

文章编号: 1001-2486(2009)03-0053-07

## 基于 DSP 的 SMI 方法快速实现研究\*

任磊<sup>1,2</sup>, 王永良<sup>2</sup>, 陈辉<sup>2</sup>, 陈建文<sup>2</sup>

(1. 国防科技大学 电子科学与工程学院, 湖南 长沙 410073;

2. 空军雷达学院 兵器运用工程军队重点实验室, 湖北 武汉 430019)

**摘要:** 空时自适应处理(STAP) 权值计算有数据域和均方域两种方法, 分别以 QR 分解和样本协方差矩阵求逆(SMI) 方法为代表。QR 分解方法可以映射到脉动阵上并行实现, 但实现复杂且设计成本较高; SMI 方法实现则相对简单, 但需要对样本协方差矩阵直接求逆。首先考察了不同矩阵求逆方法的内在并行性, 基于 DSP 支持的片内并行技术, 提出并实现了 SMI 方法的单 DSP 分块并行处理, 进一步给出了数值稳定性分析和改善方法, 实验结果证明了方法的有效性。

**关键词:** 分块矩阵; 片内并行; 数值稳定性

中图分类号: TN957 文献标识码: A

## Research on Fast Implementation of SMI Method Based on DSP

REN Lei<sup>1,2</sup>, WANG Yong-liang<sup>2</sup>, CHEN Hui<sup>2</sup>, CHEN Jian-wen<sup>2</sup>

(1. College of Electronic Science and Engineering, National Univ. of Defense Technology, Changsha 410073, China;

2. Key Laboratory of Radar Armament and Utilization Engineering, Air Force Radar Academy, Wuhan 430019, China)

**Abstract:** Two methods are used to compute the STAP weights in application, named as data domain algorithm and mean square domain algorithm, which are represented by QR decomposition algorithm and sample matrix inversion (SMI) algorithm respectively. In comparison, SMI algorithm can be realized simply with sample covariance matrix inversion, QR decomposition algorithm can be mapped onto systolic array with a high design cost. The inner parallelism of different algorithms of matrix inversion is reviewed firstly, the implementation of SMI method based on single DSP is proposed based on the on-chip parallel technique, the numerical stability of the proposed method is also given, and experiments with the data measured verify the validity of the proposed method.

**Key words:** block matrix; on-chip parallel; numerical stability

STAP 技术的研究自 Brennan 等在 1973 年提出到现在, 已历经三十多年的发展<sup>[1]</sup>, 目前研究的热点难点问题之一是大数据流下的高精度实时系统实现, 其中权值计算需要联合多通道数据进行, 运算量大且频繁, 往往是实时处理的瓶颈之一。权值计算有数据域和均方域两种方法, 分别以 QR 分解和 SMI 方法为代表。QR 分解方法可以被映射到脉动阵并行实现, 然而脉动阵实现的结构缺乏通用性和可重构性, 导致这种设计的费用太高, 所以某些系统的设计并不适合采用这种结构。SMI 方法需要对样本协方差矩阵直接求逆, 较 QR 分解方法数值动态范围大, 对误差较敏感, 但实现相对简单, 且随着高性能处理器的发展, 数值精度已不再是主要困扰。DSP 在功耗、易开发性和浮点精度等方面有优势, 因此常常用在复杂算法和对数值精度要求高的场合。本文基于 DSP 支持的片内并行技术, 提出并实现了 SMI 方法的分块并行处理, 还进一步讨论了处理的数值稳定性, 给出了适合工程的数值稳定性改善方法, 实验结果证明了方法的有效性。

## 1 矩阵求逆

矩阵求逆应用广泛<sup>[2]</sup>, 虽然其串行实现的运算量经过 Strassen 算法<sup>[3]</sup> 改进后由  $O(N^3)$  量级降低到

\* 收稿日期: 2008-12-08

基金项目: 国家自然科学基金重点资助项目(60736009)

