

doi: 10.3969/j.issn.1000-8349.2025.02.07

多相滤波器组技术在低频全天总功率 测量实验中的应用

王兴涛¹, 刘雄伟¹, 王伟华², 吴亚军³, 余赞³, 杨小龙³

(1. 西华师范大学 物理与天文学院, 南充 637002; 2. 常州工学院 计算机科学与信息工程学院, 常州 213032; 3. 中国科学院 上海天文台, 上海 200030)

摘要: 作为探测宇宙再电离时期的重要方法之一, 低频全天总功率实验的核心目标是在频谱为 50~200 MHz 的全天总功率谱中寻找宇宙再电离时期极其微弱的中性氢 21 cm 信号。目前通常使用快速傅里叶变换 (fast Fourier transform, FFT) 来计算功率谱, 其存在的频谱泄漏问题, 使得射频干扰 (radio frequency interference, RFI) 无法彻底剔除, 从而严重干扰了微弱的宇宙再电离信号的提取。多相滤波器组 (polyphase filter bank, PFB) 是一种将信号按频率均匀划分的技术, 可以替代 FFT 计算功率谱。介绍了 PFB 的基本原理及基于统一计算架构 (compute unified device architecture, CUDA) 的实现, 并且通过仿真比较了 PFB 和 FFT 的性能, 还比较了不同阶数的 PFB 性能差异。结果表明, PFB 技术具有更平坦的带通响应、更窄的过渡带、更好的带外抑制。最后, 对天马望远镜的观测数据及低频总功率实验产生的数据进行处理, 验证了 PFB 技术的优越性, 为全天总功率实验的频谱分析提供了一种有效的解决方法。

关键词: CUDA; PFB 技术; GPU 并行计算; 信号处理; 多相滤波器; FFT 运算

中图分类号: P161 **文献标识码:** B

1 引言

随着宇宙膨胀降温, 早期宇宙进入“黑暗时期”, 无发光天体, 仅中性气体弥漫。高密度暗物质聚集形成暗晕, 吸引气体形成第一代发光天体。这些天体释放的高能光子有一部分逃离并电离星系际中性氢, 在宇宙演化到红移 $z = 6$ 时, 星系际气体基本完全电离, 此过程被称为“宇宙再电离”^[1]。中性氢原子会产生 21 cm 发射线, 是研究宇宙黎明时期的重要探针。然而, 来自宇宙黎明和再电离时期的中性氢发射线非常微弱, 与以银河系同步辐射为主导的前景相比, 信号低 5 个量级。因此, 对这些微弱信号的研究和检测极具挑战性。此外,

收稿日期: 2024-07-16; 修回日期: 2024-08-21

资助项目: 科技部 SKA 专项 (2020SKA0110103); 国家自然科学基金重点项目 (12433012)

通讯作者: 吴亚军, wuyajun@shao.ac.cn; 王兴涛, 1368981402@qq.com

在 50~200 MHz 频段内, 人类活动产生的射频干扰 (radio frequency interference, RFI) 极为复杂。因此, 消除 RFI 等干扰及提取中性氢 21 cm 发射线, 是未来研究宇宙早期历史和结构演化必须要克服的问题。

傅里叶变换 (Fourier transform, FT)^[2] 算法是射电天文信号处理的重要方法之一。对于有限时间长度的观测来说, 实际使用离散傅里叶变换 (discrete Fourier transform, DFT) 来实现快速傅里叶变换算法, 以高效地计算信号的频谱。然而 DFT 算法有两个较明显的缺点: (1) 频率泄漏 (leakage), 即单一频率输入频率信号, 特别是输入信号较强时, 在频谱上会有多个频率点的响应; (2) 栅栏损失 (scalloping loss), 即由于 DFT 算法获得的频谱中单个频率分组内的频率响应并不平坦, 导致不同频率分组之间存在能量损失^[3]。为克服 DFT 算法这两个缺点, 获得更高精度的频谱结果, 多相滤波器组 (polyphase filter bank, PFB)^[4] 技术被提出。多相滤波器组是一种可以有效克服 DFT 算法缺点的另一个频谱计算方法, 它可以将信号分解成多个组成部分, 并根据每个部分的特点进行有针对性的处理。这种技术在处理对频谱质量要求较高的天文数据时, 能提高数据处理效果, 在频谱质量上有显著优势。

