



**QUEEN'S
UNIVERSITY
BELFAST**

Bounded Selective Spanning With Extended Fast Enumeration for MIMO-OFDM Systems Detection

Wu, Y., & McAllister, J. (2017). Bounded Selective Spanning With Extended Fast Enumeration for MIMO-OFDM Systems Detection. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 64(9), 2556-2568. DOI: 10.1109/TCSI.2017.2706722

Published in:

IEEE Transactions on Circuits and Systems I: Regular Papers

Document Version:

Peer reviewed version

Queen's University Belfast - Research Portal:

[Link to publication record in Queen's University Belfast Research Portal](#)

Publisher rights

© 2017 IEEE. This work is made available online in accordance with the publisher's policies. Please refer to any applicable terms of use of the publisher.

General rights

Copyright for the publications made accessible via the Queen's University Belfast Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

The Research Portal is Queen's institutional repository that provides access to Queen's research output. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact openaccess@qub.ac.uk.

Bounded Selective Spanning with Extended Fast Enumeration for MIMO-OFDM Systems Detection

Yun Wu, *Member, IEEE*, and John McAllister, *Senior Member, IEEE*,

Abstract—Sphere decoders allow receivers in Multiple-Input Multiple-Output (MIMO) communications systems to detect QAM symbols with quasi-optimal accuracy and low complexity compared to the ideal Maximum Likelihood (ML) detector. However, their high complexity relative to simple linear detectors means that the latter are still usually adopted, despite their lower detection performance. Configurable sphere decoders such as Selective Spanning Fast Enumeration (SSFE) allow complexity to be reduced at the cost of lower performance and are hence ideal for transceivers for Internet-of-Things (IoT) equipment, where scale, operating context and resource and energy budgets vary dramatically. However, SSFE still suffers performance limitations due to the internal heuristics employed for symbol selection and enumeration and real-time, software-defined realisations for even moderately demanding MIMO standards, such as 802.11n, have not been recorded. This paper presents a new variant of SSFE which, by employing novel fast symbol enumeration and modulation dictionary spanning heuristics increases performance and computational efficiency to the point where very substantial reductions in resource can be achieved without impacting detection accuracy relative to SSFE. This is demonstrated via a series of FPGA-based detectors 2×2 and 4×4 , 16-QAM 802.11n MIMO.

Index Terms—Field Programmable Gate Array (FPGA), Processor, Real-Time, Multi-Input Multi-Output (MIMO), Orthogonal Frequency-Division Multiplexing (OFDM), Sphere Decoder, 802.11n

I. INTRODUCTION

MULTIPLE-INPUT, Multiple-Output (MIMO) communications systems [1] exploit spatial diversity to provide wireless communications channels of unprecedented capacity and throughput. These capabilities have seen MIMO adopted to increasing degrees in standards relevant to the IoT, such as 802.11n [2], LTE and LTE-Advanced [3], 5G [4] and the emergence of MIMO technologies of very large-scale for next generation cellular communications [5].

A MIMO system employs multiple antennas at both transmitter and receiver terminals to enhance both signal quality and data rate by exploiting spatial diversity [6]. Fig. 1 shows a generic model of MIMO system with N_t transmitting antennas and N_r receiving antennas.

In the context of IoT, equipment can vary dramatically in its scale (i.e. number of antennas), modulation density, operating environment and channel quality, energy budget and data rates [7]. This means that embedded MIMO transceiver algorithms and architectures must be highly adaptable for use in different contexts. One particularly problematic area is symbol detection: the estimation of transmitted symbols

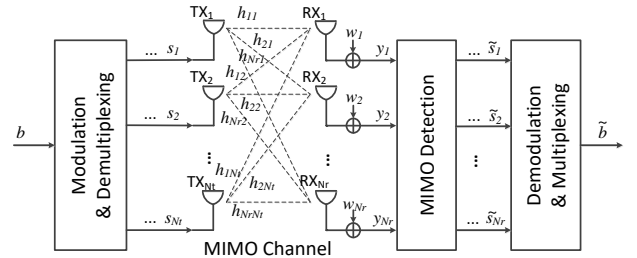


Fig. 1: MIMO System Model

from those received. Simple equalizers are low-cost but offer low performance detection, but more capable approaches, such as sphere decoders can be highly computationally complex. In order to bridge this gap between detection performance and cost, a series of 'adaptive' detectors have emerged which permit trade-off of detection performance with cost [8], [9], a critical capability for IoT equipment.

Selective Spanning with Fast Enumeration (SSFE) [10] is particularly promising approach. It enables very low cost and potentially good performance by design-time tuning of the computational complexity and detection performance [10], [8]. However, despite these advantages, SSFE currently suffers from restrictions. Specifically,

- It is only capable of enumerating 8 candidate symbols, irrespective of the cardinality of the modulation dictionary. This limits detection accuracy for even moderately-sized modulation schemes.
- It enumerates redundant or known-impossible symbols in certain scenarios, again potentially restricting detection accuracy.

This paper resolves these issues. It presents a novel Boundary Selective Scanning with Extended Fast Enumeration (BSS-EFE) algorithm¹ with the following characteristics:

- A novel Boundary SS (BSS) heuristic which avoids enumerating known-impossible symbols, improving BER performance by an average of 1 dB (25.9%) SNR as compared to SSFE.
- A novel Extended FE (EFE) heuristic, which can span all members of a QAM constellation. By combining with BSS, this enhances performance by a further 1 dB SNR whilst reducing computational complexity by 16.3% as compared to SSFE.
- Novel FPGA-based BSS-EFE detectors are presented which enable real-time detection for 2×2 and 4×4 16-

Authors undertook this work at the Institute of Electronics, Communications and Information Technology (ECIT), Queen's University Belfast, UK e-mail: {yun.wu, jp.mcallister}@qub.ac.uk

¹A preliminary version of this work was presented in [11]

QAM 802.11n with performance and/or cost substantially in advance of SSFE.

The rest of this paper is organized as follows. Sections III and IV describes SSFE and quantify its effectiveness, before Sections V and VI introduce BSS-EFE. The performance and complexity of BSS-EFE are described in Section VII, before Section VIII realises and analyses BSS-EFE.

II. BACKGROUND

A. MIMO Detection and Sphere Decoding

In a typical $N_t \times N_r$ MIMO system, such as that in Fig. 1 a bit stream \mathbf{b} is modulated and multiplexed onto N_t transmitters to form a transmitted symbol vector $\mathbf{s} \in \mathbb{C}^{N_t \times 1}$. At the N_r -element receiver, the received symbol vector $\mathbf{y} \in \mathbb{C}^{N_r \times 1}$, is retrieved and an estimate $\tilde{\mathbf{s}}$ of \mathbf{s} made. The relationship between $\tilde{\mathbf{s}}$ and \mathbf{y} is assumed to be of the form (1)

$$\mathbf{y} = \sqrt{\frac{\rho}{N_t}} \cdot H \cdot \mathbf{s} + \mathbf{w} \quad (1)$$

where $\mathbf{w} \in \mathbb{C}^{N_r \times 1}$ is the vector of mutually independent and identically distributed complex Additive White Gaussian Noise (AWGN) elements with power σ_w^2 . ρ is the Signal-to-Noise Ratio (SNR) denoted by $\frac{\sigma_s^2}{\sigma_w^2}$, where σ_s^2 the signal power of \mathbf{s} . The Rayleigh-distributed fading multipath channel between transmitter and receiver is represented by $H \in \mathbb{C}^{N_r \times N_t}$ where $h_{i,j}$, represents the unit-power fading path between the i^{th} receive and j^{th} transmit antennas.

MIMO detectors seek a solution that minimizes the error between the estimated and actual transmitted signals, expressed as [12]

$$\min_{\mathbf{x}_i} \|\mathbf{y} - H \cdot \mathbf{x}_i\|^2, \quad \mathbf{x}_i \in D_{N_t}^{M_c}, \quad i \in [1, M_c^{N_t}] \quad (2)$$

where M_c denotes the scale of modulation type, $D_{N_t}^{M_c} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{M_c^{N_t}}\}$ denotes the set of all $M_c^{N_t}$ possible transmit symbol vectors over N_t transmit antennas and $\mathbf{x}_i = \{\tilde{s}_1, \tilde{s}_2, \dots, \tilde{s}_{N_t}\}$ is one of the estimated candidate vectors from $D_{N_t}^{M_c}$.

The optimal ML detector attempts to solve the problem denoted in (2) by exhaustively searching all possible symbol vectors; the result $\tilde{\mathbf{s}}_{ML}$ is that with the smallest Euclidean Distance of all candidate symbol vectors in $D_{N_t}^{M_c}$. This establishes the upper bound on error rate for uncoded MIMO detectors [13]. However, the solution space $D_{N_t}^{M_c}$, is of size $M_c^{N_t}$ leading to exponentially increasing ML detection complexity $O(M_c^{N_t})$. Modern wireless standards continuously increase both the number of antennas N_t and the density of the modulation scheme M_c [3] and the associated exponential growth in ML detection complexity is rapidly rendering it computationally infeasible. For real-time implementation, sub-optimal detectors are employed instead of ML.

Sphere decoders [14] support quasi-optimal detection performance whilst offering reduced computational complexity relative to ML [15]. In general, sphere decoding algorithms

traverse a partial hyperspherical-space around the received symbol $D_{N_t}^C$ [14] with detection objective given in (2)²:

$$\min_{\mathbf{x}} \|\mathbf{y} - H \cdot \mathbf{x}\|^2 \quad (3)$$

via QR decomposition of H this can be reformulated as [7]:

$$\min_{\mathbf{x}} \|R \cdot (\mathbf{y}_{ZF} - \mathbf{x})\|^2 \quad (4)$$

where \mathbf{y}_{ZF} is a received symbol which has undergone ZF equalization.

