

基于软输出固定复杂度球形译码的高效迭代检测算法

沈弘* 赵春明

(东南大学移动通信国家重点实验室 南京 210096)

摘要: 该文在软输出固定复杂度球形译码(SFSD)算法的基础上,提出一种低复杂度高性能的MIMO迭代检测方法。该算法利用迭代过程中译码器的反馈信息更新SFSD检测算法的软输出,获得明显的迭代增益,并利用多级比特映射星座图的特点大大降低分支度量的运算次数。针对SFSD算法预处理复杂度较高的问题,该文将检测顺序调整和QR分解两个预处理步骤相结合,从而减少了矩阵求逆运算。在长期演进方案(LTE)下行链路环境中的仿真结果表明,该文提出的算法性能十分接近最优的最大后验概率(MAP)检测,并且实现复杂度相对于MAP有显著的下降。

关键词: 多输入多输出;软输出固定复杂度球形译码(SFSD);迭代检测;最大后验概率(MAP)检测;长期演进方案(LTE)

中图分类号: TN92

文献标识码: A

文章编号: 1009-5896(2012)07-1659-06

DOI: 10.3724/SP.J.1146.2011.00931

Efficient Iterative Detection Based on the Soft Fixed-complexity Sphere Decoder

Shen Hong Zhao Chun-ming

(National Mobile Communications Research Lab, Southeast University, Nanjing 210096, China)

Abstract: This paper proposes an efficient iterative detection algorithm for MIMO systems, based on the Soft Fixed-complexity Sphere Decoder (SFSD). The algorithm updates the soft output of SFSD using the feedback of the decoder, thus obtaining substantial iterative gain. The number of the branch metric calculation is reduced by exploiting the feature of the multi-level bit mapping constellation. The steps of the detection ordering determination and the QR decomposition are combined, in order to reduce the number of matrix inversion operations in the preprocessing stage. Simulation results in the Long Term Evolution (LTE) downlink system show that, the performance of the proposed algorithm approaches very close to that of the Maximum *A Posteriori* (MAP) detection with much lower complexity.

Key words: MIMO; Soft Fixed-complexity Sphere Decoder (SFSD); Iterative detection; Maximum *A Posteriori* (MAP) detection; Long Term Evolution (LTE)

1 引言

多输入多输出(MIMO)技术可以在不增加发射功率和带宽的前提下显著提高无线信道容量,已成为长期演进方案(LTE)等通信标准的物理层关键技术之一。空分复用是MIMO技术的一种重要形式,其基本思想是将多天线信道分解为多个独立的并行子信道,通过这些子信道上发送不同的数据流提高传输速率。由于空分复用系统会引入数据流之间的干扰,因而在接收端需要通过MIMO检测算法去

除空域上的干扰,恢复发送数据流。目前文献中已经提出各种MIMO检测算法,其中迭代检测^[1-4]是一种能够有效逼近MIMO信道容量的方案。最大后验概率(MAP)检测是最优的迭代检测算法,该方法需要遍历整个发送向量空间,因而在采用高阶调制或者数据流数较多的情况下,其复杂度非常高以至于难以实现。针对这个问题,文献[1]提出一种低复杂度的基于最小均方误差(MMSE)的软干扰抵消算法,但是其性能与MAP检测相比有一定的差距;文献[3]对传统的硬输出球形译码(SD)算法进行改进,提出一种列表球形译码(LSD)迭代检测算法,该算法的性能虽然接近MAP检测,但是其复杂度会随着噪声强度和信道条件的变化而改变,尤其在低信噪比或者信道条件差的情况下,复杂度仍然很高,因而不利于ASIC实现。文献[4]则在非迭代硬

2011-09-06 收到, 2012-05-08 改回

新一代宽带无线移动通信网科技重大专项(2010ZX03002-011, 2010ZX03001-002), 爱立信(中国)通信有限公司基金, 国家自然科学基金(61101087)和高等学校博士学科点专项科研基金(20110092120011)资助课题

*通信作者: 沈弘 shenhong@seu.edu.cn

输出 M 算法^[5]基础上提出一种迭代树搜索算法 (ITS), 其优点是计算复杂度恒定, 然而当采用高阶调制时, 该算法仍需要较多的排序运算, 并且某些比特的软输出值需要预先设定, 导致性能上的损失。最近, 文献[6,7]针对硬输出 SD 算法复杂度变化的缺点, 提出一种固定复杂度球形译码(FSD)算法, 该算法能以较低的复杂度获得近似硬输出 SD 算法的性能, 且不需要排序运算, 适合 ASIC 实现。文献[8]提出实数域上的 FSD 算法, 有效降低了高阶 MIMO 系统(如 8×8)的检测复杂度。在 FSD 的基础上, 文献[9]又提出一种软输出 FSD(SFSD)算法, 该方法通过比特翻转和路径添加两个步骤产生较高精度的软量。然而, SFSD 算法并没有利用译码器反馈的先验信息更新检测器的软输出, 因而其性能与 MAP 迭代检测算法相比仍有很大差距, 另外, 它的预处理复杂度较大, 需要多次求解矩阵的逆。

