PES University, Bangalore

(Established under Karnataka Act No. 16 of 2013)

MAY 2020: IN SEMESTER ASSESSMENT (ISA) B.TECH. IV SEMESTER

UE18MA251- LINEAR ALGEBRA

MINI PROJECT REPORT

ON

APPLICATIONS OF LINEAR ALGEBRA IN MU-MIMO

Submitted by

1. CHANDAN SHANKAR M PES1201801986
2. ABHISHEK T H PES1201801869

Branch & Section : ELECTRONICS AND COMMUNICATION , Sec: D

PROJECT EVALUATION

(For Official Use Only)

Sl.No.	Parameter	Max Marks	Marks Awarded
1	Background & Framing of the problem	4	
2	Approach and Solution	4	
3	References	4	
4	Clarity of the concepts & Creativity	4	
5	Choice of examples and understanding of the topic	4	
6	Presentation of the work	5	
	Total	25	

Name of the Course Instructor : VRINDA KAMATH .K

Signature of the Course Instructor :

Introduction:

Wireless Communication is a method of transmitting information from one point to other, without using any connection like wires, cables or any physical medium. The most developed method of wireless communication currently is MIMO (multiple input, multiple output) MIMO is a technology for wireless communications in which multiple antennas are used at both the source (transmitter) and the destination (receiver). In MIMO systems, a transmitter sends multiple streams by multiple transmit antennas. The transmit streams go through a matrix channel which consists of all paths between the transmit antennas at the transmitter and receiving antennas at the receiver.[1] Then, the receiver gets the received signal vectors by the multiple receiving antennas and decodes the received signal vectors into the original information. Mainly there are two processes in transmission the uplink transmission and the downlink transmission. Downlink transmission has two major steps beamforming and precoding. In this paper we will be looking into the method of Precoding employed.

Beamforming is a technique that focuses a wireless signal towards a specific receiving device, rather than having the signal spread in all directions from a broadcast antenna, as it normally would. The resulting more direct connection is faster and more reliable than it would be without beamforming.

Precoding is a technique that exploits transmit diversity by weighting the information stream, i.e. the transmitter sends the coded information to the receiver to achieve pre-knowledge of the channel. This technique will reduce the corrupted effect of the communication channel. accordingly. Linear precoding techniques at the downlink aim to focus each signal at its desired terminal and mitigate interference towards other terminals

Precoding is a generalization of beamforming to support multi-stream (or multi-layer) transmission in multi-antenna wireless communications. In conventional single-stream beamforming, the same signal is emitted from each of the transmit antennas with appropriate weighting (phase and gain) such that the signal power is maximized at the receiver output. When the receiver has multiple antennas, single-stream beamforming cannot simultaneously maximize the signal level at all of the receive antennas.[2]

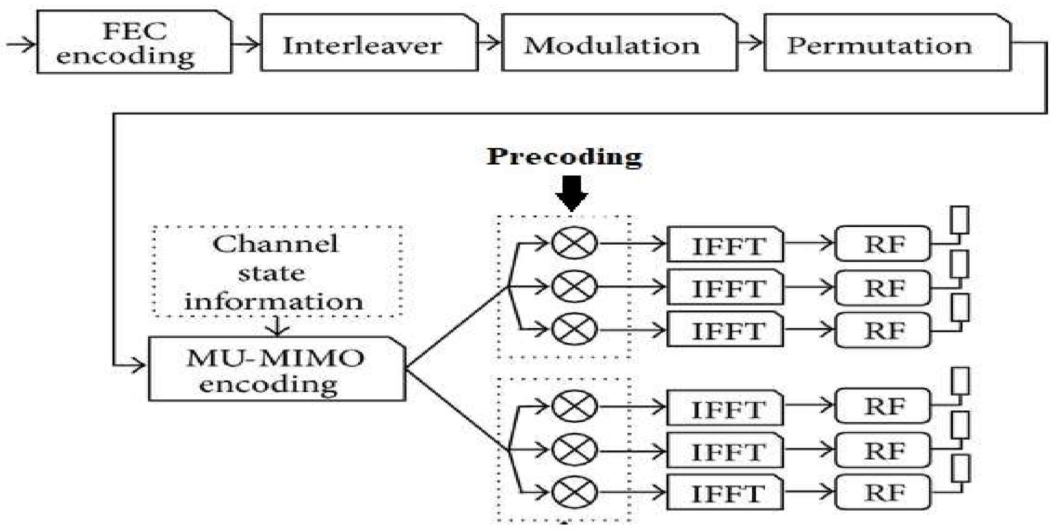

Figure 1: Downlink transmission model

A Simple MIMO system can be modelled as

$$\mathbf{y=hx+n} \tag{1}$$

where y is receiving vector, x is transmitting vector, h is channel matrix and n is noise vector. This is shown in the figure 1 and 2.