Since R is upper triangular, the final row of the matrix product in (4) has only a single non-zero entry, which can be considered interference-free [16]. Therefore, the euclidean distance can be calculated recursively through R , row-by-row, in reverse order. At each row are calculated the Partial Euclidean Distance (PED) and the Accumulated PED (APED), given by:

$$PED_l = \sum_{j=l}^{N_t} r_{l,j}^2 \|y_{e_j} - x_j\|^2, \quad l \in [1, N_t] \quad (5)$$

and

$$APED_l = \sum_{j=l}^{N_t} PED_j \quad (6)$$

where y_{ZF_j} and x_j are the j^{th} element of \mathbf{y}_{ZF} and \mathbf{x} and l is the index of current layer.

Sphere decoding enables a trade-off of complexity and detection performance by limiting $APED_l$ in (6) within a threshold d [15]. Sphere decoding can be expressed as a tree search structure as shown in Fig. 2 with d a specific radius from the root limiting the search scope. By retaining only the leaf nodes within d , (4) becomes a tree search problem [17]. Among the retained leaves, that with the lowest $APED$ is the final solution. By carefully selecting d and the search strategy, sphere decoder can outperform linear detection with considerably lower complexity than ML detection [17].

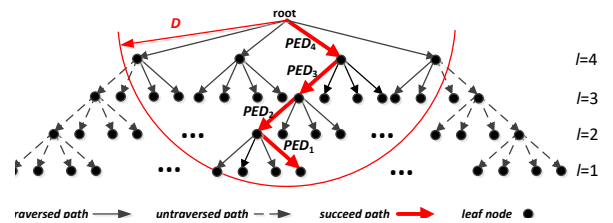


Fig. 2: The Generic Tree-Search Structure of Sphere Decoder

B. Sphere Decoder Enabled Real-Time Detection

The sphere decoder tree-search problem is well known and strategies such as Depth First Search (DFS), Breadth First Search (BFS) and Best Metric First Search (BMFS) have been devised to adapt computational efficiency, determinism, parallelism and performance to support realisations with varying

²Note that all of data values from this point forward have been sorted according to a policy specific to the class of sphere decoder used

performance and cost [18]. From the perspective of real-time implementation, a number of such approaches are notable, in particular two BFS approaches: the Fixed Complexity Sphere Decoder (FSD) [19] and Selective Spanning Fast Enumeration (SSFE) [20]. The former employs a hybrid brute-force/low-complexity approach which enables quasi-ML detection using deterministic dataflow through a fixed tree structure, whilst avoiding computationally demanding operations such as the sort operations prolific in other approaches [9]. These characteristics are important from the perspective of real-time implementation as they enable high performance custom accelerators to be realised, as demonstrated in a number of works [9]. SSFE has similar capabilities but has the added benefit of being adaptable to trade-off algorithm performance and complexity whilst maintaining the same features which lend themselves so well to efficient realisation.

III. SELECTIVE SPANNING WITH FAST ENUMERATION - SSFE

A. The SSFE Algorithm

SSFE is a BFS sphere decoding method which allows the designer of a system, at design time, to determine the number of symbols enumerated at each layer of a tree, so that they can control and trade-off detection performance and implementation resource and energy cost. Each configuration is defined by a vector $\mathbf{v} \in (\mathbb{Z}^+)^m$, with each $v_i \in \mathbf{v}$ defining the number of symbols enumerated at layer i of the tree. Fig. 3 demonstrates the influence of \mathbf{v} on the tree structure for two example values.

SSFE is generally composed of three phases: *preprocessing*, *symbol enumeration* and *result selection*. Preprocessing performs two functions: it equalizes the received signal vector \mathbf{y} to produce \mathbf{y}_{ZF} and orders the channel matrix H such that the i^{th} detected layer is determined by :

$$k_i = \arg \max_{l=[1, N_t]} \mathbf{norm}_l. \quad (7)$$

where \mathbf{norm}_l is the norm of l^{th} column of H .

This metric always orders the decoded signals from highest to lowest power. During symbol enumeration, SSFE performs APED calculations given by (5); $\hat{\mathbf{y}}_n$ at each stage is given by

$$\hat{\mathbf{y}}_n = \mathbf{y}_n - \sum_{l=n+1}^{N_t} \frac{r_{n,l}}{r_{l,l}} (\hat{\mathbf{s}}_l - \hat{\mathbf{y}}_l), \quad (8)$$

SSFE employs a 'fast' enumeration heuristic spanning a partial set of the QAM constellation. The heuristic enumerates candidate symbols based on the sliced value of the ZF equalized symbol $\hat{\mathbf{s}}_n$. A sequence of candidate symbols $S = \{\hat{\mathbf{s}}_i\}_{i=0}^7$ are enumerated described in (9), where $d = \hat{\mathbf{y}}_n - \mathcal{Q}(\hat{\mathbf{y}}_n)$, $\phi = \Re(d) > \Im(d)$, $R_{sgn} = \text{sgn}(\Re(d))$ and $I_{sgn} = \text{sgn}(\Im(d))$

$$\left\{ \begin{array}{l} \hat{\mathbf{s}}_1 = \hat{\mathbf{s}}_0 + 2 \cdot (R_{sgn} \cdot \phi + j \cdot I_{sgn} \cdot (!\phi)), \\ \hat{\mathbf{s}}_2 = \hat{\mathbf{s}}_0 + 2 \cdot (R_{sgn} \cdot (!\phi) + j \cdot I_{sgn} \cdot \phi), \\ \hat{\mathbf{s}}_3 = \hat{\mathbf{s}}_0 + 2 \cdot (R_{sgn} + j \cdot I_{sgn}), \\ \hat{\mathbf{s}}_4 = \hat{\mathbf{s}}_0 - 2 \cdot j \cdot I_{sgn}, \\ \hat{\mathbf{s}}_5 = \hat{\mathbf{s}}_3 - 4 \cdot j \cdot I_{sgn}, \\ \hat{\mathbf{s}}_6 = \hat{\mathbf{s}}_0 - 2 \cdot R_{sgn}, \\ \hat{\mathbf{s}}_7 = \hat{\mathbf{s}}_3 - 4 \cdot R_{sgn}, \end{array} \right. \quad (9)$$

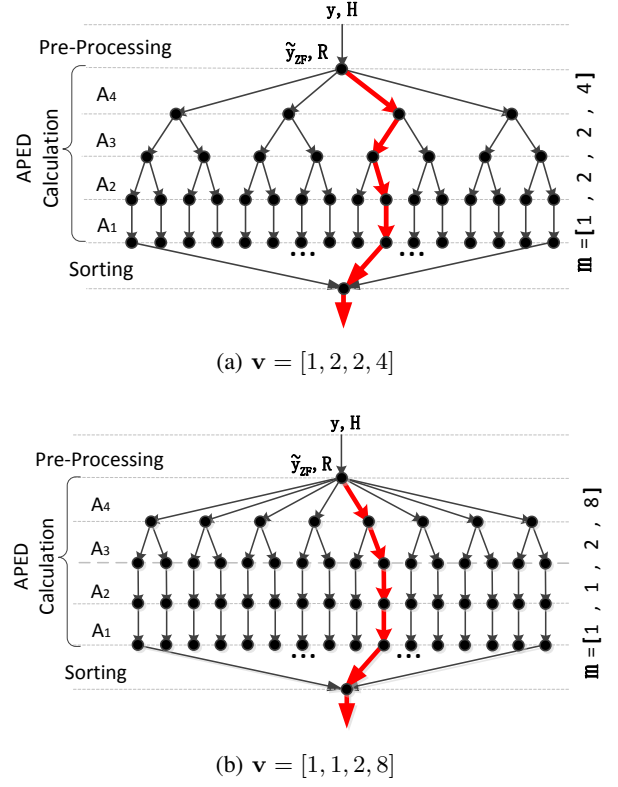


Fig. 3: SSFE Configuration Examples

The final detected symbol vector $\hat{\mathbf{s}}$ is then selected from the paths formed by the combinations of enumerated symbols according to

$$\hat{\mathbf{s}} = \arg \min_{\hat{\mathbf{s}}_j} (APED_j), \quad j \in [1, \prod_{l=1}^{N_t} v_l] \quad (10)$$

B. SSFE: Analysis

SSFE is very effective when a designer is concerned not only with detection performance, but also by the performance and cost of realising the detector. The number of candidates enumerated can be controlled to trade detection performance, which increases as the number of candidates increases, with the resource, performance and energy cost of enumerating those candidates. SSFE places control of this balance in the hands of the designer, who defines the configuration vector \mathbf{v} . However, there are some limitations associated with SSFE:

- Selective spanning: Regardless of the size of the modulation dictionary, a maximum of eight candidate symbols may be enumerated; in the case of even 16-QAM, this represents only 50% of the candidate dictionary and potentially restricts performance.
- Fast enumeration: Fig. 4 illustrates the behaviour of the FE process in the case of an arbitrary value of $\mathbf{y}_{ZF} = (3.5, 3.5)$ assuming 16-QAM. As this shows, $\hat{\mathbf{s}}_0$ takes a value at the edge of the valid constellation; when the enumeration process is followed, six of the eight symbols enumerated are outside the valid constellation

and therefore impossible. The computations expended considering these symbols is therefore wasted.

