The Chase Family of Detection Algorithms for Multiple-Input Multiple-Output Channels

Deric W. Waters and John R. Barry

School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, GA 30332-0250, USA
{deric, barry}@ece.gatech.edu

***Abstract* — We introduce a new family of detectors for multiple-input multiple-output (MIMO) channels called *Chase detectors* because of their resemblance to the well-known Chase algorithm for soft decoding of error-control codes. A Chase detector is fully specified by only three simple parameters; nevertheless, it reduces to a wide range of previously reported MIMO detectors as special cases, including the maximum-likelihood and decision-feedback detectors. Based on the Chase framework we propose two new detectors, the B-Chase and L-Chase detectors, both of which perform well on fading channels. In fact, the L-Chase detector is shown to outperform the BLAST-ordered decision-feedback detector by 9.8 dB, while simultaneously requiring 17% *fewer* computations, on a 4-input 4-output Rayleigh-fading channel with uncoded 4-QAM inputs. Under the same conditions, the B-Chase detector falls only 0.2 dB short of the minimum-mean-squared-error sphere detector, while requiring 50% fewer computations.**

I. Introduction

The promise of high spectral efficiency and diversity to fading has led to widespread interest in multiple-input multiple-output (MIMO) communications. A practical obstacle to the realization of a MIMO system is the complexity of detection. For example, the complexity of the maximum-likelihood (ML) detector [1,2] grows exponentially with the number of channel inputs. A popular reduced-complexity alternative is the BLAST ordered decision-feedback (BDF) detector [3] (also known as the successive interference canceller); its performance can approach that of the ML detector when there are many more channel outputs than inputs [3], but otherwise the BDF detector is significantly inferior to the ML detector.

The large gap in both performance and complexity between the ML and BDF detectors has motivated the search for alternatives. Various combinations of the ML and BDF detectors have been proposed that improve on the performance of the BDF detector at the cost of increased complexity [4,5]. Other algorithms that achieve near-ML performance while reducing the average complexity are the minimum mean-squared-error (MMSE) sphere detector of [6], and the MMSE BDF detector combined with lattice reduction of [7].

This paper introduces a new family of *Chase detectors*, which includes as special cases the ML [2], BDF [3], ML-BDF [4], parallel [5], and partial decision-feedback (PDF) [8] detectors. The Chase family provides a unified framework for comparing a variety of existing detectors. Furthermore, we propose the L-Chase and B-Chase detectors as new special cases that perform well on fading channels. We will demonstrate that the L-Chase detector greatly outperforms the BDF detector, despite its reduced complexity. We will also show that the B-Chase detector can approach ML performance with less complexity than previously reported detectors.

In Section II we introduce the Chase framework for defining detection algorithms, and show how existing detectors fit into the framework. In Section III we derive two new instances of the Chase detector family. In Section IV we present some performance and complexity results, and in Section V we make concluding remarks.

II. Chase Detection: A General Framework

This paper considers a memoryless channel with N inputs $\boldsymbol{a} = [a_1, \dots a_N]^T$ and M outputs $\boldsymbol{r} = [r_1, \dots r_M]^T$:

$$\boldsymbol{r} = \mathbf{H}\boldsymbol{a} + \boldsymbol{w}\,, \tag{1}$$

where $\mathbf{H} = [\boldsymbol{h}_1, \dots \boldsymbol{h}_N]$ is a complex $M \times N$ channel matrix whose i-th column is $\boldsymbol{h}_i$, and where $\boldsymbol{w} = [w_1, \dots w_M]^T$ is noise. We assume that the columns of $\mathbf{H}$ are linearly independent, which implies $M \ge N$. We assume that the noise components are i.i.d. complex Gaussian random variables with $\mathrm{E}[\boldsymbol{w}\boldsymbol{w}^*] = \sigma^2\mathbf{I}$, where $\boldsymbol{w}^*$ denotes the conjugate transpose of $\boldsymbol{w}$. Further, we assume that the inputs are uncorrelated and chosen from the same unit-energy alphabet A, so that $\mathrm{E}[\boldsymbol{a}\boldsymbol{a}^*] = \mathbf{I}$.

