机器人动力学：方程与算法

Roy Featherstone & David Orin

24-28 April 2000

摘要

本文回顾了机器人动力学研究领域的一些成就，从递归牛顿-欧拉算法的发展到今天的成就。本文给出了最重要的动力学计算的方程与算法，并以通用符号表达，以方便它们的表述和比较。

1 简介

自从二十多年前最早的工作以来，机器人动力学领域已经做出了许多贡献。在机构动力学领域，机器人学界特别关注计算效率的问题。事实上，许多动力学中最有效的算法，适用于广泛的机制，都是由机器人研究人员开发的 [23,33,10]。

虽然计算效率对于以更高速度运行的日益复杂的机构的模拟和控制仍然很重要，但动力学问题的其它方面也很重要。算法应该用一组紧凑的方程来表示，以便于开发和实现。另一方面，这些方程与从中获得最大计算效率的递归集之间应该有明确的关系。在这方面，空间符号和空间算子代数 [11,29] 的使用非常有效。另外，对于具有一般几何形状和关节结构的机器人机构，开发对其具有适用性的算法也很重要。虽然早期的算法 [23,33,10] 适用于具有旋转关节或棱柱关节的单一开链式机械手，但通用关节模型已被开发出来并应用于更复杂的构型 [11]。

本文的目的是回顾机器人动力学中的一些主要贡献。最重要的递归计算的方程和算法将在第 3 节中以一种通用、简洁的符号表示。如前所述，这些算法直接适用于树状结构的机制。然后，闭环系统的方程将以紧凑的形式推导，接着是对全局分析技术的讨论。由于篇幅所限，我们无法将该领域的所有重要贡献的参考文献包括在内。尽管如此，被引用最多的论文一般都包括在内。

2 机器人动力学的基础工作

早期的机器人动力学研究致力于以最有效的形式表达机器人操纵器和其它单一开链式系统的运动方程。为机器人分析、控制和模拟的最常见计算开发了算法。在本节中，重点将放在概述主要贡献和最常被引用的工作上。不幸的是，空间不允许我们对该领域的大量文献进行全面回顾。更多的参考文献见该领域的一些早期书籍 [7,11]。

表达运动方程的经典方法是基于该问题的拉格朗日公式 [18,31]。使用拉格朗日动力学开发的算法是 $O(N^4)$，并且必须适应实时控制的需要。高效的低阶算法被用于三个主要的计算。

1. 逆向动力学，其中所需的关节执行器扭矩/动力是根据操纵器的轨迹 (位置、速度和加速度) 的规范计算出来的，

2. 正向动力学，其中规范应用的关节执行器扭矩/动力，并确定关节加速度，以及

3. 操纵器惯量矩阵，将关节加速度映射到关节力上。

逆向动力学用于前馈控制，而正向动力学则需要用于仿真。惯量 (质量) 矩阵在分析中使用，在反馈控制中用于线性化动力学，并且是许多正向动力学公式的一个组成部分。

第一批开发机器人逆向动力学 $O(N)$ 算法的研究人员使用求解问题的牛顿-欧拉 (NE) 公式。Stepanenko 和 Vukobratovic [30] 开发了一种用于人类肢体动力学的递归 NE 方法，Orin 等人 [26] 通过将动力和力矩引用到局部连杆坐标，从而使递归方法更加高效，以便实时控制行走机器的一条腿。Luh、Walker 和 Paul [23] 通过将大多数的量引用到连杆坐标，开发了一个非常高效的递归算法 (RNEA)。RNEA 是被引用最多的。Hollerbach [16] 开发了一个 $O(N)$ 递归拉格朗日公式，但发现就算法中所需要的乘法和加/减法的数量而言，其效率远低于 RNEA。这些年来，在效率方面取得了进一步的效率。Balafoutis 等人 [3] 以及 He 和 Goldenberg [15] 的结果具有代表性，他们比 RNEA 的早期实现快了 1.7 倍 (对于 6-DoF 机器人)。

Walker 和 Orin [33] 将 RNEA 用于逆向动力学 [23]，作为正向动力学的高效算法的基础。他们的方法 3，后来被 Featherstone [11] 命名为复合刚体算法 (CRBA)，用于计算操纵器链外端的复合刚体集的惯量参数。惯量矩阵的列通过连续应用逆向动力学来非常有效地计算，其中关节速度设置为零，并且关节加速度设置为零或一个单位向量。由于这意味着每次只有一个关节在运动，逆向动力学就简化为对处于静态平衡的一组基础连杆，以及处于链条外端运动的一个复合刚体的一个非常简单的分析。由于需要求解一个线性方程组，其大小随 N 的增长而增长，因此该算法是 $O(N^3)$。对于小的 N，一阶项主导了计算，因此结果非常有效。

最早的已知 $O(N)$ 的正向动力学算法是由 Vereschagin [32] 开发的。该算法使用一个递归公式来评估运动方程的 Gibbs-Appel 形式，并适用于具有旋转关节和棱柱关节的无支链。递归公式通过动态规划技术获得。该算法与铰接体算法 (ABA) 非常相似，但这篇论文远远领先于它的时代，并在十年内一直默默无闻。后来，Armstrong 为球形关节的机构开发了一种 $O(N)$ 算法 [1]，然后 Featherstone 开发了 ABA [10]。该算法的第一个版本适用于具有单自由度关节的机械手，但第二个版本包括一个通用关节模型，而且速度更快 [11]。就所需的算术运算总数而言，对于 $N > 9$，ABA 比 CRBA 更有效 [11]。另外，Brandl 等人 [8] 使用与 Featherstone [11,10] 类似的有效变换和连杆坐标，对 ABA 做了进一步的改进，使其在 $N = 6$ 时与 CRBA 大致相当。多年来，在效率方面有了进一步的提高，McMillan 和 Orin [24] 是减少计算量的代表 (又减少了 15%)。

CRBA 的效率与计算关节空间惯量 (质量) 矩阵的效率直接相关 [33]。Featherstone [11] 使用有效的变换和连杆坐标，将惯量矩阵的计算减少了约 30%。多年来还取得了一些其它的进展 [4,25]，总体上比文献 [33] 有了接近 2 倍的改进。Lilly 和 Orin [22] 开发了四种计算惯量矩阵的方法。他们改进的复合刚体方法包括操纵器 Jacobian 矩阵的计算，因此它对计算操作空间的惯量矩阵非常有效。

Khatib [20] 开发了一种机器人动力学的操作空间公式，其中的方程是在用于指挥机器人的同一坐标系统中表达：笛卡尔坐标和末端执行器的方向。该公式在混合运动/动力控制和相关应用中特别成功 [21]。