PFB 架构的历史可以追溯到 1973 年, 当时由 Schafer 和 Rabiner^[5] 在设计语音分析-合成系统时首次引入。随后, 在 1976 年, Bellanger 等人^[6] 对 PFB 架构进行了详尽的阐述, 使其概念更加清晰。然而, 尽管 PFB 架构的理论基础早已确立, 但它在射电天文学光谱测量中的应用却相对较晚。直到 1991 年, 在美国宇航局的任务中, Zimmerman 和 Gulakis^[7] 验证了多相 DFT 系统的旁瓣抑制性能远远优于窗口 DFT 系统, PFB 架构才被正式引入射电天文学领域, 为这一领域的研究提供了新的视角和工具。进入 21 世纪, Buntun^[8] 在 2000 年进一步推动了 PFB 在射电天文学中的普及, 他建议将 PFB 用于射电干涉仪相关器系统, 这一建议得到了广泛的认可和采纳。随着 PFB 系统的日益普及, 其在射电天文学领域的应用也越来越广泛, 成为研究宇宙的重要工具之一, 并在国内也得到许多研究人员的重视。2010 年, 陈林杰等人^[9] 提出了基于多相滤波器的设计方案, 通过多相滤波器实现信号的滤波, 对滤波后的信号再进行检波, 然后通过积分控制, 最后得到信号的频谱强度; 2011 年, 朱凯等人^[10] 利用多相滤波器组进行频谱计算; 2015 年, 吴俊霖等人^[11] 将 400 MHz 带宽分为 16 个带宽为 25 MHz 的通道, 对其中任一通道做通道数为 6 000, 频率分辨率为 4 kHz 的频谱细化系统, 可用于 FAST (Five-hundred-meter Aperture Spherical Radio Telescope) 河内中性氢观测; 2015 年, 赵欣等人^[12] 设计实现了基于 Hamming 窗的多相滤波多通道数字终端, 可以应用于 FAST 河外中性氢巡天; 2016 年, 张夏等人^[13] 研究信号失真的相关情况, 并最终还原出真实信号; 2020 年, 刘晔等人^[14] 开发设计了实时带宽 10 MHz~2 GHz 的数字频谱仪, 完成了快速扫描和脉冲监测两种不同观测模式开发设计与性能测试; 2023 年, 张海龙等人^[15] 详细分析了 PFB 的频率特性, 对比数字滤波器组 (mathematical filter bank, MFB) 幅频响应, 发现 PFB 可产生更平坦的带通响应, 更窄以及更陡峭的过渡带, 从而有效缓解了频谱泄漏问题。

在本工作中, 我们使用的频率范围为 50~200 MHz, 用于低频全天总功率实验中接收和检测宇宙再电离信号的数字接收机, 设计和测试基于图形处理器 (graphics processing unit,

GPU) 的 PFB 频谱生成与处理原型程序。利用这一程序, 我们分别对仿真模拟数据和实测数据进行频谱分析, 并与传统 FFT 的结果进行对比。结果显示, 我们的 PFB 频谱生成和处理程序可以有效实现对频率泄漏的抑制, 这对于低频全天总功率实验中射电干扰信号的去除, 及后续宇宙再电离这样微弱信号的提取有重要的帮助。本文第 2 章介绍多相滤波器组技术原理和结构设计; 第 3 章介绍基于统一计算架构 (compute unified device architecture, CUDA) 的多相滤波器组技术的算法, 通过仿真数据处理, 对 PFB 的性能进行分析, 对比多相滤波器组技术与传统 FFT 在频谱响应上的优缺点; 第 4 章使用 PFB 技术对实测和实验数据进行频谱分析, 并验证 PFB 技术在低频全天总功率实验中的应用前景; 第 5 章分析该技术在数据处理上的影响, 并进行了讨论和总结, 同时也对本工作需要改进和提高的方面进行讨论。

