

机载雷达空时自适应处理的实时实现

范西昆^{①②} 王永良^② 陈辉^②

^①(国防科学技术大学电子科学与工程学院 长沙 410073)

^②(空军雷达学院兵器运用工程军队重点实验室 武汉 430010)

摘要 多通道机载雷达空时自适应 (STAP) 算法运算量巨大, 不易实时实现。本文详细分析了部分自适应 STAP 的 3 个计算步骤, 指出了其内在并行性。针对多片 DSP 系统提出了一种部分自适应 STAP 并行算法, 给出了该算法的任务划分、执行模型、任务映射策略以及性能评价函数。该算法在不同的计算阶段之间进行数据重映射。通过数据试验证明这种基于多片 DSP 系统的 STAP 并行算法具有较高的实时性能。

关键词 机载雷达, 空时自适应处理, 并行处理, 数字信号处理

中图分类号: TN959.73

文献标识码: A

文章编号: 1009-5896(2006)12-2224-04

Real-Time Implementation of Airborne Radar Space-Time Adaptive Processing

Fan Xi-kun^{①②} Wang Yong-liang^② Chen Hui^②

^①(School of Electronic Science and Engineering, NUDT, Changsha 410073, China)

^②(Key Lab of Wuhan Radar Institute, Wuhan 430010, China)

Abstract The Space-Time Adaptive Processing (STAP) algorithm of multi-channel airborne radar requires very high computing power and hardly implements. In this paper, detail analysis of computation steps of partially adaptive STAP algorithm is presents, which indicates that there is a natural, inherent parallelism in STAP algorithm. A parallel algorithm based on multi-DSP system of partially adaptive STAP is presented. The execution model, task mapping strategy and performance evaluation functions of this algorithm are also illustrated. Data remapping is used between successive computation steps of STAP. The effectiveness of the implementation is demonstrated with experimental results.

key words Airborne radar, STAP, Parallel processing, DSP

1 引言

空时自适应处理(STAP)可以明显改善机载雷达对低空、低速目标的探测性能。因此, 研究者将其视为新一代机载相控阵雷达信号处理的核心技术^[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。文献[2,3]利用高性能计算机 (HPC) 实现STAP算法, 而文献[4]通过将相同的STAP算法在通用高性能计算机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 Ts101、201 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 节简要论述了 STAP 算法; 第 3 节研究了部分自适应算法内在并行性, 并给出了其任务模型; 第 4 节简要介绍了一种多 DSP 并行处理系统; 第 5 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该算法的任务划分、执行模型、任务映射策略以及性能评价函数; 第 6 节通过数据实验, 得出结论。

第 7 节总结全文。

2 STAP 算法

设 N 个通道, 接收 M 个脉冲, 则第 k 个距离单元的空时快拍数据为 $MN \times 1$ 维矢量, 形如(1)式

$$\mathbf{X}_k = [\mathbf{x}_{k,1,1}, \dots, \mathbf{x}_{k,N,1}, \mathbf{x}_{k,1,2}, \dots, \mathbf{x}_{k,N,2}, \dots, \mathbf{x}_{k,1,M}, \dots, \mathbf{x}_{k,N,M}]^T \quad (1)$$

其中 3 个下标分别表示距离、通道和脉冲。每一个空时快拍由加性的杂波, 有源干扰, 热噪声和可能出现的目标组成。直观来看, STAP 算法的输入数据为立方体结构。

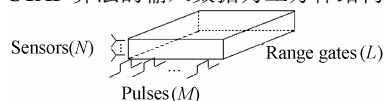


图 1 STAP 输入数据

Fig.1 Input data cube of STAP

空时导向矢量定义为空域导向矢量和时域导向矢量式的 Kronecker 积, $\mathbf{S} = \mathbf{S}_s \otimes \mathbf{S}_t$ 。空域导向矢量和时域导向矢量为式(2)和式(3)。其中, f_s 和 f_t 分别为归一化空频和时频。

$$\mathbf{S}_s = [1 \ e^{j2\pi f_s} \ e^{j2^2 2\pi f_s} \ e^{j3^2 2\pi f_s} \ \dots \ e^{j(N-1)^2 2\pi f_s}]^T \quad (2)$$

$$\mathbf{S}_t = [1 \ e^{j2\pi f_t} \ e^{j2^2 2\pi f_t} \ e^{j3^2 2\pi f_t} \ \dots \ e^{j(N-1)^2 2\pi f_t}]^T \quad (3)$$

最小方差失真响应 (MVDR) 意义下的最优 STAP 自适应权值和滤波器输出形如式(4), 其中 $\hat{\mathbf{R}}$ 为估计的干扰协方

差矩阵, $\hat{\mathbf{R}} = \frac{1}{K} \sum_i^K \mathbf{X}_i^H \mathbf{X}_i$, K 应当满足 2~3 倍的系统自由度。

$$\mathbf{W}_{\text{opt}} = \frac{\hat{\mathbf{R}}^{-1} \mathbf{S}}{\mathbf{S}^H \hat{\mathbf{R}}^{-1} \mathbf{S}}, \quad \mathbf{y} = \mathbf{W}^H \mathbf{X}_k \quad (4)$$

