Application of Sphere Decoding to MSDD

Lutz Lampe, Robert Schober
Dept. of Electrical and Computer Engineering
The University of British Columbia
Canada

Volker Pauli, Christoph Windpassinger
Lehrstuhl für Informationsübertragung
Universität Erlangen-Nürnberg
Germany

Abstract

In multiple-symbol differential detection (MSDD) for power-efficient transmission over fading channels without channel state information, blocks of N received symbols are jointly processed to decide on $N-1$ data symbols. The search space for the maximum-likelihood (ML) estimate is therefore (complex) $N-1$ dimensional, and MSDD quickly becomes computationally intractable as N grows. Sphere decoding (SD) is a widely used approach to find the ML estimate in such high-dimensional spaces. In this paper, we devise the application of SD to accomplish MSDD, and we refer to the resulting technique as multiple-symbol differential sphere decoding (MSDSD). We present an efficient algorithm for MSDSD, whose excellent performance versus complexity trade-off is verified by various simulation results and by comparisons with other, suboptimum approaches known from the literature.

1 Introduction

Maximum-likelihood (ML) decoding (or detection) algorithms with polynomial expected complexity, collectively referred to as sphere decoding (SD) algorithms, have recently attracted considerable attention in multiple-input multiple-output (MIMO) communication problems, cf. e.g. [1]-[6]. In this context, the problem usually considered is to find the solution

$$\hat{\boldsymbol{s}} = \underset{\boldsymbol{s}}{\operatorname{argmin}} ||\boldsymbol{r} - \boldsymbol{H}\boldsymbol{s}||^2 \tag{1}$$

for an N_s-dimensional[1] input $\boldsymbol{s}$ transmitted over an $N_r \times N_s$-dimensional channel $\boldsymbol{H}$ ($N_r \geq N_s$) with N_r-dimensional output $\boldsymbol{r}$ ($||\cdot||$: Euclidean norm). Given that the channel $\boldsymbol{H}$ is known to the receiver, (1) can be rewritten as [6]

$$\hat{\boldsymbol{s}} = \underset{\boldsymbol{s}}{\operatorname{argmin}} ||\boldsymbol{U}(\boldsymbol{s} - \boldsymbol{s}_{\mathrm{LS}})||^2 \tag{2}$$

where the Cholesky factorization $\boldsymbol{H}^H\boldsymbol{H} = \boldsymbol{U}^H\boldsymbol{U}$ with upper triangular matrix $\boldsymbol{U}$ and the unconstrained least-squares solution $\boldsymbol{s}_{\mathrm{LS}} \in \mathbb{C}^{N_s}$ are used ($.^H$: Hermitian transpose).

In this paper, we consider the application of SD to multiple-symbol differential detection (MSDD) of differential phase-shift keying (DPSK) over an *unknown* frequency-nonselective (flat) fading channel. In MSDD, N consecutively received samples are collected for joint detection of $N-1$ consecutively transmitted data symbols, cf. e.g. [7], [8]. Due to this block processing structure, we can model transmission with MSDD as MIMO system with $N_s = N-1$, $N_r = N$. It is known that for optimum MSDD with respect to the ML criterion (ML-MSDD) error-rate performance steadily improves with increasing dimension or *observation window size* N. On the other hand, detection complexity grows exponentially in N. In order to significantly reduce this detection complexity while maintaining ML performance, we propose to apply SD to MSDD, which we refer to as multiple-symbol differential sphere decoding (MSDSD)[2]. MSDSD is motivated by the observation that by Cholesky factorization of the channel statistics, a decision metric of the same form as (2) can be obtained. Hence, we can quite straightforwardly extend SD strategies originally designed for (1), i.e., for MIMO detection with channel state information (CSI) at the receiver, to MSDD without CSI.

In the recent past, several low-complexity implementations for MSDD have been proposed. Mackenthun [9] developed an algorithm with complexity of the order $N\log_2(N)$ per N symbol block. This algorithm is only equivalent to ML-MSDD for channels that are time-invariant over an N symbol period. For the general time-varying flat fading case, linear-prediction based decision-feedback differential detection (DF-DD) receivers have been proposed, cf. e.g. [10], [11], [12]. Although DF-DD does not achieve the performance of ML-MSDD, it offers substantial performance improvements over conventional differential detection (CDD) with $N = 2$ while its per-symbol complexity is only linear in N. Other suboptimum *ad hoc* algorithms use a two-step strategy, e.g. [13], [14], [15]. First, a candidate list is set up by using CDD with $N = 2$, and then

[1] Since we use the equivalent complex baseband description, in general all vector and matrix components are complex-valued.

[2] In the context of this paper, the terms "decoding" and "detection" refer to the same procedure and are used interchangeably. For "MSDSD" we stick with the term "decoding" since it is preferred in the SD literature.

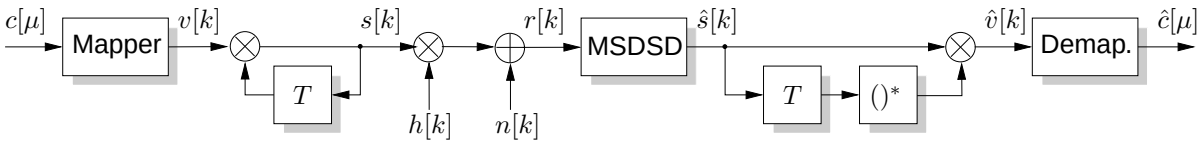

Fig. 1. Block diagram of differential transmission and MSDD.

these candidates are tested using the MSDD metric with increased window size. Again, these detection schemes do not achieve the performance of ML-MSDD. This is in contrast to the proposed MSDSD, which in fact performs true ML-MSDD, but keeps detection complexity reasonably low by the application of SD.

