Sphere Decoding Algorithm for Space-Time Block Codes

Mohammad Arslan Zulfiqar
zulfiqar@wisc.edu

April 16, 2009

1 Motivation

The key challenge faced by future wireless communication systems is to provide high-data rate wireless access at high quality of service. Combined with the fact that the spectrum is a scarce resource and the propagation conditions are hostile due to fading and due to interference from users, this requirement calls for a means to radically increase spectral efficiency and to improve the link reliability. Multiple-input multiple output wireless technology seems to meet these demands by offering increased spectral efficiency through spatial multiplexing gain, and improved link reliability due to antenna diversity gain. That is why, MIMO technology has been proposed for next generation wireless systems (WiMax, LTE, 802.11n). But, the gains achievable in MIMO systems come at the cost of increased complexity. D. Perels et al. [1] show that their ASIC implementation of MIMO-OFDM WLAN physical layer based on the IEEE 802.11a standard increases in area by a factor of 6.5 in going from a SISO to a 4 x 4 MIMO system. Due to the increased complexity, a significant focus in research on MIMO systems has been directed at sub-optimal decoding strategies [2, 3]. Examples of less complex decoding techniques include zero-forcing and V-BLAST. But, they come at the price of reduced performance at the receiver. The optimal Maximum likelihood strategy which gives the best performance is based on exhaustive search and becomes too complex as the number of antennas and data rates increase.

2 Objective

For the class project, I would like to implement the sphere decoding algorithm proposed by [4]. It has been shown that the sphere decoding algorithm has almost the same performance as the maximum likelihood decoder but requires computational effort implementable in practice [5]. I would like to do the implementation in MATLAB and explore different opportunities for optimization.

3 Game Plan

The approach I will adopt for this project is:

- First, I will develop a complete understanding of the algorithm by studying the relevant literature [4, 5]. A brief description of the algorithm is given in Section 4.
- Implement the basic algorithm in floating-point using MATLAB.
- Optimize the algorithm using techniques learnt in class (unfolding, retiming, algorithmic transformation etc.) as well as those proposed in research [6–8]. Specifically, I will examine the constraints and trade-offs of these methods in detail.
- If time permits, convert the algorithm to fixed-point implementation using 16 and 32 bit storage variables and study the performance degradation in terms of SNR on the probability of error. Currently, most of the hardware used in mobile handsets is based on fixed-point arithmetic due to its low-power consumption. So, it would be beneficial to study the impact on performance in converting the algorithm to fixed-point.

4 Sphere Decoding Algorithm

4.1 Basic Setup

In this section, I'll present a brief overview of the Sphere decoding algorithm. More details including derivation will be included in the final report. The material presented in this section is mostly referenced from [5]. For a MIMO system, the received signal vector $\mathbf{x}$ is modeled as

$$\mathbf{x} = \mathbf{H}\mathbf{s} + \mathbf{v}$$

where $\mathbf{x}$ is the $n \times 1$ received signal vector, $\mathbf{H}$ is the known $n \times m$ channel matrix, $\mathbf{s}$ is the $m \times 1$transmitted signal vector and $\mathbf{v}$ represents a $n \times 1$ i.i.d Gaussian noise vector with $\mathcal{N}(0, \sigma^2)$ distributed entries.

Now, the optimal Maximum Likelihood detector that minimizes the average probability of error $Pr(\hat{\mathbf{s}} \neq \mathbf{s})$ is given as

$$\hat{s} = \underset{\mathbf{s} \in \mathcal{D}_L^m}{\operatorname{argmin}} ||\mathbf{x} - \mathbf{H}\mathbf{s}||^2 \tag{1}$$

where $\mathcal{D}_L$ is the set of alphabets for a L-QAM constellation e.g., for 16-QAM, we may have $\mathcal{D}_L = \{-3, -1, 1, 3\}$.

The non-linear optimization problem given by (1) is an exponentially complex problem in m. Finding its exact solution is, in general, NP hard. There are two possible approaches one can take:

1. Brute force, which involves searching over the entire lattice space $\mathcal{D}_L^m$.

2. Exploit the structure of the lattice and search only those $s \in \mathcal{D}_L^m$ that lie in sphere of radius d around $\mathbf{x}$. This is where the sphere decoding strategy comes in.

