Modified K-best Receiver for Multi-antenna Vehicular Networks

Shuangshuang Han*†, Fenghua Zhu‡§, Yingchun Wang*, Dongpu Cao†¶,
Gang Xiong*‡ and Fei-Yue Wang*, *Fellow, IEEE*
*The State Key Laboratory of Management and Control for Complex Systems,
Institute of Automation Chinese Academy of Sciences, Beijing, China
†Qingdao Academy of Intelligent Industries, Qingdao, Shandong Province, China.
‡The Cloud Computing Center, Chinese Academy of Sciences, Dongguan 523808, China.
§The Beijing Engineering Research Center of Intelligent Systems and Technology,
Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China
¶The Center for Automotive Engineering, Cranfield University, Cranfield MK43 0AL, U.K.

***Abstract*—Nowadays, with the fast growth of Internet of Vehicles (IoV), wireless communication techniques for IoV is becoming more and more important for vehicular networks. To achieve more reliable information and reduce the computational complexity, receiver detection is one of the most significant techniques. In this paper, several modified K-best detection algorithms are developed to take advantages of the list decoder, which provide a flexible performance and complexity trade-off. Different from the existing algorithms, the proposed K-best algorithms could make use of the a priori probability to generate the list. Simulation results demonstrate that the proposed low complexity algorithms can achieve a significant performance gain over existing ones, especially for the networks with high order constellations.**

I. Introduction

Vehicular networks (VANETs) are gaining more and more attention because of the rapid development of the wireless communication technologies. Thus, a huge amount of services may be devised for much smarter transportation systems. VANETs are regarded as the most promising technologies to support much safer and more efficient transportation systems, which enable vehicles to quickly and accurately collect significant and essential traffic information and simultaneously notice other neighbouring vehicles.

With the increasing number of wireless devices accessing VANETs, the data rate demand and the system capacity drive the need for more sophisticated communications technologies. For the improvements in reliability and transmit speed of the information transmission, multiple antennas techniques are recommended to exploit spatial diversity for VANETs, which use multiple antennas at both the transmitter and receiver to improve communication performance. Multiple-input multiple-output (MIMO) [1], [2] obtains significant improvements in data throughput and link range without extra bandwidth or transmit power. In MIMO systems, detection methods at receivers play an important role in communication performance and system complexity.

Corresponding author: Gang Xiong, email: gang.xiong@ia.ac.cn

As known, the maximum likelihood (ML) detection achieves the optimal system performance, but its complexity increases with increasing number of antennas and the order of constellation. Thus, a low complexity technique referred to as sphere decoding (SD) is proposed for lattice code decoding. This SD approach is extended to coded MIMO systems [4], which iteratively detect and decode any linear space-time mapper combined with an outer channel code. Computing the exact log likelihood ratio (LLR) [4] requires a complexity exponential in the number of antennas and in the size of the constellation. A list SD and max-log approximation are used to approach the optimal performance with low complexity. However, the complexity of SD based algorithms depends on signal-to-noise (SNR). Consequently, a K-best SD is proposed for both uncoded and coded MIMO systems [5], which has a constant complexity across whole SNR region. Another different approach [6]–[9] to adopt soft information in iterative detection and decoding is using nonlinear interference cancellation.

In this paper, our main idea is to develop a class of algorithms that can provide a flexible performance and complexity trade-off for vehicular networks. In the proposed modified algorithms, we also generate a list of K lattice points. Different from [4] which uses max-log, we make use of all the lattice points in the list to generate the LLR value, i.e., sum-log. The way to generate the list is also different from that in [4]. In the proposed algorithms, the a priori probability is used to update the list at each stage, where the a priori probability is approximated using Gaussian and non-Gaussian approximations. We also discuss several variations of the basic algorithms and efficient implementation in MIMO-OFDM systems. Finally, simulation results approve that the proposed algorithms achieve a superior performance over existing ones particularly for the system with high order modulation.

II. System Model

In this paper, as shown in Fig. 1, a MIMO-OFDM system with $\tilde{M}$ transmit antennas and N receive antennas is

978-1-5386-1526-3/17/$31.00 ©2017 IEEE

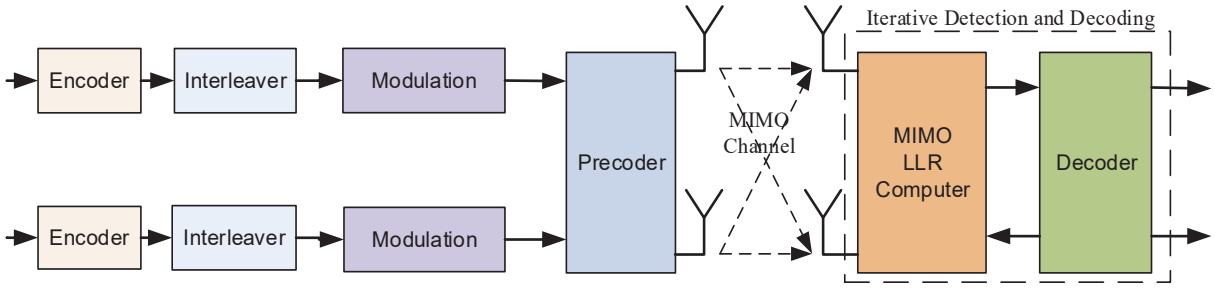

Fig. 1. System Model.