本文首先将 SFSD 算法推广到迭代检测中, 使其能够有效利用译码器反馈的软信息, 并利用多级比特映射星座图^[4]的特性大大减少分支度量的计算次数, 之后又将检测顺序调整和 QR 分解这两个预处理步骤进行结合, 在降低计算量的同时保证性能损失很小。最终的仿真结果表明, 本文提出的算法能够达到近似 MAP 的性能, 并且复杂度远远低于 MAP。

本文结构如下: 第 2 节给出系统模型。第 3 节介绍 FSD 和 SFSD 算法。第 4 节提出基于 SFSD 的高效迭代检测算法。第 5 节给出 LTE 系统下各种算法的仿真性能。第 6 节总结全文。文中 $(\bullet)^T$ 和 $(\bullet)^H$ 分别表示矩阵的转置和共轭转置, \mathbf{I}_N 表示 $N \times N$ 的单位矩阵, $\mathbf{0}_{M \times N}$ 表示 $M \times N$ 的零矩阵, $E(\bullet)$ 表示统计平均, $|\bullet|$ 和 $\|\bullet\|$ 表示复数的模和向量的欧几里德范数。

2 系统模型

图 1 给出了采用迭代接收机的 MIMO 系统框图。信息比特序列首先经过信道编码器和交织器, 由此产生的比特序列再被映射为符号序列(假设共有 $Q = 2^q$ 个星座符号, q 为调制阶数), 然后经过串并转换分为 N_T 个子数据流, 并从不同的天线上发射。接收端信号为

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{n} \quad (1)$$

其中 $\mathbf{s} = [s_1, \dots, s_{N_T}]^T$ 表示发送信号并满足 $E(\mathbf{s}\mathbf{s}^H) = \sigma_s^2 \mathbf{I}_{N_T}$; $\mathbf{n} = [n_1, \dots, n_{N_R}]^T$ 表示高斯白噪声且满足 $E(\mathbf{n}\mathbf{n}^H) = N_0 \mathbf{I}_{N_R}$ 。检测器利用 \mathbf{y} 和译码器反馈的先验信息 \mathbf{L}_{A1} 计算软量 \mathbf{L}_{D1} , 解交织后得到译码器的先验信息 \mathbf{L}_{A2} , 译码器利用 \mathbf{L}_{A2} 计算软量 \mathbf{L}_{D2} , 再通过

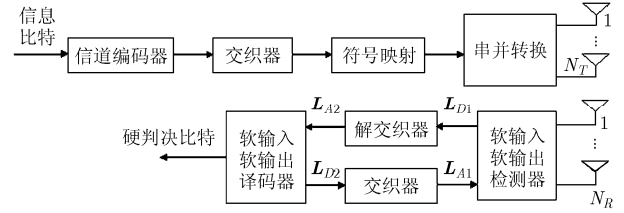


图 1 MIMO 迭代接收机系统框图

交织得到检测器的先验信息 \mathbf{L}_{A1} , 这样即构成一次迭代。当译码正确或达最大迭代次数时译码器输出硬判决结果。

最优的软输入软输出检测算法是 MAP 算法, 其输出软量可采用 Max-log 近似得到^[3]。

$$\begin{aligned} L_{D1}(s_n^k) = \max_{s \in \mathcal{S}_n^{k,0}} & \left\{ -\frac{1}{N_0} \|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2 + \frac{1}{2} \mathbf{d}^T \mathbf{L}_{A1} \right\} \\ & - \max_{s \in \mathcal{S}_n^{k,1}} \left\{ -\frac{1}{N_0} \|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2 + \frac{1}{2} \mathbf{d}^T \mathbf{L}_{A1} \right\} \\ & - L_{A1}(s_n^k) \end{aligned} \quad (2)$$

其中 s_n^k 表示符号 s_n 的第 k 个比特, $\mathbf{d}^T = [1 - 2s_1^1, \dots, 1 - 2s_1^q, \dots, 1 - 2s_{N_T}^1, \dots, 1 - 2s_{N_T}^q]$, $\mathbf{L}_{A1} = [L_{A1}(s_1^1), \dots, L_{A1}(s_1^q), \dots, L_{A1}(s_{N_T}^1), \dots, L_{A1}(s_{N_T}^q)]^T$, $\mathcal{S}_n^{k,0}$ 和 $\mathcal{S}_n^{k,1}$ 分别表示满足 $s_n^k = 0$ 和 $s_n^k = 1$ 的发送符号向量集合。由于 MAP 算法需要计算 Q^{N_T} 次 $\|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2$, 因此当数据流数较多或者调制阶数较高时, 复杂度会变得非常高。对于多输入多输出-正交频分复用(MIMO-OFDM)系统, 由于不同子载波的检测是独立进行的, 因而本节介绍的迭代接收方案可以在 MIMO-OFDM 系统每个子载波上直接使用。