2 多相滤波器组原理

多相滤波器组技术将一个输入信号分成多个子信号进行并行处理, 数字信号通过滤波器, 公式表达如下:

$$y(n) = \sum_{m=-\infty}^{\infty} h(m)x(n-m) \quad , \quad (1)$$

其中, $y(n)$ 为输出信号, $h(m)$ 为数字滤波器的系数, $x(n)$ 为输入信号。对于多相滤波器组, 其等效于将原型滤波器进行频谱搬移, 得到输入信号中相应的频谱输出。单个信道的输出是:

$$\begin{aligned} X_k(n) &= \sum_{m=-\infty}^{\infty} x(nD-m)h(m)e^{-jw_k m} = \\ &= \sum_{m=-\infty}^{\infty} x(nD-m)h(m)W_K^{-km} \quad , \end{aligned} \quad (2)$$

其中, $k = 0, 1, 2, \dots, K-1$, $w_k = 2k/K$, $W_K = e^{j2/K}$, K 是通道数, D 是抽取因子, 第 k 个通道拥有将中心频率移到 w_k 的带通滤波功能^[16]。

$$\begin{cases} x_i(m) = x(mK-i) \\ p_i(m) = h(mK+i) \end{cases} \quad , \quad (3)$$

其中, $i = 0, 1, 2, \dots, K-1$, $x_i(m)$ 是 $x(m)$ 的多相滤波。如果 $D = K$, 则式 (2) 表示为:

$$\begin{aligned}
 X_k(n) &= \sum_{m=-\infty}^{\infty} x(nD - m)h(m)e^{-j2\pi \frac{k}{K} m} = \\
 &= \sum_{i=0}^{K-1} \sum_{r=-\infty}^{\infty} x(rK - i)h(n - rK + i)e^{j2\pi \frac{k}{K} (rK - i)} = \\
 &= \sum_{i=0}^{K-1} \sum_{r=-\infty}^{\infty} x_i(r)p_i(n - r)e^{-j2\pi \frac{k}{K} i} = \\
 &= \sum_{i=0}^{K-1} [x_i(n)p_i(n)]e^{-j2\pi \frac{k}{K} i} = DFT[x_i(n)p_i(n)] \quad .
 \end{aligned}
 \tag{4}$$

由上述推导可以得出, 多相滤波器组可以使用 FFT 快速实现。

多相滤波器组的基本原理如图 1 所示, 实现了滤波器阶数 $n_{\text{tap}} = m$ 和信道数 $n_{\text{chan}} = N$ 的多相滤波器组。左边依次将输入信号按时间分成 N 组, 每组依次通过子滤波器, 其 N 组输出 $y(n)$ 依次做 N 点 FFT, 最后的输出 $Y(n)$ 是按频率分解的信号。

