第 20 卷 第 6 期
1994 年 11 月

自 动 化 学 报
ACTA AUTOMATICA SINICA

Vol. 20, No. 6
Nov., 1994

机器人反向动力学方程的并行计算[1)]

刘鲁源　刘 畅　王欣东

(天津大学自动化系　天津　300072)

摘　　要

以 PUMA560 机器人的分解牛顿-欧拉反向动力学方程为模型，提出了方程分解的原则，由此得到 AOE (Activity On Edge) 有向图. 以此为基础，按照深度和时差的概念建立了 L-W 优先表，并导出了一种启发式的调度算法. 该算法在微处理机个数一定的情况下，可得到最小调度时间. 最后，以 Stanford 机器人的递推牛顿-欧拉反向动力学方程为例，说明了该算法的有效性.

关键词: 机器人动力学方程，并行计算，多微处理机.

1　引言

机器人控制用到的力矩计算技术及自适应扰动控制等均需要机器人关节力或关节力矩的计算. 由于机器人的物理参数和惯性矩阵可以辨识，各关节的位置和运动速度可以测量，而加速度可通过适当的公式计算出来，因此利用机器人反向动力学方程计算关节力或关节力矩是可行的. 在与计算关节力或力矩的相关控制方案中，计算精度和计算时间直接影响机器人运动的轨迹和速度. 当机器手沿着一定的轨迹运动时，必须实时地计算相应的关节力或力矩，因此减小反向动力学方程的计算时间是十分必要的.

许多学者从机器人反向动力学方程本身进行了减小计算时间的努力，并且取得了很大的进展. 但单一微处理机仍不能满足快速实时计算的要求. 于是，机器人反向动力学方程的并行计算及在多微处理机上进行的研究便成了一个新课题.

2　机器人反向动力学方程的分解

以具有六个转动关节的 PUMA560 机器人为样机，采用并行性较好的分解牛顿-欧拉反向动力学方程[1](相对于自身坐标系)作为研究的模型，将方程分解为若干子任务，使这些子任务之间具有最大的并行性和不相关性. 分解的原则是：尽可能以机器人反向动力学中的运动学或动力学参数的计算作为子任务，如果某参数的计算时间与其它参数的计算时间相差很大，则将它再分解成几个子任务，使各子任务计算时间的数量级基本一致；分解的各子任务数目在调度算法允许的范围内尽可能多，以满足各子任务之间最大的

本文于1992年 5 月 3 日收到.
1) 国家自然科学基金资助项目.

并行性和不相关性. 分解后的子任务方程是:

$1/i\ \omega_k^k = A_k^{k-1}Z_0$, $2/i\ V_k^{k*} = \omega_k^k \times P_k + \omega_k^k \times S_k$,

$3/i\ \omega_i^k = A_k^{k-1}\omega_i^{k-1}$, $4/i\ V_i^k = A_k^{k-1}V_i^{k-1} + \omega_i^k \times P_k$,

$5/i\ V_i^{k*} = V_i^k + \omega_i^k \times S_k$, $6/i\ \omega^k = \omega_i^k q_i' + \omega_k^k q_i' + \omega^k$,

$7/i\ VEC1 = \omega^{k-1} \times Z_0 q_k'$, $8/i\ \alpha^k = A_k^{k-1}(\alpha^{k-1} + Z_0 q_k'' + VEC1)$,

$9/i\ VEC2 = A_k^{k-1}a^{k-1}$, $10/i\ VEC3 = \omega^k \times P_k$,

$11/i\ VEC4 = \alpha^k \times P_k$,

$12/i\ a^k = \omega^k \times VEC3 + VEC2 + VEC4$,

$13/i\ VEC5' = \omega^k \times S_k$, $14/i\ VEC5 = \omega^k \times VEC5'$,

$15/i\ VEC6 = \alpha^k \times S_k$, $16/i\ F_k = m_k(a^k + VEC5 + VEC6)$,

$17/i\ VEC7 = J_k\alpha^k$, $18/i\ VEC8' = J_k\omega^k$,

$19/i\ VEC8 = \omega^k \times VEC8'$, $20/i\ N_k = VEC8 + VEC7$,

$21/i\ VEC9 = (N_k)^T \cdot \omega_i^k$, $22/i\ VEC10 = (F_k)^T \cdot V_i^{k*}$,

$23/i\ \tau_i = n_{n+1} \cdot \omega_i^{n+1} + f_{n+1}V_i^{n+1} = n_{n+1} \cdot A_{n+1}^n\omega_i^n + f_{n+1} \cdot A_{n+1}^n V_i^n$,

$24/i\ \tau_i = \tau_i + VEC9 + VEC10 + b_i q_i'$.

