

A Practical and Efficient Algorithm for Distributed Interference Alignment Based on Cognitive Radio

Hufei Zhu, Bin Li (*Member, IEEE*) ,

Dept. of Corporate Research,
Huawei Technologies Co., Ltd.,
Shenzhen, P.R. China.

Wen Chen (*Member, IEEE*) ,

Dept. of Electronic Engineering,
Shanghai Jiao Tong Univ.,
Shanghai, P.R.China.

Abstract—We propose a practical and efficient minor subspace tracking (MST) algorithm to implement Distributed Interference Alignment (IA) based on Cognitive Radio (CR) that employs Interference Subspace Tracking (IST). The proposed MST algorithm utilizes the parallelizable matrix-vector products instead of the back and forward substitution in the existing MST algorithm for IST-IA, which is an inherent serial process unsuitable for the parallel implementation. Thus the proposed MST algorithm is more suitable for the practical parallel implementation than the existing MST algorithm for IST-IA. The proposed MST algorithm also requires less computational complexity than the existing MST algorithm for IST-IA.

Keywords-Interference Alignment (IA); Distributed IA; Cognitive Radio (CR); Interference Subspace Tracking (IST); minor subspace tracking (MST)

I. INTRODUCTION

The interference ultimately limits the performance of many wireless systems [1]. Recently, the idea of interference alignment (IA) was proposed for wireless networks [2]. Theoretically IA can achieve the capacity that is much higher than previously believed [2]. It designs interfering signals from different sources such that they “cast overlapping shadows” on the signal subspace of interfered receivers. Then part of the signal space at the receiver is free from interference [2].

IA schemes in [2] require global channel knowledge at all the transmitters, which can be an overwhelming overhead in practice. To alleviate this problem, a Distributed IA scheme taking a cognitive radio (CR) approach was proposed in [3], where each transmitter only requires local channel knowledge and the covariance matrix of its effective noise (consisting of the AWGN and the interference). In the Distributed IA scheme, each transmitter adopts an unselfish approach to minimize the interference to unintended receivers, which is the defining feature of CR [4]. This CR approach then leads to interference alignment [3]. In CR systems, it is a key challenge for a given transmitter to estimate the strength of interference caused at unintended receivers. To meet this challenge, the CR-based Distributed IA scheme utilizes the reciprocity of the two-way communication channel in Time Division Duplex (TDD) networks: the signal subspace with the least interference for a given receiver also causes the least interference to other users during the reciprocal network transmission.

In [5], the Interference Subspace Tracking IA (IST-IA) algorithm was proposed, as a practical implementation of

Distributed IA [3] in a cellular communication system. In the IST-IA algorithm, each transmitter requires neither local channel knowledge nor the effective noise covariance matrix, and only utilizes a few received signal vectors to track the beamforming matrices. Compared to the Distributed IA algorithm [3], the IST-IA algorithm has a similar performance, and requires less system overhead and complexity.

This paper is organized as follows. Section II overviews the system model. Section III describes the existing Distributed IA algorithms, including the IST-IA algorithm. A practical and efficient MST algorithm for IST-IA is proposed in Section IV. Then the implementation complexity is analyzed in Section V. Finally, we make conclusion in Section VI.

In the following sections, $(\bullet)^T$ and $(\bullet)^H$ denote matrix transposition and matrix conjugate transposition, respectively.

II. SYSTEM MODEL

As in [5], let us consider a TDD cellular network with K Base Stations (BSs) and K User Equipments (UEs). All the UEs and BSs are equipped with M and N antennas, respectively. For the uplink transmission (from UEs to BSs), the $N \times 1$ received signal vector $\mathbf{y}^{[k]}$ at BS k ($k=1, 2, \dots, K$) is

$$\mathbf{y}^{[k]} = \sum_{i=1}^K g_{ki} \mathbf{H}^{[ki]} \mathbf{x}^{[i]} + \mathbf{w}^{[k]}, \quad (1)$$

where g_{ki} is the distance dependent path-loss gain between UE i and BS k; the $N \times M$ matrix $\mathbf{H}^{[ki]}$ consists of fading channel coefficients between UE i and BS k that are independent and identically distributed block Rayleigh fading, i.e., the coefficients are assumed to be constant during training and data transmission phases; $\mathbf{x}^{[i]}$ is the $M \times 1$ transmit signal vector by UE i with transmit power $E \|\mathbf{x}^{[i]}\|^2 = P$; $\mathbf{w}^{[k]}$ is the $N \times 1$ AWGN noise vector with zero mean and unit variance. For the downlink transmission (from BSs to UEs), the $M \times 1$ received signal vector $\mathbf{z}^{[i]}$ at UE i ($i=1, 2, \dots, K$) is

