Reduced Complexity Space Division Multiplexing Receivers

Geert Awater, Allert van Zelst and Richard van Nee

Lucent Technologies Bell Labs, Zadelstede 1-10, 3431 JZ Nieuwegein, The Netherlands
email: awater@lucent.com

Abstract — This paper presents Space Division Multiplexing (SDM) as a technology to satisfy the user demand for ever higher bit rates in wireless Local area network technology. SDM uses multiple transmit and receive antennas to increase bit rate proportionally, without reducing range, delay spread tolerance and spectrum occupancy. The optimal SDM receiver is the maximum likelihood decoder, which has the problem that its computational complexity is exponential in the number of antennas. This paper discusses complexity reduction, looks at existing, approximate algorithms and proposes new, exact algorithms. The Symbol Error rate performance and the computational complexity are compared and discussed.

I. Introduction

The development of wireless Local Area Network technology is driven primarily by an ever-increasing need for user bandwidth and system capacity. Traditionally, higher bit rates have been obtained using larger modulation constellations and higher symbol rates, resulting in a larger signal bandwidth. Equalizers or OFDM (Orthogonal Frequency Division Multiplexing) are employed to combat the increased inter-symbol interference caused by multi-path propagation. The problems associated with this approach are:

- Large bandwidths are mostly available in higher frequency bands. As free space signal loss decreases inversely proportional with carrier frequency, the link budget is tighter than for lower frequencies. In addition, high frequency RF circuitry is more costly and less power efficient.
- Since more bits per second are transmitted, E_b/N_0 decreases, reducing the link budget even more.
- Higher signaling rates imply higher sensitivity to inter-symbol interference (ISI).

Space division Multiplexing (SDM) [3] where information is transmitted and received over several transmit antennas and receive antennas in parallel, avoids those problems.

- Capacity gain can be realized in existing bands. Thus there is no additional free space loss, and cost efficient RF designs can be reused.
- The receiver diversity order is equal to the number of receive antennas [1]. As antennas are added, the SNR gain compensates for the loss in E_b/N_0 caused by the fact that the available energy is be divided over multiple transmitters.
- As the signaling rate remains the same, there is no additional ISI (in fact delay spread robustness increases [2])

The price is in the additional RF circuitry (of which parts, such as local oscillators can be shared), and higher signal processing complexity. This paper evaluates complexity and proposes methods to reduce it.

We consider a communication system, which transmits symbols at discrete times. If M and N denote the number transmit and receive antennas. The system is described by:

$$\mathbf{x} = \mathbf{H}\mathbf{s} + \mathbf{n} \tag{1}$$

where $\mathbf{s}$ is the M-dimensional transmit vector, each element of which is chosen from a (complex-valued) constellation, $\mathbf{H}$ is the $N \times M$ complex, random channel matrix and $\mathbf{n}$ is an N-dimensional complex AWGN vector. We assume Rayleigh flat fading (see [2] for extension to the case of frequency selective fading) for each element of $\mathbf{H}$, reflecting a rich-scattered, indoor radio environment.

We assume that the channel matrix is known by the receiver and remains constant over the duration of the data packet. The training phase, during which the receiver learns $\mathbf{H}$, is outside the scope of this paper.

II. Optimal Receiver Algorithm

Reference [3] estimates the transmitted symbol by multiplication of $\mathbf{x}$ by the Moore-Penrose pseudo-inverse of the channel matrix. An improved algorithm is to use MMSE (Minimum Mean Squared Error) estimator that would be optimal if $\mathbf{s}$ was a continuous, rather than a discrete-valued vector. A further improvement, which is to use decision feedback, is described as well. The best element (with smallest estimated error) of the transmitted vector is computed and sliced, i.e. mapped to one of its possible discrete values. The received vector $\mathbf{x}$ is then compensated for this decision, and the next best element is decoded, and so on, until a complete estimate of the transmitted vector, $\hat{\mathbf{s}}$ is obtained. These methods have in common that they have a computational complexity of $O(M^3)$ and they achieve a receiver diversity of order $N - M + 1$. This implies that the number of receive antennas must be equal or greater than the number of transmit antennas for this communication system to work.

