

并行实现空时自适应算法

范西昆^{1,2}, 王永良², 陈 辉², 李 强², 母其勇²

(1. 国防科学技术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 430010)

摘要: 针对多片 DSP(数字信号处理器) 并行处理系统, 研究了空时自适应处理(STAP) 算法的实时实现. 在分析了部分自适应 STAP 算法内在并行性的基础上, 提出了一种任务级的 STAP 并行处理算法. 给出了该算法在多 DSP 并行处理系统上的算法映射. 通过数据实验证明这种基于多 DSP 并行处理系统的 STAP 并行算法具有较高的实时性能.

关键词: STAP; 并行处理; DSP

中图分类号: TN957 **文献标识码:** A **文章编号:** 0372-2112(2005)12-2222-04

Parallel Implementation of Space-Time Adaptive Processing Algorithms

FAN Xi-kun^{1,2}, WANG Yong-liang², CHEN Hui², LI Qiang², MU Qi-yong²

(1. School of Electronic Science and Engineering, NUDT, Changsha, Hunan 410073, China;

2. Key Lab of Wuhan Radar Academy, Wuhan, Hubei 430010, China)

Abstract: Space-time adaptive processing (STAP) technique that is well known in the area of airborne radars requires very high computing power. In order to meet the real-time requirements in STAP applications, Multi-DSP system is used. In this paper, a method is presented to design parallel algorithms of partially adaptive STAP approaches. Data remapping is used between successive computation steps of STAP. Task-level parallelism is used for these steps. As an example, the method is applied to Element-space post-Doppler STAP approaches. We also implemented the parallel STAP algorithm on a Multi-DSP system. The effectiveness of our implementation is demonstrated with experimental results. The STAP algorithm is highly signal processing oriented and hence executes efficiently on DSP-based systems.

Key words: STAP; parallel processing; DSP

1 引言

STAP 实质上是一个多普勒频率与空间频率的二维滤波问题, 用于检测淹没在强地杂波和各种干扰中的目标^[1]. 实时性是雷达系统的本质特征之一, 而 STAP 达到实时处理的关键点在于系统能否在线实时计算出自适应权. 对实时 STAP 的实现可分为软件方法和硬件方法. 硬件实现的方法是通过某种半导体工艺来设计 STAP 算法的专用芯片. 在文献[2]中, Farrina 等使用 VLSI 技术的高度专用的 Systolic 阵列来解决这一问题. 软件方法是采用基于通用并行计算机选择一种语言来编制算法的软件包, 也可以采用某种可编程 DSP 芯片. 硬件的实现方法是针对特定算法设计处理器, 因此, 它的处理速度快、吞吐量大, 但它的设计难度大, 设计周期长. 软件实现方法在实时性能上肯定不如硬件实现方法, 但它也有自己的优势, 例如: 可移植性好, 对技术更新具有更强的稳健性等. 在通用并行计算机上对 STAP 的研究^[3,4], 往往是通过实现这一密集计算的算法, 来研究大规模并行计算中更普适性的规律, 如性能

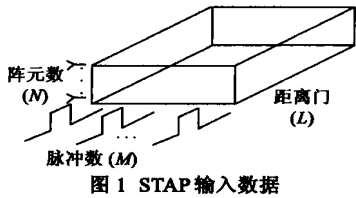
更优良的通信模型的提出. 另外, 通用超级计算机有过多的系统辅助操作, 这并不利于其应用到实时信号处理中. 文献[5]通过将相同的 STAP 算法在 IBM SP2 与双片 ADSP21062 的运行结果进行比较, 得出 DSP 芯片是雷达信号处理应用的最佳选择. 近年来, 各大 DSP 厂商相继推出了一系列更高性能的 DSP 芯片, 如 ADI 的 TigerSHARC 系列等. 这就为 STAP 这样的密集计算问题的实时实现提供了更强有力的硬件支持.

本文的目的是研究多 DSP 并行处理系统实时实现 STAP 算法. 本文的后续部分是这样安排的: 第 2 节研究了部分自适应算法内在并行性, 第 3 节简要介绍了一种多 DSP 并行处理系统, 第 4 节针对多 DSP 并行处理系统, 提出了一种 STAP 并行处理算法并给出算法映射; 第 5 节通过数据实验, 得出结论. 第 6 节总结全文. 下面我们展开论述.

