上海交通大學

SHANGHAI JIAO TONG UNIVERSITY

学士学位论文

THESIS OF BACHELOR

论文题目：大规模 MIMO 检测算法研究及芯片实现

学生姓名：	楼梦旦
学生学号：	515021910657
专　　业：	微电子科学与工程
指导教师：	贺光辉
学院 (系)：	电子信息与电气工程学院

大规模 MIMO 检测算法研究及芯片实现

摘 要

大规模多输入多输出（Massive Multi-Input Multi-Output, Massive MIMO）技术是第五代通信系统的关键技术之一，它能实现比传统 MIMO 技术更高的频谱效率，是无线通信技术达到 Gbps 的传输速率的核心。接收端信号检测的 BER 性能和复杂度是该技术实用化的关键，然而，随着基站天线规模的扩大以及用户数量的增加，检测复杂度急剧增加。因此，高检测性能低复杂度的算法，高吞吐率的大规模 MIMO 检测器是目前急需解决的问题。

在算法研究方面，本文基于 Non-Stationary Richardson 迭代和 Second-Order Richardson 迭代分别提出了两种信号检测算法。通过采用近似特征值估计、低复杂度初始化、以及矩阵向量积等策略，不仅有效避免了大规模乘法 Gram 矩阵的计算及其直接求逆，同时还消除了除法运算，从而在保证近乎最优性能的同时降低了超过 25% 的算法复杂度。仿真结果表明，天线配置为 128Œ16 的 MIMO 系统中，本文提出的两种算法在迭代三次时，检测性能十分接近精确求逆的最小均方误差算法。此外，针对用户数更多的大规模 MIMO 系统，以及实际信道，基于 Second-Order Richardson 迭代的算法相较于当前算法具备更优的性能。

在芯片实现方面，本文采用 Verilog HDL 硬件描述语言基于 Second-Order Richardson 检测算法采用脉动阵列结构，完成了首个支持可变用户数大规模 MIMO 系统的硬件设计。在 Xilinx Virtex-7 XC7VX980T FPGA 上的实现，吞吐率达 1302Mbps，硬件效率（Throughput/LUTs) 达 18043，明显优于 NS，CG，GS，PCI 和 SDJC 算法。此外，通过 SMIC 40nm，1.21V 工艺库进行芯片实现，检测器吞吐率达 3.0Gbps，时间延时低至 59 个时钟周期，归一化能量效率相较于 Weji、NS 和 CG 检测器分别提高为 1.48×、15.1× 和 1.76×。为下一代高性能通用 Gbps 大规模 MIMO 检测器的设计奠定了理论和实际基础。

关键词：大规模 MIMO 系统，检测算法，低复杂度，Second-order Richardson 迭代，Non-Stationary Richardson 迭代, 可变用户数

RESEARCH ON MASSIVE MIMO DETECTION ALGORITHM AND CHIP IMPLEMENTATION

ABSTRACT

Massive Multi-Input Multi-Output (Massive MIMO) technology is one of the key technologies of the fifth-generation communication system. It can achieve higher spectral efficiency than traditional MIMO technology, and Gbps transmission rate. The BER performance and complexity of the signal detection at the receiving end is the key to the practical application of this technology. However, as the size of the base station antenna increases and the number of users increases, the detection complexity increases dramatically. Therefore, high detection performance, low complexity algorithms, AND high throughput massive MIMO detector is urgently needed to be implemented.

In terms of algorithm research, two signal detection algorithms are proposed based on Non-Stationary Richardson iteration and Second-Order Richardson iteration. By adopting strategies such as approximate eigenvalue estimation, low complexity initialization, and matrix vector product, not only the calculation of large-scale multiplication Gram matrix and its direct inversion are effectively avoided, but also the division operation is eliminated, thereby achieving near optimal performance. Meanwhile, the algorithm complexity is reduced by over 25%. The simulation results show that the antenna configuration is 128×16 MIMO system. The two algorithms proposed in this paper are close to the minimum mean square error algorithm with exact inversion when the number of iteration is three. In addition, for large-scale MIMO systems with more users and actual channels, the algorithm based on Second-Order Richardson algorithm attains performance improvement comparing with current algorithms.

In this paper, based on the Second-Order Richardson detection algorithm, the first hardware design supporting the variable user number massive MIMO system is completed with systolic array structure. The implementation on the Xilinx Virtex-7 XC7VX980T FPGA has a throughput of 1302Mbps and the hardware efficiency (Throughput/LUTs) of 18043, which is significantly better than the NS, CG, GS, PCI and SDJC algorithms. In addition, the SMIC 40nm, 1.21V process library is used for chip implementation, the throughput of detector reaches 3.0Gbps, latency is as low as 59 clock cycles. Meanwhile, normalized energy efficiency compared to Weji, NS and CG detectors increased to 1.48×, 15.1×, and 1.76×, respectively. It lays a theoretical and practical foundation for the design of next-generation high-performance generic Gbps massive MIMO detectors.

KEY WORDS: Massive MIMO, Detection, Low Complexity, Second-Order Richardson, Non-Stationary Richardson, Variable Number of Users

目 录

毕业设计期间发表和投稿的学术论文 **iv**

插图索引

表格索引

第一章　绪论

1.1　研究背景及意义

自 1901 年，意大利发明家马可尼领导的实验小组发出无线电信号以来，无线通信技术迅猛发展，从只能提供基本语音业务的第一代移动通信（1^{st}-generation，1G）到传输速率达 100Mbps 的第四代移动通信系统（4^{th}-generation，4G），无线通信技术为人们的生活创造了无数的福祉和便利，成为推动人类社会向前发展的动力之一。然而，随着互联网的迅猛发展，车联网、云端服务、增强现实、智能医疗等技术与服务逐渐走入大众生活，移动数据流量需求将经历爆炸式的增长。与此同时，高质量的用户服务在传输速率、传输延迟和数据可靠性等方面对移动通信提出了更高的要求。移动通信基础作为“万物互联”的基础，正面临着前所未有的变革以满足新时期的极高要求。

为满足用户在各类场景下的移动通信体验需求，IMT-2020(5G) 推进组发布的《5G 愿景与需求白皮书》[1] 表明第五代移动通信（5^{th}-generation, 5G）系统支持超过 10Gbps 的峰值传输速率和数倍于 4G 的频谱效率，其中的关键技术之一，便是大规模 MIMO 系统。

多输入多输出（Multiple-Input Multiple-Output，MIMO）技术是无线通信领域的一个巨大突破。1908 年马可尼（Marconi）率先提出了利用多天线来抑制信道衰落的观点，90 年代贝尔实验室的工作进一步推动了 MIMO 技术在无线通信领域的发展。从 3GPP LTE（3rd Generation

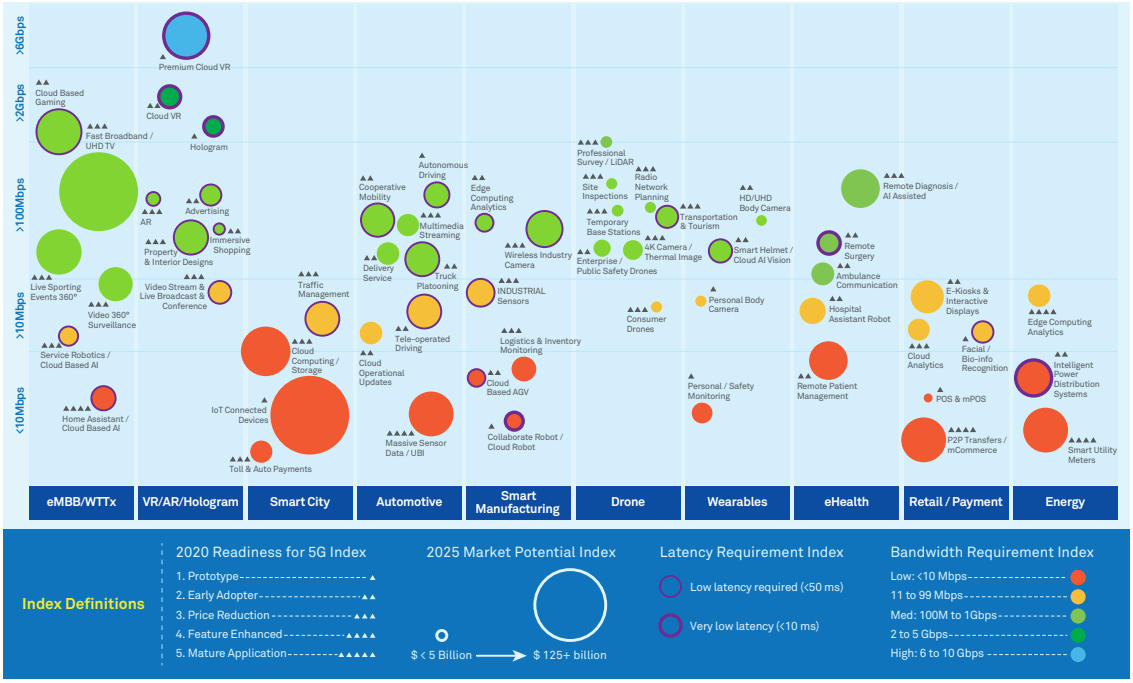


图 1–1 5G 潜在应用市场准备矩阵

Partnership Project Long Term Evolution）开始，MIMO 技术就被加入标准，并成功应用于 3G 和 4G 等系统中。MIMO 系统中，发射端和接收端同时配置多根天线进行无线传输，发送端并行发送多路信号，而接受端则需要从接收到的多个发射信号的叠加信号中恢复出原始发送信号。采用空间分集、空间复用和波束赋形等手段，利用多径传播的特点，MIMO 系统能够在不增加带宽情况下成倍地提高系统的容量和频谱利用率。

当前对传统 MIMO 技术的研究已经非常成熟，包括点对点（Point to Point）MIMO 和多用户（Multi-user）MIMO。然而，由于传统 MIMO 技术基站的天线数目较少，提供的系统性能有限，已不能满足日益增长的移动通信要求。与此同时，其演进技术大规模 MIMO（Massive MIMO or Large-Scale MIMO）由于仅需改变基站侧设备而无需更新用户终端设备就能够具有频谱效率更高、连接更稳定等特质，已成为 5G 的研究热点之一。

大规模 MIMO 技术是一项新兴技术，通过在基站配置几十根到上百根的大型天线阵列来同时服务数十个用户。利用大规模天线阵列带来的空间分集增益，能实现频谱效率提高近十倍的同时保证连接的可靠性。此外，由于终端接收功率不变，基站侧单根天线的发射功率相较于传统 MIMO 系统大大减小，低成本的功率放大器便能够满足要求，因而具有降低硬件开销和提升基站能量效率的潜力[2]。但是，天线数目的巨大增加在带来上述优势的同时也伴随着复杂度的急剧提升，尤其在下行链路预编码和上行链路信号检测中，算法复杂度高，硬件实现难度大，是 5G 实用化亟待解决的问题。

以上行链路的信号检测为例，它指利用信道估计模块得到的信道信息（Channel State Information, CSI），从接收信号中正确的恢复发送信号。在大规模 MIMO 系统中，基站采用空分复用（Space Division Multiplexing，SDM）技术，多个用户共享时域和频域资源。在一个复杂的传播环境中，不同数据流从多个方向同时到达，不同用户数据之间相互污染（contamination）和干扰（inference），信道的各种衰落和干扰等都为数据的恢复带来了困难。因此需要设计相应的检测算法来尽可能准确地恢复数据。然而，尽管半导体技术的快速发展显著提升了集成电路的计算能力，但是功率瓶颈（power bottleneck）仍限制着 IC 产业的发展[3]，因此，在大规模 MIMO 系统（如 128×8，128×16）中，传统的检测算法因硬件实现复杂度过高已不再适合实际应用。国内外学者针对大规模 MIMO 系统中的信号检测问题展开了广泛研究，旨在提出适用于大规模 MIMO 系统的低复杂度算法和硬件实现方案。

本次毕业设计以大规模 MIMO 系统上行链路的检测技术作为研究重点，对检测算法进行了深入的学习和研究。本文提出了两种分别基于 non-Stationary Richardson 迭代和 Second-Order Richardson 迭代的低复杂度线性检测算法。算法充分利用了大规模 MIMO 系统的特性，实现了接近最优的最小均方误差（Minimum Mean Square Error, MMSE）检测性能。两种算法均避免了 MMSE 算法中复杂的矩阵求逆运算，并采用三种策略进行优化。权衡两种算法的优缺点之后，对基于 Second-Order Richardson 迭代的检测算法进行了 FPGA 实现和 ASIC 实现，并将两者结果分别与现有文献进行了对比。结果满足大规模 MIMO 系统高吞吐率，高硬件效率和低误比特率的要求，为未来大规模 MIMO 系统信号检测的研究提供了新的思路和方法。

1.2 国内外研究现状及存在的不足

通过调研国内外大规模 MIMO 系统预编码和检测的相关文献，本节概括了目前该领域研究工作的成果以及存在的不足。

1.2.1 国内外研究现状

大规模 MIMO 系统检测算法的复杂度主要来源于两个方面，一个是计算 Gram 矩阵（信道矩阵 $\mathbf{H}$ 与其共轭转置的乘积，$\mathbf{H}^H\mathbf{H}$）时的大规模矩阵乘法，另一个是 Gram 矩阵的求逆运算。其中，国内外研究主要研究方向为降低 Gram 矩阵求逆复杂度，可以分为直接法和迭代法两大类。

直接法的思路是利用 Gram 矩阵为埃尔米特矩阵（Hermitian matrix）的性质，通过 LU 分解，QR 分解等方法得到精确求逆值。这种方法尽管可以得到完整的检测性能，但是硬件实现复杂度高。迭代法的思路是利用大规模 MIMO 系统 Gram 矩阵主对角元素占优的性质将矩阵求逆问题转换为线性方程组的求解问题，用迭代的方法来获得近似求逆值。这种方法，尽管性能有一定的损失，但是可以显著降低计算复杂度，硬件实现友好。迭代法根据其特点可以分为多项式展开法，经典迭代法，梯度搜索法和混合迭代法四种。

康奈尔大学由 C.Studer 教授带领的研究团队提出的基于纽曼级数（Neumann Series, NS）展开[4] 的方法是多项式展开法的代表。NS 算法利用纽曼级数展开的前 N 项作为 Gram 矩阵逆矩阵的近似值。该方法通过 FPGA 验证，在信道矩阵规模为 128×8 时，能够获得 621Mb/s 的高吞吐率。但是，当 N 较小时该算法相较于精确求逆的 MMSE 性损失较大，而高阶级数的使用则会使复杂度急剧增大以至于超过直接求逆的复杂度。

经典迭代法则通过线性方程的求解来近似逼近精确求逆值。代表算法有理查德森迭代算法（Richardson,RI)[5], 高斯塞德迭代算法（Gauss-Seidel，GS）[6]，Successive Overrelaxation 迭代（SOR）算法[7] 以及清华魏少军团队提出的并行 Chebyshev 迭代的算法（Parallel Chebyshev Iteration, PCI）[8] 等。该类方法避免了矩阵求逆计算，迭代形式简单，可以方便的通过增加迭代次数来提升性能，一般来说通过 3 次迭代就可以获得与精确求逆的 MMSE 算法几乎一致的性能。在相同的误比特率条件下，经典迭代法的复杂度均小于 NS 算法，但是在算法并行度和硬件吞吐率方面要低于 NS 算法。

梯度搜索法的本质也是线性方程的求解问题，但是与经典迭代法不同的是，它将线性方程的求解问题进一步转化为二次型的最优化问题。代表算法有最速下降法（Steepest Descent, SD）[9]，共轭梯度法（Conjugate Gradient, CG）[10]。此类方法同样不需要对 Gram 矩阵精确求逆，它通过求梯度来获得较快的收敛速度，尤其是第一次迭代的时候能显著提高收敛速度，但是在后续迭代中效果不是很好。此外，这类方法引入了除法运算，提高了算法复杂度和硬件实现难度。

混合迭代法是近年来出现的基于上述算法，存优去粕，将不同方法糅合的算法。此类方法的特征是在初次迭代时通过梯度搜索法得到较快的收敛速度，然后在后续的迭代中采用较为简单的迭代方法降低复杂度。相应的，它存在和梯度共轭法一样的缺点，即引入了除法运算。代表算法有混合 SD 算法和 JC 算法的 SDJC 算法[11] 和混合牛顿算法和 Richardson 算法的 NRI 算法[12]。

前文提到 MMSE 算法复杂度的另一个主要来源是计算 Gram 矩阵时的大规模矩阵乘法，而上述算法除了 PCI 算法之外均没有对这部分计算进行减免和优化。目前，能够避免 Gram 矩阵计算的 MMSE 检测算法主要有优化坐标下降法（Optimized Coordinate Descent, OCD），但是这种算法引入了较多的除法运算操作，从另一个角度提升了硬件实现复杂度。此外由于这种方法无法从 Gram 矩阵的特性中受益，性能表现较差。

1.2.2 目前存在的不足

如前文所述，预编码和检测算法的研究是近似的，综合上述对预编码和检测领域研究文献的讨论，大规模 MIMO 系统的检测算法还存在以下几方面的不足。

第一，现有算法在复杂度和性能上没有取得很好的平衡。上述算法往往通过迭代次数的增加来提升性能，而这也意味着复杂度的进一步提升。以 NS 算法为例，只有在级数保留项大于等于 4 的时候才能获得可接受的性能，而此时算法复杂度已经超过了直接求逆的复杂度。

第二，上述算法没有避免 Gram 矩阵的计算。在信道矩阵 $\mathbf{H} = B \times U$ 时，即基站配置 B 根天线同时服务 U 个用户，计算 Gram 矩阵的实数乘法复杂度为 $O(4BU^2)$，这个复杂度已经超过了对 Gram 矩阵直接求逆的复杂度。因此，想要进一步降低检测算法的复杂度，必须简化 Gram 矩阵的计算。

第三，由于迭代的特性所导致的迭代结果间的数据依赖，目前大部分的硬件实现结果的吞吐率都不能满足 5G 通信所要求的的 100Mbps 1Gbps 的传输速率。而直接减少迭代次数来获得吞吐率提升的做法会导致系统性能显著下降。目前线性检测算法的文献多以论述算法为主，对硬件结构优化的研究较少。大规模 MIMO 系统亟待一个低复杂度，高吞吐率，高硬件效率的硬件结构。

基于目前研究成果和不足，本文对大规模 MIMO 系统检测算法进行了深入研究，研究内容如下三个方面所示：

第一，基于 Non-Stationary Richardson 迭代和 Second-Order Richardson 迭代，本文提出了两种低复杂度的检测算法，分别简称为 proposed-1 算法和 proposed-2 算法。proposed-1 算法相较 proposed-2 算法复杂度更低，proposed-2 算法具有收敛速度快，对不同规模 MIMO 系统适应性高的特点。

第二，本文采用了三种策略对算法进行优化。首先，利用信道矩阵的统计特性，无需进行特征值计算就获得了迭代所需的最优参数，极大地降低了算法复杂度。其次，提出的低复杂度初始化策略加速了迭代的收敛。最后，利用矩阵向量乘操作代替大规模矩阵乘法，大幅减少了乘法次数，从 $O(4BU^2)$ 减少到 $\triangle\mathcal{BU}$。

第三，本文对两种算法进行权衡之后，基于 Second-Order Richardson 迭代的检测算法提出了高吞吐率、高硬件效率的硬件架构并完成了硬件实现。然后给出了硬件实现结果和 ASIC 实现结果与现有文献的比较。结果显示在算法保证更优性能的前提下，本文的实现结果还具有较高的吞吐率和硬件效率。

本文共分为六章具体安排如下：

第二章主要介绍大规模 MIMO 系统及其检测算法，包括大规模 MIMO 系统上下行链路的系统模型，大规模 MIMO 的信道矩阵特性，检测技术的原理，常见的检测算法并具体分析

其各自的优缺点。

第三章详细讨论了本文所提出的基于 Non-Stationary Richardson 迭代和基于 Second-Order Richardson 迭代的低复杂度高性能检测算法，介绍了从算法层面提出的改进策略，使得改进后的算法能够在保证与 MMSE 检测性能相近的前提下降低低复杂度。同时，从数学推导和仿真验证两个角度对收敛性进行分析；给出了算法的误比特率性能在不同条件下的仿真结果和复杂度分析并与现有算法进行了比较，并对提出的两种进行了横向对比等。

第四章基于 Second-Order Richardson 检测算法进行 FPGA 实现，展示了本文所采用的硬件架构，并对关键模块的实现进行了详细介绍。

第五章基于第四章的硬件实现，给出了 FPGA 实现结果和 ASIC 实现结果，给出了与现有成果的比较结果。

第六章是对全文工作的总结，分析现有研究成果，总结创新点与不足，对未来的工作进行指导与展望。

1.3 本文用到的数学符号说明

本文以大写粗体字母表示矩阵，下标表示其维度，如 $B \times U$ 维信道矩阵表示为 $\mathbf{H}_{B\times U}$，其中 $h_{i,j}$ 表示矩阵 $\mathbf{H}$ 的第 i 行和第 j 列元素。矩阵 $\mathbf{H}$ 的转置，共轭转置和逆矩阵分别表示为 $\mathbf{H}^T$，$\mathbf{H}^H$，$\mathbf{H}^{-1}$. U 阶单位阵和 U 阶零矩阵分别用 $\mathbf{I}_U$ 和 $\mathbf{0}_N$ 表示。

向量由小写粗体字母表示，下标 i 表示 $\mathbf{x}_i$ 的第 i 个元素。向量 $\mathbf{x}$ 的 $L2$ 范数定义为 $\|\mathbf{x}\|_2 = \sqrt{\Sigma_k |x_k|^2}$，$\Re(\cdot)$ 表示向量的实部，$\Im(\cdot)$ 表示向量的虚部。

大写字母 B 表示基站天线数，大写字母 U 表示用户天线总数。$\mathbb{R}$ 表示实数集，$\mathbb{C}$ 表示复数集。

第二章　大规模 MIMO 系统中的检测技术

本章主要介绍大规模 MIMO 系统及检测技术的原理，并分析现有研究各算法的优缺点。首先介绍大规模 MIMO 系统上下行链路的基本模型，信号检测与预编码的数学模型。其次，对大规模 MIMO 系统的信道矩阵特性做详细介绍，并分析其对检测的影响。最后，详细介绍与分析了大规模 MIMO 系统现有的典型检测算法。

2.1　大规模 MIMO 系统模型

在 5G 通信中，大规模 MIMO 系统主要应用可以分为如图2–1所示的多小区多用户多天线和单小区多用户多天线的通信场景。多小区的通信场景中，每个小区的多个用户与该小区配备了大规模天线阵列的基站进行通信；为了独立研究检测问题，本文不考虑小区间的干扰，将其中一个小区独立出来得到如图2–1(b) 所示单小区内一个基站与多个用户进行通信的场景。

大规模 MIMO 系统可以分为对称和非对称两种形式，其判断标准为基站天线数与用户数量的比值大小。对称的大规模 MIMO 系统指基站的天线数量与同时服务的用户数量接近的情况，在该条件下预编码和信号检测只能采用直接法求逆，迭代算法收敛的速度很慢，性能较差且复杂度高。在非对称的大规模 MIMO 系统中，用户数量远小于基站天线的数量，通常为数十个，此时信道矩阵具备明显的硬化效应，采用近似求逆法的信号预编码和检测的性能相较对称条件下的近似求逆有很大的提升[15]。换言之，非对称的大规模 MIMO 系统有利于降低预编码和检测的复杂度。因此，目前预编码和检测的相关研究大多基于非对称大规模 MIMO 系统展开，本文选取了信道矩阵维度为 128×8，128×16 的大规模 MIMO 系统典型配置，同时考虑 128×32 的系统配置，使算法在满足非对称的情形的同时，也逐渐往对称情形过渡。

根据信号的传递方向，大规模 MIMO 通信系统可以分为下行链路和上行链路。下行链路指基站将信号经过预编码之后发送给用户，由用户独立完成检测；上行链路指多路用户将信号发送到基站，基站接收到受损信号后进行检测。

(a) 多小区多用户多天线通信[13]　　(b) 单小区单用户多天线通信[14]

