

机械手的显式递推牛顿-欧拉动力学 及其分散自适应控制的并行方案

刘美华 常文森 张良起

(自动控制系)

摘 要

文中推导了PUMA 560机械手的显式递推牛顿-欧拉动力学方程,提出了分散自适应控制的并行实现方案。算法复杂性分析显示了该实现方案的高效性。

关键词: 机器人, 分散控制, 自适应控制, 动力学, 并行处理, 人工智能

1 引 言

目前,工业机器人都采用独立关节PID控制方式。由于该方式没有考虑机器人的动力学效应,其控制性能受到很大程度上的限制。尤其是当机器人作高速运动时,其控制性能变得很差。因此,人们逐步认识到动力学补偿的重要性,提出了许多机器人动力学控制方案^[1~3]。然而,机器人动力学控制目前尚处于理论和仿真研究阶段而没能得到实际应用。其中主要有两个原因:其一是它要求过大的在线计算量,其二是它要求准确地已知机器人的动力学模型。

递推形式的牛顿-欧拉方程是现有机器人动力学建模中最有效的一种方法,其计算复杂性为 $O(N)$ 数量级,即与 N 成正比,这里 N 为机器人的自由度数。尽管如此,它仍然不能满足一般六自由度机器人实时控制的要求。克服这一困难有两个途径:其一是充分利用具体机器人的结构特点,其二是挖掘动力学计算的并行性。一般形式的动力学模型适用于任何机器人。很明显,这一通用性的获得是以增加模型的复杂性为代价的。实际上,工业机器人大都具有非常简单的结构形式。在针对具体机器人进行动力学控制器设计时有必要充分利用机器人的结构特点,这样可以大大简化其动力学模型^[4~7]。特别地,文献^[4]建立了PUMA560机械手的显式拉格朗日动力学模型,而文献^[5]则给出了一直接驱动机械手的递推牛顿-欧拉方程。本文在文献^{[4],[5]}的基础上,推导了PUMA560机械手的显式递推牛顿-欧拉动力学模型。通过计算复杂性比较,进一步显示了递推牛顿-欧拉动力学与拉格朗日动力学相比的简易性。

机器人控制的多CPU并行实现是近年来来的一个热门研究课题^[7~11]。封闭形式的拉格朗日动力学方程的各项相互独立,因而具有明显的并行性^[8],而递推形式的牛顿-欧

拉方程是一个递归问题,许多研究者运用求解递归问题的方法提出了递推牛顿-欧拉动力学的并行实现^{[9],[10]}。然而,这些并行实现的并行效率都比较低。文献^[11]详细地分析了递推牛顿-欧拉方程的并行性,提出了非常高效的并行计算算法。但由于它要求过多的处理单元和复杂的通信,其实现特别昂贵。本文在显式递推牛顿-欧拉方程的基础上,提出了动力学计算的六处理机并行实现方案。这种充分利用机器人的结构特点和并行处理的设计思想将导致一个非常高效而且便于实现的机器人动力学控制的并行计算算法。

在机器人的动力学控制中,另一个有待解决的问题是它要求准确地已知其动力学模型,因为动力学控制的实质在于对消非线性动力学作用,以实现闭环系统的完全解耦和全局线性化。由于许多系统参数(如作用负载、关节摩擦以及结构变形等)实际上是不确定的,因而也无法准确得到。这就使得非线性动力学不能得到完全对消而使系统性能变差,甚至导致不稳定。为此,近年来已提出许多机器人的自适应控制方案^{[12],[13]}。这些方案在不同程度上提高了系统对模型不确定性的适应能力,改善了其控制性能。但是,这些高级的控制方案都比较复杂,难以实时实现。为此,我们提出了自适应控制的分散设计^{[14],[15]}。这些分散自适应控制不但具有良好的控制性能,而且结构简单,便于实现。本文在提出机器人显式牛顿-欧拉动力学模型的基础上,进一步对文献^[14]的分散自适应控制提出了一种高效的多处理机并行实现方案。

2 显式递推牛顿-欧拉动力学

机器人的动力学建模主要有两种常用的方法,即拉格朗日方程和牛顿-欧拉方程。封闭形式的拉格朗日方程是机器人控制器设计和综合的有效工具,而递推形式的牛顿-欧拉方程则是实时控制、求解逆动力学问题的有效方法。递推牛顿-欧拉方程包含向前和向后两组递推运算:向前递推(即从基座到末端执行器)求解正运动学问题以及各杆受到的净力和净力矩,向后递推(即从末端执行器到基座)求解关节力矩。对于一台有 N 个自由度的旋转关节式机器人,其递推牛顿-欧拉方程的一般形式如下:

向前递推 ($i=1,2,\dots,N$):

$$\omega_i = A_{i-1}^i (\omega_{i-1} + z_0 \dot{q}_i) \quad (1)$$

$$\dot{\omega}_i = A_{i-1}^i [\dot{\omega}_{i-1} + z_0 \ddot{q}_i + \omega_{i-1} \times (z_0 \dot{q}_i)] \quad (2)$$

$$v_i = A_{i-1}^i v_{i-1} + \dot{\omega}_i \times p_i^* + \omega_i \times (\omega_i \times p_i^*) \quad (3)$$

$$\dot{v}_{ci} = v_i + \dot{\omega}_i \times r_i + \omega_i \times (\omega_i \times r_i) \quad (4)$$

$$F_i = m_i \dot{v}_{ci} \quad (5)$$

$$N_i = I_i \dot{\omega}_i + \omega_i \times (I_i \omega_i) \quad (6)$$

向后递推 ($i=N, N-1, \dots, 1$):