It is intuitively obvious that the second approach will be less computationally intensive. But, there are two questions that immediately come to mind if the second approach is considered:

1. How to choose the radius of the sphere d:

   Note that in (1), $(1/\sigma^2)||\mathbf{v}||^2 = (1/\sigma^2)||\mathbf{x} - \mathbf{Hs}||^2$ is distributed as $\mathcal{X}^2$ random variable with n degrees of freedom. Now, if we define $d^2 = \alpha n \sigma^2$, then solving $Pr(||\mathbf{x} - \mathbf{Hs}||^2 < d^2) = 1 - \epsilon$ for α for a small ϵ will ensure that with high probability, the sphere of radius d, contains a lattice point.

2. Once d is chosen, how do we tell which points lie inside the sphere:

   The Sphere decoding algorithm proposed by [4] deals with this issue. The basic premise of the algorithm is that in one-dimension, a sphere reduces to an interval. Hence, if we consider the multi-dimensional problem one dimension at a time, then we only need to deal with intervals, which makes the problem easier. Of course, this is just a crude explanation. The derivation of the exact algorithm is provided below.

4.2 Derivation

Assume $n \geq m$ i.e., there are at least as many equations as unknowns. Also, the lattice point $\mathbf{Hs}$ lies inside the sphere of radius d centered at $\mathbf{x}$ if and only if

$$d^2 \geq ||\mathbf{x} - \mathbf{Hs}||^2 \tag{2}$$

Taking the QR factorization of the channel matrix $\mathbf{H}$, we get

$$\mathbf{H} = \mathbf{Q} \begin{bmatrix} \mathbf{R} \\ \mathbf{0}_{(n-m)\times m} \end{bmatrix}$$

where $\mathbf{R}$ is an upper triangular $m \times m$ matrix and $\mathbf{Q} = [\mathbf{Q_1 Q_2}]$ is an $n \times n$ orthogonal matrix. Eq. (2) can now be written as

$$d^2 \geq \left|\left| \mathbf{x} - [\mathbf{Q_1 Q_2}] \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} \mathbf{s} \right|\right|^2 = ||\mathbf{Q_1^* x} - \mathbf{Rs}||^2 + ||\mathbf{Q_2^* x}||^2$$

where $(.)^*$ denotes Hermitian matrix transposition. In other words,

$$d'^2 - ||\mathbf{Q_2^* x}||^2 \geq ||\mathbf{Q_1^* x} - \mathbf{Rs}||^2$$

Defining $\mathbf{y} = \mathbf{Q_1^* x}$ and $d'^2 = d^2 - ||\mathbf{Q_2^* x}||^2$, we can rewrite the above equation as

$$d'^2 \geq \sum_{i=1}^{m} \left(y_i - \sum_{j=i}^{m} r_{i,j} s_j \right)^2$$

where $r_{i,j}$ denotes the (i, j) entry of $\mathbf{R}$. Notice, that the indices of the inner sum go from i to m and not from 1 to m. This is because of the upper triangular property of $\mathbf{R}$. This enables us to expand the RHS of the above equation as

$$d'^2 \geq (y_m - r_{m,m}s_m)^2 + (y_{m-1} - r_{m-1,m}s_m - r_{m-1,m-1}s_{m-1})^2 + \cdots \quad (3)$$

where the first term depends only on s_m, the second term on $\{s_m, s_{m-1}\}$ and so on. Therefore, a necessary condition for $\mathbf{Hs}$ to lie inside the sphere is that $d'^2 \geq (y_m - r_{m,m}s_m)^2$. This leads to

$$\left\lceil \frac{-d' + y_m}{r_{m,m}} \right\rceil \leq s_m \leq \left\lfloor \frac{d' + y_m}{r_{m,m}} \right\rfloor \quad (4)$$

Now, for every s_m satisfying (4), define ${d'_{m-1}}^2 = d'^2 - (y_m - r_{m,m}s_m)^2$ and $y_{m-1|m} = y_{m-1} - r_{m-1,m}s_m$. Looking at (3), we see that

$$\left\lceil \frac{-d'_{m-1} + y_{m-1|m}}{r_{m-1,m-1}} \right\rceil \leq s_{m-1} \leq \left\lfloor \frac{d'_{m-1} + y_{m-1|m}}{r_{m-1,m-1}} \right\rfloor \quad (5)$$

One can proceed in a similar fashion for s_{m-2} and so on until s_1, thereby obtaining all lattice points belonging to (2). An overview of the algorithm is given in Alg. 1.

