International Journal of Engineering Research and Technology.
ISSN 0974-3154 Volume 11, Number 8 (2018), pp. 1199-1212

http://www.irphouse.com

UPDATING THE LINEAR DECODER FOR THE UPLINK IN MIMO SYSTEM USING SVD ALGORITHM IN LINEAR DETECTION

Soraya Norma Mustika[1]
[1] Electrical Engineering,University of Brawijaya, VeteranKetawanggede, Kota Malang, Indonesia, 65145

Huang-Wan Jen[2]
[2] Institute Communication Engineering, National Sun Yat-sen University, No. 70, Lianhai Rd., Kaohsiung, 804, Taiwan

Rini Nurhasanah[3]
[3] Electrical Engineering,University of Brawijaya, VeteranKetawanggede, Kota Malang, Indonesia, 65145

Wijono[4]
[4] Electrical Engineering,University of Brawijaya, VeteranKetawanggede, Kota Malang, Indonesia, 65145

ABSTRACT:

MIMO (multi input multi output) has its limitation, because of that many researcher have developed a massive MIMO system. In linear detector massive MIMO system one of the main problem is placed on Uplink in MIMO system, because Massive MIMO system have to deal with huge matrix inversion to find approximate transmitted data. Because of this, we are trying to find new way to approximate transmitted data while reducing the complexity in the inverse huge matrix. In this paper, we are using SVD (Singular Value Decomposition) for two common linear detection which is Zero Forcing and Minimum Mean-Square Error (MMSE). To fulfil that, SVD algorithm when updating one user join or leave the base station, Given Rotation and GolubReinsch algorithm applied. If we need to minimalize the error, we have to redundant the reinsch algorithm. The trade-off of redundant the Reinsch algorithm is the complexityis higher. But still, the complexity is less than Neumann series and the exact inversion.

Keywords - GolubReinsch Algorithm, Given Rotation, SVD, MIMO, Updating.

I. INTRODUCTION

Massive multi-input multi-output (MIMO) systems have been extensively developed due to high spectrum efficiency, robust link reliability and extended coverage, compared with the conventional MIMO systems[13-16]. In the uplink of massive MIMO systems, linear detectors such as Zero Forcing (ZF) and LMMSE(linear minimum mean-square error) detectors are near optimal [1]-[6]. Although the linear detectors is much simpler than the optimal maximum likelihood (ML) detector, the computational complexity is dramatically increased with the dimensionality of the massive MIMO system. To reduce the complexity, some works focus on the numerical method to approximate the LMMSE detector [1,2,6] or ZF detector [3-5]. Specifically, the authors adopt the Neumann Series (NS) to approximate the matrix inversion. However, the numerical errors of the NS approximation is largewith lower order.Other numerical methods such as Gauss Seidel method [2] and the Newton iteration method [6] have been proposed to reduce the numerical errors. Nevertheless, the reduced errors still dominate the detection error performance when SNR is high.

Instead tso approximating the linear detectors numerically, we would like to obtain the linear detectors through the singular value decomposition (SVD) of the channel matrix. Specifically, we adopt power iterative method [X] to initialize the SVD of each channel matrix. Under the assumption of slow fading environment, the channel matrix may be changed slightly when one user joins or leaves the cell, or when the channel condition of one user changes. In this case, re-calculating the SVD of the new channel matrix is not efficient. In this work, we will propose two update algorithms to update the SVD of the channel matrix for the cases when one user joins or leaves, respectively. Specifically, when one user joins or leaves the cell, we first adopt the Gram-Schmidt procedure or Gives rotation to update the decomposition of the new channel matrix as $\mathbf{H}=\mathbf{URV}^H$, where the matrices **U** and **V** are semi-unitary, while the matrix **R** is upper triangular. To obtain the SVD of the new channel matrix, we first adopt the Householder transformation [X] to reduce the matrix **R** as a bi-diagonal matrix while keeping the matrices **U** and **V** semi-unitary. Then, we take turs to apply the Givens rotation and the Golub Reinsch method to reduce the off-diagonal terms recursively until the matrix **R** can be approximated as a diagonal matrix. Note that the SVD of the updated channel matrix is also helpful for the BS to design the precoder for the downlink transmission.

Throughout the remainder of this letter, vectors and matrices are denoted by boldface lower and upper symbols, respectively. In addition, the notations $\mathbf{X}^{-1}$ and $\mathbf{X}^{-H}$ represent the inverse and Hermitian transpose of matrix **X**, respectively. The notation $[\mathbf{X}]_{i:j,m:n}$ stands for a sub matrix of the matrix **X** by extracting **X** from the *i*-th row to the *j*-th row and from the *m*-th column to the *n*-th column. Finally, $\|\mathbf{x}\|_2$ and $[\mathbf{x}]_i$ denote the Euclidean norm and *i*-th entry of vector **x.**

II. SYSTEM MODEL

Consider a downlink system where the base station (BS) receives signals sent from K users. It is assumed that the BS is equipped with M antennas and each user has one antenna due to cost consideration. Denote $\mathbf{x} \in \mathbb{C}^{K\times 1}$ as the signal transmitted by K users with $\mathrm{E}[\mathbf{x}] = \mathbf{0}$ and $\mathrm{E}[\mathbf{x}\mathbf{x}^{\boldsymbol{H}}] = E_x\mathbf{I}$. The signal received at the BS is then given by

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}, \tag{1}$$

where is an $M \times K$ channel matrix $(M \geq K)$, and $\mathbf{n} \in \mathbb{C}^{M\times 1}$ is an white Gaussian noise vector with covariance matrix $\sigma^2\mathbf{I}$. It is assumed that the communication is fully scattered and suffers Rayleigh fading, i.e., the entries of $\mathbf{H}$ are i.i.d. Gaussian with $CN(0,1)$.

