

微机继电保护中的并行 自适应滤波算法

许昌继电器研究所 陆振华

摘要 本文首次在电力系统故障滤波中引入基于Transputer的并行计算技术及自适应滤波方法,并将两者结合起来提出了解决微机继电保护滤波问题的新方案。文中给出了该方案的技术经济论证,数学证明及实验结果,给出了该方案与卡尔曼滤波的比较结果,给出了并行自适应滤波的并行计算复杂度,速度倍数及应用结果分析,最后给出该方案中对处理机台数 P 的优化选择结果。

关键词 微机保护 自适应滤波 并行算法

一 问题的提出

微机继电保护的滤波算法中,速度和精度是两个最重要的指标,而它们又是相互矛盾的,为解决这一对矛盾,人们展开了种种不同的算法,从最早的pardar-70, Mann—Morrison导数算法到后来的全周及半周付氏算法以及数据拟合法,微分方程算法,卡尔曼滤波算法,人们提出的算法已达数百种。众多的算法中,卡尔曼滤波算法对速度与精度矛盾解决的较为成功,但卡尔曼滤波算法也有其存在的问题而使其未能迅速为人们所采用。首先,算法中所需要的噪声参数估计是很难确定的,它和不同的故障类型、故障点、系统接线等息息相关,只能通过统计试验进行估算,这样做是比较复杂和困难的,同时也会带来估计上的误差以致引起发散,另外,计算量过大也是其存在的问题。为此我们采用自适应滤波来减小误差提高速度,用并行计算技术来解决运算量过大的问题,并据此提出新的滤波方案,给出了相应的理论证明与实验结果。

二 自适应滤波算法及计算结果比较

自适应滤波的目的是在进行滤波的同时,利用观测数据带来的信息,不断地在线估计和校正模型参数或稳态滤波增益矩阵,以减小状态估值误差,提高滤波的精度,降低滤波的收敛时间。

在电力系统继电保护的模型中,我们假定忽略过渡过程中频率 ω 的偏移(可以证明,该偏移所引起的影响是相当小的),即假设系统模型和观测模型精确已知,噪声参数是未知的,亦即 Φ_k 和 H_k 为精确,均值和方差是未知的。系统可写为:

$$\text{状态方程: } \mathbf{x}_k = \Phi_k \mathbf{x}_{k-1} + \omega_{k-1}$$

$$\text{观测方程: } Z_k = H_k \mathbf{x}_k + V_k$$

其中, $\mathbf{x}_k \in \mathbb{R}^n$, $z_k \in \mathbb{R}^1$, Φ_k , H_k 为已知阵, ω_k 和 V_k 是独立的正态白噪声,且 $E[\omega_k] = \mathbf{q}$, $E[V_k] = r$, $\text{cov}[\omega_k, \omega_j] = Q_k \delta_{kj}$, $\text{cov}[v_k, v_j] = R_k \delta_{kj}$, 噪声均值 \mathbf{q} , r 和协方差 Q , R 是未知的。

自适应滤波问题是基于观测 $\{Z_k, Z_{k-1}, \dots, Z_1\}$ 求噪声统计估值和状态 \mathbf{x}_k 的滤波值。下面给出自适应滤波算法步骤。

算法1:

$$1^\circ \text{给出初值 } \mathbf{q}, r, Q, R, K_0 \text{ 令 } K = 1$$

$$2^\circ P_{k|Z_{k-1}} = \Phi_k P_{k-1|Z_{k-1}} \Phi_k^T + Q_{k-1}$$

$$3^\circ K_k = P_{k|Z_{k-1}} H_k^T (H_k P_{k|Z_{k-1}} H_k^T + R_{k-1})^{-1}$$

$$4^\circ P_{k|Z_k} = (I - K_k H_k) P_{k|Z_{k-1}}$$

$$5^\circ \hat{\mathbf{x}}_{k|Z_{k-1}} = \Phi_k \hat{\mathbf{x}}_{k-1|Z_{k-1}} + \mathbf{q}_{k-1}$$

$$6^\circ \varepsilon_k = Z_k - H_k \hat{\mathbf{x}}_{k|Z_{k-1}} - r_{k-1}$$

$$7^\circ \hat{\mathbf{x}}_{k|Z_k} = \hat{\mathbf{x}}_{k|Z_{k-1}} + K_k \varepsilon_k$$

$$8^\circ \text{判别 } \|\hat{\mathbf{x}}_{k|Z_k} - \hat{\mathbf{x}}_{k-1|Z_{k-1}}\| \leq \varepsilon \text{ 或 } K < K_0$$