Reference [2] derives the optimal receiver, which is a maximum likelihood detector (MLD). Given the received symbol vector $\mathbf{x}$, the optimal receiver algorithm optimizes, for all possible transmitted signals $\mathbf{s}_j$, the conditional probability $\Pr\{\mathbf{s}_j \mid \mathbf{x}\}$. The vector which maximizes this probability, is given by:

$$\hat{\mathbf{s}} = \arg\min_i \left\| \mathbf{x} - \mathbf{H}\mathbf{s}_i \right\| \tag{2}$$

In [1] it is shown that this detector realizes order N receiver diversity, independent of the number of transmitter antennas, M. If c is the number of constellation points, there are c^M different transmit vectors $\mathbf{s}_j$. The maximum likelihood detector multiplies $\mathbf{s}_j$ by the known channel matrix and then calculates the Euclidean distance from the received vector. Thus, the complexity of the maximum likelihood decoder is exponential in the number of transmit antennas, $O(c^M)$.

The question that we will address in this paper is how we can reduce the complexity of the MLD algorithm, without compromising its receiver performance.

III. Reduced complexity MLD

By rewriting (2) as

$$\hat{\mathbf{s}} = \arg\min_i \left\| \mathbf{x} - \sum_{j=0}^{M-1} \mathbf{H}_j \cdot \mathbf{s}_{ij} \right\| \tag{3}$$

where $\mathbf{H}_j$ is the j-th column of $\mathbf{H}$, and $\mathbf{s}_{ij}$ is the j-th component of vector $\mathbf{s}_i$, and if the components of $\hat{\mathbf{s}}$ are estimated one after the order, it is seen that the MLD algorithm is related to sequence estimation. The MLD problem can be visualized as a tree, as shown in figure 1 which depicts a simple decoding example for $M = 2$. On the horizontal axis is $t = 0, 1, 2$, the decoding step, while the vertical axis represents n(t), which is defined as the distance between $\mathbf{x}$ and the partial estimate of the received vector after t steps:

$$n(t) = \left\| \mathbf{x} - \sum_{j=1}^{t} \mathbf{H}_j \cdot \mathbf{s}_{ij} \right\| \tag{4}$$

In the figure, it is assumed that BPSK (Binary Phase Shift Keying) modulation is used, i.e. $\mathbf{s}_{ij} \in \{-1,+1\}$ and $c = 2$. This is why every node has two branches, i.e. 2 sub-nodes. In the example the maximum likelihood estimate of $\mathbf{s}$, which minimizes $n(M)$, is $(+1,-1)^T$

The computational effort of an exhaustive evaluation of the decoding tree can be reduced by using well known maximum likelihood sequence estimation techniques, such as Fano's algorithm, stack decoding or "retain K best paths" ("K-best" for short). See [4] and [5] for an overview of various decoding methods and their relative performance and computational and storage complexity.

At each step t, the K-best algorithm has a list of K nodes in the tree. For each node in the list, the algorithm calculates the norms of its sub-nodes according to (4). Of the $c{\cdot}K$ resulting norms, the K best are put in the list and the corresponding nodes become the new survivors. The complexity of this algorithm is linear in M. It has the additional advantage of a convenient implementation since its processing time is constant.

These algorithms only approximate a true maximum likelihood detector. In the example of figure 1, the 1-best algorithm would find the correct $\mathbf{s}$ correctly. However, if the branches of the tree would cross each other, this algorithm would make a decoding error.

Intuitively it makes sense to make the algorithm *greedy* by sorting the columns of H in descending order of their norm. Thus, in the branches close to the root, the largest strides are made, which makes the probability that branches cross further down the tree smaller. Hence, it is expected that a K-best would introduce fewer errors.

Simulation results, comparing K-best decoding with exhaustive decoding are given in section VI.

IV. Exact MLD: survivor algorithm

We try to reduce the MLD receiver complexity without resorting to approximation, by exploiting the N-dimensional geometry of the problem as well as the freedom of choosing the order in which to estimate the components of $\mathbf{s}$. The simplest algorithm (called Survivor) finds, for a partial $\mathbf{s}$ estimate, of which only the first t components $(\mathbf{s}_1,\ldots,\mathbf{s}_t)$ are known, a best case and worst case estimate of $n(M)$, which is the distance between $\mathbf{x}$ and the estimated receive vector.

