ORVD-Trellis based MIMO Detection

Dominik Auras, Rainer Leupers, Gerd Ascheid
Institute for Communication Technologies and Embedded Systems
RWTH Aachen University, 52056 Aachen, Germany
email: auras@ice.rwth-aachen.de

***Abstract*—Non-linear detection for multi-antenna (MIMO) systems using iterative detection and decoding offers superior communications performance at the cost of an increased computational complexity. Various algorithms from literature solve the underlying search problem using quite diverse approaches. Detection based on a Trellis diagram to structure the search has shown close-to-optimal performance, however also a relatively high complexity. This paper presents a novel Trellis based algorithm that achieves a similar communications performance at a significantly reduced complexity, requiring only 6% of the original metric evaluations.**

I. Introduction

Multi-antenna (MIMO) receivers for bit-interleaved coded modulation with iterative decoding and detection (BICM-ID) have received significant research attention. Since optimal detection is prohibitively complex for hardware implementation, various sub-optimal algorithms have been devised. While linear detectors basically perform estimation followed by demapping, non-linear detectors treat the detection problem as a search. Within that class, the detectors differ in the way that they structure the search, e.g. sphere decoders use a depth-first search strategy. The soft-in soft-out (SISO) path-preserving trellis-search (PPTS) based detector presented in [1] uses a Trellis structure representing the search space, thus is a specially structured breadth-first (BF) search. Its communications performance is close to the optimal brute-force max-log detector. Further advantages are the lower local search complexity compared to BF based detection, and that it obtains information on every possibly transmitted symbol vector, which is not guaranteed for traditional BF based detection. However, its complexity depends exponentially on the modulation alphabet size. This leads to a poor implementation efficiency, as evidenced by the very large VLSI area [1].

Contribution

A hardware-friendly trellis-search based soft-in soft-out MIMO detection algorithm for use in a BICM-ID receiver is presented. It features a significantly reduced complexity compared to [1] at a similar communications performance.

Outline

The next section describes the assumed system model. It follows an informal (Sec. III) and then formal (Sec. IV) description of the algorithm. Sec. V describes possible variants. We evaluate the communications performance (Sec. VI) and the complexity (Sec. VII), then compare to [1] (Sec. VIII).

II. System Model

We consider a spatial-multiplexing $N_t \times N_r$ MIMO system with BICM-ID. A message $\boldsymbol{b} \in \{0,1\}^{N_b}$ is encoded with rate $r = N_b/N_c$ and interleaved, yielding the code word $\boldsymbol{c} \in \{0,1\}^{N_c}$. Let $\mathcal{X} \subset \mathbb{R}$ be a modulation alphabet of size $M = |\mathcal{X}|$ with $K = \log_2 |\mathcal{X}|$ bits per symbol. The code word is partitioned into multiple sub-vectors $\boldsymbol{c}_n \in \{0,1\}^{KN_t}$. They are subsequently mapped to symbol vectors $\boldsymbol{x}_n \in \mathcal{X}^{N_t}$ that are transmitted independently. Assuming a frequency-flat fading channel characterized by $\boldsymbol{H}_n \in \mathbb{R}^{N_r \times N_t}$, the received symbol vector at time n is $\tilde{\boldsymbol{y}}_n = \boldsymbol{H}_n \boldsymbol{x}_n + \boldsymbol{w}_n$ where $\boldsymbol{w}_n \in \mathbb{R}^{N_r}$ is a white Gaussian noise process with $\mathbb{E}[\boldsymbol{w}_n \boldsymbol{w}_n^T] = N_0 \boldsymbol{I}_{N_r}$. In the remainder, the time index n is dropped for convenience. Using iterative MIMO decoding following the Turbo Principle, detector and channel decoder exchange extrinsic information $\tilde{\boldsymbol{\lambda}}^e = \tilde{\boldsymbol{\lambda}}^p - \tilde{\boldsymbol{\lambda}}^a$ in terms of log-likelihood ratios (LLRs), usually denoted as soft decision, hence soft-in soft-out (SISO), where $\tilde{\boldsymbol{\lambda}}^p$ are the detector's posterior LLRs and $\tilde{\boldsymbol{\lambda}}^a$ are the prior LLRs fed back from the decoder.