图 2–1 大规模 MIMO 移动通信场景

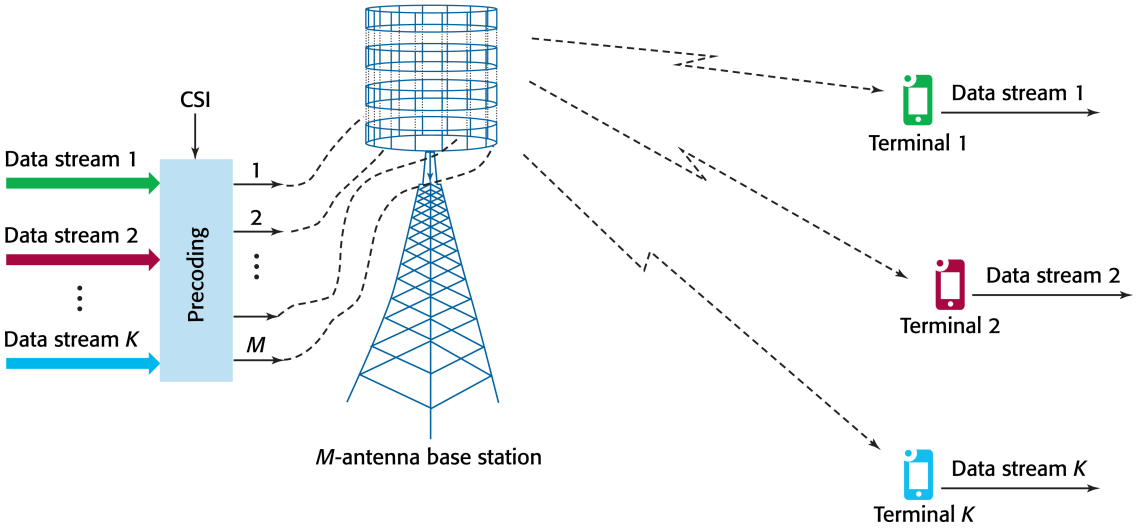


图 2–2 大规模 MIMO 系统下行链路

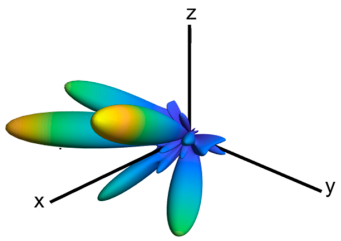


图 2–3 大规模 MIMO 系统波束成形直观效果

2.1.1 下行链路：预编码

如图2–2所示为大规模 MIMO 系统下行链路示意图[16]。基站侧的信号经过信道编码，交织，调制和预编码之后通过大规模的天线阵列发送给多路用户，信号传递时信道中的各种干扰和噪声会使信号受损，用户接收到受损信号后独立完成检测，获取属于自己的信息。对基站与用户间信号传递空间的信道特征加以表征即为信道信息（channel state information，CSI）。

受限于用户侧的设备体积和功率，用户难以完成复杂的信号检测，而基站侧往往配置了处理能力更强的设备。因此，为了简化用户端接收机的设计，必须在基站侧对待发送的信号进行预处理，以尽可能的消除信道传递中引入的用户间干扰和噪声。这种利用信道信息对信号预处理的技术即为预编码技术。大规模 MIMO 系统的预编码技术可以通过波束成形实现，其直观理解如图2–3所示[17]，通过利用多路径特征，在不同方向上使用不同的波束实现信号的预处理。

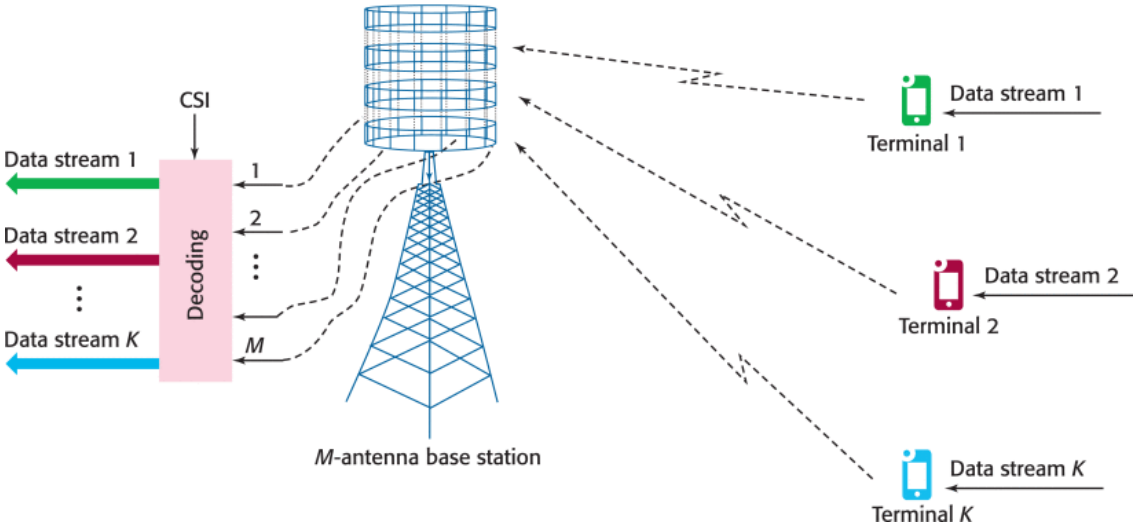


图 2–4 大规模 MIMO 系统上行链路

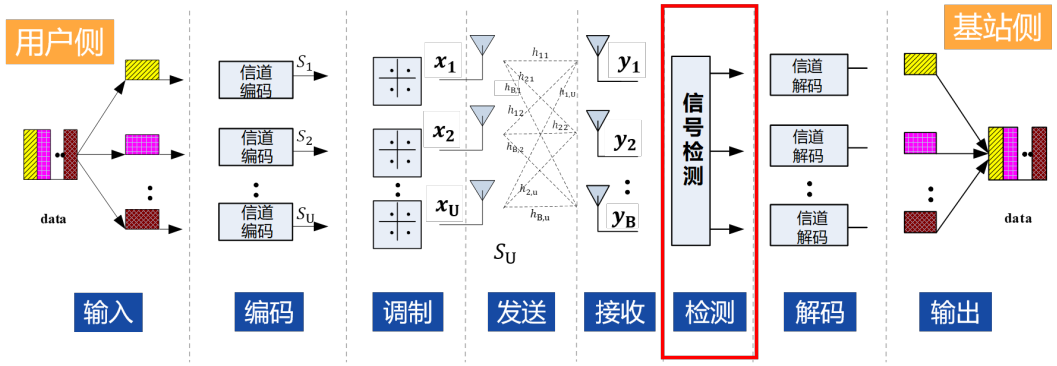
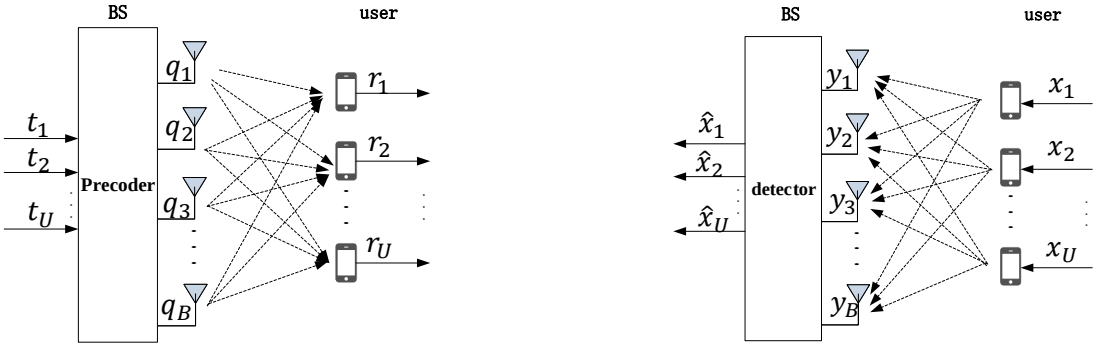


图 2–5 大规模 MIMO 系统上行链路中检测所在的位置

2.1.2 上行链路：检测

如图2–4所示为大规模 MIMO 系统上行链路示意图[16], 其中检测算法在大规模 MIMO 系统上行链路中的位置如图2–5所示。每个用户将各自经过编码，交织和调制后的信息同时发送至基站，信道中的各种干扰和噪声会对传输的信号产生污染，如用户间干扰（Inter-User Interference, IUI）。基站接收到受损的多用户联合信号后，首先通过导频序列进行信道估计，获取当前的信道信息，然后利用信道信息对信号进行检测，最后经过解交织和译码等步骤还原出各用户发送的信息。

上行链路的关键问题为信号检测，其难点在于如何消除用户间干扰和噪声从而尽可能准确的恢复出用户发送的原始信号。



(a) 下行链路预编码数学模型　　(b) 上行链路检测数学模型

图 2–6 大规模 MIMO 系统数学模型

2.1.3 大规模 MIMO 系统预编码和检测的数学模型

大规模 MIMO 系统下行链路的数学模型如图2–6(a) 所示, 假设基站发射的符号向量为 **t**，预编码矩阵为维度等于为 $B \times U$ 的复数矩阵 **P**，则基站 B 维复数发射向量 **q** 可以表示为：

$$\mathbf{q} = \mathbf{P}\mathbf{t} \tag{2–1}$$

用户的接收向量可表示式 (2–2) 所示, 其中噪声 $\mathbf{n}_d \in \mathbb{C}_{U\times 1}$ 满足独立同分布（i.i.d)，每个元素均值为 0，方差为 σ^2，$\mathbf{H}_d$ 表示 $U \times B$ 维理想信道估计。独立研究预编码技术时通常假设基站拥有理想的信道估计，因此如何减少用户信号 **r** 之间的干扰是预编码技术的研究重点。如前文所述，预编码与信号检测的本质是一致的，因此限于本文篇幅，此处仅提供预编码的数学模型而不对预编码算法进行介绍。

$$\mathbf{r} = \mathbf{H}_d\mathbf{q} + \mathbf{n}_d \tag{2–2}$$

大规模 MIMO 系统上行链路的系统数学模型如图2–6(b) 所示，U 个单天线用户发射的信息经过编码调制后并行发送给基站，基站在接收到受损的混合信号之后通过信道估计和检测技术得到信息比特流，然后通过解交织和译码得到接收比特。大规模 MIMO 检测的数学模型可以用式 (2–3) 表示。其中，$\mathbf{x} \in \mathbb{C}_{U\times 1}$ 表示用户侧发送的信号向量，用户 i 的传输功率定义为 $\mathbb{E}\{|x_i|^2\} = E_s$，$\mathbf{y} \in \mathbb{C}_{B\times 1}$ 表示基站接收向量。噪声 $\mathbf{n}_U \in \mathbb{C}_{B\times 1}$ 满足均值为 0，方差为 σ^2 的高斯独立同分布 (i.i.d)。同样的，为了独立研究检测技术，假设基站拥有理想的信道估计矩阵 $\mathbf{H}_U \in \mathbb{C}_{B\times U}$。

$$\mathbf{y} = \mathbf{H}_U\mathbf{x} + \mathbf{n}_U \tag{2–3}$$

与预编码类似的，上行链路检测的关键问题为如何根据信道估计和接收向量 **y** 尽可能的消除各天线间的干扰以获得更优的性能。由于本文着重讨论大规模 MIMO 系统的检测算法，为方便起见，下文中信道矩阵简写为 **H**, 噪声简写为 **n** 因此接收向量可以表示为

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \tag{2–4}$$

其中，**H** 可以表示为式 (2–5)。

$$\mathbf{H} = \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,U} \\ h_{2,1} & h_{1,2} & \cdots & h_{1,U} \\ \cdots & \cdots & \ddots & \cdots \\ h_{B,1} & h_{B,2} & \cdots & h_{B,U} \end{bmatrix} \tag{2–5}$$

2.2 大规模 MIMO 系统检测算法

传统 MIMO 系统的检测算法可以分为最优（optimal）和次优（sub-optimal）检测算法两大类。最优的检测算法包括最大似然检测算法和最大后验概率检测（Maximum A Posteriori estimation, MAP）两种，但是两种算法均因复杂度过高而无法硬件实现，通常被用作性能的参照标准。为了降低检测的复杂度，一系列次优检测算法被提出，可分为非线性检测算法和线性检测算法两大类。非线性检测算法如干扰消除（Interference Cancellation, IC) 检测算法，树搜索（Tree Search) 算法等。典型线性检测算法则包括迫零法（Zero Forcing, ZF) 和最小均方误差检测算法等。在传统 MIMO 系统中，非线性检测算法能够以一定的复杂度为代价获得好于线性检测算法的性能。

然而，随着大规模 MIMO 系统中天线数增加为传统 MIMO 系统的数倍，传统的检测算法变得不适应。非线性算法的复杂度显著提高难以硬件实现，同时线性算法由于信道硬化等效应已经能够获得近乎最优 (near-optimal) 的性能。因此，线性算法以其复杂度大幅降低，性能较优的特点吸引国内外众多学者展开研究。本文将详细分析线性算法，而对非线性算法原理只做简要的介绍，不给出具体的实现细节分析。

2.2.1 非线性检测算法

(1) 干扰消除算法

干扰消除算法可以分为串行干扰消除（Sequential Interference Cancellation, SIC）算法和并行干扰消除算法（Parallel Interference Cancellation, SIC）等，其基本思想为通过判决反馈在接收端对每个用户消除已检测信号对待检测信号产生的影响。两者的区别在于串行干扰消除采用对多个用户逐个进行数据判决的策略，而并行干扰消除则在每一阶都同时对多个用户的多址干扰进行处理，但是两种方法都是多级的。

串行干扰消除结构简单，但是时延较长，且前阶判决错误会对后续的判决产生较大影响，使得各阶性能严重下降。并行干扰虽然时延小但是计算量大。两者对用户的功率要求较高，在多径信道中，功率控制不理想，检测性能降低。

(2) 树搜索算法

传统 MIMO 系统中树搜索算法包括球形译码（Sphere Decoding, SD）算法和 K-Best 算法等。SD 算法的基本思想为在给定的球形半径 d 中对所有格点进行搜索，若球内没有找到相应的格点则扩大搜索半径 d 继续搜索，直到搜索到相应的格点作为检测结果。它不像采用穷尽

搜索的 ML 算法需要在所有格点的范围内进行搜索，由此降低了复杂度和搜索时间。SD 算法属于深度优先算法，树的第 k 层节点对应为落在球形半径 d 内，深度为 k 的格点。K-Best 算法首先对信道矩阵的共轭转置矩阵 $\mathbf{H}'$ 进行 QR 分解得到 $\mathbf{H}' = \mathbf{QR}$，其中 $\mathbf{Q}$ 为正交矩阵，$\mathbf{R}$ 为上三角矩阵。然后构造 K 层树形结构，每层的子节点数目与调制方式相关，以欧氏距离为标准，按照广度优先搜索策略来得到最终检测结果。目前对于 SD 算法和 K-Best 已有许多改进策略，但是这两种算法在大规模 MIMO 仍有复杂度过高的缺点，目前多用于小规模如 4×4 的 MIMO 系统。

2.2.2 线性检测算法

大规模 MIMO 系统线性检测算法的原理可表示为式 (2–6)，其中 $\hat{\mathbf{x}}$ 为基站还原的信息向量，$\mathbf{W}$ 为权重矩阵。

$$\hat{\mathbf{x}} = \mathbf{W}\mathbf{y} \tag{2–6}$$

根据权重矩阵的不同可以将线性算法分为匹配滤波（Matched Filter, MF）检测算法，迫零检测算法和最小均方误差检测算法。由于匹配滤波算法保留了较强的干扰，性能较差。只有当信道矩阵 $\mathbf{H}$ 具有较好的行正交性时，匹配滤波算法能够获得较好的性能，因此不给出详细介绍。

(1) ZF

迫零检测算法的权重矩阵可以表示为式 (2–7)。

$$\mathbf{W} = (\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H \tag{2–7}$$

因此迫零检测算法的检测结果可以表示为

$$\hat{\mathbf{x}} = (\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H\mathbf{y} = \mathbf{x} + (\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H\mathbf{n} \tag{2–8}$$

因为迫零法的权重矩阵为信道矩阵 $\mathbf{H}$ 的右伪逆，通过 ZF 信道检测，信道的影响被完全消除，相当于消除了用户间的干扰，但是它同时会受到噪声增强效应的影响。这种不良效应在低信噪比的情况下尤为突出，严重影响 ZF 检测算法的性能。因此 ZF 检测算法更适用于高信噪比的场景。

(2) 最小均方误差检测算法（MMSE）

不同于 ZF 算法，MMSE 算法旨在寻求噪声增强与消除干扰之间的一个平衡，其权重矩阵可以表示为

$$\mathbf{W} = (\mathbf{H}^H\mathbf{H} + \sigma^2\mathbf{I}_U)^{-1}\mathbf{H}^H = A^{-1}\mathbf{H}^H \tag{2–9}$$

其中 $\sigma^2 = N_0/E_S$。因此基站的检测结果可以表示为

$$\hat{\mathbf{x}} = \mathbf{A}^{-1}\mathbf{H}^H\mathbf{y} \tag{2–10}$$

因为 $\mathbf{A}^{-1}$ 不是对角阵，因此 MMSE 算法没有完全消除用户间的干扰。相比于 ZF 算法，MMSE 算法几乎没有提升复杂度，但是在低信噪比的条件下，拥有比 ZF 算法更优的误码性能，高

信噪条件下，MMSE 算法的性能趋近于 ZF 算法。可以看出 MMSE 检测算法更适合于大规模 MIMO 检测系统。

2.3 基于 MMSE 检测的迭代算法

2.3.1 大规模 MIMO 系统的信道矩阵特性

本节主要介绍大规模 MIMO 系统信道矩阵的特性及其对信号检测的影响，便于读者理解后文的具体迭代算法介绍。Gram 矩阵定义为 $\mathbf{G} = \mathbf{H}^H\mathbf{H}$，MMSE 算法在此基础上引入了噪声，表达式为

$$\mathbf{A} = \mathbf{H}^H\mathbf{H} + \sigma^2\mathbf{I}_U \tag{2–11}$$

在大规模 MIMO 系统中，Gram 矩阵 $\mathbf{A}$ 具备以下两条性质：

(a) 共轭对称正定

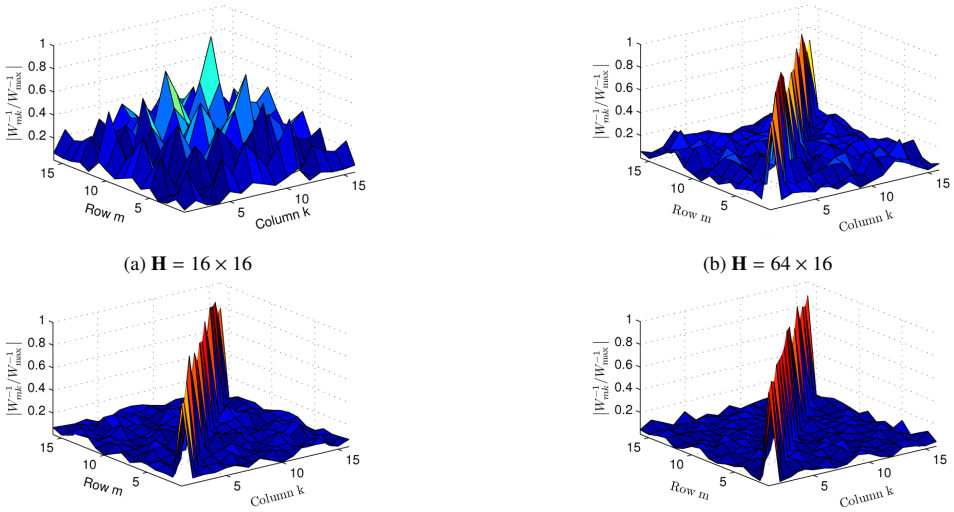
$$\begin{cases} \mathbf{A} = \mathbf{A}^H \\ \mathbf{x}\mathbf{A}\mathbf{x}^H = \mathbf{x}\mathbf{H}\mathbf{H}^H\mathbf{x}^H = (\mathbf{x}\mathbf{H})(\mathbf{x}\mathbf{H})^H > 0 \end{cases} \tag{2–12}$$

因此 $\mathbf{A}$ 是一个共轭对称正定矩阵。

(b) 主对角元素占优

信道硬化效应的本质为 Gram 矩阵主对角元素占优，在大规模 MIMO 系统上行链路中，当基站天线数与用户数的比值很大时，信道矩阵的各行将趋于正交[6]。由图2–7可以看出，只有在大规模 MIMO 系统（图2–7(c) 和图2–7(d)）中，矩阵主对角元素才显著占优，此时信道硬化效应明显。

大规模 MIMO 系统中的信道硬化效应使基于 MMSE 检测的线性算法取得接近最优的性能，这是因为此时等效信道矩阵由于主对角元素占优，可近似于一个对角阵，这等价于用户间干扰变得十分小，与传统小规模 MIMO 系统中非线性算法达到的效果一致。



(a) $\mathbf{H} = 16 \times 16$　(b) $\mathbf{H} = 64 \times 16$

(c) $\mathbf{H} = 128 \times 16$　(d) $\mathbf{H} = 256 \times 16$

图 2–7 瑞利信道下，不同天线配置的 MIMO 系统 Gram 矩阵分布

上文对线性检测算法和大规模 MIMO 系统信道矩阵特性进行了阐述，大规模 MIMO 系统中 MMSE 检测算法能够获得近乎最优的性能，但是考虑到天线矩阵规模的扩大，MMSE 算法复杂度仍较高，难以硬件实现。目前主流研究集中在基于 MMSE 算法的权重矩阵对其进行优化上，旨在进一步降低复杂度的同时尽可能减少性能的损失。

2.3.2 迭代算法的求逆思路

MMSE 算法复杂度主要来源于 Gram 矩阵 A 的计算及其求逆，通过避免矩阵直接求逆可以大幅降低算法复杂度。观察线性方程组 $\mathbf{Ax} = \mathbf{b}$ 的求解等式和 MMSE 检测结果如式2–13所示。令 $\mathbf{b} = \mathbf{H}^H\mathbf{y}$，可以看到两者的表达式完全等价。

$$\begin{cases} \hat{\mathbf{x}} = \mathbf{A}^{-1}b \\ \hat{\mathbf{x}} = \mathbf{A}^{-1}\mathbf{H}^H\mathbf{y} \end{cases} \tag{2–13}$$

得益于 Gram 矩阵主对角元素占优的性质，迭代法能够很快的逼近精确解 $\mathbf{x}$。由此出发可以用迭代法对 Gram 矩阵近似求逆以降低复杂度。根据1.2.1的分类，下面将对各类算法的具体原理，迭代形式及其优缺点展开介绍。各迭代算法的评估指标为性能与硬件复杂度，其中性能用误比特率衡量，复杂度主要用乘法和除法运算次数衡量。

2.3.3 多项式展开法

(1) 纽曼级数法（NS）

NS 算法[4] 的迭代形式可以表示为

$$\mathbf{A}^{-1} = \Sigma_{n=0}^{\infty}(\mathbf{D}^{-1}(\mathbf{D}-\mathbf{A}))^n\mathbf{D}^{-1} \tag{2–14}$$

其中 $\mathbf{D}$ 为 $\mathbf{A}$ 的对角阵。NS 的精度由迭代次数决定，当迭代次数 n 小于 3 时，NS 求逆的复杂度为 $O(U^2)$，小于直接求逆的复杂度 $O(U^3)$，但是其误码性能远差于直接求逆的 MMSE 算法。当迭代次数大于等于 3 时，NS 算法的复杂度进一步提升，与直接求逆持平甚至超过直接求逆，这时采用 NS 算法就得不偿失了。

2.3.4 经典迭代法

经典迭代法的基本思想为每一次迭代结果 $\mathbf{x}_k$ 加上一个增量得到新的迭代结果 $\mathbf{x}_{k+1}$，逐渐逼近精确解 $\mathbf{A}^{-1}\mathbf{b}$。本小节方法迭代的形式如式2–15所示，其中 $\triangle$ 表示每次迭代的增量，k 为当前迭代次数。

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \triangle \tag{2–15}$$

本小节将介绍不同增量表达式的检测算法。

(1) RI

Richardson 算法[5] 的迭代形式可以表示为

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha(\mathbf{b} - \mathbf{A}\mathbf{x}_k) \tag{2–16}$$

