

Optimized Configurable Architectures for Scalable Soft-Input Soft-Output MIMO Detectors With 256-QAM

Mohammad M. Mansour, *Senior Member, IEEE*, and Louay M. A. Jalloul, *Senior Member, IEEE*

Abstract—This paper presents an optimized low-complexity and high-throughput MIMO signal detector core for detecting spatially multiplexed data streams. The core architecture supports various layer configurations up to 4, while achieving near-optimal performance, and configurable modulation constellations up to 256-QAM on each layer. The core is capable of operating as a soft-input soft-output log-likelihood ratio (LLR) MIMO detector which can be used in the context of iterative detection and decoding. High area-efficiency is achieved via algorithmic and architectural optimizations performed at two levels. First, distance computations and slicing operations for an optimal 2-layer maximum *a posteriori* MIMO detector are optimized to eliminate use of multipliers and reduce the overhead of slicing in the presence of soft-input LLRs. We show that distances can be easily computed using elementary addition operations, while optimal slicing is done via efficient comparisons with soft decision boundaries, resulting in a simple feed-forward pipelined architecture. Second, to support more layers, an efficient channel decomposition scheme is presented that reduces the detection of multiple layers into multiple 2-layer detection subproblems, which map onto the 2-layer core with a slight modification using a distance accumulation stage and a post-LLR processing stage. Various architectures are accordingly developed to achieve a desired detection throughput and run-time reconfigurability by time-multiplexing of one or more component cores. The proposed core is applied also to design an optimal multiuser MIMO detector for LTE. The core occupies an area of 1.58 MGE and achieves a throughput of 733 Mbps for 256-QAM when synthesized in 90-nm CMOS.

Index Terms—Maximum Likelihood detection, MIMO systems, 256-QAM, Reconfigurable MIMO detector, Sphere decoder.

I. INTRODUCTION

MULTIPLE-INPUT MULTIPLE-OUTPUT (MIMO) systems have become mainstream technology for achieving high spectral efficiencies in wireless communications standards such as IEEE 802.11ac [1] and the 3GPP Long-Term Evolution (LTE) [2]. Detection of spatially-multiplexed MIMO

streams plays a key role in receiver design, both in terms of performance and complexity, and has remained to be an active area of research [3]–[7]. The focus has been on developing area-/energy-efficient VLSI implementations of MIMO detectors that are capable of achieving close to optimal performance.

A plethora of MIMO detectors have appeared in the literature on this subject, offering various performance-complexity trade-offs. Suboptimal zero-forcing (ZF) and minimum mean-squared error (MMSE) detectors [5], as well as nonlinear parallel and successive interference cancellation schemes [8]–[11], require relatively low complexity but sacrifice performance. On the other hand, tree-search or list-based detectors require substantially higher complexity but can offer (near-)ML performance, such as the well-known sphere decoding algorithm [12]–[19]. Other tree-search schemes, such as the K-Best algorithm [20]–[26], address the non-deterministic throughput aspects of sphere decoders. Practical implementation aspects have been investigated in [18], [23], [25]–[39].

Subspace detection based on channel decomposition offers a good compromise between performance and complexity (e.g., see [40]–[43]). In these schemes, the effective MIMO channel matrix is decomposed into parallel subchannels that can be used to detect subsets of streams in parallel. By allowing subspaces to overlap, additional diversity can be gathered by putting a low reliable data stream into several detection sets. The LORD algorithm proposed in [44], [45] can be viewed as a special class of subspace MIMO detectors. It achieves ML performance (in the max-log-MAP [46] sense) on 2 transmit antennas, but its performance degrades when the number of antennas increases. In [47], the LORD algorithm was generalized to 4-transmit antennas by using matrix inversion to decompose the channel into single streams.

Support for ever increasing data rates has come through an increase in the number of supported spatial streams, or through the use of more bandwidth via carrier aggregation [48]. LTE-Advanced uses up to 8 spatial streams, or the aggregation of five component carriers for a bandwidth of 100 MHz, which lead to staggering speeds of over 1 Gbps. While the receiver complexity to detect 8 spatial layers remains to be very challenging especially for dense constellations, the use of carrier aggregation with distinct or separate physical layers and convergence at higher layers seems more tractable. Since each physical layer of a component carrier is required to support 2 or 4 spatial layers, the need for the hardware optimization of these MIMO detector cores becomes paramount, especially if near-ML performance is desired, higher-order modulations such as 256-QAM are to be supported, and high-throughput processing is a must.

Manuscript received November 08, 2014; revised March 25, 2015; accepted June 06, 2015. Date of publication June 16, 2015; date of current version August 19, 2015. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Joseph Cavallaro.

M. M. Mansour is with the Department of Electrical and Computer Engineering at the American University of Beirut, Beirut 1107 2020, Lebanon (e-mail: mmansour@ieee.org).

L. Jalloul is with Qualcomm Inc., San Jose, CA 95134 USA (e-mail: jalloul@ieee.org).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TSP.2015.2446441

Contributions: We propose in this work an optimized and configurable 2×2 soft-input soft-output maximum a posteriori (MAP) MIMO detector, and use it as a basic building block for constructing high-throughput detectors for higher-order layers. The key features and advantages of the proposed detector core are: 1) scalability in supporting multiple layers, 2) flexibility in accommodating multiple layer-configurations and detection of subsets of layers, 3) configurability of supported constellations per layer, 4) support for soft-input log-likelihood ratios (LLRs) from channel decoder, 5) near-ML performance, and 6) reduced-complexity and high-throughput operation. We develop extensive optimizations at both the algorithmic and architectural levels targeted for a 2×2 soft-input soft-output MAP MIMO detector, as well as its extension to support more spatial layers. In particular, optimizations of distance computations (to eliminate multipliers and simplify slicing) are shown to result in substantial reduction in computational complexity when supporting constellations up to 256-QAM. Furthermore, the complexity of a 1D slicer is shown to play a key role in the overall complexity of the detector, when soft-input LLRs are supported. To this end, an efficient slicing scheme based on *soft* decision boundaries is presented. Moreover, a low-complexity scheme that decomposes a MIMO channel into multiple subsets of decoupled streams is proposed. It is shown that decoupled streams can be detected efficiently and in parallel using the optimized 2×2 core. Moreover, the 2×2 core is applied in the context of multi-user (MU-MIMO) for joint modulation classification and data detection. The core has been implemented on an FPGA, and synthesized as well using a generic 90 nm ASIC CMOS library.

The rest of the paper is organized as follows. After introducing the system model in Section II, Section III presents the optimizations targeted for a 2-layer MAP MIMO detector in terms of distance computations and slicing. Key equations for distances and soft decision boundaries are derived assuming both zero and non-zero input LLRs. Section IV proposes a matrix decomposition scheme to support detection of more spatial streams. We show that the key distance equations scale in a straightforward fashion from the 2-layer case, where only a new distance-accumulation and a post-LLR processing phases are needed. In Section V, single and multi-core detector architectures are developed. The core is applied in Section VI part of MU-MIMO detection for constellation estimation and data detection. Synthesis and simulations results are reported in Section VII. Finally, Section VIII ends with concluding remarks.

II. SYSTEM MODEL

Consider a MIMO system with N transmit and $M \geq N$ receive antennas. The equivalent complex baseband input-output system relation can be modeled as $\tilde{\mathbf{y}} = \mathbf{H}\mathbf{x} + \mathbf{n}$, where $\tilde{\mathbf{y}} \in \mathcal{C}^{M \times 1}$ is the received complex signal vector, $\mathbf{H} \in \mathcal{C}^{M \times N}$ is the complex channel matrix, $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_N]^T \in \mathcal{X} = \mathcal{X}_1 \times \dots \times \mathcal{X}_N$ is the $N \times 1$ transmitted complex symbol vector, and $\mathbf{n} \in \mathcal{C}^{M \times 1}$ is a zero-mean complex Gaussian circularly symmetric random noise vector with covariance $\sigma^2 \mathbf{I}_M$. Each symbol x_n belongs to a complex constellation \mathcal{X}_n of size $Q_n = 2^{q_n}$, and is associated via the map $\mathbf{b}(\cdot)$ with a coded bit-interleaved vector $\mathbf{b}(x_n) = \mathbf{b}_n = [b_{n,1} \ b_{n,2} \ \dots \ b_{n,q_n}]^T$ of length q_n over the set $\{-1, +1\}$, where binary 0 maps to $+1$.

Authorized licensed use limited to: American University of Beirut. Downloaded on April 17, 2024 at 07:33:46 UTC from IEEE Xplore. Restrictions apply.

Let $|\mathcal{X}| = Q = 2^q$, and denote the binary vector associated with the overall symbol vector \mathbf{x} as $\mathbf{b}(\mathbf{x}) = [\mathbf{b}_1; \dots; \mathbf{b}_N] = [b_{n,j}]$, for $n = 1, \dots, N$, and $j = 1, \dots, q_n$. Motivated by recent standards, we assume rectangular QAM constellations, where $\mathcal{X}_n = \mathcal{P}_n \times \mathcal{P}_n$, and \mathcal{P}_n is a 1D P_n -PAM constellation with $P_n = \sqrt{Q_n}$.

We assume \mathbf{H} is known to the receiver, has full column rank and is decomposed as $\mathbf{H} = \mathbf{Q}\mathbf{L}$, where $\mathbf{Q} \in \mathcal{C}^{M \times N}$ is a unitary matrix and $\mathbf{L} \in \mathcal{C}^{N \times N}$ is a lower triangular matrix (LTM) with positive and real diagonal elements. Since \mathbf{Q} is unitary, it preserves Euclidean norm as well as noise statistics. Hence we use the transformed relation $\mathbf{y} \triangleq \mathbf{Q}^* \tilde{\mathbf{y}} = \mathbf{L}\mathbf{x} + \mathbf{Q}^* \mathbf{n} \in \mathcal{C}^{N \times 1}$ to model the MIMO system.

A hard-decision (HD) maximum a posteriori (MAP) MIMO detector achieves log-max [46] optimal performance by finding the symbol vector \mathbf{x} in \mathcal{X} that is closest to the received vector \mathbf{y} under the unscaled “distance” metric [16]:

$$d(\mathbf{x}) \triangleq \|\mathbf{y} - \mathbf{L}\mathbf{x}\|^2 - \mathbf{b}^T(\mathbf{x})\boldsymbol{\lambda}, \quad (1)$$

where $\boldsymbol{\lambda} = [\lambda_1; \dots; \lambda_N] = [\lambda_{n,j}]$ is a column vector of *a priori* LLR values $\lambda_{n,j} \in \mathcal{R}$ associated with the bits in $\mathbf{b}(\mathbf{x})$, assuming these bits are statistically independent:

$$\lambda_{n,j} = \frac{1}{\sigma^2} \ln \frac{\text{Prob}(b_{n,j} = +1)}{\text{Prob}(b_{n,j} = -1)}.$$

The subvector $\boldsymbol{\lambda}_n = [\lambda_{n,1}, \dots, \lambda_{n,q_n}]^T$ is associated with the bits $\mathbf{b}(x_n)$ of the n th symbol x_n . The hard-decision MAP solution of the MIMO detection problem is given by¹

$$d^{\text{MAP}} = \min_{\mathbf{x} \in \mathcal{X}} d(\mathbf{x}) \text{ and } \mathbf{x}^{\text{MAP}} = \arg \min_{\mathbf{x} \in \mathcal{X}} d(\mathbf{x}). \quad (2)$$

For joint iterative MIMO detection and decoding however, soft-input soft-output MIMO detectors are required. A log-max optimal soft-input soft-output MAP MIMO detector computes $2q$ other minimum distance metrics as follows:

$$\Lambda_{n,j}^{\text{MAP}} = \min_{\mathbf{x} \in \mathcal{X}_{n,j}^{(+1)}} d(\mathbf{x}) - \min_{\mathbf{x} \in \mathcal{X}_{n,j}^{(-1)}} d(\mathbf{x}), \quad (3)$$

for $n = 1, \dots, N$ and $j = 1, \dots, q_n$, where $\mathcal{X}_{n,j}^{(+1)} = \{\mathbf{x} \in \mathcal{X} : b_{n,j} = +1\}$ and $\mathcal{X}_{n,j}^{(-1)} = \{\mathbf{x} \in \mathcal{X} : b_{n,j} = -1\}$ are the subsets of symbol vectors in \mathcal{X} that have their corresponding j th bit in the n th symbol $+1$ and -1 , respectively.

III. OPTIMIZED MIMO MAP DETECTION FOR 2 LAYERS

Finding the MAP solutions in (2) and (3) require computing $\prod_{n=1}^N Q_n$ distance metrics. When $N = 2$, a simplification [44] can be applied to reduce the number of computations from $Q_1 \cdot Q_2$ to $Q_1 + Q_2$. Triangularizing the channel matrix as $\mathbf{H} = \mathbf{Q}\mathbf{L}$ with \mathbf{Q} being unitary, we obtain:

$$\mathbf{y} - \mathbf{L}\mathbf{x} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} - \begin{bmatrix} \alpha & 0 \\ \gamma & \beta \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix},$$

where $\mathbf{y} = \mathbf{Q}^* \tilde{\mathbf{y}}$, with $\alpha, \beta \in \mathcal{R}^+$ and $\gamma \in \mathcal{C}$. Then (1) becomes

$$d(\mathbf{x}) = f_1(x_1) + f_2(x_2 | x_1), \text{ where} \\ f_1(x_1) = |y_1 - \alpha x_1|^2 - \mathbf{b}^T(x_1)\boldsymbol{\lambda}_1, \text{ and} \\ f_2(x_2 | x_1) = |y_2 - \gamma x_1 - \beta x_2|^2 - \mathbf{b}^T(x_2)\boldsymbol{\lambda}_2. \quad (4)$$

¹The quantities d^{MAP} in (2) and $\Lambda_{n,j}^{\text{MAP}}$ in (3) need to be scaled by $\sigma^2/2$.

The minimum distance in (2) can then be computed as

$$\min_{\mathbf{x} \in \mathcal{X}} d(\mathbf{x}) = \min_{\substack{x_1 \in \mathcal{X}_1 \\ x_2 \in \mathcal{X}_2}} \{f_1(x_1) + f_2(x_2 | x_1)\} \quad (5)$$

$$\begin{aligned} &= \min_{x_1 \in \mathcal{X}_1} \{f_1(x_1) + \min_{x_2 \in \mathcal{X}_2} f_2(x_2 | x_1)\} \\ &= \min_{x_1 \in \mathcal{X}_1} \{f_1(x_1) + f_2(\hat{x}_2(x_1) | x_1)\} \\ &= \min_{x_1 \in \mathcal{X}_1} d(x_1, \hat{x}_2(x_1)) \end{aligned} \quad (6)$$

where

$$\hat{x}_2(x_1) = \arg \min_{x_2 \in \mathcal{X}_2} \{|y_2 - \gamma x_1 - \beta x_2|^2 - \mathbf{b}^T(x_2)\boldsymbol{\lambda}_2\}. \quad (7)$$

Denote the set of sliced symbol vectors for all x_1 in (6) by

$$\mathcal{O}_1 = \{[x_1 \hat{x}_2(x_1)]^T : x_1 \in \mathcal{X}_1\}. \quad (8)$$

The bit LLRs of symbol x_1 , for $j = 1, \dots, q_1$, are given by

$$\Lambda_{1,j}^{\text{MAP}} = \min_{x_1 \in \mathcal{X}_{1,j}^{(+1)}} d(x_1, \hat{x}_2(x_1)) - \min_{x_1 \in \mathcal{X}_{1,j}^{(-1)}} d(x_1, \hat{x}_2(x_1)). \quad (9)$$