Orthogonal Real-Value Decomposition (ORVD): We assume that the channel matrix $\boldsymbol{H}$ is derived from a complex-valued channel model $\widetilde{\boldsymbol{H}} \in \mathbb{C}^{N_r/2 \times N_t/2}$ such that

$$\boldsymbol{H} = \begin{pmatrix} \mathrm{Re}\{\tilde{h}_{1,P_1}\} & \mathrm{Im}\{\tilde{h}_{1,P_1}\} & \mathrm{Re}\{\tilde{h}_{1,P_2}\} & \cdots \\ -\mathrm{Im}\{\tilde{h}_{1,P_1}\} & \mathrm{Re}\{\tilde{h}_{1,P_1}\} & -\mathrm{Im}\{\tilde{h}_{1,P_2}\} & \cdots \\ \mathrm{Re}\{\tilde{h}_{2,P_1}\} & \mathrm{Im}\{\tilde{h}_{2,P_1}\} & \mathrm{Re}\{\tilde{h}_{2,P_2}\} & \cdots \\ -\mathrm{Im}\{\tilde{h}_{2,P_1}\} & \mathrm{Re}\{\tilde{h}_{2,P_1}\} & -\mathrm{Im}\{\tilde{h}_{2,P_2}\} & \cdots \\ \vdots & \vdots & & \end{pmatrix} \tag{1}$$

which is a structural constraint on $\boldsymbol{H}$. The columns of the matrix $\widetilde{\boldsymbol{H}}$ are sorted before the conversion to $\boldsymbol{H}$ by their l^2-norm in ascending order, where the P_1-th column of $\widetilde{\boldsymbol{H}}$ has the smallest norm. An orthogonal matrix $\boldsymbol{Q} \in \mathbb{R}^{N_r \times N_t}$ with $\boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}$ and an upper-triangular matrix $\boldsymbol{R} \in \mathbb{R}^{N_t \times N_t}$ are computed such that

$$\begin{pmatrix} \boldsymbol{Q} & \boldsymbol{Q}_b \\ \boldsymbol{Q}_c & \boldsymbol{Q}_d \end{pmatrix} \begin{pmatrix} \boldsymbol{R} \\ \boldsymbol{0} \end{pmatrix} = \begin{pmatrix} \boldsymbol{H} \\ N_0 \boldsymbol{I} \end{pmatrix} \tag{2}$$

which is the decomposition of the regularized channel matrix. Note that $\boldsymbol{QR} = \boldsymbol{H}$ holds. The system model becomes

$$\boldsymbol{y} = \boldsymbol{Q}^T \tilde{\boldsymbol{y}} = \boldsymbol{Rx} + \boldsymbol{Q}^T \boldsymbol{w} \tag{3}$$

where $\boldsymbol{y} \in \mathbb{R}^{N_t}$. The input $\boldsymbol{\lambda}^a = N_0 \tilde{\boldsymbol{\lambda}}^a$ and output LLRs $\tilde{\boldsymbol{\lambda}}^e = \boldsymbol{\lambda}^e / N_0$ are scaled and permuted, where both permutations depend on the column sorting P_i of $\widetilde{\boldsymbol{H}}$.

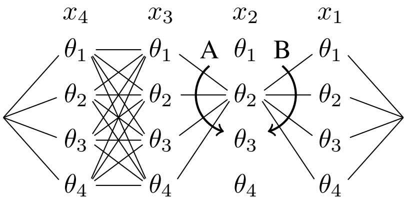

Fig. 1. Trellis structure representing $\boldsymbol{x} \in \mathcal{X}^4$ with $M = 4$. Every stage is fully connected to the next stage (as for x_4 to x_3, some edges not drawn for visual purposes). A is an example for the incoming path selection in PR$_1$, and B for the outgoing path selection in PR$_2$ (cf. Sec. IV, Path Reduction).