3 FSD 和 SFSD 算法原理

FSD 算法本质上是一种基于宽度优先原则的树形检测算法。对于前 P 层数据流, 该算法采用遍历搜索的方法, 而对后 $N_T - P$ 层数据流则只保留一条局部最优路径。当 P 满足 $(N_R - N_T)(P + 1) + (P + 1)^2 \geq N_R$ 时, FSD 算法的性能与 SD 算法几乎一致(与非迭代 MAP 算法的性能也十分接近)。下面将介绍 FSD 算法的具体步骤。该算法首先通过以下步骤确定检测顺序并置换信道矩阵 \mathbf{H} 的列向量:

(1) 令 $i = 1$, $\mathbf{H}_1 = \mathbf{H}$, $\mathbf{T}_1 = \{1, 2, \dots, N_T\}$ 。

(2) 计算 $\mathbf{G}_i = (\mathbf{H}_i^H \mathbf{H}_i)^{-1} \mathbf{H}_i^H$ 及其各行的模平方, 当 $i \leq P$ 时, 找出模平方最大的行向量 $(\mathbf{G}_i)_{k_i}$; 当 $i > P$ 时, 找出模平方最小的行向量 $(\mathbf{G}_i)_{k_i}$ 。

(3) 交换 \mathbf{H} 第 $\mathbf{T}_i(k_i)$ 和 $N_T - i + 1$ 列; 删除 \mathbf{H}_i 第 k_i 列得 \mathbf{H}_{i+1} ; 删除 \mathbf{T}_i 第 k_i 个元素得 \mathbf{T}_{i+1} 。

(4) 如果 $i = N_T - 1$, 则此时 \mathbf{H}_{i+1} 只包含一列, \mathbf{T}_{i+1} 只剩余一个元素, 表示已经完成检测顺序的调

整；否则 i 的值增加 1，并跳转回(2)进入下一轮循环。

将按照上述步骤得到的信道矩阵记作 $\tilde{\mathbf{H}}$ ，则接收信号可以写为

$$\mathbf{y} = \tilde{\mathbf{H}}\tilde{\mathbf{s}} + \mathbf{n} \quad (3)$$

其中 $\tilde{\mathbf{s}} = [\tilde{s}_1, \dots, \tilde{s}_{N_T}]^T = [s_{T_{N_T}(k_{N_T})}, \dots, s_{T_1(k_1)}]^T$ 。对 $\tilde{\mathbf{H}}$ 进行 QR 分解： $\tilde{\mathbf{H}} = \mathbf{Q}\mathbf{R}$ ，其中 \mathbf{Q} 是酉矩阵， \mathbf{R} 是上三角阵。令 $\mathbf{r} = \mathbf{Q}^H\mathbf{y}$ ，并将 $\mathbf{y} = \tilde{\mathbf{H}}\tilde{\mathbf{s}} + \mathbf{n}$ 代入可得

$$\mathbf{r} = \mathbf{R}\tilde{\mathbf{s}} + \mathbf{Q}^H\mathbf{n} = \begin{bmatrix} R_{1,1} & \cdots & R_{1,N_T} \\ & \ddots & \vdots \\ & & R_{N_T,N_T} \end{bmatrix} \begin{bmatrix} \tilde{s}_1 \\ \vdots \\ \tilde{s}_{N_T} \end{bmatrix} + \mathbf{n}' \quad (4)$$

其中 \mathbf{n}' 和 \mathbf{n} 同分布。根据式(4)可定义符号 \tilde{s}_i ($i = N_T, \dots, 1$) 对应分支度量 u_i 和路径度量 w_i 为

$$u_i = \left| r_i - \sum_{j=i}^{N_T} R_{i,j} \tilde{s}_j \right|^2 \quad (5)$$

$$w_i = \sum_{j=i}^{N_T} u_j \quad (6)$$

显然，总路径度量值 $w_1 = \|\mathbf{r} - \mathbf{R}\tilde{\mathbf{s}}\|^2$ 。FSD 算法依次对 $\tilde{s}_{N_T}, \dots, \tilde{s}_{N_T-P+1}$ 这 P 层数据流进行遍历搜索，而对于剩余各层，则保留一个使得当前分支度量 u_i 最小的符号 \tilde{s}_i ，容易求出

$$\tilde{s}_i = Q \left[\left| r_i - \sum_{j=i+1}^{N_T} R_{i,j} \tilde{s}_j \right| / R_{i,i} \right], \quad i = N_T - P, \dots, 1 \quad (7)$$