In this section we introduce the *Chase detector*, a general detection strategy for MIMO channels that reduces to a variety of previously reported detectors as special cases. The Chase detector defines a simple framework for not only comparing existing MIMO detection algorithms but also proposing new ones. Specifically, a Chase detector is defined by five steps, as illustrated in Fig. 1, and as outlined below:

This research was supported in part by National Science Foundation grants CCR-0082329 and CCR-0121565.

Step 1. Identify $i \in \{1, \ldots N\}$, the index of the first symbol to be detected.

Step 2. Generate a sorted list $\{s_1, \ldots s_q\}$ of candidate values for the i-th symbol, by computing the q elements of the alphabet nearest to y_i, where $\boldsymbol{y} = (\mathbf{H}^*\mathbf{H})^{-1}\mathbf{H}^*\boldsymbol{r}$.

Step 3. Determine a set of q residual vectors $\{\boldsymbol{r}_1, \ldots \boldsymbol{r}_q\}$ by cancelling the contribution to $\boldsymbol{r}$ from the i-th symbol, assuming each candidate from the list is in turn correct:

$$\boldsymbol{r}_j = \boldsymbol{r} - \boldsymbol{h}_i s_j . \tag{2}$$

Step 4. Apply each of $\{\boldsymbol{r}_1, \ldots \boldsymbol{r}_q\}$ to its own independent subdetector, which makes decisions about the remaining $N-1$ symbols (all but the i-th symbol). Together with s_j, the j-th subdetector defines a *candidate* hard decision $\hat{\boldsymbol{a}}_j$ regarding the input $\boldsymbol{a}$.

Step 5. Choose as the final hard decision $\hat{\boldsymbol{a}}$ the candidate hard decision $\{\hat{\boldsymbol{a}}_1, \ldots \hat{\boldsymbol{a}}_q\}$ that best represents the observation $\boldsymbol{r}$ in a minimum mean-squared-error sense:

$$\hat{\boldsymbol{a}} = \underset{\{\hat{\boldsymbol{a}}_1, \ldots \hat{\boldsymbol{a}}_q\}}{\operatorname{argmin}} \| \boldsymbol{r} - \mathbf{H}\hat{\boldsymbol{a}}_j \|^2 . \tag{3}$$

The Chase detector can be viewed as a generalization of its namesake, the well-known Chase *algorithm* for soft decoding of binary error-control codes [9]. The Chase algorithm begins by identifying the p least reliable bits of a received codeword, and enumerates all 2^p corresponding binary vectors while fixing the remaining more reliable bits. This is analogous to Steps 1 and 2, except in Step 1, only one symbol is identified instead of p, and in Step 2, only a subset of the most likely values are enumerated. The Chase algorithm decodes each of the 2^p binary vectors using a simple hard-decoding algorithm, producing candidate hard decisions for the codeword. This is analogous to the cancellation and subdetection in Steps 3 and 4. Finally, the Chase algorithm chooses the candidate codeword that best matches the received observations in a way precisely analogous to that in Step 5.

To uniquely define an instance of the Chase detector requires that the following three parameters be identified:

- A strategy for selecting i in Step 1.
- A list length q for Step 2.
- A subdetector algorithm for Step 4.

Table 1 summarizes how the ML, BDF, PDF, ML-BDF, and parallel detectors may be specified as Chase detectors using these three parameters. The last two rows of Table 1 describe new detectors that will be proposed in the next section.

Table 1: Special cases of the Chase detector.

Detector	First-Symbol Index i	List Length q	Subdetector
ML [2]	any	$\|A\|$	ML
BDF [3]	♦BLAST$_1$	1	BDF
PDF [8]	♦BLAST$_1$	1	Linear
Parallel [5]	using (9) with $\alpha_q = \infty$	$\|A\|$	any
B-Chase	according to (6)	$1 < q \le \|A\|$	BDF
L-Chase	according to (6)	$1 < q \le \|A\|$	Linear

♦ The index BLAST$_1$ signifies the first index of the BLAST ordering [3].