considered. In an OFDM block, there are N_s subcarriers and M transmitted data streams[1]. The constellation $\mathcal{Q}_m$ is used for stream m, and C_m denotes the number of bits per constellation symbol. The incoming bits of each stream m of length $N_s C_m R_m$, $m = 1, \ldots, M$ is encoded by a channel code with rate R_m, resulting in a bit vector $\mathbf{b}_m$. The encoded bits are then modulated by a mapping function $x_{i,m} = \mathcal{M}_m(\mathbf{b}_m((i-1)C_m + 1 : iC_m))$, $i = 0, ..., N_s - 1$, where $x_{i,m}$ is the transmitted symbol over the i-th subcarrier and the m-th antenna. The time domain sequence is then generated by the Inverse Discrete Fourier Transform (IDFT) of the data block $x_{0,m}, \ldots, x_{N_s-1,m}$, which is derived as

$$X_{j,m} = \frac{1}{\sqrt{N_s}} \sum_{i=0}^{N_s-1} x_{i,m} e^{j2\pi ij/N_s}, j = 0, \ldots, N_s - 1. \quad (1)$$

The time domain symbol $X_{j,m}$ is assumed to satisfy the component-wise energy constraint $E\|X_{j,m}\|^2 = \mathcal{E}_s/M$. A cyclic prefix (CP) is added to mitigate the residual intersymbol interference (ISI) because of the previous OFDM symbol. After parallel-to-serial (P/S) conversion, the signal is transmitted from the corresponding antenna. The channel between m-th transmit antenna and n-th receive antenna is given as

$$h_{n,m}(t) = \sum_{l=0}^{\Gamma_{n,m}-1} \alpha_{n,m,l} \delta(t - \tau_{n,m,l}), \quad (2)$$

where $\Gamma_{n,m}$ denotes the number of taps, $\alpha_{n,m,l}$ is the l-th complex path gain, and $\tau_{n,m,l}$ is the corresponding path delay. In this paper, the channel is set to be constant for each OFDM data block, i.e. block fading channel.

At the receiver, after the corresponding process, the received signal in frequency domain is derived as

$$y_{i,n} = \sum_{m=1}^{M} H_{i,n,m} x_{i,m} + w_{i,n}, \quad (3)$$

where $i = 0, \ldots, N_s - 1; n = 1, \ldots, N$, n is the receiver antenna index, $w_{i,n}$ is the additive white Gaussian noise (AWGN) with 0 mean and variance σ^2, and

$$H_{i,n,m} = \frac{1}{\sqrt{N_s}} \sum_{l=0}^{\Gamma_{n,m}-1} \alpha_{n,m,l} e^{-j2\pi \lceil \tau_{n,m,l}/T_s \rfloor i/N_s}, \quad (4)$$

[1]We have $M \leq \tilde{M}$ due to possible beamforming at the transmitter. In this case, we consider an equivalent channel with M transmit antennas

where T_s means the symbol duration. The vector form of Eq. (3) is given as

$$\mathbf{y}_i = \mathbf{H}_i \mathbf{x}_i + \mathbf{w}_i, i = 0, \ldots, N_s - 1. \quad (5)$$

It is essential to mention that we can consider Eq. (5) as a MIMO system on each subcarrier. For simplicity, the subscript i in Eq. (5) is neglected for the following parts of this paper.

III. Iterative Detection and Decoding

At the receiver, several iterations of soft information exchange [4] occur between the decoder and MIMO detector (Fig. 1). The MIMO detector in this case generates soft *a posteriori* information $\mathbf{L}$ by processing the received signal $\mathbf{y}$ and the *a priori* information $\mathbf{L}_A$ from the decoder. This reliability information is expressed by *a posteriori* probability (APP) in the form of log-likelihood ratios (LLR). For example, The LLR of bit $x_i (i = 1, 2, \ldots, M_x)$ is defined as

$$L(x_i) = \log \frac{\Pr[x_i = +1]}{\Pr[x_i = -1]}. \quad (6)$$

Note that the amplitude levels -1 and $+1$ represent binary 0 and 1, respectively.

For the first iteration, the $\mathbf{L}_A$ is initialized to 0, and the extrinsic information $\mathbf{L}_E = \mathbf{L} - \mathbf{L}_A$ generated by the MIMO detector is deinterleaved to serve as the *a priori* information for the decoder. The decoder then generates the extrinsic information for the next iteration. This process continues until a stopping criterion is met, such as a predefined iteration number or a performance bound. In the final iteration, the decoder obtains the *a posteriori* information $\mathbf{L}_D$ on the uncoded bits $\mathbf{b}$, which is sent to the slicer that outputs the final bit estimates $\hat{\mathbf{b}}$.

As discussed above, soft information needs to be exchanged between the detector and decoder. The naive SD algorithm can be modified to give soft information. One jointly iterative detection and decoding method has been proposed [4], which generates soft information by a list version of the SD (LSD).

IV. Modified K-best Algorithms

A. Motivation

There are several issues with the existing algorithms in Section II.

- Many practical wireless communications standards (e.g., LTE) now adopt high order constellations such as

64QAM or large number of antennas. The max-log approximation in [4] may not work well with high order constellations as the number of terms in the summation computation is large. Moreover, the LSD may be hard to be implemented in hardware directly due to its sequential nature.

- The Gaussian approximation based algorithms avoid the max-log approximation but the Gaussian assumption incurs some performance loss. It is commented in [8] that the performance of Gaussian approximation algorithms is not good for higher order modulations.

We wonder whether we could combine these two strategies and take the advantages of both. The main contribution in this section is summarized as follows. A combination of K-best algorithm and the non-Gaussian approximation is proposed. In the K-best algorithm, K branches are kept at each decoding stage and the branches are pruned using the non-Gaussian approximation. Instead of using only the maximum of the K remaining metrics in K-best algorithms, we use the sum of all the K metrics to compute the LLR. The resulting algorithm is readily parallelized in hardware.

