IEICE TRANS. COMMUN., VOL.E94–B, NO.1 JANUARY 2011

PAPER

New Factorization Algorithms for Channel-Factorization Aided MMSE Receiver in MIMO Systems

Chih-Cheng KUO[†a)], ***Student Member***, **Wern-Ho SHEEN**[††], ***and*** **Chang-Lung HSIAO**[†], ***Nonmembers***

SUMMARY Channel-factorization aided detector (CFAD) is one of the important low-complexity detectors used in multiple input, multiple output (MIMO) receivers. Through channel factorization, this method transforms the original MIMO system into an equivalent system with a better-conditioned channel where detection is performed with a low-complexity detector; the estimate is then transferred back to the original system to obtain the final decision. Traditionally, the channel factorization is done with the lattice reduction algorithms such as the Lenstra-Lenstra-Lovász (LLL) and Seysen's algorithms with no consideration of the low-complexity detector used. In this paper, we propose a different approach: the channel factorization is designed specifically for the minimum mean-square-error (MMSE) detector that is a popular low-complexity detector in CFADs. Two new types of factorization algorithms are proposed. Type-I is LLL based, where the well-known DLLL-extended algorithm, the LLL algorithm working on the dual matrix of the extended channel matrix, is a member of this type but with a higher complexity. DLLL-extended is the best-performed factorization algorithm found in the literature, Type-II is greedy-search based where its members are differentiated with different algorithm's parameters. Type-II algorithms can provide around 0.5–1.0 dB gain over Type-I algorithms and have a fixed computational complexity which is advantageous in hardware implementation.

key words: *MIMO detection, MMSE channel factorization, lattice-reduction aided detection*

1. Introduction

Multiple input, multiple output (MIMO) is a model for a range of communication problems including the multiple transmit and receive antenna systems [1], code-division multiple-access systems [2], inter-symbol interference channels [3], etc. The canonical form of the model is given by*.

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{w} \tag{1}$$

where $\boldsymbol{y} = [y_1 \cdots y_n]^T \in C^n$, $\boldsymbol{x} = [x_1 \cdots x_m]^T \in C^m$, $\boldsymbol{w} = [w_1 \cdots w_n]^T \in C^n$ are the received signal vector, transmitted signal vector, and noise vector, respectively, $\boldsymbol{H}$ is the channel matrix, and C is the set of complex numbers. For QAM constellations, it is easy to see that, after proper shifting and scaling, $\boldsymbol{x} = [x_1 \cdots x_m]^T \in \Omega^m$, where $\Omega \subset Z$, and Z is the set of complex integers.

The problem of detecting the MIMO signal in (1) is to find estimate $\hat{\boldsymbol{x}} \in \Omega^m$ that minimizes the error probability $P\{\hat{\boldsymbol{x}} \neq \boldsymbol{x}\}$, given the received signal vector $\boldsymbol{y}$ and the channel matrix $\boldsymbol{H}$. Maximum likelihood (ML) detector is the optimum detector that minimizes the error probability. Without a special structure on Ω^m, however, the complexity of the ML detector grows exponentially with m and/or $|\Omega|$, the cardinality of Ω, because all the signal vectors need to be searched exhaustively for optimum detection. For the case of $\Omega \subset Z$, on the other hand, the complexity of the ML detector can be reduced by using the sphere decoding [4], [5], where the searching is limited to within a sphere around $\boldsymbol{y}$, although its complexity may still be too demanding for some applications. In view of this, different reduced-complexity sub-optimal detectors have been proposed for practical systems, including the linear and nonlinear detectors [6]–[9]. For an ill-conditioned channel, unfortunately, the performance of these detectors is significantly inferior to that of the ML detector.

Recently, channel-factorization aided detectors (CFAD) have been proposed to narrow the performance gap between the ML and traditional reduced-complexity detectors [10]–[19], where channel factorization is done with lattice-reduction algorithms such as the Lenstra-Lenstra-Lovász (LLL) [20], [21] and Seysen's algorithms [22], [23]. Thus, the method is also called the lattice-reduction aided detector, LRAD. (In this paper, we prefer to use the term CFAD because the channel factorization is not necessarily done with a lattice-reduction algorithm.) In particular, in [10]–[13], the LLL-based LRAD was proposed to improve performance over the conventional zero-forcing (ZF) MIMO detector. The authors of [14]–[16] proposed the MMSE-based LRAD to further improve its ZF counterpart. Later on, the LLL algorithm was proposed to operate on the dual lattice rather than the original lattice to reduce effective noise power [17], [18]. In addition, the Seysen's algorithm [23] which simultaneously reduces lattice basis and its dual was proposed to improve the performance of the LLL algorithm. Lastly the authors of [19] conducted a comprehensive performance comparison on the reduction algorithms mentioned above.

Traditionally, the channel factorization algorithms in a lattice-reduction aided detector (LRAD) are designed somewhat intuitively; neither a specific detector nor a cost function is involved in search of good factorization algorithms.

Manuscript received March 10, 2010.
Manuscript revised July 27, 2010.

[†]The authors are with the Department of Electrical Engineering, National Chiao Tung University, Hsinchu, 300, Taiwan, ROC.
[††]The author is with the Department of Information and Communication Engineering, Chaoyang University of Technology, Wufong, Taichung, 41349, Taiwan, ROC.
a) E-mail: cckuo.cm90g@nctu.edu.tw

DOI: 10.1587/transcom.E94.B.222

*Throughout this paper, bold capital letters denote matrices and small letters denote vectors. $(\cdot)^H$ and $(\cdot)^T$ represent the operations of conjugate transpose and transpose of a matrix or vector, respectively, and $(\cdot)^*$ denotes the complex conjugate of a complex value.

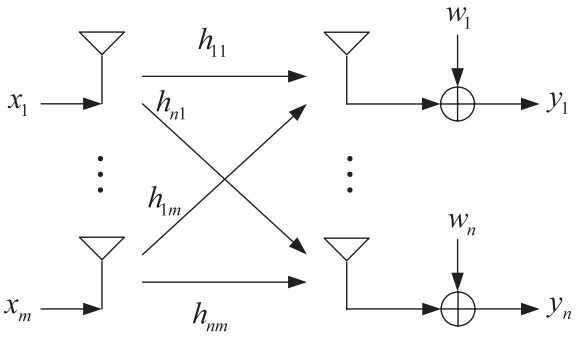

Fig. 1 The considered MIMO channel with m transmit and n receive antennas.

The LLL and Seysen's algorithms were employed in previous works to obtain a relatively short (orthogonal) basis with no consideration of the low-complexity detector used. In this work, a new approach is proposed: the factorization algorithm is designed specifically for the MMSE detector with the aim to minimize the cost function of sum mean-squared-error (MSE). Two new types of factorization algorithms are proposed. Type-I is LLL-based, where the best-performed factorization algorithm found in the literature, i.e., the DLLL-extended algorithm, is a member of this type but with a higher complexity. In this part, this work provides a theoretical foundation for the DLLL-extended algorithm. Type-II is greedy-search based, where its members are differentiated with different algorithm's parameters. Type-II algorithms can provide around 0.5–1.0 dB gain over Type-I algorithms and have a fixed computational complexity which is advantageous in hardware implementation.

The remainder of this paper is organized as follows. Section 2 gives the system model and a review on the MMSE detector. Section 3 discusses CFAD which can be viewed as an extension of LRAD. The proposed algorithms are presented in Sect. 4, and simulation results are in Sect. 5. Lastly, conclusions are given in Sect. 6.

2. System Model and MMSE Detector

Figure 1 is the considered flat-faded MIMO channel with m transmit and $n \geq m$ receive antennas, where $h_{i,j}$ denotes the complex-valued gain from transmit antenna j to receive antenna i, $1 \leq j \leq m$ and $1 \leq i \leq n$. Using the notations in (1), the channel matrix is $\boldsymbol{H} = [h_{i,j}]$. The popular correlated model $\boldsymbol{H} = \boldsymbol{J}_R^{1/2} \boldsymbol{F} \boldsymbol{J}_T^{1/2}$ in [15] is adopted in this work, where $\boldsymbol{F}$ consists of zero-mean, uncorrelated complex Gaussian coefficients of unit variance, and $\boldsymbol{J}_T$ and $\boldsymbol{J}_R$ are the spatial correlation matrices at transmitter and receiver, respectively. Furthermore, as in [15], we adopt the commonly used correlation matrices

$$\boldsymbol{J}_T = \begin{bmatrix} 1 & \rho & \rho^4 & \cdots & \rho^{(m-1)^2} \\ \rho & 1 & \rho & \ddots & \vdots \\ \rho^4 & \rho & 1 & \ddots & \rho^4 \\ \vdots & \ddots & \ddots & \ddots & \rho \\ \rho^{(m-1)^2} & \cdots & \rho^4 & \rho & 1 \end{bmatrix}, \tag{2}$$

and

$$\boldsymbol{J}_R = \begin{bmatrix} 1 & \rho & \rho^4 & \cdots & \rho^{(n-1)^2} \\ \rho & 1 & \rho & \ddots & \vdots \\ \rho^4 & \rho & 1 & \ddots & \rho^4 \\ \vdots & \ddots & \ddots & \ddots & \rho \\ \rho^{(n-1)^2} & \cdots & \rho^4 & \rho & 1 \end{bmatrix}, \tag{3}$$

with $0 \leq \rho \leq 1$. Note that $\rho = 0.0$ gives the uncorrelated channel and $\rho = 1.0$ gives the fully correlated one.

The signal vector $\boldsymbol{x} \in \Omega^m$ has independent and identical distributed (i.i.d.) entries with the power constraint $E[\|\boldsymbol{x}\|^2] = m\sigma_x^2$, where $\|\cdot\|^2$ denotes the squared Euclidean norm, and $\boldsymbol{w} = [w_1 \cdots w_n]^T$ is a circularly symmetric complex Gaussian vector with the correlation matrix $E[\boldsymbol{w}\boldsymbol{w}^H] = \sigma_w^2 \boldsymbol{I}_n$. $\boldsymbol{x}$ and $\boldsymbol{w}$ are independent of each other, and $\boldsymbol{I}_n$ denotes the $n \times n$ identity matrix.

Basically, a linear detector is to find an $\boldsymbol{x} \in \Omega^m$ that is closest to the filtered vector $\boldsymbol{B}^H \boldsymbol{y}$, i.e.,

$$\widehat{\boldsymbol{x}} = \arg \min_{\boldsymbol{x} \in \Omega^m} \left\| \boldsymbol{B}^H \boldsymbol{y} - \boldsymbol{x} \right\|^2 = Q\left[\boldsymbol{B}^H \boldsymbol{y} \right], \tag{4}$$

where $\boldsymbol{B}^H$ is the receive filter, and $Q[\cdot]$ is the operation of rounding its argument to the nearest $\hat{\boldsymbol{x}} \in \Omega^m$. For the MMSE linear detector, $\boldsymbol{B} = [(\boldsymbol{H}^H \boldsymbol{H} + \sigma_w^2 \boldsymbol{I}_m / \sigma_x^2)^{-1} \boldsymbol{H}^H]^H$ [6]–[9]. Without considering the effect of noise, i.e., $\sigma_w^2/\sigma_x^2 = 0$, the detector degenerates to ZF linear detector, where inter-symbol interference in $\boldsymbol{y}$ is cancelled completely. It is well known that linear detectors suffer from severe noise enhancement in an ill-conditioned channel and have diversity order of $n - m + 1$ which is less than the full diversity order n [6]–[9].

3. Channel-Factorization Aided Detection

In the literature, LRAD has been proposed to improve the performance of the traditional reduced-complexity detectors while retains a low complexity [10]–[19]. It was shown in [17], [18] that LRAD achieves full diversity order. Recall that in this paper LRAD will be viewed as a special case of a more general class of detectors, CFAD, where channel factorization can be done with any algorithms including the LLL and Seysen's lattice-reduction algorithms.

