

铰接刚体动力学的李群公式

Junggon Kim

12/10/2012, Ver 2.01

摘要

在大多数旧式的动力学教科书中，通常描述刚体线性（或平移）和角度（或旋转）运动的公式是分别处理的。例如，著名的牛顿第二定律， $f = ma$ ，对于一个刚体的平移运动，有它的伙伴公式，所谓的欧拉方程，它描述了机体的旋转运动。然而，将平移和旋转分离，使得推导铰接式刚体系统（如机器人）的运动方程非常复杂。

在第 1 节中，使用李群公式推导出一个在 3D 空间中运动的刚体的优雅的统一运动方程。在第 2 节中，用刚体的几何公式重写了递归牛顿-欧拉算法（逆向动力学）、铰接机体算法（正向动力学），以及一个用于开链或树形铰接机体帧统的通用递归算法（混合动力学）。在第 3 节中，将描述受约束系统（如闭环机构）的动力学。最后，在第 4 节中，给出了动力学算法的解析导数，这将有助于优化和灵敏度分析。¹

1 刚体动力学

本节以几何方式描述单一刚体的运动方程。

1.1 刚体运动

要描述一个刚体的运动，我们需要同时表示机体的位置和方向。设 $\{B\}$ 是附着在刚体上的一个坐标帧，而 $\{A\}$ 是一个任意坐标帧，从现在起所有坐标帧都是右手笛卡尔坐标帧，我们可以定义一个 3×3 矩阵

$$R = [x_{ab}, y_{ab}, z_{ab}] \quad (1)$$

其中 $x_{ab}, y_{ab}, z_{ab} \in \mathbb{R}^3$ 是 $\{B\}$ 的坐标轴相对于 $\{A\}$ 的坐标。这种形式的矩阵称为旋转矩阵 (*rotation matrix*)，因为它可以用来描述刚体相对于参考帧的方向（或旋转）。因为旋转矩阵的列是相互正交的，并且坐标帧是右手的，旋转矩阵有两个性质，

$$RR^T = R^T R = I, \quad \det R = 1, \quad (2)$$

并被标志为 $SO(3)$ ²。

设 $p \in \mathbb{R}^3$ 是 $\{B\}$ 原点相对于 $\{A\}$ 原点的位置向量，并且 $R \in SO(3)$ 是 $\{B\}$ 相对于 $\{A\}$ 的旋转矩阵。刚体运动的构型空间可用一对 (R, p) 来表示，并被标志为 $SE(3)$ 。一个 4×4 矩阵，

$$T = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} \quad (3)$$

¹GEAR (用于铰接刚体仿真的几何引擎) 是本文算法的一个 C++ 实现。(<http://www.cs.cmu.edu/~junggon/tools/gear.html>)

²符号 SO 是特殊正交 (*special orthogonal*) 的缩写，并且“特殊”指的是 $\det R = +1$ 而不是 ± 1 。更多细节，参见文献 [2]。

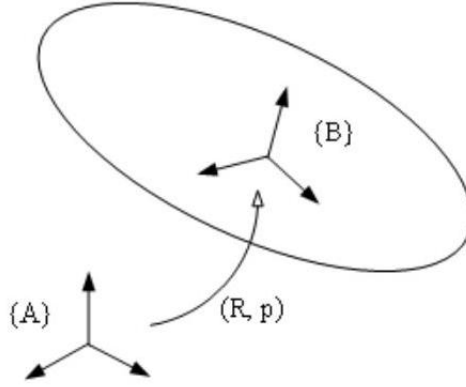


图 1: 刚体的坐标帧

被称为 $T = (R, p) \in SE(3)$ 的齐次表示 (*homogeneous representation*), 其逆矩阵可用以下方法获得

$$T^{-1} = \begin{bmatrix} R^T & -R^T p \\ 0 & 1 \end{bmatrix}. \quad (4)$$

从现在开始, 一个简单的声明, $T \in SE(3) : \{A\} \rightarrow \{B\}$, 将被用于通知 $T \in SE(3)$ 表示坐标帧 $\{B\}$ 相对于另一坐标帧 $\{A\}$ 的方向和位置。

$SE(3)$ 的李代数, 被标志为 $se(3)$, 被确定为一个 6 维向量空间 $(w, v) \in \mathbb{R}^6$, 其中 $w \in so(3)$, 为 $SO(3)$ 的李代数。 $\xi = (w, v) \in se(3)$ 也可以表示为 4×4 矩阵,

$$\xi = \begin{bmatrix} [w] & v \\ 0 & 0 \end{bmatrix} \quad (5)$$

其中 $[w] = \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix} \in \mathbb{R}^{3 \times 3}$ 是一个斜对称矩阵。

在 $\xi \in se(3)$ 上的 $T \in SE(3)$ 的伴随作用, $\text{Ad} : SE(3) \times se(3) \rightarrow se(3)$, 定义为

$$\text{Ad}_T \xi = T \xi T^{-1}. \quad (6)$$

从方程 (3), (4) 和 (5), Ad_T 可被视为一个线性变换, $\text{Ad}_T : se(3) \rightarrow se(3)$, 其由一个 6×6 矩阵定义

$$\text{Ad}_T = \begin{bmatrix} R & 0 \\ [p]R & R \end{bmatrix} \quad (7)$$

其中 $T = (R, p) \in SE(3)$ 。在 $\xi^* \in dse(3)$ 上的 T 的协伴随作用, $\text{Ad}_T^* : dse(3) \rightarrow dse(3)$, 是 ξ 的对偶, 由 6×6 矩阵定义

$$\text{Ad}_T^* = \text{Ad}_T^T. \quad (8)$$

1.2 广义的速度与动力

设 $T(t) = (R(t), p(t)) \in SE(3)$ 是附着于刚体的一个坐标帧相对于一个惯性帧的运动轨迹。刚体的广义速度 (*generalized velocity*) 定义为

$$V = T^{-1} \dot{T} = \begin{bmatrix} [w] & v \\ 0 & 0 \end{bmatrix} \quad (9)$$

其中 $[w] = R^T \dot{R}$ 并且 $v = R^T \dot{p}$ 。 $w \in \mathfrak{R}^3$ 的物理意义是相对于该惯性帧附着在机体上的坐标帧的旋转 (或角度) 速度, 但以机体坐标帧表达。类似地, $v \in \mathfrak{R}^3$ 表示这个坐标帧原点相对于该惯性帧的速度, 仍以机体帧表达。广义速度是 $se(3)$ 的一个元素, 并可简单地看作是一个 6 维向量, 即,

$$V = \begin{pmatrix} w \\ v \end{pmatrix}. \quad (10)$$

由于广义速度是 $se(3)$ 的一个实例, 因此它遵循在方程 (7) 中定义的伴随变换规则。设 $\{A\}, \{B\}$ 是两个不同的坐标帧, 它们附着在同一个刚体上, 并且 $T_a, T_b \in SE(3)$ 表示这两个坐标帧相对于惯性帧的方向和位置。然后, 从方程 (6) 和 (9) 看出, $\{A\}$ 和 $\{B\}$ 的广义速度具有以下关系:

$$V_b = \text{Ad}_{T_{ba}} V_a \quad (11)$$

其中 $T_{ba} \in SE(3) : \{B\} \rightarrow \{A\}$ 。

当坐标帧附着在刚体上时, 作用在机体上的广义动力 (*generalized force*) 可被定义为

$$F = \begin{pmatrix} m \\ f \end{pmatrix} \quad (12)$$

其中 $m \in \mathfrak{R}^3$ 和 $f \in \mathfrak{R}^3$ 分别标志在机体帧中观察, 作用在机体上的力矩和动力。广义动力称为 $dse(3)$ 的成员, 具有以下变换规则,

$$F_b = \text{Ad}_{T_{ab}}^* F_a \quad (13)$$

其中 F_a 和 F_b 标志从不同机体帧 $\{A\}$ 和 $\{B\}$ 观察到的广义动力, 并且 $T_{ab} \in SE(3) : \{A\} \rightarrow \{B\}$ 。

1.3 广义惯量和动量

刚体的动能由以下体积积分给出为

$$e = \int_{\text{vol}} \frac{1}{2} \|v\|^2 d\mathbf{m} \quad (14)$$

这意味着组成机体的所有质量粒子的动能之和。通过引入一个附着在机体上的坐标帧, 方程 (14) 可以重构为以下简单的二次型,

$$e = \frac{1}{2} V^T \mathcal{I} V \quad (15)$$

其中 $V \in \mathfrak{R}^6$ 是机体的广义速度, 而 $\mathcal{I} \in \mathfrak{R}^{6 \times 6}$, 即广义惯量 (*generalized inertia*), 表示相对于机体帧的质量和分布。

为了得到机体广义惯量的显式形式, 设 $r \in \mathfrak{R}^3$ 为一个机体点相对于机体帧的位置, 并且 $(R, p) \in SE(3)$ 分别表示机体帧相对于惯性帧的方向和位置。使用 $\|v\|^2 = \|\dot{p} + \dot{R}r\|^2$, $R^T \dot{R} = [w]$ 和 $R^T \dot{p} = v$, 方程 (14) 可重写为

$$e = \frac{1}{2} \int_{\text{vol}} \left(\|\dot{p}\|^2 + 2\dot{p}^T \dot{R}r + \|\dot{R}r\|^2 \right) d\mathbf{m} \quad (16)$$

$$= \frac{1}{2} \left\{ \dot{p}^T \dot{p} \int_{\text{vol}} d\mathbf{m} - 2\dot{p}^T R \left(\int_{\text{vol}} [r] d\mathbf{m} \right) w + w^T \left(\int_{\text{vol}} [r]^T [r] d\mathbf{m} \right) w \right\} \quad (17)$$

$$= \frac{1}{2} \left\{ m v^T v - 2v^T \left(\int_{\text{vol}} [r] d\mathbf{m} \right) w + w^T \left(\int_{\text{vol}} [r]^T [r] d\mathbf{m} \right) w \right\} \quad (18)$$

$$= \frac{1}{2} V^T \mathcal{I} V \quad (19)$$

其中 $V = (w, v)$ 是机体的广义速度，而广义惯量 \mathcal{I} 具有以下显式矩阵形式：

$$\mathcal{I} = \begin{bmatrix} \int_{\text{vol}} [r]^T [r] d\mathbf{m} & \int_{\text{vol}} [r] d\mathbf{m} \\ \int_{\text{vol}} [r]^T d\mathbf{m} & \mathbf{m}\mathbf{1} \end{bmatrix}. \quad (20)$$

广义惯量是对称正定的，并且它的上对角项， $I = \int_{\text{vol}} [r]^T [r] d\mathbf{m}$ ，是众所周知的刚体相对于机体帧的 3×3 惯量矩阵的定义。如果机体帧的原点位于质心上，则由于 $\int_{\text{vol}} [r] d\mathbf{m} = 0$ ，因此广义惯量成为块对角矩阵。另外，如果机体帧的方向也与机体的主轴重合，则广义惯量成为对角线矩阵。

设 $\{A\}$ 和 $\{B\}$ 是附着在机体上的坐标帧， \mathcal{I}_a 和 \mathcal{I}_b 是对应于这两个坐标帧的机体的广义惯量。使用方程 (11)，(15)，以及在坐标帧变化时机体动能应保持不变的事实，可以获得以下广义惯量之间的变换规则：

$$\mathcal{I}_b = \text{Ad}_{T_{ab}}^* \mathcal{I}_a \text{Ad}_{T_{ab}} \quad (21)$$