其中，为满足 RI 算法的收敛性，α 需要满足以下条件

$$0 < \alpha < \frac{2}{\lambda_{max}(\mathbf{H}^H\mathbf{H})} \tag{2–17}$$

α 为迭代的松弛因子，其数值为常数，相较于 NS 算法而言，其一次迭代的复杂度较高。但是随着迭代次数的增加，其复杂度增加缓慢。在相同迭代次数下，RI 能够获得比 NS 更优的误码性能。

(2) GS

与 RI 算法不同的是，GS 算法[6] 将 Gram 矩阵分为下三角矩阵 **L** 和对角阵 **D** 进行求解，其表达式为

$$\mathbf{x}_{k+1} = (\mathbf{L} + \mathbf{D})^{-1}(\mathbf{b} - \mathbf{L}^H\mathbf{x}_k) \tag{2–18}$$

尽管 GS 算法拥有比 RI 算法更快的收敛速率，在相同的迭代次数下能够获得更好的误码性能，但是由于引入了除法操作，复杂度相较 RI 算法显著提高。

(3) SOR

在 GS 算法的基础上，SOR 算法[7] 通过引入松弛因子 ω，使算法拥有更快的收敛速度。其迭代形式可表示为

$$\mathbf{x}_{k+1} = (\mathbf{L} + \frac{1}{\omega})^{-1}\left[\left((\frac{1}{\Omega}\mathbf{D}) - \mathbf{L}^H\right)\mathbf{x}_k + \mathbf{b}\right] \tag{2–19}$$

在 $0 < \omega < 2$ 的条件下，SOR 算法总是收敛的。值得注意的是，在每次迭代中 SOR 算法比 GS 算法多了 U 次除法运算，复杂度有所提高，硬件实现更不友好。但是在相同迭代次数下，能够获得比 GS 算法更优的性能。

(4) PCI

PCI 算法[8] 基于切比雪夫迭代，利用大规模 MIMO 系统的信道特性做了一定的改进，其优点有二，一个是避免了除法运算，一个是消除了 Gram 矩阵的计算，大幅降低了计算的复杂度。同时它还针对平坦衰落瑞利信道进行了优化，进一步降低了复杂度，但这也是其弱点之一，即该算法不普适于所有的信道模型。PCI 算法的迭代形式如式 (2–20) 所示。

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k + \sigma_k \\ \sigma_k &= \rho_k\mathbf{r}_k + \phi_k\sigma_k \\ \mathbf{r}_k &= \mathbf{y}^{MF} - N_0E_s^{-1}\mathbf{x}_k - \mathbf{H}^H(\mathbf{H}\mathbf{x}_k) \end{aligned} \tag{2–20}$$

其中，ρ_k，ϕ_k 为根据迭代次数计算出的参数。同时 PCI 算法的另一个不足为它需要存储前两次迭代结果，提高了数据结果间的依赖性，不利于硬件实现。

(a) SD 算法　　(b) CG 算法

图 2–8 SD 算法和 CG 算法搜索方向示意图

2.3.5 梯度搜索法

梯度搜索法与二次型最优化问题的求解思路类似，通过将求逆问题转化为求极值问题求解。二次型的最优化问题常见的方法有 SD 算法，CG 算法，牛顿迭代法等。例如函数

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{b}^T\mathbf{x} \tag{2–21}$$

求梯度可得

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = grad(f) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_4} \end{bmatrix} = \mathbf{A}\mathbf{x} - \mathbf{b} \tag{2–22}$$

其迭代形式可以用式2–23表示。通过计算梯度来确定搜索方向 □, 然后乘以搜索步长 △ 得到迭代的增量 □ · △。

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \square \cdot \triangle \tag{2–23}$$

(1) SD

梯度的负方向是函数局部下降最快的方向，因此最速下降法（SD 算法）[9] 通过在每次迭代中求解出梯度，取其负方向作为搜索方向，以此循环迭代收敛至精确解附近。其迭代形式可表示为

$$\mathbf{x}_{k+1} = \mathbf{x}_k + a_k + \mathbf{r}_k \tag{2–24}$$

其中 $\mathbf{r}_k$ 为迭代搜索方向，$a_k = \frac{\mathbf{r}_k{}^T\mathbf{r}_k}{\mathbf{r}_k^T\mathbf{A}\mathbf{r}_k}$ 为迭代搜索步长（step size）。该方法在迭代初期尤其是第一次迭代时收敛速度很快，但是随着迭代次数的增加，容易陷入局部最优的困境。越接近精确解，搜索步长越小，收敛速度相应的变慢，其收敛路径，如图2–8所示呈锯齿状像精确解靠近[18]。

(2) CG

为了弥补 SD 算法后期收敛速度慢的缺点，梯度下降法（CG 算法）对搜索方向进行了优化，即每一次迭代后的残差向量都与之前所有的残差向量正交[10]。其完整的迭代形式如下所示

CG 检测算法

Input: $\mathbf{H}$, $\mathbf{y}$, σ^2, K

Initialization:

1. $\mathbf{b} = \mathbf{H}^H\mathbf{y} and \mathbf{A} = \mathbf{H}^H\mathbf{H} + \sigma^2\mathbf{I}_U$
2. $\mathbf{v}_0 = 0, \mathbf{r}_0 = \mathbf{b}, \mathbf{p}_0 = \mathbf{r}_0$

for k $= \mathbf{1}, \mathbf{2} \cdots$, K **do**

3. $\mathbf{e}_k = \mathbf{A}\mathbf{p}_{k-1}$
4. $\alpha_k = \|\mathbf{r}_{k-1}^2/(\mathbf{p}_{k-1}^H\mathbf{e}_{k-1})$
5. $\mathbf{v}_k = \mathbf{v}_{k-1} + \alpha_k\mathbf{p}_{k-1}$
6. $\mathbf{r}_k = \mathbf{r}_{k-1} - \alpha_k\mathbf{e}_{k-1}$
7. $\beta_k = \|\mathbf{r}_k\|^2/\|\mathbf{r}_{k-1}\|^2$
8. $\mathbf{p}_k = \mathbf{r}_k + \beta_k\mathbf{p}_k$

end for

其中，$\mathbf{p}_k$ 为每次迭代的搜索方向，可以看出 CG 算法第一次迭代与 SD 算法第一次迭代的搜索方向是一致的。尽管 CG 算法在相同迭代次数内拥有比 SD 算法更优的性能，但是由于其每次迭代需要两次除法，复杂度有所提高。

2.3.6 混合迭代法

混合迭代法代表算法有 SDJC 算法, 也有 NRI 算法等较新的研究[11, 12]。其基本迭代形式如下所示。

混合迭代法基本形式

Iterations

for k = 1 : n do

1. iteration 1

end for other iteration

for k = n+1 : K do

2. iteration method 2

end for

(1) SDJC

SDJC 算法结合了一次最速下降法和 K-1 次雅克比迭代法，其中 Jacobi 迭代属于经典迭代法中最为简单的一种。SD 算法在第一次迭代时以较高的收敛速度逼近精确解，然后通过形式简单复杂度低的 Jacobi 迭代进一步获得更优的性能。在大规模 MIMO 系统中，尤其是基站天线数 B 远大于 U 的条件下，仅需数次迭代就能获得近乎最优的性能，因此 SDJC 算法中仅使用了一次 SD 迭代。该方法的缺点一个是 SD 算法引入了除法提高了复杂度，另一个是在如 128×32 规模的 MIMO 系统中，该算法失去收敛性。

(2) NRI

NRI 算法[12] 结合了牛顿迭代法和 RI 算法。以一定次数的牛顿迭代提高收敛速度，然后利用一定次数低复杂度的 RI 迭代算法逼近直接求逆的值。但是，由于前述方法包括 NRI 在内，在迭代次数等于 3 时就已经逼近 MMSE 的性能，因此 n 值可以发挥的空间很小。此外，牛顿迭代法的迭代形式如式 (3–1) 所示。注意到指数项 2^{k-1}，因此当 n 为 2 时，牛顿迭代的复杂度就达到了 $O(2BU^2 + 2U^3)$，难以硬件实现。

$$\mathbf{x}_k = \mathbf{x}_{k-1} + (\mathbf{I}_U - \mathbf{P}_0^{-1}\mathbf{H}^H\mathbf{H} - \mathbf{V})^{2^{k-1}}\mathbf{x}_{k-1} \tag{2–25}$$

2.3.7 当前迭代算法的主要问题

上节对现有基于 MMSE 检测的迭代算法进行了分类介绍，分析每一种方法的优缺点，可以看出当前的迭代算法有三点共同的不足。

第一，没有在算法的复杂度和性能上取得很好的权衡，总是以牺牲复杂度或者性能来获得另一个指标的提升。

第二，没有避免 Gram 矩阵的计算，只是对求逆部分进行了优化，而大规模 MIMO 系统中计算 Gram 矩阵的复杂度已经超过了对其求逆的复杂度。

第三，算法仅适用于非对称的大规模 MIMO 系统，当其逐渐向对称的大规模 MIMO 系统过渡时，保持相同迭代次数的条件下，性能损失明显。

2.4 本章小结

本章详细的介绍了大规模 MIMO 系统模型以及检测算法的通用数学模型。通过对非线性算法与线性算法的比较，选取了当前主流研究: 基于 MMSE 检测的迭代算法进行详细介绍与分析，最终总结当前迭代算法存在的不足。下一章将据此提出两种新的低复杂度，性能近乎最优的迭代算法。

第三章　基于 Non-Stationary RI 和 Second-Order RI 的检测算法

本章首先引入一种新的基于 Non-Stationary Richardson 的迭代算法，接着根据大规模 MIMO 系统的信道特性，提出三项优化策略。该算法利用这三种优化策略，降低了初始化复杂度，同时避免了 Gram 矩阵及其特征值的计算。考虑到算法在不同的信道模型以及 Gram 矩阵主对角元素占优不明显系统下的普适性，本章提出一种普适于上述条件的新的基于 Second-Order Richardson 的迭代算法。该算法在更实际的信道模型和用户数较多的大规模 MIMO 系统，如 128×32 的 MIMO 系统时，仍能保持近乎最优的性能，而复杂度的提升可忽略不计。

随后，本章基于大规模 MIMO 系统仿真平台，在不同信道模型和天线配置下对现有算法以及本文提出的两种算法的 BER 性能进行评估，给出了各算法的复杂度分析及比较。

3.1　基于 Non-Stationary Richardson 的迭代算法

本小节提出一种新的基于 Non-Stationary Richardson 迭代的检测算法（proposed-1），该算法的基本原理也是通过求解线性方程组来迭代逼近精确解，以此避免 Gram 的精确求逆，可以归类在上文所述的经典迭代法中。相较于传统 Richardson(RI) 算法[5]，该算法采用随迭代次数变化的松弛因子 α_k 代替原来的常数松弛因子 α。松弛因子的表达式如式 (3–1) 所示。

$$\alpha_k = \frac{2}{(\lambda_{\max} + \lambda_{\min}) - (\lambda_{\max} - \lambda_{\min})cos(\frac{(2k-1)\pi}{2K})} k = 1, 2, \cdots, K \tag{3–1}$$

其中，K 为根据天线配置和信道特征选定的总的迭代次数。当 K 确定之后，可以通过上式计算出每一步迭代松弛因子的值。此算法的优点在于可以针对不同的迭代次数而采用变化的松弛因子，以此来获得更快的收敛速度。该算法的完整表达式如式 (3–2) 所示[19]。

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k(b - A\mathbf{x}_k) \tag{3–2}$$

直接引入 Non-Stationary Richardson 算法会存在以下几个问题：

第一，没有避免 Gram 矩阵的计算，计算 $\mathbf{H}^H\mathbf{H}$ 会引入 $O(2BU^2)$ 次乘法操作，这一部分的复杂度已经超过了矩阵直接求逆的复杂度。

第二，Gram 矩阵的最大最小特征值 $\lambda_{\max}$ 和 $\lambda_{\min}$ 如果采用直接计算的方法，一方面意味着 Gram 矩阵的计算不可能避免，另一方面会额外增加较多复杂度。

第三，初始化的赋值，即 $\mathbf{x}_0$ 的取值难以确定。若是采用传统的初始化策略，令 $\mathbf{x}_0 = 0$，则在一定的迭代次数内性能损失较大。

综上所述，算法的直接引入是不可行的，需要采取一定的优化策略，使得基于 Non-Stationary Richardson 的迭代算法能够避免 Gram 矩阵的计算和简化特征值的计算来降低算法的复杂度，然后通过合适的初始化策略来进一步提高算法在固定迭代次数内的性能。

3.2 优化策略

针对上节提到的问题，本小节将详细介绍具体的优化策略。

3.2.1 近似特征值

在大规模 MIMO 系统中，尤其是当 $B \gg U$ 时，Gram 矩阵分布与 Wishart Matrix 分布类似[20]。而对于 Wishart 矩阵 $\mathbf{M}_s = \frac{1}{s}\mathbf{V}_s\mathbf{V}_s$，$\mathbf{V}_s \in \mathbb{R}_{n\times s}$，当其矩阵维度较大时，该矩阵的最大和最小特征值趋向于 $\left(1+\sqrt{n/s}\right)^2$ 和 $\left(1-\sqrt{n/s}\right)^2$。因此，对于 $\mathbf{G} = \mathbf{H}^H\mathbf{H}$ 而言，其中 $\mathbf{H} \in \mathbb{C}_{B\in U}$，令 $\mathbf{H}^H = \mathbf{V}_s, U = s, B = n$，则有

$$\lambda_{\max} = B(1+\sqrt{U/B})^2\,,\ \lambda_{\min} = B(1-\sqrt{U/B})^2 \tag{3–3}$$

考虑到基于 MMSE 检测的算法，其 Gram 矩阵的表示式为 $\mathbf{A} = \mathbf{H}^H\mathbf{H} + \sigma^2$。因此，可以得到 Gram 矩阵的最大和最小特征值分别如式3–4所示。

$$\lambda_{\max} = B(1+\sqrt{U/B})^2 + \sigma^2\,,\ \lambda_{\min} = B(1-\sqrt{U/B})^2 + \sigma^2 \tag{3–4}$$

由此，我们可以简单的通过信道矩阵 $\mathbf{H}$ 的维度和信道噪声计算出所需特征值。由于这三个参数为输入数据，因此特征值能够提前计算，而不必消耗硬件资源。

3.2.2 低复杂度初始化策略

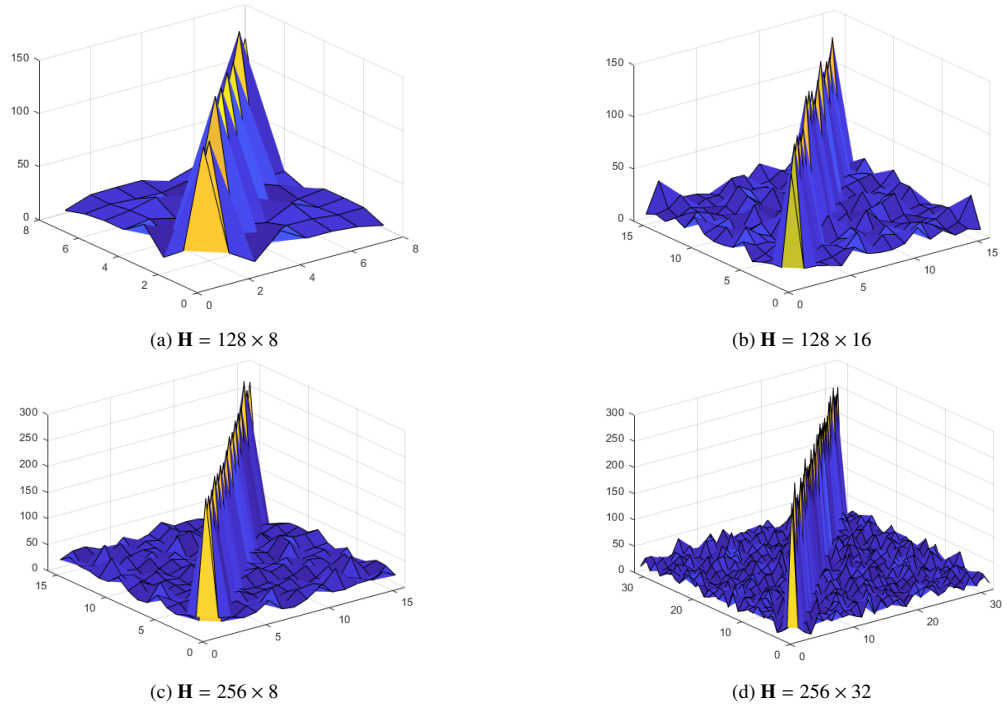
在迭代初始解的选取方面，当前文献主要存在两种方法，一种是无初始化，即直接从零开始迭代。另一种是选取 Neumann 展开的第一项作为初始解。Neumann 展开的形式如式 (3–5) 所示，其中 $\mathbf{D}$ 为 Gram 矩阵 $\mathbf{A}$ 的对角阵。

$$\mathbf{A}^{-1} = \mathbf{D}^{-1} + \Sigma_{n=1}^{\infty}(\mathbf{I}_U - \mathbf{D}^{-1}\mathbf{A})^n\mathbf{D}^{-1} \approx \mathbf{D}^{-1} \tag{3–5}$$

若是将 $\mathbf{D}^{-1}\mathbf{t}$ 作为精确解自然是不可行的，但是考虑到大规模 MIMO 系统中当 $B \gg U$ 时，Gram 矩阵主对角占优的特质，Neumann 展开的第一项远大于剩余项，因此将 $\mathbf{D}^{-1}\mathbf{t}$ 作为初始解是可行的。而且此初始化方法相当于进行一次 NS 迭代，对算法性能的提升十分有利。虽然计算 $\mathbf{D}$ 的实数乘法复杂度 $O(4BU)$ 远小于计算 Gram 矩阵的实数乘法复杂度 $O(2BU^2)$，但是考虑到在得到 $\mathbf{D}$ 之后，计算 $\mathbf{D}^{-1}$ 仍需要 U 次除法，而过多的除法硬件实现不友好。为减少除法次数，需要对该初始化策略做一定的改进。

前文提到在瑞利信道中，当 $B \gg U$ 时有 Gram 矩阵主对角占优的性质，如图2–7。在更为实际的信道 Correlation 信道模型下，相关系数 $\xi = 0.5$ 时，如图3–1所示，Gram 矩阵仍具备主对角占优显著的特质。观察图2–7和图3–1，注意到无论是在瑞利信道还是更为实际的 correlation 信道下 Gram 矩阵的对角线各个元素的值大小接近方差很小，因此利用它们的平均值来表征 $\mathbf{D}$ 是一种很好的近似，如式3–6所示。

$$\mathbf{D}^{-1}\mathbf{t} = \frac{U}{(\Sigma_U diag(\mathbf{D}))}\mathbf{I}_U\mathbf{t} = \eta\mathbf{t} \tag{3–6}$$



(a) $\mathbf{H} = 128 \times 8$　(b) $\mathbf{H} = 128 \times 16$

(c) $\mathbf{H} = 256 \times 8$　(d) $\mathbf{H} = 256 \times 32$

图 3–1 相关系数 ξ=0.5 的 correlation 信道下，不同天线配置的 MIMO 系统 Gram 矩阵分布

但是这样的近似使得 Gram 矩阵的计算仍不可避免，考虑到上小节对特征值的计算，我们可以用 $\lambda_{\max}$ 和 $\lambda_{\min}$ 的平均值来近似 η 的值，得到如下方程。

$$\mathbf{D}^{-1}\mathbf{t} = \eta\mathbf{t} = \frac{1}{\frac{\lambda_{\max}+\lambda_{\min}}{2}}\mathbf{I}_U\mathbf{t} = \frac{1}{B+U+\sigma^2}\mathbf{t} \tag{3–7}$$

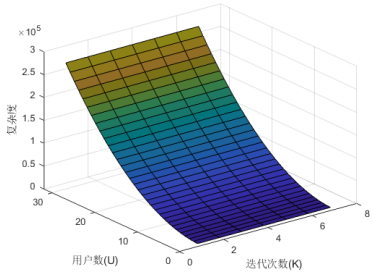
因此，$\mathbf{x}_0$ 的初始赋值的复杂度大大降低，从原来的 4BU 次实数乘法和 U 次除法操作减小到复杂度为 0。另一方面，令 $\mathbf{t} = \mathbf{H}^H\mathbf{y}$ 也就是 MF 检测算法的最终结果能够有效保证初始值接近精确值。

3.2.3　矩阵向量乘

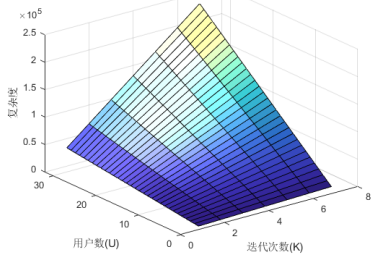
通过上面两小节的优化策略，初始值 $\mathbf{x}_0$ 和特征值的计算都仅仅利用了 Gram 矩阵的性质，避开了直接利用 Gram 矩阵的元素值。接下来本小节将介绍通过分步矩阵向量乘代替大规模的矩阵乘法，如式 (3–8) 所示。

$$\mathbf{A}\mathbf{x}_k = (\mathbf{H}^H\mathbf{H} + \sigma^2\mathbf{I}_U)\mathbf{x}_k = \mathbf{H}^H(\mathbf{H}\mathbf{x}_k) + \sigma^2\mathbf{x}_k \tag{3–8}$$

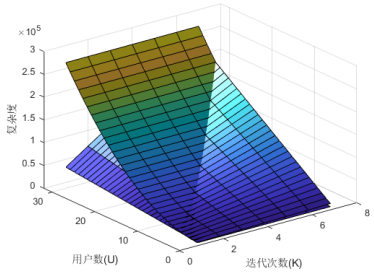
利用该策略，计算 $\mathbf{A}\mathbf{x}_k$ 的复数乘法运算次数可以从 $O(2BU^2 + 4U^2)$ 减小到 $O(8BU + 2U)$，极大的降低了复杂度。设迭代次数为 K，分解前计算 Gram 矩阵和 K 次 Gram 矩阵与向量 $\mathbf{x}_k$ 乘积所需的实数乘法次数为 $2BU^2 + 4KU^2$；分解后计算 K 次两个矩阵向量积和一个向量数乘的组合所需的实数乘法运算次数为 $8KBU + 2KU$。



(a) 分步计算前复杂度



(b) 分步计算后复杂度



(c) 分步计算前与分步计算后复杂度比较图

图 3–2 B=128 时，分步计算前后复杂度随用户数和迭代次数变化图

通常情况下，迭代次数小于 4 次就能获得很好的性能，以及在实际情况中用户数量一般大于 8，因此从图3–2，可以得出分解后的复杂度远低于分解前的复杂度。此外，根据二者的复杂度函数，用户数越多，矩阵向量积分解的优势越明显。

因此，利用矩阵向量积避免对 Gram 矩阵的直接计算，可使现有算法的复杂度降低很多。

3.3 LLR 计算

前文得到的计算结果仅为硬判决符号 $\mathbf{x}_{k+1}$，无法用于采用信道编码的系统，本文采用近似计算 LLR 来得到软比特输出, 如式 (3–9) 所示，其中 $L_{i,b}$ 表示第 i 个用户的第 b 个比特的软比特输出,X_b^0 和 X_b^1 表示星座图符号子集。

$$L_{i,b}(\hat{x}_i) = \rho_i\left(\min_{x\in X_b^0}\left|\frac{\hat{x}_i}{\mu_i} - x\right|^2 - \min_{x\in X_b^1}\left|\frac{\hat{x}_i}{\mu_i} - x\right|^2\right) \tag{3–9}$$

展开精确解 $\hat{\mathbf{x}}$ 的计算公式，如式 (3–10) 所示。

$$\hat{\mathbf{x}} = \mathbf{A}^{-1}\mathbf{H}^H\mathbf{y} = \mathbf{A}^{-1}\mathbf{H}^H\mathbf{H}\mathbf{x} + \mathbf{A}^{-1}\mathbf{H}^H\mathbf{n} \tag{3–10}$$