作者简介: 任磊(1980-), 男, 讲师, 博士生。

$O(N^{\log_2 7})$  量级,但对于包含高维矩阵求逆或需要频繁矩阵求逆的数值处理应用,矩阵求逆处理仍可能是实时实现的瓶颈之一。文献[4]通过比较认为,迭代法适用于对数值精度要求较高而对实时性要求不高的场合,而直接求逆法能更好地满足实时性要求。STAP 系统对实时性要求很高,所以本文重点研究直接矩阵求逆方法。伴随矩阵求逆法和初等变换法不具有很好的内在并行性,而分块矩阵求逆法是基于递归分裂的思想,本身具有好的内在并行性,基于此思想可以加速矩阵求逆。

## 2 分块矩阵求逆并行性分析及实现

分块矩阵求逆基于递归分裂的思想,具有一定的内在并行性,常见的分块矩阵求逆有以下三种<sup>[2]</sup>:

$$A^{-1} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}^{-1} = \begin{bmatrix} S_1 & -S_1 S_2 \\ -S_3 S_1 & S_4 + S_3 S_1 S_2 \end{bmatrix} \quad (1)$$

其中,  $S_1 = (A_{11} - A_{12} A_{22}^{-1} A_{21})^{-1}$ ,  $S_2 = A_{12} A_{22}^{-1}$ ,  $S_3 = A_{22}^{-1} A_{21}$ ,  $S_4 = A_{22}^{-1}$ , 或

$$A^{-1} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}^{-1} = \begin{bmatrix} S_1 & -A_{11}^{-1} A_{12} S_2 \\ -A_{22}^{-1} A_{21} S_1 & S_2 \end{bmatrix} \quad (2)$$

其中,  $S_1 = (A_{11} - A_{12} A_{22}^{-1} A_{21})^{-1}$ ,  $S_2 = (A_{22} - A_{21} A_{11}^{-1} A_{12})^{-1}$ , 或

$$A^{-1} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}^{-1} = \begin{bmatrix} (A_{11} - A_{12} A_{22}^{-1} A_{21})^{-1} & (A_{21} - A_{22} A_{12}^{-1} A_{11})^{-1} \\ (A_{12} - A_{11} A_{21}^{-1} A_{22})^{-1} & (A_{22} - A_{21} A_{11}^{-1} A_{12})^{-1} \end{bmatrix} \quad (3)$$

由上面的三个公式可以看出,每一子块都可以与其他子块并行计算,即具有内在并行性。假设可逆矩阵  $A$  的阶数为  $N = 2^n$ , 对其进行递归分解,这个求逆过程包括的操作主要有子矩阵加减(+/-)、子矩阵乘法( $\times$ )、子矩阵求逆(Inv)三种。对  $A$  分块一次,得到 4 个  $N/4$  阶的子阵  $A_{ij}$  ( $i = 1, 2; j = 1, 2$ )。下面分析比较它们采用不同实现方法的运算量。

### 2.1 分析 1: 串行实现

把  $A$  分解一次,可得 4 个  $N/4$  阶的子阵,根据式(1)~(3),以串行方法实现一次递归求逆的运算量分析为表 1,表中数字为进行相应操作的次数,表 2 同。

由表 1 可以看出,当采用串行方法求逆矩阵时,需要逐一计算逆矩阵的各分块部分,运算量为各部分之和,此时式(1)的运算量最小,而式(3)的运算量最大。下面基于各公式的内在并行性来分析采用并行方法实现时的运算量。

### 2.2 分析 2: 并行实现

因为式(1)~(3)中各个子块的逆可并行计算,则可知矩阵求逆的时间应以最耗时部分为标准。设每子块的求逆计算为串行,此时的运算量分析见表 2。

由表 2 可见,式(1)的运算量变成了最大,而式(3)的运算量最小,且因为运算操作步骤少,根据水桶原理,运算操作越少,结果的数值精度越高,此时式(3)体现出更好的平衡性。

表 2 并行实现运算量分析(单位:运算次数)