$$f_i = A_{i+1}^i f_{i+1} + F_i \quad (7)$$

$$n_i = A_{i+1}^i n_{i+1} + N_i + p_i^* \times f_i + r_i \times F_i \quad (8)$$

$$\tau_i = n_i^T (A_{i-1}^i z_0) \quad (9)$$

其中, $z_0 = [0 \ 0 \ 1]^T$ 为常向量; 边界条件 $\omega_0 = \dot{\omega}_0 = 0$, $v_0 = [0 \ 0 \ g]^T$, 这里 $g = 9.80621 \text{m/s}^2$, f_{N+1} 和 n_{N+1} 分别为负载作用在末端执行器上的力和力矩向量; τ_i 为关节 i 上的输入力矩; \dot{q}_i 和 \ddot{q}_i 分别为关节 i 的运动速度和加速度; 三维向量 ω_i 、 $\dot{\omega}_i$ 、

$\dot{\theta}_i, \dot{\theta}_{ci}, F_i, N_i, r_i, p_i^*, f_i$ 以及 n_i 均在与杆 i 相连的坐标系中表示, 其中 $\omega_i, \dot{\omega}_i, \dot{\theta}_i, \dot{\theta}_{ci}, F_i, N_i$ 以及 r_i 分别为杆 i 的角速度、角加速度、线加速度、质心加速度、受到的净力、净力矩、以及质心坐标; p_i^* 为由 $i-1$ 坐标系原点到 i 坐标系原点的向量; f_i 和 n_i 分别为杆 $i-1$ 作用在杆 i 上的力和力矩; A_{i-1}^i 为杆 i 坐标系对杆 $i-1$ 坐标系的 3×3 旋转变换矩阵; I_i 为杆 i 对其质心的 3×3 惯量矩阵, 且在杆 i 坐标系中表示。

为了提高系统的伺服频率以改善其控制性能, 在实际控制器设计时有必要针对具体机器人对一般形式的动力学方程进行简化^[4~7]。实际上, 现有机器人大都具有比较简单的结构形式。考察以上递推牛顿-欧拉方程便可发现, 对于一个给定的机器人(如 PUMA), A_{i-1}^i, p_i^*, r_i 以及 I_i ($i=1, 2, \dots, N$) 中含有许多“0”元素; 此外, 还有 $z_0 = [0 \ 0 \ 1]^T$ 以及初始条件 $\omega_0 = \dot{\omega}_0 = 0, \dot{\theta}_0 = [0 \ 0 \ g]^T$ 。由于“0”或“1”与某数相乘或“0”与某数相加并不需要任何运算, 因而对于给定的具体机器人, 可以将以上一般形式的递推牛顿-欧拉方程显在地写出。在这一过程中, “0”或“1”参与的运算被自动省去。由此产生的显式递推牛顿-欧拉方程在一般形式的基础上得到了大大简化。基于这一设计思想, 我们采用文献[4]中给出的结构参数, 推导了 PUMA560 机械手的显式递推牛顿-欧拉动力学方程(见附录)。为了表明该显式模型的简易性, 并对它与文献[4]提出的显式拉格朗日方程进行比较, 表 1 列出了 PUMA560 机械手的一般形式及显式拉格朗日和牛顿-欧拉动力学的计算量。可以看出, 由于充分地利用了 PUMA560 机械手的结构特点, 显式模型比一般模型简单得多。另外, 通过两种显式模型之间的比较, 可进一步得出一个结论, 即递推牛顿-欧拉方程比拉格朗日方程简单, 显式牛顿-欧拉方程的计算效率是显式拉格朗日方程的计算效率的 2.5 倍。因此, 显式牛顿-欧拉动力学是现有动力学计算中最高效的一种方法。

表 1 PUMA560 机械手的四种动力学模型的计算复杂性比较

模 型	乘法	加法
一般形式的拉格朗日模型 ^[16]	66271	51548
一般形式的递推牛顿-欧拉模型 ^[16]	852	738
显式拉格朗日模型 ^[4]	739	426
显式递推牛顿-欧拉模型(本文)	228	220

3 动力学计算的并行实现

尽管显式递推牛顿-欧拉方程仍是一个相关问题, 即递推中每一步的计算都依赖前几步的计算结果, 但是, 其递推求解中存在并行性。首先, 可以通过引入中间变量, 将原显式方程写成一种紧凑的向量递推形式, 利用向量的各分量之间的相互独立性, 引入并行处理; 其次, 可以挖掘向量递推在递推求解过程中本身的并行性。下面, 仍以 PUMA560 机械手为例, 说明提出的动力学计算及自适应控制的多处理机并行实现方案。

附录中给出了 PUMA560 机械手的紧凑向量递推形式的显式牛顿-欧拉动力学方程, 它是在原显式递推牛顿-欧拉方程的基础上经过引入适当的中间变量, 写成适合于三 CPU 并行计算的结果。对于显式方程中的每个向量, 引入三个 CPU 并行地计算其三个分量。在此并行实现方案下, PUMA560 机械手的显式递推牛顿-欧拉方程的并行处理时间为 $96t_* + 81t_+$ 。这里 t_* 和 t_+ 分别为乘法和加法的运算时间(对于带协处理器 80287