两式中若有一式成立则结束,否则转9°

$$9^\circ \mathbf{q}_k = \frac{1}{K} [(K-1)\mathbf{q}_{k-1} + \hat{\mathbf{x}}_{k|Z_k} - \Phi_k \hat{\mathbf{x}}_{k-1|Z_{k-1}}]$$

$$10^\circ Q_k = \frac{1}{K} [(K-1)Q_{k-1} + K_k \varepsilon_k \varepsilon_k^T K_k^T + P_{k|Z_k} - \Phi_k P_{k-1|Z_{k-1}} \Phi_k^T]$$

$$11^\circ r_k = \frac{1}{K} [(K-1)r_{k-1} + Z_k - H_k \hat{\mathbf{x}}_{k|Z_{k-1}}]$$

$$12^\circ R_k = \frac{1}{K} [(K-1)R_{k-1} + \varepsilon_k \varepsilon_k^T - H_k P_{k|Z_{k-1}} H_k^T]$$

转2°

使用上述算法对电压模型和电流模型进行仿真计算并与卡尔曼滤波算法比较发现,对电压模型,该算法的滤噪性能比卡尔曼算法要更有效,对电流模型而言,该算法收敛和卡尔曼收敛速度相近。然而,可以进行一些小的改变将电流模式改造为电压模式。那就是预先施行一次减法滤波,而抑制大部分衰减直流分量,而对未能完全消去的部分作为噪音由电压模式滤除。我们可以算出未能消去的衰减直流分量造成的相对误差。

设衰减直流分量 $\mathbf{x}_k = A_0 e^{\mathbf{m} \Delta T Z^T}$, 其中 τ 为衰减常数,于是相对误差: $\frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{\mathbf{x}_k} = e^{\mathbf{m} \Delta T Z^T} - 1$, 对16点/周采样时 $\Delta T = 0.00125$ 秒,而 τ 的范围为0.02~0.04秒,可算

出此时相对误差在3.08%~6.06%，如此大的未消去的衰减直流分量加入其余谐波噪音是完全可以电压模式的自适应滤波消除的。因而，我们下面的讨论仅针对电压模型而言。

仿真计算表明自适应滤波和卡尔曼滤波相比有如下特点：

1. 滤波精度大为提高，并能避免发散现象。
2. 滤波能更快的收敛。
3. 滤波对初值极不敏感，初值的影响可以忽略。
4. 极强的抑制噪声的能力。
5. 计算量较大。