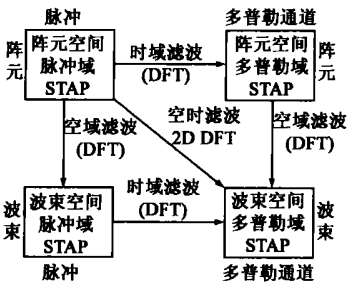
2 STAP 算法

STAP 是对一个输入数据立方体执行的一系列计算. 输入数据立方体是在一个被称为一个相参处理间隔 (CPI) 的时间

间隔内, 由 N 个阵元所接收到的 M 个脉冲的回波信号, 其中每一个脉冲的距离门数为 L . 如图 1 所示.



按照文献 [6] 的研究, STAP 可以分为四类, 如图 2 所示. 每一个方框代表了不同的变换后, 一个距离门的数据平面. 全自适应 STAP 算法 (阵元空间-脉冲域类算法) 是将分离的自适应权应用到所有脉冲和通道, 这要求求解 MN 阶线性方程组, 其中 N 是阵元数, M 是每一个 CPI 中的脉冲



数. 在现有计算能力条件下, 性能最优的全自适应 STAP 是很难做到实时实现的. 因此, 对于 STAP 在实际系统中的实现, 部分自适应 STAP 更具吸引力.

部分适应 STAP 是将全程自适应 STAP 分解成一系列计算量较小, 相互独立的, 更易处理的自适应问题, 并可获得次最优的性能. 部分自适应 STAP 的计算可分为三步.

第一步: 域变换, 即非自适应滤波. 该步骤用于降低自适应权值求解问题的维数, 它决定自适应权值计算所在的域. 非自适应滤波即可以是到频域的变换, 在每一个通道所有的脉冲执行 FFT 运算 (阵元空间-多普勒域类算法), 也可以是到波束空间的变换, 所有通道同一个脉冲的非自适应波束形成 (波束空间-脉冲域类算法). 空间和时间变换可以同时执行 (波束空间-多普勒域类算法).

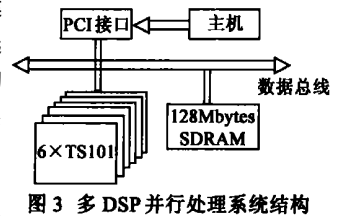
第二步: 自适应权值计算. 在自适应所在的域内, 自适应权值是通过求线性方程组的最小二乘解得出. 给定空时导向矢量 s , 自适应权 w 与干扰协方差矩阵的关系为 $Rw = s$. 协方差矩阵 R 由 $\bar{R} = X^H X$ 公式估计得出, 其中训练数据矩阵 X 是输入数据矩阵的子集. 出于数值稳定性和动态范围方面的考虑, 可以通过 QR 分解直接从 X 计算 R 的 Cholesky 分解 [1]. 计算 $A = QX$, 其中 A 是一个上三角阵, Q 是一个正交的酉矩阵, $Q^H Q = I$. 因此, $\bar{R} = X^H X$ 等价于 $\bar{R} = A^H A$. 由于 A 是三角阵, 由如下两步回代操作解出 $W: A^H y = s, Aw = y$.

第三步: 波束形成, 即将自适应权值应用到输入数据. 为一个矩阵矢量相乘的过程, $Z = Yw$, 其中 Y 是输入数据, Z 是该波束的输出数据.

由以上的分析可以看出, 部分自适应 STAP 的计算步骤都是对不同的数据集重复地执行同一运算. 域变换是针对每一个通道或每一个脉冲, 重复执行 FFT 运算, 而自适应是对不同的数据集求其最小二乘解, 计算是彼此独立的. 因此, STAP 在计算上具有并行性, 这是 STAP 并行化的基础. 利用并行处理技术对 STAP 进行处理, 不仅能够满足其实时需求, 同时增加了雷达信号处理系统的灵活性和可升级性.

