Fixed-Complexity Soft MIMO Detection via Partial Marginalization

Erik G. Larsson and Joakim Jaldén

Erik G. Larsson and Joakim Jaldén, Fixed-Complexity Soft MIMO Detection via Partial Marginalization, 2008, IEEE Transactions on Signal Processing, (56), 8(1), 3397-3407.
http://dx.doi.org/10.1109/TSP.2008.925260

Fixed-Complexity Soft MIMO Detection via Partial Marginalization

Erik G. Larsson and Joakim Jaldén

***Abstract*—This paper presents a new approach to soft demodulation for MIMO channels. The proposed method is an approximation to the exact a posteriori probability-per-bit computer. The main idea is to marginalize the posterior density for the received data exactly over the subset of the transmitted bits that are received with the lower signal-to-noise-ratio (SNR), and marginalize this density approximately over the remaining bits. Unlike the exact demodulator, whose complexity is huge due to the need for enumerating all possible combinations of transmitted constellation points, the proposed method has very low complexity. The algorithm has a fully parallel structure, suitable for implementation in parallel hardware. Additionally, its complexity is fixed, which makes it suitable for pipelined implementation. We also show how the method can be extended to the situation when the receiver has only partial channel state information, and how it can be modified to take soft-input into account. Numerical examples illustrate its performance on slowly fading 4×4 and 6×6 complex MIMO channels.**

***Index Terms*—Detection, MIMO, soft information.**

I. Introduction

We consider the problem of separating coded information symbols that have undergone transmission over a channel that introduces crosstalk. This problem is encountered in many applications. Of particular interest are multiple-antenna (MIMO) systems [1], where the signals received by a receiving antenna array are superpositions of waveforms sent by a transmitter array, and where the gains and phases in the superposition are determined by the radio propagation environment. After appropriate filtering and sampling this leads to the well-known data model $\boldsymbol{y} = \boldsymbol{H}\boldsymbol{s} + \boldsymbol{e}$ (to be made precise in Section II-A) where $\boldsymbol{y}$ is received data, $\boldsymbol{H}$ is a channel matrix, $\boldsymbol{s}$ is a symbol vector with elements from a finite constellation, and $\boldsymbol{e}$ is noise. The problem is then to detect $\boldsymbol{s}$ from $\boldsymbol{y}$. Essentially the same problem occurs in multiuser detection for CDMA [2] and for single-carrier transmission over channels that induce intersymbol interference. In these cases, the matrix $\boldsymbol{H}$ usually has a specific structure. We are interested in the fundamental aspects of the model $\boldsymbol{y} = \boldsymbol{H}\boldsymbol{s} + \boldsymbol{e}$ and we shall refer to it in general terms as a MIMO model.

The problem of detecting $\boldsymbol{s}$ from $\boldsymbol{y}$ has stimulated a large body of research [2]. One can easily show that if the noise $\boldsymbol{e}$ is Gaussian then obtaining the maximum-likelihood solution is equivalent to minimizing the Euclidean distance $\|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{s}\|^2$ with respect to $\boldsymbol{s}$ over the finite set spanned by all possible combinations of constellation points that can constitute the vector $\boldsymbol{s}$. Unfortunately this problem is NP-hard for general $\boldsymbol{H}$ and $\boldsymbol{y}$ [3] which implies that there are no known efficient (i.e. polynomial-time) solutions. Naive solutions, like neglecting the integer constraint and then projecting the so-obtained solution onto the finite set of permissible $\boldsymbol{s}$ (this is called zero-forcing [ZF]), in general works poorly except if $\boldsymbol{H}$ is well conditioned. One can do somewhat better by using ZF decision-feedback-equalization (ZF-DFE) detection ("nulling-and-cancelling"), whereby the elements of $\boldsymbol{s}$ are detected one by one, and in an order that can be optimally chosen [1]. Many other more sophisticated methods, that find the ML solution with high probability, exist. Unfortunately these methods are in general computationally very complex. This is true also in an average sense if $\boldsymbol{H}$ is random (i.e., for a fading channel). The popular "sphere decoding" method [4], for example, is much more efficient than a brute-force search, but it still admits an average complexity that is exponential in the dimension of $\boldsymbol{s}$ [5].

In practice, the information bits that constitute $\boldsymbol{s}$ are usually part of a codeword over GF(2). For decoding, one then not only wants to know what the most likely vector $\boldsymbol{s}$ is, but it is also important to obtain reliability information (soft decisions, see Section II-B) for each bit. Direct computation of such soft decisions involves the summation of a number of terms that grows exponentially with the dimension of $\boldsymbol{s}$ and polynomially in the size of the signal constellation. The problem of computing good soft decisions is fundamentally much more important for slowly fading channels (i.e., when a codeword spans only one realization of $\boldsymbol{H}$) than in fast fading (where a codeword spans many realizations of $\boldsymbol{H}$). The reason is that in fast fading, simple linear preprocessing (for example, premultiplying $\boldsymbol{y}$ by a [pseudo-]inverse of $\boldsymbol{H}$) can be used to separate the symbols in $\boldsymbol{s}$ at a marginal loss of average mutual information between the transmitter and the receiver [1, Sec. 8.3]. This means that in fast fading, "most demodulators" have fairly good performance. By contrast, in slow fading, decoupling linear processing imposes a significant penalty on the outage mutual information. In particular, one can show that such linear preprocessing severely limits the diversity order of the system [1]. The main motivation for our work is therefore slowly fading channels.

Contributions: We present a novel approach to the problem of computing soft decisions for the model $\boldsymbol{y} = \boldsymbol{H}\boldsymbol{s} + \boldsymbol{e}$. The idea

Manuscript received March 31, 2007; revised March 27, 2008. This work was supported in part by the Swedish Research Council (VR). The work of E. Larsson was supported by Grant from the Knut and Alice Wallenberg Foundation, for being a Royal Swedish Academy of Sciences (KVA) Research Fellow. Parts of this work were presented at IEEE Globecom 2007. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Leslie Collins.

E. G. Larsson is with the Department of Electrical Engineering (ISY), Linköping University, 581 83 Linköping, Sweden (e-mail: erik.larsson@isy.liu.se).

J. Jaldén is with the Institute of Communications and Radio-Frequency Engineering (INTHFT), Vienna University of Technology, A-1040 Vienna, Austria (e-mail: joakim.jalden@nt.tuwien.ac.at).

Digital Object Identifier 10.1109/TSP.2008.925260

is a new way of approximating the marginal posterior probabilities for the information bits of which $\boldsymbol{s}$ is composed. Our new method has two major advantages: First, it provides impressive performance at low computational cost. Second, the computations required by the algorithm can be performed in parallel, and unlike competing methods its computational cost is *fixed* (i.e., independent of the particular realizations of $\boldsymbol{H}$, $\boldsymbol{s}$ and $\boldsymbol{e}$). This means that it is suitable for pipelined implementation and on parallel hardware architectures. We also show how the method can be extended to take into account uncertainty of the receiver's knowledge of the channel $\boldsymbol{H}$.

This paper is the journal version of [6]. The main contributions with respect to [6] include extensions to imperfect channel state information at the receiver, extensions to soft-input, and more extensive discussion.

Notation: $(\cdot)^*$ = complex conjugate; $(\cdot)^T$ = transpose; $\sigma_{\max}(\cdot)$ = largest singular value; $\sigma_{\min}(\cdot)$ = smallest singular value; $\otimes$ = Kronecker product; $\kappa(\cdot)$ = condition number of a matrix; $\boldsymbol{I}$ = identity matrix; $\Pi_{\boldsymbol{X}}$ = orthogonal projection onto the range of $\boldsymbol{X}$; $\Pi_{\boldsymbol{X}}^{\perp}$ = orthogonal projection onto the orthogonal complement of the range of $\boldsymbol{X}$.

II. Preliminaries

A. Channel Model

We consider a real-valued discrete-time matrix-vector channel model of the form

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{s} + \boldsymbol{e} \tag{1}$$

where $\boldsymbol{y} \in \mathbb{R}^{n_r}$ is a received n_r-vector, $\boldsymbol{s} \in \mathbb{R}^{n_t}$ is a transmitted n_t-vector, $\boldsymbol{H} \in \mathbb{R}^{n_r \times n_t}$ is a channel matrix of dimension $n_r \times n_t$ and $\boldsymbol{e} \in \mathbb{R}^{n_r}$ is an n_r-vector of noise. The channel $\boldsymbol{H}$ is perfectly known to the receiver, unless otherwise stated (this will be relaxed in Section V-A). The noise vector $\boldsymbol{e}$ has independent Gaussian elements with zero mean and variance $N_0/2$. Hence

$$p(\boldsymbol{y}|\boldsymbol{s}) = \frac{1}{\sqrt{\pi^{n_t} N_0^{n_t}}} \exp\left(-\frac{1}{N_0}\|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{s}\|^2\right). \tag{2}$$

In most applications, $n_r \geq n_t$, but this is not strictly required unless explicitly stated.

Equation (1) can model any linear communication channel. In particular, it can model a standard complex-baseband MIMO channel model of the form $\bar{\boldsymbol{y}} = \bar{\boldsymbol{H}}\bar{\boldsymbol{s}} + \bar{\boldsymbol{e}}$, with the real parts representing the inphase component of the signal and the imaginary part representing the quadrature component ($\bar{\boldsymbol{y}} = \boldsymbol{y}_I + j\boldsymbol{y}_Q$, for example). For such a complex MIMO channel, set n_r to twice the number of receive antennas, set n_t to twice the number of transmit antennas, and then take

$$\boldsymbol{y} = \begin{bmatrix} \mathrm{Re}\{\bar{\boldsymbol{y}}\} \\ \mathrm{Im}\{\bar{\boldsymbol{y}}\} \end{bmatrix}, \quad \boldsymbol{s} = \begin{bmatrix} \mathrm{Re}\{\bar{\boldsymbol{s}}\} \\ \mathrm{Im}\{\bar{\boldsymbol{s}}\} \end{bmatrix}, \quad \boldsymbol{e} = \begin{bmatrix} \mathrm{Re}\{\bar{\boldsymbol{e}}\} \\ \mathrm{Im}\{\bar{\boldsymbol{e}}\} \end{bmatrix}$$

and

$$\boldsymbol{H} = \begin{bmatrix} \mathrm{Re}\{\bar{\boldsymbol{H}}\} & -\mathrm{Im}\{\bar{\boldsymbol{H}}\} \\ \mathrm{Im}\{\bar{\boldsymbol{H}}\} & \mathrm{Re}\{\bar{\boldsymbol{H}}\} \end{bmatrix}. \tag{3}$$

(This requires that the symbols in $\bar{\boldsymbol{s}}$ come from a separable constellation, such as quadrature phase-shift keying [QPSK], rectangular M-ary quadrature amplitude modulation [M-QAM], but *not* M-ary phase shift keying [M-PSK] for $M \neq 2$ or 4.) More generally, the model (1) can describe a MIMO system that uses linear space-time block coding (STBC) to implement a complex-field code over a small number of channel symbols. To use the model (1) in this way, one must replace $\boldsymbol{H}$ with an "equivalent channel matrix" whose structure depends on the particular STBC being used. See [7, ch. 7], for example, for the precise mathematics involved. (In the special case of orthogonal STBCs, $\boldsymbol{H}$ would be proportional to an orthogonal matrix; demodulation is then trivial.) For generality, in the rest of the paper we will make no assumptions on what structure $\boldsymbol{H}$ in (1) may have, except for in the extension to imperfect channel state information (Section V-A) and the numerical results (Section VI).