对于脉冲数和通道数较多的情形, 实现最优的空时自适应处理是非常困难的, 实际工程中一般采用部分自适应的方法。该类方法由 3 个计算步骤组成。第 1 步, 域变换, 或非自适应滤波。通过构造变换矩阵 $\mathbf{T}$, 将空时数据 $\mathbf{X}$ 和导向矢量 $\mathbf{S}$ 与 $\mathbf{T}$ 进行内积运算, $\hat{\mathbf{X}} = \mathbf{T}_L^H \mathbf{X}_k$, $\hat{\mathbf{S}} = \mathbf{T}_L^H \mathbf{S}$ 达到降维的目的。不同的部分自适应 STAP 算法的变换矩阵不同^[1]。这一步对于工程应用来讲就是一维或二维 FFT 运算。第 2 步, 自适应权值求解。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。自适应权 $\mathbf{W}$ 与干扰协方差矩阵的关系为, 出于数值稳定性和动态范围方面的考虑, 一般采用 QRD-SMI 方法。通过 QR 分解直接从 $\mathbf{X}$ 计算 $\mathbf{R}$ 的 Cholesky 分解^[1]。计算 $\mathbf{A} = \mathbf{Q}\mathbf{X}$, 其中 $\mathbf{A}$ 是一个上三角阵, $\mathbf{Q}$ 是一个正交的酉矩阵, $\mathbf{Q}^H \mathbf{Q} = \mathbf{I}$ 。因此, $\hat{\mathbf{R}} = \mathbf{A}^H \mathbf{A}$ 。由于 $\mathbf{A}$ 是三角阵, 由如下两步回代操作解出 $\mathbf{W}$: $\mathbf{A}^H \mathbf{z} = \hat{\mathbf{S}}$, $\mathbf{A} \mathbf{W} = \mathbf{z}$ 。第 3 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, $\mathbf{y} = \mathbf{W}^H \hat{\mathbf{X}}$ 。

3 任务模型

由以上的算法分析我们可以看出, 部分自适应 STAP 实质上它是将最优的 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, 它是由 3 个计算阶段组成如图 2, 而每一个计算阶段又是由若干子任务组成。

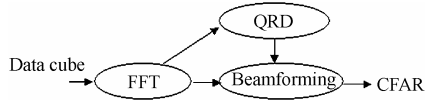


图 2 部分自适应 STAP 算法的任务模型

Fig.2 Task model of partially adaptive STAP algorithm

我们以阵元空间-多普勒域类算法为例分析图 2 所示的任务模型。对于图 1 中 $M \times N \times L$ 维的数据立方体, 计算过程如下。

(1) 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 3(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 3(a) 的例子转化为如图 3(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。

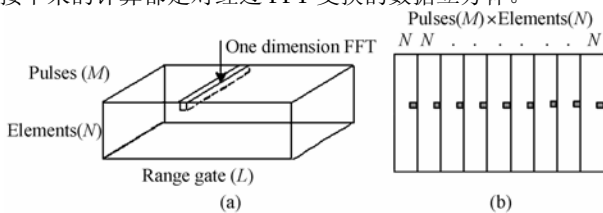


图 3 域变换计算部分

Fig.3 Computational part of domain transforming

(2) 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 4 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_{ls} 应当满足自适应权的收敛条件, 如图 4(a), 我们取 $L_{\text{ls}} = 3 \times (3N)$, 这样求解权值的数据矩阵为 $(3N) \times (9N)$ 维。图 4(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。

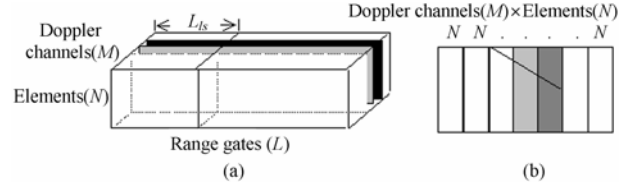


图 4 自适应计算部分

Fig.4 Adaptive weight computing

(3) 波束形成: 该阶段是将计算得到的权值应用到多普勒滤波后的数据。从计算的角度来看是一个 $3N$ 维权矢量与 $3N \times L$ 维数据矩阵相乘的过程, 得到一个 L 维的数据矢量。

4 多 DSP 并行处理系统

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, 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。

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。图 5 为一个簇式结构的多 DSP 并行处理系统, 系统由 6 片 TigerSHARC TS-101 DSP 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。

