APPROXIMATE ML DETECTION FOR MIMO SYSTEMS WITH VERY LOW COMPLEXITY

Markus Rupp, Gerhard Gritsch, Hans Weinrichter

Vienna University of Technology,Institute of Communications and Radio-Frequency Engineering
Gusshausstrasse 25/389, 1040 Vienna, Austria (Europe)
email: (mrupp,ggritsch,jweinri)@nt.tuwien.ac.at

ABSTRACT

Recently many Space-Time Coding schemes for multiple antenna systems (MIMO) have been proposed in order to achieve high data rate when transmitting over wireless channels. However, most of such schemes rely on Maximum Likelihood (ML) detection which can become quite complex when many antennas are involved and higher modulation schemes are utilized. On the other hand, the high diversity gains of MIMO channels are easily lost when low-complexity receivers like ZF or MMSE are applied. It is thus of utmost importance to look for low-complexity receivers which achieve almost ML performance. In this paper, a new scheme for approximate ML detection for typical flat Rayleigh fading channels is proposed achieving ML performance in the area of Bit Error Ratio between 10^{-4} and 0.1 as it is of interest in wireless communications. The proposed scheme allows to transmit 64QAM schemes on 4×4 antenna schemes with a detection complexity of only 1% of a brute force ML receiver.

1. INTRODUCTION

Once implementation of algorithms under real-time constraints becomes an issue, algorithms may exhibit specific difficulties. The Maximum Likelihood (ML) detection is a scheme well-known to be very robust and well suited for practical implementation offering best detection performance whereas linear receivers such as ZF or MMSE suffer from numerical challenges. The ML detector calculates the squared distance d_i^2 between the received vector $\mathbf{y}$ and every possible signal constellation $\mathbf{x}_i$:

$$d_i^2 = \|\mathbf{y} - \mathbf{H}\mathbf{x}_i\|^2 .$$

$\mathbf{H}$ denotes the MIMO channel matrix tpically of a flat Rayleigh fading channel and $\|.\|$ denotes the l_2-norm operator. The number of calculations is $|\mathscr{A}|^{n_T}$, whereby $|\mathscr{A}|$ and n_T denote the size of the symbol alphabet and the number of transmit antennas, respectively. For this reason, its complexity grows exponentially with the number of signal points and transmit antennas.

Therefore, with the growing complexity of multiple input-multiple output systems (MIMO) the ML detection scheme becomes too complex to implement. Depending on the utilized modulation scheme, many simplifications have been proposed [1]-[3] converting expensive - to - realize multiplications into cheap add/sub-structures. However, since ML detection requires a full search its exponential complexity still remains. Lowering the complexity of the basic operation involved does not solve this problem but only eases it. Other strategies [4]-[6] turn the exponential complexity into a polynomial one.

This paper proposes a new approximate ML detection scheme with much lower complexity. All operations can be implemented in simple add/sub structures as proposed in [1]-[3], but in order to save much more complexity a pre-selection scheme has been invented that selects such parts of the constellation map that are most likely to be among the ML-candidates. By such pre-selection scheme it is not required to search through all possible combinations of transmit-symbols and thus saves a tremendous amount of complexity. Note that the complexity of the proposed scheme is still exponentially increasing with n_T.

In the following Sec. 2 the principle of the pre-selection scheme is explained for 16QAM modulation. Simulation results of the approximate ML detector are compared to the brute force ML detector performance. In Sec. 3 the principle is extended to 64QAM. Conclusions close the paper.

This work has been funded by the Christian Doppler Pilot Laboratory for Design Methodology of Signal Processing Algorithms.

2. APPROXIMATE ML-DETECTION FOR 16QAM

This contribution focuses on MIMO systems with four transmit antennas (n_T=4) and four receive antennas (n_R=4). The brute force ML detector has to calculate $|\mathscr{A}|^4$ distances. If 4QAM is used as modulation scheme, then $|\mathscr{A}|^4 = 4^4 = 256$ distances have to be calculated. The resulting complexity is not too large for implementation, but if modulation formats with higher symbol alphabet like 16QAM or 64QAM are used, then the number of distance calculations grows extremely, namely to $|\mathscr{A}|^4 = 16^4 = 65.536$ or to $|\mathscr{A}|^4 = 64^4 = 16.777.216$, respectively. Especially, today the 64QAM system is not practically realizable, because of the limited capability of nowadays processors.

