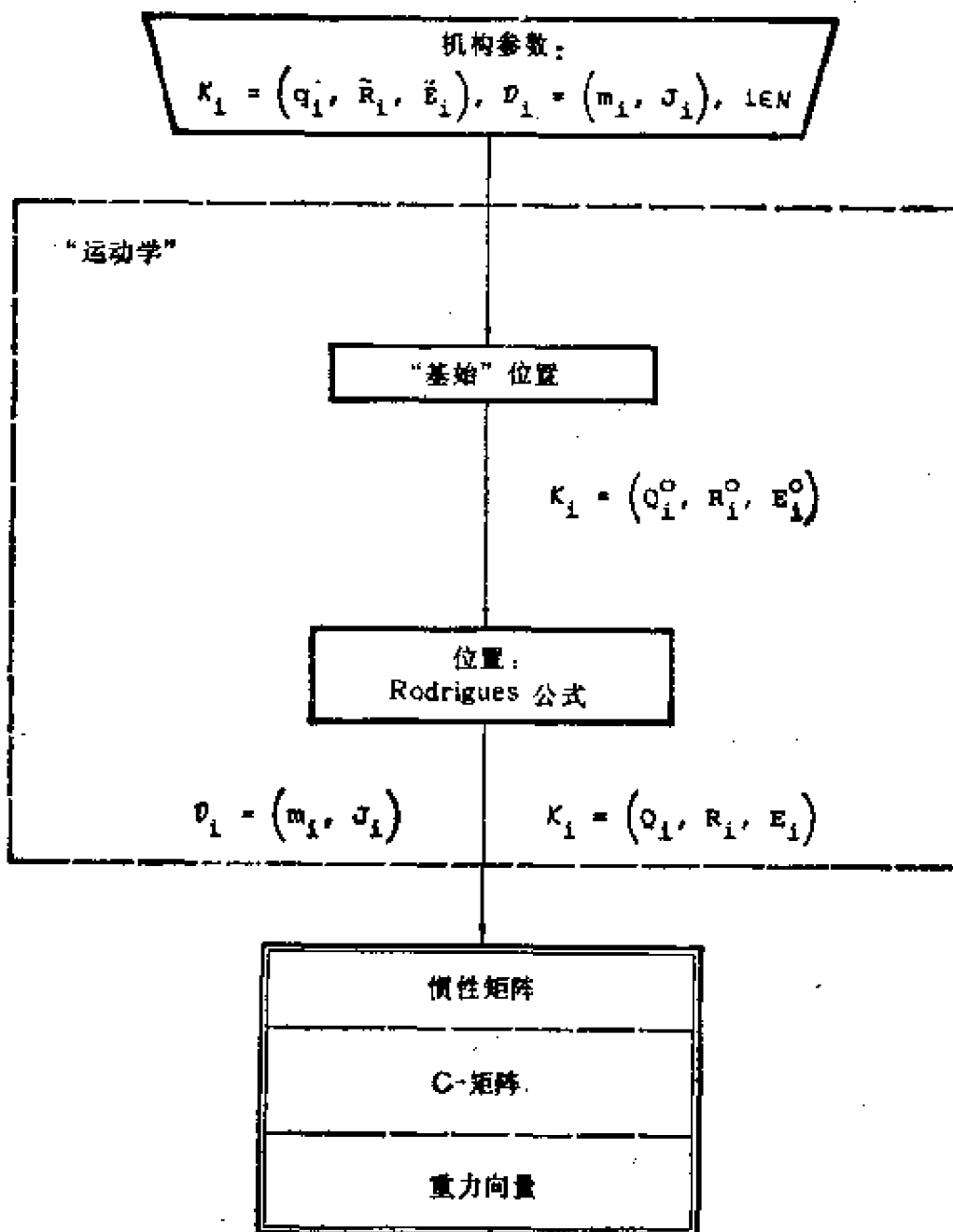


图 2.6 建立闭式动力学模型矩阵算法的总示意图

可能的,例如,基于 Denavit-Hartenberg 矩阵的方法^[10],等等。图 2.7 中详细给出了全部完整算法。

输入量

n —— 自由度数

$q = (q_1, \dots, q_n)$ —— 关节坐标向量

$\xi_i = \begin{cases} 0 & \text{—— 关节 } i \text{ 是旋转关节} \\ 1 & \text{—— 关节 } i \text{ 是滑动关节} \end{cases}$

r_{ii} —— 连杆 i 坐标系表示的, 由关节 i 到连杆 i 质量中心的距离向量

$r_{i,i+1}$ —— Q_i 坐标表示的, 由关节 $i+1$ 到连杆 i 质量中心的距离向量

e_i —— 关节 i 的轴对 Q_i 的单位向量

m_i —— 连杆 i 的质量

J_{i1}, J_{i2}, J_{i3} —— 连杆 i 对主惯性轴的惯性矩

r_{01} —— 第 1 个关节到参考坐标系原点的距离向量

e_1 —— 用参考系表示的第 1 个关节轴的单位向量

(指符集 $i \in \{1, \dots, n\}$)

基始位置

对 $i \in \{1, \dots, n\}$:

$$\tilde{R}_i = e_i \times (r_{ii} \times e_i), \quad \tilde{a}_i = \tilde{R}_i / |\tilde{R}_i|$$

$$R_i = e_i^* \times (r_{i,i+1}^* \times e_i^*), \quad a_i^* = -R_i / |R_i|$$

$$\tilde{b}_i = e_i \times \tilde{a}_i, \quad b_i^* = e_i^* \times a_i^*$$

$$Q_i^* = [a_i^* \ b_i^* \ a_{i2}^*] = [a_i^* \ b_i^*] [a_i \ \tilde{b}_i]^T$$

$$r_{i1}^* = Q_i^* r_{ii}, \quad r_{i,i+1}^* = Q_i^* r_{i,i+1}$$

$$e_{i+1}^* = Q_i^* e_{i+1}$$

机构位置

对 $i \in \{1, \dots, n\}$:

$$q_{ij} = \begin{bmatrix} e_i \times (a_i^* \times e_i) \tilde{E}_i \\ (e_i \times a_i^*) \tilde{E}_i \\ (e_i \cdot a_i^*) e_i \tilde{E}_i + a_i^* \tilde{E}_i \end{bmatrix}^T \begin{bmatrix} \cos q_i \\ \sin q_i \\ 1 \end{bmatrix}, \quad j = 1, 2, 3$$

$$Q_i = [q_{i1} \ q_{i2} \ q_{i3}]$$

$$r_{ii} = Q_i r_{ii} + q_{i1} e_i \tilde{E}_i, \quad r_{i,i+1} = Q_i r_{i,i+1}$$

$$e_{i+1} = Q_i e_{i+1}$$

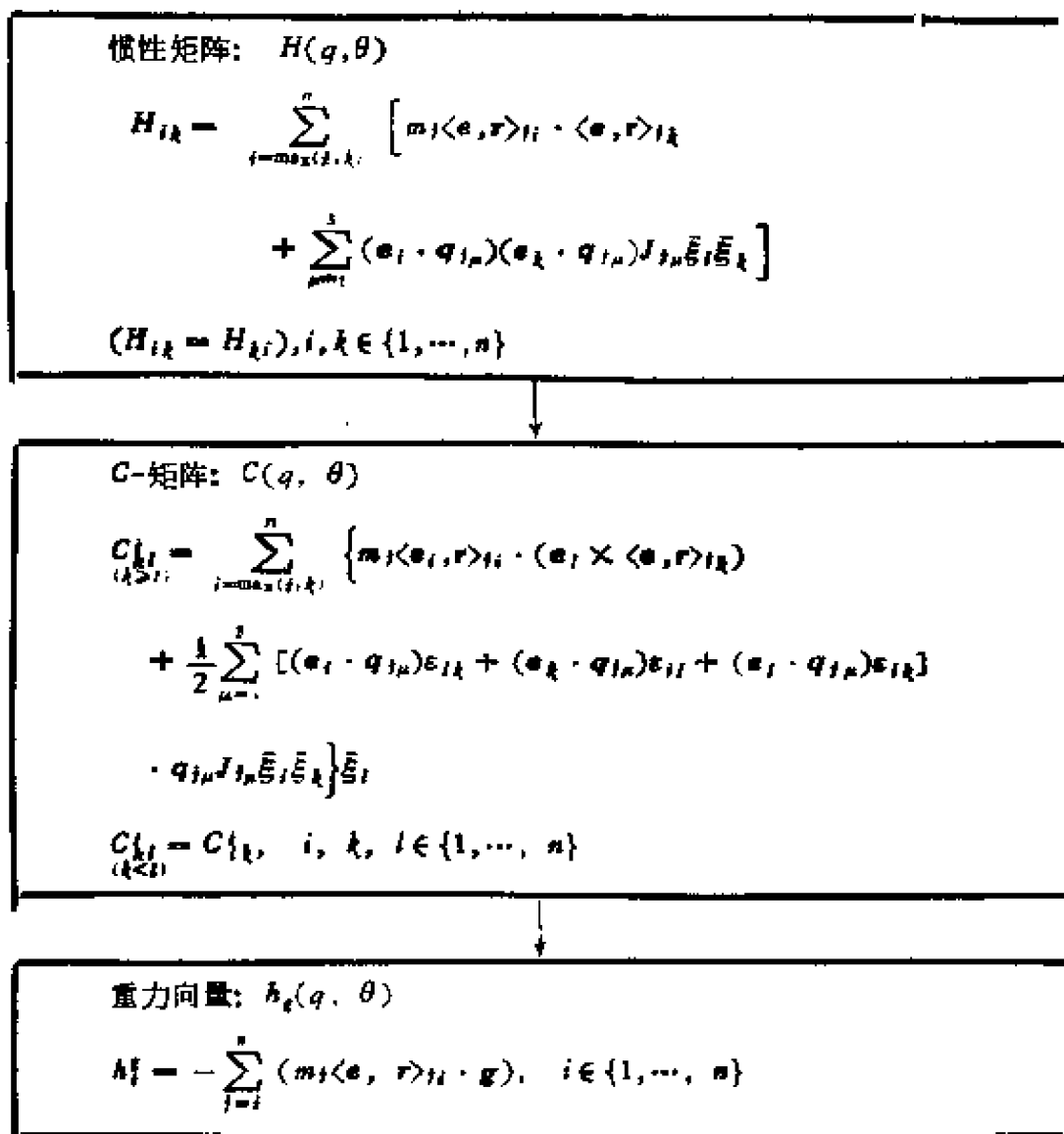


图 2.7 建立闭式动力学模型矩阵算法的流程图

$$\langle e, r \rangle \equiv e_i \times r_i \bar{e}_i + e_j \bar{e}_j, e_i \equiv e_i \times e_i$$

2.4 动力学模型矩阵特性: 对称性、正定性和反对称性

动力学模型式 (2.3.1) 和 (2.3.2) 中的矩阵 $H(q, \theta)$ 和 $C^j(q, \theta)$ 的特性可用下列定理系统化:

定理 1

动力学模型式(2.3.1)中惯性矩阵 $H(q, \theta)$ 是对称的和正定的。

证明

按照分析过的方法, 矩阵 $H(q, \theta)$ 的元素 (i, k) 由式(2.3.9)给出。从而, 元素 (k, i) 是

$$H_{ki} = \sum_{j=\overline{(k,i)}}^n \left[m_j \langle e, r \rangle_{ik} \cdot \langle e, r \rangle_{ij} + \sum_{\mu=1}^3 (e_k \cdot q_{j\mu})(e_i \cdot q_{j\mu}) J_{j\mu} E_k E_i \right]$$

由于

$$\overline{(i,k)} = \overline{(k,i)}$$

因此可直接得到

$$H_{ki} = H_{ik}, \quad i \in N, k \in N$$

这表示了惯性矩阵的对称性。

定理的第二部分要利用正定性的定义进行证明。

定义

矩阵 $H \in R^{n \times n}$ 是正定的, 当且仅当对每个 $x \neq 0, x \in R^n$, 必成立 $x^T H x > 0$ 。

让我们考虑惯性矩阵元素式(2.3.9)的构造。式(2.3.9)右边表示式的求和是对指标集 j 进行的, 而 j 的取值在极限值 $\overline{(i,k)} = \max(i, k)$ 和 n 的范围内。通过引入上一节定义的整值函数 η_{ij} 和 η_{ki} ($\eta_{ij} = 1, j \geq i; \eta_{ij} = 0$, 其它), 把求和范围延伸到从 1 到 n :

$$H_{ik} = \sum_{j=1}^n \left\{ m_j (\langle e, r \rangle_{ji} \eta_{ij}) \cdot (\langle e, r \rangle_{jk} \eta_{kj}) \right. \\ \left. + \sum_{\mu=1}^3 [(\mathbf{e}_i \cdot \mathbf{q}_{i\mu}) \eta_{ij} \xi_i] \cdot [(\mathbf{e}_k \cdot \mathbf{q}_{k\mu}) \eta_{kj} \xi_k] J_{i\mu} \right\} \quad (2.4.1)$$

引入下列符号:

$$P_{ij}^* = \sqrt{m_j} (\langle e, r \rangle_{ji} \eta_{ij}) \quad (2.4.2) \\ P_{ij}^* = \sqrt{J_{i\mu}} (\mathbf{e}_i \cdot \mathbf{q}_{i\mu}) \eta_{ij} \xi_i, \mu = 1, 2, 3$$

并把它们代入式(2.4.1), 得到

$$H_{ik} = \sum_{j=1}^n \sum_{\mu=0}^3 P_{ij}^* \cdot P_{kj}^* \quad (2.4.3)$$

由式(2.4.3)可见, 矩阵 $H(q, \theta)$ 可表示为

$$H = \sum_{j=1}^n \sum_{\mu=0}^3 \begin{bmatrix} P_{1j}^* \cdot P_{1j}^* \cdots P_{1j}^* \cdot P_{2j}^* \\ \vdots \\ P_{nj}^* \cdot P_{nj}^* \cdots P_{nj}^* \cdot P_{nj}^* \end{bmatrix}$$

或

$$H = \sum_{j=1}^n \sum_{\mu=0}^3 \begin{bmatrix} P_{1j}^* \\ \vdots \\ P_{nj}^* \end{bmatrix} \cdot [P_{1j}^* \cdots P_{nj}^*] \quad (2.4.4)$$

引入矩阵

$$H_j^* = [P_{1j}^* \cdots P_{nj}^*] \in R^{3 \times n}$$

则得到

$$H = \sum_{j=1}^n \sum_{\mu=0}^3 (H_j^*)^T H_j^* \quad (2.4.5)$$

让我们用 $x \in R^n$ 表示任意向量, 并使 $x \neq 0$, 构造二次型

$$\begin{aligned}
x^T H x &= x^T \left[\sum_{j=1}^n \sum_{k=0}^j (H_j^k)^T H_j^k \right] x \\
&= \sum_{j=1}^n \sum_{k=0}^j x^T (H_j^k)^T H_j^k x \\
&= \sum_{j=1}^n \sum_{k=0}^j (H_j^k x)^T (H_j^k x) \quad (2.4.6)
\end{aligned}$$

因为 $H_j^k x \in R^3$ 是一个向量, 并且 $x \neq 0$ 成立, 所以得到

$$(H_j^k x)^T (H_j^k x) > 0 \quad (2.4.7)$$

这就证明了 $x^T H x > 0$, 也就是, 惯性矩阵是正定的, 定理 1 全部得到证明。

定理 2

动力学模型式(2.3.1)中哥氏和离心效应矩阵 $C(q, \theta)$ 的子矩阵 $C^i(q, \theta)$, $i \in N$ 是

1) 对称的, 也就是

$$C_{kl}^i = C_{lk}^i, \quad k, l \in N$$

2) 对指符集 i 和 k 是反对称的, 也就是, 必成立

$$\begin{aligned}
C_{kl}^i &= -C_{li}^k, \quad k, i \geq l \\
C_{kl}^i &= 0, \quad i = k \geq l
\end{aligned} \quad (2.4.8)$$

证明

按照上一节的推导, 矩阵 $C_{kl}^i \in R^{n \times n}$ 和 $C_{li}^k \in R^{n \times n}$ 是对称的, 因此, 式(2.3.31)中的矩阵 $C^i \in R^{n \times n}$ 也是对称矩阵。

为证明特性 2), 让我们从表示式(2.3.32)开始, 并构造

$$C_{(kl)}^i = \sum_{j=(k,i)}^n \left\{ m_j \langle e, r \rangle_{jk} \cdot (e_i \times \langle e, r \rangle_{ji}) \right\}$$

$$\begin{aligned}
& + \frac{1}{2} \sum_{\mu=1}^i [(e_k \cdot q_{i\mu})e_{li} + (e_i \cdot q_{i\mu})e_{kl} \\
& + (e_i \cdot q_{i\mu})e_{ki}] \cdot q_{i\mu} J_{i\mu} \xi_i \xi_k \xi_i \} \xi_i \quad (2.4.9)
\end{aligned}$$

由于下列各式成立:

$$\langle e, r \rangle_{ik} \cdot (e_i \times \langle e, r \rangle_{ii}) = -\langle e, r \rangle_{ii} \cdot (e_i \times \langle e, r \rangle_{ik})$$

$$e_{ij} = -e_{ji}, \quad i, j \in N$$

$$\overline{(k, i)} = \overline{(i, k)}$$

因此通过把这些公式代入式(2.4.9), 并与式(2.3.32)进行比较, 就能直接得到反对称关系式(2.4.8). 如果 $i = k$, 可直接断定, 式(2.4.9)求和式等于零, 这就是定理 2 的证明.

这些定理是重要的, 因为这些定理描述了动力学模型矩阵的主要特性, 而且按照图 2.7 的流程图, 这些定理的应用也大大简化了构造非递推模型所需的运算次数. 为举例说明这一点, 我们通过应用对称性, 计算惯性矩阵元素的运算次数将由 n^2 减少到

$$n_H = \frac{n(n+1)}{2}$$

容易证明, 通过应用 C -矩阵的对称性和反对称性, 计算这些矩阵元素的运算次数将由 n^3 减少到

$$n_c = \frac{n(n^2-1)}{3}$$

例如, 对 $n = 6$, 运算次数由 216 次减少到 70 次, 也就是, 运算次数减少到原运算次数的 32.4%.

2.5 闭式线性化模型

n 个自由度的机器人操作手的动力学模型式(2.3.1)是一个对关节坐标向量 q 的 n 维复杂、非线性微分方程系统。另外,作为哥氏和离心力的结果,系统对变量 \dot{q} 也是非线性关系,这就是要引入计算机辅助设计机器人线性化模型的原因。

系统式(2.3.1)的全微分可表示为

$$\delta P = H(q, \theta) \delta \ddot{q} + 2[\dot{q}^T C(q, \theta)] \delta \dot{q} + D(q, \dot{q}, \ddot{q}, \theta) \delta q \quad (2.5.1)$$

式中

$$D = \frac{\partial H(q, \theta)}{\partial q} \ddot{q} + \dot{q}^T \frac{\partial C(q, \theta)}{\partial q} \dot{q} + \frac{\partial h_g(q, \theta)}{\partial q} \quad (2.5.2)$$

矩阵 $[\dot{q}^T C(q, \theta)] \in R^{n \times n}$ 有如下结构:

$$[\dot{q}^T C(q, \theta)] = \begin{bmatrix} \dot{q}^T C^1(q, \theta) \\ \vdots \\ \dot{q}^T C^n(q, \theta) \end{bmatrix} \quad (2.5.3)$$

构造线性化模型问题可表述如下:

问题的提出

我们要研究建立线性化模型矩阵 $D \in R^{n \times n}$ 的方法,这种方法是以 2.3 节介绍的构造闭式动力学模型算法为基础的。在建立偏导数矩阵

$$\frac{\partial H(q, \theta)}{\partial q}, \quad \frac{\partial C(q, \theta)}{\partial q}, \quad \frac{\partial h_g(q, \theta)}{\partial q} \quad (2.5.4)$$

时,我们要利用不产生累积误差,即数值微分法等等这样的运算。

问题的解

我们要利用在 2.3 节中给出的符号来研究算法:

$$N = \{1, \dots, n\}, I = \{1, \dots, i\}$$

$$\eta_{ij} = \begin{cases} 1, & i > j, \\ 0, & i \leq j, \end{cases} \quad i \in N, j \in N$$

$$\eta_{ji} = 1 - \eta_{ij}$$

我们要利用开运动链的某些特性来求解问题, 可把问题总结为下述三个定理:

定理 1

非合成向量 α_i (非合成向量 α_i 是这样—一个向量, 它对连杆 i 的局部坐标系是一个常向量) 对关节坐标 $q_s, s \in N$ 的偏导数给定为

$$\frac{\partial \alpha_i}{\partial q_s} = (\mathbf{e}_s \times \alpha_i) \xi_s \eta_{si} \quad (2.5.5)$$

式中 \mathbf{e}_s 是运动副 s 中关节轴的单位向量。

证明

当运动副 s 是旋转副 ($\xi_s = 0$) 和 $s \leq i$ ($\eta_{si} = 1$) 时, 则确定偏导数

$$\frac{\Delta \alpha_i}{\Delta q_s} = \lim_{\Delta q_s \rightarrow 0} \frac{\Delta \alpha_i}{\Delta q_s} \Big|_{\Delta q_j = 0, j \neq s}$$

的问题可归结为著名的理论力学基本旋转定理。按照这个定理, 如果只对轴 \mathbf{e}_s 发生旋转, 且 α_i 是非合成向量, 则

$$\lim_{\Delta q_s \rightarrow 0} \frac{\Delta \alpha_i}{\Delta q_s} = \mathbf{e}_s \times \alpha_i$$

如果 $s > i$, 则对轴 s 旋转时, 向量 α_i 不改变它的位置, 因此 α_i 对 q_s 的偏导数等于零。另外, 对滑动副情况 ($\xi_i = 1$), 向量 α_i 不变化, 也就是

$$\frac{\partial \alpha_i}{\partial q_s} = 0$$

这就是定理 1 的证明。

定理 2

向量 r_{ij} , $i \in N$, $i \geq j$ (r_{ij} 是联接第 j 个运动副的关节到第 i 个连杆重力中心的向量) 对关节坐标 $q_s, s \in N$ 的偏导数由表示式

$$\frac{\partial r_{ij}}{\partial q_s} = (\mathbf{e}_s \times \overline{r_{i,(j,s)}} \xi_s + \mathbf{e}_s \xi_s \eta_{ij}) \eta_{ij} \quad (2.5.6)$$

给出, 式中 \mathbf{e}_s 是第 s 个关节轴的单位向量。

证明

让我们考虑三种情况。

1) $s \leq j$. 在这种情况下, r_{ij} 表现为非合成向量, 因此可直接应用定理 1. 因为对 $s \leq j$, η_{ij} 的值等于零, 所以完成了证明。

2) $j < s \leq i$. 在这种情况下, 可把向量 r_{ij} 分解为两个分量

$$r_{ij} = r_{is} + R_{ij}$$

如图 2.8 所示。假设运动副 $\{C_{s-1}, C_s\}$ 是旋转的, 且 q_s 变化 Δq_s , 那么 R_{ij} 保持不变, 而向量 r_{is} 对轴 \mathbf{e}_s 旋转。因此, 仍能应用定理 1. 因为对 $s > j$, 有 $\overline{(j,s)} = s$, 所以式(2.5.6)可简化为定理 1 的对应表示式。

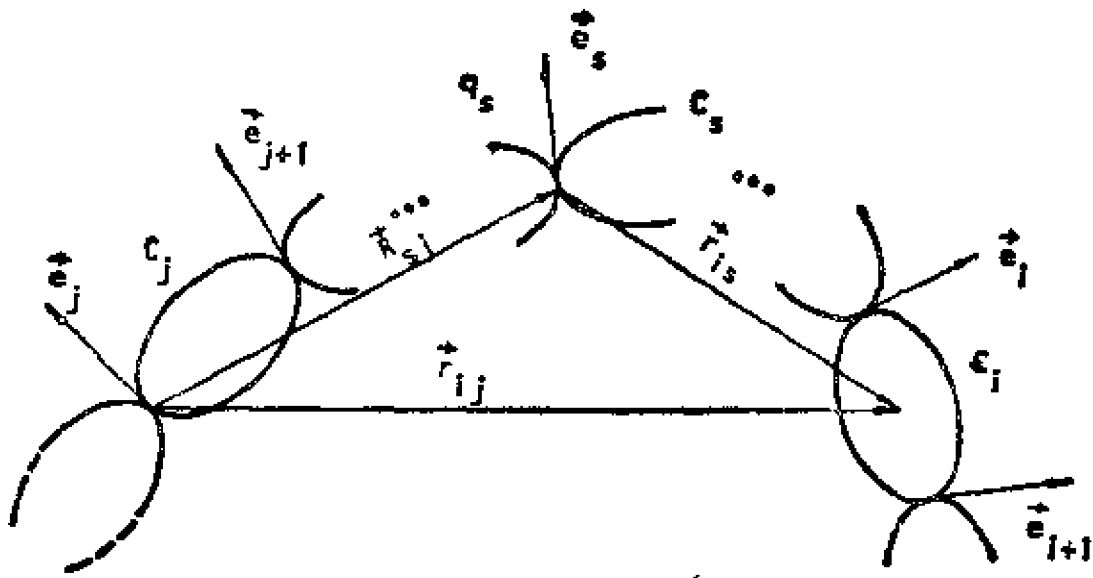


图 2.8 运动链的某些特征向量

如果运动副 \$(C_{s-1}, C_s)\$ 是滑动的, 对 \$q_s\$ 改变 \$\Delta q_s\$, 那么向量 \$r_{ij}\$ 保持不变, 而向量 \$R_{sj}\$ 改变 \$e_s \Delta q_s\$. 因此, 向量 \$r_{ij}\$ 对 \$q_s\$ 的偏导数简化为向量 \$e_s\$, 由式(2.5.6)给出.

3) \$s > i\$. 在这种情况下, \$q_s\$ 改变时, \$r_{ij}\$ 保持不变, 因此使 \$r_{ij}\$ 的偏导数等于零, 这刚好是式(2.5.6)得到的结果.

定理 2 完全得证.

定理 3

两个向量(可以是合成向量) \$\underline{a}_i\$ 和 \$\underline{b}_i\$, \$i \in N\$ 的叉积或点积的偏导数给定为

$$\frac{\partial}{\partial q_s} (\underline{a}_i \times \underline{b}_i) = \frac{\partial \underline{a}_i}{\partial q_s} \times \underline{b}_i + \underline{a}_i \times \frac{\partial \underline{b}_i}{\partial q_s}, \quad s \in N \quad (2.5.7)$$

因为这是向量分析的基本定理, 在此不再证明.

为了建立线性化模型式(2.5.1)和(2.5.2), 让我们考虑在

2.3 节中(图 2.7)得到的、动力学模型矩阵的非递推形式。惯性矩阵、C-矩阵和重力向量的表示式包括下列元素:

$$\langle \mathbf{e}, \mathbf{r} \rangle_{ij} = \mathbf{e}_i \times \mathbf{r}_{ij} \xi_i + \mathbf{e}_j \xi_i, \quad i \geq j \quad (2.5.8)$$

$$\mathbf{e}_{ij} = \mathbf{e}_i \times \mathbf{e}_j, \quad j \geq i \quad (2.5.9)$$

和标量

$$P_{ij\mu} = \mathbf{e}_i \cdot \mathbf{q}_{j\mu}, \quad i \geq j \quad (2.5.10)$$

对式(2.5.8)先应用定理 3, 再应用定理 1 和定理 2, 得到

$$\begin{aligned} \langle \mathbf{e}, \mathbf{r} \rangle_{ij}^{(s)} &= (\mathbf{e}_i^{(s)} \times \mathbf{r}_{ij} + \mathbf{e}_j \times \mathbf{r}_{ij}^{(s)}) \xi_i + \mathbf{e}_j^{(s)} \xi_i \\ &= \{(\mathbf{e}_{ij} \times \mathbf{r}_{ij} + \mathbf{e}_{ij}) \eta_{ij} \xi_i + [\mathbf{e}_j \\ &\quad \times (\mathbf{e}_i \times \mathbf{r}_{i,(\overline{j,i})}) \xi_i + \mathbf{e}_{j,i} \xi_i \eta_{ij}] \eta_{ij}\} \xi_i \quad (2.5.11) \end{aligned}$$

式中 $(\cdot)^{(s)}$ 表示变量 (\cdot) 对 q_s 的偏导数。如下表示式

$$\langle \mathbf{e}, \mathbf{r} \rangle_{ij}^{(s)} = \begin{cases} \mathbf{e}_i \times (\mathbf{e}_j \times \mathbf{r}_{ij} + \mathbf{e}_j) \xi_i \xi_j, & s \leq j \\ \mathbf{e}_j \times \langle \mathbf{e}, \mathbf{r} \rangle_{ij} \xi_j, & j < s \leq i \\ 0, & s > i \end{cases} \quad (2.5.12)$$

等价于表示式(2.5.11)。对式(2.5.9)应用定理 3 和定理 1, 得到

$$\begin{aligned} \mathbf{e}_{ij}^{(s)} &= \mathbf{e}_i^{(s)} \times \mathbf{e}_j + \mathbf{e}_i \times \mathbf{e}_j^{(s)} \\ &= (\mathbf{e}_{ii} \times \mathbf{e}_j \eta_{ii} + \mathbf{e}_i \times \mathbf{e}_{ij} \eta_{ij}) \xi_i \quad (2.5.13) \end{aligned}$$

或

$$\mathbf{e}_{ij}^{(s)} = \begin{cases} \mathbf{e}_i \times \mathbf{e}_{ij} \xi_i, & s \leq i \\ \mathbf{e}_i \times \mathbf{e}_{ij} \xi_i, & i < s \leq j \\ 0, & s > j \end{cases} \quad (2.5.14)$$

用同样方法, 由式(2.5.10)得到

$$\begin{aligned} P_{ij\mu}^{(s)} &= \mathbf{e}_i^{(s)} \cdot \mathbf{q}_{j\mu} + \mathbf{e}_i \cdot \mathbf{q}_{j\mu}^{(s)} \\ &= \mathbf{q}_{j\mu} \cdot \mathbf{e}_{ii} (\eta_{ii} - \eta_{ij}) \xi_i \quad (2.5.15) \end{aligned}$$

现在让我们利用定理 3 得到惯性矩阵元素的偏导数

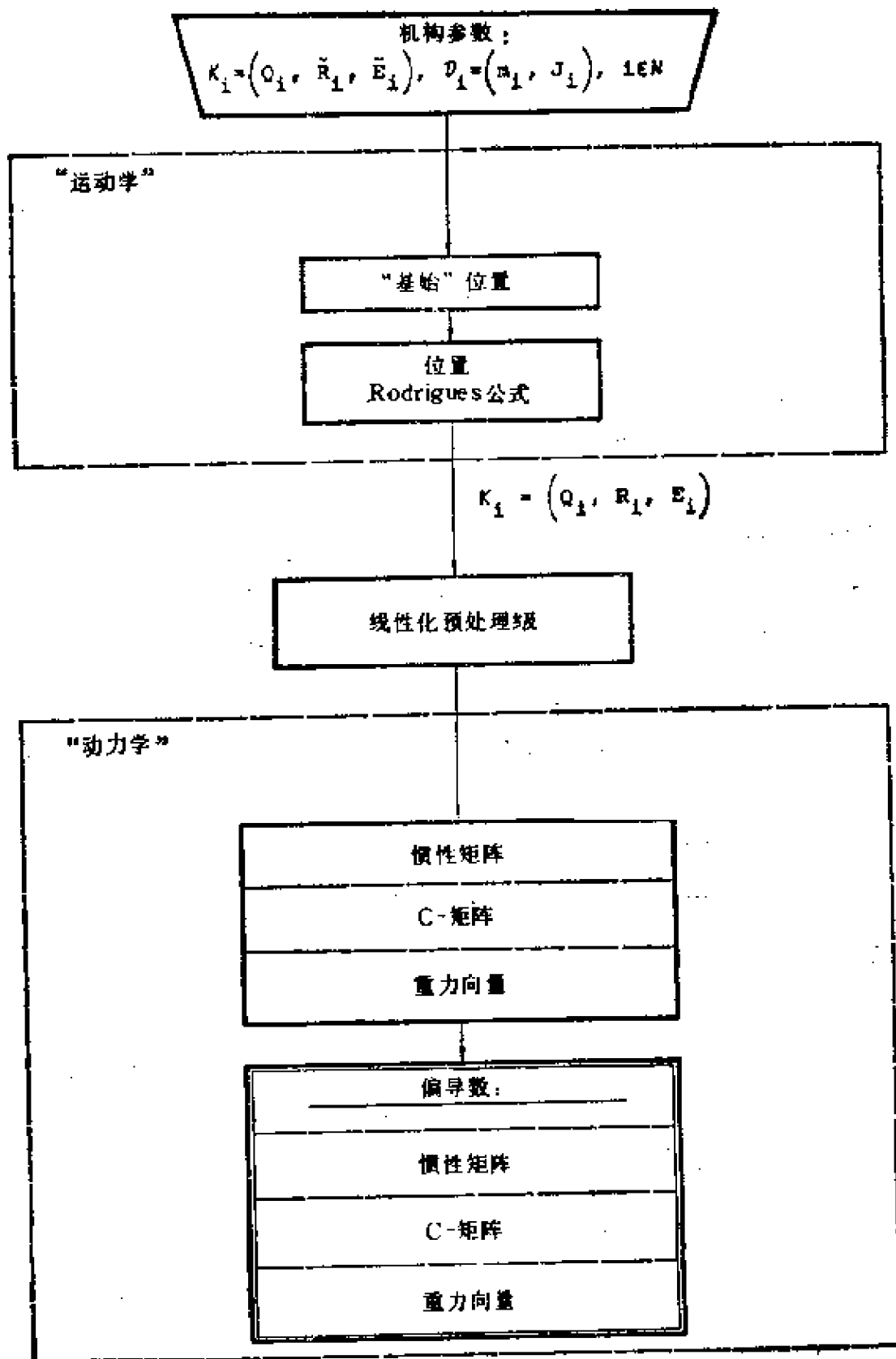


图 2.9 构造非线性和线性化模型的总流程图

$$\frac{\partial H_{ik}}{\partial q_r} = \sum_{j=\max(i,k)}^n \left[m_j (\langle \mathbf{e}, \mathbf{r} \rangle_{ij}^{(j)} \cdot \langle \mathbf{e}, \mathbf{r} \rangle_{ik} + \langle \mathbf{e}, \mathbf{r} \rangle_{ij} \cdot \langle \mathbf{e}, \mathbf{r} \rangle_{ik}^{(j)} + \sum_{\mu=1}^3 (P_{ij\mu}^{(j)} P_{k\mu} + P_{ij\mu} P_{k\mu}^{(j)}) J_{j\mu} \xi_i \xi_k \right] \quad (2.5.16)$$

用类似方法可获得 C-矩阵和重力向量元素的偏导数的表示式。这样，提出的问题都得到了解决。

最后，我们要说明，正像从式(2.5.1)看到的那样，构造线性化模型不仅需要确定偏导数矩阵式(2.5.4)，而且需要动力学模型矩阵 $H(q, \theta)$ 和 $C(q, \theta)$ 。

图 2.9 给出了构造非线性和线性化模型的总流程图。图 2.10 中给出了全部表示式。

线性化预处理级

$$\begin{aligned} \langle \mathbf{e}, \mathbf{r} \rangle_{ij}^{(j)} &= \{ (\varepsilon_{i1} \times r_{i1} + \varepsilon_{i2}) \eta_{i1} \bar{\xi}_i + [\varepsilon_{i1} \times (\mathbf{e} \times r_{i1} \eta_{i1})] \bar{\xi}_i \\ &\quad + \varepsilon_{i2} \bar{\xi}_i \eta_{i1} \} \bar{\xi}_i \\ \varepsilon_{ij}^{(j)} &= (\varepsilon_{i1} \times \mathbf{e} \eta_{i1} + \varepsilon_{i2} \times \varepsilon_{i1} \eta_{i1}) \bar{\xi}_i \\ p_{ij\mu}^{(j)} &= q_{j\mu} \cdot \varepsilon_{ij} (\eta_{i1} - \bar{\eta}_{i1}) \bar{\xi}_i \end{aligned}$$

惯性矩阵、C-矩阵和重力向量的偏导数

$$\begin{aligned} H_{ik}^{(j)} &= \sum_{j=\max(i,k)}^n \left[m_j \langle \mathbf{e}, \mathbf{r} \rangle_{ij}^{(j)} \cdot \langle \mathbf{e}, \mathbf{r} \rangle_{ik} + \langle \mathbf{e}, \mathbf{r} \rangle_{ij} \cdot \langle \mathbf{e}, \mathbf{r} \rangle_{ik}^{(j)} \right. \\ &\quad \left. + \sum_{\mu=1}^3 (p_{ij\mu}^{(j)} p_{k\mu} + p_{ij\mu} p_{k\mu}^{(j)}) J_{j\mu} \bar{\xi}_i \cdot \bar{\xi}_k \right] \\ (C_{ij}^{(j)})_{(k>l)} &= \sum_{j=\max(i,k)}^n \left\{ m_j [\langle \mathbf{e}, \mathbf{r} \rangle_{ij}^{(j)} \cdot (\varepsilon_{i1} \times \langle \mathbf{e}, \mathbf{r} \rangle_{ik}) + \langle \mathbf{e}, \mathbf{r} \rangle_{ij} \cdot (\varepsilon_{i1} \times \langle \mathbf{e}, \mathbf{r} \rangle_{ik}^{(j)}) \right. \\ &\quad \left. - (\varepsilon_{i1} \times \langle \mathbf{e}, \mathbf{r} \rangle_{ik}) \bar{\xi}_i \eta_{i1} + \langle \mathbf{e}, \mathbf{r} \rangle_{ij} \cdot (\varepsilon_{i1} \times \langle \mathbf{e}, \mathbf{r} \rangle_{ik}^{(j)}) \right\} \end{aligned}$$

$$\begin{aligned}
& + \frac{1}{2} \sum_{k=1}^3 [(p_{i1k}^{(1)} \varepsilon_{1k} + p_{k1i}^{(2)} \varepsilon_{1i} + p_{i1k}^{(3)} \varepsilon_{1k} \\
& + p_{i1k} \varepsilon_{ik}^{(2)} + p_{k1i} \varepsilon_{ii}^{(1)} + p_{i1k} \varepsilon_{ik}^{(1)}) \cdot q_{1\mu} + (p_{i1k} \varepsilon_{1k} \\
& + p_{k1i} \varepsilon_{1i} + p_{i1k} \varepsilon_{1k}) \cdot (e_r \times q_{1\mu}) \bar{E}_i \bar{E}_{i+1}] J_{1\mu} \bar{E}_i \bar{E}_k \} \bar{E}_i \\
(h_i^{(1)})^{(1)} = & - \sum_{j=1}^n (m_j \langle e_j, r \rangle \dot{q}_j^2) \cdot g
\end{aligned}$$

图 2.10 构造线性化模型的算法

2.6 闭式灵敏度模型

让我们考虑动力学模型矩阵式(2.3.1)与参数向量 θ 的关系。参数向量由运动学参数

$$\begin{aligned}
\tilde{R}_i = & \{\tilde{r}_{ii}, \tilde{r}_{i,i+1}\} \subset K_i \\
\tilde{E}_i = & \{\tilde{e}_i\} \subset K_i
\end{aligned} \tag{2.6.1}$$

和动力学参数

$$D_i = \{m_i, J_{i1}, J_{i2}, J_{i3}\}, i \in N \tag{2.6.2}$$

组成,其中运动学参数是对局部坐标系 $Q_i, i \in N$ 而言的。运动学参数改变对动力学模型矩阵的影响是非线性的(图 2.7),但是,经常把这些参数看作是常数,因为它们实际上表示机构的几何参数。这些参数发生变化时,我们要注意这样的事实,它们对惯性矩阵和 G -矩阵元素的影响是一类二次函数,而对重方向量元素的影响是线性函数

这一节我们考虑动力学参数式(2.6.2)有限改变的情况,作为工件参数发生变化的结果,动力学参数的改变经常在第 n 个连杆出现。用 s 表示连杆数,这个连杆产生的偏差为

$$\Delta\theta_s = \{\Delta m_s, \Delta J_{s1}, \Delta J_{s2}, \Delta J_{s3}\} \tag{2.6.3}$$

现在我们可以说明下列问题,

问题的提出

用式(2.3.1)描述机器人机构模型。假设向量 q , \dot{q} 和 \ddot{q} 不变,当连杆 i 动力学参数向量出现偏差 $\Delta\theta_i$ (式(2.6.3))时,需要确定驱动转矩(力)向量的偏差 ΔP 。

问题的解

让我们考虑惯性矩阵 $H(q, \theta)$ 元素 H_{ik} 的表示式。假设连杆 i 的动力学参数改变了 Δm_i 和 $\Delta J_{i,\mu}$, $\mu = 1, 2, 3$, 则由式(2.3.9)得到

$$\Delta H_{ik} = \left[\langle e, r \rangle_{ii} \cdot \langle e, r \rangle_{ik} \Delta m_i + \sum_{\mu=1}^3 (e_i \cdot q_{i,\mu})(e_k \cdot q_{i,\mu}) \Delta J_{i,\mu} \xi_i \xi_k \right] \eta_{(i,k),i} \quad (2.6.4)$$

引入下列符号,对 $\mu = 1, 2, 3$

$$H_i^*(i, k) = \langle e, r \rangle_{ii} \cdot \langle e, r \rangle_{ik} \eta_{(i,k),i} \quad (2.6.5)$$

$$H_i^*(i, k) = (e_i \cdot q_{i,\mu})(e_k \cdot q_{i,\mu}) \xi_i \xi_k \eta_{(i,k),i} \quad (2.6.6)$$

把它们代入式(2.6.4),得到

$$\Delta H_{ik} = \sum_{\mu=0}^{n_d} H_i^*(i, k) \Delta \theta_i(\mu + 1) \quad (2.6.7)$$

式中

$$\Delta \theta_i(1) = \Delta m_i$$

$$\Delta \theta_i(\mu + 1) = \Delta J_{i,\mu}, \quad \mu = 1, 2, 3$$

$$n_d = 3$$

式(2.6.7)可表示为矩阵形式:

$$\Delta H(q, \theta) = \sum_{\mu=0}^{n_d} H_i^*(q) \Delta \theta_i(\mu + 1) \quad (2.6.8)$$

式中 $H_i^\mu(q): R^n \rightarrow R^{n \times n}$.

我们用类似方法处理 C -矩阵式(2.3.32)。在对参数改变做同样假设条件下,得到

$$\Delta C_{kl}^i = \sum_{\mu=0}^{n_d} C_{kl}^{i\mu}(k, l) \Delta \theta_i(\mu + 1) \quad (2.6.9)$$

式中对 $k \geq l$

$$C_{kl}^{i0}(k, l) = \langle e, r \rangle_{il} \cdot (e_i \times \langle e, r \rangle_{ik}) \xi_i \eta_{(i, k)}, \quad (2.6.10)$$

$$\begin{aligned} C_{kl}^{i\mu}(k, l) = & \frac{1}{2} [(e_i \cdot q_{i,\mu}) \epsilon_{ik} + (e_k \cdot q_{i,\mu}) \epsilon_{il} \\ & + (e_l \cdot q_{i,\mu}) \epsilon_{ik}] \cdot q_{i,\mu} \xi_i \xi_k \xi_l \eta_{(i, k)}, \end{aligned} \quad (2.6.11)$$

式(2.6.9)也可以表示为矩阵形式:

$$\Delta C^i(q, \theta) = \sum_{\mu=0}^{n_d} C^{i\mu}(q) \Delta \theta_i(\mu + 1) \quad (2.6.12)$$

式中 $C^{i\mu}(q): R^n \rightarrow R^{n \times n}$.

