LATTICE REDUCTION AIDED SELECTIVE SPANNING WITH FAST ENUMERATION FOR SOFT-OUTPUT MIMO DETECTION

Ubaid Ahmad, Min Li, Sofie Pollin, Claude Desset, Liesbet Van der Perre, Rudy Lauwereins

Interuniversity Micro-Electronics Center (IMEC) vzw
Kapeldreef-75, Leuven, B-3001, Belgium
Email: {ubaid, limin, pollins, desset, vdperre, lauwerei}@imec.be

ABSTRACT

Lattice Reduction (LR) is a promising technique to improve the performance of linear MIMO detectors. However, LR-aided linear hard output MIMO detection is still far from optimal. Practical systems use soft output information to exploit gains from forward-error-correcting codes to achieve near-optimal performance. In this paper, LR-aided Selective Spanning with Fast Enumeration (LR-SSFE) is proposed as a candidate list generation method for soft output MIMO detection. The proposed algorithm uses heuristics based on simple arithmetic operations, which results in a completely deterministic and regular data flow. Hence, LR-SSFE can be efficiently implemented on a parallel programmable architecture. LR-SSFE is compared to the Fixed Candidates Algorithm (FCA) in terms of performance and complexity, which is another LR-aided candidate list generation method. Under the same performance constraints LR-SSFE has a significantly lower complexity than FCA.

1. INTRODUCTION

The optimal solution to the Multiple Input Multiple Output (MIMO) detection problem is the Maximum a posteriori (MAP) detector, however its complexity increases exponentially with the number of antennas, and thus, suboptimal methods have to be used in practice. The challenge is to have MIMO detectors that can achieve performance comparable to the MAP detector while having a lower complexity. Linear MIMO detectors, such as Zero Forcing (ZF) or Minimum Mean Square Error (MMSE), are attractive choices for MIMO detection due to their low computational cost. However, they cannot efficiently remove the inter-stream interference and suffer from noise amplification. LR-aided ZF/MMSE have been proposed in [1] to improve the performance with sub-optimal linear detectors. Although LR-aided linear MIMO detection achieves the same diversity order as Maximum Likelihood (ML), there still exists a gap between LR-aided hardoutput MIMO detection and MAP, which can be reduced further with soft-output MIMO detection techniques. It has been shown in [2] and [3], that LR-aided soft output MIMO detectors can achieve near optimal performance. The method proposed in [4] uses the covariance matrix of the noise along with a nearest-neighbor search method to reduce the complexity of LR-aided soft output MIMO detection. However, the complexity of these algorithms is still very high for practical implementation. Three soft output LR-aided MIMO detection methods are proposed in [5]. The Fixed Candidates Algorithm (FCA) and Fixed Radius Algorithm (FRA), are based on the K-best and Sphere detection approaches, respectively. FCA can be efficiently implemented on a ASIC. However, its implementation on a parallel programmable architecture will require extensive data shuffling and memory rearrangement, which would result in low hardware resource utilization.

In order to achieve near-MAP performance with LR-aided soft output MIMO detection on a parallel programmable architecture, we propose LR-aided SSFE (Selective Spanning with Fast Enumeration). LR-SSFE takes advantage of the low complexity candidate list generation method SSFE [6], along with LR-aided linear detection to achieve near optimal performance. Heuristics are used to replace the spanning-sorting-deleting process in FCA [5]. Deterministic data flow and low cost arithmetic operations in LR-SSFE will result in an efficient implementation on a parallel programmable baseband architecture. Comparing with FCA, LR-SSFE achieves the same performance with significantly lower complexity.

The remainder of this paper is organized as follows: Section 2 describes the system model, LR-aided linear MIMO detection and LR-aided soft output MIMO detection. The LR-SSFE algorithm is proposed in Section 3, while Section 4 details the simulation results. Afterwards, conclusions are drawn in Section 5.