依据子任务方程可得到各子任务的优先关系和计算时间. 不失一般性, 这里假定装载数据时间为 10μs,一次加或减时间为 40μs,一次乘或除时间为 50μs, 空操作为 0.

3 多微处理机的调度算法

多微处理机的调度算法用以解决分解后子任务并行计算问题. 分解后的子任务相互关系可以用 AOE (Activity On Edge) 有向图形象地表示,因此, 对子任务的调度便转化为对 AOE 有向图的调度. 分解牛顿-欧拉 (N-E) 反向动力学方程, 分解后子任务的 AOE 图如图 1 所示. 图中结点表示子任务,结点左边的数字表示子任务的序号,结点上边的数字表示子任务的执行时间. 有向边的方向总是向下(在图中略去), 即表示上面的结点所代表的子任务优先于与它相关联的下面结点所代表的子任务.

图中每个结点均具有最早完成时间 TE(N_i)、最晚完成时间 TL(N_i) 和时差三个属性. 最早完成时间的定义为:

$$\mathrm{TE}(N_1) = 0,$$

$$\mathrm{TE}(N_i) = \max_k \sum_{i \in \varphi_k} t_i,\ (i \neq 1),$$

其中 N_1 为起点, φ_k 表示从 N_1 到 N_i 第 k 条路径的时间.

最晚完成时间的定义为

$$\mathrm{TL}(N_n) = \mathrm{TE}(N_n),$$

$$\mathrm{TL}(N_i) = \mathrm{TL}(N_n) - \max_k \sum_{i \in \varphi_k} t_i,$$

其中 N_n 为终点, φ_k 表示从 N_i 到 N_n 第 k 条路径的时间.

所谓时差是指结点的最晚完成时间与最早完成时间之差.

