# Efficient Inverse Cholesky Factorization for Alamouti Matrices in G-STBC and Alamouti-like Matrices in OMP

Hufei Zhu, Ganghua Yang
Communications Technology Laboratory
Huawei Technologies Co. Ltd., P. R. China
Email: zhuhufei@huawei.com, yang.yangganghua@huawei.com

Wen Chen
the Department of Electronic Engineering
Shanghai Jiao Tong University, Shanghai, P. R. China
Email: wenchen@sjtu.edu.cn

***Abstract*—We apply the efficient inverse Cholesky factorization to Alamouti matrices in Groupwise Space-time Block Code (G-STBC) and Alamouti-like matrices in Orthogonal Matching Pursuit (OMP) for the sub-Nyquist sampling system. By utilizing some good properties of Alamouti or Alamouti-like matrices, we save about half the complexity. Then we propose the whole square-root algorithms for G-STBC and OMP, respectively. The proposed square-root G-STBC algorithm has an average speedup of $2.96 \sim 3.6$ with respect to the sub-optimal G-STBC algorithm. On the other hand, when comparing the complexities of all the steps except the projection step, the complexity for the proposed square-root OMP algorithm is about $30\%$ of that for the fast OMP algorithm by matrix inverse update.**

## I. Introduction

The equivalent channel matrix in Group-wise space-time block code (G-STBC) is an Alamouti matrix [1]–[3], i.e., a block matrix involving $2 \times 2$ Alamouti sub-blocks $\begin{bmatrix} c_1 & -c_2^* \\ c_2 & c_1^* \end{bmatrix}$ (where $c_1$ and $c_2$ are complex scalars). Alamouti matrices have some good properties [3], which have been utilized to develop efficient matrix manipulations [1] and a fast recursive G-STBC algorithm [2].

On the other hand, in the sub-Nyquist sampling system [4], the Orthogonal Matching Pursuit (OMP) algorithm [5]–[7] computes the inversion of Alamouti-like matrices, i.e., block matrices involving $2 \times 2$ Alamouti-like sub-blocks $\begin{bmatrix} c_1 & c_2^* \\ c_2 & c_1^* \end{bmatrix}$ (that is obtained by deleting the negative sign in the Alamouti sub-block). Similar to Alamouti matrices, Alamouti-like matrices have also some good properties that can be utilized to reduce the computational complexity, as shown in this paper.

In this paper, the efficient inverse Cholesky factorization proposed by us recently [8] is applied to Alamouti matrices in G-STBC and Alamouti-like matrices in OMP. We exploit the properties of Alamouti or Alamouti-like matrices to reduce the complexity dramatically. Then we develop efficient square-root algorithms for G-STBC and OMP, respectively.

In the following sections, $(\bullet)^T$, $(\bullet)^*$ and $(\bullet)^H$ denote matrix transposition, matrix conjugate, and matrix conjugate transposition, respectively. $\mathrm{I}_M$ is the identity matrix with size $M$, and $\mathbf{0}_M$ is the zero column vector with length $M$.

## II. System Models

In this section, we introduce the system model for G-STBC and that for OMP in the sub-Nyquist sampling system [4].

### A. System Model for G-STBC

The considered G-STBC system [2] consists of $2M$ transmit antennas and $N(\geq M)$ receive antennas in a rich-scattering and flat-fading wireless channel. The transmit side de-multiplexes a single data stream $a_{11}, a_{12}, \cdots, a_{M1}, a_{M2}$ into $M$ layers. Each layer $a_{m1}, a_{m2}$ $(m = 1, 2, \cdots, M)$ is encoded by an independent Alamouti encoder, and then it is transmitted by its respective 2 transmit antennas in 2 time slots. Let $h_{n,i}$ $(n = 1, 2, \cdots, N$ and $i = 1, 2, \cdots, 2M)$ denote the channel coefficient from transmit antenna $i$ to receive antenna $n$. Let $x_{n1}$ and $x_{n2}$ $(n = 1, 2, \cdots, N)$ denote the signal received at antenna $n$ in the $1^{st}$ time slot and the $2^{nd}$ time slot, respectively. Then the equivalent channel model [2] for G-STBC can be represented as

$$\mathbf{x} = \mathbf{H}\mathbf{a} + \mathbf{w}, \tag{1}$$

where $\mathbf{w}$ is the complex Gaussian noise vector with statistically independent entries, the equivalent receive signal vector

$$\mathbf{x} = \begin{bmatrix} x_{11} & x_{12}^* & \cdots & x_{N1} & x_{N2}^* \end{bmatrix}^T, \tag{2}$$

the equivalent transmit signal vector

$$\mathbf{a} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{M1} & a_{M2} \end{bmatrix}^T, \tag{3}$$

and the $2N \times 2M$ equivalent channel matrix

$$\mathbf{H} = \begin{pmatrix} h_{11} & h_{12} & \cdots & h_{1,2M-1} & h_{1,2M} \\ h_{12}^* & -h_{11}^* & \cdots & h_{1,2M}^* & -h_{1,2M-1}^* \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ h_{N1} & h_{N2} & \cdots & h_{N,2M-1} & h_{N,2M} \\ h_{N2}^* & -h_{N1}^* & \cdots & h_{N,2M}^* & -h_{N,2M-1}^* \end{pmatrix}. \tag{4}$$

Assume $E(\mathbf{a}\mathbf{a}^H) = \sigma_a^2 \mathrm{I}_{2M}$ and $E(\mathbf{w}\mathbf{w}^H) = \sigma_w^2 \mathrm{I}_{2N}$. The minimum mean-square error (MMSE) estimation of $\mathbf{a}$ is [2]

$$\hat{\mathbf{a}} = \mathbf{R}^{-1}\mathbf{H}^H\mathbf{x}, \tag{5}$$

where

$$\mathbf{R} = \mathbf{H}^H\mathbf{H} + \sigma_w^2/\sigma_a^2 \mathbf{I}_{2M}. \tag{6}$$

978-1-4673-1881-5/12/$31.00 ©2012 IEEE

The recursive G-STBC algorithm [2] detects $M$ Alamouti codes iteratively with the optimal ordering. In each iteration, the undetected Alamouti code with the highest signal-to-noise ratio (SNR) is detected by (5), and then its effect is cancelled.