的 Intel80286 16 位处理器, $t_* = 11.8\mu\text{s}$, $t_+ = 8.75\mu\text{s}$; 对于带协处理器 MC68881 的 Motorola MC68020 32 位处理器, $t_* = 3.1\mu\text{s}$, $t_+ = 2.8\mu\text{s}$)。假定 $t_* = t_+ = 1$ (这是评价和分析并行算法时常用的假设条件), 那么采用三台处理器并行计算 PUMA560 机械手的显式递推牛顿-欧拉动力学方程的速度增长倍数

$$S_3 = 448 / (96 + 81) \approx 2.53$$

并行处理效率为

$$E_3 = 2.53 / 3 \approx 0.843$$

再利用向量递推本身在递推过程中的并行性, 可以进一步减小计算时间。对于 PUMA560 机械手, 表 2 列出了其串行向量递推形式的显式牛顿-欧拉动力学方程的内并行性。可见, 整个显式模型的计算可分为 15 级, 下级依赖上级的处理结果, 而同级内各计算可并行进行。

表 2 PUMA560 显式递推牛顿-欧拉方程计算的并行性

并行级	并行任务 (见附录)	最大运算量
1	1	$1t_* + 1t_+$
2	2, 3, 7	$2t_* + 1t_+$
3	4, 8, 9, 10, 13	$3t_* + 2t_+$
4	5, 11, 14, 15, 16, 19, 42, 46	$4t_* + 5t_+$
5	6, 12, 17, 20, 21, 22, 23, 38	$4t_* + 5t_+$
6	18, 24, 25, 26, 27, 33, 34, 41, 45	$3t_* + 2t_+$
7	28, 30, 37	$2t_* + 2t_+$
8	29	$1t_*$
9	31, 32	$1t_* + 2t_+$
10	35, 36	$2t_* + 2t_+$
11	39	$2t_* + 2t_+$
12	40, 43	$4t_* + 4t_+$
13	44, 47	$5t_* + 5t_+$
14	48	$5t_* + 5t_+$
15	49, 50	$3t_* + 2t_+$

如果对处理器的台数不加以限制, 那么, 可以将表 2 中的并行级别作为并行处理步骤。因而需要 9 组处理器, 其中每组包含三个 CPU, 并行计算递推向量的三个分量。这时, 逆动力学的并行处理时间为 $42t_* + 40t_+$ 。其速度增长倍数为

$$S_{27} = 448 / 82 \approx 5.46$$

而并行处理效率为

$$E_{27} = 5.46 / 27 \approx 0.202$$

可以看出, 该并行实现方案的并行处理效率很低。实际上, 从表 2 可以看出, 多数处理器在绝大多数处理时间内都处于空闲状态。

为了比较充分地利用处理器，作者兼顾计算时间和并行效率两项指标，提出了采用两组处理器，即 6 CPU 的并行处理方案。表 3 列出了采用两组处理器的并行计算步骤

表 3 并行计算程序表

执行步骤	任务分配 (参见附录)		计算时间	
	A 组处理器 (CPU ₁ ~CPU ₃)	B 组处理器 (CPU ₄ ~CPU ₆)	t_0	t_1
1	1	—	1	1
2	2	3	1	
3	4	7	2	1
4	8	9	1	
5	5	10	3	2
6	6	11	4	5
7	13	46	2	2
8	14	15	1	
9	12	16	2	4
10	19	45	2	1
11	20	41	1	
12	17	42	4	5
13	18	21	3	2
14	22	23	2	1
15	24	25	1	
16	33	37	1	
17	26	38	3	2
18	27	34	2	2
19	28	30	2	2
20	29	—	1	
21	31	32	1	2
22	35	36	2	2
23	39	40A	3	3
24	43	40	2	2
25	44	47	5	5
26	48A	48B, 48	3	3
27	49	50	3	2

总计算时间为 $58f_0 + 49f_1$.

及其计算时间。该并行实现方案的速度增长倍数

$$S_0 = 448 / (58 + 49) \approx 4.18$$

其并行处理效率为

$$E_0 = 4.18 / 6 \approx 0.696$$

应当指出，对于并行算法，采用乘法和加法的次数来估计算法的计算时间具有很大的近似。因为并行实现时，CPU 之间存在通讯和同步的问题。由于通讯和同步的时间因实现方式的不同而不同。本文没有考虑这部分时间。

4 分散自适应控制的并行实现

理想的机器人控制方案既要具有良好的控制性能,又要结构简单、易于实现。正如引言中指出的那样,分散自适应控制^{[14]、[15]}是一种可取的机器人控制方案。为了进一步减小该分散自适应控制方案的在线计算量,作者基于以上给出的显式递推牛顿-欧拉动力学计算的并行算法,提出了文献[14]的分散自适应控制的并行实现方案。

现在简略地回顾一下文献[14]的分散自适应控制。

该分散自适应控制器具有如下控制结构^[14]:

$$\tau(t) = \tau^d(t) + \delta\tau(t) \quad (10)$$

其中 $\tau^d(t)$ 为额定力矩,用来补偿机器人沿期望轨迹 $\{q_d(t), \dot{q}_d(t), \ddot{q}_d(t)\}$ 运动时各关节间的非线性耦合作用。它可以基于拉格朗日动力学方程或牛顿-欧拉动力学方程计算。采用上节中提出的显式牛顿-欧拉模型计算(将 q, \dot{q} 及 \ddot{q} 分别用 q_d, \dot{q}_d 及 \ddot{q}_d 取代,所求得的 $\tau(t)$ 便是 $\tau^d(t)$),以提高控制算法的运算速度。