In this section, we assume squared-QAM is used at all transmit antennas, which is the case in many wireless communications standards. But the proposed algorithm can be readily extended to other general constellations. With squared-QAM, we can write (5) as a real system, i.e.,

$$\begin{bmatrix}\Re(\mathbf{y}_i)\\ \Im(\mathbf{y}_i)\end{bmatrix} = \begin{bmatrix}\Re(\mathbf{H}_i) & -\Im(\mathbf{H}_i)\\ \Im(\mathbf{H}_i) & \Re(\mathbf{H}_i)\end{bmatrix}\begin{bmatrix}\Re(\mathbf{x}_i)\\ \Im(\mathbf{x}_i)\end{bmatrix} + \begin{bmatrix}\Re(\mathbf{w}_i)\\ \Im(\mathbf{w}_i)\end{bmatrix}, \quad (7)$$

where $\Re(\cdot)$ and $\Im(\cdot)$ are the real and the imaginary parts of $(\cdot)$, respectively. With a slight abuse of notations, we still use (5) to represent the real system (7) in this section with the entries of $\mathbf{x}_i$ from PAM.

B. K-Best algorithm with Distribution Approximation

In this subsection, we consider extending the K-best algorithm in computing the LLR value. The LSD only considers the maximum term among all the $2^{\sum_{m=1}^{M} C_m - 1}$ terms, and the list is generated by using $\Pr(\mathbf{y}|x_1, ..., x_M)$ only without using the a priori information $\Pr(x_{m'})$, $m' = 1, ..., M$. Moreover, when the LSD comes to the i-th data stream, it only checks the symbols satisfying

$$\left(\tilde{y}_i - R_{i,i}x_i - \sum_{j=i+1}^{M} R_{i,j}\tilde{x}_i\right)^2 + \sum_{j=i+1}^{M}\left(\tilde{y}_j - \sum_{l=j}^{M} R_{j,l}\tilde{x}_l\right)^2 \le r^2, \quad (8)$$

where the QR decomposition of $\mathbf{H}$ is $\mathbf{H} = \mathbf{QR}$, $R_{i,j}$ is the (i,j)-th entry of $\mathbf{R}$, $\tilde{\mathbf{y}} = \mathbf{Q}^H\mathbf{y}$ and $\tilde{x}_j$ is the trial value of x_j. The key idea is to use the Gaussian approximation or non-Gaussian approximation as a metric to guide the search taking into account the effects of stream i on streams $1, ..., i-1$.

As LSD, we also want to find a list of K lattice points. Different from LSD, we try to find a list $\mathcal{L}_{i,\pm 1}$ containing K points for each $b_i = \pm 1$. The LLR value of the bit b_i in [4] is then approximated as

$$L(b_i|\mathbf{y}) \approx \log \frac{\sum_{\mathbf{x}\in\mathcal{L}_{i,+1}} \Pr(\mathbf{x}|\mathbf{y})}{\sum_{\mathbf{x}\in\mathcal{L}_{i,-1}} \Pr(\mathbf{x}|\mathbf{y})}. \quad (9)$$

The second difference from the LSD is that we use sum-log rather than max-log. The third difference lies in the way we generate the list. We want to find K lattice points $\mathbf{x} \in \mathcal{X}_{i,\pm 1}$ such that $\Pr(\mathbf{x}|\mathbf{y})$ is maximized rather than $\Pr(\mathbf{y}|\mathbf{x})$ is maximized, where the a priori information is exploited in the former case. There are two ways to generate the list using modified K-best algorithm: sum-algorithm and max-algorithm.

1) Sum-Algorithm: For the sum-algorithm, at the initial step, assuming that b_i belongs to data stream m, we first check each $\tilde{x} \in \mathcal{X}_{i,\pm 1}^{m}$ to find the K candidates such that $\Pr(x_m|\mathbf{y})$ is maximized and add m into a set $\mathcal{V}$. We can write $\Pr(\tilde{x}_m|\mathbf{y})$ as

$$\begin{aligned}\Pr(\tilde{x}_m|\mathbf{y}) &= \sum_{\mathbf{x}_{-m}} \Pr(\mathbf{x}_{-m}, \tilde{x}_m|\mathbf{y})\\ &\propto \sum_{\mathbf{x}_{-m}} \Pr(\mathbf{y}|\mathbf{x}_{-m}, \tilde{x}_m)\Pr(\mathbf{x}_{-m}). \quad (10)\end{aligned}$$

Direct computation of (10) requires $2^{\sum_{m=1}^{M} C_m} - 1$ summation, which may be computational prohibitive. We can replace the summation in (10) as an integral

$$\Pr(\tilde{x}_m|\mathbf{y}) \propto \int \sum_{\mathbf{x}_{-m}} \Pr(\mathbf{y}|\mathbf{x}_{-m}, \tilde{x}_m) f(\mathbf{x}_{-m}) d\mathbf{x}_{-m}, \quad (11)$$

where $f(\mathbf{x}_{-m})$ is the matched pdf of $\mathbf{x}_{-m}$, which could be either Gaussian or non-Gaussian. For example, with Gaussian approximation, we have

$$\begin{aligned}\Pr(\tilde{x}_m|\mathbf{y}) \propto\ & \exp\big(-(\mathbf{y} - \mathbf{H}_{-m}\boldsymbol{\mu}_{-m} - \mathbf{h}_m\tilde{x}_m)^H\\ & \mathbf{R}_m^{-1}(\mathbf{y} - \mathbf{H}_{-m}\boldsymbol{\mu}_{-m} - \mathbf{h}_m\tilde{x}_m)\big), \quad (12)\end{aligned}$$