### *B. System Model for OMP in Sub-Nyquist Sampling System*

Compressive sensing (CS) theory [9] asserts that a signal vector $\mathbf{s} \in \mathbb{C}^N$ with at most $K$ nonzero components can be exactly recovered from a measurement vector $\mathbf{y} \in \mathbb{C}^M$ with $M \ll N$. The signal vector $\mathbf{s}$ is called $K$-sparse. The measurement vector

$$\mathbf{y} = \mathbf{T}\mathbf{s}, \tag{7}$$

where the full rank $M \times N$ matrix $\mathbf{T} = \left[ \begin{array}{cccc} \mathbf{t}_1 & \mathbf{t}_2 & \cdots & \mathbf{t}_N \end{array} \right]$ is commonly called the dictionary, and its $i^{th}$ column, i.e. $\mathbf{t}_i$, is commonly called an atom [6].

The signal recovery problem is to find the sparse solution to the underdetermined linear system (7), i.e., to represent $\mathbf{y}$ as a weighted sum of the least number of atoms. The signal recovery algorithms include OMP [5]–[7], a greedy algorithm that selects atoms from $\mathbf{T}$ iteratively to approximate $\mathbf{y}$ gradually. In each iteration, OMP selects the atom best matching the current residual (i.e. the approximation error), renews the weights for all the already selected atoms (to obtain the least squares approximation of $\mathbf{y}$), and then updates the residual accordingly.

In [4], the signal vector (i.e. X(f) in [4, left line 18 on page 387]) is conjugate symmetry, and so is the dictionary[1]. Thus the OMP algorithm is slightly modified to select a symmetric pair of atoms in each iteration [4, left line 16 on page 387]. Accordingly we modify the OMP algorithm described in [6, left lines 27-34 on page 2] to summarize the OMP algorithm in [4]. We also follow these conventions in [6]: the set $\mathbf{\Gamma}^n$ is a set containing $n$ indices, and the matrix $\mathbf{T}_{\mathbf{\Gamma}^n}$ is a sub-matrix of $\mathbf{T}$ containing only those columns of $\mathbf{T}$ with indices in $\mathbf{\Gamma}^n$. Finally, we summarize the OMP algorithm in [4] as follows.

**The OMP Algorithm in [4]**

```
1. Initialise $\mathbf{\Gamma}^0 = \emptyset$, the residual $\tilde{\mathbf{y}}^0 = \mathbf{y}$.
2. for $k = 1, 2, \cdots, K/2$
    (2.a) Projection: $\nabla^k = \mathbf{T}^H \tilde{\mathbf{y}}^{k-1}$.
    (2.b) $i^k = \underset{i=1}{\overset{N}{\operatorname{argmax}}} \left( \nabla_i^k \right)$.
    (2.c) Select a symmetric pair of atoms by
    $\mathbf{\Gamma}^{2k} = \mathbf{\Gamma}^{2k-2} \cup i^k \cup \left( N - i^k + 1 \right)$.
    (2.d) Renew the weights for the selected $2k$ atoms:
        $\mathbf{s}_{2k} = \left( \mathbf{T}_{\mathbf{\Gamma}^{2k}}^H \mathbf{T}_{\mathbf{\Gamma}^{2k}} \right)^{-1} \mathbf{T}_{\mathbf{\Gamma}^{2k}}^H \mathbf{y}.$   (8)
    (2.e) Update the residual by $\tilde{\mathbf{y}}^k = \mathbf{y} - \mathbf{T}_{\mathbf{\Gamma}^{2k}} \mathbf{s}_{2k}$.
3. Output: $\tilde{\mathbf{y}}^{K/2}$, $\mathbf{s}_{2 \times K/2} = \mathbf{s}_K$.
```

## III. Efficient Inverse Cholesky Factorizations for Alamouti Matrices and Alamouti-like Matrices

We need to compute matrix inversions in (5) and (8). $\mathbf{R}$ in (5) is an Alamouti matrix [2]. On the other hand, in (8) let

$$\mathbf{R}_{\mathbf{\Gamma}^{2k}} = \mathbf{T}_{\mathbf{\Gamma}^{2k}}^H \mathbf{T}_{\mathbf{\Gamma}^{2k}}. \tag{9}$$

Obviously, $\mathbf{R}_{\mathbf{\Gamma}^{2k}} = \left[ \mathbf{t}_{i^1} \ \mathbf{t}_{i^1}^* \cdots \mathbf{t}_{i^k} \ \mathbf{t}_{i^k}^* \right]^H \left[ \mathbf{t}_{i^1} \ \mathbf{t}_{i^1}^* \cdots \mathbf{t}_{i^k} \ \mathbf{t}_{i^k}^* \right] =$ $\begin{bmatrix} \mathbf{t}_{i^1}^H \mathbf{t}_{i^1} & \mathbf{t}_{i^1}^H \mathbf{t}_{i^1}^* & \cdots & \mathbf{t}_{i^1}^H \mathbf{t}_{i^k} & \mathbf{t}_{i^1}^H \mathbf{t}_{i^k}^* \\ \mathbf{t}_{i^1}^T \mathbf{t}_{i^1} & \mathbf{t}_{i^1}^T \mathbf{t}_{i^1}^* & \cdots & \mathbf{t}_{i^1}^T \mathbf{t}_{i^k} & \mathbf{t}_{i^1}^T \mathbf{t}_{i^k}^* \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{t}_{i^k}^H \mathbf{t}_{i^1} & \mathbf{t}_{i^k}^H \mathbf{t}_{i^1}^* & \cdots & \mathbf{t}_{i^k}^H \mathbf{t}_{i^k} & \mathbf{t}_{i^k}^H \mathbf{t}_{i^k}^* \\ \mathbf{t}_{i^k}^T \mathbf{t}_{i^1} & \mathbf{t}_{i^k}^T \mathbf{t}_{i^1}^* & \cdots & \mathbf{t}_{i^k}^T \mathbf{t}_{i^k} & \mathbf{t}_{i^k}^T \mathbf{t}_{i^k}^* \end{bmatrix}$ is an Alamouti-like matrix.