Rodriguez [28] 认识到卡尔曼滤波的概念和技术与正向动力学问题之间的相似性，并为多体动力学的研究开发了空间算子代数框架。这使他和 JPL [29] 的其他人能够开发出质量矩阵的替代因子分解，从而推导出 ABA。Jain [17] 使用空间算子代数框架为操纵器动力学提供了一个统一的公式。由此，他能够比较之前发表的各种 $O(N^3)$、$O(N^2)$ 和 $O(N)$ 算法。Ascher, Pai 和 Cloutier [2] 使用空间算子框架来统一 CRBA 和 ABA 的推导，作为解决同一线性系统的两种消除方法。他们还表明，ABA 比 CRBA 更准确。

上述工作都与刚体动力学有关，并且因此适用于任意可以由刚体系统对机器人机构进行充分建模的情况。某些形式的非刚体行为，如关节轴承的顺应性，相对容易纳入刚体模型；但弹性连杆则更为复杂。Book [6] 解决了这个问题，他开发了一个高效的、递归的拉格朗日公式 (使用 4×4 矩阵)，用于具有弹性连杆的串行链的逆向和正向动力学。使用了弹性位移的通用模态公式。

3 方程与算法

本节介绍主要算法和相关方程。为简洁起见，方程式用空间符号表示；但不熟悉这种符号的读者应该仍然能够理解这些材料。

空间向量是 6×1 向量，包含速度、加速度和动力等物理量的线性和角度分量。出于技术原因，它们被分为两个向量空间：M^6 中的运动类型向量和 F^6 中的动力类型向量。张量，如惯量，用 6×6 矩阵表示。基础代数是向量空间的对偶系统。文献 [12] 的附录中对当前版本的空间代数进行了简要描述，而在文献 [11] 中对先前版本进行了详细描述。

3.1 递归牛顿-欧拉算法

具有树状结构的通用机器人机构可以由一组编号为 $1\dots N$ 的 N 个可移动连杆 (刚体)、一个编号为 0 的固定基座连杆和一组连接在连杆之间的 N 个关节建模，以便关节 i 从连杆 $\lambda(i)$ 连接到连杆 i，其中 $\lambda(i)$ 是树中连杆 i 的父级的连杆编号，将基座连杆作为根节点。选择的数字应确保 $\lambda(i)<i$。在无支链运动链的特殊情况下，$\lambda(i)=i-1$，连杆和接头从底部到尖端连续编号。

如果我们设 $\mathbf{v}_i$ 是连杆 i 的速度，并且 $\mathbf{v}_i^J$ 是跨过关节 i 的速度，则

$$\mathbf{v}_i^J=\mathbf{v}_i-\mathbf{v}_{\lambda(i)}. \tag{1}$$

关节速度也可以用以下形式描述

$$\mathbf{v}_i^J=\mathbf{h}_i\dot{\mathbf{q}}_i, \tag{2}$$

其中，$\mathbf{h}_i$ 是一个 $6\times d_i$ 矩阵，其张成第 i 个关节的运动自由度子空间，$\dot{\mathbf{q}}_i$ 是一个 $d_i\times1$ 的关节自由度变量向量，d_i 是第 i 个关节的自由度 (DoF)。在 1-DoF 关节的特殊情况下，$\mathbf{h}_i$ 是一个描述关节运动轴的向量。

结合方程 (1) 和 (2) 产生

$$\mathbf{v}_i=\mathbf{v}_{\lambda(i)}+\mathbf{h}_i\dot{\mathbf{q}}_i, \tag{3}$$

这是计算连杆速度的标准递归公式。加速度的等效公式只是方程 (3) 的时间导数：

$$\mathbf{a}_i=\mathbf{a}_{\lambda(i)}+\dot{\mathbf{h}}_i\dot{\mathbf{q}}_i+\mathbf{h}_i\ddot{\mathbf{q}}_i, \tag{4}$$

其中，$\mathbf{a}_i$ 是连杆 i 的加速度，$\ddot{\mathbf{q}}_i$ 是关节加速度变量的向量。[1]对于旋转关节和棱柱关节以及许多其他特殊情况，

$$\dot{\mathbf{h}}_i=\mathbf{v}_i\times\mathbf{h}_i.$$

给定基座的速度和加速度，$\mathbf{v}_0$ 和 $\mathbf{a}_0$，这些公式依次计算每个连杆的速度和加速度，从基座向外延伸到终端连杆。典型的算法如下所示：

[1]该方程使用空间加速度。大多数已发表的作品使用标准线性加速度和角加速度来代替，或者单独使用 (如文献 [23])，或者使用 6-D 表示法 (如空间算子代数)。这种差异在文献 [11,§2.7] 中进行了解释。使用非空间加速度的论文在方程 (4) 和 (5) 中有不同的表达。

```
for i = 1 to N do
  v_i = v_λ(i) + h_i q̇_i;
  a_i = a_λ(i) + ḣ_i q̇_i + h_i q̈_i;
end for
```

性质 $\lambda(i) < i$ 确保 $\mathbf{v}_{\lambda(i)}$ 在 $\mathbf{v}_i$ 之前计算。

连杆 i 的运动方程为

$$\mathbf{f}_i + \mathbf{f}_i^x = \mathbf{I}_i\mathbf{a}_i + \mathbf{v}_i \times \mathbf{I}_i\mathbf{v}_i, \tag{5}$$

其中，$\mathbf{I}_i$ 是连杆 i 的空间惯量 (6×6 矩阵)，$\mathbf{f}_i$ 是通过关节施加到连杆 i 的净动力，并且 $\mathbf{f}_i^x$ 是作用在连杆 i 上的所有其它力的总和。该方程结合了刚体线性运动和角向运动的牛顿方程和欧拉方程。

$\mathbf{f}_i^x$ 可能包括来自弹簧、阻尼器、力场、与环境的接触等因素的贡献；但假设其值已知，或者至少是可以从已知量中计算出来的 (在这种情况下，$\mathbf{f}_i^x$ 的计算被认为是一个单独的问题)。$\mathbf{f}_i^x$ 也可以包括重力对连杆 i 的影响；但是均匀重力场的特殊情况可以通过给基座施加一个虚拟加速度来进行最有效地模拟：如果 $\mathbf{g}$ 是重力加速度向量，则将 $-\mathbf{g}$ 加至 $\mathbf{a}_0$ [23]。