由于机器人沿期望轨迹运动时各关节间的耦合作用得到了补偿,因此将机器人的运动方程沿期望轨迹线性化,得到机器人的离散化摄动状态模型。再采用分散设计技术,将机器人的每个关节作为一个子系统来考虑。于是,可以提出如下分散形式的摄动状态模型:

$$\begin{aligned} \begin{bmatrix} \delta q_i(k+1) \\ \delta \dot{q}_i(k+1) \end{bmatrix} &= A_{ii}(k) \begin{bmatrix} \delta q_i(k) \\ \delta \dot{q}_i(k) \end{bmatrix} + B_{ii}(k) \delta \tau_i(k) \\ &+ C_{ii}(k) \xi_i(k), \quad i=1, 2, \dots, N \end{aligned} \quad (11)$$

其中 $C_{ii}(k)\xi_i(k)$ 代表各子系统之间的耦合作用及外界干扰的影响。

基于分散模型(11),修正力矩 $\delta\tau_i(i=1, 2, \dots, N)$ 由极小化性能指标

$$\tau_i = E\{[\delta q_i(k+1)]^2 + p_i'[\delta \dot{q}_i(k+1)]^2 + \gamma_i'[\delta \tau_i(k)]^2\} \quad (12)$$

得到,以保证位移误差和速度误差同时达到极小。由此产生的 $\delta\tau_i(k)$ 可表达为

$$\delta\tau_i(k) = -[f_i\delta q_i(k) + g_i\delta \dot{q}_i(k)]/h_i \quad (13)$$

其中控制器参数 f_i, g_i 及 h_i 由递推最小二乘辨识算法基于如下估计模型提供:

$$\phi_i(k) = h_i\delta\tau_i(k-1) + f_i\delta q_i(k-1) + g_i\delta \dot{q}_i(k-1) + \varepsilon_i(k) \quad (14)$$

式中 $\phi_i(k)$ 为辅助输出,定义为

$$\phi_i(k) = \delta q_i(k) + p_i\delta \dot{q}_i(k) + \gamma_i\delta\tau_i(k-1) \quad (15)$$

整个控制算法总结如下:

- (1) 按显式递推牛顿-欧拉方程计算 $\tau^d(k)$;
- (2) 按分散自适应控制算法计算 $\delta\tau(k)$;
- (3) 计算 $\tau(k) = \tau^d(k) + \delta\tau(k)$, 并置 $k=k+1$ 返回(1)。

详细的控制算法及其仿真研究可参见文[14]。

很显然,该分散自适应控制算法的计算量主要由 $\tau^d(k)$ 和 $\delta\tau(k)$ 的计算所决定。由于这两部分的计算相互独立,它们可以分开并行地进行。又由于在分散自适应控制中,各关节子系统控制器是彼此独立的,因而可用多CPU实行并行处理。对于每一关节子系统设置一个CPU,用来完成该关节子系统控制器所要求的计算,包括递推参数估计

和分散自适应控制律 $\delta\tau_i(k)$ 的获取, $i=1,2,\dots,N$ 。于是, 每个 CPU 上的处理时间为 $59t_*+48t_+$ 。对于额定力矩 $\tau^d(k)$ 的计算, 作者仍用上节中给出的并行算法, 即用 6 个 CPU 按显式递推牛顿-欧拉方程并行计算 $\tau^d(k)$ 。对于 PUMA560, $\tau^d(k)$ 的并行计算时间为 $58t_*+49t_+$ 。因此, 对于 PUMA560 来说, 分散自适应控制器的并行实现共需 12 个 CPU, 且整个控制算法的并行处理时间取决于这 12 个 CPU 中的最大处理时间, 即 $59t_*+48t_+$ 。另外, $\tau^d(k)+\delta\tau(k)$ 需要一个并行加运算, 因而整个并行算法的计算时间为 $59t_*+49t_+$ 。表 4 列出了采用带协处理器 80287 的 Intel80286 和带协处理器 MC68881

表 4 分散自适应控制在 PUMA560 上的串行和并行实现时间

实现方式	各 CPU 上的最大处理时间	Intel	Motorola
		80286/80287	MC68020/MC68881
串行	$582t_*+508t_+$	11.3ms	3.2ms
并行	$59t_*+49t_+$	1.13ms	0.32ms

的 Motorola MC68020 的串行和并行处理时间。可以看出, 本文提出的充分利用机器人的结构特点和挖掘算法并行性的设计方法产生了一个高效的机器人控制的实现方案, 适用于高速机器人运动的快速伺服控制。

5 结 束 语

本文推导了 PUMA560 机械手的显式递推牛顿-欧拉动力学方程, 提出了动力学计算和分散自适应控制的并行实现方案。由于充分地利用了机器人的结构特点和向量递归的并行性, 其计算效率非常高。算法复杂性分析显示了并行实现的快速性。

参 考 文 献

- [1] Luh J.Y.S., etc. IEEE Trans., 1980, AC-25, (3): 468
- [2] Freund E., Int. J. Robotics Research, 1982; 1(1): 65
- [3] Tarn T.J., etc. Proc. 23rd IEEE CDC, 1984; 736
- [4] Armstrong B., etc., Proc. IEEE Int. Conf. Robotics & Automation, 1986; 510
- [5] Kanade T., etc. Proc, 23rd IEEE CDC, 1984; 1345
- [6] Horak D.T. Trans. ASME, J. DSC, 1984; 4: 350
- [7] Liu C.-H., etc. IEEE J. Robotics & Automation, RA-2, 1986; 2: 110
- [8] Khatib O., etc. Proc. IEEE Int. Conf. Robotics & Automation, 1986; 1381
- [9] Luh J.Y.S., etc. IEEE Trans., SMC-12, 1982; 2: 214
- [10] Lee C.S.G., etc. Proc. IEEE Int. Conf. Robotics & Automation, 1986; 851
- [11] Lathrop R.H., Int. J. Robotics Research, 1985; 4(2): 80
- [12] Hsia T.C. Proc. IEEE Int. Conf. Robotics & Automation, 1986; 183
- [13] 刘美华. 国防科技大学学报, 1987; (4): 18
- [14] Liu M.-H., etc., Proc. 26th IEEE CDC, 1987
- [15] 刘美华等. 华中工学院学报, 1987年, (4): 141
- [16] Hollerbach J.M.. IEEE Trans., SMC-10, 1980; 11: 730