Throughout we assume that the vector $\boldsymbol{s}$ in (1) has elements that belong to a finite alphabet $\mathcal{S}$, for example binary phase-shift keying (BPSK) or pulse amplitude modulation (PAM). Each element of $\boldsymbol{s}$, say s_k, is composed of m information bits. Hence the vector $\boldsymbol{s}$ is composed of $n_t m$ bits, which we denote $b_1, \ldots, b_{n_t m}$. For example, for BPSK modulation per real dimension we have $m = 1$. (This corresponds to QPSK modulation for a complex-valued MIMO channel.) For $m = 2$, we have 4-PAM (for a complex-valued MIMO channel, this corresponds to 16-QAM). We assume that the bits $b_1, \ldots, b_{n_t m}$ are mutually independent. This can usually be guaranteed in practice by an interleaver. To each bit b_i we associate an *a priori* log-likelihood ratio (LLR)

$$L(b_i) = \log\left(\frac{P(b_i = 1)}{P(b_i = 0)}\right)$$

which expresses what the detector knows about the bit *before* the data $\boldsymbol{y}$ were observed. Typically, if the demodulator is used as a building block in an iterative (turbo) receiver, this *a priori* LLR is the extrinsic output from the channel decoder.

B. Exact, Optimal Demodulation for the Model (1)

The task of the demodulator is to compute posterior LLRs for all information bits that constitute $\boldsymbol{s}$. The following calculation gives an explicit expression for this LLR [8]: [see (4) at the bottom of the next page]. Here $b_{i'}(\boldsymbol{s})$ is the i'th information bit in the transmitted vector $\boldsymbol{s}$. The notation $\boldsymbol{s} : b_i(\boldsymbol{s}) = \beta$ means the set of all vectors $\boldsymbol{s}$ for which $b_i(\boldsymbol{s})$ equals β, and $P(\boldsymbol{s})$ denotes the *a priori* probability that the vector $\boldsymbol{s}$ was transmitted. In (a), we marginalized over all possible $\boldsymbol{s}$. In (b), we used Bayes theorem. In (c), we used that all bits are *a priori* independent. In (d), we used (2).

For simplicity of exposition we shall now assume that all bits are equally likely to be 0 or 1 before observing $\boldsymbol{y}$. (This assumption is made throughout Sections III, IV, and V-A, but will be relaxed in Section V-B.) Then we have $P(b_i = 1) = P(b_i = 0) = 1/2$ so (4) can be written

$$L(b_i|\boldsymbol{y}) = \log\left(\frac{\sum_{\boldsymbol{s}:b_i(\boldsymbol{s})=1} \exp\left(-\frac{1}{N_0}\|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{s}\|^2\right)}{\sum_{\boldsymbol{s}:b_i(\boldsymbol{s})=0} \exp\left(-\frac{1}{N_0}\|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{s}\|^2\right)}\right). \tag{5}$$

The fundamental problem with computing (5) is that the sums contain altogether 2^{mn_t} terms. This makes direct evaluation of (5) infeasible in many cases of practical interest and one will have to resort to approximations (see Section III). That said, for

small m, n_t (e.g., complex MIMO with two transmit antennas and QPSK modulation; that is, $n_t = 4$ and $m = 1$) one can compute $L(b_i|\boldsymbol{y})$ brute-force at a modest effort. When doing so, it is useful to compute the numerator and denominator of (5) recursively, exploiting the so-called "Jacobian logarithm" identity [9]:

$$\log(e^a + e^b) = \max(a,b) + \log\left(1 + e^{-|a-b|}\right). \quad (6)$$

Equation (6) can be implemented efficiently and with good numerical stability even in fixed-point arithmetic. In particular, the last term can be implemented via a table-lookup as a function of $|a-b|$.

III. Previous Related Work and Known Suboptimal Approaches

We next review some existing approaches to the approximate computation of (5).

A. Max-Log Approximation

The idea behind this method is to approximate each of the sums in (5) with their largest term. This gives

$$L(b_i|\boldsymbol{y}) \approx \log\left(\frac{\max_{\boldsymbol{s}:b_i(\boldsymbol{s})=1} \exp\left(-\frac{1}{N_0}\|\boldsymbol{y}-\boldsymbol{H}\boldsymbol{s}\|^2\right)}{\max_{\boldsymbol{s}:b_i(\boldsymbol{s})=0} \exp\left(-\frac{1}{N_0}\|\boldsymbol{y}-\boldsymbol{H}\boldsymbol{s}\|^2\right)}\right). \quad (7)$$

(Note that this corresponds to omitting the second term in (6).)

Finding the maximum term in each sum is equivalent to minimizing $\|\boldsymbol{y}-\boldsymbol{H}\boldsymbol{s}\|^2$ subject to the constellation constraint on $\boldsymbol{s}$, and this is an NP-hard optimization problem. Therefore the max-log approximation, while it is conceptually simple, still suffers from being computationally very intensive. Minimizing $\|\boldsymbol{y}-\boldsymbol{H}\boldsymbol{s}\|^2$ by hard decision sphere decoding [4] is feasible, but incurs exponential average complexity [5]. This limits the range of problem sizes which may be addressed. Moreover, the complexity of this approach is random (i.e., it depends on the realizations of $\boldsymbol{H}$, $\boldsymbol{e}$ and $\boldsymbol{s}$) which may introduce random decoding delays. Additionally, sphere decoding is based on searching through a tree in a sequential manner, and, therefore, difficult to execute efficiently on parallel hardware architectures.

An alternative approach, which is more attractive from a computational complexity point of view, is to only approximately solve the maximization problems in (7) using a suboptimal hard decision detector, e.g., ZF or ZF-DFE. Unfortunately such approximations tend to perform poorly, especially when $\boldsymbol{H}$ is poorly conditioned. One could of course also choose another (presumably stronger) detector for the hard decision detection problem in the max-log approximation in (7). One such approach was proposed in [10] where a so-called semidefinite relaxation detector was used to solve the maximization problem in (7). However, most "near-optimal" hard decision detectors tend to be significantly more complex than the ZF and ZF-DFE detectors.

B. List-Sphere Decoding

The expression in (5) can be approximated by restricting the sums to a smaller set of admissible symbol vectors. The list-sphere-decoder [11] uses the conventional sphere decoder algorithm to find the list $\mathcal{L}$ of all candidate vectors that lie inside a sphere

$$\mathcal{L} \triangleq \left\{\boldsymbol{s} \in \mathcal{S}^{n_t} : \|\boldsymbol{y}-\boldsymbol{H}\boldsymbol{s}\|^2 \le C\right\}$$

for some constant C. It then approximates (5) according to

$$L(b_i|\boldsymbol{y}) \approx \log\left(\frac{\sum_{\boldsymbol{s}\in\mathcal{L}:b_i(\boldsymbol{s})=1} \exp\left(-\frac{1}{N_0}\|\boldsymbol{y}-\boldsymbol{H}\boldsymbol{s}\|^2\right)}{\sum_{\boldsymbol{s}\in\mathcal{L}:b_i(\boldsymbol{s})=0} \exp\left(-\frac{1}{N_0}\|\boldsymbol{y}-\boldsymbol{H}\boldsymbol{s}\|^2\right)}\right).$$

An advantage of this method is its conceptual simplicity. However, it is based on the sphere decoding algorithm, whose expected complexity grows exponentially with n_t [5]. Additionally, its complexity is random and the list-sphere decoder in-

$$\begin{aligned}
L(b_i|\boldsymbol{y}) &= \log\left(\frac{P(b_i=1|\boldsymbol{y})}{P(b_i=0|\boldsymbol{y})}\right) \overset{(a)}{=} \log\left(\frac{\sum_{\boldsymbol{s}:b_i(\boldsymbol{s})=1} P(\boldsymbol{s}|\boldsymbol{y})}{\sum_{\boldsymbol{s}:b_i(\boldsymbol{s})=0} P(\boldsymbol{s}|\boldsymbol{y})}\right) \overset{(b)}{=} \log\left(\frac{\sum_{\boldsymbol{s}:b_i(\boldsymbol{s})=1} p(\boldsymbol{y}|\boldsymbol{s})P(\boldsymbol{s})}{\sum_{\boldsymbol{s}:b_i(\boldsymbol{s})=0} p(\boldsymbol{y}|\boldsymbol{s})P(\boldsymbol{s})}\right) \\
&\overset{(c)}{=} \log\left(\frac{\sum_{\boldsymbol{s}:b_i(\boldsymbol{s})=1} p(\boldsymbol{y}|\boldsymbol{s})\left(\prod_{i'=1}^{n_t m} P\left(b_{i'}=b_{i'}(\boldsymbol{s})\right)\right)}{\sum_{\boldsymbol{s}:b_i(\boldsymbol{s})=0} p(\boldsymbol{y}|\boldsymbol{s})\left(\prod_{i'=1}^{n_t m} P\left(b_{i'}=b_{i'}(\boldsymbol{s})\right)\right)}\right) \\
&= \log\left(\frac{\sum_{\boldsymbol{s}:b_i(\boldsymbol{s})=1} p(\boldsymbol{y}|\boldsymbol{s})\left(\prod_{i'=1,i'\neq i}^{n_t m} P\left(b_{i'}=b_{i'}(\boldsymbol{s})\right)\right)\cdot P(b_i=1)}{\sum_{\boldsymbol{s}:b_i(\boldsymbol{s})=0} p(\boldsymbol{y}|\boldsymbol{s})\left(\prod_{i'=1,i'\neq i}^{n_t m} P\left(b_{i'}=b_{i'}(\boldsymbol{s})\right)\right)\cdot P(b_i=0)}\right) \\
&= \log\left(\frac{\sum_{\boldsymbol{s}:b_i(\boldsymbol{s})=1} p(\boldsymbol{y}|\boldsymbol{s})\left(\prod_{i'=1,i'\neq i}^{n_t m} P\left(b_{i'}=b_{i'}(\boldsymbol{s})\right)\right)}{\sum_{\boldsymbol{s}:b_i(\boldsymbol{s})=0} p(\boldsymbol{y}|\boldsymbol{s})\left(\prod_{i'=1,i'\neq i}^{n_t m} P\left(b_{i'}=b_{i'}(\boldsymbol{s})\right)\right)}\right) + L(b_i) \\
&\overset{(d)}{=} \log\left(\frac{\sum_{\boldsymbol{s}:b_i(\boldsymbol{s})=1} \exp\left(-\frac{1}{N_0}\|\boldsymbol{y}-\boldsymbol{H}\boldsymbol{s}\|^2\right)\left(\prod_{i'=1,i'\neq i}^{n_t m} P\left(b_{i'}=b_{i'}(\boldsymbol{s})\right)\right)}{\sum_{\boldsymbol{s}:b_i(\boldsymbol{s})=0} \exp\left(-\frac{1}{N_0}\|\boldsymbol{y}-\boldsymbol{H}\boldsymbol{s}\|^2\right)\left(\prod_{i'=1,i'\neq i}^{n_t m} P\left(b_{i'}=b_{i'}(\boldsymbol{s})\right)\right)}\right) + L(b_i). \quad (4)
\end{aligned}$$

herits the implementation issues of the original sphere decoder. There is also an issue relating to how various parameters of the algorithm should be selected in order for the subsets $\{\boldsymbol{s} \in \mathcal{L} : b_i(\boldsymbol{s}) = \beta\}$ to always contain at least one member. Note that selecting C properly for the list-sphere decoder is more critical for the performance than for the hard-decision sphere decoder, where the decoder complexity, assuming Schnorr-Euchner enumeration with adaptive radius updates [4], is not strongly affected by the initial radius.