To reuse the already listed equations for brevity, let us use (i)$_{\mathbf{\Gamma}}$ to express equation (i) with the subscript $k$ of each matrix replaced by $\mathbf{\Gamma}^k$ and with overlines added to all vectors and scalars. Then $\mathbf{R}_{2k}$ (i.e. the $2k \times 2k$ submatrix in the first $2k$ rows and columns of $\mathbf{R}$) and $\mathbf{R}_{\mathbf{\Gamma}^{2k}}$ can be represented as [2]

$$\mathbf{R}_{2k} = \begin{bmatrix} \mathbf{R}_{2k-2} & \mathbf{v}'_{2k-2} & \mathbf{v}''_{2k-2} \\ \mathbf{v}'^H_{2k-2} & \beta_{2k-1} & r_{2k-1,2k} \\ \mathbf{v}''^H_{2k-2} & r_{2k-1,2k}^* & \beta_{2k} \end{bmatrix} \tag{10}$$

and (10)$_{\mathbf{\Gamma}}$, respectively. The inverse Cholesky factor [8] of $\mathbf{R}_{2k}$ is $\mathbf{F}_{2k}$ that satisfies

$$\mathbf{F}_{2k} \mathbf{F}_{2k}^H = \mathbf{R}_{2k}^{-1}. \tag{11}$$

We can simply apply equations (11) and (17) in [8] twice to compute $\mathbf{F}_{2k}$ from $\mathbf{F}_{2k-2}$. Then we have

$$\mathbf{F}_{2k} = \begin{bmatrix} \mathbf{F}_{2k-2} & \mathbf{u}'_{2k-2} & \mathbf{u}''_{2k-2} \\ \mathbf{0}_{2k-2}^H & \lambda_{2k-1} & f_{2k-1,2k} \\ \mathbf{0}_{2k-2}^H & 0 & \lambda_{2k} \end{bmatrix}, \tag{12}$$

where we can employ equation (17) in [8] to compute

$$\begin{cases} \lambda_{2k-1} = 1 / \sqrt{\beta_{2k-1} - \mathbf{v}'^H_{2k-2} \mathbf{F}_{2k-2} \mathbf{F}_{2k-2}^H \mathbf{v}'_{2k-2}}, & \text{(13a)} \\ \mathbf{u}'_{2k-2} = -\lambda_{2k-1} \mathbf{F}_{2k-2} \mathbf{F}_{2k-2}^H \mathbf{v}'_{2k-2}. & \text{(13b)} \end{cases}$$

We can also employ equation (17) in [8] to compute the last column of $\mathbf{F}_{2k}$. However, we will find some algorithms that are more efficient, to reduce the computational complexity.

### *A. Inverse Cholesky Factorization for Alamouti Matrices*

$\mathbf{R}_{2k}$ is an Alamouti matrix, and so are $\mathbf{R}_{2k}^{-1}$ [3, Lemma 1 (Invariance Under Inversion)]. Then the LDL$^T$ factors of $\mathbf{R}_{2k}^{-1}$ are Alamouti matrices [3, Lemma 3 (Invariance Under Triangular Factorization)], which include the unit lower triangular $\mathbf{L}$ and the diagonal $\mathbf{D} = \operatorname{diag}\left( \begin{array}{cccc} d_1 & d_2 & \cdots & d_{2k} \end{array} \right)$. Accordingly $\mathbf{L}\sqrt{\mathbf{D}} = \mathbf{L} \cdot \operatorname{diag}\left( \begin{array}{ccc} \sqrt{d_1} & \cdots & \sqrt{d_{2k}} \end{array} \right)$ is also an Alamouti matrix, which is the unique Cholesky factor of $\mathbf{R}_{2k}^{-1}$ [10, Theorem 4.2.5 (Cholesky Factorization)], and is equivalent to $\mathbf{F}_{2k}$ (since the permuted $\mathbf{F}_{2k}$ is the Cholesky

[1] The dictionary $\mathbf{T} = \tilde{\mathbf{S}}\tilde{\mathbf{F}}\mathbf{D}$ [4, equation (19)]. $\tilde{\mathbf{F}} = \left[ \begin{array}{ccc} \bar{\mathbf{F}}_{L_0} & \cdots & \bar{\mathbf{F}}_{-L_0} \end{array} \right]$ [4, right line 19 on page 380] is conjugate symmetry since $\bar{\mathbf{F}}_{-i} = \bar{\mathbf{F}}_i^*$ [4, equation (18)]. $\tilde{\mathbf{S}}$ is real, since $\tilde{\mathbf{S}}_{ik} = \alpha_{ik}$ [4, right line 22 on page 380] and $\alpha_{ik}$ is real [4, the line below equation (4)]. Moreover, $\mathbf{D} = diag\left( \begin{array}{ccc} d_{L_0} & \cdots & d_{-L_0} \end{array} \right)$ [4, right line 23 on page 380] has $d_{-i} = d_i^*$ [4, equation (16)]. Thus we can verify that $\mathbf{T} = \tilde{\mathbf{S}}\tilde{\mathbf{F}}\mathbf{D} = \left[ \begin{array}{ccc} \tilde{\mathbf{S}}\bar{\mathbf{F}}_{L_0} d_{L_0} & \cdots & \tilde{\mathbf{S}}\bar{\mathbf{F}}_{-L_0} d_{-L_0} \end{array} \right]$ has $\tilde{\mathbf{S}}\bar{\mathbf{F}}_{-i} d_{-i} = (\tilde{\mathbf{S}}\bar{\mathbf{F}}_i d_i)^*$, i.e., $\mathbf{T} = \tilde{\mathbf{S}}\tilde{\mathbf{F}}\mathbf{D}$ is conjugate symmetry.