其中 $Q[\bullet]$ 表示星座图上的硬判决运算。将最终产生的路径集合记作 \mathbf{L}_{FSD} ，则硬判决输出为总路径度量值最小的路径：

$$\hat{\mathbf{s}} = \arg \min_{\tilde{\mathbf{s}} \in \mathbf{L}_{\text{FSD}}} \|\mathbf{r} - \mathbf{R}\tilde{\mathbf{s}}\|^2 \quad (8)$$

由于 FSD 算法只遍历前 P 层，使得有些路径上某些比特位只有 0 或 1，不利于计算输出软量。SFSD 算法在 \mathbf{L}_{FSD} 的基础上添加路径，保证所有比特位软量都可以计算，过程如下：

(1) 对 \mathbf{L}_{FSD} 中路径按度量值排序，即 $\|\mathbf{r} - \mathbf{R}\tilde{\mathbf{s}}^{(1)}\|^2 \leq \dots \leq \|\mathbf{r} - \mathbf{R}\tilde{\mathbf{s}}^{(Q^P)}\|^2$ ； $i=1$ ，最大循环次数为 N_{SFSD} 。

(2) 在第 i 次循环时，首先依次翻转符号 $\tilde{s}_{N_T-P}^{(i)}$ 的各个比特，得到 q 个新增的符号，再按照式(7)分别扩展出 q 条完整的路径；对于符号 $\tilde{s}_{N_T-P-1}^{(i)}, \dots, \tilde{s}_1^{(i)}$ 按照相同的方法产生新路径，这样第 i 次循环一共产生 $q(N_T - P)$ 条路径。

(3) 如果 $i = N_{\text{SFSD}}$ ，则结束 SFSD 算法的树搜索过程，否则跳转回上一步。

用 \mathbf{L}_{SFSD} 表示 SFSD 算法产生的路径集合(包括 FSD 算法产生的 Q^P 条路径)，则最终的比特软量可以采用如下公式计算：

$$L(\tilde{s}_n^k)$$

$$= \frac{1}{N_0} \left\{ \min_{\tilde{\mathbf{s}} \in \mathbf{L}_{\text{SFSD}} \cap \mathcal{S}_n^{k,1}} \|\mathbf{r} - \mathbf{R}\tilde{\mathbf{s}}\| - \min_{\tilde{\mathbf{s}} \in \mathbf{L}_{\text{SFSD}} \cap \mathcal{S}_n^{k,0}} \|\mathbf{r} - \mathbf{R}\tilde{\mathbf{s}}\| \right\} \quad (9)$$

其中 \tilde{s}_n^k 表示符号 \tilde{s}_n 的第 k 个比特。这里我们采用欧几里德范数计算软量，可以有效地提高软量精度，该方法最早在文献[10]中提出。

文献[6,7]通过理论分析和仿真指出硬输出 FSD 算法能以很低的复杂度获得近似最优硬输出的性能，其根本原因是 FSD 能根据不同数据流信噪比自适应的调整搜索方式。文献[9]则指出，SFSD 算法在 FSD 产生的符号向量集合基础上，通过比特翻转和添加路径两个步骤得到一个新的集合 \mathbf{L}_{SFSD} ，第一可以保证集合 \mathbf{L}_{SFSD} 中不会发生某个比特位只存在‘0’或者‘1’的情况，第二能确保新增加的符号向量与近似最优的 $\hat{\mathbf{s}}_{\text{FSD}}$ 尽可能地接近，从而使得检测器输出的软量十分精确且接近于最优软输出。

4 基于 SFSD 算法的迭代检测算法

SFSD 算法虽然能产生较高精度软量，但是并没有利用译码器先验信息，因而不能获得迭代增益，此外该算法预处理需要较多的运算量。本文将针对这两个问题分别进行改进。

4.1 软输入软输出的 SFSD 算法

当检测器先验信息 $\mathbf{L}_{\text{AI}} \neq 0$ 时，分支度量 u_i 和路径度量 w_i 需考虑 \mathbf{L}_{AI} 的作用，定义如下：

$$u_i = -\frac{1}{N_0} \left| r_i - \sum_{j=i}^{N_T} R_{i,j} \tilde{s}_j \right|^2 + \frac{1}{2} \mathbf{d}_i^T \mathbf{L}_{\text{AI},i} \quad (10)$$

$$w_i = \sum_{j=i}^{N_T} u_j \quad (11)$$

其中， $\mathbf{d}_i^T = [1 - 2\tilde{s}_i^1, \dots, 1 - 2\tilde{s}_i^q]$ ， $\mathbf{L}_{\text{AI},i} = [L_{\text{AI}}(\tilde{s}_i^1), \dots, L_{\text{AI}}(\tilde{s}_i^q)]^T$ 。