III. Two New Chase Detectors

In this section we introduce the B-Chase and L-Chase detectors, as summarized in the last two rows of Table 1. The two detectors are distinguished by the type of subdetector they employ; the B-Chase detector uses BDF [3,10] subdetectors, whereas the L-Chase detector uses linear [1] subdetectors. These new detectors are further parameterized by the strategy for selecting i in Step 1, as well as the list length q, which could be any integer in the set $\{1, \ldots |A|\}$.

In the following subsection we analyze the performance benefit of using a list detector. In the subsection that follows we propose strategies for selecting the index i in Step 1. Finally, in the last subsection we propose computationally efficient implementations of the L-Chase and B-Chase detectors.

A. *The SNR Gain of a List Detector*

We define α_q^2 as the *effective gain* in SNR provided by a list detector of length q, relative to a list detector of length one (i.e., relative to a *conventional* decision device). We will introduce this concept using the unit-energy 4-QAM alphabet $\{e^{\pm j\pi/4}, e^{\pm j3\pi/4}\}$ as an example. By symmetry we can assume that $e^{j\pi/4}$ was transmitted, without loss of generality. In this case, a conventional decision device ($q = 1$) will be correct for any observation $y = |y|e^{j\phi}$ in the first quadrant of the complex plane, satisfying $|\phi - \pi/4| < \pi/4$. The distance from the transmitted symbol to the nearest decision boundary is thus $d_1 = 1/\sqrt{2}$. In contrast, a list detector with $q = 2$ will be *correct* (produce a list that contains the transmitted symbol) for any observation $y = |y|e^{j\phi}$ in the half-plane $|\phi - \pi/4| < \pi/2$. The

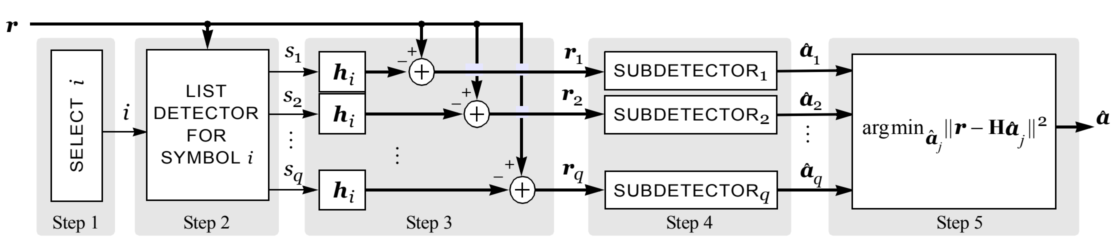

Fig. 1. Block diagram of the Chase detector.

distance d_2 to the nearest decision boundary is thus $d_2 = 1$. Therefore, relative to a conventional decision device, the list detector with $q = 2$ provides an SNR gain of $\alpha_2^2 = (d_2/d_1)^2 = 2$.

Generalizing the above example to arbitrary alphabets A and arbitrary list lengths q, we define d_q as the minimum distance from any valid symbol in A to its decision boundary for a length-q list detector. In these terms, the effective SNR gain of a length-q list detector is:

$$\alpha_q^2 = (d_q/d_1)^2. \tag{4}$$

The parameter α_q is a function of the symbol alphabet A as well as the list length q. Obviously, $\alpha_1 = 1$ and $\alpha_{|A|} = \infty$, but it is also easy to calculate α_q for other values of q. Specifically, for 4-QAM we have $\alpha_2^2 = \alpha_3^2 = 2$; for 16-QAM we have $\alpha_2^2 = 2$, $\alpha_4^2 = 4$, $\alpha_8^2 = 8$, and $\alpha_{12}^2 = 13.6$; while for 64-QAM we have $\alpha_4^2 = 4$, $\alpha_8^2 = 8$, $\alpha_{16}^2 = 18.5$, $\alpha_{32}^2 = 38.4$, and $\alpha_{48}^2 = 57.8$.

