

MIMO 系统中一种自适应的迭代检测译码算法

韩湘 赵丹 魏急波 王建新
(国防科技大学电子科学与工程学院 长沙 410073)

摘要:针对常用的非穷尽列表形式后验概率检测算法直接采用恒定且较大的列表长度,导致列表冗余度大的问题,该文提出了一种自适应长度的列表球形译码算法(Adaptive Size List Sphere Decoding, ASLSD)。在算法中通过更新检测半径和设置停止条件,使检测列表长度可随信噪比和迭代次数自适应变化。而且通过将列表操作与 LSD (List Sphere Decoding)检测相结合,避免了符号序列在不同半径下的重复检测。仿真表明,在较小性能损失的前提下,该算法可以大大减小所需检测列表的长度,进而有效降低接收机的复杂度。

关键词:列表球形译码;最大后验概率检测;度量

中图分类号:TN92

文献标识码:A

文章编号:1009-5896(2008)03-0625-05

An Adaptive Iterative Detection and Decoding Algorithm for MIMO Systems

Han Xiang Zhao Dan Wei Ji-bo Wang Jian-xin

(Dept of Electronic Science and Engineering, National University of Defence Technology, Changsha 410073, China)

Abstract: To further reduce the redundancy of the existing non-exhaustive list MAP detection algorithms, which is induced by defining a fixed and large list size, an Adaptive Size List Sphere Decoding (ASLSD) algorithm is proposed. Through updating radius and setting stop criterion, the resulted detection list has a variable length which is adaptive with the SNR and the iteration. Moreover, by combining LSD with a list, the repeated detection with different radius is avoided. Simulation shows that with slight loss in performance, the proposed algorithm leads to a much shortened detection list, which means a simplified receiver.

Key Words: List Sphere Decoding (LSD); MAP detection; Metric

1 引言

后验概率检测是 MIMO 系统中迭代检测译码算法的重要组成部分。但在 MIMO 系统中,基于穷尽形式列表的后验概率检测算法其复杂度随发送天线数和调制阶数的增加呈指数上升。为了降低 MAP 检测算法的复杂度,近年来提出了基于非穷尽形式列表的 APP 检测算法,主要包括:列表球形译码(List Sphere Decoding, LSD)^[1-3]和列表序列检测(LISt-Sequential, LISS)^[4]。在这两种算法中,列表长度是决定系统性能和复杂度的重要参数。但是目前这两种算法均直接设定了恒定且较大的列表长度,导致检测复杂度增加。实际上,发送符号后验概率的取值将随 SNR 和迭代次数的增加逐渐集中于少量符号序列。这时如仍采用与低 SNR 或初始迭代时相同的列表长度,将使列表中引入大量后验概率取值很小的符号序列,增加了接收机的复杂度。

针对该问题,本文提出了一种自适应长度的列表球形译码算法(ASLSD)。该算法的优点在于:(1)以逐渐增大的半径检测符号序列,直至满足停止条件,因此所得检测列表长度

可随信噪比和迭代次数自适应变化。(2)将列表操作与 LSD 检测相结合,避免了符号序列在不同半径下的重复检测。仿真表明,在较小的性能损失前提下,该算法可以显著减小检测列表长度,从而有效降低接收机的复杂度,获得更优的性能和复杂度的折中。

2 系统模型

假设 MIMO 系统采用 N_t 个发送天线和 N_r 个接收天线,且 $N_t = N_r$ 。二进制信息比特 b 先经过编码、交织,形成编码比特 $\mathbf{c} = (c_1 \cdots c_{N_t})^T$, 其中 $\mathbf{c}_i = (c_{i1} \cdots c_{iM})$, $M = \log_2 K$, K 为调制的星座点数。编码比特经映射后,得到发送符号序列,表示为 $\mathbf{x} = (x_1 \cdots x_{N_t})^T$, 其中 $x_i = \text{map}(c_i)$ 。发送符号通过信道后,如图 1 所示,第 n 时刻接收信号的形式为