2.1. Principle of Reducing the Search Set

The search for the symbol constellation (transmit vector) with the smallest distance from the received vector $\mathbf{y}$ is performed in two steps.

The first step is to find out the quadrant constellation which fits best. This is done by representing each quadrant by one signal point in the center of the quadrant. For this reason, the symbol alphabet is reduced to four signal points (green '$\times$' in Fig. 1), instead of 16 points (blue '$+$'). With this reduction, there are only $4^{n_T} = 4^4 = 256$ possible symbol vectors (quadrant constellations) $\mathbf{x}_i^q$ to be checked. These special symbol vectors $\mathbf{x}_i^q$ are called quadrant constellation vectors. For these 256 quadrant constellation vectors, the squared distance d_i^2 from the received vector $\mathbf{y}$ has to be calculated:

$$d_i^2 = \|\mathbf{y} - \mathbf{H}\mathbf{x}_i^q\|^2 , i = 1..256. \quad (1)$$

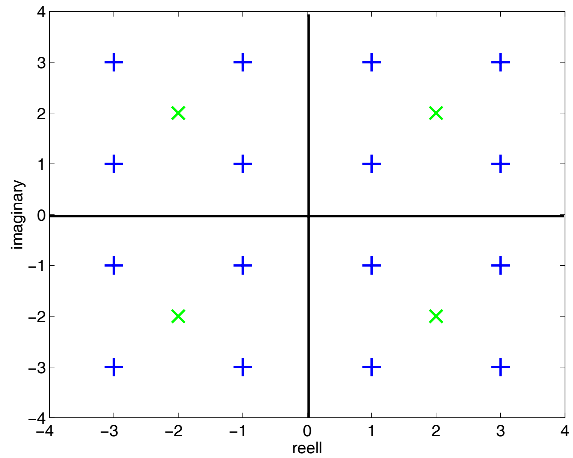

Figure 1: 16QAM signal constellation (blue '+') and representing points of the quadrants (green '×').

Note that the calculated distance (Eqn. (1)) is not the true distance in terms of ML detection. This calculated distance is the distance to a representative signal constellation and therefore the resulting minimum is not necessarily the minimum in the ML sense. For this reason, a systematic error (compared to ML detection) is made. This calculation (Eqn. (1)) can be performed without multiplication since the quadrant constellation points follow a 4QAM (see [1] for more details). The final distance d_i^2 according to (1) can also be computed by an l_2-norm approximation without losing much performance as will be explained further ahead. Ordering those values by magnitude, a distance profile is received. As an intermediate step, only those n_{OCC} values with smallest distance are selected for further processing.

The second step is to perform the ML search process based on these selected values, i.e., for each of these n_{OCC} quadrant constellations the corresponding nearest neighbor 16QAM signal constellations (16QAM signal vectors) are determined. This is done by utilizing the 16QAM signal points, which are the nearest neighbors of the predetermined quadrant constellation points. For example, in Fig. 1 the nearest neighbor 16QAM signal points of the upper left quadrant point are simply the signal points (blue '+') surrounding the upper left green '×'. For this reason, for each quadrant point there are four 16QAM signal points. Because of four transmit antennas and four possible 16QAM signal points per transmit antenna, once again the distances of 256 possible signal constellations have to be calculated. For all ($n_{\mathrm{OCC}} \times 256$) 16 QAM signal constellations, the distances are calculated, finally selecting the one with the smallest value.

Note that the center value is connected to the final four constellation points by a simple relation. Each quadrant constellation point is surrounded by another 4QAM constellation with half the length of the previous 4QAM, as explained above (see again in [1]-[3] for more details). Thus, the previous results on the quadrant constellation points can be reused as a starting point. Thus, even at this second step, no multiplication is required.