现在让我们考虑重力向量 $h_i(q, \theta)$ 与参数的关系。考虑到在式(2.3.33)中,向量 G_i 由 $m_i g$ 给出,得到

$$\Delta h_i^j = \langle e, r \rangle_{il} \cdot g \eta_{il} \Delta m_i, \quad (2.6.13)$$

引入符号

$$h_i^j(i) = \langle e, r \rangle_{il} \cdot g \eta_{il}, \quad (2.6.14)$$

并把它代入式(2.6.13),得到

$$\Delta h_i^j = h_i^j(i) \Delta m_i, \quad (2.6.15)$$

用向量形式表示,这个关系式变为

$$\Delta h^i(q, \theta) = \sum_{\mu=0}^{n_d} h_i^{\mu}(q) \Delta \theta_i(\mu + 1) \quad (2.6.16)$$

式中 $h_i^{\mu}(q): R^n \rightarrow R^n$.

$$h_i^{\mu 0} = [h_i^{\mu}(1) \cdots h_i^{\mu}(n)]^T \quad (2.6.17)$$

$$h_i^{\mu} = [0 \cdots 0]^T \in R^n, \mu = 1, 2, 3 \quad (2.6.18)$$

按照“问题的提出”，在已知 q , \dot{q} 和 \ddot{q} 条件下，需要确定 ΔP 与参数偏差 $\Delta\theta_i$ 的关系。由式 (2.6.8), (2.6.12) 和 (2.6.16) 得到, $\Delta\theta_i$ 对模型矩阵的影响是线性的。作为对式 (2.3.1) 线性化的结果, 得到

$$\Delta P = \Delta H(q, \theta)\ddot{q} + \dot{q}^T \Delta C(q, \theta)\dot{q} + \Delta h^{\mu}(q, \theta) \quad (2.6.19)$$

$$\Delta P = \sum_{\mu=0}^{n_d} (H_i^{\mu}(q)\ddot{q} + \dot{q}^T C_i^{\mu}(q)\dot{q} + h_i^{\mu}(q))\Delta\theta_i, (\mu + 1) \quad (2.6.20)$$

引入矩阵

$$\begin{aligned} H_i(q) &= [H_i^0(q) \vdots H_i^1(q) \vdots \cdots \vdots H_i^{n_d}(q)] \\ C_i(q) &= [C_i^0(q) \vdots C_i^1(q) \vdots \cdots \vdots C_i^{n_d}(q)] \\ h_i^{\mu}(q) &= [h_i^{\mu 0}(q) \vdots 0 \vdots \cdots \vdots 0] \end{aligned} \quad (2.6.21)$$

则模型式 (2.6.20) 可简化为

$$\Delta P = (H_i(q)\ddot{q} + \dot{q}^T C_i(q)\dot{q} + g_i^{\mu}(q))\Delta\theta_i, \quad (2.6.22)$$

也就是提出问题的解。在一定条件下, 模型式 (2.6.22) 称为“灵敏度”模型。

现在让我们分析模型式 (2.6.22) 的某些特性, 也就是分析参与构造公式 (2.6.22) 的矩阵

$$H_i^{\mu}(q), C_i^{\mu}(q), h_i^{\mu}(q), \mu \in \{0, 1, 2, 3\}$$

的特性:

- 1) 模型式 (2.6.22) 的矩阵对机构动力学参数是独立的;
- 2) 模型式 (2.6.22) 既适用于小的, 也适用于大的参数偏差;
- 3) 矩阵 $H_i^{\mu}(q), \mu \in \{0, \dots, n_d\}$ 是惯性效应的结果。这个矩阵是对称的和正定的。

证明

矩阵 $H^*(q) \in R^{n \times n}$ 的对称性可直接由矩阵式 (2.6.5) 和 (2.6.6) 得到。通过类似于 2.4 节定理 1 的证明方法可以证明正定性。引入符号

$$\begin{aligned} P_{ii}^0 &= \langle e, r \rangle_{ii} \eta_{ii} \\ P_{ii}^\mu &= (e_i \cdot a_{i\mu}) \eta_{ii} \xi_i, \quad \mu = 1, 2, 3 \end{aligned} \quad (2.6.23)$$

由于

$$\eta_{(i,k),i} = \eta_{ii} \eta_{ki}$$

因此式(2.6.5)和(2.6.6)可写为

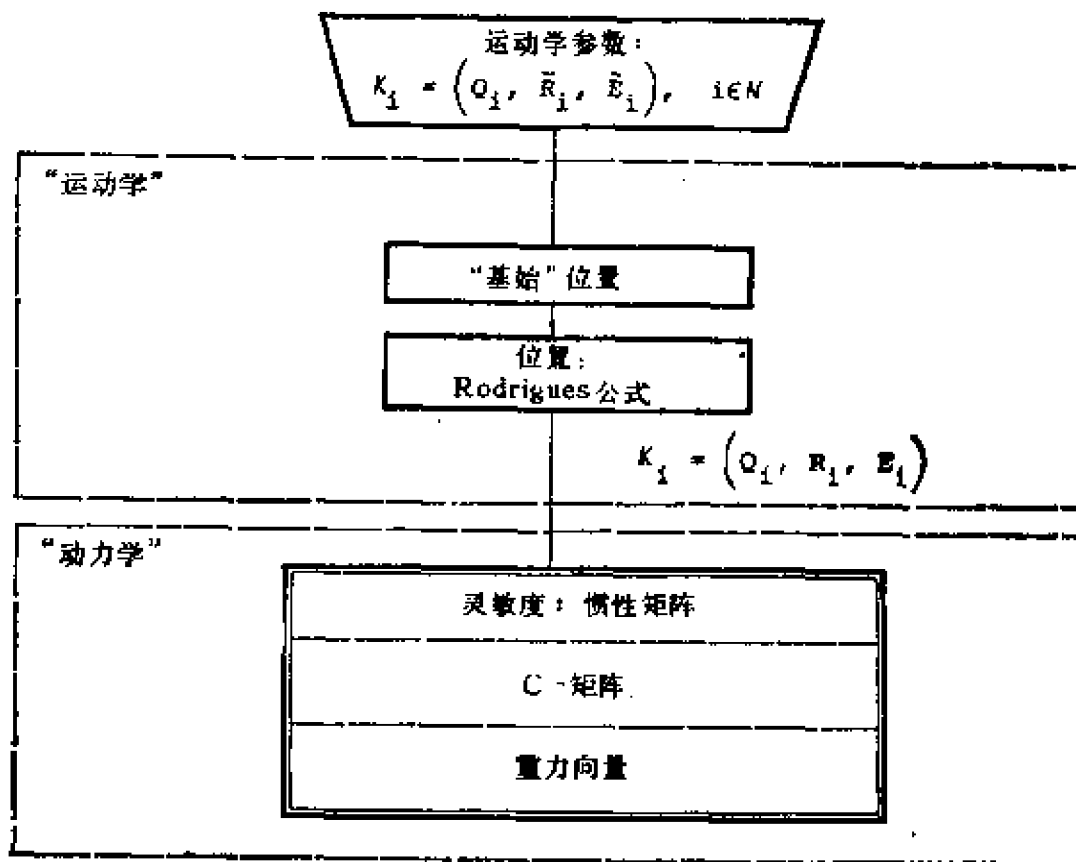


图 2.11 构造灵敏度矩阵的总流程图

$$H_i^\mu(i, k) = P_{ii}^\mu \cdot P_{kk}^\mu, \quad \mu \in \{1, \dots, n_i\} \quad (2.6.24)$$

式(2.6.24)与式(2.4.3)相似, 因而仿照步骤式(2.4.3)~(2.4.7), 得到

$$x^T H_i^\mu(q) x > 0 \quad (2.6.25)$$

式中 $x \in R^n$ 是任意向量, 并满足 $x \neq 0$. 这就是对正定性的证明.

4) 矩阵 $C_i^\mu(q) \in R^{n \times n}$ 是作为哥氏和离心效应的结果而存在的. 这些矩阵是对称的, 也就是

$$C_i^\mu(k, l) = C_i^\mu(l, k), \quad k, l \in N \quad (2.6.26)$$

另外, 反对称性也成立, 即

$$C_i^\mu(k, l) = -C_i^\mu(l, k), \quad k, l \geq 1 \quad (2.6.27)$$

证明

相对于指标集 k 和 l 的对称性, 式(2.6.26)是矩阵

播性矩阵、C-矩阵和重力向量的灵敏度

$$H_i^\mu(i, k) = \langle e, r \rangle_{ii} \cdot \langle e, r \rangle_{kk} \bar{\eta}_{i, k}^{\mu} \bar{\eta}_{i, k}^{\mu}$$

$$H_i^\mu(i, k) = (e_i \cdot q_{i, \mu})(e_k \cdot q_{k, \mu}) \bar{\xi}_i \bar{\xi}_k \bar{\eta}_{i, k}^{\mu} \bar{\eta}_{i, k}^{\mu}$$

式中 $i, k \in N, \mu \in \{1, 2, 3\}$.

$$C_i^\mu(k, l) = \langle e, r \rangle_{ii} \cdot (e_l \times \langle e, r \rangle_{kk}) \bar{\xi}_i \bar{\eta}_{i, k}^{\mu} \bar{\eta}_{i, l}^{\mu}$$

$$C_i^\mu(k, l) = \frac{1}{2} [(e_i \cdot q_{i, \mu}) \varepsilon_{ik} + (e_k \cdot q_{i, \mu}) \varepsilon_{il} \\ + (e_l \cdot q_{i, \mu}) \varepsilon_{ik}] \cdot q_{i, \mu} \bar{\xi}_i \bar{\xi}_k \bar{\xi}_l \bar{\eta}_{i, k}^{\mu} \bar{\eta}_{i, l}^{\mu}$$

式中 $i, k, l \in N, \mu \in \{1, 2, 3\}$.

$$h_i^\mu(i) = \langle e, r \rangle_{ii} g \cdot \bar{\eta}_{ii}^{\mu}$$

式中 $i \in N$.

图 2.12 构造灵敏度矩阵算法 (i 是具有动力学变量参数的连杆数)

$C^i(q, \theta)$ 对称性的直接结果, $C^i(q, \theta)$ 的元素由式(2.3.32)给出. 反对称性要在式(2.6.10)的基础上证明

$$C_i^\mu(i, l) = \langle e, r \rangle_{ii} \cdot (e_l \times \langle e, r \rangle_{ii}) \bar{\xi}_i \bar{\eta}_{i, l}^{\mu} \bar{\eta}_{i, l}^{\mu}$$

$$\begin{aligned}
&= -\langle e_i, r \rangle_{ii} \cdot (e_i \times \langle e_i, r \rangle_{ik}) \xi_i \dot{q}_k \\
&= -C_i^\mu(k, l)
\end{aligned}$$

式中 $\mu = 0, i \geq 1$. 对 $\mu = 1, 2, 3$ 的情况, 反对称性用类似方法由式(2.6.11)证明.

最后, 我们要说明, 构造模型式(2.6.22)比构造完全的动力学模型大大减少了数值运算. 图 2.11 给出了计算灵敏度模型的总流程图, 图 2.12 给出了详细算法.

附 录 2.1

考虑运动学变量 $\beta_i^0 \in W_i$. 在 2.3 节中我们描述了这一变量的非递推形式的推导方法, 并由表示式(2.3.14)给出最终结果. 在本附录介绍由 β_i^0 的复杂表示式(在 2.3 节式(2.3.14)之前已给出这种表示式)来获得式(2.3.14)的详细方法.

按照 2.3 节中给出的推导, 运动学变量 β_i^0 可表示为

$$\begin{aligned}
\beta_i^0 = & \sum_{j=1}^i \sum_{k=1}^{j-1} \sum_{l=1}^{k-1} \beta_{ijk}^{(1)} + \sum_{k=1}^i \sum_{l=1}^{k-1} \beta_{ikl}^{(2)} \\
& + \sum_{j=1}^i \sum_{k=1}^j \sum_{l=1}^j \beta_{ijk}^{(3)} - \sum_{j=1}^i \sum_{k=1}^{j-1} \sum_{l=1}^{j-1} \beta_{ikl}^{(4)}
\end{aligned} \tag{A2.1.1}$$

其中

$$\beta_{ijk}^{(1)} = e_{ik} \times R_{j,i-1} \xi_k \xi_l \dot{q}_k \dot{q}_l \tag{A2.1.2}$$

$$\beta_{ikl}^{(2)} = (e_{ik} \times r_{kk} \xi_k + 2e_{ik} \xi_k) \xi_l \dot{q}_k \dot{q}_l \tag{A2.1.3}$$

$$\beta_{ijk}^{(3)} = e_k \times (e_l \times r_{ij}) \xi_k \xi_l \dot{q}_k \dot{q}_l \tag{A2.1.4}$$

$$\beta_{ikl}^{(4)} = e_k \times (e_l \times r_{i-1,l}) \xi_k \xi_l \dot{q}_k \dot{q}_l \tag{A2.1.5}$$

这样, 问题就归结为要把式(A2.1.1)变换为等价式(2.3.15).

为此目的,让我们考虑恒等式

$$\sum_{l=1}^i \sum_{k=1}^{l-1} \sum_{l=1}^{k-1} \beta_{kl}^{(i)} = \sum_{k=1}^i \sum_{l=1}^i \left(\eta_{kl} \sum_{j=k+1}^i \beta_{jl}^{(i)} \right) \quad (\text{A2.1.6})$$

类似于推导表示式(2.3.12),让我们对式(A2.1.6)的右边引入缩短符号

$$\sum_{k=1}^i \sum_{l=1}^i \eta_{kl} P_{kl}, \quad P_{kl} = \sum_{j=k+1}^i \beta_{jl}^{(i)}$$

并且,不但注意到恒等式

$$\sum_{k=1}^i \sum_{l=1}^i \eta_{kl} P_{kl} = \sum_{k=1}^i \sum_{l=1}^i \frac{1}{2} (\eta_{kl} P_{kl} + \eta_{lk} P_{lk}) \quad (\text{A2.1.7})$$

成立,而且注意到方程式(A2.1.7)右边双重求和表示式对指标集 k 和 l 是对称的,也就是

$$q_{kl} = q_{lk}$$

成立,式中

$$q_{kl} = \frac{1}{2} (\eta_{kl} P_{kl} + \eta_{lk} P_{lk}) = \begin{cases} \frac{1}{2} P_{kl}, & k \geq l \\ \frac{1}{2} P_{lk}, & k < l \end{cases} \quad (\text{A2.1.8})$$

上述对称性允许我们在 $k \geq l$ 时,用 $\frac{1}{2} P_{kl}$ 代替 q_{kl} (对 $k < l$, 则改变指标集次序即可),也就是

$$\sum_{k=1}^i \sum_{l=1}^i q_{kl} = \sum_{k=1}^i \sum_{l=1}^i \frac{1}{2} P_{kl} \quad (\text{A2.1.9})$$

于是,可把式(A2.1.7)变为

$$\sum_{k=1}^i \sum_{l=1}^i \eta_{kl} P_{kl} = \sum_{k=1}^i \sum_{l=1}^i \frac{1}{2} P_{kl} \quad (\text{A2.1.10})$$

我们把这个关系式用到式(A2.1.6),并考虑到由式(A2.1.2)

给出 $\beta_{ikl}^{(1)}$, 得到

$$\begin{aligned} \sum_{i=1}^l \sum_{k=1}^{i-1} \sum_{j=1}^{k-1} \beta_{ikl}^{(1)} &= \sum_{k=1}^l \sum_{j=1}^k \frac{1}{2} \left(\mathbf{e}_{ik} \right. \\ &\quad \left. \times \sum_{i=k+1}^l \mathbf{R}_{i,j-1} \right) \xi_k \xi_l \dot{q}_k \dot{q}_l \\ &= \sum_{k=1}^l \sum_{i=1}^l \frac{1}{2} \left(\mathbf{e}_{ik} \times \mathbf{R}_{ik} \right) \xi_k \xi_l \dot{q}_k \dot{q}_l \end{aligned} \quad (\text{A2.1.11})$$

式中 \mathbf{R}_{ik} 是第 k 个连杆质量中心到连杆 i 质量中心的距离向量。

现在让我们考虑式 (A2.1.1) 的第二项, 类似于上述推导, 得到

$$\begin{aligned} \sum_{k=1}^l \sum_{l=1}^{k-1} \beta_{kl}^{(2)} &= \sum_{k=1}^l \sum_{l=1}^k \eta_{kl} \beta_{kl}^{(2)} \\ &= \sum_{k=1}^l \sum_{l=1}^k \frac{1}{2} \beta_{kl}^{(2)} \\ &= \sum_{k=1}^l \sum_{l=1}^k \left(\frac{1}{2} \mathbf{e}_{kl} \times \mathbf{r}_{kl} \xi_k \right. \\ &\quad \left. + \mathbf{e}_{lk} \xi_k \right) \xi_l \dot{q}_k \dot{q}_l \end{aligned} \quad (\text{A2.1.12})$$

利用恒等式

$$\sum_{i=1}^l \sum_{k=1}^i \sum_{l=1}^k \beta_{ikl}^{(2)} = \sum_{k=1}^l \sum_{l=1}^k \left(- \sum_{i=\overline{k,l}}^l \beta_{ikl}^{(2)} \right) \quad (\text{A2.1.13})$$

可把式 (A2.1.1) 的第三项变换为等价式。因为方程式 (A2.1.13) 右边双重求和表示式对指标集 k 和 l 不是对称的, 所以我们要利用恒等式

$$\sum_{k=1}^i \sum_{l=1}^i P_{kl} = \sum_{k=1}^i \sum_{l=1}^i \frac{1}{2} (P_{kl} + P_{lk}) \quad (\text{A2.1.14})$$

于是, 可把式 (A2.1.13) 变为

$$\begin{aligned} \sum_{i=1}^i \sum_{k=1}^i \sum_{l=1}^i \beta_{ikl}^{(3)} = & \sum_{k=1}^i \sum_{l=1}^i \left\{ \frac{1}{2} \sum_{j=(k,l)}^i [e_k(e_l \times r_{lj}) \right. \\ & \left. + e_l \times (e_k \times r_{jl})] \xi_k \xi_l \dot{q}_k \dot{q}_l \right\} \quad (\text{A2.1.15}) \end{aligned}$$

类似于处理第三项的方法, 可把式 (A2.1.1) 的第四项变换为对应的等价式

$$\begin{aligned} \sum_{i=1}^i \sum_{k=1}^{i-1} \sum_{l=1}^{i-1} \beta_{ikl}^{(4)} = & \sum_{k=1}^i \sum_{l=1}^i \left(\sum_{j=(k+1,l+1)}^i \beta_{ijk}^{(4)} \right) \\ = & \sum_{k=1}^i \sum_{l=1}^i \left[\frac{1}{2} \sum_{j=(k+1,l+1)}^i (\beta_{ijk}^{(4)} + \beta_{ilk}^{(4)}) \right] \\ = & \sum_{k=1}^i \sum_{l=1}^i \left\{ \frac{1}{2} \sum_{j=(k+1,l+1)}^i [e_k \times (e_l \right. \\ & \left. \times r_{l-1,j}) + e_l \times (e_k \times r_{j-1,l})] \xi_k \xi_l \dot{q}_k \dot{q}_l \right\} \quad (\text{A2.1.16}) \end{aligned}$$

$$\begin{aligned}
C'_B(k,l) = & \frac{1}{2} \left\{ \mathbf{e}_{ik} \times R_{ik} \xi_k + \mathbf{e}_{ik} \times r_{kk} \xi_k + 2\mathbf{e}_{ik} \xi_k \right. \\
& + \sum_{i=k}^l [\mathbf{e}_k \times (\mathbf{e}_i \times r_{ii}) + \mathbf{e}_i \times (\mathbf{e}_k \times r_{ii})] \xi_k \\
& - \sum_{i=k+1}^l [\mathbf{e}_k \times (\mathbf{e}_i \times r_{i-1,i}) \\
& \left. + \mathbf{e}_i (\mathbf{e}_k \times r_{i-1,i}) \right] \xi_k \Big\} \xi_l \quad (\text{A2.1.18})
\end{aligned}$$

考虑到

$$\begin{aligned}
R_{ik} + r_{kk} &= r_{ik} \\
\sum_{i=k}^l r_{ii} - \sum_{i=k+1}^l r_{i-1,i} &= r_{ik}
\end{aligned}$$

得到

$$\begin{aligned}
C'_B(k,l) = & \frac{1}{2} [\mathbf{e}_{ik} \times r_{ik} \xi_k + 2\mathbf{e}_{ik} \xi_k \\
& + \mathbf{e}_k (\mathbf{e}_l \times r_{ik}) \xi_k + \mathbf{e}_l \times (\mathbf{e}_k \times r_{ik}) \xi_k] \xi_l \quad (\text{A2.1.19})
\end{aligned}$$

因为

$$\begin{aligned}
\mathbf{e}_{ik} &= \mathbf{e}_i \times \mathbf{e}_k \\
(\mathbf{e}_i \times \mathbf{e}_k) \times r_{ik} + \mathbf{e}_k \times (\mathbf{e}_i \times r_{ik}) &= \mathbf{e}_i \times (\mathbf{e}_k \times r_{ik})
\end{aligned}$$

则得到

$$\begin{aligned}
C'_B(k,l) &= [\mathbf{e}_l \times (\mathbf{e}_k \times r_{ik}) \xi_k + \mathbf{e}_{ik} \xi_k] \xi_l \\
&= \mathbf{e}_l \times [(\mathbf{e}_k \times r_{ik}) \xi_k + \mathbf{e}_k \xi_k] \xi_l
\end{aligned}$$

或按照已采用的符号

$$\langle \mathbf{e}, \mathbf{r} \rangle_{ik} = \mathbf{e}_k \times r_{ik} \xi_k + \mathbf{e}_k \xi_k$$

有

$$C'_B(k,l) = \mathbf{e}_l \times \langle \mathbf{e}, \mathbf{r} \rangle_{ik} \xi_l \quad (\text{A2.1.20})$$

这就是问题的解。

附 录 2.2

在本附录中，我们要介绍一系列等式变换，通过这些变换，把变量 b_i^0 简化为最简单形式的式(2.3.25)和(2.3.26)。为此，把式(2.3.20)和(2.3.21)代入式(2.3.19)，得到

$$b_i^0 = - \sum_{k=1}^l \sum_{l=1}^{k-1} \sum_{\mu=1}^3 (\mathbf{e}_{lk} \cdot \mathbf{q}_{l\mu}) \mathbf{q}_{i\mu} J_{i\mu} \xi_k \xi_l \dot{q}_k \dot{q}_l + \sum_{k=1}^l \sum_{l=1}^k \lambda_{kl}^i \dot{q}_k \dot{q}_l \quad (\text{A2.2.1})$$

让我们考虑等式(A2.2.1)右边的第一项。应用附录2.1中同样的方法，完成对指标集 k 和 l 的“对称化”(见式(A2.1.6)——(A2.1.9))，得到

$$\sum_{k=1}^l \sum_{l=1}^{k-1} \sum_{\mu=1}^3 (\mathbf{e}_{lk} \cdot \mathbf{q}_{l\mu}) \mathbf{q}_{i\mu} J_{i\mu} \xi_k \xi_l \dot{q}_k \dot{q}_l - \sum_{k=1}^l \sum_{l=1}^k \left[\frac{1}{2} \sum_{\substack{\mu=1 \\ (k>l)}}^3 (\mathbf{e}_{lk} \cdot \mathbf{q}_{l\mu}) \mathbf{q}_{i\mu} J_{i\mu} \xi_k \xi_l \dot{q}_k \dot{q}_l \right] \quad (\text{A2.2.2})$$

很明显，式(A2.2.1)右边的第二项是 λ_{kl}^i 的函数， λ_{kl}^i 由式(2.3.22)给出：

$$\lambda_{kl}^i = \sum_{\mu=1}^3 P_{kl\mu}^i (J_{i(\mu+1)} - J_{i(\mu+2)}) \xi_k \xi_l \quad (\text{A2.2.3})$$

式中

$$P_{kl\mu}^i = (\mathbf{e}_k \cdot \mathbf{q}_{i(\mu+1)}) (\mathbf{e}_l \cdot \mathbf{q}_{i(\mu+2)}) \mathbf{q}_{i\mu} \quad (\text{A2.2.4})$$

通过简单变换，式(A2.2.3)可变换为

$$\lambda_{kl}^i = \sum_{\mu=1}^3 (P_{k|l(\mu+2)}^i - P_{l|k(\mu+1)}^i) J_{i\mu} \xi_k \xi_l \quad (\text{A2.2.5})$$

另外,我们需要把式(2.3.21)变换为对指符集 k 和 l 对称的形式,这已在式(1.4.4)中给出。应用类似于附录 2.1 中的方法,得到

$$C_i^i(k, l) = \frac{1}{2} (\lambda_{kl}^i + \lambda_{lk}^i), \quad i \geq k, l \quad (\text{A2.2.6})$$

式中

$$C_i^i(k, l) = C_i^i(l, k)$$

把式(A2.2.5)代入式(A2.2.6),得到

$$C_i^i(k, l) = \frac{1}{2} \sum_{\mu=1}^3 (P_{l|k(\mu+2)}^i - P_{k|l(\mu+1)}^i + P_{k|l(\mu+2)}^i - P_{l|k(\mu+1)}^i) J_{i\mu} \xi_k \xi_l \quad (\text{A2.2.7})$$

让我们注意,因为

$$q_{i\mu} = q_{i(\mu+1)} \times q_{i(\mu+2)}$$

所以下式成立:

$$\begin{aligned} P_{l|k(\mu+2)}^i - P_{k|l(\mu+1)}^i &= (e_k \cdot q_{i\mu}) [(e_l \cdot q_{i(\mu+1)}) q_{i(\mu+2)} \\ &\quad - (e_l \cdot q_{i(\mu+2)}) q_{i(\mu+1)}] \\ &= (e_k - q_{i\mu}) (q_{i\mu} \times e_l) \end{aligned} \quad (\text{A2.2.8})$$

用同样方法,得到

$$P_{k|l(\mu+2)}^i - P_{l|k(\mu+1)}^i = (e_l \cdot q_{i\mu}) (q_{i\mu} \times e_k) \quad (\text{A2.2.9})$$

把式(A2.2.8)和(A2.2.9)代入式(A2.2.7),得到

$$\begin{aligned} C_i^i(k, l) &= \frac{1}{2} \sum_{\mu=1}^3 [(e_k \cdot q_{i\mu}) (q_{i\mu} \times e_l) \\ &\quad + (e_l \cdot q_{i\mu}) (q_{i\mu} \times e_k)] J_{i\mu} \xi_k \xi_l \end{aligned} \quad (\text{A2.2.10})$$

上式与式(2.3.24)恒等。最后,让我们把式(A2.2.2)和

(A2.2.10) 代入式 (A2.2.1), 得到

$$b_i^j = \sum_{k=1}^i \sum_{l=1}^i \left\{ \frac{1}{2} \sum_{\substack{\mu=1 \\ (k>l)}}^3 [((\mathbf{e}_k \times \mathbf{e}_l) \cdot \mathbf{q}_{i\mu}) \mathbf{q}_{i\mu} \right. \\ \left. + (\mathbf{e}_k \cdot \mathbf{q}_{i\mu})(\mathbf{q}_{i\mu} \times \mathbf{e}_l) + (\mathbf{e}_l \cdot \mathbf{q}_{i\mu})(\mathbf{q}_{i\mu} \times \mathbf{e}_k)] J_{i\mu} \xi_k \xi_l \dot{q}_k \dot{q}_l \right\} \quad (\text{A2.2.11})$$

由上式可直接得到式(2.3.26)。这就确定了动力学系数 b_i^j 的最简单闭式表示式。

第三章 计算机辅助生成机器人 数值-符号模型

3.1 引言

在第二章中，我们分析了机器人操作手数学建模的几种算法；研究了机器人的闭式动力学模型，这种模型很便于程序实现；我们还进一步指出了如何用这种模型证明惯性矩阵的对称性和正定性， C -矩阵的反对称性；解决了线性化和灵敏度模型的生成问题。在这一章中，我们要考虑上述算法的计算机实现问题。

我们在第一章中已经指出，目前有三种基本方法可完成程序实现，它们分别称为数值法、符号法和数值-符号(解析)法。按照处理计入建模方法中变量的类型可对这些方法进一步分类，一般说来，不管运动学变量，还是动力学变量，它们都是参数、关节坐标和关节速度的函数。因为这些变量可以是标量函数，也可以是向量函数，所以我们可把这些变量看作是

- 1) 数值量(由给定参数、关节坐标和关节速度算出)；
- 2) 给定参数和广义坐标的函数。

情况 1) 就是在第一章中所述的数值建模法。这类方法的沉重负担是要进行大量的浮点乘法/加法运算，因而难于实时实现。情况 2) 的基本任务是要解决下列问题：

- a) 从计算机存储单元占用量和运算速度的观点看，如何选择自变量；

b) 如何用统一的方式描述任意变量对自变量的函数关系;

c) 如何实现函数间的代数运算;

d) 如何优化函数,以减少浮点乘法/加法运算次数。(例如,函数 $z_1 = (\cos^2 x + \sin^2 x)y$ 和函数 $z_2 = y$ 是等价的,但显然, z_2 的结构比 z_1 简单。)

在这一章中,我们要考虑上述的所有问题。第一个问题是进行相当精细和有效的初步分析。正像在第一章中阐述的那样,可把任意参数、关节坐标及其导数当作自变量。这样做的结果就得到非数值量,即符号建模法。不过,描述机器人连杆运动学和动力学特性的参数数量是相当大的。例如,我们在第一章已经指出,大约用 100 个参数才能描述六个自由度的操作手。考虑到需要附加 12 个关节坐标/速度和遍及整个建模过程进行的非线性代数运算,可以断定这种方法是相当复杂的。

我们看到,由于编制软件的复杂性,使用符号法会遇到很多困难。另一方面,对给定机器人,经常把参数取为常数。因此,如果只把关节坐标和关节速度作为自变量,那么就可把自变量的数目减少到 $2n$ 个 (n 为关节/连杆的数目)。已经证明,甚至在这种简化情况下,描述哥氏和离心效应的所有表示式的解析结构也是相当复杂的。由于这个缘故,当然最精巧和最有效的解决方法就是只把关节坐标当作自变量。在 2.3 节中描述的闭式模型就属于这种方法,因为它的动力学模型矩阵 $H(q)$, $C(q)$ 和 $h_g(q)$ 只是关节坐标的函数。

通过上述讨论可知,最适宜的方法显然是数值-符号法,即把参数取为常数,而把关节坐标当作符号(自变量)处理。

我们在 3.2 节中讨论问题 b),研究函数-变量关系的解析

结构.可以证明,任何一个运动学或动力学变量均可用唯一的一列数据表示.通过引入由关节坐标空间 $Q \subset R^n$ 到新定义自变量空间 $X \subset R^{3n}$ 的特定非线性变换,就能获得这种唯一的形式.这是一种由复杂三角函数到多项式的变换.正像在第一章中指出的那样,这些多项式可用数值数据的两个集合系数向量和指数矩阵描述.这一对集合唯一地定义了多项式,即变量.我们把这对集合称为多项式矩阵.这种矩阵描述了模型变量的解析结构.接着,我们要证明一个十分重要的定理,它阐明 B -矩阵的元素只对应集合 $\{0,1,2\}$.从存储单元占用量的角度看,这是一个非常有意义的定理.

在 3.3 节中,我们研究多项式矩阵代数.我们要证明,多项式矩阵空间构成一个向量空间.自然,我们要引入标量的加法、乘法及叉积和点积等运算.可以证明,这些运算适合在数字计算机上编制程序.

然后,我们着手解决一个基本问题,即多项式矩阵的唯一性问题.正如我们将要看到的那样,可用无穷多个多项式矩阵表示同一个变量.这是从自变量空间 $Q \subset R^n$ (关节坐标)变换到空间 $X \subset R^{3n}$ 的展开式的一个基本结论. $Q \rightarrow X$ 的变换还导致在 X 空间变量之间出现非线性关系式.这意味着,确实存在不同的,但是等价的多项式矩阵.现在,我们遇到的问题是,如何从众多的等价多项式矩阵的集合中,确定出一个唯一的表示式.我们将要证明,确实存在这样的表示式,它对应一个具有最小行数的多项式矩阵.我们还要研究把任意多项式矩阵简化为最小行数多项式矩阵的算法.

在这一章中的第二部分专门论述机器人建模所得结果的应用.在 3.5 节中,我们介绍一种用解析形式表示的构造非线性动力学模型的算法.然后,我们研究两种本质不同的构

造机器人线性化模型的算法，第一种算法是利用 2.5 节中给出的方程，第二种算法是建立在多项式矩阵偏导数定理的基础上(参阅 3.6 节)。对应用来说，第二种算法是相当精巧和有效的算法。下面我们探讨推导解析灵敏度模型的问题。然后，我们处理一个很实际的问题，这就是论述构造机器人近似模型的问题。基于这种思想，我们采用的方法是建立在预先规定误差判据的基础上的，这种算法要分析多项式的有关项，即对应多项式矩阵有关行，在不违反规定误差的条件下，它们可以忽略不计。在本章结尾，我们提供一个例子，从减少数值运算的复杂程度和实时应用的观点看，它说明了近似模型的重要性。

3.2 变量的数值-符号法

考虑具有 n 个连杆和 n 个关节的机器人操作手。在 2.2 节中，对于用旋转和滑动关节联接的简单运动链，我们已把运动学变量的集合定义为

$$K_i = (Q_i, R_i, E_i), \quad i \in N = \{1, \dots, n\}$$

式中

Q_i ——附设于连杆 i 的局部坐标系， $Q_i = (q_{i1}, q_{i2}, q_{i3})$;

R_i ——由关节到质量中心的距离向量集合， $R_i = \{r_{ii}, r_{i,i+1}\}$;

E_i ——为关节轴的单位向量集合， $E_i = \{e_i\}$ 。

由 2.3 节中(图 2.7)的闭式模型方程可以看到，模型方程的变量(向量和标量表示式)由 K_i 和

$$D_i = (m_i, J_{i1}, J_{i2}, J_{i3}), \quad i \in N$$

得到，现在让我们引入“机构变量”，或简称为“变量”的概念。

定义

机构变量是

- 1) $K_i, \forall i \in N$ 的元素;
- 2) 通过 $K_i, \forall i \in N$ 的线性运算得到的变量;
- 3) 通过 $K_i, \forall i \in N$ 的非线性运算得到的变量。

所谓线性运算指的是向量加法和减法，而非线性运算指的是标量乘法和向量乘法。例如，由 2) 定义的变量可以是

$$r_{ij} = r_{ij} = r_{i,i+1} + r_{i+1,i+1} + \dots + r_{ij}, i > j$$

由 3) 定义的变量可以是

$$\begin{aligned} e_{ij} &= e_i \times e_j, \\ \langle e, r \rangle_{ij} &= e_i \times r_{ij} \xi_j + e_j \xi_j \\ P_{ij\mu} &= e_i \cdot q_{j\mu} \end{aligned} \quad (3.2.1)$$

机构变量可用三种方法描述：数值法、符号法和混合法（把数值法和符号法结合起来）。变量的数值表示式表示实数集合，通过给定关节坐标向量确定它的值。变量的符号表示式由数学符号组成，它定义了它对参数和关节坐标的函数关系。描述变量的第三种形式称为数值-符号法或解析法。在这里，参数用数值处理，关节坐标用符号处理。正像第一章中阐述的那样，基于解析描述变量的建模算法是非常有用的。

变量的数值-符号表示法

考虑局部坐标系 $Q_i, i \in N$ 的单位向量集合。确定这些向量的基本方程是有限旋转公式(2.2.4)，重写如下：

$$q_{ij} = [q_{ij}^{(3)} q_{ij}^{(2)} q_{ij}^{(1)}] \begin{bmatrix} \cos q_i \\ \sin q_i \\ 1 \end{bmatrix}, j = 1, 2, 3 \quad (3.2.2)$$

式中

$$\begin{aligned} \mathbf{q}_{ii}^{(3)} &= \mathbf{e}_i \times (\mathbf{q}_{ii}^0 \times \mathbf{e}_i) \xi_i \\ \mathbf{q}_{ii}^{(2)} &= \mathbf{e}_i \times \mathbf{q}_{ii}^0 \xi_i \\ \mathbf{q}_{ii}^{(1)} &= (\mathbf{e}_i \cdot \mathbf{q}_{ii}^0) \mathbf{e}_i \xi_i + \mathbf{q}_{ii}^0 \xi_i \end{aligned} \quad (3.2.3)$$

给定关节坐标 q_1, \dots, q_{i-1} , 我们就能确定向量 \mathbf{e}_i 和 \mathbf{q}_{ii}^0 . 让我们确定它们对 q_{i-1} 的解析关系. 由式(2.2.5), 有

$$\mathbf{e}_i = Q_{i-1} \tilde{\mathbf{e}}_i \quad (3.2.4)$$

式中 $\tilde{\mathbf{e}}_i$ 表示关节 i 对连杆 $i-1$ 坐标系的单位向量. 我们还有

$$\mathbf{q}_{ii}^0 = Q_{i-1} \tilde{\mathbf{q}}_{ii}^0 \quad (3.2.5)$$

式中 $\tilde{\mathbf{q}}_{ii}^0$ 由第 $i-1$ 个坐标系给出. 由式(3.2.4)和(3.2.5), 得到

$$\mathbf{q}_{ii}^{(k)} = Q_{i-1} \tilde{\mathbf{q}}_{ii}^{(k)}, \quad k = 1, 2, 3 \quad (3.2.6)$$

把上式代入式(3.2.2), 得到

$$\mathbf{q}_{ii} = Q_{i-1} [\tilde{\mathbf{q}}_{ii}^{(3)} \tilde{\mathbf{q}}_{ii}^{(2)} \tilde{\mathbf{q}}_{ii}^{(1)}] \begin{bmatrix} \cos q_i \\ \sin q_i \\ 1 \end{bmatrix} \quad (3.2.7)$$

考虑到

$$Q_{i-1} = [q_{i-1,1} \ q_{i-1,2} \ q_{i-1,3}]$$

和

$$\mathbf{q}_{i-1,j} = [q_{i-1,j}^{(2)} \ q_{i-1,j}^{(1)}] \begin{bmatrix} \cos q_{i-1} \\ \sin q_{i-1} \\ 1 \end{bmatrix} \quad (3.2.8)$$

并把它们代入式(3.2.7), 得到

$$q_{ii} = [q_{ij}^{(9)} \cdots q_{ij}^{(1)}] \begin{bmatrix} \cos q_{i-1} \cos q_i \\ \cos q_{i-1} \sin q_i \\ \cos q_{i-1} \\ \hline \sin q_{i-1} \cos q_i \\ \sin q_{i-1} \sin q_i \\ \sin q_{i-1} \\ \hline \cos q_i \\ \sin q_i \\ 1 \end{bmatrix} \quad (3.2.9)$$

原来,式(3.2.9)右边列向量元素和笛卡儿乘积的元素

$$(\cos q_{i-1}, \sin q_{i-1}, 1) \times (\cos q_i, \sin q_i, 1) \quad (3.2.10)$$

是一致的。这个向量简单表示为 $\text{Rot}(q_{i-1}, q_i)$ 。于是,方程式(3.2.2)和(3.2.9)简化为

$$q_{ii} = Q_{ij}^{(9)} \text{Rot} q_i = Q_{ij}^{(9)} \text{Rot}(q_{i-1}, q_i) \quad (3.2.11)$$

式中 $Q_{ij}^{(9)} \in R^{3 \times 3}$ 和 $Q_{ij}^{(9)} \in R^{3 \times 3}$ 分别对应矩阵 $[q_{ij}^{(9)} q_{ij}^{(2)} q_{ij}^{(1)}]$ 和 $[q_{ij}^{(9)} \cdots q_{ij}^{(1)}]$ 。很明显,式(3.2.11)的一般形式是

$$q_{ii} = Q_{ij}^{(9)} \text{Rot}(q_1, \cdots, q_i) \quad (3.2.12)$$

矩阵 $Q_{ij}^{(9)} \in R^{3 \times 3}$ 与关节坐标无关,但它是运动学参数的函数。向量 $\text{Rot}(q_1, \cdots, q_i) \in R^3$ 由笛卡儿乘积

$$(\cos q_1, \sin q_1, 1) \times \cdots \times (\cos q_i, \sin q_i, 1) \quad (3.2.13)$$

的元素组成。

现在,我们容易阐明变量的符号表示和数值-符号(解析)表示之间的区别。在这两种表示中,都假设关节坐标 q_1, \cdots, q_i 是未知的,因此, $\text{Rot}(q_1, \cdots, q_i)$ 的元素不能用实数代替,而必须保留符号。基本区别是矩阵 $Q_{ij}^{(9)}$ 中元素的不同。如果我们用 $Q_{ij}^{(9)}$ 的元素表示参数函数,那么就假设使用了符

号变量表示。如果用元素表示按给定操作手已知参数而计算出来的实数,那么我们就使用了变量的数值-符号形式。让我们考虑运动学变量集合 K_i 的其余元素; $R_i = \{r_{ii}, r_{i,i+1}\}$ 和 $E_i = \{e_i\}$ 给定为

$$\begin{aligned} r_{ii} &= Q_i \tilde{r}_{ii} + q_i e_i \xi_i \\ r_{i,i+1} &= Q_i \tilde{r}_{i,i+1} \\ e_{i+1} &= Q_i \tilde{e}_{i+1} \end{aligned} \quad (3.2.14)$$

把方程式(3.2.12)代入矩阵 $Q_i = [q_{i1} q_{i2} q_{i3}]$, 得到

$$Q_i = [Q_{ii}^{(q)} | Q_{i,i+1}^{(q)} | Q_{i3}^{(q)}] \text{Rot}(q_1, \dots, q_i) \quad (3.2.15)$$

把上式代入式(3.2.14), 得到

$$\begin{aligned} r_{ii} &= R_{ii}^{(q)} \text{Rot}(q_1, \dots, q_i) + q_i e_i \xi_i \\ r_{i,i+1} &= R_{i,i+1}^{(q)} \text{Rot}(q_1, \dots, q_i) \\ e_{i+1} &= E_{i+1}^{(q)} \text{Rot}(q_1, \dots, q_i) \end{aligned} \quad (3.2.16)$$

这些方程式的第一式表明, 式

$$v_i = v_i^{(q)} \text{Rot}(q_1, \dots, q_i) \quad (3.2.17)$$

不具有普遍性。此外, 通过对 K_i 的非线性运算得到的变量也不能用式(3.2.17)表示。例如, 标量积 $v_i \cdot v_i$ 就不能用式(3.2.17)表示, 因为它包含不计入 $\text{Rot}(q_1, \dots, q_i)$ 的一系列函数。因此, 我们引入更为一般的形式:

$$v = V^{(m)} \begin{bmatrix} (\cos q_1)^{c_{11}} \dots (\cos q_n)^{c_{n1}} (\sin q_1)^{s_{11}} \dots (\sin q_n)^{s_{n1}} q_1^{q_{11}} \dots q_n^{q_{n1}} \\ \vdots \\ (\cos q_1)^{c_{1m}} \dots (\cos q_n)^{c_{nm}} (\sin q_1)^{s_{1m}} \dots (\sin q_n)^{s_{nm}} q_1^{q_{1m}} \dots q_n^{q_{nm}} \end{bmatrix} \quad (3.2.18)$$

式中

$V^{(m)}$ 为 $l \times m$ 常阵(如果变量 v 是 3D 向量, 则 $l = 3$; 对标量, 则 $l = 1$);

$c_{ij}, s_{ij}, u_{ij} (i = 1, \dots, n; j = 1, \dots, m)$ 分别是余弦、正弦和关节坐标的指数。

很明显, 当 $m = 3^i$ 时, 式(3.2.18)容易简化为式(3.2.17)。请注意, m 并不固定为 $\text{Rot}(q_1, \dots, q_i)$ 的维数。一般, 对用式(3.2.17)表示的向量, $m \leq 3^i$ 成立。维数 3^i 包括了关节坐标的余弦和正弦的所有可能组合。

式(3.2.18)的解析式可表示为

$$v = \sum_{k=1}^m V_k (\cos q_1)^{c_{1k}} \cdots (\cos q_n)^{c_{nk}} (\sin q_1)^{s_{1k}} \cdots (\sin q_n)^{s_{nk}} q_1^{u_{1k}} \cdots q_n^{u_{nk}} \quad (3.2.19)$$

式中 V_k 是矩阵 $V^{(m)}$ 的第 k 列。让我们引入新的自变量向量

$$(x_1, \dots, x_N) = (\cos q_1, \dots, \cos q_n, \sin q_1, \dots, \sin q_n, q_1, \dots, q_n) \quad (3.2.20)$$

和指数向量

$$(\varepsilon_{1k}, \dots, \varepsilon_{Nk}) = (c_{1k}, \dots, c_{nk}, s_{1k}, \dots, s_{nk}, u_{1k}, \dots, u_{nk}) \quad (3.2.21)$$

显然, 式中 $\varepsilon_{jk}, j = 1, \dots, n$ 表示正整数或零。把式(3.2.20)和(3.2.21)代入式(3.2.19), 得到

$$v = \sum_{k=1}^m V_k x_1^{\varepsilon_{1k}} \cdots x_N^{\varepsilon_{Nk}}, \quad N = 3n \quad (3.2.22)$$

这种形式称为多项式数值-符号式, 或简称为多项式。

如果我们要在数字计算机上实现式(3.2.22), 那么就必须把它变换成适当的数值形式。由于这个缘故, 我们需引入多项式(3.2.22)的矩阵表示, 并把它称为“多项式矩阵”。

多项式矩阵的定义

由多项式(3.2.22)给定的变量 ν 的多项式矩阵表示一个对

$$S_\nu^{(n)} = (V^{(n)T}, E_\nu)$$

式中 T 表示向量的转置, $V^{(n)T}$ 是

$$V^{(n)T} = \begin{bmatrix} V_1^T \\ \vdots \\ V_m^T \end{bmatrix}$$

其中 V_i 由式(3.2.22)定义, E_ν 表示指数矩阵

$$E_\nu = \begin{bmatrix} c_{11} \cdots c_{n1} & s_{11} \cdots s_{n1} & u_{11} \cdots u_{n1} \\ \vdots & \vdots & \vdots \\ c_{1m} \cdots c_{nm} & s_{1m} \cdots s_{nm} & u_{1m} \cdots u_{nm} \end{bmatrix} \quad (3.2.23)$$

由上述定义式看出, 多项式矩阵完全定义了对应多项式的系数和指数。

运动学和动力学变量多项式矩阵的重要特性可通过如下定理表述。

定理

如果用 ν 表示机器人操作手的一个任意变量, 那么对应指数矩阵 E_ν 的元素属于集合 $\{0, 1, 2\}$, 对于运动学变量的情况, E_ν 的元素属于集合 $\{0, 1\}$ 。

在附录 3.1 中, 我们严格地证明了这个定理。

3.3 多项式矩阵代数

正如在上一节看到的那样, 机器人操作手的任何模型变

量都可用多项式

$$v = \sum_{k=1}^m V_k x_1^{k1} \cdots x_N^{kN}, \quad N = 3n$$

表示, 式中自变量 x_i 和 x_{n+i} 是相互关联的, 有

$$x_i^2 + x_{n+i}^2 = 1, \quad i \in N \quad (3.3.1)$$

这个变量也可用多项式矩阵

$$S_v^{(m)} = \left\{ \begin{bmatrix} V_1^T \\ \vdots \\ V_m^T \end{bmatrix}, \begin{bmatrix} \varepsilon_{11} \cdots \varepsilon_{M1} \\ \vdots \\ \varepsilon_{1m} \cdots \varepsilon_{Nm} \end{bmatrix} \right\} \quad (3.3.2)$$

表示. 让我们证明一个定理, 它描述多项式矩阵之间的数学运算.