where $\boldsymbol{\mu}_{-m}$ and $\mathbf{R}_m$ are defined as

$$\begin{aligned}\boldsymbol{\mu}_{-m} &= [\mu_1, \ldots, \mu_{m-1}, \mu_{m+1}, \ldots, \mu_M]^T\\ \mathbf{R}_m &= \mathbf{H}_{-m}\mathrm{diag}\{\nu_1^2, \ldots, \nu_{m-1}^2, \nu_{m+1}^2, \ldots, \nu_M^2\}\mathbf{H}_{-m}^H\\ &\quad +\sigma^2\mathbf{I}_N \quad (13)\end{aligned}$$

with mean μ and variance ν^2. The K $\tilde{x}_m$ with largest $\Pr(\tilde{x}_m|\mathbf{y})$ are added into a list $\mathcal{L}$, which is initialized to be ϕ.

The progress then goes to $x_1, x_2, \ldots, x_M$. Before it reaches x_j, $j \neq m$, we have $V = m, 1, \ldots, j-1$ and the list $\mathcal{L}$ contains K candidates, each of which has the form $\mathbf{x}_{\mathcal{V}} = [x_m, x_1, \ldots, x_{j-1}]^T$. For each $\mathbf{x}_{\mathcal{V}} \in \mathcal{L}$, we compute $\Pr(\mathbf{x}_{\mathcal{V}}, \tilde{x}_j|\mathbf{y})$ for each $\tilde{x}_j \in \mathcal{Q}_j$. Among the resulting $K|\mathcal{Q}_j|[\mathbf{x}_{\mathcal{V}}^T, \tilde{x}_j]^T$, we only choose K of them such that $\Pr(\mathbf{x}_{\mathcal{V}}, \tilde{x}_j|\mathbf{y})$ is maximized, update the list $\mathcal{L}$ with the K chosen vectors and add j into $\mathcal{V}$. We can approximate

$\Pr(\mathbf{x}_{\mathcal{V}}, \tilde{x}_j|\mathbf{y})$ in the same way as in (11). In case of Gaussian approximation, we have

$$\begin{aligned}&\Pr(\mathbf{x}_{\mathcal{V}}, \tilde{x}_j)\\ &\propto \exp\big(-(\mathbf{y}-\mathbf{H}_{-\{\mathcal{V},j\}}\boldsymbol{\mu}_{-\{\mathcal{V},j\}}-\mathbf{H}_{\mathcal{V}}\mathbf{x}_{\mathcal{V}}-\mathbf{h}_j\tilde{x}_j)^H\\ &\qquad \mathbf{R}^{-1}_{\{\mathcal{V},j\}}(\mathbf{y}-\mathbf{H}_{-\{\mathcal{V},j\}}\boldsymbol{\mu}_{-\{\mathcal{V},j\}}-\mathbf{H}_{\mathcal{V}}\mathbf{x}_{\mathcal{V}}-\mathbf{h}_j\tilde{x}_j)\big),\end{aligned} \tag{14}$$

where $\boldsymbol{\mu}_{-\mathcal{A}}$ constitutes the entries of $\boldsymbol{\mu}$ that are not in $\mathcal{A}$, $\mathbf{H}_{-\mathcal{A}}$ is consisted of the columns of $\mathbf{H}$ that are not in $\mathcal{A}$ and

$$\mathbf{R}_{\{\mathcal{V},j\}} = \mathbf{H}_{-\{\mathcal{V},j\}}\text{diag}\{\boldsymbol{\nu}^2_{\{\mathcal{V},j\}}\}\mathbf{H}^H_{-\{\mathcal{V},j\}} + \sigma^2\mathbf{I}_N. \tag{15}$$

The process ends when $j = M$.

2) Max-Algorithm: Different from the sum-algorithm where $\Pr(\mathbf{x}_{\mathcal{V}}, \tilde{x}_j|\mathbf{y})$ is maximized consecutively, in the max-algorithm, we maximize $\Pr(\tilde{\mathbf{x}}|\mathbf{y})$ directly. At the first step, for each $\tilde{x}_m \in \mathcal{X}^m_{i,\pm1}$, we find the corresponding $\tilde{\mathbf{x}}_{-m}$ such that

$$\begin{aligned}\tilde{\mathbf{x}}_{-m} &= \arg\max_{\mathbf{x}_{-m}\in\mathcal{X}^{-m}} \Pr(\tilde{x}_m, \mathbf{x}_{-m}|\mathbf{y})\\ &= \arg\max_{\mathbf{x}_{-m}\in\mathcal{X}^{-m}} \Pr(\mathbf{y}|\tilde{x}_m, \mathbf{x}_{-m})\Pr(\tilde{x}_m, \mathbf{x}_{-m}),\end{aligned} \tag{16}$$

where $\mathcal{X}^{-m}$ includes all possible lattice points. We put K $\tilde{x}_m$ into the list $\mathcal{L}$ such that $\Pr(\tilde{x}_m, \tilde{\mathbf{x}}_{-m}|\mathbf{y})$ is largest and add m into a set $\mathcal{V}$. As solving (16) has a high complexity, we therefore replace $\Pr(\tilde{x}_m, \mathbf{x}_{-m})$ with its continuous Gaussian or non-Gaussian approximations and relax the discrete set $\mathcal{X}^{-m}$ into a continuous set $\mathcal{C}^{-m}$.