对二状态电压模型不同噪声时仿真计算结果比较见图1、图2所示：

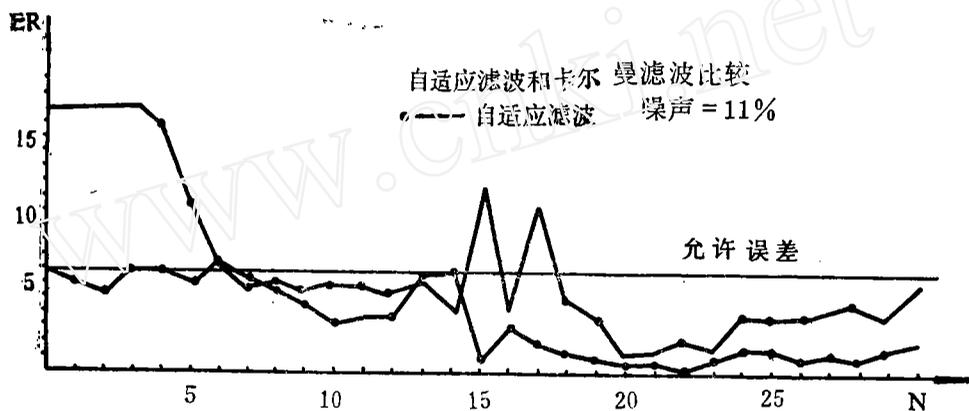


图 1

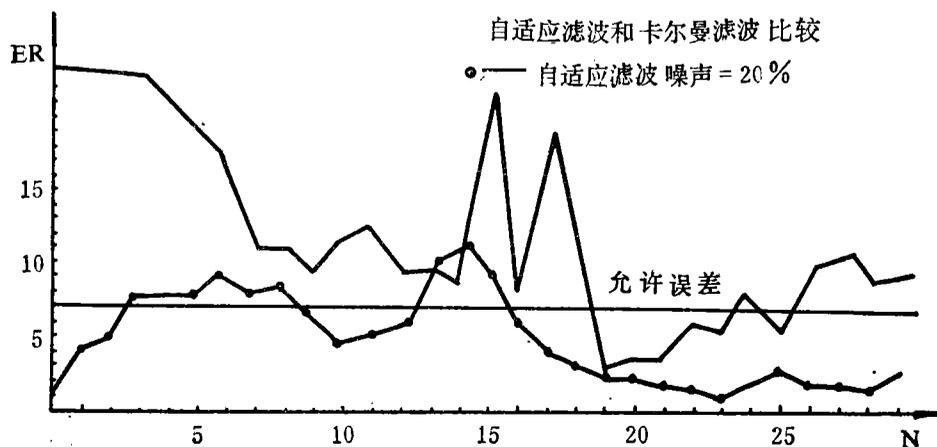


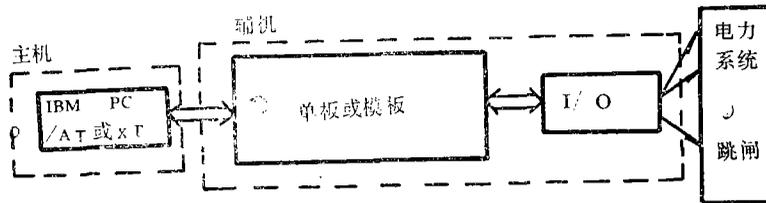
图 2

仿真计算结果表明，自适应滤波已解决了卡尔曼滤波所存在的几个问题，但与此同

时，计算量过大的问题却暴露的更为明显，为此，我们采用并行计算技术来解决。

三 并行处理方案

硬件框图见 3 所示：



注：单板指Transputer单板，模板指TRAMS模板

图 3

说明：

1. 硬件模型为由Transputer组成的分布式或多处理机系统，主机为 IBMPC / AT或XT，硬件框图如上。

2. 处理机的台数 P 限制在两个区间内研究，即 $0 < p \leq n$ 和 $n < p \leq n^2$ ，其中 n 是状态维数。作此假定主要是根据电力系统的实际状况及产品经济性来考虑。

3. 乘除运算和加减运算的时间在不同类的Transputer板是大不相同的，如T8类板产品比T4类产品快数倍，这在我们具体选型时加以考虑，这里仅假定乘除运算时间为加减运算时间的 L 倍。

4. 通讯和数据传输的时间暂时不计。

5. 每台处理机有任意多个存储单元。

技术经济可行性

英国INMOS公司于八十年代中期推出的以精简指令系统结构(RISC)而设计的计算机—Transputer即是其杰出代表。该机实质是置于一半导体芯片上的计算机，它的独特之处在于芯片内部兼有处理器，存储器及通信电路，既可在一般电路中使用，亦可以用来构成新型并行结构计算机系统，其功能强，处理速度快，价格便宜，并有编程功能，由于该公司开发出了并行语言ocean，并行C、并行Pascal，Ada等语言编程器，再加上功能强大的Helios(太阳神)分布式操作系统，使其更加具有活力。该系统可以用来进行科学计算、超级计算机、人工智能计算机仿真、数字信号处理、实时控制等方面，因而给我们进行实时故障滤波提供了强有力的支持。同时该公司为方便用户使用，提供多种模块式硬件产品TRAMS，该模板插入IBM PC/AT或XT，sun和VAX机上空闲插槽，即可进行并行处理。目前国内清华大学已将其国产化，生产出带有2—cpu和4—cpu的Transputer模板，价格比国外便宜许多，这为我们今后的使用提供了保证。