附录 PUMA560机械手的显式递推牛顿-欧拉方程

对 PUMA 560 机械手设置 Denavit-Hartenberg 坐标系, 如图 1 所示。其坐标系参数和惯量参数分别在表 1 和表 2 中给出。

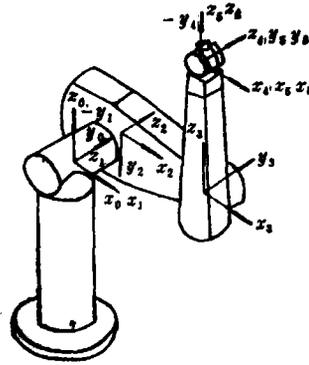


图 1 PUMA 560 机械手及其坐标系设置

表 1 坐标系参数

i	α_i	q_i	a_i	d_i
1	-90	q_1	0	0
2	0	q_2	0	d_2
3	90	q_3	a_3	d_3
4	-90	q_4	a_4	d_4
5	90	q_5	0	0
6	0	q_6	0	0

表 2 惯量参数(kg·m²)

i	$I_{i\ x x}$	$I_{i\ y y}$	$I_{i\ z z}$
1	—	1.5	—
2	1.3	0.524	5.249
3	0.086	0.086	0.0125
4	1.8×10^{-3}	0.2013	1.8×10^{-3}
5	0.3×10^{-3}	0.1794	0.3×10^{-3}
6	0.15×10^{-3}	0.15×10^{-3}	0.193

由表 1 得

$$A_1^0 = \begin{bmatrix} c_1 & 0 & -s_1 \\ s_1 & 0 & c_1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$A_2^1 = \begin{bmatrix} c_2 & -s_2 & 0 \\ s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$A_3^2 = \begin{bmatrix} c_3 & 0 & s_3 \\ s_3 & 0 & -c_3 \\ 0 & 1 & 0 \end{bmatrix}$$

$$A_4^3 = \begin{bmatrix} c_4 & 0 & -s_4 \\ s_4 & 0 & c_4 \\ 0 & -1 & 0 \end{bmatrix}$$

$$A_5^4 = \begin{bmatrix} c_5 & 0 & s_5 \\ s_5 & 0 & -c_5 \\ 0 & 1 & 0 \end{bmatrix} \quad A_6^5 = \begin{bmatrix} c_6 & -s_6 & 0 \\ s_6 & c_6 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

其中 $c_i = \cos(q_i)$, $s_i = \sin(q_i)$, $i = 1, 2, \dots, 6$ 。根据正交性, A_{i-1}^i 可表示如下

$$A_{i-1}^i = (A_i^{i-1})^T, \quad i = 1, 2, \dots, 6$$

由表 1 还可得到

$$p_1^* = [0 \ 0 \ 0]^T \quad p_2^* = [0 \ 0 \ p_{2z}^*]^T \quad p_3^* = [p_{3x}^* \ p_{3y}^* \ 0]^T$$

$$p_4^* = [p_{4x}^* \ p_{4y}^* \ 0]^T \quad p_5^* = p_6^* = [0 \ 0 \ 0]^T$$

其中 $p_{2z}^* = 0.2435$, $p_{3x}^* = 0.4318$, $p_{3y}^* = -0.0936$,

$$p_{4x}^* = -0.0203, \quad p_{4y}^* = -0.4331$$

另外, 由文献[4], 得到如下质心参数:

$$r_1 = [0 \ 0]^T \quad r_2 = [r_{2x} \ r_{2y} \ r_{2z}]^T \quad r_3 = [0 \ r_{3y} \ r_{3z}]^T$$

$$r_4 = [0 \ r_{4y} \ 0]^T \quad r_5 = [0 \ 0 \ 0]^T \quad r_6 = [0 \ 0 \ r_{6z}]^T$$

其中 $r_{2x} = 0.068$, $r_{2y} = 0.006$, $r_{2z} = -0.016$, $r_{3y} = 0.070$,

$$r_{3z} = 0.014, \quad r_{4y} = 0.019, \quad r_{6z} = 0.032$$

有了以上参数, 其显式递推牛顿-欧拉方程如下。

$$1: \quad \omega_2 = [-\dot{q}_1 s_2 \quad -\dot{q}_1 c_2 \quad \dot{q}_2]^T$$

$$2: \quad [\omega_{2xx} \ \omega_{2yy} \ \omega_{2xz}]^T = [\omega_{2x}^2 \ \omega_{2y}^2 \ \omega_{2z}^2]^T$$

$$3: \quad [\omega_{2xy} \ \omega_{2xz} \ \omega_{2yz}]^T = [\omega_{2x}\omega_{2y} \ \omega_{2x}\omega_{2z} \ \omega_{2y}\omega_{2z}]^T$$

$$4: \quad \dot{\omega}_2 = [-\ddot{q}_1 s_2 + \omega_{2yz} \quad -\ddot{q}_1 c_2 + \omega_{2xz} \quad \ddot{q}_2]^T$$