对于 $\tilde{s}_{N_T}, \dots, \tilde{s}_{N_T-P+1}$ 这 P 层数据流，由于进行遍历搜索，保留的路径和 SFSD 算法是一致的，唯一的区别在于需要根据式(10)和式(11)计算分支度量和路径度量。而对于数据流 $\tilde{s}_{N_T-P}, \dots, \tilde{s}_1$ ，除了度量计算的方法不同以外，SFSD 算法中寻找局部最优路径的方法，即式(7)不再有效。最直接的方法是计算出所有分支的度量值，并找出其中的最小值：

$$\hat{s}_i = \arg \max_{\tilde{s}_i \in \Omega} \left\{ -\frac{1}{N_0} \left| r_i - R_{i,i} \tilde{s}_i - \sum_{j=i+1}^{N_T} R_{i,j} \tilde{s}_j \right|^2 + \frac{1}{2} \mathbf{d}_i^T \mathbf{L}_{\text{AI},i} \right\} \quad (12)$$

其中 Ω 是星座符号集合。本文提出的迭代 SFSD 算法在搜索发送符号向量集合时使用的方法和非迭代

SFSD 算法是一致的, 包括 FSD 算法、比特翻转以及路径添加 3 部分。与 SFSD 算法的区别在于重新定义了计算软量时使用的路径度和分支度量。根据之前对 SFSD 算法的分析可以推断, 迭代 SFSD 算法同样能产生十分精确的软量, 并且由于有效地利用译码器反馈的信息, 其性能接近于 MAP 迭代检测, 事实上仿真结果也验证了这一结论。

将式(12)与式(7)进行对比可以发现, 由于引入了先验信息, 此时无法采用简单的符号硬判决运算, 这就导致对于调制阶数较高的情况(如 16QAM), 需要计算很多次分支度量值, 显然对于实现是不利的。文献[4]利用多级比特映射星座图的特性, 有效地简化了迭代树搜索算法中的度量计算, 下面将采用文献[4]中的方法降低式(12)的计算复杂度。

图 2 中的实心圆表示采用多级比特映射的 16 QAM 星座图, 其中每个星座符号都用 4 个比特表示。所谓多级比特映射的含义是: 前两个比特代表了该符号所在的象限, 例如“00”表示第 1 象限; 而后两个比特则确定了该符号在象限中的位置, 这样只需要两步搜索即可确定符号的具体位置(对于 64 QAM 星座图则需要 3 步搜索)。下面仍以 16 QAM 为例介绍如何利用多级比特映射减少搜索局部最优路径的运算量。图 2 中的空心圆表示每个象限内星座符号的中心, 记作 $s_{C1}, s_{C2}, s_{C3}, s_{C4}$ 。首先将这 4 个符号值分别代入式(10)计算分支度量:

$$u_{i,k} = -\frac{1}{N_0} \left| r_i - R_{i,i} s_{Ck} - \sum_{j=i+1}^{N_T} R_{i,j} \tilde{s}_j \right|^2 + \frac{1}{2} \mathbf{d}_{i,k}^T \tilde{\mathbf{L}}_{A1,i}, \quad k = 1, 2, 3, 4 \quad (13)$$

其中 $\mathbf{d}_{i,1}^T = [1, 1], \mathbf{d}_{i,2}^T = [1, -1], \mathbf{d}_{i,3}^T = [-1, 1], \mathbf{d}_{i,4}^T = [-1, -1], \tilde{\mathbf{L}}_{A1,i} = [L_{A1}(\tilde{s}_i^1), L_{A1}(\tilde{s}_i^2)]^T$ 。假设 $u_{i,2}$ 最大, 则根据式(10)可计算出 s_{C2} 所在象限 4 个符号的分支度量, 其中最大度量值对应符号即 \hat{s}_i 。

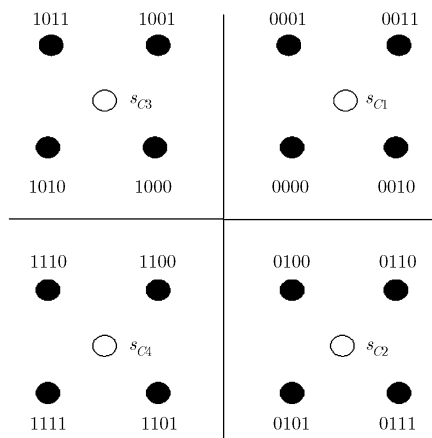


图 2 多级比特映射的 16 QAM 星座图

显然对于 16QAM 调制, 采用该简化方法求解一次 \hat{s}_i 只需 8 次分支度量运算, 而式(12)需要计算 16 次分支度量。以 4×4 MIMO 为例, 可统计出 $N_{\text{SFSD}} = 3$ 时需计算 84 次 \hat{s}_i , 这样共可以节省 $8 \times 84 = 672$ 次分支度量运算。从前面给出的例子可以看到, 这种多级搜索方案首先判断 \hat{s}_i 所在象限, 然后在象限中进一步搜索, 可以精确确定 \hat{s}_i 所在位置, 因而几乎不造成性能损失, 最终仿真结果也表明它对性能影响很小, 因而是一种有效的简化方法。