In each node we have a partially estimated receive vector:

$$\mathbf{y} = \sum_{j=1}^{t} \mathbf{H}_j \cdot \mathbf{s}_{ij} \tag{5}$$

We will refer to $\mathbf{y}$ as the *node location*. An upper bound on the distances from $\mathbf{x}$ corresponding to all possible values of the remaining components $(\mathbf{s}_{t+1},\ldots,\mathbf{s}_M)$ is when all remaining $\mathbf{H}$-columns $(\mathbf{H}_t,\ldots,\mathbf{H}_{M-1})$ are all aligned and point exactly in the direction of $\mathbf{x}$. The worst case distance occurs when all remaining columns of are aligned but point away from $\mathbf{x}$, which gives a lower bound. The distances between $\mathbf{x}$ and final estimated receive vector are in the interval:

$$\left\| \mathbf{x} - \mathbf{y} \right\| \pm \sum_{j=t+1}^{M} \left\| \mathbf{H}_j \right\| \tag{6}$$

The Survivor algorithm starts with received vector $\mathbf{x}$, computes node locations $\mathbf{y}$ for every possible value of $\mathbf{s}_1$ and computes the interval in (6). This is shown in figure **2**. The $\mathbf{y}$ vectors are $\mathbf{H}_1$ and $-\mathbf{H}_1$, for $\mathbf{s}_1 = +1$ and -1, respectively. The best case vector $\mathbf{b}_1$ has a length equal to that of the remaining $\mathbf{H}$ columns, $\|\mathbf{H}_2\|$ in this case, and is pointing in the direction of $\mathbf{x}$. The worst case vector $\mathbf{w}_1$ has the same length, but is pointing opposite direction. The worst and best case distances are equal to $\|\mathbf{x}-\mathbf{b}_1\|$ and $\|\mathbf{x}-\mathbf{w}_1\|$, and are calculated according to (6). The vectors $\mathbf{b}_2$ and $\mathbf{w}_2$ are the best and worst case vector for the second node location, $-\mathbf{H}_1$.

The algorithm continues with $\mathbf{s}_2$ and steps through all components of $\mathbf{s}$ sequentially to construct the full decoding tree. However, at each step t, nodes whose *best case* distance is larger than the *worst case* distance of any other node can be eliminated. All nodes in the sub-trees of eliminated nodes need not be evaluated since they would give a distance from $\mathbf{x}$ which can never be better than that from the surviving nodes. This is illustrated in figure 5, which shows an example where at a particular step t, two nodes can be eliminated. It is expected that non-overlapping do occur so that only part of the node needs to be evaluated, thus reducing coding complexity. Note that the eventual complexity of this method depends on the communication channel — both the values of the elements of channel matrix **H** and noise vector **n**.

V. QR survivor MLD

An advanced version of this algorithm, called QR1 is based on the QR-decomposition. We apply a unitary coordinate transform, to upper-triangularize **H**. It is well-known that **H** can be factored as **H** = **QR**, provided that **H** has rank M:

$$\mathbf{R} = \mathbf{Q}^{*} \cdot \mathbf{H} \tag{7}$$

Here **R** and **H** are both $N \times M$ matrices and **Q*** is an $N \times N$ matrix. Basically, **Q** is an orthonormal basis of the vector space spanned by **H**, and **R** is upper triangular. **Q** can be obtained from **H** using Gram-Schmidt orthogonalization, or equivalent, computationally more stable methods involving a series of Householder or Givens rotations [6]. Since **Q** is unitary, **Q**'s inverse is equal to its conjugate transpose ($\mathbf{Q}^{*} = \mathbf{Q}^{-1}$). Each received **x** is transformed to orthonormal coordinate system according to:

$$\mathbf{x}' = \mathbf{Q}^{*} \cdot \mathbf{x} \tag{8}$$

The problem is now represented in a coordinate system where we can better bound the best case and worst case distance. This is because the dimension of the columns of **R** (i.e. **H** represented in the orthonormal system) is reduced by one as the algorithms descends deeper into the tree.