B. *Selection Algorithms: Specifying the index* i

The performance of a Chase detector depends strongly on the *selection algorithm* that is adopted in Step 1 for selecting the index i of the first symbol to be detected. In this subsection we propose a selection algorithm that is preferable to that proposed in [5] in two ways; it is drastically less complex while achieving nearly the same performance, and it applies to the case when $q < |A|$. We also describe better-performing selection algorithms that are too complex to implement in practice but are nevertheless useful as performance benchmarks.

Proposed Selection Algorithm. The list in Step 2 of the Chase detector is simply the list of valid alphabet symbols closest to:

$$\begin{aligned} y_i &= \mathbf{c}_i \boldsymbol{r} \\ &= a_i + n_i, \end{aligned} \tag{5}$$

where $\mathbf{c}_i$ is the i-th row of the pseudoinverse $\mathbf{C} = (\mathbf{H}^*\mathbf{H})^{-1}\mathbf{H}^*$, and where the noise variance is $\mathrm{E}[|n_i|^2] = \sigma^2\|\mathbf{c}_i\|^2$.

We now argue that the choice of i must balance two opposing goals: (1) that the transmitted symbol be on the list, and (2) that the subsequent subdetectors perform well. Indeed, if our only concern were to ensure that the transmitted symbol be on the list, we would choose i so that variance of the noise n_i is small, or equivalently, so that the pseudoinverse row norm $\|\mathbf{c}_i\|$ is small. On the other hand, if our only concern were to ensure that the subdetectors perform well when making decisions about the remaining $N - 1$ symbols, we would choose i so that the effective MIMO channel seen by the subdetectors is as "orthogonal" as possible. Specifically, we would choose i so that $\boldsymbol{h}_i$ is the column of $\mathbf{H}$ that is *least orthogonal* to the other columns, which is precisely the i that corresponds to the pseudoinverse row with maximum norm $\|\mathbf{c}_i\|$.

Therefore, to balance the two opposing goals, we should choose i so that the noise variance $\sigma^2\|\mathbf{c}_i\|^2$ is large, but not so large that the list does not contain the transmitted symbol. We want the actual symbol to be on the list, but just barely. In other words, we should choose i so that the effective SNR of the list detector is neither too small nor too large.

We propose *choosing* i *so that the effective SNR seen by the list detector is as close as possible to a target value of* β^{-2}. Since the effective SNR with list detection is $\alpha_q^2 / \sigma^2\|\mathbf{c}_i\|^2$, assuming a unit-energy alphabet, this reduces to:

$$i = \underset{k \in \{1, 2, \ldots, N\}}{\arg\min} \left| \|\mathbf{c}_k\| - \frac{\beta\alpha_q}{\sigma} \right|. \tag{6}$$

Tuning the factor β provides an additional degree of freedom in balancing the opposing goals.

The complexity of the selection algorithm (6) is very low. In fact, many subdetectors (like the BDF) already require the norms of the rows of $\mathbf{C}$, in which case (6) may be implemented with no additional computations.

Benchmark Selection Algorithms. We now describe benchmark selection algorithms that perform better than (6), but whose high complexity makes them impractical. They depend on $\mathbf{H}^{(k)} = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_{k-1}, \boldsymbol{h}_{k+1}, \ldots, \boldsymbol{h}_N]$, the channel matrix after the k-th column has been removed. They will be described in terms of the following pair of Cholesky decompositions:

$$(\mathbf{H}\Pi)^*(\mathbf{H}\Pi) = \mathbf{G}^*\mathbf{G}, \tag{7}$$

$$(\mathbf{H}^{(k)}\Pi^{(k)})^*(\mathbf{H}^{(k)}\Pi^{(k)}) = (\mathbf{G}^{(k)})^*\mathbf{G}^{(k)}, \tag{8}$$

where Π and $\Pi^{(k)}$ are permutation matrices that represents the symbol ordering and the BLAST ordering [3] of the submatrix $\mathbf{H}^{(k)}$, respectively, and where $\mathbf{G}$ and $\mathbf{G}^{(k)}$ are lower triangular matrices with real positive diagonal elements $\{g_{1,1}, \ldots g_{N,N}\}$ and $\{g_{1,1}^{(k)}, \ldots g_{N,N}^{(k)}\}$, respectively.