factor of the correspondingly permuted $\mathbf{R}_{2k}^{-1}$ [8]). Thus $\mathbf{F}_{2k}$ must be an Alamouti matrix. Correspondingly in (12),

$$\begin{cases} f_{2k-1,2k} = 0, & \text{(14a)} \\ \lambda_{2k} = \lambda_{2k-1}, & \text{(14b)} \\ \mathbf{u}''_{2k-2} = \text{Alamouti}(\mathbf{u}'_{2k-2}), & \text{(14c)} \end{cases}$$

where (14c) means that $\mathbf{u}''_{2k-2}$ is obtained from $\mathbf{u}'_{2k-2}$ directly since $\left[\begin{array}{cc} \mathbf{u}'_{2k-2} & \mathbf{u}''_{2k-2} \end{array}\right]$ is an Alamouti matrix.

### *B. Inverse Cholesky Factorization for Alamouti-like Matrices*

It is easy to verify that the sum, difference, or product of two Alamouti-like matrices is another Alamouti-like matrix. Then we can prove that the inversion of an Alamouti-like matrix is another Alamouti-like matrix, as the proof for [3, Lemma 1 (Invariance Under Inversion)].

Now let us consider $(12)_{\mathbf{\Gamma}}$. We employ $(13b)_{\mathbf{\Gamma}}$ to obtain

$$\left[\begin{array}{c} \bar{\mathbf{u}}''_{2k-2} \\ \bar{f}_{2k-1,2k} \end{array}\right] = -\bar{\lambda}_{2k} \mathbf{F}_{\mathbf{\Gamma}^{2k-1}} \mathbf{F}_{\mathbf{\Gamma}^{2k-1}}^{H} \left[\begin{array}{c} \bar{\mathbf{v}}''_{2k-2} \\ \bar{r}_{2k-1,2k} \end{array}\right] \qquad (15)$$

where $\mathbf{F}_{\mathbf{\Gamma}^{2k-1}} = \left[\begin{array}{cc} \mathbf{F}_{\mathbf{\Gamma}^{2k-2}} & \bar{\mathbf{u}}'_{2k-2} \\ \mathbf{0}_{2k-2}^{H} & \bar{\lambda}_{2k-1} \end{array}\right]$. From (15) we deduce

$$\begin{cases} \bar{\mathbf{u}}^{''}_{2k-2} = -\bar{\lambda}_{2k} \hat{\mathbf{u}}^{''}_{2k-2} - \bar{\lambda}_{2k} \hat{f}_{2k-1,2k} \bar{\mathbf{u}}^{'}_{2k-2}, & \text{(16a)} \\ \bar{f}_{2k-1,2k} = -\bar{\lambda}_{2k} \bar{\lambda}_{2k-1} \hat{f}_{2k-1,2k}, & \text{(16b)} \end{cases}$$

where

$$\begin{cases} \hat{f}_{2k-1,2k} = \bar{\mathbf{u}}^{'H}_{2k-2} \bar{\mathbf{v}}''_{2k-2} + \bar{\lambda}_{2k-1} \bar{r}_{2k-1,2k}, & \text{(17a)} \\ \hat{\mathbf{u}}^{''}_{2k-2} = \mathbf{F}_{\mathbf{\Gamma}^{2k-2}} \mathbf{F}_{\mathbf{\Gamma}^{2k-2}}^{H} \bar{\mathbf{v}}^{''}_{2k-2}. & \text{(17b)} \end{cases}$$

We can avoid computing (17b) and obtain $\hat{\mathbf{u}}^{''}_{2k-2}$ from $\mathbf{F}_{\mathbf{\Gamma}^{2k-2}} \mathbf{F}_{\mathbf{\Gamma}^{2k-2}}^{H} \bar{\mathbf{v}}^{'}_{2k-2} = -\bar{\mathbf{u}}'_{2k-2} / \bar{\lambda}_{2k-1}$ directly, since $\mathbf{F}_{\mathbf{\Gamma}^{2k-2}} \mathbf{F}_{\mathbf{\Gamma}^{2k-2}}^{H} \left[\begin{array}{cc} \bar{\mathbf{v}}'_{2k-2} & \bar{\mathbf{v}}''_{2k-2} \end{array}\right] = \mathbf{R}_{\mathbf{\Gamma}^{2k-2}}^{-1} \left[\begin{array}{cc} \bar{\mathbf{v}}'_{2k-2} & \bar{\mathbf{v}}''_{2k-2} \end{array}\right]$ is an Alamouti-like matrix.

It can be seen from $(12)_{\mathbf{\Gamma}}$ that the Alamouti-like sub-block in the last two rows and columns of $\mathbf{F}_{\mathbf{\Gamma}^{2k}} \mathbf{F}_{\mathbf{\Gamma}^{2k-2}}^{H}$ is $\left[\begin{array}{cc} \bar{\lambda}_{2k-1} & \bar{f}_{2k-1,2k} \\ 0 & \bar{\lambda}_{2k} \end{array}\right] \left[\begin{array}{cc} \bar{\lambda}_{2k-1} & 0 \\ \bar{f}^{*}_{2k-1,2k} & \bar{\lambda}_{2k} \end{array}\right] = \left[\begin{array}{cc} \bar{\lambda}_{2k-1}^{2} + \left|\bar{f}_{2k-1,2k}\right|^{2} & \bar{f}_{2k-1,2k} \bar{\lambda}_{2k} \\ \bar{f}^{*}_{2k-1,2k} \bar{\lambda}_{2k} & \bar{\lambda}_{2k}^{2} \end{array}\right]$, from which we deduce

$$\bar{\lambda}_{2k}^{2} = \bar{\lambda}_{2k-1}^{2} + \left|\bar{f}_{2k-1,2k}\right|^{2}. \qquad (18)$$

Finally we substitute (16b) into (18) to obtain

$$\bar{\lambda}_{2k} = 1 / \sqrt{\bar{\lambda}_{2k-1}^{-2} - \left|\hat{f}_{2k-1,2k}\right|^{2}}. \qquad (19)$$

Now we only need to compute (17a), (19) and (16) to obtain the last column of $\mathbf{F}_{2k}$.