In this context, a MIMO system with four complex-valued transmit streams, e.g. four antennas in the complex baseband, offers $N_t = 8$ real-valued transmit layers (two per stream).

III. ORVD-Trellis based MIMO Detection

Fig. 1 shows an exemplary Trellis diagram. There is one Trellis stage per transmit layer x_t. Each stage contains one node per symbol $\theta_m \in \mathcal{X}$. A path from left to right represents one symbol vector $\boldsymbol{x}$. It is attributed with a transmission likelihood given $\boldsymbol{y}$. We want to find L very likely paths through every Trellis node. From these $L \times N_t \times M$ paths, we compute the output LLRs.

A. Definitions

Let $\theta_m \in \mathcal{X}$ denote the m-th symbol of the modulation alphabet, $\varphi_t : \{0,1\}^K \mapsto \mathcal{X}$ be the per-layer mapping rule and φ_t^{-1} its inverse, and $\boldsymbol{x}^{(m,t,l)}$ denote the l-th path going *through* the m-th node at the t-th stage, with $x_i^{(m,t,l)} \in \mathcal{X} \cup \{0\}$ and the constraint $x_t^{(m,t,l)} = \theta_m$. Partial paths are paths with zeros indicating not yet determined entries. To *extend a path* means we replace the highest-index zero.

Attributed to each path is a metric defined as

$$\begin{aligned}\mu(\boldsymbol{x}) &= \|\boldsymbol{y} - \boldsymbol{R}\boldsymbol{x}\|^2 + \boldsymbol{c}^T \boldsymbol{\lambda}^a \\ &= \textstyle\sum_{t=1}^{N_t} \mu_t(\boldsymbol{x})\end{aligned} \tag{4}$$

where the code bit vector $\boldsymbol{c}$ maps to $\boldsymbol{x}$ and the per-layer metric function is defined as

$$\mu_t(\boldsymbol{x}) = \left| y_t - \textstyle\sum_{j=t}^{N_t} r_{t,j} x_j \right|^2 + \varphi_t^{-1}(x_t)^T \boldsymbol{\lambda}_t^a \tag{5}$$

with $\boldsymbol{\lambda}_t^a$ holding the prior LLRs associated with the t-th layer. The most likely path has the lowest metric value.

B. Special Structure of R

The matrix $\boldsymbol{R}$ has a very special structure, e.g. for a system with four transmit layers, we obtain

$$\boldsymbol{R} = \begin{pmatrix} a & 0 & b & c \\ 0 & a & -c & b \\ 0 & 0 & d & 0 \\ 0 & 0 & 0 & d \end{pmatrix} \tag{6}$$

with only four distinct real-valued coefficients. An implementation should exploit this to process two layers in parallel and reduce the number of operations (e.g. multiplications).

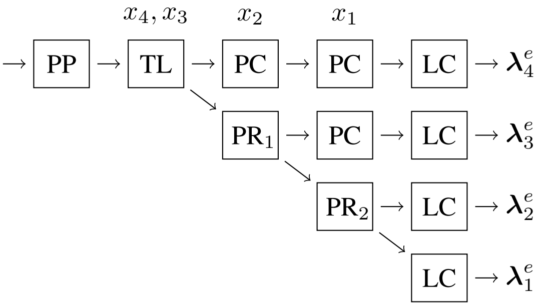
x_4, x_3
x_2
x_1
PP
TL
PC
PC
LC
$\boldsymbol{\lambda}_4^e$
PR$_1$
PC
LC
$\boldsymbol{\lambda}_3^e$
PR$_2$
LC
$\boldsymbol{\lambda}_2^e$
LC
$\boldsymbol{\lambda}_1^e$

Fig. 2. Overview of the algorithm: Preprocessing (PP), Top Layers (TL), Path Completion (PC), Path Reduction (PR) and LLR Computation (LC) process

IV. Algorithm

Fig. 2 depicts an overview of the proposed algorithm. The Preprocessing includes the channel column sorting, ORVD, QR decomposition, LLR scaling and permutation. The Top Layers and Path Reduction processes pass from left to right through the Trellis. Intermediate partial paths generated by these are extended to full length by the Path Completion. Finally, the LLR Computation combines all paths, where only paths through nodes at stage x_t are used to generate the t-th layer's $\boldsymbol{\lambda}_t^e$.