$$5: \quad \dot{v}_2 = [-g s_2 + p_{2z}^*(\dot{\omega}_{2y} - \omega_{2xy}) \quad -g c_2 + p_{2z}^*(\omega_{2xx} + \omega_{2yz}) \quad -p_{2z}^*(\omega_{2xx} + \omega_{2yy})]^T$$

$$6: \quad \dot{v}_{c2} = \begin{bmatrix} \dot{v}_{2x} - r_{2x}(\omega_{2yy} + \omega_{2zz}) + r_{2y}(\omega_{2xy} - \dot{\omega}_{2z}) + r_{2z}(\omega_{2xz} + \dot{\omega}_{2y}) \\ \dot{v}_{2y} + r_{2x}(\dot{\omega}_{2z} + \omega_{2xy}) - r_{2y}(\omega_{2xx} + \omega_{2zz}) + r_{2z}(\omega_{2yz} - \dot{\omega}_{2x}) \\ \dot{v}_{2z} + r_{2x}(\omega_{2xz} - \dot{\omega}_{2y}) + r_{2y}(\omega_{2yz} + \dot{\omega}_{2x}) - r_{2z}(\omega_{2xx} + \omega_{2yy}) \end{bmatrix}$$

$$7: \quad \omega_3 = [\omega_{2x} c_3 + \omega_{2y} s_3 \quad \omega_{2z} + \dot{q}_3 \quad \omega_{2x} s_3 - \omega_{2y} c_3]^T$$

$$8: \quad [\omega_{3xx} \ \omega_{3yy} \ \omega_{3xz}]^T = [\omega_{3x}^2 \ \omega_{3y}^2 \ \omega_{3z}^2]^T$$

$$9: \quad [\omega_{3xy} \ \omega_{3xz} \ \omega_{3yz}]^T = [\omega_{3x}\omega_{3y} \ \omega_{3x}\omega_{3z} \ \omega_{3y}\omega_{3z}]^T$$

$$10: \quad \dot{\omega}_3 = [\dot{\omega}_{2x} c_3 + \dot{\omega}_{2y} s_3 - \dot{q}_3 \omega_{3z} \quad \dot{\omega}_{2z} + \ddot{q}_3 \quad \dot{\omega}_{2x} s_3 - \dot{\omega}_{2y} c_3 + \dot{q}_3 \omega_{3x}]^T$$

$$11: \quad \dot{v}_3 = \begin{bmatrix} \dot{v}_{2x} c_3 + \dot{v}_{2y} s_3 - p_{3x}^*(\omega_{3xz} + \omega_{3yy}) + p_{3y}^*(\omega_{3xy} - \dot{\omega}_{3z}) \\ \dot{v}_{2z} + p_{3x}^*(\dot{\omega}_{3z} + \omega_{3xy}) - p_{3y}^*(\omega_{3xz} + \omega_{3xx}) \\ \dot{v}_{2x} s_3 - \dot{v}_{2y} c_3 + p_{3x}^*(\omega_{3xz} - \dot{\omega}_{3y}) + p_{3y}^*(\dot{\omega}_{3x} + \omega_{3yz}) \end{bmatrix}$$

$$12: \quad \dot{v}_{c3} = \begin{bmatrix} \dot{v}_{3x} + r_{3y}(\omega_{3xy} - \dot{\omega}_{3z}) + r_{3z}(\omega_{3xz} + \dot{\omega}_{3y}) \\ \dot{v}_{3y} - r_{3y}(\omega_{3xx} + \omega_{3zz}) + r_{3z}(\omega_{3yz} - \dot{\omega}_{3x}) \\ \dot{v}_{3z} + r_{3y}(\omega_{3yz} + \dot{\omega}_{3x}) - r_{3z}(\omega_{3xx} + \omega_{3yy}) \end{bmatrix}$$

$$13: \quad \omega_4 = [\omega_{3x} c_4 + \omega_{3y} s_4 \quad -(\omega_{3z} + \dot{q}_4) \quad \omega_{3y} c_4 - \omega_{3x} s_4]^T$$

$$14: \quad [\omega_{4xx} \ \omega_{4yy} \ \omega_{4xz}]^T = [\omega_{4x}^2 \ \omega_{4y}^2 \ \omega_{4z}^2]^T$$

$$15: \quad [\omega_{4xy} \ \omega_{4xz} \ \omega_{4yz}]^T = [\omega_{4x}\omega_{4y} \ \omega_{4x}\omega_{4z} \ \omega_{4y}\omega_{4z}]^T$$

$$16: \quad \dot{\omega}_4 = [\dot{\omega}_{3x} c_4 + \dot{\omega}_{3y} s_4 + \dot{q}_4 \omega_{4z} \quad -(\dot{\omega}_{3z} + \ddot{q}_4) \quad \dot{\omega}_{3y} c_4 - \dot{\omega}_{3x} s_4 - \dot{q}_4 \omega_{4x}]^T$$

$$17: \dot{r}_3 = \begin{bmatrix} \dot{r}_{3x}c_4 + \dot{r}_{3y}s_4 - p_{4x}^*(\omega_{4yy} + \omega_{4zz}) + p_{4y}^*(\omega_{4xy} - \dot{\omega}_{4z}) \\ - \dot{r}_{3z} + p_{4x}^*(\dot{\omega}_{4z} + \omega_{4xy}) - p_{4y}^*(\omega_{4xx} + \omega_{4zz}) \\ \dot{r}_{3y}c_4 - \dot{r}_{3x}s_4 + p_{4x}^*(\omega_{4xz} - \dot{\omega}_{4y}) + p_{4y}^*(\dot{\omega}_{4x} + \omega_{4yz}) \end{bmatrix}$$

