Iterative Superlinear-Convergence SVD Beamforming Algorithm and VLSI Architecture for MIMO-OFDM Systems

Cheng-Zhou Zhan, Yen-Liang Chen, *Student Member, IEEE*, and An-Yeu Wu, *Member, IEEE*

***Abstract*—In this paper, we propose a singular value decomposition (SVD) algorithm with superlinear-convergence rate, which is suitable for the beamforming mechanism in MIMO-OFDM channels with short coherent time, or short training sequence. The proposed superlinear-convergence SVD (SL-SVD) algorithm has the following features: 1) superlinear-convergence rate; 2) the ability of being extended smaller numbers of transmit and receive antennas; 3) being insensitive to dynamic range problems during the iterative process in hardware implementations; and 4) low computational cost. We verify the proposed design by using the VLSI implementation with CMOS 90 nm technology. The post-layout result of the design has the feature of 0.48 mm^2 core area and 18 mW power consumption. Our design can achieve 7 M channel-matrices/s, and can be extended to deal with different transmit and receive antenna sets.**

***Index Terms*—Beamforming, multiple-input multiple-output (MIMO)-orthogonal frequency division multiplexing (OFDM), precoding, singular value decomposition (SVD), superlinear.**

I. Introduction

THE demand of high-throughput wireless transmissions, such as IEEE 802.11n WLAN systems and IEEE 802.16e WiMAX systems, continues to grow. The antenna arrays at both transmitter and receiver construct multiple-input multiple-output (MIMO) transceivers to enhance the data throughput significantly. In MIMO orthogonal frequency division multiplexing (MIMO-OFDM) wireless systems, the data streams can be demultiplexed into several substreams transmitted by different antennas to improve the bit-error-rate (BER) or throughput performance of the overall communication system by utilizing the transmit diversity.

The singular value decomposition (SVD) of the channel matrix in MIMO-OFDM system is proved to be able to derive the singular vector matrix for optimum linear precoding and linear receivers [1]. In modern MIMO-OFDM communication systems with high-throughput requirement, such as IEEE 802.11n, the time interval of sending the precoding matrix to the transmitter is specified [2]. In the critical case, we have only 46 μs to derive the precoding matrices of 128 subchannels [2]. In other words, the time for the SVD of one complex matrix is limited to about 400 ns. When the channels have short coherent time, the information derived by SVD should be sent from the receiver to the transmitter as soon as possible to keep the beamforming performance. The decomposing time and accuracy will therefore greatly affect the beamforming performance.

The right singular vector matrix derived from the SVD results of the channel matrix is the optimal precoding matrix for linear detectors such as zero-forcing (ZF) and minimum mean square error (MMSE) detectors [3]. There have been researches about the SVD algorithms for MIMO-OFDM applications. Traditional power iterative algorithm [4] can also be used to solve the SVD problem. However, it has only linear-convergence rate. It will be much slower when the channel matrix has multiple similar singular values. An algorithm of updating the singular vectors of the channel matrix by periodic pre- and post-multiplication by Jacobi rotation matrices was proposed in [5] with high computational cost. In [6], the authors proposed an adaptive SVD algorithm with practical hardware implementations in [7] for MIMO applications without *channel state information (CSI)*. Nevertheless, their convergence time requires hundreds of samples per channel matrix. The disadvantage of long convergence time is not suitable for MIMO channels with short coherent time or short training sequence. Another adaptive SVD beamforming algorithm with perturbation theory was also proposed in [8]. Nevertheless, the computational cost is also high. The algorithm in [8] with iterative division will apparently cause performance degradation in practical hardware implementations with severe quantization effect.

In [9], a hardware-efficient SVD algorithm VLSI architecture for steering matrix computation was proposed. It utilizes bidiagonalization, diagonalization, and Givens rotation to achieve high processing throughput. The resulting VLSI implementations with 0.18 μm technology requires 3.3 μs to complete the SVD of one complex matrix, which is still more than 8 times the critical requirement (i.e., 400 ns) in IEEE 802.11n systems. In addition, the algorithms mentioned above have only linear convergence speeds. Hence, these algorithms may not be suitable for the MIMO channels with short coherent time, or short response time requirement with the specification in the MIMO-OFDM systems.

In this work, we propose a superlinear-convergence SVD (SL-SVD) algorithm and architecture with four features. 1) The

Manuscript received January 17, 2011; revised August 30, 2011 and December 16, 2011; accepted February 16, 2012. Date of publication March 08, 2012; date of current version May 11, 2012. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Warren J. Gross.