Top Layers: We process the two top layers together. Compute for $t \in \{N_t - 1, N_t\}$ the metric values

$$\mu_t(\theta_m) = |y_t - r_{t,t}\theta_m|^2 + \varphi_t^{-1}(\theta_m)^T \boldsymbol{\lambda}_t^a \tag{7}$$

for all symbols $\theta_m \in \mathcal{X}$. For both t, select the L smallest metrics and their respective symbols $x_t^{(l)}$. Initialize the partial paths for stage N_t with

$$\boldsymbol{x}^{(m,N_t,l)} = \left(0, \ldots, x_{N_t-1}^{(l)}, \theta_m\right)^T \tag{8}$$

and for the next stage $N_t - 1$ with

$$\boldsymbol{x}^{(m,N_t-1,l)} = \left(0, \ldots, \theta_m, x_{N_t}^{(l)}\right)^T . \tag{9}$$

All $\boldsymbol{x}^{(m,N_t,l)}$ go to the Path Completion, and all $\boldsymbol{x}^{(m,N_t-1,l)}$ go to the first Path Reduction.

Path Completion: We want to extend every input partial path $\boldsymbol{x}^{(m,t,l)}$ by one step. Replace the next zero $x_n^{(m,t,l)}$ with a suitable decision, where n is the highest index of the remaining zeros. Cancel the known interference

$$y^{(\mathrm{ic})} = y_n - \sum_{j=n+1}^{N_t} r_{n,j} x_j^{(m,t,l)} \tag{10}$$

and determine the nearest symbol

$$x_{\mathrm{c}} = \underset{x \in \mathcal{X}}{\arg\min} \left\| y^{(\mathrm{ic})} / r_{n,n} - x \right\| . \tag{11}$$

Assemble a bit vector $\boldsymbol{c}_a$ containing the sign bits of $\boldsymbol{\lambda}_n^a$, and map that vector to a $x_{\mathrm{a}} = \varphi_n(\boldsymbol{c}_a)$. For every bit index b in $\boldsymbol{c}_a$, create a copy $\boldsymbol{c}_{a\bar{b}}$ of $\boldsymbol{c}_a$ with the bit at that respective index b inverted, then map it to $x_{\mathrm{a}\bar{b}}$. Subsequently, find

$$x_{\mathrm{ca}} = \underset{x \in \{x_{\mathrm{a}}, x_{\mathrm{a}\bar{b}} \forall b | x \neq x_{\mathrm{c}}\}}{\arg\min} |x - x_{\mathrm{c}}| . \tag{12}$$

This is called *optimized hybrid enumeration* (OHE) in [2].

Select from x_{c} and x_{ca} the symbol that has a lower per-layer metric value μ_n. Extend the current partial path by putting the winning symbol at $x_n^{(m,t,l)}$. If zeros remain, feed the path to the next Path Completion, otherwise to the LLR Computation.

LLR Computation: Once every path at stage t is completed, we first reduce the L paths for every node

$$\mu\left(\boldsymbol{x}^{(m,t)}\right) = \min_{l} \mu\left(\boldsymbol{x}^{(m,t,l)}\right) \tag{13}$$

then compute the extrinsic LLRs by combining the per-symbol metrics for every bit index $b = 1 \dots K$ according to

$$\begin{aligned} \lambda^e_{b+(t-1)K} = & \min_{\boldsymbol{x}^{(m,t)}: x_t^{(m,t)} \in \mathcal{X}_b^1} \mu\left(\boldsymbol{x}^{(m,t)}\right) \\ - & \min_{\boldsymbol{x}^{(m,t)}: x_t^{(m,t)} \in \mathcal{X}_b^0} \mu\left(\boldsymbol{x}^{(m,t)}\right) \\ - & \lambda^a_{b+(t-1)K} \end{aligned} \tag{14}$$

where $\mathcal{X}_b^0$ and $\mathcal{X}_b^1$ are the subsets of $\mathcal{X}$ with the b-th bit set to 0 or 1 respectively. Nodes at stage t influence only the t-th layer's LLRs $\boldsymbol{\lambda}_t^e$.