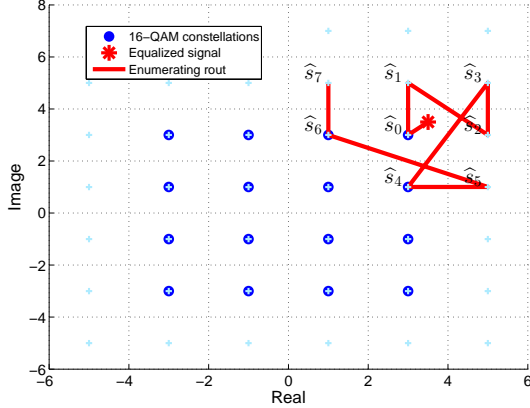


Fig. 4: Example SSFE Enumeration Procedure

Hence, SSFE offers many advantages in the context of some restrictions. By enabling design-time trade-off of performance and cost, whilst avoiding sort operations at each layer, as experienced in K -best detectors [9], SSFE places powerful control of performance and cost in the hands of the designer. However, its scope and the efficiency of its selective spanning and fast enumeration strategies potentially restrict performance and increase cost. The aims of this paper are two-fold:

- **Boundary Selective Spanning:** This work aims to eliminate the inefficiency in the SSFE selective spanning processing to ensure that only valid QAM constellation points are enumerated.
- **Extended Fast Enumeration:** This work aims to extend the customisability of SSFE by allowing enumeration of customisable numbers of constellation points up to the limit imposed by the modulation dictionary.

We will resolve these issues in the following contexts:

- **'Hard' detection:** Standalone estimation of the transmitted symbol in the absence of side-channel information from channel decoding is assumed. The alternative 'soft' approaches are frequently based on hard detection algorithms and hence this work may be further developed to realise a soft detection strategy.
- **MIMO scale:** IoT devices and networks promise MIMO topologies of widely differing scales, from a minimum of two-antenna units to much larger-scale topologies for top-end LTE and LTE-A communications. This paper focuses on 2×2 and 4×4 MIMO employing 16-QAM modulation as standard in, for example, 802.11n Wi-Fi.

IV. SSFE ENUMERATION EFFICIENCY

The SSFE symbol enumeration heuristic is independent of modulation scheme and has two potential limitations: it can enumerate only eight candidate symbols and in its current form may enumerate symbols outside of the valid modulation dictionary Ω .

The former limitation is clear from examination of the enumeration heuristic, presented in Section III and the latter is illustrated in Fig. 4. In that case, the enumeration route traverses a sequence where $\hat{s}_1 - \hat{s}_3$, \hat{s}_5 and \hat{s}_7 all lie outside of the valid constellation. Accordingly, enumerating these symbols represents redundant computation, since it is known that they are infeasible, yet are not treated as such.

To quantify the efficiency of the SSFE enumeration heuristic it is desired to characterise for each received symbol the set of enumerated candidates which are invalid, i.e. $|J|$ where $J = \{j : j \in S, J \notin \Omega\}$. To gauge this efficiency, equiprobable transmitted symbols in Ω are assumed for each antenna in the long term. The specific symbols enumerated and hence J , depending on the location of the ZF equalized symbol \hat{y}_n , as described in (9) relative to the valid constellation. By considering the possible values \hat{y}_n may take and their orientation relative to the feasible constellation points, broad classes of redundant enumeration scenarios emerge.

Consider the value of \hat{y}_n relative to lower and upper bound thresholds, $\tau_l = \sqrt{M_c} - 2$ and $\tau_u = \sqrt{M_c} - 1$ where $M_c = |\Omega|$ is the cardinality of the modulation dictionary. Based on the value of \hat{y}_n with respect to these bounds, four classes of enumeration route can be identified, each of which corresponds to a different number of invalid symbols $i \in \{2, 3, 4, 5\}$ being enumerated. These classes are defined as $S_2 - S_5$ below.

$$\begin{aligned}
 S_2 &= \left\{ \hat{y}_n : \begin{aligned} &(\tau_2 > |\Re(\hat{y}_n)| \geq \tau_1 \wedge |\Im(\hat{y}_n)| < \tau_1) \vee \\ &(\tau_2 > |\Im(\hat{y}_n)| \geq \tau_1 \wedge |\Re(\hat{y}_n)| < \tau_1) \end{aligned} \right\} \\
 S_3 &= \left\{ \hat{y}_n : \begin{aligned} &(|\Re(\hat{y}_n)| \geq \tau_2 \wedge |\Im(\hat{y}_n)| < \tau_1) \vee \\ &(|\Im(\hat{y}_n)| \geq \tau_2 \wedge |\Re(\hat{y}_n)| < \tau_1) \end{aligned} \right\} \\
 S_4 &= \left\{ \hat{y}_n : \begin{aligned} &(\tau_2 > |\Re(\hat{y}_n)| \geq \tau_1 \wedge |\Im(\hat{y}_n)| > \tau_1) \vee \\ &(\tau_2 > |\Im(\hat{y}_n)| \geq \tau_1 \wedge |\Re(\hat{y}_n)| > \tau_1) \end{aligned} \right\} \\
 S_5 &= \left\{ \hat{y}_n : (|\Re(\hat{y}_n)| \geq \tau_2 \wedge |\Im(\hat{y}_n)| \geq \tau_2) \right\}
 \end{aligned}$$

Fig. 5 illustrates each of these regions for a 16-QAM constellation. Note that $\bigcap_{i=2}^5 S_i = \emptyset$. Adopting $P(i)$ to denote the probability that \hat{y}_n falls into S_i , then the probability that a given entry of a sequence of k enumerated symbols is outside of the the valid constellation set Ω , $P(\hat{s}_n \notin \Omega)$ is given by

$$P(\hat{s}_n \notin \Omega) = \sum_{i=2}^5 P(i) \frac{i}{k} \quad (11)$$

The variation in relative frequency of occurrence of each of $S_2 - S_5$ with SNR is illustrated in Fig. 6 for $k = 8$, 16-QAM modulation and $i \in \{2, 3, 4, 5\}$. These measures are derived by Monte-Carlo simulation for 1×10^5 symbols. The values of $\gamma_i = P(i) \frac{i}{k}$ and the overall proportion of symbols enumerated which are invalid are all quoted. As this shows, both γ_i and $P(\hat{s}_n \notin \Omega)$ gradually stabilize as SNR increases to a proportion of approximately 19% of enumerated symbols which are invalid.

Repeating this procedure for modulation schemes varying between 4-QAM to 64-QAM with $k = 4$ and $k = 8$ produces the enumeration efficiency measurements given in Fig. 7. As this shows, as the order of the modulation scheme increases, the proportion of redundant enumerations decreases, but in

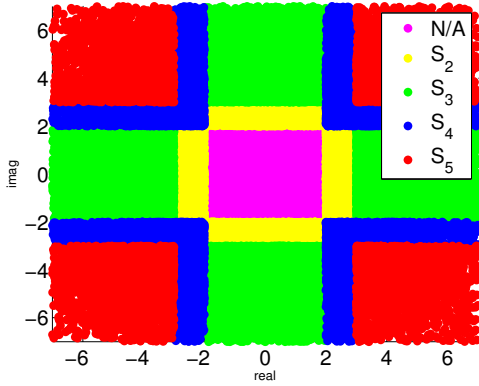


Fig. 5: 16-QAM Invalid Enumeration Regions

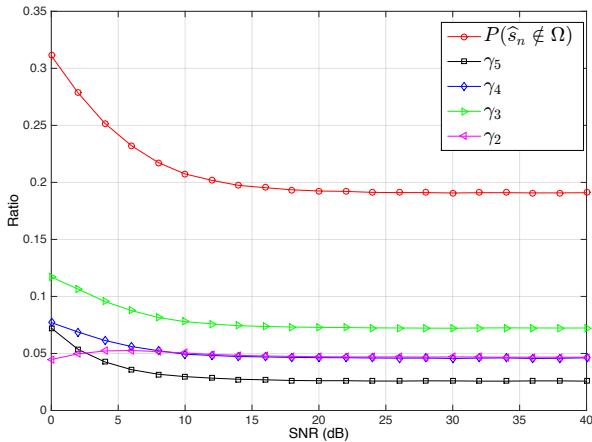


Fig. 6: Enumeration Redundancy variation with SNR (16-QAM, $k = 8$)

all cases at least 10% of enumerated symbols are invalid over all tested modulation schemes. Hence, even if highly efficient architectures are produced for SSFE, over 10% of its operations are redundant at each antenna when 4 or 8 symbols are enumerated.

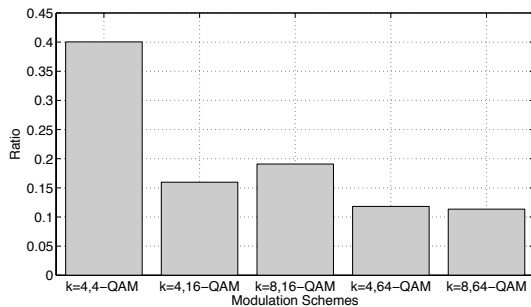


Fig. 7: Enumeration Redundancy for Different Schemes

Motivated by these observations, a bounded spanning method is described to solve the redundant enumeration issue in Section V, before a novel strategy is proposed in Section VI

to resolve the constraint on the size of the set of enumerated symbols.

V. BOUNDED SELECTIVE SPANNING FOR QAM

In Fig. 4, the enumerated symbols $\{\hat{s}_1, \hat{s}_2, \hat{s}_3, \hat{s}_5, \hat{s}_7\}$ are known not to be possible, but are still enumerated, leading to at least 10% redundancy in the computational effort expended enumerating symbols. This section describes a Bounded Selective Spanning (BSS) process which avoids enumerating symbols known to be impossible, instead enumerating valid alternatives, with the ultimate goal of increasing the computational efficiency of the detection process and hence potentially either increase detection performance for the same computational effort, or reduce implementation cost for the same detection performance requirement.

The challenge in defining the BSS scheme is to identify a heuristic which maps each invalid enumerated point to a valid alternative to increase detection performance (i.e. minimise BER), whilst minimising computational complexity. For instance, one potential solution could be to record the sequence of enumerated symbols, identifying the invalid symbols as they are enumerated, and map each of these to the closest (by Euclidean distance measurement) valid symbol which is not enumerated. However, to do so would require the calculation of a series of highly complex Euclidean distance measurements, including square root operations, and comparison operators. It would be much more desirable to exploit a scheme which applies simple, primitive arithmetic or logical operations to map these invalid symbols to valid alternatives as they are enumerated.