Let $H \doteq [\boldsymbol{h}_1 \boldsymbol{h}_2 \cdots \boldsymbol{h}_m]$, where $\{\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_m\}$ is a set of linearly independent vectors in C^n. The set of points $\Lambda_H = \{v | v = \sum_{i=1}^m \boldsymbol{h}_i x_i, x_i \in Z\}$ is called a lattice of dimension m, generated by the basis $\{\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_m\}$, and $\boldsymbol{H} \doteq [\boldsymbol{h}_1 \boldsymbol{h}_2 \cdots \boldsymbol{h}_m]$ is the generator matrix. It is clear that

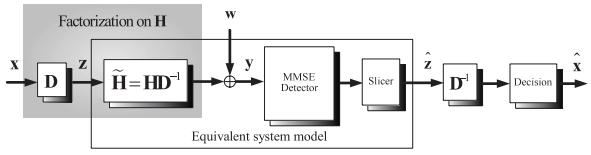

Fig. 2 A simplified diagram for the CFAD-MMSE detector.

different bases can be used to generate the same lattice. In particular, $\boldsymbol{H}$ and $\widetilde{\boldsymbol{H}}$ generate the same lattice Λ_H if and only if $\boldsymbol{H} = \widetilde{\boldsymbol{H}}\boldsymbol{D}$, where $\boldsymbol{D}$ is a unimodular matrix [21]. A complex integer matrix $\boldsymbol{D}$ is called unimodular if $|\det(\boldsymbol{D})| = 1$. Clearly, $\boldsymbol{D}^{-1}$ is also a unimodular matrix. In (1), the noiseless received signal vector is a lattice point in Λ_H. Therefore, the detection problem is to find a lattice point in Λ_H that is as close as possible to the received signal vector $\boldsymbol{y}$.

A block diagram of the channel-factorization aided MMSE detector (CFAD-MMSE) is shown in Fig. 2; the original system in (1) is transformed into an equivalent one by factorizing the channel matrix into $\boldsymbol{H} = \widetilde{\boldsymbol{H}}\boldsymbol{D}$, where $\boldsymbol{D}$ is a unimodular matrix. That is,

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{w} = \widetilde{\boldsymbol{H}}\boldsymbol{D}\boldsymbol{x} + \boldsymbol{w} = \widetilde{\boldsymbol{H}}\boldsymbol{z} + \boldsymbol{w}, \tag{5}$$

where $\boldsymbol{z} = \boldsymbol{D}\boldsymbol{x}$ is a symbol vector in the transform domain. If the channel factorization is done with a lattice-reduction algorithm, then it is the well-known LRAD. The key idea of CFAD is firstly to find a factorization such that $\boldsymbol{H}$ is better-conditioned than $\boldsymbol{H}$, then a reduced-complexity detection, i.e., "Slicer," is performed in the $\boldsymbol{z}$-domain to obtain an initial estimate $\boldsymbol{z}$. The "Slicer" in Fig. 2 performs the operation of element-wise rounding after a proper shifting and scaling as that given in [16]. Eventually, the estimate is transformed back to the original $\boldsymbol{x}$-domain to obtain the final estimate

$$\widehat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}\in\Omega^m} \left\|\boldsymbol{D}^{-1}\widehat{\boldsymbol{z}} - \boldsymbol{x}\right\|^2. \tag{6}$$

The LLL algorithm is well-known for searching a good factorization $\boldsymbol{H} = \widetilde{\boldsymbol{H}}\boldsymbol{D}$. Through size reduction and reordering of the sequence of basis vectors, a relatively short (orthogonal) basis can be found with a polynomial time complexity [20], [21]. The complex version of the LLL algorithm in [24] is summarized in Table 1. The LLL algorithm can be applied to the primal lattice [10]–[16] generated by the generator matrix $\boldsymbol{H}$, the dual lattice [17], [18] generated by the generator matrix $\boldsymbol{H}^{\#} \doteq [(\boldsymbol{H}^H\boldsymbol{H})^{-1}\boldsymbol{H}^H]^H$, and the extended lattice generated by $\underline{\boldsymbol{H}} \doteq [\boldsymbol{H}^T, \sigma_w\boldsymbol{I}_m/\sigma_x]^T$ [14]. Seysen's algorithm (SA) is another popular lattice-reduction algorithm for the channel factorization [22], [23]. Since SA reduces the primal and dual lattices simultaneously, it has a similar performance as that of LLL applied to the dual lattice, as to be shown in Sect. 5. Very recently, it was reported in [19] that the LLL algorithm applying on the dual lattice $\underline{\boldsymbol{H}}^{\#}$ and the SA algorithm on $\underline{\boldsymbol{H}}$ give the best performance if the MMSE detector is used as the low-complexity in Fig. 2 to obtain the initial estimate $\boldsymbol{z}$. In Fig. 2, the slicer is used to lower the detection complexity [16], which is perfectly fine for an infinite constellation because $\boldsymbol{z} = \boldsymbol{D}\boldsymbol{x} \in Z^m$ for $\boldsymbol{x} \in Z^m$. For a finite constellation, however, there is boundary error effect; that is, $\boldsymbol{D}^{-1}\widehat{\boldsymbol{z}}$ may not belong to Ω^m. This explains why (6) is needed for the final detection.

Table 1 Complex version of the LLL algorithm.

Input: lattice matrix $\mathbf{H} = [\mathbf{h}_1\mathbf{h}_2\cdots\mathbf{h}_m] \overset{\text{GSO}}{=} \mathbf{Q}\mathbf{U}^T = [\mathbf{q}_1\mathbf{q}_2\cdots\mathbf{q}_m]\begin{bmatrix} 1 & & & & \\ \mu_{2,1} & 1 & & 0 & \\ \mu_{3,1} & \mu_{3,2} & 1 & & \\ \vdots & \vdots & \vdots & \ddots & \\ \mu_{m,1} & \mu_{m,2} & \mu_{m,3} & \cdots & 1 \end{bmatrix}^T$.

Output: $\widetilde{\mathbf{H}}$ and unimodular matrix $\mathbf{T}$ such that $\mathbf{H} = \widetilde{\mathbf{H}}\mathbf{T}$.

1. $i = 2$, and $\mathbf{T} = \mathbf{I}_m = \begin{bmatrix} \mathbf{t}_1 \\ \vdots \\ \mathbf{t}_m \end{bmatrix}$
2. while $(i \le m)$ do
3. $\mathbf{h}_i = \mathbf{h}_i - [\![\mu_{i,i-1}]\!]\mathbf{h}_{i-1}$, $\mathbf{t}_{i-1} = \mathbf{t}_{i-1} + [\![\mu_{i,i-1}]\!]\mathbf{t}_i$
4. update GSO of $\mathbf{H}$ based on effective procedures in [20]
5. if $\left(\|\mathbf{q}_i + \mu_{i,i-1}\mathbf{q}_{i-1}\|^2 < \delta\|\mathbf{q}_{i-1}\|^2\right)$, then
6. swap $\mathbf{h}_{i-1}$ and $\mathbf{h}_i$, swap $\mathbf{t}_{i-1}$ and $\mathbf{t}_i$
7. update GSO of $\mathbf{H}$ based on effective procedures in [20]
8. $i = \max\{2, i-1\}$
9. else
10. for $j = i-2$ to 1 do
11. $\mathbf{h}_i = \mathbf{h}_i - [\![\mu_{i,j}]\!]\mathbf{h}_j$, $\mathbf{t}_j = \mathbf{t}_j + [\![\mu_{i,j}]\!]\mathbf{t}_i$
12. update GSO of $\mathbf{H}$ based on effective procedures in [20]
13. end for
14. $i = i + 1$
15. end if
16. end while
17. $\widetilde{\mathbf{H}} = \mathbf{H}$

4. The Proposed Factorization Algorithms for CFAD-MMSE Detector

Traditionally, the LLL and Seysen's algorithms are used in CFAD for the channel factorization $\boldsymbol{H} = \widetilde{\boldsymbol{H}}\boldsymbol{D}$ with no consideration of which type of low-complexity detector is used in the $\boldsymbol{z}$-domain detection. In this section, we propose a different approach: channel factorization is designed specifically for the CFAD-MMSE detector, and thus improve the detector performance of the detector either in bit error rate or detector complexity over other channel factorization methods in the literature.

4.1 MMSE Criterion

Let $\boldsymbol{G}_{MMSE}$ be the MMSE filter for the initial detection in the $\boldsymbol{z}$-domain, given the factorization $\boldsymbol{H} = \widetilde{\boldsymbol{H}}\boldsymbol{D}$. It can be shown that

$$\begin{aligned} \boldsymbol{G}_{MMSE} &= \arg\min_{G} E\left[\|\boldsymbol{G}\boldsymbol{y} - \boldsymbol{z}\|^2\right] \\ &= \boldsymbol{D}\left(\boldsymbol{H}^H\boldsymbol{H} + \frac{\sigma_w^2}{\sigma_x^2}\boldsymbol{I}_m\right)^{-1}\boldsymbol{H}^H \end{aligned} \tag{7}$$

where $\boldsymbol{G}$ is the receiver matrix. From Appendix A, the covariance matrix of the error vector $\boldsymbol{G}_{MMSE}\boldsymbol{y} - \boldsymbol{z}$ is

$$\begin{aligned} \boldsymbol{\Phi}_{MMSE} &\doteq E\left[(\boldsymbol{G}_{MMSE}\boldsymbol{y} - \boldsymbol{z})(\boldsymbol{G}_{MMSE}\boldsymbol{y} - \boldsymbol{z})^H\right] \\ &= \sigma_w^2\boldsymbol{D}\left(\boldsymbol{H}^H\boldsymbol{H} + \frac{\sigma_w^2}{\sigma_x^2}\boldsymbol{I}_m\right)^{-1}\boldsymbol{D}^H = \boldsymbol{D}\boldsymbol{A}\boldsymbol{D}^H, \end{aligned} \tag{8}$$

where $\boldsymbol{A} = \sigma_w^2(\boldsymbol{H}^H\boldsymbol{H} + \sigma_w^2\boldsymbol{I}_m/\sigma_x^2)^{-1}$ is a positive definite matrix, and the sum MSE (mean square error) is

$$tr\left(\boldsymbol{\Phi}_{MMSE}\right) = tr\left(\boldsymbol{DAD}^H\right). \tag{9}$$

where $tr(\cdot)$ denotes the sum of the diagonal elements of a square matrix. Using (8) and (9), our goal is to find the factorization $\boldsymbol{H} = \widetilde{\boldsymbol{H}}\boldsymbol{D}_{opt}$ by solving the following optimization problem

$$\begin{aligned}\boldsymbol{D}_{opt} &= \arg\min_{D}\left[tr\left(\boldsymbol{DAD}^H\right)\right],\\ &\quad \text{s.t. } \boldsymbol{D} \text{ is a unimodular matix}\end{aligned} \tag{10}$$

Note that there are infinite numbers of unimodular matrices in (10), and thus finding the optimal solution by exhaustive search is not possible. Here, two types algorithms are proposed to obtain approximate solutions efficiently: one is LLL-based, and the other is greedy-search based with column-wise optimization.

4.2 Type-I Algorithms (LLL Based)

By applying singular-value-decomposition (SVD), the channel matrix $\boldsymbol{H}$ can be expressed as

$$\boldsymbol{H} = \boldsymbol{U}\begin{bmatrix}\Delta\\ \boldsymbol{0}\end{bmatrix}\boldsymbol{V}^H, \tag{11}$$

where $\boldsymbol{U}$ and $\boldsymbol{V}$ are unitary matrices with dimension of $n\times n$ and $m\times m$, respectively, and $\Delta = \mathrm{diag}(\Delta_1, \Delta_2, \cdots, \Delta_m)$ is an $m\times m$ diagonal matrix with the singular values $\Delta_k > 0$, $k = 1, \ldots, m$. Using (11), the matrix $\boldsymbol{A}$ becomes

$$\boldsymbol{A} = \boldsymbol{\Upsilon}^H\boldsymbol{\Upsilon}, \tag{12}$$

where $\boldsymbol{\Upsilon} = \boldsymbol{\Gamma}^{1/2}\boldsymbol{V}^H$ is an $m\times m$ nonsingular matrix, and

$$\Gamma = \mathrm{diag}\left(\frac{\sigma_w^2\sigma_x^2}{\Delta_1^2\sigma_x^2+\sigma_w^2}, \frac{\sigma_w^2\sigma_x^2}{\Delta_2^2\sigma_x^2+\sigma_w^2}, \cdots, \frac{\sigma_w^2\sigma_x^2}{\Delta_m^2\sigma_x^2+\sigma_w^2}\right). \tag{13}$$

More generally, we have the following lemma.