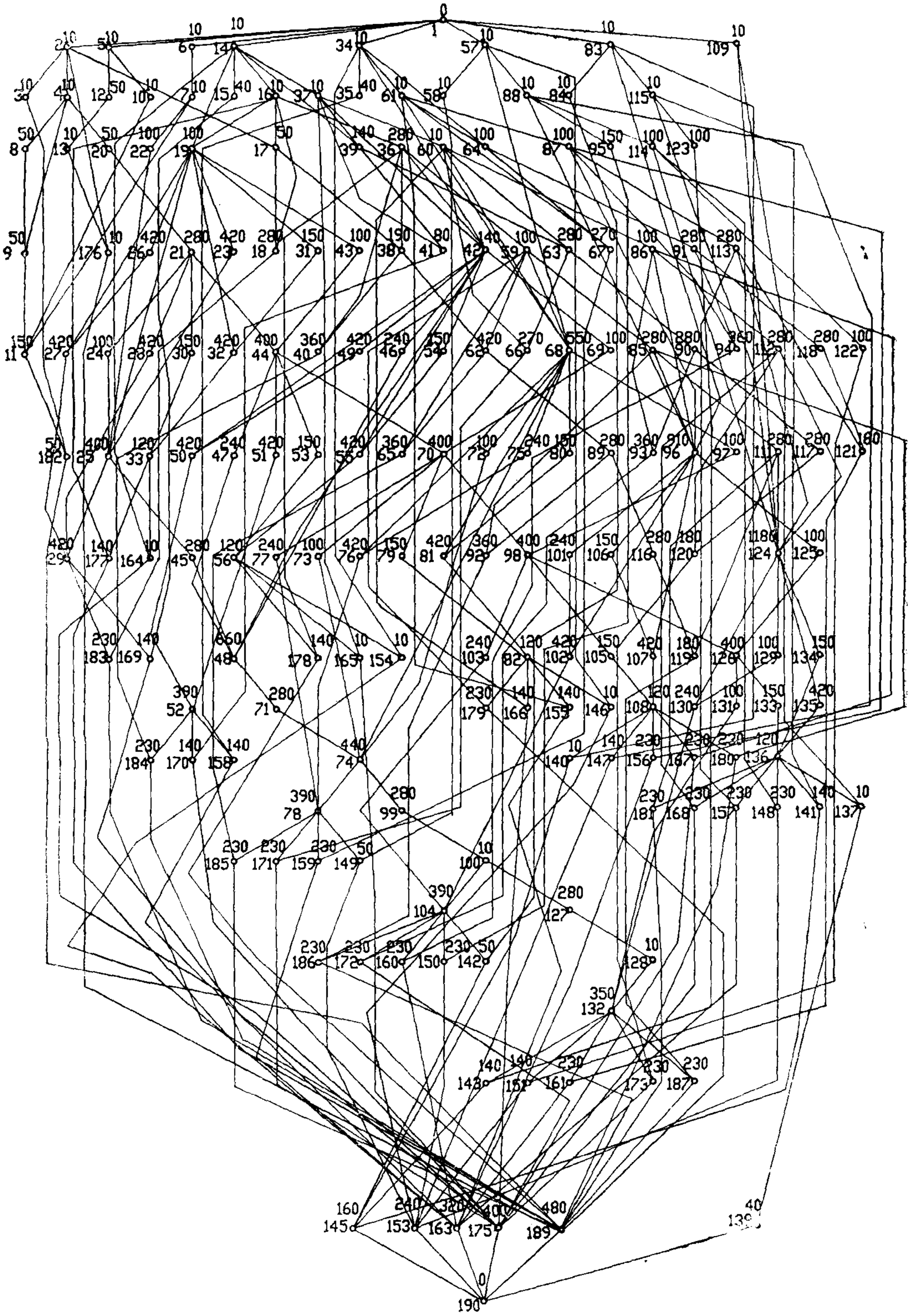


图 1　分解 N-E 反向动力学方程的 AOE 有向图

关键路径是从 AOE 有向图中起点到终点路径上结点时间总和最长的一条路径，这条路径的时间值为关键路径时间 t_{cr}，即

$$t_{cr} = \max_k \sum_{i \in \varphi_k} t_i,$$

其中 φ_k 代表从起点到终点的第 k 条路径时间．显然有

$$t_{cr} = \mathrm{TE}(N_n) = \mathrm{TL}(N_n).$$

多微处理机的调度问题属于强 NP 难解问题．在多微处理机个数任意及各子任务时间可以不相同的情况下，得到最优的调度结果是不可能的．但对于机器人反向动力学方程，寻求一种次优的调度算法是十分必要，也是可能的．据此提出相应的调度算法．具体步骤如下：

1）输入各结点的时间和微处理机的个数 P．

2）利用 AOE 有向图建立各结点的优先关系表和各结点处理时间表，以下分别简称为 OPEN 和 OPEN1 表．

3）确定每个结点的深度（Level）．第 i 个结点的“Level”用 l_i 表示，它是从终点 N_n 到第 i 个结点中最长路径上各结点时间之和．即

$$l_i = \max_k \sum_{j \in \pi_k} t_j,$$

其中 π_k 为从终点到第 i 个结点的第 k 条路径时间．

4）确定每个结点的时差 W_i，

$$W_i = \mathrm{TL}(N_i) - TE(N_i).$$

为了不使关键路径上的结点延迟，每个结点所允许延迟的最大值是它的时差．

5）利用 l_i 的值，由大到小对每个结点进行排序．当 l_i 相同时，利用时差 W_i 对其结点由小到大排序，形成 L-W 优先表．此表表示了对结点调度的先后顺序．

6）依据 OPEN 表，把 L-W 表中第 1 个结点分派给微处理机 P_1．

7）利用 OPEN1 表，确定该微处理机完成这个结点所需的时间 $\mathrm{TM}(P_1)$．

8）依据 OPEN 表，把 L-W 表中下一个结点分派给 $\mathrm{TM}(P_i)$ 最小的微处理机．但是，当有某一微处理机在已安排的结点之前有空闲时间，可以考虑把该结点插入在这个空闲时间中，其原则是：

（1）符合 OPEN 表；

（2）使可能推后结点的推后时间加上它原先可能延迟的时间小于等于它的时差．

9）利用 OPEN1 表，计算该结点分派给这个微处理机后的到达时间 $\mathrm{TM}(P_i)$．

10）如果 L-W 表中结点分派完，转 11）．否则转 8），继续调度．

11）输出分派给每个微处理机的结点执行顺序及 $\mathrm{TM}(P_i)$．

该算法是一种基于启发式的调度算法，其特点是：1）利用深度(Level)和时差等概念，对子任务进行了排序，排序后的 L-W 表是很有效的，并且显著地减少了该算法的时间复杂度和空间复杂度；2）提出了建立优先权表的观点；3）在进行深度搜索中，充分利用了空闲时间，使空闲时间尽可能地小，这样不必计算结点的启发函数值就可使调度时间达到极小值，从而进一步减少了算法的时间复杂度和空间复杂度；4）该算法可用以解决微处理机