定理

机器人操作手向量变量所对应的多项式矩阵集合构成一个向量空间.

证明

设 $V = \{S_v^{(m)} | v \in R^3, m \in N\}$ 是向量 $v \in R^3$ 的多项式矩阵集合, 我们用函数 G_+ 表示由 $V \times V$ 到 V 的变换. 假设

$$(S_1, S_2) \in V \times V$$

式中 S_i 表示 $S_{v_i}^{(m_i)}$. 引入加法

$$S_{v_1}^{(m_1)} + S_{v_2}^{(m_2)} = \left(\begin{bmatrix} V^{(m_1)T} \\ \hline V^{(m_2)T} \end{bmatrix}, \begin{bmatrix} E_{v_1} \\ \hline E_{v_2} \end{bmatrix} \right) \quad (3.3.3)$$

我们还定义函数 G , 它表示由 $R \times V$ 到 V 的变换. 如果 $(r, S) \in R \times V$, 那么把集合 V 的元素 $G_+((r, S))$ 写为 $r \cdot S$, 也就是

$$rS_p^{(m)} = G_+((r, S_p^{(m)}))$$

量 $rS_p^{(m)}$ 称为 r 和 $S_p^{(m)}$ 的积。当 r 是实数时, 函数 G_+ 定义为

$$rS_p^{(m)} = ([rV^{(m)T}], E_r) \quad (3.3.4)$$

当 r 表示一个符号变量 x_i 时, 乘积定义为

$$x_i S_p^{(m)} = \left(V^{(m)T}, \begin{bmatrix} \varepsilon_{11} \cdots \varepsilon_{i-1,1} \varepsilon_{i1} + 1 \varepsilon_{i+1,1} \cdots \varepsilon_{N1} \\ \vdots \\ \varepsilon_{1m} \cdots \varepsilon_{i-1,m} \varepsilon_{im} + 1 \varepsilon_{i+1,m} \cdots \varepsilon_{Nm} \end{bmatrix} \right) \quad (3.3.5)$$

$$i \in \{1, \dots, N\}$$

集合 V 表示一个向量空间, 如果关系式 G_+ 满足下列条件:

1) 对 V 中任意元素 S_1 和 S_2 , 有

$$S_1 + S_2 = S_2 + S_1$$

2) 对 V 中任意元素 S_1, S_2 和 S_3 , 有

$$S_1 + (S_2 + S_3) = (S_1 + S_2) + S_3$$

3) 在 V 中存在一个唯一的零元素, 使得对 V 中任意元素 S , 都满足

$$S + 0 = 0 + S = S$$

4) 在 V 中对任意元素 S , 都存在一个唯一的元素 $-S$, 使得

$$S + (-S) = (-S) + S = 0$$

利用函数 G_+ 定义式(3.3.3)容易证明条件 1)和 2)。零元素是

$$S_0^{(m)} = ([000], [0 \cdots 0])$$

逆元素 $-S$ 是

$$-S_p^{(m)} = (-V^{(m)T}, E_r)$$

多项式矩阵唯一性问题将在下一节中考虑。

函数 G 满足下列条件:

1) 对 R 中任意元素 r_1, r_2 和 V 中任意元素 S , 有

$$r_1 \cdot (r_2 \cdot S) = (r_1 r_2) \cdot S$$

2) 对 V 中任意元素 S , 有

$$1 \cdot S = S$$

3) 对 R 中任意元素 r 和 V 中任意元素 S_1, S_2 , 有

$$r \cdot (S_1 + S_2) = r \cdot S_1 + r \cdot S_2$$

4) 对 R 中任意元素 r_1, r_2 和 V 中任意元素 S , 有

$$(r_1 + r_2) \cdot S = r_1 \cdot S + r_2 \cdot S$$

我们利用函数 G 的定义式(3.3.4)和(3.3.5)容易证明这些条件。下面, 我们把 $r \cdot S$ 简写为 rS 。这就完成了定理的证明。

现在, 让我们引入多项式矩阵的标量积和向量积。向量积定义为

$$S_{v_1}^{(m_1)} \times S_{v_2}^{(m_2)} = \sum_{k=1}^{m_1} \sum_{l=1}^{m_2} (v_{1k} \times v_{2l}) x^{\varepsilon_{1k}^1 + \varepsilon_{2l}^1} \dots x_N^{\varepsilon_{1k}^1 + \varepsilon_{2l}^1} \quad (3.3.6)$$

式中 ε_{1k}^1 属于指数矩阵 E_{v_1} , ε_{2l}^1 属于指数矩阵 E_{v_2} 。利用多项式矩阵, 多项式(3.3.6)可表示为

$$\begin{aligned} & (V^{(m_1)T}, E_{v_1}) \times (V^{(m_2)T}, E_{v_2}) \\ & = ((V^{(m_1)} \times V^{(m_2)})^T, E_{v_1} + E_{v_2}) \end{aligned} \quad (3.3.7)$$

式中 $V^{(m_1)} \times V^{(m_2)} \in R^{3 \times (m_1 m_2)}$ 是一个矩阵, 它的列向量是 $v_{1k} \times v_{2l}$, $k = 1, \dots, m_1$, $l = 1, \dots, m_2$, $E_{v_1} + E_{v_2}$ 也是一个矩阵, 它的行向量是 $[\varepsilon_{1k}^1 + \varepsilon_{2l}^1 \dots \varepsilon_{1k}^1 + \varepsilon_{2l}^1]$

类似地, 我们把标量积 $S_1 \cdot S_2$ 定义为

$$\begin{aligned} & (V^{(m_1)T}, E_{v_1}) \cdot (V^{(m_2)T}, E_{v_2}) \\ & = ((V^{(m_1)} \cdot V^{(m_2)})^T, E_{v_1} + E_{v_2}) \end{aligned} \quad (3.3.8)$$

标量变量 s 的多项式矩阵与向量变量 v 的多项式矩阵的积定义为

$$(V_i^{(m_1)})^T, E_r)(V_i^{(m_2)})^T, E_s) = ((V_i^{(m_1)} V_i^{(m_2)})^T, E_r + E_s) \quad (3.3.9)$$

例

上面建立的所有运算都能容易地应用到构造机器人建模算法的任意表示式。例如,用解析形式表示的有限旋转公式(2.2.4)变为

$$S_{q_{ii}} = S_{q_i} \times S_{q_i} \xi_i + S_{q_i} x_{i+1} \xi_i + (S_{q_i} \cdot S_{q_{ii}}) S_{q_i} \xi_i + S_{q_{ii}} \xi_i \quad (3.3.10)$$

式中 $S_{q_i} = S_{q_{ii}} \times S_{q_i}$ 。

3.4 多项式矩阵的优化

考虑用多项式形式表示的变量 v

$$v = \sum_{k=1}^m V_k x_1^{N_1 k} \cdots x_N^{N_N k}$$

式中

$$\begin{aligned} N &= 3n \\ x_i &= \cos q_i \\ x_{n+i} &= \sin q_i, \\ x_{2n+i} &= q_i \end{aligned} \quad i \in N \quad (3.4.1)$$

显然,多项式自变量 $x_j, j = 1, \dots, 3n$ 是相互关联的。例如,人所共知的正弦和余弦函数之间的三角关系式有下列等式:

$$\begin{aligned} x_i^2 &= 1 - x_{n+i}^2 \\ x_i^4 &= 1 - 2x_{n+i}^2 + x_{n+i}^4 \end{aligned} \quad (3.4.2)$$

另一方面,让我们考虑多项式

$$w = \sum_{k=1}^{m+1} w_k x_1^{k-1} \cdots x_N^{k-1}$$

并假设

$$w_k = v_k, \quad k \in \{1, \dots, m\}$$

$$w_{m+1} = w_m$$

$$\xi_{ik} = \varepsilon_{ik}, \quad j \in \{1, \dots, N\}, \quad k \in \{1, \dots, m-1\}$$

$$\xi_{im} = \varepsilon_{im}, \quad j \in \{1, \dots, i-1, i+1, \dots, N\}$$

$$\varepsilon_{im} = \varepsilon_{im} + 2$$

$$\xi_{i,m+1} = \varepsilon_{jm}, \quad j \in \{1, \dots, n+i-1, n+i+1, \dots, N\}$$

$$\varepsilon_{n+i,m} = \varepsilon_{n+i,m} + 2$$

考虑到关系式(3.4.2)中的第一式,我们看到,多项式 v 和 w 是恒等的。它们对应的多项式矩阵也是等价的,即

$$S_v^{(m)} = S_w^{(m+1)}$$

$$(V_v^{(m)T}, E_v) = (V_w^{(m+1)T}, E_w)$$

可见,我们能够用两个不同的多项式表示同一个变量 v 。通过数学归纳法,我们能导出如下结论:对任意数 $m_1 \geq m$, 必存在 v_1 , 使得

$$(v_v^{(m)T}, E_v) = (v_{v_1}^{(m_1)T}, E_{v_1}) \quad (3.4.3)$$

也就是说,存在无限多个不同的多项式矩阵对应同一个变量 v 。这些多项式矩阵

$$S_{v_1}^{(m_1)} = (V_{v_1}^{(m_1)T}, E_{v_1}) \quad (3.4.4)$$

都是等价的。上述矩阵是 m_1 维的,但具有特殊意义,它是确定对应变量 v 的最小维多项式矩阵

$$S_v^{(m)} = \min_{m_1} (V_{v_1}^{(m_1)T}, E_{v_1}) \quad (3.4.5)$$

对应这个矩阵的多项式称为变量 ν 的最小式。对给定变量 ν ，如果满足下列三个条件，这个最小式则被认为是唯一的。

- 1) 在 $V_k^{(m)T}$ 中，不存在这样两个行，它们都是零向量；
- 2) 在 E_k 中，没有恒等的行；
- 3) 不可能找到这样两个行，它们能合并成一个行，条件是对任意脚标 i ，满足恒等式

$$x_i^2 + x_{i+i}^2 = 1$$

换句话说，不存在这样两个行，当 $\nu_k = \nu_j$ 时，它们的单项式

$$M_k = V_k x_1^{k_1} \cdots x_N^{k_N}$$

$$M_j = V_j x_1^{j_1} \cdots x_N^{j_N}$$

满足恒等式

$$\begin{aligned} M_k + M_j &= \nu_k x_1^{k_1} \cdots x_N^{k_N} (x_i^2 + x_{i+i}^2) \\ &= \nu_{k+i} x_1^{k_1} \cdots x_N^{k_N} \end{aligned}$$

如果条件 1) 不满足，则像矩阵 E_k^T 的对应行一样，矩阵 $V_k^{(m)T}$ 的零行可以剔除，使矩阵维数 m 减小。如果条件 2) 不满足，也就是在 E_k 中存在两个恒等的行，那么可在矩阵 V_k^T 中，把这两行相加得到一新行，使矩阵维数减少一维。

最后的结论表明，对应最小式的多项式矩阵是变量 ν 的唯一表示式。下面讲到的多项式矩阵向量空间的概念就是指最小式空间。实际上，这个向量空间满足唯一性和逆元素的特性（参阅 3.3 节）。从任意多项式矩阵出发，确定最小式的方法如图 3.1 所示。

例

考虑自由度 $n = 3$ 的操作手，它的多项式矩阵为

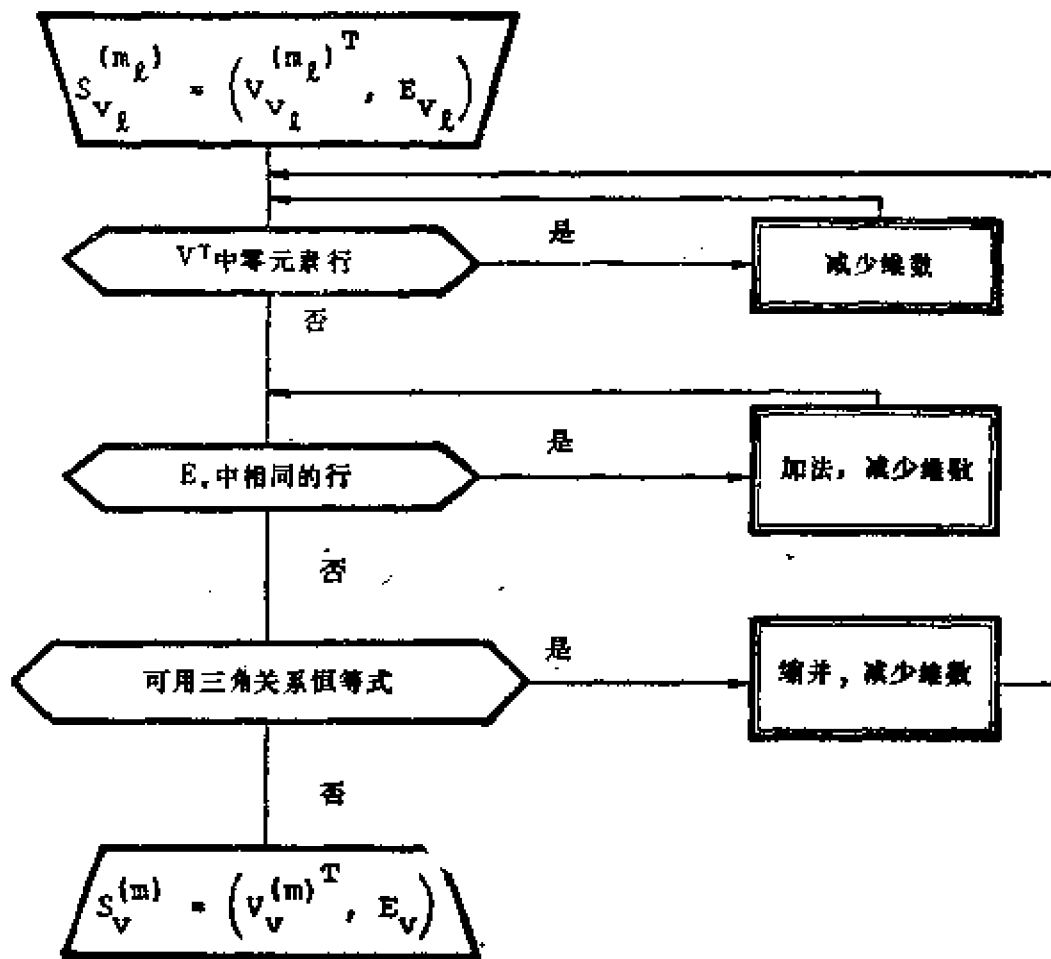


图 3.1 多项式矩阵维数的最小化

$$S_V^{(6)} = \left(\begin{array}{c} 1. \\ -3. \\ 2. \\ -3. \\ 2. \\ 0. \end{array} \right), \left(\begin{array}{ccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \end{array} \right)$$

应用上述方法, 我们可消去 V^T 中的零元素和 E_6 的对应行 (第六行)。根据三角关系式

$$x_3^2 + x_4^2 = 1$$

我们把第二行和第四行缩并为一行

$$\left(\begin{bmatrix} -3. \\ -3. \end{bmatrix}, \begin{bmatrix} 0 & 0 & 2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 2 & 0 & 0 & 0 \end{bmatrix} \right) \\ = ([-3.], [000 \quad 100 \quad 000])$$

而得到的新行与 E_5 的第五行相同, 因此, 这两行又可合并为一行

$$([-1.], [000 \quad 100 \quad 000])$$

最后, 我们得到对应最小式的矩阵

$$S_2^{(3)} = \left(\begin{bmatrix} 1. \\ -1. \\ 2. \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \right)$$

我们看到, 可把最小化过程形式化, 并且便于在数字计算机上实现。

3.5 非线性模型

我们在2.3节中曾阐述了建立非线性动力学模型的方法, 在那里, 不仅把关节坐标和关节速度, 而且把所有参数都用数值量处理。因此, 得到的动力学模型矩阵的元素也用数值量表示。按照这种观点, 这种算法属于一类数值方法。另外, 我们在3.2节中引入了多项式和多项式矩阵, 用它们作为机器人变量的数值-符号表示式。最后, 我们在3.3节中介绍了多项式矩阵代数。在这一节中, 我们将应用上述理论导出机器人的解析模型。

我们假设操作手参数 $K_i = (\tilde{Q}_i, \tilde{R}_i, \tilde{E}_i)$ 和 $D_i = (m_i,$

J_i), $i \in N$ 都已给定(见图 2.7)。另外假设关节坐标的数值是未知的, 并把它们作为符号变量处理。

推导机器人闭式动力学模型的基本方法包括三个阶段。首先, 在所有关节坐标等于零的条件下, 计算描述机器人构形的向量。然后, 在给定某些关节坐标条件下, 计算构形向量。最后, 计算动力学模型矩阵的元素。

当零关节坐标的构形与关节坐标向量 q 无关时, 正像在基本方法中描述的那样(图 2.7), 可对任何位置的向量进行数值计算。因此, 可把确定运动学变量集合

$$K_i = (Q_i^0, R_i^0, E_i^0), i \in N$$

看作是已知量(对 $q = 0$)。

现在让我们考虑计算机机构实际构形 ($q \neq 0$) 这一阶段。注意到多项式矩阵

$$S_{e_i^0} = (e_i^0, [0 \cdots 0])$$

$$S_{q_{ij}} = (q_{ij}^0, [0 \cdots 0]), j = 1, 2, 3$$

已经确定。我们采用如下方法应用有限旋转公式。首先, 把所有连杆对轴 e_1 一齐旋转一个角度 q_1 。在旋转运动链过程中, 假设连杆的相对布局保持不变, 虽然运动链对参考系的布局已经改变。接着, 除第一个连杆之外, 把所有连杆对轴 e_2 一齐旋转一个角度 q_2 。然后, 类似地, 除前两个连杆外, 把所有连杆对轴 e_3 一齐旋转一个角度 q_3 , 等等(图 1.2)。我们用 I 表示对它做了旋转的关节标志。在 e_i 旋转前, q_{ij}^0 的多项式矩阵表示为 $S_{q_{ij}}^I$ 。我们引入多项式矩阵 S_{e_i} , 并且用

$$S_q = S_{e_i} \times S_{q_{ij}}^I$$

表示操作手在绕 e_i 旋转前的构形。现在, 我们就可应用解析形式的有限旋转公式(对 $i \geq 1$)

$$S_{q_{ij}^{(l+1)}} = [S_{e_l} \times S_{q_{x_l}} + S_{q_{x_{n+l}}} + (S_{e_l} \cdot S_{q_{ij}^{(l)}})S_{e_l}] \xi_l + S_{q_{ij}^{(l)}} \xi_l \quad (3.5.1)$$

为了对 $l = 1, \dots, n$ 应用关系式(3.5.1), 必须建立 $S_{e_{l+1}}$. 利用式(2.2.3), 得到

$$e_{l+1} = [q_{l1}^{(l+1)} q_{l2}^{(l+1)} q_{l3}^{(l+1)}] \tilde{e}_{l+1}$$

或

$$e_{l+1} = \sum_{i=1}^3 \tilde{e}_{l+1}^{(i)} q_{li}^{(l+1)} \quad (3.5.2)$$

式中 $\tilde{e}_{l+1}^{(i)}$ 表示 \tilde{e}_{l+1} 的第 i 个分量. 当 \tilde{e}_{l+1} 表示参数向量, 而它的分量是数值量时, 则式(3.5.2)在数值-符号域变为

$$S_{e_{l+1}} = \sum_{i=1}^3 \tilde{e}_{l+1}^{(i)} S_{q_i^{(l+1)}} \quad (3.5.3)$$

利用有递推脚标 l 的关系式(3.5.1)和(3.5.3), 我们容易由 Q_i 和 $E_i, i \in N$ 得到向量的多项式矩阵. 由运动学变量集合看, 尚需确定 $R_i = \{r_{ii}, r_{ii+1}\}$ 的多项式矩阵. 当

$$r_{ii} = Q_i \tilde{r}_{ii} + q_i e_i \xi_i$$

或

$$r_{ii} = [q_{i1} q_{i2} q_{i3}] \tilde{r}_{ii} + q_i e_i \xi_i$$

或等价于

$$r_{ii} = \sum_{l=1}^3 \tilde{r}_{ii}^{(l)} q_{il} + q_i e_i \xi_i$$

那么 r_{ii} 的数值-符号表示式变为

$$S_{r_{ii}} = \sum_{l=1}^3 \tilde{r}_{ii}^{(l)} S_{q_{il}} + \xi_i q_i S_{e_i} \quad (3.5.4)$$

式中 $\tilde{r}_{ii}^{(l)}$ 是 \tilde{r}_{ii} 的第 l 个分量, 及

$$\xi_i q_i S_{e_i} = \left(e_i, E_{e_i} + \begin{bmatrix} 0 \cdots 0 & \xi_i^{(2n+i)} & 0 \cdots 0 \\ \vdots & \vdots & \vdots \\ 0 \cdots 0 & \xi_i & 0 \cdots 0 \end{bmatrix} \right)$$

类似地, 向量

$$r_{i,i+1} = Q_i \tilde{r}_{i,i+1}$$

也就是

$$r_{i,i+1} = \sum_{l=1}^3 \tilde{r}_{i,i+1}^{(l)} q_{il}$$

可在数值-符号域写为

$$S_{r_{i,i+1}} = \sum_{l=1}^3 \tilde{r}_{i,i+1}^{(l)} S_{q_{i,l}} \quad (3.5.5)$$

这样, 我们就确定了所有连杆的位置。下一步, 我们确定动力学模型矩阵元素的多项式矩阵。为了达到这一目的, 对由 $K_i, i \in N$ 的非线性运算所得的变量表示式(3.2.1)进行考察。把这些表示式变换到数值-符号域, 得到

$$S_{e_{ij}} = S_{e_i} \cdot S_{e_j} \quad (3.5.6)$$

$$S_{(e,r)ij} = S_{e_j} \cdot S_{r_{ij}} \xi_i + S_{e_j} \xi_i \quad (3.5.7)$$

$$S_{r_{ij\mu}} = S_{e_j} \cdot S_{q_{i\mu}} \quad (3.5.8)$$

其中

$$S_{r_{ij}} = S_{r_{ij}} = S_{r_{j,j+1}} + S_{r_{j+1,j+1}} + \cdots + S_{r_{ii}} \quad (3.5.9)$$

当然, 式(3.5.6)~(3.5.9)中的每个代数运算都包括维数的最小化。

按照图 2.7 中给出的算法, 直接得到

$$S_{H_{ik}} = \sum_{j=\max(i,k)}^n \left[m_j S_{(e,r)ij} \cdot S_{(e,r)jk} + \sum_{\mu=1}^3 S_{r_{i\mu}} \cdot S_{r_{k\mu}} J_{i\mu} \xi_i \xi_k \right], \quad i, k \in N \quad (3.5.10)$$

类似地,得到

$$S_{c_{ij}} = \sum_{j=\max(i,k)}^n \{m_j S_{(c,r)ji} \cdot (S_{e_i} \times S_{(c,r)ik}) + \frac{1}{2} \sum_{\mu=1}^3 (S_{p i \mu} S_{o j k} + S_{p k \mu} S_{o i l} + S_{p l \mu} S_{o j k}) \cdot S_{q i \mu} J_{i \mu} \xi_i \xi_k\} \xi_l, \quad i, k, l \in N \quad (3.5.11)$$

重力向量 $h_c(q, \theta)$ 的元素变为

$$S_{h_c} = - \sum_{j=i}^n (m_j S_{(c,r)ji} \cdot g) \quad (3.5.12)$$

式中 g 表示常值重力向量。这样,我们就得到了用解析形式表示的任意动力学模型矩阵的元素。

3.6 线性化和灵敏度模型

在本书第二章中,我们研究了模型线性化问题,并得到了偏导数矩阵

$$\frac{\partial H(q, \theta)}{\partial q}, \quad \frac{\partial C(q, \theta)}{\partial q}, \quad \frac{\partial h_c(q, \theta)}{\partial q}$$

公式推导是利用在 2.5 节中提出和证明的三个定理实现的,而没有采用具有累积误差的数值微分法。图 2.10 中总结了上述算法,它包括两个步骤。第一步,计算运动学变量集合向量的偏导数。第二步,计算动力学模型矩阵的偏导数。现在,我们利用 3.3 节中提出的定理,把这两步变换到符号域。例如,第一步变为

$$S_{(c,r)ij}^{(p)} = \{(S_{e_{ij}} \times S_{r_{ij}} + S_{z_{ij}}) \eta_{ij} \xi_i + [S_{e_i} \times (S_{r_i} \times S_{r_{ij}}) \xi_i + S_{e_{ij}} \xi_i \eta_{ij}]\} \eta_{ij} \xi_i$$

$$S_{q_{ij}^{(r)}} = (S_{ax_i} \times S_{ej} \eta_{ej} + S_{ej} \times S_{e_{ij}} \eta_{ij}) \dot{E}_i \quad (3.6.1)$$

$$S_{p_{ij}^{(r)}} = S_{q_{ij}^{(r)}} \cdot S_{a_{ij}} (\eta_{ij} - \eta_{ji}) \dot{E}_i$$

式中上标 (r) 表示对关节坐标 q_i 的偏导数。类似地，我们可变换第二步骤中的方程。但是，尽管原理简单，这种方法在计算上仍是不可行的，因为在多项式矩阵之间的运算量远大于(约 $2n$ 倍) 构造非线性模型基本方法的运算量。这就使我们想到，与其那样，还不如直接研究在符号域计算多项式矩阵导数的算法。因而，我们假设，我们已经利用数字计算机获得了动力学模型矩阵所有元素的多项式矩阵。用 h 表示动力学模型矩阵的任意元素，那么它对应的多项式矩阵表示为

$$S_h = (h^{(m)}, E_h)$$

式中 $h^{(m)} \in R^{m \times 1}$, $E_h \in R^{m \times 3n}$ 。得到的结果是，线性化模型的导数可简化为计算偏导数

$$h^{(r)} = \frac{\partial h(q, \theta)}{\partial q_i}$$

或用符号式表示为

$$S_h^{(r)} = \frac{\partial}{\partial q_i} (h^{(m)}, E_h) = (h^{(m,r)}, E_h^{(r)}) \quad (3.6.2)$$

多项式矩阵 $S_h^{(r)}$ ，即 S_h 对 q_i 的偏导数可用如下定理计算。

定理

设 $S_h = (h^{(m)}, E_h)$ 是多项式矩阵，它的第 k 行元素给定

$$([h_k], [e_{1k} \cdots e_{3n,k}]), k \in \{1, \cdots, m\} \quad (3.6.3)$$

那么对应变量 $\partial h / \partial q_i$ 的多项式矩阵，可通过用

$$\begin{pmatrix} -h_k \varepsilon_{1k} \\ h_k \varepsilon_{n+i,k} \\ h_k \varepsilon_{2n+i,k} \end{pmatrix}, \begin{pmatrix} \cdots \varepsilon_{1k} - 1 \cdots \varepsilon_{n+i,k} + 1 \cdots 0 \cdots \\ \cdots \varepsilon_{1k} + 1 \cdots \varepsilon_{n+i,k} - 1 \cdots 0 \cdots \\ \cdots 0 \cdots 0 \cdots \varepsilon_{2n+i,k} - 1 \cdots \end{pmatrix} \\ k = 1, \dots, m \quad (3.6.4)$$

代替第 k 行(3.6.3)获得。其后就可应用维数最小化算法了。

证明

考虑多项式

$$h = \sum_{k=1}^m h_k x_1^{2k} \cdots x_{2n+i}^{2k} \quad (3.6.5)$$

和描述 q_i 函数的变量

$$x_1 = \cos q_i,$$

$$x_{n+i} = \sin q_i,$$

$$x_{2n+i} = q_i,$$

因此,有

$$\begin{aligned} \frac{\partial x_1}{\partial q_i} &= -x_{n+i} \\ \frac{\partial x_{n+i}}{\partial q_i} &= x_1 \\ \frac{\partial x_{2n+i}}{\partial q_i} &= 1 \end{aligned} \quad (3.6.6)$$

把式(3.6.5)对 q_i 进行微分,并考虑到式(3.6.6),得到

$$\frac{\partial h}{\partial q_i} = \frac{\partial h}{\partial x_1} x_{n+i} + \frac{\partial h}{\partial x_{n+i}} x_1 + \frac{\partial h}{\partial x_{2n+i}}$$

从而有

$$\frac{\partial h}{\partial q_i} = \sum_{k=1}^m (-h_k \varepsilon_{1k} x_1^{2k} \cdots x_{n+i}^{2k-1} x_{2n+i}^{-1} x_1^{2k+1} \cdots$$

$$\begin{aligned}
& x_{n+1, k}^{\varepsilon_{n+1, k}} x_{n+2, k}^{\varepsilon_{n+2, k}+1} x_{n+3, k}^{\varepsilon_{n+3, k}+1} \cdots x_{1, k}^{\varepsilon_{1, k}} \\
& + h_k \varepsilon_{n+1, k} x_1^{\varepsilon_{1, k}} \cdots x_{i-1, k}^{\varepsilon_{i-1, k}} x_i^{\varepsilon_{i, k}+1} x_{i+1, k}^{\varepsilon_{i+1, k}} \cdots \\
& \cdots x_{n+1, k}^{\varepsilon_{n+1, k}+1} x_{n+2, k}^{\varepsilon_{n+2, k}} x_{n+3, k}^{\varepsilon_{n+3, k}+1} \cdots x_{1, k}^{\varepsilon_{1, k}} \\
& + h_k \varepsilon_{2n+1, k} x_1^{\varepsilon_{1, k}} \cdots x_{2n+2, k}^{\varepsilon_{2n+2, k}+1} x_{2n+3, k}^{\varepsilon_{2n+3, k}+1} x_{2n+4, k}^{\varepsilon_{2n+4, k}} \cdots x_{1, k}^{\varepsilon_{1, k}}
\end{aligned}$$

这就证明了定理。

请注意，这个定理的实际应用是很简单的。由式(3.6.4)可以断定，计算 $\partial h / \partial q_i$ 需要 $3m$ 次乘法和 $5m$ 次加法/减法运算。另一方面我们注意到，在式(3.6.4)中，项 $h_k \varepsilon_{i, k}$ 实际上只包括浮点加法，这是因为 $\varepsilon_{i, k}$ 属于集合 $\{0, 1, 2\}$ 。由于对项 $h_k \varepsilon_{n+1, k}$ 和 $h_k \varepsilon_{2n+1, k}$ 也有同样结果，因此，计算偏导数 $\partial h / \partial q_i$ 既不需要浮点乘法运算，也不需要定点乘法运算。因为指数矩阵常常包含很多零元素，所以它的加法运算次数比前述方法少得多。这样求得的多项式矩阵 $S_k(s)$ 的行数等于 $3m$ 。鉴于所得结果并不是最小维数矩阵，因此必须用 3.4 节中描述的算法(图 3.1)使维数最小化，从而得到

$$S_k^{(M)} = \min_{m'} S_k^{(m')}$$

式中 M 是最小维数， $M \leq 3m$ 。

例

让我们计算多项式矩阵

$$S_k^{(3)} = \left(\begin{bmatrix} 1. \\ 1. \\ 2. \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \right)$$

对 q_1 的偏导数。通过直接应用上述定理，得到

$$S_k^{(2)} = \left(\begin{bmatrix} 1. \\ -2. \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \right)$$

很明显, $S_{(i,0)}^{(2)}$ 有最小行数。

现在, 让我们推导用解析形式表示的闭式灵敏度模型。正像在 2.6 节中指出的那样, 灵敏度模型确定了驱动转矩 (力) 的增量——一个向量, 它是在 q, \dot{q} 和 \ddot{q} 不偏离标称值的条件下, 补偿机器人某连杆 ($i = s$) 动力学参数偏差

$$\Delta\theta_s = \{\Delta m_s, \Delta J_{s1}, \Delta J_{s2}, \Delta J_{s3}\}$$

所需要的。可以证明, 这个连杆的线性化动力学参数表示为

$$\Delta P = \sum_{\mu=0}^{\mu_d} (H_s^\mu(q)\ddot{q} + \dot{q}^T C_s^\mu(q)\dot{q} + h_s^{\mu+1}(q))\Delta\theta_s(\mu+1)$$

式中 H_s^μ, C_s^μ 和 $h_s^{\mu+1}$ 由图 2.12 总结的公式确定。通过检查这些公式, 我们看到, 这些公式是利用集合 $K_i, i \in N$ 的运动学变量之间的非线性组合建立起来的。把这些公式变换成解析形式, 就得到关系式(3.5.6)——(3.5.9)。随后, 我们得到惯性矩阵的元素

$$S_{H_{ik}^\mu} = S_{(i,\mu)il} \cdot S_{(i,\mu)ik} \overline{\eta_{(i,k),\mu}} \quad (3.6.7)$$

$$S_{H_{ik}^\mu} = S_{pl\mu} S_{pk\mu} \xi_i \xi_k \overline{\eta_{(i,k),\mu}}$$

式中 $i, k \in N, \mu = \{1, 2, 3\}$ 。

类似地, 我们得到 C-矩阵的元素

$$S_{C_{ikl}^{j\mu}} = S_{(i,\mu)il} \cdot (S_{e_l} \times S_{(i,\mu)ik}) \xi_l \overline{\eta_{(i,k),\mu}} \quad (3.6.8)$$

$$S_{C_{ikl}^{j\mu}} = \frac{1}{2} (S_{pl\mu} S_{iik} + S_{pk\mu} S_{iil} + S_{pl\mu} S_{iik}) \cdot S_{q,\mu} \xi_i \xi_k \xi_l \overline{\eta_{(i,k),\mu}} \quad (3.6.9)$$

式中 $i, k, l \in N, \mu \in \{1, 2, 3\}$ 。

最后, 我们得到重力向量的元素

$$S_{h_s^{\mu+1}(i)} = S_{(i,\mu)il} \overline{g} \eta_{ik} \quad (3.6.10)$$

让我们注意，灵敏度模型矩阵具有和相应非线性模型矩阵同样的特性，总结如下：

- 1) 灵敏度模型的多项式矩阵与动力学参数独立；
- 2) 矩阵 $S_{H_{ik}^{\mu}}$, $\mu = 0, \dots, 3$ 是对称的和正定的；
- 3) 矩阵 $S_{c_{ikl}^{\mu}}$ 对指标集 k 和 l 是对称的，对上标 i 和脚标 k 是反对称的，即

$$S_{c_{ikl}^{\mu}} = -S_{c_{lki}^{\mu}}, \quad k, i \geq l$$

3.7 近似模型

前面已经指出，机器人操作手的近似数学模型在中型或现代微型计算机上进行实时应用起着十分重要的作用。与精确模型相比，近似模型只需较少的浮点乘法/加法运算。但是，在采用近似模型时，由于利用数学判据^[25,27]或忽略某些动力学效应^[59,60]，必然显含或隐含地引入一定的误差。在这一节中，我们要提出一种简单方法，以便获得用解析形式表示的近似模型。

设 $S_{H_{ik}^{\mu}}$, $S_{c_{ikl}^{\mu}}$ 和 $S_{h_i^{\mu}}$ 是用解析形式表示的动力学模型矩阵集合。如果用 h 表示动力学模型矩阵元素，那么对应的多项式矩阵为

$$S_k = (h^{(m)}, E_k)$$

式中 m 是矩阵行数。这个矩阵等价于多项式

$$h(x) = \sum_{k=1}^m h_k x_1^{i_1 k} \cdots x_N^{N k}$$

其中

$$x_i = \cos q_i$$

$$x_{n+i} = \sin q_i$$

$$x_{2n+i} = q_i$$

其中 $i \in N$. 让我们引入由变量集合 $x = \{x_i; i = 1, \dots, N\}$ 到域 $z = \{z_i; i = 1, \dots, N\}$ 的线性变换, 把它定义为

$$z_i = a_i x_i, \quad i = 1, \dots, N \quad (3.7.1)$$

式中系数 a_i 给定为

$$a_i = 1, \quad i = 1, \dots, 2n \quad (3.7.2)$$

$$a_i = \sup |q_i(t)|, \quad t \in R^+, \quad i = 2n+1, \dots, 3n \quad (3.7.3)$$

如果对关节坐标的机械设计约束给定为

$$\bar{q}_i \leq q_i \leq \bar{q}_i$$

那么式(3.7.3)就变为

$$a_i = \max(|q_i|, |\bar{q}_i|) \quad (3.7.4)$$

我们看到, 上述变换实际上是对变量规范化, 也就是, 有

$$|z_i| \leq 1, \quad i = 1, \dots, N \quad (3.7.5)$$

把式(3.7.1)代入多项式 $h(x)$, 得到

$$h(x) = \sum_{k=1}^m h_k^* z_1^{s_{1k}} \cdots z_N^{s_{Nk}} \quad (3.7.6)$$

式中

$$h_k^* = h_k a_1^{s_{1k}} \cdots a_N^{s_{Nk}} = h_k a_{1n}^{s_{1k}} \cdots a_{3n}^{s_{3k}}$$

由条件式(3.7.5), 得到

$$\sup_{\substack{z_i \in Z \\ (i=1, \dots, N)}} |h_k^* z_1^{s_{1k}} \cdots z_N^{s_{Nk}}| \leq |h_k^*| \quad (3.7.7)$$

下面, 我们着手做系数 h_k^* 的规范化. 为此目的, 首先确定使 $|h_k^*|$ 为最大的 $k \in \{1, \dots, m\}$, 即

$$h^* = \max_k |h_k^*| \quad (3.7.8)$$

然后把多项式(3.7.6)的系数如下规范化

$$h(z) = \bar{h}^* \sum_{k=1}^m x_k^* z_1^{e_{1k}} \cdots z_N^{e_{Nk}} \quad (3.7.9)$$

式中

$$x_k^* = \frac{h_k^*}{\bar{h}^*}$$

显然

$$|x_k^*| \leq 1, k \in \{1, \dots, m\} \quad (3.7.10)$$

于是,对应的多项式矩阵变为

$$h(x) = \left(\bar{h}^* \begin{bmatrix} x_1^* \\ \vdots \\ x_m^* \end{bmatrix}, \begin{bmatrix} \varepsilon_{11} & \cdots & \varepsilon_{N1} \\ \vdots & & \vdots \\ \varepsilon_{1N} & \cdots & \varepsilon_{NN} \end{bmatrix} \right) \quad (3.7.11)$$

请注意,上式中指数矩阵对应Z域。

现在,我们给出近似多项式函数的定义.称函数 $h_s(x)$ 与函数 $h(x)$ 近似,且误差小于或等于 ε , 如果不等式

$$|h(x) - h_s(x)| \leq \bar{h}^* \varepsilon \quad (3.7.12)$$

成立,其中 $\forall x \in Z$, $\varepsilon > 0$. 我们看到, ε 实际上表示最大相对误差。

现在,我们确立如下定理。

定理

相对误差不超过 ε 的,函数 $h(x)$ 的近似多项式 $h_s(x)$ 为

$$h_s(x) = \bar{h}^* \sum_{k \in K^c} x_k^* z_1^{e_{1k}} \cdots z_N^{e_{Nk}} \quad (3.7.13)$$

式中 K^c 表示 $K \subset \{1, \dots, m\}$ 的补集.由如下不等式定义

证明

由人所共知的不等式

$$\left| \sum_{k \in K} x_k^* \right| \leq \sum_{k \in K} |x_k^*| \quad (3.7.15)$$

开始,并注意到 $|z_i| \leq 1, i \in \{1, \dots, n\}$, 得到

$$\left| \sum_{k \in K} x_k^* z_1^{i_1 k} \cdots z_N^{i_N k} \right| \leq \sum_{k \in K} |x_k^*| \quad (3.7.16)$$

另一方面,按照式(3.7.13),有

$$\sum_{k \in K} x_k^* z_1^{i_1 k} \cdots z_N^{i_N k} = (h(x) - h_0(x)) / \bar{h}^* \quad (3.7.17)$$

把上式代入式(3.7.16),得到

$$|h(x) - h_0(x)| \leq \bar{h}^* \sum_{k \in K} |x_k^*| \quad (3.7.18)$$

考虑到式(3.7.14),我们看到

$$|h(x) - h_0(x)| \leq \bar{h}^* \varepsilon$$

上式实际上就是前述近似多项式的定义。

我们还可用其它的不同方法来定义相对误差 ε 。既然 h 是动力学模型矩阵的元素,那么就能用观测矩阵中的最大矩阵来定义相对误差。例如,考虑惯性矩阵元素

$$S_{H_{ik}} = (V_{H_{ii}}^{(n_{ik})^T}, E_{H_{ik}})$$

式中 $i \in N, k \in N$ 。对系数和变量进行规范化后,得到

$$S_{H_{ik}} = (\bar{H}_{ik}^* V_{H_{ii}}^{(n_{ik})^T}, E_{H_{ik}}^*)$$

式中 z 表示 Z 域。常数 \bar{H}^* 定义为

$$\bar{H}^* = \max_{i,k} (\bar{H}_{ik}^*)$$

这个常数是惯性矩阵的所有元素所共有的。

此外,相对误差 ε 也可用模型矩阵的行元素来定义,于

是,我们用

$$\bar{H}_i^* = \max_k (\bar{H}_{ik}^*)$$

代替 \bar{H}^* 。因为 H_{ik} ($k \neq i$) 表示在第 i 和第 k 个自由度之间的惯性耦合, 在这里我们(在已定义相对误差 ε 的意义上)忽略了耦合效应。

由上述定义可见,算法是很简单的,且便于计算机实现。

例

计算多项式矩阵

$$h = \left(\begin{array}{c} \left[\begin{array}{c} 22.5 \\ 14.1 \\ 0.2 \\ -0.1 \\ 7.6 \end{array} \right], \left[\begin{array}{cccccc} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{array} \right] \end{array} \right)$$

的近似矩阵,给定相对误差为 0.5%, 关节坐标 q_3 的机械设计约束给定为

$$-0.3 \leq q_3 \leq 0.3$$

首先,我们必须把 $h(x)$ 变换到 Z 域。按照式(3.7.2)和(3.7.4),有

$$a_i = 1, \quad i = 1, \dots, 6$$

$$a_i = 0.3, \quad i = 9$$

通过引入式(3.7.1),得到

$$h = \left(\begin{array}{c} \left[\begin{array}{c} 22.5 \\ 14.1 \\ 0.06 \\ -0.03 \\ 7.6 \end{array} \right], E \end{array} \right)$$

式中 E 表示初始指数矩阵, 在规范化过程中, 它是不变的。按照式(3.7.8)和(3.7.9), 得到

$$h = \left(22.5 \begin{bmatrix} 1 \\ 0.627 \\ 0.0027 \\ 0.0013 \\ 0.338 \end{bmatrix}, E \right)$$

因为

$$0.0027 + |-0.0013| < \varepsilon = 0.005$$

我们看到, 可消掉多项式矩阵 h 中的第三和第四行。最后, 得到问题的解为

$$h_{\varepsilon=0.005} = \left(22.5 \begin{bmatrix} 1 \\ 0.627 \\ 0.338 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \right)$$

附 录 3.1

在这里, 我们要建立一个关于模型变量多项式矩阵的基本定理。

定理

考虑简单开运动链形式的机器人操作手。设 v 表示任意的运动学或动力学变量。那么多项式矩阵

$$S_v^m = (v^{(m)}, E_v)$$

中 E_v 矩阵的元素必属于集合 $\{0, 1, 2\}$ 。

证明

在上一节中, 我们已经说明, 运动学变量可用 $\text{Rot}(q_1, \dots, q_n)$ 表示. 我们知道, Rot 元素也就是笛卡儿乘积式 (3.2.13) 的元素. 由于乘积的各个分量中, 没有相同的元素, 所以任意指数的最大值都不能超过 1. 因此, 运动学变量多项式矩阵的指数都属于集合 $\{0, 1\}$.