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{N_R} \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1N_T} \\ h_{21} & h_{22} & \cdots & h_{2N_T} \\ \vdots & \ddots & \ddots & \vdots \\ h_{N_R1} & h_{N_R2} & \cdots & h_{N_RN_T} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N_T} \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ \vdots \\ n_{N_R} \end{bmatrix}$$

Fig. 2. MIMO system model

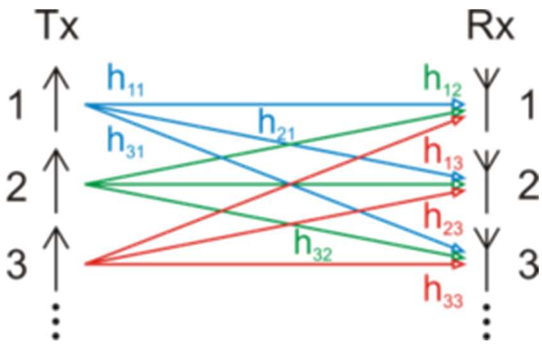

Fig. 3. MIMO system model.

Currently Multi User Multiple Input Multiple Output systems are employed in almost all wireless application. Multi-User-MIMO is an innovative breakthrough that enables simultaneous communication to multiple devices, improving overall speed and enabling network multitasking. Multiple simultaneous communication between the two entities was groundbreaking and outranked its previous generation tech in terms of exchange in data rates, outstanding speed etc., But it came with major tradeoffs.

- **The problem and justification:**

Firstly, the technology of MU-MIMO and its implementation was too complex and needed high computational potential hardware. Secondly there were other problems like interference or mix up of data between the users called the multi-user interference (MUI) which in turn affects the performance. This problem was tackled by various MU-MIMO schemes which were aimed to cancel the MUI via linear precoding algorithms at the transmitter itself. Some of them are zero-forcing channel- forcing channel-inversion (ZF-CI), Block diagonalization (BD), QR-BD etc.BD is a well-known linear precoding scheme for MU-MIMO downlink transmission. The ZF-CI precoding scheme [3] introduces ZF precoder at the transmitter to cancel the MUI. However, it results in low performance due to the noise amplification, especially when the channel is ill-conditioned. Although the BD precoding scheme improves the sum-rate, the computational costs of BD precoding scheme is very high because of the two SVD operations for each user. In QR-BD the first SVD operation is replaced by QR decomposition to mitigate the MUI [4]. However, the QR decomposition needs to be implemented for each user, which still results in heavy computational complexity.

In other words, every linear precoding schemes listed above have major drawbacks and there is dire need of a new scheme with more merits especially when the technology is moving towards its next stage 5G.

- **Aim of the topic:**

In this paper we look at a new approach towards linear precoding with the aim of reducing the computational complexities and thereby efficiency. The proposed scheme first introduces QR decomposition to the Hermitian transpose of a combined channel matrix to find the null space vectors of MUI. The QR decomposition is implemented only once. Then, the Gram Schmidt Orthonormalization method is applied to find the orthonormal basis of the null space vectors. Thus, the proposed scheme avoids SVD operation of the BD precoding and requires only one QR decomposition operation to significantly reduce the computational complexity.

Thus, there is a huge scope to this method as it can produce the same output of much complex important algorithms with less complexity, less effort, more speed and more reliability.

Review of Literature:

System Model: -

We consider a downlink Multi-User-MIMO system with one base station (BS) and K users, where the BS is equipped with N_T transmit antennas and each user with N_k antennas as shown in Figure 3.We assume that the perfect instantaneous channel state information (CSI) for each user is available at the BS. N_k multiple data streams S_k for user k are pre-coded by the corresponding precoding matrix W_k at the transmitter. All these streams from K users are combined together after precoding and emitted N_T from transmit antennas at BS.

The multiple user MIMO system will be modelled by the channel matrix as

$$\boldsymbol{H}_k = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1N_T} \\ h_{21} & h_{22} & \cdots & h_{2N_T} \\ \vdots & \vdots & \ddots & \vdots \\ h_{N_k 1} & h_{N_k 2} & \cdots & h_{N_k N_T} \end{bmatrix}$$

where, $\mathbf{H}_k \in \mathbf{C}^{N(K) \times N(T)}$ is the MIMO channel matrix for user k and the element h_{ij} indicates the channel impulse response coupling the j -th transmit antenna to the i -th receive antenna.