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

其中 $\mathbf{y}(n)$ 为 $N_r \times 1$ 的接收信号矢量, $\mathbf{H} = [h_{ij}]$ 为 $N_r \times N_t$ 复信道矩阵。 $\mathbf{x}(n)$ 为 $N_t \times 1$ 的发送信号矢量。 $\mathbf{v}(n) \sim N(0, \sigma_v^2)$ 为复高斯白噪声。这里假定接收端已知信道响应。

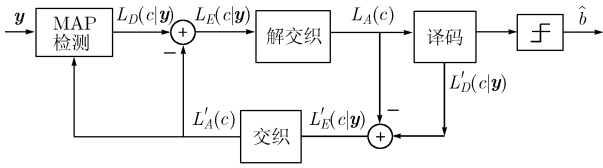


图1 基于迭代检测译码的MIMO系统接收机结构

依据发送信号的先验信息和接收信号，MAP检测器计算各编码比特的外信息，以对数似然比的形式表示为 $L_E(c_{ij} | \mathbf{y})$ 。设编码比特取值为 $\{-1, 1\}$ ，则 $L_E(c_{ij} | \mathbf{y})$ 可以表示为

$$L_E(c_{ij} | \mathbf{y}) = L_D(c_{ij} | \mathbf{y}) - L_A(c_{ij}) = \ln \frac{p(\mathbf{y} | c_{ij} = 1)}{p(\mathbf{y} | c_{ij} = -1)} \quad (2)$$

$$L_A(c_{ij}) = \ln \frac{p(c_{ij} = 1)}{p(c_{ij} = -1)}$$

$$L_D(c_{ij} | \mathbf{y}) = \ln \frac{\sum_{x, c_{ij}=1} p(\mathbf{y} | \mathbf{x}) \prod_k p(x_k)}{\sum_{x, c_{ij}=-1} p(\mathbf{y} | \mathbf{x}) \prod_k p(x_k)}$$

$$p(\mathbf{y} | \mathbf{x}) = \frac{1}{(\pi\sigma_v^2)^{N_r}} \exp\left\{-\frac{1}{\sigma_v^2} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2\right\}$$

将外信息 $L_E(c_{ij} | \mathbf{y})$ 解交织后作为编码比特的先验信息 $L_A(c)$ 送入APP(A Posteriori Probabilities)译码器。APP译码器在产生硬判决译码的同时，提供编码比特的外信息 $L'_E(c' | \mathbf{y})$ 。该外信息经交织后，作为更新后的编码比特先验信息 $L'_A(c)$ 送入检测器，完成一次迭代过程。

这里，基于Fincke-Pohst球形译码思想^[5]和式(2)，定义符号序列 \mathbf{x} 与接收信号 \mathbf{y} 的度量为

$$M(\mathbf{x}) = \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 - \sigma_v^2 \sum_k \ln p(x_k) \\ = \|\mathbf{z} - \mathbf{R}\mathbf{x}\|^2 - \sigma_v^2 \sum_k \ln p(x_k) = \sum_{k=1}^{N_t} M_k(\mathbf{x}) \quad (3)$$

其中 $\mathbf{H} = \mathbf{Q}\mathbf{R}$ ， \mathbf{Q} 为酉矩阵， \mathbf{R} 为上三角矩阵， $\mathbf{z} = \mathbf{Q}^H \mathbf{y}$ 。 $(\cdot)^H$ 表示共轭转置。第 k 层的度量为

$$M_k(\mathbf{x}) = \left\| z_k - r_{k,k}x_k - \sum_{j=k+1}^{N_t} r_{k,j}x_j \right\|^2 - \sigma_v^2 \ln p(x_k)。$$

3 Adaptive Size-LSD 检测算法

3.1 ASLSD 算法流程

ASLSD算法主要包括3个步骤：

(1)搜索 给定半径 r_0 ，搜索所有度量小于 r_0 的路径，同时将检测到的状态(包括比特取值和其对应的度量值)记录在列表中。

(2)验证停止条件 定义候选路径为度量小于 r_0 的完整路径。验证搜索到的候选路径是否满足停止条件。如满足则终止。

(3)半径更新 如不满足停止条件，则参照停止条件和检测状态给出一个扩大的半径 $r(r_0 < r)$ ，沿着当前列表所保存的状态开始新的搜索。交替进行搜索和验证，半径更新3个

