弹道数据处理融合算法的并行计算*

童　丽　曾泳泓　朱炬波

(国防科技大学系统工程与数学系　长沙　410073)

摘　要　针对参考文献 [1] 中提出的融合多信源信息的融合算法，讨论了其中大计算量的测元遴选问题，并给出了它的并行算法。最后详细地分析了此并行算法的高效性和可扩展性，给出了加速比的仿真结果。

关键词　融合算法，多测元遴选，并行算法

分类号　TP301.6，V557

A Parallel Algorithm for the Fusion Method for Estimate of Trejectory

Tong Li　Zeng Yonghong　Zhu Jubo

(Department of Systems Engineering and Mathematics, NUDT, Changsha, 410073)

Abstract　The number of operations of the algorithm for measurement element selection is very large. The paper deals with this question. And its parallel algorithm is presented, with regard to the fusion method in reference [1]. The scalability for the algorithm is analyzed. The predicted speed-ups are given.

Key words　fusion algorithm, measurement element selection, parallel algorithm

航天技术的发展带来了超大量的数据，如何从这些数据中获取所需的信息已成为航天技术发展迫切需要解决的关键技术之一。多信源数据处理中的融合技术，通常用在信号检测，弹道跟踪等方面。就弹道跟踪而言，在一定的技术条件下，提高弹道跟踪数据的精度主要靠采用多设备联用，并使用先进的数据处理方法。文献 [1] 即研究了这个问题，即如何更好地融合多信源的有效信息，如何判断和消除观测误差，特别是系统误差，对有效信息的污染，以及如何考虑多信源的数据选择问题。

我们针对 [1，2] 所提出的通过各测量信道 (源) 的数据对轨道的影响，及考虑在各信源数据各种组合所估计弹道之间的差别，来识别较大系统误差源的方法，将提供一个有效的并行算法，从而使大计算量的异常测元的遴选和剔除工作得到较大程度改进。

1　融合算法原理

1.1　概述

设 $X(t) = (x(t), y(t), z(t), \dot{x}(t), \dot{y}(t), \dot{z}(t))$ 为一弹道，$s_k(t)$ 为从信源 k 得到的数据:

$$s_k(t) = f_k(X(t)) + e_k(t) + \epsilon_k(t), k = 1, 2, \cdots, K \qquad (1)$$

其中，$f_k(\quad)$ 是已知函数，$e_k(t)$ 为信源 k 系统误差，$\epsilon_k(t)$ 为零均值随机误差。

通常可以采用将 $X(t)$ 参数化的手段。如参考文献[2] 中采用样条函数表示 $X(t)$，即 $X(t) = B(t)\beta$，其中 $B(t)$ 由已知样条基构成，β 为待估参数。这样式 (1) 可表成

$$S_k = G_k(\beta) + e_k + \epsilon_k, k = 1, 2, \cdots, K$$

问题是如何使用 K 个信源的数据来估计参数 β。通常，工程分析方法虽能发现大部分的系统误差，往往有少数系统误差不能用完全直接的方法发现并消除，称它具有偶发性的系统误差，这种情况我们用下列式子描述:

* 国家自然科学基金和国家 863 项目资助
1999 年 3 月 19 日收稿
第一作者：童丽，女，1973 年生，讲师

$$S_k = G_k(\beta) + \delta_k e_k + \epsilon_k, k = 1, 2, \cdots, K$$

其中 δ_k 为一列独立随机变量, 且与 ϵ_k 亦独立。$P(\delta_k = 1) = p_k, P(\delta_k = 0) = 1 - p_k, p_k$ 通常是个较小的正数。下面, 对于模型 (3), 研究如何选取信源的数据, 以得到尽可能好的弹道估计。

1.2 线性模型

先假设 $G_k(\beta)$ 为线性函数, 即 $G_k(\beta) = G_k\beta$, 将多信源的数据联立起来, 记 $G = (G_1^T, \cdots, G_k^T)^T, S = (S_1^T, \cdots, S_k^T)^T, \Delta = (\delta_1 e_1^T, \cdots, \delta_k e_k^T)^T, \epsilon = (\epsilon_1^T, \cdots, \epsilon_k^T)^T$ 得

$$S = G\beta + \Delta + \epsilon \tag{4}$$

对于式 (4), 设 G 为列满秩, 易知 β 的最小二乘估计为

$$\hat{\beta} = (G^TG)^{-1}G^TS = \beta + (G^TG)^{-1}G^T(\Delta + \epsilon) \tag{5}$$