When $\mathcal{C}^{-m}$ is bounded, the boundary on x_j is defined by the largest and smallest elements in $\mathcal{Q}_j$. For example, when $\mathcal{Q}_j = \{-3, -1, 1, 3\}$, we choose $-3 \le x_j \le 3$. When the non-Gaussian approximation is used, we need to solve

$$\begin{aligned}\hat{\mathbf{x}}_{-m} &= \arg\max_{\mathbf{x}_{-m}\in\mathcal{C}^{-m}} \|\mathbf{y}-\mathbf{H}_{-m}\mathbf{x}_{-m}-\mathbf{h}_m\tilde{x}_m\|^2\\ &\quad +2\sigma^2\mathbf{r}^T_{-m}\mathbf{x}_{-m} + \sigma^2\mathbf{x}^T_{-m}\mathbf{A}_{-m}\mathbf{x}_{-m}.\end{aligned} \tag{17}$$

As (17) is quadratic in $\mathbf{x}_{-m}$, when the objective function of (17) is convex, $\hat{\mathbf{x}}_{-m}$ can be found using convex optimization tools. If not, we find a local minimum around

$$\arg\max_{\mathbf{x}_{-m}\in\mathcal{C}^{-m}} \|\mathbf{y}-\mathbf{H}_{-m}\mathbf{x}_{-m}-\mathbf{h}_m\tilde{x}_m\|^2. \tag{18}$$

We can set $\tilde{\mathbf{x}}_{-m} = \hat{\mathbf{x}}_{-m}$ or map $\hat{\mathbf{x}}_{-m}$ to the closet lattice point in $\mathcal{X}^{-m}$. Comparing with (8), (17) uses the a priori information through $\mathbf{r}_{-m}$ and $\mathbf{A}_{-m}$ and it counts the effect of symbols $\tilde{x}_m$ on $\Pr(\tilde{x}_m, \tilde{\mathbf{x}}_{-m}|\mathbf{y})$.

The process then goes to $x_1, x_2, \ldots, x_M$. Before it reaches x_j, $j \ne m$, $V = m, 1, \ldots, j-1$ and the list $\mathcal{L}$ and each $\tilde{x}_j \in \mathcal{Q}_j$, we find the corresponding $\tilde{\mathbf{x}}_{-\{\mathcal{V},j\}}$ such that

$$\tilde{\mathbf{x}}_{-\{\mathcal{V},j\}} = \arg\max_{\mathbf{x}_{-\{\mathcal{V},j\}}\in\mathcal{X}^{-\{\mathcal{V},j\}}} \Pr(\tilde{\mathbf{x}}_{\mathcal{V}}, \tilde{x}_j, \mathbf{x}_{-\{\mathcal{V},j\}}|\mathbf{y}). \tag{19}$$

Among the resulting $K|\mathcal{Q}_j|[\mathbf{x}^T_{-\{\mathcal{V},j\}}, \tilde{x}_j]^T$, we only choose K of them such that $\Pr(\tilde{\mathbf{x}}_{\mathcal{V}}, \tilde{x}_j, \mathbf{x}_{-\{\mathcal{V},j\}}|\mathbf{y})$ is maximized, update the list $\mathcal{L}$ with the K chosen vectors and add j into $\mathcal{V}$. As in (17), we can approximate $\hat{\mathbf{x}}_{-m}$ by solving

$$\begin{aligned}\tilde{\mathbf{x}}_{-\{\mathcal{V},j\}} &= \arg\max_{\mathbf{x}_{-\{\mathcal{V},j\}}\in\mathcal{X}^{-\{\mathcal{V},j\}}} \|\mathbf{y}-\mathbf{H}_{-\mathcal{V},j}\mathbf{x}_{-\mathcal{V},j}\\ &\quad -\mathbf{H}_{\mathcal{V}}\tilde{\mathbf{x}}_{\mathcal{V}}-\mathbf{h}_{\mathcal{V}}\tilde{x}_{\mathcal{V}}\|^2 + 2\sigma^2\mathbf{r}^T_{-m}\mathbf{x}_{-m}\\ &\quad +\sigma^2\mathbf{x}^T_{-m}\mathbf{A}_{-m}\mathbf{x}_{-m}.\end{aligned} \tag{20}$$

where the notations are similar to those in (14) and (17).

The difference between the sum-algorithm and the max-algorithm lies in the fact that the effects of $\mathbf{x}_{-\{\mathcal{V},j\}}$ is removed from $\Pr(\tilde{\mathbf{x}}_{\mathcal{V}}, \tilde{x}_j, \mathbf{x}_{-\{\mathcal{V},j\}}|\mathbf{y})$ by summing over all possible $\mathbf{x}_{-\{\mathcal{V},j\}}$ in the former case while we take the max $\mathbf{x}_{-\{\mathcal{V},j\}}$ maximizing this probability in the latter case. The max-algorithm is similar to the decision feedback sphere decoding algorithm in [12] for uncoded MIMO systems. When $\mathcal{C}$ is unbounded and Gaussian approximation is used, it is easy to see that solving (16) is equivalent to solving

$$\begin{aligned}&\min_{\mathbf{x}_{-m}\in\mathcal{C}^{-m}} \|\mathbf{y}-\mathbf{H}_{-m}\mathbf{x}_{-m}-\mathbf{h}_m\tilde{x}_m\|^2\\ &\qquad +(\mathbf{x}_{-m}-\mu_{-m})^H\boldsymbol{\Lambda}_{-m}(\mathbf{x}_{-m}-\mu_{-m}).\end{aligned} \tag{21}$$

where $\boldsymbol{\Lambda}_{-m} = \text{diag}\{\nu^2_1, \ldots, \nu^2_{m-1}, \nu^2_{m+1}, \ldots, \nu^2_M\}$. In this case, the sum algorithm is equivalent to the max algorithm. The basic two algorithms can also be extended in various ways. We give a brief introduction of some important variations in the following.