The received signal Y_k for user k is given by

$$y_k = H_k W_k s_k + H_k \sum_{i=1, i \neq k}^{K} W_i s_i + n_k , \qquad (2)$$

where, $\mathbf{W_k} \in \mathbf{C}^{\mathbf{N(T)\ X\ N(k)}}$ is the precoding matrix for user k and $\mathbf{S_k}$ is the transmit signal vector for user k. Note that each user transmits N_k data streams and therefore $S_k \in C^{N(k)\ X\ 1}$. $\mathbf{n_k} \in C^{N(k)\ X\ 1}$ is the k -th user's Gaussian noise. Therefore, final MIMO system is modelled to get the signal will be,

$$\mathbf{Y_K = H_K W S + n_k} \qquad (3)$$

$$\mathbf{W = [W_1\ W_2\ \ldots\ldots\ W_K]} \qquad (4)$$

$$s = [s_1 \quad s_2 \quad \ldots \quad s_K]^T , \qquad (5)$$

where, $\mathbf{W_k} \in \mathbf{C}^{\mathbf{N(T)\ X\ KN(k)}}$ and $S_k \in C^{KN(k)\ X\ 1}$ is the total precoding matrix and the total transmit signals for k users respectively. We denote $N_R = KN_k$.
The total MIMO channel matrix for k users is formulated as

$$H_S = [H_1^T \quad H_2^T \quad \cdots \quad H_K^T]^T , \qquad (6)$$

the total channel matrix $\mathbf{H_S}$ excluding the k-th user's channel is given by

$$\mathbf{G_k} = [\ \mathbf{H_1}^T \ldots\ \mathbf{H_{K-1}}^T\ \ \mathbf{H_{K+1}}^T\ \ldots\ \mathbf{H_k}^T]^T \qquad (7)$$

where $\mathbf{G_k} \in C^{A\ X\ N(T)}$ and $A = N_R - N_k$.

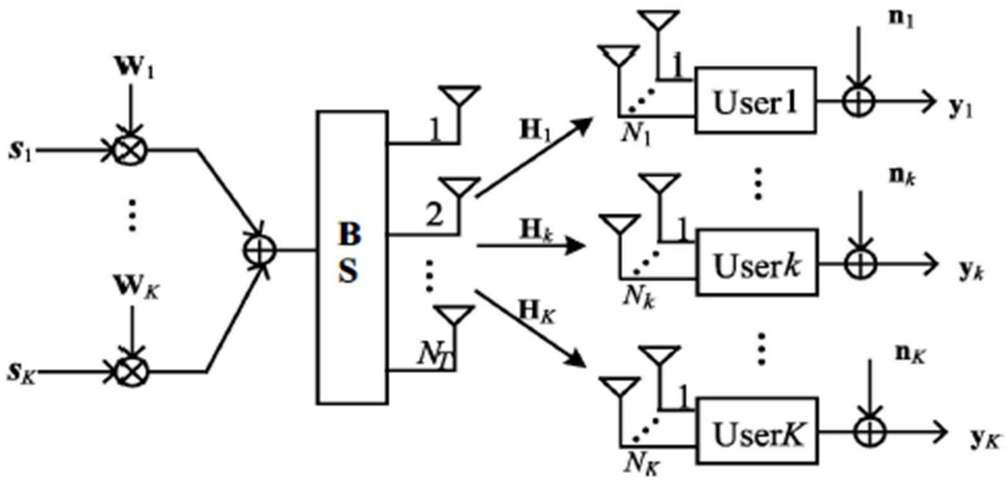

Fig. 4. MU-MIMO system.

Review of Conventional BD and QR-BD Precoding Algorithms

Although in earlier times the zero-forcing channel- forcing channel-inversion algorithm was used the algorithm had its disadvantages and is very rarely employed. We will not be discussing this algorithm in this paper. However, we will be discussing the more conventional and common BD and QR-BD precoding algorithms in order to compare with the proposed QR-GSO precoding algorithm.

BD Precoding Algorithm:

In general,BD algorithm is implemented in two steps:-

The first step aims to compute an orthonormal basis **E** for the null space of $\mathbf{G_k}$ defined in (7), which can be used to eliminate the MUI[5]. Thus the equivalent parallel SU-MIMO channels are obtained.

The second step is used to parallelize each user's streams and obtain the maximum precoding gain by **F** .

Therefore precoding matrix **W=E F .**

Step 1:

Obtain the first precoding matrix **E**.

For user k, do SVD to $\mathbf{G_k}$ as

$$\mathbf{G_k = U_k \Lambda_k V_k^H = U_k \Lambda_k [\, V_k^{(1)}\, Vk^{(0)}]^H} \tag{8}$$

where $\mathbf{U_k}$ is the unitary matrix containing the left singular vectors, $\mathbf{V_k}$ is the unitary matrix containing the right singular vectors and $\mathbf{\Lambda_k}$ is the diagonal matrix containing the singular values of $\mathbf{G_k}$. Dividing $\mathbf{V_k}$ into two parts**:** $\mathbf{V_k}^{(1)}$ consists of the first B_k non-zero singular vectors, $\mathbf{V_k^{(0)}}$ consists of the last $N_T - B_k$ zero singular vectors, where B_k is the rank of $\mathbf{G_k}$. since $\mathbf{V_k^{(0)}}$ forms an orthonormal basis for the null space of $\mathbf{G_k}$, the first precoding matrix $\mathbf{E}_k$ for user k of the BD precoding algorithm is obtained as

$$\mathbf{E_k = V_k^{(0)}.} \tag{9}$$

After traversing all the users, the first precoding matrix E for k users is

$$\mathbf{E = [\, E_1\, , \ldots, E_{k-n}, \ldots, E_k\,].} \tag{10}$$

Step 2:

Obtain the second precoding matrix **F** to parallelize each user's streams and achieve the maximum precoding gain for each user.