各公式中最耗时部分	+/-	$\times$	Inv	比较
式(1)中的 $S_4 + S_3 S_1 S_2$	2	6	2	最大
式(2)中的 $-A_{11}^{-1} A_{12} S_2$	1	4	2	居中
式(3)中的 $(A_{11} - A_{12} A_{22}^{-1} A_{21})^{-1}$	1	2	2	最小

由上面分析可知,若构建含有 4 个处理节点的并行系统,就可以发挥分块矩阵求逆的内在并行性。本文基于指令级和数据级的片内并行技术,提出并实现了 SMI 方法的单 DSP 分块并行处理,无需构建

表 1 串行实现运算量分析(单位:运算次数)

Tab. 1 Analysis of computational load of serial realization (unit: operating times)

公式	+/-	$\times$	Inv	比较
(1)	2	8	2	最小
(2)	2	8	4	居中
(3)	4	8	8	最大

多节点并行系统。下面给出具体思想和方法。

如果采用多节点并行系统,其加速比往往会受到处理器间的通信与同步的制约,且增加了处理成本。高性能 DSP 支持片内并行技术,如哈佛结构、超标量、单指令多数据流(SIMD)和超长指令字(VLIW)、多级深度指令流水等。利用 SIMD 技术和 VLIW 技术的并行可以理解为“空间”的并行处理,配合多级流水技术,则构成了“空时”的片内并行技术。基于该技术,本文基于单 DSP 从数据级和指令级着手,提出并实现了 SMI 方法的单 DSP 分块并行处理。

TS201 在一个时钟周期内可以执行完 4 条指令,同时读写不同数据。令  $NN = (N/2) \times (N/2)$ ,且假设已知  $A_j(i) (i = 1, 2; j = 1, 2)$  的逆  $A_j^{-1}$  (通过分裂递推得到),则式(3)的数据读取、乘法和加法可以表示为下面的描述代码。其中,“;”表示指令行的结束。

```
for (i = 0; i < NN; i++) {
    //数据读写代码举例如下
    fetch(A11(i); fetch(A12(i)); fetch(A21(i)); fetch(A22(i)); ;
    fetch(A11-1(i)); fetch(A12-1(i)); fetch(A21-1(i)); fetch(A22-1(i)); ;
    .....
    //乘法和加法代码举例如下
    r3:2 = q[j0+ j1]; yr5:4 = l[k8+ = k9]; fr23 = r19* r21; fr11 = r8+ r9; ;
}
```

片内并行处理的优势在于:(1)避免了多处理器间的通信和同步,简化方案,节约实现成本;(2)VLIW 和 SIMD 技术从指令级和数据级的优化使编译后的代码更高效;(3)4 子块求逆的运算操作相似,代码具有通用性,只需关注合理分配寄存器;(4)4 子块求逆的运算耗时理论上是相同的,所以运算时间是紧凑的。

### 3 分块矩阵求逆的数值稳定性

从数学角度来讲,STAP 权值的求解是在最小平方误差准则下求解超定线性方程组的最小二乘问题,需显式计算采样协方差矩阵,相当于对采样数据进行平方运算,大大增加了数据动态范围和矩阵的条件数,使得在许多实际情况下协方差矩阵病态加剧,影响算法数值特性,尤其在矩阵求逆时这种影响比较严重,本文将对此进行分析研究。矩阵的病态对数值稳定性的影响的讨论已经很多,本文主要从工程角度讨论舍入误差和数值溢出的影响<sup>[5]</sup>。舍入误差对滤波器输出的影响总是以一种随机扰动的方式存在,可以利用适当的噪声分析模型和线性系统理论进行分析。而内部溢出必须防止,因为它们会产生更大的误差,尤其是对于有限数值动态范围的处理器而言。

#### 3.1 舍入误差

舍入噪声的根源主要在于乘法器,是由于对结果进行舍弃或截断量化产生的。数学上把一个带有舍入操作的系统可以抽象为一个带有外部误差输入的、具有无限精度的系统,对乘加操作的输出进行舍入后的等价模型表示如图 1 所示<sup>[5]</sup>。