$$18: \dot{r}_{c1} = [\dot{v}_{1x} + r_{1y}(\omega_{1xy} - \dot{\omega}_{4z}) \quad \dot{v}_{1y} - r_{1y}(\omega_{4xx} + \omega_{4zz}) \quad \dot{v}_{1z} + r_{4y}(\omega_{4yz} + \dot{\omega}_{4x})]^T$$

$$19: \omega_5 = [\omega_{1x}c_5 + \omega_{1y}s_5 \quad \omega_{4x} + \dot{q}_5 \quad \omega_{4x}s_5 - \omega_{4y}c_5]^T$$

$$20: [\omega_{5xy} \quad \omega_{5xz} \quad \omega_{5yz}]^T = [\omega_{5x}\omega_{5y} \quad \omega_{5x}\omega_{5z} \quad \omega_{5y}\omega_{5z}]^T$$

$$21: \dot{\omega}_5 = [\dot{\omega}_{1x}c_5 + \dot{\omega}_{4y}s_5 - \dot{q}_5\omega_{5z} \quad \dot{\omega}_{4z} + \dot{q}_5 \quad \dot{\omega}_{1x}s_5 - \dot{\omega}_{4y}c_5 + \dot{q}_5\omega_{5x}]^T$$

$$22: \dot{r}_{c5} = \dot{r}_5 = [\dot{v}_{4x}c_5 + \dot{v}_{1y}s_5 \quad \dot{v}_{4z} \quad \dot{v}_{1x}s_5 - \dot{v}_{1y}c_5]^T$$

$$23: \omega_6 = [\omega_{5x}c_6 + \omega_{5y}s_6 \quad \omega_{5y}c_6 - \omega_{5x}s_6 \quad \omega_{5z} + \dot{q}_6]^T$$

$$24: [\omega_{6xx} \quad \omega_{6yy} \quad -]^T = [\omega_{6x}^2 \quad \omega_{6y}^2 \quad -]^T$$

$$25: [\omega_{6xy} \quad \omega_{6xz} \quad \omega_{6yz}]^T = [\omega_{6x}\omega_{6y} \quad \omega_{6x}\omega_{6z} \quad \omega_{6y}\omega_{6z}]^T$$

$$26: \dot{\omega}_6 = [\dot{\omega}_{5x}c_6 + \dot{\omega}_{5y}s_6 + \omega_{6y}\dot{q}_6 \quad \dot{\omega}_{5y}c_6 - \dot{\omega}_{5x}s_6 - \omega_{6x}\dot{q}_6 \quad \dot{\omega}_{5z} + \dot{q}_6]^T$$

$$27: \dot{r}_6 = [\dot{v}_{5x}c_6 + \dot{v}_{5y}s_6 \quad \dot{v}_{5y}c_6 - \dot{v}_{5x}s_6 \quad \dot{v}_{5z}]^T$$

$$28: \dot{r}_{c6} = \begin{bmatrix} \dot{v}_{6x} + r_{6z}(\omega_{6xz} + \dot{\omega}_{6y}) \\ \dot{v}_{6y} + r_{6z}(\omega_{6yz} - \dot{\omega}_{6x}) \\ \dot{v}_{6z} - r_{6z}(\omega_{6xx} + \omega_{6yy}) \end{bmatrix}$$

$$29: F_6 = [m_6\dot{v}_{c6x} \quad m_6\dot{v}_{c6y} \quad m_6\dot{v}_{c6z}]^T$$

$$30: N_6 = [I_{6xx}\dot{\omega}_{6x} + (I_{6zz} - I_{6yy})\omega_{6yz} \quad I_{6yy}\dot{\omega}_{6y} + (I_{6xx} - I_{6zz})\omega_{6xz} \quad I_{6zz}\dot{\omega}_{6z}]^T$$

$$31: f_6 = [f_{7x} + F_{6x} \quad f_{7y} + F_{6y} \quad f_{7z} + F_{6z}]^T$$

$$32: n_6 = [n_{7x} + N_{6x} - r_{6z}F_{6y} \quad n_{7y} + N_{6y} + r_{6z}F_{6x} \quad n_{7z} + N_{6z}]^T$$

$$33: F_5 = [m_5\dot{v}_{c5x} \quad m_5\dot{v}_{c5y} \quad m_5\dot{v}_{c5z}]^T$$

$$34: N_5 = [I_{5xx}\dot{\omega}_{5x} + (I_{5zz} - I_{5yy})\omega_{5yz} \quad I_{5yy}\dot{\omega}_{5y} \quad I_{5zz}\dot{\omega}_{5z} + (I_{5yy} - I_{5xx})\omega_{5xz}]^T$$

$$35: f_5 = [f_{4x}c_6 - f_{6y}s_6 + F_{5x} \quad f_{6x}s_6 + f_{6y}c_6 + F_{5y} \quad f_{6z} + F_{5z}]^T$$

$$36: n_5 = [n_{4x}c_6 - n_{6y}s_6 + N_{5x} \quad n_{6x}s_6 + n_{6y}c_6 + N_{5y} \quad n_{6z} + N_{5z}]^T$$