To obtain the bit LLRs of x_2 however, we triangularize \mathbf{H} as $\mathbf{Q}'\mathbf{L}'$ so that a zero appears in the upper left corner:

$$\mathbf{y}' - \mathbf{L}'\mathbf{x} = \begin{bmatrix} y'_1 \\ y'_2 \end{bmatrix} - \begin{bmatrix} 0 & \alpha' \\ \beta' & \gamma' \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix},$$

where $\mathbf{y}' = \mathbf{Q}'^* \tilde{\mathbf{y}}$; $\alpha', \beta' \in \mathcal{R}^+$ and $\gamma' \in \mathcal{C}$. Then (1) becomes

$$\begin{aligned} d(\mathbf{x}) &= f'_2(x_2) + f'_1(x_1 | x_2), \text{ where} \\ f'_2(x_2) &= |y'_2 - \alpha' x_2|^2 - \mathbf{b}^T(x_2)\boldsymbol{\lambda}_2, \text{ and} \\ f'_1(x_1 | x_2) &= |y'_1 - \beta' x_2 - \gamma' x_1|^2 - \mathbf{b}^T(x_1)\boldsymbol{\lambda}_1, \end{aligned}$$

and the minimum distance in (2) can be computed as

$$\begin{aligned} \min_{\mathbf{x} \in \mathcal{X}} d(\mathbf{x}) &= \min_{x_2 \in \mathcal{X}_2} \{f'_2(x_2) + \min_{x_1 \in \mathcal{X}_1} f'_1(x_1 | x_2)\} \\ &= \min_{x_2 \in \mathcal{X}_2} \{f'_2(x_2) + f'_1(\hat{x}_1 | x_2)\} \\ &= \min_{x_2 \in \mathcal{X}_2} d(\hat{x}_1(x_2), x_2), \end{aligned} \quad (10)$$

where $\hat{x}_1(x_2) = \arg \min_{x_1 \in \mathcal{X}_1} f'_1(x_1 | x_2)$. Denote the set of sliced symbol vectors for all x_2 in (10) by

$$\mathcal{O}_2 = \{[\hat{x}_1(x_2) \ x_2]^T : x_2 \in \mathcal{X}_2\}.$$

The bit LLRs of symbol x_2 , for $j = 1, \dots, q_2$, are given by

$$\Lambda_{2,j}^{\text{MAP}} = \min_{x_2 \in \mathcal{X}_{2,j}^{(+1)}} d(\hat{x}_1(x_2), x_2) - \min_{x_2 \in \mathcal{X}_{2,j}^{(-1)}} d(\hat{x}_1(x_2), x_2). \quad (11)$$

Since \mathbf{Q} and \mathbf{Q}' are unitary, the MAP solutions in (6) and (10) are identical. To find the hard-decision (HD)-MAP solution, only 1-sided QLD is needed on either layer 1 or 2. If $Q_1 \leq Q_2$, a list of Q_1 distances $\mathcal{D}_1 = \{d(x_1, \hat{x}_2(x_1)) : x_1 \in \mathcal{X}_1\}$ is generated by enumerating all symbols $x_1 \in \mathcal{X}_1$ and the minimum is selected. If $Q_2 < Q_1$, a list of Q_2 distances $\mathcal{D}_2 = \{d(\hat{x}_1(x_2), x_2) : x_2 \in \mathcal{X}_2\}$ is generated and the minimum is selected. However, to generate soft LLRs, 2-sided QLDs are needed, and both lists of distances must be generated to select the appropriate minima according to (9) and (11).

A. Distance Metric Optimizations

For efficient distance computations, we separate the real and imaginary parts of all complex variables, and exploit the fact that the real and imaginary parts of each QAM symbol are mapped independently into 1D PAM symbols, i.e., some bits are used only for mapping of the real part and some only for the imaginary part (see Fig. 1). Note that this mapping is used in, e.g., the IEEE 802.11ac [1] and LTE [2] standards. Under this assumption, we can split the bias term $\mathbf{b}^T(x_n)\boldsymbol{\lambda}_n$ into a part $\mathbf{b}_{nR}^T\boldsymbol{\lambda}_{nR} \triangleq \mathbf{b}^T(x_{nR})\boldsymbol{\lambda}_{nR}$ associated with the bits of the real part of the QAM symbol, and a part $\mathbf{b}_{nI}^T\boldsymbol{\lambda}_{nI} \triangleq \mathbf{b}^T(x_{nI})\boldsymbol{\lambda}_{nI}$ associated with the bits of the imaginary part. Let $\gamma = \gamma_R + j\gamma_I$, $x_n = x_{nR} + jx_{nI}$, $y_n = y_{nR} + jy_{nI}$ for $n = 1, 2$. Then the distance in (4) becomes

$$d(\mathbf{x}) = f_{1R}(x_{1R}) + f_{1I}(x_{1I}) + f_{2R}(x_{2R}|x_1) + f_{2I}(x_{2I}|x_1), \quad (12)$$

where the terms on the righthand side are given by

$$\begin{aligned} f_{1R}(x_{1R}) &= (y_{1R} - \alpha x_{1R})^2 - \mathbf{b}_{1R}^T\boldsymbol{\lambda}_{1R} \\ f_{1I}(x_{1I}) &= (y_{1I} - \alpha x_{1I})^2 - \mathbf{b}_{1I}^T\boldsymbol{\lambda}_{1I} \\ f_{2R}(x_{2R}|x_1) &= (y_{2R} - \gamma_R x_{1R} + \gamma_I x_{1I} - \beta x_{2R})^2 - \mathbf{b}_{2R}^T\boldsymbol{\lambda}_{2R} \\ f_{2I}(x_{2I}|x_1) &= (y_{2I} - \gamma_R x_{1I} - \gamma_I x_{1R} - \beta x_{2I})^2 - \mathbf{b}_{2I}^T\boldsymbol{\lambda}_{2I}. \end{aligned}$$

Expanding (12), minimizing with respect to x_{2R} and x_{2I} , and removing irrelevant terms, we obtain the following key equation:

$$\begin{aligned} \bar{d}(\mathbf{x}) &= \bar{f}_{1R}(x_{1R}) + \bar{f}_{1I}(x_{1I}) \\ &\quad + \min_{x_{2R} \in \mathcal{P}_2} \bar{f}_{2R}(x_{2R} | x_1) + \min_{x_{2I} \in \mathcal{P}_2} \bar{f}_{2I}(x_{2I} | x_1), \end{aligned} \quad (13)$$

where \mathcal{P}_2 is the 1D PAM constellation in \mathcal{X}_2 of layer 2, and

$$\bar{f}_{1R}(x_{1R}) = Ax_{1R}^2 + Cx_{1R} - \mathbf{b}_{1R}^T\boldsymbol{\lambda}_{1R} \quad (14)$$

$$\bar{f}_{1I}(x_{1I}) = Ax_{1I}^2 + Dx_{1I} - \mathbf{b}_{1I}^T\boldsymbol{\lambda}_{1I} \quad (15)$$

$$\begin{aligned} \bar{f}_{2R}(x_{2R} | x_1) &= (Ex_{1R} + Fx_{1I})x_{2R} \\ &\quad + (Bx_{2R}^2 + Gx_{2R} - \mathbf{b}_{2R}^T\boldsymbol{\lambda}_{2R}) \end{aligned} \quad (16)$$

$$\begin{aligned} \bar{f}_{2I}(x_{2I} | x_1) &= (Ex_{1I} - Fx_{1R})x_{2I} \\ &\quad + (Bx_{2I}^2 + Hx_{2I} - \mathbf{b}_{2I}^T\boldsymbol{\lambda}_{2I}). \end{aligned} \quad (17)$$

The constant coefficients in (14)–(17) are given by

$$A = \alpha^2 + |\gamma|^2, \quad B = \beta^2, \quad (18)$$

$$C = -2(\alpha y_{1R} + \gamma_R y_{2R} + \gamma_I y_{2I}),$$

$$D = -2(\alpha y_{1I} - \gamma_I y_{2R} + \gamma_R y_{2I}),$$

$$E = +2\beta\gamma_R, \quad F = -2\beta\gamma_I,$$

$$G = -2\beta y_{2R}, \quad H = -2\beta y_{2I}, \quad (19)$$

and can be precomputed off-line from \mathbf{H} and \mathbf{y} . The HD-MAP solution is obtained by populating all Q_1 distances in (13) and selecting the minimum. The same applies for the LLRs.

B. Slicing Assuming Zero Prior LLRs

Assuming the input LLRs $\boldsymbol{\lambda}$ are zero, the rightmost term in (1) vanishes and the MAP detection problem reduces to a least-squares integer ML problem. Then \hat{x}_2 in (7) can be obtained by

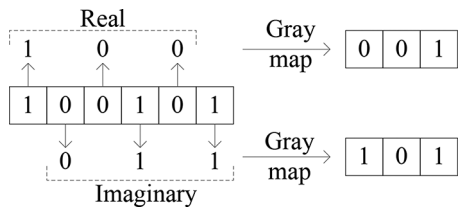


Fig. 1. Gray-coded mapping for 64-QAM in LTE [2].

slicing $(y_2 - \gamma x_1)/\beta \in \mathcal{C}$ to the nearest constellation point in \mathcal{X}_2 using the operator $[u]\mathcal{X}_n \triangleq \arg \min_{x \in \mathcal{X}_n} |u - x|$:

$$\hat{x}_2 = [(y_2 - \gamma x_1)/\beta]\mathcal{X}_2 \in \mathcal{X}_2. \quad (20)$$

By separating the real and imaginary parts as $\hat{x}_2 = \hat{x}_{2R} + j\hat{x}_{2I}$, the slicing operation in (20) splits into:

$$\hat{x}_{2R} = [(y_{2R} - \gamma_R x_{1R} + \gamma_I x_{1I})/\beta]\mathcal{P}_2 \in \mathcal{P}_2, \quad (21)$$

$$\hat{x}_{2I} = [(y_{2I} - \gamma_R x_{1I} - \gamma_I x_{1R})/\beta]\mathcal{P}_2 \in \mathcal{P}_2, \quad (22)$$

where $\mathcal{P}_2 = \{p_1, p_2, \dots, p_{P_2}\}$ is the P_2 -PAM constellation, and $P_2 = \sqrt{Q_2}$. The operations in (21)–(22) reduce to simple comparisons with the (deterministic) decision boundaries of \mathcal{P}_2 as follows. Let $z_2 = y_2 - \gamma x_1 = z_{2R} + jz_{2I}$ where

$$\begin{aligned} z_{2R} &= y_{2R} - \gamma_R x_{1R} + \gamma_I x_{1I}, \\ z_{2I} &= y_{2I} - \gamma_R x_{1I} - \gamma_I x_{1R}. \end{aligned} \quad (23)$$

Assume the constellation points are ordered such that $p_i < p_k$ if $i < k$. Then \hat{x}_{2R} maps to the point p_i that satisfies

$$\beta \frac{p_{i-1} + p_i}{2} \leq z_{2R} < \beta \frac{p_i + p_{i+1}}{2} \quad (24)$$

for $i = 1, \dots, P_2$, where $p_0 = -\infty$ and $p_{P_2+1} = +\infty$. Similarly for \hat{x}_{2I} . Hence the actual distances $f_2(x_2|x_1)$ themselves need not be computed for all x_2 and a given x_1 in order to find the symbol x_2 that minimizes $f_2(x_2|x_1)$ in (7). Therefore, (6) requires only $|\mathcal{X}_1| = Q_1$ distance computations. By the same argument, (10) requires only $|\mathcal{X}_2| = Q_2$ distance computations.

C. Slicing Assuming Non-Zero Prior LLRs

When the prior terms are included in the distance computations, slicing cannot be directly applied in (7) since the decision boundaries now depend on the bias term $\mathbf{b}^T(x_2)\boldsymbol{\lambda}_2$. We develop next an optimal scheme that enables efficient slicing similar to (24) based on [49]. In [50], a scheme that computes suboptimal slicing boundaries was presented. Compared to our approach, [50] incurs a performance loss with equivalent complexity.

The real part of \hat{x}_2 in (7) is given by

$$\hat{x}_{2R} = \arg \min_{x_{2R} \in \mathcal{P}_2} \{(z_{2R} - \beta x_{2R})^2 - \mathbf{b}^T(x_{2R})\boldsymbol{\lambda}_{2R}\}.$$

To decide in favor of $p_i \in \mathcal{P}_2$, then $\forall k \neq i$, we must have

$$(z_{2R} - \beta p_i)^2 - \mathbf{b}^T(p_i)\boldsymbol{\lambda}_{2R} < (z_{2R} - \beta p_k)^2 - \mathbf{b}^T(p_k)\boldsymbol{\lambda}_{2R}. \quad (25)$$

This condition can be formulated in terms of *decision boundaries* $R(p_i, p_k) = R(p_k, p_i)$:

$$R(p_i, p_k) = B \cdot (p_i + p_k) - \frac{\mathbf{b}^T(p_i) - \mathbf{b}^T(p_k)}{p_i - p_k} \boldsymbol{\lambda}_{2R}, \quad \forall k \neq i, \quad (26)$$

between p_i and all other p_k 's in \mathcal{P}_2 . Assuming the points in \mathcal{P}_2 are ordered such that $p_i < p_k$ if $i < k$, then for p_1 to satisfy (25), we must have $2\beta z_{2R} < R(p_1, p_k)$ for all $p_k > p_1$. For p_{P_2} to satisfy (25), we must have $2\beta z_{2R} > R(p_{P_2}, p_k)$ for all $p_k < p_{P_2}$. For any other internal point p_i , $i \neq 1, i \neq P_2$, we must have $2\beta z_{2R} < R(p_i, p_k)$ for all $p_k > p_i$, and $2\beta z_{2R} > R(p_i, p_k)$ for all $p_k < p_i$. These conditions can be combined into a single condition for $i = 1, \dots, P_2$, as follows:

$$\max_{k=0, \dots, i-1} R(p_i, p_k) \leq 2\beta z_{2R} < \min_{k=i+1, \dots, P_2+1} R(p_i, p_k), \quad (27)$$

where $p_0 = -\infty$, $p_{P_2+1} = +\infty$, $\mathbf{b}(p_0) = \mathbf{b}(p_{P_2+1}) = \mathbf{0}_{1 \times q_2/2}$. Note that (26) and (27) reduce to (24) when $\boldsymbol{\lambda}_{2R} = \mathbf{0}_{1 \times q_2/2}$.

Substituting (23) for z_{2R} in (27), using the constants (19)–(19), and accounting for sign change, we obtain the following slicing condition that is suitable for hardware implementation:

$$\begin{aligned} \max_{k=i+1, \dots, P_2+1} R(p_i, p_k) - G &\leq E x_{1R} + F x_{1I} \\ &< \min_{k=0, \dots, i-1} R(p_i, p_k) - G. \end{aligned} \quad (28)$$

Note that in (28), the maximum on the lefthand side is now taken over all points $p_k \in \mathcal{P}$ that are *greater* than p_i as opposed to *smaller* than p_i as was done in (27) due to the change in sign. Similarly for the minimum on the righthand side in (28).

A similar analysis applied to compute $\hat{x}_{2I} = \min_{x_{2I} \in \mathcal{P}_2} f_{2I}(x_{2I}|x_1)$ leads to the decision regions $I(p_i, p_k)$:

$$I(p_i, p_k) = B \cdot (p_i + p_k) - \frac{\mathbf{b}^T(p_i) - \mathbf{b}^T(p_k)}{p_i - p_k} \boldsymbol{\lambda}_{2I}, \quad (29)$$

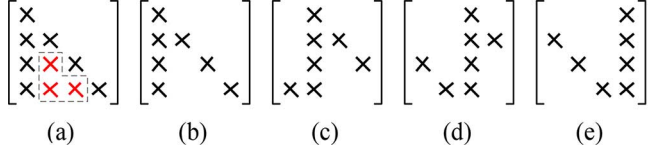
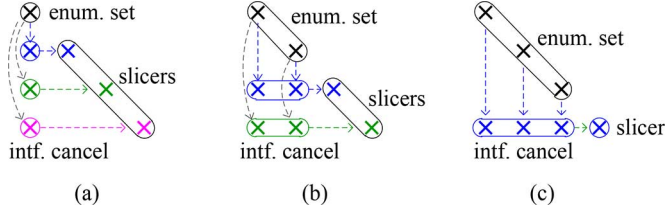
using now $\boldsymbol{\lambda}_{2I}$, and the associated slicing condition:

$$\begin{aligned} \max_{k=i+1, \dots, P_2+1} I(p_i, p_k) - H &\leq E x_{1I} - F x_{1R} \\ &< \min_{k=0, \dots, i-1} I(p_i, p_k) - H. \end{aligned} \quad (30)$$

Note that by construction of the decision boundaries in (27) (and their imaginary counterparts), the proposed approach is *optimal*. The approach in [50] however is suboptimal because it employs heuristics to compute simplified but suboptimal decision boundaries.