Consider an example for the case $N = M$. We assume that the columns of **H** have been sorted according to their lengths. Then **Q** is determined, inverted (transposed and conjugated) and used to obtain **R** and **x**'. **R** is upper triangular, and its columns are still sorted according to their length, since multiplication by unitary matrix **Q*** is a length-preserving. We start with the leftmost columns of **R** (which has no zeroes) and calculate $n(1)$, for the c possible values of $\mathbf{s}_1$, according to (4). The decision which paths survive is made as with the Survivor algorithm.

For the next step, $t = 2$, we make use of the fact that the remaining columns of $\mathbf{R}_2,\ldots,\mathbf{R}_M$ span a hyperplane of dimension $M-1$. Shift this plane by adding an offset **y**. Now we the best and worst case **x**' estimates must also lie within this shifted hyperplane. The best case estimate is now made by assuming that all remaining $\mathbf{R}_j$ vectors point in the direction of *the projection of* **x**' *on the hyperplane*, as opposed to the direction of **x**' itself, as in the Survivor algorithm. The worst case estimates are made by assuming that the remaining $\mathbf{R}_j$ vectors point away from the projection of **x**' on the hyperplane.

This principle is illustrated in figure 3, which considers the same example as in figure 2. The coordinate transform, in this case is a rotation, which upper-triangularizes **R,** so that $\mathbf{R}_2$ is horizontal. The receive vector **x** has been rotated by the same angle, giving **x**'. The **y** vectors after one step are $\mathbf{R}_1$ and $-\mathbf{R}_1$. The best case estimates for the first **y** value, is $\mathbf{b}_1$, the projection of **x**' on the hyperplane $\mathbf{y} + \lambda\mathbf{R}_2$. If we define direction vector **z** to be the difference vector between the projection of **x**' and **y**, then the best case vector is given by

$$\mathbf{b} = \begin{cases} \dfrac{\mathbf{z}}{\|\mathbf{z}\|} \cdot \displaystyle\sum_{i=t+1}^{M} \|\mathbf{R}_i\| & \text{if } \|\mathbf{z}\| > \displaystyle\sum_{i=t+1}^{M} \|\mathbf{R}_i\| \\ \mathbf{z} & \text{otherwise} \end{cases} \tag{9}$$

The "otherwise case" of this equation deals with the situation where the sum of lengths of the remaining **R** columns overshoots the projection of **x**'. The norm of **b** gives the best case bound. $\mathbf{w}_1$ is the worst case vector, which lies within the hyper plane, but points in the opposite direction of **b**. Vectors $\mathbf{b}_2$ and $\mathbf{w}_2$ are the best and worst case vectors for the second node location **y**.

A further refinement of this algorithm, called QR2, does not simply look at the lengths of the remaining vectors, but takes their direction into account. It projects the remaining **R** onto direction vector **z**, *in the direction* of the projection of **x**'.

$$\mathbf{b} = \begin{cases} \dfrac{\mathbf{z}}{\|\mathbf{z}\|^2} \cdot \displaystyle\sum_{j=t+1}^{M} \left|\mathbf{R}_j^{\mathrm{T}} \cdot \mathbf{z}\right| & \text{if } \|\mathbf{z}\|^2 > \displaystyle\sum_{j=t+1}^{M} \left|\mathbf{R}_j^{\mathrm{T}} \cdot \mathbf{z}\right| \\ \mathbf{z} & \text{otherwise} \end{cases} \tag{10}$$

Since projections are shorted than the sum of lengths, as is shown in figure 4, the best case intervals are tighter and a further reduction in the number of evaluated nodes can be expected. Note that formula (10) covers the case of real valued vectors, where the absolute value of the inner product ensures that **b** is pointing in the same direction as **z**. Without proof we state that if $p = \mathbf{R}_j^{*} \cdot \mathbf{z}$ is a complex inner product that we must use p if $\mathrm{Re}(p) > 0$ and $-p$ otherwise.