As an illustration, consider Fig. 8 which shows (shaded) the valid set of symbols in the constellation set Ω for 16-QAM, with those beyond the boundary known to be outside of Ω and therefore invalid.

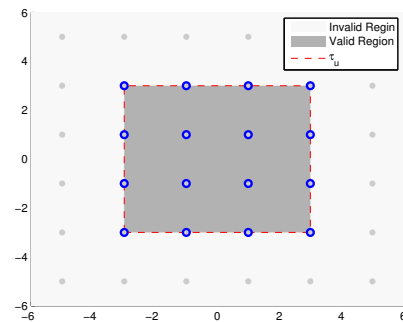


Fig. 8: The Bound of 16-QAM

For a given QAM constellation size, the set of valid points Ω are defined as $\{ \{|\Re(\hat{y}_n)| \leq \tau_u\} \vee \{|\Im(\hat{y}_n)| \leq \tau_u\} \}$. Hence τ_u demarcates a 'container' such that when the enumeration path for a given \hat{y}_n crosses the threshold, the corresponding enumerated symbol $\hat{s}_i \notin \Omega$ and is therefore invalid. Specifically, \hat{s}_i is considered invalid when $\hat{s}_i \in \{ \{|\Re(\hat{y}_n)| > \tau_u\} \vee \{|\Im(\hat{y}_n)| > \tau_u\} \}$, where $\tau_u = \sqrt{M_c} - 1$ as in Section IV.

Each symbol, however, is composed of a real part and an imaginary part, either of which may be valid or invalid. We propose a BSS scheme that for each enumerated symbol \hat{s}_i an alternative symbol $\tilde{s}_i \in \Omega$ is enumerated. \tilde{s}_i are translated from \hat{s}_i by considering the real and imaginary parts of \hat{s}_i to alternatives independently in the case where they exceed τ_u in Fig. 8, otherwise they remain unchanged. This is achieved by adding to \hat{s}_i an offset δ_i such that

$$\tilde{s}_i = \hat{s}_i + \delta_i \quad \text{for } i = 0, \dots, 7 \quad (12)$$

where

$$\delta_i \begin{cases} = 0 & \text{when } \hat{s}_i \in \Omega \\ \neq 0 & \text{otherwise} \end{cases} \quad (13)$$

That is, when $\hat{s}_i \in \Omega$, then that candidate symbol is retained, and when $\hat{s}_i \notin \Omega$ an alternative candidate is identified. The real and imaginary components of the offset are calculated to provide a horizontal translation (real component) or vertical translation (imaginary component) when required. To derive the directions of these translations, we note that the real (imaginary) components' translations must have positive polarity when $\Re\{\hat{s}_0\} > 0$ ($\Im\{\hat{s}_0\} > 0$) and negative when $\Re\{\hat{s}_0\} < 0$ ($\Im\{\hat{s}_0\} < 0$). Hence the direction of translation can be determined based only on the polarity of \hat{s}_0 . To determine the size of the translation, it is noted that, since the SSFE enumeration heuristic successively enumerates the symbols closest to the sliced value of the equalized signal, the points enumerated will form a cluster around the first point enumerated, with the number of neighbouring points enumerated in each direction around \hat{s}_0 at maximum given by $\lfloor \sqrt{v_i} \rfloor - 1$. Finally, the offset takes the value zero when the magnitude of the respective component exceeds τ_u , otherwise a non-zero translation is applied. Accordingly, the magnitude of the translation required can be encapsulated in a parameter q_i .

$$q_i = \lfloor \sqrt{v_i - 0.1} \rfloor + 1 \quad (14)$$

and the offset δ_i derived according to

$$\delta_i = -2 \cdot q_i \cdot \{(\Re(\hat{s}_i) > \tau_u) \cdot \text{sgn}(\Re(\hat{s}_0)) + j \cdot (\Im(\hat{s}_i) > \tau_u) \cdot \text{sgn}(\Im(\hat{s}_0))\} \quad (15)$$

The symbols enumerated by BSS-FE, $\{\tilde{s}_i\}_{i=0}^7$ are defined by adding δ to the symbols identified by the standard SSFE enumeration heuristic; i.e. $\{\hat{s}_i\}_{i=0}^7$ are as given by (9). Since \hat{s}_0 is always sliced to a valid constellation point, the additive offset given by (15) is only applied to the remaining enumerating symbols. With the defined enumerating bound, an offset is defined to regulate the invalid enumerating symbol back to the modulated constellation plate.

Fig. 9 presents an example of the effect of this translation for an enumeration path in S_4 . It shows how the symbols $\hat{s}_2 - \hat{s}_5$, enumerated by SSFE, but which are invalid, are mapped to valid points $\tilde{s}_2 - \tilde{s}_5$ via the additive offsets $\delta_2 - \delta_5$. Note that none of $\hat{s}_2 - \hat{s}_5$ would otherwise have been enumerated and in each case are the closest possible points to $\hat{s}_2 - \hat{s}_5$ whilst

increasing distance from it, in keeping with the original SSFE strategy.

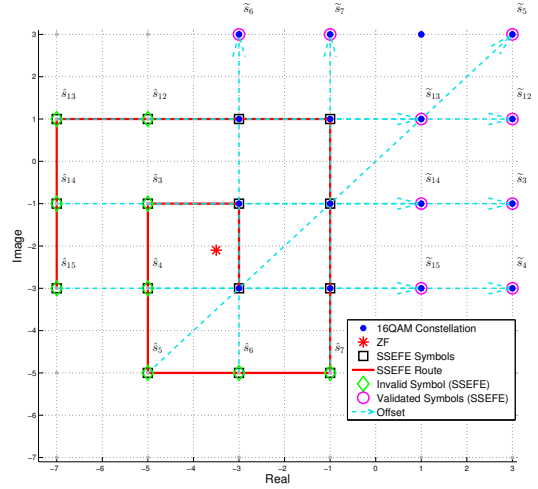


Fig. 9: Bounded Spanning

It should be noted that, whilst the addition of the offset will increase computational complexity of BSS-FE above the levels experienced by SSFE, this extension is in keeping with the reliance of SSFE on only simple arithmetic and logical operators, avoiding euclidean distance measurements or complex arithmetic operations such as square-root³. Applying BSS, the enumeration efficiency is 100%, with no invalid QAM constellation symbols considered during the enumeration process.

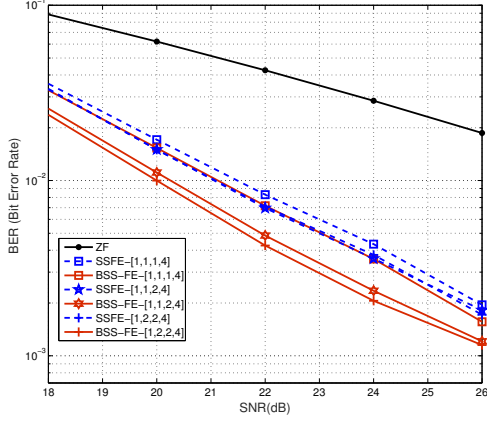
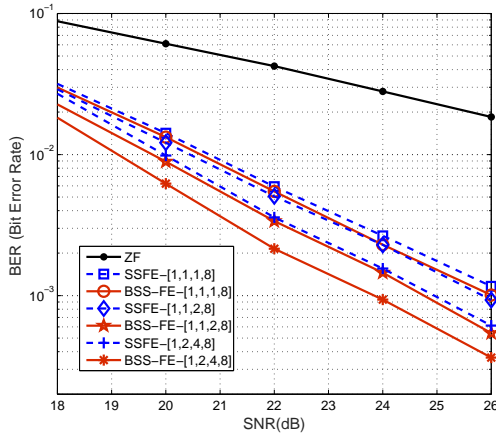
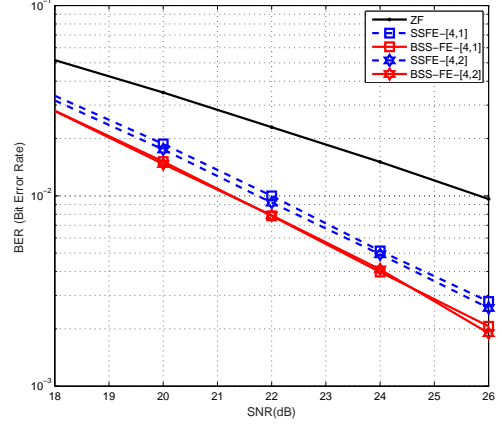
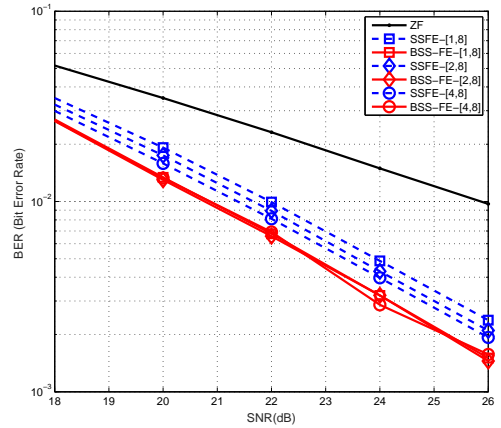
A. BSS-FE: BER v.s. Complexity

The BSS scheme proposed potentially increases detection performance since it eliminates the possibility of enumerating invalid points, which will always result in a symbol error. To illustrate this performance increase the Bit Error Rate (BER) performances of SSFE employing the standard enumeration scheme, and SSFE employing BSS (BSS-FE) are analysed in the context of an ideal MIMO OFDM fading channel with AWGN when perfect complex Gaussian cyclic-prefix (CP) is employed. Uncoded hard decisions are made in the context of 4×4 MIMO employing 16-QAM for 1×10^5 48-symbol frames, according to the 802.11n standard [2]. The BER for a variety of configuration vectors \mathbf{v} are shown in Fig. 10.