The B-Chase detector can be viewed as a bank of BDF detectors in parallel, with the first-stage decision device replaced by a list detector. Since the effective SNR of the list detector is a factor α_q^2 larger than the SNR of the first stage of a BDF detector, a straightforward generalization of the BLAST ordering would choose i according to:

$$i = \underset{k \in \{1, 2, \ldots, N\}}{\arg\max} \min\{\alpha_q g_{1,1}, g_{1,1}^{(k)}, \ldots, g_{N-1,N-1}^{(k)}\}. \tag{9}$$

In other words, the chosen index should maximize the worst-case SNR, with the understanding that the list detector amplifies the SNR of the first stage by a factor of α_q^2. The parallel selection algorithm [5] is a special case of (9) when $q = |A|$ and $\alpha_q = \infty$. The first index chosen by the BLAST ordering is also a special case of (9) when $q = 1$ and $\alpha_q = 1$. Implementing (9) when $\alpha_q \neq 1$ requires calculating the BLAST ordering for each of the N possible submatrices $\mathbf{H}^{(k)}$, which requires $O(MN^3)$ computations [10].

We now develop a selection algorithm for the L-Chase detector, which uses linear subdetectors. If j were detected first in Step 1, the post-detection SNR of the k-th symbol in the linear subdetector would be inversely proportional to $\|\boldsymbol{c}_k^{(j)}\|^2$, where $\boldsymbol{c}_k^{(j)}$ is the k-th row of the pseudoinverse of $\mathbf{H}^{(j)}$. Therefore, the selection algorithm that maximizes the minimum post-detection SNR with linear subdetection, where the SNR of the first symbol is again scaled by α_q^2, is:

$$i = \underset{j \in \{1, 2, \dots, N\}}{\arg\min} \max\{\alpha_q^{-2}\|\boldsymbol{c}_j\|^2, \|\boldsymbol{c}_1^{(j)}\|^2 \dots, \|\boldsymbol{c}_{N-1}^{(j)}\|^2\}. \quad (10)$$

The total complexity of (10) is $O(MN^2)$, since $\|\boldsymbol{c}_k^{(j)}\|^2 = \|\boldsymbol{c}_k\|^2 - |p_{k,j}|^2\|\boldsymbol{c}_j\|^2$, where $p_{k,j} = \boldsymbol{c}_k\boldsymbol{c}_j{}^*/\|\boldsymbol{c}_j\|^2$.

C. Implementing the Subdetectors

In this subsection we describe computationally efficient implementations of the two new Chase detectors, beginning with the B-Chase detector. We propose to implement the B-Chase detector using the modified decorrelation decision-feedback algorithm of [10], in a further modified form that forces the i-th symbol to be detected first. This algorithm calculates $\mathbf{G}$, $\mathbf{G}^{-1}$, and Π, where the final $N-1$ columns of Π implement the BLAST ordering of $\mathbf{H}^{(i)}$. The receiver thus begins by computing $\boldsymbol{z} = (\mathbf{G}^{-1})^*(\mathbf{H}\Pi)^*\boldsymbol{r}$, which reduces to:

$$\boldsymbol{z} = \mathbf{G}\tilde{\boldsymbol{a}} + \boldsymbol{v}, \quad (11)$$

where $\tilde{\boldsymbol{a}} = \Pi^*\boldsymbol{a}$ is a permuted version of the transmitted symbol vector. The l-th subdetector then finds its decision vector $\hat{\boldsymbol{a}}_l$ using classical DF, except that the first decision is hard-wired to s_l. The index of the best decision vector is found according to:

$$\hat{\boldsymbol{a}}_f = \arg\min_{\hat{\boldsymbol{a}}_l}\|\boldsymbol{z} - \mathbf{G}\hat{\boldsymbol{a}}_l\|^2. \quad (12)$$