III. PROPOSED ALGORITHM

3.1. SVD-based Linear Detection

The optimal detection of the users' signal is maximum likelihood detection. However, the computational complexity is increased exponentially with the number of users. Luckily, linear detections are near optimal when the number of received antennas is sufficiently large. Given the received signal in (1), the linear detector of$\mathbf{x}$ can be expressed by

$$\hat{\mathbf{x}} = (\mathbf{H}^H\mathbf{H} + \rho\mathbf{I})^{-1}\mathbf{H}^H\mathbf{y}, \tag{2}$$

Where $\rho = 0$ for ZF detection, and for LMMSE detection. However, the computational complexity of the matrix inversion in (2) is large for a massive MIMO systems. In this work, we employ the SVD of the channel matrix to reduce the required complexity. Specifically, denote the SVD of the channel matrix as

$$\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H, \tag{3}$$

Where $\mathbf{U}$ is an $M \times K$ semi-unitary matrix with orthonormal columns, $\mathbf{\Sigma} = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_K)$ is diagonal with descending singular values, and $\mathbf{U}$ is a $K \times K$ unitary matrix. Given the SVD, the linear detector is then reduced to

$$\hat{\mathbf{x}} = \mathbf{V}(\mathbf{\Sigma} + \rho\mathbf{\Sigma}^{-\mathbf{1}})^{-1}\mathbf{U}^H\mathbf{y}. \tag{4}$$

Notably, the matrix to be inversed is diagonal, and the linear detector requires complexity of $\mathcal{O}(MK + K^2)$. Nevertheless, conventional method to perform the SVD of the channel matrix requires a complexity of $\mathcal{O}(MK^2 + K^3)$. In this study, we employ power iterative method [X] to obtain the SVD numerically at the beginning stage. The power iterative method is helpful to reduce the complexity of the SVD to $\mathcal{O}(MK^2)$.

Assume that the channel is slowly faded, and the entries in the channel matrix vary subtly within a long coherence time. However, if only one or two users newly join or leave the cellular system, the SVD of the channel matrix may change significantly. In this case, it may require re-calculation of the SVD of the channel matrix, although channel coefficients regarding to the most users remains the same. To avoid the re-calculation of the SVD, we propose two numerical algorithms to update the SVD of the channel matrix when one user joins or leaves the cellular system, as will be descirbed in Sec.III-B and Sec. III-C.

3.2. Updating the SVD of the channel matrix when one user newly joinsthe cellular system

In this section, we will employ the Gram-Schmidt procedure, Householder transformation, andthe Golub-Reinsch method to update the SVD of the channel matrix for the case with one newly joined user. Specifically, denote $\tilde{\mathbf{H}} = [\mathbf{H}\ \tilde{\mathbf{h}}]$ be the updated channel matrix, where the channel vector from the newly joined user, denoted as $\tilde{\mathbf{h}}$, is inserted to the last column of the channel matrix without the loss of generality. In this section, we will propose a numerical algorithm, which is accomplished by three steps, to update the SVD of the channel matrix $\tilde{\mathbf{H}} = [\mathbf{H}\ \tilde{\mathbf{h}}]$.

1) Step 1: Given the SVD of the channel matrix in (3),the channel matrix $\tilde{\mathbf{H}}$can be decomposed as

$$\tilde{\mathbf{H}} = \underbrace{[\mathbf{U} \quad \tilde{\mathbf{u}}]}_{\breve{\mathbf{U}}_a} \underbrace{\begin{bmatrix} \mathbf{\Sigma} & \mathbf{U}^H\tilde{\mathbf{h}} \\ \mathbf{0} & \alpha \end{bmatrix}}_{\breve{\mathbf{R}}_a} \underbrace{\begin{bmatrix} \mathbf{V}^H & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}}_{\breve{\mathbf{V}}_a^H}, \tag{5}$$

where$\tilde{\mathbf{u}} = \frac{1}{\alpha}(\mathbf{I} - \mathbf{U}\mathbf{U}^H)\tilde{\mathbf{h}}$ and $\alpha = \|(\mathbf{I} - \mathbf{U}\mathbf{U}^H)\tilde{\mathbf{h}}\|_2$. The matrix factorization in (5) is resulted from the Gram-Schmidt procedure. It is easily verified that the matrices $\breve{\mathbf{U}}_a \in \mathcal{C}^{M\times(K+1)}$ and $\breve{\mathbf{V}}_a \in \mathcal{C}^{(K+1)\times(K+1)}$ are semi-unitary. However, the matrix $\breve{\mathbf{R}}_a$ is non-diagonal. Hence, we will reduce the upper triangular matrix $\breve{\mathbf{R}}_a$ as a bi-diagonal matrix and a diagonal matrix in *Step 2* and *Step 3* sequentially.

2) Step 2: Since the matrix $\breve{\mathbf{R}}_a$ only has at most two non-zero entries in each row, we can easily transform the upper triangular matrix$\breve{\mathbf{R}}_a$ as a bidiagonal matrix through Householder transformations [X] which aims to obtain a mirror vector with respect to a hyper-plane. Specifically, let $\mathbf{s}$ be an arbitrary $m \times 1$ vector and let $\mathbf{e}_m = [1\ 0 \cdots 0]^T$ be an $m \times 1$ elementary vector with the first entry being one and others being zero. It can be shown that we can find an $m \times m$Householder transformation matrix

$$\mathbf{Q}(\mathbf{s}) = \mathbf{I}_m - \frac{(\mathbf{s} - \|\mathbf{s}\| \cdot \mathbf{e}_m)(\mathbf{s} - \|\mathbf{s}\| \cdot \mathbf{e}_m)^H}{(\mathbf{s} - \|\mathbf{s}\| \cdot \mathbf{e}_m)^H \mathbf{s}}, \tag{6}$$