定义 $\mathbf{E} = \mathbf{A}^{-1}\mathbf{H}^H\mathbf{H}$ 以及 $\mathbf{F} = \mathbf{E}\mathbf{A}^{-1}$。因此，等效矩阵增益可以表示为 $\hat{x}_i = \mu_i x_i + t_i$，其中 $\mu_i = \hat{E}_{i,i}$, 以及噪声项 $t_i^2 = \Sigma_{j\neq i}^U |E_{i,j}|^2 + F_{i,i}\sigma^2 \approx F_{i,i}\sigma^2$[6]。可以将 LLR 计算近似为式3–11。

$$\rho_i = \frac{\mu_i^2}{t_i^2} = \frac{(\hat{E}_{i,i})^2}{\sigma^2 F_{i,i}} = \frac{\hat{E}_{i,i}}{\sigma^2(\hat{A}_{i,i}^{-1})} \approx \frac{A_{i,i}}{\sigma^2} \approx \frac{B+U+\sigma^2}{\sigma^2} \tag{3–11}$$

3.4 基于优化策略的 proposed-1 算法

利用本文优化策略进行改进之后，可以将 Non-Stationary Richardson 迭代算法，以较低的复杂度引入大规模 MIMO 系统检测工作中。

3.4.1 proposed-1 算法流程

完整的算法流程如表3–1所示，为方便起见，后文将称这种算法为 proposed-1 算法。

表 3–1 基于 Non-Stationary Richardson 迭代的检测算法 (proposed-1)

proposed-1: 基于 Non-Stationary Richardson 迭代的检测算法

Input:,$\mathbf{H},\mathbf{y},\sigma^2$,K

Initialization:

1. $\lambda_{\max}=B\left(1+\sqrt{\frac{U}{B}}\right)^2+\sigma^2$, $\lambda_{\min}=B\left(1-\sqrt{\frac{U}{B}}\right)^2+\sigma^2$
2. $\beta=\frac{1}{B+U+\sigma^2},\mathbf{y}^{MF}=\mathbf{H}^H\mathbf{y}$
3. $\mathbf{x}_1=\beta\mathbf{y}_{MF}$

iterations: Non-Stationary Richardson iteration

for k = 1,2,$\cdots$,K

4. $\alpha_k=\frac{2}{(\lambda_{\max}+\lambda_{\min})-(\lambda_{\max}-\lambda_{\min})cos(\frac{(2k-1)\pi}{2K})}$
5. $\tilde{\mathbf{h}}=\mathbf{H}\mathbf{x}_k$
6. $\mathbf{x}_{k+1}=\mathbf{x}_k+\alpha_k(\mathbf{y}^{MF}-\mathbf{H}^H\tilde{\mathbf{h}}-\sigma^2\mathbf{x}_k)$

end for

output LLR computation

7. $\rho_i\leftarrow(B+U+\sigma^2)/\sigma^2$
8. $L_{i,b}(\hat{x}_i)=\rho_i\left(\min_{x\in X_b^0}|\frac{\hat{x}_i}{\mu_i}-x|^2-\min_{x\in X_b^1}|\frac{\hat{x}_i}{\mu_i}-x|^2\right)$

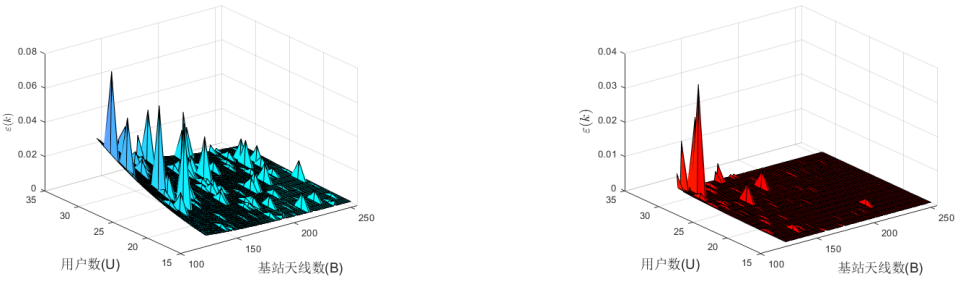
3.4.2 proposed-1 算法收敛性分析

本节对提出的 proposed-1 算法的收敛性进行分析，并证明该算法收敛。定义误差项为当前迭代结果 $\mathbf{x}_k$ 与精确解 $\hat{x}=\mathbf{A}^{-1}\mathbf{y}^{MF}$ 的差值，可以表示为

$$\mathbf{e}_k=\mathbf{x}_k-\hat{\mathbf{x}} \tag{3–12}$$

算法的收敛性可以通过 $\mathbf{e}_k$ 的收敛性来确认。因为此处仅为验证算法的收敛性，因此不考虑复杂度，$\mathbf{x}_{k+1}$ 表达式简写为式 (3–13)。

$$\mathbf{x}_{k+1}=\mathbf{x}_k+\alpha_k(\mathbf{y}^{MF}-\mathbf{A}\mathbf{x}_k) \tag{3–13}$$



(a) K=6　　(b) K=11

图 3–3 第 k 次迭代误差 $\varepsilon(k)$ 的值随用户数 (U) 和基站天线数 (B) 变化趋势图

取 $\mathbf{e}_k$ 的二范数 $\|\mathbf{e}_k\|_2$，由式 (3–13) 推导得到式 (3–14)。

$$\begin{aligned}\mathbf{e}_k &= \mathbf{x}_k - \hat{\mathbf{x}} = \mathbf{x}_{k-1} + \alpha_k(\mathbf{y}^{MF} - \mathbf{A}\mathbf{x}_{k-1}) - \hat{\mathbf{x}} \\ &= \mathbf{e}_{k-1} + \alpha_k\mathbf{A}(\mathbf{A}^{-1}\mathbf{y}^{MF} - \mathbf{x}_{k-1}) = (1-\alpha_k\mathbf{A})\mathbf{e}_k \\ &= \prod_{k=1}^{\infty}(1-\alpha_k\mathbf{A})^k\mathbf{e}_0 = \varepsilon(k)\mathbf{e}_0\end{aligned} \tag{3–14}$$

图3–3给出了当总迭代次数 K=6 和 K=11 时，$\prod_{k=1}^{\infty}(1-\alpha_k\mathbf{A})^k$ 的值随用户数 U 和基站天线数 B 变化趋势图。从图3–3可以看出, 随着迭代次数的增多，$\varepsilon(k)\to 0$，由于 $\mathbf{e}_0$ 为定值，因此随着 $k\to\infty$，有 $\mathbf{e}_k\to 0$，proposed-1 算法收敛性得证。

$$k\to\infty \quad\Rightarrow\quad \mathbf{e}_k\to 0 \quad\Rightarrow\quad \text{proposed-1 算法是收敛的} \tag{3–15}$$

3.4.3 proposed-1 算法的优势与不足

图3–4对比了不同算法的复杂度和 BER，可以看出通过一系列的优化策略，proposed-1 算法存在以下优势：

第一，降低了硬件实现复杂度。通过避免 Gram 矩阵的计算，以及简化初始值和特征值的计算，相较于现有算法 proposed-1 算法的复杂度极大的降低，在 1 次迭代时相较于其他算法复杂度降低了 67.5%，2 次迭代时降低了 46.1%，3 次迭代时降低了 25.6%，同时略低于 PCI 算法的复杂度。

第二，在相同迭代次数下，提高了算法的 BER 性能。初始化策略令 proposed-1 算法在第一次迭代就能获得较好的性能；在 128×16 的条件下，迭代次数为 3 次时，proposed-1 算法的 BER 性能已接近直接求逆的 MMSE 算法检测性能。

第三，迭代形式简单，仅需存储前一次迭代的结果，降低数据间依赖性，有利于硬件实现。

算法的不足之一为总迭代次数需要提前确定才能够计算出每次迭代的参数 α_k。如图3–5所示，在总的迭代次数不同的情况下，每一次迭代的参数是不一样的。以第一次迭代为例，

- 若是总迭代 1 次，那么有 $\alpha_1(K=1)=\frac{2}{\lambda_{\max}+\lambda_{\min}}$；
- 若是总迭代 2 次，那么有 $\alpha_1(K=2)=\frac{2}{\lambda_{\max}+\lambda_{\min}+(\lambda_{\max}-\lambda_{\min})\frac{\sqrt{2}}{2}}=\frac{4}{(2-\sqrt{2})\lambda_{\max}+(2+\sqrt{2})\lambda_{\min}}\neq\alpha_1(K=1)$

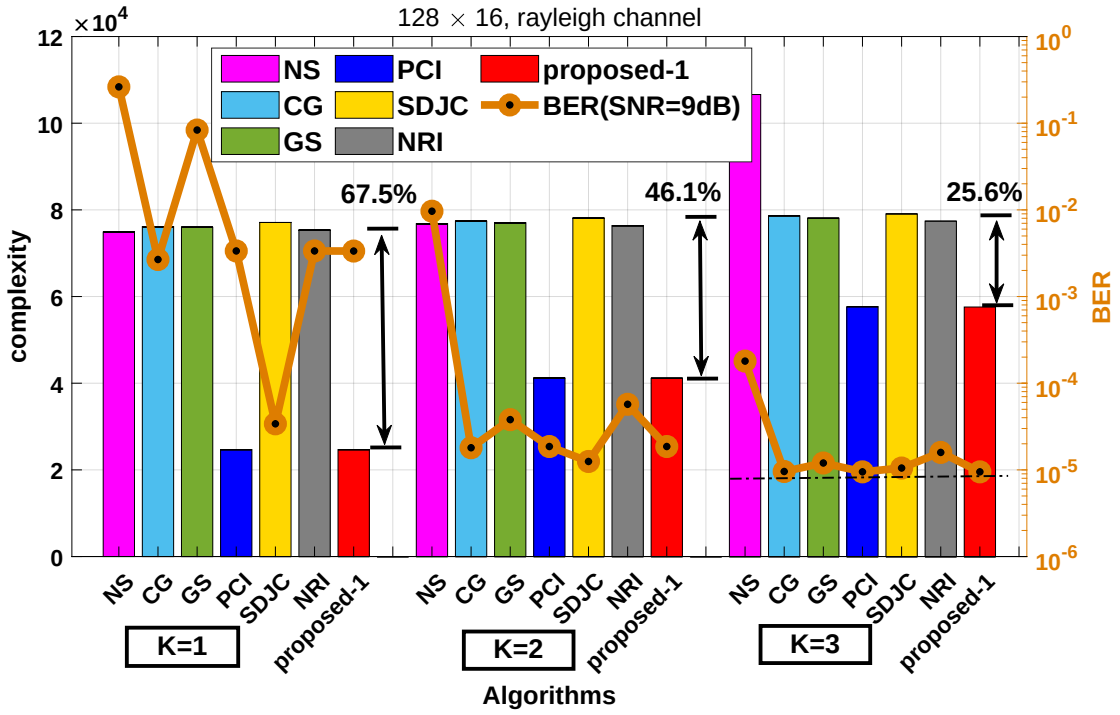


图 3–4 迭代次数 K=1~3, 不同算法复杂度和 BER 性能联合比较图，其中 BER 取 SNR=9dB 条件下的值

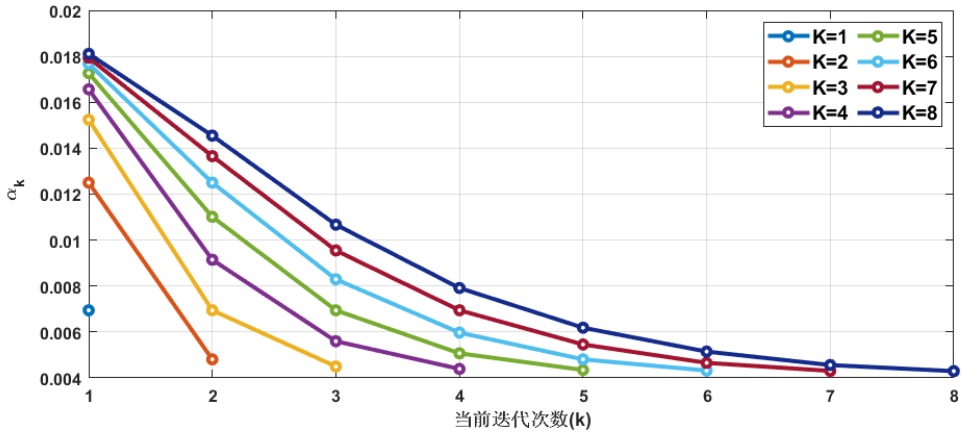


图 3–5 第 k 次迭代 α_k 的值与总迭代次数 K 关系图，128×16 系统

这意味着对于任意一个信道模型，当设定的迭代次数无法满足性能要求时，不能简单的通过增加一次迭代来提高 BER 性能，而是需要通过重新从 initialization 开始新的迭代，这无疑会降低检测的效率，增加检测的难度。proposed-1 算法另外一个不足来自于它不能适应多种信道模型。如前文提到的 correlation 信道模型，相关系数较大时，前文提到的算法包括 proposed-1 算法性能受损增大，而迭代次数的增加会导致复杂度的进一步提升。

因此，需要一个能够适用于更为实际模型且可随意增加迭代次数，同时仍能保持低复杂度高性能的检测算法。

3.5 基于优化策略和 Second-Order Richardson 迭代的 proposed-2 算法

本节借鉴 proposed-1 算法变化参数的思路，提出了采用3.2小节优化策略的二阶 Richardson 迭代算法，并给出该算法的完整流程和收敛性分析。为方便起见后文将称此种算法为 proposed-2 算法。

3.5.1 proposed-2 算法流程

proposed-2 算法迭代属于经典迭代法的范畴，二阶指的是每一次迭代结果的计算需要利用前两次的迭代结果，由此充分利用迭代结果间的相互关系。其迭代的基本形式为

$$\mathbf{x}_{k+1} = \mathbf{x}_{k-1} + \gamma_{opt}\alpha_{opt}(\mathbf{y}^{MF} - \mathbf{A}\mathbf{x}_k) + \alpha_{opt}(\mathbf{x}_k - \mathbf{x}_{k-1}) \tag{3–16}$$

其中 $\alpha_{opt} = \frac{2}{\lambda_{\max}+\lambda_{\min}}$，$\gamma_{opt}$ 为矩阵 $\mathbf{C} = \mathbf{I}_U - \alpha_{opt}\mathbf{A}$ 谱半径的函数，如式 (3–17) 所示[21]。

$$\begin{aligned} \varrho(\mathbf{C}) &= -\alpha_{opt}\lambda_{\min} + 1 = \frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}} \\ \gamma_{opt} &= \frac{2}{1 + \sqrt{1 - \varrho^2(\mathbf{C})}} \end{aligned} \tag{3–17}$$

3.2小节提出的优化策略包括近似特征值计算和用矩阵向量乘来避免计算 Gram 矩阵仍能应用于 proposed-2 算法。由于二阶算法需要借用前两次迭代的结果，这意味着初始化需要对 $\mathbf{x}_0$ 和 $\mathbf{x}_1$ 赋值。考虑到 Neumann 级数展开的第二项 $(\mathbf{I}_U - \mathbf{D}^{-1}\mathbf{A})\mathbf{D}^{-1}$ 计算复杂度高，难以作为初始值，因此采用 $\mathbf{0}$ 作为其中一个初始值避免复杂度的提升，可以用式 (3–18) 表示, 其中，$\mathbf{y}^{MF} = \mathbf{H}^H\mathbf{y}$。此外 proposed-2 算法的软比特输出同 proposed-1 算法，不予赘述。

$$\mathbf{x}_0 = \mathbf{0}; \qquad \mathbf{x}_1 = \frac{2}{\lambda_{\max} + \lambda_{\min}}\mathbf{y}^{MF} \tag{3–18}$$

完整的 proposed-2 算法流程如表3–2所示。

表 3–2 基于 Second-Order Richardson 迭代的检测算法 (proposed-2)

proposed-2: 基于 Second-Order Richardson 迭代的检测算法

Input: $\mathbf{H}$, $\mathbf{y}$, σ^2, K

Initialization:

1. $\lambda_{\max} = B\left(1+\sqrt{\frac{U}{B}}\right)^2 + \sigma^2$, $\lambda_{\min} = B\left(1-\sqrt{\frac{U}{B}}\right)^2 + \sigma^2$
2. $\alpha_{opt} = \frac{2}{\lambda_{\max}+\lambda_{\min}}$
3. $\varrho(\mathbf{C}) = -\alpha_{opt}\lambda_{\min} + 1 = \frac{\lambda_{\max}-\lambda_{\min}}{\lambda_{\max}+\lambda_{\min}}$, $\gamma_{opt} = \frac{2}{1+\sqrt{1-\varrho^2(\mathbf{C})}}$
4. $\mathbf{y}^{MF} = \mathbf{H}^H\mathbf{y}$, $\mathbf{x}_0 = \mathbf{0}$, $\mathbf{x}_1 = \frac{2}{\lambda_{\max}+\lambda_{\min}}\mathbf{y}^{MF}$

iteration:second order Richardson iteraion

for k = 1 : K do

5. $\tilde{\mathbf{h}} = \mathbf{H}\mathbf{x}_k$
6. $\mathbf{x}_{k+1} = \mathbf{x}_{k-1} + \gamma_{opt}\alpha_{opt}(\mathbf{y}^{MF} - \mathbf{H}^H\tilde{\mathbf{h}} - \sigma^2\mathbf{x}_k) + \gamma_{opt}(\mathbf{x}_k - \mathbf{x}_{k-1})$

end for

output LLR computation

6. $\rho_i \leftarrow (B + U + \sigma^2)/\sigma^2$
7. $L_{i,b}(\hat{x}_i) = \rho_i\left(\min_{x\in X_b^0}|\frac{\hat{x}_i}{\mu_i} - x|^2 - \min_{x\in X_b^1}|\frac{\hat{x}_i}{\mu_i} - x|^2\right)$

3.5.2 proposed-2 算法收敛速率比较

本小节首先证明 proposed-2 算法是收敛的，然后计算 proposed-2 算法的收敛速率并与现有算法进行比较。

误差项 $\mathbf{e}_k$ 的定义同式 (3–12) 一致，有

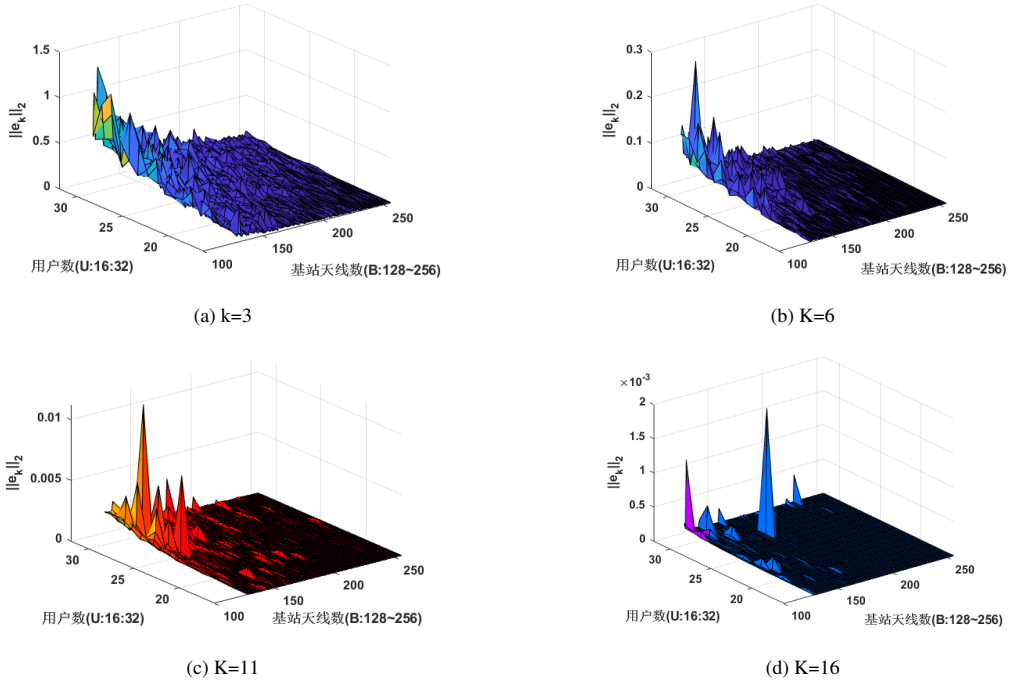
$$\mathbf{e}_k = \mathbf{x}_k - \hat{x} = \mathbf{x}_k - \mathbf{A}^{-1}\mathbf{y}^{MF} \tag{3–19}$$

图3–6表示在迭代次数 K 分别为 3，6，11，16 时，$\mathbf{e}_k$ 的值随基站天线数 (B) 和用户数 (U) 关系，从图中可以看出 $\mathbf{e}_k$ 的值随着 K 的增大逐渐减小，由此可以判定 propsoed-2 算法是收敛的。

由式 (3–19) 进一步推导得式 (3–20)。

$$\begin{aligned}\mathbf{e}_{k+1} &= \mathbf{x}_{k+1} + \gamma_{opt}\alpha_{opt}(\mathbf{y}_{MF} - \mathbf{A}\mathbf{x}_k) + \gamma_{opt}(\mathbf{x}_k - \mathbf{x}_{k-1}) - \mathbf{A}^{-1}\mathbf{y}_{MF}\\ &= \mathbf{e}_{k-1} - \gamma_{opt}\alpha_{opt}\mathbf{A}\mathbf{e}_k + \gamma_{opt}(\mathbf{e}_k - \mathbf{e}_{k-1})\\ &= \gamma_{opt}(\mathbf{I}_U - \alpha_{opt}\mathbf{A})\mathbf{e}_k + (1-\gamma_{opt})\mathbf{e}_{k-1}\\ &= \gamma_{opt}\mathbf{C}\mathbf{e}_k + (1-\gamma_{opt})\mathbf{e}_{k-1}\end{aligned} \tag{3–20}$$

数学上通常用谱半径来衡量收敛速率，收敛半径越小，收敛速率越快。proposed-2 算法误



(a) k=3　(b) K=6

(c) K=11　(d) K=16

图 3–6 迭代次数 K=3，6，11，16 条件下，$\|\mathbf{e}_k\|_2$ 的值随基站天线数 (B) 和用户数 (U) 关系图

差的迭代矩阵 T_{pro} 和 PCI[8] 算法误差的迭代矩阵如式3–21所示。

$$\begin{aligned}\begin{pmatrix}\mathbf{e}_k\\ \mathbf{e}_{k+1}\end{pmatrix}&=\begin{pmatrix}\mathbf{0} & \mathbf{I}_U\\ (1-\gamma_{opt})\mathbf{I}_U & \gamma_{opt}\mathbf{C}\end{pmatrix}\begin{pmatrix}\mathbf{e}_{k-1}\\ \mathbf{e}_k\end{pmatrix}=\mathbf{T}_{pro}\begin{pmatrix}\mathbf{e}_{k-1}\\ \mathbf{e}_k\end{pmatrix}\\ \begin{pmatrix}\mathbf{e}_k\\ \mathbf{e}_{k+1}\end{pmatrix}&=\begin{pmatrix}\mathbf{0} & \mathbf{I}_U\\ (1-\gamma_k)\mathbf{I}_U & \gamma_k\mathbf{C}\end{pmatrix}\begin{pmatrix}\mathbf{e}_{k-1}\\ \mathbf{e}_k\end{pmatrix}=\mathbf{T}_{pci}\begin{pmatrix}\mathbf{e}_{k-1}\\ \mathbf{e}_k\end{pmatrix}\end{aligned}\tag{3–21}$$

其中 γ_k 为 PCI 算法依据迭代次数变化的参数。图3–7展示了 128 × 32MIMO 系统下，proposed-2 算法和 PCI 算法在不同迭代次数下谱半径变化比较图，可以看出当迭代次数 ≤4 时，PCI 算法的谱半径总是小于等于 proposed-2 算法，尤其是第一次迭代时差距最为明显，这意味着 proposed-2 算法拥有比 PCI 算法更快的收敛速率。文献[8] 中证明了 PCI 算法的收敛速率高于 NS 算法和 CG 算法。因此算法收敛速率存在如下关系：

$$\textbf{convergence rate:}\quad \text{proposed-2} \geq \text{PCI} > \text{CG} > \text{NS}\tag{3–22}$$

3.6 算法的 BER 性能与复杂度比较

本节首先介绍基于 MATLAB 的大规模 MIMO 系统仿真平台，并将现有算法与本文提出的两种算法的 BER 性能进行仿真与对比，接着评估本文提出的两种算法复杂度，并与现有算法的复杂度进行对比。