After the first precoding operation, the MU-MIMO channel is transformed into parallel SU-MIMO channels as

$$\mathbf{H_S E = [\, H_1 E_1, \ldots, H_{k-n} E_{k-n}, \ldots, H_k E_k\,]} \tag{11}$$

We define the effective channel matrix without MUI for user k as $\mathbf{D_k = H_k E_k}$, and the rank of D_k is denoted as **J** . For user k, in order to obtain the maximum precoding gain, the second SVD operation is implemented on the effective channel matrix as

$$\mathbf{D_k = U_k \Lambda_k V_k^H = U_k \Lambda_k [\, V_k^{(1)}\;\; V_k^{(0)}\,]^H,} \tag{12}$$

where $\mathbf{U_K}$ and $\mathbf{V_K}$ are unitary matrices, $\mathbf{\Lambda_k}$ is the diagonal matrix containing the singular values of Dk. $\mathbf{V_k}^{(1)}$ consists of the first **J** singular vectors of $\mathbf{V_K}$. The second precoding matrix for user k of the BD precoding algorithm is obtained as

$$\mathbf{F_k = V_k}^{(1)} \tag{13}$$

After traversing all the users, the second precoding matrix for k users is expressed as

$$\mathbf{F = [\, F_1\, , \ldots, F_{k-n}, \ldots, F_k]} \tag{14}$$

From the above two steps, the overall precoding matrix is expressed as **W=E F**

QR-BD Precoding Algorithm:

The QR-BD precoding algorithm utilizes pseudo-inverse and QR decomposition to find orthonormal basis for the null space of $\mathbf{G_k}$ in an alternative way.QR-BD precoding algorithm will be implemented in two steps.[4]

Step 1:

Obtain the first precoding matrix **E** to mitigate the MUI.

The QR-BD precoding algorithm first calucates the pseudo-inverse of the matrix $\mathbf{G_k}$ for each user as

$$\mathbf{P_k = G_k^H (G_k\, G_k^H)^{-1}} \qquad \textbf{(15)}$$

where $\mathbf{P_k}$ is the pseudo-inverse matrix.

Then, the QR decomposition is implemented on $\mathbf{I\text{-}P_kG_k}$ **as**

$$\mathbf{I\text{-}P_kG_k = Q_k\, R_k = [\ Q_k^1\ \ Q_k^2\ ] \begin{bmatrix} R_K^1 \\ R_K^2 \end{bmatrix}} \ , \qquad \textbf{(16)}$$

where $\mathbf{R}_K^1 \in C^{(N(T) - B(k))\ X\ N(T)}$ is an upper triangular matrix, $\mathbf{Q_k}^1 \in C^{N(T)\ X(N(T) - B(k))}$ is an orthogonal matrix whose columns form an orthonormal basis for $\mathbf{I - P_k\, G_k}$ **,** B_k is the rank of $\mathbf{G_k}$ **.**

Since $\mathbf{G_k(I\text{-}P_kG_k)=0}$**,** then $\mathbf{G_k\, Q_K^1\ R_K^1=0}$. As $\mathbf{R}_K^1$ is invertible, the equation $\mathbf{G}_K\ \mathbf{Q}_K^1=0$ holds true. Thus, $\mathbf{Q}_K^1$ forms an orthonormal basis for the null space of $\mathbf{G}_K^1$.

The first precoding matrix for user k of the QR-BD precoding algorithm is obtained as

$$\mathbf{E_k = Q_K^1} \qquad \textbf{(17)}$$

After traversing all the users, the first precoding matrix for K users is

$$\mathbf{E=[\ E_1\ , \ldots, E_{k\text{-}n}, \ldots, E_k\ ].} \qquad \textbf{(18)}$$

Step 2:

Obtain the second precoding matrix **J** to parallelize each user's streams and achieve the maximum precoding gain for each user.