$$\mathbf{z}^{[i]} = \sum_{k=1}^K g_{ki} (\mathbf{H}^{[ki]})^T \bar{\mathbf{x}}^{[k]} + \bar{\mathbf{w}}^{[i]}, \quad (2)$$

where $\bar{\mathbf{x}}^{[k]}$ and $\bar{\mathbf{w}}^{[i]}$ are the $N \times 1$ transmit signal vector by BS k and the $M \times 1$ AWGN noise vector, respectively.

BS k only serves UE k in its own cell, for $k=1, 2, \dots, K$. Correspondingly in the uplink, BS k receives inter-cell interference from all $K-1$ UEs in other cells, while in the downlink, UE k receives inter-cell interference from all $K-1$ non-serving BSs. As in [5], we focus on uplink transmission in what follows, but the algorithm also applies to downlink.

III. EXISTING DISTRIBUTED IA ALGORITHMS

In this section, we review the Distributed IA algorithm in [3] and the IST-IA algorithm in [5].

A. Distributed IA

The Distributed IA scheme [3] is proposed for the TDD interference network. Unlike the prior “selfish” transmission strategy where each transmitter tries to maximize its own transmission signal-to-interference-and-noise ratio (SINR) at the desired receiver [2], the CR-based Distributed IA scheme follows an “unselfish” (i.e. “altruistic”) transmission strategy where each transmitter primarily tries to minimize the interference to other concurrent transmissions. By network duality [6], this “unselfish” strategy only requires the transmitter to find the subspace over which it observes the minimal interference when as a receiver.

The Distributed IA algorithm starts with arbitrary transmit and receive filters $\mathbf{V}^{[k]}$, $\mathbf{U}^{[k]}$, and iteratively update these filters to approach interference alignment. Assume that it works in the system model of Section II. Then in each iteration, there are uplink and downlink transmissions. During the first uplink transmission, UE k ($k = 1, 2, \dots, K$) at the transmitter side precodes the $D \times 1$ symbol vector $\hat{\mathbf{x}}^{[k]} = [\hat{x}_1^{[k]} \ \hat{x}_2^{[k]} \ \dots \ \hat{x}_D^{[k]}]$ by an arbitrary $M \times D$ precoding matrix $\mathbf{V}^{[k]}$, i.e., in (1) we have

$$\mathbf{x}^{[k]} = \mathbf{V}^{[k]} \hat{\mathbf{x}}^{[k]}. \quad (3)$$

At the receiver side, BS k applies an $N \times D$ interference suppression matrix $\mathbf{U}^{[k]}$ to obtain

$$\hat{\mathbf{y}}^{[k]} = (\mathbf{U}^{[k]})^H \mathbf{y}^{[k]}. \quad (4)$$

In (4), $\mathbf{U}^{[k]}$ is chosen to make the signal $\hat{\mathbf{y}}^{[k]}$ lie in the D -dimensional subspace with the minimal interference. Correspondingly BS k needs to know the interference covariance matrix (including the AWGN), i.e.,

$$\mathbf{Q}^{[k]} = \sum_{i=1, i \neq k}^K \frac{P}{D} \mathbf{H}^{[ki]} \mathbf{V}^{[i]} \mathbf{V}^{[i]H} \mathbf{H}^{[ki]H} + \mathbf{I}_N. \quad (5)$$

Then $\mathbf{U}^{[k]}$ is chosen as the D eigenvectors corresponding to the smallest D eigenvalues of $\mathbf{Q}^{[k]}$.