Path Reduction: We jointly process all L input paths $\boldsymbol{x}^{(m,t,l)}$ of the previous stage t. Temporarily extend every path with every modulation symbol, then select the best new partial paths. For every $\boldsymbol{x}^{(m,t,l)}$, construct $m' = 1 \dots M$ candidates $\boldsymbol{x}^{(m',m,t,l)}$ with $x_{t-1}^{(m',m,t,l)} = \theta_{m'}$. For all $M \times M \times L$ candidates, compute a corresponding metric

$$\mu\left(\boldsymbol{x}^{(m',m,t,l)}\right) = \sum_{j=t-1}^{N_t} \mu_j\left(\boldsymbol{x}^{(m',m,t,l)}\right). \tag{15}$$

For every m', select the $l' = 1 \dots L$ best *incoming* paths

$$\boldsymbol{x}^{(m',t-1,l')} = \arg\overset{(l')}{\min_{m,l}} \mu\left(\boldsymbol{x}^{(m',m,t,l)}\right) \tag{16}$$

with the smallest metric (Eq. 15) from the $M \times L$ candidates leading into node m' at the next stage $t-1$. Symbol *A* in Fig. 1 designates an example for this selection in the PR$_1$ process ($t = 3$) at node $m' = 2$.

For every m, select the $l' = 1 \dots L$ best *outgoing* paths

$$\boldsymbol{x}^{(m,t,l')} = \arg\overset{(l')}{\min_{m',l}} \mu\left(\boldsymbol{x}^{(m',m,t,l)}\right) \tag{17}$$

from the $M \times L$ candidates leaving node m at the current stage t. Symbol *B* in Fig. 1 is a corresponding example in the PR$_2$ process ($t = 2$) at node $m = 2$.

If this is the last stage, deliver the paths to the LLR Computation. Otherwise, feed $\boldsymbol{x}^{(m,t,l')}$ to Path Completion and pass $\boldsymbol{x}^{(m',t-1,l')}$ to the next Path Reduction.

V. Algorithm Variants

The following suggested changes generate distinct variants of the proposed algorithm: i) Vary $L = 1, 2, 4$. ii) Replace the l^2-norm in Eq. 5 by the l_1-norm. iii) Introduce a compensation factor to the min-functions in Eq. 13 and Eq. 14, known as max-star. iv) Apply the QR decomposition to the *unsorted* complex-valued channel matrix. v) Leave out the channel matrix regularization. vi) Instead of the OHE in the Path Completion, use full expansion and sorting as in the Path Reduction.

VI. Performance Evaluation

A 40 MHz IEEE 802.11n-like scenario is considered assuming a MIMO system with four receive and transmit antennas, i.e., $N_t = 8$ real-valued transmit layers, and perfect knowledge of the spatially uncorrelated Rayleigh channel. We use a tail-biting convolutional code with polynomials $[133, 171]_8$ and puncturing, a random interleaver and a max-log BCJR decoder. The frame lengths are 864, 1728 and 2592 code bits for 4-/16-/64-QAM respectively. We support seven modulation and coding schemes (MCS) borrowed from the 11n standard: 4-QAM with r-1/2 and r-3/4, 16-QAM with r-1/2 and r-3/4, and 64-QAM with r-2/3, r-3/4 and r-5/6 (11n-MCS 25 to 31), where r denotes the code rate. The gray-mapped QAM schemes are decomposed into 2-/4-/8-ASK for our real-valued model. For every data point, we simulated at least 10^5 frames. Erroneous frames are automatically retransmitted as part of an automated repeat-request (ARQ) scheme, on average $1/(1-\text{FER})$ times for successful reception [3], where FER denotes the frame error rate. The *effective spectral efficiency* expressed as the on average correctly transmitted information bits per symbol is $\eta = rKN_t(1-\text{FER})$.