2.2. Complexity

This section focuses on the complexity of our proposed receiver. As mentioned in the previous subsection, the total search is subdivided into two parts. For the first part, 4^{n_T} distances have to be calculated. For one distance calculation, 32 complex add/sub operations are necessary, whereby shift operations (which have extremely low complexity) and the effort for calculating the squared norm are neglected. As mentioned above, the l_2-norm can be calculated by an approximation without requiring multipliers. For the complexity calculation the number of operations of this approximation is also neglected, because these operations are the same for both, the ML and the Approximate ML. The number of computations in the second step depends on the number of considered quadrant constellations with minimum distance: n_{OCC}. The distance calculations in the second step needs the same number of complex add/sub operations as in the first step. Therefore, the whole complexity in terms of complex add/sub operations $n_{\mathrm{CAS,AML}}$ can be written as:

$$n_{\mathrm{CAS,AML}} = 32\, n_{\mathrm{DC,AML}} = 32\,(1 + n_{\mathrm{OCC}})\, 4^{n_T}\ .$$

Where $n_{\mathrm{DC,AML}}$ denotes the number of distance calculations and the index 'AML' denotes Approximate ML.

As mentioned in the introduction, the number of distance calculations for a ML Detector can be calculated as:

$$n_{\mathrm{DC,ML}} = |\mathscr{A}|^{n_T} = 16^{n_T}\ .$$

Note that in this case one distance calculation needs 16 complex multiplications and 16 complex add/sub operations and these multiplications are much more expensive regarding gate count and/or computation time. Obviously, the reduction of complexity of our detector compared with a ML detector depends on n_{OCC} and on the corresponding gate numbers necessary for each operation. For the special case of 16QAM and $n_T = 4$: $n_{\mathrm{DC,ML}} = 16^4 = 65.536$. The approximate ML detector on the other hand needs only $n_{\mathrm{DC,AML}}$=256, 2.816, 5.376, 7.963, and 10.496 for n_{OCC} =1, 10, 20, 30, and 40, respectively. Assuming that the distance calculation of the ML costs as much as for the approximate ML (a very pessimistic comparison for our proposed scheme), the complexity of the proposed scheme is only 0.39%, 4.30%, 8.20%, 12.11% and 16.02% of the brute force ML Detector, respectively.

2.3. Simulation Results

2.3.1. *Exact calculation of the l_2-norm*

To evaluate the performance of our proposed detector, simulations were performed, with the exactly calculated l_2-norm. The BER for several values of Signal to Noise Ratio (SNR) and five values of n_{OCC}=1,10,20,30 and 40 were simulated. The channels were independent identically distributed complex Gaussian values with zero mean and unit variance. These parameterized BER-curves are depicted in Fig. 3. The simulated performance of the approximate ML detector is compared to the simulated performance of the brute force ML detector, which is the dashed dotted dark green curve in Fig. 3 (labeled by 'ML'). The blue, red, black, green and magenta BER-curves (labeled by 'n_{OCC}=1, 10, 20, 30 and 40') show the performance of our detector for n_{OCC}=1, 10, 20, 30 and 40. As can be seen, the performance in the low SNR domain is equal to the ML detector and the approximate ML behaves identical regardless which n_{OCC}-value is used. Bit errors are caused by noise in a similar way as for the brute force ML detector. In the high SNR domain, the ML detector has obviously the best performance. The performance of the approximate ML strongly depends on the parameter n_{OCC}. For all applied n_{OCC}-values the BER-curves end in an error floor. Here, bit errors are no longer caused by noise, but by the systematic error which is intended to reduce the search set and thus the search time. The larger the parameter n_{OCC} is, the higher is the probability of finding the true ML candidate. Therefore, the influence of the systematic error and thus the level of the error floor can be controlled by the parameter n_{OCC}. Note that for n_{OCC}=40 the BER performance equals the ML performance down to a BER of 10^{-4}. Thus, for a typical BER-range in wireless systems ML performance can be achieved with only 16% of the ML complexity.