In the downlink transmission, the transmitters and receivers switch roles, while the steps are similar. Firstly the transmitters (BSs) set the precoding matrices

$$\bar{\mathbf{V}}^{[k]} = \mathbf{U}^{[k]} \quad (6)$$

for $k=1, 2, \dots, K$, where $\mathbf{U}^{[k]}$ is calculated in the uplink phase. The receivers (UEs) obtain the interference covariance matrix

$\bar{\mathbf{Q}}^{[k]}$. Then the interference suppression matrix $\bar{\mathbf{U}}^{[k]}$ is updated with the D eigenvectors corresponding to the smallest D eigenvalues of $\bar{\mathbf{Q}}^{[k]}$. Correspondingly the precoding matrices for the next uplink phase are updated by $\mathbf{V}^{[k]} = \bar{\mathbf{U}}^{[k]}$. (7)

The algorithm iterates uplink and downlink transmissions for T times. In each of the T iterations, it updates $\mathbf{V}^{[k]} = \bar{\mathbf{U}}^{[k]}$ and $\mathbf{U}^{[k]} = \bar{\mathbf{V}}^{[k]}$, which include the uplink and downlink precoding matrices and the uplink and downlink interference suppression matrices. The total interference in the system is monotonically reduced after each iteration, and the convergence of the Distributed IA algorithm can be proved [3].

B. IST-IA

Similarly to [3], the IST-IA algorithm [5] requires a training phase consisting of T iterations, to update the precoding and interference suppression matrices to approach interference alignment. However, IST-IA does not need a priori knowledge of the interference covariance matrix (5). For each uplink or downlink iteration, IST-IA spends KL training symbol periods to estimate the interference suppression matrices and obtain the precoding matrices accordingly.

In the uplink transmission of the first iteration, the $D \times 1$ training symbol vector $\hat{\mathbf{x}}^{[k]}$ is precoded by an arbitrary precoding matrix $\mathbf{V}^{[k]}$, as shown in (3). Then $\mathbf{x}^{[k]} = \mathbf{V}^{[k]} \hat{\mathbf{x}}^{[k]}$ is sent by UE k. These symbols are used by BSs to update the interference suppression matrix $\mathbf{U}^{[k]}$. To train BS k, the UEs in all other cells transmit L symbols, while the direct transmission from UE k is silenced. From (1), it can be seen that the l -th ($l = 1, \dots, L$) symbol vector received by BS k is

$$\mathbf{y}^{[k]}(l) = \sum_{i=1, i \neq k}^K g_{ki} \mathbf{H}^{[ki]}(l) \mathbf{x}^{[i]}(l) + \mathbf{w}^{[k]}(l) \quad (8)$$

The network schedules the transmissions from UE 2, UE 3, \dots , UE K to BS 1 firstly to train BS 1; then it schedules the transmissions from UE 1, UE 3, \dots , UE K to BS 2 to train BS 2. This procedure continues until each BS receives L training symbol vectors from the UEs in all other cells.

A simple approach to update the interference suppression matrix $\mathbf{U}^{[k]}$ is to estimate the covariance matrix (5) from the L received training vector and then obtain $\mathbf{U}^{[k]}$ via eigenvector decomposition (EVD), which is called as EVD-IA [5]. The IST-IA was proposed in [5] as an alternative solution, which is based on the minor subspace tracking (MST) algorithm [7]. Compared to EVD-IA, IST-IA significantly reduces the computational complexity, with minor performance degradation [5]. Moreover, when the channel is time varying during data transmission, IST-IA could be easily deployed to track the interference suppression matrices [5].

Downlink transmissions begin after the BSs have updated their interference suppression matrices $\mathbf{U}^{[k]}$ via MST. Each BS precodes D independent symbols with the precoding matrix

$\bar{\mathbf{V}}^{[k]} = \mathbf{U}^{[k]}$, i.e. (6). Similar to the uplink, each UE receives L training symbol vectors and then compute the interference suppression matrix $\bar{\mathbf{U}}^{[k]}$ via MST. The precoding matrices for the next uplink phase are updated by $\mathbf{V}^{[k]} = \bar{\mathbf{U}}^{[k]}$, i.e. (7). The training period continues for T training iterations.

The IST-IA algorithm [5] is summarized as follows.

The IST-IA algorithm

Training Period:

Initialize $\mathbf{U}^{[k]}$, $\mathbf{V}^{[k]}$ ($k = 1, \dots, K$): random matrix with orthonormal vectors;

for $t = 1$ to T (T training iterations) **do**

 Uplink: UE \rightarrow BS

for $k = 1$ to K (BS k) **do**

for $l = 1$ to L **do**

for $i = 1$ to K , $i \neq k$ (UE i) **do**

 generate training vector $\mathbf{x}^{[i]}(l) = \mathbf{V}^{[k]} \hat{\mathbf{x}}^{[i]}(l)$

end for

 BS k receives concurrent interference symbol vectors transmitted from the $K - 1$ UEs (i.e. UE i , $i = 1$ to K , $i \neq k$), as shown in (8).

end for

 Find $\mathbf{U}_t^{[k]}(L)$ by the MST Algorithm in the end of this subsection.

end for

 Downlink: BS \rightarrow UE, dual to the uplink part: obtain $\mathbf{V}_t^{[k]}(L)$.