3 多 DSP 并行处理系统

STAP 算法运算量大, 数据吞吐量高, 单片 DSP 已不能满足其需求. 因此, 只有多片 DSP 构成的并行处理系统才能满足其处理能力, 这样不仅使系统处理速度大大提高, 同时系统性能也得到明显改善, 如实时性、精度和动态范围上都得到很大提高. ADI 出品 TigerSHARC 系列在浮点计算能力较其他同类 DSP 具有的优势, 同时它具有的强大的实现多处理器并行处理的能力使其成为构建 DSP 并行处理系统的首选器件. 图 3 为一个簇式结构的多 DSP 并行处理系统, 系统由 6 片 TigerSHARC TS-101DSP 芯片和一个 128M 的共享存储器组成, 6 片中



一片是主控芯片. TS 101 为 32 位处理器, 主频 250MHZ, 流水深度 6 级, 有 6M 位片内存储器. 系统由 PCI 总线与基于 Windows NT 的主机相连, 执行代码由主机下载到主控 DSP 芯片并由它控制其它芯片工作, 数据由主机下载到 SDRAM 中或片内 RAM 中.

4 STAP 并行算法及算法映射

并行算法的设计是针对特定并行计算模型的 [7]. 针对第 3 节中的并行处理系统, 我们设计部分自适应 STAP 粗粒度、任务级的并行算法. 将该类方法的三个计算步骤中相互独立的计算划分为任务. 对于域变换, 计算任务是指 FFT 运算, 对于自适应计算部分, 一个任务是指对于特定数据矩阵和导向矢量的线性方程组的求解过程. 波束形成部分的一个任务是一个矩阵和矢量的乘法运算. 将这些任务分配给相应的处理器完成. 如图 4 所示, 每一个计算步骤完成后, 系统应当进行同步处理并对数据进行重映射. 这样就可以把 STAP 表示成一系列的与通信任务, 而且通信是与计算分离的. 采取这种任务级的并行算法, 是为了在计算过程中尽量减少处理器之间以及处理器与外部存储器之间的通信.

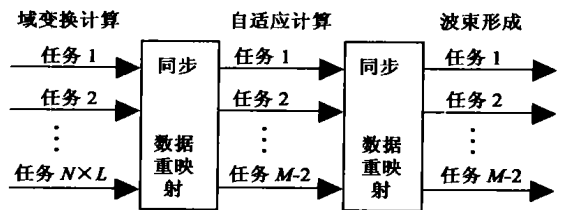


图 4 部分自适应 STAP 并行算法

延迟和吞吐率是评估实时系统重要的指标, 我们这样定义上述算法的延迟和吞吐率: 设系统在每个计算步骤所花费的时间分别为 $T_i (i = 1, 2, 3)$, 则延迟 T_L 和吞吐率 D_{TM} 分别为:

$$T_L = \sum_i T_i \quad (1)$$

$$D_{TM} = \frac{1}{\max_{1 \leq i \leq 3} T_i} \quad (2)$$

我们以阵元空间-多普勒域类算法为例来描述我们的并

行算法, 波束空间-脉冲域类算法和波束空间-多普勒域类算法的并行算法与此类似. 对于图 1 中 $M \times N \times L$ 维的数据立方体, 阵元空间-多普勒域类算法的计算过程如下.

域变换计算: 该步骤是在时间上, 将数据从脉冲域变换到多普勒域. 如图 5(a) 所示, 它是沿脉冲轴做 M 点 FFT, 每一个阵元和距离门都要执行, 所以共要计算 $N \times L$ 次 FFT, 把如图 5(a) 的例子转化为如图 5(b). 经过 FFT 后, 每一个 $N \times L$ 维数据称为一个多普勒通道, 共有 M 个多普勒通道. 接下来的计算都是对经过 FFT 变换的数据立方体.