2. SYSTEM MODEL

Consider a spatially multiplexed MIMO system with M transmit and N receive antennas denoted as $M \times N$. The vector of received symbols $y \in \mathbb{C}^{N\times 1}$ is given as

$$y = \mathbf{H}s+n \qquad (1)$$

where $s \in \mathbb{C}^{M\times 1}$ denotes the vector of transmitted symbols taken independently from a Quadrature Amplitude Modulation (QAM) constellation with $\text{E}\left[ss^H\right] = 1_M$, and $n \in \mathbb{C}^{N\times 1}$ is the vector of independent complex Gaussian noise samples where $n_i \sim N(0,\sigma^2)$ for $1 \le i \le N$. $\mathbf{H} \in \mathbb{C}^{N\times M}$ denotes the MIMO channel matrix and is considered to be perfectly known at the receiver. The channel is considered to be i.i.d Rayleigh flat fading with unit variance.

2.1. Lattice Reduction-aided Linear MIMO Detection

LR-aided linear MIMO detection has been proposed in [1]. To perform LR-aided MIMO detection first a reduced lattice basis is obtained as $\widetilde{\mathbf{H}} = \mathbf{HT}$ where $\mathbf{T} \in \mathbb{C}^{M\times M}$ is a uni-modular matrix with $det(\mathbf{T}) = \pm 1$. In this paper, the CLLL algorithm [7] [8] is considered for LR. The system equation (1) can be rewritten as

$$y = \mathbf{HTT}^{-1}s{+}n \tag{2}$$

Then Moore-Penrose pseudo inverse of the transformed channel matrix, $\widetilde{\mathbf{H}}^\dagger$ is applied to obtain

$$\widetilde{\mathbf{H}}^\dagger y = \mathbf{T}^{-1}s + \widetilde{\mathbf{H}}^\dagger n = \mathbf{T}^{-1}\left[a\left(\bar{s} + \frac{1}{2}1_v\right)\right] + \widetilde{\mathbf{H}}^\dagger n \tag{3}$$

where $\bar{s}$ are the points of the scaled and shifted QAM constellation in the domain ($\bar{s}$), as shown in Fig.1 in case of QPSK. The scaling factor $a = [\sqrt{2}, \sqrt{4/10}]$ in case of QPSK and 16-QAM, respectively. The shift vector 1_v is an $M \times 1$ vector of $[1 + 1j]$. Now (3) can be rewritten as

$$\frac{1}{a}\widetilde{\mathbf{H}}^\dagger y - \frac{1}{2}\mathbf{T}^{-1}1_v = \mathbf{T}^{-1}\bar{s} + \frac{1}{a}\widetilde{\mathbf{H}}^\dagger n$$

$$\hat{z} = \mathbf{T}^{-1}\bar{s} + w \tag{4}$$

The hard estimate is obtained by rounding $z_r = \lceil \hat{z} \rfloor$ to the nearest complex integer and transforming back to the ($\bar{s} - domain$), $\bar{s}_{LRZF} = \mathbf{T}z_r$. The final hard detection step is shifting and scaling back to the original QAM constellation to obtain the LR-aided ZF hard estimate as

$$\hat{s}_{LRZF} = \left[a\left(\bar{s}_{LRZF} + \frac{1}{2}1_v\right)\right] \tag{5}$$

2.2. Lattice Reduction-aided Soft-Output MIMO Detection

The goal of soft output MIMO detection is to obtain reliability information about the detected symbols. Soft output MIMO detection usually consists of two parts: (a) A list generator that gives a list of candidate symbol vectors, denoted by $\mathbf{L} \subseteq \Omega^M$, where Ω^M is the set containing all the possibilities of $M \times 1$ vector symbol s; (b) A Log-likelihood-ratio (LLR) generator that approximates the *a posteriori probabilities*(APP), the approximation becomes near optimal when $\mathbf{L} = \Omega^M$. Generating the candidate list dominates the performance and complexity of soft output MIMO detection. A low complexity tree-searching method is proposed in [5] by performing QR-decomposition of $\mathbf{T}^{-1}$ as $\mathbf{T}^{-1} = \mathbf{Q}_T\mathbf{R}_T$, to obtain

$$\left\|\hat{z} - \mathbf{T}^{-1}\bar{s}\right\|^2 = \left\|\mathbf{Q}_T^H\hat{z} - \mathbf{R}_T\bar{s}\right\|^2 \tag{6}$$