Fig. 3 visualizes the communications performance of the four most important algorithm variants ($L = 2, 4$, l_1/l^2-norm, rest as in Sec. IV) and of the PPTS [1] for one $I = 1$ and two $I = 2$ detector-decoder iterations. The top dashed curves mark the SNR points where the brute-force max-log detector achieved 10% FER. For each algorithm, it shows the envelope

$$\eta\left(\text{SNR}\right) = \max_{r,K} rKN_t\left(1 - \text{FER}\left(\text{SNR}, k, r\right)\right) \tag{18}$$

over the algorithm's seven per-MCS spectral efficiency curves. An MCS is deemed operational if FER $\leq 10\%$.

Observations: For lower SNRs, all five are very close. At high SNR, the $L = 2$ variants perform significantly worse than the PPTS, especially the $(l^2, L = 2)$ variant is very bad at the first iteration. The $L = 4$ variants outperform ($I = 1$) or closely match ($I = 2$) the PPTS algorithm. At the first iteration for $L = 4$, the l_1-norm is better than the l^2-norm, while this changes in the second iteration.

Also tested but not shown: i) The channel matrix regularization brings a significant advantage and should not be left out. ii) The channel matrix sorting provides an average gain of around $0.2\,\text{dB}$, thus could be left out. iii) The max-star variants, both for inner and outer reduction, have shown no significant advantage over the max-log variant, despite being computationally a lot more complex. iv) The $L = 1$ variants perform very bad. v) The full expansion is slightly better than the OHE based path completion, but a lot more complex, hence the OHE is preferable (at least for $M > 2$).

As apparent in Fig. 3, both the PPTS and our algorithm with $L = 4$ perform close to the max-log bound. The following complexity analysis indicates that the proposed algorithm is less complex and hence more hardware-friendly.