此时舍入误差  $e(n)$  可以假设是均匀分布的白噪声,是一个广义静态随机过程,且与信号不相关。因此按照线性系统内理论分析,设输出字长为  $b$  比特,舍入误差可以表示为

$$-\frac{2^{-(b-1)}}{2} \leq e(n) \leq \frac{2^{-(b-1)}}{2} \quad (4)$$

其均值为 0,方差为  $\sigma_e^2 = 2^{-2b}/3$ ,即每增加 1bit 的字长,误差就可以减少 4 倍。式(3)中各子块运算步骤相似,具有较好的误差均衡性。不失一般性,这里仅分析其中一个子块,以  $(A_{11} - A_{12}A_{22}^{-1}A_{21})^{-1}$  为例,设矩阵  $A$  为  $2N \times 2N$  维,  $A_j (i, j = 1, 2)$  为  $N \times N$  维。若  $A_{22}^{-1}$  已知,则舍入误差存在于  $A_{12}A_{22}^{-1}A_{21}$  的两次

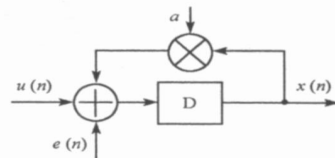


图 1 舍入误差模型  
Fig. 1 The model of round off errors

复数矩阵乘法和 $(\Theta)^{-1}$ 的求逆中,其中, $\Theta = A_{11} - A_{12}A_{22}^{-1}A_{21}$ 。两个 $N \times N$ 维复矩阵乘法的结果仍然为 $N \times N$ 维复矩阵,结果矩阵中的每一元素都是经过 $N$ 次复数乘法和 $N-1$ 次复数加法得到,而 $N$ 次复数乘法又含有 $4N$ 次实数乘法和 $2N$ 次实数加法。一次复数乘法为

$$p = (a + bj)(c + dj) = (ac - bd) + (ad + bc)j \quad (5)$$

引入舍入误差后表示为

$$Q(p) = [(ac + \varepsilon_1) - (bd + \varepsilon_2)] + [(ad + \varepsilon_3) + (bc + \varepsilon_4)]j = (ac - bd) + (ad + bc)j + \varepsilon_m \quad (6)$$

其中, $\varepsilon_m = \varepsilon_1 - \varepsilon_2 + (\varepsilon_3 + \varepsilon_4)j$ , $\varepsilon_p (p = 1, 2, 3, 4)$ 在 $[-\frac{1}{2}2^{-b}, \frac{1}{2}2^{-b}]$ 之间服从均匀分布且互不相关,与输入信号数据也不相关,所以

$$E[|\varepsilon_p|^2] = \frac{1}{12}2^{-2b}, \quad p = 1, 2, 3, 4 \quad (7)$$

则

$$|\varepsilon_m|^2 = |\varepsilon_1 - \varepsilon_2|^2 + |\varepsilon_3 + \varepsilon_4|^2 \quad (8)$$

复数噪声 $\varepsilon_m$ 的方差用 $\sigma_{\varepsilon_m}^2$ 表示,则

$$\sigma_{\varepsilon_m}^2 = E[|\varepsilon_m|^2] = \sum_{i=1}^4 E[|\varepsilon_i|^2] = 4 \cdot \frac{1}{12}2^{-2b} = \frac{1}{3}2^{-2b} \quad (9)$$

两次复数矩阵乘法的舍入误差与两次复数乘法的舍入误差量级相同,根据以上分析可知,两次复数乘法后引入的舍入误差表示为复数噪声的方差为

$$\sigma_{\varepsilon_{mn}}^2 = 4\sigma_{\varepsilon_m}^2 = \left(\frac{4}{3}\right)2^{-2b} \quad (10)$$

两个 $N \times N$ 维矩阵相乘得到的结果矩阵中的每一元素是经过 $N$ 次如上复数乘法,所以引入舍入噪声

$$\sigma_{\varepsilon_{mnn}}^2 = 4N\sigma_{\varepsilon_m}^2 = \left(\frac{4N}{3}\right)2^{-2b} \quad (11)$$