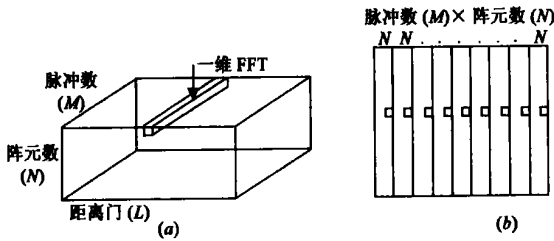


图 5 域变换计算部分

自适应权值的计算: 即求线性方程的最小二乘解. 这一步是将经过 FFT 的数据立方体分配给多个最小二乘问题. 首先, 将数据立方体按脉冲分为 M 个数据平面. 该算法是将 M 个数据平面划分到 $M-2$ 个最小二乘问题. 如图 5 所示, 对于第 i 个问题, 将含有 $N \times L$ 个多普勒通道的第 i 个数据平面, 与第 $i-1$ 、 $i-2$ 平面并联, 构成维数是 $3N \times L$ 的数据矩阵. 计算权值的距离门数 L_{is} 应当满足自适应权的收敛条件, 如图 6(a), 我们取 $L_{is} = 3 \times (3N)$, 这样求解权值的数据矩阵为 $3N \times 9N$ 维. 图 6(b) 给出了用于自适应权值求解的数据矩阵, 其中斜线为一个矩阵的对角线.

波束形成: 将 $3N$ 维权向量与 $3N \times L$ 维数据矩阵相乘, 得到一个 L 维的数据向量.

因此, 域变换部分有 $N \times L$ 个计算任务, 自适应计算部分有 $M-2$ 个计算任务, 波束形成部分 $M-2$ 个计算任务. 将这些任务映射到多 DSP 并行处理系统, 即将任务划分给各个 DSP 芯片. 为了便于描述, 在这里我们用实例说明系统的处理流程. 设 6 个阵元的天线, 在一个 CPI 内发射 32 个脉冲, 距离门数为 512, 分 I 、 Q 双通道存储. 因此, 系统的输入数据为 2 个 $6 \times 32 \times 512$ 维的数据立方体. 第一步, 要将这个数据立方体从主机下载到系统的共享存储器中. 第二步, 按图 5(a), 按阵元将数据立方体划为 6 个数据平面, 以 DMA 方式分别发送到 6 个 DSP 中. 在每一个 DSP 中做 512 次 32 点 FFT. 所有芯片都完成后, 再由 DMA 方式, 将所有数据发送回 SDRAM, 进行数据重映射. 第三步, 按照如图 6(b) 排列数据, 共有 30 个最

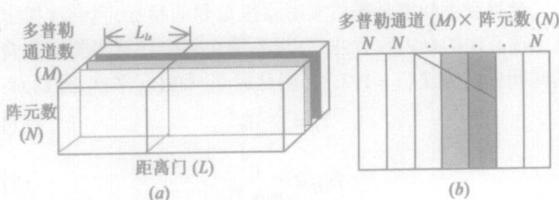


图 6 自适应计算部分

小二乘问题. 将这 30 个任务交给 6 个处理器, 每个处理器分配了 5 个任务. 每个处理器应计算得出 5 个 18 维的权值向量, 所有处理器计算完毕后, 系统进行同步操作. 第四步, 波束形成, 将权值向量与其计算出的数据矩阵相乘, 每个处理器执行 5 次, 得到 5 个 512 维数据向量. 再将这 5 个向量以 DMA 方式发回 SDRAM.

5 数据试验

我们用仿真数据对系统进行测试. 用 MATLAB 产生如下参数的仿真数据: 10 行 6 列矩形平面阵, 沿机身正侧面放置, 波束指向阵列法线方向, 即高低角 $\varphi_0 = 0^\circ$, 方位角 $\theta_0 = 90^\circ$, 载机高度 $H = 8\text{km}$, 工作波长 $\lambda = 0.23\text{m}$, 脉冲重复频率 $f_r = 2434\text{Hz}$, 阵元间距 $d = 0.115\text{m}$, 载机速度 $V = 140\text{m/s}$, 二维杂波谱斜率 $\alpha = \lambda_r/4v = 1$, 即呈对角分布. 发射天线高低向采用 -20dB 切比雪夫加权, 方位向采用 -30dB 切比雪夫静态加权, 接收天线高低向 -20dB , 水平方向是自适应的. 时域多普勒滤波器用 -60dB 切比雪夫加权. 单元杂噪比 60dB , 时域脉冲数 $K = 32$.