The final decision vector is $\hat{\boldsymbol{a}} = \Pi\hat{\boldsymbol{a}}_f$. Implementing (12) requires minimal complexity because the l-th subdetector has already calculated most of the elements of $\mathbf{G}\hat{\boldsymbol{a}}_l$.

We now describe an efficient implementation of the L-Chase detector, which is basically a set of q PDF detectors in parallel. We propose using the low-complexity noise-predictive implementation of the PDF detector presented in [8], except that the index of the first symbol detected is forced to be i. In this case, the decision in Step 5 reduces to:

$$\hat{\boldsymbol{a}}_f = \arg\min_{\hat{\boldsymbol{a}}_l}\|\mathbf{G}(\tilde{\mathbf{C}}\boldsymbol{r} - \hat{\boldsymbol{a}}_l)\|^2, \quad (13)$$

where $\tilde{\mathbf{C}}$ is the pseudoinverse of the channel matrix $\mathbf{H}$ with the first and i-th rows swapped. The final decision is obtained by swapping the first and i-th elements of $\hat{\boldsymbol{a}}_f$.

The average complexity of the B-Chase and L-Chase algorithms can be significantly reduced by exploiting the lower triangular structure of $\mathbf{G}$ in a manner reminiscent of sphere detection [6]. Specifically, the cost in (12) or (13) need not be calculated for each candidate decision. Instead, a cost *threshold* can be established with the cost of the first subdetector's decision. The cost calculation of subsequent subdetectors can be aborted whenever this threshold is exceeded. Furthermore, the threshold can be reduced each time a lower cost is found.

IV. Numerical Results

This section examines the performance and complexity of Chase detectors on Rayleigh-fading channels, assuming the channel $\mathbf{H}$ is known to the receiver. Fig. 2 shows bit-error rate versus $SNR = E[\|\mathbf{H}\boldsymbol{a}\|^2]/E[\|\boldsymbol{w}\|^2]$, averaged over 10^6 realizations of the channel model (1) with 4×4 Rayleigh-fading channels and 16-QAM inputs. The solid curves represent the performance of the L-Chase and B-Chase detectors for various list sizes, using (6) as the selection algorithm. The uppermost curve shows the performance of the traditional BDF detector, a special case of the B-Chase detector with $q = 1$ and $\beta = 0$. Even with a list length of only $q = 2$, the L-Chase detector outperforms the BDF detector by 2 dB at a BER of 10^{-3}. Larger list lengths lead to enormous performance gains. Specifically, when $q = 16$ the B-Chase detector falls only 0.1 dB short of the ML detector, while the L-Chase detector falls 1.1 dB short of the ML detector.

The dashed lines in Fig. 2 show the performance of the Chase detectors when the benchmark selection algorithms of (9) and (10) are used in place of (6). The advantage of (9) and (10) is seen to range from 0.2 dB to 0.8 dB at 10^{-3} BER, depending on the list size and subdetector type. These gains are minimal and not enough to justify the enormous increase in complexity required to implement (9) and (10).

The full benefit of the B-Chase and L-Chase detectors is best understood in the context of a performance-complexity trade off. Fig. 3 shows a plot of performance versus complexity, where performance is measured by the SNR required to reach BER $= 10^{-3}$, and complexity is measured by the 99% quantile of real computations, i.e., the number of real computations exceeded 1% of the time. These results were averaged over 10^5 realizations of the channel model with either 4-QAM or 64-QAM inputs, as indicated. The number of real computations was counted during simulations of the algorithms described in this paper, assuming that a complex multiply requires eight (floating-point) operations, a complex division or addition require two operations each, a complex magnitude requires three operations, and a square root requires one operation.