such that $\mathbf{Q}(\mathbf{s})\mathbf{s} = \|\mathbf{s}\| \cdot \mathbf{e}_m$.For the uppertriangular matrix $\breve{\mathbf{R}}_a$, we can find two sequences of Householder transformation matrices $\{\mathbf{Q}_1, \mathbf{Q}_2, \cdots, \mathbf{Q}_{K-2}\}$ and $\{\tilde{\mathbf{Q}}_1, \tilde{\mathbf{Q}}_2, \cdots, \tilde{\mathbf{Q}}_{K-2}\}$, such that

$$\tilde{\mathbf{R}}_a \triangleq \mathbf{Q}_{K-2}\cdots\mathbf{Q}_2\mathbf{Q}_1\breve{\mathbf{R}}_a\tilde{\mathbf{Q}}_1\tilde{\mathbf{Q}}_2\cdots\tilde{\mathbf{Q}}_{K-2} \tag{7}$$

is a bi-diagonal matrix, where

$$\tilde{\mathbf{Q}}_k = \mathrm{diag}(\mathbf{I}_k, \mathbf{Q}(\tilde{\mathbf{q}}_k)^H), \tag{8}$$

$$\mathbf{Q}_k = \mathrm{diag}(\mathbf{I}_k, \mathbf{Q}(\mathbf{q}_k)), \tag{9}$$

are block diagonal and

$$\tilde{\mathbf{q}}_k = [\mathbf{Q}_{k-1}\cdots\mathbf{Q}_1\breve{\mathbf{R}}_a\tilde{\mathbf{Q}}_1\cdots\tilde{\mathbf{Q}}_{k-1}]^H_{k+1,k+1:K}, \tag{10}$$

$$\mathbf{q}_k = [\mathbf{Q}_{k-1}\cdots\mathbf{Q}_1\breve{\mathbf{R}}_a\tilde{\mathbf{Q}}_1\cdots\tilde{\mathbf{Q}}_k]_{k+1:K,k+1}, \tag{11}$$

can be obtained sequentially. As a consequence, the decomposition of the matrix $\tilde{\mathbf{H}}$ can be updated as $\tilde{\mathbf{H}}=\tilde{\mathbf{U}}_a\tilde{\mathbf{R}}_a\tilde{\mathbf{V}}_a^H$, where matrices $\tilde{\mathbf{U}}_a \triangleq \breve{\mathbf{U}}_a\mathbf{Q}_1^H\cdots\mathbf{Q}_{K-2}^H$ and $\tilde{\mathbf{V}}_a \triangleq \breve{\mathbf{V}}_a\tilde{\mathbf{Q}}_1\cdots\tilde{\mathbf{Q}}_{K-2}$ are semi-unitary and the matrix $\tilde{\mathbf{R}}_a$ is bi-diagonal.

3) Step 3-1: In this stage, we will apply the recursive method proposed by Golub and Reinsch [x] to reduce the values of the off-diagonal terms in $\tilde{\mathbf{R}}_a$, in order to approximate the SVD of the channel matrix. However, the values of the off-diagonal terms in $\tilde{\mathbf{R}}_a$ is in general large, which may require a large number of iterations to eliminate those off-diagonal terms. To reduce the complexity, we perform a sequences of Givens rotation to reduce the value of the off-diagonal entries before applying theGolubReinsch method. To begin with, let us define $\mathbf{G}(\alpha,\beta)$ as an 2×2 matrix given by

$$\mathbf{G}(\alpha,\beta) = \begin{bmatrix} \frac{\alpha^*}{\sqrt{|\alpha|^2+|\beta|^2}} & \frac{\beta^*}{\sqrt{|\alpha|^2+|\beta|^2}} \\ \frac{\beta}{\sqrt{|\alpha|^2+|\beta|^2}} & \frac{-\alpha}{\sqrt{|\alpha|^2+|\beta|^2}} \end{bmatrix}. \tag{12}$$

It can be easily verified that

$$\mathbf{G}(\alpha,\beta)\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \sqrt{|\alpha|^2+|\beta|^2} \\ 0 \end{bmatrix}.$$

We can perform a sequence of the Gives rotations on $\tilde{\mathbf{R}}_a$, such that

$$\hat{\mathbf{R}}_a \triangleq \mathbf{G}_{K-1}\cdots\mathbf{G}_2\mathbf{G}_1\tilde{\mathbf{R}}_a\tilde{\mathbf{G}}_1\tilde{\mathbf{G}}_2\cdots\tilde{\mathbf{G}}_{K-1} \tag{13}$$

is an updated bidiagonal matrix with a smaller value of off-diagonal entries, where

$$\tilde{\mathbf{G}}_k = \mathrm{diag}(\mathbf{I}_{k-1}, \mathbf{G}(\tilde{\alpha}_k,\tilde{\beta}_k)^T, \mathbf{I}_{K-k-1}) \tag{14}$$

$$\mathbf{G}_k = \mathrm{diag}(\mathbf{I}_{k-1}, \mathbf{G}(\alpha_k,\beta_k), \mathbf{I}_{K-k-1}) \tag{15}$$

are block diagonal with

$$\begin{bmatrix} \tilde{\alpha}_k \\ \tilde{\beta}_k \end{bmatrix} = \Big[\mathbf{G}_{k-1}\cdots\mathbf{G}_1\tilde{\mathbf{R}}_a\tilde{\mathbf{G}}_1\cdots\tilde{\mathbf{G}}_{k-1}\Big]^T_{k,k:k+1} \tag{16}$$

$$\begin{bmatrix} \alpha_k \\ \beta_k \end{bmatrix} = \Big[\mathbf{G}_{k-1}\cdots\mathbf{G}_1\tilde{\mathbf{R}}_a\tilde{\mathbf{G}}_1\cdots\tilde{\mathbf{G}}_k\Big]_{k:k+1,k}. \tag{17}$$

As a consequence, the decomposition of the matrix $\tilde{\mathbf{H}}$ can be updated as

$\tilde{\mathbf{H}} = \hat{\mathbf{U}}_a \hat{\mathbf{R}}_a \hat{\mathbf{V}}_a^H$, where matrices $\hat{\mathbf{U}}_a \triangleq \tilde{\mathbf{U}}_a \mathbf{G}_1^H \cdots \mathbf{G}_{K-1}^H$ and $\hat{\mathbf{V}}_a \triangleq \tilde{\mathbf{V}}_a \tilde{\mathbf{G}}_1 \cdots \tilde{\mathbf{G}}_{K-1}$ are semi-unitary and the matrix $\hat{\mathbf{R}}_a$ is bi-diagonal with smaller off-diagonal entries.