IV. EXTENSION TO HIGHER-ORDER LAYERS

The previous optimizations cannot be directly extended to $N \geq 3$ layers because the structure of the lower triangular matrix \mathbf{L} includes off-diagonal terms that prevent searching for the MAP solution by enumerating symbols in one layer and finding the minima through slicing individually on all other layers in parallel. More specifically, in Fig. 2(a), the presence of the demarked entries in the LTM implies that determining the MAP solution requires enumerating symbols on the first $N - 1$ layers and slicing only on the last layer, as is typically done

Fig. 2. 4×4 structures: (a) Full; (b)–(e) punctured structures for every layer.Fig. 3. (a) $(1,3 \times 1,3)$, (b) $(2,2 \times 2,2)$, (c) $(3,1 \times 3,1)$ punctured structures.

in tree-search detectors (e.g., [30]), and hence still requiring $O(\prod_n Q_n)$ complexity rather than $O(\sum_n Q_n)$.

One desirable structure of \mathbf{H} for a 4-layer MIMO system would be as shown in Fig. 2(b), in which the demarked entries are zeroed out. Here, by enumerating symbols on layer 1, the minimum distances and associated symbols on layers 2 to 4 can be searched for in parallel through slicing only on the corresponding layers, similar to the 2-layer system. This suffices to compute the LLRs associated with the bits on layer-1 symbol. A similar process is repeated by decomposing \mathbf{H} according to the structures shown in Figs. 2(c)–(e) [47] to compute the LLRs for bits associated with layers 2 to 4.

Other “punctured” structures are also possible for a 4×4 system as shown in Fig. 3. They differ in 1) the number of layers over which symbols are enumerated (*enumeration or detection set*), 2) the submatrix structure used to propagate these enumerated symbols and cancel their interference effect from the remaining layers (*interference cancellation set*), and 3) the number of layers in which the minimum distance and associated symbol can be obtained by slicing after interference cancellation (*slicer set*). Let U denote the size of the enumeration set, S the size of the slicer set, and $S \times U$ the size of the interference cancellation set. We refer to this structure using the triplet $(U, S \times U, S)$. For example, in Fig. 3(a), we enumerate over $U = 1$ layer only, cancel interference from this layer to the 3 other layers using a 3×1 structure, and slice over $S = 3$ layers. In the structure in Fig. 3(b), we enumerate over $U = 2$ layers, cancel interference using a 2×2 structure, and slice over $S = 2$ layers.

LLR values are generated for bits in symbols included in the detection set only. Complementary structures that enumerate symbols on other decoupled layers are required to generate their respective LLRs. For example, the $(1,3 \times 1,3)$ structure requires 3 similar structures to generate LLRs for layers 2 to 4 (Fig. 2(c)–(e)). When $U > 1$, decoupled layers can overlap by placing a stream with low reliability in multiple detection sets.

A. WL Decomposition (WLD) Scheme

In [51], a decomposition scheme was introduced to transform \mathbf{H} into a punctured LTM \mathbf{L} with a desired structure via a projection matrix \mathbf{W} . In this section, we extend the scheme to handle soft-input MIMO detection using prior LLRs fed from a soft-input-soft-output channel decoder. We assume $N = M$.

We seek a matrix $\mathbf{W} = [\mathbf{w}_1 \mathbf{w}_2 \cdots \mathbf{w}_N] \in \mathcal{C}^{N \times N}$ such that $\mathbf{W}^* \mathbf{H} = \mathbf{L}$ is a punctured LTM and $\mathbf{L} = [l_{ij}] \in \mathcal{C}^{N \times N}$ with $l_{ii} \in \mathcal{R}^+$. In general, if \mathbf{L} is punctured, then \mathbf{W} is non-unitary and hence does not preserve Euclidean norm:

$$\mathbf{y} \triangleq \mathbf{W}^* \tilde{\mathbf{y}} = \mathbf{L} \mathbf{x} + \mathbf{W}^* \mathbf{n} \quad (31)$$

$$g(\mathbf{x}) \triangleq \|\mathbf{y} - \mathbf{L} \mathbf{x}\|^2 \neq \|\tilde{\mathbf{y}} - \mathbf{H} \mathbf{x}\|^2 = d(\mathbf{x}) \quad (32)$$

However, if we impose the condition

$$\text{diag}(\mathbf{W}^* \mathbf{W}) = [1 \ 1 \ \cdots \ 1]_{1 \times N}^T,$$

then the transformed noise vector $\mathbf{W}^* \mathbf{n}$ has an unaltered covariance matrix $\mathbb{E}[\mathbf{W}^* \mathbf{n} \mathbf{n}^* \mathbf{W}] = \sigma^2 \mathbf{I}_N$.

To induce a specific pattern of zeros below the main diagonal in \mathbf{L} , we choose the columns of \mathbf{W} to be orthogonal to the columns of $\mathbf{H} = [\mathbf{h}_1 \ \mathbf{h}_2 \ \cdots \ \mathbf{h}_N]$ where these zeros are to be introduced. More specifically, let $\mathcal{I}_n, n = 1, \dots, N$, be the column index sets where puncturing is desired in each row n of \mathbf{H} . Denote $\mathbf{H}_{\mathcal{I}_n}$ the submatrix formed by the columns of \mathbf{H} whose index belongs to set \mathcal{I}_n . Define the column vector $\tilde{\mathbf{w}}_n = \mathbf{P}_{\mathcal{I}_n}^\perp \mathbf{h}_n$, where

$$\mathbf{P}_{\mathcal{I}_n}^\perp = \mathbf{I}_N - \mathbf{H}_{\mathcal{I}_n} (\mathbf{H}_{\mathcal{I}_n}^* \mathbf{H}_{\mathcal{I}_n})^{-1} \mathbf{H}_{\mathcal{I}_n}^*, \quad (33)$$

and $\mathbf{H}_{\mathcal{I}_n} = \{\mathbf{h}_m | m \in \mathcal{I}_n\}$. Then the column vectors of \mathbf{W} are given by

$$\mathbf{w}_n = \frac{\tilde{\mathbf{w}}_n}{\|\tilde{\mathbf{w}}_n\|} = \frac{\mathbf{P}_{\mathcal{I}_n}^\perp \mathbf{h}_n}{\sqrt{\mathbf{h}_n^* \mathbf{P}_{\mathcal{I}_n}^\perp \mathbf{h}_n}}.$$

Furthermore, it was shown in [51] that \mathbf{L} and $\mathbf{W}^* \tilde{\mathbf{y}}$ can be derived using a simple modification to the standard QL decomposition procedure [52]. This avoids the need for expensive matrix inversion operations in (33). On modern vector digital signal processors (DSPs), matrix QLD operations are natively supported and optimized part of the instruction set. For example, on a CEVA XC-4210 processor [53], QL decomposition of a 4×4 complex matrix requires only 12 clock cycles. Hence, we assume that the channel matrix \mathbf{H} has been preprocessed by a similar DSP, and detection is performed based on the transformed system in (31). Note that because of (32), the solution to the detection problem is no longer optimal (but still achieves near-optimal performance as demonstrated in Section VII).

B. Optimized Detection Algorithm Using WLD

We next present an optimized detection algorithm based on the WLD scheme, by extending the $N = 2$ case of Section III. For simplicity, we only consider decompositions of the form $(1, S \times 1, S)$, similar to Fig. 2. The N layers are decoupled by first circularly shifting the columns of \mathbf{H} , and then performing WLD on the permuted \mathbf{H} . We refer to the decomposition whose detection set is the m th layer as the m th WLD of \mathbf{H} . To simplify notation, we describe the detection steps for $m = 1$. The same steps apply to detect the other layers with an appropriate adjustment to the layer indices. Let

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_N \end{bmatrix}, \quad \mathbf{L} = \begin{bmatrix} \alpha & 0 & 0 & 0 & 0 \\ \gamma_2 & \beta_2 & 0 & 0 & 0 \\ \gamma_3 & 0 & \beta_3 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \gamma_N & 0 & \cdots & 0 & \beta_N \end{bmatrix}, \quad (34)$$

be the transmitted symbol vector, received signal vector, and the WL-decomposed channel matrix in normal order, respectively, where: $y_n \in \mathcal{C}$, $x_n \in \mathcal{X}_n$ for $n = 1, \dots, N$; $\alpha, \beta_n \in \mathcal{R}^+$ and $\gamma_n \in \mathcal{C}$ for $n = 2, \dots, N$. Then the distance metric $g(\mathbf{x})$ of \mathbf{x} from \mathbf{y} based on \mathbf{L} in (32) can be written as

$$g(\mathbf{x}) = f_1(x_1) + \sum_{n=2}^N f_n(x_n | x_1), \quad (35)$$

where

$$f_1(x_1) = |y_1 - \alpha x_1|^2 - \mathbf{b}^T(x_1)\boldsymbol{\lambda}_1, \text{ and}$$

$$f_n(x_n | x_1) = |y_n - \gamma_n x_1 - \beta_n x_n|^2 - \mathbf{b}^T(x_n)\boldsymbol{\lambda}_n,$$

for $n = 2, \dots, N$. We next minimize $g(\mathbf{x})$ similar to (5):

$$g^{\text{WL}} \triangleq \min_{\mathbf{x} \in \mathcal{X}} \left\{ f_1(x_1) + \sum_{n=2}^N f_n(x_n | x_1) \right\} \quad (36)$$

$$= \min_{x_1 \in \mathcal{X}_1} \left\{ f_1(x_1) + \sum_{n=2}^N \min_{x_n \in \mathcal{X}_n} f_n(x_n | x_1) \right\} \quad (37)$$

$$= \min_{x_1 \in \mathcal{X}_1} \left\{ f_1(x_1) + \sum_{n=2}^N f_n(\hat{x}_n(x_1) | x_1) \right\}$$

$$= \min_{x_1 \in \mathcal{X}_1} g(x_1, \hat{x}_2(x_1), \dots, \hat{x}_N(x_1)) \quad (38)$$

where

$$\hat{x}_n(x_1) = \arg \min_{x_n \in \mathcal{X}_n} \{ |y_n - \gamma_n x_1 - \beta_n x_n|^2 - \mathbf{b}^T(x_n)\boldsymbol{\lambda}_n \}.$$

Denote the set of sliced symbol vectors for all possible x_1 in (38) by (defined similar to (8) but for any $N \geq 2$)

$$\mathcal{O}_1 = \{ [x_1 \hat{x}_2(x_1) \cdots \hat{x}_N(x_1)]^T : x_1 \in \mathcal{X}_1 \}.$$

The symbol vector that minimizes (35) is denoted as

$$\mathbf{x}^{\text{WL}} \triangleq \arg \min_{\mathbf{x} \in \mathcal{O}_1} g(\mathbf{x}). \quad (39)$$

To efficiently determine g^{WL} , we optimize the distance computations in (36) by splitting the complex quantities into their real and imaginary components:

$$f_1(x_1) = f_{1\text{R}}(x_{1\text{R}}) + f_{1\text{I}}(x_{1\text{I}})$$

$$f_{1\text{R}}(x_{1\text{R}}) = (y_{1\text{R}} - \alpha x_{1\text{R}})^2 - \mathbf{b}_{1\text{R}}^T \boldsymbol{\lambda}_{1\text{R}}$$

$$f_{1\text{I}}(x_{1\text{I}}) = (y_{1\text{I}} - \alpha x_{1\text{I}})^2 - \mathbf{b}_{1\text{I}}^T \boldsymbol{\lambda}_{1\text{I}}$$

and

$$f_n(x_n) = f_{n\text{R}}(x_{n\text{R}}) + f_{n\text{I}}(x_{n\text{I}})$$

$$f_{n\text{R}}(x_{n\text{R}} | x_1) = (y_{n\text{R}} - \gamma_{n\text{R}} x_{1\text{R}} + \gamma_{n\text{I}} x_{1\text{I}} - \beta_n x_{n\text{R}})^2$$

$$- \mathbf{b}_{n\text{R}}^T \boldsymbol{\lambda}_{n\text{R}}$$

$$f_{n\text{I}}(x_{n\text{I}} | x_1) = (y_{n\text{I}} - \gamma_{n\text{R}} x_{1\text{I}} - \gamma_{n\text{I}} x_{1\text{R}} - \beta_n x_{n\text{I}})^2$$

$$- \mathbf{b}_{n\text{I}}^T \boldsymbol{\lambda}_{n\text{I}}$$

for $n \geq 2$. Substituting back in (37), expanding terms, minimizing w.r.t. $x_{n\text{R}}$ and $x_{n\text{I}}$, and eliminating irrelevant terms, we obtain

$$\bar{g}(\mathbf{x}) = \bar{f}_{1\text{R}}(x_{1\text{R}}) + \bar{f}_{1\text{I}}(x_{1\text{I}})$$

$$+ \sum_{n=2}^N \left(\min_{x_{n\text{R}} \in \mathcal{P}_n} \bar{f}_{n\text{R}}(x_{n\text{R}} | x_1) + \min_{x_{n\text{I}} \in \mathcal{P}_n} \bar{f}_{n\text{I}}(x_{n\text{I}} | x_1) \right) \quad (40)$$

where

$$\bar{f}_{1\text{R}}(x_{1\text{R}}) = A x_{1\text{R}}^2 + C x_{1\text{R}} - \mathbf{b}_{1\text{R}}^T \boldsymbol{\lambda}_{1\text{R}}$$

$$\bar{f}_{1\text{I}}(x_{1\text{I}}) = A x_{1\text{I}}^2 + D x_{1\text{I}} - \mathbf{b}_{1\text{I}}^T \boldsymbol{\lambda}_{1\text{I}}$$

$$\bar{f}_{n\text{R}}(x_{n\text{R}}) = (E_n x_{1\text{R}} + F_n x_{1\text{I}}) x_{n\text{R}}$$

$$+ (B_n x_{n\text{R}}^2 + G_n x_{n\text{R}} - \mathbf{b}_{n\text{R}}^T \boldsymbol{\lambda}_{n\text{R}})$$

$$\bar{f}_{n\text{I}}(x_{n\text{I}}) = (E_n x_{1\text{I}} - F_n x_{1\text{R}}) x_{n\text{I}}$$

$$+ (B_n x_{n\text{I}}^2 + H_n x_{n\text{I}} - \mathbf{b}_{n\text{I}}^T \boldsymbol{\lambda}_{n\text{I}})$$

Similar to (18)–(19), the constants above are given by:

$$A = \alpha^2 + \sum_{n=2}^N |\gamma_n|^2, \quad B_n = \beta_n^2$$

$$C = -2\alpha y_{1\text{R}} - 2 \sum_{n=2}^N (\gamma_{n\text{R}} y_{n\text{R}} + \gamma_{n\text{I}} y_{n\text{I}})$$

$$D = -2\alpha y_{1\text{I}} + 2 \sum_{n=2}^N (\gamma_{n\text{I}} y_{n\text{R}} - \gamma_{n\text{R}} y_{n\text{I}})$$

$$E_n = +2\beta_n \gamma_{n\text{R}}, \quad F_n = -2\beta_n \gamma_{n\text{I}},$$

$$G_n = -2\beta_n y_{n\text{R}}, \quad H_n = -2\beta_n y_{n\text{I}}.$$

Using $g(\mathbf{x})$ (or $\bar{g}(\mathbf{x})$), the LLRs of the bits in layer 1 are

$$\Lambda_{1,j}^{\text{WL}} = \min_{x_1 \in \mathcal{X}_{1,j}^{(+1)}} g(x_1, \hat{x}_2(x_1), \dots, \hat{x}_N(x_1))$$

$$- \min_{x_1 \in \mathcal{X}_{1,j}^{(-1)}} g(x_1, \hat{x}_2(x_1), \dots, \hat{x}_N(x_1)). \quad (41)$$

The bit-LLRs in the remaining $N - 1$ layers are similarly obtained by using the other $N - 1$ complementary WL structures of \mathbf{H} (see Fig. 2). Finally, (26)–(28) for $N = 2$ can be used to slice $\hat{x}_{n\text{R}} = \min_{x_{n\text{R}} \in \mathcal{P}_n} f_{n\text{R}}(x_{n\text{R}} | x_1)$, and (29)–(30) to slice $\hat{x}_{n\text{I}} = \min_{x_{n\text{I}} \in \mathcal{P}_n} f_{n\text{I}}(x_{n\text{I}} | x_1)$, but with the constants B, E, F, G, H replaced by B_n, E_n, F_n, G_n, H_n , and $\mathcal{P}_2, \boldsymbol{\lambda}_{2\text{R}}, \boldsymbol{\lambda}_{2\text{I}}$ by $\mathcal{P}_n, \boldsymbol{\lambda}_{n\text{R}}, \boldsymbol{\lambda}_{n\text{I}}$.