Lemma 1: With $\boldsymbol{\Upsilon}$ in (12), $\boldsymbol{A} = \boldsymbol{C}^H\boldsymbol{C}$ if and only if $\boldsymbol{C} = \boldsymbol{P}\boldsymbol{\Upsilon}$, where $\boldsymbol{P}^H\boldsymbol{P} = \boldsymbol{I}_m$ is an $l\times m$ matrix with $l \geq m$.

Proof: For the if part, $\boldsymbol{C}^H\boldsymbol{C} = (\boldsymbol{P}\boldsymbol{\Upsilon})^H\boldsymbol{P}\boldsymbol{\Upsilon} = \boldsymbol{\Upsilon}^H\boldsymbol{\Upsilon} = \boldsymbol{A}$. For the only if part, using (12), one gets $\boldsymbol{C}^H\boldsymbol{C} = \boldsymbol{\Upsilon}^H\boldsymbol{\Upsilon}$. Since $\boldsymbol{\Upsilon}$ is nonsingular, $\boldsymbol{\Upsilon}^{-H}\boldsymbol{C}^H\boldsymbol{C}\boldsymbol{\Upsilon}^{-1} = \boldsymbol{I}_m$. Identify $\boldsymbol{P} = \boldsymbol{C}\boldsymbol{\Upsilon}^{-1}$, the proof is done. In addition, $\boldsymbol{C} = \boldsymbol{P}\boldsymbol{\Upsilon}$ is full column-ranked due to that $\boldsymbol{\Upsilon}$ is nonsingular and $\boldsymbol{P}$ is full column-ranked.

Define $\boldsymbol{D}^H = [\boldsymbol{d}_1\boldsymbol{d}_2\cdots\boldsymbol{d}_m]$, that is $\boldsymbol{d}_k$ is k-th column vector of matrix $\boldsymbol{D}^H$. Using $\boldsymbol{A} = \boldsymbol{C}^H\boldsymbol{C}$, the sum MSE becomes

$$tr\left(\boldsymbol{\Phi}_{MMSE}\right) = \sum_{k=1}^{m}\boldsymbol{d}_k^H\boldsymbol{C}^H\boldsymbol{C}\boldsymbol{d}_k = \sum_{k=1}^{m}\|\boldsymbol{C}\boldsymbol{d}_k\|^2, \tag{14}$$

where $\boldsymbol{C}\boldsymbol{d}_k$, $k = 1, \cdots, m$ are lattice points of the lattice $\Lambda_C = \{\boldsymbol{v}|\,\boldsymbol{v} = \sum_{k=1}^{m}\boldsymbol{c}_k x_k, x_k \in Z\}$, generated by the generator matrix $\boldsymbol{C} \doteq [\boldsymbol{c}_1\cdots\boldsymbol{c}_m]$. Furthermore, $\boldsymbol{C}\boldsymbol{d}_k$, $k = 1, \cdots, m$ are linearly independent since $\boldsymbol{D}^H$ is unimodular and $\boldsymbol{C}$ is full column-ranked, and therefore $\{\boldsymbol{C}\boldsymbol{d}_k\}_{k=1}^m$ is a basis of the lattice Λ_C. Consequently, the optimization problem in (10) becomes to find the basis of Λ_C that has the smallest sum squared norm $\sum_{k=1}^m\|\boldsymbol{C}\boldsymbol{d}_k\|^2$, which is a well-known lattice reduction problem. Let $\{\boldsymbol{c}_{opt,k}\}_{k=1}^m$ be a basis of Λ_C that has the smallest sum squared norm, and $\boldsymbol{C}_{opt} = [\boldsymbol{c}_{opt,1}, \cdots, \boldsymbol{c}_{opt,m}]$. Since $\boldsymbol{C}_{opt}$ and $\boldsymbol{C}$ generate the same lattice, $\boldsymbol{C} = \boldsymbol{C}_{opt}\boldsymbol{T}_{opt}$ for a unimodular matrix $\boldsymbol{T}_{opt}$. In addition, $\boldsymbol{C}_{opt} = \boldsymbol{C}\boldsymbol{D}_{opt}^H$, thus $\boldsymbol{D}_{opt} = (\boldsymbol{T}_{opt}^{-1})^H$. Given $\boldsymbol{D}_{opt}$, the desirable factorization is $\boldsymbol{H} = \widetilde{\boldsymbol{H}}_{opt}\boldsymbol{D}_{opt}$.

The LLL lattice-reduction algorithm will be adopted here as a practical way to search for $\boldsymbol{D}_{opt}$. In this case, if $\widetilde{\boldsymbol{C}}$ denotes the reduced basis by the LLL algorithm, that is $\boldsymbol{C} \overset{LLL}{=} \widetilde{\boldsymbol{C}}\boldsymbol{T}$, where $\boldsymbol{T}$ is a unimodular matrix, then the approximate solution is obtained as $\widehat{\boldsymbol{D}} = (\boldsymbol{T}^{-1})^H$ and $\boldsymbol{H} = \widetilde{H}\widehat{\boldsymbol{D}}$. From Appendix B, it is shown that $\boldsymbol{D}$ is the same for any decomposition $\boldsymbol{A} = \boldsymbol{C}^H\boldsymbol{C}$, and thus the detection performance is independent of what particular decomposition result $\boldsymbol{C}$ is used. The complete algorithm is summarized in Table 2.

Table 2 The Proposed Type-I Algorithm.

Input: $\mathbf{H}$, σ_x^2, σ_w^2
Output: $\mathbf{D}$, a unimodular matrix.

1: Compute $\mathbf{A} = \sigma_w^2\left(\mathbf{H}^H\mathbf{H} + \frac{\sigma_w^2}{\sigma_x^2}\mathbf{I}_m\right)^{-1}$.
2: Perform the decomposition $\mathbf{A} = \mathbf{C}^H\mathbf{C}$ (for example by Cholesky decomposition).
3: Perform the LLL algorithm to obtain $\mathbf{C} \overset{LLL}{=} \widetilde{\mathbf{C}}\mathbf{T}$.
4: Identify $\mathbf{D} = \left(\mathbf{T}^{-1}\right)^H$.

Notice that the matrix $\boldsymbol{A}$ can also be rewritten as

$$\begin{aligned}\boldsymbol{A} &= \sigma_w^2\left(\boldsymbol{H}^H\boldsymbol{H} + \frac{\sigma_w^2}{\sigma_x^2}\boldsymbol{I}_m\right)^{-1} = \sigma_w^2\left(\underline{\boldsymbol{H}}^H\underline{\boldsymbol{H}}\right)^{-1}\\ &= \sigma_w^2\left(\underline{\boldsymbol{H}}^H\underline{\boldsymbol{H}}\right)^{-1}\underline{\boldsymbol{H}}^H\underline{\boldsymbol{H}}\left(\underline{\boldsymbol{H}}^H\underline{\boldsymbol{H}}\right)^{-1} = \boldsymbol{E}^H\boldsymbol{E},\end{aligned} \tag{15}$$

where $\boldsymbol{E} = \sigma_w\underline{\boldsymbol{H}}(\underline{\boldsymbol{H}}^H\underline{\boldsymbol{H}})^{-1}$ and $\underline{\boldsymbol{H}}$ is extended channel matrix in [14], [19]. In fact, $\boldsymbol{E}$ is the dual matrix of $\underline{\boldsymbol{H}}$ scaled by σ_w, i.e., $\boldsymbol{E} = \sigma_w[(\underline{\boldsymbol{H}}^H\underline{\boldsymbol{H}})^{-1}\underline{\boldsymbol{H}}^H]^H = \sigma_w\underline{\boldsymbol{H}}^{\#}$. Therefore, applying the LLL algorithm, we have $\boldsymbol{E} \overset{LLL}{=} \widetilde{\boldsymbol{E}}\boldsymbol{T}$ and $\underline{\boldsymbol{H}}^{\#} \overset{LLL}{=} \widetilde{\underline{\boldsymbol{H}}}\boldsymbol{T}$. In other words, the scheme $\underline{\boldsymbol{H}}^{\#} \overset{LLL}{=} \widetilde{\underline{\boldsymbol{H}}}\boldsymbol{T}$ used in [19] is a member of the proposed Type-I algorithms. But, since $\underline{\boldsymbol{H}}^{\#}$ has the dimension of $(n+m)\times m$, the complexity of applying LLL algorithm on $\underline{\boldsymbol{H}}^{\#}$ is more complex than the one on $\boldsymbol{C}$, which has the dimension $m\times m$.

4.3 Type-II Algorithms (Greedy-Search Based)

A greedy-search algorithm is proposed here as an alternative to Type-I to obtain an approximate solution of (10). As to be shown in Sect. 5, this algorithm performs better than Type-I and has a fixed computational complexity which is considered to be advantageous in hardware implementation.

Using $\boldsymbol{D}^H = [\boldsymbol{d}_1\boldsymbol{d}_2\cdots\boldsymbol{d}_m]$, the sum MSE can be rewritten as

$$tr(\boldsymbol{\Phi}_{MMSE}) = \sum_{k=1}^{m} \boldsymbol{d}_k^H \boldsymbol{A} \boldsymbol{d}_k = \sum_{k=1}^{m} mse_k, \tag{16}$$

where $mse_k \doteq \boldsymbol{d}_k^H \boldsymbol{A} \boldsymbol{d}_k$. In this algorithm, firstly we observe that $mse_k \doteq \boldsymbol{d}_k^H \boldsymbol{A} \boldsymbol{d}_k$ depending only on $\boldsymbol{d}_k$, $k = 1, \ldots, m$. Therefore, the updating of the matrix $\boldsymbol{D}^H$ can be done one column at a time to minimize mse_k, from the first to the last column. This iteration can be repeated again and again until no improvement is possible, starting from $\boldsymbol{D}^H = \boldsymbol{I}_m$.