2.3.2. *Approximation of the l_2-norm*

In order to obtain a complete multiplier free realization of the proposed detector, the l_2-norm is approximated by:

$$\begin{aligned} |d| &\approx \frac{5}{8}\left(|\Re\{d\}|+|\Im\{d\}|\right)+\frac{3}{8}\max\left(|\Re\{d\}|,|\Im\{d\}|\right) \\ &= \begin{cases} |\Re\{d\}|+\frac{5}{8}|\Im\{d\}| & ;|\Re\{d\}|>|\Im\{d\}| \\ |\Im\{d\}|+\frac{5}{8}|\Re\{d\}| & ;\text{else}, \end{cases} \end{aligned} \quad (2)$$

a common approximation in hardware implementations [7], where $\Re\{d\}$ and $\Im\{d\}$ denote the real part of the distance d and the imaginary part of the distance d, respectively. To give the reader a flavour how tight this approximation is, the approximation (Eqn. (2)) and the true value for the l_2-norm are shown in Fig. 2. With this ap-

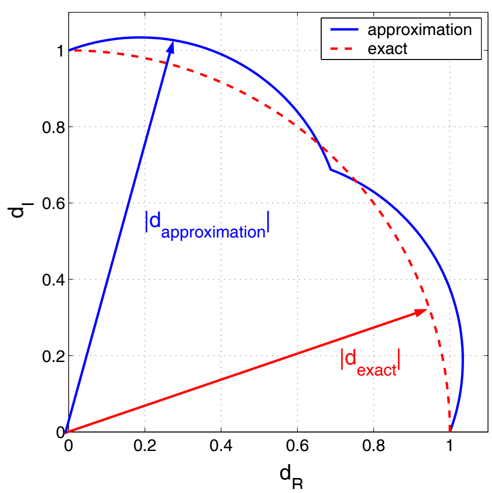

Figure 2: Exact l_2-norm (red dashed line) and the l_2-norm approximation (blue solid line).

proximation, the entire signal processing of the detector does not require multiplications and therefore the hardware realization of such a receiver is very efficient and fast. The performance loss of the receiver using the approximation compared to the exact l_2-norm is shown in Fig. 3. As it can be further seen in Fig. 3, the performance degradation of the complete multiplier free detector is quite small.

3. APPROXIMATE ML-DETECTION FOR 64QAM

For 64QAM the detection consists of three steps. The first step searches for the n_{OCC1} best fitting quadrants, whereby each quadrant is represented by one signal point (red circles in Fig. 4). These n_{OCC1} best fitting quadrant constellations are used to calculate the start values for the second search step. In the second search step, the best fitting intermediate signal points are determined. This is done

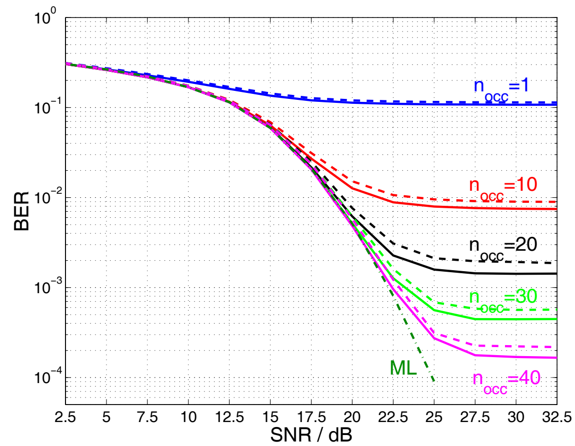

Figure 3: Exact l_2-norm performance (solid lines) and the l_2-norm approximation performance (dashed lines) of 16QAM Approximate ML receiver.

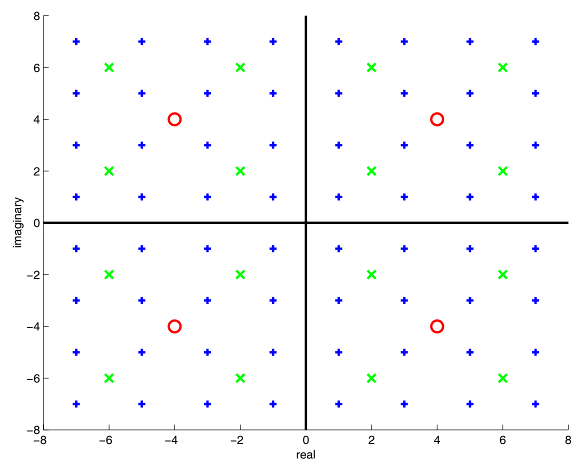

Figure 4: The 64QAM signal constellation points (blue '+'), the quadrant signal constellation points (red circles) and the intermediate step signal points (green '×').