Input: $\mathbf{Q} = [\mathbf{Q_1}\mathbf{Q_2}], \mathbf{R}, \mathbf{x}, \mathbf{y} = \mathbf{Q_1^*}\mathbf{x}, d$
where for the known channel matrix $\mathbf{H} = \mathbf{QR}$ (QR factorization)

1. Set $k = m, {d'_m}^2 = d^2 - ||\mathbf{Q_2^*}\mathbf{x}||^2, y_{m|m+1} = y_m$
2. (Bounds for s_k) Set $UB(s_k) = \lfloor \frac{d'_k + y_{k|k+1}}{r_{k,k}} \rfloor, s_k = \lceil \frac{-d'_k + y_{k|k+1}}{r_{k,k}} \rceil - 1$
3. (Increase s_k) $s_k = s_k + 1$. If $s_k \leq UB(s_k)$, go to 5, else go to 4
4. (Increase k) $k = k + 1$; if $k = m + 1$, terminate algorithm, else go to 3
5. (Decrease k) If $k = 1$, go to 6. Else $k = k - 1$,
$y_{k|k+1} = y_k - \sum_{j=k+1}^{m} r_{k,j}s_j$, ${d'_k}^2 = {d'_{k+1}}^2 - (y_{k+1|k+2} - r_{k+1,k+1}s_{k+1})^2$
and go to 2
6. Solution found. Save $\mathbf{s}$ and its distance from $\mathbf{x}$,
${d'_m}^2 - {d'_1}^2 + (y_1 - r_{1,1}s_1)^2$, and go to 3

Algorithm 1: Sphere Decoding Algorithm

5 Implementation Considerations

In order to meet the real-time constraints of modern wireless standards, there is need for faster algorithms. This will be the biggest design goal of my project.

From Alg. 1, it is evident that the implementation of the sphere decoding algorithm involves m nested loops with break conditions occurring whenever the the upper bound $UB(.)$ is violated for a transmitted signal component in a particular dimension. The algorithm also involves significant number of floating point operations. Hence, to achieve speedups in the algorithm, I'll specially focus on the following techniques:

- Vectorization of the algorithm
- Loop-unfolding
- Minimization of floating-point operations

6 Why I Chose this Project

The main reasons I chose this project are:

- Most of the research done on MIMO systems has been of theoretical nature. Focus on different implementation strategies and considerations is a recent trend in research. Hence, I'm hoping that this problem would prove to be more than just a class project.
- This project will enable me to explore and apply the different algorithmic techniques we've learnt in class thus improving my comprehension of these concepts.

References

[1] D. Perels, S. Haene, P. Luethi, A. Burg, N. Felber, W. Fichtner, and H. Blcskei, "ASIC Implementation of a MIMO-OFDM Transceiver for 192 Mbps WLANs," *Proceedings of ESSCIRC*, 2005.

[2] P. Luethi, C. Studer, S. Duetsch, E. Zgraggen, H. Kaeslin, N. Felber, and W. Fichtner, "Gram-Schmidt-based QR Decomposition for MIMO Detection: VLSI Implementation and Comparison," *IEEE Asia Pacific Conference on Circuits and Systems*, 2008.

[3] M. Borgmann and H. Blcskei, "Interpolation-Based Efficient Matrix Inversion for MIMO-OFDM Receivers," *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Eighth Asilomar Conference on*, 2004.

[4] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis." *Mathematics of Computation*, 1985.

[5] H. Blcskei, D. Gesbert, C. Papadias, and A. V. der Veen, *Space-Time Wireless Systems: From Array processing to MIMO communications*. Cambridge, 2006.

[6] Z. Safar, W. Su, and K. Liu, "A fast sphere decoding algorithm for space-frequency block codes." *EURASIP Journal of Applied Signal Processing*, 2006.

[7] A. Burg, M. Borgmann, M. Wenk, and M. Zellweger, "Vlsi implementation of mimo detection using the sphere decoding algorithm." *IEEE Journal of Solid-State Circuits*, 2005.

[8] M. Damen, H. Gamal, and G. Caire, "On maximum-likelihood detection and the search for the closest lattice point." *IEEE Transactions on Information Theory*, 2003.