Similar to the BD precoding algorithm, the second precoding matrix for QR-BD precoding algorithm is obtained by implementing the SVD operation on the effective channel matrix as

$$\mathbf{G_k = U_k\, \Lambda_k V_k^H\ = U_k\, \Lambda_k\, [\ V_k^{(1)}\ \ V_k^{(0)}\ ]^H} \qquad \textbf{(19)}$$

the rank of $\mathbf{G_k}$ is denoted as D_k **.**

The second precoding matrix **F** for user k of the QR-BD precoding algorithm is obtained as

$$\mathbf{F_k = V_k^{(1)}}\ , \qquad \textbf{(20)}$$

where $\mathbf{V_k}^{(1)}$ consists of the first **J** singular vectors of $\mathbf{V_k}$ **.** After traversing all the users, the second precoding matrix **F** for K users is

$$\mathbf{F = [\ F_1\ , \ldots, F_{k\text{-}n}, \ldots, F_k]} \qquad \textbf{(21)}$$

From the above two steps, the overall precoding matrix is expressed as **W=E F .**

The idea of the paper is to find a way to obtain the same output of the above algorithms while skipping some complex steps that consumes too much computation or simply that consumes too much cpu.

The above two problems will be fine as long as the number of users(transmitters and recievers) remain small.But in reality the number of users will be in hundreds of millions.Thus there will be lag in processing leading to inefficiency.

Report on present investigation:

Proposed QR-GSO Precoding Algorithm

In this section, we describe the proposed QR-GSO precoding aldorithm, where an alternative way of computing the null space of $\mathbf{G_k}$ is introduced by using the improved QR decomposition and Gram Schmidt Orthonormalization method.

This is also implemented in two steps.

Step 1:

Obtain the first precoding matrix **E** to mitigate the MUI.

In order to obtain **E** , we implement the QR decomposition on the channel matrix

$$\mathbf{T}^{H} = \mathbf{QR} \tag{22}$$

where $\mathbf{Q} \in \mathbb{C}^{N(T) \text{ X } N(R)}$ is an orthogonal matrix and $\mathbf{R} \in \mathbb{C}^{N(R) \text{ X } N(R)}$ is an upper triangular matrix. Thus, the combined channel matrix **T** can be reprensted as

$$\mathbf{T=LQ}^{H} \tag{23}$$

where $\mathbf{L=R}^{H}$ **.**

Then the pseudo-inverse of the combined channel matrix **T** is calculated in an alternative way as

$$\begin{aligned} \mathbf{P1} &= \mathbf{T}^{H} (\mathbf{T}\,\mathbf{T}^{H})^{-1} = \mathbf{QR} (\mathbf{LQ}^{H}\, \mathrm{QR})^{-1} \\ &= \mathbf{QL}^{-1} = \mathbf{Q}\,[\,\mathbf{L}_1 \ldots \mathbf{L}_{k-n} \ldots\ldots \mathbf{L_k}\,] \end{aligned} \tag{24}$$

where **P1** is the pseudo-inverse of the matrix **T**, and $\mathbf{L_K} \in \mathbb{C}^{N(T) \text{ X } N(K)}$ is the sub-matrix of $L^{-1} \in \mathbb{C}^{N(T) \text{ X } N(R)}$ for user k. Since **(P1) T=I,** we have $\mathbf{G}_k\mathbf{QL}_K = 0$. Therefore, $\mathbf{QL}_K$ is the null space vector of MUI.

In order to reduce the computational complexity, were use a simplified method to obtain the inverse of **L**. Since **L** is a lower triangle matrix, we transform **L** into a unit lower triangle matrix **A** by multiplying a diagonal matrix **M**, whose diagonal elements are the reciprocal of **L**'s,

$$\mathbf{A = ML} \tag{25}$$

Then we calculate the inverse of **A** in a simplifies way as

$$A^{-1} = \sum_{i=0}^{N_R-1} (-K)^{i} \tag{26}$$

where **K=A-I**, **I** is an $N_R \times N_R$ unit matrix. Then we can obtain the inverse of matrix **L** as

$$\mathbf{LL}^{H} = \mathbf{A}^{-1}\mathbf{M} = [\,\mathbf{L_1} \ldots \mathbf{L}_{k-n} \ldots\ldots \mathbf{L_k}\,] \tag{28}$$

It is worth nothing that $\mathbf{Q}\ \mathbf{L}^{o}_{k}$ is not an orthonormal basis, we should find the orthonormal basis of $\mathbf{QL}_K$ as the precoding matrix for user k. Thus, the Gram Schmidt Orthonormalization method is applied to find the orthonormal basis $\mathbf{L}^{o}_{k}$ for $\mathbf{L}_k$ as

$$\mathbf{L}^{o}_{k} = \mathrm{GSO}(\mathbf{L}_{k}) \qquad \textbf{(29)}$$

Since **Q** is an orthogonal matrix, the columns of $\mathbf{Q}\ \mathbf{L}^{o}_{k}$ form an orthonormal basis for the null space of $\mathbf{G}_k$. Hence, the first precoding matrix $\mathbf{E}_k$ for user k of the proposed QR-GSO precoding algorithm is given by

$$\mathbf{E}_k = \mathbf{Q}\mathbf{L}^{o}_{k} \qquad \textbf{(30)}$$

After traversing all the users, the first precoding matrix **E** for K users is expressed as

$$\mathbf{E} = [\ \mathbf{E}_1\ , \ldots, \mathbf{E}_{k\text{-}n}, \ldots, \mathbf{E}_k\ ] \qquad \textbf{(31)}$$

Step 2:

Obtain the second precoding matrix **F** to parallelize each user's streams and achieve the maximum precoding gain for each user.