Transputer系列产品的重要特点之一是价格低廉，如上组成的硬件方案中滤波仅是在故障时才使用，可以做为集成保护的一个组成部分，因此将不会造成硬件的浪费。比

如用上述系统作为微机大机组保护的核心部件就有其它多cpu工作方式所不可比拟的优点。Transputer目前正在美国、日本、欧洲等迅速推广,采用者日多,因此其硬件价格还会急骤下降,因此,Transputer的使用将不会给继电保护产品造成大的经济负担而过多提高产品的价格。

四 并行自适应滤波算法的计算复杂度

理论上并行计算复杂度的研究就是寻求某一问题所需的最少并行步数及其所需的处理机台数的上界,但对我们的实际系统,我们对处理机台数已限定了区域,因此下面的讨论即针对 $0 < p \leq n$ 和 $n < p \leq n^2$ 之内而言,对该两个区间给出相应的矩阵相乘的内积算法。

内积算法 1 ($0 < p \leq n$)

令 $A = (a_{ij})^{m \times n}$, $B = (b_{ij})^{n \times l}$, 则 $C = (c_{ij})^{m \times l} = A \times B$ 的内积方法是:

$$c_{ij} = (a_{i1}, a_{i2}, \dots, a_{in}) (b_{1j}, b_{2j}, \dots, b_{nj})^T$$

$$i = 1, 2, \dots, m, j = 1, 2, \dots, l$$

对于Transputer组成的多处理机模型,每个 c_{ij} 用 p 台机计算,则共需乘法为 $m \times l \times \lceil \frac{n}{p} \rceil$, 加法为 $m \times l \times \lceil \frac{n}{p} \rceil \times \lceil \log p \rceil$

内积算法 2 ($n < p \leq n^2$)

其余条件同上,考虑每个 C_{ij} 用1台机计算需 n 次乘法, $n-1$ 次加法,则共需乘法为 $n \lceil \frac{ml}{p} \rceil$ 加法为 $(n-1) \lceil \frac{ml}{p} \rceil$ 。

为了后面定理使用,我们先给出几个引理,篇幅所限,其证明略去。

引理 1 两个 $n \times n$ 的矩阵相乘,采用内积算法 1,使用 p ($0 < p < n$) 台处理机并行计算,需乘法步为 $\widetilde{CT}(p) = n^2 \lceil \frac{n}{p} \rceil$, 加法步为 $\overline{CT}(p) = n^2 \lceil \frac{n}{p} \rceil \lceil \log p \rceil$ 。

引理 2 两个 $n \times n$ 的矩阵相加,用 p ($0 < p < n$) 台处理机并行计算,需乘法步为 $\widetilde{CT}(p) = 0$, 加法步为 $\overline{CT}(p) = \lceil \frac{n^2}{p} \rceil$ 。

引理 3 一个 $n \times n$ 的矩阵乘以 $n \times 1$ 的向量,用 p ($0 < p < n$) 台机并行计算,需乘法步数 $\widetilde{CT}(p) = n \lceil \frac{n}{p} \rceil$ 加法步为: $\overline{CT}(p) = n \lceil \frac{n}{p} \rceil \lceil \log p \rceil$ 。

引理 4 一个 $n \times 1$ 向量乘以 $1 \times n$ 向量,用 p ($0 < p \leq n$) 台机并行处理,需乘法步 $\lceil \frac{n^2}{p} \rceil$ 次。

引理 5 一个 $1 \times n$ 向量乘以 $n \times 1$ 向量,采用内积算法,用 p ($0 < p \leq n$) 台处理机并行计算,需乘法 $\lceil \frac{n}{p} \rceil$, 加法 $\lceil \frac{n}{p} \rceil \lceil \log p \rceil$ 。