在构造机器人动力学模型过程中(图 2.7), 我们需要处理变量 r_{ij} , 它表示为

$$r_{ij} = r_{ij} - r_{i,j+1} + r_{i+1,j+1} - \dots + r_{ii}, i > j \quad (\text{A3.1.1})$$

把式(3.2.16)代入上式, 得到

$$\begin{aligned} r_{ij} = & \sum_{k=i}^{i-1} (R_{kk}^{(q_k)} - R_{i,k+1}^{(q_k)}) \text{Rot}(q_1, \dots, q_k) \\ & + \sum_{k=i}^i q_k e_k \xi_k + R_{ii}^{(q_i)} \text{Rot}(q_1, \dots, q_i) \end{aligned} \quad (\text{A3.1.2})$$

现在让我们证明, 如果 $k \leq i$, 那么 $\text{Rot}(q_1, \dots, q_k)$ 是 $\text{Rot}(q_1, \dots, q_i)$ 的子集(子向量). 设 $i = k + 1$, 则 $\text{Rot}(q_1, \dots, q_i)$ 变为

$$\begin{aligned} & (\cos q_{k+1}, \sin q_{k+1}, 1) \times (\cos q_k, \sin q_k, 1) \times \dots \\ & \times (\cos q_1, \sin q_1, 1) \end{aligned} \quad (\text{A3.1.3})$$

由于第一个集合内的零元素 1 与其余集合的所有元素之间的笛卡儿乘积等于集合 $\text{Rot}(q_1, \dots, q_k)$, 这就证明了 $i = k + 1$ 的情况. 应用数学归纳法, 容易证明 $k \leq i$ 的情况. $\text{Rot}(q_1, \dots, q_k)$ 的这一性质称为“包含特性”. 利用这一特性, 表示式(A3.1.3)变为

$$r_{ij} = R_{ii}^{(q_i)} \text{Rot}(q_1, \dots, q_i) + \sum_{k=i}^i q_k e_k \xi_k \quad (\text{A3.1.4})$$

我们看到,这个变量的指数属于集合 $\{0,1\}$ 。

现在,让我们考察变量式(3.2.1),它们是通过 $K_i, i \in N$ 的元素之间的非线性代数运算得到的。例如,向量

$$e_{ij} = e_i \times e_j, \quad i > j$$

就是这样一种向量。其中 e_i 和 e_j 是相对参考系给出的。首先,我们要确定对局部坐标系 Q_{i-1} 的向量 \tilde{e}_{ij} 。显然, \tilde{e}_i 是 Q_{i-1} 中的常向量,而 \tilde{e}_j 为(图3.2)

$$\tilde{e}_j = \tilde{E}_j^{(j)} \text{Rot}(q_1, \dots, q_i), \quad i = i - j \quad (\text{A3.1.5})$$

把 \tilde{e}_i 与常向量 \tilde{e}_j 相乘,得到

$$\tilde{e}_{ij} = \tilde{e}_{ij}^{(j)} \text{Rot}(q_1, \dots, q_i), \quad i = i - j \quad (\text{A3.1.6})$$

式中 $\tilde{e}_{ij}^{(j)}$ 表示对 Q_{i-1} 的常向量。因为对 Q_{i-1} 的任意常向量 a_{i-1} 都能表示为

$$a_{i-1} = A_{i-1}^{(i-1)} \text{Rot}(q_1, \dots, q_{i-1})$$

那么 $\tilde{e}_{ij}^{(j)}$ 的任一元素都可用同样方式表示。所以,我们能写出

$$e_{ij} = e_{ij}^{(j)} \text{Rot}(q_1, \dots, q_i), \quad i > j \quad (\text{A3.1.7})$$

换句话说, e_{ij} 的多项式矩阵的指数属于集合 $\{0,1\}$ 。

我们用同样方法考虑变量

$$\langle e, r \rangle_{ij} = e_j \times r_{ij} \tilde{e}_i + e_i \tilde{e}_j, \quad i > j$$

把式(A3.1.1)给出的 r_{ij} 代入上式,得到

$$\begin{aligned} \langle e, r \rangle_{ij} = & \sum_{k=i}^{i-1} (e_j \times r_{kk} - e_j \times r_{k,k+1}) \tilde{e}_j \\ & + e_j \times r_{ij} \tilde{e}_i + e_i \tilde{e}_j \end{aligned} \quad (\text{A3.1.8})$$

请注意, r_{kk} 和 $r_{k,k+1}$ 是对 Q_k 的常向量,并可用 $\text{Rot}(q_1, \dots, q_k)$ 把向量 $e_j \times r_{kk}$ 和 $e_j \times r_{k,k+1}$ 同样表示为 e_{ij} 。考虑到 Rot 函数的包含特性,我们看到,可用 $\text{Rot}(q_1, \dots, q_i)$ 表示向量 $\langle e, r \rangle_{ij}$ 。所以,对应 $\langle e, r \rangle_{ij}$ 的多项式矩阵的指数

属于集合 $\{0,1\}$ 。对向量 $e_i \times \langle e, \tau \rangle_{ii}, 1 \leq j \leq i$, 也有同样结论。

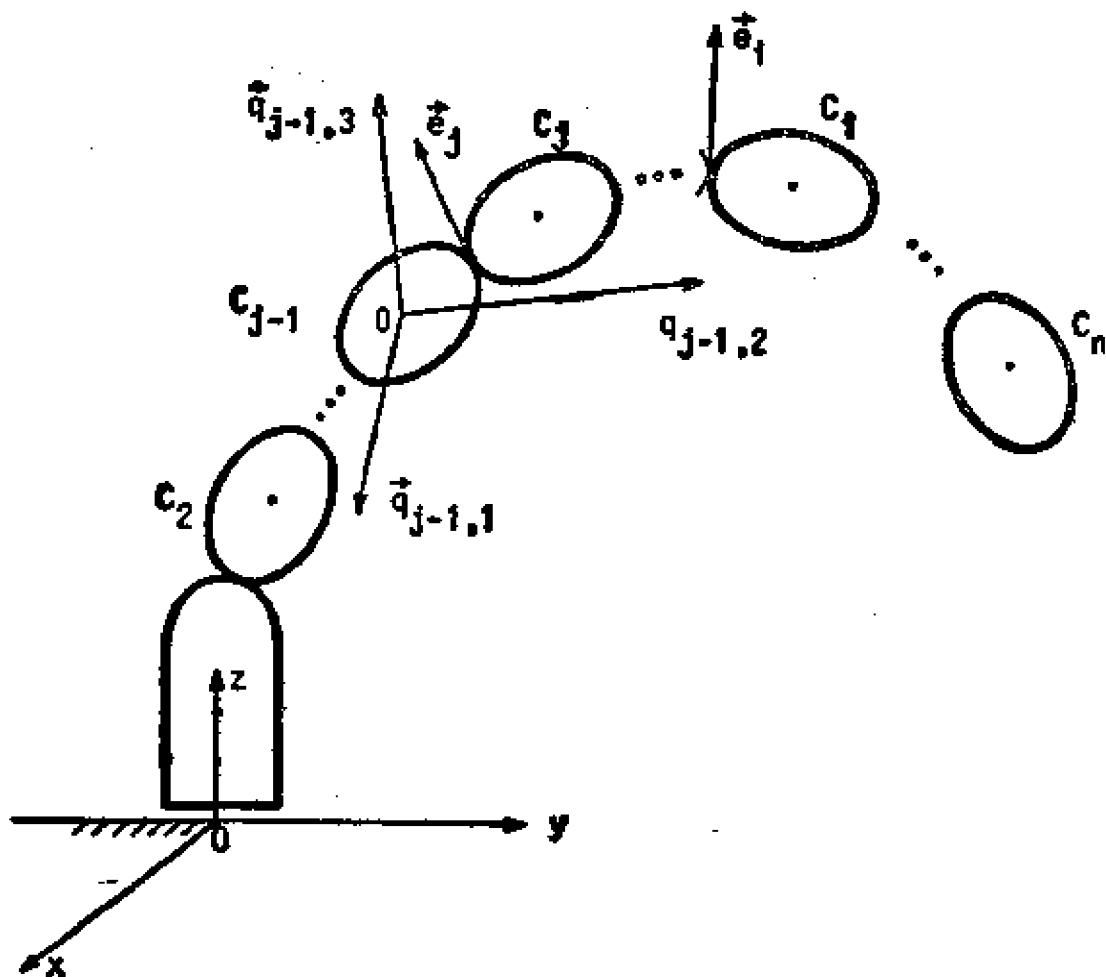


图3.2 确定变量 e_{ij} 的多项式矩阵的向量图

最后,我们考虑变量

$$p_{i\mu} = e_i \cdot q_{i\mu}$$

当 e_i 和 $q_{i\mu}$ 相对 Q_i 和 Q_j 分别为常向量时, 我们可以按 e_{ij} 的情况处理。所以, 对应多项式矩阵的指数属于集合 $\{0, 1\}$ 。

让我们转到图 2.7 总结的动力学模型矩阵的表示式, 例

如,矩阵 $H(q, \theta)$ 的元素 (i, k) 包括乘积 $\langle e, r \rangle_{ji} \cdot \langle e, r \rangle_{ik}$, $j \geq i \geq k$, 两个向量都可用 $\text{Rot}(q_1, \dots, q_i)$ 表示. 由于这两个向量对应的指数属于集合 $\{0, 1\}$, 因此得到, 它们点积的多项式矩阵的指数不超过 2. 对动力学模型矩阵的其它元素也能导出同样的结论. 所以, 任何动力学或运动学变量的指数只能是 0, 1 或 2.

第四章 模型优化与实时程序码的生成

4.1 引言

在前两章中，我们阐述了计算机辅助构造机器人解析模型所需的理论背景。它表明，在解析建模方面，我们遇到了复杂的，能够变换为多变量多项式的三角函数。我们看到，这种多项式的任一多项式都是由指数属于集合 $\{0,1\}$ 的单项式的和组成的。现在，我们从实际观点提出一个非常重要的问题：如何用最少次数的浮点乘法/加法运算来计算这种多项式？本章的第一部分就专门论述这个问题。在本章的第二部分，我们将研究一种计算多项式的优化算法。这种算法包括几个步骤。在这种方法中，我们把全局优化简化为一系列逐次的局部优化。一般说来，尽管这样做不能达到绝对最小，但它大大简化了问题的解。可以证明，这种算法很便于计算机实现。

在 4.2 节中，我们还要引入说明局部优化算法的计算图。这种计算图表示了浮点乘法/加法运算的优化顺序，同时定义了计算图的矩阵表示式。我们从矩阵表示式开始，论述了计算多项式的问题。问题的解法由一个定理给出，该定理表明，计算过程可简化为一系列递推关系式，每一个关系式最多包含一次乘法运算和一次或多次加法运算。我们将会看到，如果把这些关系式变换为程序语句，那么就可用计算机计算多项式，变换问题可由计算机本身有效地解决。完成这种任务的计算机程序称为“专家程序”。专家程序的重要性在于，它

能完全自动地构造机器人的数字模型。

此后,我们对得到的(计算机生成的)程序进行编译,并把它用于实时应用、设计目的、仿真和控制综合。输出文件的程序语言不必与用于专家程序设计的语言相同。例如,专家程序可用高级程序语言编写(像 FORTRAN, PASCAL 等),而输出文件可以是任意微型计算机的汇编语言。

在本章的结尾,我们将以几个例子来说明几个专家程序的功能。

4.2 多项式矩阵的优化计算

考虑多项式矩阵

$$h = (h^{(m)T}, E_k)$$

它对应

$$h(x) = \sum_{k=1}^m h_k x_1^{i_1 k} \cdots x_N^{i_N k} \quad (4.2.1)$$

按照附录 3.1 列出并证明的定理,对于任意机器人操作手,都有

$$e_{ik} \in \{0, 1, 2\}, i \in \{1, \dots, N\}, k \in \{1, \dots, m\}$$

现在,让我们引入变量域 $X = \{x_i: i = 1, \dots, N\}$ 和空间 $Y = \{y_j: j = 1, \dots, 2N\}$ 之间的变换,对 $i = 1, \dots, N$, 使满足

$$y_{2i-1} = x_i, y_{2i} = x_i^2$$

利用这一非线性变换,多项式(4.2.1)变为

$$h(y) = \sum_{k=1}^m h_k y_1^{i_1 k} \cdots y_M^{i_M k} \quad (4.2.2)$$

式中 $M = 2N$ 。将上式与式(4.2.1)比较,对 $i = 1, \dots, N$ 和

$k = 1, \dots, m$, 我们得到 σ

$$\sigma_{2i-1,k} = \begin{cases} 0, & \varepsilon_{ik} = 0, 2 \\ 1, & \varepsilon_{ik} = 1 \end{cases} \quad (4.2.3)$$

$$\sigma_{2i,k} = \begin{cases} 0, & \varepsilon_{ik} = 0, 1 \\ 1, & \varepsilon_{ik} = 2 \end{cases}$$

我们注意到, 对 $i = 1, \dots, M$ 和 $k = 1, \dots, m$

$$\sigma_{ik} \in \{0, 1\} \quad (4.2.4)$$

很明显, 多项式(4.2.2)的指数矩阵

$$E\% = \begin{bmatrix} \sigma_{11} \cdots \sigma_{M1} \\ \vdots \quad \quad \quad \vdots \\ \sigma_{1m} \cdots \sigma_{Mm} \end{bmatrix} \quad (4.2.5)$$

是一个二进制矩阵, 也就是只有元素 0 或 1 的矩阵。从实际观点看, 这是一个非常重要的特性, 因为仅用一位存储单元就能存储矩阵式(4.2.5)中的任一元素。因此, 整个矩阵式(4.2.5)存储在

$$m(r + M) = m(r + 6n)$$

位存储器内, 式中 r 表示实数所占用的位数, n 表示操作手的自由度数。

现在, 让我们考虑计算多项式 $h(x)$ 的运算速度。从式(4.2.1)可以看出, 浮点乘法的运算次数等于

$$n_M = \sum_{i=1}^N \sum_{k=1}^m \varepsilon_{ik} \quad (4.2.6)$$

而浮点加法的运算次数等于

$$n_A = m$$

由于 $\varepsilon_{ik} \in \{0, 1, 2\}$, 所以满足

$$0 \leq n_M \leq 2mN = 6mn \quad (4.2.7)$$

因此, 我们看到, 在 $\varepsilon_{ik} = 2, i = 1, \dots, N, k = 1, \dots, m$ 的最

$$I_{ik} = \begin{cases} 1, \beta_i \mathcal{L} s_k \\ 0, \beta_i \overline{\mathcal{L}} s_k \end{cases} \quad (4.2.13)$$

式中 $\overline{\mathcal{L}}$ 表示 \mathcal{L} 的逆关系。

让我们回到多项式(4.2.1)的优化计算上来。对任意一个 $\beta_i, i \in \{1, \dots, 3^N - 1\}$, 让我们按照以上的定义, 来确定 L_i 和 $dg\beta_i$ 。然后选择使乘积 $(dg\beta_i)L_i$ 为最大的那个 β_i , 也就是

$$L_i dg\beta_i = \max_j (L_j dg\beta_j) \quad (4.2.14)$$

式中

$$L_i = \begin{cases} L_i - 1, L_i \geq 1 \\ 0, L_i = 0 \end{cases}$$

现在,我们必须抽取满足条件 $\beta_i \mathcal{L} s_k$ 的那些行 s_k 。我们把对应脚标 k 的集合表示为

$$\kappa_i = \{k'_1, \dots, k'_{L_i}\} \quad (4.2.15)$$

现在,我们把多项式 $h(x)$ (4.2.1) 分为如下两部分:

$$h(x) = h_{11}(x) + h_{12}(x) \quad (4.2.16)$$

式中

$$h_{11}(x) = \sum_{k \in \kappa_i} h_k x_1^{i_1 k} \dots x_N^{i_N k}$$

$$h_{12}(x) = \sum_{k \in \kappa_i^c} h_k x_1^{i_1 k} \dots x_N^{i_N k} \quad (4.2.17)$$

其中 κ_i^c 是 κ_i 对 $\{1, \dots, m\}$ 的补集。多项式 $h_{11}(x)$ 可表示为

$$h_{11}(x) = M_{11}(x) \sum_{k \in \kappa_i} h_k x_1^{i_1 k - \theta_{11}} \dots x_N^{i_N k - \theta_{N1}} \quad (4.2.18)$$

式中

$$M_{11}(x) = x_1^{\theta_{11}} \dots x_N^{\theta_{N1}} \quad (4.2.19)$$

这后面的几个表示式是在假设 $\beta_j \leq \varepsilon_k, k \in \kappa_j$ 条件下得到的结果。值得注意的是,在式(4.2.18)右边求和式中的各个单项式的指数都是正的,即

$$\varepsilon_{jk} - \beta_{jk} \geq 0, j \in \{1, \dots, N\}, k \in \kappa_j$$

我们看到, $M_{11}(x)$ 中的乘法运算次数等于

$$\sum_{i=1}^N \beta_{1i} = \text{dg}\beta_1$$

因为单项式 $M_{11}(x)$ 表示和式(4.2.18)的加数 L_i 的通项,所以得到,乘法运算次数减少了

$$(L_i - 1)\text{dg}\beta_i = L'_i \text{dg}\beta_i$$

由式(4.2.14)看到,这个乘积最大。这意味着,这个算法在使浮点乘法运算次数最小意义上是最优的。因此,乘法运算次数 n_M 等于

$$n_M = n_{M(\max)} - L'_i \text{dg}\beta_i \quad (4.2.20)$$

式中 $n_{M(\max)}$ 由式(4.2.6)给出, $L'_i = L_i, \text{dg}\beta_i = \text{dg}\beta_i$ 。

优化计算的第二步是从多项式

$$h_{11}(x) = \sum_{k \in \kappa_1} h_k x_1^{\varepsilon_{1k}} \cdots x_M^{\varepsilon_{Mk}} \quad (4.2.21)$$

$$h_{12}(x) = \sum_{k \in \kappa_2} h_k x_1^{\varepsilon_{1k}} \cdots x_N^{\varepsilon_{Nk}}$$

着手做,式中

$$\varepsilon'_{jk} = \varepsilon_{jk} - \beta_{jk}, j = 1, \dots, M$$

为求解这两个多项式,我们需要按照式(4.2.10)——(4.2.15)求出 $\beta_i, \text{dg}\beta_i, L_i$ 和 κ_i 的值。于是得到

$$\beta_{11}, \text{dg}\beta_{11}, L_{11}, \kappa_{11}, \text{对 } h_{11}(x)$$

$$\beta_{12}, \text{dg}\beta_{12}, L_{12}, \kappa_{12}, \text{对 } h_{12}(x)$$

正像在式(4.2.16)和(4.2.17)中指出的那样,我们可按集合 κ_{11}

和 κ_{12} 描述的方法,把多项式分解为子多项式,于是得到

$$h'_{11}(x) = h_{111}(x) + h_{112}(x) \quad (4.2.22)$$

$$h'_{12}(x) = h_{121}(x) + h_{122}(x)$$

式中两个子多项式 $h_{111}(x)$ 和 $h_{121}(x)$ 都可表示为式(4.2.18)的形式:

$$h_{111}(x) = M_{111}(x)h'_{111}(x) \quad (4.2.23)$$

$$h_{121}(x) = M_{121}(x)h'_{121}(x)$$

式中 M_{111} 和 M_{121} 是类似于式(4.2.19)的单项式,而 h'_{111} 和 h'_{121} 是相应的和式。

我们用字母分解来结束第二步, n_M 简化为

$$n_M = n_{M(\max)} - L'_1 dg\beta_{11} - L'_{11} dg\beta_{11} - L'_{12} dg\beta_{12} \quad (4.2.24)$$

而多项式 $h(x)$ 简化为

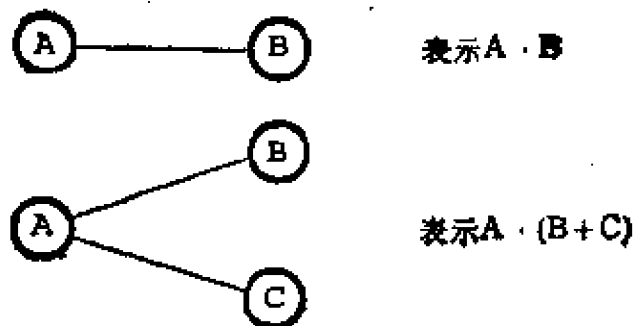
$$h(x) = M_{11}(M_{111}h'_{111} + h_{112}) + M_{121}h'_{121} + h_{122} \quad (4.2.25)$$

第三步包括多项式 h'_{111} , h_{112} , h'_{121} 和 h_{122} 的分解。重复这个过程,直到多项式简化为常数。因此, $h(x)$ 的形式取决于

1) 单项式 ($M(x) = x_1^{\beta_1} \cdots x_m^{\beta_m}$);

2) 常数 $h_k, k = 1, \cdots, m$ 。

这种形式可以象征性地用无向计算图 Γ 表示,如图 4.1 所示。单项式和常数赋予图的结点,每一浮点乘法运算对应于支路。如果两个结点用图的支路联接,那么相应的单项式相乘;如果很多支路连到一个结点,那么结点单项式与相应支路多项式的和相乘。因此,两个基本计算图的元素表示为



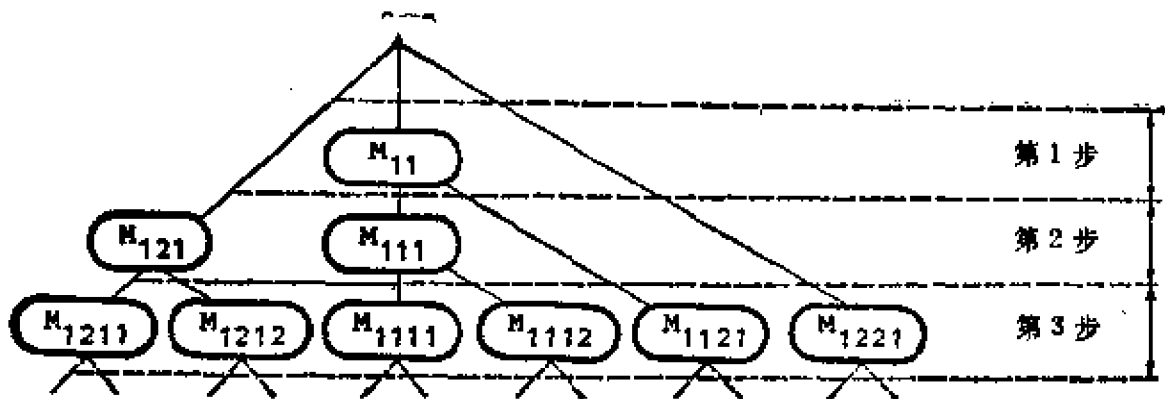
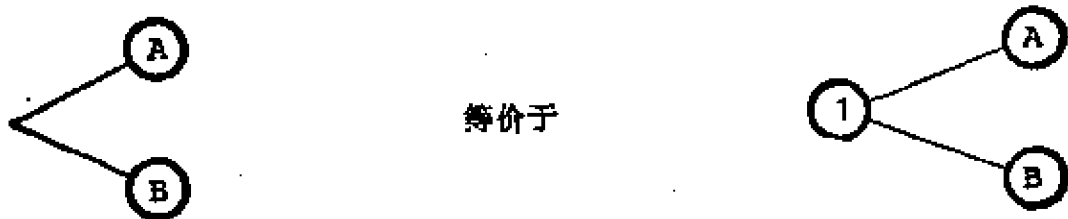


图 4.1 多项式 $h(x)$ 的计算图表示

同样, 如果一个结点上既没有单项式, 又没有常数, 那么这个结点就被认为是 1。例如下图对应



$$1(A + B) = A + B.$$

为了用计算机实现上述算法, 引入计算图的某种矩阵表示是方便的。它允许以数字据方式来处理计算图。首先, 我们注意到, 可把结点单项式看做是子计算图。例如, 单项式 $M(x) = h_1 x_1 x_3$ 可以表示为这样的子图



这种图称为基本计算图。对于这样任意一个基本计算图 Γ , 我们都用两种矩阵 $E = [e_{ij}]$ 和 $V = [v_{ij}]$ 联接起来, 其中 e_{ij} 表示赋予结点的变量脚标, v_{ij} 表示这个结点的输出支路数。用 x_i 表示变量和常数, 这里

$$X_i = \begin{cases} x_i, & 1 \leq i \leq N \\ h_{i-N}, & N < i \leq N + m \end{cases} \quad (4.2.26)$$

任何赋予结点的变量都用脚标唯一确定。为了建立计算图的矩阵 E 和 V ，我们用一个简单的例子来说明。

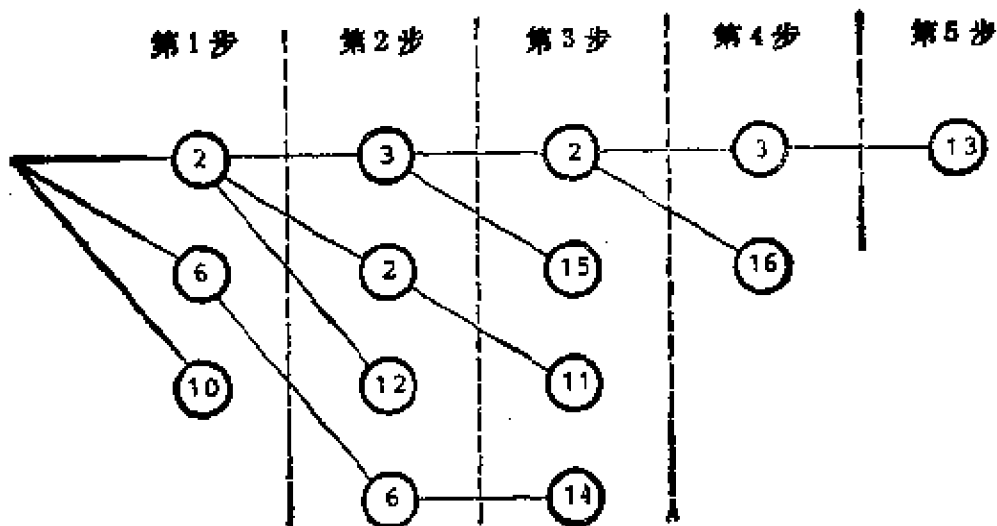


图 4.2 基本计算图

考虑图 4.2 所示计算图。结点中的数据是变量 X_i 的脚标，这里，我们有 $N=9$ 和 $m=7$ 。结点可以变换为 E 矩阵

$$E = \begin{bmatrix} 2 & 3 & 2 & 3 & 13 \\ 6 & 2 & 15 & 16 & 0 \\ 10 & 12 & 11 & 0 & 0 \\ 0 & 6 & 14 & 0 & 0 \end{bmatrix}$$

其中列数对应步数，行数对应“并行”结点数。 V 矩阵的行和列具有同样意义。 V 矩阵的元素表示对应结点的输出支路数，于是得到

$$V = \begin{bmatrix} 3 & 2 & 2 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

这一简单的例子说明了如何建立计算图 Γ 的 E 和 V 矩阵。 E 和 V 唯一地定义了计算多项式的最优运算序列。

最后,我们得出下列结论:

- 1) 优化算法简单,便于计算机实现;
- 2) 优化计算是一步一步地实现的,因此,得到的乘法运算次数并非绝对最小,但接近最小;
- 3) 在某些优化步骤中,因为对很多脚标 i , $\max(L; dg\beta_i)$ 都是相同的,所以会出现多个最优值。我们可对这些序列进行优化,最后找出绝对最优值。然而,这样做会使优化过程更加复杂。

例

考虑多项式矩阵

$$h(x) = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 2 & 0 & 2 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 2 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

按照式(4.2.6),最大的 n_M 等于

$$n_{M(\max)} = 18$$

现在让我们应用优化算法来说明各步的运算。

第一步

按照表示式(4.2.9)—(4.2.13),得到下表:

i	序列 $\beta_i(4.2.9)$	$dg\beta_i$ (4.2.10)	L_i (4.2.12) — (4.2.13)	$L_i dg\beta_i$
1	0 0 0 0 0 0 0 1	1	4	3
3	0 0 0 0 0 0 0 1 0	1	2	1
81	0 0 0 0 1 0 0 0 0	1	2	1
729	0 0 1 0 0 0 0 0 0	1	4	3
730	0 0 1 0 0 0 0 0 1	2	4	6

在表中，我们只给出了那些满足 $L_i \geq 1$ 的序列 β_i ，其余， $L_i dg\beta_i = 0$ 。在 $\beta_i = [001\ 000\ 001]$ ， $i = 730$ 时，乘积 $L_i dg\beta_i$ 最大。在 $k = 1, 2, 3$ 和 5 时，满足条件 $\beta_i \in \varepsilon_k$ ，也就是

$$\kappa_i = \{1, 2, 3, 5\}$$

现在多项式 $h(x)$ 变为

$$h(x) = M_{11}(x)h'_{11}(x) + h_{12}(x)$$

按照式(4.2.19)，有

$$M_{11}(x) = x_3 x_5$$

按照式(4.2.17)，有

$$h_{12}(x) = h_4 x_5 x_8$$

按照式(4.2.18)， $h'_{11}(x)$ 变为

$$h'_{11}(x) = \begin{pmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_5 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{pmatrix}$$

乘法运算次数减少到

$$n_M = n_{M(\max)} = L_i dg\beta_i = 12$$

第二步

现在，我们来优化计算 $h'_{11}(x)$ 。多项式 $h_{12}(x)$ 在上一步中已简化为最优形式，

在 $h_{11}(x)$ 中, 只有唯一的行 β_l 满足 $L'_i \geq 1$, 即

$$\beta_l = [001\ 000\ 000]$$

式中 $l = 729$, $\text{dg}\beta_l = 1$ 且 $L'_i = 1$. 由于 $\kappa_l = \{3, 4\}$, $h_{11}(x)$ 可以表示为

$$h_{11}(x) = M_{111}(x)h'_{111}(x) + h_{112}(x)$$

式中

$$M'_{111}(x) = x_3$$

$$h'_{111}(x) = \left(\begin{bmatrix} h_3 \\ h_5 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 & 2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \right)$$

$$h_{112}(x) = \left(\begin{bmatrix} h_1 \\ h_2 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \right)$$

很明显, 这两个多项式矩阵不能再进一步优化了, 它们的解析形式是

$$h'_{111}(x) = h_3 x_3^2 x_6 + h_5 x_4$$

$$h_{112}(x) = h_1 x_1 + h_2 x_8$$

由于 $L'_i \text{dg}\beta_l = 1$, 所以 n_M 减少 1 次, 即

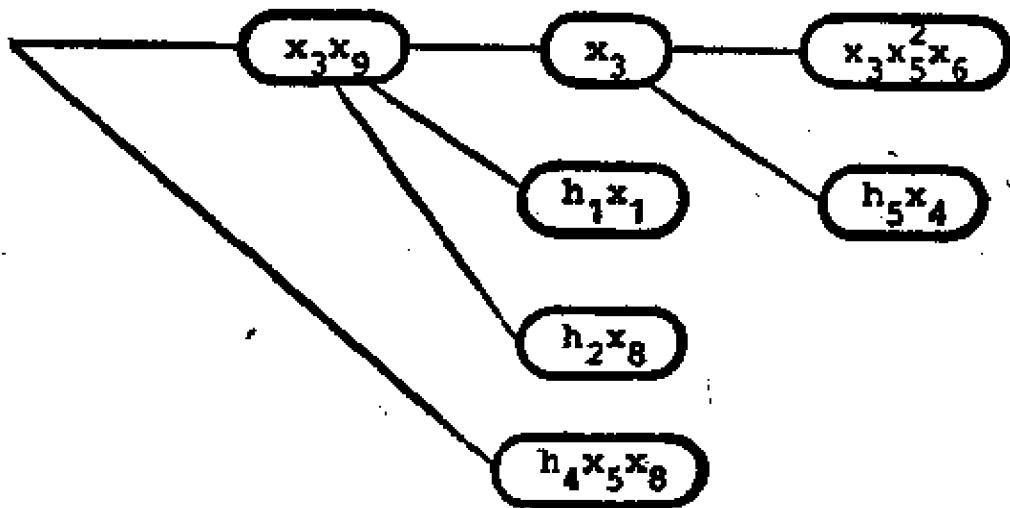


图 4.3 $h(x)$ 的计算图

$$n_M = 11$$

优化到此结束,多项式 $h(x)$ 最后得到的形式是

$$h(x) = x_3 x_9 (x_3 (h_3 x_5^2 x_6 + h_5 x_4) + h_1 x_1 + h_2 x_8) + h_4 x_5 x_8$$

图 4.3 给出了计算图。我们从图中看到, n_M 确实等于 11。
 n_M 从 18 下降到 11,也就是下降了 40%。

现在让我们来计算图 4.3 中的矩阵表示式。首先,我们建立等价于图 4.3 的基本计算图(图 4.4)。然后,直接写出 E 和 V 矩阵

$$E = \begin{bmatrix} 3 & 9 & 3 & 5 & 5 & 6 & 12 \\ 5 & 8 & 1 & 4 & 14 & 0 & 0 \\ 0 & 0 & 8 & 10 & 0 & 0 & 0 \\ 0 & 0 & 13 & 11 & 0 & 0 & 0 \end{bmatrix}$$

$$V = \begin{bmatrix} 1 & 3 & 2 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

请注意,基本计算图的任何自由(终端)结点都赋予一个常数。

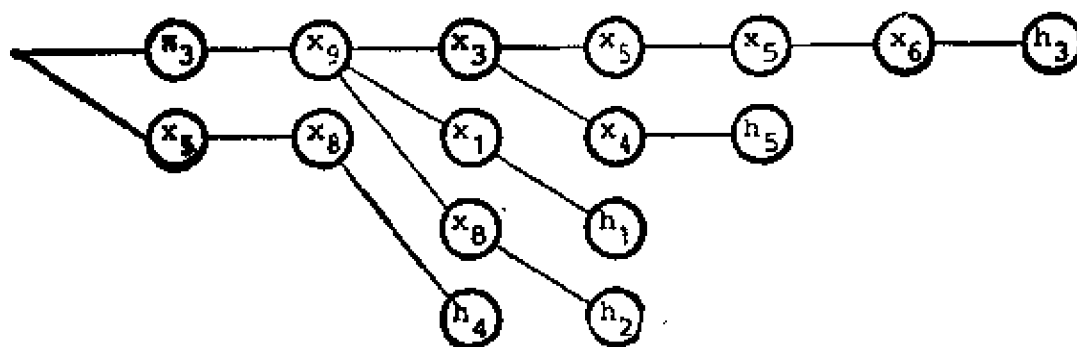


图 4.4 $h(x)$ 的基本计算图

4.3 实时程序码的生成

在前一节中，我们分析了如何生成多项式 $h(x)$ 的基本计算图和对应的 E 和 V 矩阵的问题。利用这两个矩阵，就能够构成一个简单的多项式计算算法。

定理

设 E 和 V 是多项式 $h(x)$ 基本计算图 Γ 的 $n_r \times n_c$ 矩阵。我们用 $x_{i,j}$ 表示对应矩阵 E 第 (i,j) 个元素的变量/常数。我们还引入两个辅助向量。

多项式 $h(x)$ 按下列算法计算：

第一步

$$q(i) = X_{i,n_c}, i \in N_{or} = \{1, \dots, n_r\} \quad (4.3.1)$$

第二步

$$p(i) = X_{i,n_c-1}(q(k_{i-1} + 1) + \dots + q(k_i)) \\ \forall i \in N, \quad (4.3.2)$$

式中

$$k_i = \sum_{j=1}^i V(j, n_c - 1), k_0 = 0$$

⋮

第 $(2l)$ 步

$$p(i) = X_{i,n_c-2l+1}(q(k_{i-1} + 1) + \dots + q(k_i)), \forall i \in N, \quad (4.3.3)$$

式中

$$k_i = \sum_{j=1}^i V(j, n_c - 2l + 1), k_0 = 0$$

第 $(2l + 1)$ 步

$$q(i) = X_{i, n_c - 2i} (p(k_{i-1} + 1) + \dots + p(k_i)), \forall i \in N, \quad (4.3.4)$$

式中

$$k_i = \sum_{j=1}^i V(j, n_c - 2i), \quad k_0 = 0$$

⋮

第 $n_c + 1$ 步

$$n_c + 1 = \begin{cases} \text{偶数时, } h(x) = \sum_{i=1}^{n_c} p(i) \\ \text{奇数时, } h(x) = \sum_{i=1}^{n_c} q(i) \end{cases} \quad (4.3.5)$$

这一定理的证明在附录 4.1 中给出。

应当指出，上述的算法提供了计算多项式 $h(x)$ 的计算机程序的基础。按照递推表示式(4.3.1)–(4.3.5)生成的程序称为“专家程序”。专家程序的输出同样是一种高级或低级语言编写的计算机程序。

由式(4.3.1)–(4.3.5)，我们确定了由专家程序生成的基本表示式

$$p_i = x_j (q_{i1} + \dots + q_{in})$$

式中脚标 i, j 和 n 均为给定。例如，对 $i = 1, j = 2$ 和 $n = 3$ ，我们可用 FORTRAN 程序，带有浮点处理器 8087 的微处理机 INTEL8086 的汇编程序，以及基于微处理机 INTEL8085 的微型计算机的汇编程序等来描述这个方程。

1) FORTRAN

$$P1 = X2 * (Q11 + Q12 + Q13)$$

2) 8086/8087汇编


```

FLD Q11; Q11 → ST
FADD Q12; ST + Q12 → ST
FADD Q13; ST + Q13 → ST
FMUL X2; ST * X2 → ST
FST P1; ST → P1

```

其中“ST”表示堆栈存储器的第一个单元，“→”表示数据传送。所有语句都适用浮点数。

3) 8085 汇编

```

LXI H, Q11; Q11 → HL
LXI D, Q12; Q12 → DE
CALL FPADD; Q11 + Q12 → HL, exp → A
SHLD TEMP; HL → TEMP
STA TEMP + 2; A → TEMP + 2
LXI H, TEMP; TEMP → HL
LXI D, Q13; Q13 → DE
CALL FPADD; Q11 + Q12 + Q13 → HL, exp → A
SHLD TEMP; HL → TEMP
STA TEMP + 2; A → TEMP + 2
LXI D, X2; X2 → DE
CALL FPMUL; X2 * (Q11 + Q12 + Q13) → HL,
           exp → A
SHLD P1; HL → P1
STA P1 + 2; A → exp → P1

```

其中

HL, DE, BC 表示16位寄存器;

A表示8位累加器;

Q11, Q12, Q13 和 X2 表示浮点数据的存储器地址;前16

位表示尾数,后 8 位表示阶码;

FPADD 和 FPMUL 表示浮点加法和乘法的子程序。

这些例子说明,低级的程序设计语言需要高级复杂的专家程序,而生成高级语言程序却非常简单。

例

考虑图 4.4 所示基本计算图和前一节得到的相应 B 和 V 矩阵,用提出的定理生成如下算法:

$q_1 = x_{12}$
$p_1 = x_6 q_1$
$q_1 = x_5 p_1$
$q_2 = x_{14}$
$p_1 = x_5 q_1$
$p_2 = x_4 q_2$
$p_3 = x_{10}$
$p_4 = x_{11}$

续

$q_1 = x_3(p_1 + p_2)$
$q_2 = x_1 p_2$
$q_3 = x_8 p_3$
$q_4 = x_{13}$
$p_1 = x_9(q_1 + q_2 + q_3)$
$p_2 = x_8 q_4$
$q_1 = x_3 p_1$
$q_2 = x_5 p_2$
$H = q_1 + q_2$