Without loss of generality, consider that the kth-column of $\boldsymbol{D}^H$ is to be updated. A new column vector $\boldsymbol{d}_{k,new}$ is proposed as

$$\begin{aligned} \boldsymbol{d}_{k,new} = {} & \alpha_1 \boldsymbol{d}_1 + \cdots + \alpha_{k-1} \boldsymbol{d}_{k-1} + \boldsymbol{d}_k + \alpha_{k+1} \boldsymbol{d}_{k+1} + \cdots \\ & + \alpha_m \boldsymbol{d}_m, \end{aligned} \tag{17}$$

where $\{\alpha_m\}_{m \neq k}$ are parameters to be optimized to lower mse_k. It is shown in Appendix C that $\boldsymbol{D}_{new}^H = [\boldsymbol{d}_1, \ldots, \boldsymbol{d}_{k-1}, \boldsymbol{d}_{k,new}, \boldsymbol{d}_{k+1}, \ldots, \boldsymbol{d}_m]$ is unimodular provided that $\boldsymbol{D}^H$ is unimodular, and $\{\alpha_m\}_{m \neq k}$ are complex integers. With this new $\boldsymbol{d}_{k,new}$, $mse_{k,new}$ is given by

$$mse_{k,new} = \boldsymbol{d}_{k,new}^H \boldsymbol{A} \boldsymbol{d}_{k,new} = \begin{bmatrix} \boldsymbol{\alpha}_1^H & 1 & \boldsymbol{\alpha}_2^H \end{bmatrix} \boldsymbol{R} \begin{bmatrix} \boldsymbol{\alpha}_1 \\ 1 \\ \boldsymbol{\alpha}_2 \end{bmatrix}, \tag{18}$$

where $\boldsymbol{\alpha}_1 = [\alpha_1 \cdots \alpha_{k-1}]^T$, $\boldsymbol{\alpha}_2 = [\alpha_{k+1} \cdots \alpha_m]^T$, and $\boldsymbol{R} = \boldsymbol{D}\boldsymbol{A}\boldsymbol{D}^H$. Define $\boldsymbol{D}_1 = [\boldsymbol{d}_1, \ldots, \boldsymbol{d}_{k-1}]^H$ and $\boldsymbol{D}_2 = [\boldsymbol{d}_{k+1}, \ldots, \boldsymbol{d}_m]^H$. The matrix $\boldsymbol{R}$ can be partitioned as

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{D}_1 \boldsymbol{A} \boldsymbol{D}_1^H & \boldsymbol{D}_1 \boldsymbol{A} \boldsymbol{d}_k & \boldsymbol{D}_1 \boldsymbol{A} \boldsymbol{D}_2^H \\ \boldsymbol{d}_k^H \boldsymbol{A} \boldsymbol{D}_1^H & \boldsymbol{d}_k^H \boldsymbol{A} \boldsymbol{d}_k & \boldsymbol{d}_k^H \boldsymbol{A} \boldsymbol{D}_2^H \\ \boldsymbol{D}_2 \boldsymbol{A} \boldsymbol{D}_1^H & \boldsymbol{D}_2 \boldsymbol{A} \boldsymbol{d}_k & \boldsymbol{D}_2 \boldsymbol{A} \boldsymbol{D}_2^H \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_{1,1} & \boldsymbol{R}_{1,2} & \boldsymbol{R}_{1,3} \\ \boldsymbol{R}_{2,1} & \boldsymbol{R}_{2,2} & \boldsymbol{R}_{2,3} \\ \boldsymbol{R}_{3,1} & \boldsymbol{R}_{3,2} & \boldsymbol{R}_{3,3} \end{bmatrix}, \tag{19}$$

and (18) becomes

$$\begin{aligned} mse_{k,new} &= \begin{bmatrix} \boldsymbol{\alpha}_1^H & \mathbf{1} & \boldsymbol{\alpha}_2^H \end{bmatrix} \begin{bmatrix} \boldsymbol{R}_{1,1} & \boldsymbol{R}_{1,2} & \boldsymbol{R}_{1,3} \\ \boldsymbol{R}_{2,1} & \boldsymbol{R}_{2,2} & \boldsymbol{R}_{2,3} \\ \boldsymbol{R}_{3,1} & \boldsymbol{R}_{3,2} & \boldsymbol{R}_{3,3} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_1 \\ \mathbf{1} \\ \boldsymbol{\alpha}_2 \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{\alpha}_1^H \boldsymbol{\alpha}_2^H \end{bmatrix} \begin{bmatrix} \boldsymbol{R}_{1,1} & \boldsymbol{R}_{1,3} \\ \boldsymbol{R}_{3,1} & \boldsymbol{R}_{3,3} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_1 \\ \boldsymbol{\alpha}_2 \end{bmatrix} + \begin{bmatrix} \boldsymbol{\alpha}_1^H & \boldsymbol{\alpha}_2^H \end{bmatrix} \begin{bmatrix} \boldsymbol{R}_{1,2} \\ \boldsymbol{R}_{3,2} \end{bmatrix} \\ &\quad + \begin{bmatrix} \boldsymbol{R}_{2,1} & \boldsymbol{R}_{2,3} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_1 \\ \boldsymbol{\alpha}_2 \end{bmatrix} + \boldsymbol{R}_{2,2} \\ &= \boldsymbol{\alpha}_{\underline{k}}^H \boldsymbol{S}_{2,2} \boldsymbol{\alpha}_{\underline{k}} + \boldsymbol{\alpha}_{\underline{k}}^H \boldsymbol{S}_{2,1} + \boldsymbol{S}_{1,2} \boldsymbol{\alpha}_{\underline{k}} + \boldsymbol{S}_{1,1} \end{aligned} \tag{20}$$

where $\boldsymbol{\alpha}_{\underline{k}} = [\boldsymbol{\alpha}_1^T, \boldsymbol{\alpha}_2^T]^T$, $\boldsymbol{S}_{2,2} = \begin{bmatrix} \boldsymbol{R}_{1,1} & \boldsymbol{R}_{1,3} \\ \boldsymbol{R}_{3,1} & \boldsymbol{R}_{3,3} \end{bmatrix}$, $\boldsymbol{S}_{2,1} = [\boldsymbol{R}_{1,2}^T, \boldsymbol{R}_{3,2}^T]^T$, $\boldsymbol{S}_{1,2} = [\boldsymbol{R}_{2,1} \ \boldsymbol{R}_{2,3}]$ and $\boldsymbol{S}_{1,1} = \boldsymbol{R}_{2,2}$.

By differentiating $mse_{k,new}$ with respect to $\boldsymbol{\alpha}_{\underline{k}}$ and setting the result equal to zero, the optimal vector of $\boldsymbol{\alpha}_{\underline{k}}$, $\boldsymbol{\alpha}_{\underline{k},opt}$, is obtained by

$$\boldsymbol{S}_{2,2} \boldsymbol{\alpha}_{\underline{k},opt} = -\boldsymbol{S}_{2,1}. \tag{21}$$

Furthermore, define $\boldsymbol{D}_{\underline{k}} \doteq [\boldsymbol{D}_1^T, \boldsymbol{D}_2^T]^T$ be the matrix obtained by deleting the k-th row of the matrix $\boldsymbol{D}$. Then

$$\boldsymbol{S}_{2,2} = \begin{bmatrix} \boldsymbol{R}_{1,1} & \boldsymbol{R}_{1,3} \\ \boldsymbol{R}_{3,1} & \boldsymbol{R}_{3,3} \end{bmatrix} = \boldsymbol{D}_{\underline{k}} \boldsymbol{A} \boldsymbol{D}_{\underline{k}}^H. \tag{22}$$

Since $\boldsymbol{D}_{\underline{k}}$ has full row-rank (because $\boldsymbol{D}$ has full row-rank), from Appendix D, $\boldsymbol{S}_{2,2}$ is positive definite. Thus,

$$\boldsymbol{\alpha}_{\underline{k},opt} = \begin{bmatrix} \boldsymbol{\alpha}_{1,opt} \\ \boldsymbol{\alpha}_{2,opt} \end{bmatrix} = -\boldsymbol{S}_{2,2}^{-1} \boldsymbol{S}_{2,1}, \tag{23}$$

and

$$\boldsymbol{d}_{k,new} = \boldsymbol{D}^H \begin{bmatrix} \boldsymbol{\alpha}_{1,opt} \\ 1 \\ \boldsymbol{\alpha}_{2,opt} \end{bmatrix}. \tag{24}$$

Generally, the elements of $\boldsymbol{\alpha}_{\underline{k},opt}$ ar not complex integers and thus need to be rounded to ones in order to keep $\boldsymbol{D}_{new}^H$ a unimodular matrix (see Appendix C). Denote $\left[\left[\boldsymbol{\alpha}_{\underline{k},opt}\right]\right]_j$ be the rounding operation on the jth element of the vector $\boldsymbol{\alpha}_{\underline{k},opt}$, where more than one rounded values can be retained in order to improve performance, $\psi_{[\boldsymbol{\alpha}_{\underline{k}}]_j}$ be the set of retained complex integers in the rounding $\left[\left[\boldsymbol{\alpha}_{\underline{k},opt}\right]\right]_j$, and $\Psi_{\boldsymbol{\alpha}_{\underline{k}}} = \{\boldsymbol{\alpha}_{\underline{k}} = [\alpha_1 \cdots \alpha_{k-1} \alpha_{k+1} \cdots \alpha_m]^T, \alpha_j \in \psi_{[\boldsymbol{\alpha}_{\underline{k}}]_j}\}$. Then, the final $\boldsymbol{\alpha}$ to be used in updating can be obtained by

$$\boldsymbol{\alpha}^{upd} = \arg \min_{\boldsymbol{\alpha}_{\underline{k}} \in \Psi_{\boldsymbol{\alpha}_{\underline{k}}}} \begin{bmatrix} \boldsymbol{\alpha}_1^H & 1 & \boldsymbol{\alpha}_2^H \end{bmatrix} \boldsymbol{R} \begin{bmatrix} \boldsymbol{\alpha}_1 \\ \mathbf{1} \\ \boldsymbol{\alpha}_2 \end{bmatrix} \tag{25}$$

In our experience with extensive simulations, $|\psi_{[\boldsymbol{\alpha}_{\underline{k}}]_j}| > 2$ provides very little improvement, where $|\psi_{[\boldsymbol{\alpha}_{\underline{k}}]_j}|$ is the cardinality of the set $\psi_{[\boldsymbol{\alpha}_{\underline{k}}]_j}$. Therefore, $|\psi_{[\boldsymbol{\alpha}_{\underline{k}}]_j}| = 2$, $\forall j$ will be used in all discussions regarding Type-II algorithms. Consider the example of $m = n = 4$. If $\boldsymbol{\alpha}_{\underline{k},opt} = [-0.3 + 0.6i, 0.4 - 0.1i, 0.7 + 0.9i]^T$, where $i = \sqrt{-1}$, then $\psi_{[\boldsymbol{\alpha}_{\underline{k}}]_1} = \{i, 0\}$, and $\psi_{[\boldsymbol{\alpha}_{\underline{k}}]_2} = \{0, 1\}$, and $\psi_{[\boldsymbol{\alpha}_{\underline{k}}]_3} = \{1 + i, i\}$, and it leads to

$$\Psi_{\boldsymbol{\alpha}_{\underline{k}}} = \left\{ \begin{bmatrix} i \\ 0 \\ 1+i \end{bmatrix}, \begin{bmatrix} i \\ 0 \\ i \end{bmatrix}, \begin{bmatrix} i \\ 1 \\ 1+i \end{bmatrix}, \begin{bmatrix} i \\ 1 \\ i \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1+i \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ i \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 1+i \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ i \end{bmatrix} \right\}. \tag{26}$$

Finally, the new update of $\boldsymbol{d}_k$ is given by $\boldsymbol{d}_{k,upd} = \boldsymbol{D}^H [(\boldsymbol{\alpha}_1^{upd})^H \ 1 \ (\boldsymbol{\alpha}_2^{upd})^H]^H$, and $mse_{k,upd} = \boldsymbol{d}_{k,upd}^H \boldsymbol{A} \boldsymbol{d}_{k,upd}$, if $mse_{k,upd} < mse_k$. Otherwise, no update is performed and $\boldsymbol{d}_k$ remains no change. The complete algorithm is summarized in Table 3, where the algorithm is terminated if the maximum number of iteration N_I is reached. It is worthy to note that since at each step of updating $mse_{k,upd} \leq mse_k$ and the minimal mse_k is bounded below, the algorithm converges, although it may not converge to the minimal mse_k.

4.4 Complexity Analysis

The complexity of the proposed algorithms is analyzed in this sub-section based on the parameters of $m, n, N_I, |\psi_{[\boldsymbol{\alpha}_{\underline{k}}]_j}| = \kappa$, and $|\Omega|$. Since the detailed complexity calculation is quite tedious, only the final results are summarized here. Table 4 summaries the complexity analysis along with that of other algorithms considered in this work. In this analysis, the complexity of an algorithm is divided into two parts: the

Table 3 The Proposed Type-II Algorithm.

Input: $\mathbf{H}$, σ_x^2, σ_w^2, $\mathbf{A}$, $\mathbf{D}^H = \mathbf{I}_m$ and N_I, the maximum number of iterations.
Output: $\mathbf{D}^H$, a unimodular matrix.

