ABSTRACT — Two powerful techniques for improving the performance of a detector in a multiple-input multiple-output communications channel are decision feedback and lattice reduction. We propose a new detector, the DOLLAR detector, that combines decision feedback and lattice reduction in a novel way. The DOLLAR detector is based on a simple lattice reduction algorithm that is designed to exploit the power of decision feedback. The combination of the new lattice reduction algorithm and decision feedback leads to near-optimal performance with complexity comparable to the BLAST-ordered decision-feedback (BODF) detector. For example, over a 4-input 4-output Rayleigh-fading channel with 16-QAM inputs, the DOLLAR detector outperforms the BODF detector by 6 dB while increasing complexity by 16%.

1. INTRODUCTION

The promise of dramatic increases in spectral efficiency has led to widespread interest in multiple-input multiple-output (MIMO) wireless communications systems [1]. A practical obstacle to realizing this promise is the high complexity of MIMO detection. The BLAST-ordered decision-feedback (BODF) detector [2] is a suboptimal detection strategy that is easy to implement, but its performance falls far short of the optimal maximum-likelihood (ML) detector. Numerous detectors have been proposed to close this gap [3]–[10].

Lattice-aided detection has recently emerged as a powerful tool for approaching ML performance with reduced complexity [3]–[6]. These detectors exploit the lattice structure of the channel output for common input alphabets like QAM. First, they use a lattice-reduction algorithm (such as the LLL algorithm [11][6]) to find an effective channel matrix whose columns are more orthogonal than the original channel matrix. Second, they apply a linear or decision-feedback detector to the effective channel. Since the effective channel matrix can be nearly orthogonal, these low-complexity detectors can achieve near-ML performance [3]–[6].

The viability of lattice-aided detection is limited in practice by the high complexity of lattice-reduction algorithms such as the LLL algorithm. Particularly on wireless channels that vary rapidly with time, the high overhead of lattice reduction can negate much of the computational savings.

The lattice-aided BODF detector of [5] achieves near-ML performance by combining LLL lattice reduction with MMSE BODF detection. This paper proposes a similar detector that is based on a simple observation: LLL lattice reduction is overkill when accompanied by BODF detection. The presence of the BODF detector relaxes considerably the lattice reduction constraints. We propose a lattice reduction algorithm that is much simpler than the LLL algorithm, that exploits the presence of the BODF detector.

The proposed detector is called the DOLLAR detector, which is short for double-sorted low-complexity lattice-reduced decision-feedback detector. The low-complexity lattice reduction used by the DOLLAR detector results in an effective channel that is less orthogonal than the effective channel created by LLL lattice reduction. The DOLLAR detector relies on decision feedback to overcome the noise enhancement left behind by its low-complexity lattice reduction. The DOLLAR detector significantly outperforms the BODF detector with negligible complexity increase.

The V-BLAST or spatial multiplexing architecture is a landmark MIMO system that achieves high spectral efficiency with low complexity. Replacing the BODF detector by the DOLLAR detector in such a system will lead to dramatic gains in SNR, with no penalty in spectral efficiency and with only slightly greater implementation complexity.

The remainder of the paper is organized as follows. In Section 2 we give some background on lattices. In Section 3 we describe a general framework for lattice-aided DF detection that includes the BODF detector as a special case. In Section 4, we describe the proposed DOLLAR detector along with a computationally efficient implementation. In Section 5 we quantify the performance and complexity of the DOLLAR detector, and in Section 6 we make concluding remarks.

2. BACKGROUND AND CHANNEL MODEL

A complex integer is defined as a complex number whose real and imaginary parts are both integers. A complex lattice is defined as the set of all linear
combinations of a set of linearly independent basis vectors \( \{b_1, \ldots, b_N\} \) with complex integer coefficients. The lattice dimension is defined as the number \( (N) \) of basis vectors. In terms of the matrix \( B = [b_1, \ldots, b_N] \), the lattice points can be written as \( Bx \) where \( x \) is a vector of complex integers.

The basis for a lattice is not unique. If \( B \) is a basis, the product \( BT \) will also be a basis whenever \( T \) is an \( N \times N \) unimodular matrix; i.e., whenever \( T \) has complex integer entries and a determinant of either \( \pm 1 \) or \( \pm j \). Note that the inverse of a unimodular matrix is also unimodular. Trivial examples of unimodular matrices include the identity matrix and permutation matrices. Lattice reduction is a technique for finding the unimodular \( T \) matrix that transforms one basis into another, usually with the goal of making the new basis as orthogonal as possible.