算法的 FORTRAN 码形式:

$$P1 = X6 * X12$$

$$Q1 = X5 * P1$$

$$P1 = X5 * Q1$$

$$P2 = X4 * X14$$

$Q1 = X3 * (P1 + P2)$
 $Q2 = X1 * P2$
 $Q3 = X8 * X10$
 $P1 = X9 * (Q1 + Q2 + Q3)$
 $P2 = X8 * X13$
 $Q1 = X3 * P1$
 $Q2 = X5 * P2$
 $H = Q1 + Q2$

在这里,为了尽量减小计算时间,省略了 $q = X$ 这类语句。

用 8086/8087 汇编语言,本程序变为

```

FLD X6; X6 → ST
FMUL X12; ST * X12 → ST
FST P1; ST → P1
FLD X5; X5 → ST
FMUL P1; ST * P1 → ST
FST Q1; ST → Q1
FLD X5; X5 → ST
FMUL Q1; ST * Q1 → ST
FST P1; ST → P1
FLD X4; X4 → ST
FMUL X14; ST * X14 → ST
FST P2; ST → P2
FLD P1; P1 → ST
FADD P2; ST + P2 → ST
FMUL X3; X3 * ST → ST
FST Q1; ST → Q1
FLD X1; X1 → ST

```

FMUL P2; ST * P2 → ST
 FST Q2; ST → Q2
 FLD X8; X8 → ST
 FMUL X10; ST * X10 → ST
 FST Q3; ST → Q3
 FLD Q1; Q1 → ST
 FADD Q2; ST + Q2 → ST
 FADD Q3; ST + Q3 → ST
 FMUL X9; ST * X9 → ST
 FST P1; ST → P1
 FLD X8; X8 → ST
 FMUL X13; ST * X13 → ST
 FST P2; ST → P2
 FLD X3; X3 → ST
 FMUL P1; ST * P1 → ST
 FST Q1; ST → Q1
 FLD X5; X5 → ST
 FMUL P2; ST * P2 → ST
 FST Q2; ST → Q2
 FLD Q1; Q1 → ST
 FADD Q2; ST + Q2 → ST
 FST H; ST → H

最后,我们必须指出,专家程序可以用任何语言编写。

附 录 4.1

在这里,我们将要证明 4.3 节中叙述的定理。为了证明

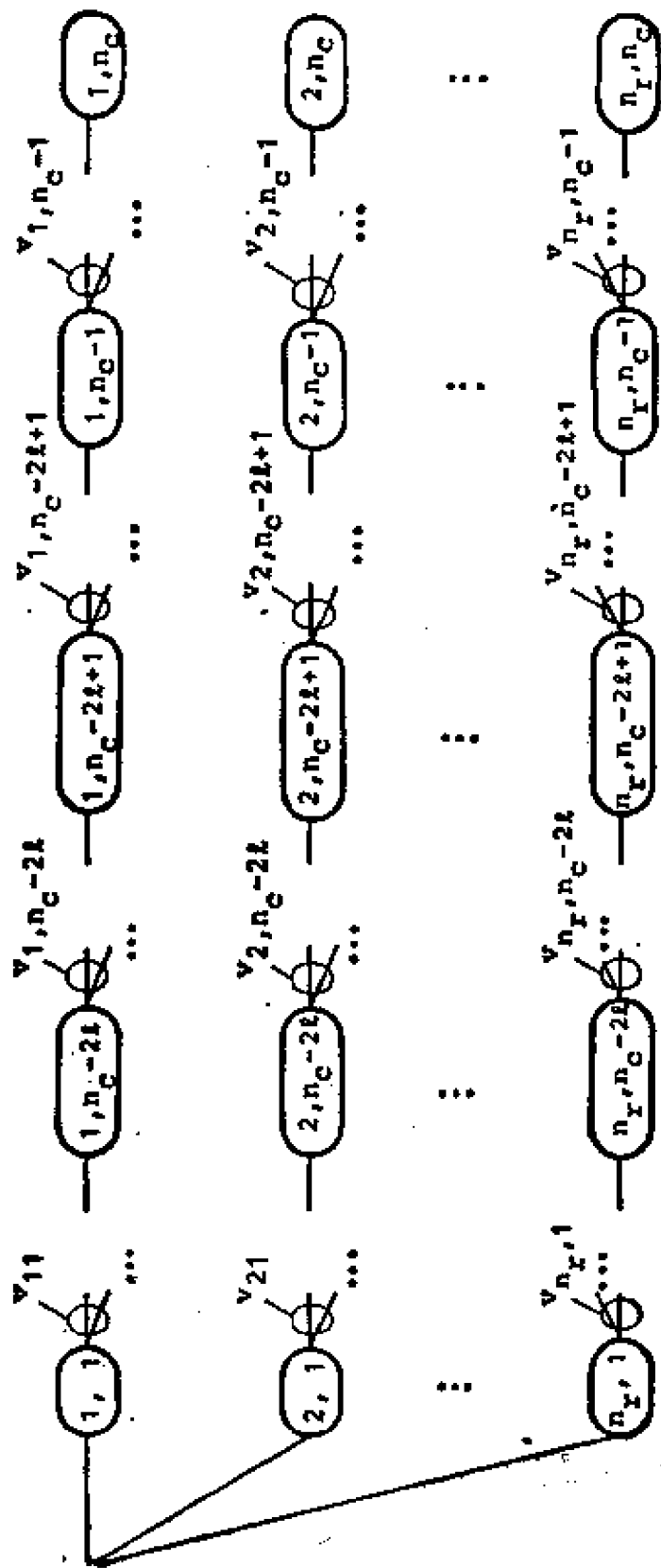


图 4.5 基本计算图的一般形式

这一定理，先考虑如图 4.5 所示的一般基本计算图 Γ ，与 B 矩阵脚标对应的脚标赋予结点。因此，结点 (i, j) 相当于元素 e_{ij} ，也就是变量 X_{ij} 。图上还表示了结点的输出支路数 v_{ij} 。

把与结点 (i, n_c) 对应的变量表示为

$$q(i) = X_{i, n_c}, \quad i \in N,$$

考虑结点 $(1, n_c - 1)$ ，从这一结点出发有 $v_{1, n_c - 1}$ 个支路，这些支路到终结点 $(1, n_c), \dots, (v_{1, n_c - 1}, n_c)$ 结束。我们用一个结点替换这样一个子图，使其包括结点 $(1, n_c - 1)$ 、这个结点的所有输出支路及其终结点。

很明显，多项式

$$p(1) = X_{1, n_c - 1}(q(1) + \dots + q(v_{1, n_c - 1})) \quad (\text{P4.1.1})$$

必须赋给这个结点。如果我们同样作用于结点 $(i, n_c - 1)$ ， $i = 2, \dots, n_r$ ，那么，就正好得到多项式(4.3.2)。因此，我们得到的图等价于含有结点 $p(1), \dots, p(n_r)$ 的基本计算图 Γ 。

我们可用类似方法证明定理中其它的表示式。在第 $2l - 1$ 步，假设结点 $(i, n_c - 2l + 1)$ ， $i \in N$ ，和结点的输出支路用等价结点 $q(1), \dots, q(n_r)$ 替换。现在考虑结点 $(1, n_c - 2l + 1)$ 。正如式(P4.1.1)的推导那样，我们得到一个由该结点，该结点向右的输出支路，以及以 $q(1), \dots, q(v_{1, n_c - 2l + 1})$ 作为支路端点的结点等组成的子图，这个子图可用赋予多项式

$$p(1) = X_{1, n_c - 2l + 1}(q(1) + \dots + q(v_{1, n_c - 2l + 1})) \quad (\text{P4.1.2})$$

的结点替换。同样的过程作用于结点 $(i, n_c - 2l + 1)$ ，我们就得到定理的第 $2l$ 步的多项式(4.3.3)。

表示式(4.3.4)和(4.3.5)可以用同样的方法证明。

第五章 实 例

5.1 引 言

在这一章中，我们以工业工艺过程中应用的真实机器人作为实例来验证前几章中介绍的所有结论。我们考虑的实例包括轻型机器人（UMS-1 型和 UMS-2 型，南斯拉夫产品）和可靠的福德斯（Dependable-Fordath）重型机器人（美国产品），选择这些机器人是因为它们有不同的机械结构。UMS-2 型机器人具有圆柱型结构，UMS-1 型机器人具有拟人结构（类似于人的手臂），而重型机器人具有节肢结构。很明显，选择不同结构的机构是为了说明数学模型的复杂程度依赖于结构的复杂程度。

第一个实例是圆柱型结构机器人（圆柱型机器人），它有一个旋转自由度和两个平移自由度。第二个实例考虑节肢机器人，它有三个旋转自由度，第二个和第三个关节的轴是平行的。第三个实例讨论拟人机器人，它有三个自由度，但没有平行轴。每个实例都给出了所需几何和动力学参数的详尽运动学简图。然后对前几章中描述的，构造非线性数值-符号模型的方法加以说明。它包括确定机构中各连杆的基始位置和实际位置，确定惯性矩阵， C -矩阵，以及确定重力向量等所有步骤的输出结果。这些步骤的输出都用相应的多项式矩阵表示，从中可直接导出多项式，即解析函数式。这样建立的动力学模型矩阵可用来说明，动力学模型矩阵的所有特性，如对称

性、正定性和反对称性(第二章)都是成立的。

我们还利用各个实例来说明 2.5 节和 3.6 节中介绍的,建立线性化数值-符号模型的方法,其中用到了 3.6 节中证明过的多项式矩阵的偏导数定理。结果表明,这个定理对程序实现非常简单和实用。

借助各个实例,我们还要说明构造机器人灵敏度模型的方法。我们将考虑 2.6 节和 3.6 节中描述的,动力学参数(质量,惯性矩)的变化对驱动转矩或驱动力的影响。可以证明,灵敏度模型所需数值计算量大大少于操作手完整动力学模型所需的数值计算量。

通过节肢机器人的实例,我们还要介绍建立机器人近似模型的方法。在 2.6 节和 3.7 节中从理论上论述了这种方法。在本例中可以看到有可能忽略掉那些不显著的动力学效应。例如,相对重力效应而言,哥氏效应常常可以忽略不计。在本例中还给出了最大误差估计,也就是在给定允许误差条件下,要确定出哪些效应可以忽略不计。

在每个实例的末尾,我们都说明了数值-符号模型优化计算的方法。对第四章引入的所有概念,例如优化计算序列、基本计算图和计算图的矩阵表示等等也都做了说明。我们特别考虑了如何利用专家程序生成模型程序码的问题。由程序码可确定出所需浮点乘法和加法运算的确切次数,这对微型计算机实现来说尤为重要。我们还通过一个实例比较了建立线性化模型和非线性模型所需的运算次数。这样,前述所有理论结果,即对应程序包的所有模块都得到了阐明。

5.2 圆柱型机器人

让我们考虑圆柱型结构操作手,如图 5.1 所示。这个机

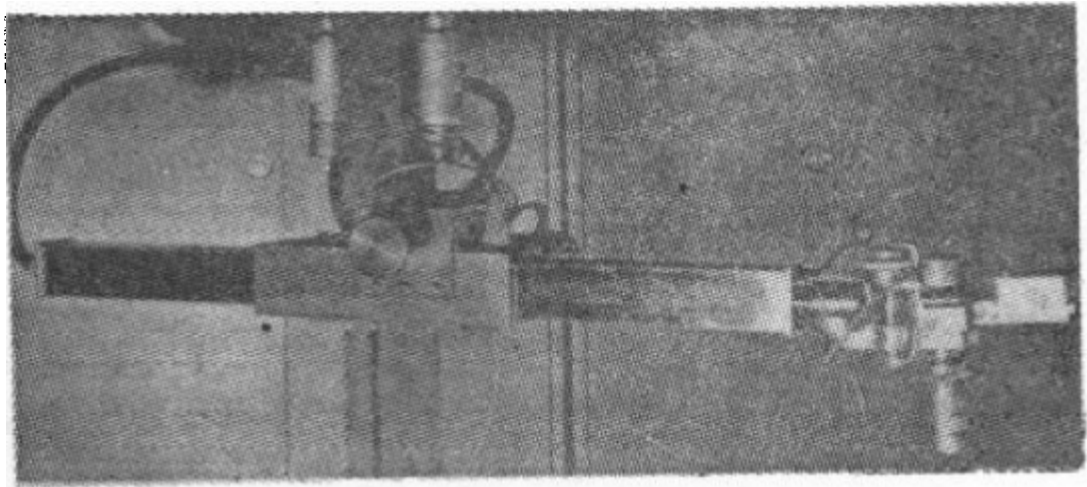


图 5.1 圆柱型机器人

机器人有三个基本自由度，用永磁电动机驱动。具有一到三个自由度的抓手和气动真空吸盘等等各种器件均可联接在第三个连杆的末端。为了系统地展示前述算法，我们先对这个操作手机构的结构建立一个对应的模型(图 5.2)。 C_1, C_2 和 C_3 表示机构的连杆， Z_1, Z_2 和 Z_3 表示关节联接。第一关节是

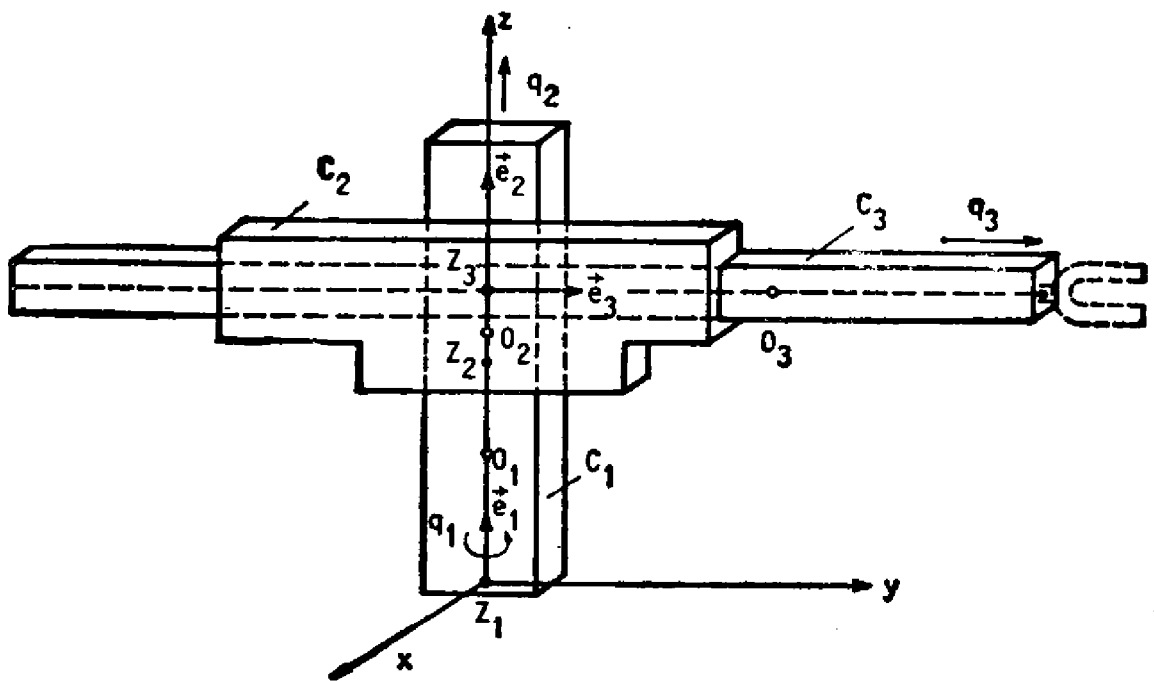


图 5.2 圆柱型机器人机构模型

旋转的,后两个关节是平移的。因此,这个机构具有 RTT 结构。机构的连杆及其全部参数如图 5.3 所示。正如 2.2 节所述,每一连杆都可用排好的参数集合 $C_i(K_i, D_i)$ 描述,其中 $K_i = (Q_i, \tilde{R}_i, \tilde{E}_i)$ 是运动学参数集合, $D_i = (m_i, J_i)$ 是动力学参数集合。在图 5.3 中对每个连杆均设定了内坐标系 $Q_i = (q_{i1}, q_{i2}, q_{i3})$ 。在这个坐标系中定义了距离向量集合 $\tilde{R}_i = \{\tilde{r}_{ii}, \tilde{r}_{i,i+1}\}$ 和关节轴向量 $\tilde{E}_i = \{\tilde{e}_i\}$ 。实际操作手的数

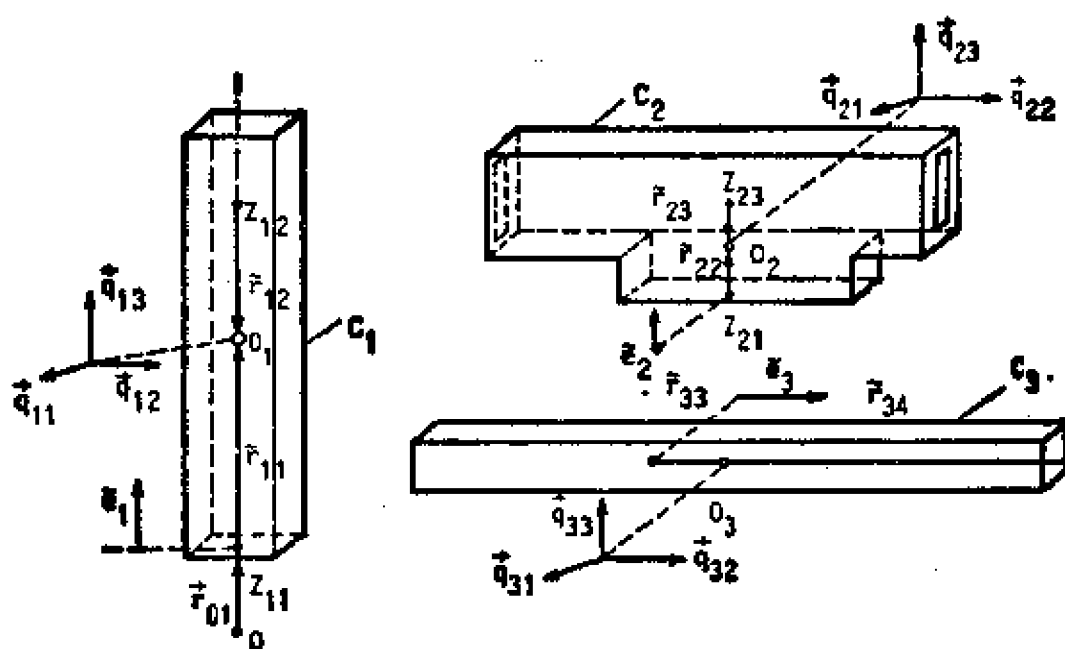


图 5.3 圆柱型机器人机构中连杆的运动学参数

据由表 5.1 给出, ξ_i 表示第 i 个关节的类型, $\xi_i = 0$ 表示旋转关节, $\xi_i = 1$ 表示平移关节。应当注意,所有关节联接: $z_1 = z_{11}$, $z_2 = z_{12} = z_{21}$ 和 $z_3 = z_{23} = z_{32}$, 也就是,所有运动副 $\{C_0, C_1\}$, $\{C_1, C_2\}$ 和 $\{C_2, C_3\}$ 均为第 5 级, 运动副 $\{C_0, C_1\}$ 为第 1 子级, 其中 C_0 表示基座, 其余两个运动副为第 2 子级。

有算法构造机器人闭式动力学模型矩阵(图 2.7 流程图), 必须规定另外两个输入向量 r_{01} 和 e_1

$$r_{01} = (0., 0., 0.)$$

$$e_1 = (0., 0., 1.)$$

表 5.1 圆柱型机器人的运动学参数

i	$r_{ii}[m]$	$r_{i,i+1}[m]$	\tilde{e}_i	$\tilde{\xi}_i$
1	(0., 0., 0.188)	(0., 0., -0.192)	(0., 0., 1.)	0
2	(0., 0., 0.01)	(0., 0., -0.01)	(0., 0., 1.)	1
3	(0., 0.01, 0.)	(0., -0.44, 0.)	(0., 1., 0.)	1

表 5.2 圆柱型机器人的动力学参数

i	$m_i[\text{kg}]$	$J_{ii}[\text{kg m}^2]$	$J_{ii}[\text{kg m}^2]$	$J_{is}[\text{kg m}^2]$
1	—	—	—	0.0294
2	7.	—	—	0.0550
3	4.15	—	—	0.3180

圆柱型机器人非线性解析模型的推导

算法的第一步是计算机构的基始位置。由上列数据可得下列初值

$$Q_i^0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$r_{ii}^0 = \tilde{r}_{ii}, \quad r_{i,i+1}^0 = \tilde{r}_{i,i+1}, \quad i = 1, 2, 3$$

$$e_{i+1}^0 = \tilde{e}_{i+1}, \quad i = 1, 2$$

在算法的第二步, 我们确定机构连杆的位置, 它们是关节坐标 q_1, q_2 和 q_3 的函数。因而, 这一步的输出变量并非具

体数值,而是可以用多项式矩阵,即用多项式描述的函数. 3.5节中介绍了建立期望多项式矩阵的方法. 期望多项式矩阵即

$$S_{qij}, i \in N, j = 1, 2, 3$$

$$S_{r11}, S_{r_{i,i+1}} \text{ 和 } S_{r_i}$$

这些矩阵由计算机生成,对 $i = 1, 2, 3$, 我们得到

$$S_{q11} = \left(\begin{bmatrix} 1.000 & 0.000 & 0.000 \\ 0.000 & 1.000 & 0.000 \end{bmatrix}, \begin{bmatrix} 100 & | & 000 \\ 000 & | & 100 \end{bmatrix} \right)$$

$$S_{q12} = \left(\begin{bmatrix} 0.000 & 1.000 & 0.000 \\ -1.000 & 0.000 & 0.000 \end{bmatrix}, \begin{bmatrix} 100 & | & 000 \\ 000 & | & 100 \end{bmatrix} \right)$$

$$S_{q13} = ([0.000 \ 0.000 \ 1.00 \ 0], [000:000])$$

应当注意,所得多项式矩阵是最小型. 也就是说,通过对多项式矩阵的一系列代数运算后(见符号域的 Rodrigues 公式(3.5.1)),再按照 3.4 节中介绍的算法,完成对矩阵维数的优化.

多项式矩阵的多项式 $S_{qij} = (V_{qij}^T, E_{qij})$ 直接由

$$V_{q11}^T = \begin{bmatrix} 1.000 & 0.000 & 0.000 \\ 0.000 & 1.000 & 0.000 \end{bmatrix}$$

$$E_{q11} = \begin{bmatrix} 1 & 0 & 0 & | & 0 & 0 & 0 \\ 0 & 0 & 0 & | & 1 & 0 & 0 \end{bmatrix}$$

得到. 引入记号

$$V_{q11}^T = \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}$$

得到

$$\begin{aligned} q_{11} &= v_1 x_1 + v_2 x_2 \\ &= \begin{bmatrix} 1.000 \\ 0.000 \\ 0.000 \end{bmatrix} x_1 + \begin{bmatrix} 0.000 \\ 1.000 \\ 0.000 \end{bmatrix} x_2 \end{aligned}$$

按照式(3.4.1)

$$x_1 = \cos q_1$$

$$x_4 = \sin q_1$$

或

$$q_{11} = \begin{bmatrix} \cos q_1 \\ \sin q_1 \\ 0 \end{bmatrix}$$

对其它多项式也做类似解释。

运动学变量的多项式矩阵 $R_i = \{r_{ii}, r_{i,i+1}\}, i \in \{1, 2, 3\}$ 见表 5.3。可以看出, $S_{r_{22}}$ 和 $S_{r_{33}}$ 也依赖于多项式自变量 $x_2 = q_2$ 和 $x_3 = q_3$, 它们是第二、三两个自由度的线性移动的结果。关节轴向量的多项式矩阵列于表 5.4 中。可以看出, 这些多项式矩阵都很简单, 并且只有 S_{e_1} 依赖于 $x_1 = \cos q_1$ 和 $x_4 = \sin q_1$, 它们是机器人圆柱型结构的结果。这样, 所有运动学变量的多项式矩阵 $K_i, i \in N$ 便确定下来了。

算法的下一步, 由式(3.5.10)生成机构惯性矩阵各元素的解析式。这就得到了表 5.5 列出的多项式矩阵, 其中只对 $i \leq j$ 的情况给出了元素 H_{ij} 的多项式矩阵。其余元素可直接由惯性矩阵的对称性

$$S_{H_{ij}} = S_{H_{ji}}, \quad i, j \in N$$

得到, 证明参阅 2.4 节。

模型的 C-矩阵各元素的多项式矩阵, 在下一步由式(3.5.11)生成 (图 2.7 流程图)。结果总结在表 5.6 中。元素 C_{31}^i 显然与元素 C_{13}^i 相同, 这是矩阵 $C^i \in R^{n \times n}, i \in N$ 具有对称性的结果。由这个表还能总结出下式成立:

$$S_{C_{13}^i} = -S_{C_{31}^i}$$

这就是 2.4 节中证明过的反对称性。现在, 我们可以得到结

表 5.3 距离向量的多项式矩阵

i	$S_{r,i}$
1	$([0.0.0.188], [000\ 000\ 000])$
2	$\left(\begin{bmatrix} 0.0.0.01 \\ 0.0.1. \end{bmatrix}, \begin{bmatrix} 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1 \end{bmatrix} \right)$
3	$\left(\begin{bmatrix} 0. & 0.01 & 0. \\ -0.01 & 0. & 0. \\ 0. & 1. & 0. \\ -1. & 0. & 0. \end{bmatrix}, \begin{bmatrix} 1\ 0\ 0 & 0\ 0\ 0 & 0\ 0\ 0 \\ 0\ 0\ 0 & 1\ 0\ 0 & 0\ 0\ 0 \\ 1\ 0\ 0 & 0\ 0\ 0 & 0\ 0\ 1 \\ 0\ 0\ 0 & 1\ 0\ 0 & 0\ 0\ 1 \end{bmatrix} \right)$
i	$S_{r,i+1}$
1	$([0.0.-0.192], [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0])$
2	$([0.0.-0.01], [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0])$
3	$\left(\begin{bmatrix} 0. & -0.44 & 0. \\ 0.44 & 0. & 0. \end{bmatrix}, \begin{bmatrix} 1\ 0\ 0 & 0\ 0\ 0 & 0\ 0\ 0 \\ 0\ 0\ 0 & 1\ 0\ 0 & 0\ 0\ 0 \end{bmatrix} \right)$

表 5.4 关节轴单位向量的多项式矩阵

i	S_{e_i}
1	$([0.0.1.], [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0])$
2	$([0.0.1.], [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0])$
3	$\left(\begin{bmatrix} 0.1.0. \\ -1.0.0. \end{bmatrix}, \begin{bmatrix} 1\ 0\ 0 & 0\ 0\ 0 & 0\ 0\ 0 \\ 0\ 0\ 0 & 1\ 0\ 0 & 0\ 0\ 0 \end{bmatrix} \right)$

论, 矩阵 $C^i, i \in N$ 有下列形式:

$$C^i = \begin{bmatrix} 0 & 0 & C_{13}^i \\ 0 & 0 & 0 \\ C_{13}^i & 0 & 0 \end{bmatrix}$$

表 5.5 圆柱型机器人惯性矩阵元素的多项式矩阵

i, j	$S_{H_{ij}}$
1,1	$\left(\begin{bmatrix} 0.40281 \\ 0.08300 \\ 4.15000 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix} \right)$
1,2	$([0.00000], [0 & 0 & 0 & 0 & 0 & 0])$
1,3	$([0.00000], [0 & 0 & 0 & 0 & 0 & 0])$
2,2	$([11.1500], [0 & 0 & 0 & 0 & 0 & 0])$
2,3	$([0.00000], [0 & 0 & 0 & 0 & 0 & 0])$
3,3	$([4.15000], [0 & 0 & 0 & 0 & 0 & 0])$

表 5.6 圆柱型机器人 C-矩阵非零元素的多项式矩阵

i	k, l	$S_{C_{kl}^i}$
1	1,3	$\left(\begin{bmatrix} 0.04150 \\ 4.15000 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \right)$
1	3,1	$\left(\begin{bmatrix} 0.04150 \\ 4.15000 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \right)$
3	1,1	$\left(\begin{bmatrix} -0.04150 \\ -4.15000 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \right)$

$$C^2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$C^3 = \begin{bmatrix} -C_{13}^1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

算法的最后一步,是按照

$$S_{k_1}^G = ([0.00000], [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0])$$

$$S_{k_2}^G = ([109.3815], [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0])$$

和

$$S_{k_3}^G = ([0.00000], [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0])$$

来计算重力向量 $S_{k_i}^G, i \in N$.

到此为止,我们确定了圆柱型机器人动力学模型中所有元素的多项式矩阵。为明确起见,表 5.7 中列出了所有非零多项式矩阵的解析式。表中利用了解析式(3.2.19)~(3.2.22)的定义。让我们注意,按照这些定义, $x_3 = q_3$ 是第 3 个关节的内坐标。

圆柱型机器人线性化模型的建立

在 2.5 节中,我们介绍了建立机器人闭式线性化模型式(2.5.1)的算法,并把算法总结在图 2.9 和图 2.10 中。在 3.6 节中,我们描述了构造对应数值-符号模型的方法,其一种方法是以 2.5 节中所述算法为基础,另一种方法是以多项式矩阵微分定理为依据。这些算法是等价的,并且容易在数字计算机上实现。这些算法的输出是下列多项式矩阵:

$$S_{H_{ij}^{(s)}}, S_{C_{ij}^{(s)}} \text{ 和 } S_{k_i}^{G(s)}$$

正如 3.6 节中所述,符号“(s)”代表算子 $\frac{\partial}{\partial q_i}$ 。表 5.8 中列出了

偏导数矩阵 $H_{ij}^{(s)}$ 和 $C_{ij}^{(s)}$ 各项中所有非零多项式矩阵。由表 5.7 可直接得到

$$S_{k_i}^{G(s)} = 0, \quad i, s \in N$$

如果把所得多项式矩阵变换为多项式,则容易得到圆柱型机

器人线性化模型的解析式。为此目的,我们先建立矩阵

$$D = \frac{\partial H}{\partial \dot{q}} \ddot{q} + \dot{q}^T \frac{\partial C}{\partial q} \dot{q} + \frac{\partial h_G}{\partial q}$$

然后再建立完整的线性化模型式(2.5.1)。把表 5.8 中的偏导数代入上述矩阵 D 的表示式,得到

$$D = \begin{bmatrix} 0. & 0. & (0.083 + 8.30x_2)\dot{q}_1 + 8.30\dot{q}_1\dot{q}_3 \\ 0. & 0. & 0. \\ 0. & 0. & -4.15\dot{q}_1^2 \end{bmatrix}$$

表 5.7 圆柱型机器人动力学模型矩阵非零元素的对应多项式

惯性矩阵		
(i,j)	多项式 $S_{H_{ij}}$	
(1,1)	$0.40281 + 0.08300x_2 + 4.15x_3^2$	
(2,2)	11.1500	
(3,3)	4.1500	
C-矩阵		
q	(k,l)	多项式 $S_{C_{kl}^q}$
1	1,3	$0.04150 + 4.15000x_2$
重力向量		
i	多项式 S_{h_G}	
2	109.38151	

把上式代入线性化模型式(2.5.1),直接得到

$$\begin{aligned} \delta P_1 = & (0.40281 + 0.08300x_2 + 4.15000x_3^2)\delta\dot{q}_1 \\ & + 2(0.04150 + 4.15000x_2)(\dot{q}_3\delta\dot{q}_1 + \dot{q}_1\delta\dot{q}_3) \\ & + ((0.083 + 8.300x_2)\dot{q}_1 + 8.300\dot{q}_1\dot{q}_3)\delta q_3 \end{aligned}$$

$$\begin{aligned} \delta P_2 &= 11.1500\delta q_2 \\ \delta P_3 &= 4.15000\delta q_3 - 2(0.04150 + 4.15000x_2) \\ &\quad \times q_1\delta q_1 - 4.15000q_1^2\delta q_3 \end{aligned}$$

式中 $x_2 = q_3$.

表 5.8 圆柱型机器人动力学模型矩阵各项偏导数的对应多项式矩阵

惯性矩阵的偏导数			
i, j	k, l	e	$S_{H_{ij}^{(k,l)}}$
1, 1		3	$\left(\begin{bmatrix} 0.08300 \\ 8.30000 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \right)$
C-矩阵的偏导数			
i	k, l	e	$S_{C_{ij}^{(k,l)}}$
1	1, 3	3	$([4.15000], [0 \ 0 \ 0 \ 0 \ 0 \ 0])$
1	3, 1	3	$([4.15000], [0 \ 0 \ 0 \ 0 \ 0 \ 0])$
3	1, 1	3	$([-4.15000], [0 \ 0 \ 0 \ 0 \ 0 \ 0])$

圆柱型机器人灵敏度模型的建立

在 2.6 节中,我们介绍了建立灵敏度模型的算法,并把算法总结在图 2.11 和图 2.12 中,在 3.6 节中,我们描述了对应的数值-符号域模型。

在本例中,假设第 3 个连杆的可变动力学参数为

$$\Delta\theta_3 = \{\Delta m_3, \Delta J_{31}, \Delta J_{32}, \Delta J_{33}\}$$

并由此确定对应的灵敏度函数。为此目的,我们要建立灵敏度矩阵:

$H_s^\mu(q), C_s^\mu(q)$ 和 $h_s^{\mu}(q)$

其中 $s = 3$ 是具有可变参数的连杆序号, $\mu = 0$ 对应 Δm_s , 而 $\mu = 1, 2, 3$ 对应参数 $\Delta J_{s\mu}$ 的不同值。按照式(3.6.7)计算惯性矩阵的灵敏度模型的各个元素, 如表 5.9 所示。

表 5.9 圆柱型机器人惯性矩阵的灵敏度矩阵非零元素的多项式矩阵

i, j	s	μ	S_{ij}^{μ}
1,1	3	0	$\left(\begin{bmatrix} 0.020 \\ 1.000 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix} \right)$
2,2	3	0	$([1.000], [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0])$
3,3	3	0	$([1.000], [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0])$
1,1	3	3	$([1.000], [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0])$

利用式(3.6.8)和(3.6.9), 除元素

$$C_{313}^{1,0} = C_{331}^{1,0} = -C_{311}^{3,0}$$

之外, 得到 $C_{ijk}^{\mu} = 0, \{i, k, l\} \in N^3, s = 3, \mu \in \{0, 1, 2, 3\}$ 。这些元素的多项式矩阵为

$$S_{C_{313}^{1,0}} = \left(\begin{bmatrix} 0.010 \\ 1.000 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \right)$$

除开 $i = 2$ 和 $\mu = 0$ 之外, 由式(3.6.10)得到

$$S_{h_{ii}^{\mu}} = 0, s = 3, i \in N$$

对这一情况, 有

$$S_{h_{12}^{3,0}} = ([9.81], [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0])$$

到此为止, 我们就得到了灵敏度模型式 (2.6.20) — (2.6.22) 的全部元素。可以证明, 参数 J_{31} 和 J_{32} 对系统动力

学没有影响。

圆柱型机器人解析模型的优化计算和程序码的生成

让我们考虑圆柱型机器人机构的动力学模型。表 5.5 和表 5.6 中给出了描述这种模型的多项式矩阵。尽管这些矩阵的结构已经十分简单,但为了说明在第四章中研究的算法,下面我们仍然给出动力学模型矩阵各元素优化计算的结果。首先,让我们考虑表 5.5 中列出的多项式矩阵 $S_{H_{ij}}$ 。对其中的每一个矩阵,我们都生成了优化计算序列,用矩阵式表示时,它们都用两个矩阵

$$E_{H_{ij}} \text{ 和 } V_{H_{ij}}$$

表示,其中 E 矩阵和 V 矩阵分别表示优化基本计算图的结点和支路(第四章)。计算结果总结在表 5.10 中。在此基础上,容易画出优化计算图。例如,就元素 H_{11} 而言,可得如下计算图:

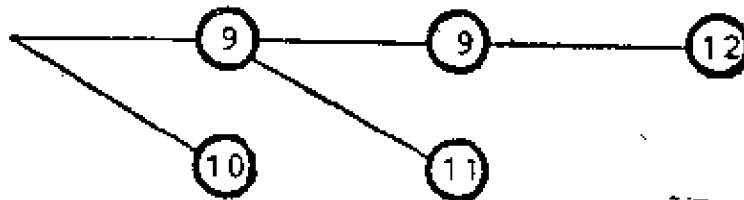


表 5.10 圆柱型机器人惯性矩阵基本计算图的矩阵表示

i, j	$E_{H_{ij}}$	$X_k, k > 9$	$V_{H_{ij}}$
1,1	$\begin{bmatrix} 9 & 9 & 12 \\ 10 & 11 & 0 \end{bmatrix}$	$X_{10} = 0.40281$ $X_{11} = 0.08300$ $X_{12} = 4.15000$	$\begin{bmatrix} 2 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
2,2	$[10 \ 0]$	$X_{10} = 11.1500$	$[0 \ 0]$
3,3	$[10 \ 0]$	$X_{10} = 4.15000$	$[0 \ 0]$

其中 $x_9 = q_3$.

现在, 让我们考虑如何生成系统 C-矩阵的优化计算序列. 由于矩阵元素 $C^i, (i = 1, 2, 3)$ 可用元素 C_{13}^i 表示, 因此我们只给出这一元素的矩阵表示就够了,

$$E_{C_{13}^i} = \begin{bmatrix} 9 & 11 \\ 10 & 0 \end{bmatrix}, V_{C_{13}^i} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

式中 $X_{10} = 0.04150, X_{11} = 4.15000$. 元素 h_i^c 的优化序列是简单的, 因为这些元素总是零或常数.

按照 4.3 节中所述定理, 我们可建立动力学模型矩阵实时计算的程序码. 程序码由 FORTRAN 语言的专家程序完成. 专家程序同样自动产生下列内容:

- 1) SUBROUTINE <name>,
- 2) COMMON-block,
- 3) RETURN 和 END 类型的结束指令,

在这里, <name> 是用户传送给专家程序的所需子程序名. 子程序的输入是机器人的关节角度和关节速度, 把它们写入 COMMON-block/UG/, 子程序的输出是动力学模型矩阵:

$$H(q, \dot{q}) \text{ 和 } h(q, \dot{q}, \theta)$$

这些数据存放在 COMMON-block/MATRIC/中.