其中 $T_{ab} \in SE(3) : \{A\} \rightarrow \{B\}$ 。如果给定质量 m 和在质心坐标帧³中的惯量矩阵， $I_c \in \mathbb{R}^{3 \times 3}$ ，可以由方程 (21) 得到任意机体帧的广义转动惯量，而不必使用方程 (20)，为

$$\mathcal{I} = \begin{bmatrix} RI_c R^T + m[p]^T [p] & m[p] \\ m[p]^T & m\mathbf{1} \end{bmatrix} \quad (22)$$

其中 $(R, p) \in SE(3)$ 表示质心帧相对于机体帧的方向和位置。

刚体的广义动量 (*generalized momentum*) 定义为

$$L = \mathcal{I}V \quad (23)$$

其中 \mathcal{I} 和 V 是机体的广义惯量和速度，在附着在机体上的一个坐标帧中表达。

L_a 和 L_b 是一个机体的广义动量，分别在不同的机体帧中表达， $\{A\}$ 和 $\{B\}$ 。使用方程 (11) 和 (21) 可以推导出广义动量的以下变换规则：

$$L_b = \text{Ad}_{T_{ab}}^* L_a \quad (24)$$

这与广义动力是一样的，实际上，广义动量也被称为 $dse(3)$ 。

1.4 $se(3)$ 和 $dse(3)$ 的时间导数

回想一下，一个 3 维向量 $x = \sum_{i=1}^3 x_i \hat{e}_i$ 的时间导数，在一个移动坐标帧 $\{\hat{e}_i\}$ 中表达，可获得为

$$\frac{d}{dt}x = \sum_{i=1}^3 \left\{ \left(\frac{d}{dt}x_i \right) \hat{e}_i + x_i \left(\frac{d}{dt}\hat{e}_i \right) \right\} = \dot{x} + w \times x \quad (25)$$

其中 $\dot{x} = \sum_{i=1}^3 \left(\frac{d}{dt}x_i \right) \hat{e}_i = \left(\frac{dx_1}{dt}, \frac{dx_2}{dt}, \frac{dx_3}{dt} \right)$ 是 x 的分量上的时间导数，而 w 是移动帧的角速度。

在移动帧中表达的 $se(3)$ 和 $dse(3)$ 的时间导数具有更一般化的形式，获得导数的一种简单方法是将 $se(3)$ (或 $dse(3)$) 变换到静止坐标帧中，在静止坐标帧中进行微分，并 (正确地) 假设 $se(3)$ (或 $dse(3)$) 的导数可以用 $se(3)$ (或 $dse(3)$) 的规则进行变换，将导数变换回原始移动坐标帧。⁴ 注意，将静止坐标帧中的向量相对于时间进行微分只是得到它的分量导数。

³一个坐标帧，其原点位于机体质心上。

⁴在文献 [1] 中，用这种方法得到了空间速度的时间导数，它类似于本文中的广义速度。

引理 1. 设 $X \in se(3)$ 可在附着于刚体的一个移动帧中表达。则 X 的时间导数可通过下式获得

$$\frac{d}{dt}X = \dot{X} + \text{ad}_V X \quad (26)$$

其中 \dot{X} 是 X 的分量上的时间导数，并且 $\text{ad}_V : se(3) \rightarrow se(3)$ 是一个线性变换，定义为

$$\text{ad}_V = \begin{bmatrix} [w] & 0 \\ [v] & [w] \end{bmatrix} \quad (27)$$

其中 $V = (w, v) \in se(3)$ 是机体的广义速度。

证明. 设 $T = (R, p) \in SE(3)$ 标志移动帧相对于静止在空间中的惯性帧的方向和位置。通过将 X 变换到惯性帧，在那里进行微分，然后将结果变换回原来的机体帧，就可以得到

$$\frac{d}{dt}X = \text{Ad}_{T^{-1}} \frac{d'}{dt} (\text{Ad}_T X) = \dot{X} + \text{Ad}_{T^{-1}} \frac{d'}{dt} (\text{Ad}_T) X$$

其中 $\frac{d'}{dt}$ 表示分量微分，并且 $\dot{X} = \frac{d'}{dt}X$ 。使用方程 (7) 和 $V = (w, v) = (R^T \dot{R}, R^T \dot{p})$ ，可以证明 $\text{Ad}_{T^{-1}} \frac{d'}{dt} (\text{Ad}_T) = \text{ad}_V$ ，如下所示：

$$\begin{aligned} \text{Ad}_{T^{-1}} \frac{d'}{dt} \text{Ad}_T &= \begin{bmatrix} R^T & 0 \\ -R^T[p] & R^T \end{bmatrix} \begin{bmatrix} \dot{R} & 0 \\ [p]\dot{R} + [p]\dot{R} & \dot{R} \end{bmatrix} \\ &= \begin{bmatrix} [w] & 0 \\ [v] & [w] \end{bmatrix} \end{aligned}$$

□

引理 2. 设 $Y \in dse(3)$ 在附着于刚体的移动帧中表达。则 Y 的时间导数可通过下式获得

$$\frac{d}{dt}Y = \dot{Y} - \text{ad}_V^* Y \quad (28)$$

其中 \dot{Y} 是 Y 的分量上的时间导数，并且 $\text{ad}_V^* : dse(3) \rightarrow dse(3)$ 是一个线性变换，定义为

$$\text{ad}_V^* = \text{ad}_V^T = \begin{bmatrix} [w] & 0 \\ [v] & [w] \end{bmatrix}^T \quad (29)$$

其中 $V = (w, v) \in se(3)$ 是机体的广义速度。

证明. 类似于引理 1 的证明， $Y \in dse(3)$ 的导数可由下式获得

$$\frac{d}{dt}Y = \text{Ad}_T^* \frac{d'}{dt} (\text{Ad}_{T^{-1}}^* Y) = \dot{Y} + \text{Ad}_T^* \frac{d'}{dt} (\text{Ad}_{T^{-1}}^*) Y.$$

使用方程 (8) 和 $V = (w, v) = (R^T \dot{R}, R^T \dot{p})$ ，可以证明 $\text{Ad}_T^* \frac{d'}{dt} \text{Ad}_{T^{-1}}^* = -\text{ad}_V^*$ ，如下所示：

$$\begin{aligned} \text{Ad}_T^* \frac{d'}{dt} \text{Ad}_{T^{-1}}^* &= \begin{bmatrix} R^T & -R^T[p] \\ 0 & R^T \end{bmatrix} \begin{bmatrix} \dot{R} & [p]\dot{R} + [p]\dot{R} \\ 0 & \dot{R} \end{bmatrix} \\ &= \begin{bmatrix} [w] & [v] \\ 0 & [w] \end{bmatrix} = -\text{ad}_V^T = -\text{ad}_V^* \end{aligned}$$

□

1.5 刚体的几何动力学

刚体的运动方程可以写为

$$F = \frac{d}{dt} L \quad (30)$$

其中 F 表示作用在机体上的广义动力的净总和，并且 L 是机体的广义动量。使用方程 (23) 和 (28)，刚体的运动方程可以写为

$$F = \mathcal{I}\dot{V} - \text{ad}_V^* \mathcal{I}V \quad (31)$$

其中 \dot{V} 是机体广义速度的分量上的时间导数。注意， $\dot{L} = \dot{\mathcal{I}}V + \mathcal{I}\dot{V} = \mathcal{I}\dot{V}$ ，因为 \mathcal{I} 的分量不变化。

机体的动力学方程是坐标不变的，即方程的结构在坐标帧变化时仍然保持不变。设 $\{A\}$ 和 $\{B\}$ 为机体的坐标帧， V_a 和 V_b 为广义速度， F_a 和 F_b 为广义动力，并且 \mathcal{I}_a 和 \mathcal{I}_b 分别为 $\{A\}$ 和 $\{B\}$ 对应的广义惯量。使用变换规则，方程 (11)，(13)，和 (21)，可以很容易地将相对于 $\{A\}$ 的动力学方程， $F_a = \mathcal{I}_a \dot{V}_a - \text{ad}_{V_a}^* \mathcal{I}_a V_a$ ，变换为在 $\{B\}$ 中的方程， $F_b = \mathcal{I}_b \dot{V}_b - \text{ad}_{V_b}^* \mathcal{I}_b V_b$ ，并且这表明方程 (31) 在坐标帧变化下的坐标不变性 (coordinate invariance)。

2 开链系统的动力学

设 $q \in \mathbb{R}^n$ 标志系统中所有关节的坐标集合，并且对于开链系统， n 等于系统的自由度。系统的动力学方程可以写为

$$M\ddot{q} + b = \tau \quad (32)$$

其中 $M(q) \in \mathbb{R}^{n \times n}$ 是系统的对称质量矩阵， $b(q, \dot{q}) \in \mathbb{R}^n$ 表示科里奥利项 (Coriolis term)、离心项和重力项，并且 $\tau \in \mathbb{R}^n$ 标志对应于系统坐标 q 的扭矩 (或动力) 向量。

我们假设当前系统状态 (q, \dot{q}) 是完全已知的。用规定的加速度 \ddot{q} 计算关节力矩 (或平移坐标的力) τ 被称为逆向动力学 (*inverse dynamics*)。它通常用于获得所需的关节力矩，使系统沿指定的关节轨迹移动。另一方面，根据给定的关节力矩 τ 计算得到的关节加速度 \ddot{q} 被称为正向动力学 (*forward dynamics*)，通常通过积分计算的加速度来及时模拟系统运动，以获得下一个时间步骤的系统状态。

通常，在模拟过程中，输入到关节坐标的命令可以是扭矩或加速度，并且对于所有坐标，命令类型不必相同。让我们把运动方程改写为

$$M \begin{pmatrix} \ddot{q}_u \\ \ddot{q}_v \end{pmatrix} + b = \begin{pmatrix} \tau_u \\ \tau_v \end{pmatrix} \quad (33)$$

其中，下标 “ u ” 标志具有规定加速度的关节坐标，下标 “ v ” 表示具有给定或已知扭矩的坐标。我们通过求解运动方程，从规定的 (\ddot{q}_u, τ_v) 可以获得 (τ_u, \ddot{q}_v) ，这被称为混合动力学 (*hybrid dynamics*)⁵。混合动力学的一个可能的解是重新排列方程 (33) 并使用直接矩阵求逆进行求解。例如，扭矩指定的关节坐标的结果加速度可获取为

$$\ddot{q}_v = M_{vv}^{-1} (\tau_v - b_v - M_{uv} \ddot{q}_u) \quad (34)$$

其中 $M = \begin{bmatrix} M_{uu} & M_{uv} \\ M_{uv} & M_{vv} \end{bmatrix}$ ， $b = \begin{pmatrix} b_u \\ b_v \end{pmatrix}$ 并且 $q = \begin{pmatrix} q_u \\ q_v \end{pmatrix}$ 。但是，该方法对于一个复杂的系统并不有效，因为它需要建立质量矩阵，并求逆对应于未规定的关节的子矩阵，这导致 $O(n^2) + O(n_b^3)$ 算法，其中 n 和 n_v 分别标志所有坐标数和未规定坐标数。