步骤，直到满足停止条件。最终基于所得列表计算后验信息。

由于ASLSD算法是基于递增的半径，为避免符号序列在不同半径下的重复检测必须保留已检测到的状态，因此基于列表进行检测。在ASLSD算法中共需建立两个列表，一个是路径列表(path_list)，用于存储检测过程中各路径的状态，另一个是候选列表(cand_list)，用于存储候选路径。ASLSD算法的搜索过程即主要基于路径列表实现，其具体步骤为

步骤1 (初始化)设置检测半径为 r ，path_list和cand_list为空，检测层数 $k = N_t$

步骤2 计算本层内所有分支的度量值，将其比特取值与对应的度量值存入path_list

while ($k \geq 1$) do:

步骤3 检测第 k 层各分支的度量值是否大于 r ，如小于 r ，置其标记位为0；如大于 r ，则在其标记位填入 r ，并将其移至堆栈的末尾。

步骤4 如果 $k=1$ ，转至步骤6；否则 $k=k-1$ ；

步骤5 扩展堆栈中第 $k+1$ 层标记位为0的分支，即用其扩展后的 M 个分支替代原来在堆栈中的分支。

End

步骤6 将候选路径从path_list中移入cand_list。

从以上步骤可以看出，虽然ASLSD与LISS均基于列表操作，但LISS算法每次仅扩展Top分支，因此需要在整个列表中根据度量值排序以确定Top分支，而ASLSD算法同时扩展本层内满足半径限制的所有分支，因此只需将本层内各分支的度量值与半径相比较即可。相比之下，ASLSD算法具有比LISS更低的复杂度。

3.2 停止条件及半径更新

在ASLSD算法中，更新半径 r 的确定直接影响到算法的复杂度和系统性能。如果 r 选择过小，将导致很多节点被多次重复检测。如果 r 选择过大，则会引入很多冗余路径。因此，本文根据停止条件和已检测的状态来确定更新半径 r 。

3.2.1 停止条件

检测的目的是使所检测到序列的 $p(\mathbf{x}, \mathbf{y})$ 之和可以基本体现 $p(\mathbf{y})$ 。考虑到 $p(\mathbf{y}) = \sum_{\mathbf{x} \in \Omega} p(\mathbf{x}, \mathbf{y}) =$

$\sum_{\mathbf{x} \in \Omega} p(\mathbf{y} | \mathbf{x}) p(\mathbf{x})$ ，可以通过设置相似系数 α 来表征由候选序列得到的 $\hat{p}(\mathbf{y})$ 与遍历所有 \mathbf{x} 序列得到的 $p(\mathbf{y})$ 的相似程度。

假定已检测到的候选序列数为 K ，其度量按照从小到大的顺序排列为 $\{M_1 \dots M_K\}$ 。并注意到 $p(\mathbf{x}, \mathbf{y}) = C e^{-M(\mathbf{x})/\sigma_v^2}$ ，给定 α 后，停止条件进一步可以写为

$$\sum_{k=1}^K e^{-M_k/\sigma_v^2} \geq \alpha \sum_{k=1}^N e^{-M_k/\sigma_v^2} \\ = \alpha \left(\sum_{k=1}^K e^{-M_k/\sigma_v^2} + \sum_{k=K+1}^N e^{-M_k/\sigma_v^2} \right), \quad N = 2^{MN_t} \quad (4)$$

将候选路径移入候选列表后，路径列表中余下的路径为

剩余路径。令 \mathbf{x}_{k-N_t} 表示第 $k-N_t$ 层已检测到的某一 \mathbf{x} 符号序列, 其与接收信号 \mathbf{y} 的联合概率密度为

$$\begin{aligned} p(\mathbf{x}_{k-N_t}, \mathbf{y}) &= C e^{-\sum_{i=1}^{N_t} M_i(\mathbf{x}_{k-N_t}) / \sigma_v^2} \\ &= C \prod_{i=1}^{k-1} e^{-M_i(\mathbf{x}_{k-N_t}) / \sigma_v^2} \cdot e^{-\sum_{i=k}^{N_t} M_i(\mathbf{x}_{k-N_t}) / \sigma_v^2} \end{aligned} \quad (5)$$