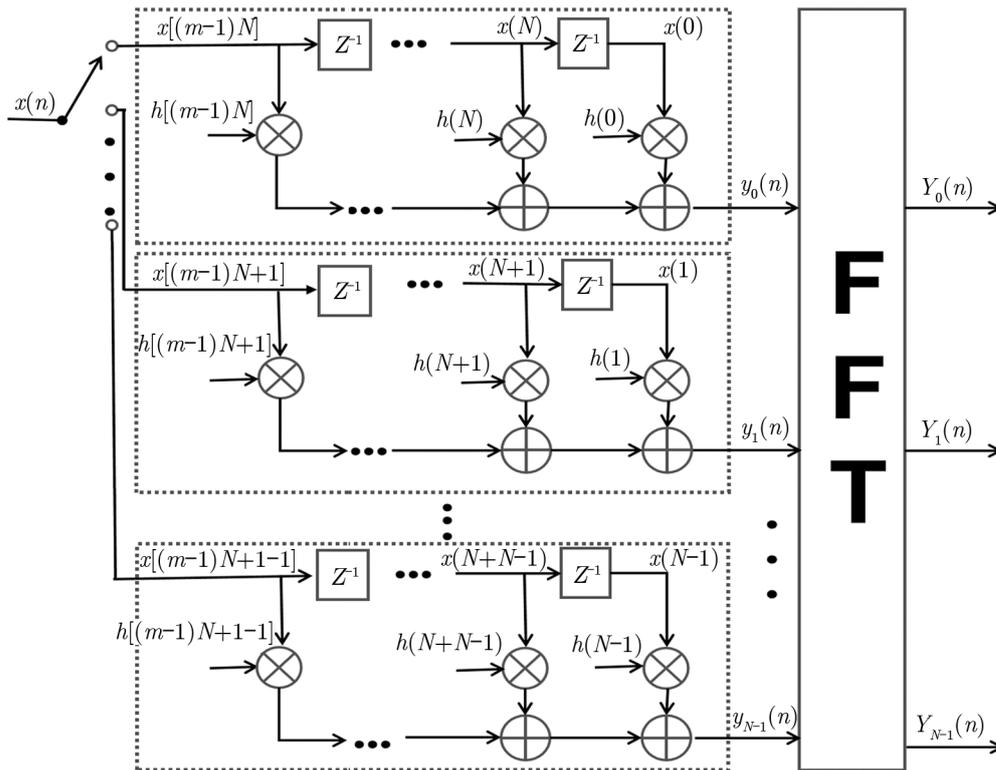


图 1 多相滤波器结构

3 CUDA 编程实现与性能分析

多相滤波器组提升信号处理精度的同时，也大幅增加了计算复杂度和运算量。针对这一问题，我们采用 GPU 卡进行加速处理，使用 CUDA 编译了相关软件。

基于并行架构的多相滤波器组的算法流程如图 2 所示。数据通过 GPU 调用，将数字信号进行数据分解并行分配给多个 GPU 线程，输入数据为一维数组，程序中根据信道数 n_{chan} 来对输入数据切分，将其组织成二维数组，GPU 中将数据长度为 n 的数组分解，若 $n_{chan} = N$ ，前 N 个数据作为第一列， $N+1$ 至 $2N$ 为第二列，以此类推，其中又以第 1、 $N+1$ 、 $2N+1$... 为第一组数据。同理得到其他信道数据分配情况。调用多个核心对每组数据进行 FIR 和 FFT，最后将得到的结果返回到中央处理器 (central processing unit, CPU)。其中在 CUDA 中使用 cuFFT 库中 cufftExecR2C() 函数，可以实现高效并行 FFT 计算。