⁵我们遵循文献 [1] 的术语。

逆向动力学	$\ddot{q} \rightarrow \tau$
正向动力学	$\tau \rightarrow \ddot{q}$
混合动力学	$(\ddot{q}_u, \tau_v) \rightarrow (\tau_u, \ddot{q}_v)$

表 1: 动力学算法的输入与输出

在表 1 中总结了逆向、正向和混合动力学的输入和输出。注意，混合动力学是传统逆向动力学和正向动力学的推广，即当 $q_u = q$ (逆向动力学) 和 $q_v = q$ (正向动力学) 时，可以将其视为混合动力学的极端情况。

2.1 递归逆向动力学

在文献 [3] 中提出了一个使用第 1 节所示几何符号的递归牛顿-欧拉逆向动力学算法。在这里，该算法略有修改，以支持多自由度关节的公式。

设 $\{0\}$ 是在空间中静止的惯性帧， $\{i\}$ 是附着在开链系统的第 i 个机体的坐标帧，并且 $\{\lambda(i)\}$ 是附着在第 i 个机体的父机体的坐标帧。另外，设 $T_i \in SE(3) : \{0\} \rightarrow \{i\}$ ， $T_{\lambda(i)} \in SE(3) : \{0\} \rightarrow \{\lambda(i)\}$ ，并且 $T_{\lambda(i),i} \in SE(3) : \{\lambda(i)\} \rightarrow \{i\}$ 。从 $T_i = T_{\lambda(i)}T_{\lambda(i),i}$ 和方程 (6)，第 i 个机体的广义速度可以重写为

$$V_i = T_i^{-1} \dot{T}_i \quad (35)$$

$$= T_{\lambda(i),i}^{-1} T_{\lambda(i)}^{-1} \left(\dot{T}_{\lambda(i)} T_{\lambda(i),i} + T_{\lambda(i)} \dot{T}_{\lambda(i),i} \right) \quad (36)$$

$$= \text{Ad}_{T_{\lambda(i),i}^{-1}} V_{\lambda(i)} + S_i \dot{q}_i \quad (37)$$

其中 $S_i \dot{q}_i = T_{\lambda(i),i}^{-1} \dot{T}_{\lambda(i),i} \in se(3)$ 表示第 i 个机体相对于其父机体的相对速度。 $S_i = S_i(q_i) \in (se(3) \times n_i)$ 被称为连接第 i 个机体与其父机体的关节的雅可比矩阵，并且 $q_i \in \mathbb{R}^{n_i}$ 表示该关节的坐标向量。

如方程 (31) 所示，需要系统中所有物体广义速度的分量上的时间导数来建立每个机体的动力学方程。回想对于任意矩阵 A ， $\dot{A}^{-1} = -A^{-1} \dot{A} A^{-1}$ ，并且对于任意 $\xi_1, \xi_2 \in se(3)$ ， $\text{ad}_{\xi_1} \xi_2 = \xi_1 \xi_2 - \xi_2 \xi_1$ ，因此可以对于 V_i 推导以下公式， V_i 的分量上的时间导数为：

$$\dot{V}_i = \frac{d'}{dt} \left(T_{\lambda(i),i}^{-1} V_{\lambda(i)} T_{\lambda(i),i} \right) + \dot{S}_i \dot{q}_i + S_i \ddot{q}_i \quad (38)$$

$$= -T_{\lambda(i),i}^{-1} \dot{T}_{\lambda(i),i} T_{\lambda(i),i}^{-1} V_{\lambda(i)} T_{\lambda(i),i} + T_{\lambda(i),i}^{-1} \dot{V}_{\lambda(i)} T_{\lambda(i),i} + T_{\lambda(i),i}^{-1} V_{\lambda(i)} \dot{T}_{\lambda(i),i} + \dot{S}_i \dot{q}_i + S_i \ddot{q}_i \quad (39)$$

$$= \text{Ad}_{T_{\lambda(i),i}^{-1}} \dot{V}_{\lambda(i)} + \text{ad}_{\text{Ad}_{T_{\lambda(i),i}^{-1}} V_{\lambda(i)}} S_i \dot{q}_i + \dot{S}_i \dot{q}_i + S_i \ddot{q}_i \quad (40)$$

$$= \text{Ad}_{T_{\lambda(i),i}^{-1}} \dot{V}_{\lambda(i)} + \text{ad}_{V_i} S_i \dot{q}_i + \dot{S}_i \dot{q}_i + S_i \ddot{q}_i \quad (41)$$

方程 (37) 和 (41) 非常适合递归地计算开链系统中每个机体，从地面到系统末端的广义速度，及其分量方向的时间导数，因为地面的速度和加速度在大多数情况下是已知的。⁶ 注意， $\dot{S}_i \neq 0$ ，因为关节雅可比矩阵 S_i 通常是 q_i 的函数。

设 $F_i \in dse(3)$ 是由父机体通过连接关节传递给第 i 个机体的广义动力，并且 $F_i^{\text{ext}} \in dse(3)$ 是环境作用于第 i 个机体的广义动力。 F_i 和 F_i^{ext} 两者都在 $\{i\}$ 中表达。根据方程 (31)，第 i 个机体

⁶可以假设 $V_0 = 0$ 和 $\dot{V}_0 = (0, g)$ ，其中 $g \in \mathbb{R}^3$ 标志在惯性帧中观察的重力向量，具有适当的方向和大小。

表 2: 递归逆向动力学

```

while forward recursion do
   $T_{\lambda(i),i}$  = function of  $q_i$ 
   $V_i = \text{Ad}_{T_{\lambda(i),i}^{-1}} V_{\lambda(i)} + S_i \dot{q}_i$ 
   $\dot{V}_i = \text{Ad}_{T_{\lambda(i),i}^{-1}} \dot{V}_{\lambda(i)} + \text{ad}_{V_i} S_i \dot{q}_i + \dot{S}_i \dot{q}_i + S_i \ddot{q}_i$ 
end while
while backward recursion do
   $F_i = \mathcal{I}_i \dot{V}_i - \text{ad}_{V_i}^* \mathcal{I}_i V_i - F_i^{\text{ext}} + \sum_{k \in \mu(i)} \text{Ad}_{T_{i,k}^{-1}}^* F_k$ 
   $\tau_i = S_i^T F_i$ 
end while

```

的运动方程可以写为

$$F_i + F_i^{\text{ext}} - \sum_{k \in \mu(i)} \text{Ad}_{T_{i,k}^{-1}}^* F_k = \mathcal{I}_i \dot{V}_i - \text{ad}_{V_i}^* \mathcal{I}_i V_i \quad (42)$$

其中，方程式左侧表示作用在机体上的净力， $\mu(i)$ 是第 i 个机体的子机体集合，并且 $-\text{Ad}_{T_{i,k}^{-1}}^* F_k$ 是广义动力，由第 k 个子机体传递过来，在 $\{i\}$ 中表达。应该注意的是，对于每个机体的广义动力 F_i ，可通过方程 (42) 递归计算，从末端到地面，因为末端机体没有子机体。

开链系统的递归逆向动力学算法如表 2 所示，以下是几何逆向动力学的符号列表：

- i = 第 i 个机体的索引。
- $\lambda(i)$ = 第 i 个机体的父机体的索引。
- $\mu(i)$ = 第 i 个机体的子机体的索引集合。
- $q_i \in \mathbb{R}^{n_i}$ = 连接第 i 个机体与其父机体的第 i 个关节的坐标。
- $\tau_i \in \mathbb{R}^{n_i}$ = 第 i 个关节施加的扭矩 (或动力)。
- $T_{\lambda(i),i} \in SE(3) : \{\lambda(i)\} \rightarrow \{i\}$ ，一个 q_i 的函数。
- $V_i \in se(3)$ = 第 i 个机体的广义速度，在机体帧 $\{i\}$ 中观察。
- $\dot{V}_i \in se(3)$ = V_i 的分量上的时间导数。
- $S_i \in (se(3) \times n_i) = T_{\lambda(i),i}$ 的雅可比矩阵，在 $\{i\}$ 中观察。
 $S_i = \left[T_{\lambda(i),i}^{-1} \frac{\partial T_{\lambda(i),i}^{-1}}{\partial q_i^1}, \dots, T_{\lambda(i),i}^{-1} \frac{\partial T_{\lambda(i),i}^{-1}}{\partial q_i^{n_i}} \right]$ ，其中 $q_i^k \in \mathbb{R}$ 标志第 i 个关节的第 k 个坐标，即 $q_i = (q_i^1, \dots, q_i^{n_i})$ 。
- \mathcal{I}_i = 第 i 个机体的广义惯量，在 $\{i\}$ 中观察。
- $F_i \in dse(3)$ = 从其父机体通过连接关节传递到第 i 个机体的广义动力，在 $\{i\}$ 中观察。
- $F_i^{\text{ext}} \in dse(3)$ = 环境作用在第 i 个机体上的广义动力，在 $\{i\}$ 中观察。

2.2 递归正向动力学

Featherstone [1] 发现第 i 个机体的动力学方程可以重新表述为以下形式,

$$F_i = \hat{\mathcal{I}}_i \dot{V}_i + \hat{\mathcal{B}}_i \quad (43)$$

其中 $\hat{\mathcal{I}}_i$ 被称为机体的铰接机体惯量 (*articulated body inertia*), 并且 $\hat{\mathcal{B}}_i$ 是关联的导向力 (*bias force*)。他还指出, 可以递归地计算开链系统中每个机体对应的铰接机体惯量和导向力, 使用这些新的量, 可以用 $O(n)$ 算法求解正向动力学。在文献 [4] 中报告了一个铰接机体惯量法的李群公式。在这里, 提出了一种支持多自由度关节模型的更一般形式的几何公式。

从方程 (43) 开始, 我们将证明相同形式的动力学方程仍然适用于第 i 个机体的父机体 (第 $\lambda(i)$ 个机体), 并且 $\hat{\mathcal{I}}_{\lambda(i)}$ 和 $\hat{\mathcal{B}}_{\lambda(i)}$ 可由 $\hat{\mathcal{I}}_i$ 和 $\hat{\mathcal{B}}_i$ 计算出来, 这导致了它们的逆向递归过程。

假设第 i 个机体的运动方程可以写为方程 (43)。对于在方程 (43) 中的 \dot{V}_i , 通过将方程 (41) 代入, 可以得到

$$F_i = \hat{\mathcal{I}}_i \left(\text{Ad}_{T_{\lambda(i),i}^{-1}} \dot{V}_{\lambda(i)} + \text{ad}_{V_i} S_i \dot{q}_i + S_i \ddot{q}_i + \dot{S}_i \dot{q}_i \right) + \hat{\mathcal{B}}_i, \quad (44)$$

并从 $S_i^T F_i = \tau_i$, 未知的 \ddot{q}_i 可写为

$$\ddot{q}_i = \left(S_i^T \hat{\mathcal{I}}_i S_i \right)^{-1} \left\{ \tau_i - S_i^T \hat{\mathcal{I}}_i \left(\text{Ad}_{T_{\lambda(i),i}^{-1}} \dot{V}_{\lambda(i)} + \text{ad}_{V_i} S_i \dot{q}_i + \dot{S}_i \dot{q}_i \right) - S_i^T \hat{\mathcal{B}}_i \right\}. \quad (45)$$