至此, 由专家程序获得的源程序就可用适当编译器进行编译了. 图 5.4 列出了程序清单.

```
SUBROUTINE UMS2
COMMON/MATRIC/H(6,6), C(6,6,6), G(6)
COMMON/UG/Q(6)
X7 = Q(1)
X8 = Q(2)
X9 = Q(3)
```

```

G(3) = 0.
G(2) = 0.10938E + 03
G(1) = 0.
C(3,2,2) = 0.
C(3,2,1) = 0.
Q1 = X9 * (-0.41500E + 01)
C(3,1,1) = Q1 - 0.41500E - 01D
C(2,3,3) = 0.
C(2,1,1) = 0.
C(1,3,3) = 0.
C(1,3,2) = 0.
C(1,2,2) = 0.
H(3,3) = 0.41500E + 01
H(3,2) = 0.
H(3,1) = 0.
H(2,2) = 0.11150E + 02
H(2,1) = 0.
P1 = X9 * 0.41500E + 01
Q1 = X9 * (p1 + 0.83000E - 01)
H(1,1) = Q1 + 0.40281E + 00

```

图 5.4 由专家程序(输出)获得的建立圆柱型
机器人动力学模型矩阵的程序

分析图 5.4 中的程序可以看出，尽管我们有了优化计算方法，并得到了计算程序，但仍然存在一些冗余操作。这就是程序第 12 行和第 24 行的乘法运算是完全相同的。产生这种冗余性的原因，是因为矩阵 H 和矩阵 C 不同项的优化计算序列是独立生成的。然而，如果要求算法把不同项中有相同表

示式这一点考虑进去，则势必大大增加其自身的复杂性和输出码的生成时间。当然，如果考虑到 2.4 节中描述的对称性和反对称性，就可减少这种冗余操作。

最后，在图 5.4 中简单地数一数浮点乘法运算次数 n_M 和加法运算次数 n_A ，有

$$n_M = 3, n_A = 3$$

在 PDP11/70 小型计算机上该程序执行时间约为 $100 \mu s$ 。至此，圆柱型机器人动力学模型矩阵的计算在功能非常有限的微型计算机上照样可以完成。

5.3 节肢机器人

在本例我们考察节肢结构的操作手，图 5.5 给出了这种机器人的一个例子。

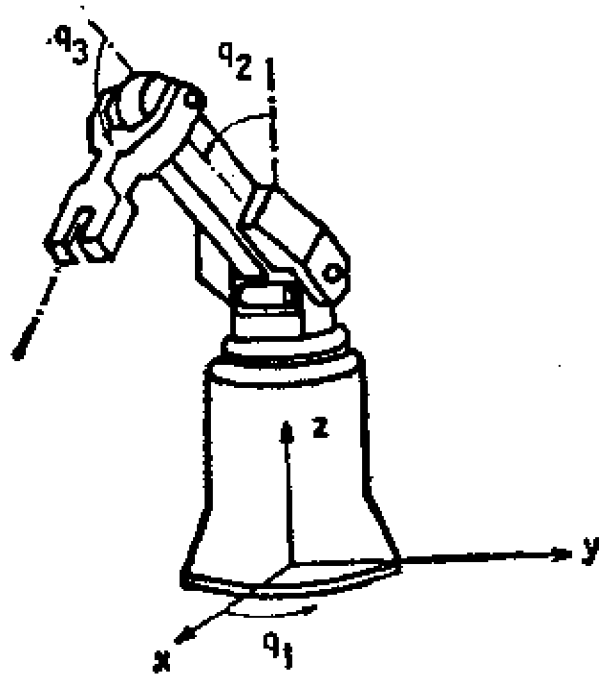


图 5.5 节肢机器人

和前例一样,这种机器人机构结构的对应模型如图 5.6 所示。机构的连杆用 C_1, C_2 和 C_3 表示,关节联接用 Z_1, Z_2 和 Z_3 表示。由于三个自由度均为旋转型,因此这种机构具有 RRR 型结构。各连杆及其几何参数分别在图 5.7 中给出。

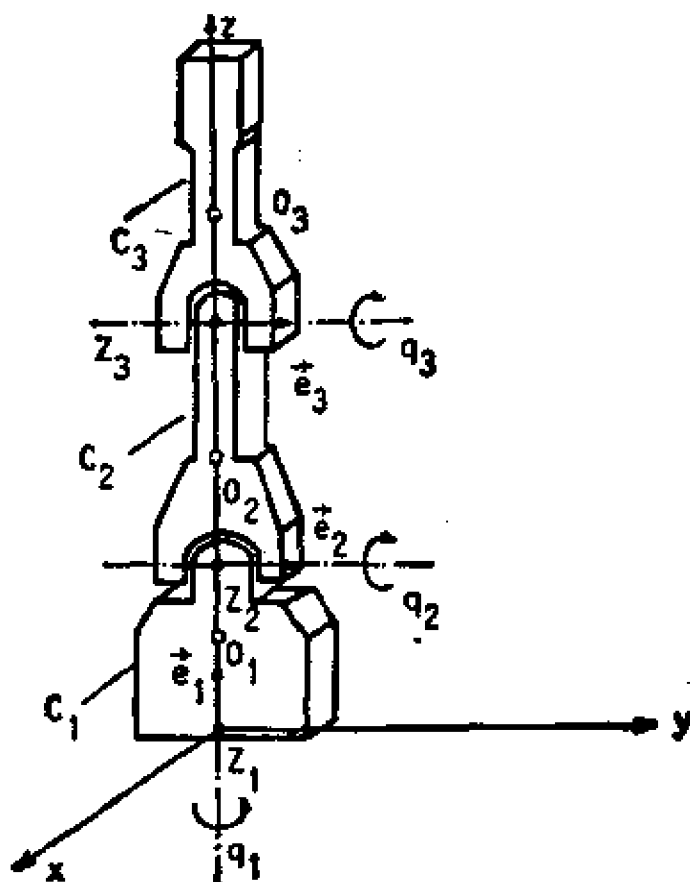


图 5.6 节肢机器人机构的模型

机构连杆运动学参数 $\tilde{R}_i = \{\tilde{r}_{ii}, \tilde{r}_{i,i+1}\}$ 和 $\tilde{E}_i = \{\tilde{e}_i\}$, $i \in N = \{1, 2, 3\}$ 的数据由表 5.11 列出。

机构的动力学参数见表 5.12。表中未列的参数对构造节肢机器人的动力学模型是无关紧要的。

节肢机器人非线性解析模型的推导

建立非线性解析模型算法(图 2.7)的第一步是确定机构

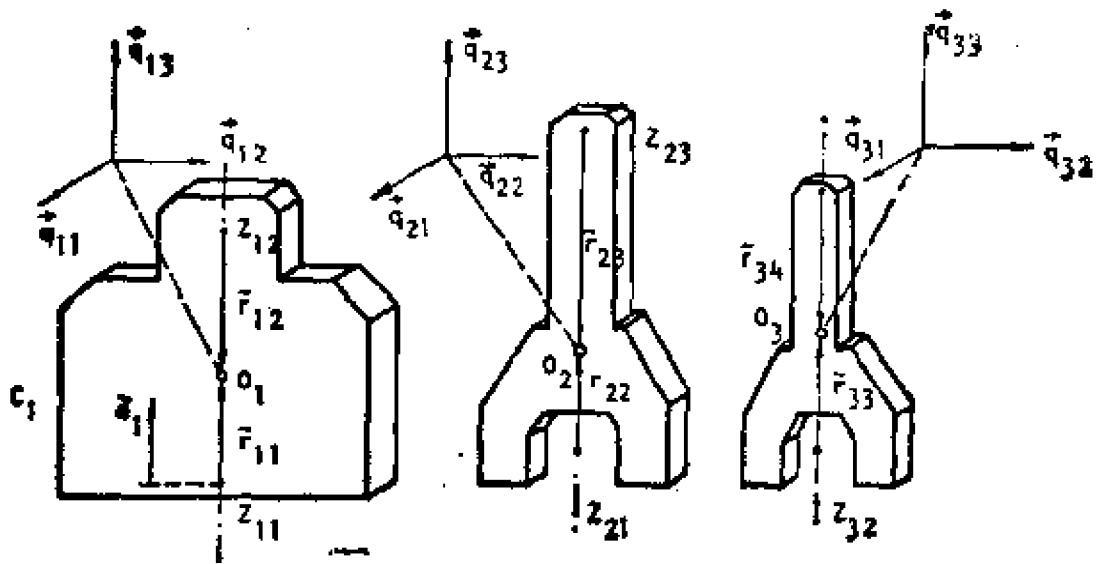


图 5.7 节肢机器人机构的各个连杆

表 5.11 节肢机器人的运动学参数

i	$r_{ij}[m]$	$r_{i,i+1}[m]$	δ_i	ξ_i
1	(0.5, 0.5, 0.25)	(0.5, 0.5, -0.25)	(0.5, 0.5, 1.)	0
2	(0.5, 0.5, 0.55)	(0.5, 0.5, -0.45)	(0.5, 1.5, 0.)	0
3	(0.5, 0.5, 0.20)	(0.5, 0.5, -0.618)	(0.5, 1.5, 0.)	0

表 5.12 节肢机器人的运动学参数

i	$m_i[\text{kg}]$	$J_{i1}[\text{kgm}^2]$	$J_{i2}[\text{kg m}^2]$	$J_{i3}[\text{kg m}^2]$
1	—	—	—	4.57
2	1.52.	—	21.74	—
3	67.	—	1.22	—

表 5.13 节肢机器人局部坐标系单位向量
所对应的多项式矩阵

i	j	S_{eij}
1	1	$\left(\begin{bmatrix} 1. & 0. & 0. \\ 0. & 1. & 0. \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \right)$
	2	$\left(\begin{bmatrix} 0. & 1. & 0. \\ -1. & 0. & 0. \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \right)$
	3	$([0.0.1.], [0 0 0 0 0 0])$
2	1	$\left(\begin{bmatrix} 1. & 0. & 0. \\ 0. & 1. & 0. \\ 0. & 0. & -1. \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \right)$
	2	$\left(\begin{bmatrix} 0. & 1. & 0. \\ -1. & 0. & 0. \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \right)$
	3	$\left(\begin{bmatrix} 0. & 0. & 1. \\ 1. & 0. & 0. \\ 0. & 1. & 0. \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \right)$
3	1	$\left(\begin{bmatrix} 1. & 0. & 0. \\ 0. & 0. & -1. \\ 0. & 1. & 0. \\ 0. & 0. & -1. \\ -1. & 0. & 0. \\ 0. & 1. & 0. \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \right)$
	2	$\left(\begin{bmatrix} 0. & 1. & 0. \\ -1. & 0. & 0. \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \right)$
3	3	$\left(\begin{bmatrix} 0. & 0. & 1. \\ 1. & 0. & 0. \\ 0. & 1. & 0. \\ 1. & 0. & 0. \\ 0. & 1. & 0. \\ 0. & 0. & -1. \end{bmatrix}, \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \right)$

表 5.14 节肢机器人的多项式矩阵 $S_{r,i}$, $S_{r,ii}$ 和 $S_{r,i+1}$

i	$S_{r,i}$
1	$([0.0.1.], [0\ 0\ 0\ 0\ 0\ 0])$
2	$\left(\begin{bmatrix} 0. & 1. & 0. \\ -1. & 0. & 0. \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \right)$
3	$\left(\begin{bmatrix} 0. & 1. & 0. \\ -1. & 0. & 0. \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \right)$
i	r_{ii}
1	$([0.0.0.25], [0\ 0\ 0\ 0\ 0\ 0])$
2	$\left(\begin{bmatrix} 0.55 & 0. & 0. \\ 0. & 0.55 & 0. \\ 0. & 0. & 0.55 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \right)$
3	$\left(\begin{bmatrix} 0. & 0. & 0.2 \\ 0.2 & 0. & 0. \\ 0. & 0.2 & 0. \\ 0.2 & 0. & 0. \\ 0. & 0.2 & 0. \\ 0. & 0. & -0.2 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \right)$
i	$S_{r,i+1}$
1	$([0.0.-0.25], [0\ 0\ 0\ 0\ 0\ 0])$
2	$\left(\begin{bmatrix} 0. & 0. & -0.45 \\ -0.45 & 0. & 0. \\ 0. & -0.45 & 0. \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \right)$

的基始位置。这一步的输出是

$$Q_i^0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$r_{ii}^0 = \hat{r}_{ii}, \quad r_{i,i+1}^0 = \hat{r}_{i,i+1}, \quad i = 1, 2, 3$$

和

$$e_{i+1}^0 = \hat{e}_{i+1}, \quad i = 1, 2$$

在算法的下一步是确定机构各连杆的位置，它们是关节坐标 q_1, q_2 和 q_3 的函数。这一步的输出是多项式矩阵

$$S_{q_i}, S_{r_{ii}}, S_{r_{i,i+1}}, S_{e_j}, \quad i \in N, \quad j = 1, 2, 3$$

由计算机生成这些矩阵，得到表 5.13 和表 5.14 列出的数据。在表中指数矩阵只有 6 列，第 7, 8, 9 列显然是零向量。很明显，系统没有滑动关节时，这种说法才成立。

表 5.15 节肢机器人惯性矩阵各元素的多项式矩阵

i, j	$S_{H_{ij}}$
1,1	$\left(\begin{array}{c} 4.57000 \\ 112.98000 \\ 26.80000 \\ 26.80000 \\ 2.68000 \\ 5.36000 \\ 2.68000 \end{array} \right), \left(\begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 2 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 2 & 0 & 2 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 2 & 0 & 0 & 0 & 2 \end{array} \right)$
1,2	$([0.00000], [0 \ 0 \ 0 \ 0 \ 0 \ 0])$
1,3	$([0.00000], [0 \ 0 \ 0 \ 0 \ 0 \ 0])$
2,2	$\left(\begin{array}{c} 138.62001 \\ 26.80000 \end{array} \right), \left(\begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{array} \right)$
2,3	$\left(\begin{array}{c} 3.90000 \\ 13.40000 \end{array} \right), \left(\begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{array} \right)$
3,3	$([3.90000], [0 \ 0 \ 0 \ 0 \ 0 \ 0])$

算法的再下一步是按照方程式(3.5.10),建立机器人机构惯性矩阵各项的多项式矩阵,其结果列于表 5.15 中。由于惯性矩阵具有对称性(参阅 2.4 节),因此在表 5.15 中只包含元素 H_{ij} , $i \leq j$ 。从表中可以看出, $H_{12} = H_{21}$ 和 $H_{13} = H_{31}$ 都是零元素。这表明,第一个自由度(绕垂直轴旋转)对第二个和第三个自由度(绕水平轴旋转)是惯性“解耦”的,因为 $H_{33} = 3.9 = \text{const}$ (常数),所以第三个自由度是承受常量惯性负荷的。

为了便于说明,让我们写出矩阵 $H(q, \theta) \in R^{3 \times 3}$

$$H(q, \theta) = \begin{bmatrix} 4.57 + 112.98s^2q_2 & & \\ + 26.8cq_3s^2q_2 & & \\ + 26.8cq_2sq_2sq_3 & 0 & 0 \\ + 2.68c^2q_3s^2q_2 & & \\ + 5.36cq_2cq_3sq_2sq_3 & & \\ + 2.68c^2q_2s^2q_3 & & \\ \hline 0 & 138.62 + 26.8cq_3 & 3.9 + 13.4cq_3 \\ \hline 0 & 3.9 + 13.4cq_3 & 3.9 \end{bmatrix}$$

它与表 5.15 中数据相对应,式中 cq_i 和 sq_i 分别表示 $\cos q_i$ 和 $\sin q_i$ 。

算法的再下一步是按照方程式 (3.5.11) 生成 C -矩阵各元素的数值-符号型。 C -矩阵的所有元素都可用 C_{21}^1 , C_{41}^1 , C_{32}^1 和 C_{33}^1 等四个元素表示,其余元素利用对称性和反对称性得到。于是,矩阵 C^i , $i \in N$ 可写为

$$C^1 = \begin{bmatrix} 0 & C_{21}^1 & C_{41}^1 \\ C_{21}^1 & 0 & 0 \\ C_{41}^1 & 0 & 0 \end{bmatrix}$$

表 5.16 节肢机器人 C-矩阵非零元素的多项式矩阵

i	k, l	Sc_{kl}^i
1	2,1	$\begin{bmatrix} 112.980 \\ 26.800 \\ 13.400 \\ 2.680 \\ 2.680 \\ -13.400 \\ -2.680 \\ -2.680 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 & 0 & 1 \\ 0 & 1 & 2 & 0 & 1 & 0 \\ 0 & 2 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 2 & 1 \\ 0 & 0 & 1 & 0 & 2 & 1 \\ 0 & 1 & 0 & 0 & 1 & 2 \end{bmatrix}$
1	3,1	$\begin{bmatrix} 13.400 \\ 2.680 \\ 2.680 \\ -13.400 \\ -2.680 \\ -2.680 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 2 & 0 & 1 & 0 \\ 0 & 2 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 2 & 1 \\ 0 & 0 & 1 & 0 & 2 & 1 \\ 0 & 1 & 0 & 0 & 1 & 2 \end{bmatrix}$
2	3,2	$([-13.400], [0 \ 0 \ 0 \ 0 \ 0 \ 1])$
2	3,3	$([-13.400], [0 \ 0 \ 0 \ 0 \ 0 \ 1])$

$$C^2 = \begin{bmatrix} -C_{21}^1 & 0 & 0 \\ 0 & 0 & C_{32}^2 \\ 0 & C_{32}^2 & C_{33}^2 \end{bmatrix}$$

$$C^3 = \begin{bmatrix} -C_{31}^1 & 0 & 0 \\ 0 & -C_{32}^2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

表 5.16 列出了 C-矩阵各元素的多项式矩阵。

算法的最后一步是按照方程式(3.5.12), 计算重力向量的多项式矩阵 S_{iG} , $i \in N$, 得到

$$S_{A_1^G} = 0$$

$$R_{A_2^G} = \left(\begin{bmatrix} -1477.3861 \\ -131.4540 \\ -131.4540 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \right)$$

$$S_{A_3^G} = \left(\begin{bmatrix} -131.4540 \\ -131.4540 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \right)$$

至此，节肢机器人动力学模型矩阵所有元素对应的多项式矩阵就完全确定了。

节肢机器人线性化模型的建立

为获得机器人线性化模型的数值-符号型，必须确定多项式矩阵：

$$S_{H_{ij}^{(s)}}, S_{C_{ij}^{(s)}} \text{ 和 } S_{A_j^G(s)}$$

$$i, j, k, s \in N, (s) = \frac{\partial}{\partial q_i}$$

为此，需要利用 3.6 节介绍的，多项式矩阵偏导数定理。按照该定理所述算法，就可得到计算机辅助生成线性化模型的程序。节肢机器人惯性矩阵各项偏导数所对应的多项式矩阵列于表 5.17。表中只包括非零偏导数。可以看出，节肢机器人惯性矩阵的偏导数矩阵有下列结构：

$$\frac{\partial H(q, \theta)}{\partial q_2} = \begin{bmatrix} H_{11}^{(2)} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\frac{\partial H(q, \theta)}{\partial q_3} = \begin{bmatrix} H_{11}^{(3)} & 0 & 0 \\ 0 & H_{22}^{(3)} & H_{23}^{(3)} \\ 0 & H_{32}^{(3)} & 0 \end{bmatrix}$$

表 5.17 节肢机器人 C-矩阵非零元素的多项式矩阵

i, j	r	$S_{H_{ij}^{(r)}}$
1,1	2	$\begin{pmatrix} 225.960 \\ 53.600 \\ 26.800 \\ -53.600 \\ 5.360 \\ 5.360 \\ -10.720 \\ -2.680 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 2 & 1 \\ 0 & 1 & 2 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 2 & 1 \\ 0 & 1 & 0 & 0 & 1 & 2 \end{pmatrix}$
1,1	3	$\begin{pmatrix} -26.800 \\ 26.800 \\ -5.360 \\ 5.360 \\ -10.720 \\ 5.360 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 & 2 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 2 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 2 \\ 0 & 2 & 1 & 0 & 0 & 1 \end{pmatrix}$
2,2	3	$([-26.800], [0 \ 0 \ 0 \ 0 \ 0 \ 1])$
2,3	3	$([-13.400], [0 \ 0 \ 0 \ 0 \ 0 \ 1])$

动力学模型 C-矩阵各项的偏导数由表 5.18 给出, 可以看出, 像惯性矩阵情况一样, 节肢机器人 C-矩阵的偏导数矩阵含有大量零元素。

构造线性化模型的最后一步是建立重力向量各元素偏导数的多项式矩阵, 我们得到

$$S \frac{\partial h_2^G}{\partial q_2} = \left(\begin{bmatrix} -1477.3861 \\ -131.4540 \\ 131.4540 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \right)$$

$$S \frac{\partial h_2^G}{\partial q_3} = \left(\begin{bmatrix} 131.4540 \\ -131.4540 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \right)$$

表 5.18 节肢机器人 C-矩阵各项偏导数的多项式矩阵

i	k,l	j	$S_{C_{kl}^{(j)}}$
1	2,1	2	$\begin{pmatrix} 115.660 \\ -231.320 \\ 26.800 \\ -53.600 \\ -53.600 \\ -5.360 \\ 10.720 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 2 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 2 & 2 \end{pmatrix}$
1	2,1	3	$\begin{pmatrix} -26.800 \\ 13.400 \\ -10.720 \\ 2.680 \\ -5.360 \\ -13.400 \\ -5.360 \\ 10.720 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 2 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 1 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2 \end{pmatrix}$
1	3,1	2	$\begin{pmatrix} 13.400 \\ -26.800 \\ 2.680 \\ -5.360 \\ -5.360 \\ 10.720 \\ -26.800 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2 \\ 0 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}$
1	3,1	3	$\begin{pmatrix} -13.400 \\ -10.720 \\ 2.720 \\ -5.360 \\ -13.400 \\ -5.360 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 2 \\ 0 & 0 & 1 & 0 & 2 & 0 \\ 0 & 0 & 2 & 0 & 2 & 0 \end{pmatrix}$
2	3,2	3	$([-13.400], [0 \ 0 \ 1 \ 0 \ 0 \ 0])$
2	3,3	3	$([-13.400], [0 \ 0 \ 1 \ 0 \ 0 \ 0])$

$$S \frac{\partial h_3^G}{\partial q_2} = \left(\begin{bmatrix} -131.4540 \\ 131.4540 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \right)$$

$$S \frac{\partial h_3^G}{\partial q_1} = \left(\begin{bmatrix} 131.4540 \\ -131.4540 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \right)$$

至此,线性化模型矩阵的推导全部完成了。

节肢机器人灵敏度模型的建立

在这一节中,我们以节肢机器人为例,说明建立灵敏度模型的方法(参阅 3.6 节)。在圆柱型机器人例子中,我们假设第 3 个连杆的参数是可变的。在这里,我们假设第 2 个连杆的参数是可变的。这一假设是可行的,因为构造灵敏度模型的算法与选择哪个连杆具有可变参数是无关的。

让我们考虑参数集合

$$\theta_2 = \{m_2, J_{21}, J_{22}, J_{23}\}$$

并假设是可变的。我们要建立灵敏度矩阵:

$$H_i^s(q), C_i^s(q) \text{ 和 } h_i^s(q)$$

其中 $S = 2$ 是具有可变参数连杆的序号, $\mu = 0$ 对应参数 m_2 , $\mu = 1, 2, 3$ 对应参数 $\Delta J_{i,\mu}$ 。按照式(3.6.7),我们得到表 5.19 中所列数据。从表中可以看出,灵敏度模型比完全的非线性模型大大简化了。还可以看出,由于参数 J_{21} 和 J_{23} 不包括在模型矩阵中,因此这些参数变化的灵敏度等于零。

节肢机器人近似模型的建立

在 3.7 节中,我们介绍了一种根据规定的允许误差,建立机器人近似模型的方法。我们首先进行规范化,通过线性变换 $z_i = a_i x_i$, 把自变量 $x = \{x_1, \dots, x_N\}$ 域变换为 $z = \{z_1, \dots, z_N\}$ 域,其中 a_i 由

表 5.19 节肢机器人灵敏度模型非零元素的多项式矩阵

惯 性 矩 阵			
i, j	s	μ	$S_{H_{ij}^{in}}$
1,1	2	0	$([0.3025], [000 \ 020])$
2,2	2	0	$([0.3025], [000 \ 000])$
2,2	2	2	$([1.0000], [000 \ 000])$
C-矩阵			
i, k, l	s	μ	$S_{H_{ikl}^c}$
1,2,1	2	0	$([0.3025], [010 \ 010])$
重 力 向 量			
i	s	μ	$S_{H_{ii}^{Gx}}$
2	2	0	$([5.39551], [000 \ 010])$

$$a_i = 1, \quad i = 1, \dots, 2n$$

$$a_i = \sup_t |q_i(t)|, \quad i = 2n + 1, \dots, 3n$$

确定。由于节肢机器人只有旋转自由度,而没有平移关节,因此动力学模型矩阵与坐标 x_{2n+1}, \dots, x_{3n} 无关。这就允许我们可令

$$a_i = 1, \quad i = 1, \dots, 3n$$

它对应单位变换 $X \rightarrow X$ 。

现在让我们考虑表 5.15 中列出的惯性矩阵。为了完成系数的规范化,采用对行最大元素的规范化(见式(3.7.20)),得到

$$S_{H_{11}} = \left(112.98 \begin{bmatrix} 0.0404 \\ 1.0000 \\ 0.2372 \\ 0.2372 \\ 0.0237 \\ 0.0474 \\ 0.0237 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 2 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 2 & 0 & 2 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 2 & 0 & 0 & 0 & 2 \end{bmatrix} \right)$$

$$S_{H_{22}} = \left(138.62 \begin{bmatrix} 1.0000 \\ 0.1933 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \right)$$

$$S_{H_{23}} = \left(138.62 \begin{bmatrix} 0.0281 \\ 0.0967 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \right)$$

$$S_{H_{32}} = \left(138.62 \begin{bmatrix} 0.2910 \\ 1.0000 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \right)$$

$$S_{H_{33}} = (13.40[0.2910], [0 \ 0 \ 0 \ 0 \ 0 \ 0])$$

根据定义式(3.7.12), 设给定允许误差是 5%。现在我们利用 3.7 节中的定理, 由于 $|0.0237| + |0.0237| < 0.05$, 因此可以从 H_{11} 的结构矩阵中消去第 5 和第 7 行。同理, 多项式矩阵 $S_{H_{22}}$ 的第 1 行也可消去, 于是得到

$$\tilde{S}_{H_{11}} = \left(112.98 \begin{bmatrix} 0.0404 \\ 1.0000 \\ 0.2372 \\ 0.2372 \\ 0.0474 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 2 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix} \right)$$

$$\tilde{S}_{H_{22}} = (138.62[0.0967], [0 \ 0 \ 1 \ 0 \ 0 \ 0])$$

$$\tilde{S}_{H_{ij}} = S_{H_{ij}}, \quad (i, j) = \begin{cases} (2, 2) \\ (3, 2) \\ (3, 3) \end{cases}$$

式中 \tilde{S} 表示多项式矩阵的近似值。驱动转矩现可由近似动力学方程(见式(2.2.17)和(2.3.1))

$$\begin{aligned} P_1 &= \tilde{H}_{11}\dot{q}_1 + h_1 \\ P_2 &= \tilde{H}_{22}\dot{q}_2 + \tilde{H}_{23}\dot{q}_3 + h_2 \\ P_3 &= \tilde{H}_{32}\dot{q}_2 + \tilde{H}_{33}\dot{q}_3 + h_3 \end{aligned}$$

表示, 式中

$$\begin{aligned} h_1 &= 2C_{21}^1\dot{q}_1\dot{q}_2 + 2C_{31}^1\dot{q}_1\dot{q}_3 + h_1^G \\ h_2 &= C_{12}^1\dot{q}_1^2 + 2C_{32}^2\dot{q}_2\dot{q}_3 + C_{33}^2\dot{q}_3^2 + h_2^G \\ h_3 &= C_{13}^1\dot{q}_1^2 + C_{32}^2\dot{q}_2^2 + h_3^G \end{aligned}$$

$C_{kl}^i (i, k, l \in N)$ 在表 5.16 中给出。

现在的任务是在式(3.7.12)的意义下来确定 h_1, h_2 和 h_3 的近似表示式, 但要求允许误差不超过 5%。为此, 先在内部速度为最大 $\dot{q}_{i(\max)}$ 条件下把 $h_i, i \in N$ 的表示式规范化。例如

$$|\dot{q}_1|_{\max} = 1 \frac{\text{rad}}{\text{s}}$$

$$|\dot{q}_2|_{\max} = |\dot{q}_3|_{\max} = 2 \frac{\text{rad}}{\text{s}}$$

让我们在 h_1, h_2 和 h_3 的表示式中引入 $v_i = \frac{\dot{q}_i}{|\dot{q}_i|_{\max}}$, 得到

$$\begin{aligned} h_1 &= 4C_{21}^1v_1v_2 + 4C_{31}^1v_1v_3 + h_1^G \\ h_2 &= -C_{21}^1v_1^2 + 8C_{32}^2v_2v_3 + 4C_{33}^2v_3^2 + h_2^G \\ h_3 &= -C_{31}^1v_1^2 - 4C_{32}^2v_2^2 + h_3^G \end{aligned}$$

式中 $|v_i| \leq 1$ 。

接着, 我们建立 $h_i, i \in N$ 的多项式矩阵, 让我们对 C-矩阵和重力向量各元素引入下列符号:

$$S_{C_{kl}^i} = (V_{kl}^i, E_{kl}^i), \quad k, l, i \in N$$

$$S_{h_i^G} = (V_i^G, E_i^G), i \in N$$

式中指数矩阵按自变量排列集合

$$x = (\cos q_1, \cos q_2, \cos q_3, \sin q_1, \sin q_2, \sin q_3)$$

定义, * 可简写为

$$x = (\cos q_i | \sin q_i)$$

我们用元素 v_1, v_2 和 v_3 扩充自变量集合, 得到

$$x_s = (\cos q_i | \sin q_i | v_i)$$

于是, h_1, h_2 和 h_3 可表示为下列多项式矩阵:

$$h_1 = \left(\begin{array}{c} 4V_{21}^1 \\ \hline 4V_{31}^1 \\ \hline V_1^G \end{array} \right), \left(\begin{array}{c|ccc} E_{21}^1 & 1 & 1 & 0 \\ \hline & \dots & & \\ & 1 & 1 & 0 \\ \hline E_{31}^1 & 1 & 0 & 1 \\ & \dots & & \\ & 1 & 0 & 1 \\ \hline E_1^G & 0 & 0 & 0 \\ & \dots & & \\ & 0 & 0 & 0 \end{array} \right)$$

$$h_2 = \left(\begin{array}{c} -V_{21}^1 \\ \hline 8V_{32}^2 \\ \hline 4V_{33}^2 \\ \hline V_2^G \end{array} \right), \left(\begin{array}{c|ccc} E_{21}^1 & 2 & 0 & 0 \\ \hline & \dots & & \\ & 2 & 0 & 0 \\ \hline E_{32}^2 & 0 & 1 & 1 \\ & \dots & & \\ & 0 & 1 & 1 \\ \hline E_{33}^2 & 0 & 0 & 2 \\ & \dots & & \\ & 0 & 0 & 2 \\ \hline E_2^G & 0 & 0 & 0 \\ & \dots & & \\ & 0 & 0 & 0 \end{array} \right)$$

$$h_3 = \left(\begin{array}{c} -V_{31}^1 \\ \hline -4V_{32}^2 \\ \hline V_3^6 \end{array} \right), \left(\begin{array}{c|ccc} E_{31}^1 & 2 & 0 & 0 \\ & \dots & & \\ \hline E_{32}^2 & 0 & 2 & 0 \\ & & \dots & \\ \hline E_3^6 & 0 & 0 & 0 \\ & & \dots & \\ & 0 & 0 & 0 \end{array} \right)$$

即

$$h_i = (V_i, E_i), i \in N$$

利用表 5.16 中的数据,得到

$$S_h = \left(\begin{array}{c|cccccc} 451.920 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 107.200 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 53.600 & 0 & 2 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 10.720 & 0 & 1 & 2 & 0 & 1 & 0 & 1 & 1 & 0 \\ 10.720 & 0 & 2 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ -53.600 & 0 & 0 & 0 & 0 & 2 & 1 & 1 & 1 & 0 \\ -10.720 & 0 & 0 & 1 & 0 & 2 & 1 & 1 & 1 & 0 \\ -10.720 & 0 & 1 & 0 & 0 & 1 & 2 & 1 & 1 & 0 \\ \hline 53.600 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 10.720 & 0 & 1 & 2 & 0 & 1 & 0 & 1 & 0 & 1 \\ 10.720 & 0 & 2 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ -53.600 & 0 & 0 & 0 & 0 & 2 & 1 & 1 & 0 & 1 \\ -10.720 & 0 & 0 & 1 & 0 & 2 & 1 & 1 & 0 & 1 \\ -10.720 & 0 & 1 & 0 & 0 & 1 & 2 & 1 & 0 & 1 \end{array} \right)$$

$$S_{A_1} = \begin{bmatrix} 112.980 & 0 & 1 & 0 & 0 & 1 & 0 & 2 & 0 & 0 \\ -26.800 & 0 & 1 & 1 & 0 & 1 & 0 & 2 & 0 & 0 \\ -13.400 & 0 & 2 & 0 & 0 & 0 & 1 & 2 & 0 & 0 \\ -2.680 & 0 & 1 & 2 & 0 & 1 & 0 & 2 & 0 & 0 \\ -2.680 & 0 & 2 & 1 & 0 & 0 & 1 & 2 & 0 & 0 \\ 13.400 & 0 & 0 & 0 & 0 & 2 & 1 & 2 & 0 & 0 \\ 2.680 & 0 & 0 & 1 & 0 & 2 & 1 & 2 & 0 & 0 \\ 2.680 & 0 & 1 & 0 & 0 & 1 & 2 & 2 & 0 & 0 \\ \hline -107.200 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ -53.600 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 2 \\ \hline -1477.386 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -131.454 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ -131.454 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$S_{A_2} = \begin{bmatrix} -13.400 & 0 & 1 & 1 & 0 & 1 & 0 & 2 & 0 & 0 \\ -2.680 & 0 & 1 & 2 & 0 & 1 & 0 & 2 & 0 & 0 \\ -2.680 & 0 & 2 & 1 & 0 & 0 & 1 & 2 & 0 & 0 \\ 13.400 & 0 & 0 & 0 & 0 & 2 & 1 & 2 & 0 & 0 \\ 2.680 & 0 & 0 & 1 & 0 & 2 & 1 & 2 & 0 & 0 \\ 2.680 & 0 & 1 & 0 & 0 & 1 & 2 & 2 & 0 & 0 \\ \hline 53.600 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 2 & 0 \\ \hline -131.454 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ -131.454 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

现在，我们对这些多项式矩阵运用 3.7 节中提供的近似方法。由于允许误差为 5%，因此可按表 5.20 消去一些行，因为消去那些行不是唯一的，所以表 5.20 中提供的数据只表示

一种消去行的可能组合。可以看出，多项式矩阵中占总数约 30% 的行被消去了。但要注意，这并非表明乘法/加法运算次数也同样减少 30%，因为这样做破坏了 C -矩阵的反对称性，而原来多项式矩阵 $S_{k_i}, i \in N$ 是具有这种特性的。例如，从 S_{k_1} 中消去 2 和 8 行时，同时计入 S_{k_1} 的多项式矩阵 $S_{c_{21}^1}$ 的分量(第 1 行除外)也消掉了。所以，从 S_{k_1} 中消去 2 和 8 行达不到减少乘法/加法运算次数的效果。另一方面，从 S_{k_1} 中消去 13 和 14 行，从 S_{k_2} 中消去 5 和 6 行，确实减少了运算次数。由此得出结论，在简化模型矩阵后，必须检查运算次数，即计算时间的减少量，并依此判断引入简化模型是否相宜。下面对节肢机器人精确模型和近似模型的计算量加以比较。

表 5.20 节肢机器人结构矩阵 S_{k_i} 的简化方法

i	最大系数 V_i^{max}	5% V_i^{max}	消去行
-----	------------------	----------------	-----

和表 5.23 中。在表中我们假设 $x_i = \cos q_i, x_{3+i} = \sin q_i, i \in 1, 2, 3$, 并且当 $k > 6$ 时, $x_k = \text{常数}$ 。

表 5.21 节肢机器人惯性矩阵基本计算图的矩阵表示

i, j	$E_{H_{ij}}$	$X_k, k > 6$	$V_{H_{ij}}$
1,1	$\begin{bmatrix} 5 & 5 & 3 & 3 & 11 \\ 6 & 6 & 8 & 9 & 12 \\ 7 & 6 & 2 & 3 & 13 \\ 0 & 0 & 2 & 10 & 0 \\ 0 & 0 & 0 & 2 & 0 \end{bmatrix}$	$X_7 = 4.57$ $X_8 = 112.98$ $X_9 = 26.80$ $X_{10} = 26.80$ $X_{11} = 2.68$ $X_{12} = 5.36$ $X_{13} = 2.68$	$\begin{bmatrix} 2 & 2 & 2 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$
2,2	$\begin{bmatrix} 3 & 8 \\ 7 & 0 \end{bmatrix}$	$X_7 = 138.62$ $X_8 = 26.80$	$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$
2,3	$\begin{bmatrix} 3 & 8 \\ 7 & 0 \end{bmatrix}$	$X_7 = 3.90$ $X_8 = 13.40$	$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$

在 4.3 节中介绍了按上述三个表所列数据建立程序码的方法。所得程序码可用来实时计算动力学模型矩阵。应当注意,这个程序码是自动得到的,即由所谓专家程序(用 4.3 节中提供的算法)生成的。就 FORTRAN 语言来说,专家程序可自动生成 SUBROUTINE, COMMON, RETURN 和 END 等类型的指令。这些程序码可用来进行编译。生成的程序码清单由图 5.8 给出。让我们再次说明,子程序的输入是内部角度和速度,这些量读入 COMMON/UG/。子程序的输出是动力学模型矩阵:

$$H(q, \dot{q}) \text{ 和 } h(q, \dot{q}, \theta)$$

式中 $h = \dot{q}^T C \dot{q} + h^G$ 。这些数据存入 COMMON/MATRIC/。

表 5.22 节肢机器人 C-矩阵基本计算图的矩阵表示

i	k, l	$E_{C_{kl}^i}$	$X_k, k > 6$	$V_{C_{kl}^i}$
1	2,1	$\begin{bmatrix} 5 & 2 & 3 & 3 & 10 \\ 2 & 5 & 7 & 8 & 14 \\ 0 & 2 & 6 & 6 & 13 \\ 0 & 0 & 6 & 3 & 11 \\ 0 & 0 & 6 & 12 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 9 & 0 \end{bmatrix}$	$\begin{aligned} X_7 &= 112.98 \\ X_8 &= 26.80 \\ X_9 &= 13.40 \\ X_{10} &= 2.68 \\ X_{11} &= 2.68 \\ X_{12} &= -13.40 \\ X_{13} &= -2.68 \\ X_{14} &= -2.68 \end{aligned}$	$\begin{bmatrix} 2 & 3 & 2 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$
1	3,1	$\begin{bmatrix} 5 & 2 & 3 & 3 & 8 \\ 2 & 5 & 6 & 7 & 12 \\ 0 & 2 & 6 & 6 & 11 \\ 0 & 0 & 3 & 3 & 9 \\ 0 & 0 & 0 & 10 & \\ 0 & 0 & 0 & 6 & \end{bmatrix}$	$\begin{aligned} X_7 &= 13.40 \\ X_8 &= 2.68 \\ X_9 &= 2.68 \\ X_{10} &= -13.40 \\ X_{11} &= -2.68 \\ X_{12} &= -2.68 \end{aligned}$	$\begin{bmatrix} 2 & 2 & 2 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$
2	3,2	[6 7]	$X_7 = -13.40$	[1 0]
2	3,3	[6 7]	$X_7 = -13.40$	[1 0]

表 5.23 节肢机器人重力向量基本计算图的矩阵表示

i	$X_k, k > 6$		
2	$\begin{bmatrix} 5 & 3 & 8 \\ 2 & 7 & 9 \\ 0 & 6 & 0 \end{bmatrix}$	$\begin{aligned} X_7 &= -1477.386 \\ X_8 &= -131.454 \\ X_9 &= -131.454 \end{aligned}$	$\begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$
3	$\begin{bmatrix} 5 & 3 & 7 \\ 6 & 2 & 8 \end{bmatrix}$	$\begin{aligned} X_7 &= -131.454 \\ X_8 &= -131.454 \end{aligned}$	$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}$

按照图 5.8 所示 PRTROP 程序, 建立节肢机器人模型所需运算次数总结在表 5.24 中。我们注意到, cos/sin 等类

型的函数可用查表法和线性插值法实现，这略微增加了乘法

SUBROUTINE ARTROP

C

COMMON/MATRIC/H(6,6),C(6,6,6),G(6)

COMMON/UG/Q(6)

C

X7 = Q(1)

X8 = Q(2)

X2 = COS(X8)

X5 = SIN(X8)

X9 = Q(3)

X3 = COS(X9)

X6 = SIN(X9)

C

P2 = X3 * (-0.13145E + 03)

P1 = X2 * (-0.13145E + 03)

Q2 = X5 * P2

Q1 = X6 * P1

G(3) = Q1 + Q2

P2 = X2 * (-0.13145E + 03)

P1 = X3 * (-0.13145E + 03)

Q2 = X6 * P2

Q1 = X5 * (P1 - 0.14774E + 04)

G(2) = Q1 + Q2

G(1) = 0.

Q1 = X6 * 0.13400E + 02

C(3,2,2) = Q1

$$C(3,2,1) = 0.$$

$$P4 = X2 * (-0.26800E + 01)$$

$$P3 = X3 * (-0.26800E + 01)$$

$$P2 = X2 * 0.26800E + 01$$

$$P1 = X3 * 0.26800E + 01$$

$$Q4 = X2 * P4$$

$$Q3 = X2 * (P3 - 0.13400E + 02)$$

$$Q2 = X6 * P2$$

$$Q1 = X5 * (P1 + 0.13400E + 02)$$

$$P3 = X3 * Q4$$

$$P2 = X3 * Q3$$

$$P1 = X6 * (Q1 + Q2)$$

$$Q2 = X6 * P3$$

$$Q1 = X5 * (P1 + P2)$$

$$C(3,1,1) = Q1 + Q2$$

$$Q1 = X6 * (-0.13400E + 02)$$

$$C(2,3,3) = Q1$$

$$P4 = X3 * (-0.26800E + 01)$$

$$P3 = X3 * 0.26800E + 01$$

$$P2 = X6 * 0.26800E + 01$$

$$P1 = X3 * (-0.26800E + 01)$$

$$Q4 = X2 * (P4 - 0.13400E + 02)$$

$$Q3 = X5 * (P3 + 0.13400E + 02)$$

$$Q2 = X6 * P2$$

$$Q1 = X3 * (P1 - 0.26800E + 02)$$

$$P3 = X2 * Q4$$

$$P2 = X6 * Q3$$

$$P1 = X2 * (Q1 + Q2 - 0.11298E + 03)$$

$$Q2 = X6 * P3$$

$$Q1 = X5 * (P1 + P2)$$

$$C(2,1,1) = Q1 + Q2$$

$$C(1,3,3) = 0.$$

$$C(1,3,2) = 0.$$

$$C(1,2,2) = 0.$$

$$H(3,3) = 0.39000E + 01$$

$$Q1 = X3 * 0.13400E + 02$$

$$H(3,2) = Q1 + 0.39000E + 01$$

$$H(3,1) = 0.$$

$$Q1 = X3 * 0.26800E + 02$$

$$H(2,2) = Q1 + 0.13862E + 03$$

$$H(2,1) = 0.$$

$$P3 = X2 * 0.26800E + 01$$

$$P2 = X3 * 0.53600E + 01$$

$$P1 = X3 * 0.26800E + 01$$

$$Q3 = X2 * P3$$

$$Q2 = X2 * (P2 + 0.26800E + 02)$$

$$Q1 = X3 * (P1 + 0.26800E + 02)$$

$$P3 = X6 * Q3$$

$$P2 = X6 * Q2$$

$$P1 = X5 * (Q1 + 0.11298E + 03)$$

$$Q2 = X6 * P3$$

$$Q1 = X5 * (P1 + P2)$$

$$H(1,1) = Q1 + Q2 + 0.45700E + 01$$

C

RETURN

END

图 5.8 由专家程序的输出获得的,计算节肢机器人动力学模型矩阵的程序

表 5.24 建立节肢机器人模型矩阵所需的数值运算次数

cos/sin	浮点乘法	浮点加法/减法
4	49	23

和加法运算的次数。这种算法的计算精确度通常可达 2—4 位有效数字,一般来说这是满足要求的。

在上述情况下,利用 PDP 11/70 型计算机时,建立模型的时间为 $t = 500\mu\text{s}$,而在 SIN/COS 函数具有 6 位有效数字情况下,计算时间约为 1ms。

在生成近似模型程序码时(见表 5.20),约减少 10 次乘法运算,这可节省

$$\frac{-\Delta n_{\text{mul}}}{n_{\text{mul}}} = 17\%$$

的时间。如果机器人有更多的自由度,可望节省更多的运算次数。

5.4 拟人机器人

让我们考虑拟人结构机器人,如图 5.9 所示。

拟人机器人的模型如图 5.10 所示。 C_1, C_2 和 C_3 表示连杆, Z_1, Z_2 和 Z_3 表示关节联接。和节肢机器人一样,本例



图 5.9 拟人机器人

机器人也具有三个旋转自由度。节肢机器人的两个轴 e_2 和 e_3 是平行的, 与其不同, 拟人机器人的两个轴 e_2 和 e_3 是正交的。这就是拟人机器人的数学模型比前述两例更复杂的原因。

连杆及其全部几何参数均在图 5.11 中分别标明。表 5.25

表 5.25 拟人机器人的运动学参数

i	$r_{i,i}[m]$	$r_{i,i+1}[m]$	e_i	ξ_i
1	(0.,0.,0.50)	(0.14,0.,0.)	(0.,0.,1.)	0
2	(0.175,0.,0.)	(-0.175,0.,0.)	(0.,1.,0.)	0
3	(0.230,0.,0.)	(-0.130,0.,0.)	(0.,0.,1.)	0

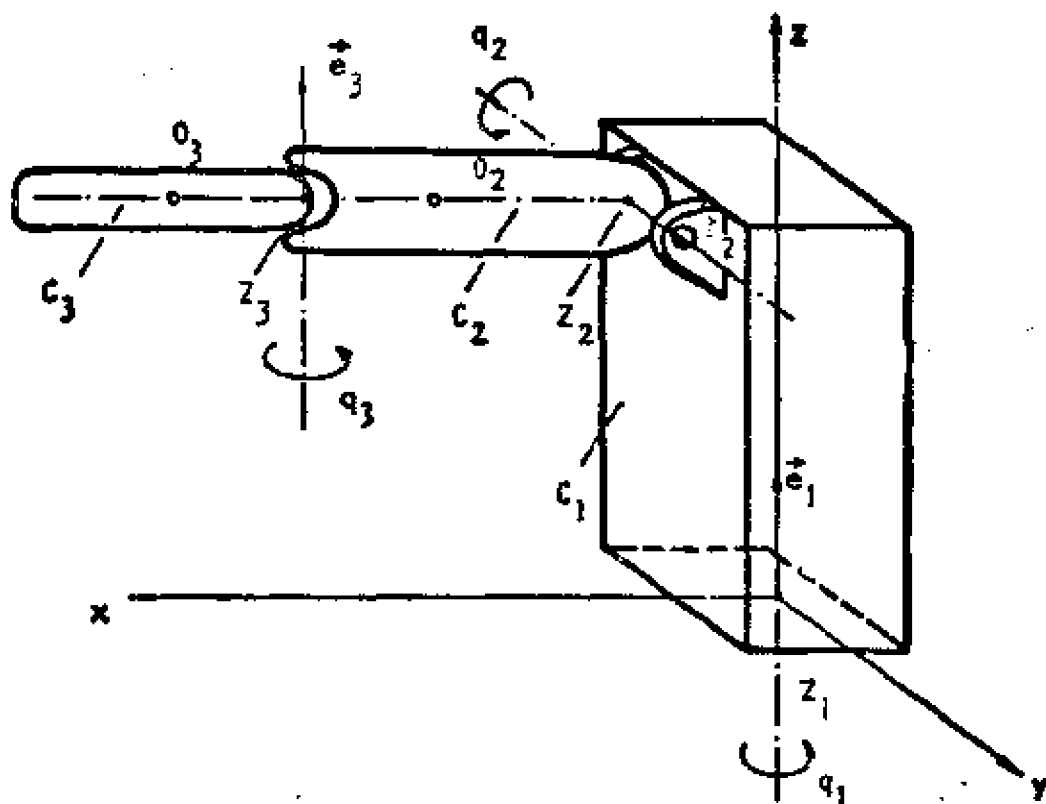


图 5.10 拟人操作手的模型

表 5.26 拟人机器人的动力学参数

i	m_i [kg]	J_{i1} [kg m ²]	J_{i2} [kg m ²]	J_{i3} [kg m ²]
1	—	—	—	0.16
2	2.450	0.00377	0.03530	0.03530
3	0.723	0.00031	0.00488	0.00488

中给出了连杆运动学参数的数据 $\tilde{R}_i = \{\tilde{r}_{ii}, \tilde{r}_{i,i+1}\}$ 和 $\tilde{B}_i = \{\tilde{\theta}_i\}, i \in N = \{1, 2, 3\}$.