对 $(\Theta)^{-1}$ 的求逆采用递归的方法来求解,即把 $N \times N$ 维的子矩阵进一步分解成4个 $(N/2) \times (N/2)$ 维的子矩阵,可知以此方法一直分解到 $2 \times 2$ 维矩阵,共需 $(\log_2 N - 1)$ 次,结果矩阵中的每一元素都是经过

$$T = N + \frac{N}{2} + \frac{N}{2^2} + \dots + \frac{N}{2^{\log_2 N - 1}} = 2N(1 - 2^{-\log_2 N}) \quad (12)$$

次复数乘法(当 $N$ 比较大时, $T = 2N$ ),所以引入的舍入误差表示为复数噪声的方差为

$$\sigma_{\varepsilon_{N}}^2 = T \cdot \sigma_{\varepsilon_{mnn}}^2 = 2N(1 - 2^{-\log_2 N}) \cdot \left(\frac{4}{3}\right)2^{-2b} \quad (13)$$

当 $N = 2$ 时, $\sigma_{\varepsilon_N}^2 = \left(\frac{8}{3}\right)2^{-2b}$ 表示 $2 \times 2$ 维矩阵的求逆舍入误差,因为此时求矩阵的模也要进行复数的乘法。可见,舍入误差的传递随着矩阵维数的增加而变大,式(1)的曲线如图2所示。从图3中有效字长

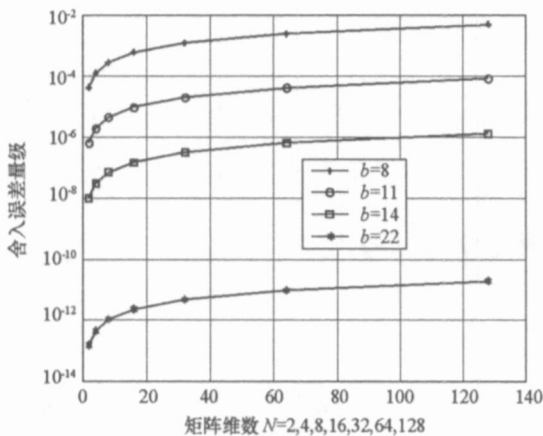


图2 舍入误差随矩阵阶数增加的变化曲线

Fig. 2 The curve of round off errors and dimension of matrix

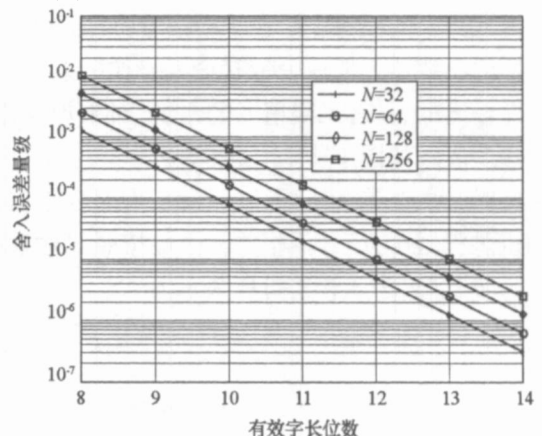


图3 有效字长对舍入误差的影响

Fig. 3 The influence of effective word on round off errors

对舍入误差的影响可以进一步看出,舍入误差可以通过增加有效字长而被控制在合理范围内。

### 3.2 数值溢出

SMI 方法计算权值的过程中有许多乘加运算,数值累加会产生数值溢出。数值溢出会产生更大的误差,尤其是当接收信号的动态范围很大、杂波干扰非均匀非平稳时。为防止内部溢出,可以采用缩放的方法避免产生更大的误差。缩放(Scaling)是在保证传输函数从输入到输出不变的前提下,重新调整一定的内部增益参数,以便限制内部信号的范围能够适应硬件的一种处理。文献[6]中给出了通过构造一对角矩阵来平衡复正定 Hermitian 矩阵的方法(样本协方差矩阵是 Hermitian 矩阵)。文献[7]也对 Hermitian 矩阵的缩放稳定性做了分析。本文对其机理分析讨论如下。