from (4), where $\mathbf{Q}_T$ is a unitary matrix and $\mathbf{R}_T$ is an upper-triangle matrix. FCA [5] is based on K-best principle, choosing the K-best candidates at each layer. Once the K-best candidates are found the LR-ZF hard estimate $\hat{s}_{LRZF}$ is added to the candidate list. Although K-best involves modular and repetitive operations that can be parallelized in VLSI architectures it has various problems when implementing on a parallel programmable baseband architectures [6]:(1) extensive shuffling incurs significant cycle and energy overhead; (2) data-dependent memory-operations and computations will significantly degrade the hardware resource-utilization on a programmable architecture; (3) the complexity of the spanning-sorting-deleting process is still too high.

3. LATTICE REDUCTION AIDED SSFE

We propose LR-aided SSFE to overcome the aforementioned problems for implementation on a parallel programmable architecture. The idea is to first generate the LR-aided ZF hard estimate $\bar{s}_{LRZF}$ in the $\bar{s} - domain$ and then the candidate list is built around this estimate with efficient heuristics. Sorting-deleting process is eliminated in the LR-SSFE which results in a regular deterministic data flow. Simulation results show that these characteristics significantly reduce the complexity.

A spanning-tree can be constructed to generate a set of candidates minimizing the distance in (6). The level of the tree is $M + 1$; mark the root-level as $i = M + 1$ and the leaf-level as $i = 1$. Each node at level $i \in \{M + 1, .., 2\}$ is expanded to $\mathcal{M}_{\mathcal{QAM}}$ nodes at level $i + 1$, where $\mathcal{M}_{\mathcal{QAM}}$ is the constellation size. In this tree each node at level $i \in \{M, .., 2, 1\}$ is uniquely described by the partial vector symbols $\bar{\mathbf{s}}^i = [\bar{s}_i, \bar{s}_{i+1}, .., \bar{s}_M]$, the leaves at level $i = 1$ correspond to all possible vector-symbols Ω^M.

Annotate the root node with $T_{M+1} = 0$ and starting from Level $i = M$, the PED (Partial Euclidean Distance) of partial symbol vector $\bar{\mathbf{s}}^i = [\bar{s}_i, \bar{s}_{i+1}, .., \bar{s}_M]$ is $T_i(\bar{\mathbf{s}}^i) = T_{i+1}(\bar{\mathbf{s}}^{i+1}) + ||e_i(\bar{\mathbf{s}}^i)||^2$, where the PED-increment $||e_i(\bar{\mathbf{s}}^i)||^2$ is

$$||e_i(\bar{\mathbf{s}}^i)||^2 = ||Q_i\hat{z} - \sum_{j=i}^{M} R_{ij}\bar{s}_j||^2 \tag{7}$$

where Q_i is the i^{th} row of Q_T^H and R_{ij} are the i^{th} row and j^{th} column entries of $\mathbf{R}_T$. $||e_i(\bar{\mathbf{s}}^i)||^2$ is non-negative, so the PED increases monotonically from root to leaves. Hence, the formulation in (6) has now been transformed to a tree-search problem. The optimal solution is to find the leaf at level $i = 1$ with the minimal PED, $T_1(\bar{\mathbf{s}}^1)$.

3.1. LR aided SSFE in $\bar{s}-domain$

The SSFE [6], is uniquely characterized by a vector $\mathbf{m} = [m_1, \ldots, m_M]$. Starting from root level $i = M$, SSFE spans each node at level $i+1$ to m_i nodes at level i. The spanned nodes are never deleted. Hence, the total number of nodes at level i is $\prod_{k=i}^{M} m_k$. If the node at level $i = M+1$ has the associated partial symbol vector being $\bar{\mathbf{s}}^{i+1} = [\bar{s}_{i+1}, .., \bar{s}_M]$, the spanning is to select a set of $\bar{\mathbf{s}}^i = [\bar{s}_i, \bar{s}_{i+1}, .., \bar{s}_M]$ in a way that PED-increment $||e_i(\bar{\mathbf{s}}^i)||^2$ is minimized. Although, $||e_i(\bar{\mathbf{s}}^i)||^2$ can be minimized using FCA [5], which is essentially finding the K-Best closest points to the LR-aided hard estimate. However, this requires calculating the PEDs at each level and then sorting and selecting the K-best points. This is avoided in the proposed LR-SSFE by using fast enumeration in the $\bar{s}-domain$. Our approach is different from the SSFE proposed in [6] where the points are enumerated in the original QAM constellation (s), around the received symbols y. In LR-SSFE the points are enumerated in the $\bar{s}-domain$ instead, around the LR-aided hard estimate ($\bar{s}_{LRZF}$). Since, all the constellation points in the $\bar{s}-domain$ are integers (Fig.1, QPSK) this facilitates the use of simple arithmetic operations in enumeration.