在该软件中，我们设计了一个原型滤波器的频率响应 (如图 3 所示)，该滤波器具有适中的主瓣宽度和滚降特性，旁瓣水平较低，显示出良好的频域特性。

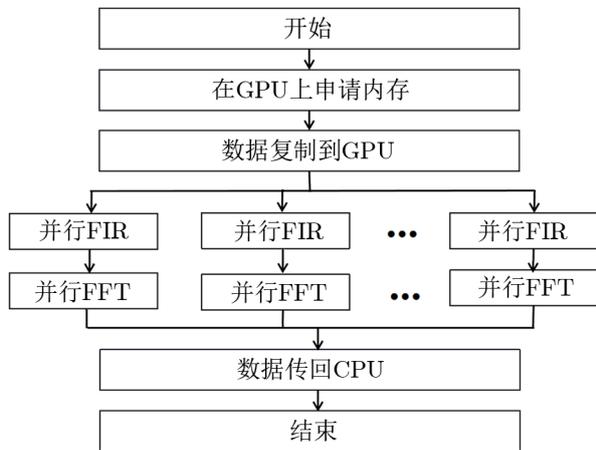


图 2 多相滤波器流程

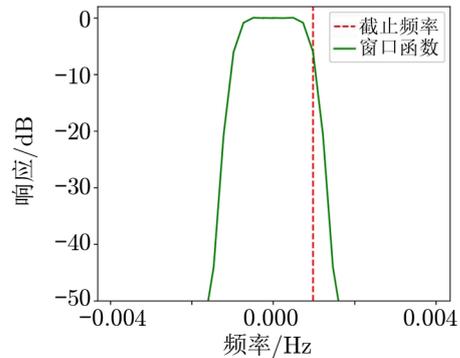


图 3 PFB 所使用的原型低通滤波器的频谱

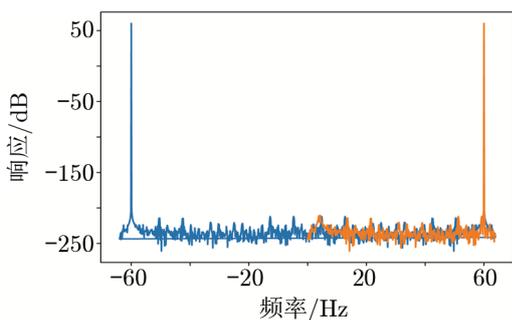


图 4 60 Hz 正弦信号的 FFT 与基于 CUDA 的 PFB 技术频谱

为了验证该软件功能的有效性，我们对一个采样率为 128 S/s，频率为 60 Hz 的仿真信号分别通过该软件与 FFT 进行频谱分析。结果如图 4 所示，在频率轴上 60 Hz 处看到了该信号，即软件成功地实现了频谱分析功能。图中 PFB 技术谱分析的频率响应左右不对称的主要原因是，在 CUDA 的 cuFFT 包中使用的傅里叶变换类型为实数至复数的变换类型，其输出的有效位数为输入的 1/2。

在该软件成功运行的情况下，我们进行了以下实验：使用数字信号长度 $n = 32\ 768$ ，滤

波器阶数 $n_{\text{tap}} = 8$, 信道数 $n_{\text{chan}} = 1024$, 以 0.05 Hz 步进, 在频率范围 $59.85 \sim 60.15 \text{ Hz}$ 内共生成 61 组仿真数据, 分别进行频谱分析。我们描绘出了 PFB 在一个信道内的频率响应, 并与 FFT 进行了比较。

图 5 是单个通道响应数值的图像, 在该仿真中信道带宽为 0.125 Hz 。如图所示, FFT 的 1 dB 带宽为 0.08 Hz , PFB 的 1 dB 带宽为 0.1 Hz ; FFT 的 20 dB 带宽为 0.22 Hz , PFB 的 20 dB 带宽为 0.15 Hz 。由此可知 PFB 相较于 FFT 拥有更窄的过渡带。

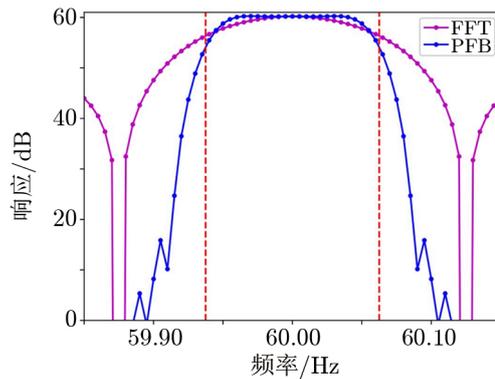


图 5 FFT 与基于 CUDA 实现 PFB 在 60 Hz 邻近频率的单一信道 dB 对比

同时, 改变 PFB 中的滤波器阶数 (通过调整滤波器阶数 n_{tap} 为 4、8、12、16) 后, 对仿真数据进行频谱分析, 比较其影响, 结果如图 6 所示, 红色虚线表示截止频率, 随着滤波器阶数的增加, 有效带宽越宽、过渡带越窄、带外抑制越高。