4) Step 3-2: To further reduce the off-diagonal entries of $\hat{\mathbf{R}}_a$, we apply theGolubReinsch method following by the Givens Rotation. Specifically, let $\{\mathbf{P}_1, \cdots, \mathbf{P}_{K-1}\}$ and $\{\tilde{\mathbf{P}}_1, \tilde{\mathbf{P}}_2, \cdots, \tilde{\mathbf{P}}_{K-1}\}$ be two sequences of givens rotations obtained by

$$\tilde{\mathbf{P}}_k = \mathrm{diag}(\mathbf{I}_{k-1}, \mathbf{G}(\tilde{\tau}_k, \tilde{\mu}_k)^T, \mathbf{I}_{K-k-1}) \tag{18}$$

$$\mathbf{P}_k = \mathrm{diag}(\mathbf{I}_{k-1}, \mathbf{G}(\tau_k, \mu_k), \mathbf{I}_{K-k-1}) \tag{19}$$

with parameters

$$\tilde{\tau}_k = [\hat{\mathbf{R}}_k]_{k,k}^2 - \lambda_k \tag{20}$$

$$\tilde{\mu}_k = [\hat{\mathbf{R}}_k]_{k,k} [\hat{\mathbf{R}}_k]_{k,k+1} \tag{21}$$

$$\begin{bmatrix} \tau_k \\ \mu_k \end{bmatrix} = \left[\hat{\mathbf{R}}_k \tilde{\mathbf{P}}_k\right]_{k:k+1,k}, \tag{22}$$

where $\hat{\mathbf{R}}_k = \mathbf{P}_{k-1} \cdots \mathbf{P}_1 \hat{\mathbf{R}}_a \tilde{\mathbf{P}}_1 \cdots \tilde{\mathbf{P}}_{k-1}$ and λ_k is the eigenvalue of the submatrix $[\hat{\mathbf{R}}_k]_{k:k+1,k:k+1}$ which is closer to $[\hat{\mathbf{R}}_k]_{k+1,k+1}$.With the Givens rotations, the matrix $\tilde{\mathbf{H}}$ can be factorized as $\tilde{\mathbf{H}} = \check{\mathbf{U}}_a \check{\mathbf{R}}_a \check{\mathbf{V}}_a^H$, where $\check{\mathbf{U}}_a \triangleq \hat{\mathbf{U}}_a \mathbf{P}_1^H \cdots \mathbf{P}_{K-1}^H$ and $\check{\mathbf{V}}_a \triangleq \hat{\mathbf{V}}_a \tilde{\mathbf{P}}_1 \cdots \tilde{\mathbf{P}}_{K-1}$ are semi-unitary and $\check{\mathbf{R}}_a \triangleq \mathbf{P}_{K-1} \cdots \mathbf{P}_1 \hat{\mathbf{R}}_a \tilde{\mathbf{P}}_1 \cdots \tilde{\mathbf{P}}_{K-1}$ is bi-diagonal with much smaller off-diagonal entries.

5) Step 4: Repeat Step 3 when the values of the off-diagonal entries in $\check{\mathbf{R}}_a$ are sufficiently small.

It is worth noting that although the Step 3-2 can gradually reduce the off-diagonal entries of the bi-diagonal matrix $\check{\mathbf{R}}_a$, combining the Givens rotation in Step 3-1 is helpful to the convergence of the algorithm

2.3. Updating the SVD of the channel matrix when one user leavesthe cellular system

In this section, we consider the case that one user leaves the cell. Specifically, given the SVD of the original channel matrix $\mathbf{H} = [\bar{\mathbf{H}}\ \mathbf{h}_K]$, we will update the SVD of the matrix $\bar{\mathbf{H}}$, which is obtained by deleting the last column of $\mathbf{H}$ based on the following steps.

1) *Step 1*: To eject the last column $\mathbf{h}_K$ off the matrix $\mathbf{H}$, we can find a sequence of Givens rotation matrices $\{\tilde{\mathbf{B}}_1, \tilde{\mathbf{B}}_2, \cdots, \tilde{\mathbf{B}}_{K-1}\}$, such that

$$\tilde{\mathbf{B}}_{K-1} \tilde{\mathbf{B}}_{K-2} \cdots \tilde{\mathbf{B}}_1 \mathbf{V} = \begin{bmatrix} \check{\mathbf{V}}_r & \mathbf{0}_{K-1} \\ \mathbf{0}_{K-1}^T & 1 \end{bmatrix}, \tag{23}$$

where

$$\tilde{\mathbf{B}}_k = \mathrm{diag}(\mathbf{I}_{k-1}, \mathbf{G}(\tilde{\gamma}_k, \tilde{\omega}_k), \mathbf{I}_{K-k-1}) \tag{24}$$

is block diagonal with parameters

$$\tilde{\gamma}_k = -[\tilde{\mathbf{B}}_{k-1} \cdots \tilde{\mathbf{B}}_1 \mathbf{V}]^*_{k+1,K} \tag{25}$$

$$\tilde{\omega}_k = [\tilde{\mathbf{B}}_{k-1} \cdots \tilde{\mathbf{B}}_1 \mathbf{V}]^*_{k,K}. \tag{26}$$