Update $\mathbf{U}^{[k]} = \mathbf{U}_t^{[k]}(L)$, $\mathbf{V}^{[k]} = \mathbf{V}_t^{[k]}(L)$.

end for

DATA Transmission Period.

The square root QR inverse iteration MST algorithm in [7] provides a good tradeoff between complexity and performance. Thus the IST-IA algorithm applies the efficient MST algorithm in [7], to track the eigenvectors corresponding to the smallest D eigenvalues of the empirical covariance matrix [5].

The efficient MST algorithm in [5] is summarized as follows.

- **Initialization** (for the 1st iteration): let $\mathbf{U}^{[k]}(0)$ be a random matrix with orthonormal vectors; $\mathbf{R}_x^{[k]}(0) = \Delta \cdot \mathbf{I}_N$, where Δ is a small positive real number.

- **Recursion:**

for $l = 1$ to L **do**

Input: $\mathbf{y}^{[k]}(l)$

 Find an upper-right triangular $\mathbf{R}_x^{[k]}(l)$ such that

$$\begin{bmatrix} 0 \dots 0 \\ \mathbf{R}_x^{[k]}(l) \end{bmatrix} = \mathbf{G}(l) \begin{bmatrix} (1-\beta)^{1/2} (\mathbf{y}^{[k]}(l))^H \\ \beta^{1/2} \mathbf{R}_x^{[k]}(l-1) \end{bmatrix}, \quad (9)$$

where the exponential forgetting factor [7] β is a real number, and $\mathbf{G}(l)$ is a $(N + 1) \times (N + 1)$ unitary matrix.

Find $\mathbf{B}^{[k]}(l)$ such that

$$(\mathbf{R}_x^{[k]}(l))^H \mathbf{B}^{[k]}(l) = \mathbf{U}^{[k]}(l-1). \quad (10)$$

Find $\mathbf{A}^{[k]}(l)$ such that

$$\mathbf{R}_x^{[k]}(l) \mathbf{A}^{[k]}(l) = \mathbf{B}^{[k]}(l). \quad (11)$$

QR decompose $\mathbf{A}^{[k]}(l)$ into

$$\mathbf{A}^{[k]}(l) = \mathbf{U}^{[k]}(l) \tilde{\mathbf{R}}^{[k]}(l). \quad (12)$$

end for

Output: $\mathbf{U}^{[k]}(L)$. Moreover, initialize $\mathbf{U}^{[k]}(0) = \mathbf{U}^{[k]}(L)$ and $\mathbf{R}_x^{[k]}(0) = \mathbf{R}_x^{[k]}(L)$ for the following tth ($2 \leq t \leq T$) iteration.

IV. A PRACTICAL AND EFFICIENT MST ALGORITHM FOR IST-IA

The efficient MST algorithm utilized in [5] requires the back and forward substitutions [7], [8] to find solutions for the triangular systems (10) and (11), respectively. However, the back or forward substitution is an inherent serial process that is unsuitable for the parallel implementation [9]. In this section, we deduce a practical and efficient MST algorithm for IST-IA. To some extent, the deduction is inspired by that in [10] to obtain the recursive V-BLAST algorithm.

Let us substitute (11) into (10) to obtain

$$(\mathbf{R}_x^{[k]}(l))^H \mathbf{R}_x^{[k]}(l) \mathbf{A}^{[k]}(l) = \mathbf{U}^{[k]}(l-1). \quad (13)$$

Define

$$\Phi_x^{[k]}(l) = (\mathbf{R}_x^{[k]}(l))^H \mathbf{R}_x^{[k]}(l), \quad (14)$$

which is then substituted into (13) to obtain

$$\Phi_x^{[k]}(l) \mathbf{A}^{[k]}(l) = \mathbf{U}^{[k]}(l-1). \quad (15)$$

Also define

$$\mathbf{W}_x^{[k]}(l) = (\Phi_x^{[k]}(l))^{-1}, \quad (16)$$

and correspondingly represent (15) as

$$\mathbf{A}^{[k]}(l) = \mathbf{W}_x^{[k]}(l) \mathbf{U}^{[k]}(l-1). \quad (17)$$

Now instead of the back and forward substitutions in (10) and (11) to find $\mathbf{A}^{[k]}(l)$, we utilize (17) to compute $\mathbf{A}^{[k]}(l)$

directly by the parallelizable matrix-vector products [9]. We need to find a method to update $\mathbf{W}_x^{[k]}(l)$ for $l=0, 1, 2, \dots, L$.