1: $i = 0$
2: while $(i < N_I)$ do
3: for $k = 1$ to m do
4: $\boldsymbol{\alpha}_{\underline{k},\text{opt}} = -\mathbf{S}_{2,2}^{-1}\mathbf{S}_{2,1}$, and $\boldsymbol{\alpha}^{\text{upd}} = \arg\min_{\boldsymbol{\alpha}_{\underline{k}} \in \Psi_{\boldsymbol{\alpha}_{\underline{k}}}} \left[\boldsymbol{\alpha}_1^H \; 1 \; \boldsymbol{\alpha}_2^H\right] \mathbf{R} \begin{bmatrix} \boldsymbol{\alpha}_1 \\ 1 \\ \boldsymbol{\alpha}_2 \end{bmatrix}$
5: $\mathbf{d}_{k,\text{upd}} = \mathbf{D}^H \left[\left(\boldsymbol{\alpha}_1^{\text{upd}}\right)^H \; 1 \; \left(\boldsymbol{\alpha}_2^{\text{upd}}\right)^H\right]^H$, and $mse_{k,\text{upd}} = \mathbf{d}_{k,\text{upd}}^H \mathbf{A}\, \mathbf{d}_{k,\text{upd}}$
6: if $\left(mse_{k,\text{upd}} < mse_k\right)$, then
7: $\mathbf{d}_k = \mathbf{d}_{k,\text{upd}}$, $mse_k = mse_{k,\text{upd}}$
8: end if
9: end for
10: if (no update for all $k = 1$ to m), then
11: $i = N_I$
12: else
13: $i = i + 1$
14: end if
15: end while

initialization and main-body parts, where the complexity of Cholesky decomposition, Gram Schmidt Orthogonalization (GSO) and matrix inversion are those given in [27]. Note that for Type-I algorithm, because the QR decomposition [26] for the upper triangular matrix $\boldsymbol{C}$ is readily available with $\boldsymbol{Q} = \boldsymbol{I}_m$, and $\boldsymbol{R} = \boldsymbol{C}$ the complexity of GSO operation is reduced significantly. Table 5 gives the complexity for obtaining $\boldsymbol{d}_{k,upd}$ in Type-II algorithms which is used to calculate the main-body complexity of the Type-II algorithm. Since the updating needs to done m times in each iteration, the complexity of the main-body part is $m \cdot N_I$ times of that given in Table 5. The complexity of the MMSE detector is also calculated where the calculation is divided into three parts as given in Table 6.

5. Simulation Results

This section provides simulations to compare the proposed algorithms and those in the literature in the aspects of performance and complexity for the CFAD-MMSE detector. In the simulations, the data vectors $\boldsymbol{x}$ are transmitted on a frame-by-frame basis, with 200 data vectors per frame. To-

Table 4 Computational complexity of different channel factorization algorithms.

Algorithms	Computational Complexity					
	Initialization			Main-body of the Algorithm		
	Operations	Number of real multiplications	Number of real additions	Operations	Number of real multiplications	Number of real additions
LLL-extended	GSO for $\underline{\mathbf{H}}$	$4n \cdot m^2 + 4m^3 + 2m^2 + 6m$	$4n \cdot m^2 + 4m^3 + m^2 + 3m$	LLL	variable	variable
SA-extended	Calculation of $\underline{\mathbf{H}}^H \underline{\mathbf{H}}$ and $\left(\underline{\mathbf{H}}^H \underline{\mathbf{H}}\right)^{-1}$	$2n \cdot m^2 + 2n \cdot m$ $2m^3 + 2m^2$	$2n \cdot m^2 + 2n \cdot m$ $2m^3 + 2m^2$	SA	variable	variable
DLLL-extended	Calculation of $\underline{\mathbf{H}}^{\#}$	$6n \cdot m^2 + 2n \cdot m$ $2m^3 + 2m^2$	$6n \cdot m^2 + 2n \cdot m$ $2m^3 + 2m^2$	LLL	variable	variable
	GSO for $\underline{\mathbf{H}}^{\#}$	$4n \cdot m^2 + 4m^3 + 2m^2 + 6m$	$4n \cdot m^2 + 4m^3 + m^2 + 3m$	Calculation of $\left(\underline{\mathbf{T}}^{-1}\right)^H$	$2m^3 + 2m^2$	$2m^3 + 2m^2$
Type-I Algorithms	Calculation of $\mathbf{A}$	$2n \cdot m^2 + 2n \cdot m$ $2m^3 + 2m^2$	$2n \cdot m^2 + 2n \cdot m$ $2m^3 + 2m^2$	LLL	variable	variable
	Calculation of $\mathbf{C}$	$\frac{2m^3}{3}$	$\frac{2m^3}{3}$			
	GSO for $\mathbf{C}$	$2m^2 + 6m$	$m^2 + 3m$	Calculation of $\left(\mathbf{T}^{-1}\right)^H$	$2m^3 + 2m^2$	$2m^3 + 2m^2$
Type-II Algorithms	Calculation of $\mathbf{A}$	$2n \cdot m^2 + 2n \cdot m$ $2m^3 + 2m^2$	$2n \cdot m^2 + 2n \cdot m$ $2m^3 + 2m^2$	N_I and $\left\lvert\psi_{\lceil \underline{\alpha}_{\underline{k}} \rceil_j}\right\rvert = \kappa$	$\frac{N_I}{3}\left(2m^4 + 6m^3 + 6m^2 - 14m\right)$	$\frac{N_I}{3}\left(2m^4 + 18m^3 - 6m^2 - 14m + 18m \cdot \left(\kappa^{m-1} - 1\right)\right)$

Table 5 Computational complexity for obtaining $\boldsymbol{d}_{k,upd}$ in Type-II algorithms with $\left|\psi_{[\alpha_{\underline{k}}]_j}\right| = \kappa$.

Algorithm	Computational complexity of $\mathbf{d}_{k,\mathrm{upd}}$		
Type-II Algorithms	Operations	Number of real multiplications	Number of real additions
	Solve linear Equation set for $\boldsymbol{\alpha}_{\underline{k},\mathrm{opt}}$ in (21)	$\frac{2(m-1)^3}{3}$	$\frac{2(m-1)^3}{3}$
	Obtain $\boldsymbol{\alpha}^{\mathrm{upd}}$ in (25)	$4m^2-4$	$4m^2-4+6\cdot(\kappa^{m-1}-1)$
	Obtain integer vector $\mathbf{d}_{k,\mathrm{upd}}$	0	$4m^2-4m$

Table 6 Computational complexity of MMSE detector.

Algorithm	Computational complexity of MMSE detector		
MMSE detector in Fig. 2	Operations	Number of real multiplications	Number of real additions
	Calculation of $\mathbf{G}_{MMSE}$ in (7) and $\mathbf{D}^{-1}$	$8n\cdot m^2+2m^3+2m^2$	$8n\cdot m^2+2m^3+2m^2$
	Calculation of $\mathbf{G}_{MMSE}\mathbf{y}$ and $\mathbf{D}^{-1}\hat{\mathbf{z}}$	$4n\cdot m+4m^2$	$4n\cdot m+4m^2$
	Obtain $\hat{\mathbf{x}}$ by (6)	$4m\cdot\lvert\Omega\rvert$	$4m\cdot\lvert\Omega\rvert$

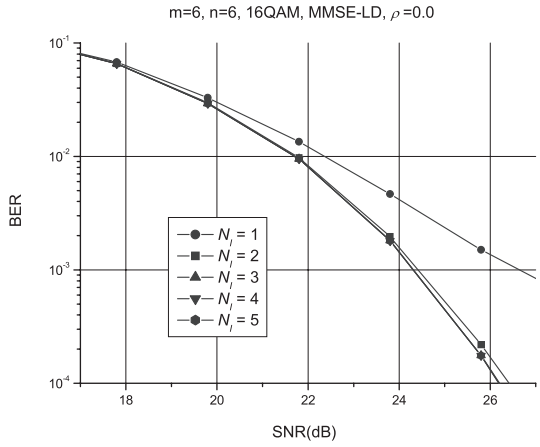

Fig. 3 The effect of N_I of Type-II algorithm on BER for the case of m = 6, n = 6, 16QAM, ρ = 0.0.

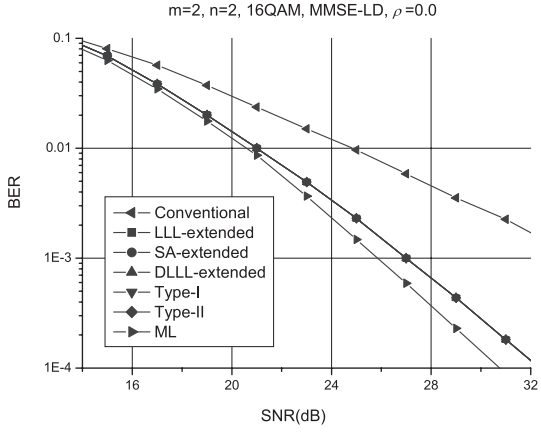

Fig. 4 BER comparisons of different channel factorizations for CFAD-MMSE in the case of m = 2, n = 2, 16QAM, ρ = 0.0.

tal of 10^4 frames are simulated. Signal constellation is fixed to 16QAM for easy comparisons between cases with different antenna numbers, although similar conclusions can be drawn for other constellation sizes according to our results not shown here. The channel is block faded; that is, $\boldsymbol{H}$ remains unchanged over a frame and changes independently from frame to frame. Signal to noise power ratio (SNR) is defined as $m\cdot\sigma_x^2/\sigma_w^2$. A total of five factorization algorithms are considered, including the LLL (LLL-extended [14]) and Syesen's algorithms (SA-extended [19], [23]) working on $\underline{\boldsymbol{H}}$, the LLL algorithm working on the dual matrix of $\underline{\boldsymbol{H}}$ (DLLL-extended [18], [19]), and the proposed Type-I and Type-II algorithms. It has been shown in [19] that reduction working on $\underline{\boldsymbol{H}}$ outperforms that on $\boldsymbol{H}$. Therefore, only those algorithms working on $\underline{\boldsymbol{H}}$ are compared here. The complex version of the LLL algorithm in Table 1 (with δ = 0.999) is used in all channel factorization methods that use LLL, where $[[\cdot]]$ stands for the operation of rounding its element to the nearest complex integer. The Cholesky decomposition is used to obtain $A = \boldsymbol{C}^H\boldsymbol{C}$ in Type-I algorithm, and $|\psi_{[\alpha_{\underline{k}}]_j}|$ = κ = 2 in Type-II algorithm.

The bit-error-rate (BER) for uncorrelated MIMO channels (ρ = 0.0) are compared first. Figure 3 shows the effect of iteration number N_I on the BER performance of Type-II Algorithm for the case of $m = n = 6$. As can be seen, there is almost no performance improvement with $N_I > 3$. As a result, we use $N_I = 3$ for the subsequent comparisons. Figures 4, 5, 6 and 7 show the comparisons between different channel factorizations for the cases of $m = n = 2$, $m = n = 4$, $m = n = 6$, and $m = n = 8$ respectively. In these figures, the performance of conventional (non-factorized) MMSE detector and optimum maximum-likelihood (ML) is also provided for reference. With smaller numbers of antennas, e.g., $m = n = 2, 4$, all the considered channel factorization algorithms perform similarly especially for $m = n = 2$; the CFAD, however, provides significant improvement over the conventional MMSE detector.

As expected, DLLL-extended and Type-I have the same performance because DLLL-extended is a member of the Type-I algorithms, as discussed in Sect. 4. SA-extended performs closely to Type-I and outperforms LLL-extended by about 1.5–2.5 dB at BER = 10^{-4} for $m = n = 6, 8$. Type-II has the best performance with a 0.5 dB gain margin over

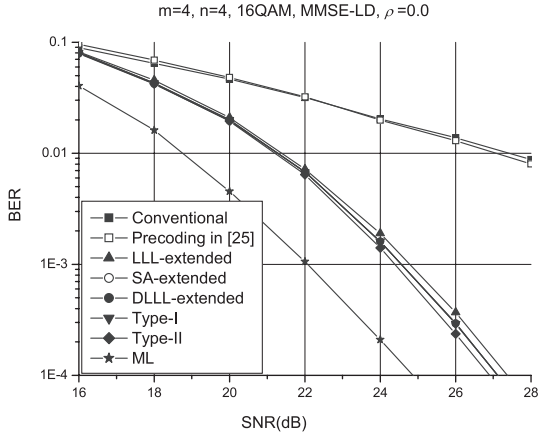

Fig. 5 BER comparisons of different channel factorizations for CFAD-MMSE in the case of $m = 4$, $n = 4$, 16QAM, $\rho = 0.0$.