This paper is organized as follows. Section 2 briefly introduces the system model for DPSK transmission with MSDD. In Section 3, the new detection technique is derived and the MSDSD algorithm is explained in detail. Simulation results illustrating the favorable performance and complexity properties of MSDSD are presented in Section 4. Finally, Section 5 concludes the paper.

2 System Model

The block diagram of our discrete-time communication system in the equivalent complex low-pass domain is depicted in Figure 1. At the transmitter, we apply the classical M-ary DPSK [16]. $\log_2(M)$ binary data symbols $c[\mu]$ are Gray mapped to M-ary data-carrying differential symbols $v[k]$ taken from the MPSK signal constellation $\mathcal{V} \triangleq \{v = \mathrm{e}^{\mathrm{j}2\pi m/M} | m = 0, 1, \ldots, M-1\}$ ($\mu, k \in \mathbb{Z}$: binary and M-ary symbol discrete-time index, respectively). From $v[k]$ the current transmit symbol $s[k]$ is obtained via differential encoding

$$s[k] = v[k]s[k-1] \; . \tag{3}$$

The transmission channel is assumed frequency-nonselective and slowly time-varying during one modulation interval of length T and is represented by the complex fading gain $h[k]$. The received sample $r[k]$ can be modelled as

$$r[k] = h[k]s[k] + n[k] \; , \tag{4}$$

where $n[k]$ denotes the additive white Gaussian noise (AWGN) with variance σ_n^2. We consider $h[k]$ to be zero-mean complex Gaussian distributed (Rayleigh fading) with autocorrelation function

$$\varphi_{hh}[\kappa] \triangleq \mathcal{E}\{h[k+\kappa]h^*[k]\} = \mathrm{J}_0(2\pi B_f T \kappa) \tag{5}$$

($\mathcal{E}$: expectation, $\mathrm{J}_0(\cdot)$: zeroth order Bessel function of the first kind) according to the widely used Clarke fading model with maximum normalized fading bandwidth $B_f T$.

At the receiver, the multiple-symbol differential sphere decoder processes N consecutively received samples

$$\boldsymbol{r}[k_N] \triangleq [r[(N-1)k-(N-1)] \; \ldots \; r[(N-1)k]]^T$$

(k_N: vector symbol discrete-time index) to obtain ML estimates $\hat{s}[k]$ of the corresponding N transmit symbols

$$\boldsymbol{s}[k_N] \triangleq [s[(N-1)k-(N-1)] \; \ldots \; s[(N-1)k]]^T \; ,$$

and via differential decoding estimates $\hat{v}[k]$ of the $N-1$ differential symbols

$$\boldsymbol{v}[k_N] \triangleq [v[(N-1)k-(N-2)] \; \ldots \; v[(N-1)k]]^T \; .$$

Thereby, no explicit channel state information, i.e., no knowledge of

$$\boldsymbol{h}[k_N] \triangleq [h[(N-1)k-(N-1)] \; \ldots \; h[(N-1)k]]^T$$

is required. We note that due to differential encoding consecutive blocks $\boldsymbol{r}[k_N]$ overlap by one scalar received symbol [8]. The value of N is referred to as the observation window size, and since with growing N the memory of the fading process is more completely taken into account, the performance of MSDSD shall improve with increasing N [17].

3 Multiple-Symbol Differential Sphere Decoding

In this section, the MSDSD algorithm is derived. For this purpose, the decision rule of ML-MSDD is reviewed and adapted to the application of SD in Section 3.1. Then, Section 3.2 provides the actual SD algorithm. For the sake of compact presentation, we omit time indices k_N, k and use x_i to refer to the ith component of vector $\boldsymbol{x}$.

3.1 ML-MSDD

ML-MSDD delivers the ML estimate $\hat{\boldsymbol{s}}$ based on the observation of $\boldsymbol{r}$. Hence, the decision rule reads

$$\hat{\boldsymbol{s}} = \underset{\boldsymbol{s}}{\operatorname{argmax}}\{p(\boldsymbol{r}|\boldsymbol{s})\} \; , \tag{6}$$

where $p(\boldsymbol{r}|\boldsymbol{s})$ denotes the conditioned probability density function (pdf) of $\boldsymbol{r}$ given $\boldsymbol{s}$. Inserting the expression for the pdf for the case of Rayleigh fading considered in this paper, e.g. [17, Eq. (6)], (6) can be written as

$$\hat{\boldsymbol{s}} = \underset{\boldsymbol{s}}{\operatorname{argmin}}\{\boldsymbol{r}^H \boldsymbol{R}_{\boldsymbol{rr}}^{-1} \boldsymbol{r}\} \; , \tag{7}$$

with the correlation matrix

$$\boldsymbol{R}_{\boldsymbol{rr}} \triangleq \mathcal{E}\{\boldsymbol{r}\boldsymbol{r}^H | \boldsymbol{s}\} \; . \tag{8}$$

We can further use the relations ($\mathrm{diag}\{\boldsymbol{x}\}$: diagonal matrix with components of $\boldsymbol{x}$ on main diagonal, $.^*$:

(componentwise) complex conjugate, $\boldsymbol{I}_N$: $N \times N$ identity matrix)

$$\begin{aligned} \boldsymbol{R}_{\boldsymbol{rr}} &= \operatorname{diag}\{\boldsymbol{s}\}\boldsymbol{C}\operatorname{diag}\{\boldsymbol{s}^*\}\,, & (9)\\ \boldsymbol{C} &\triangleq \boldsymbol{R}_{\boldsymbol{hh}} + \sigma_n^2\boldsymbol{I}_N\,, & (10)\\ \boldsymbol{R}_{\boldsymbol{hh}} &\triangleq \mathcal{E}\{\boldsymbol{h}\boldsymbol{h}^H\}\,, & (11)\\ \operatorname{diag}\{\boldsymbol{s}^*\}\boldsymbol{r} &= \operatorname{diag}\{\boldsymbol{r}\}\boldsymbol{s}^*\,, & (12) \end{aligned}$$

to rewrite decision rule (7) as

$$\hat{\boldsymbol{s}} = \underset{\boldsymbol{s}}{\operatorname{argmin}}\{(\operatorname{diag}\{\boldsymbol{r}\}\boldsymbol{s}^*)^H\boldsymbol{C}^{-1}\operatorname{diag}\{\boldsymbol{r}\}\boldsymbol{s}^*\}\,. \quad (13)$$

Apparently, expression (13) is a quadratic form in $\boldsymbol{s}$. Therefore, we apply the Cholesky factorization of the inverse matrix

$$\boldsymbol{C}^{-1} = \boldsymbol{L}\boldsymbol{L}^H \quad (14)$$

and further define

$$\boldsymbol{U} \triangleq (\boldsymbol{L}^H\operatorname{diag}\{\boldsymbol{r}\})^*\,, \quad (15)$$

where $\boldsymbol{L}$ and $\boldsymbol{U}$ are lower and upper triangular matrices, respectively, to finally arrive at

$$\hat{\boldsymbol{s}} = \underset{\boldsymbol{s}}{\operatorname{argmin}}\{||\boldsymbol{U}\boldsymbol{s}||^2\}\,. \quad (16)$$

We note that decision rule (16) equals (2) with all-zero vector $\boldsymbol{s}_{\mathrm{LS}}$. Hence, ML-MSDD can be regarded as a shortest vector problem, which in many case can efficiently be solved by SD, cf. [3], [18] for transmission with CSI. We note that the brute-force solution to (16) is to test M^{N-1} vectors $\boldsymbol{s}$ corresponding to all possible differential vectors $\boldsymbol{v}$, and its complexity grows exponentially in N.

It is also worth pointing out that an expression similar to (16) was used as branch metric for ML sequence detection without CSI in [19], [20]. In fact, whereas (16) uses an upper triangular matrix $\boldsymbol{U}$, a lower triangular matrix is devised in [19], [20] in order to meet causality constraints, i.e., to decide on s_i prior to s_j for $i < j$.

3.2 Sphere Decoding Algorithm

The sphere decoder only examines those candidate vectors $\boldsymbol{s}$ that lie inside a sphere of radius R [3]:

$$||\boldsymbol{U}\boldsymbol{s}||^2 \le R^2\,. \quad (17)$$

Due to the upper triangular form of $\boldsymbol{U}$, condition (17) can be checked componentwise, i.e., having found (preliminary) decisions $\hat{s}_l$ for the last $N-i$ components s_l, $i+1 \le l \le N$, we obtain a condition for the ith component s_i, $1 \le i \le N$. To see this, let $u_{i\iota}$ denote the entry of $\boldsymbol{U}$ in row i and column ι, $1 \le i, \iota \le N$, and introduce the squared *length*

$$d_{i+1}^2 = \sum_{l=i+1}^{N}\left|\sum_{\iota=l}^{N} u_{l\iota}\hat{s}_\iota\right|^2 \quad (18)$$

accounting for the last $N-i$ components. Then, possible values s_i have to satisfy the length criterion

$$d_i^2 = \left|u_{ii}s_i + \sum_{\iota=i+1}^{N} u_{i\iota}\hat{s}_\iota\right|^2 + d_{i+1}^2 \le R^2\,. \quad (19)$$

Once a valid vector $\hat{\boldsymbol{s}}$ is found, i.e., $i = 1$ is reached, the radius R is dynamically updated by

$$R := ||\boldsymbol{U}\hat{\boldsymbol{s}}||$$

and sphere decoding is repeated starting with $i = N$ and new radius R. If for some index i condition (19) cannot be met, i is incremented and another value s_i is tested. Sphere decoding is finished if no vector is found inside the sphere with radius R.

Our MSDSD algorithm is summarized by the pseudocode in Figure 2. The organization agrees in principle with the SD algorithm proposed in [3, Section III.B] for the closest point search in lattices. In particular, code lines 10–14, 15–20, and 22–28 correspond to parts "Case A", "Case B", and "Case C" in [3, Section III.B]. In the following, we address the specific details of our MSDSD algorithm.

3.2.1 Phase Ambiguities

Since the MSDD metric is invariant to a phase shift common to all components of $\boldsymbol{s}$ such that the same differential vector $\boldsymbol{v}$ results, without loss of generality we fix $s_N = 1$ (line 2) and start sphere decoding with $i = N-1$ (line 3).