从方程 (42), 第 $\lambda(i)$ 个机体的动力学方程变为

$$F_{\lambda(i)} = \mathcal{I}_{\lambda(i)} \dot{V}_{\lambda(i)} - \text{ad}_{V_{\lambda(i)}}^* \mathcal{I}_{\lambda(i)} V_{\lambda(i)} - F_{\lambda(i)}^{\text{ext}} + \sum_{k \in \mu(\lambda(i))} \text{Ad}_{T_{\lambda(i),k}^{-1}}^* F_k, \quad (46)$$

并对于在方程 (46) 中的 F_k , 通过将方程 (44) 代入, 可以得到

$$\begin{aligned} F_{\lambda(i)} = & \mathcal{I}_{\lambda(i)} \dot{V}_{\lambda(i)} - \text{ad}_{V_{\lambda(i)}}^* \mathcal{I}_{\lambda(i)} V_{\lambda(i)} - F_{\lambda(i)}^{\text{ext}} \\ & + \sum_{k \in \mu(\lambda(i))} \text{Ad}_{T_{\lambda(i),k}^{-1}}^* \left\{ \hat{\mathcal{I}}_k \left(\text{Ad}_{T_{\lambda(i),k}^{-1}} \dot{V}_{\lambda(i)} + \text{ad}_{V_k} S_k \dot{q}_k + S_k \ddot{q}_k + \dot{S}_k \dot{q}_k \right) + \hat{\mathcal{B}}_k \right\}. \end{aligned} \quad (47)$$

对于在方程 (47) 中未知数的 \ddot{q}_k , 通过将方程 (45) 代入, 并根据 $\dot{V}_{\lambda(i)}$ 排列方程, 可以得到对于第 $\lambda(i)$ 个机体的动力学方程, 所期望的形式如下:

$$F_{\lambda(i)} = \hat{\mathcal{I}}_{\lambda(i)} \dot{V}_{\lambda(i)} + \hat{\mathcal{B}}_{\lambda(i)} \quad (48)$$

其中

$$\hat{\mathcal{I}}_{\lambda(i)} = \mathcal{I}_{\lambda(i)} + \sum_{k \in \mu(\lambda(i))} \text{Ad}_{T_{\lambda(i),k}^{-1}}^* \left\{ \hat{\mathcal{I}}_k - \hat{\mathcal{I}}_k S_k \left(S_k^T \hat{\mathcal{I}}_k S_k \right)^{-1} S_k^T \hat{\mathcal{I}}_k \right\} \text{Ad}_{T_{\lambda(i),k}^{-1}} \quad (49)$$

$$\begin{aligned} \hat{\mathcal{B}}_{\lambda(i)} = & - \text{ad}_{V_{\lambda(i)}}^* \mathcal{I}_{\lambda(i)} V_{\lambda(i)} - F_{\lambda(i)}^{\text{ext}} \\ & + \sum_{k \in \mu(\lambda(i))} \text{Ad}_{T_{\lambda(i),k}^{-1}}^* \left\{ \hat{\mathcal{B}}_k + \hat{\mathcal{I}}_k \left(\text{ad}_{V_k} S_k \dot{q}_k + \dot{S}_k \dot{q}_k \right) \right. \\ & \left. + \hat{\mathcal{I}}_k S_k \left(S_k^T \hat{\mathcal{I}}_k S_k \right)^{-1} \left(\tau_k - S_k^T \hat{\mathcal{I}}_k \left(\text{ad}_{V_k} S_k \dot{q}_k + \dot{S}_k \dot{q}_k \right) - S_k^T \hat{\mathcal{B}}_k \right) \right\}. \end{aligned} \quad (50)$$

综上所述, 开链系统正向动力学的 $O(n)$ 算法由以下三个主要递归过程组成:

1. 正向递归: 使用方程 (37) 对于每个机体递归计算 V_i 。

表 3: 递归正向动力学

while forward recursion do $T_{\lambda(i),i} = \text{function of } q_i$ $V_i = \text{Ad}_{T_{\lambda(i),i}^{-1}} V_{\lambda(i)} + S_i \dot{q}_i$ $\eta_i = \text{ad}_{V_i} S_i \dot{q}_i + \dot{S}_i \dot{q}_i$ end while while backward recursion do $\hat{\mathcal{I}}_i = \mathcal{I}_i + \sum_{k \in \mu(i)} \text{Ad}_{T_{i,k}^{-1}}^* \Pi_k \text{Ad}_{T_{i,k}^{-1}}$ $\hat{\mathcal{B}}_i = -\text{ad}_{V_i}^* \mathcal{I}_i V_i - F_i^{\text{ext}} + \sum_{k \in \mu(i)} \text{Ad}_{T_{i,k}^{-1}}^* \beta_k$ $\Psi_i = \left(S_i^T \hat{\mathcal{I}}_i S_i \right)^{-1}$ $\Pi_i = \hat{\mathcal{I}}_i - \hat{\mathcal{I}}_i S_i \Psi_i S_i^T \hat{\mathcal{I}}_i$ $\beta_i = \hat{\mathcal{B}}_i + \hat{\mathcal{I}}_i \left\{ \eta_i + S_i \Psi_i \left(\tau_i - S_i^T \left(\hat{\mathcal{I}}_i \eta_i + \hat{\mathcal{B}}_i \right) \right) \right\}$ end while while forward recursion do $\ddot{q}_i = \Psi_i \left\{ \tau_i - S_i^T \hat{\mathcal{I}}_i \left(\text{Ad}_{T_{\lambda(i),i}^{-1}} \dot{V}_{\lambda(i)} + \eta_i \right) - S_i^T \hat{\mathcal{B}}_i \right\}$ $\dot{V}_i = \text{Ad}_{T_{\lambda(i),i}^{-1}} \dot{V}_{\lambda(i)} + S_i \ddot{q}_i + \eta_i$ $F_i = \hat{\mathcal{I}}_i \dot{V}_i + \hat{\mathcal{B}}_i$ end while

2. 逆向递归: 使用方程 (49) 和 (50) 对于每个机体递归计算 $\hat{\mathcal{I}}_i$ 和 $\hat{\mathcal{B}}_i$ 。

3. 正向递归: 使用方程 (45) 和 (41) 对于每个机体递归地计算 \ddot{q}_i 和 \dot{V}_i 。

表 3 中显示了开链系统的正向动力学算法, 为了方程的简单和效率, 增加了一些中间变量, 如 η_i, Ψ_i, Π_i 和 β_i 。

2.3 递归混合动力学

在文献 [5] 中报道了一种用于开链系统混合动力学的几何递归算法, 但该表述基于单自由度关节。本文提出了更一般的几何混合动力学形式, 以更方便地支持多自由度的关节。

可以用类似于第 2.2 节的正向动力学的方法推导开链系统的混合动力学。让我们回到方程 (47)。与正向动力学的情况不同, 由于在方程 (47) 中对应于主动关节的 \ddot{q}_k 是已知的, 因此可以根据 $\dot{V}_{\lambda(i)}$ 将方程 (45) 代入方程 (47), 仅为被动关节的 \ddot{q}_k 来排列方程 (47), 如下所示:

$$F_{\lambda(i)} = \hat{\mathcal{I}}_{\lambda(i)} \dot{V}_{\lambda(i)} + \hat{\mathcal{B}}_{\lambda(i)} \quad (51)$$

其中

$$\begin{aligned}
\hat{\mathcal{I}}_{\lambda(i)} &= \mathcal{I}_{\lambda(i)} + \sum_{k \in \{\mu(\lambda(i)) \cap u\}} \text{Ad}_{T_{\lambda(i),k}^{-1}}^* \hat{\mathcal{I}}_k \text{Ad}_{T_{\lambda(i),k}^{-1}} \\
&+ \sum_{k \in \{\mu(\lambda(i)) \cap v\}} \text{Ad}_{T_{\lambda(i),k}^{-1}}^* \left\{ \hat{\mathcal{I}}_k - \hat{\mathcal{I}}_k S_k \left(S_k^T \hat{\mathcal{I}}_k S_k \right)^{-1} S_k^T \hat{\mathcal{I}}_k \right\} \text{Ad}_{T_{\lambda(i),k}^{-1}} \\
\hat{\mathcal{B}}_{\lambda(i)} &= -\text{ad}_{V_{\lambda(i)}}^* \mathcal{I}_{\lambda(i)} V_{\lambda(i)} - F_{\lambda(i)}^{\text{ext}} \\
&+ \sum_{k \in \{\mu(\lambda(i)) \cap u\}} \text{Ad}_{T_{\lambda(i),k}^{-1}}^* \left\{ \hat{\mathcal{B}}_k + \hat{\mathcal{I}}_k \left(\text{ad}_{V_k} S_k \dot{q}_k + \dot{S}_k \dot{q}_k + S_k \ddot{q}_k \right) \right\} \\
&+ \sum_{k \in \{\mu(\lambda(i)) \cap v\}} \text{Ad}_{T_{\lambda(i),k}^{-1}}^* \left\{ \hat{\mathcal{B}}_k + \hat{\mathcal{I}}_k \left(\text{ad}_{V_k} S_k \dot{q}_k + \dot{S}_k \dot{q}_k \right) \right. \\
&\left. + \hat{\mathcal{I}}_k S_k \left(S_k^T \hat{\mathcal{I}}_k S_k \right)^{-1} \left(\tau_k - S_k^T \hat{\mathcal{I}}_k \left(\text{ad}_{V_k} S_k \dot{q}_k + \dot{S}_k \dot{q}_k \right) - S_k^T \hat{\mathcal{B}}_k \right) \right\}.
\end{aligned} \tag{52}$$

其中“ u ”和“ v ”分别标志系统中规定的加速度和规定的扭矩。表 4 显示了开链系统的混合动力学算法。

3 受约束系统的动力学

设 $q \in \mathbb{R}^n$ 标志无约束开链系统中的一组关节坐标，并且 $A(q)\dot{q} = 0$ 表示对系统 ($A \in \mathbb{R}^{m \times n}$) 施加的一组 m 维约束。例如，对于闭环系统，可以通过切割关节环来获得 (虚拟的) 无约束开链系统，并且被强制执行的闭环约束可以写为 $f(q) = 0$ ，其可线性化为 $A\dot{q} = 0$ 的形式，其中 $A = \frac{\partial f}{\partial q}$ 是受约束的雅可比矩阵。

受约束系统的运动方程可被写为：

$$M\ddot{q} + b = \tau + A^T \lambda \tag{54}$$

$$A\dot{q} = 0 \tag{55}$$

其中方程 (54) 表示无约束开链系统的运动方程，附加约束力 $J^T \lambda$ 作用在构型空间 q 上，并且拉格朗日乘子 $\lambda \in \mathbb{R}^m$ 给出约束力的相对大小。

类似于上一节 (第 2 节)，我们将推导受约束系统的逆向动力学 ($\ddot{q} \rightarrow \tau$)，正向动力学 ($\tau \rightarrow \ddot{q}$) 和混合动力学 ($(\ddot{q}_u, \tau_v) \rightarrow (\tau_u, \ddot{q}_v)$)，其中 $q = (q_u, q_v)$ 。我们假设当前系统状态 (q, \dot{q}) 已知且与约束一致，并且 A 为满秩。

3.1 受约束系统的逆向动力学

当给定系统的当前状态 (q, \dot{q}) 和加速度 (\ddot{q}) 时，逆向动力学找到满足运动方程的关节力矩 (τ)。我们假定状态和规定加速度 (q, \dot{q}, \ddot{q}) 满足约束 $A\dot{q} = 0$ 及其微分 $A\ddot{q} + \dot{A}\dot{q} = 0$ ，并着重于求解在方程 (54) 中具有附加约束力的无约束系统的运动方程。