C. Zero-Forcing With Heuristics

This method is introduced as a "simplest-possible" baseline to see what one can do with extremely low (and non-random) complexity. The idea is to first preprocess the received vector $\boldsymbol{y}$ with a ZF linear filter, to obtain

$$\tilde{\boldsymbol{y}} \triangleq (\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T\boldsymbol{y} = \boldsymbol{s} + (\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T\boldsymbol{e} = \boldsymbol{s} + \tilde{\boldsymbol{e}}$$

where $\tilde{\boldsymbol{e}}$ is zero-mean Gaussian with covariance matrix $(N_0/2)\cdot(\boldsymbol{H}^T\boldsymbol{H})^{-1}$. Then, by neglecting the correlation between the elements in $\tilde{\boldsymbol{e}}$, the bits that constitute $\boldsymbol{s}$ can be demodulated with a standard demodulator for the *scalar* channel

$$\tilde{y}_k = s_k + \tilde{e}_k$$

where $\tilde{e}_k$ is Gaussian with zero mean and variance $(N_0/2)\cdot[(\boldsymbol{H}^T\boldsymbol{H})^{-1}]_{k,k}$. (This scheme requires $n_r \geq n_t$, otherwise $\boldsymbol{H}^T\boldsymbol{H}$ is not invertible.)

In fast fading the loss introduced by the linear preprocessing is small and this scheme in fact is close to optimal [1]. However, in slow fading, which is the case of interest, the scheme performs poorly due to its inability to handle ill-conditioned channel matrix realizations. In particular, it offers a diversity order of at most $n_r - n_t + 1$ (out of $n_r n_t$ "possible"). One can show that the heuristic method presented here is equivalent to max-log using ZF detection without clipping. Also note that somewhat better performance can be obtained by using an MMSE filter instead of ZF, but this does not change the fundamental problems of the scheme.

IV. New Approach to Demodulation for the Model (1)

We next present our proposed approach to the problem of computing (5). The idea is to perform a two-step marginalization of the posterior density for $\boldsymbol{y}$. More precisely, we propose to perform *exact* marginalization over a carefully chosen, fixed number, say r, of the $n_t m$ bits and to *approximately* marginalize over the remaining $n_t m - r$ bits, using the max-log philosophy. (The value of r will be a user parameter of the algorithm, and we will later discuss how to choose it.)

In what follows, we explain the approach in detail. Consider $L(b_i|\boldsymbol{y})$ in (5). By writing out the marginalization over all information bits explicitly, (5) can be equivalently written as (8), shown at the bottom of the page, where

$$\mu(b_1,\ldots,b_{n_t m}) \triangleq \exp\left(-\frac{1}{N_0}\left\|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{s}(b_1,\ldots,b_{n_t m})\right\|^2\right). \tag{9}$$

In (9), $\boldsymbol{s}(b_1,\ldots,b_{n_t m})$ stands for the symbol vector $\boldsymbol{s}$ which corresponds to the bits $\{b_1,\ldots,b_{n_t m}\}$. Let $\mathcal{I}$ be a bit index permutation on $[1,\ldots,n_t m]$. Then we marginalize (8) *exactly* over $b_{\mathcal{I}_1},\ldots,b_{\mathcal{I}_r}$ and *approximately*, using the max-log philosophy, over $b_{\mathcal{I}_{r+1}},\ldots,b_{\mathcal{I}_{n_t m}}$. This is made explicit in the following two equations. Let l be the unique integer such that $\mathcal{I}_l = i$. For i such that $i \in \{\mathcal{I}_1,\ldots,\mathcal{I}_r\}$ we use (10), shown at the bottom of the page. For all other values of i we use (11), shown at the bottom of the page. As they stand, (10) and (11) still require the solution of a number of NP-hard maximization problems. To overcome this, we propose to use computationally less

$$L(b_i|\boldsymbol{y}) = \log\left(\frac{\sum_{b_1=0}^{1}\cdots\sum_{b_{i-1}=0}^{1}\sum_{b_{i+1}=0}^{1}\cdots\sum_{b_{n_t m}=0}^{1}\mu(b_1,\ldots,b_{i-1},1,b_{i+1},\ldots,b_{n_t m})}{\sum_{b_1=0}^{1}\cdots\sum_{b_{i-1}=0}^{1}\sum_{b_{i+1}=0}^{1}\cdots\sum_{b_{n_t m}=0}^{1}\mu(b_1,\ldots,b_{i-1},0,b_{i+1},\ldots,b_{n_t m})}\right) \tag{8}$$

$$L(b_i|\boldsymbol{y}) \approx \log\left(\frac{\sum_{b_{\mathcal{I}_1}=0}^{1}\cdots\sum_{b_{\mathcal{I}_{l-1}}=0}^{1}\sum_{b_{\mathcal{I}_{l+1}}=0}^{1}\cdots\sum_{b_{\mathcal{I}_r}=0}^{1}\left(\max_{b_{\mathcal{I}_{r+1}},\ldots,b_{\mathcal{I}_{n_t m}}}\mu(b_1,\ldots,b_{i-1},1,b_{i+1},\ldots,b_{n_t m})\right)}{\sum_{b_{\mathcal{I}_1}=0}^{1}\cdots\sum_{b_{\mathcal{I}_{l-1}}=0}^{1}\sum_{b_{\mathcal{I}_{l+1}}=0}^{1}\cdots\sum_{b_{\mathcal{I}_r}=0}^{1}\left(\max_{b_{\mathcal{I}_{r+1}},\ldots,b_{\mathcal{I}_{n_t m}}}\mu(b_1,\ldots,b_{i-1},0,b_{i+1},\ldots,b_{n_t m})\right)}\right). \tag{10}$$

$$L(b_i|\boldsymbol{y}) \approx \log\left(\frac{\sum_{b_{\mathcal{I}_1}=0}^{1}\cdots\sum_{b_{\mathcal{I}_r}=0}^{1}\left(\max_{b_{\mathcal{I}_{r+1}},\ldots,b_{\mathcal{I}_{l-1}},b_{\mathcal{I}_{l+1}},\ldots,b_{\mathcal{I}_{n_t m}}}\mu(b_1,\ldots,b_{i-1},1,b_{i+1},\ldots,b_{n_t m})\right)}{\sum_{b_{\mathcal{I}_1}=0}^{1}\cdots\sum_{b_{\mathcal{I}_r}=0}^{1}\left(\max_{b_{\mathcal{I}_{r+1}},\ldots,b_{\mathcal{I}_{l-1}},b_{\mathcal{I}_{l+1}},\ldots,b_{\mathcal{I}_{n_t m}}}\mu(b_1,\ldots,b_{i-1},0,b_{i+1},\ldots,b_{n_t m})\right)}\right). \tag{11}$$

expensive approximations to these maxima, namely those provided by the hard decision ZF or ZF-DFE detectors. We advocate the ZF-DFE detector and we will use it in our numerical examples. It provides better performance than ZF but it has only slightly higher complexity than ZF on slowly fading channels. The reason is that it is enough to compute all matrix inverses and the optimal detection orders pertinent to the ZF-DFE detector once for each channel realization.[1]

Thus, our approach is composed of two approximations: (i) replacing the marginalization over $b_{\mathcal{I}_{r+1}}, \ldots, b_{\mathcal{I}_{n_t m}}$ by a max-log operation, and (ii) solving this max-log problem approximately using a low-complexity method. In Sections IV-A and IV-B, we will motivate these approximations and also discuss the choice of the bit ordering $\mathcal{I}$. In order to facilitate good ZF or ZF-DFE solutions to the maximization problems in (10) and (11), the bit index permutation $\mathcal{I}$ must be chosen so that the bits $\{b_{\mathcal{I}_{r+1}}, \ldots, b_{\mathcal{I}_{n_t m}}\}$ in (10) and the bits $\{b_{\mathcal{I}_{r+1}}, \ldots, b_{\mathcal{I}_{l-1}}, b_{\mathcal{I}_{l+1}}, \ldots, b_{\mathcal{I}_{n_t m}}\}$ in (11) belong to a set of whole symbols most of the time.[2] Therefore we will work with entire *symbols* rather than individual *bits* when we create the partitioning between bits over which to marginalize exactly and bits over which to marginalize approximately. (Recall, that bits and symbols coincide for $m = 1$.) To accomplish this, we will in what follows assume that r is an integral multiple of m, say $p \triangleq r/m$, and that the bit index permutation $\mathcal{I}$ is defined by a *symbol index* permutation $\mathcal{J}$ on $[1, \ldots, n_t]$ as follows:

$$\mathcal{I}_i \triangleq m(\mathcal{J}_k - 1) + q$$

where k and q, $0 \leq q < m$, are the unique integers for which $i = m(k-1) + q + 1$, i.e., $k = \lfloor (i-1)/m \rfloor + 1$ and $q = (i-1) \bmod m$. For example, for $m = 2$ we have that

$$\mathcal{J} = \{3, 1, 2, 4\} \Rightarrow \mathcal{I} = \{5, 6, 1, 2, 3, 4, 7, 8\}.$$

By convention, for $r = 0$ there are no explicit sums in (10) and (11), i.e., only approximate marginalization takes place. This means that the method reduces to the max-log approximation, using ZF-DFE detection to perform the approximate maximization in (7).