We can also find another sequence of Gives rotation matrices $\{\mathbf{B}_1, \mathbf{B}_2, \cdots, \mathbf{B}_{K-1}\}$, such that

$$\breve{\mathbf{R}} \triangleq \mathbf{B}_{K-1} \cdots \mathbf{B}_2 \mathbf{B}_1 \mathbf{R} \tilde{\mathbf{B}}_1^H \tilde{\mathbf{B}}_2^H \cdots \tilde{\mathbf{B}}_{K-1}^H \tag{27}$$

is upper triangular, where

$$\mathbf{B}_k = \mathrm{diag}(\mathbf{I}_{k-1}, \mathbf{G}(\gamma_k, \omega_k), \mathbf{I}_{K-k-1}) \tag{28}$$

is block diagonal with parameters

$$\gamma_k = [\mathbf{R} \tilde{\mathbf{B}}_1^H \tilde{\mathbf{B}}_2^H \cdots \tilde{\mathbf{B}}_{K-1}^H]_{k,k} \tag{29}$$

$$\omega_k = [\mathbf{R} \tilde{\mathbf{B}}_1^H \tilde{\mathbf{B}}_2^H \cdots \tilde{\mathbf{B}}_{K-1}^H]_{k+1,k}. \tag{30}$$

With the sequences of Givens rotations, the matrix $\mathbf{H}$ can be decomposed as

$$[\bar{\mathbf{H}}\ \mathbf{h}_K] = \breve{\mathbf{U}} \breve{\mathbf{R}} \begin{bmatrix} \breve{\mathbf{V}}_r^H & \mathbf{0}_{K-1} \\ \mathbf{0}_{K-1}^T & 1 \end{bmatrix}, \tag{31}$$

here $\breve{\mathbf{U}} \triangleq \tilde{\mathbf{U}} \mathbf{B}_1^H \mathbf{B}_2^H \cdots \mathbf{B}_{K-1}^H$ and $\breve{\mathbf{V}}_r$ are semi-unitary. It can be easily shown that the deflated matrix $\bar{\mathbf{H}}$ can be factorized by

$$\bar{\mathbf{H}} = \breve{\mathbf{U}}_r \breve{\mathbf{R}}_r \breve{\mathbf{V}}_r^H \tag{32}$$

where $\breve{\mathbf{U}}_r \in \mathbb{C}^{M \times (K-1)}$ is a sub-matrix of $\breve{\mathbf{U}}$ obtained by deleting the last column, and $\breve{\mathbf{R}}_r$ is the leading principal minor of $\breve{\mathbf{R}}$ of order $K-1$. Note that the matrix $\breve{\mathbf{U}}_r$ has orthonormal columns and $\breve{\mathbf{R}}_r$ is upper triangular.

2) *Step 2*: Similar to *Step 2* in *Sec. III-B*, we can find two sequences of Householder transformation matrices, such that the deflated matrix can be decomposed as

   $\bar{\mathbf{H}} = \tilde{\mathbf{U}}_r \tilde{\mathbf{R}}_r \tilde{\mathbf{V}}_r^H$, where matrices $\tilde{\mathbf{U}}_r$ and $\tilde{\mathbf{V}}_r$ are semi-unitary and the matrix $\tilde{\mathbf{R}}_a$ is reduced to bi-diagonal.

3) *Step 3*: Repeat the method in Step 3-1 and Step 3-2 to reduce the values of the off-diagonal entries in $\tilde{\mathbf{R}}_a$ until the off-diagonal entries are negligible.

IV. NUMERICAL RESULT

In this chapter we provide computational complexity and simulation result. Computational complexity when one user join the base station is

Otherwise

$$\begin{aligned} & M + 2M^2 + \sum_{i=2}^{K-1} K(M-i)^2 + (M-i)^2 \\ , \\ & + \sum_{i=2}^{K-1} M(M-i)^2 + 12M(M+K+3) \end{aligned} \tag{32}$$

when one user leave the base station, computational complexity show as :

$$MK + 2K + M + 2M^2 + \sum_{i=2}^{K-1} K\left(M - i\right)^2 + \left(M - i\right)^2 + \sum_{i=2}^{K-1} M\left(M - i\right)^2 + 8M\left(M + K + 3\right) \quad (33)$$

M shows the number of antennas in the base station whereas K shows how many users are in the base station. while shows the iteration used. The number of iterations used in each desired error will be shown in the results. It appears that the more iterations the greater the computational complexity. Computational complexity in our proposed algorithm is keep iterated by using sum, not multiplication. If we used original Zero Forcing we have to multiply large matrix in number column and row because in the next generation we used very big number of antenna and user ini one base station.

For zero forcing and MMSE detection we need so it can be seen that the computational complexity is much lower than the original zero forcing. We need to inverse matrix to get approximation number for real signal. In this case, if we used SVD , so we can easily inverse matrix without calculating big number multiplication.

In this chapter show comparison zero forcing using conventional method with zero forcing using proposed method. In other hand, this section also compare MMSE using conventional method with using proposed method..