The L-Chase and B-Chase detectors operate in different regimes of the performance-complexity plane. Specifically, the L-Chase detector is appropriate when *complexity* is at a premium, where the BDF detector would normally be used. In many cases the L-Chase detector is less complex than the BDF detector while performing better. For example, with 4-QAM inputs, the L-Chase detector with $q = 4$ outperforms the BDF detector by 9.8 dB while requiring 17% less complexity. Similarly, with 64-QAM inputs the L-Chase detector with $q = 4$ outperforms the BDF detector by 3.9 dB while requiring 10% less complexity.

In contrast to L-Chase, the B-Chase detector is appropriate when *performance* is at a premium, when a near-ML detector like the MMSE sphere detector would normally be used. With 4-QAM inputs, the B-Chase detector with $q = 4$ falls only 0.2 dB short of the MMSE sphere detector performance, while requiring 50% less complexity. Likewise, with 64-QAM, the B-Chase detector with $q = 32$ falls short of the MMSE sphere detector by only 0.4 dB while reducing complexity by 29%.

The results of Fig. 3 also reveal the strong impact of the alphabet size on the relative merits of the different detectors. Specifically, when the alphabet is small (4-QAM), the L-Chase detector exhibits a preferable performance-complexity trade-off for all list lengths. Furthermore, a maximum list length of $q = 4$ provides enormous performance gains at very little cost in complexity. In contrast, when the alphabet is large (64-QAM), the L-Chase detector is obviously preferable only for relatively short list lengths of $q \leq 8$. After this diminishing returns set in, and for larger list lengths the B-Chase detector becomes preferable. The B-Chase detector is seen to range from the BDF detector to an approximation of the MMSE sphere detector as the list length ranges from its minimum to maximum value.

V. Conclusion

The Chase family of detection algorithms for MIMO channels is a combination of a list detector and a parallel bank of subdetectors. The general Chase detector reduces to a variety of existing MIMO detectors as special cases. Based on the Chase framework, we proposed the B-Chase and L-Chase detectors, two new members of the Chase family. Using efficient implementations and a new selection algorithm, these new detectors demonstrate an attractive performance-complexity trade off. The L-Chase detector is especially attractive because it can simultaneously improve the performance and reduce the complexity of the BDF detector. For example, on a 4×4 Rayleigh-fading channel with 4-QAM uncoded inputs, the L-Chase detector outperforms the BDF detector by 9.8 dB, while simultaneously requiring 17% fewer computations. On the other hand, the B-Chase detector is attractive when near-ML performance is required. For example, on a 4×4 Rayleigh-fading channel with 4-QAM uncoded inputs, the B-Chase detector performs only 0.2 dB worse than the MMSE sphere detector, but is half as complex.

References

[1] S. Verdú, *Multiuser Detection*, Cambridge University Press, 1998.

[2] A. Chan and I. Lee, "A new reduced-complexity sphere decoder for multiple antenna systems," *IEEE Conf. on Commun.*, pp. 460-464, 2002.

[3] G. J. Foschini, G. Golden, R. Valenzuela, and P. Wolniansky, "Simplified processing for wireless communication at high spectral efficiency," *IEEE J. Select. Areas Commun.*, vol. 17, no. 11, pp. 1841-1852, 1999.

[4] W. J. Choi, R. Negi, and J. Cioffi, "Combined ML and DFE decoding for the V-BLAST system," *IEEE Conf. on Commun.*, pp. 1243-1248, 2000.

[5] Y. Li, and Z. Luo, "Parallel detection for V-BLAST system," *Proc. IEEE Conf. on Commun. (ICC)*, vol. 1, pp. 340-344, May 2002.

[6] M. O. Damen, H. E. Gamal, and G. Caire, "On maximum-likelihood detection and the search for the closest lattice point," *IEEE Trans. on Info. Th.*, vol. 49, no. 10, Oct. 2003.