受约束系统的逆向动力学解可能不是唯一的，这取决于约束和执行器的数量。例如，当所有关节都可以驱动时，任意约束力 λ 及其相应的关节力矩 $\tau = M\ddot{q} + b - A^T \lambda$ 变为方程 (54) 的一个解，并且一个特定的解是 $\tau = M\ddot{q} + b$ 且 $\lambda = 0$ ，其中系统的行为如同无约束的开环系统。一般情况下，如果系统是冗余驱动或有太多的执行器，使系统以规定的方式移动所需的关节力矩是不唯一的，在

表 4: 递归混合动力学

```

while forward recursion do
   $T_{\lambda(i),i}$  = function of  $q_i$ 
   $V_i = \text{Ad}_{T_{\lambda(i),i}^{-1}} V_{\lambda(i)} + S_i \dot{q}_i$ 
   $\eta_i = \text{ad}_{V_i} S_i \dot{q}_i + \dot{S}_i \dot{q}_i$ 
end while
while backward recursion do
   $\hat{\mathcal{I}}_i = \mathcal{I}_i + \sum_{k \in \mu(i)} \text{Ad}_{T_{i,k}^{-1}}^* \Pi_k \text{Ad}_{T_{i,k}^{-1}}$ 
   $\hat{\mathcal{B}}_i = -\text{ad}_{V_i}^* \mathcal{I}_i V_i - F_i^{\text{ext}} + \sum_{k \in \mu(i)} \text{Ad}_{T_{i,k}^{-1}}^* \beta_k$ 
  if  $i \in u$  then
     $\Pi_i = \hat{\mathcal{I}}_i$ 
     $\beta_i = \hat{\mathcal{B}}_i + \hat{\mathcal{I}}_i (\eta_i + S_i \ddot{q}_i)$ 
  else
     $\Psi_i = (S_i^T \hat{\mathcal{I}}_i S_i)^{-1}$ 
     $\Pi_i = \hat{\mathcal{I}}_i - \hat{\mathcal{I}}_i S_i \Psi_i S_i^T \hat{\mathcal{I}}_i$ 
     $\beta_i = \hat{\mathcal{B}}_i + \hat{\mathcal{I}}_i \left\{ \eta_i + S_i \Psi_i \left( \tau_i - S_i^T (\hat{\mathcal{I}}_i \eta_i + \hat{\mathcal{B}}_i) \right) \right\}$ 
  end if
end while
while forward recursion do
  if  $i \in u$  then
     $\dot{V}_i = \text{Ad}_{T_{\lambda(i),i}^{-1}} \dot{V}_{\lambda(i)} + S_i \ddot{q}_i + \eta_i$ 
     $F_i = \hat{\mathcal{I}}_i \dot{V}_i + \hat{\mathcal{B}}_i$ 
     $\tau_i = S_i^T F_i$ 
  else
     $\ddot{q}_i = \Psi_i \left\{ \tau_i - S_i^T \hat{\mathcal{I}}_i \left( \text{Ad}_{T_{\lambda(i),i}^{-1}} \dot{V}_{\lambda(i)} + \eta_i \right) - S_i^T \hat{\mathcal{B}}_i \right\}$ 
     $\dot{V}_i = \text{Ad}_{T_{\lambda(i),i}^{-1}} \dot{V}_{\lambda(i)} + S_i \ddot{q}_i + \eta_i$ 
     $F_i = \hat{\mathcal{I}}_i \dot{V}_i + \hat{\mathcal{B}}_i$ 
  end if
end while

```

这种情况下，我们往往要选择一个最佳的解。相反，如果执行器的数量少于移动系统所需的最小值，就会像预期的那样没有解。

设 $q = (q_a, q_p)$ ，其中 q_a 和 q_p 分别标志驱动关节坐标和非驱动或被动关节坐标。被动关节力矩 τ_p 不必为零，但假定它们由系统状态确定。我们希望找到主动关节扭矩 (τ_a) 以满足方程 (54) 或

$$\begin{pmatrix} \tau_{ar} \\ \tau_{pr} \end{pmatrix} = \begin{pmatrix} \tau_a + A_a^T \lambda \\ \tau_p + A_p^T \lambda \end{pmatrix} \quad (56)$$

其中 $\tau_r = M\ddot{q} + b$ 是无约束开链系统的逆向动力学解，下标 “a” 和 “p” 分别标志主动坐标帧和被动坐标帧的分量，并且下标 “r” 表示无约束动力学解。由于我们已经知道 τ_p ，我们可以通过求解方程 (56) 的下一行或下式来获得约束力 λ ，

$$A_p^T \lambda = \tau_{pr} - \tau_p. \quad (57)$$

如果 $m = n_p$ ，其中 m 和 n_p 分别标志约束数量和非驱动坐标 ($A_p \in \mathbb{R}^{m \times n_p}$)，存在一个唯一的解 $\lambda = (A_p^T)^{-1} (\tau_{pr} - \tau_p)^T$ 。从方程 (56) 的上一行，主动关节扭矩可获得为

$$\tau_a = \tau_{ar} - A_a^T (A_p^T)^{-1} (\tau_{pr} - \tau_p). \quad (58)$$

例如，在闭环约束两个平移轴和一个旋转轴 ($m = 3$) 的平面五杆机构系统中，如果存在两个主动关节 (即五个关节中的三个为被动关节，并且 $n_p = 3$)，只要 A_p 为满秩或构形不奇异，就可以从方程 (58) 获得唯一的主动关节。

但是，如果 $m > n_p$ (例如，具有三个或更多主动关节的平面五杆机构)⁸，对于方程 (57) 就有无穷多个的解，可以写为

$$\lambda = (A_p^T)^\dagger (\tau_{pr} - \tau_p) + \mathcal{N}(A_p^T) \quad (59)$$

其中 $(\cdot)^\dagger$ 和 $\mathcal{N}(\cdot)$ 分别标志矩阵的 Moore-Penrose 伪逆和零空间，从而得出主动关节力矩的一般解，如下所示。

$$\tau_a = \tau_{ar} - A_a^T \lambda \quad (60)$$

$$= \tau_{ar} - A_a^T \left\{ (A_p^T)^\dagger (\tau_{pr} - \tau_p) + \mathcal{N}(A_p^T) \right\} \quad (61)$$

我们可以在这里选择最佳解决方案：

- 约束力最小化：其解变为 $\tau_a = \tau_{ar} - A_a^T \lambda^*$ ，其中 $\lambda^* = (A_p^T)^\dagger (\tau_{pr} - \tau_p)$ 。
- 最小主动关节力矩：我们可以通过在方程 (61) 中探索 A_p^T 的零空间找到它。例如，在 $n > m > n_p$ 的情况下，

$$\tau_a = \left(\mathbf{1} - A_a^T N (A_a^T N)^\dagger \right) \left\{ \tau_{ar} - A_a^T (A_p^T)^\dagger (\tau_{pr} - \tau_p) \right\} \quad (62)$$

$$\lambda = (A_p^T)^\dagger (\tau_{pr} - \tau_p) + N s^* \quad (63)$$

$$s^* = (A_a^T N)^\dagger \left\{ \tau_{ar} - A_a^T (A_p^T)^\dagger (\tau_{pr} - \tau_p) \right\} \quad (64)$$

⁷我们假设 A 为满秩。

⁸我们称之为冗余驱动系统。

最小化 $\|\tau_a\|^2$, 其中 $\mathbf{1} \in \mathbb{R}^{n_a \times n_a}$ 标志单位矩阵, 并且 $N \in \mathbb{R}^{m \times (m-n_p)}$ 表示 $A_p^T \in \mathbb{R}^{n_p \times m}$ 的零空间的基。作为这种情况的特例, 如果所有关节都被驱动 ($n > m, n_p = 0$), N 成为一个单位矩阵, 并因此最小关节力矩和相应的约束力可以写为

$$\tau = \tau_r - A^T \lambda \quad (65)$$

$$\lambda = (A^T)^\dagger \tau_r = (AA^T)^{-1} A \tau_r, \quad (66)$$

这也可以直接从方程 (56) 推导得出。在 $n < m$ 的情况下, $\tau_a = 0$ 为最小力矩解, 这意味着由于约束过多, 系统形成一个不可移动的结构。

在 $m < n_p$ 的情况下 (例如, 只有一个主动关节的平面五杆机构), 我们无法找到使系统精确地按规定 (q, \dot{q}, \ddot{q}) 运动的主动关节力矩。然而, 即使在这样一个系统中, 我们仍然可以找到一个主动关节力矩的解, 它必须适用于使主动关节精确地遵循规定的加速度 (\ddot{q}_a), 同时让其他被动关节以被动方式移动。为此参见受约束系统的混合动力学 (第 3.3 节)。

3.2 受约束系统的正向动力学

与求解依赖于系统性质和构型的约束系统的逆向动力学相反, 正向动力学找到响应于所施加关节力矩 (τ) 的结果加速度 (\ddot{q}) 总是有唯一的解。

通过微分受约束方程 (55) 以得到 $A\ddot{q} + \dot{A}\dot{q} = 0$, 并将其与方程 (54) 结合, 我们可以获得的约束力为

$$\lambda = - (AM^{-1}A^T)^{-1} (A\ddot{q}_r + \dot{A}\dot{q}) \quad (67)$$

其中 $\ddot{q}_r = M^{-1}(\tau - b)$ 是无约束开链系统的正向动力学解, 假设不存在约束 [2]。一旦从方程 (67) 获得约束力后, 通过求解考虑约束力的无约束开链系统方程 (54) 的正向动力学, 我们可以获得关节加速度。

为在方程 (67) 中提高计算效率, 我们可以使用在第 2.2 节中描述的开链系统的 $O(n)$ 递归正向动力学算法。递归算法不仅可用于获得 \ddot{q}_r , 而且可通过运行算法 m 次来计算 $M^{-1}A^T$ 。更具体地说, 我们通过调用正向动力学算法 (但忽略项 b , 即与重力和关节速度相关的项), 可以用 A^T 的第 i 列设置关节扭矩, 以计算 $M^{-1}A_i^T$, 并对所有列重复此操作。注意, 在重复过程中, 通过重用动力学算法中已计算的中间量, 可以节省大量计算。

我们可以通过将方程 (54) 拆分为两个方程来进一步节省计算量

$$M\ddot{q} + b = \tau + A^T \lambda \longrightarrow \begin{cases} M\ddot{q}_r + b = \tau \\ M\ddot{q}_c = A^T \lambda \end{cases} \quad (68)$$

并且这表明, 将无约束开链系统的两个解求和, 可以得到约束系统的正向动力学解, 即,

$$\ddot{q} = \ddot{q}_r + \ddot{q}_c \quad (69)$$

其中 \ddot{q}_r 是假定不存在约束的无约束开链系统的解, 并且 \ddot{q}_c 是将约束力施加到开链系统时忽略上述项 b 的解。总之, 我们需要一次完整版本的递归开链正向动力学算法 ($\ddot{q} = M^{-1}(\tau - b)$) 的调用, 以及 $m + 1$ 次简化版本的动力学算法 ($\ddot{q} = M^{-1}\tau$) 的调用。结果算法的复杂度变为 $O(n) + O(m^3)$, 其中 m 标志约束数量, $O(m^3)$ 用于求逆 $AM^{-1}A^T$ 。