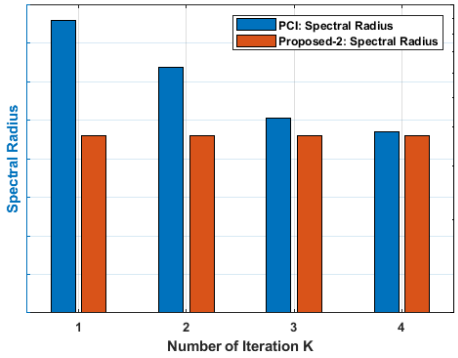


图 3–7 proposed-2 算法和 PCI 算法在不同迭代次数下谱半径变化比较，128 × 32 系统

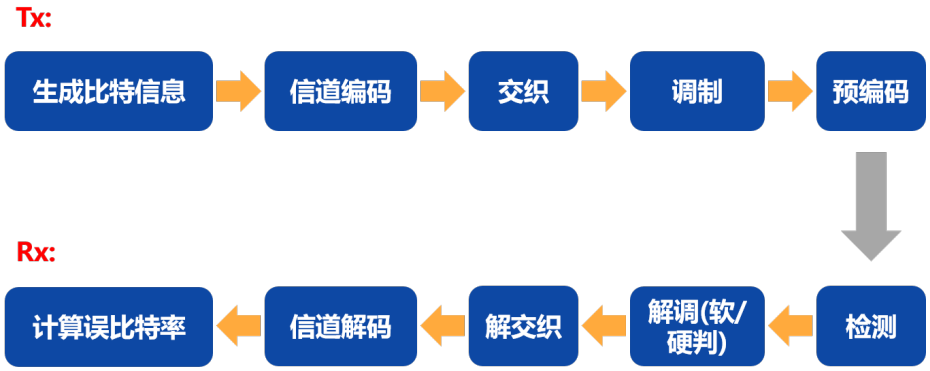


图 3–8 大规模 MIMO 系统仿真平台结构

3.6.1 大规模 MIMO 系统仿真平台

为了对算法的性能进行评估，本文搭建了基于 MATLAB 的大规模系统仿真平台，对 proposed-1 算法，proposed-2 算法并选取典型算法仿真得出 BER 性能进行对比。仿真平台包括了大规模 MIMO 系统通信链路的必要的处理过程，如图3–8所示。

仿真平台采用了 64QAM 调制方案，1/2 编码率以及 $[133_o, 171_o]$ 卷积编码。

3.6.2 瑞利信道下，不同算法 BER 性能对比

(1) 本文提出的算法与传统 RI 算法 BER 性能对比

图3–9展示了 proposed-1 算法，propsoed-2 算法和传统 RI 算法在 128 × 16 大规模 MIMO 系统下，迭代次数为 1,2,3 时的 BER 性能对比。

分析图3–9可知，proposed-1 算法采用变化的松弛因子，proposed-2 算法利用前两次迭代的结果，相比采用固定的松弛因子且仅利用前 1 次迭代结果的传统 RI[5] 方法，性能得到了较大的提升。例如，为了获得 BER=10^{-5} 的性能，proposed-1 算法和 proposed-2 算法迭代两次相比于 RI 算法迭代三次有 0.9dB 的性能提升。

而对比 proposed-1 算法和 proposed-2 算法，可以看到在第一次迭代时，propsoed-2 算法的

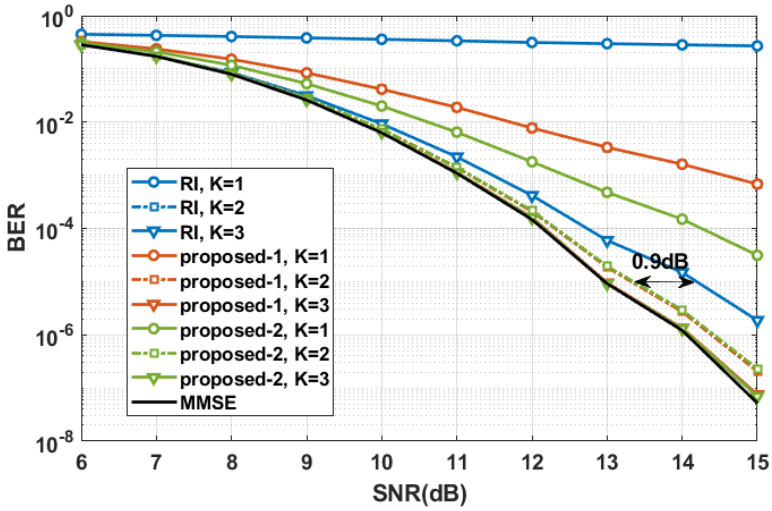


图 3–9 本文提出的算法与传统 RI 算法 BER 性能对比,128 × 16 系统, K=1~3

性能要远远好于 proposed-1 算法，在两次迭代和三次迭代时两者性能接近，尤其是三次迭代的结果与 MMSE 的结果几乎一致，得到了近乎最优的性能。

(2) 不同算法的 BER 性能对比

本节选取了三种维度的大规模 MIMO 系统来表征不同天线配置对算法性能的影响。其中，B × U = 128 × 8 代表 Gram 矩阵主对角占优显著的情况，B × U = 128 × 16 代表典型的主对角占优情况，而 B × U = 128 × 32 则代表了 Gram 矩阵主对角占优不显著的情况。

基于第二章对现有算法的分类与分析，本文选取了各类方法中的典型算法进行仿真得到其 BER 性能。选取的方法分别为多项式展开法的 NS 算法[4]；经典迭代法的 GS 算法和 PCI 算法[6, 8]；梯度搜索法的 CG 算法[10]；混合迭代法的 SDJC 算法和 NRI 算法[11, 12]。

图3–10展示了 128 × 8 大规模 MIMO 系统下，不同算法在迭代次数 K=1 时 BER 性能对比。可以看出，proposed-1 算法与 NRI 算法，SDJC 算法以及 PCI 算法的 BER 性能几乎一致；propsoed-2 算法的性能要明显优于其他算法，例如，在 BER = 10^{-4} 时，proposed-2 算法相较于 proposed-1 算法，NRI 算法，SDJC 算法以及 PCI 算法有 0.5dB 的性能提升。

图3–11展示了 128 × 8 大规模 MIMO 系统下，不同算法在迭代次数 K=2 时 BER 性能对比。与一次迭代相比，所有算法进一步提升，除了 SDJC 算法，NS 算法和 CG 算法外其余算法均得到了和 MMSE 相同的 BER 性能。进一步的，在相同的性能下，propsoed-1 算法复杂度低于其余算法。

图3–12展示了 128 × 16 大规模 MIMO 系统下，不同算法在迭代次数分别为一次和两次时的 BER 性能对比。

- 当 K=1 时，SDJC 算法的性能优于其余算法，这是由于其第一次迭代混合了梯度搜索 SD 算法的结果和 Jacobi 迭代算法，因此第一次迭代收敛速率高。然而，尽管第一次迭代复用了梯度值和 SD 算法迭代结果，复杂度并没有相应的降低。SDJC 算法迭代一次

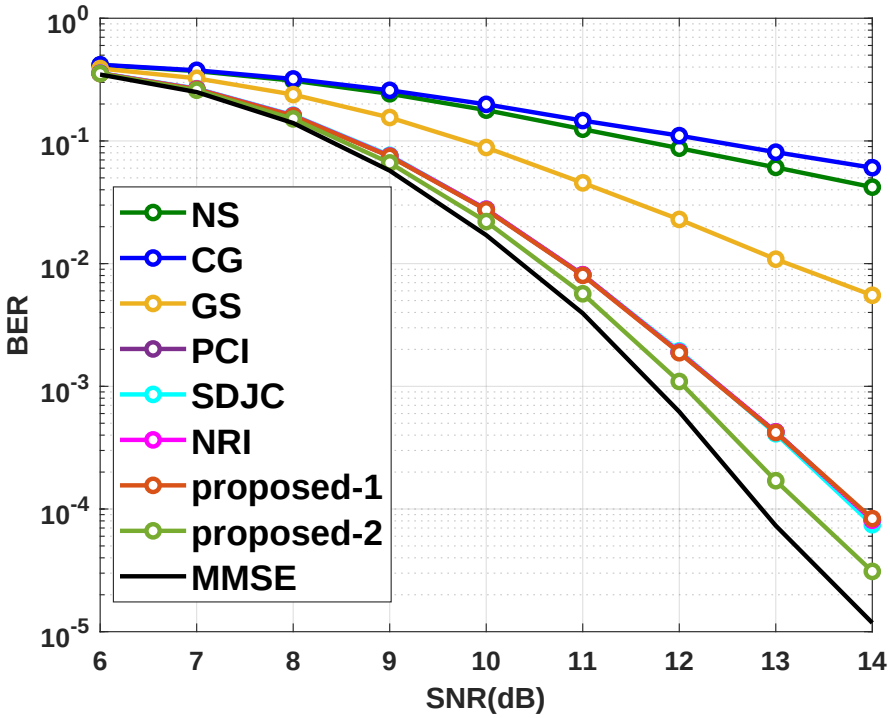


图 3–10 瑞利信道下，不同算法在迭代次数 K=1 时 BER 性能对比，128 × 8 MIMO 系统

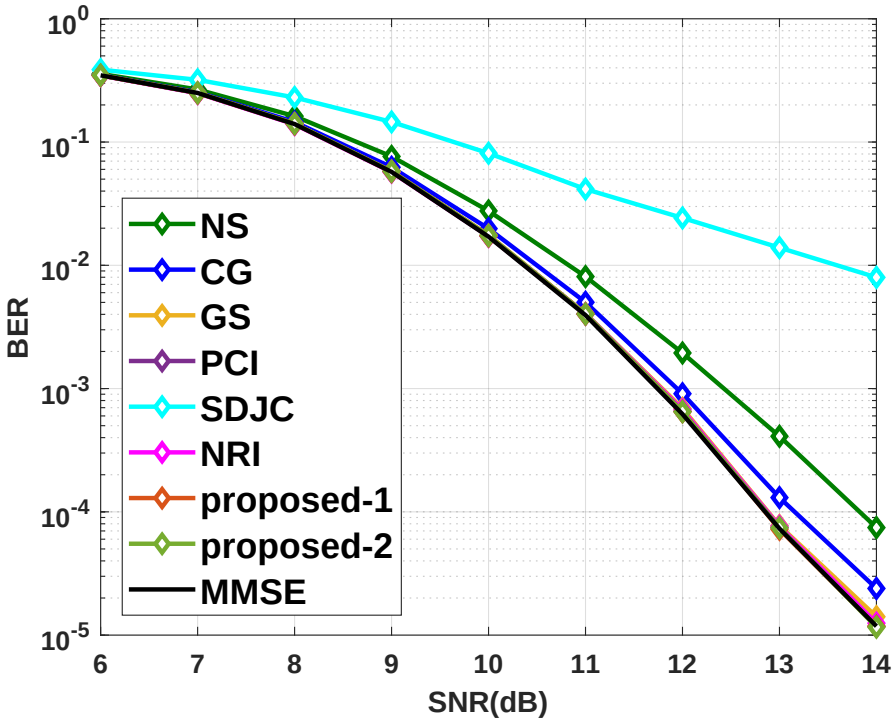


图 3–11 瑞利信道下，不同算法在迭代次数 K=2 时 BER 性能对比，128 × 8 MIMO 系统

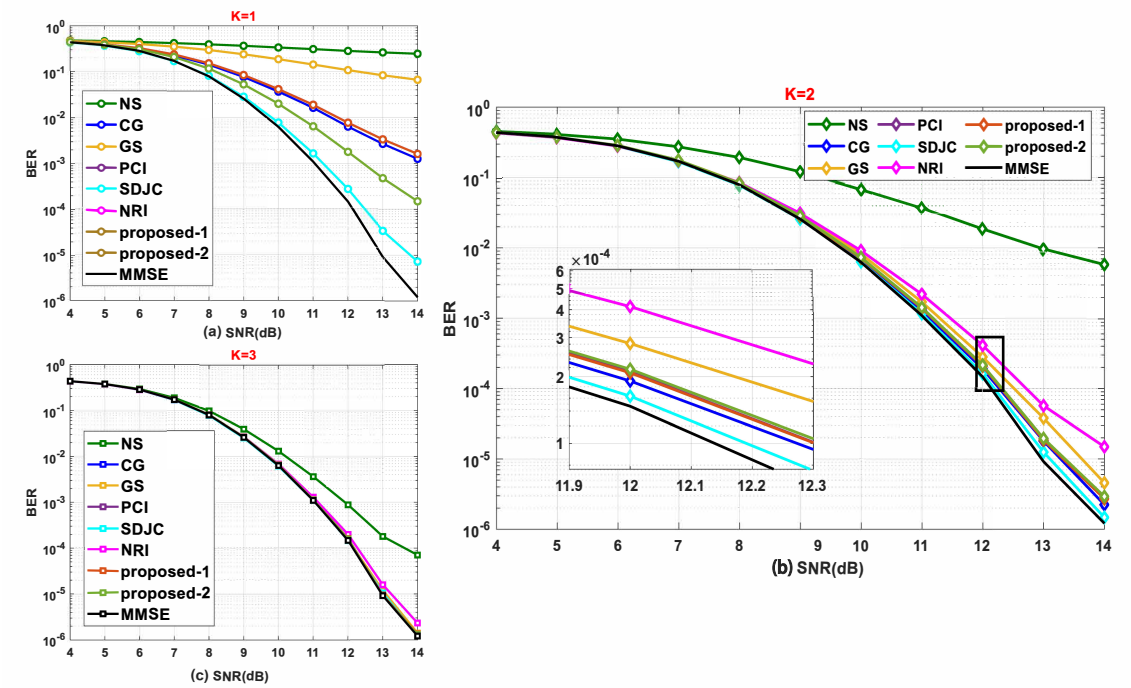


图 3–12 瑞利信道下，不同算法在迭代次数 K=1~3 时 BER 性能对比，128 × 16MIMO 系统

所需的复杂度为 $O(2BU^2 + 4BU + 8U^2 + 10U)$ 同时还有 U+1 次除法运算，而本文提出的 proposed-1 算法两次迭代的复杂度仅需 $O(20BU + 10U)$，小于 SDJC 算法一次迭代的复杂度。proposed-2 算法的性能优于其余算法，同样这得益于其较快的收敛速率。

- 当 K=2 时，本文提出的两种算法已达到接近 MMSE 的 BER 性能。CG 算法的第一次迭代退化为 SD 算法，第二次迭代时，利用更高效的搜索方向，以两次除法运算复杂度为代价，获得了较大的性能提升。而本文提出的 propsoed-1 算法在二次迭代时，具有与 CG 算法的几乎一致的收敛速度，例如在 BER = 10^{-3}，K=1 到 K=2，CG 算法有 2.9dB 的性能提升。SDJC 算法尽管仍能获得优于其余算法的性能，Jacobi 算法收敛速度慢的特性已略见一瞥。
- 当 K=3 时，除 NS 算法外，其余算法均获得和 MMSE 一致的 BER 性能。同样的，proposed-1 算法具有最小复杂度，proposed-2 算法和 PCI 算法略高于 proposed-1 算法，但仍低于其余算法。

图3–13展示了在 128 × 32 大规模 MIMO 系统下，不同算法的 BER 性能对比，虚线表示迭代一次，实线表示迭代 4 次。SDJC 算法迭代 4 次，BER 性能没有提高，这意味着在大规模 MIMO 系统向对称天线配置过渡时 SDJC 算法不能收敛，这是因为最速下降 SD 算法和 Jacobi 迭代结合后不能适应主对角元素占优不明显的情况。相同迭代次数下，propsoed-2 算法具有相较于图中其余算法更优的 BER 性能，尤其是迭代一次时。proposed-2 算法每一次迭代需要前两次迭代的结果，因此对于不同规模的大规模 MIMO 系统具有更强的适应性。

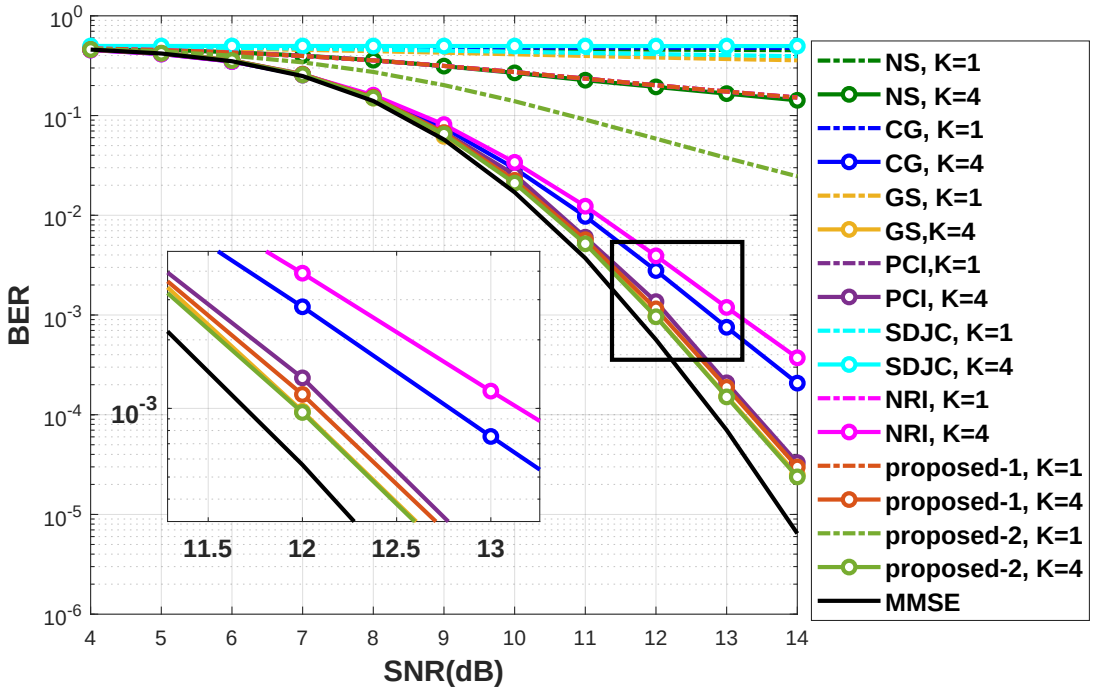


图 3–13 瑞利信道下，不同算法在迭代次数 K=1，4 时 BER 性能对比，128 × 32MIMO 系统

3.6.3 correlation 信道下，不同算法 BER 性能对比

图3–14展示了在不同规模的 MIMO 系统下，不同算法迭代一次时在 correlation 信道模型下的性能比较。对比三幅图可以发现，随着用户数的增加，图中所有算法包括 MMSE 算法，性能均有所下降。其中，NS 算法和 CG 算法在迭代次数小于等于 3 时，不再收敛。本文提出的两种算法收敛速度明显快于 GS 算法，SDJC 算法，值得指出的是，proposed-2 算法在迭代一次时收敛速度还远快于 PCI 算法，NRI 算法和 proposed-1 算法。以 128 × 16 大规模 MIMO 系统为例，在 BER 为 10^{-2} 时，proposed-2 算法相较于 PCI 算法，NRI 算法和 proposed-1 算法有 2.6dB 的性能提升。

图3–15展示了相关系数为 0.3 的 correlation 信道，128 × 32MIMO 系统下不同算法迭代 4 次的 BER 性能图。可以看到，即使在 Gram 矩阵主对角元素占优不明显的 MIMO 系统下，仅需增加一次迭代便可以达到近乎最优的性能。其中 proposed-2 算法和 proposed-1 算法的性能明显优于 NRI 算法和 SDJC 算法，尤其是 SDJC 算法和 NS 算法失去了收敛性。其中，proposed-2 算法相较于 CG 算法，PCI 算法和 proposed-1 算法性能有一定的提升。

图3–16展示了在 128 × 16 大规模 MIMO 系统，correlation 信道，相关系数 $\xi = 0.3$，不同算法迭代 2 次和 3 次时 BER 性能比较，虚线代表迭代两次，实线代表迭代三次。观察图中红圈位置，本文所提的两种算法迭代 2 次的结果与 CG 算法迭代 3 次的性能相当，明显优于 NRI 算法和 NS 算法迭代 3 次的性能，这意味着本文提出的算法可以在不牺牲性能的前提下，减少 1 次迭代所需的复杂度。此外，proposed-1 算法，proposed-2 算法，PCI 算法和 GS 算法迭代 3 次时获得了近乎最优的性能，与 MMSE 相比，在 BER 为 10^{-5} 处，仅有 0.01dB 的性能损失。

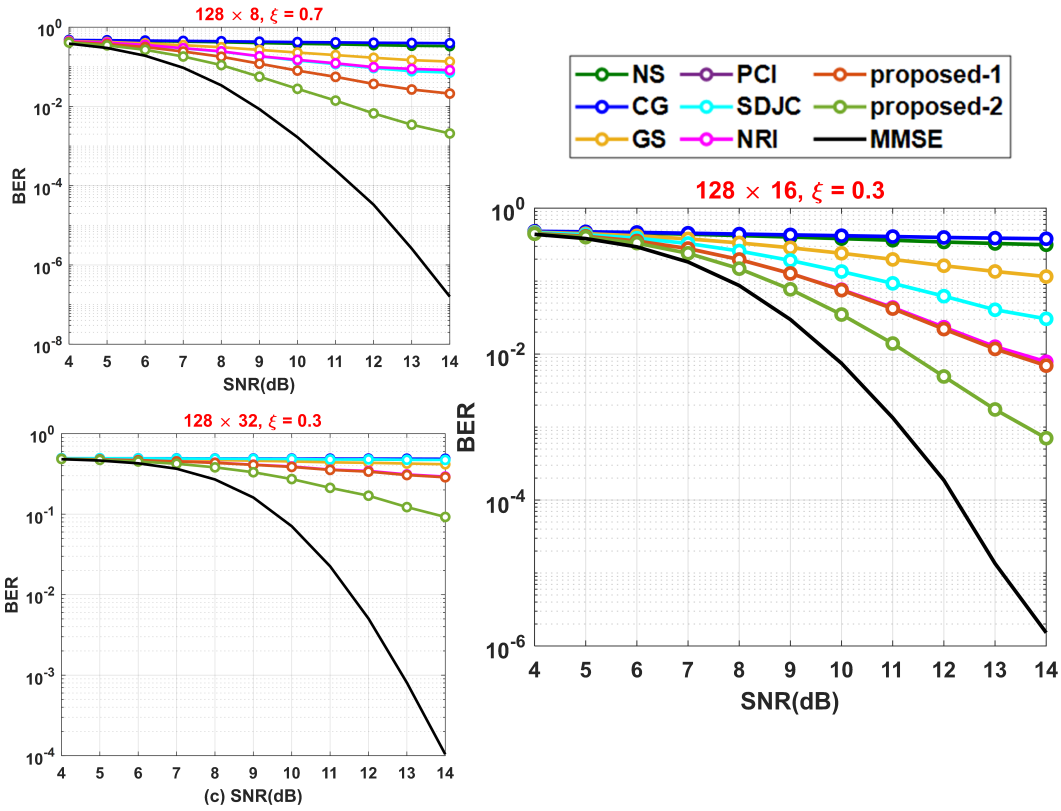


图 3–14 相关系数 $\xi = 0.3$ 的 correlation 信道下，不同算法在迭代次数 K=1 时在不同规模的 MIMO 系统下的 BER 性能比较

3.6.4 算法复杂度分析

在评估大规模 MIMO 算法检测算法复杂度时，通常以实数乘法和除法的次数作为评价指标，因为这两种计算在硬件实现中消耗的时间和资源都远超加法和减法，通过二者的运算次数可以很好的反映算法的复杂度。由于除法操作的硬件实现难度高于乘法，因此应尽量避免除法运算。

由于 proposed-1 和 proposed-2 算法的复杂度构成基本一致，因此不再单独分析。算法的计算过程可以分为两个部分：初始化，K 次迭代，每次迭代中主要包括 $\tilde{\mathbf{h}} = \mathbf{H}\mathbf{x}_k$ 和 $\mathbf{H}^H\tilde{\mathbf{h}}$ 的计算。分别计算其实数乘法运算次数，得到如表3–3所示的结果，其中 K 为迭代的次数。