## IV. Efficient square-root algorithms for G-STBC and OMP in Sub-Nyquist Sampling System

### *A. Efficient square-root algorithm for G-STBC*

We use $\mathbf{F}_{2m}$ (i.e. the equivalent Cholesky factor of $\mathbf{R}_{2m}^{-1}$) instead of $\mathbf{R}_{2m}^{-1}$ in the recursive G-STBC algorithm [2], to develop a square-root algorithm for G-STBC. Similar to the square-root BLAST algorithm [8], in the initialization phase we permute the channel matrix $\mathbf{H}$ to be

$$\mathbf{H}_{2M} = \left[\begin{array}{cccc} \mathbf{h}_{2t_1-1} & \mathbf{h}_{2t_1} \cdots & \mathbf{h}_{2t_M-1} & \mathbf{h}_{2t_M} \end{array}\right] \qquad (20)$$

with $t_M, t_{M-1}, \cdots, t_1$ to be the assumed successive detection order of $M$ Alamouti codes, while in the iterative detection phase, we perform interference cancellation in

$$\mathbf{z}_{2M} = \mathbf{H}_{2M}^{H} \mathbf{x}, \qquad (21)$$

and we block-upper triangularize $\mathbf{F}_{2m}$ by

$$\mathbf{F}_{2m} \mathbf{\Sigma} = \begin{bmatrix} \mathbf{F}_{2m-2} & \mathbf{u}'_{2m-2} & \mathbf{u}''_{2m-2} \\ \mathbf{0}_{2m-2}^{H} & f^{*}_{2m,2m} & -f^{*}_{2m,2m-1} \\ \mathbf{0}_{2m-2}^{H} & f_{2m,2m-1} & f_{2m,2m} \end{bmatrix} \qquad (22)$$

where $\mathbf{\Sigma}$ is unitary. From (11), (21) and (5) we obtain $\hat{\mathbf{a}}_{2m} = \mathbf{F}_{2m} \mathbf{F}_{2m}^{H} \mathbf{z}_{2m}$, i.e., $\hat{\mathbf{a}}_{2m} = (\mathbf{F}_{2m} \mathbf{\Sigma})(\mathbf{F}_{2m} \mathbf{\Sigma})^{H} \mathbf{z}_{2m}$, into which we substitute (22) to estimate the last two entries of $\hat{\mathbf{a}}_{2m}$ by

$$\begin{array}{c} \left[\begin{array}{c} \hat{a}_{2m-1} \\ \hat{a}_{2m} \end{array}\right] = \left[\begin{array}{cc} f^{*}_{2m,2m} & f^{*}_{2m,2m-1} \\ f_{2m,2m-1} & f_{2m,2m} \end{array}\right] \times \\ \left[\begin{array}{ccc} \mathbf{u}^{'H}_{2k-2} & f_{2m,2m} & f^{*}_{2m,2m-1} \\ \mathbf{u}^{''H}_{2k-2} & -f_{2m,2m-1} & f^{*}_{2m,2m} \end{array}\right] \mathbf{z}_{2m} \end{array}. \qquad (23)$$

The estimations $\hat{a}_{2m-1}$ and $\hat{a}_{2m}$ are quantized into $a_{2m-1}$ and $a_{2m}$, respectively, whose effect is then cancelled in $\mathbf{z}_{2m}$ by [2]

$$\mathbf{z}_{2m-2} = \mathbf{z}_{2m}^{[-2]} - \left[\mathbf{v}^{'}_{2m-2} \; \mathbf{v}^{''}_{2m-2}\right] \left[a_{2m-1} \; a_{2m}\right]^{T}, \qquad (24)$$

where $\mathbf{z}_{2m}^{[-2]}$ is $\mathbf{z}_{2m}$ with the last two entries removed, and $\mathbf{v}^{'}_{2m-2}$ and $\mathbf{v}^{''}_{2m-2}$ are in $\mathbf{R}_{2m}$, as shown in (10).

Table I summarizes the proposed square-root G-STBC algorithm. Notice that the transformation $\mathbf{\Sigma}$ in step (b) can be performed by the Givens rotations for Alamouti matrices that include inner rotations and outer rotations [1]. Generally, in the two minimum length rows (found in step (a)) that can be represented as rows $2k-1$ and $2k$, the entries in the fist $2k-2$ columns are all zeros. Let $\mathbf{\Omega}_{i,j}^{n}$ denote a complex Givens rotation matrix that rotates the $i^{th}$ entry and the $j^{th}$ entry in each row of $\bar{\mathbf{F}}_m$ and zeros the $i^{th}$ entry in the $n^{th}$ row [8]. Then the transformation $\mathbf{\Sigma}$ in step (b) can be performed by

$$\mathbf{\Sigma} = \prod_{j=k}^{m} \mathbf{\Omega}_{2j,2j-1}^{2k-1} \prod_{j=k}^{m-1} \left(\mathbf{\Omega}_{2j-1,2j+1}^{2k-1} \mathbf{\Omega}_{2j,2j+2}^{2k}\right), \qquad (25)$$

where $\mathbf{\Omega}_{2j,2j-1}^{2k-1}$ is an inner rotation that zeros two entries (i.e. $f_{2k-1,2j}$ and $f_{2k,2j-1}$) in an Alamouti sub-block and rotates the other two entries (i.e. $f_{2k-1,2j-1}$ and $f_{2k,2j}$) into real values, and $\mathbf{\Omega}_{2j-1,2j+1}^{2k-1} \mathbf{\Omega}_{2j,2j+2}^{2k}$ is an outer rotation that zeros the remaining two real values (i.e. $f_{2k-1,2j-1}$ and $f_{2k,2j}$) in a previously inner-rotated Alamouti sub-block [1]. Since $\mathbf{F}_{2m}$ is still an Alamouti matrix after an inner or outer rotation, only half of the entries need to be computed [1].