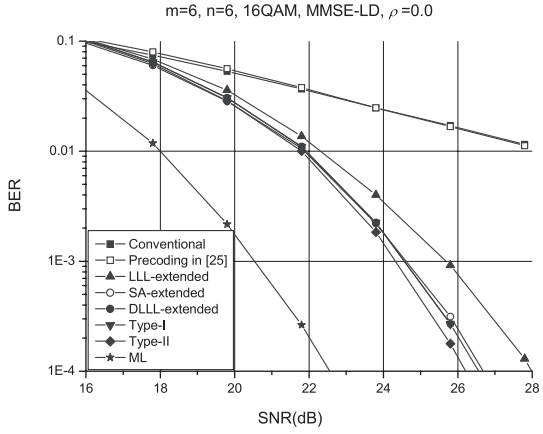

Fig. 6 BER comparisons of different channel factorizations for CFAD-MMSE in the case of $m = 6$, $n = 6$, 16QAM, $\rho = 0.0$.

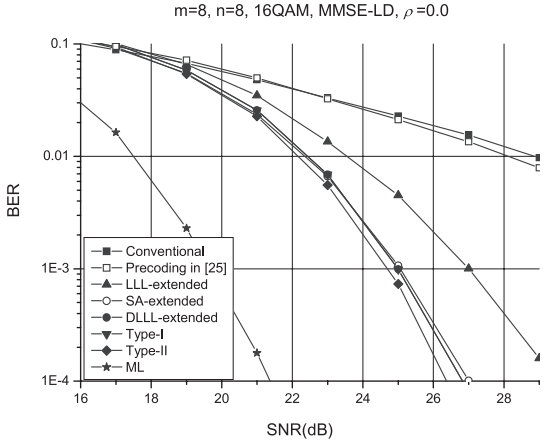

Fig. 7 BER comparisons of different channel factorizations for CFAD-MMSE in the case of $m = 8$, $n = 8$, 16QAM, $\rho = 0.0$.

Type-I (DLLL-extended) and SA-extended. In addition, it is evidently that CFAD detectors are capable of attaining the same diversity order, i.e., the slope of BER curve at high SNR region, as that of ML. In Figs. 5, 6 and 7, the performance of the best-performed pre-coder for un-coded† system proposed in [25] is also given for comparison purpose. Clearly, the pre-coder's performance is inferior to that of the proposed CFAD methods. In [25], [29], the performance improvement with pre-coding was shown to be much higher for the cases of $m > n$.

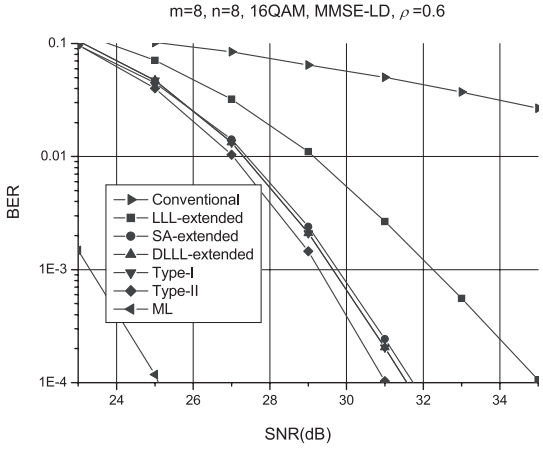

Fig. 8 BER comparisons of different channel factorizations for CFAD-MMSE in the case of $m = 8$, $n = 8$, 16QAM, $\rho = 0.6$.

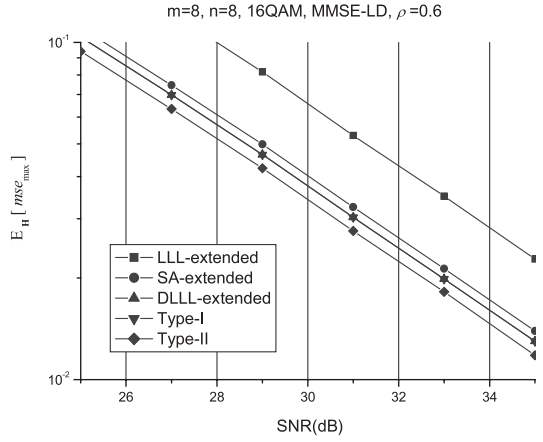

Fig. 9 $E_H\,[mse_{\max}]$ comparisons of different channel factorizations for CFAD-MMSE in the case of $m = 8$, $n = 8$, 16QAM, $\rho = 0.6$.

The comparisons over the correlated MIMO channels ($\rho = 0.6$) are shown in Fig. 8 for the case of $m = n = 8$. As shown in the figure, the channel correlation degrades the performance of the conventional MMSE detector very dramatically. In addition, Type-II outperforms Type-I (DLLL-extended) and SA-extended by about 0.7 dB. In Fig. 9, we compare $E_H[mse_{\max}]$ between different algorithms, where $mse_{\max} = \max_k\{mse_k\}$ is the worst square error among all the receive branches, and $E_H[\cdot]$ denotes the average operation over $\boldsymbol{H}$. $mse_{\max}$ dominates the BER performance. As is shown, Type-II shows its superiority over others.

†Note that the "code" of "pre-coder" and "pre-coding" means spatial beam-forming at the transmitter. On the other hand, the "code" of "un-coded system" means forward error correction coding.

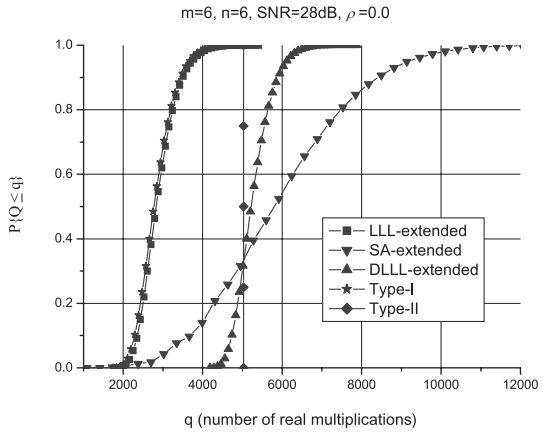

Fig. 10 Complexity comparisons of different channel factorization algorithms.

Figure 10 is an example of computational complexity comparison between different channel factorizations, where the empirical CDF (cumulative density function) of the number of real multiplications is shown for $m = n = 6$, SNR = 28 dB, $\rho = 0.0$. Since the hardware implementation cost of a multiplication is much higher than that of an addition [28], only multiplications is taken into consideration here for complexity comparison. Noticeably, the LLL-based methods (LLL-extended, DLLL-extended and Type-I) and SA-extended all have a variable complexity; Type-I has the less complexity, then, LLL-extended, DLLL-extended and SA-extended. The same order of complexity is observed for other cases not shown here. In this specific example, Type-II is more complex than LLL and SA-based algorithms for about 35% of the channel realizations. Nevertheless, Type-II has a fixed computational complexity which is considered to be advantageous in hardware implementation.

In this following, the complexity of MIMO receiver (channel factorization plus MMSE detection) is compared specifically when Type-I, SA-extended and DLLL-extended are employed as the factorization algorithm. Recall that these algorithms have a similar BER performance as shown in Figs. 4–8. The comparison is made from two aspects: *hardware complexity* and *computational complexity per data vector*. For data vectors where pilots are located, both channel factorization and MMSE detection are required to be performed, and, therefore, for a fixed hardware clock rate, extra circuitry is needed for the computation of the factorization algorithm, and that increases hardware complexity. Table 7 shows the hardware complexity ratio of channel factorization to overall MIMO receiver for the considered factorization algorithms, where N_{cf} and N_{MIMO} stand for number of real multiplications needed for channel factorization and MIMO MMSE detector, respectively. As is seen, the ratio ranges from 44% to 74%. Therefore, how to reduce the complexity of channel factorization algorithm is an important issue. In addition, from Table 7, it can be shown that the saving of *hardware complexity* of overall MIMO receiver offered by Type-I ranges from 21% to 43%. For example, the saving for SA-extended for the case of $P\{\mathrm{Q} \le \mathrm{q}\} = 0.5$ is

$$\frac{(5804 + 2904) - (2772 + 2904)}{5804 + 2904} = 34\%.$$

The *computational complexity per data vector*, evaluated by the number of real multiplications per data vector $(N_{cf}/p) + N_{MIMO}$, is also employed for comparisons, where p is the number of data vectors in a frame. Figure 11 is such a comparison for $P\{Q \le q\} = 0.9$. Similar results are observed for 10 and 50 percentiles although they are not shown here for brevity. As is shown, the complexity saving by Type-I is quite significant for small p. For example, for $p = 6$ the savings are $(4307 - 3479)/4307 = 19\%$ and $(3887 - 3479)/3887 = 11\%$ for SA-extended, and DLLL-extended respectively. As expected, the complexity saving becomes smaller for larger p.

In practical mobile cellular systems, channel estimation is usually done for every 0.5 to 1 ms in order to cover mobility up to 350 km/hour [30], [31]. For example, in the 3GPP-LTE specification, there are 7 (OFDM) symbols in a slot (0.5 ms) where time-frequency multiplexed pilots are

Table 7 Hardware complexity ratio of channel factorization algorithm to overall MIMO receiver, i.e., $\frac{N_{cf}}{N_{cf}+N_{MIMO}}$, for the case of $m = n = 6$, 16QAM, SNR = 28 dB, $\rho = 0.0$.

Comparison points	SA-extended	DLLL-extended	Type-I
$P\{Q \le q\} = 0.1$	$\frac{3686}{3686+2904} = 56\%$	$\frac{4735}{4735+2904} = 62\%$	$\frac{2287}{2287+2904} = 44\%$
$P\{Q \le q\} = 0.5$	$\frac{5804}{5804+2904} = 67\%$	$\frac{5220}{5220+2904} = 64\%$	$\frac{2772}{2772+2904} = 49\%$
$P\{Q \le q\} = 0.9$	$\frac{8415}{8415+2904} = 74\%$	$\frac{5897}{5897+2904} = 67\%$	$\frac{3449}{3449+2904} = 54\%$

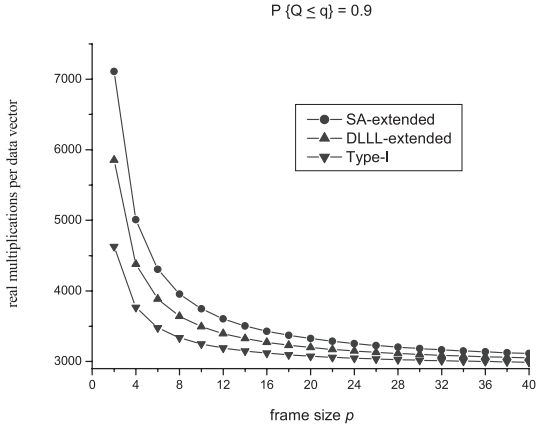

Fig. 11 Number of real multiplications per data vector for different channel factorization algorithms with $m = n = 6$, 16QAM, SNR = 28 dB, $\rho = 0.0$ and $P\{Q \le q\} = 0.9$.

used for the cell-specific channel estimation [30]. Also, in the IEEE 802.16 m specification, there are 5 to 7 (OFDM) symbols in a sub-frame (around 1 ms) where time-frequency multiplexed pilots are used for channel estimation. In this type of systems, Type-I algorithm is particular useful. In wireless LAN systems, on the other hand, the so-called preamble-based training is employed where pilot signals are placed at the beginning of a data packet. In this case, if the packet size p is large, says over 30, the saving provided by Type-I becomes quite small.