如果我们把 $\mathbf{f}_i^J$ 定义为从连杆 $\lambda(i)$ 通过关节 i 传递给连杆 i 的动力，则

$$\mathbf{f}_i = \mathbf{f}_i^J - \sum_{j\in\mu(i)} \mathbf{f}_j^J \tag{6}$$

其中 $\mu(i)$ 是连杆 i 的子级集合：

$$\mu(i) = \{j \mid \lambda(j) = i\}.$$

结合方程 (5) 和 (6) 产生

$$\mathbf{f}_i^J = \mathbf{I}_i\mathbf{a}_i + \mathbf{v}_i \times \mathbf{I}_i\mathbf{v}_i - \mathbf{f}_i^x + \sum_{j\in\mu(i)} \mathbf{f}_j^J, \tag{7}$$

这是一个递归公式，用于计算关节力，从末端连杆开始，朝着基座工作。此公式的典型实现如下所示：

```
for i = 1 to N do
  f_i^J = I_i a_i + v_i × I_i v_i − f_i^x
end for
for i = N to 1 do
  if λ(i) ≠ 0 then
    f_λ(i)^J = f_λ(i)^J + f_i^J
  end if
end for
```

最后一步是从关节力的空间向量中提取关节力变量的 $d_i \times 1$ 向量 $\boldsymbol{\tau}_i$。这是由以下步骤完成

$$\boldsymbol{\tau}_i = \mathbf{h}_i^{\mathrm{T}}\mathbf{f}_i^J. \tag{8}$$

方程 (3)、(4)、(7) 和 (8) 共同构成了通用树状结构机制的 RNEA。

为获得最大效率，应在连杆坐标中计算这些方程。如果我们为每个连杆指定一个坐标系统，并在其所指连杆的坐标中表示每个空间向量，则方程 (3)、(4) 和 (7) 需要在表达式中的适当位置包含

坐标变换。修改后的方程为

$$\begin{aligned}\mathbf{v}_i &= {}^i\mathbf{X}^M_{\lambda(i)}\mathbf{v}_{\lambda(i)} + \mathbf{h}_i\dot{\mathbf{q}}_i,\\ \mathbf{a}_i &= {}^i\mathbf{X}^M_{\lambda(i)}\mathbf{a}_{\lambda(i)} + \dot{\mathbf{h}}_i\dot{\mathbf{q}}_i + \mathbf{h}_i\ddot{\mathbf{q}}_i\end{aligned}$$

并且

$$\mathbf{f}^J_i = \mathbf{I}_i\mathbf{a}_i + \mathbf{v}_i \times \mathbf{I}_i\mathbf{v}_i + \sum_{j\in\mu(i)} {}^i\mathbf{X}^F_j\mathbf{f}^J_j,$$

其中，${}^i\mathbf{X}^M_{\lambda(i)}$ 和 ${}^i\mathbf{X}^F_j$ 分别是运动类型向量和动力类型向量的坐标变换矩阵。

完整的牛顿-欧拉算法，在连杆坐标中，看起来像这样：

for $i = 1$ **to** N **do**
  $\mathbf{v}_i = {}^i\mathbf{X}^M_{\lambda(i)}\mathbf{v}_{\lambda(i)} + \mathbf{h}_i\dot{\mathbf{q}}_i;$
  $\mathbf{a}_i = {}^i\mathbf{X}^M_{\lambda(i)}\mathbf{a}_{\lambda(i)} + \dot{\mathbf{h}}_i\dot{\mathbf{q}}_i + \mathbf{h}_i\ddot{\mathbf{q}}_i;$
  $\mathbf{f}^J_i = \mathbf{I}_i\mathbf{a}_i + \mathbf{v}_i \times \mathbf{I}_i\mathbf{v}_i - \mathbf{f}^x_i$
end for
for $i = N$ **to** 1 **do**
  $\boldsymbol{\tau}_i = \mathbf{h}^{\mathrm{T}}_i\mathbf{f}^J_i;$
  if $\lambda(i) \neq 0$ **then**
    $\mathbf{f}^J_{\lambda(i)} = \mathbf{f}^J_{\lambda(i)} + {}^{\lambda(i)}\mathbf{X}^F_i\mathbf{f}^J_i$
  end if
end for

3.2 复合刚体算法

正如在文献 [33,11] 中所解释的，如果我们将树状结构刚体系统的运动方程表示为

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

其中 $\ddot{\mathbf{q}} \in \mathrm{M}^n$ 是广义加速度向量，$\boldsymbol{\tau} \in \mathrm{F}^n$ 是广义动力向量，$\mathbf{M} : \mathbf{M}^n \mapsto \mathrm{F}^n$ 是系统质量矩阵 (关节空间惯量矩阵)，$\mathbf{C} \in \mathrm{F}^n$ 包含所有与加速度无关的项，则

$$\begin{aligned}\mathbf{M}\ddot{\mathbf{q}} &= \mathbf{D}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}}) - \mathbf{D}(\mathbf{q},\dot{\mathbf{q}},\mathbf{0})\\ &= \mathbf{D}(\mathbf{q},\mathbf{0},\ddot{\mathbf{q}}) - \mathbf{D}(\mathbf{q},\mathbf{0},\mathbf{0}),\end{aligned}$$

其中 $\mathbf{D}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})$ 是逆向动力学计算函数。速度参数可以设置为零，因为速度项会取消。(重力和 $\mathbf{f}^x$ 项也会取消。) 该方程立即为我们提供了计算 $\mathbf{M}$ 的简单算法：

$$\mathbf{M}\boldsymbol{\delta}_i = \mathbf{D}(\mathbf{q},\mathbf{0},\boldsymbol{\delta}_i) - \mathbf{D}(\mathbf{q},\mathbf{0},\mathbf{0}), \quad i = 1\ldots n,$$

其中，$\boldsymbol{\delta}_i$ 是一个 $n \times 1$ 向量，第 i 行为 1，其它地方为 0。表达式 $\mathbf{M}\boldsymbol{\delta}_i$ 是 $\mathbf{M}$ 的第 i 列。该算法为文献 [33] 中的方法 1。

我们可以从这个方程中推断出 $\mathbf{M}$ 的以下物理解释：在一个所有的速度和与加速度无关的力都为零的系统中，$\mathbf{M}$ 的 i 列是导致加速度 $\boldsymbol{\delta}_i$ 的广义动力向量。

如果每个关节都有一个自由度，则关节编号和列索引编号之间存在 $1:1$ 的对应关系；我们可以将 $\boldsymbol{\delta}_i$ 解释为关节 i 的单位加速度，而 $\mathbf{M}$ 的每个单独元素 $\mathbf{M}_{ji}$ 是关节 j 处产生加速度 $\boldsymbol{\delta}_i$ 所需的动力。