动力学参数列于表 5.26 中, 表中未列参数对构造拟人机器人的动力学模型是无关紧要的。

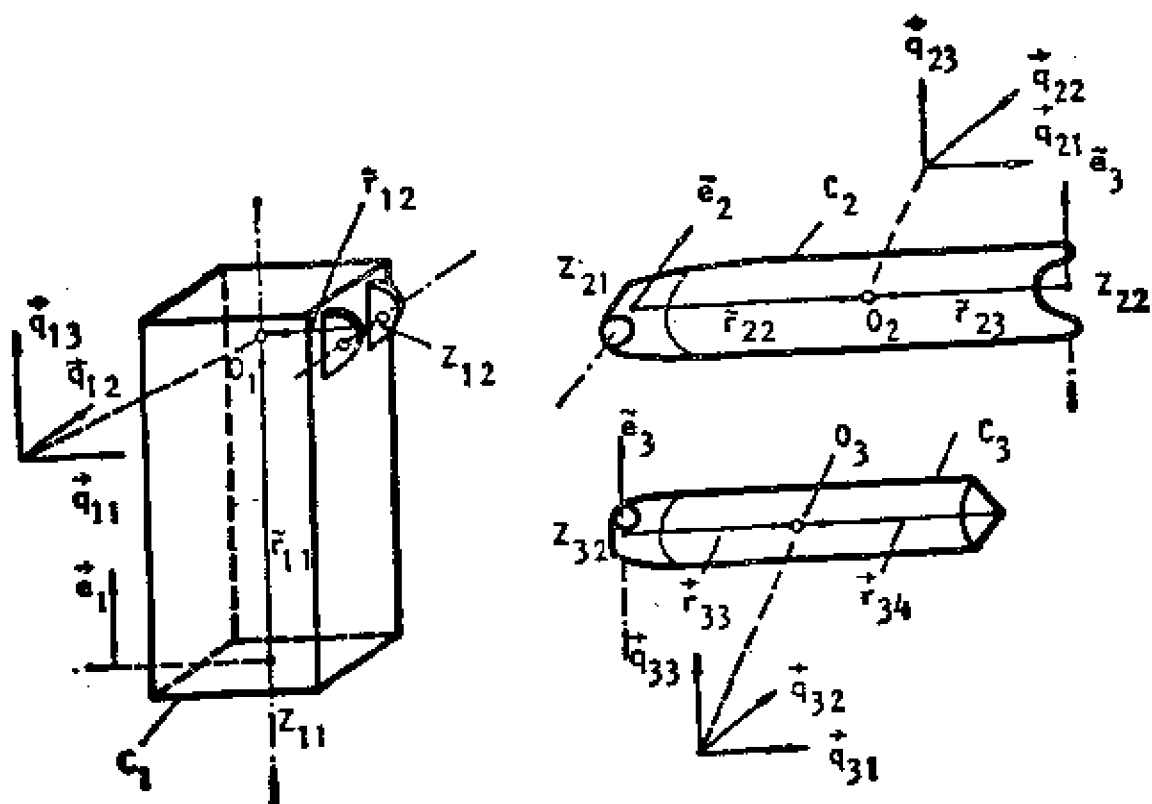


图 5.11 拟人操作手连杆的运动学参数

拟人机器人非线性符号模型的建立

建立非线性解析模型方法的第一步是确定机构的基始位置(参阅 2.3 节和图 2.7).这一步的输出是得到由局部坐标系到参考坐标系的变换矩阵:

$$Q_i^0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, i = 1, 2, 3$$

$$r_{ii}^0 = \tilde{r}_{ii}, r_{i,i+1}^0 = \tilde{r}_{i,i+1}, i = 1, 2, 3$$

和

$$e_{i+1}^0 = \tilde{e}_{i+1}, i = 1, 2$$

算法的第二步是确定机构各连杆的位置,它是关节坐标

表 5.27 拟人机器人局部坐标系单位向量对应的多项式矩阵

i	j	S_{ij}
1	1	$\left(\begin{matrix} [1. 0. 0.] \\ [0. 1. 0.] \end{matrix} , \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \right)$
	2	$\left(\begin{matrix} [0. 1. 0.] \\ [-1. 0. 0.] \end{matrix} , \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \right)$
	3	$([0.0.1.], [0 0 0 0 0 0])$
2	1	$\left(\begin{matrix} [1. 0. 0.] \\ [0. 1. 0.] \\ [0. 0. -1.] \end{matrix} , \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \right)$
	2	$\left(\begin{matrix} [0. 1. 0.] \\ [-1. 0. 0.] \end{matrix} , \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \right)$
	3	$\left(\begin{matrix} [0. 0. 1.] \\ [1. 0. 0.] \\ [0. 1. 0.] \end{matrix} , \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \right)$
3	1	$\left(\begin{matrix} [1. 0. 0.] \\ [0. 1. 0.] \\ [0. 0. -1.] \\ [0. 1. 0.] \\ [-1. 0. 0.] \end{matrix} , \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \right)$
	2	$\left(\begin{matrix} [0. 1. 0.] \\ [-1. 0. 0.] \\ [-1. 0. 0.] \\ [0. 0. 1.] \\ [0. -1. 0.] \end{matrix} , \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \right)$
	3	$\left(\begin{matrix} [0. 0. 1.] \\ [1. 0. 0.] \\ [0. 1. 0.] \end{matrix} , \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \right)$

表 5.28 拟人机器人的多项式矩阵 $S_{e_i}, S_{r_{ii}}, S_{r_{i,i+1}}$

i	S_{e_i}
1	$([0.0.1.], [0.0.0.0.0.0.])$
2	$\left(\begin{bmatrix} 0. & 1. & 0. \\ -1. & 0. & 0. \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \right)$
3	$\left(\begin{bmatrix} 0. & 0. & 1. \\ 1. & 0. & 0. \\ 0. & 1. & 0. \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \right)$
i	$S_{r_{ii}}$
1	$([0.0.0.5], [0.0.0.0.0.0.])$
2	$\left(\begin{bmatrix} 0.175 & 0. & 0. \\ 0. & 0.175 & 0. \\ 0. & 0. & -0.175 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \right)$
3	$\left(\begin{bmatrix} 0.23 & 0. & 0. \\ 0. & 0.23 & 0. \\ 0. & 0. & -0.23 \\ 0. & 0.23 & 0. \\ -0.23 & 0. & 0. \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \right)$
i	$S_{r_{i,i+1}}$
1	$\left(\begin{bmatrix} -0.14 & 0. & 0. \\ 0. & -0.14 & 0. \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \right)$
2	$\left(\begin{bmatrix} -0.175 & 0. & 0. \\ 0. & -0.175 & 0. \\ 0. & 0. & 0.175 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \right)$

q_1, q_2 和 q_3 的函数。这一步的输出是多项式矩阵 (参阅 2.5

表 5.29 惯性矩阵各元素的多项式矩阵

i, j	$S_{H,ij}$
1,1	$\left(\begin{array}{c} 0.22619 \\ 0.19520 \\ 0.19090 \\ 0.04275 \\ 0.04275 \\ 0.04656 \\ 0.11640 \end{array} \right), \left(\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 2 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 2 & 1 \end{array} \right)$
1,2	$\left(\begin{array}{c} 0.04275 \\ 0.05820 \end{array} \right), \left(\begin{array}{ccc} 0 & 0 & 1 \\ 0 & 0 & 0 \end{array} \right)$
1,3	$\left(\begin{array}{c} 0.04305 \\ 0.02328 \\ 0.05820 \end{array} \right), \left(\begin{array}{ccc} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{array} \right)$
2,2	$\left(\begin{array}{c} 0.19920 \\ 0.04275 \\ 0.11640 \end{array} \right), \left(\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 2 \\ 0 & 0 & 1 \end{array} \right)$
2,3	$([0.00000], [0 \ 0 \ 0 \ 0 \ 0 \ 0])$
3,3	$([0.04305], [0 \ 0 \ 0 \ 0 \ 0 \ 0])$

节):

$$S_{qij}, S_{r,ij}, S_{r_{i,i+1}}, S_{e,i}, i \in N, j = 1, 2, 3$$

通过计算机辅助生成这些矩阵的数据由表 5.27 和表 5.28 给出。指数矩阵有 6 列，因为第 7，第 8 和第 9 列都是零向量，这是由于没有棱柱形自由度的结果。

算法的下一步是建立惯性矩阵各元素的多项式矩阵，表 5.29 中列出了结果。因为惯性矩阵的对称性，所以表中只给

表 5.30 C-矩阵非零元素的多项式矩阵

i	k, l	$S_{C_{kl}^i}$
1	1,1	$\begin{pmatrix} -0.09545 \\ -0.19705 \\ -0.11640 \\ -0.02330 \\ -0.04290 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 2 & 0 & 1 & 0 \end{pmatrix}$
1	3,1	$\begin{pmatrix} 0.03825 \\ -0.02330 \\ -0.05820 \\ -0.03825 \\ 0.00465 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 2 & 0 & 0 & 0 & 1 \\ 0 & 2 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 2 & 1 \end{pmatrix}$
1	2,2	$\begin{pmatrix} 0.05820 \\ 0.04300 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$
1	3,2	$\begin{pmatrix} -0.04050 \\ -0.00255 \\ 0.00240 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 2 & 0 & 1 & 0 \end{pmatrix}$
1	3,3	$\begin{pmatrix} -0.02330 \\ -0.05820 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$
2	3,1	$\begin{pmatrix} 0.05820 \\ 0.04050 \\ 0.00255 \\ -0.00240 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 \end{pmatrix}$
2	3,2	$\begin{pmatrix} -0.05820 \\ -0.0428 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$

出了上三角形矩阵的数据。应当注意，第二和第三自由度是惯性解耦的。由于 $H_{33} = 0.04305 = \text{const.}$ ，可以得到，第三个关节承受的是常量惯性负荷。

表 5.31 惯性矩阵非零元素偏导数的多项式矩阵

i, j	r	$S_{H_{ij}^{(r)}}$
1,1	2	$\begin{pmatrix} -0.3904 \\ -0.19090 \\ -0.08550 \\ -0.04656 \\ -0.23280 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 2 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \end{pmatrix}$
1,1	3	$\begin{pmatrix} -0.08550 \\ 0.08550 \\ -0.04656 \\ -0.11640 \end{pmatrix}, \begin{pmatrix} 0 & 2 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 2 & 0 & 0 & 0 & 1 \end{pmatrix}$
1,2	2	$\begin{pmatrix} 0.04275 \\ 0.05820 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$
1,2	3	$\begin{pmatrix} 0.04275 \\ -0.08550 \\ 0.05820 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$
1,3	2	$\begin{pmatrix} -0.04305 \\ -0.05820 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$
1,3	3	$\begin{pmatrix} -0.02328 \\ -0.05820 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$
2,2	3	$\begin{pmatrix} -0.08550 \\ -0.11640 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$

算法的再下一步是生成 C -矩阵各元素的多项式矩阵(表 5.30)。可以看出, $C_{11}^1, C_{21}^2, C_{22}^2, C_{23}^2, C_{31}^3, C_{32}^3$ 和 C_{33}^3 均为零元素。表 5.30 中没有给出的其余元素可直接由 C -矩阵的对称性和反对称性得到(参阅 2.4 节)。

算法的最后一步是计算重力向量的多项式矩阵 $S_{h_i^0}, i \in$

表 5.32 C-矩阵各元素偏导数的多项式矩阵

i	k,l	s	$S_{C_{ij}^{kl}}$
1	2,1	2	$\begin{pmatrix} -0.09545 \\ -0.19705 \\ 0.39410 \\ -0.11640 \\ 0.23280 \\ -0.02330 \\ -0.04290 \\ 0.08580 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 2 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 2 & 0 \end{pmatrix}$
1	2,1	3	$\begin{pmatrix} 0.11640 \\ 0.02330 \\ 0.08580 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \end{pmatrix}$
1	3,1	2	$\begin{pmatrix} 0.02330 \\ 0.11640 \\ 0.08580 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \end{pmatrix}$
1	3,1	3	$\begin{pmatrix} 0.03825 \\ -0.07650 \\ -0.02330 \\ -0.05820 \\ -0.03825 \\ 0.07650 \\ 0.00465 \\ -0.00930 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2 \end{pmatrix}$
1	2,2	2	$\begin{pmatrix} -0.05820 \\ -0.04300 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$
1	2,2	3	$\begin{pmatrix} 0.05820 \\ 0.04300 \\ -0.08600 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 2 \end{pmatrix}$

表 5.32 (续)

i	k, l	s	$S_{C_{ij}^{(k)}}$
1	3,2	2	$\left(\begin{bmatrix} -0.04050 \\ 0.00255 \\ 0.00240 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 0 & 0 & 0 \end{bmatrix} \right)$
1	3,2	3	$([-0.08580], [0 \ 0 \ 1 \ 0 \ 1 \ 1])$
1	3,3	2	$([0.05820], [0 \ 0 \ 0 \ 0 \ 1 \ 1])$
1	3,3	3	$\left(\begin{bmatrix} -0.02330 \\ -0.05820 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \right)$
2	3,1	2	$\left(\begin{bmatrix} 0.05820 \\ 0.04050 \\ 0.00255 \\ -0.00240 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 2 \end{bmatrix} \right)$
2	3,1	3	$\left(\begin{bmatrix} -0.05820 \\ -0.08580 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \right)$
2	3,2	2	$([0.00000], [0 \ 0 \ 0 \ 0 \ 0 \ 0])$
2	3,2	3	$\left(\begin{bmatrix} -0.05820 \\ -0.04820 \\ 0.09640 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix} \right)$

N , 得到

$$S_{A_i^G} = 0$$

$$S_{A_7^G} = \left(\begin{bmatrix} -6.68850 \\ -1.63130 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \right)$$

$$S_{A_8^G} = ([1.63130], [0 \ 0 \ 0 \ 0 \ 1 \ 1])$$

这样, 我们就得到了拟人机器人动力学模型矩阵所有元

素的多项式矩阵。

拟人机器人线性化模型的建立

正如前例所述,构造线性化模型需要确定多项式矩阵:

$$S_{H_i^{(s)}}, S_{C_{ij}^{(s)}} \text{ 和 } S_{h_j^{(s)}}, i, j, k, s \in N$$

其中符号“(s)”表示对关节坐标 q_i 的偏导数。应用 3.6 节中所述算法可获得表 5.31, 表 5.32 和表 5.33 所列结果。三个表中未列其余元素的多项式矩阵, 或直接用对称性和反对称性得到, 或为零元素。这样, 构造线性化模型矩阵的工作就完成了。

表 5.33 重力向量各元素偏导数的多项式矩阵

i	j	$h_j^{(s)}$
2	2	$\left(\begin{bmatrix} 6.68850 \\ 1.63130 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \right)$
2	3	$([1.63130], [0 \ 1 \ 0 \ 0 \ 0 \ 1])$
3	2	$([1.63130], [0 \ 1 \ 0 \ 0 \ 0 \ 1])$
3	3	$([1.63130], [0 \ 0 \ 1 \ 0 \ 1 \ 0])$

拟人机器人解析模型的优化计算和程序码的生成

和前两例相对照, 在这里, 我们同样考虑生成拟人机器人线性化模型矩阵的问题, 在表 5.31—5.33 中给出了线性化模型的多项式矩阵。按照第四章介绍的算法, 对这些多项式矩阵需要生成对应的优化计算序列, 也就是得到最小计算图的

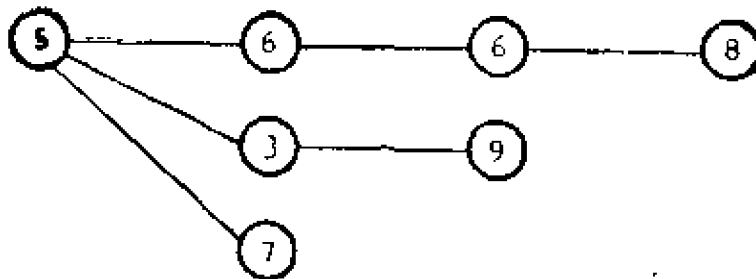
表 5.34 惯性矩阵偏导数基本计算图的矩阵表示

i, j	s	$E_{H_{ij}^{(s)}}$	$X_k, k \geq 6$	$V_{H_{ij}^{(s)}}$
1,1	2	$\begin{bmatrix} 5 & 2 & 3 & 3 & 9 \\ 0 & 3 & 7 & 11 & 0 \\ 0 & 8 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$	$X_7 = -0.3904$ $X_8 = -0.1909$ $X_9 = -0.08550$ $X_{10} = -0.04656$ $X_{11} = -0.23280$	$\begin{bmatrix} 3 & 2 & 2 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$
1,1	3	$\begin{bmatrix} 6 & 2 & 2 & 3 & 7 \\ 0 & 3 & 9 & 10 & 0 \\ 0 & 0 & 8 & 0 & 0 \end{bmatrix}$	$X_7 = -0.0855$ $X_8 = -0.0855$ $X_9 = -0.04656$ $X_{10} = -0.11640$	$\begin{bmatrix} 2 & 2 & 2 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$
1,2	2	$\begin{bmatrix} 2 & 6 & 3 & 7 \\ 0 & 0 & 8 & 0 \end{bmatrix}$	$X_7 = 0.04275$ $X_8 = 0.05820$	$\begin{bmatrix} 1 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$
1,2	3	$\begin{bmatrix} 5 & 6 & 6 & 8 \\ 0 & 3 & 9 & 0 \\ 0 & 7 & 0 & 0 \end{bmatrix}$	$X_7 = 0.04275$ $X_8 = -0.08550$ $X_9 = 0.05820$	$\begin{bmatrix} 3 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$
1,3	2	$\begin{bmatrix} 5 & 3 & 8 \\ 0 & 7 & 0 \end{bmatrix}$	$X_7 = -8.04305$ $X_8 = -0.05820$	$\begin{bmatrix} 2 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
1,3	3	$\begin{bmatrix} 6 & 2 & 8 \\ 0 & 7 & 0 \end{bmatrix}$	$X_7 = -0.02328$ $X_8 = -0.05820$	$\begin{bmatrix} 2 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
2,2	3	$\begin{bmatrix} 6 & 3 & 7 \\ 0 & 8 & 0 \end{bmatrix}$	$X_7 = -0.08550$ $X_8 = -0.11640$	$\begin{bmatrix} 2 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

矩阵 (E 和 V 矩阵) 表示。对惯性矩阵的偏导数, 表 5.34 中给出了计算结果。基于这些数据, 容易画出优化计算图。例如, 对项

$$H_{12}^{(3)} = \frac{\partial H_{12}(q, \theta)}{\partial q_3}$$

得到计算图



这个图可解释为多项式

$$H_{12}^{(3)} = x_5(x_6^2 X_1 + x_3 X_9 + X_7)$$

即三角式

$$H_{12}^{(3)} = \sin q_2((\sin^2 q_3)X_8 + (\cos q_3)X_9 + X_7)$$

式中 X_7 , X_8 和 X_9 是表 5.34 中给出的常数。

C -矩阵和重力向量的偏导数对应的 E 和 V 矩阵分别由表 5.35 和表 5.36 给出。在前两例中，生成程序码是针对机构的非线性动力学模型的，这里要证明，用相同的方法对线性化模型也能生成程序码。应当注意，程序码也是由所谓专家程序自动得到的(参阅 4.3 节)。专家程序用所需程序语言来产生输入/输出指令。例如，利用 FORTRAN 语言就能产生 SUBROUTINE, COMMON 等指令。图 5.12 给出了程序清单。子程序的输入是内部角度 (q_1, \dots, q_n) ，把它存入 COMMON-block/UG/中。子程序的输出是线性化模型矩阵：

$$\frac{\partial H(q, \theta)}{\partial q}, \frac{\partial C^i(q, \theta)}{\partial q} \text{ 和 } \frac{\partial h^{\alpha i}(q, \theta)}{\partial q}, i \in N$$

这些矩阵的元素通过下列方式

$$H_{ij}^{(n)} \rightarrow HL(i, j, s)$$

$$C_{ij}^{(n)} \rightarrow CL(i, k, l, s)$$

$$h_i^{\alpha(n)} \rightarrow HGL(i, s)$$

存入 COMMON-block/LMODL/。程序码既未设置零元素，

也未设置可直接由模型矩阵的对称性和反对称性得到的那些元素,表 5.37 中列出了按照图 5.12 给出的 LANTR 程序来建立拟人机器人线性化模型所需的运算次数。如果用查表法计算正弦和余弦函数,在 PDP11/70 型计算机上建立模型需要的时间则为 $t = 700 \mu s$, 如果正弦和余弦函数从程序库(约 6 位有效数字)调用,建立偏导数矩阵所需的时间则约为 1.3ms。

```

SUBROUTINE LANTR
COMMON/LMODL/HL(6,6,6),CL(6,6,6,6),
      HGL(6,6)
COMMON/UG/Q(6)
X8 = Q(2)
X2 = COS(X8)
X5 = SIN(X8)
X9 = Q(3)
X3 = COS(X9)
X6 = SIN(X9)

P1 = X5 * 0.16313E + 01
HGL(3,3) = X3 * P1
P1 = X6 * 0.16313E + 01
HGL(3,2) = X2 * P1
P1 = X6 * 0.16313E + 01
HGL(2,3) = X2 * P1
P1 = X3 * 0.16313E + 01
HGL(2,2) = X5 * (P1 + 0.66885E + 01)
P1 = X6 * 0.96400E - 01
Q1 = X6 * P1

```

$Q2 = X3 * (-0.58200E - 01)$
 $CL(2,3,2,3) = Q1 + Q2 - 0.48200E - 01$
 $P1 = X3 * (-0.85800E - 01)$
 $Q1 = X6 * (P1 - 0.58200E - 01)$
 $CL(2,3,1,3) = X5 * Q1$
 $P1 = X6 * (-0.24000E - 02)$
 $P2 = X3 * 0.40500E - 01$
 $Q1 = X6 * P1$
 $Q2 = X3 * (P2 + 0.58200E - 01)$
 $CL(2,3,1,2) = X2 * (Q1 + Q2 + 0.25500E - 02)$
 $P1 = X2 * 0.58200E - 01$
 $CL(1,3,3,3) = X3 * (P1 - 0.23300E - 01)$
 $P1 = X6 * 0.58200E - 01$
 $CL(1,3,3,2) = X5 * P1$
 $P1 = X6 * (-0.85800E - 01)$
 $Q1 = X5 * P1$
 $CL(1,3,2,3) = X3 * Q1$
 $P1 = X3 * 0.24000E - 02$
 $P2 = X6 * (-0.40500E - 02)$
 $Q1 = X3 * P1$
 $Q2 = X6 * P2$
 $CL(1,3,2,2) = X2 * (Q1 + Q2 - 0.25500E - 02)$
 $P1 = X6 * (-0.84000E - 01)$
 $Q1 = X3 * 0.58200E - 01$
 $Q2 = X6 * P1$
 $CL(1,2,2,3) = X2 * (Q1 + Q2 + 0.43000E - 01)$
 $P1 = X3 * (-0.43000E - 01)$
 $Q1 = X6 * (P1 - 0.58200E - 01)$

$$CL(1,2,2,2) = X5 * Q1$$

$$P1 = X6 * (-0.93000E - 02)$$

$$P2 = X6 * 0.76500E - 01$$

$$Q1 = X6 * P1$$

$$Q2 = X3 * (-0.58200E - 01)$$

$$Q3 = X6 * P2$$

$$P1 = X6 * (-0.76500E - 01)$$

$$P2 = X5 * (Q1 + 0.46500E - 01)$$

$$P3 = X3 * (-0.23300E - 01)$$

$$P4 = X2 * (Q1 + Q2 - 0.38250E - 01)$$

$$Q1 = X6 * P1$$

$$Q2 = X5 * P2$$

$$Q3 = X2 * (P3 + P4)$$

$$CL(1,3,1,3) = Q1 + Q2 + Q3 + 0.38250E - 01$$

$$P1 = X3 * 0.85800E - 01$$

$$Q1 = X2 * (P1 + 0.11640E + 00)$$

$$P1 = X6 * (Q1 + 0.23300E - 01)$$

$$Q1 = X5 * P1$$

$$CL(1,3,1,2) = Q1$$

$$CL(1,2,1,3) = Q1$$

$$P1 = X3 * 0.85800E - 01$$

$$Q1 = X5 * (P1 + 0.23280E + 00)$$

$$P1 = X5 * 0.39410E + 00$$

$$P2 = X3 * (-0.42900E - 01)$$

$$P3 = X2 * (-0.23300E - 01)$$

$$P4 = X5 * Q1$$

$$Q1 = X5 * P1$$

$$Q2 = X2 * (-0.95450E - 01)$$

```

Q3 = X3 * (P2 + P3 + P4)
CL(1,2,1,2) = Q1 + Q2 + Q3 + -0.19705E + 00
P1 = X3 * (-0.85500E - 01)
HL(2,2,3) = X6 * (P1 - 0.11640E + 00)
P1 = X2 * (-0.58200E - 01)
HL(1,3,3) = X6 * (P1 - 0.23280E - 01)
P1 = X3 * (-0.58200E - 01)
HL(1,3,2) = X5 * (P1 - 0.43050E - 01)
P1 = X6 * (-0.85500E - 01)
Q1 = X6 * 0.58200E - 01
Q2 = X6 * P1
HL(1,2,3) = X5 * (Q1 + Q2 + 0.42750E - 01)
P1 = X3 * 0.42750E - 01
Q1 = X6 * (P1 + 0.58200E - 01)
HL(1,2,2) = X2 * Q1
P1 = X3 * (-0.85500E - 01)
Q1 = X2 * (P1 + 0.11640E + 00)
P1 = X3 * 0.85500E - 01
P2 = X2 * (Q1 - 0.46560E - 01)
HL(1,1,3) = X6 * (P1 + P2)
P1 = X3 * (-0.85500E - 01)
Q1 = X3 * (P1 - 0.23280E - 01)
P1 = X3 * (-0.46560E - 01)
P2 = X2 * (Q1 - 0.39040E + 00)
HL(1,1,2) = X5 * (P1 + P2 - 0.19090E + 00)
RETURN
END

```

图 5.12 生成拟人机器人线性化动力学模型矩阵的程序

表 5.35 C-矩阵偏导数基本计算图的矩阵表示

r	k, l	s	$E_{C_{k,l}}^{(r)}$	$X_{k, k > 6}$	$V_{C_{k,l}}^{(r)}$
1	2,1	2	$\begin{bmatrix} 3 & 5 & 5 & 3 & 14 \\ 2 & 2 & 12 & 11 & 0 \\ 5 & 3 & 13 & 0 & 0 \\ 8 & 10 & 9 & 0 & 0 \\ 0 & 7 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 \end{bmatrix}$	$\begin{aligned} X_7 &= -0.09545 \\ X_8 &= -0.19705 \\ X_9 &= 0.39410 \\ X_{10} &= -0.11640 \\ X_{11} &= 0.23280 \\ X_{12} &= -0.02330 \\ X_{13} &= -0.04290 \\ X_{14} &= 0.08580 \end{aligned}$	$\begin{bmatrix} 4 & 1 & 2 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$
1	2,1	3	$\begin{bmatrix} 5 & 6 & 2 & 3 & 9 \\ 0 & 0 & 8 & 7 & 0 \end{bmatrix}$	$\begin{aligned} X_7 &= 0.11640 \\ X_8 &= 0.02330 \\ X_9 &= 0.08580 \end{aligned}$	$\begin{bmatrix} 1 & 2 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$
1	3,1	2	$\begin{bmatrix} 5 & 6 & 2 & 3 & 9 \\ 0 & 0 & 7 & 8 & 0 \end{bmatrix}$	$\begin{aligned} X_7 &= 0.02330 \\ X_8 &= 0.11640 \\ X_9 &= 0.08580 \end{aligned}$	$\begin{bmatrix} 1 & 2 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$
1	3,1	3	$\begin{bmatrix} 2 & 2 & 6 & 6 & 12 \\ 5 & 3 & 3 & 10 & 14 \\ 6 & 5 & 11 & 6 & 0 \\ 7 & 6 & 9 & 0 & 0 \\ 0 & 0 & 6 & 0 & 0 \\ 0 & 0 & 13 & 0 & 0 \\ 0 & 0 & 8 & 0 & 0 \end{bmatrix}$	$\begin{aligned} X_7 &= 0.03825 \\ X_8 &= -0.07650 \\ X_9 &= -0.02330 \\ X_{10} &= -0.05820 \\ X_{11} &= -0.03825 \\ X_{12} &= 0.07650 \\ X_{13} &= 0.00465 \\ X_{14} &= -0.00930 \end{aligned}$	$\begin{bmatrix} 2 & 3 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$
1	2,2	2	$\begin{bmatrix} 5 & 6 & 3 & 8 \\ 0 & 0 & 7 & 0 \end{bmatrix}$	$\begin{aligned} X_7 &= -0.05820 \\ X_8 &= -0.04300 \end{aligned}$	$\begin{bmatrix} 1 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$

表 5.35 (续)

i	k, l	r	$E_{C_k^{(i)}}$	$X_k, k > 6$	$V_{C_k^{(i)}}$
1	2, 2	3	$\begin{bmatrix} 2 & 6 & 6 & 9 \\ 0 & 3 & 7 & 0 \\ 0 & 8 & 0 & 0 \end{bmatrix}$	$X_7 = 0.05820$ $X_8 = 0.04300$ $X_9 = -0.08600$	$\begin{bmatrix} 3 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$
1	3, 2	2	$\begin{bmatrix} 2 & 6 & 6 & 7 \\ 0 & 3 & 3 & 9 \\ 0 & 8 & 0 & 0 \end{bmatrix}$	$X_7 = -0.04050$ $X_8 = -0.00255$ $X_9 = 0.00240$	$\begin{bmatrix} 3 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$
1	3, 2	3	[3 5 6 7]	$X_7 = -0.08580$	[1 1 1 0]
1	3, 3	2	[5 6 7]	$X_7 = 0.05820$	[1 1 0]
1	3, 3	3	$\begin{bmatrix} 3 & 2 & 8 \\ 0 & 7 & 0 \end{bmatrix}$	$X_7 = -0.02330$ $X_8 = -0.05820$	$\begin{bmatrix} 2 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
2	3, 1	2	$\begin{bmatrix} 2 & 3 & 3 & 8 \\ 0 & 6 & 7 & 10 \\ 0 & 9 & 6 & 0 \end{bmatrix}$	$X_7 = 0.05820$ $X_8 = 0.04050$ $X_9 = 0.00255$ $X_{10} = -0.00240$	$\begin{bmatrix} 3 & 2 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
2	3, 1	3	$\begin{bmatrix} 5 & 6 & 3 & 8 \\ 0 & 0 & 7 & 0 \end{bmatrix}$	$X_7 = -0.05820$ $X_8 = -0.08580$	$\begin{bmatrix} 1 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$
2	3, 2	3	$\begin{bmatrix} 6 & 6 & 9 \\ 3 & 7 & 0 \\ 8 & 0 & 0 \end{bmatrix}$	$X_7 = -0.05820$ $X_8 = -0.04820$ $X_9 = 0.09640$	$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

表 5.36 重力向量偏导数基本计算图的矩阵表示

i	r	$E_{\Delta q^{(i)}}$	$X_k, k > 6$	$V_{\Delta q^{(i)}}$
2	2	$\begin{bmatrix} 5 & 3 & 8 \\ 0 & 7 & 0 \end{bmatrix}$	$X_7 = 6.68850$ $X_8 = 1.63130$	$\begin{bmatrix} 2 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
2	3	[2 6 7]	$X_7 = 1.63130$	[1 1 0]
3	2	[2 6 7]	$X_7 = 1.63130$	[1 1 0]
3	3	[3 5 7]	$X_7 = 1.63130$	[1 1 0]

表 5.37 计算模型矩阵偏导数所需的数值运算次数

cos/sin	浮点乘法	浮点加法/减法
4	86	41

表 5.38 动力学模型矩阵基本计算图的矩阵表录

i, j		$E_{H,ij}$	$X_k, k > 6$	$V_{H,ij}$
1,1		$\begin{bmatrix} 2 & 3 & 2 & 3 & 10 \\ 6 & 2 & 12 & 13 & 0 \\ 7 & 9 & 8 & 0 & 0 \\ 0 & 6 & 11 & 0 & 0 \end{bmatrix}$	$X_7 = 0.22619$ $X_8 = 0.19520$ $X_9 = 0.19090$ $X_{10} = 0.04275$ $X_{11} = 0.04275$ $X_{12} = 0.04656$ $X_{13} = 0.11640$	$\begin{bmatrix} 3 & 2 & 2 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$
1,2		$\begin{bmatrix} 5 & 6 & 3 & 7 \\ 0 & 0 & 8 & 0 \end{bmatrix}$	$X_7 = 0.04275$ $X_8 = 0.05820$	$\begin{bmatrix} 1 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$
1,3		$\begin{bmatrix} 3 & 2 & 9 \\ 2 & 8 & 0 \\ 0 & 7 & 0 \end{bmatrix}$	$X_7 = 0.04305$ $X_8 = 0.02328$ $X_9 = 0.05820$	$\begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
2,2		$\begin{bmatrix} 3 & 3 & 8 \\ 7 & 9 & 0 \end{bmatrix}$	$X_7 = 0.19920$ $X_8 = 0.04275$ $X_9 = 0.11640$	$\begin{bmatrix} 2 & 1 & 0 \\ 2 & 1 & 0 \end{bmatrix}$
i	k	$E_{C_{k,i}}$	$X_k, k > 6$	$V_{C_{k,i}}$
1	2,1	$\begin{bmatrix} 5 & 3 & 2 & 3 & 11 \\ 0 & 2 & 10 & 9 & 0 \\ 0 & 7 & 8 & 0 & 0 \end{bmatrix}$	$X_7 = -0.09545$ $X_8 = -0.19705$ $X_9 = -0.11640$ $X_{10} = -0.02330$ $X_{11} = -0.04290$	$\begin{bmatrix} 3 & 2 & 2 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$

表 5.38 (续)

i	k	$E_{c,i}$	$X_{k>6}$	$V_{c,i}$
1	3,1	$\begin{bmatrix} 6 & 2 & 2 & 3 & 10 \\ 0 & 5 & 8 & 9 & 11 \\ 0 & 0 & 5 & 5 & 0 \end{bmatrix}$	$X_7 = 0.03825$ $X_8 = -0.02330$ $X_9 = -0.05820$ $X_{10} = -0.03825$ $X_{11} = 0.00465$	$\begin{bmatrix} 2 & 2 & 2 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}$
1	2,2	$\begin{bmatrix} 2 & 6 & 3 & 8 \\ 0 & 0 & 7 & 0 \end{bmatrix}$	$X_7 = 0.05820$ $X_8 = 0.04300$	$\begin{bmatrix} 1 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$
1	3,2	$\begin{bmatrix} 5 & 6 & 6 & 7 \\ 0 & 3 & 3 & 9 \\ 0 & 8 & 0 & 0 \end{bmatrix}$	$X_7 = -0.04050$ $X_8 = -0.00255$ $X_9 = 0.00240$	$\begin{bmatrix} 3 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$
1	3,3	$\begin{bmatrix} 6 & 2 & 8 \\ 0 & 7 & 0 \end{bmatrix}$	$X_7 = -0.02330$ $X_8 = -0.05820$	$\begin{bmatrix} 2 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
2	3,1	$\begin{bmatrix} 5 & 3 & 3 & 8 \\ 0 & 6 & 7 & 10 \\ 0 & 9 & 6 & 0 \end{bmatrix}$	$X_7 = 0.05820$ $X_8 = 0.04050$ $X_9 = 0.00255$ $X_{10} = -0.00240$	$\begin{bmatrix} 3 & 2 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
2	3,2	$\begin{bmatrix} 6 & 3 & 8 \\ 0 & 7 & 0 \end{bmatrix}$	$X_7 = -0.0582$ $X_8 = -0.0428$	$\begin{bmatrix} 2 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
i		$E_{k\varphi}$		$V_{k\varphi}$
2		$\begin{bmatrix} 2 & 3 & 8 \\ 0 & 7 & 0 \end{bmatrix}$	$X_7 = -6.68850$ $X_8 = -1.63130$	$\begin{bmatrix} 2 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
3		$[5 \ 6 \ 7]$	$X_7 = 1.63130$	$[1 \ 1 \ 0]$

有一点需要注意。如 2.5 节中所述，构造线性化模型式 (2.5.1) 不仅需要偏导数矩阵

$$\frac{\partial H}{\partial q}, \frac{\partial C}{\partial q} \text{ 和 } \frac{\partial h_0}{\partial q}$$

而且需要非线性模型矩阵 $H(q, \theta)$ 和 $C(q, \theta)$, 这就是必须确定矩阵 H 和 C 各项基本计算图的原因。在表 5.38 中给出了计算图的矩阵表示。由这些结果容易确定建立矩阵 $H(q, \theta)$ 和 $C(q, \theta)$ 所需的乘法和加法运算次数。由此得到表 5.39 所列数据(表 5.39 在下一节给出)。从这个表可以看出, 计算非线性模型矩阵所需运算次数约为

$$n_{\text{nonlin}} = 50\% n_{\text{lin}}$$

式中 n_{lin} 是建立矩阵偏导数所需的运算次数。由此可见, 建立完全线性化模型所需运算次数是建立非线性模型所需运算次数的 3 倍。

5.5 机器人解析模型的微型计算机实现, 运算时间和存储器的要求

在这一节中, 我们从存储器设备和运算速度的角度来考察在现代微型计算机上实现机器人解析模型的问题, 所述问题不仅针对三个自由度, 而且要考虑六个自由度的操作手。为保证所得结果具有实用价值, 量测是在典型的工业机器人上进行的。在前述内容中已经考察了具有 3 个关节的圆柱型、节肢和拟人结构的机器人。现在把分析扩展到实际使用的六个自由度的典型机器人。例如, 我们要考虑这样的机器人, 它的前三个自由度是圆柱型结构, 而后三个自由度是拟人结构(图 5.13)。另外, 我们考虑一个著名的机器人“斯坦福操作手”, 如图 5.14 所示, 它经常被看作是动力学分析和机器人控制综合的一个范例^[13]。最后, 我们看一个结构上比前述各例

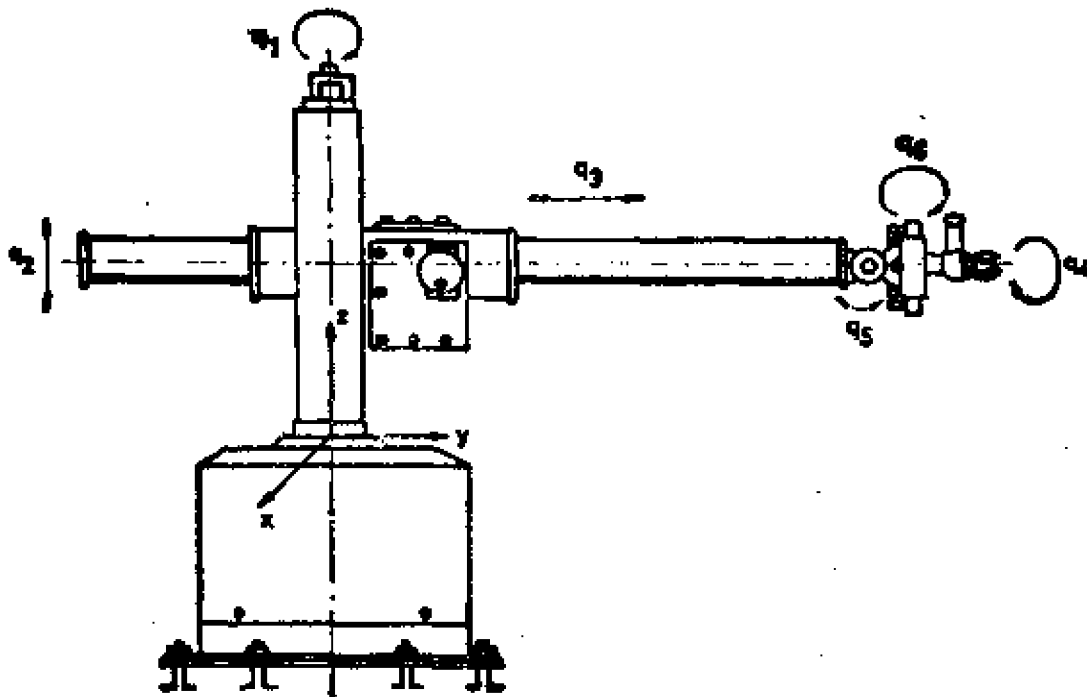


图 5.13 六个自由度的圆柱型机器人 (CL-AN)

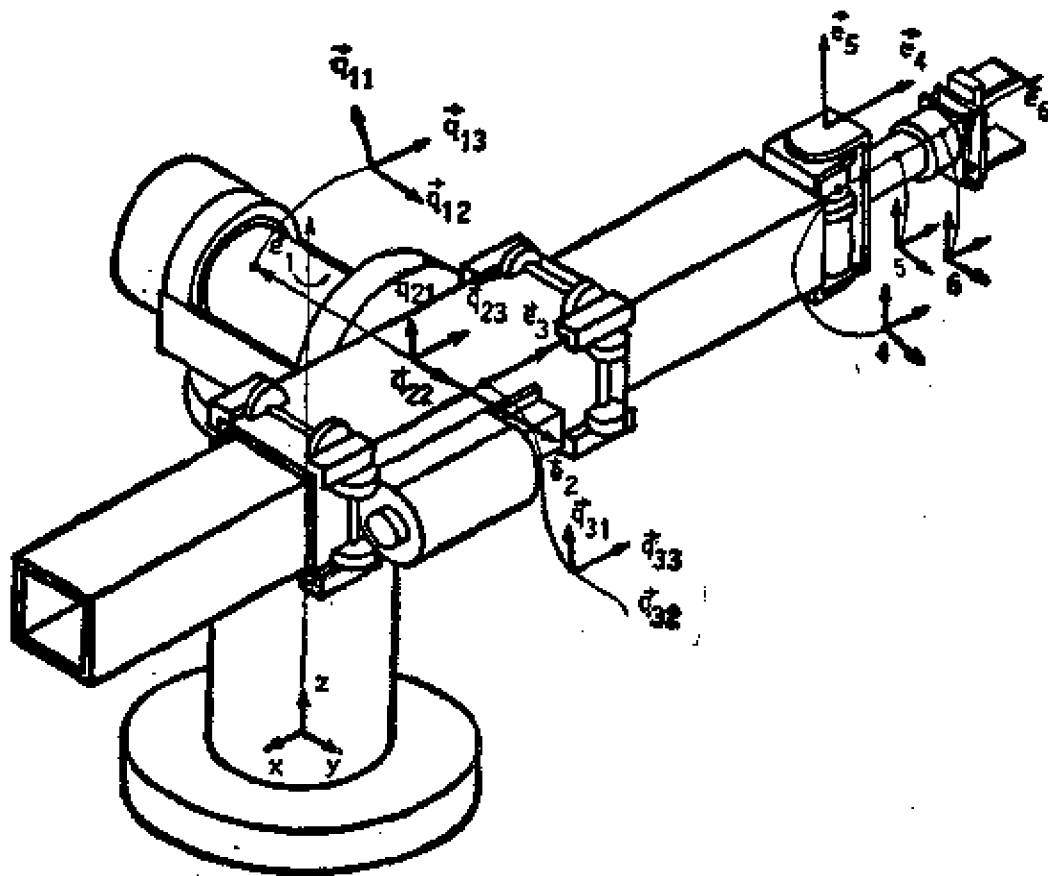


图 5.14 六个自由度的“斯坦福操作手” (ST)