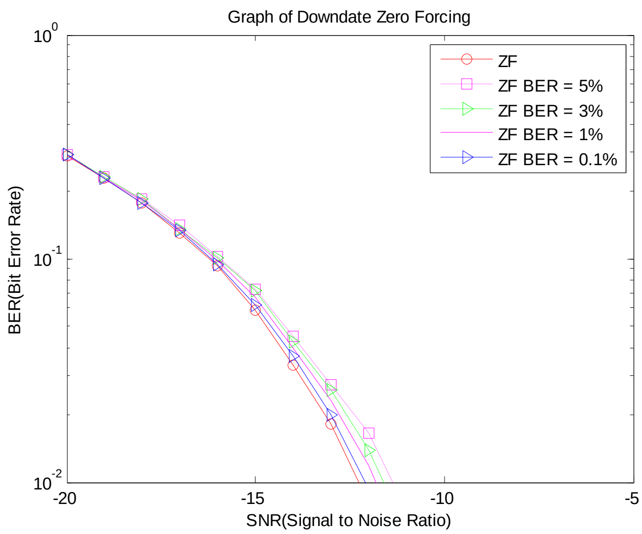

Fig. 1 Graph of SNR vs BER Downdate Zero Forcing

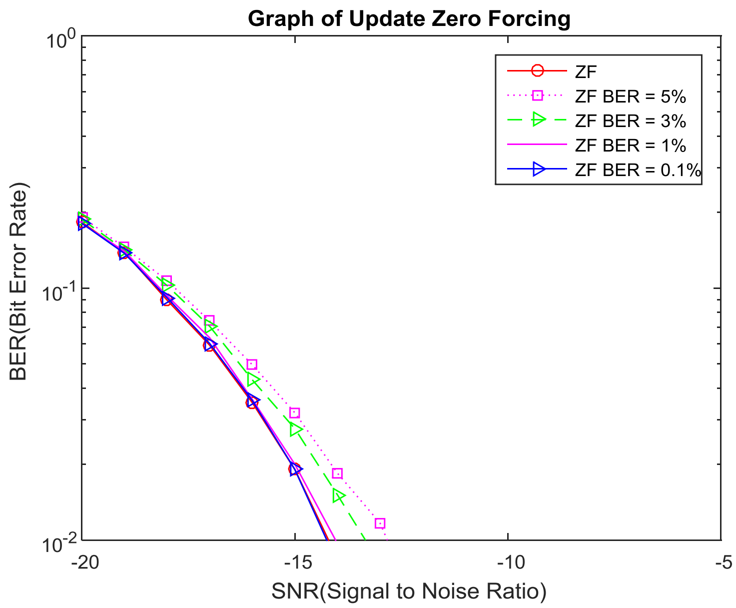

Fig. 2 Graph of SNR vs BER Update Zero Forcing

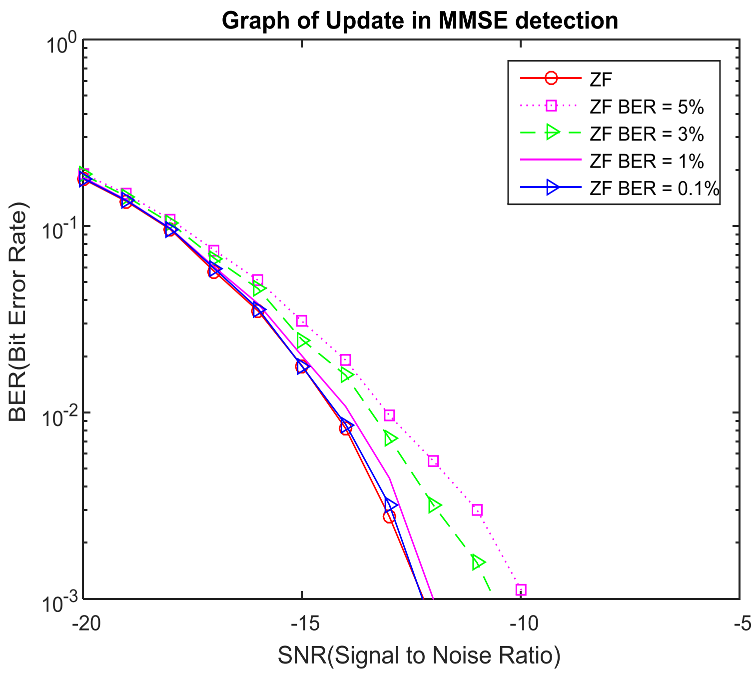

Fig. 3 Graph of SNR vs BER Update in MMSE

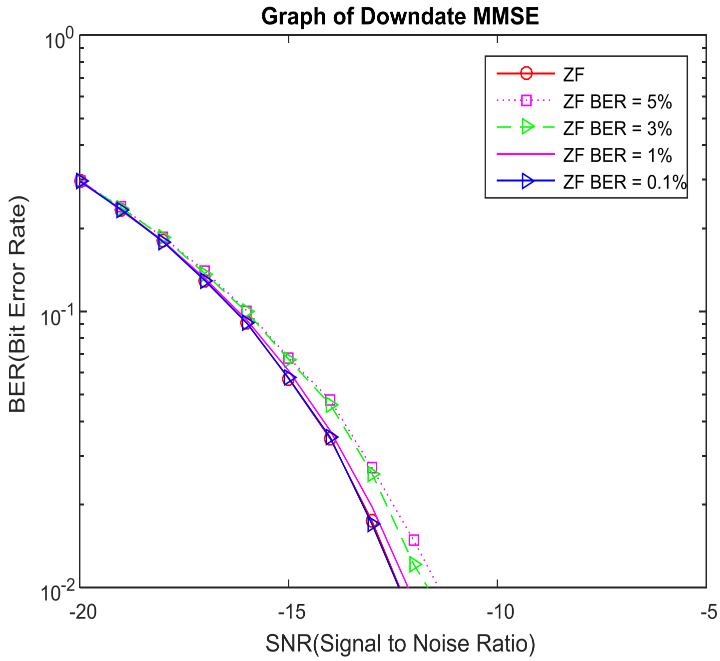

Fig. 4 Graph of SNR vs BER Downdate in MMSE

In the figure 1-4 show that convergence for each linear detection. In figure 1, graph downdate (when one user leave the base station) for zero forcing linear detection . In this figure, we show that using this research we get a variety of BER. For magenta grid strip line set BER about 5%, while light green triangle line shows the result when BER set at 3%, magenta line set BER at 1% and the last blue triangle line we set 0.1 %. The larger the BER, the smaller the computational complexity. Meanwhile in figure 2, graph update for zero forcing liear detection. For this second figure, we use a zero forcing linear detector as well. The red rounded line represents the true value of zero forcing. As for the line magenta box show zero forcing our approach with BER of 5%. The green stripe line shows the result of our approach also with 3% BER result. When using Zero Forcing approach with BER 1% we show with magenta line and triangle blue line shows zero forcing approach 0.1%.