其中当 $i < k$ 时, $M_i(\mathbf{x}_{k-N_t}) = \left\| \mathbf{z}'_i - \sum_{j=1}^{k-1} \mathbf{r}_{i,j} x_j \right\|^2 - \sigma_v^2 \ln p(x_i)$, 且

$$\mathbf{z}'_i = \mathbf{z}_i - \sum_{j=k}^{N_t} \mathbf{r}_{i,j} x_j$$

对于所有 $k-N_t$ 已给定的序列 \mathbf{x}_{k-N_t} , 其与接收信号 \mathbf{y} 的联合概率密度之和为

$$\begin{aligned} p_{k-N_t}(\mathbf{x}, \mathbf{y}) &= \sum_{x_{1-k-1} \in \Omega} p(\mathbf{x}_{k-N_t}, \mathbf{y}) \\ &= C e^{-\sum_{i=k}^{N_t} M_i(\mathbf{x}_{k-N_t}) / \sigma_v^2} p_{k-N_t}^{\text{left}}(\mathbf{x}, \mathbf{y}), \\ p_{k-N_t}^{\text{left}}(\mathbf{x}, \mathbf{y}) &= E \left[e^{-\|\mathbf{z}' - \mathbf{R}' \mathbf{x}'_{k-N_t}\|^2 / \sigma_v^2} \right] \end{aligned} \quad (6)$$

其中 $\mathbf{R}' = [R(1) \ \cdots \ R(k-1)]$ 为 R 的左上角 $(k-1) \times (k-1)$ 矩阵, $\mathbf{x}'_{k-N_t} = [(x_{k-N_t})_1 \ \cdots \ (x_{k-N_t})_{k-1}]^T$ 为 \mathbf{x} 序列的前 $(k-1)$ 个符号。

可见, 路径列表中保留了剩余路径的部分度量值, 因此可以基于路径列表来近似 $p(\mathbf{y})$ 。

(1) 初始检测 (a) 用 1 代替 $p_{k-N_t}^{\text{left}}(\mathbf{x}, \mathbf{y})$ 注意到对任一 \mathbf{x}_{k-N_t} 都有 $p(\mathbf{y} | \mathbf{x}_{k-N_t}) \leq 1$, 因此最简单的方法是将 $p_{k-N_t}^{\text{left}}(\mathbf{x}, \mathbf{y})$ 直接放大为 1。即认为, 所有剩余路径的度量都用其已检测到的度量近似。但由于无论是高信噪比情况还是低信噪比情况均有 $p_{k-N_t}^{\text{left}}(\mathbf{x}, \mathbf{y}) \ll 1$, 直接将 $p_{k-N_t}^{\text{left}}(\mathbf{x}, \mathbf{y})$ 放大到 1, 将使得度量值被过度放大, 从而使检测列表长度大大增加。

(b) 利用特定的 \mathbf{x}_{k-N_t} 序列计算 $p_{k-N_t}^{\text{left}}(\mathbf{x}, \mathbf{y})$ 由于在初始半径设置时已得到 \mathbf{x}_{ml} , 因此考虑用 $\mathbf{x}_{k-N_t}^{ml}$ 来计算 $p_{k-N_t}^{\text{left}}(\mathbf{x}, \mathbf{y})$, 这里定义 $\mathbf{x}_{k-N_t}^{ml} = ((x_{ml})_1 \ \cdots \ (x_{ml})_{k-1}, (x_{k-N_t})_k \ \cdots \ (x_{k-N_t})_{N_t})^T$ 。分析其性能, 当噪声取值较大时, 不同的 \mathbf{x}_{k-N_t} 对于 $E \left[e^{-\|\mathbf{z}' - \mathbf{R}' \mathbf{x}'_{k-N_t}\|^2 / \sigma_v^2} \right]$ 的差异并不明显, 这时可以用任一 \mathbf{x}_{k-N_t} 序列的度量来计算 $p_{k-N_t}^{\text{left}}(\mathbf{x}, \mathbf{y})$ 。另一方面, 注意到在高 SNR 时 $p(\mathbf{x}, \mathbf{y})$ 相对集中于候选路径, 剩余路径对应的 $p_{k-N_t}(\mathbf{x}, \mathbf{y})$ 取值相对较小。因此认为采用 $\mathbf{x}_{k-N_t}^{ml}$ 近似计算 $p_{k-N_t}(\mathbf{x}, \mathbf{y})$ 应不会导致性能的严重下降。综合考虑性能和复杂度的要求, 在算法中采用 $\mathbf{x}_{k-N_t}^{ml}$ 的度量来近似计算