As shown, BSS-FE offers consistently superior BER performance to SSFE in all cases; for SSFE schemes enumerating more than one symbol on multiple levels, such as the $[1, 1, 2, 4]$, BSS-FE enables approximately 1.0 ~ 1.5 dB SNR gain beyond SSFE. For SSFE schemes enumerating more than one symbol in a single layer, such as $[1, 1, 1, 4]$ performance gain varies between 0.4 ~ 0.6 dB, with an average improvement of 0.5 dB.

This increased, performance, however, comes at increased computation cost - specifically, in order to calculate and apply

³Note that the square-root required for evaluation of (14) depends only on \mathbf{v} , while is defined at compile-time and hence may be evaluated off-line

(a) BSS-FE- $[x, x, x, 4]$ (b) BSS-FE- $[x, x, x, 8]$ Fig. 10: BSS-FE BER Performance, 4×4 (a) BSS-FE- $[x, 4]$ (b) BSS-FE- $[x, 8]$ Fig. 11: BSS-FE BER Performance, 2×2

the offsets in (12). Fig. 12 illustrates the increase in the number of arithmetic operations experienced as a result of employing BSS. As this shows, BSS-FE requires 4% - 10.2% (average 7.1%) more operations than SSFE. The relative scale of the increase reduces as the enumeration varies between $\mathbf{v} = [1, 1, 1, 4]$ to $\mathbf{v} = [1, 2, 4, 8]$. This increased cost, however, has an associated reduction in BER of 5% - 36.8% (20.9% on average.)

VI. EXTENDED FAST ENUMERATION FOR QAM

As outlined in Section III, \mathbf{v} represents the unique configuration of each SSFE scheme with each $v_i \in \mathbf{v}$ representing the number of symbols enumerated at stage i in the detection tree. However, the enumeration strategy outlined in Section III is limited to eight symbols. This has a number of benefits; it allowed the inventors to devise a low-complexity, iterative heuristic which can enumerate $n \in [1, 8]$ symbols confident, whichever number of symbols are enumerated, they are the n closest to the equalized symbol in Euclidean terms, and enumerated in order of increasing distance from it. Furthermore, this can be achieved without having to measure the Euclidean distance nor incur the cost of expensive square-root and comparison operations. However, it means that the

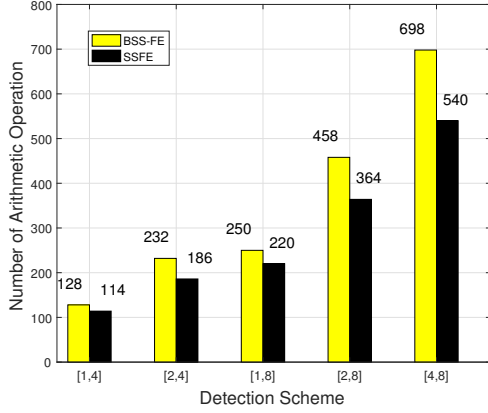
set of possible symbols for even moderately sized modulation alphabets, such as 16-QAM, cannot be fully enumerated, potentially restricting BER performance. In this section we propose an Extended Fast Enumeration (EFE) scheme to overcome this restriction.

The challenge in devising an EFE approach is to identify a heuristic which allows the same iterative nature, similarly avoids euclidean distance measurements, but is extensible to any number of symbols. This latter requirement in particular precludes hand-crafting optimised heuristics of the type used in SSFE. Hence, the EFE approach is an approximation motivated by the SSFE enumeration process.

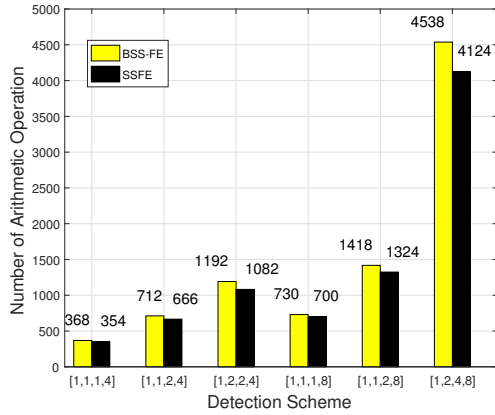
Fig. 13 illustrates an example enumeration route for 16-QAM symbols enumerated as a result of the same equalized symbol considered in Fig. 4. As it shows, the general approach is to enumerate symbols in either a 'spiral' of increasing radius around the initial estimate as that in Fig. 4⁴.

This is achieved by reformulating the original enumeration

⁴Note that, for clarity of focus on EFE, during this section we do not consider the issue of invalid symbol enumeration.



(a) 4 × 4



(b) 2 × 2

Fig. 12: BSS-FE Complexity Comparisons

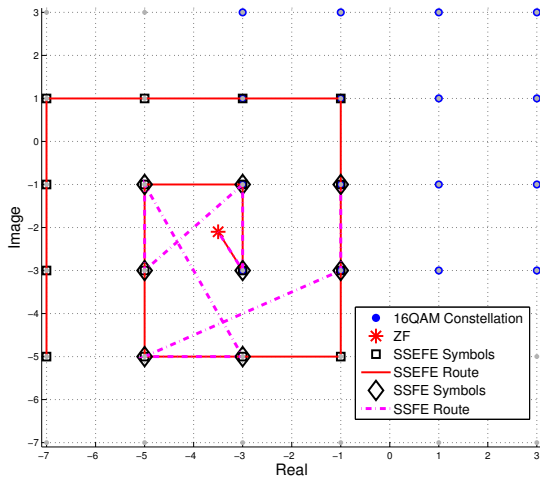


Fig. 13: Extended Fast Enumeration

conditions in (9) to the alternative in (16).

$$\begin{cases} \hat{s}_0 = \mathcal{Q}(\widehat{y}_{ZF_{nt,i}}), i \in [1, M_c^{N_t}], nt \in [1, N_t] \\ \hat{s}_{k+1} = \hat{s}_k + z_1 \cdot (-1)^{w+1}, \\ \hat{s}_{l+1} = \hat{s}_l + z_2 \cdot (-1)^{w+1}, \end{cases} \quad (16)$$

where $\widehat{y}_{ZF_{nt,i}}$ denotes the nt^{th} ZF equalized signal with

canceled noise for i^{th} enumerating symbol.

This approach ensures a consistently increasing distance from the initial symbol estimate but is not limited to 8 points but rather is directly controlled by the enumerating configuration vector \mathbf{v} via the terms w , k and l in (17).

$$\begin{cases} w \in [1, \lceil \sqrt{v_i + 0.25} - 0.5 \rceil], \\ k \in [(w-1) \cdot w + 1, w \cdot w], \\ l \in [w \cdot w + 1, (w+1) \cdot w]. \end{cases} \quad (17)$$

However, whilst these control the scale of enumeration, they do not influence the direction, which may proceed in either a clockwise or anticlockwise direction around \hat{s}_0 . The direction is important since it can have a strong influence, for the same number of enumerated points, on the number which are valid. This direction is controlled by z_1 and z_2 in (16) as defined in (18).

$$\begin{cases} z_1 = \text{sgn}(\Re(d)) \cdot \phi + j \cdot \text{sgn}(\Im(d)) \cdot (!\phi), \\ z_2 = \text{sgn}(\Re(d)) \cdot (!\phi) + j \cdot \text{sgn}(\Im(d)) \cdot \phi, \end{cases} \quad (18)$$

Consideration of this approach makes a number of important points clear. Firstly, it is important to note that complex arithmetic operations, such as square-root, are not required at run-time⁵, maintaining the commitment of SSFE to an enumeration process dependent on only simple arithmetic and logical operations. In addition, the enumeration strategy is not optimal - symbols are not enumerated in the same order as they are in SSFE; rather it a heuristic approximation in the same vain, extended to any number of symbols. Given, however, that is is an approximation, it is important to consider whether a serious impact is experienced on detection performance.

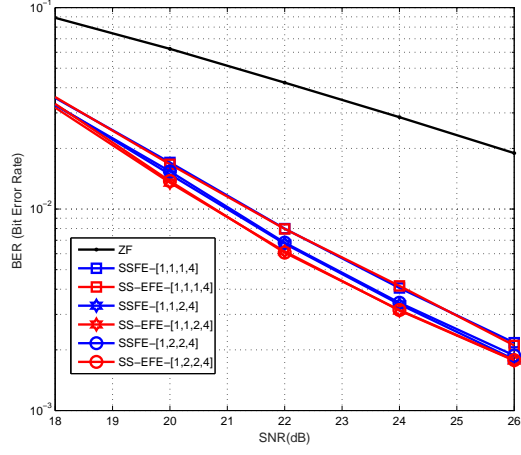
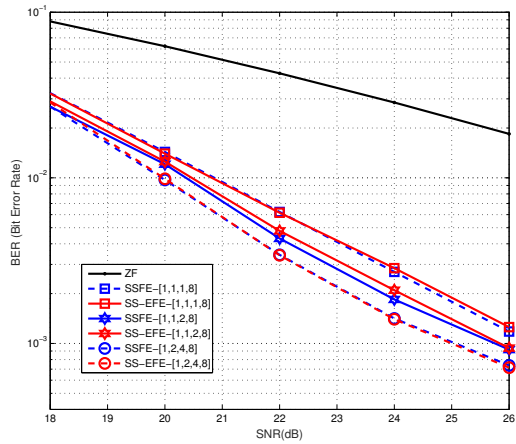
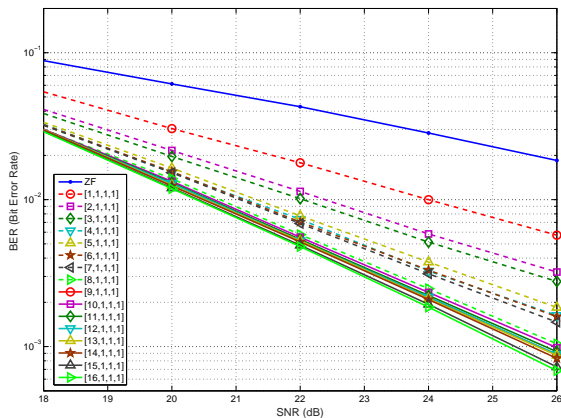
Fig. 14 illustrates the BER of a series of SSFE and SS-EFE configurations for 4×4 16-QAM MIMO systems, under similar simulation conditions employed in Section IV. As this shows, the performance difference between SSFE and SS-EFE for these configurations is negligible.