C. Post LLR Processing

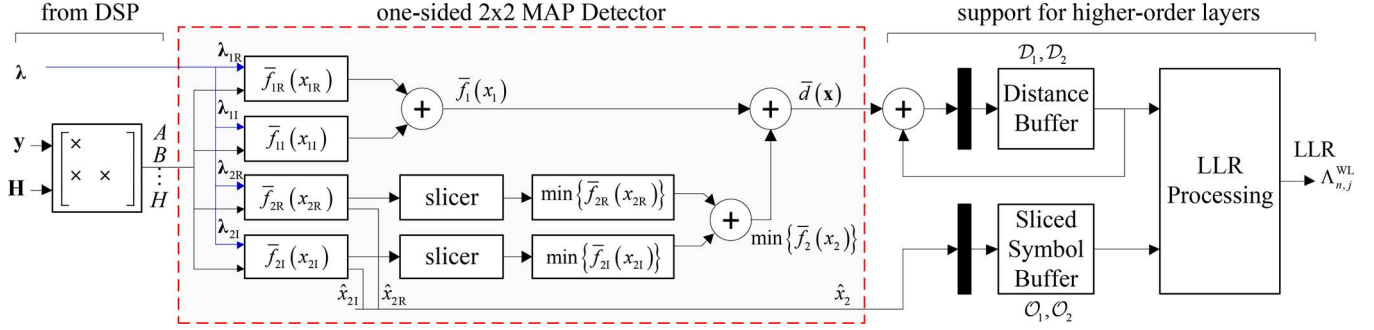
Since $g(\mathbf{x}) \neq d(\mathbf{x})$, there is no guarantee that the g^{WL} and \mathbf{x}^{WL} obtained in (38) and (39) using one WLD structure of \mathbf{H} , are the same ones obtained using the other $N - 1$ WLD structures with the columns of \mathbf{H} permuted. To avoid confusion, we refer to the quantities in (35), (38), and (39) pertaining to the m th layer WL decomposition using the subscript m : $g_m(\mathbf{x})$, g_m^{WL} , \mathbf{x}_m^{WL} .

The ‘‘WL-minimal’’ HD solution, denoted as g_{\min}^{WL} and $\mathbf{x}_{\min}^{\text{WL}}$, corresponds to the minimum of the N various g_m^{WL} values:

$$g_{\min}^{\text{WL}} = \min_m g_m^{\text{WL}}, \quad \mathbf{x}_{\min}^{\text{WL}} = \arg \min_{\mathbf{x}_m^{\text{WL}}} g_m^{\text{WL}}.$$

A similar minimization is required as well to adjust the LLR values $\Lambda_{n,j}^{\text{WL}}$ relative to the global minimum g_{\min}^{WL} and the bits of its corresponding symbol vector $\mathbf{x}_{\min}^{\text{WL}}$. This adjustment cannot be done by comparing the individual $\Lambda_{n,j}^{\text{WL}}$'s alone. One simple way is based on the list of distances $g_m(\mathbf{x})$ generated from all decompositions for $m = 1, \dots, N$, together with their corresponding symbol vectors. Let \mathcal{O}_m denote the set of symbol vectors

$$\mathcal{O}_m = \{ [\hat{x}_1(x_m) \cdots \hat{x}_{m-1}(x_m) x_m \hat{x}_{m+1}(x_m) \cdots \hat{x}_N(x_m)]^T : x_m \in \mathcal{X}_m \}, \quad m = 1, \dots, N, \quad (42)$$

Fig. 4. Block diagram of a parallel one-sided 2×2 MAP detector core, with input and output interfaces.

where the n th sliced symbol in the m th WLD is

$$\hat{x}_n(x_m) = \arg \min_{x_n \in \mathcal{X}_n} \left\{ |y_{n,m} - \gamma_{n,m}x_m - \beta_{n,m}x_n|^2 - \mathbf{b}^T(x_n)\boldsymbol{\lambda}_n \right\},$$

for $n \neq m$. Here $y_{n,m}$, $\gamma_{n,m}$, and $\beta_{n,m}$ are defined as in (34) but relative to the m th WLD of \mathbf{H} . Next, define the partitions on \mathcal{O}_m : $\mathcal{O}_{n,j,m}^{(+1)} = \{\mathbf{x} \in \mathcal{O}_m : b_{n,j} = +1\}$ and $\mathcal{O}_{n,j,m}^{(-1)} = \{\mathbf{x} \in \mathcal{O}_m : b_{n,j} = -1\}$. Then the ‘‘WL-minimal’’ LLRs are given by

$$\Lambda_{n,j,\min}^{\text{WL}} = \min_m \left\{ \min_{\mathbf{x} \in \mathcal{O}_{n,j,m}^{(+1)}} g_m(\mathbf{x}) \right\} - \min_m \left\{ \min_{\mathbf{x} \in \mathcal{O}_{n,j,m}^{(-1)}} g_m(\mathbf{x}) \right\}. \quad (43)$$

D. Discussion

The key equations for the general N -layer case derived above reduce to the optimal equations derived in Section III for $N = 2$. A comparison between the two shows that the same operations applied to compute $\bar{d}(\mathbf{x})$ in (13) are applied to compute $\bar{g}(\mathbf{x})$ in (40), but using the respective constants of layer n instead of layer 2. Hence, a 2×2 MAP detector can be viewed as a building block for constructing detectors for higher-order layers, with a simple modification to account for the extra accumulated sum terms in (40), in addition to the LLR processing of (43) at the output stage. A parallel architecture will be developed next and its complexity analyzed.

V. PARALLEL 2-LAYER DETECTOR ARCHITECTURE

Fig. 4 shows a block diagram of a parallel 2×2 MAP detector core that implements the key equations in (13)–(17). For flexibility and scalability to higher-order layers, the constellations supported on each layer are configurable from BPSK up to 256-QAM, and can be distinct on each layer. We assume the input constants (18)–(19) to the detector are supplied by an external DSP. The outputs are two lists of distances $\mathcal{D}_1, \mathcal{D}_2$ and their associated lists of symbol vectors $\mathcal{O}_1, \mathcal{O}_2$, which are fed to a post LLR processing stage to extract the LLRs values depending on the number of layers.

A. Optimized Implementation of Distance Expressions

A careful inspection of expressions (14)–(17) shows that $\bar{d}(\mathbf{x})$ can be evaluated without using multipliers, assuming

TABLE I
DISTINCT PRODUCT TERMS TO BE COMPUTED: $x, y, z \in \mathcal{P}_2$; $r, s \in \mathcal{R}$

# distinct terms	2-PAM	4-PAM	8-PAM	16-PAM
$r \cdot x $	1	2	4	8
$r \cdot x \cdot y $	1	3	10	33
$r \cdot x^2$	1	2	4	8
$(r \cdot x \pm s \cdot y) \cdot z $	2	14	116	914
$\mathbf{b}_{1R}^T \boldsymbol{\lambda}_{1R}$	1	2	4	8

the constants are pre-processed and fed as inputs to the detector. The reason is that the variables x_{1R} , x_{1I} , x_{2R} , and x_{2I} are integers that belong to a PAM constellation. More specifically, in LTE [2], they are odd integers in the set $\mathcal{P}_2 = \{2m + 1 \mid m = -P/2 + 1, \dots, 0, \dots, P/2 - 1\}$ and $P = \sqrt{Q_2}$. Hence the terms that involve the products of x_{1R} , x_{1I} , x_{2R} , x_{2I} in (14)–(17) with the constants in (18)–(19) are simply integer multiples of these constants. These product terms can be computed using basic *addition operations* with appropriate power-of-2 manipulations of the operands without using expensive multipliers. Also from symmetry, only positive multiples need to be computed. Table I summarizes the number of various distinct product terms that need to be computed for various PAM constellation sizes.

Moreover, the dot products $\mathbf{b}_{1R}^T \boldsymbol{\lambda}_{1R}$ between the input LLR vectors and all the bit vectors are simply all linear binary combinations of the $q_1/2 = (\log_2 Q_1)/2$ individual input LLRs λ_i of $\boldsymbol{\lambda}_{1R}$:

$$\pm \lambda_1 \pm \lambda_2 \pm \dots \pm \lambda_{q_1/2}.$$

Also from symmetry, only half of these sums actually need to be computed, giving a total of $2^{q_1/2 - 1}$ different sums. The same applies to other dot product terms in (15)–(17).

Next, as x_{1R} runs over the P_1 integers in \mathcal{P}_1 , the expression $(Ax_{1R}^2 + Cx_{1R} - \mathbf{b}^T(x_{1R})\boldsymbol{\lambda}_{1R})$ takes P_1 different values. However, because of the Gray mapping of the bits, then $\mathbf{b}^T(-x_{1R})\boldsymbol{\lambda}_{1R} \neq -\mathbf{b}^T(x_{1R})\boldsymbol{\lambda}_{1R}$ and hence there is no symmetry that can be exploited to save in computations here. The same argument applies to the three other expressions $(Ax_{1I}^2 + Dx_{1I} - \mathbf{b}_{1I}^T \boldsymbol{\lambda}_{1I})$, $(Bx_{2R}^2 + Gx_{2R} - \mathbf{b}_{2R}^T \boldsymbol{\lambda}_{2R})$, and $(Bx_{2I}^2 + Hx_{2I} - \mathbf{b}_{2I}^T \boldsymbol{\lambda}_{2I})$ in (15)–(17).

Finally, for the remaining sum of products of cross terms $(Ex_{1R} + Fx_{1I})x_{2R}$, as x_{2R} cycles through the P_2 integers in \mathcal{P}_2 , the expression takes P_2 different values for every pair (x_{1R}, x_{1I}) . However, for all possible (x_{1R}, x_{1I}) , repetitions occur. The number of unique values of $(Ex_{1R} + Fx_{1I})x_{2R}$ is

twice that of $(E|x_{1R}| \pm F|x_{1I}|)|x_{2R}|$ (summarized in Table I). By symmetry, these are also the same values taken by the other sub-expression $(Ex_{1I} - Fx_{1R})x_{2I}$ in (17).

Therefore, hardware complexity will be measured in terms of number of adders, in addition to number of (2:1)-multiplexers (muxes) needed to steer operands to these adders. Wider ($n : 1$)-muxes can be constructed using $n - 1(2 : 1)$ -muxes.

We next determine the actual number of adders required to compute each of the unique terms in (14)–(17), assuming 256-QAM and its underlying 1D 16-PAM constellation. The same analysis applies to other constellations. The required multiples Ax_{1R}^2 for 16-PAM are $\{9, 25, 49, 81, 121, 169, 225\} \times A$, which can be generated using 11 adders as follows:

$$\begin{aligned} 9A &= 8A + A, & 25A &= 16A + 9A, & 15A &= 16A - A \\ 49A &= 64A - 15A, & 81A &= 32A + 49A, & 7A &= 8A - A \\ 121A &= 128A - 7A, & 41A &= 32A + 9A, & 169A &= 128A + 41A \\ 31A &= 32A - A & 225A &= 256A - 31A \end{aligned}$$

Similarly, the 8 multiples $C|x_{1R}|$ can be generated using 7 adders. For $\mathbf{b}_{1R}^T \boldsymbol{\lambda}_{1R}$, 8 values of can be generated as

$$\begin{aligned} (\lambda_1 + \lambda_2) \pm (\lambda_3 + \lambda_4), & (\lambda_1 + \lambda_2) \pm (\lambda_3 - \lambda_4), \\ (\lambda_1 - \lambda_2) \pm (\lambda_3 + \lambda_4), & (\lambda_1 - \lambda_2) \pm (\lambda_3 - \lambda_4) \end{aligned}$$

with 12 adders. The other 8 are their negatives.

To generate the unique elements of $(E|x_{1R}| \pm F|x_{1I}|)|x_{2R}|$, we first generate all unique sums with $x_{2R} = 1$, i.e., $(E|x_{1R}| \pm F|x_{1I}|)$, such that $\gcd(|x_{1R}|, |x_{1I}|) = 1$, and then generate all their multiples. The number of unique sums of the form $(E|x_{1R}| \pm F|x_{1I}|)$ with co-prime coefficients $|x_{1R}|$ and $|x_{1I}|$ from the set $\{1, 3, \dots, 15\}$ is 49. We next enumerate the unique multiples from each of these 49 classes. For $(|x_{1R}|, |x_{1I}|) = (1, 1)$, there are 33×2 distinct multiples of $(E \pm F)$. For $(|x_{1R}|, |x_{1I}|) = (1, 3)$ or $(3, 1)$, there are 18×2 distinct multiples of $(E \pm 3F)$ and 18×2 of $(3E \pm F)$. For $(|x_{1R}|, |x_{1I}|) = (1, 5)$, $(5, 1)$, $(3, 5)$, or $(5, 3)$, there are 13×2 distinct multiples of each. Finally, for the remaining 42 classes, there are 8×2 distinct multiples of each. Summing all distinct multiples we obtain 914.

Table II summarizes the various constants that appear in the computation of $(E|x_{1R}| \pm F|x_{1I}|)|x_{2R}|$ for 16-PAM, and how they are generated using addition operations involving powers-of-2 operands and other already computed constants. First, the odd multiples $3E, 5E, \dots, 15E$, and $3F, 5F, \dots, 15F$, require 14 adders. The term $(E \pm F)$ and its 33×2 distinct multiples require all the 36 constants in Table II and hence need $(36 + 1) \times 2$ adders. The term $(E \pm 3F)$ requires 18 constants $\{1, 3, 5, 7, 9, 11, 13, 15, 21, 25, 27, 33, 35, 39, 45, 55, 65, 75\}$, and hence needs 2×18 adders. The same count is needed for $(3E \pm F)$. On the other hand, the term $(E \pm 5F)$ and its 13×2 distinct multiples require only 13 constants $\{1, 3, 5, 7, 9, 11, 13, 15, 21, 27, 33, 39, 45\}$ and hence need 2×13 adders. The same applies for $(5E \pm F)$, $(3E \pm 5F)$, and $(5E \pm 3F)$. For the remaining 42 classes, only 8 constants $\{1, 3, 5, 7, 9, 11, 13, 15\}$ appear and hence need $2 \times 8 \times 42$ adders. Summing all counts results in a total of 936 adders. Finally note that $(E|x_{1R}| \pm F|x_{1I}|)|x_{2R}|$ takes the same values as $(E|x_{1R}| \mp F|x_{1I}|)|x_{2I}|$ but in a different order.

TABLE II

CONSTANTS THAT APPEAR IN $(E|x_{1R}| \pm F|x_{1I}|)|x_{2I}|$ FOR 16-PAM

3=2+1	5=4+1	7=8-1	9=8+1
11=8+3	13=16-3	15=16-1	21=16+5
25=16+9	27=32-5	33=32+1	35=32+3
39=32+7	45=32+13	49=64-15	55=64-9
63=64-1	65=64+1	75=64+11	77=64+13
81=32+49	91=64+27	99=64+35	41=32+9
105=64+41	117=128-11	121=128-7	135=128+7
143=128+15	37=33+4	165=128+37	169=128+41
61=64-3	195=256-61	31=32-1	225=256-31

B. Minimization by Exhaustive Search

One approach to implement the minimizations in (13) is by exhaustive search. In (16), for every pair (x_{1R}, x_{1I}) , 16 out of 914 distinct values of $(E|x_{1R}| \pm F|x_{1I}|)|x_{2R}|$ pertaining to the 16 different x_{2R} 's are added to $Bx_{2R}^2 + Gx_{2R} - \mathbf{b}_{2R}^T \boldsymbol{\lambda}_{2R}$, and the minimum is selected. Hence a total of 16×256 adders are needed to generate all possible values of $\bar{f}_{2R}(x_{2R} | x_1)$. The same holds for $\bar{f}_{2I}(x_{2I} | x_1)$. To find the minimum among P_2 quantities, a binary tree of comparators comprised of $P_2 - 1$ adders and $P_2 - 1(2 : 1)$ -multiplexers are needed. A total of 2×256 such comparators are needed. Finally, the 256 minima from each case are added to complete the sum for $\bar{d}(\mathbf{x})$ in (13).