3) Common List Algorithm: One disadvantage of the two basic list algorithms is that we need to find two lists (one for $+1$ and the other for -1) for each bit's LLR computation. When the total number of bits is large, this may incur a high computational complexity. To reduce the complexity, we propose using the same list $\mathcal{L}$ for all bits' LLR computation. The list is generated by choosing the K lattice points such that $\Pr(\mathbf{x}|\mathbf{y})$ is maximized. Both the sum-algorithm and the max-algorithm can serve this purpose. Different from the basic algorithms which start from x_m, we start from x_1 to $x_2, \ldots$ in the common list algorithm, where x_j is from $\mathcal{Q}_j, \forall j = 1, \ldots, M$. Finally, the LLR value of the bit b_i is then approximated as

$$L(b_i|\mathbf{y}) \approx \log\frac{\sum_{\mathbf{x}\in\mathcal{X}_{i,+1}\cap\mathcal{L}}\Pr(\mathbf{y}|\mathbf{x})\Pr(\mathbf{x})}{\sum_{\mathbf{x}\in\mathcal{X}_{i,-1}\cap\mathcal{L}}\Pr(\mathbf{y}|\mathbf{x})\Pr(\mathbf{x})}. \tag{22}$$

When $\mathbf{x} \in \mathcal{X}_{i,+1} \cap \mathcal{L} = \phi$, the LSD in [4] proposes using a predetermined saturated LLR value $\pm B$, e.g., $B = 8$. We propose using $\sum_{x_m\in\mathcal{X}^m_{i,+1}}\Pr(x_m)\Pr(\mathbf{y}|x_m)$ with Gaussian or non-Gaussian approximation for $\Pr(\mathbf{y}|x_m)$ or using

$$\max_{\mathbf{x}\mathcal{C}_{i,\pm1}} \Pr(\mathbf{x})\Pr(\mathbf{y}|\mathbf{x}) \tag{23}$$

where $\mathcal{C}_{i,\pm1}$ is the real relaxation of $\mathcal{X}_{i,\pm1}$.

4) Parallel Algorithm: In the basic algorithms, we generate the list by visiting $x_m, x_1, \ldots, x_M$ sequentially. We can also generate the list in parallel by generating a list $\mathcal{L}_i$ for each x_i, where $\mathcal{L}_i$ is generated by choosing the best K_i elements in $\mathcal{Q}_i$ to maximize $\Pr(x_i|\mathbf{y})$. Finally, the list is given by $\mathcal{L} = \mathcal{L}_1 \times \mathcal{L}_2 \times \ldots \times \mathcal{L}_M$, which is of size $K = \prod_{i=1}^M$. In this

case, different lists $\mathcal{L}_i$ can be generated in parallel, which is suitable for hardware implementation.

5) Bit-wise Algorithm: The basic algorithms proceed from symbol to symbol. Both algorithms can also run on bits. For example, when set partitioning mapping is used, we have seen that the 2^C-PAM can be written as a weighted sum of bits. Both algorithms can work on bits by replacing x in both algorithms with b.

We can also derive bit-wise algorithms for arbitrary mappings. We take the sum-algorithm as an example. To compute $L(b_i|\mathbf{y})$, we start with b_i and compute $\Pr(b_i = \pm 1|\mathbf{y}) = \sum_{\mathbf{x}\in\mathcal{X}_{i,+1}} \Pr(\mathbf{x}|\mathbf{y})$. In (11), we replace every x_j except x_m with a Gaussian or non-Gaussian continuous variable and $\Pr(b_i = \pm 1|\mathbf{y})$ is computed by summing over all possible x_m in $\mathcal{X}^m_{i,\pm 1}$. We can even approximate x_m as a continuous variable. For example, when x_m is assumed to be Gaussian, we can compute the matched mean and variance as

$$\mu_{m,i,\pm 1} = \sum_{x_m \in \mathcal{X}^m_{i,\pm 1}} \Pr(x_m) x_m \tag{24}$$

and

$$\nu^2_{m,i,\pm 1} = \sum_{x_m \in \mathcal{X}^m_{i,\pm 1}} \Pr(x_m)|x_m|^2 - |\mu_{m,i,\pm 1}|^2. \tag{25}$$

When the non-Gaussian distribution is used, we can get the distribution by fitting the distribution over the symbols in $\mathcal{X}^m_{i,\pm 1}$ only. We can obtain $\Pr(b_i = \pm 1|\mathbf{y})$ as (12). When the algorithm reaches bit b_j and its corresponding symbol is x'_m , where symbols $x'_{m+1}, \ldots, x'_{m-1}, x_{m+1}, \ldots, x_M$ have not been visited. Let $\mathbf{b}_j = [b_1, \ldots, b_j, b_i]^T$. For any $\tilde{\mathbf{b}}_j$ from the list $\mathcal{L}$, we can compute the matched mean and variance for x'_m as

$$\mu_{m',\mathbf{b}_j,\tilde{\mathbf{b}}_j} = \sum_{x_{m'} \in \mathcal{X}^{m'}_{\mathbf{b}_j,\tilde{\mathbf{b}}_j}} \Pr(x_{m'}) x_{m'} \tag{26}$$

and

$$\nu^2_{m',\mathbf{b}_j,\tilde{\mathbf{b}}_j} = \sum_{x_{m'} \in \mathcal{X}^{m'}_{\mathbf{b}_j,\tilde{\mathbf{b}}_j}} \Pr(x_{m'})|x_{m'}|^2 - |\mu_{m',\mathbf{b}_j,\tilde{\mathbf{b}}_j}|^2, \tag{27}$$

where $\mathcal{X}^{m'}_{\mathbf{b}_j,\tilde{\mathbf{b}}_j}$ is the set of constellation points for x'_m such that the corresponding bits in $\mathbf{b}_j$ is equal to $\tilde{\mathbf{b}_j}$. The rest of the algorithm is identical to that of the symbol based algorithm.