上式的最后一项是观测误差带来的估计误差。

1.3 单个信源对模型的影响分析

在估计弹道参数中, 增加了信源, 一般来说要增加新的有效信息, 但另一方面, 若该信源有较大的误差, 也会带来一份对信息的"污染", 在实际问题中需要衡量这两者的得失而决定取舍, 为此先分析一下单个信源数据对模型的影响。不妨考虑信源 K, 记 $G(K) = (G_1^T, \cdots, G_{k-1}^T)^T$, 即 $G = \begin{bmatrix} G(K) \\ G_k \end{bmatrix}$, 其余 $S(K), \Delta(K), \epsilon(K)$ 意义类似, 由式 (4) 可知:

$$S(K) = G(K)\beta + \Delta(K) + \epsilon(K) \tag{6}$$

式 (6) 即为不用信源 K 的数据时的模型, 记 $\hat{\beta}(K)$ 为 β 的最小二乘估计, 则有

定理1 模型 (6) 中, 参数 β 的最小二乘估计可表为

$$\hat{\beta}(K) = \hat{\beta} - AG_k^TS_k + AG_k^T(I - G_kAG_k^T)G_kAG^T(K)S(K) \tag{7}$$

其中 $A = (G^TG)^{-1}$, $\hat{\beta}$ 为模型 (4) 中参数 β 的最小二乘估计 (式 (5))。定理1说明, 若减少信源 K 的数据, 则得到的 β 的估计 $\hat{\beta}(K)$ 与 $\hat{\beta}$ 的差异可用式 (7) 的后一项表示。因此, 可以证明当 K 信源的系统误差 c_k 越大时, 使用该信源所造成的平均均方残差也越大 (详见 [1, 2])。

1.4 信源的选择准则

当我们不知道哪个信源具有系统误差时, 用模型 (4) (或 (6)) 来描述这些数据是恰当的, 现在我们要决定是否要用信源 K 的数据, 可以考虑下述检验问题:

H_0: K 信源的系统误差为 $c_k\delta_k$ (即仅以小概率 p_k 出现系统误差);

H_1: K 信源的系统误差为 c_k1 (即确定有系统误差 c_k)。

要检验上述问题, 可采用直观的方法比较 $\hat{\beta}$ 与 $\hat{\beta}(K)$。

定理2 在 H_0 成立时,

$$E(\hat{\beta} - \hat{\beta}(K)) = AG^T\begin{bmatrix} 0 \\ \\ e_kP_k \end{bmatrix} + AG_k^T(1 - G_kAG_k^T)G_kAG^T\begin{bmatrix} e_1P_1 \\ \\ e_{k-1}P_{k-1} \\ 0 \end{bmatrix} \tag{8}$$

由定理2可知, 当 H_0 成立, 式 (8) 接近于零, 即使用信源 K 与不使用信源 K 的结果相差不会太大, 而当 H_1 成立时, 平均均方残差将会明显增大, 因而可以断定信源 K 确有系统误差。据此可以考虑不采用信源 K 的数据。

1.5 多测元遴选方法

通过计算 $\hat{\beta} - \hat{\beta}(K)$ 或 $B(t)(\hat{\beta} - \hat{\beta}(K))$ 来判定信源 K 的数据是否可用, 有时也难免因信源 K 与某个信源 j 的具体数据的某种耦合而产生错误推断。我们有下面的检测方法。

定理3 在 H_0 成立时, 记 $Z(K) = \Sigma^{1/2}\Delta X(K) \sim N(0, I_{6N(K-1)})$, 其中 $N = [m/2]$, 从而

$$Z(K)^TZ(K) = \Delta X(K)^T\Sigma^{-1}\Delta X(K) \sim \chi^2(6N(K-1)) \tag{9}$$

当 $Z(K)^{\mathrm{T}}Z(K)$ 显著地大时，应认为信源 K 有常值系统误差。这里 $\Delta X(K)$ 的直观意义是，依次去掉一个信源 $i(i=1,2,\cdots,K-1)$ 得到的估计弹道，对不同的 i（取不同的时刻）分别与相应时刻去掉信源 K 的估计弹道作差。

1.6 非线性模型

把模型线性化即可转化成线性的模型。

2 测元遴选的并行算法

2.1 理论基础

由前面的分析可知，测元遴选的工作是按照多测元遴选准则，利用一定数量的测速元和测距元组合，进行测元选择和计算的过程。这个工作量是非常巨大的，因此我们考虑对这一部分采取并行计算。

当我们选定某一个测元组合后，即要在全模型所有的数据中去掉这个测元组合所对应的数据，对剩下的数据进行估计弹道与系统误差的计算。这样，每次进行的计算除了参加估计的数据不同以外，计算的步骤和过程是一样的，因此多测元遴选的算法在理论上具有很好的并行性质。

我们在并行算法中采取下面的方法:

(1) 首先在算法中对所有测元都参加，即取全部数据的情况进行估计弹道与系统误差的计算，得到一组参数估计值;

(2) 然后构造一个二维数组，依次放入每次所选择的不同测元组合对应的测元标号;

(3) 根据所选定的不同的测元组合，我们可以将选中的测元所对应的数据均赋值为 0，这样，所需要进行的计算过程和步骤和第 1 步中所有测元都参加估计的情况完全相同，因此可以采取第 1 步中的算法计算，得到另一组对应的参数估计值。