3.2.2 Search Strategy

The ordering of hypothetical symbols s_i examined for the ith component is critical for a low SD complexity, i.e., for early finding the ML estimate $\hat{\boldsymbol{s}}$. As suggested in [3] we apply the Schnorr-Euchner search strategy [21], i.e., the symbols s_i are ordered according to the length d_i starting with the smallest value. In this way, the search can be safely terminated as soon as d_i exceeds the current sphere radius R. The appropriate ordering of s_i is accomplished by the two functions "findBest()" and "findNext()" given in Figure 2. The first function yields the *phase index* m_i ($s_i = \mathrm{e}^{\mathrm{j}2\pi m_i/M}$) of the best candidate point which minimizes d_i in (19), and the second function zig-zags through the remaining $M-1$ signal points such that d_i increases monotonically.

3.2.3 PSK Signal Constellation

In order to perform the Schnorr-Euchner search strategy for PSK constellations in a similar manner as for pulse-amplitude modulation (PAM) signal sets, cf. e.g. [3], [22], the functions "findBest()" and "findNext()" deal with the phase index m_i instead of the signal point s_i itself. Furthermore, by introducing (line F1-3) and incrementing (line F2-3) the counter n_i for the number of signal points already examined for component i, the finite constellation size is taken into account, i.e., the

search is terminated if $n_i = M$ is true (line 26). We note that this counter is a necessity, since in contrast to PAM signal sets no boundary region exists for the phase index $m_i \in \mathbb{Z}$.

3.2.4 Initial Radius

When using the Schnorr-Euchner strategy no initial radius is needed, i.e., $R \to \infty$ can be assumed. The first vector $\hat{\boldsymbol{s}}$ found might then be referred to as Babai point [3], [22]. Clearly, this initialization is appealing, since the likely event of setting the value for R too small, i.e., no point lies in the sphere, is bypassed. For this reason, we chose to initialize our MSDSD algorithm with a reasonably high value of R for the following simulation results.

4 Results and Discussion

In this section, we present simulation results and discuss the performance of MSDSD. In particular, we consider the bit-error rate (BER) and the computational complexity of MSDSD and compare them with those of Mackenthun's algorithm (MA) and DF-DD. As exemplary system parameters, 4DPSK transmission and fading bandwidth $B_f T = 0.03$ are assumed.

4.1 Bit-Error Rate

Figure 3 shows the BER curves for MSDSD, DF-DD, and MA, respectively, with observation window sizes $N = 6$ and $N = 10$ as functions of the signal-to-noise ratio (SNR) $10\log_{10}(\bar{E}_b/N_0)$ ($\bar{E}_b$ and N_0 denote the average received energy per information bit and the two-sided equivalent baseband noise power density, respectively.) As reference curves, BERs for CDD ($N = 2$) and coherent detection with perfect CSI are also plotted.

From Figure 3 we can observe a substantial gain in power efficiency of MSDSD and DF-DD with $N > 2$, respectively, compared to CDD with $N = 2$. MSDSD approaches quite closely the performance of coherent detection and, depending on the target BER, outperforms DF-DD by several dB for the same value of N. Interestingly, the performance of MA deteriorates with larger N due to the increasing mismatch between the blockwise time-invariant channel assumption and the actually time-varying channel coefficients. This, in fact, nicely illustrates the necessity for a true ML detector for the general flat fading case.

4.2 Computational Complexity

The application of SD to MSDD is certainly interesting in itself, but for practical implementation its computational complexity is of paramount importance of course. Therefore, we implemented the different algorithms in floating point C and consider the number of real multiplications required per N symbols as measure for complexity, cf. e.g. [23], [22]. Although neglecting computational costs of memory accesses, loops etc., this measure provides quite meaningful statements and is independent of particular processor architecture or programming skills.

As it was to be expected from other applications of SD, e.g. in multiple-antenna systems [2], [4], [5], [6], [22], MSDSD leads to a tremendous improvement over brute-force ML-MSDD in terms of computational complexity. Depending on N, the number of multiplications is reduced by order of magnitudes. For example, we observed that for MSDSD with $N = 10$ and $10\log_{10}(\bar{E}_b/N_0) \geq 10$ dB the expected number of visited symbols s_i for the ith component, $1 \leq i \leq N-1$, is less than four, whereas M^{N-i}, i.e, on average $\frac{1}{N-1}\sum_{i=1}^{N-1} 4^{N-i} = 38836$, symbols s_i are tested while doing brute-force ML-MSDD.

Figure 4 compares the expected complexity of MSDSD with that of DF-DD and MA, respectively, for $N = 10$ as function of the SNR. For the implementation of the DF-DD and MA we followed closely the description in [12] and the MATLAB program in [24, Section 5], respectively. As can be seen, the complexity of MSDSD decreases rapidly with increasing SNR, since the search quickly terminates for small enough noise. Especially for $10\log_{10}(\bar{E}_b/N_0) \gtrsim 15$ dB, which corresponds to $\mathrm{BER} \lesssim 2 \cdot 10^{-2}$, it is well in the order of or even below the complexity of DF-DD, which is independent of SNR. This emphasizes on the practical relevance of MSDSD, as the performance gain shown in Figure 3 does not entail an increase in complexity or is even accompanied by a complexity reduction. MA requires about a third of the multiplications of DF-DD, but as concluded from the results in Figure 3 is only an appropriate alternative for rather time-invariant fading channels.