After the first precoding operation, the effective channel matrix $\mathbf{D}_k$ for user k is formulated as

$$\mathbf{D}_k = \mathbf{H}_k\mathbf{E}_k \qquad \textbf{(32)}$$

Similar to the BD-type precoding, the second precoding matrix for QR-GSO algorithm is obtained by implementing the SVD operation on the effective channel matrix as

$$\mathbf{D}_K = \mathbf{U}_k\, \mathbf{\Lambda}_k \mathbf{V}_k^{H} = \mathbf{U}_k\, \mathbf{\Lambda}_k\, [\ \mathbf{V}_k^{(1)}\ \ \mathbf{V}_k^{(0)}\ ]^{H} \qquad \textbf{(33)}$$

the rank of $\mathbf{D}_K$ is denoted as **J** .

The second precoding matrix $\mathbf{F}_k$ for user k is obtained as

$$\mathbf{F}_k = \mathbf{V}_k^{(1)}$$

where $\mathbf{V}_k^{(1)}$ consists of the first **J** singular vectors of $\mathbf{V}_k$**.** After traversing all the users, the second precoding matrix **F** for K users is

$$\mathbf{F} = [\ \mathbf{F}_1\ , \ldots, \mathbf{F}_{k\text{-}n}, \ldots, \mathbf{F}_k\ ] \qquad \textbf{(34)}$$

From the above two steps, the overall precoding matrix is expressed as **W=E F**

Results and discussions:

In order to quantify the computational complexity and compare it between our proposed algorithm with the conventional methods we make the use of FLOPS (Floating Point Operations) .

Complexity can be expressed in terms of floating-point operations or flops required to find the solution, expressed as a function of the problem parameter. A Flop serves as a basic unit of computation.

Flops of some operations are mentioned below.[6]

- Multiplication an m x n complex matrix by an (n x p)complex matrix: 8mnp -2mp .
- QR decomposition to an m x n (m <= n) complex matrix: 16(n2m-nm2 +m3/3) .
- SVD to an m x n (m <= n) complex matrix, where only V is obtained: 32 ($nm^2 + 2m^3$)
- SVD to an m x n (m <= n) complex matrix, where U and V are both obtained: 8($4n^2m + 8nm^2 + 9m^2$)
- GSO to an m x n (m >= n) complex matrix: $8n^2 (m-n/3)$.

- Inversion of an m x m complex matrix using Gauss-Jordan elimination: $16m^3$

Table 1 to 3 conatins the information about the total number of flops required to compute different steps in respective algorithms.

For simplicity let us assume k=10 , N_k=2, $N_R = N_T = KN_K = 20$,$A = N_R - N_K = 18$ in examples to calculate total number of flops required to perform the respective operations.

Table 1: - Computational Complexity of BD

OPERATIONS	FLOPS	EXAMPLE
$\mathbf{U_k \Lambda_k V_k^H}$	$\mathbf{32 (N_T A^2 + 2 A^3)}$	580608
$\mathbf{H_S E}$	$\mathbf{8 N_k^2 N_T - 2 N_k^2}$	632
$\mathbf{U_k \Lambda_k V_k^H}$	$\mathbf{8 (4 N_T^2 N_k + 8 N_k^2 N_T + 9 N_K^3)}$	31296
E F	$\mathbf{8 N_R^2 N_T - 2 N_R N_T}$	63200 **Total = 6 ,75 ,736 flops**

Table 2: - Computational Complexity of QR-BD

OPERATIONS	FLOPS	EXAMPLE
$\mathbf{G_k G_k^H}$	$\mathbf{8 A^2 N_T - 2 A^2}$	51192
$\mathbf{(G_k G_k^H)^{-1}}$	$\mathbf{16 A^3 / 3}$	31104
$\mathbf{G_k (G_k G_k^H)^{-1}}$	$\mathbf{8 A^2 N_T - 2 A N_T}$	51120
$\mathbf{I - P_k G_k}$	$\mathbf{8 A^2 N_T - 2 A^2}$	51192

$Q_k R_k$	**$16\,(N^2_T N_K - N_T N_K^2 + N_K^3/3)$**	11563
$G_K\, Q_K^1$	**$8N_K^2 N_T - 2\, N_K^2$**	108096
$U_k \Lambda_k V_k^H$	**$8\,(4\, N^2_T N_K + 8\, N_T N_K^2 + 9\, N_K^3)$**	63200 **Total=3 ,68 ,099 flops**