[7] D. Wübben, R. Böhnke, V. Kühn, and K. Kammeyer, "Near-maximum-likelihood detection of MIMO systems using MMSE-based lattice-reduction," *IEEE Conf. on Commun.*, vol. 2, pp. 798-802, June. 2004.

[8] D. W. Waters and J. R. Barry, "Partial decision-feedback detection for multiple-input multiple-output channels," *Proc. ICC*, pp. 2668-72, 2004.

[9] D. Chase, "A class of algorithms for decoding block codes with channel measurement information," *IEEE Trans. Info. Th.*, pp. 170-182, Jan. 1972.

[10] W. Zha and S. Blostein, "Modified decorrelating decision-feedback detection of BLAST space-time system," *IEEE Conf. on Commun.*, vol. 1, pp. 335-339, May 2002.

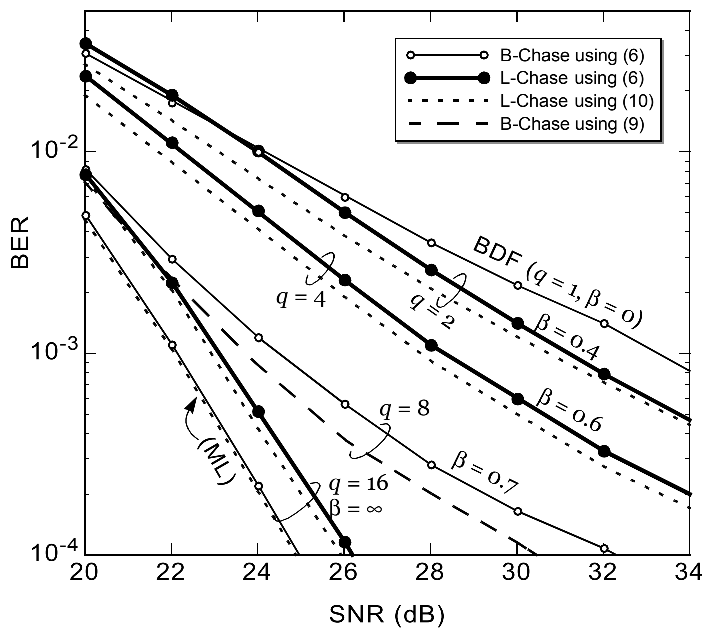

Fig. 2. Performance of the L-Chase and B-Chase detectors with varying list lengths q, over 4×4 channels with 16-QAM inputs.

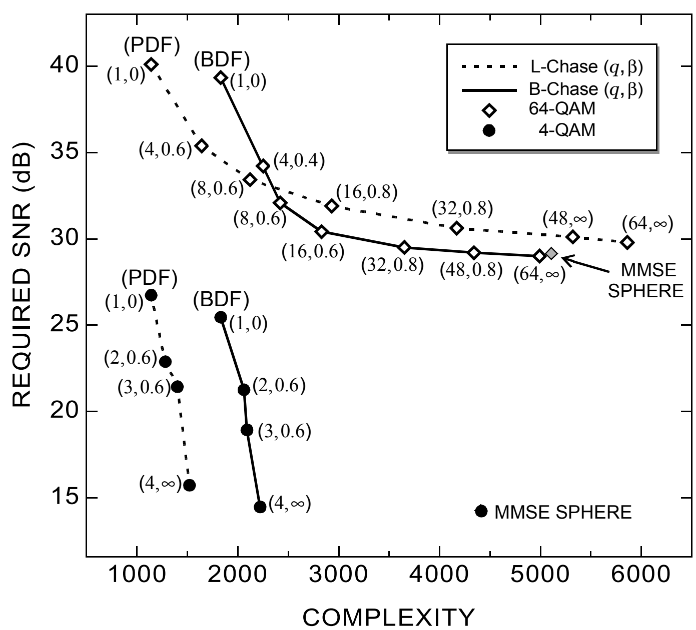

Fig. 3. The performance-complexity trade-off of Chase detectors.