The advantage of the bit-wise algorithms is that some symbols can be pruned early when the first few bits of the corresponding symbols are not chosen in the list with K elements.

V. Simulation Results

In this section, a 2×2 MIMO-OFDM system with 1024 subcarriers and 960 subcarriers are considered for data transmission. It is assumed that perfect knowledge of channel state information is known at the receiver. Power P is assigned at each transmit antenna. The Extended Vehicular A model (EVA) [11] is assumed in this section with delay profile $[0\ 30\ 150\ 310\ 370\ 710\ 1090\ 1730\ 2510]$ns and power profile $[0\ -1.5\ -1.4\ -3.6\ -0.6\ -9.1\ -7\ -12\ -16.9]$ dB. And Turbo code is used with the transfer function of the PCCC $G(D) = \left[1, \frac{g_0(D)}{g_1(D)}\right]$ [12], where $g_0(D) = 1 + D^2 + D^3, g_1(D) = 1 + D + D^3$. 64 QAM and Gray mapping are set in the simulation.

A. BER Comparison of Different Algorithms

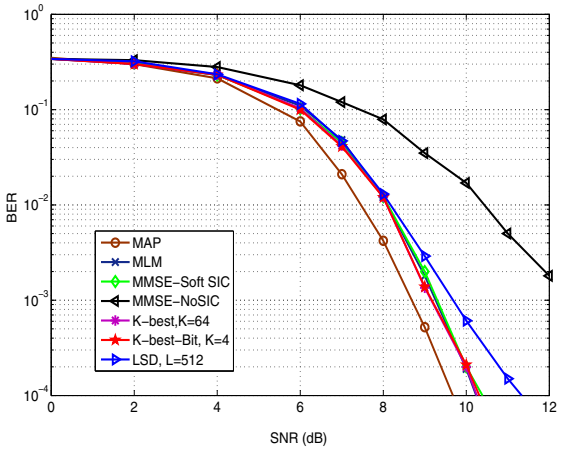

Fig. 2. BER comparison of different algorithms in a 2×2 MIMO-OFDM system over the EVA channel.

We first consider fixed scheduling, where both data streams transmit using transport block size (TBS) 1916. The bit error rates of different algorithms after simulating 20000 subframes are shown in Fig. 2. The channel varies independently from subframe to subframe. All algorithms except MMSE-No SIC use 6 iterations. It is clear that all the iterative algorithms benefit from the information exchange between the demodulation and decoder as compared with MMSE-No SIC. We can see that by using only the max term in MAP MLM (Max-log) incurs a 0.5 dB loss over MAP at BER$= 10^{-3}$. MMSE-Soft SIC only has a 0.1 dB loss over MLM at BER$= 10^{-3}$. But the former only needs to sum over 64 terms while the latter needs to compute $32 \times 64 = 2048$ terms, respectively. With the proposed K-best algorithms, K-Best, $K = 64$ has a 0.08 dB gain over MLM and K-Best-Bit, $K = 4$ has a 0.15 dB gain over MLM at BER$= 10^{-3}$. Note that K-Best-Bit, $K = 4$ only needs to sum over $K = 4$ terms in the LLR computation but with improved performance over MMSE-Soft SIC. LSD with $L = 512$ incurs a 1 dB loss over MAP at BER$= 10^{-3}$ but with a higher complexity than MMSE-Soft SIC and the proposed K-best algorithms. The proposed K-Best algorithm achieves a good performance and complexity trade-off.

B. Effects of the List Size K

In Fig. 3, the performance of the K-best algorithms are compared with different list size K. We can see that for both K-Best and K-Best-Bit doubling K gives a 0.5 dB gain at BER$= 10^{-3}$. In high SNR, the error floor is also reduced by increasing K. But doubling K also means that the complexity is roughly doubled. Moreover, more exponentials are needed to compute in the LLR computation, which is very expensive

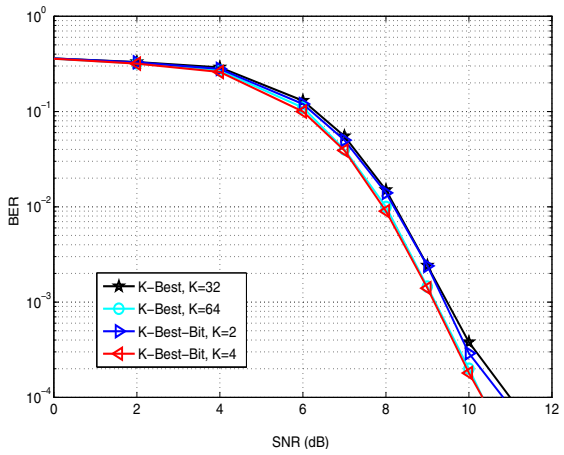

Fig. 3. BER comparison of K-best algorithms with list size K in a 2×2 MIMO-OFDM system over the EVA channel.

in hardware implementation. In practice, we may question whether the $0.2\,\mathrm{dB}$ gain is worth for the more complicated hardware implementation.