VII. BSS-EFE: COMPLEXITY AND PERFORMANCE ANALYSIS

This section measures the complexity and performance consequences of the novel enumeration strategies employed in BSS-EFE. Fig. 15 shows the BER performance of a variety of BSS-EFE configurations for 4×4 16-QAM MIMO exploiting V-BLAST antenna ordering (i.e. antennas are decoded in order of decreasing received signal power). The configurations span the least complex (enumeration of a single symbol at each antenna) up to full enumeration SSFE-[16,1,1].

As shown, the performance of the detector steadily increases with the number of symbols enumerated and, accordingly, the complexity of the detector. The arithmetic complexity of SSFE, SS-EFE, BSS-FE and BSS-EFE are described in Table I. The number of symbols which may be enumerated by SSFE is configurable, with an upper limit of eight. As defined in (9), the complexity of enumerating each symbol is variable and hence the enumeration complexity is variable depending

⁵The square-root used to evaluate w in (17) is dependent only on \mathbf{v} and may be pre-computed off-line

(a) SS-EFE- $[x, x, x, 4]$ (b) SS-EFE- $[x, x, x, 8]$ Fig. 14: SS-EFE BER Performance, 4×4 Fig. 15: BSS-EFE BER for 4×4 16-QAM MIMO

on the number of symbols enumerated at each search tree layer. To allow for this complexity, we use the vectors $\mathbf{add} = [0, 4, 6, 8, 9, 10, 11, 12]$ and $\mathbf{mul} = [0, 4, 6, 6, 6, 7, 7, 8]$, where each $\mathit{add}_i \in \mathbf{add}$ ($\mathit{mul}_i \in \mathbf{mul}$) represents the number of addition/subtraction (multiplication) operations required to

enumerate $\hat{s}_i \in [1, 8]$. Based on these, the complexity of SSFE may be represented as per the first row in Table I. Note that, since the only differences between SSFE, BSS-FE and BSS-EFE lie in the process of identifying and enumerating symbols, the other computational costs (most particularly, the APED calculations) are not included in this comparison. Note that, whilst BSS-FE, SS-EFE and BSS-EFE increase complexity relative to SSFE, they do not increase the order of the complexity and hence, in the limit, exhibit similar computational complexity.

Tables II and III enumerate the total number of arithmetic calculations required for a series of configurations of SSFE, BSS-FE, SS-EFE, and BSS-EFE. As shown, the bounded spanning and extended fast enumeration strategies each increase the complexity of the detection process. This complexity increment is only desirable in the case where disproportionately large increases in detection performance result.

Fig. 16 compares the BER performance and arithmetic complexity of a number of SSFE and BSS-EFE schemes for 4×4 , 16-QAM MIMO detection under similar experimental conditions as outlined in Section IV. As this shows, both BSS-EFE- $[1, 2, 2, 12]$ and BSS-EFE- $[1, 1, 4, 12]$ offer performance gain of more than 1 dB SNR at BER of 10^{-3} over SSFE- $[1, 2, 4, 8]$. This performance benefit accrues despite the fact that the BSS-EFE schemes enumerate, respectively, 132 and 156 symbols (as compared to the 168 enumerated by SSFE- $[1, 2, 4, 8]$) and requiring respectively 3496 and 3880 arithmetic operations, as compared to the 4124 required by SSFE- $[1, 2, 4, 8]$. Similarly, for 2×2 systems Fig. 17 compares a series of SSFE and BSS-EFE configurations. As shown, BSS-EFE- $[1, 15]$ exhibits similar detection performance to SSFE- $[4, 8]$ despite enumerating fewer symbols (30 as compared to 32 for SSFE- $[4, 8]$).

These observations point a a major opportunity enabled by BSS-EFE - it allows more efficient balancing of computational effort across the layers of the decoding tree. Specifically, by enumerating larger numbers of symbols in the first levels of the tree and fewer at lower levels, BER is potentially increased even whilst complexity is reduced. This is an approach promoted by FSD [19], [21], but which is not configurable.

The relation to FSD also suggests that it may be worthwhile to consider the use of alternative ordering strategies when decoding multiple antennas. SSFE uses a V-BLAST ordering, decoding antennas in decreasing order of signal power, but FSD adopts this approach for a subset of the antennas, before reversing the order to enumerate antennas in increasing order of received signal power. When this ordering approach is adopted for 4×4 MIMO detection with 16-QAM, the BER of a subset of the BSS-EFE configurations are as illustrated in Fig. 18.

The benefit of this approach is clear, with obvious improvements in BER performance for all 7 configurations illustrated in Fig. 18 relative to their counterparts in Fig. 15. Fig. 19 illustrates the true benefits of this capability. BSS-EFE- $[15, 1, 1, 1]$ achieves close to the BER performance of SSFE- $[1, 2, 4, 8]$, with less than 0.2 dB SNR loss but with complexity reduced substantially by 53.26%. Similarly, BSS-

TABLE I: Enumeration Arithmetic Complexity

	\pm	\times
SSFE	$\sum_{i=1}^{N_t} (\mathbf{add}_{v_i} \cdot (\prod_{j=1}^{i-1} v_j))$	$\sum_{i=1}^{N_t} (\mathbf{mul}_{v_i} \cdot (\prod_{j=1}^{i-1} v_j))$
BSS-FE	$\sum_{i=1}^{N_t} (\mathbf{add}_{v_i} \cdot (\prod_{j=1}^{i-1} v_j)) + \sum_{i=1}^{N_t-1} ((v_i - 1) \cdot \prod_{j=1}^{i-1} v_j \cdot (v_j > 1) \cdot 2)$	$\sum_{i=1}^{N_t} (\mathbf{mul}_{v_i} \cdot (\prod_{j=1}^{i-1} v_j)) + \sum_{i=1}^{N_t} (\prod_{j=1}^{i-1} (v_j > 1) \cdot 2) + (v_1 > 1) \cdot 2 + \sum_{i=2}^{N_t} ((v_i - 1) \cdot \prod_{j=1}^{i-1} v_j \cdot (v_j > 1) \cdot 2)$
SS-EFE	$\sum_{i=2}^{N_t} (\prod_{j=1}^{i-1} v_j \cdot (v_j > 1) \cdot 2) + (v_1 > 1) \cdot 2 + \sum_{i=1}^{N_t-1} ((v_i - 1) \cdot \prod_{j=1}^{i-1} v_j \cdot (v_j > 1) \cdot 2)$	$\sum_{i=2}^{N_t} (\prod_{j=1}^{i-1} v_j \cdot (m_i > 1) \cdot 6) + (v_1 > 1) \cdot 6$
BSS-EFE	$\sum_{i=2}^{N_t} (\prod_{j=1}^{i-1} v_j \cdot (v_j > 1) \cdot 2) + (v_1 > 1) \cdot 2 + \sum_{i=1}^{N_t-1} ((v_i - 1) \cdot \prod_{j=1}^{i-1} v_j \cdot (v_j > 1) \cdot 4)$	$\sum_{i=2}^{N_t} (\prod_{j=1}^{i-1} v_j \cdot (v_i > 1) \cdot 8) + (v_1 > 1) \cdot 8 + \sum_{i=2}^{N_t} ((v_i - 1) \cdot \prod_{j=1}^{i-1} v_j \cdot (v_j > 1) \cdot 2)$

TABLE II: Complexity Comparisons - 2×2

\mathbf{v}	[1,4]	[2,4]	[1,8]	[2,8]	[4,8]
SSFE	119	211	225	409	585
SS-EFE	127	235	243	459	651
BSS-FE	144	296	274	578	850
BSS-EFE	152	312	292	612	900

TABLE III: Complexity Comparisons - 4×4

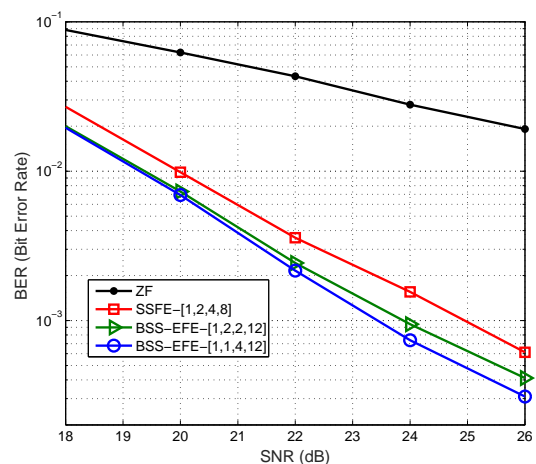
\mathbf{v}	[1,1,1,4]	[1,1,2,4]	[1,2,2,4]	[1,1,1,8]	[1,1,2,8]	[1,2,4,8]
SSFE	354	666	700	1082	1324	4124
SS-EFE	354	666	1082	702	1326	4110
BSS-FE	368	712	1192	730	1418	4538
BSS-EFE	368	704	1168	732	1404	4444

EFE-[15, 1, 1, 1] enables 2 dB SNR and 0.6 dB SNR gain respectively over SSFE-[1, 2, 4, 8] and BSS-EFE-[1, 2, 4, 8] while maintaining similar levels of complexity reduction.