4.2 简化的预处理步骤

SFSD 算法的预处理包括调整检测顺序和 QR 分解这两个步骤, 其中调整检测顺序需要 $N_T - 1$ 次矩阵求逆运算, 当数据流较多时计算量很大。事实上, SFSD 算法对最后 $N_T - P$ 层数据流的检测顺序调整和 V-BLAST 检测^[11]一致, 即首先检测信噪比最高的数据流。为避免 V-BLAST 算法的矩阵求逆运算, 文献[12]曾提出一种 SQRD 检测方法, 其基本思想是在 QR 分解同时确定检测顺序。下面我们将该方法推广到 SFSD 算法预处理中, 具体步骤如下:

(1) 令 $i = 1, \mathbf{H}_1 = \mathbf{H}, \mathbf{S} = \mathbf{T}_1 = \{1, 2, \dots, N_T\}$ 。

(2) 计算 $\mathbf{G}_i = (\mathbf{H}_i^H \mathbf{H}_i)^{-1} \mathbf{H}_i^H$, 并找出模平方最大的行向量 $(\mathbf{G}_i)_{k_i}$ 。

(3) 交换 \mathbf{H} 的第 $\mathbf{T}_i(k_i)$ 列和第 $N_T - i + 1$ 列; 交换 \mathbf{S} 的第 $\mathbf{T}_i(k_i)$ 个元素和第 $N_T - i + 1$ 个元素; 删除 \mathbf{H}_i 的第 k_i 列, 得到 \mathbf{H}_{i+1} ; 删除 \mathbf{T}_i 的第 k_i 个元素, 得到 \mathbf{T}_{i+1} 。如果 $i = P$, 则进行下一步操作, 否则 i 增加 1 并跳转回上一步。

(4) 令 $i = 1, \mathbf{Q} = \mathbf{H}, \mathbf{R} = \mathbf{0}_{N_T \times N_T}$ 。

(5) 当 $i \leq N_T - P$ 时, 按照式(14)求解 k_i :

$$k_i = \arg \min_{l=i, \dots, N_T-P} \|\mathbf{q}_l\|^2 \quad (14)$$

其中 \mathbf{q}_l 表示 \mathbf{Q} 的第 l 列; 分别交换 \mathbf{Q} 和 \mathbf{R} 的第 i 和 k_i 列, 交换 \mathbf{S} 的第 i 和 k_i 个元素。

(6) 采用 Gram-Schmidt 正交化方法更新矩阵 \mathbf{Q} 和 \mathbf{R} , 包括以下几步:

(a) $r_{i,i} = \|\mathbf{q}_i\|$;

(b) $\mathbf{q}_i = \mathbf{q}_i / r_{i,i}$;

(c) $r_{i,l} = \mathbf{q}_i^H \mathbf{q}_l, \mathbf{q}_l = \mathbf{q}_l - r_{i,l} \mathbf{q}_i, l = i + 1, \dots, N_T$ 。

(7) 如果 $i = N_T$, 则完成预处理, 否则 i 增加 1 并跳转回(5)。

该简化方法保证前 P 层数据流的检测顺序和 SFSD 算法相同, 而后面 $N_T - P$ 层数据流的检测顺序则由 SQRD 算法确定, 和原来的预处理方法相比, 不但可以省去 $N_T - P - 1$ 次矩阵求逆, 而且性能损失很小。这主要是因为 SQRD 算法能保证 QR 分解

后得到的上三角阵 R 满足如下性质：较小的对角元在左上方而较大的对角元在右下方，又由于较大对角元对应的数据流具有较高的信噪比^[12]，因而信噪比较大数据流可以先被检测。虽然这种检测顺序调整方式不完全等价于下文算法 1 中多次求逆的方法，但基本思想是一致的，即减少误差传播影响。