From (14) and (9), we can deduce

$$\begin{aligned} \Phi_x^{[k]}(l) &= (\mathbf{R}_x^{[k]}(l))^H \mathbf{R}_x^{[k]}(l) = \begin{bmatrix} 0 \cdots 0 \\ \mathbf{R}_x^{[k]}(l) \end{bmatrix}^H \begin{bmatrix} 0 \cdots 0 \\ \mathbf{R}_x^{[k]}(l) \end{bmatrix} = \quad , \text{ i.e.,} \\ & \left[(1-\beta)^{1/2} (\mathbf{y}^{[k]}(l))^H \right]^H \mathbf{G}(l)^H \mathbf{G}(l) \begin{bmatrix} (1-\beta)^{1/2} (\mathbf{y}^{[k]}(l))^H \\ \beta^{1/2} \mathbf{R}_x^{[k]}(l-1) \end{bmatrix} \\ \Phi_x^{[k]}(l) &= \beta \Phi_x^{[k]}(l-1) + (1-\beta) \mathbf{y}^{[k]}(l) \mathbf{y}^{[k]}(l)^H, \end{aligned} \quad (18)$$

where the relation of (14) is utilized. From (18), we can utilize the matrix inverse lemma [11, p.30] to obtain

$$\begin{aligned} \Phi_x^{[k]}(l)^{-1} &= (\beta \Phi_x^{[k]}(l-1) + \mathbf{y}^{[k]}(l)(1-\beta) \mathbf{y}^{[k]}(l)^H)^{-1} = \quad , \text{ i.e.,} \\ (\beta \Phi_x^{[k]}(l-1))^{-1} &- \frac{(\beta \Phi_x^{[k]}(l-1))^{-1} \mathbf{y}^{[k]}(l) \mathbf{y}^{[k]}(l)^H (\beta \Phi_x^{[k]}(l-1))^{-1}}{(1-\beta)^{-1} + (\mathbf{y}^{[k]}(l))^H (\beta \Phi_x^{[k]}(l-1))^{-1} \mathbf{y}^{[k]}(l)} \\ \mathbf{W}_x^{[k]}(l) &= \frac{\mathbf{W}_x^{[k]}(l-1)}{\beta} - \frac{\mathbf{W}_x^{[k]}(l-1) \mathbf{y}^{[k]}(l) \mathbf{y}^{[k]}(l)^H \mathbf{W}_x^{[k]}(l-1)}{\beta^2 / (1-\beta) + \beta \mathbf{y}^{[k]}(l)^H \mathbf{W}_x^{[k]}(l-1) \mathbf{y}^{[k]}(l)}. \end{aligned} \quad (19)$$

From (14) and (16), it is easy to deduce $\mathbf{W}_x^{[k]}(l)^H = \mathbf{W}_x^{[k]}(l)$. Thus firstly we can compute

$$\underline{\mathbf{y}}_w^{[k]}(l) = \mathbf{W}_x^{[k]}(l-1) \mathbf{y}^{[k]}(l), \quad (20)$$

which is then substituted into (19) to obtain

$$\mathbf{W}_x^{[k]}(l) = \frac{\mathbf{W}_x^{[k]}(l-1)}{\beta} - \frac{\underline{\mathbf{y}}_w^{[k]}(l) \underline{\mathbf{y}}_w^{[k]}(l)^H}{\beta^2 / (1-\beta) + \beta \underline{\mathbf{y}}_w^{[k]}(l)^H \underline{\mathbf{y}}_w^{[k]}(l)}. \quad (21)$$

Now we modify the efficient MST algorithm utilized in [5] into the following practical and efficient MST algorithm.