更复杂的机器人——半节肢机器人，如图 5.15 所示。这种机器人适合于焊接和油漆等工作。

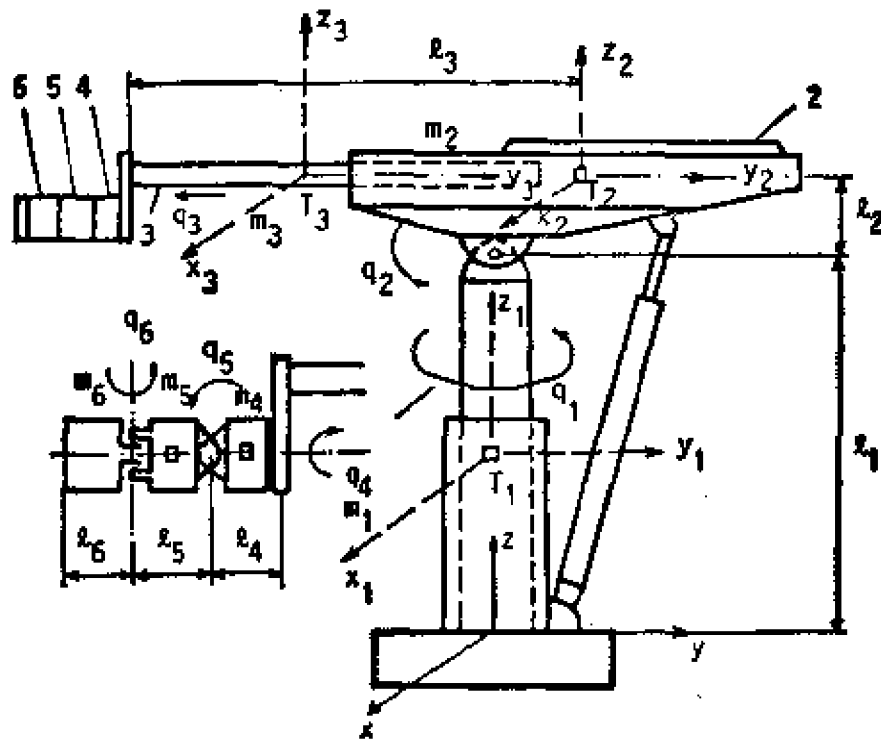


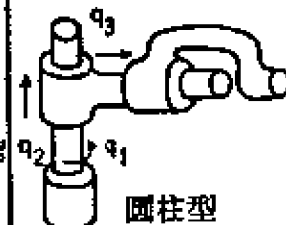
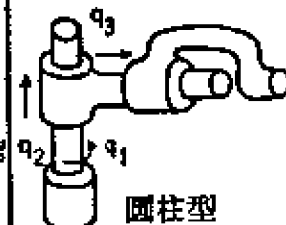
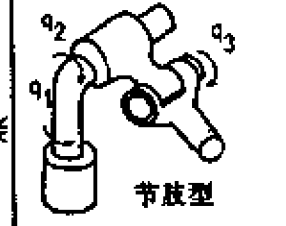

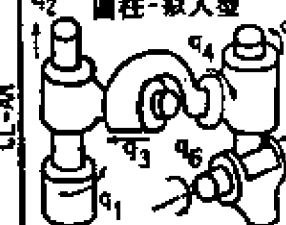
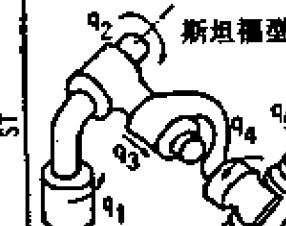
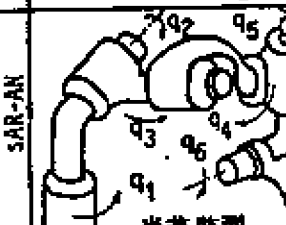
图 5.15 六个自由度的半节肢机器人 (SAR-AN)

在第四章中曾详尽描述其结构的专家程序，最初是用来生成机器人解析模型的程序码。如前所述，像其它机器人建模程序系统一样，专家程序适合在高性能计算机上建立起来，然后把所得实时运算程序配置在合适的微型计算机上。在本例中，我们在 PDP 11/70 型计算机上生成模型，然后把所得模型在微型计算机 INTEL8086（带有 8087 算术协同处理器）上，在 PDP11/03 型 16 位微型计算机上和 PDP11/70 型计算机自身上进行检验。源程序的编译和检验在微型计算机上完成。附录 5.1 中介绍了一个生成源程序的例子。这一程

序描绘了六个自由度的斯坦福操作手模型。可以看出，斯坦福操作手浮点乘法和加法/减法运算次数是三个自由度最复杂的操作手(拟人机器人)所需运算次数的 7.5 倍。

对所述的每一个操作手，我们都测试了可能与动力学模型实时实现有关的大量重要参数。浮点运算次数和计算模型矩阵所需时间都系统地列在表 5.39 中。表的第 1 列是机器人的缩写名 (CL 表示圆柱型机器人，AR 表示节肢机器人，AN 表示拟人机器人，SAR-AN 表示半节肢机器人，ST 表示斯坦福操作手)，第 2 列是机器人的运动学类型和关节坐标。从表中可以看出，三个自由度的操作手所需浮点运算次数不到 60 次，而六个自由度的操作手需要数百次运算。运算次数与运动学类型复杂程度的函数关系由图 5.16 给出。这个表还包括，在微型/小型计算机上测得的，计算各项模型矩阵所需的计算机时间。现在，我们可以得到若干有关实时实现机器人动力学模型可能性的重要结论。可以看出，所有模型都能在 PDP11/70 小型计算机上实时地建立起来。如果正弦和余弦函数用查表法计算(在表 5.39 中标以“T”的行)，那么建立一个最复杂的，三个自由度的机器人动力学模型所需时间不超过 0.6ms。如果三角函数通过调用库程序计算，那么计算时间也不超过 1.2ms。应当说明，通常没有必要调用库程序(其计算精度可达 6 位以上有效数字)，因为机器人的运动学和动力学参数常常只需 2—3 位有效数字就够了。建立六个自由度操作手的模型所需时间范围是 3—14ms。如果把 20—30ms 作为实时限制，对应 30—50Hz 采样频率，那么可以看出，六个自由度的机器人动力学模型可以在线计算。然而，微型计算机系统达到的时间在实际应用中更为重要。在表 5.39 中，考察了以 INTEL8086 型微处理机(带一协同处

表 5.39

	 圆柱型	\cos/\sin	乘法运算 次数	加法运算 次数	L/T ¹⁾	t(ms)		
						PDP11/70	INTEL 86	PDP11/03
CL	 圆柱型	0	3	3	T	0.2	0.9	1.7
					L	0.2	0.9	1.7
AR	 节肢型	4	49	22	T	0.5	4.4	11.4
					L	0.97	6.4	52.0
AN	 拟人型	4	55	31	T	0.5	5.3	13.6
					L	1.2	7.4	54.0
CL-AN	 圆柱-拟人型	6	193	80	T	2.8	16.9	41.5
					L	3.2	19.9	187.5
ST	 斯坦福型	6	372	167	T	5.0	30.2	76.5
					L	5.4	33.3	145.0
SAR-AN	 半节肢型	8	895	389	T	13.9	83.7	209.0
					L	14.4	88.0	297.0

1) T 意指三角函数值用查表法和线性插值法计算, L 意指三角函数值从函数库调用。

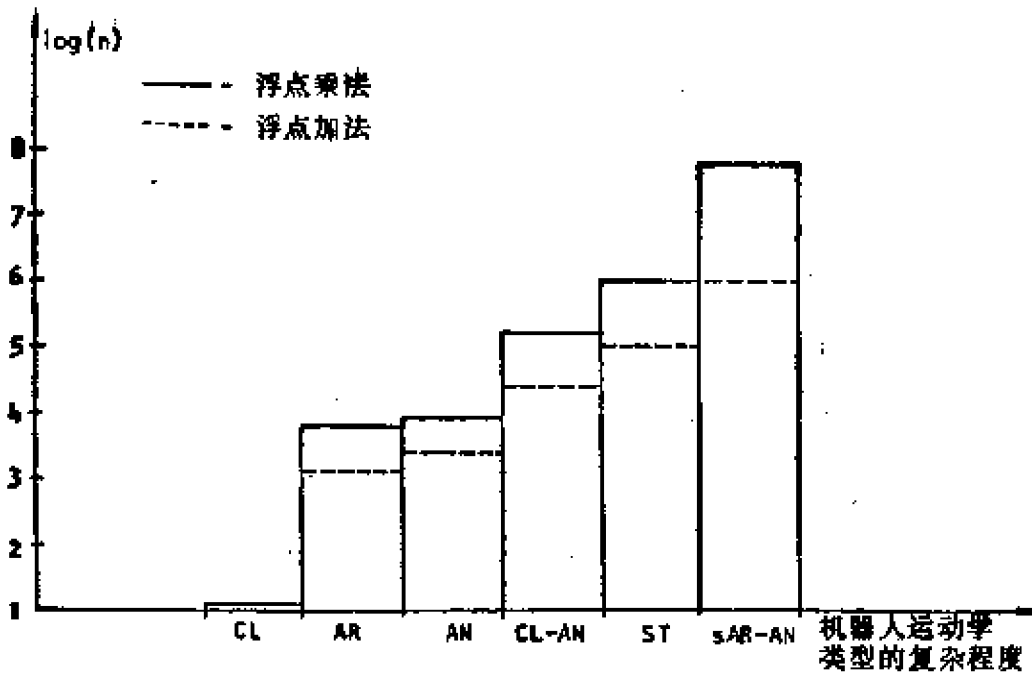


图 5.16 浮点运算次数与操作手运动学类型复杂程度的函数关系

(\log 为自然对数, n 为表 5.20 中所列的运算次数)

并行处理器或更快的微型处理机（诸如 INTEL286 等类型）也是可行的。

图 5.17 中给出了机器人动力学模型计算时间与运动学类型复杂程度的函数关系。图中标出了实时限制 20—30ms。

除程序的运行速度外，我们还要考虑机器人动力学模型的实现对存储器的要求。由于这些数据难以预测，这里提供的存储器规模都是从实际运行的计算机系统上量测出来的，量测量包括储存源程序和储存用机器码表示的目标程序所需存储器区的大小。这些数据用 8 位 1 字节的单元表示，在表 5.40 中列出了所述机器人的数据。可以看出，储存 3 个关节的机器人模型的源程序，使用机器人控制的典型微型计算机（如 INTEL8086）约需要 2000 个字节，对 6 个关节的机器人

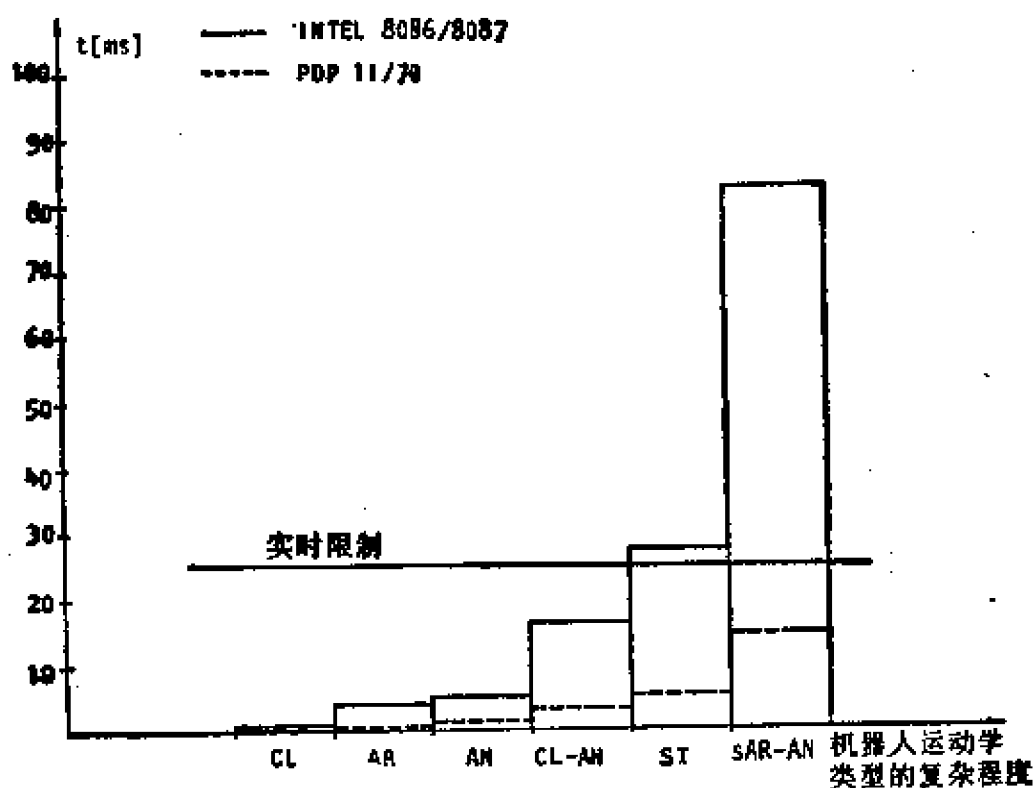


图 5.17 机器人动力学模型矩阵计算时间与其运动学类型复杂程度的函数关系

表 5.40 安置计算机器人动力学模型矩阵的程序所需的存储器

机器人 程序	存储器(字节) INTEL8086/8087					
	CL	AR	AN	CL-AN	ST	SAR-AN
源程序、	1943	2184	2385	8041	11644	30493
计算机码	1219	3809	5190	13509	21925	58210

来说, 约需要 8000—30000 个字节。对不超过三个自由度的操作手, 安置机器人模型的目标程序所需的存储器空间最多可用到 5K 个字节, 而对六个自由度的操作手, 则需要 13—58K 个字节。在图 5.18 中给出了存储器空间的需要量与操

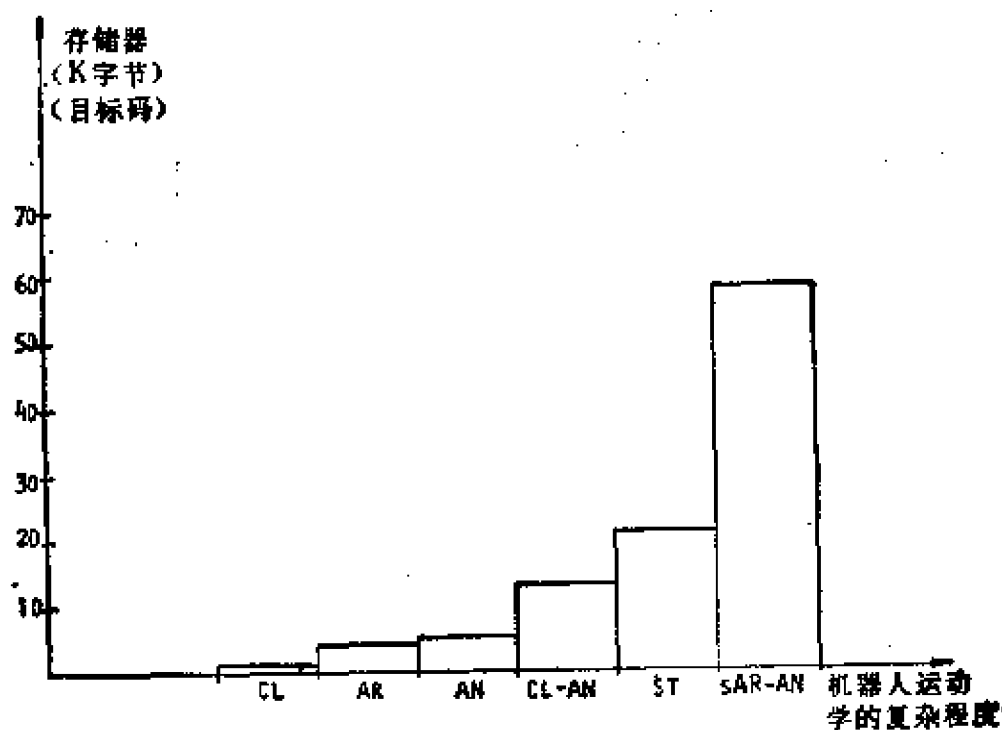


图 5.18 动力学模型计算程序存储器的需要量与机器人运动学复杂程度的函数关系

作手运动学类型复杂程度的函数关系。最后,应当注意,就现代半导体存储器价格较低来说,价格问题远不如微型计算机运算速度来得重要,因为后者关系到机器人模型实时实现的可能性。

附 录 5.1

在本附录中,我们介绍六个自由度的斯坦福操作手的动力学模型的计算程序,操作手的运动学简图如表 5.39 所示。这个程序是由专家程序的输出文件得到的。

```

SUBROUTINE STANFM
COMMON/MATRIC/H(6,6),C(6,6,6),G(6)
COMMON/UG/Q(6)
  X11 = Q ( 1 )
  X12 = Q ( 2 )
  X2  = C O S ( X12 )
  X7  = S I N ( X12 )
  X13 = Q ( 3 )
  X14 = Q ( 4 )
  X4  = C O S ( X14 )
  X9  = S I N ( X14 )
  X15 = Q ( 5 )
  X5  = C O S ( X15 )
  X10 = S I N ( X15 )

  Q1=X4*(-0.13773E+01)
  P2=X5*Q1
  P1=X2*(-0.13773E+01)
  Q2=X7*P2
  Q1=X10*P1
  Q(5)=Q1+Q2

  Q1=X7* 0.13773E+01
  P1=X9*Q1
  Q1=X10*P1
  G(4)=Q1
  G1=X2* 0.63471E+02
  G(3)=Q1
  G1=X2*(-0.13773E+01)
  P3=X4*Q1
  P2=X5*(-0.13773E+01)
  P1=X13*(-0.63471E+02)
  Q2=X10*P3
  G1=X7*(P1+P2 -0.14047E+02)
  G(2)=Q1+Q2
  G(1)=0.
  P1=X5*(-0.27697E-01)
  P2=X5* 0.14000E 01
  P1=X2* 0.12397E-01
  G4=X5*P2
  G3=X4*P1
  G2=X5* 0.27697E-01
  Q1=X13* 0.14040E+00

```

P4=X4*(Q4 -0.15000E-01)
P3=X4*(-0.12397E-01)
P2=X10*Q3
P1=X7*(Q1+Q2+ 0.90556E-01)
Q3=X7*F4
Q2=X10*P3
Q1=X2*(P1+P2)
P3=X4*(-0.21397E-01)
P2=X7*Q3
P1=X10*(Q1+Q2)
Q2=X10*P3
Q1=X7*(P1+P2)

C(3,1,1)=Q1
Q1=X13*(-0.14040E+00)
P1=X4*(Q1 -0.90556E-01)
Q1=X10*P1
C(2,5,5)=Q1
Q3=X10* 0.76500E-02
Q2=X5*(-0.20047E-01)
Q1=X13*(-0.14040E+00)
P2=X10*Q3
P1=X5*(Q1+Q2 -0.90556E-01)
Q1=X9*(P1+P2 -0.80000E-02)

P2=X4*Q2
 P1=X9*Q1
 Q2=X7*P3
 Q1=X10*(P1+P2)
 C(1,5,5)=Q1+Q2
 P3=X10*(-0.76500E-02)
 P2=X5* 0.20047E-01
 P1=X13* 0.14040E+00
 Q4=X9* 0.21397E-01
 Q3=X10* 0.27697E-01
 Q2=X10*P3
 Q1=X5*(P1+P2+ 0.90556E-01)
 P2=X2*(Q3+Q4)
 P1=X4*(Q1+Q2+ 0.80000E-02)
 Q2=X5*P2
 Q1=X7*P1
 C(1,5,4)=Q1+Q2
 C(1,5,3)=0.
 Q1=X4* 0.27697E-01
 P3=X5*Q1
 P2=X5*(-0.76500E-02)
 P1=X10* 0.20047E-01
 Q4=X4* 0.21397E-01
 Q3=X7*P3
 Q2=X5*P2
 Q1=X10*P1
 P4=X5*Q4
 P3=X2* 0.21397E-01
 P2=X10*Q3
 P1=X2*(Q1+Q2+ 0.80000E-02)
 Q3=X7*P4
 Q2=X10*P3
 Q1=X9*(P1+P2)
 C(1,5,2)=Q1+Q2+Q3
 P2=X5*(-0.27697E-01)
 P1=X13*(-0.14040E+00)
 Q2=X2* 0.21397E-01
 Q1=X7*(P1+P2 -0.90556E-01)
 P2=X4*Q2
 P1=X9*Q1
 Q1=X10*(P1+P2)
 C(1,4,4)=Q1
 C(1,4,3)=0.
 Q3=X5*(-0.80000E-02)
 Q2=X5* 0.80000E-02
 Q1=X10*(-0.12047E-01)
 P3=X5*Q3
 P2=X5*Q2
 P1=X10*Q1
 Q4=X10*(-0.80000E-02)
 Q3=X4*(P3+ 0.75000E-02)
 Q2=X10*(-0.21397E-01)
 Q1=X9*(P1+P2 -0.75000E-02)
 P3=X10*Q4
 P2=X4*Q3

P1=X9*(Q1+Q2)
 Q1=X7*(P1+P2+P3)
 C(1,4,2)=Q1
 C(1,3,3)=0.
 Q1=X2*(-0.98603E+00)
 C(1,3,2)=Q1
 Q2=X5*(-0.16000E-01)
 Q1=X2* 0.11697E-01
 P4=X5*Q2
 P3=X4*Q1
 P2=X5* 0.27697E-01
 P1=X1,* 0.14040E+00
 Q4=X2* 0.21397E-01
 Q3=X2*(P4+ 0.10000E-01)
 Q2=X10*P3
 Q1=X7*(P1+P2+ 0.90556E-01)
 P5=X4*Q4
 P4=X5* 0.21397E-01
 P3=X13* 0.98603E+00
 P2=X4*Q3
 P1=X10*(Q1+Q2)
 Q3=X10*P5
 Q2=X7*(P3+P4+ 0.21822E+00)
 Q1=X9*(P1+P2)
 C(1,2,2)=Q1+Q2+Q3
 H(5,5)= 0.28397E-01
 H(5,4)=0.
 Q1=X10*(-0.14040E+00)
 H(5,3)=Q1
 Q1=X13* 0.14040E+00
 P1=X5*(Q1+ 0.90556E-01)
 Q1=X4*(P1+ 0.28397E-01)
 H(5,2)=Q1
 P1=X13* 0.14040E+00
 Q2=X2*(-0.21397E-01)
 Q1=X5*(P1+ 0.90556E-01)
 P3=X4*Q2
 P2=X10* 0.21397E-01
 P1=X7*(Q1+ 0.28397E-01)
 Q2=X5*P3
 Q1=X7*(P1+P2)
 H(5,1)=Q1+Q2
 P1=X10* 0.28397E-01
 Q1=X10*P1
 H(4,4)=Q1+ 0.10000E-02
 H(4,3)=0.
 Q2=X5*(-0.27697E-01)
 Q1=X13*(-0.14040E+00)
 P1=X9*(Q1+Q2 -0.90556E-01)
 Q1=X10*P1
 H(4,2)=Q1
 P2=X5* 0.27697E-01
 P1=X13* 0.14040E+00
 Q3=X9* 0.21397E-01
 Q2=X10* 0.28397E-01

```

Q1=X4*(P1+P2+ 0.90554E-01)
P2=X2*(Q2+Q3)
P1=X7*Q1
Q2=X2* 0.10000E-02
Q1=X10*(P1+P2)
H(4,1)=Q1+Q2
H(3,3)= 0.64700E+01
P1=X4*(-0.14040E+00)
Q1=X10*P1
H(3,2)=Q1
Q1=X9*(-0.14040E+00)
P1=X10*Q1
Q1=X7*(P1 -0.98603E+00)
H(3,1)=Q1
P2=X10* 0.12397E-01
P1=X9* 0.16000E-01
Q2=X10*P2
Q1=X9*P1
P5=X9* 0.20000E-02
P4=X4*(Q2+ 0.17000E-01)
P3=X13* 0.64700E+01
P2=X13* 0.28079E+00
P1=X5*(Q1+ 0.12397E-01)
Q4=X7*P5
Q3=X4*P4
Q2=X13*(P3+ 0.28638E+01)
Q1=X5*(P1+P2+ 0.18111E+00)
H(2,2)=Q1+Q2+Q3+Q4+ 0.34516E+01
Q2=X5*(-0.16000E-01)
Q1=X4* 0.11697E-01
P4=X5*Q2
P3=X7*Q1
P2=X5*(-0.27697E-01)
P1=X13*(-0.14040E+00)
Q1=X4* 0.21397E-01
Q3=X4*(P4+ 0.15000E-01)
Q2=X10*P3
Q1=X2*(P1+P2 -0.90554E-01)
P3=X7*Q4
P4=X5*(-0.21397E-01)
P3=X13*(-0.98603E+00)
P2=X7*Q3
P1=X10*(Q1+Q2)
Q3=X10*P5
Q2=X2*(P3+P4 -0.21823E+00)
Q1=X9*(P1+P2)
H(2,1)=Q1+Q2+Q3
P2=X4* 0.12397E-01
P1=X4* 0.16000E-01
Q4=X4*P2
Q3=X5* 0.35394E-01
Q2=X13* 0.28079E+00
Q1=X4*P1
P8=X9* 0.12397E-01
P7=X2*(Q4+ 0.16000E-01)
P6=X2*(Q2+Q3+ 0.18111E+00)
P5=X4* 0.20000E-02
P4=X9* 0.17000E-01
P3=X13* 0.64700E+01
P2=X13* 0.28079E+00
P1=X5*(Q1+ 0.12397E-01)
Q7=X9*P8
Q6=X2*P7
Q5=X4*P6
Q4=X4*P5
Q3=X9*P4
Q2=X13*(P3+ 0.28638E+01)
Q1=X5*(P1+P2+ 0.18111E+00)
P5=X2* 0.10700E+00
P4=X9* 0.42793E-01
P3=X10*(Q6+Q7)
P2=X10*Q5
P1=X7*(Q1+Q2+Q3+Q4+ 0.34336E+01)
Q3=X2*P5
Q2=X10*(P3+P4)
Q1=X7*(P1+P2)
H(1,1)=Q1+Q2+Q3+ 0.41634E+00
RETURN
END

```


参 考 文 献

- [1] Uicker J.J., On Dynamic Analysis of Spatial Linkages Using 4x4 Matrices, Ph. D. dissertation, Northwestern University, Evanstone, August, 1965.
- [2] Uicker J.J., "Dynamic Force Analysis of Spatial Linkages", Lafayette, Ind., Oct. 10-12, 1966.
- [3] Uicker J.J., "Dynamic Force Analysis of Spatial Linkages", ASME Journal of Applied Mechanics, June, pp. 418-424, 1967.
- [4] Kahn N.E., The Near Minimum Time Control of Open Loop Articulated Kinematic Chains, Ph. D. Thesis, Stanford University, MEMO AIM. 106, 1969.
- [5] Kahn N.E., Roth B., "The Near-Minimum-Time Control of Open-loop Articulated Kinematic Chains" ASME Journal of Dynamic Systems, Measurement and Control, Vol. 93, pp. 164-172, 1971.
- [6] Ball R.S., A Treatise on the Theory of Screws, Cambridge University press, London, 1900.
- [7] Brand L., Vector and Tensor Analysis, 4th ed., New York, Wiley.
- [8] Woo L.S., Freudenstein F., "Dynamic Analysis of Mechanisms Using Screw Coordinates", ASME Journal of Engineering for Industry, February, 1971.
- [9] Yang A.T., "Inertia Force Analysis of Spatial Mechanisms", ASME Journal of Engineering for Industry, February, 1971.
- [10] Orlandea N., Berenui T., "Dynamic Continuous Path Synthesis of Industrial Robots Using Adams Computer Program", ASME Journal of Mechanical Design, No. 5, 1981.
- [11] Mahil S.S., Modeling Techniques in the Control of Anthropomorphic Industrial Manipulators, Ph. D. thesis, Lanchester Polytechnic, Coventry, 1980.
- [12] Mahil S.S., "On the Application of Lagrange's Method to the Description of Dynamic Systems", IEEE Trans. on SMC, Vol. 12, No. 6, pp. 877-890, 1982.
- [13] Renaud M., "Contribution a l-Etude de la Modelisation et de la Commande des Systemes Mecaniques Articules", These de Docteur-Ingénieur, Toulouse, 1975.
- [14] Thomas M., Tesar D., "Dynamic Modeling of Serial Manipulator Arms", ASME Journal of Dynamic Systems, Measurement and Control, Vol. 104, No. 3, pp. 218-228, 1982.
- [15] Waters R.C., Mechanical Arm Control, A.I. Memo 549, MIT Artificial Intelligence Laboratory, 1979.
- [16] Hollerbach J.M., An Iterative Lagrangian Formulation of Manipulator Dynamics, A.I. Memo 533, MIT Artificial Intelligence Laboratory, March 1980.

- [17] Hollerbach J.M., "A Recursive Formulation of Lagrangian Manipulator Dynamics", IEEE Trans. on SMC, Vol. 10, No. 11, pp. 730-736, 1980.
- [18] Denavit J., Hartenberg R.S., "A Kinematic Notation for Lower Pair Mechanisms Based on Matrices", ASME Journal of Applied Mechanics, June, 1955.
- [19] Meqahed S., Renaud M., "Minimization of the Computation Time Necessary for the Dynamic Control of Robot Manipulators", 12th ISIR, Paris, 1982.
- [20] Vukobratović M., Stepanenko Y., "Mathematical Models of General Anthropomorphic Systems", Mathematical Biosciences, Vol. 17, pp 191-242, 1973.
- [21] Stepanenko Y., Vukobratović M., "Dynamics of Articulated Open Chain Active Mechanisms", Mathematical Biosciences, Vol. 28, pp. 137-170, 1976.
- [22] Vukobratović M., "Computer Method for Dynamic Model Construction of Active Articulated Mechanisms Using Kinetostatic Approach", Journal of Mechanism and Machine Theory, Vol. 13, pp. 19-39, 1978.
- [23] Vukobratović M., Kirčanski M., "Correlation Between Denavit-Hartenberg Manipulator Kinematics and Rodrigues Formula Approach, Journal of Mechanisms and Machine Theory, (in press) 1984.
- [24] Renaud N., "An Efficient Iterative Analytical Procedure for Obtaining a Robot Manipulator Dynamic Model", Proc. of First International Symp. of Robotics Research, Bretton Woods, New Hampshire, USA, 1983.
- [25] Vukobratović M., Potkonjak V., "Contribution to Automatic Forming of Active Chain Models via Lagrangian Form", ASME Journal of Applied Mechanics, No. 1, 1979.
- [26] Vukobratović M., Potkonjak V., Dynamics of Manipulation Robots, Springer-Verlag, 1982.
- [27] Orin D., McGhee R., Vukobratović M., Hartoch G., "Kinematic and Kinetic Analysis of Open-Chain Linkages Utilizing Newton-Euler Methods", Mathematical Biosciences, pp. 107-130, 1979.
- [28] Cvetković V., Contribution to the Dynamic Control Synthesis of Manipulators, Ph. D. Thesis, Electrical Engineering Faculty, Belgrade, 1981.
- [29] Cvetković V., Vukobratović M., "Computer-Oriented Algorithms Modelling of Active Spatial Mechanisms for Application in Robotics", IEEE Trans. on SMC, Vol. SMC-12, No. 6, 1982.
- [30] Huston R.L., Kelly P.A., "The Development of Equations of Motion of Single Arm Robots", IEEE Trans. on SMC, Vol. 12, No. 3, pp. 259-266, 1982.
- [31] Kane T.R., Wang C.F., "On the Derivation of Equations of Motion", J. Soc. for Ind. and Appl. Math., Vol. 13, pp. 487-492, 1965.
- [32] Kane T.R., Dynamics, New York: Holt, Reinhart, and Winston, 1968.

- [33] Luh J.Y.S., Walker M.W., Paul R.P.C., "On-Line Computational Scheme for Mechanical Manipulators", ASME Journal of Dynamic Systems, Measurement and Control, Vol. 102; No. 2, pp. 69-76, 1980.
- [34] Medvedov V.S., Leskov V.S., Yuzhenko A.C., Control Systems of Manipulation Robots, "Nauka", Moscow, 1978.
- [35] Popov E.P., Vereschagina A.F., Zenkevitch S.L., Manipulation Robots: Dynamics and Algorithms, "Nauka", Moscow, 1978.
- [36] Kirčanski N., Vukobratović M., "Control of Manipulators by Means of Discrete Regulators", Trans. of the Institute of Measurement and Control, Vol. 2, No. 3, 1980.
- [37] Vukobratović M., Kirčanski N., "Computer-Oriented Method for Linearization of Dynamic Models of Active Spatial Mechanisms", Journal of Mechanisms and Machine Theory, Vol. 16, No. 2, 1981.
- [38] Vukobratović M., Kirčanski N., "Computer Method for Dynamic Model Linearization of Active Spatial Mechanisms", Journal "Mashinovedenie" ANUSSR, No. 6, 1982.
- [39] Vukobratović M., Kirčanski N., "An Engineering Concept of Adaptive Control for Manipulation Robots via Parametric Sensitivity Analysis", Bulletin T. LXXXI de l'Académie Serbe des Sciences et des Arts, Classe des Sciences Techniques No. 20, 1982.
- [40] Vukobratović M., Kirčanski N., "Computer Assisted Sensitivity Model Generation in Manipulation Robots Dynamics", Journal of Mechanism and Machine Theory, No. 1, 1984.
- [41] Vukobratović M., Stokić D., Control of Manipulation Robots, Springer-Verlag, 1982.
- [42] Kane T.R., "Dynamics of Nonholonomic Systems", ASME Journal of Applied Mechanics, Vol. 28, pp. 574-578, 1961.
- [43] Kane T.R., Likins P., Kinematics of Rigid Bodies in Spaceflight, Applied Mechanics, Stanford University, Rep. 204, pp. 26-119, 1971.
- [44] Huston R.L., Passerello C.E., "On the Dynamics of Chain Systems", ASME Paper No. 74-WA/Aut 11, 1974.
- [45] Huston R.L., Passerello C.E., Harlow M.W., "Dynamics of Multi-rigid-Body Systems", ASME Journal of Applied Mechanics, Vol. 45, pp. 889-894, 1978.
- [46] Whittaker E.T., Analytical Dynamics, London, Cambridge, 1937.
- [47] Popov E.P., "Control of Robots-Manipulators", Journal "Tekhnicheskaya kibernetika ANUSSR, No. 6, Moscow, 1974.
- [48] Vukobratović M., Potkonjak V., "Two New Methods for Computer Forming of Dynamic Equations of Active Mechanisms", Journal of Mechanism and Machine Theory, Vol. 14, No. 3, 1979.
- [49] Bourne S.R., Horton J.R., "The Design of the Cambridge Algebra System", Proceedings of the Second Symposium on Symbolic and Algebraic Manipulation, New York, Association for Computing Machinery, pp. 134-143, 1971.

- [50] Manova M., Bloca S., Engleman C., "Rotational Functions in MATHLAB", Symbol Manipulation Languages and Techniques, Amsterdam, North-Holland, pp. 86-102, 1966.
- [51] Martin W., Patesan R., "The MACSYMA Systems", Proceedings of the Second Symposium on Symbolic and Algebraic Manipulation, New York, Association for Computing Machinery, pp. 59-75, 1971.
- [52] Byland E., Chapman C., "Symbolic Algebra in Theoretical Seismology", Proc. of the Second Symposium on Symbolic and Algebraic Manipulation, New York, Association for Computing Machinery, pp. 352-364, 1971.
- [53] Dillon S., Computer Assisted Equation Generation in Linkage Dynamics, Ph. D. Thesis, Ohio State University, 1973.
- [54] Aldon M.J., Liégeois A., "Génération et Programmation Automatiques des Equations de Lagrange des Robots et Manipulateurs", Rapport de Recherche, INRIA, 1981.
- [55] Luh J.Y.S., Lin C.S., "Scheduling of Parallel Computation for Computer-Controlled Mechanical Manipulators", IEEE Trans. on SMC, Vol. 12, No. 2, pp. 214-234, 1982.
- [56] Vukobratović M., Kirčanski N., "New Method for Real-Time Generation of Manipulator Dynamic Models" Proc. of 1st Yugoslav-Soviet Symp. on Applied Robotics, Moscow, 1983.
- [57] Kirčanski N., Contribution to Dynamics and Control of Manipulation Robots, Ph.D. Thesis, Electrical Engineering Faculty, Belgrade, 1984.
- [58] Vukobratović M., Kirčanski N., "Computer Aided Procedure of Forming of Robot Motion Equations in Analytical Forms", Proc. of VI IPTOMM Congress, New Delhi, 1983.
- [59] Kirčanski N., Contribution to the Synthesis of Manipulation Robots Control, MSc Thesis, Electrical Engineering Faculty, Belgrade, 1980.
- [60] Vukobratović M., Kirčanski N., "A Dynamic Approach to Nominal Trajectory Synthesis for Redundant Manipulators", IEEE Trans. on SMC, Vol. 14, No. 4, 1984.
- [61] Vukobratović M., Kirčanski N., "A Method for Computer-Aided Construction of Analytical Models of Robotic Manipulators", First IEEE Conf. on Robotics, Atlanta, 1984.
- [62] Vukobratović M., Kirčanski N., "Computer Assisted Generation of Robot Dynamic Models in Analytical Form", Acta Applicandae Mathematicae, International Journal of Applying Mathematics and Mathematical Applications, Vol. 2, No. 2, 1984.
- [63] Vukobratović M., Kirčanski N., "Implementation of High Efficient Analytical Robots Models on Microcomputers", V International Symposium on Theory and Practice of Robots and Manipulators, Udine, 1984.
- [64] Vukobratović M., Kirčanski N., "One Method for Simplified Manipulator Model Construction and its Application in Quasioptimal Trajectory Synthesis", Journal of Mechanism and Machine Theory, Vol. 17, No. 6, pp. 369-378, 1982.

- [65] Leskov A.G. Medvedov V.S., "Analysis of Dynamics and Control synthesis of Manipulation Robots", *Tekhnicheskaya Kibernetika*, ANUSSR, No. 6, Moscow, 1974.
- [66] Vereschagin A.F., "Computer-Aided Modelling Method of Manipulator Dynamics", *Tekhnicheskaya Kibernetika*, ANUSSR, No. 6, Moscow 1974.
- [67] Krutko P.D., Lakota N.A., "Control Synthesis of Manipulation Robots Motion Based on Inverse Dynamics Solution", *Tekhnicheskaya Kibernetika*, ANUSSR, No. 1, Moscow, 1981.
- [68] Vukobratović M., Li Shi-Gang, "One Efficient Procedure for Generating Dynamic Manipulator Models", *Robotica* (in press).
- [69] Paul R.P., *Robot Manipulators: Mathematics, Programming, and Control*, The MIT Press, Cambridge, Massachusetts, 1981.

名 词 索 引

三 画

工件 Work-piece 60
广义的 Generalized
坐标 coordinates 110
力 force 9, 41

四 画

反对称性 Antisymmetry 60
开链机构 Open-chain mechanisms
59
牛顿-欧拉方程 Newton-Euler's
equation 8, 26
专家程序 Expert program 52,
53, 231

五 画

半节肢机器人 Semiarthropoidal
robot 231
电力液压系统 Electro-hydraulic
systems 43
加速度能量 Acceleration energy
39, 40
节肢机器人 Arthro poidal robot
49, 167, 183
可动机构 Active mechanism 65
末端执行器 End-effector 60
外部坐标 External coordinates

65

正问题 Direct problem 8, 39,
44
正定矩阵 Positive definite
matrices 60, 87, 110, 168
存储单元占用量 Memory consump-
tion 235
主要的 Principal
主惯性轴 inertia axes 13, 28
主惯性矩 mument of inertia
46

六 画

闭式模型 Closed-form model
58, 113
动力学的 Dynamic
矩阵 matrices 48
步骤 stage 56
变量 variables 56
动力学 Dynamics 8
动能 Kinetic energy 9, 25
动静平衡法 Kinetostatic method
27
多项式矩阵 Polynomial matrix
50, 53, 112, 119, 124, 128,
186, 196
关节 Joint
加速度 acceleration 14

坐标 coordinate 63, 111, 128
 速度 velocity 69
 机构的构形 Configuration of
 mechanism 60
 机构变量 Mechanism variable
 114
 有限旋转公式 Finite-rotation
 formula 17, 126

七 画

驱动 Driving
 力 forces 8, 35, 72
 转矩 torques 11, 31, 35, 72
 角的 Angular
 加速度 acceleration 69
 速度 velocity 69
 近似的 Approximate
 模型 models 53, 136
 多项式 polynomials 138
 局部坐标系 Local coordinate
 system 11, 40
 连杆 Link 11, 61
 连杆机构 Linkage mechanisms
 12
 灵敏度 Sensitivity
 矩阵 matrices 101, 179
 模型 model 53, 60, 96
 拟人机器人 Anthropomorphic
 robot 53, 207
 运动学的 Kinematic
 链 chain 63, 92
 副 pair 62, 90, 170
 参数 parameters 170
 步骤 stage 56
 变量 variables 40, 142

八 画

参考坐标系 Reference coordinate
 system 66, 81
 采样周期 Sampling period 32
 单项式 Monomials 152
 单位向量 Unit vector 13
 空间机构 Spatial mechanisms
 43
 拉格朗日方程 Lagrange's equa-
 tions 8, 25, 28
 欧拉的 Euler's
 动力学方程 dynamic
 equation 26
 参数 parameters 27, 34
 势能 Potential energy 9
 实时实现 Real-time implementa-
 tion 32, 42, 232
 线性化模型 Linearized model
 59, 89, 176
 转换矩阵 Transformation matrix
 14, 67, 79

九 画

结点 Nodes 166
 递推方法 Recursive method
 8, 24
 面向计算机的 Computer oriented
 算法 algorithm 22
 方法 procedure 27
 重力向量 Gravity vector 10, 41, 84
 科里奥利效应 Coriolis effect
 10, 41, 58, 80

十 画

浮点 Floating-point
 加法 additions 15, 42
 乘法 multiplications
 15, 32, 42, 151
 离心效应 Centrifugal effect
 10, 41, 58, 80
 特性 Property
 反对称性 of antisymmetry
 173
 包含特性 of inclusion 142
 圆柱型机器人 Cylindrical robot
 168

十 一 画

笛卡儿(乘)积 Cartesian product
 116
 符号模型 Symbolic model 9
 基本计算图 Elementary graph
 158
 基始位置 "Home" position 65, 82
 维修装置 Rehabilitation devices
 43
 唯一性 Uniqueness 121
 旋转关节 Revolute joint 13, 73

十 二 画

程序语言 Programming language
 16, 52
 惯性的 Inertial
 力 forces 70

矩阵 matrix 10, 48, 74,
 100

转矩 moments 71
 张量 tensor 13

惯性矩 Moments of inertia 72
 滑动关节 Sliding joint 13
 棱柱形关节 Prismatic joint 60

十 三 画

数值法 Numeric methods 46
 数值-符号 Numeric-symbolic
 法 method 44, 54
 型 form 54
 微型计算机实现 Microcomputer
 implementation 42

十 四 画

模型 Model
 复杂性 complexity
 43
 执行机构 of actuators
 65

其 它

Appel 方程 Appel's equations
 8, 42
 D'Alembert 原理 D'Alembert's
 principle 56
 Denavit-Hartenberg 矩阵 Denavit-Hartenberg's matrices 17
 Gibbs 函数 Gibbs function 40
 Rodrigues 公式 Rodrigues
 formula 19, 28, 45