C. Throughput Comparison of Different Algorithms

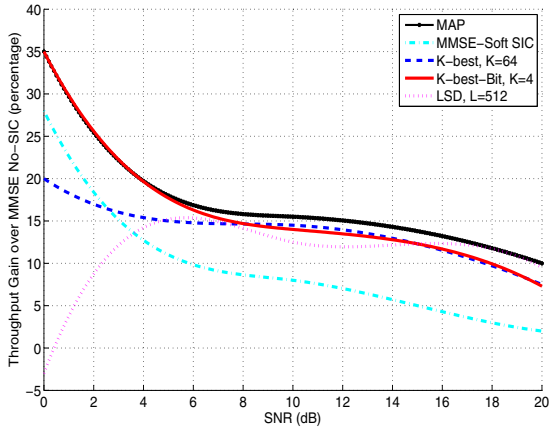

Fig. 4. Throughput gain comparison of different algorithms over MMSE No-SIC in a 2×2 MIMO-OFDM system over the EVA channel.

The BER performance does not translate to the throughput performance directly in practice. In Fig. 4, we compare the throughput gain of different algorithms over MMSE No-SIC. The throughput gain is obtained after averaging 50 channel realizations. For each channel realization, We can see that the throughput gain of all the algorithms except LSD decreases as SNR increases. This is because in high SNR the MIMO channel capacity with finite constellation is saturated to 6 bits/s/Hz as the largest constellation is 64QAM. In high SNR, all algorithms can approach this limit. The throughput of LSD is small in low SNR is because the list size is small in this case. K-Best-Bit, $K = 4$ achieves almost the same throughput as MAP in all SNRs. Both K-Best, $K = 64$ and K-Best-Bit, $K = 4$ achieve more than 5% gain over MMSE Soft-SIC when SNR is greater than $5\,\mathrm{dB}$ but the former two are potentially less complex than the latter one.

VI. Conclusion

Currently, to meet the data rate of IoV, there is a significant interest in the design of MIMO receiver. In this paper, we have developed several low-complexity modified K-best algorithms for MIMO-OFDM vehicular networks, which could provide a flexible performance and complexity trade-off. The proposed algorithms are different from the conventional K-best algorithm in the way how the K best candidates are generated and updated and how the LLR value is computed. Further, when $K = 1$, the proposed algorithms become an improved soft version of V-BLAST algorithm. When $K = +\infty$, they reduces to the optimal MAP detection. Simulation results on LTE systems demonstrate that the proposed low complexity algorithms can achieve a significant performance gain over existing ones with high order constellations.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China 61501461, 61471269, 61233001, 71232006, 61533019 and 91520301; the Early Career Development Award of SKLMCCS (Y3S9021F34); and Chinese Guangdong's S&T project (2014B010118001, 2014B090902001, 2014A010103004, 2014A050503004, 2015B010103001, 2016B090910001), Dongguan's Innovation Talents Project (Gang Xiong).

References

[1] X. Cheng, B. Yu, L. Yang, J. Zhang, G. Liu, Y. Wu, and L. Wan, "Communicating in the real world: 3D MIMO," *IEEE Wireless Commun.*, vol. 21, no. 4, pp.136–144, Aug. 2014.

[2] X. Cheng, C.-X. Wang, H. Wang, X. Gao, X.-H. You, D. Yuan, B. Ai, Q. Huo, L.-Y. Song, and B.-L. Jiao, "Cooperative MIMO channel modeling and multi-link spatial correlation properties," *IEEE J. Sel. Areas Commun*, vol. 30, no. 2, pp. 388–396, Feb. 2012.

[3] E. Viterbo and J. Boutros, "A universal lattice code decoder for fading channels," *IEEE Trans. Inf. Theory*, vol. 45, no. 5, pp. 1639–1642, Jul. 1999.

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

[5] Z. Guo and P. Nilsson, "Algorithm and implementation of the K-best sphere decoding for MIMO detection," *IEEE J. Sel. Areas Commun*, vol. 24, no. 3, pp. 491–503, Mar. 2006.

[6] X. Wang and H. Poor, "Iterative (turbo) soft interference cancellation and decoding for coded CDMA," *IEEE Trans. Commun.*, vol. 47, no. 7, pp. 1046–1061, Jul. 1999.

[7] M. Tuchler, A. Singer, and R. Koetter, "Minimum mean squared error equalization using a priori information," *IEEE Trans. Signal Process.*, vol. 50, no. 3, pp. 673–683, Mar. 2002.

[8] D. Pham, K. Pattipati, P. Willett, and J. Luo, "A generalized probabilistic data association detector for multiple antenna systems," in *IEEE International Conference on Communications*, vol. 6, June 2004, pp. 3519–3522, vol.6.

[9] S. Liu and Z. Tian, "Near-optimum soft decision equalization for frequency selective MIMO channels," *IEEE Trans. Signal Process.*, vol. 52, no. 3, pp. 721–733, Mar. 2004.

[10] C. Mecklenbrauker, A. Molisch, J. Karedal, F. Tufvesson, A. Paier, L. Bernado, T. Zemen, O. Klemp, and N. Czink, "Vehicular channel characterization and its implications for wireless system design and performance," *Proc. of the IEEE*, vol. 99, no. 7, pp. 1189–1212, 2011.

[11] "Evolved Universal Terrestrial Radio Access (E-UTRA); user equipment (UE) radio transmission and reception," The 3rd Generation Partnership Project, TS 36.101, Mar. 2014.

[12] "Evolved Universal Terrestrial Radio Access (E-UTRA); multiplexing and channel coding," The 3rd Generation Partnership Project, TS 36.212, Dec. 2013.