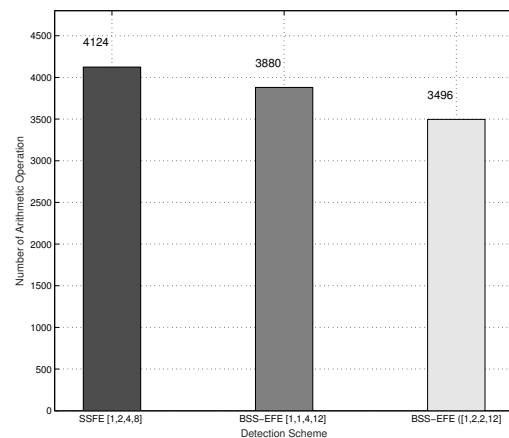
It is clear that, despite incurring a complexity increase relative to SSFE when similar configurations are considered, BSS-EFE offers increased detection performance due to its ability to avoid enumerating invalid symbols. However, it is also clear that the ability of BSS-EFE to enable configurations which SSFE cannot, due to its ability to enumerate only eight symbols, means that even for moderately dense QAM constellations, such as 16-QAM, BSS-EFE offers a very effective combination of higher detection performance and lower computation cost. Section VIII considers the performance and cost implications of this reduced complexity for real implementations.

VIII. BSS-EFE: PERFORMANCE AND COST

A series of SSFE and BSS-EFE detectors have been created for Xilinx Virtex FPGA using the FPGA Processing Element (FPE), shown to be very effective for realising sphere decoder accelerators in [22]. In all cases Virtex-5 or Virtex-6 technology is targeted (with the choice dependent on the scale of the architectures). To provide a realistic comparison scenario, all architectures are created to minimise cost and provide real-time throughput for 4×4 and 2×2 802.11n MIMO.



(a) BER Performance



(b) Arithmetic Operations

Fig. 16: SSFE/BSS-EFE Comparison, 4×4 16-QAM MIMO

The crucial architectural features of the FPE realisations (specifically, the number of SIMD units employed) and the FPGA performance and cost metrics are quoted and compared with SSFE alternatives in Tables IV - VI.

TABLE IV: 4×4 16-QAM BSS-EFE Implementations

Scheme \mathbf{v}	SSFE						BSS-EFE		
	[1,1,1,4]	[1,1,2,4]	[1,2,2,4]	[1,1,1,8]	[1,1,2,8]	[1,2,4,8]	[1,2,2,12]	[1,1,4,12]	[1,1,1,15]
SIMDs	6	6	11	8	12	34	39	40	10
DSP48E1	54	75	135	84	150	578	429	495	160
LUT ($\times 10^3$)	9.8	17.3	30.3	26.3	34.1	212.1	99.6	112.6	47.7
Clock (MHz)	351	347	341	352	343	267	283	252	337
T (Mbps)	534.9	488.5	487.1	491.5	483.3	532.5	497.6	484.8	485.3

TABLE V: 2×2 16-QAM Implementations

Scheme \mathbf{v}	SSFE				BSS-EFE				
	[1,4]	[1,8]	[2,4]	[2,8]	[1,4]	[1,8]	[2,4]	[2,8]	[1,15]
SIMDs	3	4	3	4	3	7	6	7	8
DSP48E1	28	39	28	72	24	63	50	70	100
LUT ($\times 10^3$)	3.8	7.1	4.6	11.6	3.8	11.5	7.6	14.0	21.37
Clock (MHz)	357	347	362	350	361	337	353	343	320
T (Mbps)	241.3	241.5	281.4	257.2	245.9	240.7	254.1	244.7	273.4

TABLE VI: 2×2 4-QAM Implementations

Scheme \mathbf{v}	SSFE			BSS-EFE		
	[1,2]	[1,4]	[2,4]	[1,2]	[1,4]	[2,4]
SIMDs	3	3	4	3	3	4
DSP48E1	16	28	54	24	24	56
LUT ($\times 10^3$)	2.3	4.3	7.5	3.6	3.8	7.9
Clock (MHz)	363	363	363	362	361	351
T (Mbps)	123.8	140.6	136.6	122.0	123.0	133.9

The relative cost BSS-EFE compared to SSFE are illustrated in Fig. 20 and Fig. 21. As expected, when similar configurations are compared, BSS-EFE architectures generally require greater resources in order to meet the throughput requirements - up to 29.1% more LUTs ($\mathbf{v} = [1, 1, 1, 8]$) and 16.7% more DSP48E1 units ($\mathbf{v} = [1, 1, 1, 4]$). However, as highlighted in Section VII, this increased cost is accompanied by superior detection performance and reduced BER.

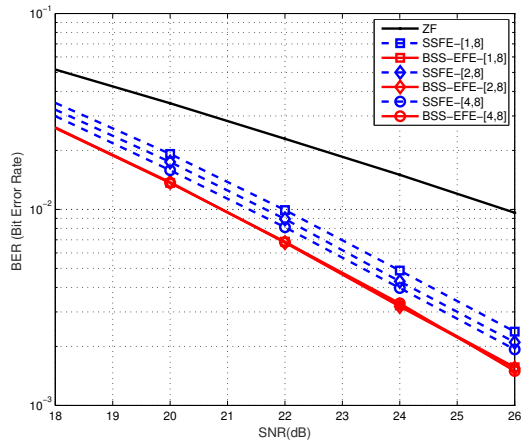
However, Section VII also reveals a secondary benefit of BSS-EFE - its ability to support configurations which SSFE cannot and which enable substantially better detection performance and reduced cost. For example, BSS-EFE-[1, 2, 2, 12] was shown in Section VII was shown to reduce complexity by 16% relative to SSFE-[1, 2, 4, 8], whilst offering superior BER performance; on implementation, this translates to reductions in LUT and DSP48E1 cost by 47% and 21% respectively. Similarly, BSS-EFE-[1, 1, 1, 15] requires 25% and 29% of the respective LUT and DSP48E1 costs of SSFE-[1, 2, 4, 8], but achieves the same BER performance mentioned. A similar scenario holds for 2×2 MIMO - for instance, BSS-EFE-[15, 1] enables reduced BER as compared to SSFE-[4, 8], yet reduces DSP48E1 cost by 26%.

IX. SUMMARY

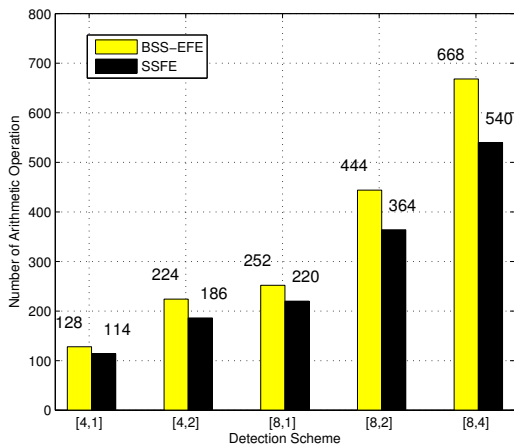
IoT MIMO equipment vary dramatically in scale, operating contexts and energy, cost and performance budgets. When realising embedded transceivers for such equipment, it is therefore vital that designers are able to trade performance and cost, within this operating context.

Symbol detection is a particular operation of concern; the leading approaches to supporting this design process, such as SSFE, include a number of operating inefficiencies and limitations with constrain performance and efficiency for even moderate MIMO topologies. This paper has introduced BSS-EFE, which overcomes these limitations. Specifically, BSS-EFE uses novel approaches to low-complexity spanning of the modulation constellation and fast enumeration of symbols which eliminate the redundancies and constraints inherent in SSFE. On a like-for-like basis this increases cost, but it increases performance also. But the central purpose of BSS-EFE is to enable increased performance at lower cost than SSFE by enabling configurations which SSFE cannot. This work has shown that these more efficient configurations can increase detection performance whilst also enabling substantial reductions in complexity and implementation cost. When realised on Xilinx FPGA, these reduced aspects of the system cost by up to 47%, whilst simultaneously reducing error rates.

Despite these promising foundations, notable avenues for further work remain to be explored. Prominent is the investigation of hybrid or 'soft' detection approaches, which resolve both the detection and decoding problems in a single component. Soft detection approaches harnessing BSS-EFE have not yet been proposed. Similarly, the BSS and EFE heuristics proposed are by no means the only ones possible, but are likely representative of families of heuristics which may solve the same problem, each with differing performance and cost requirements. It is the authors' intention that this



(a) BER Performance



(b) Arithmetic Operations

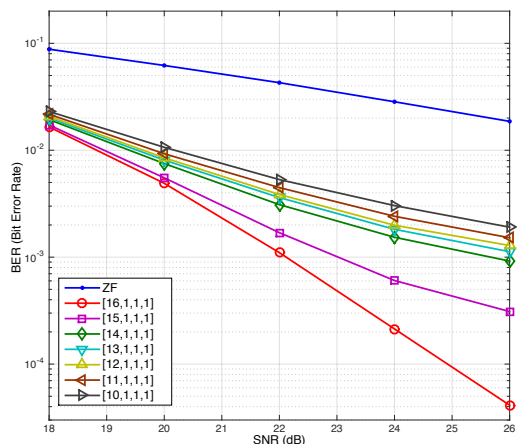
Fig. 17: SSFE/BSS-EFE Comparison, 2×2 16-QAM MIMO

Fig. 18: BSS-EFE BER with FSD Ordering

work should serve as motivation for these investigations.