如果系统包含多个自由度的关节，则我们将 $\mathbf{M}$ 视为块矩阵 — 其元素本身为矩阵的矩阵，并将块列 i 安排为对应于关节 i 的 d_i 加速度变量的 d_i 实数列组。对 $\mathbf{M}$ 的行与组合向量，如 $\ddot{\mathbf{q}}$ 和 $\boldsymbol{\tau}$，也进行类似处理。这允许我们对单个自由度和多个自由度的关节使用相同的符号。

CRBA 是一种计算 $\mathbf{M}$ 的快速算法，它从以下观察结果开始：如果上述零速度系统的加速度为 $\boldsymbol{\delta}_i$，则以连杆 i 为根的整个子树的行为就像一个单一 (复合) 刚体，加速度等于 $\mathbf{h}_i$，系统的其余部分处于静态平衡。因此，$\mathbf{M}$ 的 (块) 列 i 的元素可以根据以下算法计算。

1. 设 C_i 为复合刚体，包括以连杆 i 为根的子树中的所有连杆，并设 $\mathbf{I}_i^C$ 为 C_i 的惯量。

2. 设 $\mathbf{f}_i^C$ 为向 C_i 施加 $\mathbf{h}_i$ 加速度所需的力。该力通过关节 i 传递给 C_i；但是，由于连杆 $\lambda(i)$ 处于静态平衡状态，这意味着没有净力作用于其上，通过连杆 $\lambda(i)$ 传递的力必须与通过连杆 i 传递的力相同。通过相同的参数，基座和连杆 i 之间路径上的每个关节传递 $\mathbf{f}_i^C$。

3. 每个元件 $\mathbf{M}_{ji}$，其中 j 是基座和连杆 i 之间路径上的关节，由 $\mathbf{M}_{ji} = \mathbf{h}_j^{\mathrm{T}} \mathbf{f}_i^C$ 给出 (参见方程 (8))。

4. 每个元素 $\mathbf{M}_{ji}$，其中 $j < i$，但不在基座和连杆 i 之间的路径上，为零。

5. 元素 $\mathbf{M}_{ji}$，其中 $j > i$，由 $\mathbf{M}_{ji} = \mathbf{M}_{ij}^{\mathrm{T}}$ 给出，因为 $\mathbf{M}$ 是对称的。

复合刚体的惯量是其组成部分惯量的简单总和，因此

$$\mathbf{I}_i^C = \sum_{j \in \nu(i)} \mathbf{I}_j,$$

其中，$\nu(i)$ 是子树中每个连杆的连杆编号集合；但是使用递归公式更有效率

$$\mathbf{I}_i^C = \mathbf{I}_i + \sum_{j \in \mu(i)} \mathbf{I}_j^C. \tag{9}$$

施加在 C_i 上的动力为

$$\mathbf{f}_i^C = \mathbf{I}_i^C \mathbf{h}_i,$$

并且通过关节 $j\,(j \leq i)$ 传递的动力为

$$\mathbf{f}_i^C(j) = \begin{cases} \mathbf{f}_i^C & \text{if } i \in \nu(j) \\ \mathbf{0} & \text{otherwise;} \end{cases}$$

所以 $\mathbf{M}$ 的完整表达式是

$$\mathbf{M}_{ji} = \begin{cases} \mathbf{h}_j^{\mathrm{T}} \mathbf{I}_i^C \mathbf{h}_i & \text{if } i \in \nu(j) \\ \mathbf{h}_j^{\mathrm{T}} \mathbf{I}_j^C \mathbf{h}_i & \text{if } j \in \nu(i) \\ \mathbf{0} & \text{otherwise.} \end{cases} \tag{10}$$

方程 (9) 和 (10) 以绝对坐标表示 CRBA。该算法在连杆坐标中运行得更快 (除非 n 很大 [11])，典型的实现可能看起来像这样：

$\mathbf{M}=\mathbf{0}$;
for $i=1$ **to** N **do**
  $\mathbf{I}_i^C=\mathbf{I}_i$;
end for
for $i=N$ **to** 1 **do**
  $\mathbf{f}_i^C(i)=\mathbf{I}_i^C\mathbf{h}_i$;
  for all $j\in\nu(i)$ **do**
    $\mathbf{M}_{ij}=\mathbf{h}_i^{\mathrm{T}}\mathbf{f}_j^C(i)$;
  end for
  if $\lambda(i)\neq 0$ **then**
    $\mathbf{I}_{\lambda(i)}^C=\mathbf{I}_{\lambda(i)}^C+{}^{\lambda(i)}\mathbf{X}_i^F\mathbf{I}_i^C\,{}^i\mathbf{X}_{\lambda(i)}^M$;
    for all $j\in\nu(i)$ **do**
      $\mathbf{f}_j^C(\lambda(i))={}^{\lambda(i)}\mathbf{X}_i^F\mathbf{f}_j^C(i)$;
    end for
  end if
end for

3.3 铰接体算法

ABA 的出发点是观察到刚体系统中物体的加速度始终是作用力的线性函数。如果刚体系统的运动因对系统中的任意一个物体施加外部试验力 $\mathbf{f}$ 而受到干扰，则该物体的加速度可表达为一个线性方程

$$\mathbf{a}=\mathbf{\Phi}\mathbf{f}+\mathbf{b},$$

其中，$\mathbf{a}$ 是物体的加速度，$\mathbf{b}$ 是如果 $\mathbf{f}$ 为零时的加速度，$\mathbf{\Phi}:\mathsf{F}^6\mapsto\mathsf{M}^6$ 是一个 6×6 矩阵。如果所选物体的运动是无约束的，则 $\mathbf{\Phi}$ 是非奇异的，整个方程可以求逆，读取为

$$\mathbf{f}=\mathbf{I}^A\mathbf{a}+\mathbf{p}^A, \tag{11}$$

其中 $\mathbf{I}^A=\mathbf{\Phi}^{-1}$ 并且 $\mathbf{p}^A=-\mathbf{I}^A\mathbf{b}$。

在铰接体术语中，整个系统被称为铰接体，承受试验力的一个物体被称为手柄 (handle)，$\mathbf{I}^A$ 是手柄的铰接体惯量 (ABI)，$\mathbf{p}^A$ 是相应的导向力 (bias force)，这是使手柄加速度为零所需的力。

本练习的要点是，方程 (11) 适用于任意系统中的任意物体，无论该系统中有多少其他物体，只要所选物体具有完全的运动自由度。这就是 ABA 的 $O(N)$ 复杂性的秘密。