Let us partition $\boldsymbol{H}$ and $\boldsymbol{s}$ according to

$$\boldsymbol{H}_a \triangleq \left[\boldsymbol{h}_{\mathcal{J}_1}, \ldots, \boldsymbol{h}_{\mathcal{J}_p}\right], \quad \boldsymbol{H}_b \triangleq \left[\boldsymbol{h}_{\mathcal{J}_{p+1}}, \ldots, \boldsymbol{h}_{\mathcal{J}_{n_t}}\right]$$
$$\boldsymbol{s}_a \triangleq \left[\boldsymbol{s}_{\mathcal{J}_1}, \ldots, \boldsymbol{s}_{\mathcal{J}_p}\right]^T, \quad \boldsymbol{s}_b \triangleq \left[\boldsymbol{s}_{\mathcal{J}_{p+1}}, \ldots, \boldsymbol{s}_{\mathcal{J}_{n_t}}\right]^T.$$

Then (1) can be written $\boldsymbol{y} = \boldsymbol{H}_a \boldsymbol{s}_a + \boldsymbol{H}_b \boldsymbol{s}_b + \boldsymbol{e}$. The maxima appearing in (10) are now obtained by computing

$$\min_{\hat{\boldsymbol{s}}_b \in \mathcal{S}^{n_t - p}} \|\boldsymbol{y} - \boldsymbol{H}_a \boldsymbol{s}_a - \boldsymbol{H}_b \hat{\boldsymbol{s}}_b\|^2 \triangleq f(\boldsymbol{H}, \boldsymbol{y}, \boldsymbol{s}_a). \tag{12}$$

[1]Note that the decision feedback involved in ZF-DFE, and the associated detection orders, just refers to the nulling-cancelling mechanism used to solve the maximization problems in (10) and (11). These implementation details of ZF-DFE have nothing to do with the proposed two-step marginalization, nor with the choice of the index mapping $\mathcal{I}$.

[2]Note that ZF or ZF-DFE is fully possible with symbols that contain an arbitrary number of bits, even if not all these bits are considered "unknown"—just perform ZF or ZF-DFE detection and then discard the bits that are not if interest. Doing so degrades the quality of the result, however.

The ZF approximation of (12) is obtained according to

$$f(\boldsymbol{H}, \boldsymbol{y}, \boldsymbol{s}_a) \approx \|\boldsymbol{y} - \boldsymbol{H}_a \boldsymbol{s}_a - \boldsymbol{H}_b \hat{\boldsymbol{s}}_{b,\mathrm{ZF}}\|^2 \tag{13}$$

where

$$\hat{\boldsymbol{s}}_{b,\mathrm{ZF}} \triangleq \arg \min_{\boldsymbol{s}_b \in \mathcal{S}^{n_t - p}} \left\| \left(\boldsymbol{H}_b^T \boldsymbol{H}_b\right)^{-1} \boldsymbol{H}_b^T (\boldsymbol{y} - \boldsymbol{H}_a \boldsymbol{s}_a) - \boldsymbol{s}_b \right\|^2 \tag{14}$$

is the ZF estimate of $\boldsymbol{s}_b$ given $\boldsymbol{y}$ and $\boldsymbol{s}_a$. The maxima in (11), and their ZF approximation, are obtained in a similar manner. The ZF-DFE approximation of (12) is obtained by replacing $\hat{\boldsymbol{s}}_{b,\mathrm{ZF}}$ in (13) by its ZF-DFE counterpart.

As an aside, we note that a somewhat similar approach for computing LLR values was suggested in [12]. The method of [12] is based on mariginalizations similar to (10), although with the outer sum of (10) replaced by its maximum term, while keeping the inner maximization approximate. However, with no marginalizations on the form of (11), this method requires several different symbol permutations in order facilitate the computation of the all LLRs using (10), thereby limiting the freedom to optimize the permutations with respect to error probability performance.

We will later argue that $\mathcal{I}$ (or equivalently $\mathcal{J}$) should be chosen so that it contains the indexes corresponding to the "worst" (in some sense) bits. This idea is inspired by the work on hard detection in [13]. It is also conceptually reminiscent of the work on decoding block codes presented in [14]. It is interesting to note that, in the MIMO detection context, the philosophy of marginalizing over the "worst" symbols stands in sharp contrast to the paradigm that most existing detectors use. For example, when implementing algorithms like ZF-DFE one usually tries to detect the "best" symbols first to minimize the effects of error propagation.

A. Motivation of the Proposed Approach

Clearly, by varying r between 0 and $n_t m$ one can trade off between the exact computation in (5) and the approximation in (7), with the additional approximation that the maxima are not computed exactly. As the total number of points appearing in the sums in (10) and (11) grows as 2^r it stands to reason that for the suggested approximation to have some merit, an accurate approximation of (5) must be obtained for a relatively small value of r. This is indeed the case, and in what follows we explain why.

For a fixed $\mathcal{J}$, increasing r has two main effects. One is that since the total number of terms over which $\mu(\cdot)$ is marginalized grows as 2^r, the sums in the numerator and denominator of (8) will be better and better approximated as r increases. An additional, very important but not so obvious, consequence of increasing r is that the quality of the approximation in (13) improves. To see why this is so, consider the QR-decomposition of $\boldsymbol{H}_b$ given by $\boldsymbol{Q}_b \boldsymbol{R}_b = \boldsymbol{H}_b$ and note that for any $\hat{\boldsymbol{s}}_b$ it holds that

$$\|\tilde{\boldsymbol{y}} - \boldsymbol{H}_b \hat{\boldsymbol{s}}_b\|^2 = \left\|\boldsymbol{Q}_b^T \tilde{\boldsymbol{y}} - \boldsymbol{R}_b \hat{\boldsymbol{s}}_b\right\|^2 + \left\|\boldsymbol{\Pi}_{\boldsymbol{H}_b}^{\perp} \tilde{\boldsymbol{y}}\right\|^2 \tag{15}$$

where $\tilde{\boldsymbol{y}} \triangleq \boldsymbol{y} - \boldsymbol{H}_a \boldsymbol{s}_a$ and where $\boldsymbol{\Pi}_{\boldsymbol{H}_b}^{\perp} \triangleq \boldsymbol{I} - \boldsymbol{Q}_b \boldsymbol{Q}_b^T$ denotes projection onto the orthogonal complement of the range of $\boldsymbol{H}_b$.

Thus, the quality of the approximation in (13) can be quantified according to

$$\frac{\|\tilde{\boldsymbol{y}}-\boldsymbol{H}_b\hat{\boldsymbol{s}}_{b,\mathrm{ZF}}\|^2}{\|\tilde{\boldsymbol{y}}-\boldsymbol{H}_b\hat{\boldsymbol{s}}_{b,\mathrm{ML}}\|^2}=\frac{\frac{\left\|\boldsymbol{Q}_b^T\tilde{\boldsymbol{y}}-\boldsymbol{R}_b\hat{\boldsymbol{s}}_{b,\mathrm{ZF}}\right\|^2}{\left\|\boldsymbol{\Pi}_{\boldsymbol{H}_b}^{\perp}\tilde{\boldsymbol{y}}\right\|^2}+1}{\frac{\left\|\boldsymbol{Q}_b^T\tilde{\boldsymbol{y}}-\boldsymbol{R}_b\hat{\boldsymbol{s}}_{b,\mathrm{ML}}\right\|^2}{\left\|\boldsymbol{\Pi}_{\boldsymbol{H}_b}^{\perp}\tilde{\boldsymbol{y}}\right\|^2}+1}\geq 1 \qquad (16)$$

where $\hat{\boldsymbol{s}}_{b,\mathrm{ML}}$ is the minimizer of (12). The quality of the approximation in (13) improves with increasing r and there are two reasons for this. First, since the number of columns in $\boldsymbol{H}_b$, equal to $n_t-p=n_t-r/m$, decreases with r, it follows that $\|\boldsymbol{\Pi}_{\boldsymbol{H}_b}^{\perp}\tilde{\boldsymbol{y}}\|^2$ increases with r. By (16) this implies that the approximation in (13) becomes less sensitive to "errors" in the ZF (or ZF-DFE) estimate of $\boldsymbol{s}_b$. In fact, if r approaches n_tm the right hand side of (16) approaches its lower limit which means that the approximation becomes tight. Second, we have that

$$\left\|\boldsymbol{Q}_b^T\tilde{\boldsymbol{y}}-\boldsymbol{R}_b\hat{\boldsymbol{s}}_{b,\mathrm{ZF}}\right\|^2\approx\left\|\boldsymbol{Q}_b^T\tilde{\boldsymbol{y}}-\boldsymbol{R}_b\hat{\boldsymbol{s}}_{b,\mathrm{ML}}\right\|^2. \qquad (17)$$

The accuracy of the approximation in (17) also increases with increasing r. This is a consequence of the following result.

Proposition 1: Given an invertible matrix $\boldsymbol{A}\in\mathbb{R}^{n\times n}$, $\boldsymbol{b}\in\mathbb{R}^n$, and $\mathcal{X}\subset\mathbb{R}^n$ let

$$\hat{\boldsymbol{x}}_{\mathrm{ML}}\triangleq\arg\min_{\boldsymbol{x}\in\mathcal{X}}\|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}\|^2,\quad\text{and}$$
$$\hat{\boldsymbol{x}}_{\mathrm{ZF}}\triangleq\arg\min_{\boldsymbol{x}\in\mathcal{X}}\|\boldsymbol{A}^{-1}\boldsymbol{b}-\boldsymbol{x}\|^2.$$

Then

$$\|\boldsymbol{b}-\boldsymbol{A}\hat{\boldsymbol{x}}_{\mathrm{ML}}\|\leq\|\boldsymbol{b}-\boldsymbol{A}\hat{\boldsymbol{x}}_{\mathrm{ZF}}\|\leq\kappa(\boldsymbol{A})\|\boldsymbol{b}-\boldsymbol{A}\hat{\boldsymbol{x}}_{\mathrm{ML}}\| \qquad (18)$$

where $\kappa(\boldsymbol{A})=\sigma_{\max}(\boldsymbol{A})/\sigma_{\min}(\boldsymbol{A})$ denotes the condition number of $\boldsymbol{A}$.

Proof: See the Appendix.

Proposition 1 states that the quality of the (hard) ZF decision is bounded in terms of $\kappa(\boldsymbol{A})$. In the special case where $\boldsymbol{A}$ is a scaled unitary matrix, $\kappa(\boldsymbol{A})=1$ and $\hat{\boldsymbol{x}}_{\mathrm{ZF}}=\hat{\boldsymbol{x}}_{\mathrm{ML}}$, i.e., the approximation is tight. Applying Proposition 1 with $\boldsymbol{A}=\boldsymbol{R}_b$ to (17) and noting that $\kappa(\boldsymbol{R}_b)=\sqrt{\kappa(\boldsymbol{H}_b^T\boldsymbol{H}_b)}$ is a decreasing function of r (since removing columns from $\boldsymbol{H}_b$ cannot increase its condition number)[3] it follows that the approximation in (17) improves by increasing the value of r.