To gain more insight into the properties of MSDSD, it is quite illustrative to consider the complexity exponent [23], [22][3]

$$E_C(N) \triangleq \log_{N-1}(\text{average number of multiplications}) .$$

In Figure 5, $E_C(N)$ is plotted as a function of N for different values of $10\log_{10}(\bar{E}_b/N_0)$. The respective curves for DF-DD and MA are also included for comparison. We can observe that the behavior of $E_C(N)$ heavily depends on the particular adjusted SNR. In particular, it appears that for the larger values of $10\log_{10}(\bar{E}_b/N_0)$ the complexity exponent approaches a constant, which in turn implies polynomial expected complexity of MSDSD [23]. From Figure 5 we can also conclude that the performance gain of MSDD over DF-DD comes at practically no cost in complexity considering usual observation window sizes of $N \leq 10 \ldots 20$.

So far, we have considered expected complexity of MSDSD. Of course, in a practical implementation

[3] We use $N-1$ instead of N as basis of the logarithm, since only $N-1$ components of $\boldsymbol{s}$ are searched for.

the maximum complexity will be limited, and we are interested in the implications of this on the performance of MSDSD. Therefore, Figure 6 depicts the achieved BERs for the case that the maximum number of multiplications is limited to a certain value. MSDSD with $N = 10$ and $10 \log_{10}(\bar{E}_b/N_0) = 10$, 20, 30 dB is assumed. For a proper comparison, the BERs achieved and the maximum number of multiplications required for MSDSD without any complexity limitation are also plotted. As can be seen, the maximum number of multiplications can safely be limited to about $2 \cdot 10^3$ multiplications per N symbol block without sacrificing performance in this example. Especially for low SNRs this implies a tremendous reduction in worst-case complexity. For high SNRs we have more generally observed that setting a reasonable complexity limit does practically not affect performance, since the probability of exceeding this limit is far below the achievable BER without any limit. Hence, MSDSD is an attractive solution also in this regard.

5 Conclusions

In this paper, we have proposed the application of SD to accomplish MSDD for Rayleigh fading channels. Following the footsteps of the SD literature we derived an efficient MSDSD algorithm, whose expected complexity is orders of magnitudes below that of brute-force search typically used in MSDD. The presented simulation results verify the excellent performance versus complexity trade-off of the proposed MSDSD. It was shown that the gains in power efficiency compared to other low-complexity multiple-symbol detectors come almost for free. Furthermore, imposing reasonable limitations on the maximum MSDSD complexity did practically not affect performance.

Acknowledgement

The authors wish to thank the anonymous reviewer whose valuable comments led to an improvement of the MSDSD algorithm.

References

[1] E. Viterbo and J. Boutros. A Universal Lattice Code Decoder for Fading Channels. *IEEE Trans. Inf. Theory*, 45:1639–1642, July 1999.

[2] O. Damen, A. Chkeif, and J.-C. Belfiore. Lattice Code Decoder for Space-Time Codes. *IEEE Com. Letters*, 4(5):161–163, May 2000.

[3] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger. Closest Point Search in Lattices. *IEEE Trans. Inf. Theory*, 48(8):2201–2214, August 2002.

[4] H. Vikalo and B. Hassibi. Modified Fincke-Pohst Algorithm for Low-Complexity Iterative Decoding over Multiple-Antenna Channels. In *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Lausanne, July 2002.

[5] B. Hassibi and B.M. Hochwald. Cayley Differential Unitary Space-Time Codes. *IEEE Trans. Inf. Theory*, 48(6):1485–1503, June 2002.

[6] B.M. Hochwald and S. ten Brink. Achieving Near-Capacity on a Multiple-Antenna Channel. *IEEE Trans. Com.*, 51(3):389–399, March 2003.

[7] S.G. Wilson, J. Freebersyser, and C. Marshall. Multi-Symbol Detection of M-DPSK. In *Proc. IEEE Global Telecom. Conf. (GLOBECOM)*, pages 47.3.1–47.3.6, Dallas, November 1989.

[8] D. Divsalar and M.K. Simon. Multiple-Symbol Differential Detection of MPSK. *IEEE Trans. Com.*, 38(3):300–308, March 1990.

[9] K.M. Mackenthun, Jr. A Fast Algorithm for Multiple-Symbol Differential Detection of MPSK. *IEEE Trans. Com.*, 42(2/3/4):1471–1474, February/March/April 1994.

[10] P.Y. Kam and C.H. Teh. Reception of PSK Signals Over Fading Channels Via Quadrature Amplitude Estimation. *IEEE Trans. Com.*, 31:1024–1027, August 1983.

[11] A. Svensson. Coherent Detection Based on Linear Prediction and Decision Feedback for QDPSK. *Electronics Letters*, 30(20):1642–1643, 1994.

[12] R. Schober, W.H. Gerstacker, and J.B. Huber. Decision-Feedback Differential Detection of MDPSK for Flat Rayleigh Fading Channels. *IEEE Trans. Com.*, 47(7):1025–1035, July 1999.

[13] W. Xiaofu and S. Songgeng. Low complexity multisymbol differential detection of MDPSK over flat correlated Rayleigh fading channels. *Electronics Letters*, 34(21):2008–2009, October 1998.

[14] P. Tarasak and V.K. Bhargava. Reduced complexity multiple symbol differential detection of space-time block code . In *Proc. IEEE Wireless. Commun. and Networking Conf. (WCNC)*, New Orleans, March 2002.