各算法的优化策略不同，其复杂度构成大致可分为以下几个部分：计算 Gram 矩阵，对角阵 $\mathbf{D}$ 求逆，迭代过程。其中 Gram 矩阵 $\mathbf{A} = \mathbf{H}^H\mathbf{H} + \sigma^2\mathbf{I}_U$ 的元素关于主对角线对称，因此仅需计算出对角阵及其上三角阵元素中即可，需要实数乘法 $2BU^2$；对角阵求逆可以对每一个元素求倒数来获得，需要 U 次除法运算；迭代过程中复杂度主要来源于计算 $\mathbf{A}\mathbf{x}_k$，需要 $4U^2$ 次乘法运算，涉及求解梯度的算法包括 CG 算法和 SDJC 算法会额外增加除法复杂度。

各算法迭代复杂度表达式如表3–4所示。本文提出的两种算法具有明显的复杂度优势，且随着用户数量的增加，这种优势愈加明显。而在性能相近的算法中，本文提出的 proposed-1 算法需要的实数乘法次数最少，且不需要除法次数。

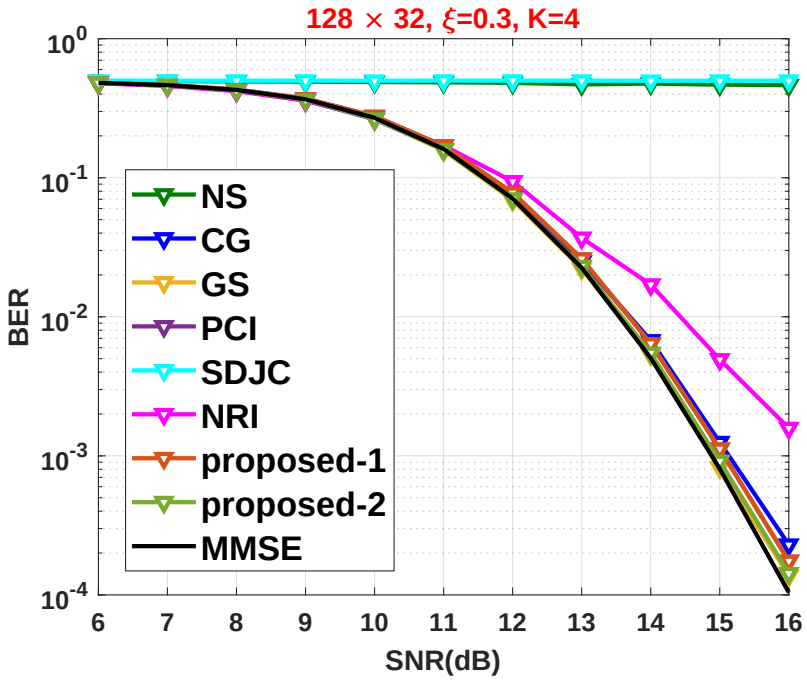


图 3–15 相关系数 $\xi = 0.3$ 的 correlation 信道下，不同算法迭代次数 K=4 时 BER 性能比较，128 × 32MIMO 系统

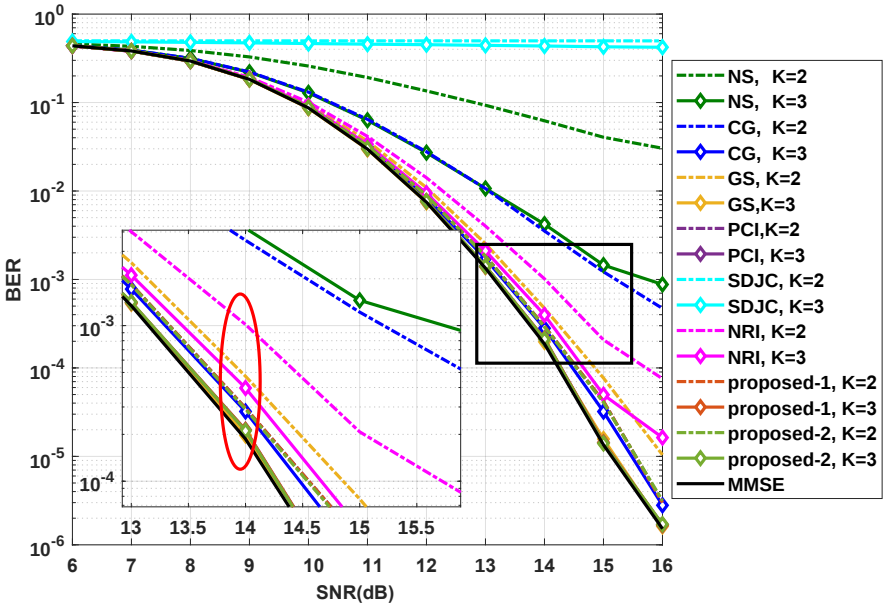


图 3–16 相关系数 $\xi = 0.3$ 的 correlation 信道下，不同算法迭代次数 K=2，3 时 BER 性能比较，128 × 16MIMO 系统

表 3–3 proposed-1 算法和 proposed-2 算法复杂度分析

步骤	中间结果	proposed-1 实数乘法次数	proposed-2 实数乘法次数
初始化	$\mathbf{y}_{MF}$	$4BU$	$4BU$
	$\mathbf{x}_1$	$2U$	$2U$
K 次迭代	$\hat{\mathbf{h}}$	$4KBU$	$4KBU$
	$\mathbf{x}_{k+1}$	$K(4BU+2U+2U)$	$K(4BU+2U+2U+2U)$
总计	-	$(8K+4)BU+(4K+2)U$	$(8K+4)BU+(6K+2)U$

表 3–4 各算法迭代复杂度表达式

算法	实数乘法次数	除法次数
NS[4]	$2BU^2+4BU\ (K=1)$	U
	$2BU^2+4BU+4U^2-4U\ (K=2)$	U
	$2BU^2+4BU+8(K-2)U^3+(20-8K)U+(2K-8)U$	U
RI[5]	$2BU^2+4BU+4KU^2+(2K+8)U$	0
CG[10]	$2BU^2+4BU+(4K+1)U^2+(14K+8)U$	2K
GS[6]	$2BU^2+4BU+4(K+1)U^2+(10+2K)U$	U
PCI[8]	$(8K+4)BU+(6K+2)U$	0
SDJC[11]	$2BU^2+4BU+(4K+8)U^2+(20-2K)U$	U+1
NRI(n=1)[12]	$2BU^2+(8K-4)BU+6U^2$	0
proposed-1	$(8K+4)BU+(4K+2)U$	0
proposed-2	$(8K+4)BU+(6K+2)U$	0

3.7 两种算法对比

从算法复杂度上，两种算法在迭代三次时，相对复杂度差如式3–23所示，复杂度的提高仅为 0.2%，可以忽略不计。

$$\text{相对复杂度差}=\frac{\text{复杂度(proposed-2)}-\text{复杂度(proposed-1)}}{\text{复杂度(proposed-1)}}=\frac{6U}{28BU+14U}=0.2\% \quad (3\text{–}23)$$

从性能上，尽管在瑞利信道，128 × 16 MIMO 系统中，两者的性能几乎一致，但是在 correlation 信道，迭代一次时 proposed-2 算法性能远优于 proposed-1 算法。

从性能和复杂度两方面来说，两种算法没有明显的优劣之分。proposed-2 算法需要存储前两次迭代的计算结果，所需的硬件资源占用更高，数据依赖性大。然而，考虑到实际信道的未知性，这意味着为达到近乎最优性能所需的迭代次数是不确定的，在小节3.4.3中提到，proposed-1 算法在总迭代次数不同时，每一次迭代的参数均不同，因此无法在原有迭代结果上增加次数来获得性能的进一步的提升，而 proposed-2 算法则没有这一问题，因此将基于 proposed-2 算法完成硬件实现。

3.8 本章小结

本章介绍了基于 Non-Stationary Richardson 迭代和 Second-Order Richardson 迭代检测算法的完整流程，并且针对大规模 MIMO 系统特性，提出了三种优化策略：特征值估计，低复杂度初始化方法和分步矩阵向量乘，从而避免了 Gram 矩阵的计算及其求逆，在提高性能的基础上极大的降低了复杂度。本章还给出了在瑞利信道以及更为实际的 correlation 信道下不同算法的 BER 性能对比。仿真结果表明，在天线配置为 128×16 的 MIMO 系统中，本文提出的两种算法节约了一次迭代的复杂度的同时性能和和 CG 算法几乎一致，优于 NS 算法和 NRI 算法。在相同的迭代次数下，propsoed-1 算法在不牺牲性能的前提下具有最低的复杂度。此外，proposed-2 能够适应不同规模的大规模 MIMO 系统，以及在迭代一次时具有明显优于其他算法的性能。

第四章　基于 Second-Order RI 检测算法的硬件实现

本章介绍基于 Second-Order Richardson 迭代的检测算法（proposed-2 算法）完成的高吞吐率、高硬件效率的硬件实现方案，介绍内容包括算法的实数化和定点化，系统架构以及关键模块的设计，整个设计基于 Verilog 语言并在 Xilinx Virtex-7 开发板上进行实现。本硬件架构适用于 $128 \times U$ 天线配置的大规模 MIMO 系统，其中 U 为 8~32 中的任意数值，即本文提出的硬件架构支持 8~32 个可变用户。

4.1　算法的实数化与定点化

4.1.1　实数化

原有的的 proposed-2 算法是基于复数的运算，在硬件实现中，需要将算法进行实数化以易于数据操作。实数化包括复数矩阵实数化和复数向量实数化两部分，分别如式 (4–1) 和式 (4–2) 所示。

$$H_{real} = \begin{bmatrix} \Re(H) & -\Im(H) \\ \Im(H) & \Re(H) \end{bmatrix} \tag{4–1}$$

$$x_{real} = \begin{bmatrix} \Re(x) \\ \Im(x) \end{bmatrix} \tag{4–2}$$

实数化后算法最终得到的计算结果 $\mathbf{x}_{k+1}$ 也是实数向量。

4.1.2　定点化

算法定点化是进行硬件设计前的重要步骤。算法的性能仿真都是基于易于操作的浮点数，但是对于硬件实现而言，浮点计算会造成大量不必要的资源开销，因此需要对算法进行定点化处理，通过限定数据位宽，指定数据精度，将浮点数转换为定点数。合理的定点化旨在减小资源消耗的同时保证性能不受太大的影响。

借助 MATLAB 的 Fixed-point designer 工具，编写 testbench 对算法多次仿真，观察输入输出以及中间数据的取值范围，从而确定每个数据定点化参数。经过多次实验，最终得到如表4–1所示的 proposed-2 算法定点化参数表，

4.2　系统整体架构

为适应更广泛的应用场景，本文提出的硬件架构支持信道矩阵 $H \in 128 \times U$ 的用户数可变系统，其中 U∈(8~32)。根据性能仿真结果，不同信道下为了达到近乎最优的性能所需要的迭代次数是不一致的。瑞利信道下信道矩阵为 128×8 和 128×16 时，2 次迭代能达到与 MMSE 几乎一致的性能，而在瑞利信道下的信道矩阵为 128×32 以及 correlation 信道下，通常需要 4

表 4–1 proposed-2 算法定点化参数表

数据		规模	数据精度
输入	$\mathbf{H}$	U×128	(1,16,13)
	$\mathbf{y}$	128×1	(1,16,10)
	σ^2	-	(1,16,12)
	K	-	(1,3,0)
中间结果	$\mathbf{y}^{MF}$	$U \times 1$	(1,16,6)
	$\alpha_{opt}\gamma_{opt}$	-	(1,16,21)
	$\mathbf{x}_0$	$U \times 1$	(1,16,13)
	$\mathbf{x}_1$	$U \times 1$	(1,16,13)
	$\tilde{\mathbf{h}}$	128×1	(1,16,10)
	$\mathbf{H}^H\tilde{\mathbf{h}}$	$U \times 1$	(1,16,5)
	$\mathbf{y}^{MF} - \sigma^2\mathbf{x}_k$	$U \times 1$	(1,16,6)
	$\gamma_{opt}(\mathbf{x}_k - \mathbf{x}_{k-1})$	$U \times 1$	(1,16,13)
	$\mathbf{x}_{k-1} + \gamma_{opt}(\mathbf{x}_k - \mathbf{x}_{k-1})$	$U \times 1$	(1,16,13)
	$\gamma_{opt}\alpha_{opt}(\mathbf{y}^{MF} - \sigma^2\mathbf{x}_k - \mathbf{H}^H\tilde{\mathbf{h}})$	$U \times 1$	(1,16,14)
	$\mathbf{x}_{k+1}$	$U \times 1$	(1,16,13)
输出结果	$L_{i,b}$	$\log_2(M)$	(1,8,3)

次迭代来满足性能要求。系统整体采用流水线和脉动阵列结构，计算模块高并行，在两次迭代时计算单元使用率达 100%。系统的整体结构如图4–1所示。

整个系统主要由三部分组成，分别是控制单元，存储单元和计算单元。控制单元负责控制 memory 的存取地址以及计算模块的逻辑控制；存储模块负责存储信道矩阵 $\mathbf{H}$ 和来自用户的信息 $\mathbf{y}$；计算单元则是整个架构的核心，包括初始化模块和两次迭代模块，以及最后计算 LLR 输出的模块，其中每次迭代模块包括两个矩阵乘向量模块，因此计算单元可以分为共 6 个模块。初始化模块计算 $\mathbf{y}^{MF}$ 和每次迭代中计算 $\mathbf{H}^H\tilde{\mathbf{h}}$ 的 stage 2 模块实现一致; 两次迭代中计算 $\tilde{\mathbf{h}}$ 的 stage 1 模块的实现一致。为提高硬件效率，仅构建两次迭代模块，若需 4 次迭代，可以方便的通过控制模块 control unit 令 initial block 暂停，将 iterative block 2 stage 2 模块计算的结果 $\mathbf{x}_3$ 作为 iterative block 1 stage 1 模块的输入，接着进行第 3 次和第 4 次迭代，最后将数据送入 LLR 模块得到软比特输出。

系统时序如图4–2所示，各级模块所需周期数大致相同，当流水线被充满之后，各级之间的计算同时进行，提高了硬件的吞吐率和资源使用率。系统的延迟为 59 cycles，其中初始化和两次迭代的 stage 2 均为 13 cycles 的延迟，而两次迭代的 stage 1 模块分别需要 9 cycles 的延迟，LLR 计算需要 3 cycles 的延迟。

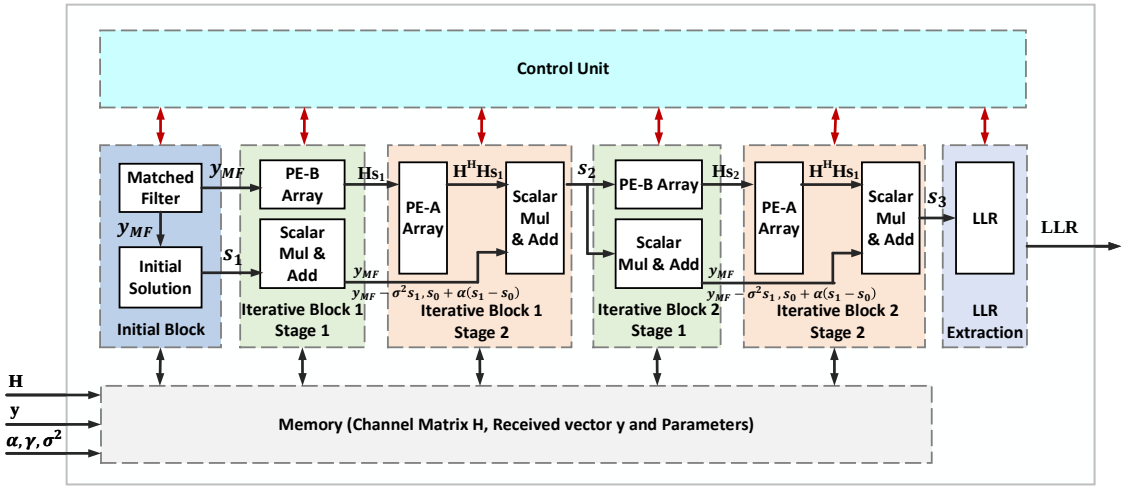


图 4–1 proposed-2 算法系统架构图

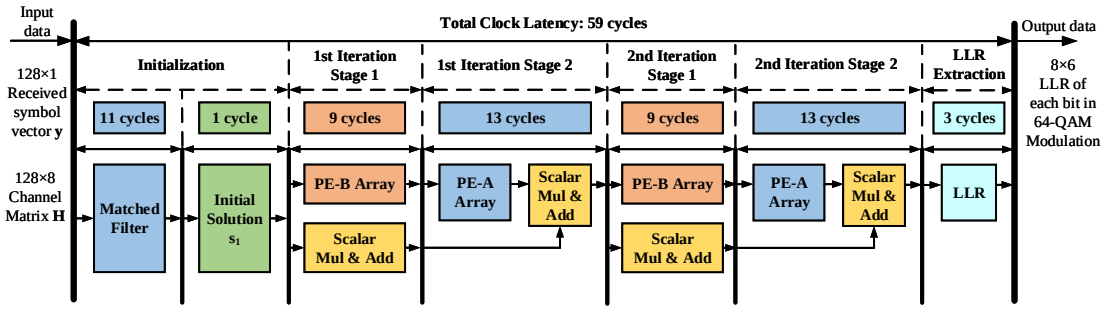


图 4–2 系统时序图

4.3 模块介绍

系统需要完成的关键计算分为三类，包括矩阵向量乘 $\tilde{\mathbf{h}} = \mathbf{H}\mathbf{x}_k$，矩阵向量乘 $\mathbf{H}^H\tilde{\mathbf{h}}$ 以及常数乘向量。其中，常数乘向量仅为简单相乘，不予详细介绍。存储模块负责存储信道矩阵 $\mathbf{H}$ 和用户信息 $\mathbf{y}$。由于 $\mathbf{H}^H$ 为其共轭转置矩阵，因此无需占用额外的存储空间。由于硬件架构需要支持可变用户数，而 ASIC 的实现要求最开始的资源分配均为确定的，因此需要设计一种硬件资源分配与用户数无关，即用户数仅影响时间维度的实现策略，下面将展开详细介绍。

4.3.1 复数乘法

硬件实现中需将复数乘法转化为实数乘法模块:

$$(a+bi)(c+di) = (ac-bd) + (ad+bc)i \tag{4–3}$$

由于硬件实现上乘法和加法的时间复杂度和面积复杂度是不一样的，乘法运算的时间和面积通常远大于加法，因此用廉价运算代替昂贵运算可以有效加速运算。本文采用的 winograd 算法，其运算规则如式 (4–4) 所示。

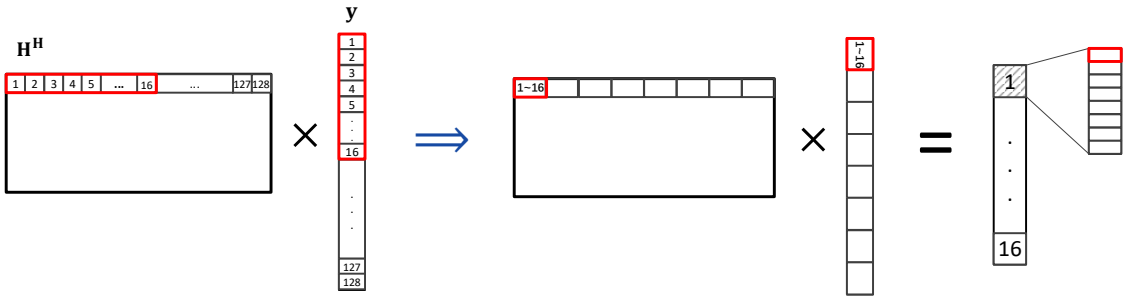


图 4–3 $\mathbb{C}^{U\times 128}\times\mathbb{C}^{128\times 1}$ 矩阵乘向量操作抽象图

$$\begin{aligned} ac-bd &= a(c-d)+d(a-b) \\ ad+bc &= b(c+d)+d(a-b) \end{aligned} \tag{4–4}$$

通过上式，一个复数乘法，由“4 乘 3 加”实数运算减小为“3 乘 5 加”的实数运算，有效的减少了 25% 的实数乘法运算。本文所有的复数乘法均默认用 winograd 算法实现。

4.3.2 矩阵向量乘：PE-A array

初始化模块计算 $\mathbf{y}^{MF}=\mathbf{H}^H\mathbf{y}$ 与迭代模块中 stage 2 计算 $\mathbf{H}^H\tilde{\mathbf{h}}$ 均是维度为 $\mathbb{C}^{U\times B}\times\mathbb{C}^{B\times 1}$ 的复数乘法操作，因此采用同样的策略实现。

图4–3以 $\mathbf{y}^{MF}=\mathbf{H}^H\mathbf{y}$ 的计算过程为例，展示了计算 $\mathbb{C}^{U\times B}\times\mathbb{C}^{B\times 1}$ 操作的抽象示意图。其中基站天线数为固定值 128，考虑到支持的用户数为 8~32，将 $\mathbf{H}^H$ 每行 128 个元素以及 $\mathbf{y}$ 分成 16×8，16 个元素为 1 组并行处理，共 8 组。一次运算可以得到 16 个元素的乘法结果, 也就是 $\mathbf{y}^{MF}$ 第一个元素所需乘法操作的 1/8。因此仅需 8 次运算并相加即可得到 $\mathbf{y}^{MF}$ 的第一个元素值。

PE-A array 模块矩阵乘向量操作示意图如图4–4所示。为方便起见，以 $\mathbf{H}^H\in\mathbb{C}^{8\times 128}$ 为例进行介绍。每个 PE-A 模块的输入为 $\mathbf{H}^H$ 和 $\mathbf{y}$, 分别用黑色实线和蓝色实线表示。

将 PE-A 从左到右排序，2 号 PE-A 的内部实现如图所示。1 号 PE-A 在第一个周期将藕粉色的 16 个元素与 $\mathbf{y}$ 深棕色的 16 个元素相乘，得到 1~16 乘积结果，经过一级寄存器传递给 2 号 PE-A。第二个周期，1 号 PE-A 处理的数据向下移动，将第一个灰色列块浅蓝色的 16 个元素与 $\mathbf{y}$ 深棕色的 16 个元素相乘；与此同时，2 号 PE-A 从存储模块中取第二个灰色列块的浅蓝色 16 个元素，与 $\mathbf{y}$ 橙棕色的 16 个元素相乘得到 17~32 乘积结果，并在同一周期接收来自 PE-A 的 1~16 个乘积结果，将两者相加后传递给下一个 PE-A 单元，因此，3 号 PE-A 单元将在第三个时钟周期接收到来自 2 号 PE-A 的 16 个乘积结果并与内部乘积 33~48 乘积结果相加传给 4 号 PE-A。以此类推，8 个时钟周期后 8 号 PE-A 将得到有 16 个数组成的 $\mathbf{H}^H_{1,:}\times\mathbf{y}$ 的结果，经过 3 级加法树共两个周期得到 $\mathbf{y}_1^{MF}$。考虑到复数乘法器的延迟，PE-A 中每个乘法符号内部有一级的时钟延迟，因此 stage 1 模块的延迟为 11 个时钟周期，与时序图一致。

图4–5展示了 PE-A 与存储模块数据流交互顺序示意。其中淡灰色列块和浅灰色列块分别代表两个不同的信道矩阵 $\mathbf{H}_1^H$ 和 $\mathbf{H}_2^H$，而棕色系与灰色系方块则表示两组不同的 $\mathbf{y}$。