This paper considers a memoryless channel with \( N \) inputs \( a = [a_1, \ldots, a_N]^T \) and \( M \) outputs \( r = [r_1, \ldots, r_M]^T \):

\[
r = Ha + w,
\]

where \( H = [h_1, \ldots, h_N] \) is a complex \( M \times N \) channel matrix, and where \( w = [w_1, \ldots, w_M]^T \) is noise. We assume that the columns of \( H \) are linearly independent, which implies \( M \geq N \). We assume that the noise components are i.i.d. complex Gaussian random variables with \( E[ww^*] = c^2I \), where \( w^* \) denotes the conjugate transpose of \( w \). Further, we assume that the inputs are uncorrelated and chosen from the same QAM alphabet \( A = \{\pm c, \pm 3c, \ldots \pm (\sqrt{q-1})c \} + \sqrt{-1}(\pm c, \pm 3c, \ldots \pm (\sqrt{q-1})c) \), where \( q = |A| \). The constant \( c \) is chosen so that the alphabet has unit energy, i.e. \( c = \sqrt{1.5/(q-1)} \).

The QAM alphabet is a scaled and translated subset of the complex integer lattice. Therefore, the noise-less channel output \( Ha \) is a scaled and translated version of a point in the \( N \) dimensional complex lattice generated by the columns of \( H \). Recovering \( a \) can thus be seen as a closest point lattice search.

3. LATTICE-AIDED DF DETECTION

In this section we describe a family of MIMO detectors called lattice-aided DF (LA-DF) detectors. The family is parameterized by the choice of a unimodular lattice reduction matrix \( T \). A variety of well-known detectors can be viewed as special cases of the LA-DF detectors, as summarized in Table 1. For example, the BODF detector is a special case when \( T \) is chosen as the BLAST ordering permutation matrix. The LLL-BODF detector of [5] chooses \( T \) according to the LLL algorithm using the equivalent real model of the channel [16]. The DOLLAR detector proposed in this paper is also a special case, when \( T \) is chosen according to the strategy described in the next section (see (14)).

In describing the LA-DF detector family we will adopt the noise-predictive view of the BODF detector [13]. In a first stage, the receiver applies a forward filter to the channel output to remove (zero-forcing) or reduce (MMSE) interference. A by-product of this filter is that it correlates the noise. In a second stage, the receiver exploits the correlation in the noise using linear prediction to reduce the noise variance. A block diagram of the LA-DF detector is shown in Fig. 1. In the following we define the forward and prediction filters \( C \) and \( P \), respectively.

\[\text{A. Forward Filter}\]

The MMSE and zero-forcing (ZF) versions of the LA-DF detector can be derived simultaneously from an extended version of the channel matrix \([14][15]\):

\[\tilde{H} = H\tilde{T},\]

where the parameter \( \alpha \) is set to zero for the ZF detector, and it is set to \( \sigma \) for the MMSE detector. A basis for the lattice spanned by the columns of \( \tilde{H} \) is:

\[
\tilde{H} = QL,
\]

Roughly speaking, we now proceed as if we were implementing the DF detector of [13] when the actual channel matrix is \( H \).

Define \( \hat{Q} \) and \( L \) by the QR decomposition of \( \tilde{H} \):

\[
\tilde{H} = \hat{Q}L,
\]

where the columns of the \((M+N) \times N\) matrix \( \hat{Q} \) are orthonormal, and where \( L \) is a lower triangular \( N \times N \) matrix with positive and real diagonal elements. The forward filter is then:

Table 1: Special cases of the lattice-aided DF detector.

<table>
<thead>
<tr>
<th>Detector</th>
<th>Reduction Matrix ( T )</th>
<th>Channel Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>DF [13]</td>
<td>Identity</td>
<td>Complex</td>
</tr>
<tr>
<td>SDF [17]</td>
<td>Sorted-QR Permutation</td>
<td>Complex</td>
</tr>
<tr>
<td>BODF [1]</td>
<td>BLAST Permutation</td>
<td>Complex</td>
</tr>
<tr>
<td>LLL-BODF [5]</td>
<td>LLL Lattice Reduction and</td>
<td>Real</td>
</tr>
<tr>
<td>DOLLAR</td>
<td>Using (14)</td>
<td>Complex</td>
</tr>
</tbody>
</table>

Fig. 1. Lattice reduced decision-feedback detector.
C = L^{-1}Q^* ,
\tag{5}

where the matrix Q is the first M rows of the matrix \( \hat{Q} \).