$$37: F_4 = [m_4\dot{v}_{c4x} \quad m_4\dot{v}_{c4y} \quad m_4\dot{v}_{c4z}]^T$$

$$38: N_4 = [I_{4xx}\dot{\omega}_{4x} + (I_{4zz} - I_{4yy})\omega_{4yz} \quad I_{4yy}\dot{\omega}_{4y} \quad I_{4zz}\dot{\omega}_{4z} + (I_{4yy} - I_{4xx})\omega_{4xz}]^T$$

$$39: f_4 = [f_{5x}c_5 + f_{5z}s_5 + F_{4x} \quad f_{5x}s_5 - f_{5z}c_5 + F_{4y} \quad f_{5y} + F_{4z}]^T$$

$$40: n_4 = \begin{bmatrix} n_{5x}c_5 + n_{5z}s_5 + N_{4x} + r_{4y}F_{4z} \\ n_{5x}s_5 - n_{5z}c_5 + N_{4y} \\ n_{5y} + N_{4z} - r_{4y}F_{4x} \end{bmatrix} + \begin{bmatrix} p_{4y}^*f_{4z} \\ -p_{4x}^*f_{4z} \\ p_{4x}^*f_{4y} - p_{4y}^*f_{4x} \end{bmatrix}$$

(40 A) (40 B)

$$41: F_3 = [m_3\dot{v}_{c3x} \quad m_3\dot{v}_{c3y} \quad m_3\dot{v}_{c3z}]^T$$

$$42: N_3 = \begin{bmatrix} I_{3xx}\dot{\omega}_{3x} + (I_{3zz} - I_{3yy})\omega_{3yz} \\ I_{3yy}\dot{\omega}_{3y} + (I_{3xx} - I_{3zz})\omega_{3xz} \\ I_{3zz}\dot{\omega}_{3z} + (I_{3yy} - I_{3xx})\omega_{3xy} \end{bmatrix}$$

$$43: f_3 = [f_{4x}c_4 - f_{4z}s_4 + F_{3x} \quad f_{4x}s_4 + f_{4z}c_4 + F_{3y} \quad -f_{4y} + F_{3z}]^T$$

$$44: \quad n_3 = \begin{bmatrix} n_{4z}c_4 - n_{4x}s_4 + N_{3x} + p_{3y}^* f_{3z} + r_{3y} F_{3z} - r_{3z} F_{3y} \\ n_{4x}s_4 + n_{4z}c_4 + N_{3y} - p_{3x}^* f_{3z} + r_{3z} F_{3x} \\ -n_{4y} + N_{3z} + p_{3x}^* f_{3y} - p_{3y}^* f_{3x} - r_{3y} F_{3x} \end{bmatrix}$$

$$45: \quad F_2 = [m_2 \dot{v}_{c2x} \quad m_2 \dot{v}_{c2y} \quad m_2 \dot{v}_{c2z}]^T$$

$$46: \quad N_2 = \begin{bmatrix} I_{2xz} \omega_{2z} + (I_{2zz} - I_{2yy}) \omega_{2yz} \\ I_{2yy} \omega_{2y} + (I_{2xz} - I_{2zx}) \omega_{2xz} \\ I_{2zz} \omega_{2z} + (I_{2yy} - I_{2xx}) \omega_{2zy} \end{bmatrix}$$

$$47: \quad [f_{2x} \quad f_{2y} \quad -]^T = [f_{3x}c_3 + f_{3z}s_3 + F_{2x} \quad f_{3x}s_3 - f_{3z}c_3 + F_{2y} \quad -]^T$$

$$48: \quad n_2 = \begin{bmatrix} n_{3x}c_3 + n_{3z}s_3 + N_{2x} \\ n_{3x}s_3 - n_{3z}c_3 + N_{2y} \\ n_{3y} + N_{2z} \end{bmatrix} + \begin{bmatrix} r_{2y} F_{2x} - r_{2z} F_{2y} - p_{2z}^* f_{2y} \\ r_{2z} F_{2x} - r_{2x} F_{2z} + p_{2x}^* f_{2z} \\ r_{2x} F_{2y} - r_{2y} F_{2x} \end{bmatrix}$$

(48A)

(48B)

$$49: \quad [\tau_6 \quad \tau_5 \quad \tau_4]^T = [n_{6x} \quad n_{6y} \quad -n_{4y}]^T$$

$$50: \quad [\tau_3 \quad \tau_2 \quad \tau_1]^T = [n_{3y} \quad n_{2x} \quad I_{1yy} \dot{q}_1 - n_{2x}s_2 - n_{2y}c_2]^T$$

以上方程前面的数字表示任务编号，用来陈述并行算法，见第3节。

Feedback Linearization for Nonlinear Singularly Perturbed Systems

Jin Liang

Abstract

This paper deals with feedback linearization for a class of nonlinear singularly perturbed systems. First, the N -order approximate system about the singular perturbation parameter ε is obtained with the concept of integral manifold. Then, the feedback linearizable relationship between the reduced system and the N -order approximate system is presented, the results are shown to provide the calculating formulas of the linearization transformations, and an example is discussed.

Key words: Nonlinear systems, Singular perturbation, Integral manifold, Linearization

Explicit Recursive Newton-Euler Dynamics and Parallel Computation of Decentralized Adaptive Control for Robotic Manipulators

Liu Meihua Chang Wensen Zhang Liangqi

Abstract

In this paper, the explicit recursive Newton-Euler dynamic equations of PUMA 560 are derived and a parallel implementational scheme of decentralized adaptive control for robotic manipulators is presented. Analysis of algorithm computational complexity shows the high efficiency of the implementational scheme.

Key words: Robot, Decentralized control, Adaptive control, Dynamics, Parallel processing, Artificial intelligence