- Initialization** (for the 1st iteration): let $\mathbf{U}^{[k]}(0)$ be a random matrix with orthonormal vectors; $\mathbf{W}_x^{[k]}(0) = (\mathbf{R}_x^{[k]}(0)^H \mathbf{R}_x^{[k]}(0))^{-1} = (1/\Delta^2) \cdot \mathbf{I}_N$, where Δ is a small positive real number.
- Recursion:**

for $l = 1$ to L **do**

Input: $\mathbf{y}^{[k]}(l)$. **Compute** $\mathbf{W}_x^{[k]}(l)$ from $\mathbf{W}_x^{[k]}(l-1)$ and $\mathbf{y}^{[k]}(l)$ by (20) and (21). **Compute** $\mathbf{A}^{[k]}(l) = \mathbf{W}_x^{[k]}(l) \mathbf{U}^{[k]}(l-1)$, i.e. (17). **QR decompose** $\mathbf{A}^{[k]}(l)$ into $\mathbf{A}^{[k]}(l) = \mathbf{U}^{[k]}(l) \tilde{\mathbf{R}}^{[k]}(l)$, i.e. (12).

end for

Output: $\mathbf{U}^{[k]}(L)$. Moreover, initialize

$\mathbf{U}^{[k]}(0) = \mathbf{U}^{[k]}(L)$ and $\mathbf{W}_x^{[k]}(0) = \mathbf{W}_x^{[k]}(L)$ for

the following tth ($2 \leq t \leq T$) iteration.

V. ANALYSIS OF IMPLEMENTATION COMPLEXITY

Let us compare the implementation complexity of the proposed MST algorithm with that of the MST algorithm in [5]. The proposed MST algorithm employs (20), (21) and (17), which only consist of the parallelizable matrix-vector products [9]. As a comparison, the MST algorithm in [5] employs the corresponding (9), (10) and (11). The back substitution to solute (10) and the forward substitution to solute (11) [7], [8] are both inherent serial processes that are unsuitable for the parallel implementation [9]. Thus it can be seen that the proposed MST algorithm is more suitable for the practical parallel implementation than the MST algorithm in [5]. On the other hand, it can be seen that the proposed MST algorithm requires less complexity than the MST algorithm in [5].

VI. CONCLUSION

We propose a practical and efficient MST algorithm to implement Distributed IA based on CR that employs IST. The proposed MST algorithm utilizes the parallelizable matrix-vector products [9] instead of the back and forward substitution in the MST algorithm in [5], which is an inherent serial process unsuitable for the parallel implementation [9]. Thus the proposed MST algorithm is more suitable for the practical parallel implementation than the MST algorithm utilized in [5]. On the other hand, the proposed MST algorithm requires less computational complexity than the MST algorithm in [5].

ACKNOWLEDGMENT

This work is supported by Major National S&T Program 2009ZX03003-002 of China.

REFERENCES

- [1] David Tse, "Breaking the Interference Barrier, Mobicom/Mobihoc Plenary", Mobicom/Mobihoc Plenary Talk, September 13, 2007.
- [2] V.R. Cadambe, S.A. Jafar, "Interference Alignment and Degrees of Freedom of the K-User Interference Channel," IEEE Transactions on Information Theory, vol 54, pp. 3425 – 3441, Aug. 2008.
- [3] K. Gomadam, V.R. Cadambe, S.A. Jafar, "Approaching the Capacity of Wireless Networks through Distributed Interference Alignment," IEEE Globecom 2008.
- [4] S. Srinivasa and S. Jafar, "The throughput potential of cognitive radio - a theoretical perspective," IEEE Communications Magazine, vol. 45, 2007.
- [5] B. Niu, A. M. Haimovich, "Interference Subspace Tracking for Network Interference Alignment in Cellular Systems," IEEE Globecom 2009.
- [6] B. Song, R. Cruz and B. Rao, "Network duality and its application to multi-user mimo wireless networks with sinr constraints," IEEE ICC 2005, vol. 4, pp. 2684–2689, 16-20 May 2005.
- [7] P. Strobach, "Square-root QR inverse iteration for tracking the minor subspace," IEEE Trans. Signal Processing, vol. 48, no. 11, Nov. 2000.
- [8] G. H. Golub and C. F. Van Loan, Matrix Computations, 3rd ed., Baltimore, MD, Johns Hopkins Univ. Press, 1996.
- [9] E. J. Baranowski, "Triangular factorization of inverse data covariance matrices", International Conference on Acoustics, Speech, and Signal Processing, 1991 (ICASSP-91), 14-17 Apr 1991, pp. 2245 - 2247, vol.3.
- [10] H. Zhu, W. Chen, and F. She, "Improved Fast Recursive Algorithms for V-BLAST and G-STBC with Novel Efficient Matrix Inversion," IEEE ICC 2009.
- [11] H. Lütkepohl, Handbook of Matrices. NY John Wiley & Sons, 1996.