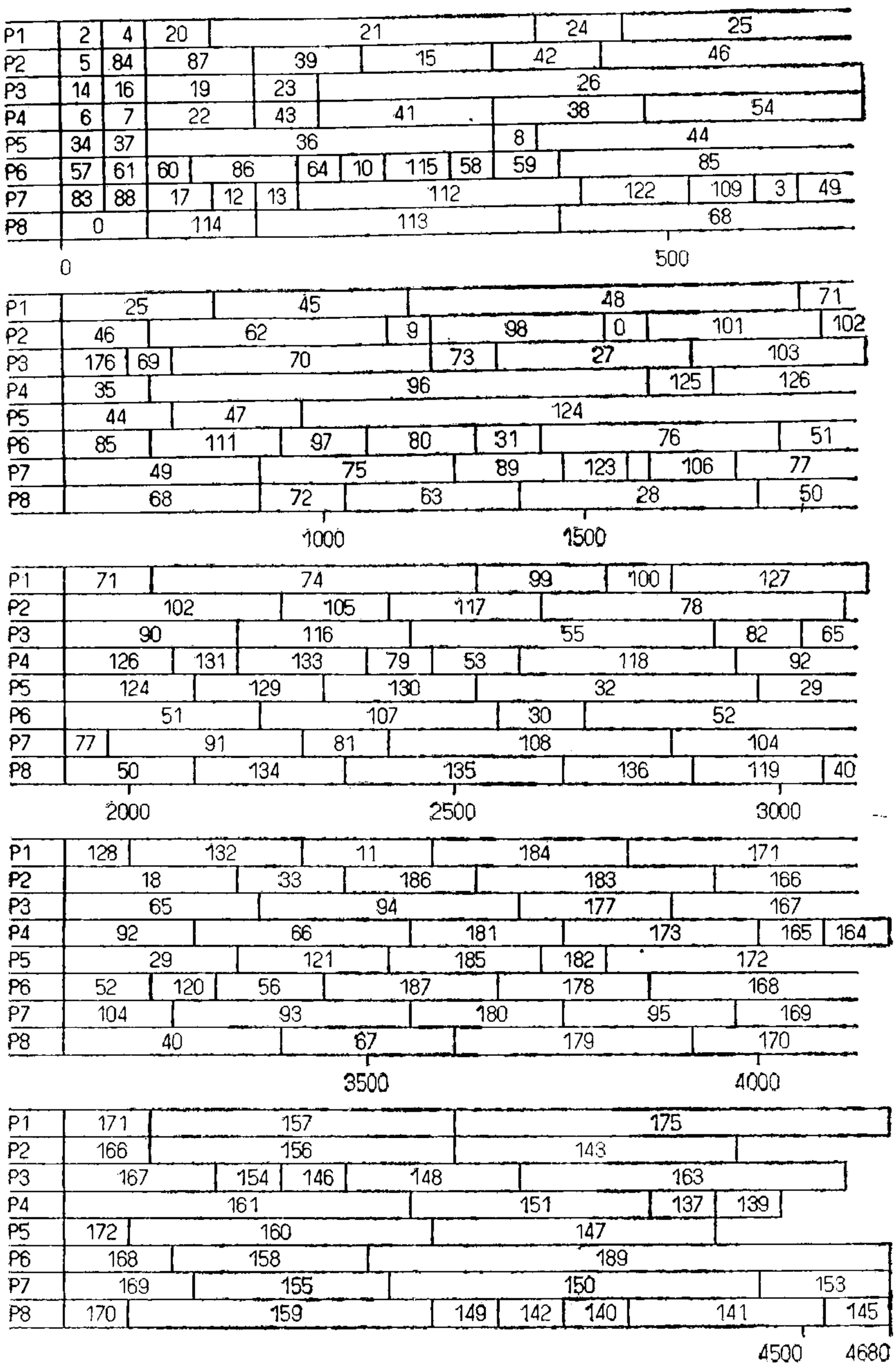


图 2　调度进程表

数任意、子任务时间不相同及子任务数目达到几百的任务.

对于 Stanford 机器人的递推牛顿-欧拉反向动力学方程形成的 AOE 有向图[3],(具有 88 个结点,132 条有向边,子任务时间在0—1110μs 内变化)将文献[2—4]的计算结果,与采用相同时间参数,用本文所提出的调度算法的计算结果进行比较,见表 1 所示. 由此可见,为了达到关键路径时间 t_{cr}, 文献[3,4]分别需 7 个, 6 个微处理机,文献[2]没有考虑,而本文仅需 5 个微处理机. 对 PUMA560 分解牛顿-欧拉方程的调度结果见表 2, 当

微处理机个数为 8 时，调度进程表见图 2 .