by utilizing the intermediate signal points, which are the nearest neighbors of the predetermined quadrant constellation points. For example, in Fig. 4 the nearest neighbor intermediate signal points of the upper left quadrant point are simply the signal points (green '×') surrounding the upper left red circle. For this reason, for each quadrant point there are four intermediate signal points. Because of four transmit antennas and four possible intermediate signal points per transmit antenna, once again the distances of 256 possible signal constellations have to be calculated. As mentioned above, each quadrant constellation is used to calculate one start value for the second search. From each quadrant constellation, there are 256 intermediate signal constellations. From the first step, there are n_{OCC1} quadrant constellations and therefore $n_{OCC1}\times 256$ distances have to be calculated. The n_{OCC2} best fitting intermediate signal constellations corresponding to the n_{OCC2} smallest distances are stored. Now, there are n_{OCC2} intermediate signal constellations per quadrant constellation and thus in total $n_{OCC1}\times n_{OCC2}$ intermediate signal constellations. These $n_{OCC1}\times n_{OCC2}$ constellations are new start values for the third and last search. From each intermediate constellation the corresponding 256 nearest neighbor 64QAM signal constellations (64QAM signal vectors) can easily be calculated by repetitive add/sub operations. For all of these $(n_{OCC1}\times n_{OCC2}\times 256)$ 64QAM signal constellations, the distances are calculated, finally selecting the one with the smallest value. Note that also in the case of 64QAM only add/sub operations are required. A significant difference between the 16QAM scheme and 64QAM scheme is that the whole search is subdivided into three parts instead of two parts for 16QAM. Therefore, the potential of reducing the search set and thus the search time is doubled. For this reason, the relative reduction in computational complexity for the 64QAM detector is higher than for the 16QAM receiver.

3.1. Complexity

The complexity analysis of the 64QAM scheme follows the same path as for the 16QAM scheme. 256 distances have to be calculated to find out the n_{OCC1} best fitting quadrants (first search), $n_{OCC1}\times 256$ distances have to be calculated to find the $n_{OCC1}\times n_{OCC2}$ best fitting intermediate signal constellation (second search) and the third and last search needs to calculate $n_{OCC1}\times n_{OCC2}\times 256$ distances. One distance calculation requires approximately 32 complex add/sub operations like in the 16QAM case. In total, the number of complex add/sub operations $n_{CAS,AML}$ is:

$$n_{CAS,AML} = 32\, n_{DC,AML} = 32\,(1 + n_{OCC1} + n_{OCC1}\, n_{OCC2})\, 4^{n_T} \;.$$

The BER vs. SNR curves were simulated for n_{OCC1}=n_{OCC2}=n_{OCC}= 1, 10, 20, and 30. Note, n_{OCC1}=n_{OCC2}=n_{OCC} is only an arbitrary choice. The resulting number of distance calculations is: $n_{DC,AML}$ = 256, 28.416, 107.776 and 238.336. Thus, the relative complexity compared to the brute force ML is: 0.001%, 0.169%, 0.642% and 1.421%.

3.2. Simulation Results

BER vs. SNR curves were simulated for several values of n_{OCC1} = n_{OCC2} = n_{OCC}. Fig. 5 shows the results of the simulations for exact l_2-norm (solid lines) and l_2-norm approximation (dashed lines). The brute force ML detector performance is indicated by the dashed dotted dark green line (labeled by 'ML'). The principle behavior of the BER-curves is the same as for the 16QAM case. As for the 16QAM case, the BER-performance of the approximate ML detector follows the performance of the brute force ML detector (low SNR domain - errors caused by noise) until it ends in an error floor (high SNR domain - errors caused by the systematic error). The level of the error floor can be controlled by the parameter n_{OCC}. An important difference to the 16QAM case is, that a certain BER is achieved with a lower relative complexity. For example, for the 16QAM case a BER of approximately 10^{-3} is achieved with a relative complexity of 10%, for 64QAM this BER is achieved with 2% relative complexity.