$p_{k-N_t}(\mathbf{x}, \mathbf{y})$ 。这时停止条件为

$$\begin{aligned} &\sum_{\mathbf{x} \in \text{cand_list}} p(\mathbf{x}, \mathbf{y}) \\ &\geq \alpha \left(\sum_{\mathbf{x} \in \text{cand_list}} p(\mathbf{x}, \mathbf{y}) + \sum_{\mathbf{x} \in \text{path_list}} p(\mathbf{x}_{k-N_t}^{ml}, \mathbf{y}) \right) \end{aligned} \quad (7)$$

(2) 迭代过程 在迭代过程中, 由于已得到符号序列的先验信息, 而且随着迭代次数的增加, $p(x_i = a) \rightarrow 1$ 或 0。这时 $E \left[e^{-\|\mathbf{z}' - \mathbf{R}' \mathbf{x}'_{k-N_t}\|^2 / \sigma_v^2} \right] \rightarrow e^{-\|\mathbf{z}' - \mathbf{R}' \mathbf{x}'_{\text{real}}\|^2 / \sigma_v^2}$, $\mathbf{x}'_{\text{real}}$ 为实际发送序列的 $(x_1 \ \cdots \ x_{k-1})$ 。另一方面, 定义 $x_i^s = \sum_{a \in A} (x_i = a) p(x_i)$ 为 x_i

的软信息, 可以得到 $e^{-\|\mathbf{z}' - \mathbf{R}' (\mathbf{x}'_{k-N_t})_s\|^2 / \sigma_v^2} \rightarrow e^{-\|\mathbf{z}' - \mathbf{R}' \mathbf{x}'_{\text{real}}\|^2 / \sigma_v^2}$, 其

中 $(\mathbf{x}'_{k-N_t})_s = [(x_{k-N_t})_1^s \ \cdots \ (x_{k-N_t})_{k-1}^s]^T$ 。因此, 可以利用 $(\mathbf{x}'_{k-N_t})_s$ 通过软度量来近似 $p_{k-N_t}(\mathbf{x}, \mathbf{y})$ 。这时, 停止准则可写为

$$\begin{aligned} &\sum_{\mathbf{x} \in \text{cand_list}} p(\mathbf{x}, \mathbf{y}) \geq \alpha \left(\sum_{\mathbf{x} \in \text{path_list}} p_{k-N_t}(\mathbf{x}_s, \mathbf{y}) + \sum_{\mathbf{x} \in \text{cand_list}} p(\mathbf{x}, \mathbf{y}) \right), \\ p_{k-N_t}(\mathbf{x}_s, \mathbf{y}) &= C e^{-\sum_{i=k}^{N_t} M_i / \sigma_v^2} \cdot e^{-\|\mathbf{z}' - \mathbf{R}' (\mathbf{x}'_{k-N_t})_s\|^2 / \sigma_v^2} \end{aligned} \quad (8)$$

3.2.2 确定更新半径 r 当停止准则不满足时, 需要确定新的半径 r ($r > r_0$) 以获得更多的候选路径。本文提出两种 r 的更新方法及其性能分析。

最简单的半径更新方法是假定 $p(\mathbf{x}, \mathbf{y})$ 集中于 $p(\mathbf{x}_{ml}, \mathbf{y})$, 将停止条件式(4)进一步缩紧为:

$$(1 - \alpha) e^{-M_1 / \sigma_v^2} \geq \alpha (N - K) e^{-M_1 / \sigma_v^2} \quad (9)$$

这里 K 为已检测路径数。因此更新的半径可表示为:

$$M_r \geq M_1 + \sigma_v^2 \ln \left[\frac{\alpha}{1 - \alpha} (N - K) \right] \quad (10)$$