5 算法性能仿真

本节将给出 LTE 下行链路环境中不同算法的性能，表 1 列出了仿真中采用的系统参数。

表 1 LTE 下行链路仿真参数

带宽	1.4 MHz
子载波数	72
信道模型	EVA70 ^[13]
天线相关性	低相关 ^[13]
MIMO 工作模式	开环空分复用
天线配置	$2 \times 2, 4 \times 4, 6 \times 6$
数据流数	2,4,6
调制方式	QPSK, 16QAM
码字个数	2
信道编码及码率	Turbo 码 ^[14] , 1/2 码率

考虑 LTE 下行链路中 3 种天线配置： 2×2 、 4×4 和 6×6 。通过仿真发现 3 种天线配置下较为理想的 N_{SFSD} 分别为 1, 3 和 6。下文“基于 SFSD 的迭代算法 1”是利用式(12)寻找每层局部最优路径，并采用原始 SFSD 算法的预处理；“基于 SFSD 的迭代算法 2”是在算法 1 基础上，利用多级比特映射星

座图特性减少分支度量计算并对简化预处理。

图 3 比较了 2×2 MIMO, 16QAM 调制情况下两种基于 SFSD 的迭代算法误帧率性能。从图中可以看到，基于 SFSD 的迭代算法 2 和算法 1 的性能几乎无区别，这表明算法 2 采用的简化是合理的。此外两种基于 SFSD 的迭代算法在第 1 次迭代时能获得 0.5 dB 性能增益(误帧率 0.1)，第 2 次迭代时增益变小，说明迭代趋于收敛。与 MAP 算法相比，不进行迭代时基于 SFSD 的迭代算法 2 约有 0.2 dB 的损失(误帧率 0.1)，而对于迭代一次和两次则几乎无损失。

图 4 比较了 4×4 MIMO, 16QAM 调制情况下两种基于 SFSD 的迭代算法性能。从图中可以看到，基于 SFSD 的迭代算法 2 的误帧率十分接近算法 1，这表明算法 2 的简化处理对性能影响很小。此外，两种基于 SFSD 的迭代算法在第 1 次迭代时即可获得 0.7 dB 的增益(误帧率 0.1)，第 2 次迭代获得增益变小，说明迭代趋于收敛。与 MAP 算法相比，不进行迭代时基于 SFSD 的迭代算法 2 约有 0.25 dB 的性能损失(误帧率 0.1)，迭代 1 次时损失为 0.1 dB(误帧率 0.1)。当迭代次数为 2 时损失变得更小。

图 5 比较了 6×6 MIMO, QPSK 调制情况下两种基于 SFSD 的迭代算法与 MAP 算法的性能。从结果中可以看到，基于 SFSD 的迭代算法 2 的误帧率十分接近算法 1，这表明即使在高阶 MIMO 情况下，算法 2 的简化处理也是有效的。在不进行迭代时基于 SFSD 的迭代算法 2 相对于 MAP 算法约有 0.25 dB 的性能损失(误帧率 0.1)，而迭代 1 次时，性能损失则减小为 0.1 dB(误帧率 0.1)，两次迭代情况下损失变得很小。

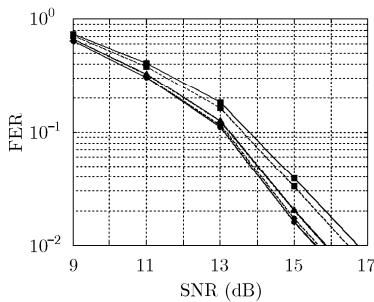


图 3 基于 SFSD 的迭代算法性能 ($N_T = N_R = 2$, 16QAM 调制)

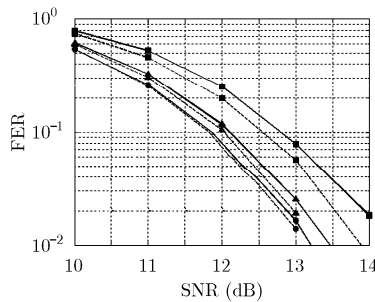


图 4 基于 SFSD 的迭代算法性能 ($N_T = N_R = 4$, 16QAM 调制)

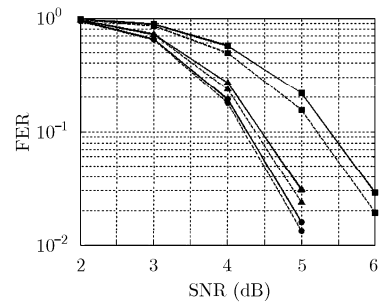


图 5 基于 SFSD 的迭代算法与 MAP 算法性能比较 ($N_T = N_R = 6$, QPSK 调制)

表2列出了MAP检测和两种基于SFSD的迭代算法的计算复杂度(单个子载波)。从表中可以发现,在 2×2 MIMO, 16QAM调制下,基于SFSD的迭代算法1计算量相对于MAP算法没有显著的减少,而算法2的计算量相对于算法1有明显下降,需要实数乘法数和实数加法数约为MAP算法的 $1/2$ 和 $1/3$ 。在 4×4 MIMO, 16QAM调制下,基于SFSD的迭代算法1和算法2的计算量均远小于MAP算法,且算法2复杂度最低。对于 6×6 MIMO, QPSK调制,两种基于SFSD迭代算法计算量远小于MAP算法,且算法2具有更低复杂度。