Efficient heuristics, called FE (Fast Enumeration), can be derived to approximate the selection-sorting operations. To derive the FE, we first rewrite (7) as

$$\begin{aligned} ||e_i(\bar{\mathbf{s}}^i)||^2 &= ||Q_i\hat{z} - \sum_{j=i+1}^{M} R_{ij}\bar{s}_j - R_{ii}\bar{s}_i||^2 \\ &= ||Q_i z_r + Q_i e_z - \sum_{j=i+1}^{M} R_{ij}\bar{s}_j - R_{ii}\bar{s}_i||^2 \quad (8) \end{aligned}$$

where $e_z = \hat{z} - z_r$ is the quantization error from the $z-domain$ (infinite integer lattice domain). Clearly, the minimization of $||e_i(\bar{\mathbf{s}}^i)||^2$ is equivalent to the minimization of $||e_i(\bar{\mathbf{s}}^i)/R_{ii}||^2$. Hence, from (8) we derive

$$\begin{aligned} ||e_i(\bar{\mathbf{s}}^i)/R_{ii}||^2 &= ||\frac{Q_i z_r}{R_{ii}} + \frac{Q_i e_z - \sum_{j=i+1}^{M} R_{ij}\bar{s}_j}{R_{ii}} - \bar{s}_i||^2 \\ &= ||\bar{s}_{LRZF} + \underbrace{e_{\bar{s}} + \frac{Q_i e_z - \sum_{j=i+1}^{M} R_{ij}\bar{s}_j}{R_{ii}}}_{e_\alpha} - \bar{s}_i||^2 \\ &= ||\bar{s}_{LRZF} + e_\alpha - \bar{s}_i||^2 \quad (9) \end{aligned}$$

where $\bar{s}_{LRZF} = Q(\frac{Q_i z_r}{R_{ii}})$ is the LR aided ZF hard estimate obtained using the slicing operator Q, which is essentially rounding and applying boundary control and $e_{\bar{s}} = (\frac{Q_i z_r}{R_{ii}}) - \bar{s}_{LRZF}$ is the quantization error.

Specifically, minimizing (9) essentially selecting the closest complex integer constellation point (in the $\bar{s}-domain$) to $\bar{s}_{LRZF} + e_\alpha$. For LR-SSFE, the FE is to select a set of closest constellation points around $\bar{s}_{LRZF} + e_\alpha$. FCA [5] finds the closest constellation points to $\bar{s}_{LRZF} + e_\alpha$ calculating the Partial Euclidean Distances to all the constellation points excluding $\bar{s}_{LRZF}$ at each layer (line S8 Table II in [5]). This requires comparing the LR-aided hard estimate to all the constellation points for making the exclusion and calculating the PEDs. To avoid this we derive the Fast Enumeration based on (9).

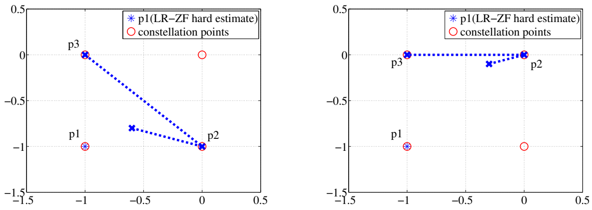
3.2. Fast Enumeration in $\bar{s}-domain$