VI. Performance Evaluation

We performed simulation for several systems with an equal number of transmit and receive antennas, ranging from 1 to 8. The transmit vector was BPSK modulated ($\mathbf{s}_i \in \{+1,-1\}$). The elements of the channel matrix have a complex Gaussian distribution with mean zero and $\mathrm{E}\{|\mathbf{H}_{ij}|^2\} = 1$. For each transmitted symbol, a new random channel is generated.

If the noise on each receive antenna is equal to σ^2, i.e. $\mathrm{E}\{\mathbf{n}_i^2\} = \sigma^2$, and the total transmitted power is 1, i.e. the power per transmit antenna is given by $\|\mathbf{s}_i^2\| = 1/M$, then the received power per receive antenna is equal to 1 and the SNR per receive antenna is given by:

$$\mathrm{SNR} = \sigma^{-2} \tag{11}$$

Figure 6 shows the symbol error probability versus the signal to noise ratio for different decoding algorithms, in a system with $N = M = 8$ antennas. Clearly, the maximum likelihood algorithm (whether implemented using the survivor, QR1 or QR2 algorithm) achieves optimal performance. The *K*-best algorithm approximates this performance for small signal to noise ratio. For larger signal to noise ratio, the symbol error rate starts to level off. Of the approximate algorithms, the *K*-best algorithm performs best. The 5-best algorithm achieves a SER of 10^{-4}, the 3-best algorithm has a SER which is an more order of magnitude higher. The performance of the myopic (1-best) and greedy algorithms is too poor to be practically useful in an 8-antenna system.

Figure 7 graphs the complexity in terms of number of evaluated nodes. The approximate algorithms that we considered all have a linear complexity in the number of transmit antennas *M*. The exact algorithms are all exponential in complexity, as is expected [7].

The Survivor algorithm is not very efficient in reducing complexity. A small reduction in complexity is achieved for small *M* (17% at most). The gain disappears for larger *M* larger than 5. The QR1 and QR2 algorithms do better , although is clear from the figure that their advantage starts to disappear as *M* becomes larger than 7. For a small number of antennas (less than 4) the QR1 and 2 methods are better than the approximate algorithms. For larger numbers, the approximate algorithms have less node evaluations.

Obviously the QR methods involve more complex operations, and it remains to be investigated whether the additional complexity expense for the QR factorization (which must be done after channel training) and the vector projections, does not outweigh the advantage of a smaller number of node evaluations. The *K*-best method is attractive for larger number of antennae, although value of *k*, must be increased with the number of transmit antenna to achieve satisfactory performance.

Figure 8 depicts the SER error floor, i.e. the performance of the various algorithms in the absence of noise for 2,3 up to 8 transmit and receive antennas. The graph shows that the myopic (1-best) algorithm has hardly practical value. Sorting **H** columns before applying the myopic algorithm (greedy), yields acceptable performance for 2 antennas. The performance quickly deteriorates for larger number of antennas. The *K*-best algorithm seems to perform well for a number of antennas that is *K*+1 to *K*+2. The MLD algorithms are not shown since their error floor is zero.

VII. Conclusions

The order *N* diversity receiver makes it possible, for a given single antenna system, to increase the bit rate by in increasing the number of transmit antennas. We maintain the same delay spread tolerance and signal bandwidth, while the reduction in the link budget, resulting from the fact that the available transmit energy must be shared among the transmit antennas, can be compensated by increasing the number of receive antenna. This opens the door to high-speed wireless systems which change *physical* limitations on link budget, delay spread, bandwidth, to *practical* limitations: cost of radio circuitry and signal processing logic. This paper discussed reduction in complexity of the latter complexity of the latter. The applicability of the algorithms presented in this paper extends to the generic multi-user detection problem.