若矩阵  $R = [r_{ij}]$ , 则构造对角缩放矩阵  $T$ , 其对角元素为  $t_{jj} = 1/\sqrt{r_{jj}}$ , 则矩阵  $R_{sc} = TRT$  的对角元素等于 1, 矩阵  $T$  能够有效地平衡矩阵  $R$ , 同时可以减少因变量字长不同而引发的硬件开销和不规则设计, 且根据文献[6]可知, 此时  $R_{sc}$  的条件数将是所有对角缩放  $R$  后得到的矩阵中最小的。为了避免额外因变量字长不同而引发的不规则性, 系统希望所有的变量指定相同的字长, 而这只有在所有的状态具有相同的能量时才行。假设有缩放矩阵  $T$ , 则缩放后的状态  $X_s$  与缩放前的状态  $X$  之间的关系定义为  $X_s = TX$ , 缩放后的状态协方差矩阵  $R_{sc}$  与缩放前的状态协方差矩阵  $R$  有如下关系:

$$R_{sc} = E[XX_s^T] = E[TXX^T T^T] = TE[XX^T] T^T = TRT^T \quad (14)$$

由于希望所有的状态都具有相同的能量, 所以选择变换矩阵  $T$  使缩放后的系统矩阵  $R_{sc}$  的主对角线上的元素都为 1。进一步, 假设  $T$  是对角矩阵, 即  $T = T^T = \text{diag}[t_{11} \ t_{22} \ \dots \ t_{NN}]$ , 则

$$T^{-1} = (T^T)^T = \text{diag}[1/t_{11} \ 1/t_{22} \ \dots \ 1/t_{NN}] \quad (15)$$

所以  $(R_{sc})_{ii} = r_{ii} \cdot t_{ii}^2 = 1$ , 所以  $t_{ii} = 1/\sqrt{r_{ii}}$ , 即通过选取矩阵  $T$  的对角线上的第  $i$  个元素为矩阵  $R$  的对角线上第  $i$  个元素的平方根, 就可以保证所有状态都有相同的能量。当矩阵求逆时, 先得到  $R_{sc} = TRT$ , 然后求  $R_{sc}^{-1}$ , 则  $R^{-1} = TR_{sc}^{-1}T$ , 这样会比直接求  $R^{-1}$  的数值精度高。下面以试验数据验证上述方法的有效性。

## 4 实测数据实验

本节首先给出 SMI 方法并行分块求逆的时间加速比较实验结果, 然后进行了数值稳定性测试, 结果表明本文方法得到较理想的加速比, 经过缩放也有效地改善了矩阵条件数和数值精度。这里采用的矩阵是在 STAP 系统中采用 3-CAP STAP 法, 联合多空域通道的对应多普勒通道数据形成的协方差矩阵, 根据空域通道数不同, 形成的协方差矩阵维数不同。

### 4.1 加速比实验

采用第 2 节中的方法, 通过并行数据读取, 并行指令执行, 从数据级和指令级进行优化, 理论上最优的情况下加速比可以接近 4。下面比较串行和并行实现式(3)的时间, 见表 3。测试环境为 Visual DSP+ + 4.0, EZ-KIT 测试版 V1.1。从表 3 可以看出, 加速比随着维数的增加而增大, 这是因为随着矩阵维数增加, 在数据级和指令级能进行的优化比例增加, 但这种增幅会逐步减小。如果进一步利用协方差矩阵是 Hermitian 矩阵的特点(即复共轭对称), 还可以减少并行处理时间。

表 3 串行与并行实现矩阵求逆的时间比较(单位:  $\mu\text{s}$ )

Tab. 3 The time comparisons between serial realization and parallel realization (unit:  $\mu\text{s}$ )