可见, 该方法仅利用了 \mathbf{x}_{ml} , 而没有考虑检测列表中已检测到的其它序列。在高 SNR 时, 假设与实际情况比较接近, 而且由于 σ_v^2 很小, 即使采用 $(N - K)$ 的放大, 也不会引起检测列表长度显著增加, 因此可以获得较好的性能。但当 SNR 较低时, 该假设不再成立, $(N - K)$ 的放大将使检测列表的长度大大增加。

为了使半径更新更加合理, 需综合利用所有已检测到的状态和停止条件。这里, 先建立两个假设: (1) 任一符号序列的 $p(\mathbf{x}, \mathbf{y})$ 在其相邻的较小范围内变化比较平缓, 即认为与第 K 个候选路径相邻的后续路径的度量也近似于 M_K ; (2) 位于底层的路径相对来说具有较小度量的概率较大。综合考虑停止条件和检测状态, r 可通过如下步骤确定。

步骤 1 由式(11)计算需要增加的候选路径数 Δ , 表明还需要 Δ 个度量为 M_K 的候选路径才能满足停止条件。

$$C \left(\sum_{k=1}^K e^{-M_k/\sigma_v^2} + \Delta e^{-M_k/\sigma_v^2} \right) \geq \alpha p(y) \quad (11)$$

步骤 2 从底层($k=1$)开始,累计各层可能扩展的分支数 $\text{num} = \text{num} + \text{left}(k) \cdot M^{k-1}$,其中 $\text{left}(k)$ 为第 k 层的剩余路径数,找到满足 $\min_k(\text{num} \geq \Delta)$ 的 k 值。即认为后续第 Δ 个度量可由第 k 层的剩余路径扩展得到,因此基于该层内的各剩余路径得到更新半径。

步骤 3 对 k 层内的所有剩余路径按前述方法扩展至完整路径,并对其扩展后的度量进行排序。计算 $\Delta_k = \Delta - \sum_{i=1}^{k-1} \text{left}(i)M^{i-1}$,即认为在第 k 层内应选择第 Δ_k 个度量为 r 。

从以上步骤可见,这种半径更新方法根据停止条件计算需要的路径数,在假设前提下用与所需路径对应的最大度量作为更新半径,避免了过多的重复检测。另一方面,通过将后续度量均缩小为 M_K ,又避免了 r 被过度放大。

3.2.3 初始半径的选择 由于检测开始时,路径列表为空,需要设置初始值 r_0 来启动检测过程。当然该初始值不能选得过大,尤其是在高 SNR 或迭代过程中。考虑到 \mathbf{x}_{ml} 具有最小的度量值 $M(\mathbf{x}_{ml})$,因此可以采用一个较小的半径得到 \mathbf{x}_{ml} ,然后设置初始半径 $r_0 = \varepsilon M(\mathbf{x}_{ml})$,这里 ε 为大于且接近 1 的实数。

4 复杂度分析

通过前述 ASLSD 算法过程可以看出,ASLSD 算法的实质是采用后验概率为度量,依据逐渐递增的半径从内向外依次作 LSD 检测。其与 LSD 算法的不同之处在于:(1)检测过程依据逐渐递增的半径进行。但是由于采用列表形式实现检测过程,列表中保存了上次检测后各路径所达到的状态和度量值,使半径更新后只需沿着已检测的状态继续搜索即可,不必再从顶层计算度量值重新检测。因此,半径递增所引入的复杂度可以显著减小。(2)ASLSD 算法需搜索 \mathbf{x}_{ml} 以计算初始半径。考虑到 \mathbf{x}_{ml} 的检测可以采用一个相对很小的半径来获得,而且目前已有很多加速的检测方法,因此可以认为其对整个检测过程复杂度的影响不大。(3)ASLSD 算法判断停止条件时需根据各剩余路径估计 $p(\mathbf{y})$ 。在对 $p(\mathbf{y})$ 的估计中,对每一条剩余路径来说,只需直接将其未检测到的各层用 ML 或软判值数值代入计算即可,计算量较小。而且随着信噪比和迭代次数的增加,后验概率取值较大的符号序列逐渐减少,其所需的半径更新次数和需计算的剩余路径数将会进一步减少。因此,令检测结束时 ASLSD 所检测到的总列表长度(候选路径和剩余路径之和)为 N_{cand} ,可以近似认为 ASLSD 算法的复杂度与得到 N_{cand} 个完整路径的 LSD 算法的复杂度相当。在仿真中,本文通过仿真比较二者所需的列表长度数目来体现两种算法的复杂度差异。