ABA 中的主要步骤是依次计算每个连杆的 ABI 和导向力，选择连杆本身作为手柄，以该连杆上的子树作为铰接体。这可以通过使用以下递归公式，从终端连杆开始并朝着基座进行计算来完成 [11]：

$$\mathbf{I}_i^A=\mathbf{I}_i+\sum_{j\in\mu(i)}\left(\mathbf{I}_j^A-\mathbf{I}_j^A\mathbf{h}_j\left(\mathbf{h}_j^{\mathrm{T}}\mathbf{I}_j^A\mathbf{h}_j\right)^{-1}\mathbf{h}_j^{\mathrm{T}}\mathbf{I}_j^A\right), \tag{12}$$

$$\mathbf{p}_i^A=\mathbf{p}_i+\sum_{j\in\mu(i)}\left(\mathbf{p}_j^\alpha+\mathbf{I}_j^A\mathbf{h}_j\left(\mathbf{h}_j^{\mathrm{T}}\mathbf{I}_j^A\mathbf{h}_j\right)^{-1}\left(\boldsymbol{\tau}_i-\mathbf{h}_j^{\mathrm{T}}\mathbf{p}_j^\alpha\right)\right), \tag{13}$$

其中

$$\mathbf{p}_i = \mathbf{v}_i \times \mathbf{I}_i \mathbf{v}_i - \mathbf{f}_i^x,$$
$$\mathbf{p}_j^\alpha = \mathbf{p}_j^A + \mathbf{I}_j^A \dot{\mathbf{h}}_j \dot{\mathbf{q}}_j.$$

计算完这些量后，下一步是计算关节加速度。假设 $\mathbf{a}_{\lambda(i)}$ 已经被计算出来，我们可以在三个未知量 $\mathbf{a}_i$、$\mathbf{f}_i^J$ 和 $\ddot{\mathbf{q}}_i$ 中构建以下方程组：

$$\mathbf{a}_i = \mathbf{a}_{\lambda(i)} + \dot{\mathbf{h}}_i \dot{\mathbf{q}}_i + \mathbf{h}_i \ddot{\mathbf{q}}_i,$$
$$\mathbf{f}_i^J = \mathbf{I}_i^A \mathbf{a}_i + \mathbf{p}_i^A,$$
$$\boldsymbol{\tau}_i = \mathbf{h}_i^{\mathrm{T}} \mathbf{f}_i^J.$$

这些方程可以对 $\ddot{\mathbf{q}}_i$ 求解得出

$$\ddot{\mathbf{q}}_i = \left(\mathbf{h}_i^{\mathrm{T}} \mathbf{I}_i^A \mathbf{h}_i\right)^{-1} \left(\boldsymbol{\tau}_i - \mathbf{h}_i^{\mathrm{T}} \left(\mathbf{I}_i^A \mathbf{a}_{\lambda(i)} + \mathbf{p}_i^\alpha\right)\right), \tag{14}$$

这反过来又可以用来评估 $\mathbf{a}_i$，如此等等。

方程 (3)、(12)、(13)、(14) 和 (4) 共同描述了绝对坐标中的 ABA。该算法在连杆坐标中运行得更快。将 ABA 转换为连杆坐标的过程与 RNEA 的相同。

3.4 闭环系统

具有闭合运动回路的系统比树结构系统复杂得多，需要进行冗长的解释才能使其合理；因此，我们建议读者参考文献 [27,34] 中的冗长解释。

主要问题是联合变量不再是独立的，因为它们受到 (可能非常复杂的) 环路闭包约束。识别一组独立的位置变量通常是不切实际的，所以大多数算法都是用一个更大的、非独立的变量集合工作，并采取措施强制执行它们之间存在的约束。

处理运动环路最常用的方法是从连通图中提取生成树，计算生成树的运动方程，然后添加模拟运动环路效果的环路闭合力。生成树是一个子图，包含原始图中的所有节点和弧的子集，并且具有树连通性。相应的树结构机制包含所有实体，但仅包含原始关节中的一个子集，回路闭合力模拟省略关节的效果。

让我们将闭环系统的运动方程表示为：

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C} = \boldsymbol{\tau} + \boldsymbol{\tau}_c, \tag{15}$$

其中 $\mathbf{M}$ 和 $\mathbf{C}$ 是指生成树，并使用运动学树方法计算，$\ddot{\mathbf{q}}$ 和 $\boldsymbol{\tau}$ 是分别包含树关节加速度和动力变量的复合向量，并且 $\boldsymbol{\tau}_c$ 是树关节力向量，模拟缺少关节的影响。运动学环路对树的运动以及树关节变量的值施加以下约束：[2]

$$\boldsymbol{\phi}(\mathbf{q}) = \mathbf{0}. \tag{16}$$

方程 (15) 和 (16) 一起提供了动力学的完整描述，但形式不方便。通常的下一步是将方程 (16) 微分两次，以得到一个涉及加速度的方程：[3]

$$\ddot{\boldsymbol{\phi}} = \boldsymbol{\phi}' \ddot{\mathbf{q}} + \dot{\boldsymbol{\phi}}' \dot{\mathbf{q}} = \mathbf{0}, \tag{17}$$

[2]该方程假设了硬性的约束。对通用的完整性和非完整性约束的处理在文献 [27,34] 中描述。

[3]在实践中，必须包括约束稳定项，因为微分方程 $\ddot{\boldsymbol{\phi}} = \mathbf{0}$ 在数值上是不稳定的 [27,34]。

其中 $\phi' = \partial\phi/\partial\mathbf{q}$。通常，$\phi'$ 可以是秩亏的，并且秩可以作为 $\mathbf{q}$ 的函数变化；因此，让我们也定义方程 (17) 的满秩版本：

$$\mathbf{L}\ddot{\mathbf{q}} = \mathbf{c}, \tag{18}$$

其中 $\mathbf{L}$ 是一个 $n_c \times n$ 矩阵，由 ϕ' 的 n_c 个线性无关的行组成，$\mathbf{c}$ 包含 $-\dot{\phi}'\dot{\mathbf{q}}$ 的相应元素，并且 n_c 是 ϕ' 的秩。

这可以证明，环路闭合力总是可以用以下形式表达

$$\boldsymbol{\tau}_c = \mathbf{L}^{\mathrm{T}}\boldsymbol{\lambda} + \boldsymbol{\tau}_a, \tag{19}$$

其中，$\boldsymbol{\lambda}$ 是未知的约束力变量 (或拉格朗日乘子) 的 $n_c \times 1$ 向量，$\boldsymbol{\tau}_a$ 表示环路闭合关节中的任意主动力。如果所有这些关节均为被动关节，则 $\boldsymbol{\tau}_a = \mathbf{0}$。