引理 6 两个 $n \times n$ 矩阵相乘, 采用内积算法 2, 用 p 台 ($n < p \leq n^2$) 处理机并行处理, 需乘法步 $n \lceil \frac{n^2}{p} \rceil$, 加法步为 $(n-1) \lceil \frac{n^2}{p} \rceil$

引理 7 两个 $n \times n$ 矩阵相加, 用 p ($n < p \leq n^2$) 台处理机并行处理, 需加法 $\lceil \frac{n^2}{p} \rceil$ 步

引理 8 一个 $n \times n$ 矩阵乘以 $n \times 1$ 向量, 用 p ($n < p \leq n^2$) 台处理机并行计算, 需乘法 $\lceil \frac{n^2}{p} \rceil$ 加法 $\lceil \frac{n^2}{p} \rceil \lceil \log \frac{p}{n} \rceil$ 这里采用内积算法 2。

引理 9 一个 $n \times 1$ 矩阵乘以 $1 \times n$ 矩阵, 用 p ($n < p \leq n^2$) 台处理机并行计算, 需乘法步为 $\lceil \frac{n^2}{p} \rceil$ 加法为零。

引理 10 一个 $1 \times n$ 矩阵乘以 $n \times 1$ 矩阵, 用 p ($n < p \leq n^2$) 台处理机并行计算, 采用内积算法 2 则需乘法步为 1, 加法步为 $\lceil \log n \rceil$ 。

几个基本定义:

速度倍数 (Speed 或 Speedup ratio) 为:

$$S_p(N) = T_1(N) / T_p(N)$$

$T_1(N)$ 是算法使用一台处理机上的运行时间, $T_p(N)$ 是使用 P 台处理机时算法的运行时间。

效率 $E_p(N) = S_p(N) / P$

效益 $\tilde{F}_p(N) = E_p(N) S_p(N)$

并行自适应滤波的算法步骤如下:

算法 2 设 $\hat{x}_{k|Z_k} \in R^n$, 用 P 台处理机, 采用同步并行算法, 迭代一次如下。

1° 并行计算 $P_{k|Z_{k-1}} = \Phi_k P_{k-1|Z_{k-1}} \Phi_k^T + Q_{k-1}$, 各量均为 $n \times n$ 阵

2° 并行计算 $K_k = P_{k|Z_{k-1}} H_k^T (H_k P_{k|Z_{k-1}} H_k^T + R_{k-1})^{-1}$

3° 并行计算 $P_{k|Z_k} = (I - K_k H_k) P_{k|Z_{k-1}}$ I 为单位阵

4° 并行计算 $\hat{x}_{k|Z_{k-1}} = \Phi_{k-1} \hat{x}_{k-1|Z_{k-1}} + q_{k-1}$

5° 并行计算 $\varepsilon_k = Z_k - H_k \hat{x}_{k|Z_{k-1}} - r_{k-1}$

6° 并行计算 $\hat{x}_{k|Z_k} = \hat{x}_{k|Z_{k-1}} + K_k \varepsilon_k$

7° 并行计算 $q_k = \frac{1}{K} ((K-1) \hat{q}_{k-1} + \hat{x}_{k|Z_k} - \Phi_k \hat{x}_{k-1|Z_{k-1}})$

8° 并行计算 $Q_k = \frac{1}{K} ((K-1) Q_{k-1} + K_k \varepsilon_k \varepsilon_k^T K_k^T + P_{k|Z_k} - \Phi_{k-1} P_{k-1|Z_{k-1}} \Phi_{k-1}^T)$

9° 并行计算 $r_k = \frac{1}{K} ((K-1) r_{k-1} + Z_k - H_k \hat{x}_{k|Z_{k-1}})$

10° 并行计算 $R_k = \frac{1}{K} ((K-1) \hat{x}_{k-1} + \varepsilon_k \varepsilon_k^T - H_k P_{k|Z_{k-1}} H_k^T)$