B. Choosing the Index Permutation $\mathcal{J}$

Naturally, the performance of the proposed method depends on the chosen symbol index permutation, $\mathcal{J}$. However, as the overall effect of $\mathcal{J}$ on the performance depends on a complex interplay between the soft-output demodulator and the outer code it is difficult to say precisely how $\mathcal{J}$ should be chosen to optimize the overall performance. Still, the approximations discussed in Section IV-A provide some insight into what properties a "good" ordering $\mathcal{J}$ should have. Note that for a given ordering $\mathcal{J}$, the performance will always increase with increased r.

In light of the discussion following Proposition 1, we would like to choose $\mathcal{J}$ as a function of $\boldsymbol{H}$, so that the condition number of $\boldsymbol{H}_b^T\boldsymbol{H}_b$ is minimized. A direct implementation of this strategy requires a search over $\binom{n_t}{p}$ possible orderings (where $p=r/m$). This is feasible only if n_t and p are small. As an approximation, we suggest to use the following ordering strategy (proposed in [13] as a search order for hard decisions).

1) Let $\mathcal{J}=\varnothing$ (empty) and let $\mathcal{J}^c=[1,\ldots,n_t]$.
2) Compute $\boldsymbol{\gamma}=\mathrm{diag}\{(\boldsymbol{H}^T\boldsymbol{H})^{-1}\}$. Let k be the index of the *largest* element of $\boldsymbol{\gamma}$.
3) Set $\mathcal{J}:=[\mathcal{J},\mathcal{J}_k^c]$.
4) Remove the kth column from $\boldsymbol{H}$.
5) Remove the kth element of $\mathcal{J}^c$.
6) If $\boldsymbol{H}$ is empty, terminate. Otherwise, repeat from step 2.

Simulations indicate that this ordering yields an improvement in overall frame error rate (FER) over the natural (fixed) symbol ordering although the magnitude of the improvement is not as large as in the hard decision setup of [13]. Simulations also show close to optimal performance at relatively small values of r.

An even simpler alternative ordering would be to just sort $[(\boldsymbol{H}^T\boldsymbol{H})^{-1}]_{k,k}$ in decreasing order and take $\mathcal{J}$ to be the resulting index vector. The method outlined in the previous paragraph performs somewhat better however, since if there are two nearly "parallel" columns of $\boldsymbol{H}$ then the condition number can improve drastically by just removing *one* of these columns.

Note that we use the ordering $\mathcal{J}$ to decide what r bits to completely marginalize over via the sums in (10) and (11). However, when (approximately) computing the maxima in (10) and (11) via ZF-DFE, we use the optimal V-BLAST ordering.

C. Complexity

Table I summarizes the computational complexity of the proposed method and its competitors. Only operations per bit are reported while disregarding the preprocessing steps performed once per outer codeword. Also, note that the figures in Table I are only rough order estimates of the actual complexity. For instance, polynomial terms preceding 2^{mn_t} in the brute-force method [due to the number of operations required to evaluate the norms appearing in (5)] are omitted. Such polynomial terms are

TABLE I
COMPLEXITY-PERFORMANCE COMPARISON

	Ops. per bit	Fixed complexity?
Exact (5), brute-force	$\sim 2^{mn_t}$	Yes
Max-log (7) (brute-force)	$\sim 2^{mn_t}$	Yes
Max-log (7) (sphere decoding)	$\sim 2^{\alpha mn_t}$ [5]	No
List-sphere decoding	$\sim 2^{\beta mn_t}$ [5]	No
Proposed (10)–(11)	$\sim n_t^2 2^r$	Yes

[3]To see this, let $\boldsymbol{H}_b$ be the original matrix with r columns, and let $\bar{\boldsymbol{H}}_b$ be a submatrix of $\boldsymbol{H}_b$ that contains $\bar{r}$ arbitrarily selected columns, where $\bar{r}<r$. Note that $\bar{\boldsymbol{H}}_b^T\bar{\boldsymbol{H}}_b$ is a principal submatrix of $\boldsymbol{H}_b^T\boldsymbol{H}_b$. By the interlacing property of eigenvalues for principal submatrices (see, e.g., [15, Th. 4.3.15]) it follows that

$$\begin{aligned}\sigma_{\max}\left(\boldsymbol{H}_b^T\boldsymbol{H}_b\right)&\geq\sigma_{\max}\left(\bar{\boldsymbol{H}}_b^T\bar{\boldsymbol{H}}_b\right)\\&\geq\sigma_{\min}\left(\bar{\boldsymbol{H}}_b^T\bar{\boldsymbol{H}}_b\right)\\&\geq\sigma_{\min}\left(\boldsymbol{H}_b^T\boldsymbol{H}_b\right)\end{aligned}$$

and, thus

$$\kappa\left(\boldsymbol{H}_b^T\boldsymbol{H}_b\right)\geq\kappa\left(\bar{\boldsymbol{H}}_b^T\bar{\boldsymbol{H}}_b\right).$$

also omitted in the figures reported for the sphere decoder implementations. The constants α and β depend on the signal-to-noise-ratio (SNR) and on the constellation, but they satisfy $\beta > \alpha > 0$ for all finite SNR [5].

In order to obtain an order-of-magnitude estimate of the complexity per bit of the proposed method, using ZF decisions to solve the approximate maximization problems in (10) and (11), we note that the bulk of the computations is in forming $f(\boldsymbol{H},\boldsymbol{y},\boldsymbol{s}_a)$ in (13) for the 2^r vectors $\boldsymbol{s}_a \in \mathcal{S}^p$. Forming $\boldsymbol{u} \triangleq (\boldsymbol{H}_b^T\boldsymbol{H}_b)^{-1}\boldsymbol{H}_b^T(\boldsymbol{y}-\boldsymbol{H}_a\boldsymbol{s}_a)$ in (14) requires on the order of n_t^2 operations, assuming $n_r \approx n_t$, $n_t \gg p$ and that the matrix multiplications and inverses are precomputed and stored. The minimization in (14) is solved by rounding the components of $\boldsymbol{u} \in \mathbb{R}^{n_t-p}$ to the closest constellation points. This requires on the order of n_t operations, again assuming $n_t \gg p$. The complexity of evaluating (13) is dominated by the cost of evaluating $\boldsymbol{H}_b\hat{\boldsymbol{s}}_{b,\mathrm{ZF}}$ which requires on the order of $n_t(n_t-p) \approx n_t^2$ operations. Thus, the overall complexity of evaluating $f(\boldsymbol{H},\boldsymbol{y},\boldsymbol{s}_a)$ for all $\boldsymbol{s}_a$ is in the order of $n_t^2 2^r$. The same is true for the ZF-DFE version of the proposed method.

V. Extensions

A. Imperfect Channel State Information (CSI) at the Receiver

In practice, the receiver is never going to know $\boldsymbol{H}$ perfectly. We can model the uncertainty in the receiver's knowledge of $\boldsymbol{H}$, and more importantly, take this uncertainty into account when computing soft decisions on individual information bits b_i. In this section we will provide one way of doing this. The philosophy used here is closely inspired by [16] and [17].

Ultimately, if $\boldsymbol{H}$ were known then one would like to compute (4), with $p(\boldsymbol{y}|\boldsymbol{s})$ replaced by $p(\boldsymbol{y}|\boldsymbol{H},\boldsymbol{s})$ to make explicit the randomness of the channel $\boldsymbol{H}$ and to emphasize the receiver's complete knowledge of it. In practice $\boldsymbol{H}$ is not known. However one usually has some estimate of $\boldsymbol{H}$, say $\hat{\boldsymbol{H}}$. This estimate may be obtained from a received training sequence, say $\boldsymbol{y}_t$ and if so $\hat{\boldsymbol{H}}$ is a direct function of $\boldsymbol{y}_t$. What is then of interest is to compute (4) with $p(\boldsymbol{y}|\boldsymbol{s})$ replaced by $p(\boldsymbol{y}|\boldsymbol{s},\boldsymbol{y}_t)$, or $p(\boldsymbol{y}|\boldsymbol{s},\hat{\boldsymbol{H}})$. It is clear that using $p(\boldsymbol{y}|\boldsymbol{s},\boldsymbol{y}_t)$ for data detection must be *equivalent to, or better* than using $p(\boldsymbol{y}|\boldsymbol{s},\hat{\boldsymbol{H}})$, assuming that no information about $\boldsymbol{H}$ is injected by the mapping $\boldsymbol{y}_t \to \hat{\boldsymbol{H}}$. We shall compute the metric based on $p(\boldsymbol{y}|\boldsymbol{s},\hat{\boldsymbol{H}})$. There are two reasons for this: First, this quantity is independent of the particular training method used. Second, under certain conditions, one can show that using $p(\boldsymbol{y}|\boldsymbol{s},\boldsymbol{y}_t)$ and $p(\boldsymbol{y}|\boldsymbol{s},\hat{\boldsymbol{H}})$ leads to equivalent decisions on $\boldsymbol{s}$ [16].

To proceed we need to make a few explicit assumptions. The calculation we make will explicitly assume that we deal with a Rayleigh fading (complex) MIMO channel, so that $\boldsymbol{H}$ in (1) has the structure of (3). This is somewhat easier to handle by using complex number notation, so througout this subsection we use the model $\bar{\boldsymbol{y}} = \bar{\boldsymbol{H}}\bar{\boldsymbol{s}} + \bar{\boldsymbol{e}}$ where all quantities are complex-valud [the notation $\bar{(\cdot)}$ emphasizes this] and in particular, $\bar{\boldsymbol{e}}$ has i.i.d. complex Gaussian elements with zero mean and variance N_0. We can rewrite this model as follows:

$$\bar{\boldsymbol{y}} = \bar{\boldsymbol{H}}\bar{\boldsymbol{s}} + \bar{\boldsymbol{e}} = (\bar{\boldsymbol{s}}^T \otimes \boldsymbol{I})\bar{\boldsymbol{h}} + \bar{\boldsymbol{e}}$$

where $\bar{\boldsymbol{h}} \triangleq \mathrm{vec}(\bar{\boldsymbol{H}})$, and

$$\bar{\boldsymbol{e}} \sim N(\boldsymbol{0}, N_0\boldsymbol{I}).$$

As before, N_0 is the noise power per complex dimension. Then, suppose that *a priori* (before observing any data or any training), we know that the channel is i.i.d. Rayleigh fading

$$\bar{\boldsymbol{h}} \sim N(\boldsymbol{0}, \rho^2\boldsymbol{I})$$

where ρ^2 is the average squared gain of the channel per complex dimension. We now assume that we can model the receiver's knowledge of the channel as follows:

$$\hat{\bar{\boldsymbol{h}}} = \bar{\boldsymbol{h}} + \bar{\boldsymbol{\delta}}$$

where

$$\bar{\boldsymbol{\delta}} \sim N(\boldsymbol{0}, \epsilon^2\boldsymbol{I})$$

and where ϵ^2 represents the accuracy with which the channel is known (i.e., the channel estimation error variance per complex dimension). Additionally, we shall assume that $\bar{\boldsymbol{e}}, \bar{\boldsymbol{h}}, \hat{\bar{\boldsymbol{h}}}$ are mutually independent. (This set of assumptions does not limit the generality of the method although it does exclude the case when the channel estimate $\hat{\bar{\boldsymbol{h}}}$ is obtained from the same observations as those used to detect the data.) In practice, ϵ^2 may either be independent of N_0, or proportional to N_0 (i.e., $\epsilon^2 = \xi \cdot N_0$ for some positive ξ). The former case corresponds to the situation where the uncertainty in knowledge of $\bar{\boldsymbol{h}}$ stems from a delay between the instant when the channel is estimated and the point in time when it is used. In the latter case, the quality of the channel estimate is proportional to the SNR. This models estimation errors due to pilots that are subject to noise whose magnitude stands in proportion to the power of the noise that affects the data symbols. The theory we present is applicable to both scenarios (or any combination of them).