现在我们可以结合方程 (15)、(18) 和 (19) 以产生

$$\begin{bmatrix} \mathbf{M} & \mathbf{L}^{\mathrm{T}} \\ \mathbf{L} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ -\boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\tau} - \mathbf{C} + \boldsymbol{\tau}_a \\ \mathbf{c} \end{bmatrix}. \tag{20}$$

该方程或类似的东西出现在大多数闭环动力学公式中的某个点上。

下一步是对 $\ddot{\mathbf{q}}$ 求解方程 (20) 或其等效项。三种主要方法是

1. 对 $\ddot{\mathbf{q}}$ 和 $\boldsymbol{\lambda}$ 直接求解方程 (20)，
2. 首先求解 $\boldsymbol{\lambda}$，然后使用结果求解 $\ddot{\mathbf{q}}$，或者
3. 对 $\ddot{\mathbf{q}}$ 求解方程 (18)(或方程 (17))，将结果代入方程 (15)，消除未知的约束力，并求解剩余的未知量。

方法 1 最简单，但通常效率最低。由于系统矩阵的大小为 $(n+n_c)\times(n+n_c)$，因此该方法为 $O\left((n+n_c)^3\right)$。

如果 $n \gg n_c$，方法 2 特别有用，并且它提供了在生成树上使用 $O(n)$ 算法的机会 [5]。根据方程 (20)，

$$\mathbf{L}\mathbf{M}^{-1}\mathbf{L}^{\mathrm{T}}\boldsymbol{\lambda} = \mathbf{c} - \mathbf{L}\mathbf{M}^{-1}\left(\boldsymbol{\tau} - \mathbf{C} + \boldsymbol{\tau}_a\right). \tag{21}$$

该方程可以通过 $O(n)$ 算法在 $O\left(nn_c^2\right)$ 操作中规划，并在 $O\left(n_c^3\right)$ 中求解。一旦 $\boldsymbol{\lambda}$ 已知，可以在 $O\left(nn_c\right)$ 运算中计算 $\boldsymbol{\tau}_c$，并通过 $O(n)$ 算法求解方程 (15)；所以总的复杂度是 $O\left(nn_c^2 + n_c^3\right)$。

如果 $n - n_c$ 很小和/或 n_c 必须在运行时确定，方法 3 就很有用。一个特殊版本的高斯消除法 (或类似程序)，配备有数值秩检验，旨在解决欠定系统，直接应用于方程 (17)，以获得

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

其中 $\mathbf{y}$ 是一个 $n - n_c$ 未知的向量 (通常是 $\ddot{\mathbf{q}}$ 元素的线性无关子集)，$\ddot{\mathbf{q}}_0$ 是方程 (17) 的任意特定解，$\mathbf{K}$ 是 $n \times (n - n_c)$ 矩阵，其属性为 $\mathbf{LK} = \mathbf{0}$。对于 $\ddot{\mathbf{q}}$ 将该表达式代入方程 (15)，并用 $\mathbf{K}^{\mathrm{T}}$ 对两侧进行预乘，以消除 $\boldsymbol{\tau}_c$ 的 $\mathbf{L}^{\mathrm{T}}\boldsymbol{\lambda}$ 分量，产生

$$\mathbf{K}^{\mathrm{T}}\mathbf{M}\mathbf{K}\mathbf{y} = \mathbf{K}^{\mathrm{T}}\left(\boldsymbol{\tau} - \mathbf{C} + \boldsymbol{\tau}_a - \mathbf{M}\ddot{\mathbf{q}}_0\right). \tag{22}$$

该方法也有立方复杂度，但如果 $n - n_c$ 较小，它可能是最有效的。据报道，它也比方法 1 更稳定 [9]。

3.5 全局分析技术

有可能将 N 个独立刚体系统的运动方程以下列形式表达出来

$$\mathbf{f} = \mathbf{Ia} + \mathbf{v} \times \mathbf{Iv}, \tag{23}$$

其中

$$\mathbf{f} = \left[\mathbf{f}_1^{\mathrm{T}}, \ldots, \mathbf{f}_N^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathrm{F}^{6N},$$
$$\mathbf{a} = \left[\mathbf{a}_1^{\mathrm{T}}, \ldots, \mathbf{a}_N^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathrm{M}^{6N},$$
$$\mathbf{I} = \mathrm{diag}\left(\mathbf{I}_1, \ldots, \mathbf{I}_N\right) : \mathrm{M}^{6N} \mapsto \mathrm{F}^{6N},$$

并如此之类。这样的复合向量和矩阵是 Rodriguez 等人 [28,29,17] 所开发的空间算子代数的起点。通过引入两个向量空间 M^{6N} 和 F^{6N}，我们给出了它们的空间代数处理，这两个向量空间分别是 N 组 M^6 和 F^6 的笛卡尔乘积。

假设上述系统受到运动学约束，将 (瞬时) 速度限制在一个 n 维子空间 $S \subset \mathsf{M}^{6N}$ 中。因此，加速度受制于

$$\mathbf{a} - \mathbf{a}_0 \in S \subset \mathsf{M}^{6N},$$

其中 $\mathbf{a}_0$ 是 (已知) 速度乘积项的向量。如果 $\mathbf{S}$ 是一个 $6N \times n$ 矩阵，其性质为 $\mathrm{Range}\,(\mathbf{S}) = S$，则

$$\mathbf{a} = \mathbf{a}_0 + \mathbf{S}\boldsymbol{\alpha}, \tag{24}$$

其中 $\boldsymbol{\alpha}$ 是未知量的向量。

这些运动学约束将对系统施加一定的力。让我们将总作用力定义为 $\mathbf{f} = \mathbf{f}_a + \mathbf{f}_c$，其中 $\mathbf{f}_a$ 是已知主动力的向量，$\mathbf{f}_c$ 是未知约束力的向量。因此，受约束系统的运动方程为

$$\mathbf{f}_a + \mathbf{f}_c = \mathbf{Ia} + \mathbf{v} \times \mathbf{Iv}. \tag{25}$$

根据虚功原理，$\mathbf{S}^{\mathrm{T}}\mathbf{f}_c = \mathbf{0}$；因此，如果我们用方程 (24) 代入 $\mathbf{a}$，并将结果预乘 $\mathbf{S}^{\mathrm{T}}$，则我们得到

$$\mathbf{S}^{\mathrm{T}}\mathbf{IS}\boldsymbol{\alpha} = \mathbf{S}^{\mathrm{T}}\left(\mathbf{f}_a - \mathbf{Ia}_0 - \mathbf{v} \times \mathbf{Iv}\right). \tag{26}$$