5 仿真结果

仿真环境设置如下:信道为瑞利快衰落环境。系统中采用(2,1,2)卷积码编码和 BCJR 方式译码。调制方式为 QPSK,对应于 4×4 天线的 MIMO 系统。ASLSD 算法的初始半径设置为 $r_0 = \varepsilon M(\mathbf{x}_{ml})$, $\varepsilon = 1.01$ 。

(1) $p(\mathbf{y})$ 估计效果 图 2 为初始检测时,不同 $p(\mathbf{y})$ 近似方法的性能比较。比较而言,“用 1 代替”的方法估计性能最差,这是由于这种方法只简单地将后续度量放大到极值,而没有利用格点约束、SNR 等其它信息,而用 $\mathbf{x}_{k-N_t}^{ml}$ 计算 $p_{k-N_t}^{\text{left}}(\mathbf{x}, \mathbf{y})$ 可以达到较小的估计误差。另外,“用 1 代替”的方法随 SNR 增加明显呈现出先升后降的过程。这是由于在低 SNR 时,到达底层的路径较多,而需补足的路径较少,因此估计误差也相对较小。但随 SNR 的增加,需补足的路径数增多,误差也逐渐增大。当 SNR 增加到使候选序列的 $p(\mathbf{x}, \mathbf{y})$ 已成为 $p(\mathbf{y})$ 的主要分量时,估计误差又开始逐渐下降。综合考虑估计性能和实现复杂度,后续仿真中在初次检测时均采用 $\mathbf{x}_{k-N_t}^{ml}$ 计算 $p_{k-N_t}^{\text{left}}(\mathbf{x}, \mathbf{y})$ 。图 3 为迭代过程中用软信息估计 $p(\mathbf{y})$ 时的估计性能。从图中可见,与前述分析一致,归一化误差均随 SNR 或迭代次数增加而减小。

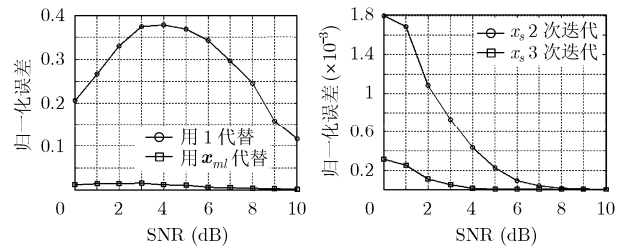


图 2 初始检测时,用 1 代替与用 $\mathbf{x}_{k-N_t}^{ml}$ 代替(QPSK)

图 3 迭代过程中用软信息补足(QPSK)

(2) 半径选择的性能和复杂度 图 4~图 5 为不同半径选择方法的性能和复杂度比较。就路径数而言,当信噪比较低时,“门限”方法的列表长度明显大于“状态”方法的列表长度。这是由于“门限”方法的假设在低信噪比条件下并不成立,而且 $\sigma_v^2 \ln[\alpha(N-K)/(1-\alpha)]$ 的放大也会使半径范围大大增加,而“状态”方法由于综合利用了已有检测状态和停止条件,从而避免了对半径的盲目放大。随着 SNR 和迭代次数的增加, $\sigma_v^2 \ln[\alpha(N-K)/(1-\alpha)]$ 对半径的影响也逐渐减弱,“门限”方法的假设条件在高 SNR 下逐渐得到满足,二者列表长度的差距逐渐减小。就性能而言,在低信噪比时,由于门限方法初始检测利用的列表长度较大,使得迭代中传递的先验信息可靠性较高,因而在性能上门限方法稍优于基于状态方法。随着信噪比的增加,二者整个检测所利用的列表长度逐渐相近,因而性能也逐渐一致。

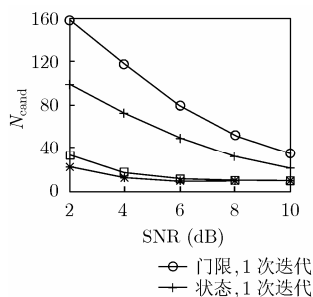


图 4 不同半径选择的 N_{cand} (QPSK)