TABLE I

THE PROPOSED SQUARE-ROOT ALGORITHM FOR G-STBC

| | |
|---|---|
| **Initialization** | Permute $\mathbf{H}$ into $\mathbf{H}_{2M}$ by (20), and permute $\mathbf{a}_{2M} = \mathbf{a}$ accordingly. Compute $\mathbf{R}_{2M}$ and $\mathbf{z}_{2M}$ by (6) and (21), respectively. Compute $\mathbf{F}_2 = (1/\sqrt{r_{1,1}})\mathbf{I}_2$. Then compute $\mathbf{F}_{2m}$ from $\mathbf{F}_{2m-2}$ iteratively by (13), (14) and (12) for $m = 2, 3, \cdots, M$, to obtain $\mathbf{F}_{2M}$. Set $m = M$. |
| **Iterative Detection** | (a) Find the minimum length odd row and even row of $\mathbf{F}_{2m}$, and permute them to the last odd row and even row, respectively. Correspondingly permute $\mathbf{a}_{2m}$, $\mathbf{z}_{2m}$, and rows and columns in $\mathbf{R}_{2m}$.<br>(b) Block-upper triangularize $\mathbf{F}_{2m}$ by (22) to obtain $\mathbf{F}_{2m-2}$.<br>(c) Compute $\hat{a}_{2m-1}$ and $\hat{a}_{2m}$ by (23), which are quantized into $a_{2m-1}$ and $a_{2m}$, respectively.<br>(d) Cancel the effect of $a_{2m-1}$ and $a_{2m}$ in $\mathbf{z}_{2m}$ by (24), to obtain $\mathbf{z}_{2m-2}$. Remove $a_{2m-1}$ and $a_{2m}$ in $\mathbf{a}_{2m}$ to obtain $\mathbf{a}_{2m-2}$. Also remove the last two rows and columns in $\mathbf{R}_{2m}$ to obtain $\mathbf{R}_{2m-2}$.<br>(e) If $m > 1$, go back to step (a) with $\mathbf{F}_{2m-2}$, $\mathbf{R}_{2m-2}$, $\mathbf{z}_{2m-2}$ and $\mathbf{a}_{2m-2}$. Let $m = m - 1$. |

### B. Efficient square-root algorithm for OMP in Sub-Nyquist Sampling System

From (8), (9) and (11)$_{\mathbf{\Gamma}}$ we can deduce

$$\mathbf{s}_{2k} = \mathbf{F}_{\mathbf{\Gamma}^{2k}} \mathbf{F}_{\mathbf{\Gamma}^{2k}}^H \mathbf{T}_{\mathbf{\Gamma}^{2k}}^H \mathbf{y}. \tag{26}$$

In (26), let

$$\mathbf{y'}_{\mathbf{\Gamma}^{2k}} = \mathbf{T}_{\mathbf{\Gamma}^{2k}}^H \mathbf{y}, \tag{27}$$

which satisfies

$$\mathbf{y'}_{\mathbf{\Gamma}^{2k}} = \left[ \begin{array}{ccc} \mathbf{y'}_{\mathbf{\Gamma}^{2k-2}}^T & \mathbf{t}_{\gamma_{2k-1}}^H \mathbf{y} & \mathbf{t}_{\gamma_{2k}}^H \mathbf{y} \end{array} \right]^T \tag{28}$$

where $\mathbf{t}_{\gamma_i}$ is the $i^{th}$ column in $\mathbf{T}_{\mathbf{\Gamma}^{2k}}$. Then we subsitute (27) and (12) into (26), to finally obtain

$$\mathbf{s}_{2k} = \left[\mathbf{s}_{2k-2}^T \; \mathbf{0}_2^T\right]^T + \left[\mathbf{f}_{2k-1} \; \mathbf{f}_{2k}\right]\left[\mathbf{f}_{2k-1} \; \mathbf{f}_{2k}\right]^H \mathbf{y'}_{\mathbf{\Gamma}^{2k}}, \tag{29}$$

where $\mathbf{f}_{2k-1}$ and $\mathbf{f}_{2k}$ are the $(2k-1)^{th}$ and $2k^{th}$ columns of $\mathbf{F}_{\mathbf{\Gamma}^{2k}}$, respectively. Moreover, from (11)$_{\mathbf{\Gamma}}$ we deduce

$$\mathbf{F}_{\mathbf{\Gamma}^2} = \left[ \begin{array}{cc} \frac{1}{\sqrt{r_{1,1}}} & -\frac{r_{1,2}}{r_{1,1}} \sqrt{r_{1,1}/(r_{1,1}^2 - |r_{1,2}|^2)} \\ 0 & \sqrt{r_{1,1}/(r_{1,1}^2 - |r_{1,2}|^2)} \end{array} \right]. \tag{30}$$

To obtain the square-root algorithm for OMP, we only need to use step (2.d') instead of step (2.d) in the OMP Algorithm in [4] (summarized in section II.B). Step (2.d') is as follow:

*if($k = 1$):* compute $\mathbf{y'}_{\mathbf{\Gamma}^2}$, $\mathbf{F}_{\mathbf{\Gamma}^2}$ and $\mathbf{s}_2$ by (27), (30) and (26) (where $\mathbf{T}_{\mathbf{\Gamma}^{2k}}^H \mathbf{y} = \mathbf{y'}_{\mathbf{\Gamma}^2}$), respectively.