We can now compute $p(\bar{\boldsymbol{y}}|\bar{\boldsymbol{s}},\hat{\bar{\boldsymbol{h}}})$. For simplicity, we shall assume that $\bar{s}_k$ are constant modulus so that $\|\bar{\boldsymbol{s}}\|^2 = n_t$. We begin by observing that $\bar{\boldsymbol{y}}$ and $\hat{\bar{\boldsymbol{h}}}$ are jointly Gaussian, conditioned on $\bar{\boldsymbol{s}}$. More precisely they have the following joint conditional distribution:

$$\begin{bmatrix}\bar{\boldsymbol{y}}\\ \hat{\bar{\boldsymbol{h}}}\end{bmatrix} \sim N\left(\boldsymbol{0}, \begin{bmatrix}(n_t\rho^2+N_0)\boldsymbol{I} & \rho^2(\bar{\boldsymbol{s}}^T\otimes\boldsymbol{I})\\ \rho^2(\bar{\boldsymbol{s}}^*\otimes\boldsymbol{I}) & (\rho^2+\epsilon^2)\boldsymbol{I}\end{bmatrix}\right).$$

The following conditional distribution is then immediate:

$$\bar{\boldsymbol{y}}|\hat{\bar{\boldsymbol{h}}},\bar{\boldsymbol{s}} \sim N\left(\frac{\rho^2}{\rho^2+\epsilon^2}(\bar{\boldsymbol{s}}^T\otimes\boldsymbol{I})\hat{\bar{\boldsymbol{h}}}, \left(n_t\rho^2+N_0-\frac{n_t\rho^4}{\rho^2+\epsilon^2}\right)\boldsymbol{I}\right)$$

or equivalently

$$\bar{\boldsymbol{y}}|\hat{\bar{\boldsymbol{h}}},\bar{\boldsymbol{s}} \sim N\left(\frac{\rho^2}{\rho^2+\epsilon^2}\hat{\bar{\boldsymbol{H}}}\bar{\boldsymbol{s}}, \left(\frac{n_t\epsilon^2}{1+\epsilon^2/\rho^2}+N_0\right)\boldsymbol{I}\right).$$

It follows that:

$$p(\bar{\boldsymbol{y}}|\hat{\bar{\boldsymbol{h}}},\bar{\boldsymbol{s}}) = \frac{1}{\pi^{n_t}\tilde{N}_0^{n_t}}\exp\left(-\frac{1}{\tilde{N}_0}\|\bar{\boldsymbol{y}}-\eta\hat{\bar{\boldsymbol{H}}}\bar{\boldsymbol{s}}\|^2\right)$$

where we have defined

$$\eta \triangleq \frac{\rho^2}{\rho^2+\epsilon^2}$$
$$\tilde{N}_0 \triangleq \frac{n_t\epsilon^2}{1+\epsilon^2/\rho^2}+N_0.$$

The counterpart to (5) is (assuming the bits are equiprobable *a priori*, for simplicity)

$$L(b_i|\bar{\boldsymbol{y}},\hat{\bar{\boldsymbol{H}}}) = \log\left(\frac{\sum_{\bar{\boldsymbol{s}}:b_i(\bar{\boldsymbol{s}})=1}\exp\left(-\frac{1}{N_0}||\bar{\boldsymbol{y}}-\eta\hat{\bar{\boldsymbol{H}}}\bar{\boldsymbol{s}}||^2\right)}{\sum_{\bar{\boldsymbol{s}}:b_i(\bar{\boldsymbol{s}})=0}\exp\left(-\frac{1}{N_0}||\bar{\boldsymbol{y}}-\eta\hat{\bar{\boldsymbol{H}}}\bar{\boldsymbol{s}}||^2\right)}\right) = \log\left(\frac{\sum_{\boldsymbol{s}:b_i(\boldsymbol{s})=1}\exp\left(-\frac{1}{N_0}||\boldsymbol{y}-\eta\hat{\boldsymbol{H}}\boldsymbol{s}||^2\right)}{\sum_{\boldsymbol{s}:b_i(\boldsymbol{s})=0}\exp\left(-\frac{1}{N_0}||\boldsymbol{y}-\eta\hat{\boldsymbol{H}}\boldsymbol{s}||^2\right)}\right) \quad (19)$$

where in the last equality we return to the real-valued problem formulation in (1), that we use in the rest of this article.

If the channel has the fading distribution assumed here (Rayleigh), and if the receiver does not have perfect channel knowledge, then the detector based on (19) is going to outperform a detector that simply uses (5) with $\boldsymbol{H}$ replaced by an estimate $\hat{\boldsymbol{H}}$. We will see this in the numerical results in Section VI. Arguably this is so because side information about $\bar{\boldsymbol{H}}$ is injected when forming the conditional distribution $p(\boldsymbol{y}|\hat{\boldsymbol{h}},\boldsymbol{s})$. Nevertheless, in practice such side information may be available, and it can be collected for example by long-term averaging over multiple received blocks. Note, however, that no matter how inaccurate knowledge the receiver has about each individual realization of $\bar{\boldsymbol{H}}$, the receiver must *perfectly* know the statistics of $\boldsymbol{H}$, that is $p(\boldsymbol{H})$. If this knowledge is inaccurate, for example, the fading is not Rayleigh, then optimal performance will not be achieved. (See [18] for some general discussion on receiver design with imperfect receiver CSI in other, specific types of fading.)

An issue that we do not touch here is that of imperfect knowledge of the noise variance N_0. The gain that could be obtained by modifying (5) or (19) to take into account such uncertainty is presumably very small, however.

B. Taking Into Account Soft Input

The proposed method can be extended in a relatively straightforward manner to take into account soft input. What is of interest is then to reformulate the a posteriori LLR in (4) into a form which can be computed using the philosophy that we developed in Section IV. We shall briefly outline how this can be done, for the case of binary modulation per real dimension, i.e., $s_k \in \{\pm 1\}$. By convention, we will let $b = 1$ correspond to $s = +1$ and $b = 0$ correspond to $s = -1$. Note that

$$\begin{aligned}\log\left(P(s_k = s)\right) &= \frac{1}{2}\left((1+s)\log\left(P(s_k=1)\right)\right.\\ &\qquad \left.+(1-s)\log\left(P(s_k=-1)\right)\right)\\ &= \frac{1}{2}\gamma_k + \frac{1}{2}\lambda_k s \end{aligned} \quad (20)$$

where we have defined

$$\begin{aligned}\gamma_k &\triangleq \frac{1}{2}\log\left(P(s_k=-1)P(s_k=1)\right)\\ &= \frac{1}{2}\log\left(P(b_k=0)P(b_k=1)\right)\end{aligned}$$

and where

$$\lambda_k \triangleq \log\left(\frac{P(s_k=1)}{P(s_k=-1)}\right) = \log\left(\frac{P(b_k=1)}{P(b_k=0)}\right) = L(b_k)$$

is a shorthand for the *a priori* log-likelihood ratio of the kth symbol (bit).

Now using (20), we can write (4) as shown in the equation at the bottom of the page. Let us define

$$\tilde{\boldsymbol{H}}_k \triangleq \begin{bmatrix}\boldsymbol{H}\\ \boldsymbol{\Lambda}_k\end{bmatrix}$$

where

$$\boldsymbol{\Lambda}_k \triangleq \operatorname{diag}\left\{\frac{N_0}{4}\lambda_1,\cdots,\frac{N_0}{4}\lambda_{k-1},0,\frac{N_0}{4}\lambda_{k+1},\ldots,\frac{N_0}{4}\lambda_{n_t}\right\}$$

Also define $\tilde{\boldsymbol{y}} \triangleq \left[\boldsymbol{y}^T 1\cdots 1\right]^T$. Then, (21) is arrived at (see the bottom of the page). Equation (21) can be computed directly using the strategy developed in Section IV.

One can interpret the augmented channel matrix $\tilde{\boldsymbol{H}}$ as adding n_t-1 "virtual observations" (virtual antennas for MIMO) which provide the *a priori* information contained in $L(b_i)$. The smaller N_0 is, the less this *a priori* information will impact the decisions made by the detector. Note that the strategy of incorporating the *a priori* LLRs into the bit metrics as a linear perturbation was used also in [10].