为使用方便, 给出算法 1 的串行计算步数。

$$\text{乘法步 } \widetilde{CT}(1) = 5n^3 + 8n^2 + 8n + 6$$

$$\text{加法步 } \overline{CT}(1) = 6n^3 + 4n^2 + 5n + 3$$

定理 1, 对于算法 2, 如果采用 $P(n < p \leq n^2)$ 台处理机并行计算, 则递推一次需:

$$\text{乘法 } \sum_{i=1}^{10} \widetilde{CT}_i(p) = (5n+6) \left[\frac{n^2}{p} \right] + 14$$

$$\text{加法 } \sum_{i=1}^{10} \overline{CT}_i(p) = 5n \left[\frac{n^2}{p} \right] + 4 \left[\frac{n^2}{p} \right] \left[\log \frac{p}{n} \right] + 4 \left[\log n \right] + 11$$

速度倍数:

$$S_p = \frac{(5L+6)n^3 + (8L+4)n^2 + (8L+5)n + 6L+3}{(5nL+6L+5n) \left[\frac{n^2}{p} \right] + 4 \left[\frac{n^2}{p} \right] \left[\log \frac{p}{n} \right] + 4 \left[\log n \right] + 14L+11}$$

证明: 利用引理 6 至引理 10 即可证得。

定理 2: 对算法 2, 用 $p(o < p \leq n)$ 台处理机并行处理, 则递推一次需:

$$\text{乘法 } \sum_{i=1}^{10} \widetilde{CT}_i(p) = (15n^2 + 4n + 8) \left[\frac{n}{p} \right] + 5 \left[\frac{n^2}{p} \right] + 4$$

$$\text{加法 } \sum_{i=1}^{10} \overline{CT}_i(p) = (5n^2 + 4n + 4) \left[\frac{n}{p} \right] \left[\log p \right] + 2 \left[\frac{n}{p} \right] + 5 \left[\frac{n^2}{p} \right] + 7$$

速度倍数

$$S_p = \frac{(5L+6)n^3 + (8L+4)n^2 + (8L+5)n + 6L+3}{(5L+5) \left[\frac{n^2}{p} \right] + 15n^2L + 4nL + 8n + 2) \left[\frac{n}{p} \right] + (5n^2 + 4n + 4) \left[\frac{n^2}{p} \right]}$$

$$[10gp] + 4L + 7$$

证明: 利用引理 1 至引理 6 即可证得。

下面使用定理 1、定理 2 来计算几个应用实例, 硬件选用 Transputer 的 T8 类板, 其一次乘为 0.9us, 一次加为 0.45us, 结果见下。

表 1 求基波至二次谐波时 $n = 4$

处理机台数	乘法步数	加法步数	时间 us	速度倍数	效率	占 16 点采样间隔百分比
P = 1	486	471	649.35			52%
P = 2	252	251	339.75	1.9	0.95	27%
P = 4	128	229	218.75	2.97	0.74	17.46%
P = 12	66	83	96.75	6.7	0.56	7.74%
P = 16	40	47	57.15	11.36	0.71	4.5%

表 2 求基波至三次谐波时 $n = 6$

处理机台数	乘法步数	加法步数	时间 us	速度倍数	效率	占16点采样间隔百分比
P = 1	1422	1473	1942.65			155%
P = 2	730	727	984.18	1.97	0.98	78.7%
P = 3	488	903	845.55	2.3	0.77	67%
P = 6	246	663	519.75	3.74	0.62	42%
P = 18	86	99	121.95	15.93	0.88	10%
P = 36	50	71	76.95	25.24	0.7	6.1%

五 处理机台数 p 的优化结果

我们讨论的最优性有三种含义：

- 1) 讨论处理机台数 p 取何值时，并行自适应滤波算法的计算时间最短。
- 2) p 取何值时， $\widetilde{F}_p(N) = E_p(N) S_p(N)$ 为最大，其中 $E_p(N)$ 是效率， $S_p(N)$ 是速度倍数， $F_p(N)$ 是效益。
- 3) 在满足一定的时间要求内，求最少的处理机台数 P 以使得系统硬件成本尽可能降低。