该方程的结构与方程 (22) 相同，因为它们都使用相同的方法应用约束，并且仅在其开始的系统中有所不同：方程 (22) 从运动学树开始，方程 (26) 从无约束刚体系统开始。方程 (26) 中的矩阵比方程 (22) 中的矩阵大，但它们具有稀疏结构，可用于描述低阶算法。

如果我们还引入一个子空间 $T \subset \mathsf{F}^{6N}$ 表示由运动约束产生的所有可能的约束力的空间，以及一个 $6N \times (6N - n)$ 矩阵 $\mathbf{T}$ 以满足 $\mathrm{Range}\,(\mathbf{T}) = T$，则

$$\mathbf{f}_c = \mathbf{T}\boldsymbol{\beta}, \quad \mathbf{T}^{\mathrm{T}}\mathbf{S} = \mathbf{0}, \quad \mathbf{T}^{\mathrm{T}}\mathbf{a} = \mathbf{T}^{\mathrm{T}}\mathbf{a}_0.$$

用方程 (25) 结合这些方程产生

$$\begin{bmatrix} \mathbf{I} & \mathbf{T} \\ \mathbf{T}^{\mathrm{T}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ -\boldsymbol{\beta} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_a - \mathbf{v} \times \mathbf{Iv} \\ \mathbf{T}^{\mathrm{T}}\mathbf{a}_0 \end{bmatrix}, \tag{27}$$

这显然与方程 (20) 类似；并对 $\boldsymbol{\beta}$ 求解得出

$$\mathbf{T}^{\mathrm{T}}\mathbf{I}^{-1}\mathbf{T}\boldsymbol{\beta} = \mathbf{T}^{\mathrm{T}}\mathbf{a}_0 - \mathbf{T}^{\mathrm{T}}\mathbf{I}^{-1}\left(\mathbf{f}_a - \mathbf{v} \times \mathbf{Iv}\right), \tag{28}$$

这与方程 (21) 类似。

假设方程 (24) 中包含的约束是由一组关节造成的，这些关节将物体连接在一起形成一个运动学树，如第 3.1 节所述。在这种情况下，方程 (24) 是单个关节约束的全局表达式

$$\mathbf{a}_i - \mathbf{a}_{\lambda(i)} = \mathbf{h}_i \ddot{\mathbf{q}}_i + \dot{\mathbf{h}}_i \dot{\mathbf{q}}_i, \quad i = 1 \ldots N.$$

该方程可用全局形式表达为

$$\mathbf{P}\mathbf{a} = \mathbf{H}\ddot{\mathbf{q}} + \dot{\mathbf{H}}\dot{\mathbf{q}}, \tag{29}$$

其中 $\mathbf{H} = \mathrm{diag}\,(\mathbf{h}_i)$，并且 $\mathbf{P}$ 是关联矩阵，[4]定义如下：

$$\mathbf{P}_{ij} = \begin{cases} \mathbf{1}_{6\times 6} & : j = i \\ -\mathbf{1}_{6\times 6} & : j = \lambda\,(i) \\ \mathbf{0}_{6\times 6} & : \text{otherwise.} \end{cases}$$

该矩阵是稀疏的、下三角的且 (平凡的) 可逆的。其逆矩阵在文献 [29] 中对应于矩阵 $\boldsymbol{\phi}^*$ 。

将方程 (29) 放入与方程 (24) 相同的形式，我们得到 $\mathbf{S} = \mathbf{P}^{-1}\mathbf{H}$ 和 $\boldsymbol{\alpha} = \ddot{\mathbf{q}}$。将这些表达式代入方程 (26)，得到系统质量矩阵的以下表达式：

$$\mathbf{M} = \mathbf{H}^{\mathrm{T}}\mathbf{P}^{-\mathrm{T}}\mathbf{I}\mathbf{P}^{-1}\mathbf{H},$$

这是在文献 [29] 中描述的牛顿-欧拉分解。该方程的算子解释直接导出 RNEA，并通过算子求逆公式推导出 ABA。

通过利用其特殊的稀疏结构，可以直接从方程 (27) 中获得另一个用于运动学树的 $O(N)$ 正向动力学算法：它仅有 $O(N)$ 个非零元素，并且有一个完美的消除顺序 [5]。

对于无支链的特殊情况，方程 (28) 中的系数矩阵是块三角对角型的。这意味着它可以在具有 $O(N)$ 个处理器的并行计算机上在 $O(\log(N))$ 时间内求解；这是约束-力算法 [14,13] 的关键步骤。

4 结论

本文回顾了机器人动力学领域的一些主要贡献。主要算法和相关方程已使用通用的简洁的符号给出。计算成本的最小化是开发机器人机构控制和仿真新公式和算法的关键。机器人学家可以被认为是开发了计算效率最高的的低阶算法，用于逆向动力学、正向动力学和惯量矩阵。

由于篇幅所限，我们无法对机器人动力学中的其它一些主题进行广泛讨论。这些包括运动符号方程的自动生成和简化，并行计算机的算法，以及对特定类别系统的应用。对于一个特定的机器人构型，用符号形式表达的运动方程通常需要最少的计算来实现。Kane 方程 [19] 的使用在这方面可能是有价值的，尽管它们的使用也可能涉及到对基本方程的大量操作，以将其简化为最简单的形式。通过使用并行算法，研究人员已经能够在 $O(N)$ 个处理器上将 $O(N)$ 算法的计算量减少到 $O(\log N)$。机器人动力学在多足机器人、生物力学系统、航天器、柔性结构和水下机器人车辆上的应用，已经将许多基本算法扩展到特定领域的需求。希望随着更复杂的系统和应用的发展，机器人动力学的研究将有助于满足性能方面不断扩大的需求。

[4]标准关联矩阵的元素是标量 $+1$，-1 和 0，而不是 6×6 矩阵。参见文献 [27,34]。(其关联矩阵 S 对应于 $-\mathbf{P}^{\mathrm{T}}$。)

5 References

1. W. W. Armstrong, "Recursive Solution to the Equations of Motion of an n-Link Manipulator", Proc. of 5th World Congress on Theory of Machines and Mechanisms, pp. 1343-1346, 1979-July.

2. U. M. Ascher, D. K. Pai and B. P. Cloutier, "Forward Dynamics Elimination Methods and Formulation Stiffness in Robot Simulation", Int. J. Rob. Research, vol. 16, no. 6, pp. 749-758, 1997.