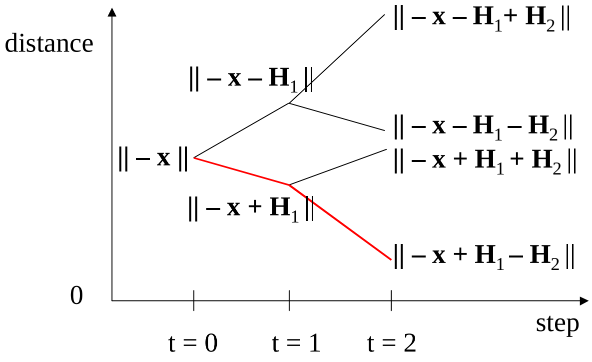

Figure 1: Maximum likelihood sequential decoding tree. On the vertical axis is the distance between x and H times the partially decoded s, on the horizontal axis the decoding step. $\mathbf{H}_i$ denotes the *i*-th column of H. The correct MLD solution is s = (+1,–1).

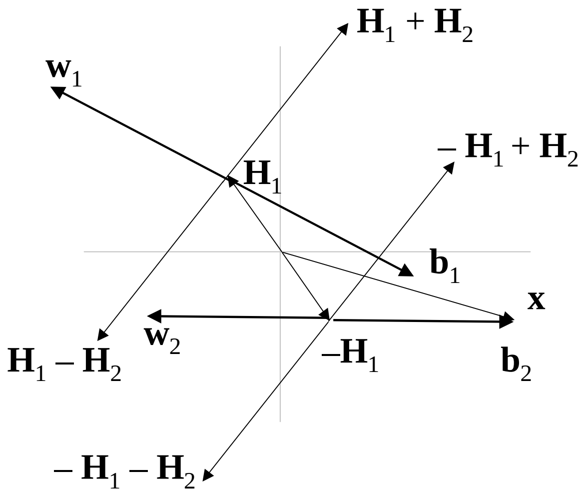

Figure 2: Example of Survivor decoding, *M* = *N* = 2, x and H are real, x uses BPSK modulation.

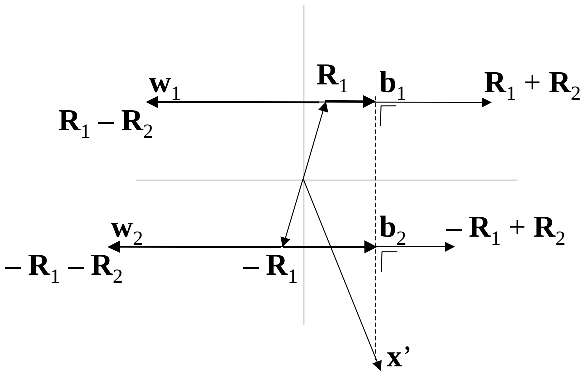

Figure 3: Example of QR survivor decoding, *M* = *N* = 2 x and H are real. $\mathbf{R}_j$ and x', are rotated channel matrix columns and receive vector, respectively. The points b and w denote best and worst case estimates, respectively. In this example, the best case estimates

coincide with the projection of x' on the R_2 hyperplanes.

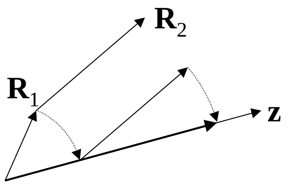

Figure 4: Difference between QR-Survivor decoding methods 1 and 2. Algorithm QR1 normalizes direction vector z and multiplies by the total length of the remaining R columns, while algorithm QR2 uses the projections of the remaining R columns on the direction vector.

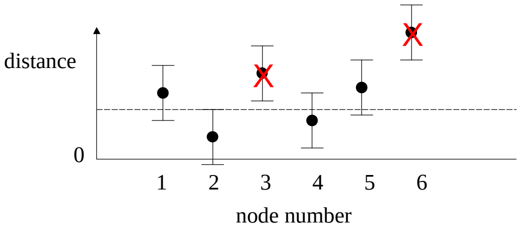

Figure 5: Node Distance versus node number, showing error intervals after partial decoding. The dashed line shows the best worst case distance. All nodes with a worst case distance, crossed out in this figure, are to be eliminated.