据定理 1 和定理 2 已求得并行自适应算法的计算复杂度

$$n < p \leq n^2 \text{ 时乘法 } \Sigma C\widetilde{T}(p) = (5n+6) \left[\frac{n^2}{p} \right] + 14$$

$$\text{加法 } \Sigma C\overline{T}(p) = 5n \left[\frac{n^2}{p} \right] + 4 \left[\frac{n^2}{p} \right] \left[\log \frac{p}{n} \right] + 4 \left[\log n \right] + 11$$

$$p \leq n \text{ 时 乘法 } \Sigma C\widetilde{T}(p) = (15n^2 + 4n + 8) \left[\frac{n}{p} \right] + 5 \left[\frac{n^2}{p} \right] + 4$$

$$\text{加法 } \Sigma C\overline{T}(p) = (5n^2 + 4n + 4) \left[\frac{n}{p} \right] \left[\log p \right] + 2 \left[\frac{n}{p} \right]$$

对于计算时间而言，显然 $p = n^2$ 时可达最短，此时乘法步数仅为：

$$\Sigma C\widetilde{T}(n^2) = 5n + 20, \text{ 加法步数仅为: } \Sigma C\overline{T}(n^2) = 5n + 8 \left[\log n \right] + 11$$

再来看 $\widetilde{F}_p(N) = E_p(N) S_p(N) = S_p^2(N) / P$

$$= \begin{cases} \frac{[(15L+6)n^3 + (8L+4)n^2 + (8L+5)n + 6L+3]^2}{P \left[(15L+5) \left[\frac{n^2}{p} \right] + (5n^2L+4nL+8n+2) \left[\frac{n}{p} \right] (5n^2+4n+4) \left[\frac{n^2}{p} \right] \right.} \\ \left. [\log p] + 4L+7 \right]^2 \quad (0 < p \leq n) \\ \frac{[(5L+6)n^3 + (8L+4)n^2 + (8L+5)n + 6L+3]^2}{P \left[(5nL+6L+5n) \left[\frac{n^2}{p} \right] + 4 \left[\frac{n^2}{p} \right] \left[\log \frac{p}{n} \right] + 4 \left[\log n \right] + 14L+11 \right]^2} \\ \quad (n < p \leq n^2) \end{cases}$$

欲求何时效益 $\tilde{F}_N(p)$ 达到最大, 由于 L, n 都是相对固定的值, 因此需求何时上式分母达最小值, 为此我们对上式分母求导, 为书写方便可令: $A = 5L + 5$ $B = 5n^2L + 4nL + 8n + 2$ $C = 5n^2 + 4n + 4$

于是 $P \leq n$ 时, 分母写为:

$$P \left(A \left[\frac{n^2}{p} \right] + B \left[\frac{n}{p} \right] + C \left[\frac{n^2}{p} \right] [\log p] + 4L + 7 \right)^2$$

对 P 求导数并经整理为: (假定 P 能整除 n)

$$- \left(A \frac{n^2}{p} + B \frac{n}{p} + C \frac{n^2}{p} \log p + 4L + 7 \right) \left(\frac{An^2}{p} + \frac{Bn}{p} + \frac{Cn^2}{p} \right)$$

由于括号内各项均为正值, 因此导数恒小于零, 分母为递减的, 因此, $p = n$ 时, 可达极小。

同理可知 $n < p \leq n^2$ 时, $p = n^2$ 可达极小, 两者综合可知 $p = n^2$ 时, 效益 $F_N(p)$ 为最优

$$\tilde{F}_N(p) = \frac{[(5L+6)n^3 + (8L+4)n^2 + (8L+5)n + 6L+3]^2}{[5nL + 20L + 5n + 8(\log n) + 11]^2 P}$$

假定我们容许的计算时间为 T_{ALL} , 则我们求满足下式的最小的 P 即为所求的最优处理机台数 p , 即 $L(\sum CT(p)) + \sum CT(p) \leq T_{ALL}/T$, T_e 为处理机一步加的运算时间。