To generate the hard-decision MAP solution, the minimum among the 256 distances $\bar{d}(\mathbf{x})$ must be taken and the corresponding constellation symbol be identified. This requires a total of 255 adders and 255 muxes. On the other hand, to compute the output LLRs of the bits in x_1 according to (9), the 256 distances in (13) must be minimized over two complementary sets for every bit and their difference be taken. The 256-QAM constellation points can be viewed as 16 columns each containing 16 points, or as 16 rows each containing 16 points. In LTE, the 4 bits corresponding to the real part of the constellation points do not change in every column, and the 4 bits corresponding to the imaginary part do not change in every row. Hence it suffices to take the minimum distances among all points in each row and among all points in each column independently. The column minima are used to compute the LLRs of the real bits by partitioning the columns into two groups of 8 columns depending on whether the bit is +1 or -1 in the column. The minimum distance among each group of columns is taken, and the difference of the two minima generates the LLR of that bit. The same applies to the imaginary bits and the row minima. Hence a total of 2×16 16-point comparators are needed, amounting to 480 adders and 480 muxes, to extract the minima, followed by 8 adders to take the differences.

Table III summarizes the core complexity using exhaustive search. The core requires 18290 adders and 8160 muxes.

C. Minimization by Slicing

We next analyze the complexity of computing $\min_{x_{2R} \in \mathcal{P}_2} \bar{f}_{2R}(x_{2R} | x_1)$ in (13) via the slicing approach by first determining $\hat{x}_{2R} = \arg \min_{x_{2R} \in \mathcal{P}_2} \bar{f}_{2R}(x_{2R} | x_1)$ followed by evaluating $\bar{f}_{2R}(\hat{x}_{2R} | x_1)$, for all possible x_1 . To minimize $\bar{f}_{2R}(x_{2R} | x_1)$, the decision boundaries $R(x_{2R}, \bar{x}_{2R})$ in (26) must be computed for all $x_{2R} \neq \bar{x}_{2R} \in \mathcal{P}_2$, and appropriate minima and maxima must be extracted from these boundaries according to (28) and compared to $Ex_{1R} + Fx_{1I}$. Similarly, to minimize $\bar{f}_{2I}(x_{2I} | x_1)$, the decision boundaries $I(x_{2I}, \bar{x}_{2I})$ in

TABLE III
RESOURCES OF DETECTOR CORE USING EXHAUSTIVE SEARCH

# adders (& muxes)	2-PAM	4-PAM	8-PAM	16-PAM
Ax_{1R}^2	0	1	4	11
$C x_{1R} $	0	1	3	7
$\mathbf{b}_{1R}^T \lambda_{1R}$	0	2	6	12
$f_{1R}(x_{1R})$	4	8	16	32
$D x_{1R} $	0	1	3	7
$\mathbf{b}_{1I}^T \lambda_{1I}$	0	2	6	12
$f_{1I}(x_{1I})$	4	8	16	32
$\tilde{f}_1(x_1) = f_{1R}(x_{1R}) + f_{1I}(x_{1I})$	4	16	64	256
Bx_{2R}^2	0	1	4	11
$G x_{2R} $	0	1	3	7
$\mathbf{b}_{2R}^T \lambda_{2R}$	0	2	6	12
$Bx_{2R}^2 + Gx_{2R} - \mathbf{b}_{2R}^T \lambda_{2R}$	4	8	16	32
$H x_{2I} $	0	1	3	7
$\mathbf{b}_{2I}^T \lambda_{2I}$	0	2	6	12
$Bx_{2I}^2 + Hx_{2I} - \mathbf{b}_{2I}^T \lambda_{2I}$	4	8	16	32
$(E x_{1R} \pm F x_{1I}) x_{2R} $	2	16	122	936
$\tilde{f}_{2R}(x_{2R} x_1)$	8	64	512	4096
$\bar{m}_{2R} = \min\{\tilde{f}_{2R}(x_{2R} x_1)\}$ muxes \rightarrow	4	48	448	3840
$\tilde{f}_{2I}(x_{2I} x_1)$	8	64	512	4096
$\bar{m}_{2I} = \min\{\tilde{f}_{2I}(x_{2I} x_1)\}$ muxes \rightarrow	4	48	448	3840
$\tilde{f}_1 + \bar{m}_{2R} + \bar{m}_{2I}$	8	32	128	512
HD solution: $\min\{\tilde{f}_1 + \bar{m}_{2R} + \bar{m}_{2I}\}$ muxes \rightarrow	3	15	63	255
soft-output LLRs muxes \rightarrow	6	28	118	488
	4	24	112	480
Total (soft-output)	60	346	2460	18290
	12	120	1008	8160

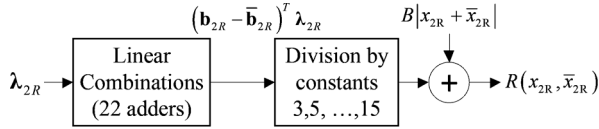


Fig. 5. Computation of decision boundaries.

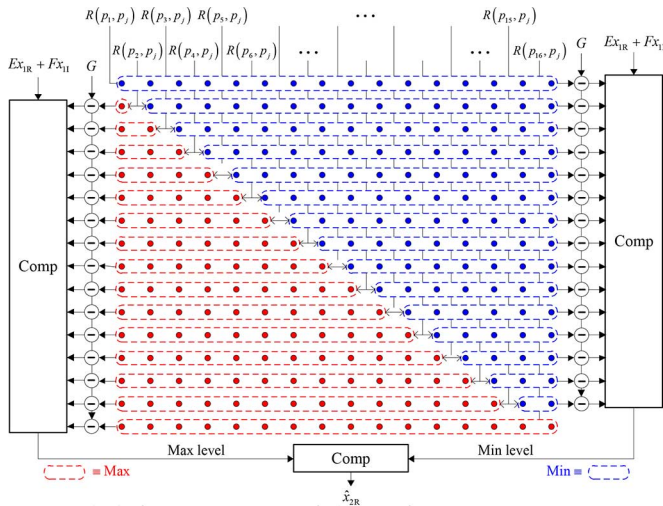


Fig. 6. Block diagram of optimized slicer architecture.

(29) must be computed for all $x_{2I} \neq \bar{x}_{2I} \in \mathcal{P}_2$, and appropriate minima and maxima must be extracted from these boundaries according to (30) and compared to $Ex_{1I} - Fx_{1R}$.

By analogy, it suffices to analyze the complexity of (26) and (28). Since $R(x_{2R}, \bar{x}_{2R}) = R(\bar{x}_{2R}, x_{2R})$, only $P_2(P_2 - 1)/2 = 120$ decision boundaries need to be computed (see Fig. 5). The sum $|x_{2R} + \bar{x}_{2R}|$ takes $P_2 - 2$ distinct non-zero values $(2, 4, \dots, 2P_2 - 4)$, and hence the product $B|x_{2R} + \bar{x}_{2R}|$ term in (26) requires 6 adders. Similarly, the difference $|x_{2R} - \bar{x}_{2R}|$ takes $P_2 - 1$ distinct non-zero values $(2, 4, \dots, 2P_2 - 2)$. For the division of $(\mathbf{b}_{2R} - \bar{\mathbf{b}}_{2R})^T \lambda_{2R}$ by these constants, where $\bar{\mathbf{b}}_{2R} = \mathbf{b}(\bar{x}_{2R})$, the term $(\mathbf{b}_{2R} - \bar{\mathbf{b}}_{2R})^T \lambda_{2R}$ takes 80 distinct values, 40 of which can be obtained by negation. These 40 values require 22 adders. The required ratios $(\mathbf{b}_{2R} - \bar{\mathbf{b}}_{2R})^T \lambda_{2R} / (x_{2R} - \bar{x}_{2R})$ take only 40 distinct values, and require divisions by 3,5,7,9,11,13,15. However, each value of $(\mathbf{b}_{2R} - \bar{\mathbf{b}}_{2R})^T \lambda_{2R}$ need not be divided by all these 7 constants. By going over all various combinations, it is easy to show that the number of divisions by the various values of $|x_{2R} - \bar{x}_{2R}|$ is as follows (constant : count):

$$2 : 4, 4 : 3, 6 : 5, 8 : 2, 10 : 5, 12 : 3, 14 : 4, 16 : 1, \\ 18 : 3, 20 : 2, 22 : 3, 24 : 1, 26 : 2, 28 : 1, 30 : 1$$

Divisions by powers-of-2 are trivial. Division by 3 covers division by 6 = 3 × 2, 12 = 3 × 4, and 24 = 3 × 8, and hence is needed 9 times. Division by 5 covers division by 10 and 20, and hence is needed 7 times. In a similar fashion, division by 7 is needed 5 times, by 9 is needed 3 times, by 11 is needed 3 times, by 13 is needed 2 times, and by 15 is needed once. The total number of such non-trivial divisions is 30. The complexity of a division-by-small-constant circuit is roughly equivalent to a small number of adders for small bit-widths. Specifically, a divide-by-3 is equivalent to 1 adder; by 5, 7, 9, and 11 are equivalent to 2 adders; and by 13 and 15 are equivalent to 3 adders. Hence, the ratios in (26) can be computed using 54 adders. Finally, computing all 120 decision boundaries by adding/subtracting the various 14 non-zero values of $B|x_{2R} + \bar{x}_{2R}|$ to the various 40 distinct ratios $(\mathbf{b}_{2R} - \bar{\mathbf{b}}_{2R})^T \lambda_{2R} / (x_{2R} - \bar{x}_{2R})$ requires 112 adders ($B|x_{2R} + \bar{x}_{2R}| = 0$ in 8 cases out of the 120).

Moving to (28), a subset of $P_2 - 1$ minimum and $P_2 - 1$ maximum regions must be extracted from these boundaries for every hypothesis point x_{2R} w.r.t. all other $P_2 - 1$ points in \mathcal{P}_2 . These can be obtained using a set of P_2 comparator trees, comprising a total of $14 \times 15 = 210$ adders and 210 (2:1)-MUXES. Next, G is subtracted from each of the $P_2 - 1$ min and $P_2 - 1$ max boundaries using 30 adders. Finally, comparisons between $Ex_{1R} + Fx_{1I}$ and these min/max boundaries are required to determine \hat{x}_{2R} according to (28). Each comparison requires 30 adders. Only 128 such comparisons are needed for $|Ex_{1R} \pm Fx_{1I}|$, requiring a total of 3840 adders. The other 128 are derived by symmetry. Fig. 6 shows the architecture of the slicer block in Fig. 4.

Based on the results from the slicers, the \hat{x}_{2R} 's are used to evaluate $\tilde{f}_{2R}(\hat{x}_{2R}|x_1)$. This is done by selecting the appropriate multiples $|(Ex_{1R} \pm Fx_{1I})\hat{x}_{2R}|$ to be added to $B\hat{x}_{2R}^2 + G\hat{x}_{2R} - \mathbf{b}^T(\hat{x}_{2R})\lambda_{2R}$. Hence 256 adders are needed, in addition to 128 (8:1)-MUXES and 256 (16:1)-MUXES.

Table IV summarizes the complexity resources of the slicer-based detector. The architecture requires 11246 adders and 10372 muxes, which amount to a 38.52% savings in adders and an increase of 27.11% in muxes compared with the previous architecture using exhaustive search minimization. The internal pipeline registers, output buffers and accumulators in

TABLE IV
RESOURCES OF DETECTOR CORE USING SLICERS

# adders (& muxes)	2-PAM	4-PAM	8-PAM	16-PAM
$\bar{f}_1(x_1) = \bar{f}_{1R}(x_{1R}) + \bar{f}_{1I}(x_{1I})$	4	16	64	256
$B x_{2R} + \bar{x}_{2R} $	0	0	2	6
$(\mathbf{b}_{2R} - \bar{\mathbf{b}}_{2R})^T \boldsymbol{\lambda}_{2R}$	0	2	8	22
$\frac{(\mathbf{b}_{2R} - \bar{\mathbf{b}}_{2R})^T \boldsymbol{\lambda}_{2R}}{x_{2R} - \bar{x}_{2R}}$	0	1	10	54
$R(x_{2R}, \bar{x}_{2R})$	0	4	24	112
min / max boundaries (MUXES)	0	6	42	210
min / max boundaries – G	2	6	14	54
$ Ex_{1R} \pm Fx_{1I} \cdot x_{2R} $	2	16	122	936
Compare $ Ex_{1R} \pm Fx_{1I} $ and min / max boundaries – G	4	48	448	3840
$\bar{f}_{2R}(\hat{x}_{2R} x_1) = Ex_{1R} \pm Fx_{1I} \cdot \hat{x}_{2R} +$ $(B\hat{x}_{2R}^2 + G\hat{x}_{2R} - \mathbf{b}^T(\hat{x}_{2R})\boldsymbol{\lambda}_{2R})$	4	16	64	256
	4	56	544	4736
$(\mathbf{b}_{2I} - \bar{\mathbf{b}}_{2I})^T \boldsymbol{\lambda}_{2I}$	0	2	8	22
$\frac{(\mathbf{b}_{2I} - \bar{\mathbf{b}}_{2I})^T \boldsymbol{\lambda}_{2I}}{x_{2I} - \bar{x}_{2I}}$	0	1	10	54
$I(x_{2I}, \bar{x}_{2I})$	0	4	24	112
min / max boundaries (MUXES)	0	6	42	210
min / max boundaries – H	2	6	14	54
Compare $ Ex_{1I} \mp Fx_{1R} $ and min / max boundaries – H	4	48	448	3840
$\bar{f}_{2I}(\hat{x}_{2I} x_1) = Ex_{1I} \mp Fx_{1R} \cdot \hat{x}_{2I} +$ $(B\hat{x}_{2I}^2 + H\hat{x}_{2I} - \mathbf{b}^T(\hat{x}_{2I})\boldsymbol{\lambda}_{2I})$	4	16	64	256
	4	56	544	4736
$\bar{f}_1(x_1) + \bar{f}_{2R}(\hat{x}_{2R} x_1) + \bar{f}_{2I}(\hat{x}_{2I} x_1)$	4	32	128	512
soft-output LLRs	6	28	118	488
muxes \rightarrow	4	24	112	480
Total	36	258	1654	11246
(2:1)-MUXS	12	148	1284	10372

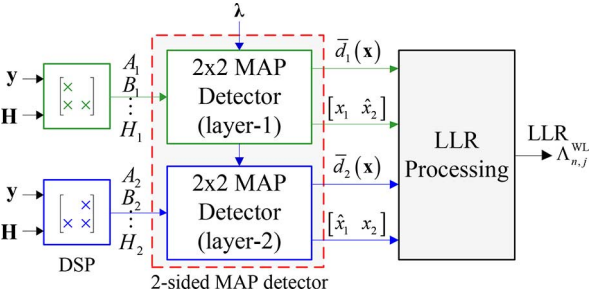


Fig. 7. Block diagram of 2-sided MAP detector.

Fig. 4 are the same between the 2 architectures, and thus are not included in the comparisons.

D. Multi-Core Detector Architectures

Depending on the target throughput and the number of antennas N in the MIMO systems, multiple detector cores similar to Fig. 4 can be configured to build a MIMO detector. Fig. 7 shows a 2-sided fully parallel 2×2 MIMO detector architecture that uses 2 separate cores to detect the two streams. Since the detection algorithm in this case is optimal, the post LLR processing stage simply implements (9) and (11), without the need for distance buffering and accumulation.

Fig. 8 shows a 4-sided fully parallel 4×4 MIMO detector that uses 4 cores to process the 4 streams. Here distance buffering and accumulation are needed before LLR processing

in order to adjust the individual LLRs according to (43). In this case, the WLD matrix inputs for all 4 streams using the decompositions in (34) are needed. If chip area is the constraining factor, a MIMO detector can be built using a single core that is time-multiplexed among the 4 streams.

VI. APPLICATION TO MU-MIMO DETECTION

Multi-user MIMO (MU-MIMO) has been proposed as a method for increasing the capacity of wireless networks [54], [55]. In MU-MIMO, multiple users are scheduled on the same physical resource blocks (PRBs). Several receiver processing methods have been proposed in the literature for MU-MIMO [55]–[58]. We consider an optimal MU-MIMO detection method based on the joint constellation estimation of the interfering user and data detection. The optimal MU-MIMO detector can be *efficiently* implemented with a slight modification of the MAP MIMO detector developed in Section III.