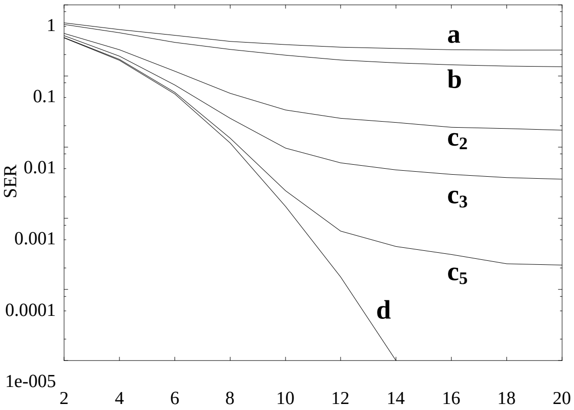

Figure 6: Bit and Symbol Error Rate versus Signal to noise ratio (SNR) for (a) greedy, (b) myopic, (c_K) *K*-best (d) exhaustive MLD for an $N = M = 8$ system.

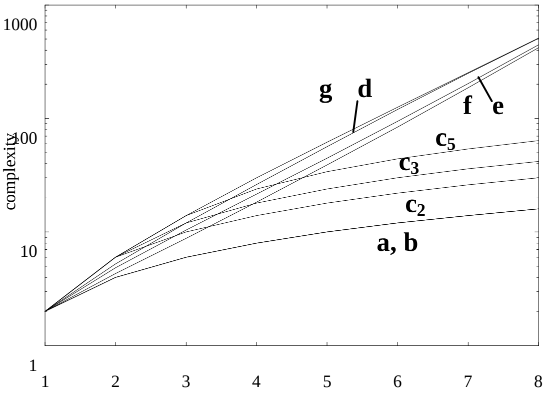

Figure 7: Node evaluation complexity versus *N*, *M* (number of TX, RX antennae), no noise, for (a) greedy, (b) myopic, (c_k) *K*-best (d) Survivor, (e) QR1, (f) QR2 and (g) exhaustive.

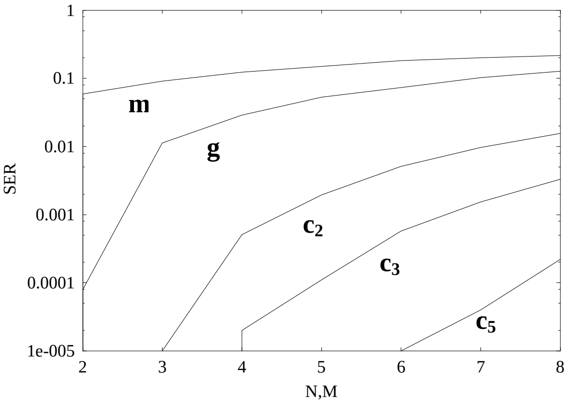

Figure 8: SER, floor versus *N*, *M* (number of TX, RX antennae) for (a) greedy, (b) myopic, (c_K) *K*-best, no noise.

REFERENCE

[1] R. van Nee, A. van Zelst and G. Awater “Maximum Likelihood Decoding in a Space Division Multiplexing System,” in proceedings IEEE VTC 2000, Tokyo, Japan, May 15–18, 2000.

[2] A. van Zelst, R. van Nee and G. Awater, “Space Division Multiplexing (SDM) for OFDM Systems,” in proceedings IEEE VTC 2000, Tokyo, Japan, May 15–18, 2000.

[3] G. D. Golden, C. J. Foschini, R. A. Valenzuela, P. W. Wolniansky, “Detection Algorithm and Initial Laboratory Results Using V-BLAST Space-Time Communication Architecture,” Electronics Letters, Vol. 35, No. 1, January 7, 1999.

[4] J. B. Anderson, S. Mohan, “Sequential Coding Algorithms: A survey and Cost Analysis," IEEE Transactions on Communications, Vol. 32, No. 5, September 1989, pp. 169–176.

[5] J. B. Anderson, “Limited Search Trellis Decoding of Convolutional Codes,” IEEE Transactions on Information Theory, Vol. 35, No. 5, September 1989, pp. 944–955.

[6] R. A. Horn, C. R. Johnson, “Matrix Analysis,” Cambridge University Press, Cambridge, UK, 1985.

[7] S. Verdú, “Computational Complexity of Optimum Multiuser Detection,” Algorithmica, Vol. 4, pp. 303–312.