Table 3: - Computational Complexity of QR-GSO

OPERATIONS	FLOPS	EXAMPLE
QR	**$16\,(N_R^2 N_T - N_R N_T^2 + N_T^3 / 3)$**	42666
L^{-1}	**$3\, N_R + 2\, N_R^2$**	860
$GSO(L_K)$	**$8\, A^2 N_T - 2\, A\, N_T$**	51120
QL^o_k	**$8\, A^2 N_T - 2\, A^2$**	51192
$H_k E_k$.	**$16\,(N^2_T N_K - N_T N_K^2 + N_K^3/3)$**	11563
U_k ΛkVkH	**$8\,(4\, N^2_T N_K + 8\, N_T N_K^2 + 9\, N_K^3)$**	63200
E F	**$8\, N_R^2 N_T - 2\, N_R N_T$**	31296 **Total= 2 ,51 ,897 flops**

For simulation the computaional complexity expressed in the notion if Flops is plotted as a function of K.The resulting graph was obtained using MatLab tool for the code used see Appendix.

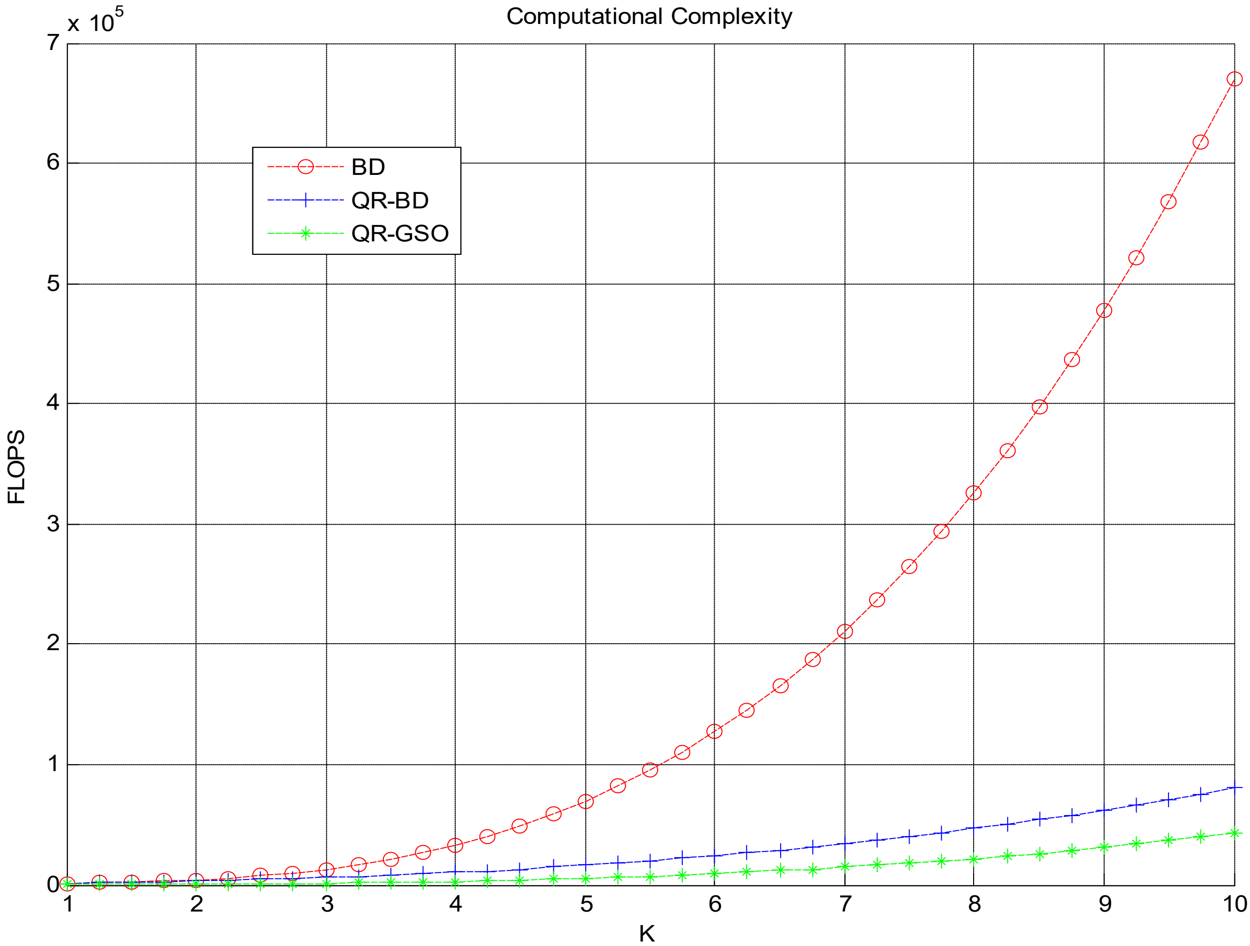

Fig. 4: graph of Flops vs K for BD, QR-BD , and the proposed QR-GSO.