A. MU-MIMO System Model

We consider a practical OFDM-based MU-MIMO system where 2 users are co-scheduled on the same PRBs, and each UE has 2 receive antennas. Let K be the number of tones in each PRB. Also let user 1 denote the user of interest with *known* constellation \mathcal{X}_S , while user 2 denotes the interfering user whose constellation \mathcal{X}_I is *unknown* to user 1's receiver. The received frequency-domain complex signal vector $\mathbf{y}[k] \in \mathcal{C}^{2 \times 1}$ at the UE of interest on the k th resource element (RE) over which the 2 users are scheduled is given by

$$\begin{aligned} \mathbf{y}[k] &= \mathbf{H}[k]\mathbf{x}[k] + \mathbf{n}[k] \\ &= \mathbf{h}_1[k]x_1[k] + \mathbf{h}_2[k]x_2[k] + \mathbf{n}[k], \quad k = 1, \dots, K, \end{aligned}$$

where $\mathbf{H}[k] = [\mathbf{h}_1[k] \mathbf{h}_2[k]] \in \mathcal{C}^{2 \times 2}$ is the complex channel matrix with $\mathbf{h}_1[k]$ and $\mathbf{h}_2[k]$ representing the cascade of the channel and precoders of user 1 and user 2, respectively; $\mathbf{x}[k] = [x_1[k] \ x_2[k]]^T$ denotes the transmitted 2×1 QAM symbol vector where $x_1[k] \in \mathcal{X}_S$, $x_2[k] \in \mathcal{X}_I$; and $\mathbf{n}[k] \in \mathcal{C}^{2 \times 1}$ is the noise vector at the k th RE modeled as a zero-mean complex Gaussian random vector with variance σ^2 .

B. ML MU-MIMO Detection

The maximum likelihood estimate of the constellation of the interfering user based on $\mathbf{y}[1], \dots, \mathbf{y}[K]$ is given by

$$\hat{\mathcal{X}}_I = \arg \max_{\mathcal{X}_I \in \mathcal{M}} p(\{\mathbf{y}[k]\}_{k=1}^K | \{\mathbf{H}[k]\}_{k=1}^K, \mathcal{X}_S, \mathcal{X}_I),$$

where K is the number of REs over which \mathcal{X}_I is constant, and

$$\mathcal{M} \triangleq \{4\text{-QAM}, 16\text{-QAM}, 64\text{-QAM}, 256\text{-QAM}\},$$

denotes the set of allowable constellations for the interferer. Assuming that $x_1[k], x_2[k], \mathbf{n}[k]$ are independent for all $k = 1, \dots, K$, the ML estimate of the interferer's constellation can then be written as

$$\begin{aligned} \hat{\mathcal{X}}_I &= \arg \max_{\mathcal{X}_I \in \mathcal{M}} \frac{1}{|\mathcal{X}_I|^K} \\ &\times \prod_{k=1}^K \sum_{x_1[k] \in \mathcal{X}_S} \times \sum_{x_2[k] \in \mathcal{X}_I} p(\mathbf{y}[k] | \mathbf{H}[k], \mathcal{X}_S, \mathcal{X}_I, x_1[k], x_2[k]), \end{aligned} \quad (44)$$

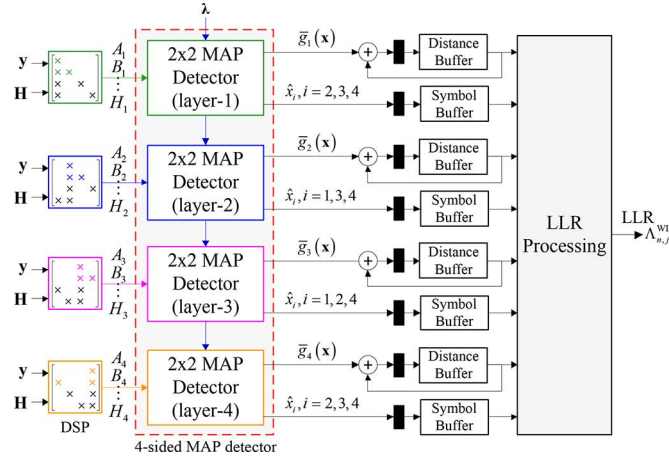


Fig. 8. Block diagram of 4-sided MAP detector.

where $|\mathcal{X}_I|$ denotes the size of the interfering user's constellation, under the assumption that

$$P(x_1[k]) = \frac{1}{|\mathcal{X}_S|}, \text{ and } P(x_2[k]) = \frac{1}{|\mathcal{X}_I|}, \quad k = 1, \dots, K.$$

Let $d(\mathbf{x}[k]) = \|\mathbf{y}[k] - \mathbf{H}[k]\mathbf{x}[k]\|^2/\sigma^2$, we can then write (44) as

$$\hat{\mathcal{X}}_I = \arg \max_{\mathcal{X}_I \in \mathcal{M}} \frac{1}{|\mathcal{X}_I|^K} \prod_{k=1}^K \sum_{\mathbf{x}[k] \in \mathcal{X}_S \times \mathcal{X}_I} \exp(-d(\mathbf{x}[k])).$$

Using the log-max approximation [59], we can approximate the ML estimate $\hat{\mathcal{X}}_I$ by [60]

$$\hat{\mathcal{X}}_I \approx \arg \min_{\mathcal{X}_I \in \mathcal{M}} \left(K \log(|\mathcal{X}_I|) + \sum_{k=1}^K \min_{\mathbf{x}[k] \in \mathcal{X}_S \times \mathcal{X}_I} d(\mathbf{x}[k]) \right), \quad (45)$$

where $\log(\cdot)$ is the natural logarithmic function.

Once the co-scheduled user's constellation, $\hat{\mathcal{X}}_I$, is estimated, then the LLR of the j th bit of the desired user QAM symbol $x_1[k]$ on the k th RE is given by [44]

$$\Lambda_{k,j}^{\text{MLL}} \approx \min_{\substack{x_1[k] \in \mathcal{X}_{S,j}^{(+1)} \\ x_2[k] \in \mathcal{X}_I}} d(\mathbf{x}[k]) - \min_{\substack{x_1[k] \in \mathcal{X}_{S,j}^{(-1)} \\ x_2[k] \in \mathcal{X}_I}} d(\mathbf{x}[k]), \quad (46)$$

where $\mathcal{X}_{S,j}^{(+1)} = \{x \in \mathcal{X}_S : b_j = +1\}$ and $\mathcal{X}_{S,j}^{(-1)} = \{x \in \mathcal{X}_S : b_j = -1\}$. As seen from (46), computing the LLRs involves the same distance computations as those needed for the co-scheduler user's constellation estimation in (45). This fact is exploited in the architecture of a joint constellation classifier and data MU-MIMO detector shown in Fig. 9, which uses an optimized one-sided MAP MIMO detector as its core. The MIMO detector processes the received signal $\mathbf{y}[k]$ assuming all 4 possible choices of the interferer's constellation. It generates 4 corresponding lists of minimum distance metrics $d(\mathbf{x}[k])$ and their associated symbol vectors $\mathbf{x}[k]$ for all the $|\mathcal{M}|$ possible hypotheses of the interferer's constellation, with $x_1[k] \in \mathcal{X}_S$.

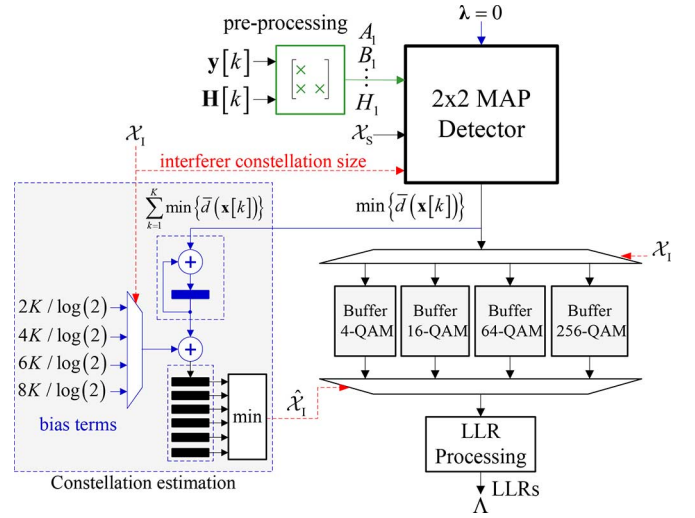


Fig. 9. Block diagram of a MU-MIMO detector.

These distances and symbols are stored in 4 buffers each of size $|\mathcal{X}_S|$ as shown in Fig. 9.

For each tone, the minimum distance from each list is passed to an adder that accumulates the minimum distances over a span of K tones, during which the interferer modulation is assumed to be static. The resulting 4 minimum accumulated distances for each interferer hypothesis are stored in a buffer. The minimum from this buffer is used to identify the interferer's constellation, and the corresponding stored distances in the buffers are selected and forwarded for LLR processing according to (46).

Note that since the interferer's modulation constellation remains static over K tones for a duration of 1 subframe in LTE (14 OFDM symbols), the particular choice of $K = 12$ results in substantial savings in computations. The detector only needs to run in the above mode to identify the interferer's constellation for one OFDM symbol in the subframe. It can then switch back to normal ML detection mode (without modulation classification) to generate the LLRs for the remaining 13 OFDM symbols for the user of interest $x_1[k]$.

Taking the LTE scenario for hardware complexity analysis, the total number of possible tones in 1 PRB in a subframe is $12 \times 14 = 168$. Of these tones, 28 are reserved for pilots (for cell specific reference signals and for UE specific pilots to support the MU-MIMO transmission mode), and 140 for data. In the hardware architecture of Fig. 9, the total number of distance computations needed to generate the LLRs from these 140 data tones is $(140 + 12 \times 5) \times |\mathcal{X}_S|$. This corresponds to an increase of only 42.86% compared to the distances computed by an ML detector with perfect knowledge of the interferer.

Fig. 10 shows the results when \mathcal{X}_S is 64-QAM, with \mathcal{X}_I being 4-, 16-, and 64-QAM using $K = 24$ resource elements. The plots show that the ML classification method has a 5 dB gain over the basic nulling approach when \mathcal{X}_S is 4-QAM, and 2 dB gain in the case of \mathcal{X}_S being 64-QAM. Therefore, the gain of the ML classification method is largest for small constellation sizes of the desired signal, i.e., the largest gain is attained when the receiver complexity is minimal.

Fig. 11 shows the performance of the joint ML classification and detection method as compared to an ML receiver that has perfect knowledge of the interfering user's constellation. Also

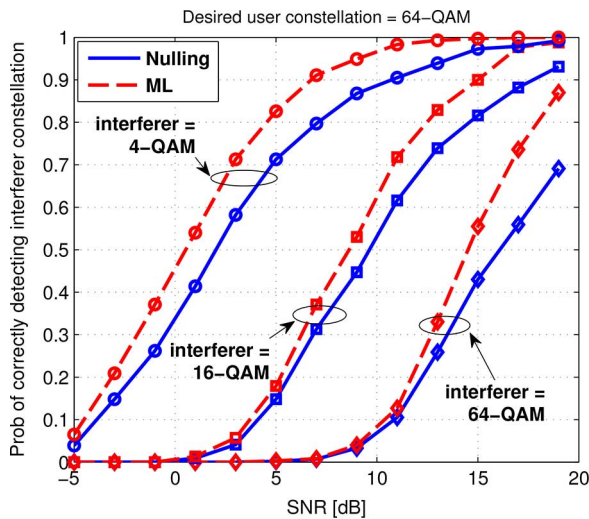


Fig. 10. Probability of correct interferer modulation constellation detection versus SNR [60]. Solid lines are for nulling approach, dashed are for the ML approach. Desired user constellation is fixed to 64-QAM and the co-scheduled user constellation is 4-, 16-, and 64-QAM. The channel is i.i.d. block fading.

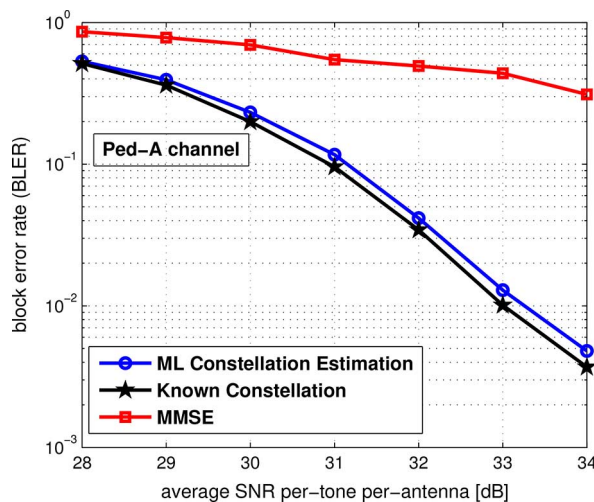


Fig. 11. BLER versus per-tone per-antenna SNR (dB). EPA channel, high correlation (0.9), 64-QAM for both users and code-rate 1/2.

shown in the figure is the performance of the linear MMSE receiver that only uses the knowledge of the interfering user's channel and does not exploit knowledge of the interferer's constellation. Both users use 64-QAM, with the turbo code of [61] and encoding rate 1/2 using block size 6144 bits. The pedestrian-A (Ped-A) [62] multi-path fading channel with high antenna correlation was used. The effective channel matrix is given by $\mathbf{H} = \mathbf{R}_t^{1/2} \mathbf{H}_c \mathbf{R}_r^{1/2}$, where \mathbf{H}_c is channel whose entries are uncorrelated and generated according to the Ped-A model, \mathbf{R}_t and \mathbf{R}_r are the transmit and receive antenna 2×2 correlation matrices, respectively, which have 1 on the diagonal entries and 0.9 on the off-diagonal. As seen from Fig. 11, the joint ML classification and detection receiver is only 0.1 dB away from an ML receiver that has perfect knowledge of the co-scheduled user constellation. The MMSE method has a significant performance degradation as compared to the joint ML classification and detection receiver.

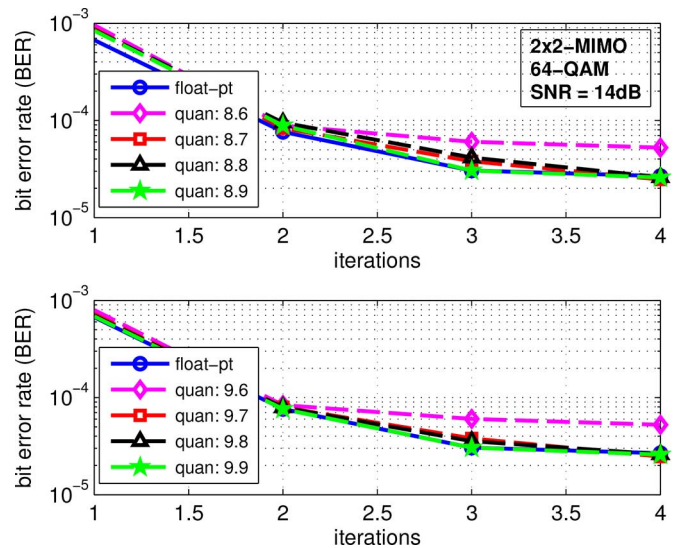


Fig. 12. BER vs. outer detection-decoding iterations for various bit-precisions at SNR = 14 dB for a 2×2 MIMO system with 64-QAM.

VII. IMPLEMENTATION AND SIMULATION RESULTS

The proposed 2×2 reconfigurable MIMO detector architecture was modeled in VHDL and synthesized on a Xilinx Virtex@-6 FPGA. The core was also synthesized using a 90 nm CMOS ASIC library. The experimental simulations below evaluate the coded bit-error rate (BER) performance of the proposed detection algorithm and the implemented core, assuming a MIMO system employing either 2 transmit and 2 receive antennas, or 4 transmit and 4 receive antennas. The channel encoder is based on the LTE turbo encoder specification [2] with interleaver length 1024, using 16-QAM, 64-QAM, and 256-QAM modulation constellations. The channel entries are assumed to be i.i.d. complex Gaussian random variables with unit variance. At the receiver end, we assume perfect channel knowledge. The turbo decoder implements the true *A Posteriori* Probability algorithm, and performs 4 full decoding iterations. Also, the detector and turbo decoder perform up to 4 outer joint detection and decoding iterations. Channel decomposition is performed externally by a pre-processing stage and the coefficients in (18)–(19) are fed as input.