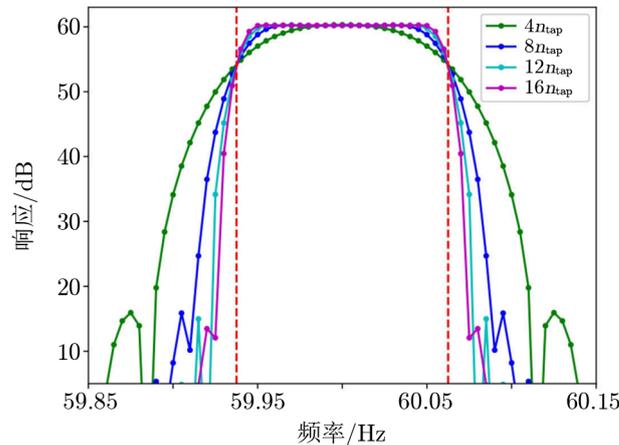


图 6 PFB 技术在不同滤波器阶数时在 60 Hz 邻近频率的单一信道频幅响应对比

4 实验与观测数据处理

我们用该软件处理了中国科学院上海天文台的天马 65 m 射电望远镜的观测数据。我们选取 S 波段观测数据, 该通道中心频率为 2 256 MHz, 将其与 FFT 频谱进行对比, 结果如图 7 所示。我们截取了 11~14 MHz 这一相对平坦的频率部分, 其中 11~14 MHz 对应的真实频率为 $2\,256 - 32 + (11 \sim 14)$ MHz, 并计算了其均方根 (root mean square, RMS) 值。经过精确测量, FFT 技术得到的 RMS 值为 163.23, 而 PFB 技术的 RMS 值为 163.08。图 8 是图 7 的 24 MHz 附近处放大图。在 24.60~24.75 MHz 范围内, PFB 的波峰比 FFT 更尖锐, 波谷更低, 展现出更优秀的频谱分析能力。

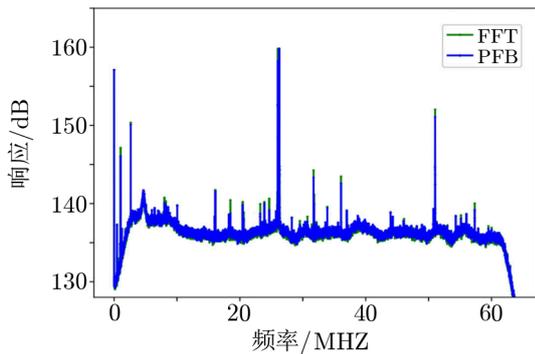


图 7 PFB 与 FFT 对射电信号处理的频谱

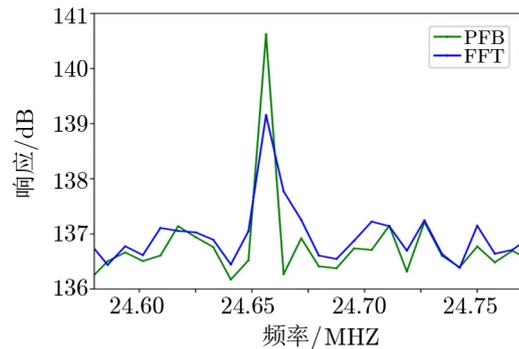


图 8 PFB 与 FFT 对射电信号处理的频谱图局部放大

另外, 我们还对低频总功率实验数字接收机产生的数据进行了处理, 采样率为 480 MHz。如图 9 所示, 在 90~100 MHz 的频段内, PFB 的 RMS 值为 31.84, 而 FFT 的 RMS 值为 21.48。特别值得注意的是, 在 10 MHz 的主峰处, PFB 的响应比 FFT 更尖锐, 并且两侧的陡峭程度更大, 抑制更明显。相较于 FFT 技术, 在低频全天总功率实验中, PFB 技术可以更容易地去除类似点频谱的射电干扰信号, 有助于后续微弱 21 cm 信号的提取。

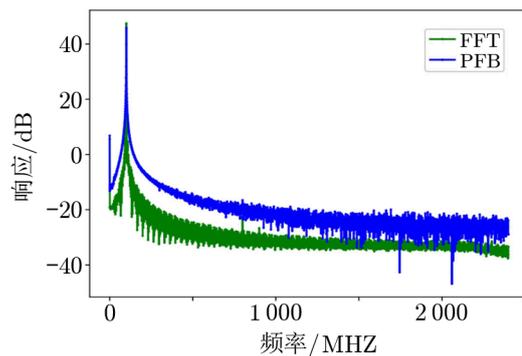


图 9 PFB 与 FFT 对 21CMA 射电信号处理的频谱

5 总结与展望

本文基于全天总功率实验, 用 CUDA 开发了 PFB 技术的相关软件, 并且通过仿真分析体现了该软件的性能: 过渡带狭窄、带通响应平坦、带外抑制高。然后处理了实测和实验数据, 验证了该方法在全天总功率谱实验中的应用前景。