In figure 1 and figure 2 the results show our approach when zero forcing diapplied. If the required is a very small BER here we show with 0.01% BER, then the result is indeed close to the actual Zero Forcing result but has a weakness that is required looping in the Given Rotation algorithm to reduce the value of bidiagonal obtained and to close the true value. Can be seen in figure one and figure two that the result of approach with BER 1% not too much different from the actual Zero Forcing results. With a 1% BER result, it is found that the computatonal complexity to produce the BER is not too high even lower than other algorithms.can be seen from figure 1 or figure 2, the result of down date user or oritma updating result, the algorithm we proposed in approach of zero forcing result when one user leave base station tend to be better than user update or when one user leave base station.

This is because when using the downdating algorithm, before using the rotation already formed SVD first rotation. So when one user is removed it is close to the actual Zero Forcing value. While updating a lot is done given rotation to make the matrix back into SVD because the result of adding the user to generate triangular matrix instead of SVD. Require a lot of iteration on given rotation to remove the bidiagonal matrix and transform into SVD.

In other hand, figure 3 show that graph update in MMSE Detection and figure 4 show that graph downdate in MMSE Detection.The figure 3 red round indicates the actual MMSE result. While the MMSE result that we proposed with BER 5% is indicated by magenta strip line. The triangle green stripe line shows the MMSE results we proposed with BER 3%. The magenta line shows our proposed MMSE with 1% and the last blue triangle line shows MMSE results with BER 0.1%.The larger the BER, the less iteration is used but the results look away from the actual. But this is a trade off where if we use a lot of iterations, the results are indeed close to the actual signal but the complexity is also higher because many summations will be used although it tends slightly compared with the actual MMSE calculations.

In the figure 4, larger the BER, less iteration is used but the results look away from the actual. But this is a trade off where if we use a lot of iterations, the results are indeed close to the actual signal but the complexity is also higher because many summations will be used although it tends slightly compared with the actual MMSE calculations. As well as the results listed in Zero Forcing detection, it produces an SVD approaching the real one difference value when we that if we update the error signal is greater than the error when downdate with the same number of iterations. This is because when downdate, the update results too many values triangular nonzero so it takes a lot of given rotation to produce the value that approximates the original MMSE. While on downdate, fewer number in triangular. Many triangular value is zero. So to get an SVD close to the real value will be easier.

This four figure compare different error in different Linear Detection. This errorhas trade off with computational complexity. If error less then, computational complexity will higher.The PDF for each iteration and for each error will denoted in figure 5 and figure 6.

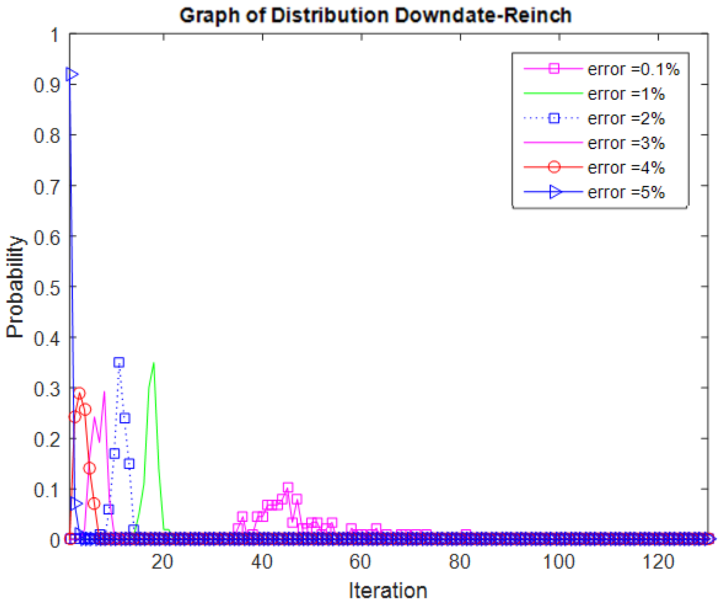

Fig. 5 Graph Distribution Downdate

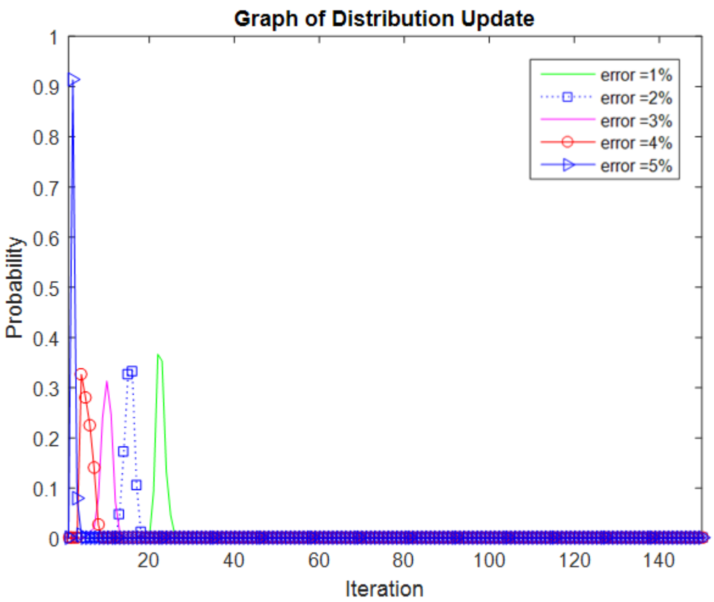

Fig. 6 Graph Distribution Update

In figure 5 and 6 show that probability function from error=1% until error =5%. Both figure show how many iteration that we used for achieve each error. In the graph the downdate distribution in figure 5 shows how many iterations are needed to generate the desired BER when one user leaves the base station. When to generate a signal with 0.1% BER shown in the round magenta line that it takes a lot of repetition starting from 30 up to more than 80 times iteration. From this result shows that if BER wants 0.1% then computational complexity will be high. While at BER 1% it takes about 18 iterations and the result is not too far compared to BER 0.1%. Here it can be seen that with iteration approximately 18 times can be produced low computation complexity and the results near the actual one for MMSE and Zero Forcing Detection.For the blue box line shows BER 2% where the loop is done about 15 times. For BER 3%, iteration required about 8 times, while for 4% BER is looped about 5 times. The last time it loops once the Bit Error Rate gets around 5%.