VI. Empirical Performance Evaluation

In this section, we present numerical results to illustrate the performance of the proposed approach. Monte Carlo simulation was used to estimate the frame-error-rate. At each SNR

$$L(s_k|\boldsymbol{y}) = \log\left(\frac{\sum_{\boldsymbol{s}:b_k=1}\exp\left(-\frac{1}{N_0}||\boldsymbol{y}-\boldsymbol{H}\boldsymbol{s}||^2+\frac{1}{2}\sum_{k'=1,k'\neq k}^{n_t}(\gamma_{k'}+\lambda_{k'}s_{k'})\right)}{\sum_{\boldsymbol{s}:b_k=0}\exp\left(-\frac{1}{N_0}||\boldsymbol{y}-\boldsymbol{H}\boldsymbol{s}||^2+\frac{1}{2}\sum_{k'=1,k'\neq k}^{n_t}(\gamma_{k'}+\lambda_{k'}s_{k'})\right)}\right)+\lambda_k$$

$$L(s_k|\boldsymbol{y}) = \log\left(\frac{\sum_{\boldsymbol{s}:b_k=1}\exp\left(-\frac{1}{N_0}||\tilde{\boldsymbol{y}}-\tilde{\boldsymbol{H}}_k\boldsymbol{s}||^2+\sum_{k'=1,k'\neq k}^{n_t}\left(\frac{N_0\lambda_{k'}^2}{16}+\frac{\gamma_{k'}}{2}\right)\right)}{\sum_{\boldsymbol{s}:b_k=0}\exp\left(-\frac{1}{N_0}||\tilde{\boldsymbol{y}}-\tilde{\boldsymbol{H}}_k\boldsymbol{s}||^2+\sum_{k'=1,k'\neq k}^{n_t}\left(\frac{N_0\lambda_{k'}^2}{16}+\frac{\gamma_{k'}}{2}\right)\right)}\right)+\lambda_k. \quad (21)$$

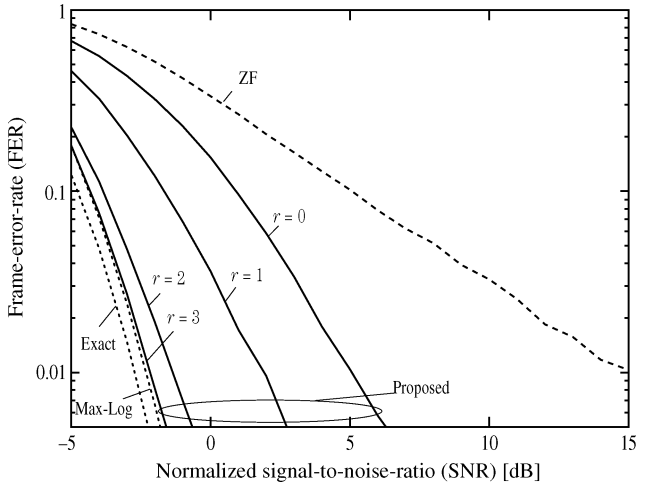

Fig. 1. Performance comparison for a 4×4 complex MIMO system ($n_r = n_t = 8$ in (1)) with QPSK modulation ($m = 1$) on a slowly fading Rayleigh channel, using a rate-1/3 convolutional code with 100 bits long codewords that span one channel realization. The figure shows the performance of (i) the proposed method [Eq. (10), (11)] with different values of r. (ii) exact [Eq. (5)] evaluated by brute-force enumeration. (iii) Max-Log [Eq. (7)] evaluated by brute-force enumeration. (iv) Heuristic zero-forcing as described in Section III-C.

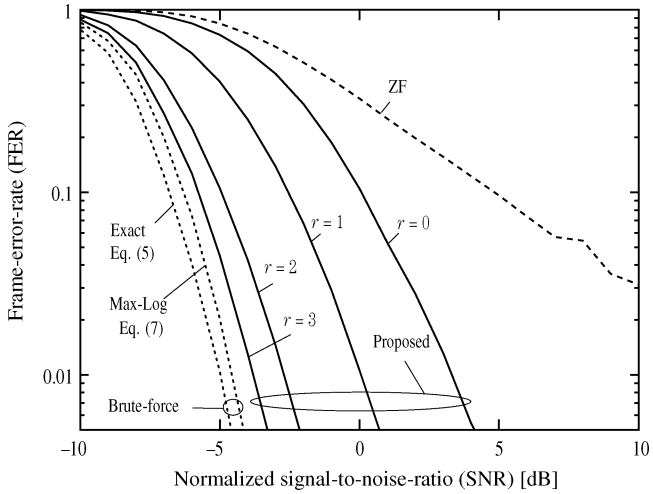

Fig. 2. Same as Fig. 1 but for a 6×6 complex MIMO channel ($n_t = n_r = 12$ in (1)) with QPSK ($m = 1$).

point, we simulated enough frames to count 500 frame errors. We compare the performance of our method to that of exact demodulation (5), and to that of max-log (7). We used brute-force enumeration of all $2^{n_t m-1}$ permissible vectors $\boldsymbol{s}$ to evaluate the numerators and denominators of (5) and (7). This is computationally costly, although the computational cost is nonrandom. Note that in principle (5) and (7) can be computed using sphere decoding, albeit at a random (and high; see Table I) computational cost.

In all simulations we consider coded transmission over a slowly fading (quasi-static) Rayleigh channel. Each codeword spans one realization of $\boldsymbol{H}$. To simulate the channel we generated $\boldsymbol{H}$ with the structure (3), where $\bar{\boldsymbol{H}}$ has independent, zero-mean complex Gaussian elements with variance N_0. We present the frame-error-rate as a function of the normalized SNR, defined as E_b/N_0 where E_b is the transmitted energy per uncoded bit.

A. Detection Performance for Different r

In the first example, we quantify the detection performance for different choices of r and different channel dimensions. In this example, we used a convolutional code with block length 100 bits and rate 1/3 as outer code. The code is decoded with the Viterbi algorithm, and there is no iteration between the decoder and the demodulator.

In Figs. 1 and 2, we show results for a 4×4 and a 6×6 complex MIMO system, respectively, with QPSK modulation. [That is, $m = 1$ and $n_t = n_r = 8$ and $n_t = n_r = 12$, respectively, in (1).] The performance of the proposed method is impressive. In fact the results suggest that the largest improvements (relative to $r = 0$) are achieved for relatively small r. Choosing $r = 3$ produces results close to exact demodulation (5) for both examples. Note that for the 6×6 system, the number of terms in (5) is 2^{11}, whereas with $r = 3$ our method only sums over 2^3 terms. This represents a complexity saving of a factor $2^{11-3} = 256$.

B. Detection With Imperfect CSI at the Receiver

Next, we illustrate the performance of the demodulator based on (19), which is tuned to handle imperfect channel knowledge at the receiver. We consider a 4×4 complex MIMO system, with all parameters being the same as in Fig. 1 (slow Rayleigh fading). We consider two cases: (i) the quality of the channel estimate being constant with respect to the SNR (ϵ^2 fixed), and (ii) the quality channel estimate degrading inverse proportionally with SNR ($\epsilon^2 = \xi \cdot N_0$, ξ fixed). Case (i) effectively models the use of an outdated channel estimate and case (ii) models channel estimation errors due to noisy pilots. Fig. 3 shows the results for case (i) and Fig. 4 shows the results for case (ii). We compare (a) the coherent metric [using (5) with the true $\boldsymbol{H}$], (b) the mismatched metric [using (5) with $\bar{\boldsymbol{H}}$ replaced by $\hat{\boldsymbol{H}}$] and (c) the proposed metric (19). In all cases, the metric was evaluated using our proposed strategy (Section IV) with $r = 3$.

It is clear that the proposed metric provides better performance than the mismatched metric. Note that in case (ii), the detection loss (the difference between the receivers having imperfect CSI and the coherent receiver) stays constant when the SNR increases. This means that the detectors (b) and (c), which use only imperfect CSI, have the same diversity order as the coherent detector, i.e., the difference to the coherent detector does not grow when the SNR increases. This is well known [7], [16]. In case (i), however, the detection loss grows with the SNR and the use of the optimal metric in (19) cannot change this fundamental fact.

C. An Example With Iterative Decoding

The goal of the final example is to illustrate the capability of the proposed soft demodulator to take soft input. Towards this end, we construct a "turbo"-type receiver by iterating between the proposed soft demodulator and the channel decoder. We iterated up to three times between the demodulator and the decoder. The demodulator was implemented by approximating (21) using the approach developed in Section IV.

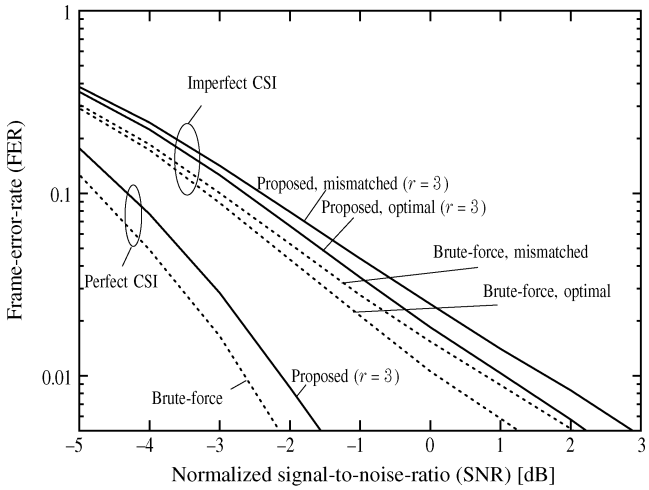

Fig. 3. Same as Fig. 1 but with imperfect channel knowledge at the receiver. Here the variance of the channel estimation error is constant with respect to the SNR. In the figure, "optimal" refers to the soft metric (19) and "mismatched" refers to using (5) with $\boldsymbol{H}$ replaced by its estimate $\hat{\boldsymbol{H}}$. Also, "brute-force" refers to exact marginalization (as in (5)) whereas "proposed" refers to our scheme presented in Section IV.

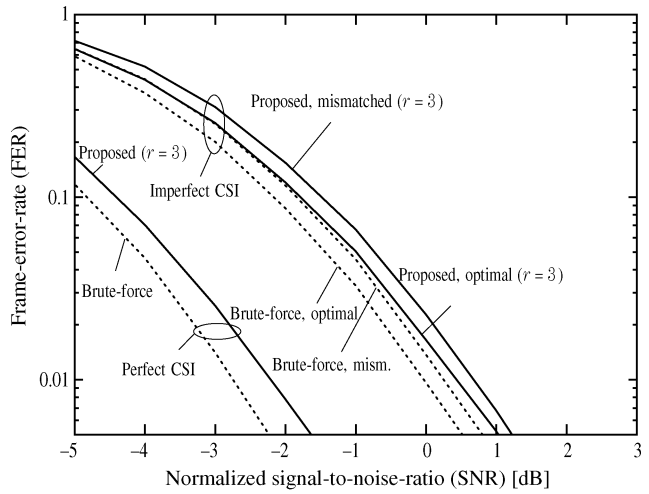

Fig. 4. Same as Fig. 3 but here the variance of the channel estimation error is inversely proportional to the SNR.

For channel coding we use in this example a rate-1/2, regular (3,6) LDPC code with block length 1000 bits.[4] This code can be efficiently decoded with belief propagation. Additionally, by looking at whether the decoder converged to a valid codeword one can say with relatively high certainty whether the correct codeword was found or not. We ran the belief propagation until a valid codeword was encountered, but not more than 25 iterations. In order to save computational efforts, we also terminated the outer iteration (between the decoder and the demodulator) whenever the decoder converged to a valid codeword.

Fig. 5 shows the results for a 4×4 complex MIMO system with QPSK modulation [that is, $m = 1$ and $n_t = n_r = 8$ in (1)]. We used $r = 3$, which was found to offer a good tradeoff between complexity and performance in Section VI-A. There is a clear gain by iterating between the demodulator and the

[4]The parity check matrix was randomly constructed, but some small-loop removal was applied. The resulting graph had girth 8.