[15] B. Li. A new reduced-complexity algorithm for multiple-symbol differential detection. *IEEE Com. Letters*, 7(6):269–271, June 2003.

[16] J.G. Proakis. *Digital Communications*. McGraw-Hill, New York, third edition, 1995.

[17] P. Ho and D. Fung. Error Performance of Multiple-Symbol Differential Detection of PSK Signals Transmitted over Correlated Rayleigh Fading Channels. *IEEE Trans. Com.*, 40:25–29, October 1992.

[18] H. Vikalo and B. Hassibi. On Sphere Decoding Algortihm. I. Expected Complexity, II. Generalizations, Second-order Statistics, and Applications to Communications. submitted to *IEEE Trans. Signal Proc., preprint available at: http://www.its.caltech.edu/ hvikalo/publications.html*, 2002.

[19] D. Makrakis, P.T. Mathiopoulos, and D.P. Bouras. Optimal Decoding of Coded PSK and QAM Signals in Correlated Fast Fading Channels and AWGN: A Combined Envelope, Multiple Differential and Coherent Detection Approach. *IEEE Trans. Com.*, 42(1):63–74, January 1994.

[20] G.M. Vitetta and D.P. Taylor. Maximum Likelihood Decoding of Uncoded and Coded PSK Signal Sequences Transmitted over Rayleigh Flat-Fading Channels. *IEEE Trans. Com.*, 43(11):2750–2758, November 1995.

[21] C.P. Schnorr and H.H. Hörner. Lattice Basis Reduction: Improved Practical Algorithms and Solving Subset Sum Problems. *Math. Programming*, 66:181–191, 1994.

[22] O. Damen, H. El Gamal, and G. Caire. On Maximum Likelihood Detection and the Search for the Closest Lattice Point. *IEEE Trans. Inf. Theory*, 49(10):2389–2402, October 2003.

[23] B. Hassibi and H. Vikalo. On the expected complexity of integer least-squares problems. In *Proc. Acoust., Speech, and Signal Proc. (ICASSP)*, pages 1497 –1500, 2002.

[24] W. Sweldens. Fast Block Noncoherent Decoding. *IEEE Com. Letters*, 5(4):132–134, April 2001.

function MSDSD$(\boldsymbol{U}, M, N, R)$
input: $N \times N$ upper triangular matrix $\boldsymbol{U}$, constellation size M, window length (dimension) N, initial radius R
output: Maximum-likelihood decision $\hat{\boldsymbol{s}}$

---

1 $d_N := |u_{NN}|$ // initialize length
2 $s_N := 1$ // fix last component of $\boldsymbol{s}$
3 $i := N - 1$ // start with component $i = N - 1$
4 $k_i := u_{(N-1)N}$ // sum of last $N - i$ components, see (18)
5 $[m_i, \mathrm{step}_i, n_i] = \text{findBest}\,(k_i, u_{ii}, M)$ // find best candidate point for ith component
6 ⟨**loop**⟩
7 $\quad d_i^2 := \left| k_i + u_{ii} \cdot \mathrm{e}^{\mathrm{j}\,\frac{2\pi}{M} m_i} \right|^2 + d_{i+1}^2$ // update squared length, see (19)
8 $\quad$ **if** $d_i < R$ **and** $n_i \leq M$ { // check sphere radius (19) and constellation size
9 $\quad\quad s_i := \mathrm{e}^{\mathrm{j}\,\frac{2\pi}{M} m_i}$ // store candidate component
10 $\quad\quad$ **if** $i \neq 1$ { // component 1 not reached yet
11 $\quad\quad\quad i := i - 1$ // move down
12 $\quad\quad\quad k_i := \sum_{\iota=i+1}^{N} u_{i\iota} s_\iota$ // add last $N - i$ components, see (18)
13 $\quad\quad\quad [m_i, \mathrm{step}_i, n_i] := \text{findBest}\,(k_i, u_{ii}, M)$ // find best candidate point for ith component
14 $\quad\quad$ }
15 $\quad\quad$ **else** { // first component reached
16 $\quad\quad\quad \hat{\boldsymbol{s}} := \boldsymbol{s}$ // best point so far
17 $\quad\quad\quad R := d_i$ // update sphere radius
18 $\quad\quad\quad i := i + 1$ // move up
19 $\quad\quad\quad [m_i, \mathrm{step}_i, n_i] := \text{findNext}\,(m_i, \mathrm{step}_i, n_i)$ // next point examined for component i
20 $\quad\quad$ }
21 $\quad$ }
22 $\quad$ **else** {
23 $\quad\quad$ **do** {
24 $\quad\quad\quad$ **if** $i == N - 1$ **return** $\hat{\boldsymbol{s}}$ **and exit** // outside sphere and no component left
25 $\quad\quad\quad i := i + 1$ // move up
26 $\quad\quad$ } **while** $n_i == M$ // while all constellation points examined
27 $\quad\quad [m_i, \mathrm{step}_i, n_i] := \text{findNext}\,(m_i, \mathrm{step}_i, n_i)$ // next point examined for component i
28 $\quad$ }
29 **goto** ⟨**loop**⟩

---

subfunction $[m_i, \mathrm{step}_i, n_i] = \text{findBest}\,(k_i, u_{ii}, M)$ // Finds MPSK signal point that minimizes (18)