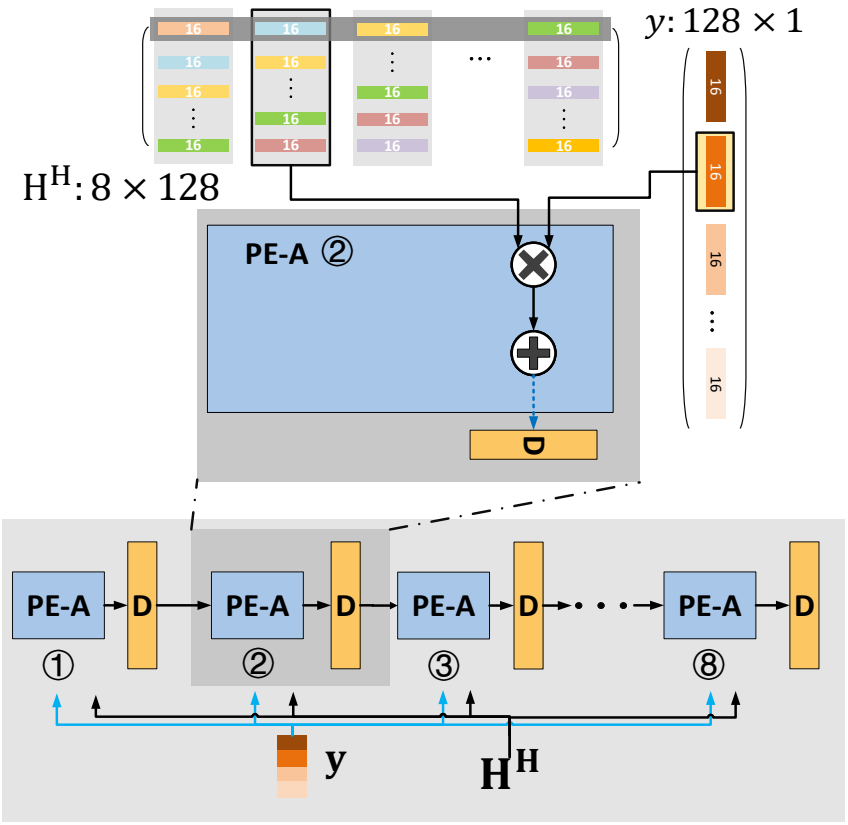


图 4–4 PE-A array 矩阵乘向量实现逻辑示意图

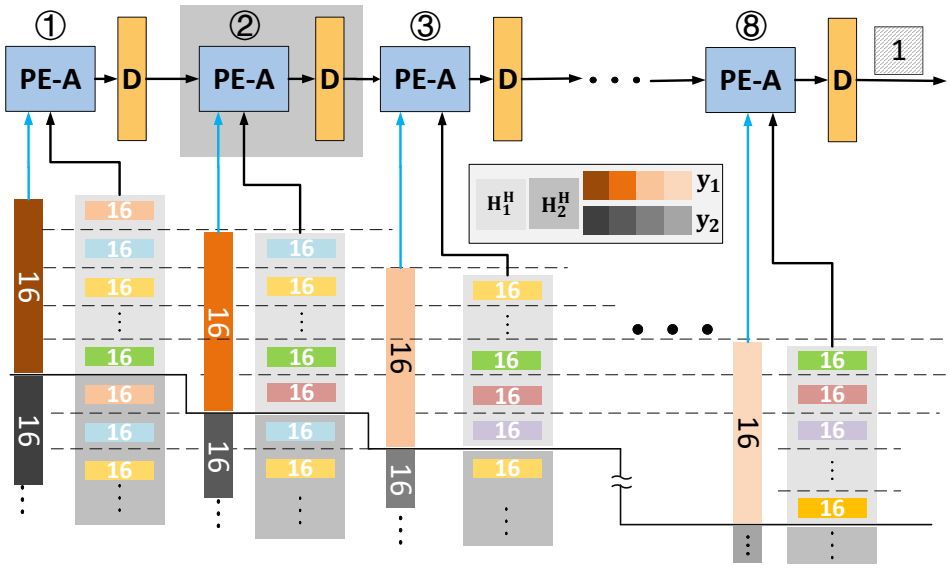


图 4–5 PE-A array 数据流示意图

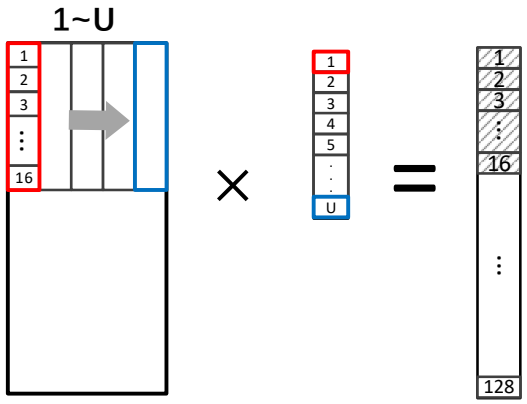


图 4–6 $\mathbb{C}^{128\times U} \times \mathbb{C}^{U\times 1}$ 矩阵乘向量操作抽象图

每个 PE 处理 $\mathbf{H}^H$ 的一个灰色列块，共 8 个 PE-A 对应一行 8 组元素，相应的，每个 PE-A 将矩阵的元素乘以 $\mathbf{y}$ 的相同 16 个元素。每个 PE-A 计算出 16 个乘积结果后，经过一级寄存器传递给下一个 PE-A 模块。每一个 PE-A 比上一个 PE-A 晚一个时钟周期开始工作，8 个时钟周期后，所有 PE-A 单元同时工作。

虚线从上至下代表时钟周期的增加，相同颜色代表在同一个时钟周期使用的数据。对比图4–4矩阵 $\mathbf{H}^H$ 元素分组的颜色，可以看到 stage 1 的架构设计不会同时使用一行中的两组数据，因此矩阵元素按照灰色方框存储，即一行存于一个 memory 中，用户数变化仅影响一个矩阵所需的 memory 个数，而不会产生数据冲突。观察图4–5可得，当 8 号 PE-A 处理 $\mathbf{H}^H_{2,113:128}$ 时，1 号 PE 已经处理完 $\mathbf{H}^H_1$ 的灰色列块紧接着开始处理第二个矩阵 $\mathbf{H}^H_2$，更换矩阵 $\mathbf{H}^H$ 的周期为 U。因此对于 stage 1 模块而言，13 个周期后，每个是时钟周期能产生一个结果传递给 stage 2 模块。

4.3.3 矩阵向量乘：PE-B array

stage 1 模块指的是计算 $\tilde{\mathbf{h}} = \mathbf{H}\mathbf{x}_k$ 的模块，实现维度为 $\mathbb{C}^{B\times U} \times \mathbb{C}^{U\times 1}$ 的矩阵乘向量操作。与4.3.2小节一致，以 $\mathrm{B} \times \mathrm{U} = 128 \times 8$ 为例展开该模块实现策略的介绍。

图4–6展示了 PE-B 矩阵乘向量操作的示意图。考虑到 stage 2 以及 matched filter 模块每个周期输出一个结果，为尽可能早的利用计算结果，此模块的基本思想为乘累加技术。矩阵 $\mathbf{H}$ 共 U 列，每列 128 个元素分为 16×8，16 个为 1 组，共 8 组。$\mathbf{H}$ 的第一列元素均乘以 $\mathbf{x}_k$ 的第一个元素 x_1。每个 PE-B 处理的数据由红色方框向右移至蓝色方框，相应的 $\mathbf{y}$ 同时从上至下移动，因此在 U 个周期后，能够得到 $\tilde{\mathbf{h}}$ 的前 16 个元素 $\tilde{h}_{1:16}$，作为 stage 1 的输入，每个周期输出一组 16 个元素，经 U 个周期将 $\tilde{\mathbf{h}}$ 的 128 个元素输出完毕，与 stage 1 模块的迭代周期保持一致，因此整个系统得以流水，各个模块并行运作。

为方便起见，本小节以 $\mathbf{x}$ 表示每一次迭代结果 $\mathbf{x}_k$，x_i 表示 $\mathbf{x}$ 的第 i 个元素。图4–7展示了 PE-B array 相互间的关系及其内部实现逻辑。棕色实线表示数据 $\mathbf{x} \in \mathbb{C}^U$ 在 PE-B 单元间顺向传递，蓝色实线表示信道矩阵 $\mathbf{H}$，红色实线表示每个 PE-B 单元的输出：$\tilde{h}_{16i:16(i+1)}$。

假设 PE-B 单元在第一个时钟周期开始工作，因此第 3 个时钟周期 3 号 PE-B 单元接收到

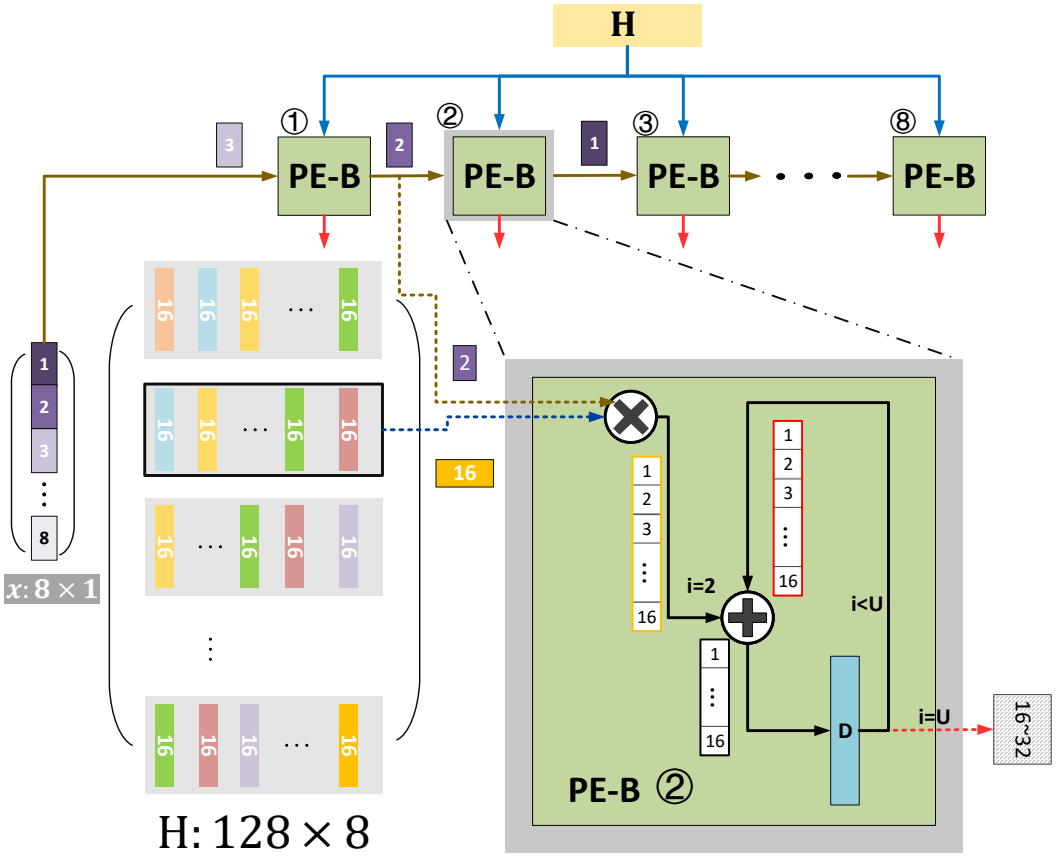


图 4–7 PE-B array 矩阵乘向量实现逻辑示意图

来自 2 号 PE-B 单元的元素 x_1; 同样的 2 号 PE-B 单元接收来自 1 号 PE-B 单元的元素 x_2，以紫色方块 2 表示。2 号 PE-B 单元的另一个输入为矩阵 **H** 的第二行，与之对应的即为 stage 2 单元图4–4$\mathbf{H}^H$ 的第二个灰色列块。2 号 PE-B 单元计算得到 $H_{17:32,2} \times x_2$ 的 16 个乘积结果与第二周期 $H_{1:16,1} \times x_1$16 个乘积结果相加得到 16 个加法结果，存入寄存器中并将 x_2 传递给 3 号 PE-B 单元。

图4–7展示了 PE-B 单元数据流示意图。PE-B 单元从左至右排序，每个 PE 处理矩阵 **H** 的一个行块，即 $\mathbf{H}^H$ 在图4–4的一个列块，PE-A 单元和 PE-B 单元对其读取策略一致，不再赘述；紫色系和灰色系方块表示两组不同的 **x**。**x** 的每一个元素以每周期一个元素的顺序输入 1 号 PE-B 单元，各个 PE-B 单元每周期向后传递 x_i 的同时，接收来自上一个 PE-B 单元的 x_{i+1} 元素。考虑到单元内部乘法器采用 winograd 算法，存在一个周期的时钟延迟，因此 1 号 PE-B 单元在第 9 个时钟周期得到 $h_{1:16}$ 与时序图4–2一致。第 10 个周期 1 号 PE-B 单元取新的子载波数据及其对应的新的 **H**, 同时 2 号 PE-B 单元生成 $h_{17:32}$。以上分析可得，所有单元工作之后 stage 1 模块每个时钟周期输出 $\tilde{\mathbf{h}}$ 的 16 个元素，与 stage 2 模块对输入数据的要求一致，因此整个系统所有单元并行工作。

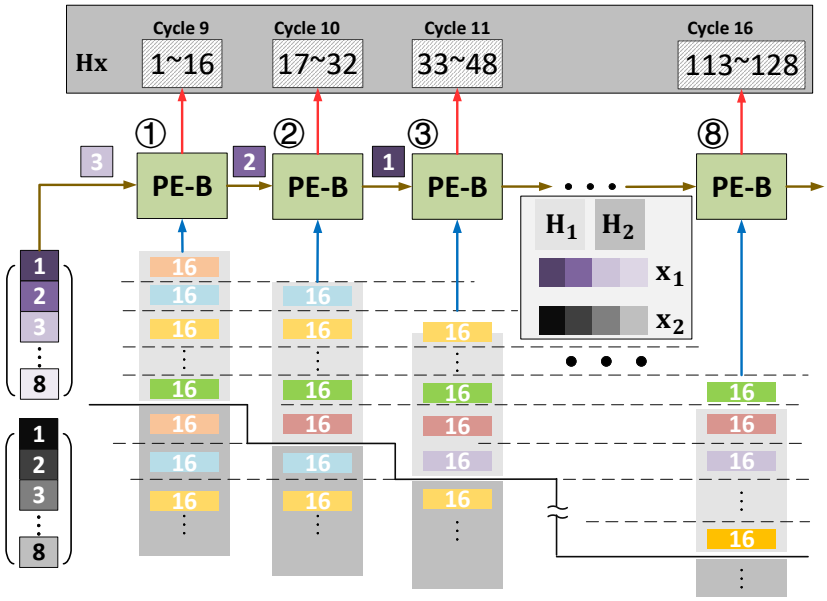


图 4–8 PE-B array 数据流示意图

4.3.4 LLR 模块

LLR 模块接受 iterative block 2 stage 2 模块输出的分别对应于 2 次迭代和 4 次迭代的符号向量 $\mathbf{x}_3$ 和 $\mathbf{x}_5$. 按照下式运算，根据调阶数 M 输出 $\log_2 M$ 比特对应的软信息，本文设计采用 64-QAM 调制方式，因此每周期输出 6 个 LLR，每个 LLR 为 8 比特。

$$L_{i,b}(\hat{x}_i) = \rho_i\left(\min_{x\in X_b^0}\left|\frac{\hat{x}_i}{\mu_i} - x\right|^2 - \min_{x\in X_b^1}\left|\frac{\hat{x}_i}{\mu_i} - x\right|^2\right) \tag{4–5}$$

4.4 本章小结

本章完成了基于 proposed-2 算法的硬件设计，对设计思路，系统架构及其核心模块进行了详细描述。本文提出的硬件架构能够支持 8 ~ 32 多用户系统，可以通过简单的控制逻辑增加迭代次数来获得更有的性能。当迭代次数为 2 时，计算使用率达 100%。此外，每个周期能够计算得到输出一个 x_i 结果，对应于 6 个 8 比特的软输出用于带编码的系统。

第五章　硬件实现结果

本章首先介绍硬件实现和验证流程，然后给出硬件实现结果并与现有算法的对比。

5.1　硬件实现和验证流程

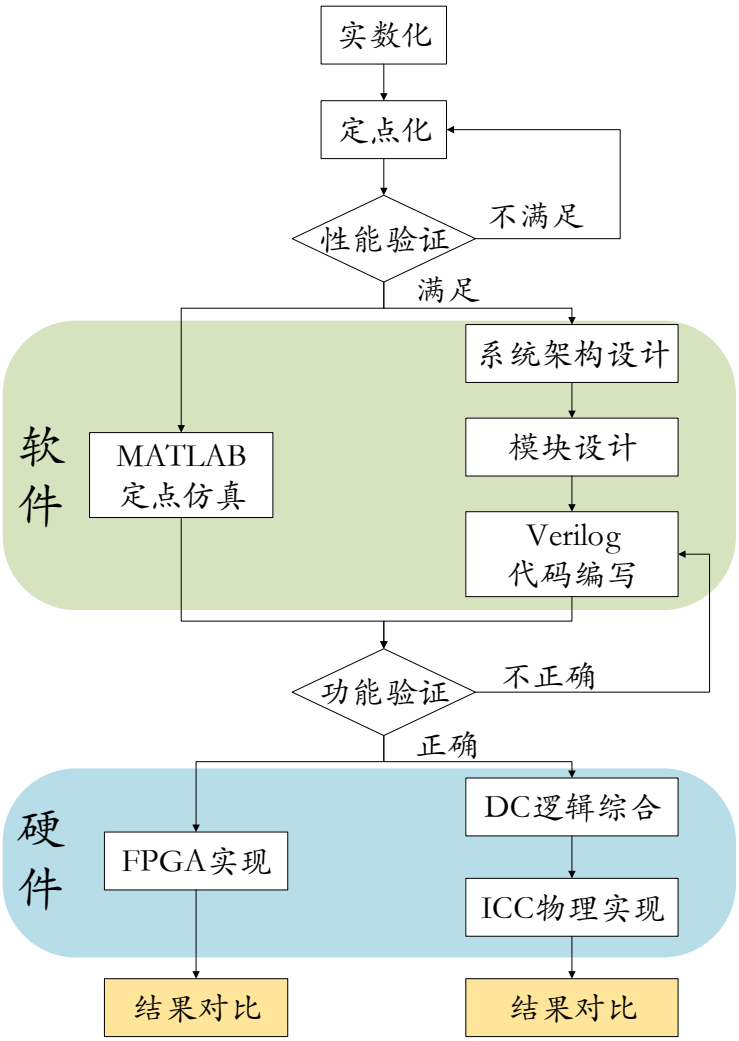


图 5–1 硬件实现和验证流程图

图5–1展示了从软件仿真到硬件实现的具体流程。首先对进行算法实数化和定点化并用 matlab 仿真验证性能的准确性，若是性能不满足则修改定点方案，若是性能满足则进行模块设计和 RTL 代码撰写。利用 modelsim 工具对硬件代码进行功能仿真，将结果与 MATLAB 定

表 5–1 FPGA 实现结果对比

Detector	NS[4]	CG[22]	GS[23]	PCI[8]	SDJC[11]	This work
the number of iteration, K	3	3	1	2	2	2
MIMO scale	128 × 8	128 × 8	128 × 8	128 × 16	128 × 16	128 × U (U ∈(8,32))
LUTs	168125	3324	18976	70288	6330	72158
FFs	193451	3878	15864	70452	28010	52426
DSP48s	1059	33	232	1064	1312	1164
Clock frequency [MHz]	317	412	309	205	200	217
Latency (clock cycles)	196	951	311	n.a.	n.a.	59
Throughput [Mb/s]	**603**	**20**	**48**	**1230**	**1.65**	**1302**
Throughput/LUTs	**4173**	**6016**	**2530**	**17499**	**260**	**18043**

点结果对比，结果完全一致表示功能正确，否则修改代码重新验证。随后利用 Xilinx Vivado 工具进行 FPGA 实现，并通过 post-implementation 报告了解资源消耗情况。逻辑综合通过 Design compiler 工具完成，物理实现通过 IC compiler 工具实现。

5.2 FPGA 实现结果对比

本设计在 VIRTEX-7 XC7VX980T FPGA 上进行了硬件实现，并与相关文献的硬件实现结果进行了对比，如表5–1所示。

对于硬件支持的 MIMO 系统的规模，仅本文提出的 propsoed-2 算法硬件架构能够支持 8 ~ 32 任意用户数的大规模 MIMO 系统。考虑到实际场景，基站的天线数变化成本较大，而用户数的切换方便且廉价，因此固定基站天线数，支持可变用户数的策略符合实际条件。

本设计具备高并行的特性，数据从输入到输出的延迟仅需 59 个时钟周期，小于 NS 算法，CG 算法，GS 算法，PCI 算法和 SDJC 算法。得益于高并行处理的脉动阵列架构，流水线单元工作率 100%，proposed-2 算法的吞吐率达到了 1302Mb/s。PCI 算法和 proposed-2 算法的具有明显优于表中其余算法的吞吐率，而本文提出的硬件架构具有比 PCI 更高的时钟频率，因此本文提出的硬件架构仍优于 PCI 算法的吞吐率，约提高了 5.8%。尽管本文的架构，消耗了更多 LUT 单元，这是为了获得高吞吐率而付出的一定代价，归一化之后，硬件效率（Throughput/LUTs）达到 18043，显著优于表中其余算法，相比 PCI 算法仍有 3% 的提升。

5.3 ASIC 实现结果对比

本文对提出的基于 Second-Order Ricahrdson 迭代检测算法的硬件设计在 SMIC 40nm，1.21V 工艺下进行 ASIC 实现。表5–2总结了大规模 MIMO 系统下，迭代检测算法 post-layout 实现的关键指标，并对不同检测算法的 ASIC 结果进行对比。能量效率定义为吞吐率与功率的比值（Throughput/Power)，面积效率定义为吞吐率与面积的比值（Throughput/Area）。对于

表 5–2 不同大规模 MIMO 检测器的 ASIC 实现结果比较

Inversion method	Weji[24]	NS[25]	PCI[8]	CG[26]	This work
Technology	65nm	45nm	65nm	45nm	40nm
MIMO Systems	128×8	128×8	128×16	128×8	$128\times U$ $U\in(8,32)$
modulation	64-QAM	64-QAM	64-QAM	64-QAM	64-QAM
Memory [KB]	3.52	131.25	37.22	1.5	16.13
Area (mm^2)	2.57	11.1	7.70	0.107	2.09
Frequency (MHz)	680	1000	680	934	500
Power(W) (@ Voltage)	0.65 (@ 1.00V)	8 (@ 0.82V)	1.66 (@ 1.00V)	0.03 (@0.81V)	0.71 (@1.21V)
Throughput [Gbps]	1.02	3.8	4.08	0.06	3.00
Energy efficiency (Gbps/W)	1.58	0.475	2.46	2.00	4.23
Area efficiency ($Gbps/mm^2$)	0.40	0.34	0.53	0.55	1.43
Normalized Energy* efficiency (Gbps/W)	**2.85**	**0.28**	**4.44**	**1.13**	**4.23**
Normalized Area* efficiency ($Gbps/mm^2$)	**1.72**	**0.48**	**2.27**	**0.78**	**1.43**

* 归一化为 40nm CMOS 工艺，假设如下：$f_{clk}\sim s, A\sim 1/s^2, and P_{dyn}\sim(1/s)(V_{dd}/V'_{dd})^2$.