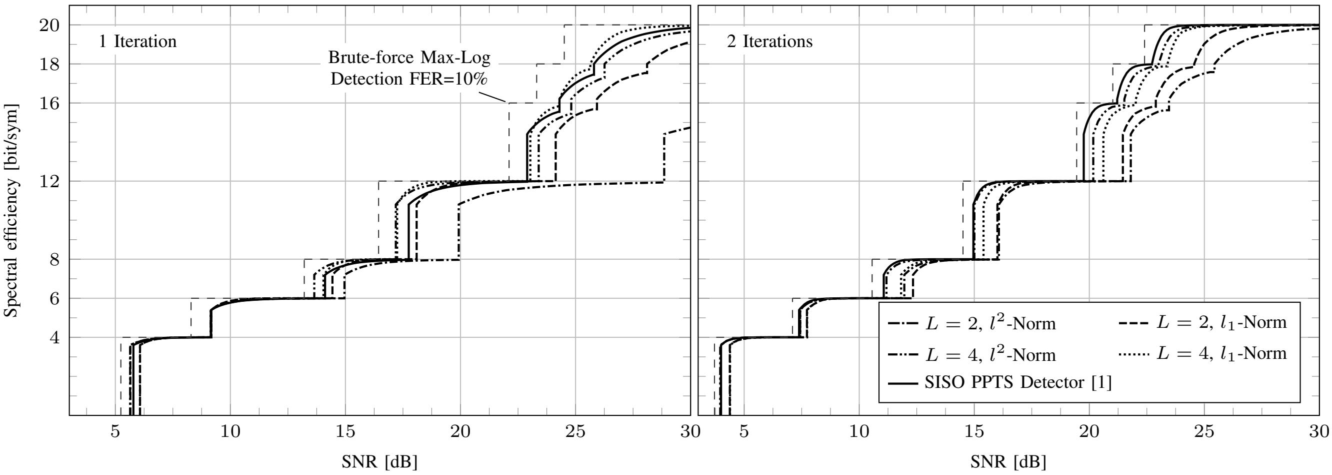

Fig. 3. Communications performance: effective spectral efficiency over SNR

TABLE I
METRIC EVALUATIONS FOR $N_t = 8, L' = 2$

	This Work	PPTS [1]	Ratio
$L = 2, M = 8$	1456	45120	3%
$L = 4, M = 8$	2896	45120	6%
$L = 2, M = 4$	536	2832	19%
$L = 4, M = 4$	1064	2832	38%
$L = 2, M = 2$	220	180	122%
$L = 4, M = 2$	436	180	242%

VII. Complexity Analysis

We use the number of metric $\mu(\boldsymbol{x})$ evaluations to approximately quantify the algorithms' computational complexities. Our algorithm requires

$$(N_t - 2)\,(N_t - 1)\,LM + 2M + (N_t - 2)\,M^2 L$$

evaluations, while the PPTS algorithm [1] with $L' = 2$ paths per Trellis node requires

$$L' M^4\,(N_t/2 - 2)\,[(N_t/2 - 1)\,/2 + 1] + M^2 + M^4$$

evaluations. For $N_t = 8$ and $L' = 2$, this reduces to $42LM + 2M + 6LM^2$ and $11M^4 + M^2$ respectively. Tbl. I lists the data for 4-/16-/64-QAM ($M = 2, 4, 8$) and $L = 2, 4$ with $L' = 2$ fixed. For higher modulation orders, the saving is significant. We only need a fraction compared to the PPTS algorithm. For $L = 4$, our algorithm is very close to [1] in terms of communications performance at 6% of the PPTS' complexity. The principal reasons for the complexity reduction are the ORVD and the OHE based path completion.

VIII. Comparison to the Reference Algorithm

This section describes differences to the PPTS detector [1] which served as basis for the proposed algorithm. We consider the PPTS with $L' = 2$ and our algorithm with $L = 4$ due to their comparable communications performance. A four-antenna 64-QAM system is used as example.

The PPTS uses complex-valued arithmetic and only the l^2-norm. For the example system, it requires a Trellis structure with four 64-node stages. Its size is the reason for the PPTS' very large hardware implementation despite supporting only 16-QAM. The best incoming/outgoing path selections identify the two smallest among 128 metric values. The channel matrix is not regularized, but we assume that the columns are sorted by their l^2-norm (though not mentioned explicitly in [1]). The path extension (corresponding to our path completion) uses full expansion, as in the PR process. Both reductions in the LLR computation use the max-star function.

We use real-valued arithmetic due to the ORVD and a Trellis with eight 8-node stages. Our OHE-based path completion is simpler than the PPTS' corresponding process. There is no compensation factor used in the LLR Computation's reductions. The l_1-norm might further simplify the hardware. Additionally, our algorithm features a reduced minimum-search complexity. Our path selections (Eq. 16 and 17) identify the four smallest out of only 32 values.

IX. Conclusions

We have introduced a new MIMO detection algorithm. For the considered 802.11n-like scenario, it operates close to the max-log optimal bound. The reduced complexity, amongst others in terms of a significantly decreased amount of metric evaluations, make it particularly hardware-friendly.

We are currently in the process of implementing the proposed algorithm as application-specific integrated circuit (ASIC) to further assess its implementation efficiency.

References

[1] Y. Sun and J. Cavallaro, "Trellis-search based soft-input soft-output MIMO detector: Algorithm and VLSI architecture," *Signal Processing, IEEE Transactions on*, vol. 60, no. 5, pp. 2617–2627, 2012.

[2] X. Chen, G. He, and J. Ma, "Vlsi implementation of a high-throughput iterative fixed-complexity sphere decoder," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 60, no. 5, pp. 272–276, May 2013.

[3] R. Comroe and J. Costello, D.J., "Arq schemes for data transmission in mobile radio systems," *Selected Areas in Communications, IEEE Journal on*, vol. 2, no. 4, pp. 472–481, July 1984.