*else:* compute $\mathbf{F}_{\mathbf{\Gamma}^{2k}}$ from $\mathbf{F}_{\mathbf{\Gamma}^{2k-2}}$ by (13)$_{\mathbf{\Gamma}}$, (17), (19), (16) and (12)$_{\mathbf{\Gamma}}$ (where (17b) in (17) does not need any computations, as above-mentioned). Compute $\mathbf{y'}_{\mathbf{\Gamma}^{2k}}$ and $\mathbf{s}_{2k}$ by (28) and (29), respectively, from $\mathbf{y'}_{\mathbf{\Gamma}^{2k-2}}$ and $\mathbf{s}_{2k-2}$.

## V. COMPLEXITY EVALUATION

In the proposed inverse Cholesky factorization for Alamouti matrices and that for Alamouti-like matrices, the complexities for the computations of the even columns (i.e. (14), (17a), (19), (16)) can be neglected, with respect to those for the computations of the odd columns (i.e. (13), (13)$_{\mathbf{\Gamma}}$). Thus compared to the inverse Cholesky factorization [8] directly applied to Alamouti or Alamouti-like matrices, the proposed inverse Cholesky factorization can save about half the complexity.

TABLE II

COMPLEXITY COMPARISON BETWEEN THE V-BLAST ALGORITHM IN [8] AND THE PROPOSED G-STBC ALGORITHM

| Step | The V-BLAST Alg [8] | The G-STBC Alg |
|---|---|---|
| Compute R | $(\frac{1}{2}M^2N)$ | $(2M^2N)$ |
| Compute F | $(\frac{1}{3}M^3)$ | $(\frac{4}{3}M^3)$ |
| Givens Rot. | $(\frac{1}{3}M^3, \frac{1}{9}M^3)$ to (0) | $(\frac{8}{9}M^3, \frac{2}{9}M^3)$ to (0) |
| Sum | from $(\frac{2}{3}M^3 + \frac{1}{2}M^2N,$ $\frac{4}{9}M^3 + \frac{1}{2}M^2N)$ to $(\frac{1}{3}M^3 + \frac{1}{2}M^2N)$ | from $(\frac{20}{9}M^3 + 2M^2N,$ $\frac{14}{9}M^3 + 2M^2N)$ to $(\frac{4}{3}M^3 + 2M^2N)$ |

Let ($j$, $k$) denote the complexity of $j$ complex multiplications and $k$ complex additions, and it is simplified to be ($j$) if $j = k$. Table I compares the computational complexity of the proposed square-root algorithm for G-STBC and that of the square-root algorithm for Vertical Bell Laboratories Layered Space-Time architecture (V-BLAST) in [8]. Notice that in the Hermitian Alamouti matrix $\mathbf{R}_{2M}$ and the upper triangular Alamouti matrix $\mathbf{F}_{2m}$, only $\frac{1}{4}$ entries need to be computed. In (25), $\mathbf{\Omega}_{2j,2j-1}^{2k-1}$ and $\mathbf{\Omega}_{2j-1,2j+1}^{2k-1}\mathbf{\Omega}_{2j,2j+2}^{2k}$ only rotates non-zero entries in the first 2j and 2j+2 rows, respectively. The complexities of a complex inner rotation and a real outer rotation are (4,2) and (2,0), respectively. Thus the average complexity of (25) is $\sum_{m=1}^{M}\left(\frac{1}{m}\sum_{k=1}^{m}\left(\sum_{j=k}^{m}(4,2)(2j) + 2\sum_{j=k}^{m-1}(2,0)(2j+2)\right)/2\right) \approx (\frac{8M^3}{9}, \frac{2M^3}{9})$, where the probabilities for $k = 1, 2, \cdots, m$ are assumed to be equal. The complexity of (25) is zero when the assumed detection order is the same as the actual order [8].

When the square-root V-BLAST algorithm [8] is directly applied to the equivalent channel model (1) that can be regarded as a $2M \times 2N$ V-BLAST system, the required complexity is from $(\frac{16}{3}M^3 + 4M^2N, \frac{32}{9}M^3 + 4M^2N)$ to $(\frac{8}{3}M^3 + 4M^2N)$. Thus when $M = N$, the average speedup in floating-point operations (flops) of the proposed square-root G-STBC algorithm over the square-root V-BLAST algorithm [8] directly applied to (1) ranges from 2.19 to 2. On the other hand, the complexity of the MMSE sub-optimal ordered successive interference cancelation (OSIC) detector for G-STBC using sorted QR decomposition (SQRD) is $(8M^3 + 4M^2N)$

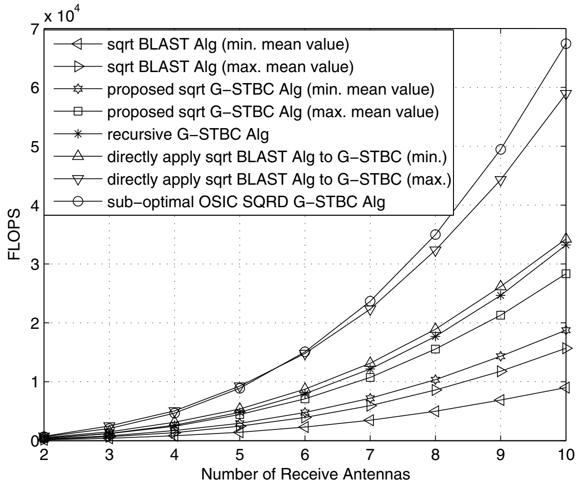

Fig. 1. Comparison of computational complexity among the sub-optimal OSIC G-STBC algorithm in [11], the OSIC G-STBC algorithm obtained by directly applying the square-root V-BLAST algorithm [8], the proposed square-root OSIC G-STBC algorithm, the recursive G-STBC algorithm [2], and the square-root V-BLAST algorithm [8].