不同的工艺技术，为了保证公平的比较，表中给出了将能量效率和面积效率归一化到工艺为 40nm，电压为 1.21V 后的比较结果。计算方案如式5–1所示。

$$f_{clk}\sim s, A\sim 1/s^2,\ \ and\ \ P_{dyn}\sim(1/s)(V_{dd}/V'_{dd})^2 \tag{5–1}$$

其中 s, A, P_{dyn} 和 V_{dd} 分别表示工艺，面积，功率和电压。这种缩放方法广泛用于比较不同工艺的不同架构，如[24, 25, 27, 28]。

可以看到，仅本文提出的架构支持可变用户数的大规模 MIMO 系统，高并行的数据架构使得 memory 的消耗有所提升，但是仍低于 NS[25] 和 PCI[8]。在 500MHz 的时钟频率下，吞吐率达到了 3Gb/s，相较于 Weji[24] 和 CG[26] 分别提高为 **2.94**× 和 **50.0**×。尽管本文的吞吐率略低于 NS 和 PCI 的实现结果，但是能量效率和面积效率相较于表中其他检测器均有所提高。按照式 (5–1) 的计算规则，将能量效率和面积效率归一化至 40nm，1.21V 工艺后，本文实现结果的归一化能量效率相较于 Weji、NS 和 CG 检测器分别提高为 **1.48**×、**15.1**× 和 **1.76**×，与 PCI 检测算法归一化能量效率相当。归一化的面积效率低于 Weji 检测器和 PCI 检测器，但是相较于 NS 检测器和 CG 检测器分别提高为 **2.98**× 和 **1.83**×。

5.4 本章小结

本章对 VLSI 硬件架构的实现结果与现有文献结果进行了对比，其中 FPGA 实现结果显示本文的硬件架构具有低延迟高吞吐率的特性。此外，ASIC 实现结果显示，在归一化能量效率方面，本文的实现优于 Weji 检测器、NS 检测器，PCI 检测器和 CG 检测器；在归一化面积效率方面相较于 NS 检测器以及 CG 检测器也有一定程度的提高。

第六章　结论

6.1　全文工作总结

本文以复杂度，性能，硬件实现以及 ASIC 实现为线索，对大规模 MIMO 系统检测算法那进行了学习与研究，现将全文的工作总结为如下 6 个部分。

(1)　本文对国内外近几年来有关大规模 MIMO 预编码算法和检测算法的文献进行了调研，通过文献的学习，了解了大规模 MIMO 系统的相关原理，预编码和检测问题的模型，信道矩阵的硬化效应对线性检测算法在性能上带来的影响。认识到当前大规模 MIMO 检测系统存在的主要问题为如何在保证性能的前提下降低线性算法的复杂度。该问题面临的两大挑战为 Gram 矩阵的计算和对 Gram 矩阵求逆。阅读检测算法文献的同时，追根溯源，对其数学本质进行学习和研究，将现有基于 MMSE 检测的迭代算法按照求逆思路分为四类，详细分析了每种算法的特点并指出其存在的不足。

(2) 基于 MATLAB 仿真平台进行仿真，该平台包含了大规模 MIMO 系统中所有必需的信号处理过程，并有上行链路和下行链路两个版本，在基础的瑞利信道模型之外，考察了更为实际的 correlation 信道模型，以此验证算法在不同模型下的通用性。通过该平台，完成了对现有检测算法的性能评估和对比工作。此外，还学习了 MATLAB fixed-point designer 工具箱的使用，并借此完成了算法定点化工作，学习了 Vivado 硬件实现的流程与硬件优化的策略，了解 Vivado 线程并行化加速综合过程的方法。

(3) 针对大规模 MIMO 系统的信道增强效应，即当基站天线数远大于用户数时 Gram 矩阵严格主对角占优的性质，本文提出了三种优化策略。进一步提高性能的基础上，大幅降低了算法乘法复杂度。同时还避免了除法运算，硬件实现更为友好。

(4) 本文提出了两种新的分别基于 Non-Stationary Richardson 迭代和 Second-Order Ricahrdson 迭代的检测算法，通过采用上述的三种优化策略，避免了 Gram 矩阵的计算。在相同的迭代次数下，本文提出的两种算法均能获得较优的性能，以瑞利信道 128×16 大规模 MIMO 系统为例，两种算法迭代三次能获得与 MMSE 一致的性能。同时，本文提出的两种算法复杂度大幅降低。两种算法除了上述共性，也各具特色。基于 Non-Stationary Ricahrdson 迭代的检测算法，尽管参数计算的形式复杂，但是所有松弛因子参数均可提前计算，因此在本文所述算法中具有最低的复杂度。基于 Second-Order Richardson 迭代的检测算法，每一次迭代的结果需要前两次迭代的结果作为参考，由此提高了收敛速度，尤其是在迭代一次时，在不同的 MIMO 系统规模抑或是不同的信道模型中，该算法能够获得明显优于 NS 算法，CG 算法，GS 算法，PCI 算法，SDJC 算法的 BER 性能。

(5) 完成了基于 Second-Order Richardson 检测算法的硬件实现，确定定点策略之后，为支持多用户 MIMO 系统，设计了两种矩阵向量乘结构，从而巧妙的避免了可变用户数所带来硬件资源浪费的情况。通过采用脉动阵列的结构，高并行计算，以及流水线策略，在 217M 的时钟频率下达到了 1302Mb/s 的高吞吐率，硬件效率（Throughput/LUTs）为 18043，远高于 NS 算法, CG 算法，GS 算法和 SDJC 算法，相较于 PCI 算法也有 3% 的提高。

(6) 在 FPGA 实现的基础上进一步完成了 ASIC 实现工作，并与不同大规模 MIMO 系统检测器 ASIC 实现结果进行比较。本文实现的检测器吞吐率在 400M 时钟频率下吞吐率达到了 2.4Gbps。对实现结果标准化至 40nm,1.21V 工艺后，本文实现结果的标准化能量效率相较于 Weji、NS 和 CG 检测器分别提高为 **1.48**×、**15.1**× 和 **1.76**×。

6.2 创新点总结

本文致力于研究适用于大规模 MIMO 系统的高性能、低复杂度的检测算法，并探究高吞吐率的硬件结构，创新点包括：

(1) 提出了三种适用于现有迭代方法的优化策略：低复杂度初始化方法，可以进一步提高算法性能；Gram 矩阵特征值估计，仅需要基站天线数目 B，同时服务的用户数 U 和噪声 N0, 即可确定最大最小近似特征值，有效降低了参数计算的复杂度；得益于上述两种优化策略，Gram 矩阵的计算以及 Gram 矩阵与迭代结果的乘积可以用分步矩阵向量乘代替，在迭代三次时仍能减少 25% 以上的复杂度。

(2) 提出了两种迭代检测算法：基于 Non-Stationary Richardson 迭代和 Second-Order Richardson 迭代。结合上述三种优化策略，可以方便的通过增加迭代次数具备十分接近精确 MMSE 检测的性能，两种算法在性能和复杂度上均优于本文所提及算法。尤其是基于 Second-Order Richardson 的迭代算法，对不同规模的天线系统以及不同的信道矩阵具有较强适应性。

(3) 提出了支持可变用户数的硬件架构，并完成了 FPGA 原型实现和芯片实现，主要包括两个矩阵向量乘模块的实现。本设计的数据调度将固定的基站天线数映射为 PE 模块数量，而用户数映射为时间延迟，因此用户数量的变化仅影响时间延迟而不影响吞吐率和硬件效率。硬件实现结果对比显示，本文具备优于其他算法的高吞吐率和高硬件效率。

6.3 未来工作展望

从本文的研究结果出发，未来仍有可以改进和拓展的内容，主要包括以下两点：

(1) 9 月份完成流片工作。

(2) 本毕业设计的芯片实现中，面积效率还有进一步提升的空间，在后端优化中，可以降低布局布线的难度和开销，以此减少布线带来的面积和功耗损失；同时还能减少线延迟，进一步提高时钟频率。

参考文献

[1] 推进组 I 2 (. IMT-2020 (5G) 推进组发布 5G 技术白皮书[J]. 中国无线电, 2015(5): 6.

[2] NGO H Q, LARSSON E G, MARZETTA T L. Energy and spectral efficiency of very large multiuser MIMO systems[J]. IEEE Transactions on Communications, 2013, 61(4): 1436-1449.

[3] YANG S, HANZO L. Fifty Years of MIMO Detection: The Road to Large-Scale MIMOs[J]. IEEE Communications Surveys Tutorials, 2015, 17(4): 1941-1988. DOI: 10.1109/COMST.2015.2475242.

[4] WU M, YIN B, WANG G, et al. Large-Scale MIMO Detection for 3GPP LTE: Algorithms and FPGA Implementations[J]. IEEE Journal of Selected Topics in Signal Processing, 2014, 8(5): 916-929. DOI: 10.1109/JSTSP.2014.2313021.

[5] GAO X, DAI L, MA Y, et al. Low-complexity near-optimal signal detection for uplink large-scale MIMO systems[J]. Electronics Letters, 2014, 50(18): 1326-1328. DOI: 10.1049/el.2014.0713.

[6] DAI L, GAO X, SU X, et al. Low-Complexity Soft-Output Signal Detection Based on Gauss–Seidel Method for Uplink Multiuser Large-Scale MIMO Systems[J]. IEEE Transactions on Vehicular Technology, 2015, 64(10): 4839-4845. DOI: 10.1109/TVT.2014.2370106.

[7] GAO X, DAI L, HU Y, et al. Matrix inversion-less signal detection using SOR method for uplink large-scale MIMO systems[C]//2014 IEEE Global Communications Conference. [S.l. : s.n.], 2014: 3291-3295. DOI: 10.1109/GLOCOM.2014.7037314.

[8] PENG G, LIU L, ZHANG P, et al. Low-Computing-Load, High-Parallelism Detection Method Based on Chebyshev Iteration for Massive MIMO Systems With VLSI Architecture[J]. IEEE Transactions on Signal Processing, 2017, 65(14): 3775-3788. DOI: 10.1109/TSP.2017.2698410.

[9] XUE Y, ZHANG C, ZHANG S, et al. Steepest Descent Method Based Soft-Output Detection for Massive MIMO Uplink[C]//2016 IEEE International Workshop on Signal Processing Systems (SiPS). [S.l. : s.n.], 2016: 273-278. DOI: 10.1109/SiPS.2016.55.

[10] YIN B, WU M, CAVALLARO J R, et al. Conjugate gradient-based soft-output detection and precoding in massive MIMO systems[C]//2014 IEEE Global Communications Conference. [S.l. : s.n.], 2014: 3696-3701. DOI: 10.1109/GLOCOM.2014.7037382.

[11] QIN X, YAN Z, HE G. A Near-Optimal Detection Scheme Based on Joint Steepest Descent and Jacobi Method for Uplink Massive MIMO Systems[J]. IEEE Communications Letters, 2016, 20(2): 276-279. DOI: 10.1109/LCOMM.2015.2504506.

[12] JIN F, LIU Q, LIU H, et al. A Low Complexity Signal Detection Scheme Based on Improved Newton Iteration for Massive MIMO Systems[J]. IEEE Communications Letters, 2019.

[13] LU L, LI G Y, SWINDLEHURST A L, et al. An Overview of Massive MIMO: Benefits and Challenges[J]. IEEE Journal of Selected Topics in Signal Processing, 2014, 8(5): 742-758. DOI: 10.1109/JSTSP.2014.2317671.

[14] WEI X, PENG W, NG D W K, et al. Joint Estimation of Channel Parameters in Massive MIMO Systems via PARAFAC Analysis[C]//2018 International Conference on Computing, Networking and Communications (ICNC). [S.l. : s.n.], 2018: 496-502. DOI: 10.1109/ICCNC .2018.8390299.

[15] HOCHWALD B M, MARZETTA T L, TAROKH V. Multiple-antenna channel hardening and its implications for rate feedback and scheduling[J]. IEEE Transactions on Information Theory, 2004, 50(9): 1893-1909. DOI: 10.1109/TIT.2004.833345.

[16] MARZETTA T L. Massive MIMO: An Introduction[J]. Bell Labs Technical Journal, 2015, 20: 11-22. DOI: 10.15325/BLTJ.2015.2407793.

[17] HEFNAWI M. Hybrid Beamforming for Millimeter-Wave Heterogeneous Networks[J]. Electronics, 2019, 8(2): 133.

[18] SHEWCHUK J R, et al. An introduction to the conjugate gradient method without the agonizing pain[Z]. 1994.

[19] YOUNG D. On Richardson's method for solving linear systems with positive definite matrices[J]. Journal of Mathematics and Physics, 1953, 32(1-4): 243-255.

[20] SILVERSTEIN J W, et al. The smallest eigenvalue of a large dimensional Wishart matrix[J]. The Annals of Probability, 1985, 13(4): 1364-1368.

[21] GOLUB G H, VARGA R S. Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order Richardson iterative methods[J]. Numerische Mathematik, 1961, 3(1): 157-168.

[22] YIN B, WU M, CAVALLARO J R, et al. VLSI design of large-scale soft-output MIMO detection using conjugate gradients[C]//2015 IEEE International Symposium on Circuits and Systems (ISCAS). [S.l. : s.n.], 2015: 1498-1501. DOI: 10.1109/ISCAS.2015.7168929.

[23] WU Z, ZHANG C, XUE Y, et al. Efficient architecture for soft-output massive MIMO detection with Gauss-Seidel method[C]//2016 IEEE International Symposium on Circuits and Systems (ISCAS). [S.l. : s.n.], 2016: 1886-1889. DOI: 10.1109/ISCAS.2016.7538940.

[24] PENG G, LIU L, ZHOU S, et al. A 1.58 Gbps/W 0.40 Gbps/mm2 ASIC Implementation of MMSE Detection for 128×8 64 -QAM Massive MIMO in 65 nm CMOS[J]. IEEE Transactions on Circuits and Systems I: Regular Papers, 2018, 65(5): 1717-1730. DOI: 10.1109/TCSI.2017 .2754282.

[25] YIN B, WU M, WANG G, et al. A 3.8Gb/s large-scale MIMO detector for 3GPP LTE-Advanced[C]//2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). [S.l. : s.n.], 2014: 3879-3883. DOI: 10.1109/ICASSP.2014.6854328.

[26] YIN B. Low complexity detection and precoding for massive MIMO systems: Algorithm, architecture, and application[D]. Ph.D. dissertation, Dept. Electr. Comput. Eng., Rice Univ., Houston, TX, USA, 2014.

[27] STUDER C, FATEH S, SEETHALER D. ASIC Implementation of Soft-Input Soft-Output MIMO Detection Using MMSE Parallel Interference Cancellation[J]. IEEE Journal of Solid-State Circuits, 2011, 46(7): 1754-1765. DOI: 10.1109/JSSC.2011.2144470.

[28] PENG G, LIU L, WEI Q, et al. A 2.69 Mbps/mW 1.09 Mbps/kGE Conjugate Gradient-based MMSE Detector for 64-QAM 128Œ8 Massive MIMO Systems[C]//2018 IEEE Asian Solid-State Circuits Conference (A-SSCC). [S.l. : s.n.], 2018: 191-194. DOI: 10.1109/ASSCC.2018.8579260.

谢　辞

本文的完成离不开实验室提供的资源和帮助，包括氛围良好的科研团队，涂家铭学长的耐心帮助以及导师贺光辉老师的建设性意见和方向指导。

首先，感谢实验室在大规模 MIMO 系统方面的研究基础，包括各类优秀文献的积累，仿真平台搭建积累的代码等，为我快速了解研究内容，找准研究方向提供了有力的帮助。尽管师兄师姐们的研究领域在传统 MIMO 系统或是随机计算与 MIMO 系统的结合，但是仍能就共通的问题相互协作和讨论，为我的研究带来灵感。

其次，感谢涂家铭学长在学业繁重、科研压力大的研一生活中，抽出大量的时间与精力同我讨论，从算法的提出到硬件结构的实现，涂家铭学长的参与和贡献使得本设计的质量进一步提高。感谢梁卓君学长分享他在硬件实现，工具使用方面的经验，帮助我解决遇到的困难。感谢李瑞学长在服务器平台使用方面的耐心解答，感谢组会上师兄师姐的提问，让我得以在设计初期就发现存在的问题。

然后，感谢贺光辉老师在大四上学期对我的学术论文投稿方面的大力支持，是他的耐心指导和细心修改使我的论文更上一个台阶，并最终得以发表。从研究起步到最终完成，贺老师用他严谨的学术作风，合理的进展规划，引导着我有序完成毕业设计。此外，本设计的完成还离不开贺老师在物质和精神上提供的无私帮助，包括良好的实验室环境，性能出色的设备，和舒缓压力的面对面交流，这些都为我提供了强大的前进动力。

最后要感谢父母的支持，他们从不给我施加压力而是毫无保留的鼓励我。感谢吴静学姐和朋友们在毕设期间对我的关心。感谢我列表里的歌曲，悲伤或是沮丧时，听歌让我得以调节心情，释放压力。“明天总要微笑吧”这句歌词一直激励着我期待明天的欢乐与希望，不被眼前的困难打倒。

RESEARCH ON MASSIVE MIMO DETECTION ALGORITHM AND CHIP IMPLEMENTATION

With the rapid development of the Internet, applications such as the Internet of Things and augmented reality have entered the public life, mobile data requirements will experience explosive growth, and the user experience of high transmission rate and low transmission delay, puts higher demands on wireless communication systems. The 5G (5^{th}-generation) mobile communication system supports peak transmission rates of over 10 Gbps and spectral efficiency of more than 100 bps/Hz, which can meet wireless communication requirements of users in various scenarios. One of the key technologies of 5G is the large-scale multiple-input multiple-output (MIMO) system. However, comparing massive MIMO systems with traditional MIMO systems, a key problem emerges that the complexity of the uplink detection algorithm and the downlink precoding algorithm increases dramatically for massive MIMO systems, which is difficult in hardware implementation. Considering the consistency of precoding and detection, two low-complexity detection algorithms with near-optimal performance are proposed in this thesis. Besides, high throughput VLSI (Very Large Scale Implementation) architecture is designed based on the proposed-2 algorithm to support various users MIMO systems. In addition, comparison with other algorithms in terms of hardware implementation and logic synthesis is provided.

Due to the channel hardening effect of massive MIMO systems, linear algorithms such as MMSE detection algorithms can achieve near-optimal performance with a lot of complexity reduced. However, the complexity is still too large for hardware implementation, which mainly comes from the calculation of the Gram matrix and its inversion. Widely researches have been made on reducing the complexity of matrix inversion. The basic idea is to transforming the problem of matrix inversion into the problem of solving the linear system of.

The Neumann series algorithm (NS) is first proposed with Neumann series expansion. However, when the number of iterations is small, the performance loss is large. When the number of iterations is greater than or equal to 3, the complexity even exceeds the exact inversion, which is not worth it[4]. The Richardson iterative algorithm (RI) introduces a relaxation factor to improve the performance, but a large number of iteration is required to obtain near-optimal performance[5]. The Gauss-Seidel (GS) method achieves near-optimal performance by decomposing the Gram matrix into an upper triangular matrix and a diagonal matrix. The CG algorithm update the estimation results with efficient search direction from the calculation of gradient. However, both methods introducing division operations, which is not friendly hardware implementation[6, 10]. Parallel Chebyshev iteration (PCI) approximates the Gram matrix inversion based on the Chebyshev polynomial, and achieves

near-optimal performance with a small number of iteration, the disadvantage of PCI is that its poor adaptability under different scale systems[8]. Joint algorithms are also explored by many researchers, such as SDJC (steepest descent, SD and Jacobi, JC) algorithm and NRI (Improved Newton iteration and Richardson iteration)[11, 12]. These algorithms intend to combine the merits of two methods to further improve the performance within smaller number of iteration. Both SD algorithm and Improved Newton algorithm provide efficient search direction by computing the gradient, especially for early iterations. And Richardson algorithm differs from Jacobi algorithm with relaxation factor, both of them employs simple iterative form. The SDJC algorithm converges fast in first iteration, unfortunately, it is unable to adapt in different scale MIMO systems and more practical correlation channels. For example, the SDJC algorithm does not converge anymore under the 128×32 MIMO system. Meanwhile, the complexity of the NRI algorithm grows exponentially with the number of Newton iterations.

In order to solve the problems encountered by the aforementioned algorithms, this thesis proposes two detection algorithms based on Non-Stationary Richardson iterative (proposed-1) and Second-Order Richardson (proposed-2) iterations, which significantly reduce the complexity while achieving near-optimal performance. With reasonable initialization method, the performance is further improved with complexity slightly increased. Optimizations specified at the characteristics of the channel hardening effect of the massive MIMO system, this thesis utilizes the eigenvalue estimation of the Gram matrix to calculate the parameter in iterative equation without additional complexity. Based on the above two optimization strategies, the computation of the Gram matrix is no longer necessary. By replacing the multiplication of the product of the Gram matrix and the iterative result with two-step matrix vector multiplication, the complexity is reduced by more than 25% when the iteration number is three.

Based on the MATLAB massive MIMO system simulation platform, the BER performance simulation and comparison of different algorithms are provided. The system model employs 64-QAM modulation, and rate 1/2 convolutional code with $[133_o, 171_o]$ polynomial. The simulation results show that for 128×16 massive MIMO system, the two proposed algorithms achieve near-optimal performance with 3 iterations under rayleigh channel and with 4 iteration under correlation model, respectively. In addition, the detailed complexity analysis is given, for example, when the number of iteration is three, the complexity of the algorithm is reduced by 45.9%, 26.7%, 26.25%, 27.1%, 25.6% comparing with NS, CG, GS, SDJC and NRI algorithm, respectively. Note that the proposed-1 algorithm consumes the lowest complexity among the algorithm mentioned in this thesis. As the previous two iteration results is utilized to update the iteration result $\mathbf{x}_k$, the proposed-2 algorithm is adaptive to different scale massive MIMO systems, such as 128×32 MIMO system. Besides, under the more realistic correlation channel model, algorithms proposed in this thesis still achieve near-optimal peformance within a small number of iterations. Convergence rate analysis of two proposed algorithms is given in this thesis, the proposed-2 algorithm converges faster than NS and CG algorihtm, especially in the first iteration.

Based on the proposed-2 algorithm, this thesis proposes a VLSI architecture that supports various number of users. The bit width for each step of data is determined through real number and fixed point simulation, and the consistency of the fixed point scheme is verified using MATLAB simulation platform. The hardware architecture design mainly includes the implementation of two matrix vector multiplication modules. By mapping the fixed number of base station antennas to the number of PE modules, and mapping the number of users to the change of time dimension, the variable number of users only affects the latency, which will not affect the throughput and hardware efficiency (throughput/LUTs). With reasonable data scheduling strategy, the waste of hardware resources caused by the variable number of users is avoided. Through the use of systolic array structure, high parallel computing, and pipeline strategy, high throughput and hardware efficiency is obtained.

In this thesis, the hardware design based on Second-Order Richardson detection algorithm is implemented using Verilog language. It is implemented on Xilinx Virtex-7 XC7VX980T FPGA (Field-Programmable Gate Array), and the high throughput rate of 1302Mb/s is achieved at 217M clock frequency. Hardware efficiency (Throughput/LUTs) achieves 18043, which is obviously superior to NS, CG, GS, PCI and SDJC algorithm, for example, having a improvement of 199% over the CG algorithm.

Using the Design Compiler tool, the hardware design proposed by this thesis is synthesized with the 40nm process, SMIC. The performance is further improved as it achieves a high throughput rate of 2.4 Gbps at a clock frequency of 400M, and the area is 3.6mm^2. The power consumption is only 0.41W at a voltage of 1.21V, thus the energy efficiency is 5.85 Gbps/W, and the area efficiency is 0.66 Gbps/mm^2.

On the basis of hardware implementation, the logic synthesis work is further completed with the Design Compiler tool and compared with the ASIC implementation results of different large-scale MIMO system detector. The detector based on Second-Order Richardson algorithm in this thesis achieves a throughput of 2.4 Gbps at a 400 MHz clock frequency. After normalizing the implementation results to 1.21V, 40nm process, comparing with Weji, NSand CG detector, the normalized energy efficiency of our work increased to $\mathbf{1.48}\times$, $\mathbf{15.1}\times$ and $\mathbf{1.76}\times$, respectively.

To sum up, the main contributions of this thesis lies in three aspects. 1) two detection algorithms are proposed based on Non-Stationary Richardson iteration and Second-Order Richardson iteration, respectively. These two algorithms achieve near-optimal performance while the complexity is significantly reduced, as over 25% complexity is reduced. Besides, the division operation is avoided of these two proposed algorithms. 2) three optimization strategies specified in massive MIMO systems are proposed to improve the performance and reduce the complexity, including low complexity initial solution, approximate eigenvalue estimation and the multiplication of Gram matrix with iteration results is replaced with 2-step multiplication of matrix and vector. 3) Completed High throughput and hardware efficiency VLSI architecture design of proposed-2 algorithm, which is able to support $128 \times U$ MIMO system and $U \in (8, 32)$. Synthesized the architecture design proposed in this thesis using the SMIC 40nm technology, high energy efficiency is obtained at no expense of area efficiency.

毕业设计期间发表和投稿的学术论文

[1] **Mengdan Lou**, Jieyu Li and Guanghui He, "AR-C3D: Action Recognition Accelerator for Human-Computer Interaction on FPGA," 2019 IEEE International Symposium on Circuits and Systems (ISCAS), Sapporo, Japan, 2019, pp. 1-4.

[2] **Mengdan Lou**, Jiaming Tu, Haibao Chen and Guanghui He, "A Low-Complexity Near-Optimal Detection Algorithm Based on Non-stationary Richardson Iteration for Massive MIMO Systems," Submitted to IEEE Wireless Communications Letters, 2019

[3] Jiaming Tu, **Mengdan Lou** and Guanghui He,"Highly Adaptable Low-complexity Signal Detection Method Based on Second-order Richardson Iteration for Uplink Massive MIMO Systems," Submitted to IEEE Transaction on Vehicular Technology, 2019