系统测试结果及按式(1)(2)计算得出的延迟和吞吐率如表 1. 由表中数据我们可以看出, 系统的吞吐率是较高的. 文献[5]中, 以 8 个通道, 64 个脉冲, 64 个距离门的数据, 在两片 ADSP21062 DSP 系统上实现 STAP 算法, 系统的执行时间达 0.0350s, 显然, 我们的系统延时要小的多, 这除了器件本身的差距外, 同时说明任务级的并行算法在降低通信量上有很大贡献.

表 1 系统实测结果及性能参数

计算步骤		时间(单位: ms)	性能	
域变换计算	FFT	1.502	延迟	5.713ms
	通信	1.709		
自适应权值计算	QR 分解	0.985		
	回代	0.42	吞吐率	311.4
通信	0.961			
权值应用	内积计算	0.136		

6 结论

本文通过分析部分自适应 STAP 的内在并行性, 针对多 DSP 并行处理系统提出了一种任务级的并行算法, 并将该算法映射到一个多 DSP 系统上. 通过仿真数据对系统进行测试, 试验结果显示该系统实时性能较高. 由此我们可以得出该并行算法是可行的, 如果以多片 DSP 为核心构造一个嵌入式系统作为 STAP 专用处理机, 将是一个非常具有吸引力的方案.

参考文献:

- [1] 王永良, 彭应宁. 空时自适应信号处理[M]. 北京: 清华大学出版社, 2001. 46-55.
Wang Yongliang, Peng Yingning. Space Time Adaptive Processing[M]. Beijing: Tsinghua University Press. 2001: 46-55. (in Chinese)
- [2] A Farina, L Timmoneri. Real time STAP techniques[J]. Electronics & Communication Engineering Journal, 1999, 11(1): 13

- 22.

- [3] K Hwang, Z Xu, M Arakawa. Benchmark evaluation of the IBM SP2 for parallel signal processing[J]. IEEE Trans Parallel and Distributed Systems, 1996, 7(5): 522- 535.
- [4] J M Lebak, A W Bojanczyk. Design and performance evaluation of a portable parallel library for space time adaptive processing[J]. IEEE Trans Parallel and Distributed Systems, 2000, 11(3): 287- 298.
- [5] K Rajan, L M Patnik. Implementation of SFAP algorithms on IBM SP2 and on ADSP 21062 dual digital signal processor systems[J]. Microprocessors and Microsystems, 2003, 27(4): 221- 227.
- [6] 王永良. 新一代机载预警雷达的空时二维自适应信号处理[D]. 西安: 西安电子科技大学, 1994.
Wang Yongliang. Space Time Adaptive Processing for the New Generation of AEW Radars[D]. Xi'an: Xidian University, 1994. (in Chinese)
- [7] 陈国良. 并行算法的设计与分析[M]. 北京: 高等教育出版社, 2002. 2- 47.
Cheng Guoliang. Design and Analysis of Parallel Computing Algorithms[M]. Beijing: High Education Press. 2002. 2- 47. (in Chinese)

作者简介:



范西昆 男, 1976 年生于陕西西安, 国防科技大学电子科学与工程学院博士研究生, 主要从事 STAP、实时信号处理等方面的研究。
E mail: vanxikun@sina.com.



王永良 男, 教授, 博士生导师, 1965 年生于浙江嘉兴, 1994 年获西安电子科技大学博士学位, 1996 年 12 月清华大学博士后出站, 现为空军雷达学院兵器运用工程军队重点实验室主任, 中国电子学会无线电分会委员, 已发表论文 150 多篇, 收入三大检索 70 余篇, 出版学术专著 2 部, 曾获高校青年教师奖和中国优秀博士后奖, 入选新世纪百千万人才工程国家级人选, 主要研究领域为: 侦察、监视与预警技术、空时二维处理、阵列信号处理等。

陈辉 男, 硕士, 讲师, 1974 年生于江苏启东, 分别于 1996 年 7 月和 1999 年 3 月获空军雷达学院信息处理专业学士和硕士学位, 现已发表论文 30 多篇, 其中收入三大检索 10 多篇, 主要研究方向: 阵列信号处理、超分辨空间谱估计和 STAP。