3.3 受约束系统的混合动力学

设 $q = (q_u, q_v)$, 其中 q_u 和 q_v 分别是具有规定加速度和给定或已知扭矩的关节坐标。受约束系统的运动方程可以重写为

$$M \begin{pmatrix} \ddot{q}_u \\ \ddot{q}_v \end{pmatrix} + b = \begin{pmatrix} \tau_u \\ \tau_v \end{pmatrix} + A^T \lambda \quad (70)$$

$$A \begin{pmatrix} \ddot{q}_u \\ \ddot{q}_v \end{pmatrix} + \dot{A}\dot{q} = 0, \quad (71)$$

并且混合动力学算法通过求解受约束动力学方程, 从给定的 (\ddot{q}_u, τ_v) 找到 (τ_u, \ddot{q}_v) 。

我们首先证明约束力 λ 可以以类似于方程 (67) 的形式获得。从方程 (70) 我们得到

$$\ddot{q}_v = M_{vv}^{-1} (\tau_v - b_v - M_{uv}\ddot{q}_u + A_v^T \lambda) \quad (72)$$

其中 $M = \begin{bmatrix} M_{uu} & M_{uv} \\ M_{uv} & M_{vv} \end{bmatrix}$, $b = \begin{pmatrix} b_u \\ b_v \end{pmatrix}$ 并且 $A = \begin{bmatrix} A_u & A_v \end{bmatrix}$ 。将其代入方程 (71) 将导致

$$(A_v M_{vv}^{-1} A_v^T) \lambda = - \left\{ A_u \ddot{q}_u + A_v M_{vv}^{-1} (\tau_v - b_v - M_{uv}\ddot{q}_u) + \dot{A}\dot{q} \right\}. \quad (73)$$

如果 $m \leq n_v$, 其中 m 和 n_v 标志约束数量和扭矩指定坐标 ($A_v \in \mathbb{R}^{m \times n_v}$, $M_{vv} \in \mathbb{R}^{n_v \times n_v}$), 对于约束力 λ 就存在唯一解, 并且导致对于 (τ_u, \ddot{q}_v) 也存在唯一解⁹。在这种情况下, 约束力可通过以下方式获得。

$$\lambda = - (A_v M_{vv}^{-1} A_v^T)^{-1} \left\{ A_u \ddot{q}_u + A_v M_{vv}^{-1} (\tau_v - b_v - M_{uv}\ddot{q}_u) + \dot{A}\dot{q} \right\} \quad (74)$$

$$= - (A_v M_{vv}^{-1} A_v^T)^{-1} \left\{ A_u \ddot{q}_u + A_v \ddot{q}_{rv} + \dot{A}\dot{q} \right\} \quad (75)$$

$$= - (A_v M_{vv}^{-1} A_v^T)^{-1} (A \ddot{q}_r + \dot{A}\dot{q}) \quad (76)$$

其中

$$\ddot{q}_{rv} = M_{vv}^{-1} (\tau_v - b_v - M_{uv}\ddot{q}_u) \quad (77)$$

标志无约束开链系统中扭矩指定坐标的结果加速度, 并且 $\ddot{q}_r = (\ddot{q}_u, \ddot{q}_{rv})$ 标志无约束系统的加速度。注意, \ddot{q}_u 是规定的加速度, 并且通过使用在第 2.3 节中描述的递归算法可以有效地获得 \ddot{q}_{rv} 。

类似于正向动力学 (第 3.2 节) 的情况, 通过调用递归开链混合动力学算法的简化版本 m 次, 可以有效地获得 $M_{vv}^{-1} A_v^T$ 。更具体地说, 可以很容易地从方程 (77) 识别出, 当忽略与重力和关节速度相关的项时, $M_{vv}^{-1} A_{vi}^T$ 可以解释为作为扭矩指定的坐标的结果加速度, 并且将输入到无约束开链系统的关节命令设置为 $(\ddot{q}_u = 0, \tau_v = A_{vi}^T)$, 其中 A_{vi}^T 标志 A_v^T 的第 i 列。

一旦我们从方程 (76) 中获得约束力 λ , 我们就可以通过使用开链系统的递归混合动力学算法求解方程 (70) 获得 (τ_u, \ddot{q}_v) 。但是, 将方程 (70) 拆分为两个方程可以提高计算效率, 如我们在第 3.2 节中所讨论的。该方程可重写为

$$M \begin{pmatrix} \ddot{q}_u \\ \ddot{q}_v \end{pmatrix} + b = \begin{pmatrix} \tau_u \\ \tau_v \end{pmatrix} + A^T \lambda \quad \longrightarrow \quad \begin{cases} M \begin{pmatrix} \ddot{q}_u \\ \ddot{q}_{vr} \end{pmatrix} + b = \begin{pmatrix} \tau_{ur} \\ \tau_v \end{pmatrix} \\ M \begin{pmatrix} 0 \\ \ddot{q}_{vc} \end{pmatrix} = \begin{pmatrix} \tau_{uc} \\ A_v^T \lambda \end{pmatrix} \end{cases} \quad (78)$$

⁹再一次, 我们假设 A 为满秩。

其中 (\ddot{q}_u, τ_u) 和 $(0, A_v^T \lambda)$ 分别是无约束开链系统两个混合动力学问题的输入。这种拆分表明, 受约束系统的原始混合动力学问题的解可获得为

$$\tau_u = \tau_{ur} + \tilde{\tau}_{uc} \quad (79)$$

$$\ddot{q}_v = \ddot{q}_{vr} + \ddot{q}_{vc} \quad (80)$$

其中 $\tilde{\tau}_{uc} = \tau_{uc} - A_u^T \lambda$ 注意, 当 $q_v = q$ 时, 混合动力学的解与在第 3.2 节中的正向动力学的解完全匹配。

如果 $m > n_v$, 约束力不是唯一确定的, 这类似于逆向动力学当执行器多于所需 ($m > n_p$) 时的情况。在这种情况下, 如果 \ddot{q}_u 设置为任意值, 对于 \ddot{q}_v 有可能不存在满足约束方程 (71) 或 $A_v \ddot{q}_v = -A_u \ddot{q}_u - \dot{A} \dot{q}$ 的解。如果仔细设置 \ddot{q}_u , 使得存在满足约束条件的加速度, 则对于加速度指定坐标 (τ_u) 的关节力矩有无穷多个解, 并可通过探索 A_v^T 的零空间来选择最优解。逆向动力学的解方程 (61) 提供了一个更简单的公式, 其中对于混合动力学中的符号, 下标 “a” 和 “p” 必须分别替换为 “u” 和 “v”。

4 几何动力学的微分

4.1 基本导数

对于使用链式规则微分递归动力学算法, 下面的导数是有用的。

引理 3. 设 $p \in \mathfrak{R}$ 是一个任意标量变量, 并且 $T \in SE(3)$ 是 p 的一个函数。则

$$\frac{\partial}{\partial p} \text{Ad}_T = \text{ad}_{\frac{\partial T}{\partial p} T^{-1}} \text{Ad}_T. \quad (81)$$

证明. 设 ξ 为一个任意 $se(3)$ 。若将 Ad_T 视为一个 6×6 矩阵, 则

$$\frac{\partial}{\partial p} (\text{Ad}_T \xi) = \frac{\partial}{\partial p} (\text{Ad}_T) \xi + \text{Ad}_T \frac{\partial \xi}{\partial p}. \quad (82)$$

Ad_T 也可以看作是一个线性映射, $\text{Ad}_T : \xi \rightarrow T \xi T^{-1}$, 并在这种情况下,

$$\frac{\partial}{\partial p} (\text{Ad}_T \xi) = \frac{\partial}{\partial p} (T \xi T^{-1}) \quad (83)$$

$$= \frac{\partial T}{\partial p} \xi T^{-1} + T \frac{\partial \xi}{\partial p} T^{-1} + T \xi \frac{\partial T^{-1}}{\partial p} \quad (84)$$

$$= \frac{\partial T}{\partial p} T^{-1} T \xi T^{-1} - T \xi T^{-1} \frac{\partial T}{\partial p} T^{-1} + T \frac{\partial \xi}{\partial p} T^{-1} \quad (85)$$

$$= \text{ad}_{\frac{\partial T}{\partial p} T^{-1}} \text{Ad}_T \xi + \text{Ad}_T \frac{\partial \xi}{\partial p}. \quad (86)$$

由于 $\xi \in se(3)$ 是任意的, 可以从方程 (82) 和 (86) 中看出 $\frac{\partial}{\partial p} \text{Ad}_T = \text{ad}_{\frac{\partial T}{\partial p} T^{-1}} \text{Ad}_T$ 。□

推论 1. 设 $p \in \mathfrak{R}$ 是一个任意标量变量, 并且 $T_{\lambda(i), i} \in SE(3) : \{\lambda(i)\} \rightarrow \{i\}$ 。则

$$\frac{\partial}{\partial p} \text{Ad}_{T_{\lambda(i), i}}^{-1} = -\text{ad}_{\frac{\partial h_i}{\partial p}} \text{Ad}_{T_{\lambda(i), i}}^{-1} \quad (87)$$

$$\frac{\partial}{\partial p} \text{Ad}_{T_{\lambda(i), i}}^{*-1} = -\text{Ad}_{T_{\lambda(i), i}}^{*-1} \text{ad}_{\frac{\partial h_i}{\partial p}}^* \quad (88)$$

表 5: 递归逆向动力学的导数

while forward recursion do $\frac{\partial h_i}{\partial p} \triangleq T_{\lambda(i),i}^{-1} \frac{\partial T_{\lambda(i),i}}{\partial p}$ $\frac{\partial V_i}{\partial p} = \text{Ad}_{T_{\lambda(i),i}^{-1}} \frac{\partial V_{\lambda(i)}}{\partial p} - \text{ad}_{\frac{\partial h_i}{\partial p}} \text{Ad}_{T_{\lambda(i),i}^{-1}} V_{\lambda(i)} + \frac{\partial S_i}{\partial p} \dot{q}_i + S_i \frac{\partial \dot{q}_i}{\partial p}$ $\frac{\partial \dot{V}_i}{\partial p} = \text{Ad}_{T_{\lambda(i),i}^{-1}} \frac{\partial \dot{V}_{\lambda(i)}}{\partial p} - \text{ad}_{\frac{\partial h_i}{\partial p}} \text{Ad}_{T_{\lambda(i),i}^{-1}} \dot{V}_{\lambda(i)} + \frac{\partial S_i}{\partial p} \ddot{q}_i + S_i \frac{\partial \ddot{q}_i}{\partial p}$ $\quad + \text{ad}_{\frac{\partial V_i}{\partial p}} S_i \dot{q}_i + \text{ad}_{V_i} \left(\frac{\partial S_i}{\partial p} \dot{q}_i + S_i \frac{\partial \dot{q}_i}{\partial p} \right) + \frac{\partial \dot{S}_i}{\partial p} \dot{q}_i + \dot{S}_i \frac{\partial \dot{q}_i}{\partial p}$ end while while backward recursion do $\frac{\partial F_i}{\partial p} = \frac{\partial \mathcal{I}_i}{\partial p} \dot{V}_i + \mathcal{I}_i \frac{\partial \dot{V}_i}{\partial p} - \text{ad}_{\frac{\partial V_i}{\partial p}}^* \mathcal{I}_i V_i - \text{ad}_{V_i}^* \left(\frac{\partial \mathcal{I}_i}{\partial p} V_i + \mathcal{I}_i \frac{\partial V_i}{\partial p} \right) - \frac{\partial F_i^{\text{ext}}}{\partial p}$ $\quad + \sum_{k \in \mu(i)} \text{Ad}_{T_{i,k}^{-1}}^* \left(\frac{\partial F_k}{\partial p} - \text{ad}_{\frac{\partial h_k}{\partial p}}^* F_k \right)$ $\frac{\partial \tau_i}{\partial p} = \frac{\partial S_i}{\partial p}^T F_i + S_i^T \frac{\partial F_i}{\partial p}$ end while
--