表 1 调度时间的比较 (ms)

微处理机个数	1	2	3	4	5	6	7
文献[2]	24.8	空	空	空	空	9.67	空
文献[3]	24.83	12.42	8.43	6.59	5.86	5.73	5.7
文献[4]	24.83	12.42	8.44	6.59	5.72	5.7	,空
本文	24.83	12.42	8.41	6.59	5.7	空	空

注: 文献[2]假定装载数据时间为 0,而文献[3,4]及本文中假定装载数据时间为 10μs.

表 2 本文提出的调度算法对图 1 的调度结果

微处理机个数	1	2	3	4	5	6	7	8	9	10
调度时间(μs)	36610	18310	12210	9160	7330	6120	5290	4680	4210	4150

4 结论

本文的调度结果未考虑子任务的通讯时间和竞争等待时间，但是对于实际的多机系统这是不容忽略的. 为此在计算子任务时间时，可直接把通讯时间考虑在子任务中，其调度算法不必进行任何修改，便可得到考虑通讯时间的调度进程表. 其次，看此进程表是否有竞争，对有竞争的子任务，依据硬件要求适当地延长一定时间，并把这个时间考虑在子任务中，重新利用调度算法得到调度进程表. 多次反复，最终得到的调度进程表应该是既考虑了通讯时间和竞争问题，又不应存在死锁现象.

随着超大规模集成电路的发展，高性能单片机的出现，为控制系统并行处理提供了良好的物质基础. 对上述调度结果，可通过采用专用 VLSI 或链路多、通讯速度高的单片机来实现.

参 考 文 献

[1] Hashimoto, K. Kimura. H, A new parallel algorithm for inverse dynamics. *Int. J. Robotics Research*, 1989, **8**(1): 63—76.

[2] Luh J Y S, Lin R C S. Scheduling of parallel computer for a computer-controlled mechamical manipulator. *IEEE Trans. Syst. Man. Cybem.*, 1982, **12**:214—234.

[3] Kasahara H, Narita S. Parallel processing of robot-arm control computation on a multiprocessor system, *IEEE J. Robotics Auto.*, 1985,**1** (2):104—113.

[4] Chen C L, Lee C S G. Efficient scheduling algorithms for robot inverse dynamics computation on a multiprocessor system. *IEEE Trans. Syst. Man. Cybern.*, 1988, **18**(5):729—743.

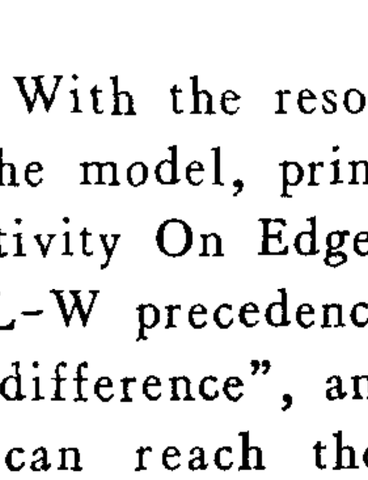
PARALLEL COMPUTATION OF ROBOT INVERSE DYNAMIC EQUATIONS

LIU LUYUAN LIU CHANG WANG XINGDONG
(*Dept. of Automation, Tianjin University, Tianjin* 300072 *P.R. China*)

ABSTRACT

With the resolved Newton-Euler inverse dynamic equations for the PUMA 560 as the model, principles for decomposing the equations are proposed and the AOE (Activity On Edge) directed graph is obtained. Based on the AOE directed graph, an L-W precedence list is constructed according to the concepts of "level" and "time difference", and a heuristic scheduling algorithm is also presented. This algorithm can reach the minimal scheduling time on a fixed number of processors. Compared with other ones the model of Stanford manipulator's recursive Newton-Eular inverse dynamic equations, the algorithm presented in this paper is effective.

Key words: Robot dynamic equations, parallel computation, multiprocessor systems.



刘鲁源　1941年生于山东．1964年毕业于山东大学电子系并留校工作．1975年至今在天津大学自动化系工作，现任智能控制研究室主任、副教授．主持或参加过多项微机控制项目，曾获机电部、国家教委、天津市科技进步二等和三等奖．近期研究方向是智能控制及并行处理．

刘　畅　1964年生于湖南．1992年在天津大学自动化系获硕士学位．现在北京有色冶金设计研究总院工作．近期研究方向是工业控制及计算机软件的应用与开发．

王欣东　1966 年生于北京，1991 年在天津大学系统工程研究所获硕士学位．现在天津大学自动化系智能控制研究室工作．研究方向是智能控制及并行处理．