A. Performance Results

The bit-precision of the detector architecture can be configured to enable tradeoff analysis between gate complexity and tolerable degradation in BER performance due to quantization noise. Fig. 12 compares the BER performance of the detector core for 2 layers and 64-QAM under various integer and fractional bit-widths, versus floating-point performance, at SNR = 14 dB. The x-axis denotes the number of joint detection and decoding iterations. The top figure corresponds to a fixed-point representation of $(I, F) = \{8.6, 8.7, 8.8, 8.9\}$, where I denotes integer bit-precision while F denotes fractional bit-precision. The bottom figure corresponds to the representation of $(I, F) = \{9.6, 9.7, 9.8, 9.9\}$. As can be seen, when F starts to drop to 6, the BER starts to degrade. There is no significant improvement in BER in going beyond $I = 9$ integer bits, as demonstrated also in Fig. 13.

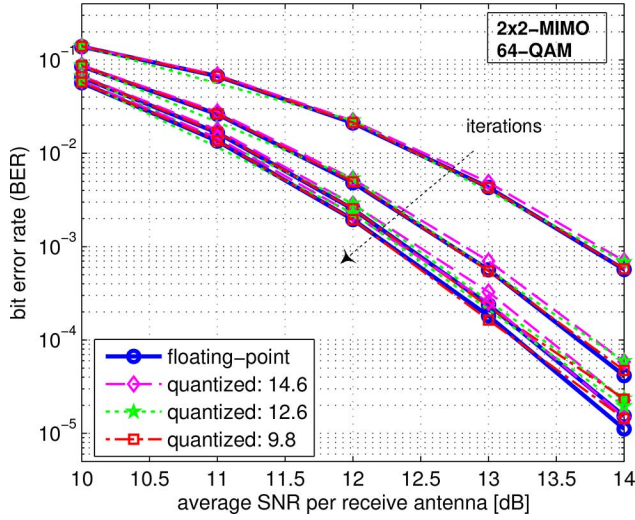


Fig. 13. BER vs. SNR for various bit-precisions and up to 4 outer detection-decoding iterations, for a 2×2 MIMO system with 64-QAM.

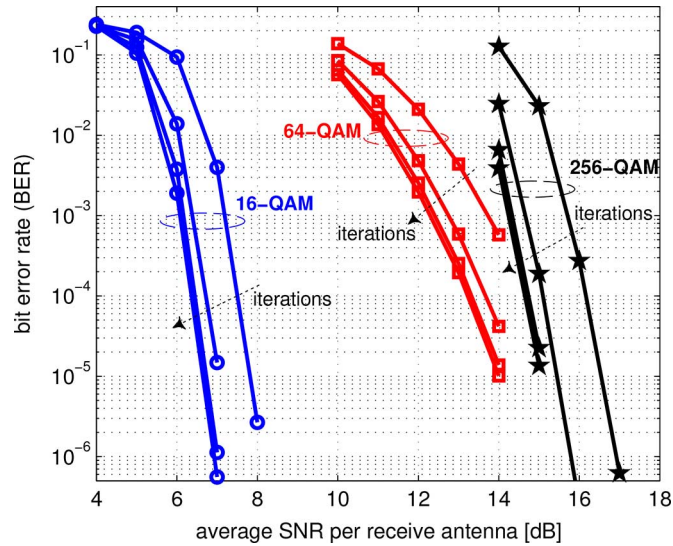


Fig. 14. BER vs. SNR for a 2×2 MIMO system with 16-, 64-, and 256-QAM.

Fig. 14 compares the BER performance of the core using 16-QAM, 64-QAM, and 256-QAM. The plots demonstrate that most of the coding gain is attained after 3 outer iterations, assuming the inner turbo decoder performs at most 4 full turbo decoding iterations.

In Figs. 15 and 16, the BER performance of a 4×4 MIMO system using the proposed WLD scheme is simulated. In Fig. 15, the plots compare the BER versus SNR of the proposed WLD scheme with $E = 1$ and 2 structures (Fig. 3(a)–(b)), versus ML, zero-forcing (ZF), the approach of [47], and the sphere decoder with radius clipping [30], for 16-QAM. Both overlapping and non-overlapping subsets are considered. Two scenarios for distance computations in (32) are followed; one based on \mathbf{H} and one on \mathbf{L} . The plots demonstrate that WLD with $E = 2$ using \mathbf{H} distances with overlapping subsets performs virtually as ML, and is less than 0.1 dB away from ML with no overlapping. Also, for single streams, \mathbf{L} distances perform better than \mathbf{H} distances. The plots correspond to one

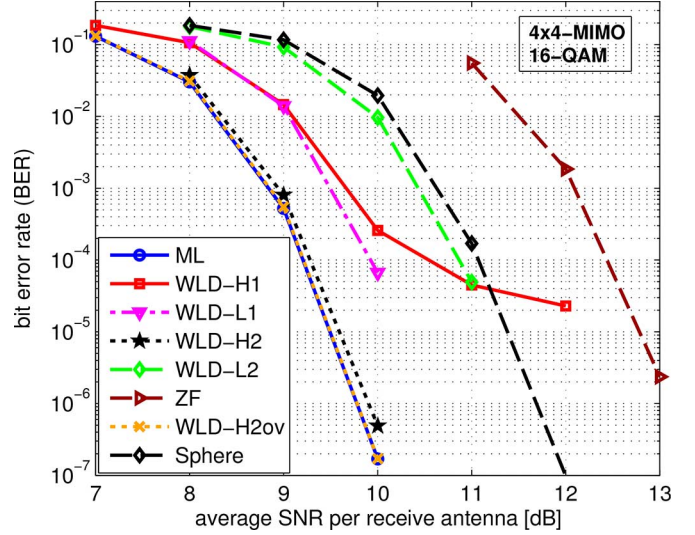


Fig. 15. BER vs. SNR plots for a 4×4 MIMO system with 16-QAM.

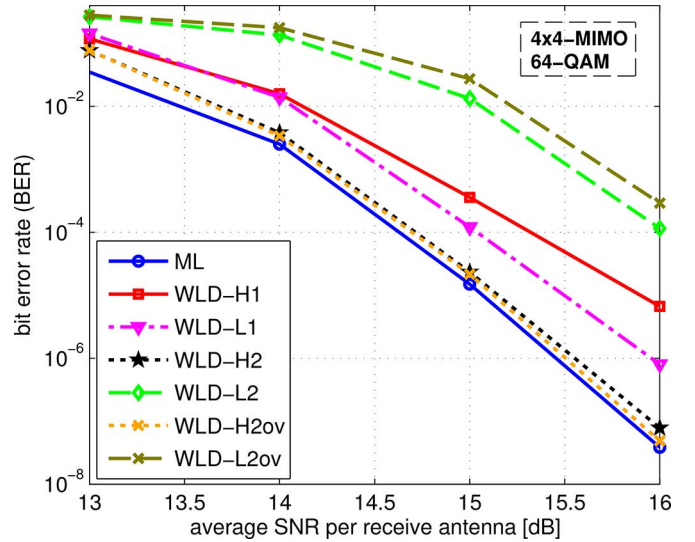


Fig. 16. BER vs. SNR plots for a 4×4 MIMO system with 64-QAM.

outer detection-decoding iteration, and 4 full internal turbo decoder iterations.

Fig. 16 compares the BER performance for 64-QAM. The plots demonstrate again that the WLD scheme with $E = 2$ using \mathbf{H} distances and overlapping subsets performs very close to ML. Fig. 17 shows the results for 256-QAM.

B. Architecture Synthesis Results

Various architecture configurations for the 2×2 core with different algorithmic features and architectural optimizations were synthesized, assuming 17-bit datapaths. The datapaths are pipelined with 6 stages and clocked at 275 MHz. The input LLRs fed from the turbo decoder are 8 bits wide. The output LLRs from the detector are passed to a dynamic scaling block (not included in this work) that scales the bit-widths down to 8 bits before feeding them to the turbo decoder.

Fig. 18 shows the gate complexity of 8 different architectures. Four architectures support reconfigurable constellations up to 64-QAM, while the other four support up to 256-QAM.

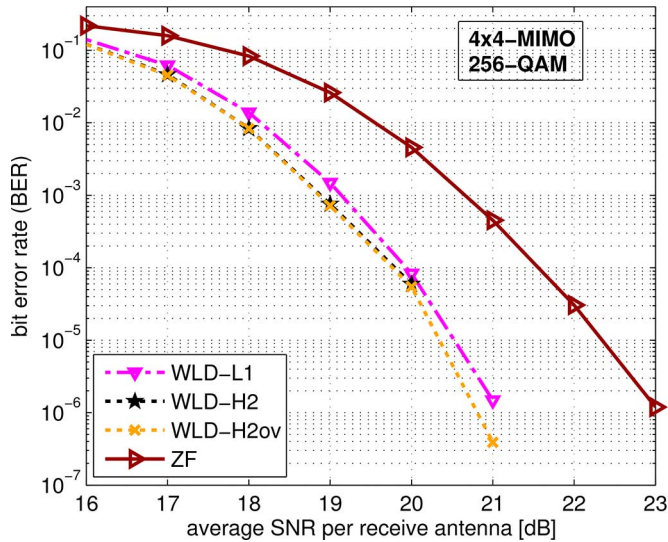
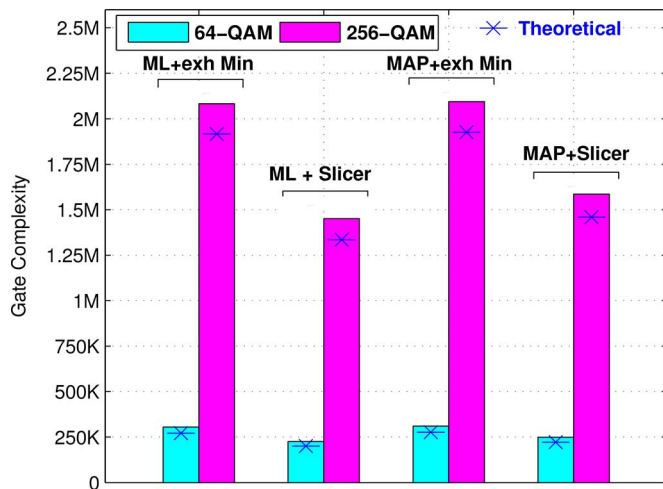
Fig. 17. BER vs. SNR plots for a 4×4 MIMO system with 256-QAM.

Fig. 18. Hardware complexity of various synthesized detector cores.

For the 64-QAM case, two architectures are designed to support soft-outputs only without soft-inputs (i.e., ML detection, see Section III.B): one based on distance minimizations using exhaustive search (Section V.B), and one based on minimization via slicing (Section V.C). The other two 64-QAM architectures support both soft-outputs and soft-inputs (i.e., MAP detection, see Section III.C), one with minimization based on exhaustive search and one via slicing. The other four 256-QAM architectures are similar. All architectures have the same input/output interfaces, external buffers, and control logic. The reported gate counts in gate-equivalent (GE) are for the core logic only.

The plots demonstrate that there is a significant increase in complexity (between 6.35x–6.82x) when supporting 256-QAM compared to 64-QAM. Furthermore, the slicer-based architectures using the proposed scheme in Section V.C offer significant reduction in complexity compared to distance minimization by search (between 19.58%–26.22% for 64-QAM, and between 24.28%–30.35% for 256-QAM). Finally, for slicer-based architectures, supporting soft-inputs for MAP detection comes with an increase in gate count between 8.49%–9.83% compared to soft-output-only ML detection. For minimization-by-search

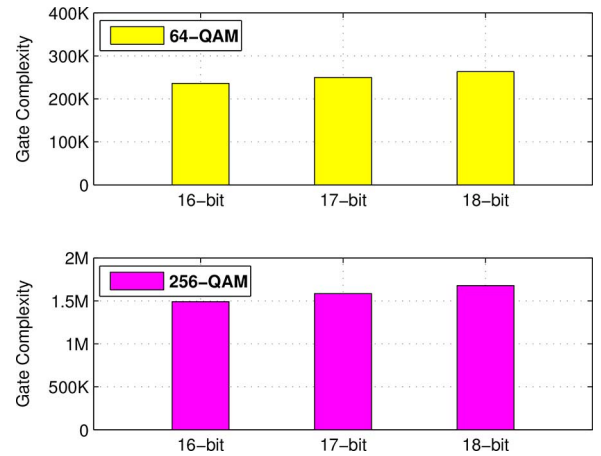


Fig. 19. Hardware complexity as a function of bit-width.

TABLE V
SUMMARY AND COMPARISON OF IMPLEMENTATION RESULTS

Reference	This work	[11]	[33]	[35]	[36]
Antennas	$\leq 4 \times 4$	$\leq 4 \times 4$	$\leq 4 \times 4$	$\leq 4 \times 4$	4×4
Modulation [QAM]	≤ 256	≤ 64	≤ 64	16	64
Algorithm	WLD	MMSE -PIC	STS-SD	Trellis search	FSD
Iterative	YES	YES	YES	YES	YES
Technology [nm]	90	90	90	65	90
Core Area [kGE] ^a	1580	410 ^b	212	1097	555
Clock freq. [MHz]	275	568	193	320	370
Maximum Throughput [Mbps]	2200 (2×2) 733 (4×4)	757	772	1200 ^c	2200
Normalized hardware efficiency [kGE/Mbps]	0.72 (2×2) 2.16 (4×4)	0.54	0.28	0.91	0.25
Power consumption in [mW] @ [Mbps]	320.56 @ 733	189.1 @ 757	87.62 @ 772	—	335.8 @ 2200
Energy efficiency in [nJ/bit]	0.44	0.25	0.11	—	0.15

^a One gate-equivalent corresponds to a 2-input drive-1 NAND gate.

^b Includes preprocessing circuitry.

^c Technology scaling to 90 nm CMOS technology according to $A \sim 1/s$, $t_{pd} \sim 1/s$, and $P_{dyn} \sim (1/s)(V_{dd}/V_{dd}')$ [11].

architectures, the overhead of supporting soft-inputs is only between 0.51%–1.71%. The gate counts predicted by the theoretical analyses in Section V are also plotted in Fig. 18. The error ranges between 8%–11%, which asserts the validity of the model used and the theoretical analysis performed.

Fig. 19 plots the gate complexity of the slicer-based MAP cores as a function of bit-width. The complexity increases roughly between 5.2%–5.9% for every added bit. A similar trend was observed when synthesizing the 256-QAM core with soft-outputs only on a Virtex-6 FPGA. The area increases from 317937 LUTs (33%) for 18 bits to 337210 LUTs (35%) for 19 bits. The area jumps to 403498 LUTs (42%) when the integer bit-width is increased to 12 bits.

The core achieves an average SNR-independent throughput of 2.2 Gbps for 2-layers with 256-QAM, when running in soft-input soft-output mode. In 4×4 mode, the core achieves a throughput of 733 Mbps and consumes 320.56 mW of power. This compares favorably with other detectors in the literature with throughput ranging from 757 Mbps at 410 kGE [11]; 772 Mbps at 212 kGE [33]; 1.2 Gbps at 1097 kGE for 16-QAM [35]; and 2.2 Gbps at 555 kGE [36] for up to 64-QAM only. Table V provides a comparative summary of our implemented detector and the detectors in [11], [33], [35], [36].

VIII. CONCLUSION

A configurable 2-layer soft-input soft-output MIMO detector core has been proposed as a basic building block for constructing detectors with more spatial streams. Optimizations targeting distance computations and slicing operations reduce the overall complexity when supporting constellations up to 256-QAM. By appropriately decomposing the MIMO channel, multi-layer detection is casted in terms of multiple parallel 2-layer detection problems, which can be mapped onto the 2-layer core. Various architectures have been developed to achieve a high target detection throughput. The proposed core has been applied as well to the design an optimal MU-MIMO detector for LTE. The core occupies an area of 1.58 MGE and achieves a throughput of 733 Mbps with 320.56 mW of power for 256-QAM when synthesized in 90 nm CMOS. Future work will target expanding the core to handle 1024-QAM.