In FE the first point is always set to $p_1 = \bar{s}_{LRZF}$ ($m_i = 1$) to guarantee the ML diversity. The closest constellation point to $\bar{s}_{LRZF} + e_\alpha$ is $p_2 = Q(\bar{s}_{LRZF} + e_\alpha)$. However, when $|\Re(e_\alpha)| < 0.5$ and $|\Im(e_\alpha)| < 0.5$, $p_2 = p_1$. To avoid this double inclusion of the same point we make use of simple operations, such as rounding and boolean logic 'OR'. When $|\Re(e_\alpha)| < 0.5$ and $|\Im(e_\alpha)| < 0.5$, finding the closest constellation points to $\bar{s}_{LRZF} + e_\alpha$ is approximately same as finding the closest points to $\bar{s}_{LRZF}$ i.e. p_1, otherwise it is finding closest constellation points to $\bar{s}_{LRZF} + e_\alpha$. For $m_i \geq 2$ more points can be efficiently enumerated based on the direction vector $d = \bar{s}_{LRZF} + e_\alpha - Q(\bar{s}_{LRZF} + e_\alpha)$. For $m_i, i \in [2, 3, 4]$ the points can be enumerated in the following way,

$$\begin{aligned} p_1 &= \bar{s}_{LRZF} \\ \zeta &= (\lceil|\Re(e_\alpha)|\rfloor > 0) \oplus (\lceil|\Im(e_\alpha)|\rfloor > 0) \\ \phi &= |\Re(d)| > |\Im(d)| \\ p_2 &= (!\zeta)(p_1 + (sgn(\Re(d))\phi + j(sgn(\Im(d))(!\phi))) \\ &\quad + (\zeta)(Q(\bar{s}_{LRZF} + e_\alpha)) \\ p_3 &= (!\zeta)(p_1) + (\zeta)(p_2) \\ &\quad + (sgn(\Re(d))(!\phi) + j(sgn(\Im(d))\phi)) \\ p_4 &= (!\zeta)(p_1) + (\zeta)(p_2) + (sgn(\Re(d)) + j(sgn(\Im(d))) \end{aligned} \quad (10)$$

where '$sgn()$' is the operator for extracting the sign of a number (positive/negative), '!' is the logic-not operator and '$\oplus$' is logical-OR. The technique applied here is to incrementally grow the set around p_1 when $\zeta = 0$, otherwise the set is built around the point $\bar{s}_{LRZF} + e_\alpha$ when $\zeta = 1$.

For example, if $\zeta = 0$ and $|\Re(d)| > |\Im(d)|$, the closest constellation p_2 to p_1 is on the horizontal-line where p_1 stays, and the distance between p_1 and p_2 is $(sgn(\Re(d))$. If $|\Re(d)| < |\Im(d)|$, p_2 is on the vertical-line where p_1 stays, and the distance is $j(sgn(\Im(d))$. In the other case, when $\zeta = 1$, points are enumerated around $\bar{s}_{LRZF} + e_\alpha$, the closest point p_2 to $\bar{s}_{LRZF} + e_\alpha$ is obtained as $Q(\bar{s}_{LRZF} + e_\alpha)$ and then further points are enumerated around p_2 using d as shown in (10). Similarly, p_3 and p_4 are enumerated with simple operations. Using this approach both explicit exclusion of the $\bar{s}_{LRZF}$ from the list and PED calculations are avoided. This makes data flow deterministic and data dependent memory re-arrangement is avoided. The FE (10) uses simple arithmetic operators like addition, subtraction and rounding. As the FE is carried out in the $\bar{s}-domain$, where all the points are integers, these operations can be implemented with low cost shift-add operations. Fig.1a and Fig.1b shows the 3 enumerated points in case of QPSK, when $\zeta = 0$ and $\zeta = 1$, respectively.