目前该软件仅用于有限信号的数据仿真和部分真实的射电信号, 下一步将处理大量射电信号, 以验证其在实际应用中的性能。射电天文学家需要处理越来越多的射电数据, 相信利用 PFB 技术进行频谱分析将会提取更多科学信息, 为今后的射电天文学发展作出力所能及的贡献。

参考文献:

- [1] 张雪莹, 郭铨, 郑倩, 等. 天文学报, 2023, 64(1): 96
- [2] 张宪超, 武继刚, 蒋增荣, 等. 电子学报, 2000, 28(5): 105
- [3] 张维良, 张彧, 杨再初, 等. 系统工程与电子技术, 2009, 31(8): 1819
- [4] 吴焰斌. 科技风, 2009, 9(3): 63
- [5] Schafer R, Rabiner L. IEEE Transactions on Audio and Electroacoustics, 1973, 21(3): 165
- [6] Bellanger M, Bonnerot G, Coudreuse M. IEEE Transactions on Acoustics, Speech, and Signal Processing, 1976, 24(2): 109
- [7] Zimmerman G A, Gulkis S. The Telecommunications and Data Acquisition Report, No.15. Pasadena, California: JPL, Calif of Tech, 1991: 141
- [8] John Bunton. <https://library.nrao.edu/public/memos/alma/memo342.pdf>, 2025
- [9] 陈林杰, 颜毅华, 刘飞, 等. 天文研究与技术, 2010, 7(2): 89
- [10] 朱凯, 甘恒谦, 朱岩, 等. 天文研究与技术, 2011, 8(1): 81
- [11] 吴俊霖, 金乘进, 朱岩, 等. 天文研究与技术, 2015, 12(4): 473
- [12] 赵欣, 金乘进, 朱岩, 等. 天文研究与技术, 2015, 12(4): 495
- [13] 张夏, 俞欣颖, 段然, 等. 天文学进展, 2016, 34(2): 249
- [14] 刘晔, 刘奇, 李健, 等. 天文研究与技术, 2020, 17(4): 429
- [15] 张海龙, 张萌, 张亚州, 等. 吉林大学学报(工学版), 2023, 53(8): 2388
- [16] Harris F J, Dick C, Rice M. IEEE Transactions on Microwave Theory and Techniques, 2003, 51(4): 1395

Application of Multiphase Filter Banks Technology in Low-Frequency All-Sky Total Power Measurement Experiments

WANG Xingtao¹, LIU Xiongwei¹, WANG Weihua²,
WU Yajun³, YU Yun³, YANG Xiaolong³

(1. School of Physics and Astronomy, China West Normal University, Nanchong 637002, China; 2. School of Computer Science and Information Engineering, Changzhou Institute of Technology, Changzhou 213032, China; 3. Shanghai Astronomical Observatory, Chinese Academy of Sciences, Shanghai 200030, China)

Abstract: As one of the important methods to detect the cosmic reionization epoch, the core goal of the low-frequency all-sky total power experiment is to detect the extremely weak neutral hydrogen 21 cm signal during the cosmic reionization period in the all-sky total power spectrum with a spectrum range of 50~200 MHz. At present, the fast Fourier transform (FFT) is usually used to calculate the power spectrum, and the spectral leakage problem makes it impossible to completely deduct Radio Frequency Interference (RFI), which seriously interferes with the extraction of weak cosmic reionization signals. Polyphase filter banks (PFB) is a technique that evenly divides signals by frequency, which can replace FFT to calculate the power spectrum. In this paper, the basic principle of PFB and its implementation based on compute unified device architecture (CUDA) are introduced, and the performance of PFB and FFT is compared through simulation, and the performance differences of PFB of different orders are compared. Finally, the measured and experimental data are processed to verify the superiority of PFB technology, which provides an effective solution for the spectrum analysis of the all-sky total power experiment .

Key words: CUDA; polyphase filter bank (PFB); GPU parallel computing; signal processing; polyphase filter; FFT operation