F1-1 $p := \frac{M}{2\pi} \left(\mathrm{angle} \left(-\frac{k_i}{u_{ii}} \right) \right)$ // unconstrained phase index ($p \in \mathbb{R}$)
F1-2 $m_i := \lfloor p \rceil$ // constrained phase index ($m_i \in \mathbb{Z}$)
F1-3 $n_i := 1$ // counter of examined candidates for component i
F1-4 $\mathrm{step}_i := \mathrm{sign}(p - m_i)$ // step size for phase index

---

subfunction $[m_i, \mathrm{step}_i, n_i] = \text{findNext}\,(m_i, \mathrm{step}_i, n_i)$ // Selects next MPSK signal point
// according to Schnorr-Euchner strategy

F2-1 $m_i := m_i + \mathrm{step}_i$ // zig-zag through MPSK constellation
F2-2 $\mathrm{step}_i = -\mathrm{step}_i - \mathrm{sign}(\mathrm{step}_i)$ // update step size
F2-3 $n_i := n_i + 1$ // count examined candidates for component i

---

Fig. 2. Pseudocode for MSDSD algorithm. $\lfloor x \rceil$ denotes the closest integer to $x \in \mathbb{R}$.

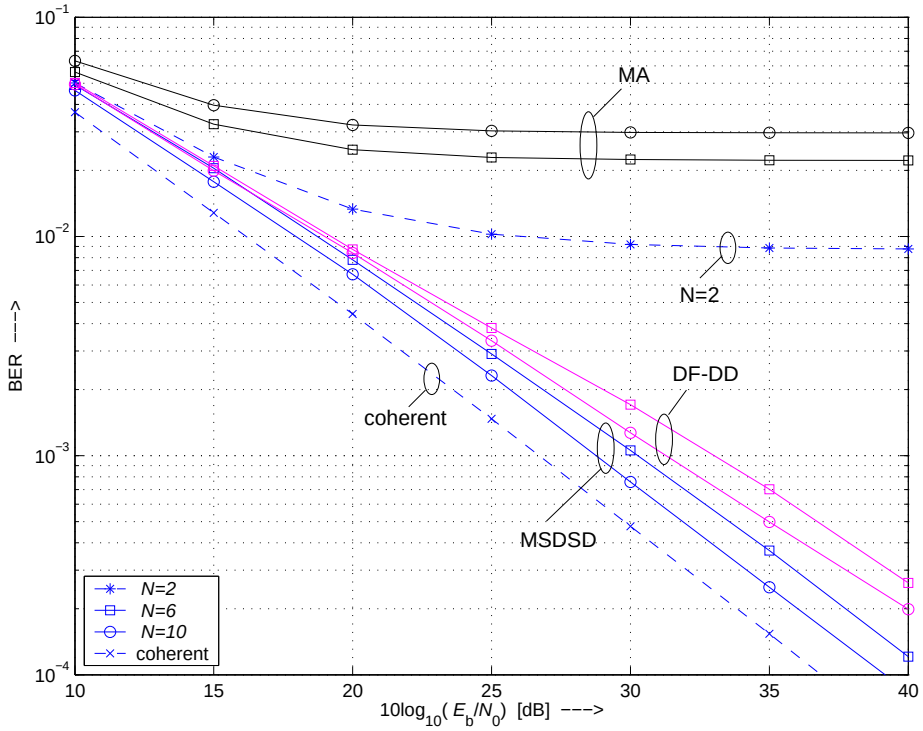

Fig. 3. BER vs. $10\log_{10}(\bar{E}_b/N_0)$ for 4DPSK and Rayleigh fading with $B_fT = 0.03$. Comparison of MSDSD, DF-DD, and MA. As reference: CDD ($N = 2$) and coherent detection with CSI.

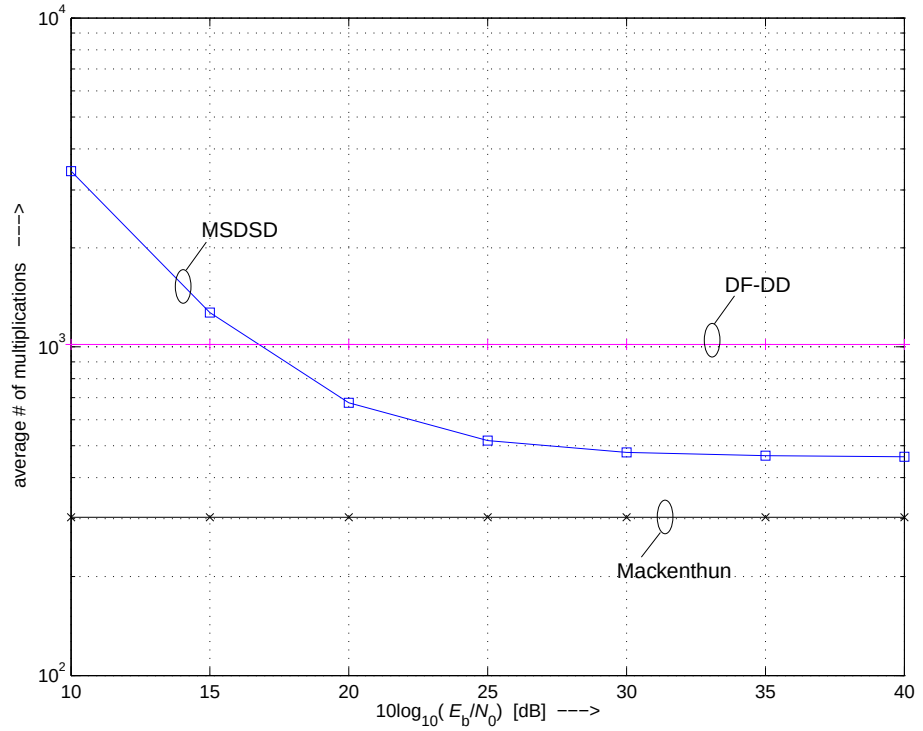

Fig. 4. Average number of multiplications vs. $10\log_{10}(\bar{E}_b/N_0)$ for 4DPSK and Rayleigh fading with $B_fT = 0.03$. Comparison of MSDSD, DF-DD, and MA. Observation window size $N = 10$.