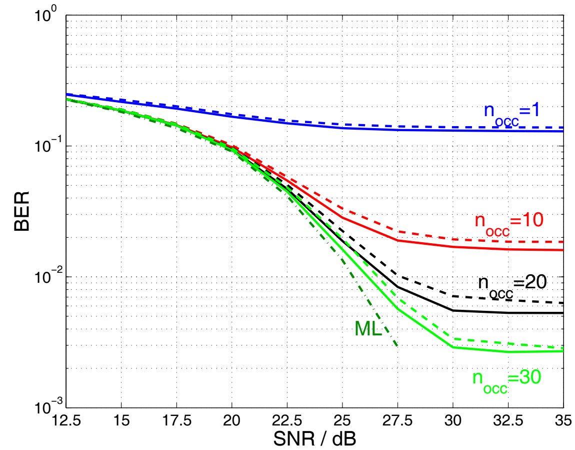

Figure 5: Exact l_2−norm performance (solid lines) and the l_2−norm approximation performance (dashed lines) of 64QAM Approximate ML receiver.

4. CONCLUSION

Our proposed receiver, the approximate ML-Detector, has the advantage that its complexity compared with the ML detector is still of exponential order but in absolute terms substantially lower. Its performance follows the performance of the brute force ML until the BER ends in an error floor, which is the most important disadvantage of this approximate ML detector. The error floor is due to the systematic error, because of the search set reduction, of our scheme. Obviously, there is a tradeoff between complexity and the level of the error floor (n_{OCC}). For 16QAM the complexity reduction is not that large. For a BER≈10^{-3} the complexity is approximately reduced to 10% of the brute force ML. In the case of 64QAM the relative reduction in computational complexity is very high. With a complexity of approximately 2% compared with the brute force ML Detector, a BER of approximately 10^{-3} is achieved. If the subsequent processing or the application can live with relative high BERs due to the error floor, then this receiver is one possibility to save a lot of computation time compared to the brute force ML detector.

5. ACKNOWLEDGMENT

The authors would like to thank Prof. Ernst Bonek for support and encouragement.

REFERENCES

[1] M.Rupp, J.Balakrishnan, "Efficient chip design for pulse shaping", SPAWC 99 in Anapolis, pp. 304-307, May 1999.

[2] H.L.Lou, M.Rupp, R.L.Urbanke, H.Viswanathan, R.Krishnamoorthy, "Efficient implementation of parallel decision feedback decoders for broadband applications", IEEE Electronics, Circuits and Systems Conference, Cyprus, pp.1475-1478, Sept. 1999.

[3] M.Rupp, H.L.Lou, "On Efficient Multiplier-Free Implementation of Channel Estimation and Equalization", Globecom 2000, San Francisco, pp. 6-10, Nov. 2000.

[4] U. Fincke, M.Phost, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis", Math. of Comp., vol. 44, pp. 463-471, April 1985

[5] M.O.Damen,H.El-Gamal,G.Caire, " On maximum-likelihood detection and the search for the closest lattice point," IEEE Trans. IT, pp. 2389-2402, Oct. 2003.

[6] H. Artés, D. Seethaler, F. Hlawatsch, "Efficient Detection Algorithms for MIMO Channels: A Geometrical Approach for Approximate ML Detection", IEEE Trans. Signal Processing, Special Issue on Signal Processing for MIMO Wirel. Com. Syst., pp. 2808-2820, Nov. 2003

[7] A.Adjoudani, E.Beck, A.Burg, G.M.Djuknic, T.Gvoth, D.Haessig, S.Manji, M.Milbrodt, M.Rupp, D.Samardzija, A.Siegel, T.Sizer II, C.Tran, S.Walker, S.A.Wilkus, P.Wolniansky, "Prototype Experience for MIMO BLAST over Third Generation Wireless System," Special Issue JSAC on MIMO Systems, vol. 21, pp. 440-451, April 2003.

[8] John G. Proakis, "Digital Communications", 4th ed. McGraw-Hill, Inc. 2002.