表2 算法复杂度比较(单个子载波)

算法名称	2×2 MIMO		4×4 MIMO		6×6 MIMO	
	16QAM		16QAM		QPSK	
	实数乘法	实数加法	实数乘法	实数加法	实数乘法	实数加法
MAP算法	1536	3223	590848	2621999	53824	344387
SFSD迭代算法1	1452	2024	7960	10876	4808	6614
SFSD迭代算法2	768	1028	4364	5666	4600	6444

6 结束语

本文提出了一种基于SFSD算法的MIMO迭代检测方案。该方法首先将译码器反馈的先验信息引入分支度量和路径度量的定义中,使得检测器软输出的精度在每次迭代过程中不断增加,获得明显的迭代增益。之后,该方案又利用多级比特映射星座图的特点,大大减少分支度量的计算次数。另外,本文还将调整检测顺序和QR分解两步相结合,减少了SFSD算法预处理中的矩阵求逆次数。在LTE下行链路环境中的仿真结果表明,本文的算法能够获得近似MAP算法的性能,并且复杂度远低于MAP。另外本文提出的算法同样适用于调制阶数更高(如64QAM)的情况,也可以应用于发送数据流更多的MIMO系统。

参考文献

- [1] Sellathurai M and Haykin S. Turbo-BLAST for wireless communications: theory and experiments [J]. *IEEE Transactions on Signal Processing*, 2002, 50(10): 2538-2546.
- [2] Haykin S, Sellathurai M, De Jong Y, et al. Turbo-MIMO for wireless communications [J]. *IEEE Communications Magazine*, 2004, 42(10): 48-53.
- [3] Hochwald B M and Ten Brink S. Achieving near-capacity on a multiple-antenna channel [J]. *IEEE Transactions on Communications*, 2003, 51(3): 389-399.
- [4] de Jong Y and Willink T J. Iterative tree search detection for MIMO wireless systems [J]. *IEEE Transactions on Communications*, 2005, 53(6): 930-935.
- [5] Yue J, Kim K J, Gibson J D, et al. Channel estimation and data detection for MIMO-OFDM systems [C]. Proceedings of IEEE GLOBECOM, San Francisco, USA, Dec. 2003: 581-585.
- [6] Barbero L G and Thompson J S. Fixing the complexity of the sphere decoder for MIMO detection [J]. *IEEE Transactions on Wireless Communications*, 2008, 7(6): 2131-2142.
- [7] Jaldén J, Barbero L G, Ottersten B, et al. The error probability of the fixed-complexity sphere decoder [J]. *IEEE Transactions on Signal Processing*, 2009, 57(7): 2711-2720.
- [8] Zheng C, Chu X, McAllister J, et al. Real-valued fixed-complexity sphere decoder for high dimensional QAM-MIMO systems [J]. *IEEE Transactions on Signal Processing*, 2011, 59(9): 4493-4499.
- [9] Barbero L G, Ratnarajah T, and Cowan C. A low-complexity soft MIMO detector based on the fixed-complexity sphere decoder [C]. Proceedings of IEEE ICASSP, Las Vegas, USA, March 2008: 2669-2672.
- [10] Kawai H, Higuchi K, Maeda N, et al. Likelihood function for QR-MMLD suitable for soft-decision Turbo decoding and its performance for OFCDM MIMO multiplexing in multipath fading channel [J]. *IEICE Transactions on Communications*, 2005, E88-B(1): 47-57.
- [11] Foschini G J. Layered space-time architecture for wireless communication in a fading environment when using multiple antennas [J]. *Bell Labs Technical Journal*, 1996, 1(2): 41-59.
- [12] Wübben D, Böhneke R, Rinas J, et al. Efficient algorithm for decoding layered space-time codes [J]. *Electronics Letters*, 2001, 37(22): 1348-1350.
- [13] 3GPP TSG-RAN, TS 36.101 v9.8.0. User Equipment (UE) radio transmission and reception (Release 9) [S]. 2011.
- [14] 3GPP TSG-RAN, TS 36.212 v9.3.0. Multiplexing and channel coding (Release 9) [S]. 2010.

沈弘: 男, 1987年生, 博士生, 研究方向包括MIMO检测、LTE物理层技术以及信道编码技术等。

赵春明: 男, 1959年生, 教授, 博士生导师, 移动通信国家重点实验室副主任, 从事未来无线与移动通信技术和新型信道纠错编译码技术的研究。