As shown in Fig. 1, the output of the forward filter is \( y = Cr \), which reduces to:
\[ y = T^{-1}a - \alpha^2L^{-1}(L^{-1})^*a + L^{-1}Q^*w = x + n , \]
\tag{6}

where \( x = T^{-1}a \), where we used the fact that \( Q^*HT = L - \alpha^2(L^{-1})^*T \) in the simplification, and where the effective noise \( n \) contains both noise and residual interference when \( \alpha \neq 0 \).

The lattice-aided detector first makes a decision \( \hat{x} \) about \( x \), and then transforms this decision into a decision about \( a \). However, the elements of \( x \) belong to an alphabet that changes with the channel. As a result, the receiver cannot use a conventional symbol slicer. Instead, lattice-aided detectors exploit the fact that \( x / 2c - s \) is a vector of complex integers when \( s = T^{-1}[1, \ldots, 1]^T(1 + \sqrt{-1})/2 \) using a slicer that incorporates scaling and shifting. This new slicer is defined as:
\[ \text{slicer}(y, s_k) = \left[ \frac{y}{2c} - s_k \right] + s_k2c, \]
\tag{7}

where \( [y] = [\text{Re}(y)] + \sqrt{-1}[\text{Im}(y)] \) independently rounds the real and imaginary parts of \( y \) to the nearest integer, and where \( s_k \) is the \( k \)-th element of \( s \).

**B. Noise-Predictive Decision-Feedback**

A lattice-aided linear detector would use the symbol slicer (7) to quantize the components of \( y \) directly. In contrast, the LA-DF detector exploits the correlation in the noise \( n \) using linear prediction to reduce its variance. In particular, after making decisions about the first \( k - 1 \) components of \( x \), it subtracts these decisions from the corresponding components of \( y \), yielding \( \{n_1, \ldots, n_{k-1}\} \) when the decisions are correct. These noise samples are then used to predict the \( k \)-th noise sample, and this prediction is subtracted before making the next decision. The noise-prediction process can be summarized recursively as:
\[ \hat{x}_k = \text{slicer}(y_k - \sum_{j < k} p_{k,j}(y_j - \hat{x}_j), s_k), \]
\tag{8}

where \( p_{k,j} \) is a linear prediction coefficient. This prediction process is captured in Fig. 1 by the strictly lower triangular prediction matrix \( P \), which contains \( p_{k,j} \) at row \( k \) and column \( j \). The prediction coefficients that minimize the mean-square prediction error under an assumption of correct decisions are \( p_{k,j} = -l_{k,j}/l_{k,k} [19] \).

The final step of the detector is to convert a decision about \( x = T^{-1}a \) into a decision about \( a \). The conversion is simple when \( \hat{x} \) is correct: we need merely multiply by \( T \). However, errors in \( \hat{x} \) may cause \( T\hat{x} \) to fall outside the alphabet \( A \). To deal with this possibility we can append a conventional symbol slicer, yielding:
\[ \tilde{a} = \text{dec}(T\hat{x}) , \]
\tag{9}

where \( \text{dec}(x) \) returns the element of \( A \) nearest \( x \).

**4. A NEW LATTICE-REDUCED DF DETECTOR**

The DOLLAR detector is a special case of the LA-DF detector of the previous section. Therefore, to fully specify the DOLLAR detector we need only specify its unimodular lattice-reduction matrix \( T \). That is the purpose of this section. Specifically, the DOLLAR lattice reduction consists of three steps: an initial sorting, a weak-Gramm-Schmidt (WGS) reduction, and a final sorting.

The DOLLAR detector begins in the same manner as the BODF detector by computing the QR decomposition of the extended channel matrix after sorting its columns:
\[ \overline{H} = QG , \]
\tag{10}

where \( \Pi_1 \) is a permutation matrix. To reduce complexity, the DOLLAR detector implements both the sorting and the QR decomposition simultaneously using the sorted-QR decomposition of [17]. The resulting permutation \( \Pi_1 \) is often but not always identical to the BLAST ordering.