Figure 6 shows the Graph Distribution Update for our proposed algorithm. In this graph does not show an Error or BER about 0.1% because to reach the value iterations required exceeding 140 iterations. In this case, computational complexity is very high to get results with BER 0.1%. This happens because the result of updating matrix there are many nonzero values in triangular given rotation matrix so it needs more computation compared with downdate algorithm which tends to have few nonzero value on triangular matrix.Green line indicates 1% error if using iteration approximately 23 times. Whereas 2% error is obtained if iteration repeat about 18 times that have been shown by blue stripe box line. 3% result obtained if using iteration about 15 times while result 4% obtained when using iteration approximately 5 times indicated by the red rounded line. Lastly, Error 5% is obtained at iteration approximately 2 times.

The comparison between MMSE and Zero Forcing detector can be seen in figure 1-4 where MMSE get more BER than Zero Forcing. This is because MMSE has more complexity when getting real user signal then it should multiply more matrix compared to Zero Forcing. Because the Zero Forcing calculation is simpler, then the error is not too much to accumulate in Zero Forcing detection. Seen from both 5 and 6, to encounter a small error of 0.1% required a very high iteration. For downdate, the required iteration is about 80 iterations and for updated it takes more than 140 iterations. From this it can be seen that in order to achieve BER of 0.1%, the trade-off obtained is that the computational complexity is also high, even smaller than the actual Zero Forcing or MMSE.

From both above graph also got that to reach BER about 1% required iteration that not too much to downdate about 20 times and for update about 23 iteration. Here with the iteration that is not too high obtained the result also approached the actual detection of MMSE and Zero Forcing. At 1% of BER is obtained Computational Complexity is not high.Here it can be seen also that MMSE and Zero Forcing have different errors due to different algorithms so that their computational complexity is different too. Zero Forcing has a simpler algorithm than MMSE detection. Although the computational complexity of MMSE detection is higher than Zero Forcing Detection, but the computational complexity of MMSE is lower than the exact calculation which must multiply the huge matrix to inverse the matrix in order to obtain the original user signal. Huge matrix caused by the number of users and many antennas in one base station.

V. CONCLUSION

Result show that our algorithm that used SVD algorithm combined with reinch algorithm and given rotation has Computational complexity less than calculation when we used manual exact MMSE and ZF linear detection. Computational Complexity for update and downdate are different. If error less than 1% computational complexity is really high.Good result shows in error 1%. The proposed algorithm prove that can be used for both MMSE and ZF detection.

REFERENCES

[1] M. Wu, B. Yin, G. Wang, C. Dick, J. R. Cavallaro, and C. Studer, Large-Scale MIMO Detection for 3GPP LTE: Algorithms and FPGA Implementations, IEEE J. Sel. Top. Signal Process., *vol. 8, no. 5,* 2014, 916–929.

[2] L. Dai, X. Gao, X. Su, S. Han, C.-L. I, and Z. Wang, Low-Complexity Soft-Output Signal Detection Based on Gauss–Seidel Method for Uplink Multiuser Large-Scale MIMO Systems, IEEE Trans. Veh. Technol., *vol. 64, no. 10,* 2015, 4839–4845.

[3] X. Qin , Z. Yan, and G. He , A Near-Optimal Detection Scheme Based on Joint Steepest Descent and Jacobi Method for Uplink Massive MIMO Systems, IEEE Communications Letters., *vol. 20,* 2015, 276-279.

[4] C-I. Wu, W-J. Huang, W-H. Chung, Robust Update Algorithms for Zero-Forcing Detection in Uplink LargeScale MIMO Systems, IEEE Communications Letters, *vol. 22*, 2018, 424-427

[5] F. Rosario, F. A. Monteiro, and A. Rodrigues, Fast Matrix Inversion Updates for Massive MIMO Detection and Precoding, IEEE Signal Processing Letters *vol. 23*,2016, 75-79.

[6] C. Tang, C. Liu, L. Yuan, and Z. Xing, High precision low complexity matrix inversion based on Newton iteration for data detection in the massive MIMO,IEEE Communications Letters., *vol. 20*, 2016, 490-493.

[7] K. L. Chung, and W-M. Yan, The complex Householder transform, IEEE transactions on signal processing., *vol. 45,* 1997, 2374-2376.

[8] D. Zhu, B. Li, and P. Liang, On the matrix inversion approximation based on Neumann series in massive MIMO systems, Communications (ICC), IEEE International Conference,2015.

[9] X.-G. Xia and B.W. Suter, On the Householder transform in C^m, Digital Signal Processing., *vol.5*, 1995, 116-117.

[10] A. Cline, I. Dhillon. Handbook of Linear Algebra, series editor ed., K. H. Rosen, Ed. (United States of America: CRC Press, 2007).

[11] G. H. Golub, C. F. V. Loan, Matrix Computations, *3rd ed.*, (Baltimore and London: Johns Hopkins Univ. Press, 1989).

[12] V. C. Venkaiah, V. Krishna, A. Paulraj, Householder transform in $\mathbf{C}^m$, Digital Signal Processing., *vol. 3*, 2002, 226-227.