**球检测**

在上一章可知，虽然ML算法性能最好，但因其算法复杂度很高，很难应用于较大的调制星座或天线数较多的系统中。人们一直希望可以在算法复杂度与算法的检测性能之间寻求一个折中，进而找到一种算法复杂度相对不太高而同时检测性能又相对较好的方法。球检测(SD: Sphere Detecting)算法可以在具有合理计算量的同时又达到最大似然检测的误码性能，由此可以用SD算法代替ML算法用于无线MIMO系统。

**球检测算法**

球检测算法的基本思想是：在一个以矢量为中心的半径为的多维球内搜索格点。通过限制或减小搜索半径来减少搜索的点数，从而降低搜索的计算复杂度。在这个多维球内距离矢量最近的格点，也就是最大似然点。

球检测的基本思想可以用下图1表示：



图1 球检测算法的示意图

**系统模型**

球检测算法将MIMO检测转化为如下问题：

 （1）

其中为球的半径。

算法的一般步骤为：首先通过矩阵分解（通常为分解或分解）将信道矩阵分为三角矩阵；然后在相应的栅格空间中求解[1]。

以分解为例，算法的理论依据如下：

设经过实值分解的矩阵为，接收到的向量通过实值变换后为，码本向量通过实值变换后为。半径，则根据上文球检测公式，我们的目标为求出（1）式中的所有符合条件的：

我们对进行分解，由上文得：

 （2）

其中为的对角线上的元素大于0的上三角矩阵，和分别为和的矩阵。根据分解的性质，矩阵为正交矩阵，其余任意向量相乘后不会改变向量的模，因此，（2）式可以进一步改写：

 （3）

其中，假设。可知是已知的。进一步将上述不等式转换为：

 （4）

其中

 （5）

可知，在时，是已知的，也是已知的。而在时，只与有关。根据上面的特点，球检测按照以下思路求解：

在（4）式中，如果只考虑，将忽略掉，不等式将简化为

 （6）

求解上述不等式可得：

 （7）

从中选择一个值，并代入（4）式中，此时仅考虑和，而忽略及其之前的所有项，这样，则有

（8）

根据上式和（7）式选择的一个值，则可以求出的范围。以此类推，就可以求出任意的范围，将其表示为，其中

 （9）

在搜索过程中，若发现的取值范围为空，则说明的取值导致了不等式无解，即它们的取值已超出了球的边界，此时球检测器将回到上一步，并取取值集合中的下一个值，然后继续搜索过程。搜索结束以后，如果没有向量输出，则说明半径过小，球为空，则增大半径，继续搜索。如果球检测最终输出了一组向量，则在其中选择经过变换后，与接收向量最近的一个作为最终的结果。

通过上面对球检测过程的描述可以看出，球检测实际就是一个树搜索过程，树共由层组成，根节点没有实际意义，除叶子节点外，每个节点的度为L，每个分支对应一个星座点。部分向量确定了一条从根节点出发，经过到达的路径。显然，为的函数，我们将定义为分支量度，而分支量度的累加

 （10）

定义为路径量度，因为是已知的，因此将其定义为根节点的路径量度。另外，任意一个向量与之间的距离定义为代价量度

 （11）

可以看出，

 （12）

由上文所述，可以将球检测算法的基本思想重新概括如下：

先在接收信号空间中预设一个以接收信号点为圆心的球，再把该球映射为发射信号空间的一个球，在此球内搜索可能的发射信号点，一旦找到一个发射信号点，即以该信号点的映射点与接收信号的距离为半径收缩预设的球，从而使后面的搜索能在更小的范围内进行。

可以看出，球形检测算法实际上是一种不断强化约束条件的搜索算法。

那么，球检测算法的一般步骤也可概括如下：

1. 给定初始半径，即，且设；
2. 在球内搜索发射信号的星座点；
3. 若搜索到一个星座点，那么令，更新半径：，返回（2），继续搜索发射信号的星座点；
4. 若没有搜索到星座点，那么令，增加球的半径，返回（2），继续搜索；
5. 迭代结束，即为最后的搜索结果。

图2给出了一个实际搜索的例子，以调制方式为例，用圆点表示星座图中的点，其值分别取中的一个，其中算法所访问到的点以及对应的路径分别给出了标记。可以看出节点B，D，E，G，I和J对算法整体的性能是没有贡献的，因为它们所处的路径没能最终到达第层，而在中途就被放弃了。

最终得到的实线为满足条件的星座符号组合，而虚线代表被放弃的路径，然后计算各实线对应的符号向量的欧式距离，选择具有最小距离所对应的符号向量作为最终的输出结果。



图2 球检测计算举例示意图

若搜索完第层仍未检测出符合条件的符号向量时，则表示该超球内部没有合适的格点，此时需要增大搜索半径，重新开始搜索过程。

上一章已经知道，ML检测也可以描述为树搜索，但是其将搜索整个树。而球检测只需要搜索路径量度小于的路径，因此复杂度被有效的降低了。图3给出了的MIMO系统，采用调制时，一个树的实例。图中加黑的路径对应正确的解向量。ML需要遍历树中的所有节点才能确定这个解，而球检测只需要处理浅色的节点即可完成搜索过程，可见浅色节点的数量要少很多，因此复杂度也低得多。



图3 球检测与ML检测树搜索示意图

通过以上对球形检测算法的简单描述可知，影响球检测算法的误码性能和计算复杂度的关键是：

1. 搜索半径的确定。

如果半径太大，那么球内就会包含过多的点，使计算复杂度接近或达到最大似然算法的计算复杂度(指数级)。如果搜索半径太小，那么可能在搜索区内没有所要搜索的符合要求的点，导致检测失败。目前被较为广泛接受的理想搜索半径是覆盖半径。覆盖半径是以点为中心的球内一定存在格点的最小半径。即在以任意矢量为中心，以覆盖半径为半径的球内至少存在一个格点。但是确定一个格的覆盖半径本身就是一个NP问题。

1. 球内是否存在有效点的确定。

如果需要依据每一个格点和矢量之间的距离来判断每一点是否在球内的话，那么这个计算量也是指数级的。

[1]E. Agrell, T Eriksson, A. Vardy, and K. Zeger, Closest point search in lattices, IEEE Trans. Inform. Theory, vol. 48, pp. 2201-2214, Aug. 2002.