(4) 重复第 (2) 步、第 (3) 步，当取完所需要的测元组合后，对所有得到的各组参数估计值作弹道差别矩阵，由该差别矩阵找出异常测元，从而得到最优估计。

假设测元共有 m 个，一般情况下 m 为 19，并且需要参加计算的测元至少为6 个，那么测元遴选的工作实际上就转换成相同步骤的 $C_m^1+C_m^2+\cdots+C_m^{m-6}$ 次的估计弹道与系统误差的问题。因此测元遴选的并行算法实际上就抽象成为 $C_m^1+C_m^2+\cdots+C_m^{m-6}$ 个组合数如何平均分配到 $nproc$ 个处理机中的问题。

2.2 并行算法的描述

考虑有 m 个测元的情况，一般 m 不会太大。根据基数 m 不大和每次测元遴选的计算量都比较大这两方面的原因，我们设计如下的并行算法（k 表示 m 个测元中对应数据应取为 0 的测元个数，$k=1,2,\cdots,m-6$）:

(1) 假设在 $nproc$ 台并行机器中都存有所有测元对应的全部数据;

(2) 从 $k=1$ 开始，依次采用字典式排序的方法产生一组 $nproc$ 个组合数，用 $nproc$ 个数组来表示这 $nproc$ 个组合数，它们分别对应 $nproc$ 台处理机上对应数据应取为 0 的测元的标号;

(3) 将这 $nproc$ 个数组分别传入 $nproc$ 个处理机中。在各处理机中将接收到的数组所对应的测元对应的数据赋值 0，同时在 $nproc$ 台处理机中进行计算，并将所得到的参数的估计值传送回主机;

(4) 主机接收到 $nproc$ 个参数的估计值后，再继续第 (2) 步的工作，接着采用字典式排序的方法产生另一组 $nproc$ 个组合数，同样用 $nproc$ 个数组来表示这 $nproc$ 个组合数，然后回到第 (3) 步进行计算，并返回参数的估计值;

(5) 直到 $k=m-6$ 为止。主机根据所有的参数估计值作弹道差别矩阵，由该差别矩阵找出异常测元，得到最优估计。

算法结束。

2.3 并行算法的分析

此并行算法的串行计算量为产生字典式排序的组合数，以及根据所有的参数估计值作弹道差别矩阵。

算法的并行计算量大约为 $C_m^1 + C_m^2 + \cdots + C_m^{m-6}/nproc$ 次计算参数的估计值。由于每次计算参数的估计值的计算量都非常大，相对而言，每次并行启动的开销和传输启动的开销都可以忽略不计。因此，我们所采用的并行算法为了完全将 $C^m + C_m^2 + \cdots + C_m^{m-6}$ 次参数估计的计算完全平均地分到 $nproc$ 台机器中，而增加了传输的次数，这不仅不会影响算法的并行效率，相反会提高算法的并行效率。由前面的算法和并行模拟可以看出，并行计算量比串行计算量大得多得多，串行计算量在总计算量中的比重不超过百分之一。

算法只需要在第 (3) 步开始时将 $nproc$ 个组合数分别传入 $nproc$ 台并行处理机中，在第 (3) 步结束时从 $nproc$ 台并行处理机向主机传送 $nproc$ 个参数的估计值，传输量很小，且在每次的并行计算过程中不需要进行数据的传输。

由前面得分析，此并行算法的第 (3) 步实行了完全意义上的并行计算，各并行处理机的工作完全相同，非并行部分的计算量只占很小的一部分，且传输量很小，因此算法能充分发挥并行机器的操作特点，具有非常高的加速比。

此并行算法的并行原理是将 $C_m^1 + C_m^2 + \cdots + C_m^{m-6}$ 次参数估计的计算量完全平均地分到 $nproc$ 台机器中。这样的并行结构使得此算法具有非常良好的可扩展性。

2.4 并行算法的仿真结果

我们编制并行算法所用的平台是 PVM 并行编程环境，所用的编程语言是 FORTRAN 语言，使用的是国防科大六系国家重点实验室的微机网络机群系统。此系统拥有 8 台主频 166Hz，内存 32MB 的 586 微机。

仿真计算结果如下:

测元个数 m	对应数据取 0 的测元个数 k	机器台数	运行时间(s)	加速比
13	1	1	157.2	
13	1	4	61.1	2.57
13	1	8	29.3	5.4
13	2	1	965.2	
13	2	4	298.7	3.21
13	2	8	133.8	7.3

致谢 感谢国防科技大学系统工程与数学系数学技术实验室和国防科大“并行与分布式处理”国家重点实验室的大力支持。

参考文献

1 吴翊，朱炬波. 弹道数据处理的融合算法. 中国科学 (E 辑)，1998，(6)

2 王正明，朱炬波. 弹道跟踪数据的节省参数模型及应用. 中国科学 (E 辑) (已接收)

3 王正明，易东云. 测量数据建模与参数估计. 长沙：国防科技大学出版社，1996

4 孙家昶等. 网络编程计算与分布式编程环境. 北京：科学出版社，1996

5 陈希孺，王松桂. 近代回归分析. 合肥：安徽教育出版社，1987