其中 $\frac{\partial h_i}{\partial p} \in se(3)$ 被定义为

$$\frac{\partial h_i}{\partial p} = T_{\lambda(i),i}^{-1} \frac{\partial T_{\lambda(i),i}}{\partial p}. \quad (89)$$

推论 2. 如果 $p = q_i^k$, 其中 q_i^k 标志第 i 个关节的第 k 个坐标, 则

$$\frac{\partial h_i}{\partial p} = S_i^k \quad (90)$$

其中 $S_i^k \in se(3)$ 标志第 i 个关节的雅可比矩阵的第 k 列, $S_i \in (se(3) \times n_i)$ 。如果 $p \notin q_i = \{q_i^1, \dots, q_i^{n_i}\}$, 则

$$\frac{\partial h_i}{\partial p} = 0. \quad (91)$$

4.2 动力学的导数

通过应用链式规则于方程 (87) 和 (88), 逆向、正向和混合动力学的递归算法可以相对于一个任意标量变量 $p \in \mathbb{R}$ 进行微分。表 5 显示了递归逆向动力学的导数, 并且它可以用给定的 $\left(\frac{\partial q}{\partial p}, \frac{\partial \dot{q}}{\partial p}, \frac{\partial \ddot{q}}{\partial p}\right)$ 求解 $\frac{\partial \tau}{\partial p}$ 。在表 6 中给出了递归正向动力学的导数, 它用给定的 $\frac{\partial \tau}{\partial p}$ 来计算 $\frac{\partial \ddot{q}}{\partial p}$ 。递归混合动力学的导数, 用给定的 $\left(\frac{\partial q}{\partial p}, \frac{\partial \dot{q}}{\partial p}, \frac{\partial \ddot{q}_u}{\partial p}, \frac{\partial \tau_v}{\partial p}\right)$ 求解 $\left(\frac{\partial \tau_u}{\partial p}, \frac{\partial \ddot{q}_v}{\partial p}\right)$, 如表 7 所示。

为获得动力学的导数, 需要提前运行递归动力学相关的方程, 此外, 还需要以下的量

$$\frac{\partial S_i}{\partial p}, \frac{\partial \dot{S}_i}{\partial p}, \frac{\partial F_i^{\text{ext}}}{\partial p}, \frac{\partial \mathcal{I}_i}{\partial p}, \text{ and } \frac{\partial h_i}{\partial p} \quad (92)$$

其对于每个机体, 应在初始化步骤中正确设置。

动力学导数的算法是如此的通用, 以至于它们可以应用于对任意标量变量的运动方程的微分。应该注意的是, 通过巧妙地实现算法, 计算速度可以比原始实现快得多, 因为在某些情况下, 在方程 (92) 中的许多量为零。例如,

表 6: 递归正向动力学的导数

while forward recursion **do**

$$\frac{\partial h_i}{\partial p} \triangleq T_{\lambda(i),i}^{-1} \frac{\partial T_{\lambda(i),i}}{\partial p}$$

$$\frac{\partial V_i}{\partial p} = \text{Ad}_{T_{\lambda(i),i}^{-1}} \frac{\partial V_{\lambda(i)}}{\partial p} - \text{ad}_{\frac{\partial h_i}{\partial p}} \text{Ad}_{T_{\lambda(i),i}^{-1}} V_{\lambda(i)} + \frac{\partial S_i}{\partial p} \dot{q}_i + S_i \frac{\partial \dot{q}_i}{\partial p}$$

$$\frac{\partial \eta_i}{\partial p} = \text{ad}_{\frac{\partial V_i}{\partial p}} S_i \dot{q}_i + \text{ad}_{V_i} \left(\frac{\partial S_i}{\partial p} \dot{q}_i + S_i \frac{\partial \dot{q}_i}{\partial p} \right) + \frac{\partial \dot{S}_i}{\partial p} \dot{q}_i + \dot{S}_i \frac{\partial \dot{q}_i}{\partial p}$$

end while

while backward recursion **do**

$$\frac{\partial \hat{\mathcal{L}}_i}{\partial p} = \frac{\partial \mathcal{L}_i}{\partial p} + \sum_{k \in \mu(i)} \text{Ad}_{T_{i,k}^{-1}}^* \left\{ \frac{\partial \Pi_k}{\partial p} - \Pi_k \text{ad}_{\frac{\partial h_k}{\partial p}} - \left(\Pi_k \text{ad}_{\frac{\partial h_k}{\partial p}} \right)^T \right\} \text{Ad}_{T_{i,k}^{-1}}$$

$$\begin{aligned} \frac{\partial \hat{\mathcal{B}}_i}{\partial p} = & -\text{ad}_{\frac{\partial V_i}{\partial p}}^* \mathcal{L}_i V_i - \text{ad}_{V_i}^* \left(\frac{\partial \mathcal{L}_i}{\partial p} V_i + \mathcal{L}_i \frac{\partial V_i}{\partial p} \right) - \frac{\partial F_i^{\text{ext}}}{\partial p} \\ & + \sum_{k \in \mu(i)} \text{Ad}_{T_{i,k}^{-1}}^* \left(\frac{\partial \beta_k}{\partial p} - \text{ad}_{\frac{\partial h_k}{\partial p}}^* \beta_k \right) \end{aligned}$$

$$\frac{\partial \Psi_i}{\partial p} = -\Psi_i \left\{ S_i^T \frac{\partial \hat{\mathcal{L}}_i}{\partial p} S_i + \frac{\partial S_i}{\partial p}^T \hat{\mathcal{L}}_i S_i + \left(\frac{\partial S_i}{\partial p}^T \hat{\mathcal{L}}_i S_i \right)^T \right\} \Psi_i$$

$$\begin{aligned} \frac{\partial \Pi_i}{\partial p} = & \frac{\partial \hat{\mathcal{L}}_i}{\partial p} - \left\{ \hat{\mathcal{L}}_i S_i \frac{\partial \Psi_i}{\partial p} S_i^T \hat{\mathcal{L}}_i + \frac{\partial \hat{\mathcal{L}}_i}{\partial p} S_i \Psi_i S_i^T \hat{\mathcal{L}}_i + \left(\frac{\partial \hat{\mathcal{L}}_i}{\partial p} S_i \Psi_i S_i^T \hat{\mathcal{L}}_i \right)^T \right. \\ & \left. + \hat{\mathcal{L}}_i \frac{\partial S_i}{\partial p} \Psi_i S_i^T \hat{\mathcal{L}}_i + \left(\hat{\mathcal{L}}_i \frac{\partial S_i}{\partial p} \Psi_i S_i^T \hat{\mathcal{L}}_i \right)^T \right\} \end{aligned}$$

$$\begin{aligned} \frac{\partial \beta_i}{\partial p} = & \frac{\partial \hat{\mathcal{B}}_i}{\partial p} + \frac{\partial \hat{\mathcal{L}}_i}{\partial p} \left\{ \eta_i + S_i \Psi_i \left(\tau_i - S_i^T \left(\hat{\mathcal{L}}_i \eta_i + \hat{\mathcal{B}}_i \right) \right) \right\} \\ & + \hat{\mathcal{L}}_i \left\{ \frac{\partial \eta_i}{\partial p} + \left(\frac{\partial S_i}{\partial p} \Psi_i + S_i \frac{\partial \Psi_i}{\partial p} \right) \left(\tau_i - S_i^T \left(\hat{\mathcal{L}}_i \eta_i + \hat{\mathcal{B}}_i \right) \right) \right. \\ & \left. + S_i \Psi_i \left(\frac{\partial \tau_i}{\partial p} - \frac{\partial S_i}{\partial p} \left(\hat{\mathcal{L}}_i \eta_i + \hat{\mathcal{B}}_i \right) - S_i^T \left(\frac{\partial \hat{\mathcal{L}}_i}{\partial p} \eta_i + \hat{\mathcal{L}}_i \frac{\partial \eta_i}{\partial p} + \frac{\partial \hat{\mathcal{B}}_i}{\partial p} \right) \right) \right\} \end{aligned}$$

end while

while forward recursion **do**

$$\begin{aligned} \frac{\partial \ddot{q}_i}{\partial p} = & \frac{\partial \ddot{\Psi}_i}{\partial p} \left\{ \tau_i - S_i^T \hat{\mathcal{L}}_i \left(\text{Ad}_{T_{\lambda(i),i}^{-1}} \dot{V}_{\lambda(i)} + \eta_i \right) - S_i^T \hat{\mathcal{B}}_i \right\} \\ & + \Psi_i \left\{ \frac{\partial \tau_i}{\partial p} - \left(\frac{\partial S_i}{\partial p}^T \hat{\mathcal{L}}_i + S_i^T \frac{\partial \hat{\mathcal{L}}_i}{\partial p} \right) \left(\text{Ad}_{T_{\lambda(i),i}^{-1}} \dot{V}_{\lambda(i)} + \eta_i \right) \right. \\ & \left. - S_i^T \hat{\mathcal{L}}_i \left(\text{Ad}_{T_{\lambda(i),i}^{-1}} \frac{\partial \dot{V}_{\lambda(i)}}{\partial p} - \text{ad}_{\frac{\partial h_{\lambda(i)}}{\partial p}} \text{Ad}_{T_{\lambda(i),i}^{-1}} \dot{V}_{\lambda(i)} + \frac{\partial \eta_i}{\partial p} \right) \right. \\ & \left. - \frac{\partial S_i}{\partial p}^T \hat{\mathcal{B}}_i - S_i^T \frac{\partial \hat{\mathcal{B}}_i}{\partial p} \right\} \end{aligned}$$

$$\frac{\partial \dot{V}_i}{\partial p} = \text{Ad}_{T_{\lambda(i),i}^{-1}} \frac{\partial \dot{V}_{\lambda(i)}}{\partial p} - \text{ad}_{\frac{\partial h_i}{\partial p}} \text{Ad}_{T_{\lambda(i),i}^{-1}} \dot{V}_{\lambda(i)} + \frac{\partial S_i}{\partial p} \ddot{q}_i + S_i \frac{\partial \ddot{q}_i}{\partial p} + \frac{\partial \eta_i}{\partial p}$$

$$\frac{\partial F_i}{\partial p} = \frac{\partial \hat{\mathcal{L}}_i}{\partial p} \dot{V}_i + \hat{\mathcal{L}}_i \frac{\partial \dot{V}_i}{\partial p} + \frac{\partial \hat{\mathcal{B}}_i}{\partial p}$$