Stopping here, with \( T = \Pi_1 \), would yield the suboptimally ordered DF detector of [17]. Its performance is limited by the weakest post-detection SNR, \( \min_j (g_{j,k}^2) \). To overcome this bottleneck, the next step is a WGS reduction [6] of \( G \), which yields:
\[ L' = GM , \]
\tag{11}

where \( M \) is a special unimodular matrix that is lower triangular with ones on the diagonal, so that \( L' \) is also a lower triangular matrix. Fig. 3 gives the pseudocode for the WGS reduction algorithm, which takes \( G \) as an input and produces \( M \) as an output.

**Function WGS-Reduction.**

\begin{tabular}{l}
Input: \( G \) \\
Output: \( M, M^{-1} \)
\end{tabular}

\begin{tabular}{l}
1 \( M = \mathbf{I}_{N \times N} \); \( M^{-1} = \mathbf{I}_{N \times N} \) \\
2 \( \text{for } j = N - 1 \text{ downto } 1 \), \\
3 \( \text{for } i = j + 1 \text{ to } N \) \\
4 \( M_{i,j} = g_{i,j} / g_{j,j} \) \\
5 \( \text{for } n = 1 \text{ to } N \) \\
6 \( m_{n,j} = m_{n,j} - M_{i,j}m_{n,i} \) \\
7 \( \text{end} \) \\
8 \( \text{end} \) \\
9 \( \text{end} \)
\end{tabular}

Fig. 2. Pseudocode for the weak-Gramm-Schmidt reduction.
Combining (10) and (11) we see that, in effect, the WGS reduction leads to a new QR decomposition:

\[
\mathbf{H} \Pi_2 \mathbf{M} = \mathbf{Q} \mathbf{L}.
\]

If we were to stop here, with \( \mathbf{T} = \Pi_1 \mathbf{M} \), the resulting LA-DF detector would again be equivalent to the suboptimally ordered detector of [17] because the diagonals of \( \mathbf{L} \) and \( \mathbf{G} \) are the same.

The final step of the DOLLAR algorithm is to sort the columns of \( \mathbf{H} \Pi_2 \mathbf{M} \). To achieve the best performance, the DOLLAR detector computes the BLAST ordering of the matrix \( \mathbf{H} \Pi_2 \mathbf{M} \). Once again, this last step can be viewed as a new QR decomposition:

\[
(\mathbf{H} \Pi_2 \mathbf{M}) \Pi_2 = \tilde{\mathbf{Q}} \mathbf{L},
\]

where \( \Pi_2 \) is the BLAST permutation matrix from this second column sort. This is obviously a special case of the general QR decomposition (4), when we adopt the following as the lattice-reduction matrix:

\[
\mathbf{T} = \Pi_2 \mathbf{M} \Pi_2.
\]

This equation defines the DOLLAR lattice reduction, and thus the DOLLAR detector.

While the determinants of \( \mathbf{L} \) and \( \mathbf{G} \) are identical, the minimum diagonal of \( \mathbf{L} \) is larger than that of \( \mathbf{G} \) whenever \( \mathbf{M} \neq \mathbf{I} \), thus improving the performance bottleneck.

A. Performance Analysis

To understand how the DOLLAR detector outperforms the BODF detector, consider a 2 × 2 example. The BODF detector performs poorly when either \( g_{2,1}^2 \) or \( g_{2,2}^2 \) is small since \( \min(g_{2,2}^2, g_{1,1}^2) \) dictates performance.

The DOLLAR detector improves the bottleneck of the BODF detector. To see this we first observe that \( \min(l_{2,1}^2, l_{2,2}^2) \) dictates the performance of the DOLLAR detector since it makes an error only when \( \hat{x} \neq x \). The diagonals of the matrix \( \mathbf{L} \) in (13) can be written as:

\[
l_{1,1}^2 = \max\left(g_{1,1}^2, -\frac{g_{1,1}^2 g_{2,2}^2}{g_{2,1}^2 + |e|^2 g_{2,2}^2}\right),
\]

\[
l_{2,2}^2 = \min(g_{2,2}^2, g_{1,1}^2 + |e|^2 g_{2,2}^2),
\]

where \( e = g_{2,1}/g_{2,2} - [g_{2,1}/g_{2,2}] \) is the rounding error, and \( |e|^2 \leq 1/2 \). Using these expressions we will show that \( \min(l_{2,1}^2, l_{2,2}^2) \geq \min(g_{2,1}^2, g_{2,2}^2) \) by considering the matrix \( \mathbf{L} \) for the cases when \( \Pi_2 = \mathbf{I} \) and is not the identity matrix.