[11, equation (19)]. When $M = N$, the average speedup (in flops) of the proposed OSIC G-STBC detector over the sub-optimal OSIC G-STBC detector [11] ranges from 2.96 to 3.6.

Assuming $N = M$, we carried out numerical experiments to count the average flops per time slot of the presented G-STBC or V-BLAST algorithms, for different $N$, the number of receive antennas. The considered G-STBC system includes $M$ Alamouti sub-streams and $2M$ transmitters, while the considered V-BLAST system includes $M$ sub-streams and $M$ transmitters. All results are shown in Fig. 1. It can be seen that they are consistent with the theoretical flops calculation.

On the other hand, in the sub-Nyquist sampling system [4], we simulate the flops of the proposed square-root OMP algorithm, the original OMP algorithm using direct matrix inversion [5], the original OMP algorithm with direct matrix inversion replaced by the inverse Cholesky factorization in [8], the fast OMP algorithm based on QR decomposition [5], [6], and the fast OMP algorithm by matrix inverse update [6], [7]. Our simulations do not include the flops of the projection step (i.e. step (2.a)) since they are the same for all the presented OMP algorithms. As the simulations for Fig. 5a in [12], we set $K = \sqrt{N}/2$, $M = K(\log_2 N - 2)$, and give flops for $N = 2^i$ $(i = 4, 5, \cdots, 10)$. The results are shown in Fig. 2. Our simulations also show that the flops for the proposed square-root OMP algorithm are only about $30\%$ of those for the fast OMP algorithm by matrix inverse update [6], [7].

## VI. Conclusion

When applying the efficient inverse Cholesky factorization [8] to Alamouti matrices in G-STBC and Alamouti-like matrices in OMP for the sub-Nyquist sampling system [4], we utilize some good properties of Alamouti or Alamouti-like matrices to save about half the complexity. Then we propose the whole square-root algorithms for G-STBC and OMP in [4], respectively. The proposed square-root G-STBC algorithm has an average speedup of $2 \sim 2.19$ with respect to the square-root V-BLAST algorithm [8] directly applied to the equivalent channel model for G-STBC, and has an average speedup of $2.96 \sim 3.6$ with respect to the sub-optimal OSIC G-STBC detector in [11]. On the other hand, when comparing the complexities of all the steps except the projection step, the complexity for the proposed square-root OMP algorithm is about $30\%$ of that for the fast OMP algorithm by matrix inverse update [6], [7].

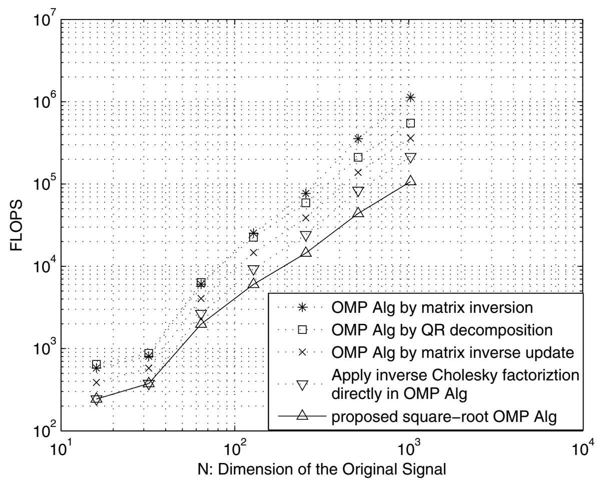

Fig. 2. Comparison of flops among the presented OMP algorithms.

## References

[1] J. Eilert, D. Wu, D. Liu, D. Wang, N. Al-Dhahir and H. Minn, "Complexity reduction of matrix manipulation for multi-user STBC-MIMO decoding", *IEEE Sarnoff Symposium*, April 30 - May 2 2007.

[2] H. Zhu, W. Chen, B. Li and F. Gao, "A Fast Recursive Algorithm for G-STBC", *IEEE Trans. on Communications*, vol. 59, no. 8, 2011.

[3] A. H. Sayed, W. M., Younis and A. Tarighat, "An invariant matrix structure in multiantenna communications", *Signal Processing Letters, IEEE*, vol. 12, Nov. 2005, pp. 749-752.

[4] M. Mishali and Y. C. Eldar, "From Theory to Practice: Sub-Nyquist Sampling of Sparse Wideband Analog Signals", *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 375C391, Apr. 2010.

[5] Joel Tropp and Anna Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit", *IEEE Trans. on Information Theory*, vol. 53, no. 12, pp. 4655-4666, December 2007.

[6] Thomas Blumensath and Mike E. Davies, "In Greedy Pursuit of New Directions: (Nearly) Orthogonal Matching Pursuit by Directional Optimisation", *EUSIPCO 2007*, Poznan, Poland, 2007.

[7] Y. Fang, L. Chen, J. Wu and B. Huang, "GPU Implementation of Orthogonal Matching Pursuit for Compressive Sensing", *ICPADS 2011.*

[8] H. Zhu, W. Chen, B. Li and F. Gao, "An Improved Square-root Algorithm for V-BLAST Based on Efficient Inverse Cholesky Factorization", *IEEE Trans. on Wireless Communications*, vol. 10, no. 1, pp. 43-48, 2011.

[9] D. Donoho, "Compressed sensing", *IEEE Trans. on Information Theory*, vol. 52, no. 4, pp. 1289 - 1306, April 2006.

[10] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd edition, Johns Hopkins University Press, 1996.

[11] M. Gomaa and A. Ghrayeb, "A Low Complexity MMSE Detector for Multiuser Layered Space-Time Coded MIMO Systems", *CCECE 2007*, 22-26 Apr. 2007.

[12] S. Kunis and H. Rauhut, "Random sampling of sparse trigonometric polynomials II - Orthogonal matching pursuit versus basis pursuit", Preprint, 2006. Available online at http://dsp.rice.edu/cs.