6. Conclusions

A new channel factorization design is proposed for the channel-factorization aided detectors, where effective factorization algorithms are sought to minimize the sum mean-squared-error of the MMSE detector. Two new types of factorization algorithms are devised; the first type is LLL based, where the best-performed factorization algorithm found in the literature, i.e., the DLLL-extended algorithm, is a member of this type but with a higher complexity. The second type is greedy-search based which can provide around 0.5–1.0 dB gain over the first type and has a fixed computational complexity which is advantageous in hardware implementation. The computational complexity of the proposed methods was analyzed and compared to the existing methods.

Acknowledgments

The authors would like to thank anonymous reviewers for their valuable comments.

References

[1] G.J. Foschini and M.J. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," Wirel. Pers. Commun., vol.6, pp.311–335, March 1998.
[2] S. Verdu, Multiuser Detection, Cambridge University Press, UK, 1998.
[3] W.H. Mow, "Maximum likelihood sequence estimation from the lattice viewpoint," IEEE Trans. Inf. Theory, vol.40, no.5, pp.1591–1600, Sept. 1994.
[4] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," Math. Comput., vol.44, pp.463–471, April 1985.
[5] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," IEEE Trans. Inf. Theory, vol.48, no.8, pp.2201–2214, Aug. 2002.
[6] G.D. Golden, G.J. Foschini, R.A. Valenzuela, and P.W. Wolniansky, "Detection algorithm and initial laboratory results using V-BLAST space-time communication architecture," Electron. Lett., vol.35, pp.14–16, Jan. 1999.
[7] G.J. Foschini, G.D. Golden, R.A. Valenzuela, and P.W. Wolniansky, "Simplified processing for high spectral efficiency wireless communication employing multi-element arrays," IEEE J. Sel. Areas Commun., vol.17, no.11, pp.1841–1852, Nov. 1999.
[8] D. Wübben, R. Böhnke, J. Rinas, V. Kühn, and K.D. Kammeyer, "Efficient algorithm for decoding layered space-time codes," Electron. Lett., vol.37, pp.1348–1350, Oct. 2001.
[9] R. Böhnke, D. Wübben, V. Kühn, and K.D. Kammeyer, "Reduced complexity MMSE detection for BLAST architecture," Proc. IEEE Globecom, pp.2258–2262, Dec. 2003.
[10] H. Yao and G. Wornell, "Lattice-reduction-aided detectors for MIMO communication systems," Proc. IEEE Globecom, pp.424–428, Nov. 2002.
[11] C. Windpasseinger and R.F.H. Fischer, "Low-complexity near-maximum-likelihood detection and precoding for MIMO systems using lattice reduction," Proc. IEEE ITW, pp.345–348, March 2003.
[12] W.H. Mow, "Universal lattice decoding: Principle and recent advances," Wireless Commun. Mobile Comput., vol.3, pp.553–569, Aug. 2003.
[13] I. Berengure and X. Wang, "MIMO antenna selection with lattice-reduction-aided linear receivers," IEEE Trans. Veh. Technol., vol.53, no.5, pp.1289–1302, Sept. 2004.
[14] D. Wübben, R. Böhnke, V. Kühn, and K.D. Kammeyer, "Near-maximum-likelihood detection of MIMO systems using MMSE-based lattice reduction," Proc. IEEE ICC, pp.798–802, June 2004.
[15] D. Wübben, V. Kühn, and K.D. Kammeyer, "On the robustness of lattice-reduction aided detectors in correlated MIMO systems," Proc. IEEE VTC, pp.3639–3643, Sept. 2004.
[16] D. Wübben, R. Böhnke, V. Kühn, and K.D. Kammeyer, "MMSE-based lattice-reduction for near-ML detection of MIMO systems," Proc. ITG Workshop on Smart Antennas, pp.106–113, March 2004.
[17] C. Ling, "Approximate lattice decoding: Primal versus dual basis reduction," Proc. IEEE ISIT, pp.1–5, July 2006.
[18] M. Taherzadeh, A. Mobasher, and A.K. Khandani, "LLL reduction achieves the receiver diversity in MIMO decoding," IEEE Trans. Inf. Theory, vol.53, pp.4801–4805, Dec. 2007.
[19] D. Wübben and D. Seethaler, "On the performance of lattice reduction schemes for MIMO data detection," Proc. IEEE ACSSC, pp.1543–1538, Nov. 2007.
[20] A.K. Lenstra, H.W. Lenstra, Jr., and L. Lovász, "Factoring polynomials with rational coefficients," Math. Ann., no.261, pp.513–534, 1983.
[21] C.P. Schnorr and M. Euchner, "Lattice basis reduction: Improved practical algorithms and solving subset sum problem," Mathematical Programming, vol.66, pp.181–191, 1994.
[22] M. Seysen, "Simultaneous reduction of a lattice basis and its reciprocal basis," Combinatorica, vol.13, pp.363–376, 1993.
[23] D. Seethaler, G. Matz, and F. Hlawatsch, "Reduced-complexity MIMO data detection using seysen's lattice reduction algorithm," Proc. IEEE ICASSP, pp.53–56, April 2007.
[24] Y.H. Gan and W.H. Mow, "Complex lattice reduction algorithm for reduced-complexity MIMO detection," Proc. IEEE Globecom, pp.2953–2957, Dec. 2005.
[25] S. Zhou and B. Li, "BER criterion and codebook construction for finite-rate precoded spatial multiplexing with linear receivers," IEEE Trans. Signal Process., vol.54, no.5, pp.1653–1665, May 2006.
[26] S.H. Friedberg, A.J. Insel, and L.E. Spence, Linear Algebra, 2nd ed., Prentice-Hall, Englewood Cliffs, NJ, 1989.
[27] G.H. Golub and C.F. Van Loan, Matrix Computations, 1st ed., The Johns Hopkins University Press, 1983.
[28] A. Lindström, M. Nordseth, and L. Bengtsson, "0.13 μm CMOS synthesis of common arithmetic units," Technical Report no.03-11, Department of Computer Engineering, Chalmers University of Technology, Aug. 2003.
[29] J.C. Roh and B.D. Rao, "Multiple antenna channels with partial channel state information at the transmitter," IEEE Trans. Wirel. Commun., vol.3, no.2, pp.677–688, March 2004.
[30] S. Sesia, I. Toufik, and M. Baker, LTE–The UMTS Long Term Evolution, Chapter 8, Wiley, 2009.
[31] Draft Amendment to IEEE Standard for Local and Metropolitan Area Networks, Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems: Advanced Air Interface, IEEE P802.16 m/D6 May 2010, Section 16.3.5.4.

Appendix A

By definition, $\boldsymbol{\Phi}_{MMSE} = E[(\boldsymbol{G}_{MMSE}\boldsymbol{y} - \boldsymbol{z})(\boldsymbol{G}_{MMSE}\boldsymbol{y} - \boldsymbol{z})^H]$, where

$$\boldsymbol{G}_{MMSE} = \boldsymbol{D}\left(\boldsymbol{H}^H\boldsymbol{H} + \frac{\sigma_w^2}{\sigma_x^2}\boldsymbol{I}_m\right)^{-1}\boldsymbol{H}^H.$$

Using $\boldsymbol{\Phi}_x = E[\boldsymbol{x}\boldsymbol{x}^H] = \sigma_x^2\boldsymbol{I}_m$, and $\boldsymbol{\Phi}_w = E[\boldsymbol{w}\boldsymbol{w}^H] = \sigma_w^2\boldsymbol{I}_n$, we have

$$\begin{aligned}\boldsymbol{\Phi}_{MMSE} &= \boldsymbol{G}_{MMSE}\left(\sigma_x^2\boldsymbol{H}\boldsymbol{H}^H + \sigma_w^2\boldsymbol{I}_n\right)\boldsymbol{G}_{MMSE}^H \\ &-\sigma_x^2\boldsymbol{G}_{MMSE}\boldsymbol{H}\boldsymbol{D}^H - \sigma_x^2\boldsymbol{D}\boldsymbol{H}^H\boldsymbol{G}_{MMSE}^H + \sigma_x^2\boldsymbol{D}\boldsymbol{D}^H \quad (\text{A}\cdot 1)\end{aligned}$$

Clearly, $\boldsymbol{H}^H(\boldsymbol{H}\boldsymbol{H}^H + \sigma_w^2\boldsymbol{I}_n/\sigma_x^2) = (\boldsymbol{H}^H\boldsymbol{H} + \sigma_w^2\boldsymbol{I}_m/\sigma_x^2)\boldsymbol{H}^H$. Since $(\boldsymbol{H}\boldsymbol{H}^H + \sigma_w^2\boldsymbol{I}_n/\sigma_x^2)$ and $(\boldsymbol{H}^H\boldsymbol{H} + \sigma_w^2\boldsymbol{I}_m/\sigma_x^2)$ are invertible, it leads to

$$\left(\boldsymbol{H}^H\boldsymbol{H} + \frac{\sigma_w^2}{\sigma_x^2}\boldsymbol{I}_m\right)^{-1}\boldsymbol{H}^H = \boldsymbol{H}^H\left(\boldsymbol{H}\boldsymbol{H}^H + \frac{\sigma_w^2}{\sigma_x^2}\boldsymbol{I}_n\right)^{-1}. \quad (\text{A}\cdot 2)$$

According to (A· 2), $\boldsymbol{G}_{MMSE}$ can be rewritten as:

$$\begin{aligned}G_{MMSE} &= \boldsymbol{D}\left(\boldsymbol{H}^H\boldsymbol{H} + \frac{\sigma_w^2}{\sigma_x^2}\boldsymbol{I}_m\right)^{-1}\boldsymbol{H}^H = \boldsymbol{D}\boldsymbol{H}^H\left(\boldsymbol{H}\boldsymbol{H}^H + \frac{\sigma_w^2}{\sigma_x^2}\boldsymbol{I}_n\right)^{-1} \\ &= \sigma_x^2\boldsymbol{D}\boldsymbol{H}^H\left(\sigma_x^2\boldsymbol{H}\boldsymbol{H}^H + \sigma_w^2\boldsymbol{I}_n\right)^{-1}, \quad (\text{A}\cdot 3)\end{aligned}$$

Thus, $\boldsymbol{G}_{MMSE}(\sigma_x^2\boldsymbol{H}\boldsymbol{H}^H + \sigma_w^2\boldsymbol{I}_n)\boldsymbol{G}_{MMSE}^H = \sigma_x^2\boldsymbol{D}\boldsymbol{H}^H\boldsymbol{G}_{MMSE}^H$, and (A· 1) becomes

$$\begin{aligned}\boldsymbol{\Phi}_{MMSE} &= \sigma_x^2\boldsymbol{D}\boldsymbol{D}^H - \sigma_x^2\boldsymbol{G}_{MMSE}\boldsymbol{H}\boldsymbol{D}^H \\ &= \sigma_x^2\boldsymbol{D}\boldsymbol{D}^H - \sigma_x^2\boldsymbol{D}\boldsymbol{H}^H\left(\boldsymbol{H}\boldsymbol{H}^H + \frac{\sigma_w^2}{\sigma_x^2}\boldsymbol{I}_n\right)^{-1}\boldsymbol{H}\boldsymbol{D}^H, \quad (\text{A}\cdot 4)\end{aligned}$$

By applying the Matrix Inversion Lemma $\boldsymbol{K}^{-1}$–$\boldsymbol{K}^{-1}\boldsymbol{L}(\boldsymbol{N}\boldsymbol{K}^{-1}\boldsymbol{L} + \boldsymbol{M}^{-1})^{-1}\boldsymbol{N}\boldsymbol{K}^{-1} = (\boldsymbol{K} + \boldsymbol{L}\boldsymbol{M}\boldsymbol{N})^{-1}$ [27] to (A· 4) with $\boldsymbol{K}^{-1} = \sigma_x^2\boldsymbol{D}\boldsymbol{D}^H$, $\boldsymbol{L} = (\boldsymbol{D}^{-1})^H\boldsymbol{H}^H$, $\boldsymbol{M}^{-1} = \sigma_w^2\boldsymbol{I}_n/\sigma_x^2$, and $\boldsymbol{N} = \boldsymbol{H}\boldsymbol{D}^{-1}/\sigma_x^2$,

$$\boldsymbol{\Phi}_{MMSE} = \sigma_w^2\boldsymbol{D}\left(\boldsymbol{H}^H\boldsymbol{H} + \frac{\sigma_w^2}{\sigma_x^2}\boldsymbol{I}_m\right)^{-1}\boldsymbol{D}^H = \boldsymbol{D}\boldsymbol{A}\boldsymbol{D}^H,$$

where

$$A = \sigma_w^2\left(\boldsymbol{H}^H\boldsymbol{H} + \frac{\sigma_w^2}{\sigma_x^2}\boldsymbol{I}_m\right)^{-1}.$$

Appendix B

In this appendix, it is proved that given $\boldsymbol{C} = \boldsymbol{P}\boldsymbol{\Upsilon}$, where $\boldsymbol{\Upsilon}$ is a non-singular matrix and $\boldsymbol{P}^H\boldsymbol{P} = \boldsymbol{I}$, if $\boldsymbol{\Upsilon} \overset{LLL}{=} \widetilde{\boldsymbol{\Upsilon}}\boldsymbol{T}_\Upsilon$, and $\boldsymbol{C} \overset{LLL}{=} \widetilde{\boldsymbol{C}}\boldsymbol{T}_C$, then $\widetilde{\boldsymbol{C}} = \boldsymbol{P}\widetilde{\boldsymbol{\Upsilon}}$ and $\boldsymbol{T}_\Upsilon = \boldsymbol{T}_C$, where $\boldsymbol{X} \overset{LLL}{=} \widetilde{\boldsymbol{X}}\boldsymbol{T}_X$ denotes that the factorization is done with the LLL algorithm.