First, if the second sort does not change the order, then \( \Pi_2 = \mathbf{I} \), and \( l_{2,j}^2 = l_{j,1}^2 \). In this case, the DOLLAR and BODF detectors have the exact same performance and achieve full diversity. From (15), \( g_{2,1}^2 > g_{2,2}^2 (1 - |e|^2) \) which implies that \( g_{2,1}^2 > g_{2,2}^2 / 2 \), and therefore \( Pr(\hat{x} \neq x) \) decays as \( \text{SNR}^{-2} \) [18].

Second, if the post sort changes the order, then we know that \( g_{2,1}^2 + |e|^2 g_{2,2}^2 < g_{2,2}^2 \) which implies that \( g_{2,1}^2 < g_{2,2}^2 / 2 \). In this case, the ratio of the minimum SNR of the DOLLAR and BODF detectors, \( \gamma \), can have one of two values:

\[
\gamma = \begin{cases} 
\frac{g_{2,2}^2}{g_{2,1}^2 + |e|^2 g_{2,2}^2}, & \text{if } l_{2,1}^2 < l_{2,2}^2 \\
g_{2,1}^2, & \text{else}
\end{cases}
\]

Obviously, \( \gamma \geq 1 \), which means that the DOLLAR detector always performs at least as good as the BODF detector. Furthermore, \( \gamma \) can be large, in which case the DOLLAR detector significantly outperforms the BODF detector.

We can quantify the relative performance of the two detectors as \( g_{2,1}^2 \) approaches zero and \( |e|^2 > 0 \), which can occur when \( \mathbf{h}_1 \) and \( \mathbf{h}_2 \) are nearly colinear. In this case, \( l_{2,1}^2 < l_{2,2}^2 \) since \( l_{2,1}^2 \) also tends to zero, and \( \gamma \) approaches the following limit:

\[
\lim_{g_{2,1}^2 \to 0} \gamma = \frac{1}{|e|^2}.
\]
lines (3)-(4) are omitted and \( M = I \). Therefore, the only additional complexity required by the DOLLAR detector during preprocessing comes from the WGS reduction. At most, the algorithm shown in Fig. 2 requires \((N^3 - N)/2\) real multiplies when the lower half of \( M^{-1} \) contains only nonzero complex numbers. Multiplying the two matrices \( M^{-1} \) and \( G^{-1} \) in line (4) of Fig. 3 can also require up to \((N^3 - N)/2\) real multiplies. This additional preprocessing complexity is relatively small since the preprocessing of the BODF detector requires approximately \( 9N^3 \) real multiplies when \( M = N \). During core processing the only additional multiplies come from mapping \( x \) to the QAM alphabet (9), which requires at most \( 3(N^2 - N)/2 \) real integer multiplies. The rest of the core processing is identical to that of the BODF detector which requires \( 9(N^2 - N)/2 \) real multiplies when \( M = N \). In this paper we consider integer and floating point multiplies to be equally complex.

5. NUMERICAL RESULTS

In this section, we present numerical results that demonstrate that the proposed DOLLAR detector achieves a very attractive performance-complexity trade-off over Rayleigh-fading channels. In all simulations we assume that the receiver knows the channel parameters \( H \) and \( \sigma^2 \). We compare the MMSE version of the proposed DOLLAR detector (\( \alpha = \sigma \)) to the MMSE versions of the LLL-linear [5], LLL-BODF [5], and sphere detectors [7].

Fig. 4 shows the bit-error-rate (BER) versus SNR = \( E[|H|a|^2]/(E[|w|^2] \log_2 q) \), averaged over \( 10^6 \) realizations of the channel model (1) with \( 2 \times 2 \) Rayleigh-fading channels and 4-QAM inputs. The DOLLAR detector achieves almost exactly ML performance, requiring about 3.5 dB less signal power than the BODF detector to reach BER = \( 10^{-3} \). This performance is even more impressive considering the fact that the DOLLAR and BODF detectors have nearly the same complexity. Specifically, to achieve the performance shown in Fig. 4 the DOLLAR detector requires at most 23.5 real multiplies per bit, while the BODF detector requires at most 21.25 real multiplies per bit when the channel is assumed to change every symbol period. In contrast, the sphere detector required 64.5 real multiplies per bit to achieve ML performance.