end while

表 7: 递归混合动力学的导数

while forward recursion do

$$\begin{aligned}\frac{\partial h_i}{\partial p} &\triangleq T_{\lambda(i),i}^{-1} \frac{\partial T_{\lambda(i),i}}{\partial p} \\ \frac{\partial V_i}{\partial p} &= \text{Ad}_{T_{\lambda(i),i}}^{-1} \frac{\partial V_{\lambda(i)}}{\partial p} - \text{ad}_{\frac{\partial h_i}{\partial p}} \text{Ad}_{T_{\lambda(i),i}}^{-1} V_{\lambda(i)} + \frac{\partial S_i}{\partial p} \dot{q}_i + S_i \frac{\partial \dot{q}_i}{\partial p} \\ \frac{\partial \eta_i}{\partial p} &= \text{ad}_{\frac{\partial V_i}{\partial p}} S_i \dot{q}_i + \text{ad}_{V_i} \left(\frac{\partial S_i}{\partial p} \dot{q}_i + S_i \frac{\partial \dot{q}_i}{\partial p} \right) + \frac{\partial \dot{S}_i}{\partial p} \dot{q}_i + \dot{S}_i \frac{\partial \dot{q}_i}{\partial p}\end{aligned}$$

end while

while backward recursion do

$$\begin{aligned}\frac{\partial \hat{\mathcal{I}}_i}{\partial p} &= \frac{\partial \mathcal{I}_i}{\partial p} + \sum_{k \in \mu(i)} \text{Ad}_{T_{i,k}}^* \left\{ \frac{\partial \Pi_k}{\partial p} - \Pi_k \text{ad}_{\frac{\partial h_k}{\partial p}} - \left(\Pi_k \text{ad}_{\frac{\partial h_k}{\partial p}} \right)^T \right\} \text{Ad}_{T_{i,k}}^{-1} \\ \frac{\partial \hat{\mathcal{B}}_i}{\partial p} &= -\text{ad}_{\frac{\partial V_i}{\partial p}}^* \mathcal{I}_i V_i - \text{ad}_{V_i}^* \left(\frac{\partial \mathcal{I}_i}{\partial p} V_i + \mathcal{I}_i \frac{\partial V_i}{\partial p} \right) - \frac{\partial F_i^{\text{ext}}}{\partial p} \\ &\quad + \sum_{k \in \mu(i)} \text{Ad}_{T_{i,k}}^* \left(\frac{\partial \beta_k}{\partial p} - \text{ad}_{\frac{\partial h_k}{\partial p}}^* \beta_k \right)\end{aligned}$$

if $i \in u$ then

$$\begin{aligned}\frac{\partial \Pi_i}{\partial p} &= \frac{\partial \hat{\mathcal{I}}_i}{\partial p} \\ \frac{\partial \beta_i}{\partial p} &= \frac{\partial \mathcal{B}_i}{\partial p} + \frac{\partial \hat{\mathcal{I}}_i}{\partial p} (\eta_i + S_i \ddot{q}_i) + \hat{\mathcal{I}}_i \left(\frac{\partial \eta_i}{\partial p} + \frac{\partial S_i}{\partial p} \dot{q}_i + S_i \frac{\partial \dot{q}_i}{\partial p} \right)\end{aligned}$$

else

$$\begin{aligned}\frac{\partial \Psi_i}{\partial p} &= -\Psi_i \left\{ S_i^T \frac{\partial \hat{\mathcal{I}}_i}{\partial p} S_i + \frac{\partial S_i}{\partial p}^T \hat{\mathcal{I}}_i S_i + \left(\frac{\partial S_i}{\partial p}^T \hat{\mathcal{I}}_i S_i \right)^T \right\} \Psi_i \\ \frac{\partial \Pi_i}{\partial p} &= \frac{\partial \hat{\mathcal{I}}_i}{\partial p} - \left\{ \hat{\mathcal{I}}_i S_i \frac{\partial \Psi_i}{\partial p} S_i^T \hat{\mathcal{I}}_i + \frac{\partial \hat{\mathcal{I}}_i}{\partial p} S_i \Psi_i S_i^T \hat{\mathcal{I}}_i + \left(\frac{\partial \hat{\mathcal{I}}_i}{\partial p} S_i \Psi_i S_i^T \hat{\mathcal{I}}_i \right)^T \right. \\ &\quad \left. + \hat{\mathcal{I}}_i \frac{\partial S_i}{\partial p} \Psi_i S_i^T \hat{\mathcal{I}}_i + \left(\hat{\mathcal{I}}_i \frac{\partial S_i}{\partial p} \Psi_i S_i^T \hat{\mathcal{I}}_i \right)^T \right\} \\ \frac{\partial \beta_i}{\partial p} &= \frac{\partial \mathcal{B}_i}{\partial p} + \frac{\partial \hat{\mathcal{I}}_i}{\partial p} \left\{ \eta_i + S_i \Psi_i \left(\tau_i - S_i^T (\hat{\mathcal{I}}_i \eta_i + \hat{\mathcal{B}}_i) \right) \right\} \\ &\quad + \hat{\mathcal{I}}_i \left\{ \frac{\partial \eta_i}{\partial p} + \left(\frac{\partial S_i}{\partial p} \Psi_i + S_i \frac{\partial \Psi_i}{\partial p} \right) \left(\tau_i - S_i^T (\hat{\mathcal{I}}_i \eta_i + \hat{\mathcal{B}}_i) \right) \right. \\ &\quad \left. + S_i \Psi_i \left(\frac{\partial \tau_i}{\partial p} - \frac{\partial S_i}{\partial p} (\hat{\mathcal{I}}_i \eta_i + \hat{\mathcal{B}}_i) - S_i^T \left(\frac{\partial \hat{\mathcal{I}}_i}{\partial p} \eta_i + \hat{\mathcal{I}}_i \frac{\partial \eta_i}{\partial p} + \frac{\partial \hat{\mathcal{B}}_i}{\partial p} \right) \right) \right\}\end{aligned}$$

end if

end while

while forward recursion do

if $i \in u$ then

$$\begin{aligned}\frac{\partial \dot{V}_i}{\partial p} &= \text{Ad}_{T_{\lambda(i),i}}^{-1} \frac{\partial \dot{V}_{\lambda(i)}}{\partial p} - \text{ad}_{\frac{\partial h_i}{\partial p}} \text{Ad}_{T_{\lambda(i),i}}^{-1} \dot{V}_{\lambda(i)} + \frac{\partial S_i}{\partial p} \dot{q}_i + S_i \frac{\partial \dot{q}_i}{\partial p} + \frac{\partial \eta_i}{\partial p} \\ \frac{\partial F_i}{\partial p} &= \frac{\partial \hat{\mathcal{I}}_i}{\partial p} \dot{V}_i + \hat{\mathcal{I}}_i \frac{\partial \dot{V}_i}{\partial p} + \frac{\partial \hat{\mathcal{B}}_i}{\partial p} \\ \frac{\partial \tau_i}{\partial p} &= \frac{\partial S_i}{\partial p}^T F_i + S_i^T \frac{\partial F_i}{\partial p}\end{aligned}$$

else

$$\begin{aligned}\frac{\partial \ddot{q}_i}{\partial p} &= \frac{\partial \Psi_i}{\partial p} \left\{ \tau_i - S_i^T \hat{\mathcal{I}}_i \left(\text{Ad}_{T_{\lambda(i),i}}^{-1} \dot{V}_{\lambda(i)} + \eta_i \right) - S_i^T \hat{\mathcal{B}}_i \right\} \\ &\quad + \Psi_i \left\{ \frac{\partial \tau_i}{\partial p} - \left(\frac{\partial S_i}{\partial p}^T \hat{\mathcal{I}}_i + S_i^T \frac{\partial \hat{\mathcal{I}}_i}{\partial p} \right) \left(\text{Ad}_{T_{\lambda(i),i}}^{-1} \dot{V}_{\lambda(i)} + \eta_i \right) \right. \\ &\quad \left. - S_i^T \hat{\mathcal{I}}_i \left(\text{Ad}_{T_{\lambda(i),i}}^{-1} \frac{\partial \dot{V}_{\lambda(i)}}{\partial p} - \text{ad}_{\frac{\partial h_i}{\partial p}} \text{Ad}_{T_{\lambda(i),i}}^{-1} \dot{V}_{\lambda(i)} + \frac{\partial \eta_i}{\partial p} \right) \right. \\ &\quad \left. - \frac{\partial S_i}{\partial p}^T \hat{\mathcal{B}}_i - S_i^T \frac{\partial \hat{\mathcal{B}}_i}{\partial p} \right\}\end{aligned}$$

$$\begin{aligned}\frac{\partial \dot{V}_i}{\partial p} &= \text{Ad}_{T_{\lambda(i),i}}^{-1} \frac{\partial \dot{V}_{\lambda(i)}}{\partial p} - \text{ad}_{\frac{\partial h_i}{\partial p}} \text{Ad}_{T_{\lambda(i),i}}^{-1} \dot{V}_{\lambda(i)} + \frac{\partial S_i}{\partial p} \dot{q}_i + S_i \frac{\partial \dot{q}_i}{\partial p} + \frac{\partial \eta_i}{\partial p} \\ \frac{\partial F_i}{\partial p} &= \frac{\partial \hat{\mathcal{I}}_i}{\partial p} \dot{V}_i + \hat{\mathcal{I}}_i \frac{\partial \dot{V}_i}{\partial p} + \frac{\partial \hat{\mathcal{B}}_i}{\partial p}\end{aligned}$$

end if

end while

- 如果 $p = q^k$, 其中 $q^k \in \mathfrak{R}$ 标志系统的第 k 个坐标, 则

$$\begin{aligned} -\frac{\partial \mathcal{I}_i}{\partial p} &= 0 \\ -\frac{\partial S_i}{\partial p} &= \frac{\partial \dot{S}_i}{\partial p} = \frac{\partial h_i}{\partial p} = 0 \text{ when } q^k \notin q_i \end{aligned}$$

- 如果 $p = \dot{q}^k$, 则

$$\begin{aligned} -\frac{\partial S_i}{\partial p} &= \frac{\partial \mathcal{I}_i}{\partial p} = \frac{\partial h_i}{\partial p} = 0 \\ -\frac{\partial \dot{S}_i}{\partial p} &= 0 \text{ when } q^k \notin q_i \end{aligned}$$

- 如果 $p = \ddot{q}^k$, 则

$$-\frac{\partial \dot{S}_i}{\partial p} = \frac{\partial S_i}{\partial p} = \frac{\partial \mathcal{I}_i}{\partial p} = \frac{\partial h_i}{\partial p} = 0.$$

References

1. Roy Featherstone, Robot Dynamics Algorithms, Kluwer Academic Publishers, 1987.
2. Richard M. Murray, Zexiang Li and S. Shankar Sastry, A Mathematical Introduction to Robotic Manipulation, CRC Press, 1994.
3. F. C. Park, J. E. Bobrow and S. R. Ploen, "A Lie group formulation of robot dynamics," International Journal of Robotics Research, vol. 14, no. 6, pp. 609-618, 1995.
4. S. R. Ploen and F. C. Park, "Coordinate-invariant algorithms for robot dynamics," IEEE Transactions on Robotics and Automation, vol. 15, no. 6, pp. 1130-1136, 1999.
5. Garrett A. Sohl and James E. Bobrow, "A recursive multibody dynamics and sensitivity algorithm for branched kinematic chains," Journal of Dynamic Systems, Measurement, and Control, vol. 123, pp. 391-399, 2001.