(a) (b)

Fig. 1. Fast Enumeration in $\bar{s} - domain$ for QPSK (a) $\zeta = 0$ (b) $\zeta = 1$

4. SIMULATION RESULTS

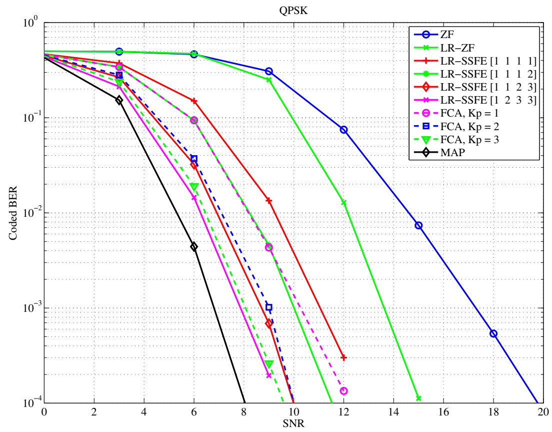
In this section, we provide performance comparison of the proposed soft output MIMO detector LR-SSFE to ZF, LR-aided ZF (LR-ZF), FCA [5] and MAP. CLLL [8] algorithm is used in all the cases for LR. A 4×4 MIMO system, with i.i.d Rayleigh Flat Fading channel is considered with complete channel state information at the receiver. The bit-flipping strategy proposed in [9] is applied to generate the LLR values from the candidate list. In all cases $1/2$-rate convolutional code of constraint length 7 with generator polynomials $[133, 171]$ is used.

4.1. BER Performance

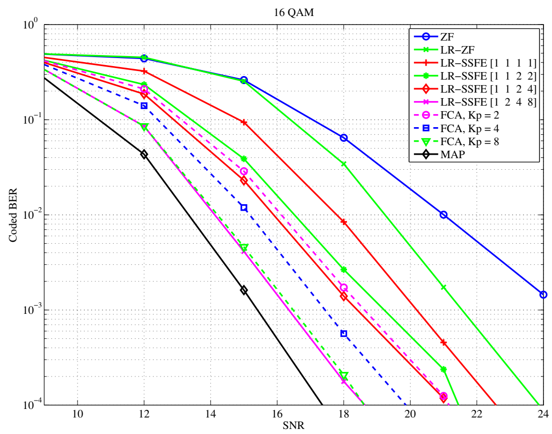
Fig.2 shows the coded BER results for QPSK and 16QAM. As expected, in all the cases LR-SSFE achieves the same diversity order as MAP. This proves the motivation for generating the candidate list using LR-aided ZF hard estimate. LR-SSFE with only one candidate $\mathbf{m} = [1111]$ provides a significant performance gain of 3dB compared to LR-ZF in case of QPSK, Fig.2a. While in case of 16QAM, Fig.2b, LR-SSFE with $\mathbf{m} = [1111]$ provides a gain of about 1dB compared to LR-ZF.

The number of candidates in FCA is $K_p + 1$ as the LR-ZF candidate is also added to the list of candidates [5]. In case of QPSK, LR-SSFE with $\mathbf{m} = [1112]$ has the same performance as FCA with $K_p = 1$, both having the same number of candidates. When $\mathbf{m} = [1233]$ LR-SSFE is only about 1dB away from MAP at a BER of 10^{-4}.

In case of 16QAM, Fig.2b, LR-SSFE with $\mathbf{m} = [1124]$ has a slightly better performance than FCA with $K_p = 2$. LR-SSFE with $\mathbf{m} = [1248]$ is only about 1dB away from MAP at a BER of 10^{-4}. This gap can be further reduced by using bigger values of m_i.



(a)



(b)

Fig. 2. BER Performance 4×4 MIMO system (a) QPSK (b) 16QAM

4.2. Complexity Comparison

To the best of our knowledge this is the first LR aided soft output MIMO detector specifically optimized for parallel programmable processors. Deterministic data flow and simplified arithmetic operations in LR-SSFE significantly reduce the implementation complexity. Table 1 shows the complexity comparison of LR-SSFE and FCA. The CMUL(Complex Multiplications) and CADD(Complex Additions) required in generating the candidate list for a 4×4 MIMO system with

16QAM are shown in Table 1, all the intermediate steps such as QR-decomposition and calculating $\widetilde{\mathbf{H}}^{\dagger}$ are also included. From Table 1 we can observe that LR-SSFE has a lower complexity compared to FCA [5].