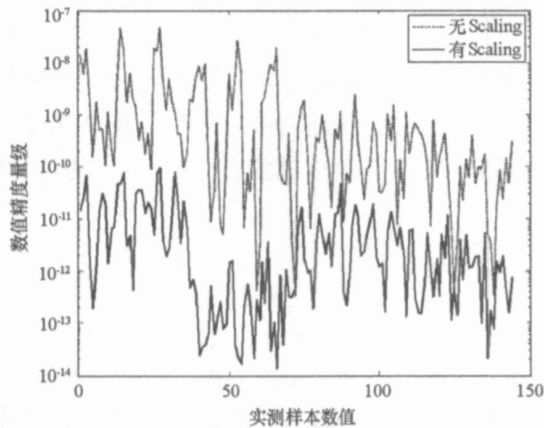
矩阵维数	串行实现	并行实现	加速比
6	32.956	11.511	2.863
9	75.424	25.681	2.937
12	141.578	47.382	2.988
18	367.904	121.421	3.030

## 4.2 数值稳定性实验

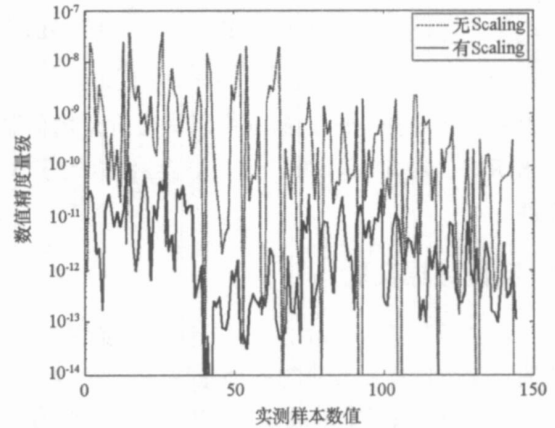
有限字长效应产生的舍入噪声误差大小取决于系统的结构、数值的表示方法和运算算法等因素。下面给出基于TS201对实测数据的分块矩阵求逆进行浮点数值精度分析,结果表明经过缩放可以有效地改善数值精度和矩阵条件数。

首先看对矩阵条件数的改善。由表4可见,不同批号的数据在权值计算前形成的协方差矩阵的条件数都有不同程度的改善。下面结合两个实验给出缩放操作前后数据协方差矩阵在TS201上进行分块矩阵求逆的数值精度改善对比。

实验一: Scaling前后矩阵求逆结果的误差分析。测试情况方法是把# 0002批数据在TS201上进行求逆,分无Scaling和有Scaling两种情况。把用Matlab运算得到的结果(当作真值)分别和上面两种情况比较,得到差值取绝对值后的数量级。由图4中可以看出,实部和虚部经过Scaling后的误差比没有Scaling的误差小了约3个数量级,波动也小了许多。



(a) 实部的误差量级比较



(b) 虚部的误差量级比较

图4 实测数据协方差矩阵Scaling前后的求逆结果比较

Fig. 4 The comparisons of inversion results between "before Scaling" and "after Scaling" of the covariance matrix of measure data

实验二: 有无Scaling两种情况的矩阵求逆结果分别乘回原矩阵,得到的结果取绝对值后与单位阵进行误差比较。测试方法是把# 0002批数据在TS201上进行求逆,分无Scaling和有Scaling两种情况。把得到的结果分别和原矩阵相乘,与单位阵进行比较,得到差值取绝对值后的数量级,该实验进一步测试了误差的传递性。由图5中可以看出,Scaling之前的误差在一个不能接受的范围,经过Scaling后的误差比没有Scaling的误差小了约3个数量级,波动也小了许多。

上面实验中的误差均值与方差表示在表5。从表5中可以看出缩放操作的有效性,误差均值变小说明数值精度提高,误差方差变小说明数值稳定性也被改善;从实验二还可以看出,Scaling方法对减弱误差的传递性效果也很明显。Scaling方法实现简单,运算量低,很适合对数值精度和数值稳定性要求高的场合使用。