对于 $0 < p \leq n$ 和 $n < p \leq n^2$ 时的情形, 写出来即是: $p \leq n$ 时, $(5L+5) \left[\frac{n^2}{p} \right] +$

$$(5n^2L + 4nL + 8n + 2) \left[\frac{n}{p} \right] + (5n^2 + 4n + 4) \left[\frac{n^2}{p} \right] [\log p] + 4L + 7 \leq T_{ALL}/T_e \quad (1)$$

$n < p \leq n^2$ 时, $(5nL + 6L + 5n) \left[\frac{n^2}{p} \right] + 4 \left[\frac{n^2}{p} \right] [\log \frac{p}{n}] + 4(\log n) + 14L + 11$

$$\leq T_{ALL}/T_e \quad (2)$$

对于(1)式, 假定 $\frac{n}{p} = x$, 可化为下式

$$(5n^2L + 9nL + 13n + 2)x + (5n^2 + 4n + 4)nx \log \frac{n}{x} + 4L + 7 \leq T_{ALL}/T_e$$

求上式左端的导数得

$$[5n^2(L-1) + 9n(L-1) - 2] + (5n^2 + 4n + 4) \log \frac{n}{x}$$

由于 $n \geq 1, L > 1, x > 0$, 因此上式大于零, 可见原式(3)为递增函数, 因此, 解出(3)中等式成立时的 x 值即可求得最小的 p , 但(3)是一超越方程, 为此利用不等式 $\log \frac{n}{x} \leq \frac{n}{x}$, 用 $\frac{n}{x}$ 代替 $\log \frac{n}{x}$ 解出此时的 x , 则(3)式一定也能成立, 将 $\frac{n}{x}$ 代替(3)

中的 $\log \frac{n}{x}$ 后(3)式成为:

$$(5n^2 + 9nL + 13n + 2)x + (5n^2 + 4n + 4)n^2 + 4L + 7 = T_{ALL}/T_e$$

解得:

$$x = \frac{T_{ALL}/T_e - 5n^2 - 4n^3 - 4n^2 - 4L - 7}{5n^2L + 9nL + 13n + 2}$$

$$\text{处理机台数 } P_{m;n} = \frac{n}{x} = \frac{n[5n^2L + 9nL + 13n + 2]}{T_{ALL}/T_e - 5n^2 - 4n^3 - 4n^2 - 7}$$

对于(2)式采取同样的方法可求解得:

$$P_{m;n} = \frac{5n^3L + 6n^2L + 5n^3}{T_{ALL}/T_e - 4n - 41\log n - 14L - 11}$$

由于 $P_{m;n}$ 的范围在 $0 < p \leq n$ 之内,因此尚需满足下式:

$$0 < \frac{5n^2L + 9nL + 13n + 2}{T_{ALL}/T_e - 5n^2 - 4n^3 - 4n^2 - 4L - 7} \leq 1$$

$P_{m;n}$ 的范围在 $n < p \leq n^2$,因此需满足下式:

$$\frac{1}{n} < \frac{5nL + 6L + 5n}{T_{ALL}/T_e - 4n - 41\log n - 14L - 11} \leq 1$$

致谢

本文得到西安交通大学葛耀中教授,吴振国副教授的细心指点和帮助,特此致谢。

参 考 文 献

- [1] A.A.Girgis and David G.Hart, Implementation of Kalman an abaptive Kalman Filtering algorithms for digital distanle pre-d tion on a vector signal process,IEEE Transactions on Power Delivey Vol.4.No.1 Fanuary, 1989.
- [2] R.K Mehra.Approouhes to Adaptive Filtening,IEEE Transatcions on Auto matic confrol, Octoke 1972 A C-17.
- [3] “离散时间系统滤波的数学方法”中国科学数学研究所概率组.
- [4] 张丽君、陈增荣.并行算法的设计与分析,湖南科学技术出版社.
- [5] 英Y柏克.多微处理器系统.
- [6] 《计算机仿真》——Transputer特辑1990.1.
- [7] 第一届全国科学计算并行算法论文集.1989.6.
- [8] 英R·W霍克尼 C·R杰斯荷普.并行计算机一体系统结构,程序设计及算法.清华大学出版社.