LR-SSFE with $\mathbf{m} = [1124]$ has almost the same BER performance as FCA with $K_p = 2$ (Fig.2b), while the number of required operations are significantly reduced. Fig.2b shows that FCA [5] with $K_p = 4$ is only about 0.5dB better than SSFE with $\mathbf{m} = [1124]$ at 10^{-4}, while the complexity of FCA is more than twice of SSFE (Table 1). Although the size of the candidate list in case of LR-SSFE with $\mathbf{m} = [1248]$ is bigger than FCA $K_p = 8$, to achieve the same BER performance with in 1dB of MAP, the cost of generating the candidate list with LR-SSFE is still significantly lower than FCA (Table 1). This complexity gap between LR-SSFE and FCA will further increase when using bigger QAM constellations. In an implementation on a parallel programmable processor, LR-SSFE will have more efficient utilization of hardware resources than FCA, as there are is no sorting-deleting in LR-SSFE.

	ZF	LR-ZF	SSFE (m)				FCA (Kp)		
			1111	1122	1124	1248	2	4	8
CMUL	270	572	820	848	886	1330	1414	1998	3166
CADD	124	298	435	480	542	1210	810	1202	1986

Table 1. Complexity Comparison of LR-SSFE and FCA [5], 4×4 MIMO system 16QAM

5. CONCLUSION

In this work, we presented LR-SSFE as a low complexity soft output MIMO detector that can achieve close to MAP performance. LR-SSFE has a completely deterministic and regular data flow which will enable efficient implementation on a parallel programmable processor. Moreover, LR-SSFE can be configured to achieve different performance/complexity trade-offs, which is highly desirable for Software Defined Radio baseband processing. Simulation results show that under the same performance constraints LR-SSFE has a lower complexity than FCA.

6. REFERENCES

[1] D. Wubben, R. Bohnke, V. Kuhn, and K.-D. Kammeyer, "MMSE-based lattice-reduction for near-ML detection of MIMO systems," in *Smart Antennas, 2004. ITG Workshop on*, march 2004, pp. 106 – 113.

[2] P. Silvola, K. Hooli, and M. Juntti, "Suboptimal soft-output MAP detector with lattice reduction," *Signal Processing Letters, IEEE*, vol. 13, no. 6, pp. 321 – 324, june 2006.

[3] D.L. Milliner and J.R. Barry, "A lattice-reduction-aided soft detector for multiple-input multiple-output channels," in *Global Telecommunications Conference, 2006. GLOBECOM '06. IEEE*, dec 2006, pp. 1 –5.

[4] V. Ponnampalam, D. McNamara, A. Lillie, and M. Sandell, "On generating soft outputs for lattice-reduction-aided MIMO detection," in *Communications, 2007. ICC '07. IEEE International Conference on*, 24-28 2007, pp. 4144 –4149.

[5] Wei Zhang and Xiaoli Ma, "Low-complexity soft-output decoding with lattice-reduction-aided detectors," *Communications, IEEE Transactions on*, vol. 58, no. 9, pp. 2621 –2629, september 2010.

[6] Min Li, B. Bougard, E.E. Lopez, A. Bourdoux, D. Novo, L. Van Der Perre, and F. Catthoor, "Selective spanning with fast enumeration: A near maximum-likelihood MIMO detector designed for parallel programmable baseband architectures," in *Communications, 2008. ICC '08. IEEE International Conference on*, may 2008, pp. 737 –741.

[7] A. K. Lenstra, H. W. Lenstra, and J. L. Lovasz, "Factoring polynomials with rational coefficients," in *Math. Ann.*, 1982, vol. 261, pp. 515 –534.

[8] Ying Hung Gan, Cong Ling, and Wai Ho Mow, "Complex lattice reduction algorithm for low-complexity full-diversity MIMO detection," *Signal Processing, IEEE Transactions on*, vol. 57, no. 7, pp. 2701 –2710, july 2009.

[9] Min Li, R. Fasthuber, D. Novo, B. Bougard, L. Van Der Perre, and F. Catthoor, "Algorithm-architecture co-design of soft-output ML MIMO detector for parallel application specific instruction set processors," in *Design, Automation Test in Europe Conference Exhibition, 2009. DATE '09.*, april 2009, pp. 1608 –1613.