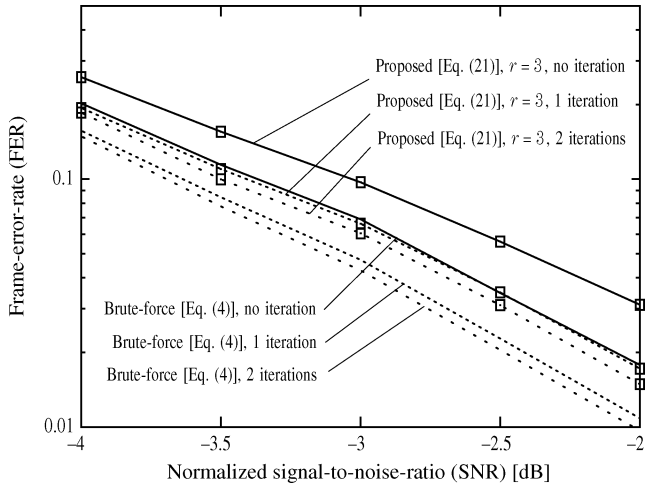

Fig. 5. Performance comparison for a 4×4 complex MIMO system with QPSK modulation ($n_t = n_r = 8$, $m = 1$) on a slowly fading Rayleigh channel, using a rate-1/2 LDPC code decoded with belief propagation. Each information block had 1000 bits and spanned one channel realization. Information between the demodulator and the channel decoder was iterated up to three times. (Note the different scale used in this figure.)

decoder. This was expected and it is consistent with previous results on iterative detection/decoding for MIMO (using other types of demodulators, including the exact MAP in (4) [8]). More importantly, the iteration gain is about as large for the exact MAP demodulator (4) as it is for our approximate demodulator (21). (This is true for other values of r too, but we omit the corresponding curves in Fig. 5 to keep the plot readable.) This suggests that there is nothing fundamental in the proposed approximations or algorithm structure that limits the usefulness of soft-input, and that the approach in Section IV is technically sound also when the *a priori* bit-probabilities are biased away from 1/2. It remains to determine whether the soft-input extension in Section V-B can be extended to higher-order constellations ($m > 1$), an open problem at this point.

VII. Discussion

We have proposed a new approach to soft detection for MIMO systems, or more generally for linear channels with crosstalk. The scheme inherits many of its advantages from the (hard-decision only) "fixed-complexity sphere decoding" algorithm [13] which has served as an inspiration source for our work. In particular, the main merits of our proposed scheme are: (1) it has fixed (nonrandom) complexity, (2) it provides very good performance at low complexity, and (3) in contrast to competing methods such as those based on sphere-decoding, many of its steps can be executed in parallel and it is therefore suitable for implementation on parallel hardware architectures. Additionally, it offers a simple way of trading performance against complexity by choosing the parameter r.

We have shown how the optimal MIMO detector can be extended to make optimum use of imperfect channel knowledge at the receiver, and that the resulting receiver can be implemented using our proposed receiver structure as well. Additionally, we have discussed how our soft detector can be modified to make use of soft-input, for the case of binary signalling per real dimension.

Our detector has been especially designed with "slow fading" in mind. In particular, it relies heavily on preprocessing of the channel matrix which must be performed once per received frame (i.e., once per channel realization). In fast fading, this preprocessing may become a significant part of the decoder complexity. However, in fast fading linear detectors like ZF generally work well (at least for systems with outer channel coding) and the motivation for using more sophisticated structures is smaller. In conclusion, we believe that our method shall be considered a strong competitor for use in practical MIMO systems operating on stationary or slowly fading channels.

APPENDIX

Proof of Proposition 1: Let $\hat{\boldsymbol{x}}_{\mathrm{ML}}$ and $\hat{\boldsymbol{x}}_{\mathrm{ZF}}$ be given as in the Proposition. By definition,

$$\|\boldsymbol{b}-\boldsymbol{A}\hat{\boldsymbol{x}}_{\mathrm{ML}}\| \le \|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}\|$$

for *any* $\boldsymbol{x} \in \mathcal{X}$. This proves the lower bound in (18).

To show the upper bound note that

$$\begin{aligned}
\|\boldsymbol{b}-\boldsymbol{A}\hat{\boldsymbol{x}}_{\mathrm{ZF}}\| &= \left\|\boldsymbol{A}(\boldsymbol{A}^{-1}\boldsymbol{b}-\hat{\boldsymbol{x}}_{\mathrm{ZF}})\right\| \\
&\overset{(a)}{\le} \sigma_{\max}(\boldsymbol{A})\|\boldsymbol{A}^{-1}\boldsymbol{b}-\hat{\boldsymbol{x}}_{\mathrm{ZF}}\| \\
&\overset{(b)}{\le} \sigma_{\max}(\boldsymbol{A})\|\boldsymbol{A}^{-1}\boldsymbol{b}-\hat{\boldsymbol{x}}_{\mathrm{ML}}\| \\
&= \sigma_{\max}(\boldsymbol{A})\left\|\boldsymbol{A}^{-1}(\boldsymbol{b}-\boldsymbol{A}\hat{\boldsymbol{x}}_{\mathrm{ML}})\right\| \\
&\overset{(c)}{\le} \sigma_{\max}(\boldsymbol{A})\sigma_{\max}(\boldsymbol{A}^{-1})\|\boldsymbol{b}-\boldsymbol{A}\hat{\boldsymbol{x}}_{\mathrm{ML}}\| \\
&\overset{(d)}{=} \frac{\sigma_{\max}(\boldsymbol{A})}{\sigma_{\min}(\boldsymbol{A})}\|\boldsymbol{b}-\boldsymbol{A}\hat{\boldsymbol{x}}_{\mathrm{ML}}\| \\
&\overset{(e)}{=} \kappa(\boldsymbol{A})\|\boldsymbol{b}-\boldsymbol{A}\hat{\boldsymbol{x}}_{\mathrm{ML}}\|.
\end{aligned}$$

Here the inequalities in (a) and (c) follow from Theorem 7.3.10 of [15]. The equality in (b) follows by the definition of $\hat{\boldsymbol{x}}_{\mathrm{ZF}}$. The equality in (d) follows since if $\boldsymbol{X} = \boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{W}^T$ denotes the singular value decomposition of a square matrix $\boldsymbol{X}$, then $\boldsymbol{X}^{-1} = \boldsymbol{W}\boldsymbol{\Sigma}^{-1}\boldsymbol{V}^T$. The final equality (e) is the definition of the condition number (also known as spectral norm).

REFERENCES

[1] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[2] S. Verdú, *Multiuser Detection*. Cambridge, U.K.: Cambridge Univ. Press, 1998.

[3] S. Verdú, "Computational complexity of multiuser detection," *Algorithmica*, vol. 4, pp. 303–312, 1989.

[4] M. O. Damen, H. E. Gamal, and G. Caire, "On maximum-likelihood detection and the search for the closest lattice point," *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2389–2401, Oct. 2003.

[5] J. Jaldén and B. Ottersten, "On the complexity of sphere decoding in digital communications," *IEEE Trans. Signal Process.*, vol. 53, no. 4, pp. 1474–1484, Apr. 2005.

[6] E. G. Larsson and J. Jaldén, "Soft MIMO detection at fixed complexity," in *Proc. IEEE GLOBECOM*, Nov. 2007.

[7] E. G. Larsson and P. Stoica, *Space-Time Block Coding for Wireless Communications*. Cambridge, U.K.: Cambridge Univ. Press, May 2003.

[8] A. Stefanov and T. M. Duman, "Turbo-coded modulation for systems with transmit and receive antenna diversity over block fading channels: System model, decoding approaches, and practical considerations," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 5, pp. 958–968, May 2001.

[9] T. K. Moon, *Error Correction Coding: Mathematical Methods and Algorithms*. New York: Wiley, 2005.

[10] B. Steingrimsson, Z.-Q. Luo, and K. Wong, "Soft Quasi-maximum-likelihood detection for multiple-antenna wireless channels," *IEEE Trans Signal Process.*, vol. 51, no. 11, pp. 2710–2719, Nov. 2003.

[11] B. M. Hochwald and S. Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Trans. Commun.*, vol. 51, no. 3, pp. 389–399, Mar. 2003.

[12] M. Siti and M. P. Fitz, "A novel soft-output layered orthogonal lattice detector for multiple antenna communications," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2006, pp. 1686–1691.

[13] L. G. Barbero and J. S. Thompson, "A fixed-complexity MIMO detector based on the complex sphere decoder," presented at the IEEE Signal Process. Advanced Wireless Commun (SPAWC), Cannes, France, Jul. 2006.

[14] D. Chase, "Class of algorithms for decoding block codes with channel measurement information," *IEEE Trans. Inf. Theory*, vol. 18, no. 1, pp. 170–182, Jan. 1972.

[15] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1985.

[16] G. Taricco and E. Biglieri, "Space-time decoding with imperfect channel estimation," *IEEE Trans. Wireless Commun.*, vol. 4, pp. 1874–1888, Jul. 2005.

[17] P. Frenger, "Turbo decoding for wireless systems with imperfect channel estimation," *IEEE Trans. Commun.*, vol. 48, pp. 1437–1440, Sep. 2000.

[18] G. Taricco and G. Coluccia, "Optimum receiver design for correlated Rician fading MIMO channels with pilot-aided detection," *IEEE J. Sel. Areas Commun.*, vol. 25, pp. 1311–1321, Sep. 2007.

Erik G. Larsson received the Ph.D. degree from Uppsala University, Uppsala, Sweden, in 2002.

Currently, he is Professor and Head of the Division for Communication Systems, Department of Electrical Engineering (ISY), Linköping University (LiU), Linköping, Sweden. He joined LiU in September 2007. He has previously been Associate Professor (Docent) at the Royal Institute of Technology (KTH) in Stockholm, Sweden, and Assistant Professor at the University of Florida and the George Washington University. His main professional interests are within the areas of wireless communications and signal processing. He has published approximately 50 papers on these topics, and is a coauthor of the textbook *Space-Time Block Coding for Wireless Communications* (Cambridge, U.K.: Cambridge Univ. Press, 2003). He holds ten patents on wireless technology.

Prof. Larsson is an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING and the IEEE SIGNAL PROCESSING LETTERS and a member of the IEEE Signal Processing Society SAM and SPCOM technical committees.

Joakim Jaldén was born in Gävle, Sweden, on May 16, 1976. He received the M.Sc. and Ph.D. degrees in electrical engineering from the Royal Institute of Technology (KTH), Stockholm, Sweden, in 2006 and 2002, respectively. Between September 2000 and May 2002, he studied at Stanford University, CA, where he also conducted research for his M.Sc. degree thesis. His Ph.D. position at KTH was funded by a grant, "Honor Graduate Student Position," awarded by the dean office.

Currently, he is holds a Postdoctoral Research position within the Institute of Communications and Radio-Frequency Engineering, Vienna University of Technology, Vienna, Austria.

Dr. Jaldén was awarded the IEEE Signal Processing (SP) Society's 2006 Young Author Best Paper Award for his work on MIMO communications, and won first prize in the Student Paper Contest at the 2007 International Conference on Acoustics, Speech and Signal Processing (ICASSP).