The paper contributes to the speed and efficiency of MU-MIMO to increase.It is through small tweeks like this and employing new algorithms in every step that makes a system fast and reliable in more practical conditions.

The method of MU-MIMO allows the exchange of multiple data streams between multiple users.All kind of communications employ this method be it satelite comunication, live broadcasting, live streaming,gaming,wifi routers etc.. where the number of users are significantly large . Any means to make the method more fast and less complex is encouraged and important.

Summary and Conclusions:

1. It is clear from the flops consumed by each algorithm that the proposed QR-GSO is way less complex and requires less computation.

2. From the example considered the total number of flops for BD was 6,75,763 ,for QR-BD it was 3,68,009 for the proposed QR-GSO 2, 51 ,875.
   i .. The difference between BD and QR-GSO is 4,23,888 flops.
   ii.. The difference between QR-BD and QR-GSO is 1,16,134 flops.

3. The above point indicates that the proposed QR-GSO algorithm requires 4 ,23, 888 less floating point operations or less computation than the conventional BD precoding algorithm considered for 10 users. QR-GSO also requires 1 ,16,134 less floating point operations than the QR-BD algorithm.These numbers mean that the QR-GSO algorithm is way less complex and performs well when the number of users become significantly large.

4. From the graph obtained by simulating the flops function as the function of number of users using MatLab, The complexity of the conventional BD precoding algorithm increases monotonically as the number of users.
   The increase in the complexity of less conventional QR-BD algorithm is not as rapid or sudden when compared to BD but nevertheless still more then QR-GSO.
   The proposed QR-GSO algorithm's comlexity is the least of three and its increase is less when compared to the BD and QR-BD algorithms.

5. The fact the output of all the three algorithms are same and the QR-GSO being the least complex makes the proposed algorithm more efficient ,more speed ,more realible in practical applications.

The advancement wireless technology is vast.The trend started with the first generation 1g technology with maximum speed 2.4Kbps . Then came the single input single output method second generation 2g technology.With the speed 384Kbps.Then third generation single user MIMO (SU-MIMO) 3g technology along with 3g+ with speeds of about 21Mbps.Currently is the era of fourth generation MU-MIMO technology 4g,LTE, LTE+ technology with speeds up to 100Mbps.The era of 5g Massive MIMO has just started with the estimated speeds 10Gbps.Better algorithm like this serves as a pavement for these technology to develop.[7]

References:

[1][2] www.wikipedia.org

[3] Spencer, Q.H., Swindlehurst, A.L. and Haardt, M. (2004) Zero-Forcing Methods for Downlink Spatial Multiplexing in Multiuser MIMO Channels. IEEE Transactions on Signal Processing

[4] Chen, R., Robert, W.H. and Andrews, J.G. (2007) Transmit Selection Diversity for Unitary Precoded Multiuser Spatial Multiplexing Systems with Linear Receivers. IEEE Transactions on Signal Processing, 55,1159-1171.

[5] Choi, L.U. and Murch, R.D. (2004) A Transmit Preprocessing Technique for Multiuser MIMO Systems Using a Decomposition Approach. IEEE Transactions on Wireless Communications, 3, 20-24.

[6] Golub, G.H. and Loan, C.F.V. (2013) Matrix Computations. 4th Edition, The Johns Hopkins University Press.

[7] Noha Hassan and Xavier Fernando (2017) Massive MIMO wireless networks an overview.

Appendix:

MatLab code:

```
clc;clear all;close all;
k=1:0.1:10;  %%variable
Nk=2;
Nr=Nk.*k;
A=(Nr-Nk);

%%BD flops
BD=32*(Nr.*(A.^2)+2*(A.^3))...
  +8*4.*Nr - 8...
  +8*(4*(Nr.^2)*2 + 8*4.*Nr + 9*8)...
  + 8*(Nr.^3 -2 *Nr.^2);

%%QRBD flops
QRBD=8*(A.^2) - 2*(A.^2)...
    + ((16* A.^3)/3)...
    + 8*(A.^2) - 2.*A.*Nr...
    +8*(A.^2) - 2*(A.^2)...
    + 16*((Nr.^2)*2 - 4*Nr +(8/3))...
    +8*4*Nr -8...
    +8*(4*(Nr.^2)*2  +32*Nr +9*8);

%%QRGSO flops
QRGSO=(16*((Nr.^3)/3))...
   +3 *Nr +2*(Nr.^2)...
   +8*(A.^2).*Nr-2.*Nr.*A...
   +8*(A.^2).*Nr-2*(A.^2)...
   +16*((Nr.^2)*2 - 4*Nr +(8/3))...
   +8*(4*(Nr.^2)*2  +32*Nr +9*8)...
   +8*Nr.^3-2*Nr.^2;

%%Plot all three
hold on;
grid on;
plot(k,BD,'r--o');
plot(k,QRBD,'--+');
plot(k,QRGSO,'g--*');
legend('BD','QR-BD','QR-GSO');
```