REFERENCES

- [1] *IEEE Draft Standard—Part 11: Wireless LAN Medium Access Control and Physical Layer Specifications—Amendment 4: Enhancements for Very High Throughput for Operation in Bands Below 6 GHz*, IEEE Std. P802.11ac/D7.0, Dec. 2013 [Online]. Available: <http://www.ieee.org>
- [2] *Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation*, 3GPP Std. TS 36.211 [Online]. Available: <http://www.3gpp.org>
- [3] A. Paulraj, R. Nabar, and D. Gore, *Introduction to Space-Time Wireless Communications*. Cambridge, U.K.: Cambridge Univ. Press, 2003.
- [4] G. B. Giannakis *et al.*, *Space-Time Coding for Broadband Wireless Communications*. New York, NY, USA: Wiley, 2006.
- [5] E. Biglieri *et al.*, *MIMO Wireless Communications*. Cambridge, U.K.: Cambridge Univ. Press, 2007.
- [6] C. Oestges and B. Clerckx, *MIMO Wireless Communications*. Oxford, U.K.: Elsevier Academic, 2007.
- [7] A. Chockalingam and B. S. Rajan, *Large MIMO Systems*. Cambridge: Cambridge Univ. Press, 2014.
- [8] B. Hassibi, "An efficient square-root algorithm for BLAST," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Istanbul, Turkey, Jun. 2000, pp. 5–9.
- [9] G. D. Golden *et al.*, "Detection algorithm and initial laboratory results using V-BLAST space-time communication architecture," *IEE Electron. Lett.*, vol. 35, no. 1, pp. 14–15, Jan. 1999.
- [10] D. Wübben, R. Böhnke, V. Kühn, and K. Kammeyer, "MMSE extension of V-BLAST based on sorted QR decomposition," in *Proc. IEEE Veh. Technol. Conf. (VTC)*, Orlando, FL, USA, Oct. 2003, pp. 508–512.
- [11] C. Studer, S. Fateh, and D. Seethaler, "ASIC implementation of soft-input soft-output MIMO detection using MMSE parallel interference cancellation," *IEEE Trans. Syst. Sci. Cybern.*, vol. 47, no. 7, pp. 1754–1765, Jul. 2011.
- [12] E. Viterbo and E. Biglieri, "A universal decoding algorithm for lattice codes," in *Proc. 14 ème Colloque GRETSI*, Juan-Les-Pins, France, Sep. 1993, pp. 611–614.
- [13] E. Viterbo and J. Boutros, "A universal lattice code decoder for fading channels," *IEEE Trans. Inf. Theory*, vol. 45, no. 5, pp. 1639–1642, Jul. 1999.
- [14] O. Damen, A. Chkeif, and J.-C. Belfiore, "Lattice code decoder for space-time codes," *IEEE Commun. Lett.*, vol. 4, no. 5, pp. 161–163, May 2000.
- [15] E. Agrell *et al.*, "Closest point search in lattices," *IEEE Trans. Inf. Theory*, vol. 48, no. 8, pp. 2201–2214, Aug. 2002.
- [16] B. Hochwald and S. Ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Trans. Commun.*, vol. 51, no. 3, pp. 389–399, Mar. 2003.
- [17] B. Hassibi and H. Vikalo, "On sphere decoding algorithm. I. Expected complexity," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 2806–2818, Aug. 2005.
- [18] 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.
- [19] D. Seethaler, J. Jaldén, C. Studer, and H. Bölcskei, "On the complexity distribution of sphere decoding," *IEEE Trans. Inf. Theory*, vol. 57, no. 9, pp. 5754–5768, Sep. 2011.
- [20] K.-W. Wong, C.-Y. Tsui, R. S.-K. Cheng, and W.-H. Mow, "A VLSI architecture of a K -best lattice decoding algorithm for MIMO channels," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Scottsdale, AZ, USA, May 2002, vol. 3, pp. 273–276.
- [21] M. Wenk *et al.*, "K-best MIMO detection VLSI architectures achieving up to 424 Mbps," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Island of Kos, Greece, May 2006, pp. 1151–1154.
- [22] S. Mondal, A. Eltawil, C.-A. Shen, and K. Salama, "Design and implementation of a sort free K -best sphere decoder," *IEEE Trans. VLSI Syst.*, vol. 18, no. 10, pp. 1497–1501, Oct. 2010.
- [23] L. Liu, F. Ye, X. Ma, T. Zhang, and J. Ren, "A 1.1-Gb/s 115-pJ/bit configurable MIMO detector using 0.13 μm CMOS technology," *IEEE Trans. Circuits Syst. II*, vol. 57, no. 9, pp. 701–705, Sep. 2010.
- [24] C.-A. Shen, A. Eltawil, and K. Salama, "Evaluation framework for K -best sphere decoders," *J. Circuits, Syst. Comput.*, vol. 19, no. 5, pp. 975–995, Aug. 2010.
- [25] M. Shabany and P. Gulak, "A 675 Mbps, 4×4 64-QAM K -Best MIMO detector in 0.13 μm CMOS," *IEEE Trans. VLSI Syst.*, vol. 20, no. 1, pp. 135–147, Jan. 2012.
- [26] M. Mahdavi and M. Shabany, "Novel MIMO detection algorithm for high-order constellations in the complex domain," *IEEE Trans. VLSI Syst.*, vol. 21, no. 5, pp. 834–847, May 2013.
- [27] D. Garrett *et al.*, "Silicon complexity for maximum likelihood MIMO detection using spherical decoding," *IEEE J. Solid-State Circuits*, vol. 39, no. 9, pp. 1544–1552, Sep. 2004.
- [28] Z. Guo and P. Nilsson, "A VLSI architecture of the Schnorr-Euchner decoder for MIMO systems," in *Proc. IEEE CAS Symp. Emerg. Technol.*, Shanghai, China, May 2004, vol. 1, pp. 65–68.
- [29] A. Burg *et al.*, "VLSI implementation of MIMO detection using the sphere decoding algorithm," *IEEE J. Solid-State Circuits*, vol. 40, no. 7, pp. 1566–1577, Jul. 2005.
- [30] C. Studer, A. Burg, and H. Bölcskei, "Soft-output sphere decoder: Algorithms and VLSI implementation," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 2, pp. 290–300, Feb. 2008.
- [31] C.-H. Yang and D. Markovic, "A flexible DSP architecture for MIMO sphere decoding," *IEEE Trans. Circuits Syst. I*, vol. 56, no. 10, pp. 2301–2314, Oct. 2009.
- [32] C.-H. Yang and D. Markovic, "A 2.89 mW 50 GOPS 16×16 16-core MIMO sphere decoder in 90 nm CMOS," in *Proc. Eur. Solid-State Circuits Conf. (ESSCIRC)*, Athens, Greece, Sep. 2009, pp. 344–347.
- [33] F. Borlenghi *et al.*, "A 772 Mbit/s 8.81 bit/nJ 90 nm CMOS soft-input soft-output sphere decoder," in *Proc. IEEE Asian Solid State Circuits Conf. (A-SSCC)*, Jeju, Korea, Nov. 2011, pp. 297–300.
- [34] L. Liu, J. Lofgren, and P. Nilsson, "Area-efficient configurable high-throughput signal detector supporting multiple MIMO modes," *IEEE Trans. Circuits Syst. I*, vol. 59, no. 9, pp. 2085–2096, Sep. 2012.
- [35] Y. Sun and J. R. Cavallaro, "Trellis-search based soft-input soft-output MIMO detector: Algorithm and VLSI architecture," *IEEE Trans. Signal Process.*, vol. 60, no. 5, pp. 2617–2627, May 2012.
- [36] X. Chen, G. He, and J. Ma, "VLSI implementation of a high-throughput iterative fixed-complexity sphere decoder," *IEEE Trans. Circuits Syst. II*, vol. 60, no. 5, pp. 272–276, May 2013.
- [37] M. M. Mansour, S. Alex, and M. Jalloul, "Reduced complexity soft-output MIMO sphere detectors—Part I: Algorithmic optimizations," *IEEE Trans. Signal Process.*, vol. 62, no. 21, pp. 5505–5520, Nov. 2014.
- [38] M. M. Mansour, S. Alex, and M. Jalloul, "Reduced complexity soft-output MIMO sphere detectors—Part II: Architectural optimizations," *IEEE Trans. Signal Process.*, vol. 62, no. 21, pp. 5521–5535, Nov. 2014.
- [39] M.-Y. Huang and P.-Y. Tsai, "Toward multi-gigabit wireless: Design of high-throughput MIMO detectors with hardware-efficient architecture," *IEEE Trans. Circuits Syst. I*, vol. 61, no. 2, pp. 613–624, Feb. 2014.
- [40] Y. Jiang, J. Li, and W. W. Hager, "Joint transceiver design for MIMO communications using geometric mean decomposition," *IEEE Trans. Signal Process.*, vol. 53, no. 10, pp. 3791–3803, Oct. 2005.
- [41] Y. Jiang, J. Li, and W. W. Hager, "Uniform channel decomposition for MIMO communications," *IEEE Trans. Signal Process.*, vol. 53, no. 11, pp. 4283–4294, Nov. 2005.
- [42] S. Ariyavisitakul, J. Zheng, E. Ojard, and J. Kim, "Subspace beamforming for near-capacity MIMO performance," *IEEE Trans. Signal Process.*, vol. 56, no. 11, pp. 5729–5733, Nov. 2008.

- [43] Y. Chen and S. Brink, "Near-capacity MIMO subspace detection," in *Proc. IEEE Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, Toronto, Canada, Sept. 2011, pp. 1733–1737.
- [44] 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)*, Istanbul, Turkey, Jun. 2006, vol. 4, pp. 1686–1691.
- [45] M. Siti and M. P. Fitz, "On layer ordering techniques for near-optimal MIMO detectors," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Hong Kong, Mar. 2007, pp. 1199–1204.
- [46] M. S. Yee, "Max-log-MAP sphere decoder," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Philadelphia, PA, USA, Mar. 2005, vol. 3, pp. 1013–1016.
- [47] E. Ojard and S. Ariyavisitakul, "Method and system for approximate maximum likelihood (ML) detection in a multiple input multiple output (MIMO) receiver" US Patent 12/207,721, Mar. 19, 2009 [Online]. Available: <http://www.google.com/patents/US20090074114>
- [48] L. A. C. Zhang and T. Meixia, "LTE-advanced and 4G wireless communications [guest editorial]," *IEEE Commun. Mag.*, vol. 50, no. 2, pp. 102–103, Feb. 2012.
- [49] A. Gomaa and L. Jalloul, "Efficient soft-input soft-output detection of dual-layer MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 3, no. 5, pp. 541–544, Oct. 2014.
- [50] C. K. Yeung, J. Lee, and S. Kim, "A simple slicer for soft detection in gray-coded QAM-modulated MIMO OFDM systems," in *Proc. IEEE 35th Sarnoff Symp. (SARNOFF)*, Newark, NJ, USA, May 2012, pp. 1–5.
- [51] M. M. Mansour, "A near-ML MIMO subspace detection algorithm," *IEEE Signal Process. Lett.*, vol. 22, no. 4, pp. 408–412, Apr. 2015.
- [52] G. H. Golub and C. F. V. Loan, *Matrix Computations*, 3rd ed. Baltimore, MD, USA: Johns Hopkins Univ. Press, 1996.
- [53] CEVA-XC4210 DSP Processors [Online]. Available: <http://www.ceva-dsp.com/CEVA-XC4210>
- [54] J. Lee, J.-K. Huan, and J. Zhang, "MIMO technologies in 3GPP LTE and LTE-Advanced," *EURASIP J. Wireless Commun. Netw.*, vol. 2009, no. 1, pp. 1–10, 2009.
- [55] J. Duplity *et al.*, "MU-MIMO in LTE systems," *EURASIP J. Wireless Commun. Netw.*, vol. 2011, no. 1, pp. 1–13, 2011.
- [56] Z. Bai *et al.*, "On the equivalence of MMSE and IRC receiver in MU-MIMO systems," *IEEE Commun. Lett.*, vol. 15, no. 12, pp. 1288–1290, Dec. 2011.
- [57] R. Ghaffar and R. Knopp, "Interference sensitivity for multiuser MIMO in LTE," in *Proc. IEEE Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jun. 2011, pp. 506–510.
- [58] R. Ghaffar and R. Knopp, "Interference-aware receiver structure for multiuser MIMO and LTE," *EURASIP J. Wireless Commun. Netw.*, vol. 40, pp. 1–17, 2011.
- [59] A. Viterbi, "An intuitive justifications and a simplified implementation of the MAP decoder for convolutional codes," *IEEE J. Sel. Areas Commun.*, vol. 16, pp. 260–264, Feb. 1998.
- [60] A. Gomaa *et al.*, "Multi-user MIMO receivers with partial state information," *IEEE Trans. Veh. Technol.*, Jan. 2015, submitted for publication.
- [61] *Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and Channel Coding*, 3GPP Std. TS 36.212 [Online]. Available: <http://www.3gpp.org>
- [62] *High Speed Downlink Packet Access: UE Radio Transmission and Reception FDD*, 3GPP Std. TR 25.890 [Online]. Available: <http://www.3gpp.org>



Mohammad M. Mansour (S'97–M'03–SM'08) received the B.E. (Hons.) and M.E. degrees in computer and communications engineering from the American University of Beirut (AUB), Beirut, Lebanon, in 1996 and 1998, respectively, and the M.S. degree in mathematics and the Ph.D. degree in electrical engineering from the University of Illinois at Urbana Champaign (UIUC), Champaign, IL, USA, in 2002 and 2003, respectively.

He was a Visiting Researcher at Broadcom, Sunnyvale, CA, USA, from 2012 to 2014, where he worked on the physical layer SoC architecture, and algorithm development for LTE-Advanced. He was on research leave with Qualcomm Flarion Technologies, Bridgewater, NJ, USA, from 2006 to 2008, where he worked on modem design and implementation for 3GPP-LTE, 3GPP2-UMB, and peer-to-peer wireless networking physical layer SoC architecture and algorithm development. He was a Research Assistant at the Coordinated Science Laboratory (CSL), UIUC, from 1998 to 2003. He was with the National Semiconductor Corporation, San Francisco, CA, with the Wireless Research group in 2000. He was a Research Assistant with the Department of Electrical and Computer Engineering, AUB, in 1997, and a Teaching Assistant in 1996. He joined the Department of Electrical and Computer Engineering, AUB, in 2003, as a faculty member, and is currently an Associate Professor. His research interests are in the area of energy-efficient and high-performance VLSI circuits, architectures, algorithms, and systems for computing, communications, and signal processing. He has six issued U.S. patents.

Prof. Mansour is a member of the Design and Implementation of Signal Processing Systems (DISPS) Technical Committee Advisory Board of the IEEE Signal Processing Society. He served as a member of the DISPS Technical Committee from 2006 to 2013. He served as an Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II (TCAS-II) from 2008 to 2013. He currently serves as an Associate Editor of the IEEE TRANSACTIONS ON VLSI SYSTEMS since 2011, and an Associate Editor of the IEEE SIGNAL PROCESSING LETTERS since 2012. He served as the Technical Co-Chair of the IEEE Workshop on Signal Processing Systems in 2011, and as a member of the Technical Program Committee of various international conferences and workshops. He was the recipient of the PHI Kappa PHI Honor Society Award twice in 2000 and 2001, and of the Hewlett Foundation Fellowship Award in 2006.



Louay M.A. Jalloul (M'91–SM'00) received the B.S. degree from the University of Oklahoma, Norman, USA, in 1985; the M.S. degree from The Ohio State University, Columbus, USA, in 1988; and the Ph.D. degree from Rutgers, The State University of New Jersey, Piscataway, NJ, USA, in 1993, all in electrical engineering.

He was a Research Associate with the Electro-Science Laboratory, The Ohio State University; and the Wireless Information Networks Laboratory (WINLAB), Rutgers University. He is currently a Technical Director with Broadcom Corporation, Sunnyvale, CA, USA. Prior to that, he was a Senior Director of Technology with Beceem Communications, Inc. (a Silicon Valley startup providing solutions for mobile broadband wireless communication systems). From September 2004 to September 2005, he was an Associate Professor with the Department of Electrical and Computer Engineering, American University of Beirut, Beirut, Lebanon. In February 2001, he joined MorphICs Technology Inc., Campbell, CA (acquired by Infineon Technologies AG in April 2003) as the Director of Systems Architecture, where he led his team in the development of the code-division multiple access (CDMA) cellular digital signal processor for the third-generation wideband CDMA standard. From 1993 to 2001, he was with Motorola Inc., taking on various functions in research and development. He contributed to the early concepts of high-speed downlink packet access and IS-2000 evolution to voice and data (1XEV-DV). He has 57 issued U.S. patents.

Dr. Jalloul has received numerous engineering awards for his innovations to Motorola products. He is a member of Eta Kappa Nu.