5 STAP 并行算法的设计与实现

并行算法的设计是针对特定并行计算模型的^[5]。针对第 4 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。这样考虑的原因在于: (a) 通信是主要的系统开销。因此, 在并行算法设计时应当尽量使数据局部化, 避免计算过程中的远程数据访问。(b) 因为我们系统的资源有限, 并行 FFT 以及并行 QR 分解这样的细粒度并行算法是不适合我们的系统的。

5.1 执行模型

嵌入式处理系统,其尺寸、重量以及功率都有严格的限制,处理器数目有限。因此在计算过程中,并行算法设计时使处理器的利用率达到最优。针对多 DSP 系统,设计了如图 6 线性执行模型。由图 6 中我们可以看出对于每一个计算阶段, n 每一个处理器分配了几乎相同数目任务。

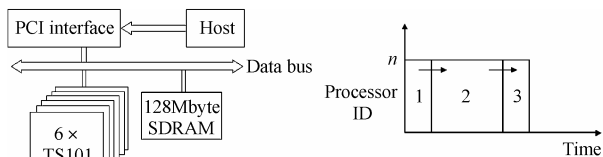


图 5 多 DSP 并行处理系统结构

Fig.5 Structure of multi-DSP system

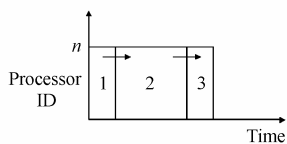


图 6 STAP 并行算法的执行模型

Fig.6 Execution model of STAP parallel algorithm

5.2 任务映射策略

在第 3 节中我们详细分析了部分自适应 STAP 的 3 个计算步骤,并将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,共计 $N \times L$ 个;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,有 $M-2$ 个;波束形成部分的一个任务是一个矩阵和矢量的乘法运算,也是 $M-2$ 个。依据执行模型,将这些任务平均地分配给各个处理器完成,避免出现处理器空闲。

5.3 数据重映射

STAP 不同的计算步骤是针对不同数据集的,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。这样就可以把计算与通信任务分离,避免了计算时处理器之间的数据交互。

图 7 为系统处理流程。为了便于描述,在这里我们用实例进一步说明。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 1 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 2 步,按图 3(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 3 步,按照如图 6(b)排列数据,共有 30 个最小二乘问题。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 维的权值矢量,所有处理器计算完毕

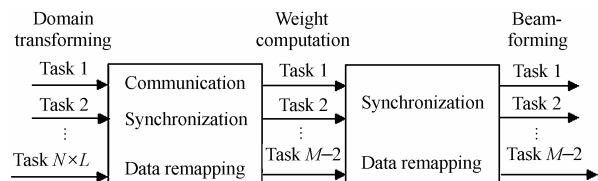


图 7 部分自适应 STAP 并行算法

Fig.7 Parallel algorithm of partially adaptive STAP

后,系统进行同步操作。第 4 步,波束形成,将权值矢量与

将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 维数据矢量。再将这 5 个矢量以 DMA 方式发回 SDRAM,以便后续处理。

5.4 性能评估指标

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_{TH} 分别为

$$T_L = \sum_i T_i, \quad D_{TH} = \frac{1}{\max_{1 \leq i \leq 3} T_i}$$

6 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 $\theta_0 = 0^\circ$,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 $f_r=2434.8\text{Hz}$,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 $\alpha=\lambda f_r/4v=1$,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),式(2)计算得出的延迟和吞吐率如表 1。由表 1 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得多,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

Tab.1 System test results and performance parameters

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	吞吐率	311.4/s
	回代	0.42		
	通信	0.961		
权值应用	内积计算	0.136		

7 结束语

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. 北京: 清华大学出版社, 2001. 46-55.
- [2] Hwang K, Xu Z, Arakawa M. Benchmark evaluation of the IBM SP2 for parallel signal processing. *IEEE Trans. on Parallel and Distributed Systems*, 1996, 7(5): 522-535.
- [3] Lebak J M, Bojanczyk A W. Design and performance evaluation of a portable parallel library for Space-Time Adaptive Processing. *IEEE Trans. on Parallel and Distributed Systems*, 2000, 11(3): 287-298.
- [4] Rajan K, Patnik L M. Implementation of STAP algorithms on IBM SP2 and on ADSP 21062 dual digital signal processor systems. *Microprocessors and Microsystems*, 2003, 5(4): 221-227.
- [5] 陈国良. 并行算法的设计与分析. 北京: 高等教育出版社, 2002: 2-47.
- 范西昆: 男, 1976 年生, 博士生, 从事 STAP、实时信号处理等方面的研究.
- 王永良: 男, 1965 年生, 教授, 博士生导师, 主要研究领域为侦查、监视与预警技术、空时二维处理、阵列信号处理等, 已发表论文 130 多篇, 收入三大检索 60 余篇, 出版专著 2 部.
- 陈 辉: 男, 1974 年生, 讲师, 主要研究方向为阵列信号处理、超分辨空间谱估计, 已发表论文 30 多篇, 其中收入三大检索 10 多篇.