表4 不同批号数据协方差矩阵

Scaling前后条件数比较

Tab. 4 The comparisons of condition number between "before Scaling" and "after Scaling" of covariance matrix of different data

数据批号	Scaling前	Scaling后
# 0001	1.3725 e+ 005	6.8042 e+ 004
# 0002	1.3614 e+ 005	6.2737 e+ 004
# 0003	1.6276 e+ 005	1.5198 e+ 004
# 0004	1.1067 e+ 005	5.7199 e+ 004
# 0005	1.2346 e+ 004	3.3035 e+ 003

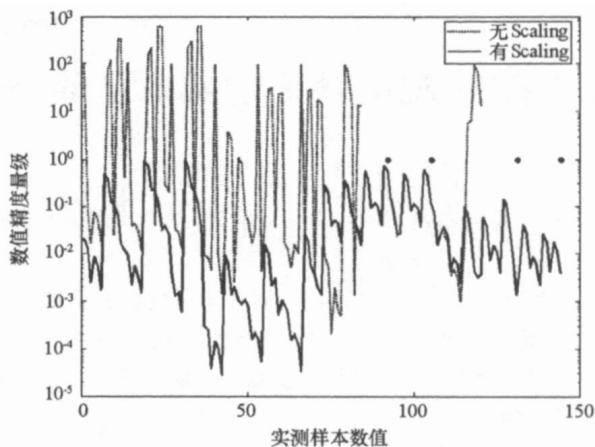


图 5 Scaling 前后矩阵求逆结果乘回原矩阵后与单位阵比较的误差量级

Fig. 5 The comparisons of future results between “before Scaling” and “after Scaling” of the covariance matrix of measure data

表 5 两实验的误差统计

Tab. 5 Error statistics of the above two experiments

	误差均值		误差方差	
	Scaling 前	Scaling 后	Scaling 前	Scaling 后
图 4(a)	2.553e-009	8.229e-012	4.554e-017	2.774e-022
图 4(b)	2.270e-009	8.282e-012	3.722e-017	2.797e-022
图 5	37.724	0.092	1.300e+004	0.032

## 5 结束语

SMI 方法是自适应处理计算权值的方法之一,与均方域 QR 分解方法比较,实现则相对简单,但需要对样本协方差矩阵直接求逆。本文首先考察了矩阵求逆的内在并行性,基于 DSP 支持的片内并行技术,提出并实现了 SMI 方法的单 DSP 分块并行处理,进一步讨论了方法的数值稳定性,实验结果证明方法的有效性。本文实现的过程中,只是利用了各子块之间的并行,若是子块内也进行数据级和指令级的并行优化,应该还可以得到更好的加速比。随着高性能处理器的发展,对片内并行技术更好的支持,并行系统不仅是多处理节点间的并行,还应挖掘优化每个处理节点的潜力。

## 参考文献:

- [1] Brennan L E, Reed I S. Theory of Adaptive Radar[J]. IEEE Trans. AES, 1973, 9(2): 237- 252.
- [2] 张贤达. 矩阵分析与应用[M]. 北京: 清华大学出版社, 2005:69- 70.
- [3] Press W H, Teukolsky S A. C 数值算法[M]. 傅祖芸, 丁岩石,等译. 北京: 电子工业出版社, 2004: 76- 77.
- [4] Ylinen M, Burian A, Takala J. Direct Versus Iterative Methods for Fix-point Implementation of Matrix Inversion[C]//IEEE ISCAS, 2004: 225- 228.
- [5] Parhi K K. VLSI 数字信号处理系统设计与实现[M]. 陈弘毅, 白国强, 吴行军,译. 北京: 机械工业出版社, 2004:277- 287.
- [6] Higham N J. Accuracy and Stability of Numerical Algorithm[M]. SIAM Philadelphia, 2002.
- [7] 赵雪. Hermite 矩阵的  $r$ - 性质及 Scaling 稳定性分析[J]. 北华大学学报(自然科学版), 2006, 7(5): 392- 394.