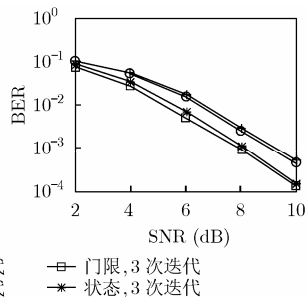


图 5 不同半径选择的误码性能(QPSK)

(3)系统性能 图 6-图 7 为 QPSK 时, ASLSD 与 LSD 的性能和检测列表长度比较。从检测列表的长度来看, LSD 采用固定且较大的列表长度(QPSK, 4×4 天线, 列表长度为 256), 而 ASLSD 所需的列表长度则可以随 SNR 和迭代次数的不同自适应的变化。当停止条件设置为(QPSK, $\alpha = 0.9$)时, ASLSD 所需的列表长度仍小于 LSD 算法, 尤其是在高 SNR 和迭代过程中其差距更加显著。因此可以认为, 与 LSD 相比, ASLSD 具有更低的复杂度。从系统性能上看, 列表长度的缩短将给 ASLSD 算法带来一定的性能损失。仿真表明, 对 QPSK, 4×4 系统来说, 其性能损失小于 0.5dB。

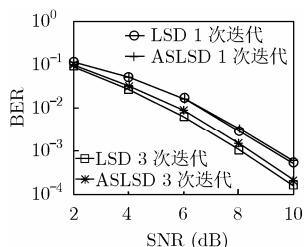


图 6 QPSK, ASLSD 与 LSD 性能比较

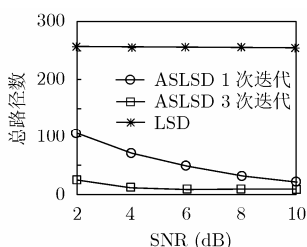


图 7 QPSK, ASLSD 与 LSD 检测列表长度比较

6 结束语

本文针对常用的非穷尽形式列表的 MAP 检测算法中直接采用恒定且较大的列表长度, 导致接收机复杂度增加的问题, 提出了一种改进的 ASLSD 算法。算法的优点在于: 采用递增的半径检测符号序列, 直至满足停止条件, 使所得检测列表长度可随信噪比、迭代次数等因素自适应变化; 将列表操作与 LSD 算法相结合, 不仅避免了符号序列在不同半径下的重复检测, 而且无需排序操作。仿真表明, 在较小性能损失的前提下, 该算法可以大大缩短检测列表的长度, 进而有效降低接收机的复杂度, 具有更优的性能和复杂度的折中。

参考文献

- [1] Hochwald B and ten Brink S. Achieving near-capacity on a multiple-antenna channel. *IEEE Trans. on Commun*, 2003, 51(3): 389-399.
- [2] Vikalo H, Hassibi B, and Kailath T. Iterative decoding for MIMO channels via modified sphere decoding. *IEEE Trans. on Wireless Commun*, 2004, 3(6): 2299-2311.
- [3] Boutros J, Gresset N, Brunel L and Fossorier M. Soft-input/soft-output lattice sphere decoder for linear channels. in Proc Globecom'03, San Francisco, USA, Dec. 2003, vol. 3: 1583-1587.
- [4] Baro S, Hagenauer J and Witzke M. Iterative detection of MIMO transmission using a list-sequential (LISS) detector. in Proc. ICC '03, Anchorage (AK), USA, IEEE, May 2003: 2653-2657.
- [5] Hassibi B and Vikalo H. On sphere decoding algorithm. I. Expected complexity. *IEEE Trans. on Signal Processing*, 2005, 53(8): 2806-2818.

韩 湘: 女, 1975 年生, 博士生, 研究方向为信道估计与检测算法、MIMO 系统。
 赵 丹: 女, 1979 年生, 博士生, 研究方向为同步与信道估计算法、OFDM。
 魏急波: 男, 1967, 教授, 研究方向为通信信号处理与通信网络。
 王建新: 男, 研究员, 研究方向为通信信号处理。