Proof: Let $\boldsymbol{X} \in \{\boldsymbol{\Upsilon}, \boldsymbol{C}\}$. Initially, in the LLL algorithm, the Gram Schmidt Orthogonalization (GSO) procedure is applied to obtain the factorization

$$\boldsymbol{X} \overset{GSO}{=} \boldsymbol{Q}_X\boldsymbol{U}_X^T,$$

where $\boldsymbol{Q}_X = [\boldsymbol{q}_{1,X}\cdots\boldsymbol{q}_{m,X}]$ is a orthogonal matrix, and $\boldsymbol{U}_X$ is a lower triangular matrix with unit main diagonal elements. Since $\boldsymbol{C} = \boldsymbol{P}\boldsymbol{\Upsilon}$ and $\boldsymbol{P}^H\boldsymbol{P} = \boldsymbol{I}$, it is easy to show that $\boldsymbol{Q}_C = \boldsymbol{P}\boldsymbol{Q}_\Upsilon$ and $\boldsymbol{U}_C = \boldsymbol{U}_\Upsilon = \boldsymbol{U}$. Let $\boldsymbol{C} = [\boldsymbol{c}_1\cdots\boldsymbol{c}_m]$, $\boldsymbol{\Upsilon} = [\boldsymbol{\Upsilon}_1\cdots\boldsymbol{\Upsilon}_m]$, and

$$\boldsymbol{U} = \begin{bmatrix} 1 & & & 0 \\ \mu_{2,1} & 1 & & \\ \vdots & \vdots & \ddots & \\ \mu_{m,1} & \mu_{m,2} & \cdots & 1 \end{bmatrix},$$

the LLL algorithm then performs the following two operations iteratively to obtain a reduced basis:

Operation 1 : $\boldsymbol{x}_i = \boldsymbol{x}_i - \left\lfloor\mu_{i,j}\right\rceil\boldsymbol{x}_j$, if $\left\lfloor\mu_{i,j}\right\rceil \neq 0$.

Operation 2: swap $\boldsymbol{x}_{i-1}$ and $\boldsymbol{x}_i$, if $\left\|\boldsymbol{q}_{i,x} + \mu_{i,i-1}\boldsymbol{q}_{i-1,x}\right\|^2 < \delta\left\|\boldsymbol{q}_{i-1,x}\right\|^2$.

where $\boldsymbol{x} \in \{\boldsymbol{c}, \boldsymbol{\Upsilon}\}$ and $\lfloor\cdot\rceil$ stands for the operation of rounding the argument to the nearest complex integer. Since $\boldsymbol{c}_i - \lfloor\mu_{i,j}\rceil\boldsymbol{c}_j = \boldsymbol{P}\boldsymbol{\Upsilon}_i - \lfloor\mu_{i,j}\rceil\boldsymbol{P}\boldsymbol{\Upsilon}_j = \boldsymbol{P}(\boldsymbol{\Upsilon}_i - \lfloor\mu_{i,j}\rceil\boldsymbol{\Upsilon}_j)$ in Operation 1, and $\|\boldsymbol{q}_{i,\boldsymbol{c}}\|^2 = \|\boldsymbol{q}_{i,\Upsilon}\|^2$ and $\|\boldsymbol{q}_{i,\boldsymbol{c}} + \mu_{i,i-1}\boldsymbol{q}_{i-1,\boldsymbol{c}}\|^2 = \|\boldsymbol{q}_{i,\Upsilon} + \mu_{i,i-1}\boldsymbol{q}_{i-1,\Upsilon}\|^2$ in Operation 2, it concludes that $\widetilde{\boldsymbol{C}} = \boldsymbol{P}\widetilde{\boldsymbol{\Upsilon}}$ and $\boldsymbol{T}_\Upsilon = \boldsymbol{T}_C$.

Appendix C

In this Appendix, we prove that $\boldsymbol{D}_{new}^H = [\boldsymbol{d}_1, \ldots, \boldsymbol{d}_{k-1}, \boldsymbol{d}_{k,new}, \boldsymbol{d}_{k+1}, \ldots, \boldsymbol{d}_m]$ is unimodular provided that $\boldsymbol{D}^H$ is unimodular, where

$$\begin{aligned}\boldsymbol{d}_{k,new} &= \alpha_1\boldsymbol{d}_1 + \cdots + \alpha_{k-1}\boldsymbol{d}_{k-1} + \boldsymbol{d}_k + \alpha_{k+1}\boldsymbol{d}_{k+1} \\ &\quad + \cdots + \alpha_m\boldsymbol{d}_m. \quad (\text{A}\cdot 5)\end{aligned}$$

and $\{\alpha_m\}_{m\neq k}$ are complex integers.

Proof: Recall that $\boldsymbol{D}^H = [\boldsymbol{d}_1, \ldots, \boldsymbol{d}_{k-1}, \boldsymbol{d}_k, \boldsymbol{d}_{k+1}, \ldots, \boldsymbol{d}_m]$. Using (A· 5), $\boldsymbol{D}_{new}^H = \boldsymbol{D}^H\boldsymbol{I}_m^k$, where $\boldsymbol{I}_m^k$ is obtained by replacing the kth-column of $\boldsymbol{I}_m$ with $[\alpha_1, \cdots, \alpha_{k-1}, 1, \alpha_{k+1}\cdots, \alpha_m]^T$. It is clear that $\det(\boldsymbol{I}_m^k) = 1$. Therefore, $|\det(\boldsymbol{D}_{new}^H)| = |\det(\boldsymbol{I}_m^k)| \cdot |\det(\boldsymbol{D}^H)| = 1$, and $\boldsymbol{D}_{new}^H$ is a unimodular matrix.

Appendix D

In this appendix, it is proved that the matrix $\boldsymbol{O}\boldsymbol{A}\boldsymbol{O}^H$ is positive definite provided the $l \times m$ matrix $\boldsymbol{O}$ has full row rank, where $l \leq m$, and $\boldsymbol{A} = \sigma_w^2(\boldsymbol{H}^H\boldsymbol{H} + \sigma_w^2\boldsymbol{I}_m/\sigma_x^2)^{-1}$.

Proof: Firstly, $\boldsymbol{A} = \sigma_w^2(\boldsymbol{H}^H\boldsymbol{H} + \sigma_w^2\boldsymbol{I}_m/\sigma_x^2)^{-1}$ is positive definite. This can be shown as follows. Let $\boldsymbol{X} = \boldsymbol{H}^H\boldsymbol{H} + \sigma_w^2\boldsymbol{I}_m/\sigma_x^2$. It is clear that $\boldsymbol{X}$ is Hermitian and semi-positive definite, i.e., $\boldsymbol{u}^H\boldsymbol{X}\boldsymbol{u} \geq 0$, for all $m \times 1$ vector $\boldsymbol{u}$. Suppose $\boldsymbol{u}^H\boldsymbol{X}\boldsymbol{u} = 0$, we have $\boldsymbol{u}^H\boldsymbol{H}^H\boldsymbol{H}\boldsymbol{u} + \sigma_w^2\boldsymbol{u}^H\boldsymbol{u}/\sigma_x^2 = 0$, and that implies $\boldsymbol{u} = \boldsymbol{0}_m$. Hence, $\boldsymbol{X}$ is positive definite, and so is $\boldsymbol{A} =$

$\sigma_w^2(\boldsymbol{X})^{-1}$. Secondly, since $\boldsymbol{A}$ is positive definite, $\boldsymbol{A} = \boldsymbol{J}^H\boldsymbol{J}$ for an upper triangular matrix $\boldsymbol{J}$ by applying, for example, the Cholesky decomposition [26]. Given $l \times 1$ vector $\boldsymbol{v}$,

$$\boldsymbol{v}^H\boldsymbol{O}\boldsymbol{A}\boldsymbol{O}^H\boldsymbol{v} = \boldsymbol{v}^H\boldsymbol{O}\boldsymbol{J}^H\boldsymbol{J}\boldsymbol{O}^H\boldsymbol{v} = \left\|\boldsymbol{J}\boldsymbol{O}^H\boldsymbol{v}\right\|^2 \geq 0,$$

therefore $\boldsymbol{O}\boldsymbol{A}\boldsymbol{O}^H$ is semi-positive definite. In addition, if $\boldsymbol{v}^H\boldsymbol{O}\boldsymbol{A}\boldsymbol{O}^H\boldsymbol{v} = 0$, it follows that $\boldsymbol{O}^H\boldsymbol{v} = \boldsymbol{0}$ (as $\boldsymbol{A}$ is positive definite), and $\boldsymbol{v} = \boldsymbol{0}$ because $\boldsymbol{O}$ has full row rank, and this concludes the proof.

Chih-Cheng Kuo was born on March 5, 1977 in Hsinchu, Taiwan. He received the B.S. degree in the department of industry engineering and management from the National Chiao Tung University (NCTU), Taiwan, in 1999, and the M.S. degree in the department of communication engineering from the same University in 2001. He is currently working toward the Ph.D. degree in the department of communication engineering at NCTU. He had been studying in several areas including communications, signal processing, optimization theory, and numerical linear algebra. His current research interests are focused on wireless and mobile communications, OFDM-CDMA, and MIMO space–time processing for wireless systems.

Wern-Ho Sheen received the B.S. degree from the National Taiwan University of Science and Technology, Taiwan in 1982, the M.S. degree from the National Chiao Tung University, Taiwan in 1984, and the Ph.D. degree from the Georgia Institute of Technology, Atlanta, USA in 1991. From 1993 to 2001, he was with the National Chung Cheng University, Taiwan, where he held positions as Professor in the Department of Electrical Engineering and the Managing Director of the Center for Telecommunication Research. From 2001 to 2009, he was a Professor of the Department of Communications Engineering, National Chiao Tung University. Currently he is a Professor of the Department of Information and Communication Engineering, Chaoyang University of Technology, Taiwan. His research interests include general areas of communication theory, cellular mobile and personal radio systems, adaptive signal processing for wireless communications, spread spectrum communications, and VLSI design for wireless communications systems. Prof. Sheen has been consulting extensively for the industry and research institutes in Taiwan.

Chang-Lung Hsiao received the B.S. and M.S. degrees in electrical engineering from the National Tsing Hua University, Hsinchu, Taiwan, in 1995 and 1998, respectively. Since 1998, he has been with the Wireless Communication Technology Division of Industrial Technology Research Institute, Hsinchu, Taiwan. His research interests include channel estimation, error correcting coding, and multiple antennas signal processing for wireless communication systems.