The DOLLAR detector maintains an attractive performance-complexity trade-off for larger channels as well. Fig. 5 shows the SNR required to reach BER \( 10^{-3} \) versus the number of real computations per bit required by each detector as averaged over \( 10^6 \) \( 4 \times 4 \) Rayleigh-fading channels with 16-QAM inputs. Since systems are often designed for the worst-case scenario, we measure the 99.9% quantile of complexity.

**Function DOLLAR.**

Input: \( H, r, \alpha, \mathcal{A} \)  
Output: \( \hat{a} \)

1. \([Q_1, G, \Pi_1] = \text{sortedQR}(H\
2. \text{Compute } G^{-1}
3. \([M, M^{-1}] = \text{WGS-Reduction}(G)
4. L'^{-1} = M^{-1}G^{-1}
5. \left[\Pi, \Pi_2\right] = \text{BLASTsort}(L'^{-1}*k)
6. Q = \text{first } M \text{ rows of } Q_1
7. T = \Pi_1M\Pi_2, \quad T^{-1} = \Pi_2^\dagger M^{-1}\Pi_1^\dagger
8. C = \Pi_2^\dagger (L')^{-1}Q^*
9. y = Cr
10. for \( k = 1 \) to \( N \),
11. \( \hat{x}_k = \text{slicer}(y_k - \sum_{j < k} p_{kj}(y_j - \hat{x}_j), s_k) \)
12. end
13. \( \hat{a} = \text{dec}(\hat{T}\hat{x}) \)

Fig. 3. Pseudocode for the DOLLAR detector. See references for sortedQR [17] and BLASTsort [19] functions.

The complexity depends upon how often the channel changes. Fig. 5 shows the complexity intervals for each detector as well as when \( L = 16 \). The complexity intervals demonstrate how \( L \) impacts complexity. While the DOLLAR detector is less complex than the LLL-linear and LLL-BODF detectors for any value of \( L \), its complexity reduction is greater the faster the channel changes. When the channel changes every symbol period (\( L = 1 \)), the LLL-linear, LLL-BODF, MMSE sphere, and ML detectors require 281, 300, 160, and 290 real multiplies per bit, respectively, while the DOLLAR detector requires only 39 real multiplies per bit.

In Fig. 5 the black diamond markers illustrate the performance-complexity trade-off when \( L = 16 \). In this case, the DOLLAR detector outperforms the BODF detector by \( 6 \) dB while increasing complexity by \( 16\% \). The DOLLAR detector outperforms the LLL-linear detector by \( 1.1 \) dB and requires \( 68\% \) less complexity. At the same time, the LLL-BODF detector outperforms the DOLLAR detector by \( 0.3 \) dB, but requires \( 72\% \) more complexity. The MMSE sphere detector further improves performance, but increases complexity even more. Not shown is the ML detector which required \( 182 \) real multiplies per bit and \( 16.1 \) dB of SNR.

6. CONCLUSION

We have proposed a new kind of lattice-aided DF detector called the DOLLAR detector. It is based on a new lattice reduction technique that sandwiches a WGS procedure between two sorting procedures. This new detector was shown to achieve near-ML performance...
with small complexity increase relative to the BODF detector. For example, over a 4-input 4-output Rayleigh-fading channel with 16-QAM inputs, the DOLLAR detector outperformed the BODF detector by 6 dB while requiring 16% more complexity. In the same setting, the LLL-BODF detector needed as much as 0.3 dB better. In another experiment, the DOLLAR detector achieved within a fraction of a dB of ML -QAM inputs, while requiring only 10% more complexity than the BODF detector.

![Fig. 4.](image1.jpg)

**Fig. 4.** Performance comparison for $2 \times 2$ Rayleigh-fading channels with 4-QAM inputs.

![Fig. 5.](image2.jpg)

**Fig. 5.** Performance-complexity trade-off for $4 \times 4$ Rayleigh-fading channels with 16-QAM inputs. Intervals of complexity are shown for $1 \leq L \leq \infty$, while the markers indicate the trade-off for $L = 16$.

### References