3. C. A. Balafoutis, R. V. Patel and P. Misra, "Efficient Modeling and Computation of Manipulator Dynamics Using Orthogonal Cartesian Tensors", IEEE Journal of Robotics and Automation, vol. 4, pp. 665-676, December 1988. View Article

4. C. A. Balafoutis and R. V. Patel, "Efficient Computation of Manipulator Inertia Matrices and the Direct Dynamics Problem", IEEE Transactions on Systems Man and Cybernetics, vol. 19, pp. 1313-1321, Sept/Oct 1989. View Article

5. D. Baraff, "Linear-Time Dynamics using Lagrange Multipliers", Proc. SIGGRAPH '96, pp. 137-146, 1996-August.

6. W. J. Book, "Recursive Lagrangian Dynamics of Flexible Manipulator Arms", Int. J. Robotics Research, vol. 3, no. 3, pp. 87-101, 1984.

7. M. Brady, J. M. Hollerbach, T. L. Johnson, T. Lozano-Perez and M. T. Mason, Robot Motion: Planning and Control, MA, Cambridge:The MIT Press, 1982.

8. H. Brandl, R. Johanni and M. Otter, "A Very Efficient Algorithm for the Simulation of Robots and Similar Multibody Systems Without Inversion of the Mass Matrix", Proc. of IFAC/IFIP/IMACS International Symposium on Theory of Robots, 1986-December.

9. R. E. Ellis and S. L. Ricker, "Two Numerical Issues in Simulating Constrained Robot Dynamics", IEEE Trans. on Systems Man and Cybernetics, vol. 24, no. 1, pp. 19-27, 1994. View Article

10. R. Featherstone, "The Calculation of Robot Dynamics using Articulated-Body Inertias", Int. J. Robotics Research, vol. 2, no. 1, pp. 13-30, 1983.

11. R. Featherstone, Robot Dynamics Algorithms, Boston/Dordrecht/Lancaster:Kluwer Academic Publishers, 1987.

12. R. Featherstone, "A Divide-and-Conquer Articulated-Body Algorithm for Parallel O(log(n)) Calculation of Rigid-Body Dynamics. Part 1: Basic Algorithm", Int. J. Robotics Research, vol. 18, no. 9, pp. 867-875, 1999.

13. R. Featherstone and A. Fijany, "A Technique for Analyzing Constrained Rigid-Body Systems and Its Application to the Constraint Force Algorithm", IEEE Trans. Robotics & Automation, vol. 15, no. 6, pp. 1140-1144, 1999. View Article

14. A. Fijany, I. Sharf and G. M. T. D'Eleuterio, "Parallel O(log N) Algorithms for Computation of Manipulator Forward Dynamics", IEEE Trans. Robotics & Automation, vol. 11, no. 3, pp. 389-400, June 1995. View Article

15. X. He and A. A. Goldenberg, "An Algorithm for Efficient Computation of Dynamics of Robotic Manipulators", Proc. of Fourth International Conference on Advanced Robotics, pp. 175-188, 1989-June.

16. J. M. Hollerbach, "A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity", IEEE Trans. on Systems Man and Cybernetics, vol. SMC-10, no. 11, pp. 730-736, 1980. View Article

17. A. Jain, "Unified Formulation of Dynamics for Serial Rigid Multibody Systems", Journal of Guidance Control and Dynamics, vol. 14, no. 3, pp. 531-542, 1991.

18. M. E. Kahn and B. Roth, "The Near Minimum-time Control of Open-loop Articulated Kinematic Chains", Journal of Dynamic Systems Measurement and Control, vol. 93, pp. 164-172, 1971.

19. T. R. Kane and D. A. Levinson, "The Use of Kane's Dynamical Equations in Robotics", Int. J. Robotics Research, vol. 2, no. 3, pp. 3-21, 1983.

20. O. Khatib, "A Unified Approach to Motion and Force Control of Robot Manipulators: The Operational Space Formulation", IEEE J. Robotics & Automation, vol. 3, no. 1, pp. 43-53, 1987. View Article

21. O. Khatib, "Inertial Properties in Robotic Manipulation: An Object-Level Framework", Int. J. Robotics Research, vol. 14, no. 1, pp. 19-36, 1995.

22. K. W. Lilly and D. E. Orin, "Alternate Formulations for the Manipulator Inertia Matrix", International Journal of Robotics Research, vol. 10, pp. 64-74, February 1991.

23. J. Y. S. Luh, M. W. Walker and R. P. C. Paul, "On-Line Computational Scheme for Mechanical Manipulators", Trans. ASME J. Dynamic Systems Measurement & Control, vol. 102, no. 2, pp. 69-76, 1980.

24. S. McMillan and D. E. Orin, "Efficient Computation of Articulated-Body Inertias Using Successive Axial Screws", IEEE Trans. on Robotics and Automation, vol. 11, pp. 606-611, 1995. View Article

25. S. McMillan and D. E. Orin, "Forward Dynamics of Multilegged Vehicles Using the Composite Rigid Body Method", Proc. of IEEE International Conference on Robotics and Automation, pp. 464-470, 1998-May. View Article

26. D. E. Orin, R. B. McGhee, M. Vukobratovic and G. Hartoch, "Kinematic and Kinetic Analysis of Open-chain Linkages Utilizing Newton-Euler Methods", Mathematical Biosciences, vol. 43, pp. 107-130, February 1979.

27. R. E. Roberson and R. Schwertassek, Dynamics of Multibody Systems, Berlin/Heidelberg/New York:Springer-Verlag, 1988.

28. G. Rodriguez, "Kalman Filtering Smoothing and Recursive Robot Arm Forward and Inverse Dynamics", IEEE Journal on Robotics and Automation, vol. RA-3, no. 6, pp. 624-639, 1987. View Article

29. G. Rodriguez, A. Jain and K. Kreutz-Delgado, "A Spatial Operator Algebra for Manipulator Modelling and Control", Int. J. Robotics Research, vol. 10, no. 4, pp. 371-381, 1991.

30. Y. Stepanenko and M. Vukobratovic, "Dynamics of Articulated Open-chain Active Mechanisms", Math. Biosciences, vol. 28, pp. 137-170, 1976.

31. J. J. Uicker, "Dynamic Force Analysis of Spatial Linkages", Transactions of the ASME Journal of Applied Mechanics, vol. 34, pp. 418-424, 1967.

32. A. F. Vereshchagin, "Computer Simulation of the Dynamics of Complicated Mechanisms of Robot Manipulators", Engineering Cybernetics, no. 6, pp. 65-70, 1974.

33. M. W. Walker and D. E. Orin, "Efficient Dynamic Computer Simulation of Robotic Mechanisms", Trans. ASME J. Dynamic Systems Measurement & Control, vol. 104, pp. 205-211, 1982.

34. J. Wittenburg, Dynamics of Systems of Rigid Bodies, Stuttgart:B. G. Teubner, 1977.