REFERENCES

[1] P. Wolniansky, G. Foschini, G. Golden, and R. Valenzuela, "V-BLAST: An Architecture for Realizing Very High Data Rates Over The Rich-

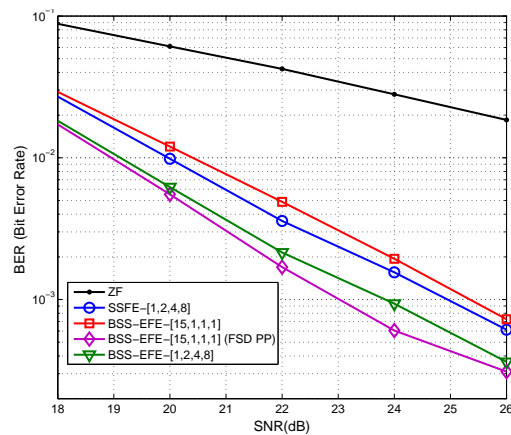
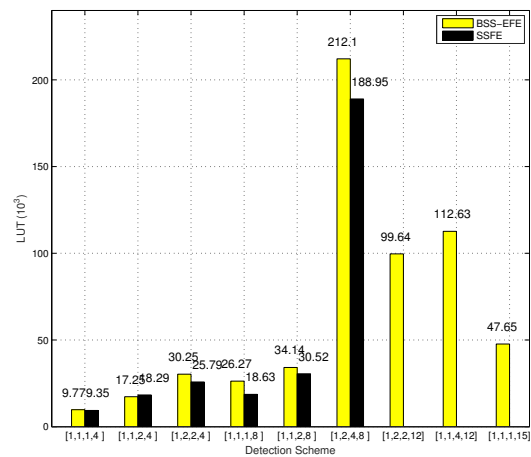
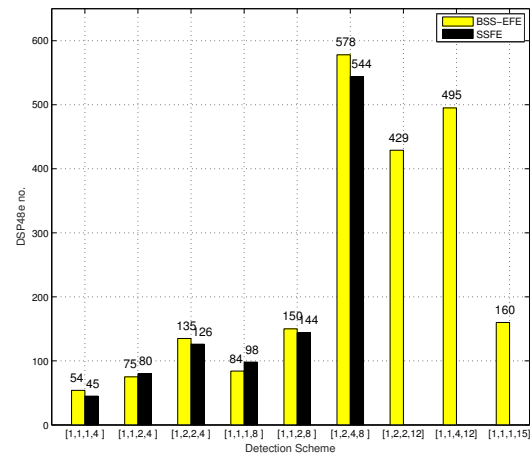


Fig. 19: BSS-EFE BER Performance (various ordering)



(a) LUT Cost



(b) DSP48E Cost

Fig. 20: Resource Cost Comparison, 4×4

Scattering Wireless Channel," in 1998 *URSI International Symposium on Signals, Systems, and Electronics, 1998. ISSSE 98.*, 1998, pp. 295–300.

[2] IEEE Standards Association, *802.11n-2009 IEEE Local and Metropolitan Area Networks Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput.* IEEE, 2009.

[3] E. Dahlman, S. Parkvall, and J. Skold, *4G LTE/LTE-Advanced for Mobile Broadband*, 1st ed. Academic Press, 2011.

[4] A. Gohil, H. Modi, and S. K. Patel, "5G Technology of Mobile Com-

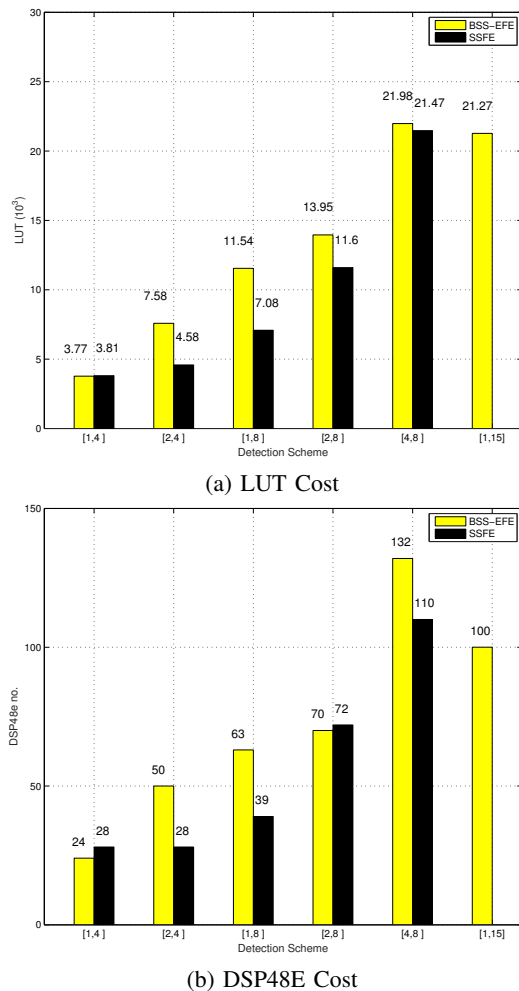


Fig. 21: Resource Cost Comparison, 2×2

munication: A Survey,” in *2013 International Conference on Intelligent Systems and Signal Processing (ISSP)*, March 2013, pp. 288–292.

- [5] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, “Massive MIMO for Next Generation Wireless Systems,” *IEEE Communications Magazine*, vol. 52, no. 2, pp. 186–195, February 2014.
- [6] A. Sibille, C. Oestges, and A. Zanella, *MIMO: From Theory to Implementation*, 1st ed. Academic Press, 2010.
- [7] T. Chiueh, P. Tsai, and I. Lai, *Baseband Receiver Design for Wireless MIMO-OFDM Communications*. John Wiley & Sons Ltd, 2012.
- [8] M. Li, B. Bougard, D. Novo, W. V. Thillo, L. V. der Perre, and F. Catthoor, “Adaptive SSFE Near-ML MIMO Detector With Dynamic Search Range and 80-103 Mbps Flexible Implementation,” in *IEEE GLOBECOM 2008 - 2008 IEEE Global Telecommunications Conference*, Nov 2008, pp. 1–5.
- [9] Z. Guo and P. Nilsson, “Algorithm and Implementation of the K-best Sphere Decoding for MIMO Detection,” *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 3, pp. 491–503, March 2006.
- [10] M. Li, B. Bougard, E. E. Lopez, A. Bourdoux, D. Novo, L. V. D. Perre, and F. Catthoor, “Selective Spanning with Fast Enumeration: A Near Maximum-Likelihood MIMO Detector Designed for Parallel Programmable Baseband Architectures,” in *2008 IEEE International Conference on Communications*, May 2008, pp. 737–741.
- [11] Y. Wu and J. McAllister, “Bounded Selective Spanning with Extended Fast Enumeration for MIMO-OFDM Systems Detection,” in *21st European Signal Processing Conference (EUSIPCO 2013)*, Sept 2013, pp. 1–5.
- [12] E. Larsson, “MIMO Detection Methods: How They Work [Lecture Notes],” *IEEE Signal Processing Magazine*, vol. 26, no. 3, pp. 91–95, May 2009.
- [13] M. O. Damen, H. E. Gamal, and G. Caire, “On Maximum-Likelihood Detection and The Search for the Closest Lattice Point,” *IEEE Trans-*

actions on Information Theory, vol. 49, no. 10, pp. 2389–2402, Oct 2003.

- [14] C. P. Schnorr and M. Euchner, “Lattice Basis Reduction: Improved Practical Algorithms and Solving Subset Sum Problems,” in *Mathematical Programming*, August 1994, pp. 181–199.
- [15] B. Hassibi and H. Vikalo, “On the Sphere-Decoding Algorithm I. Expected Complexity,” *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2806–2818, Aug 2005.
- [16] D. Wubben, R. Bohnke, J. Rinas, V. Kuhn, and K. D. Kammeyer, “Efficient Algorithm for Decoding Layered Space-Time Codes,” *Electronics Letters*, vol. 37, no. 22, pp. 1348–1350, 2001.
- [17] T.-D. Chiueh and P.-Y. Tsai, *OFDM Baseband Receiver Design for Wireless Communications*. Wiley Publishing, 2007.
- [18] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, *Introduction to Algorithms*, 3rd ed. McGraw-Hill Higher Education, 2009.
- [19] L. G. Barbero and J. S. Thompson, “Fixing the Complexity of the Sphere Decoder for MIMO Detection,” *IEEE Transactions on Wireless Communications*, vol. 7, no. 6, pp. 2131–2142, June 2008.
- [20] M. Li, B. Bougard, W. Xu, D. Novo, L. V. D. Perre, and F. Catthoor, “Optimizing Near-ML MIMO Detector for SDR Baseband on Parallel Programmable Architectures,” in *2008 Design, Automation and Test in Europe*, March 2008, pp. 444–449.
- [21] L. G. Barbero and J. S. Thompson, “Performance Analysis of a Fixed-Complexity Sphere Decoder in High-Dimensional MIMO Systems,” in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 4, May 2006, pp. IV–IV.
- [22] X. Chu and J. McAllister, “Software-Defined Sphere Decoding for FPGA-Based MIMO Detection,” *IEEE Transactions on Signal Processing*, vol. 60, no. 11, pp. 6017–6026, Nov 2012.



Yun Wu received the B.S. degree in Electronic and Information Engineering from Dalian Nationalities University, Dalian, China, in 2003; the M.Sc. in Circuits and System from Hunan University, Changsha, China, in 2007; the M.Sc. in Radio Frequency Communication Systems from University of Southampton, Southampton, U.K. 2008 and the Ph.D. degree in Electronic Engineering from Queen’s University Belfast, Belfast, U.K. in 2014. Before that, he was a wireless algorithm engineer with ZTE Shanghai R & D Centre from 2008 to 2010. He is currently a Research Fellow with Queen’s University Belfast. His current research interests include signal processing, processor architecture synthesis, and energy proportional computing.



John McAllister (S’02-M’04-SM’12) received the Ph.D. degree in Electronic Engineering from Queen’s University Belfast, U.K., in 2004. He is currently a member of academic staff in the Institute of Electronics, Communications and Information Technology (ECIT) at the same institution. His research focusses on embedded realisation of high performance signal processing architecture, in particular modelling languages, compilers and architectures for FPGA-based signal processing. He is a cofounder of Analytics Engines Ltd., a member of the Advisory Board to the IEEE Signal Processing Society Technical Committee on Design and Implementation of Signal Processing Systems (DISPS), an Associate Editor of IEEE TRANSACTIONS ON SIGNAL PROCESSING, Chief Editor of SigView and a member of the editorial board of Springer’s Journal of Signal Processing Systems.