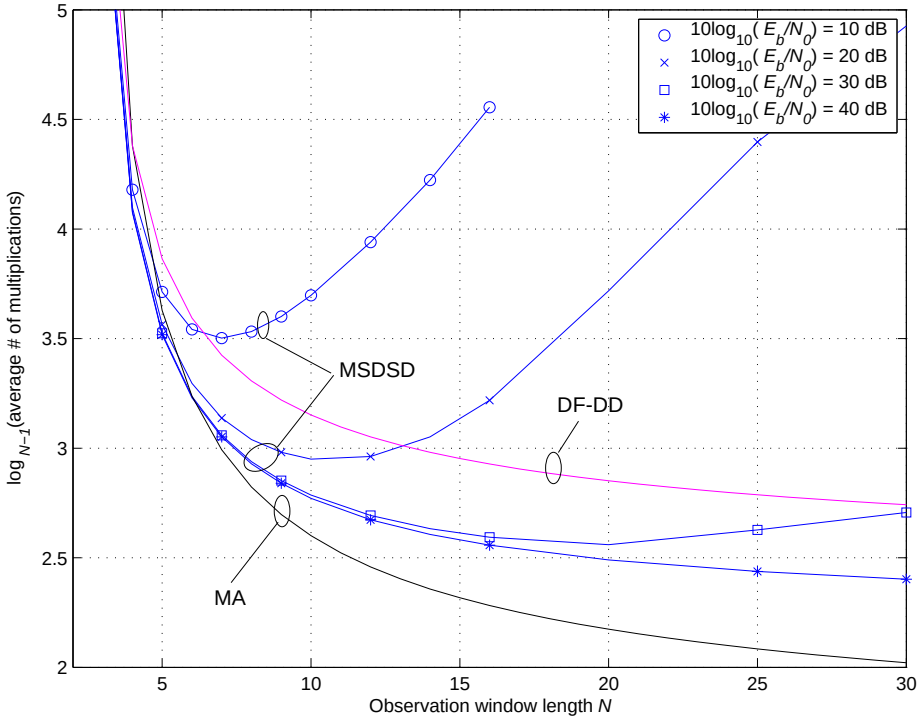

Fig. 5. Complexity exponent vs. observation window size N. 4DPSK and Rayleigh fading with $B_f T = 0.03$.

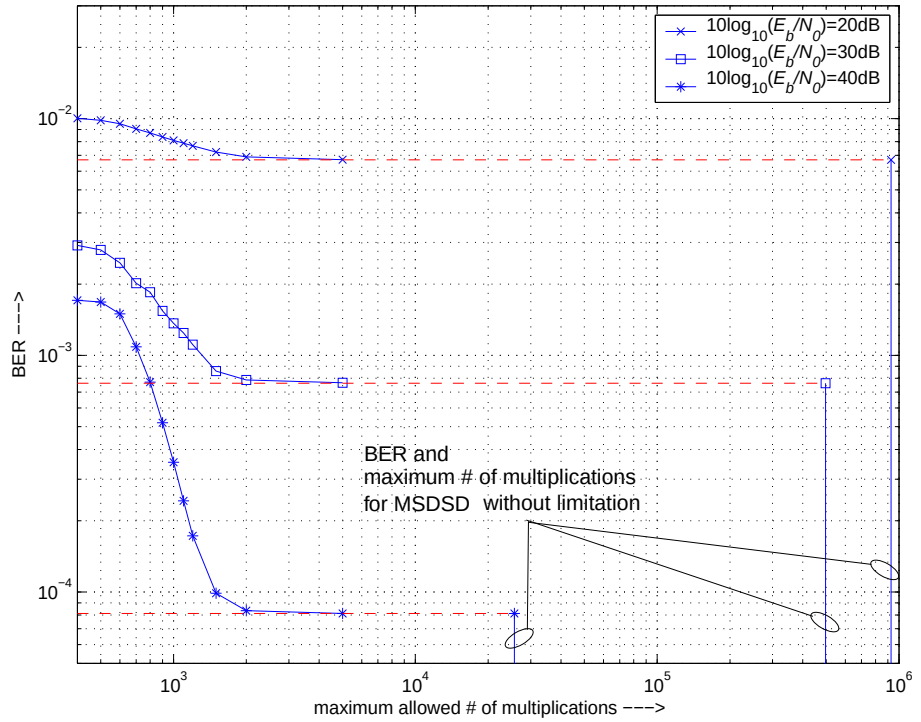

Fig. 6. BER vs. maximum allowed number of multiplications for MSDSD with $N = 10$. 4DPSK and Rayleigh fading with $B_f T = 0.03$.