The authors are with the Graduate Institute of Electronics Engineering, and Department of Electrical Engineering, National Taiwan University, Taipei, 106, Taiwan, R.O.C. (e-mail: chenjo@access.ee.ntu.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2012.2190405

property of superlinear-convergence rate makes it at least 25 times faster than the referenced works. 2) The ability of being extended to smaller numbers of transmit and receive antennas without hardware overhead. 3) The proposed SL-SVD is insensitive to the dynamic range problems during the iterative process. Only 10-bit precision is required with the system simulation in the IEEE 802.11n systems. It leads to small area, short critical path, and over five times better normalized area efficiency in VLSI implementations compared with related works. 4) The comparison of the computational cost in Section IV shows the proposed SL-SVD to have at least 25% complexity reduction compared with other algorithms of [7] and [9]. At last, we implement the hardware of the SL-SVD beamforming algorithm in 90 nm technology. The chip has the feature of 0.48 mm^2 core area and 18 mW power consumption. It not only achieves 7 M channel-matrices/s, 140 ns per matrix equivalently, which satisfies the critical specification of 400 ns per matrix in the IEEE 802.11n systems. In addition, the proposed SL-SVD design is also able to be extended to deal with different transmit and receive antenna sets. Besides, the postlayout simulation is also verified by commercial electric design automation (EDA) tools.

The paper is organized as follows. The system model is described in Section II, and the details of the operation of the proposed SL-SVD algorithm are presented in Section III. The algorithm analysis is described in Section IV. The simulation, architecture design, and VLSI implementation results are presented in Sections V, VI, and VII, respectively.

II. System Model

Consider a wireless MIMO-OFDM system in a frequency nonselective, slowly fading channel, respectively. Suppose N_t and N_r antennas are used at the transmitter and receiver. The equivalent channel model is given by

$$\mathbf{r} = \mathbf{H}\mathbf{s} + \mathbf{n}. \tag{1}$$

$\mathbf{H} \in \mathbb{C}^{N_r \times N_t}$ is the complex channel matrix with the (p, q)th element which is the random fading between the pth receive and qth transmit antennas. $\mathbf{n} \in \mathbb{C}^{N_r \times 1}$ is the additive noise source and is modeled as a zero-mean, circularly symmetric, complex Gaussian random vector with statistically independent elements. The pth element of $\mathbf{s} \in \mathbb{C}^{N_t \times 1}$ is the symbol transmitted at the pth transmit antenna, and that of $\mathbf{r} \in \mathbb{C}^{N_t \times 1}$ is the symbol received at the pth received antenna.

After deriving the CSI, we can decompose the channel matrix in the SVD form as follows:

$$\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}}. \tag{2}$$

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1 & & & \\ & \ddots & & \mathbf{0} \\ & & \sigma_t & \\ & \mathbf{0} & & \mathbf{0} \end{bmatrix}_{N_r \times N_t}, \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_t \geq 0 \tag{3}$$

$$\mathbf{U} = [\mathbf{u}_1 \quad \mathbf{u}_2 \quad \ldots \quad \mathbf{u}_{N_r}] \tag{4}$$

$$\mathbf{V} = [\mathbf{v}_1 \quad \mathbf{v}_2 \quad \ldots \quad \mathbf{v}_{N_t}]. \tag{5}$$

$\mathbf{U}$ is an $N_r \times N_r$ unitary matrix, $\mathbf{V}$ is an $N_t \times N_t$ unitary matrix, and $t = min(N_r, N_t)$. $\mathbf{u}_i$'s and $\mathbf{v}_i$'s are the corresponding left and right singular vectors. $\mathbf{\Sigma}$ is an $N_r \times N_t$ matrix with only nonnegative main diagonal entries which are the nonnegative square roots of the eigenvalues of $\mathbf{H}^{\mathrm{H}}\mathbf{H}$, and $(\cdot)^{\mathrm{H}}$ denotes the Hermitian operation. In (2), the diagonal matrix $\mathbf{\Sigma}$ is unique for a given channel matrix while the unitary matrices $\mathbf{U}$ and $\mathbf{V}$ are not unique matrices. By substituting the SVD results for the matrix $\mathbf{H}$, (1) becomes

$$\mathbf{r} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}}\mathbf{s} + \mathbf{n}. \tag{6}$$

Multiplying $\mathbf{U}^{\mathrm{H}}$ on both sides, (6) can be rewritten as

$$\mathbf{r}' = \mathbf{\Sigma}\mathbf{s}' + \mathbf{n}' \tag{7}$$

where $\mathbf{r}' = \mathbf{U}^{\mathrm{H}}\mathbf{r}$, $\mathbf{s}' = \mathbf{V}^{\mathrm{H}}\mathbf{s}$, and $\mathbf{n}' = \mathbf{U}^{\mathrm{H}}\mathbf{n}$. Note that the distribution of $\mathbf{n}'$ is invariant under unitary transformation. It means that the multiplication of the AWGN by a unitary matrix does not cause any noise enhancement. The multiantenna channel is equivalent to $min(N_t, N_r)$ independent parallel Gaussian subchannels at most. Each subchannel has a gain, which is the singular value of the channel matrix $\mathbf{H}$.

III. The Proposed SL-SVD Algorithm

Our goal is to develop an iterative SVD algorithm with high convergence rate with acceptable computational cost. Most iterative SVD algorithms try to reduce the computational cost in each iteration, however the number of required iteration times is enlarged. If we can greatly reduce the entire computation time by increasing moderate computational cost in each iteration, the overall computational cost which can be lowered with even higher convergence rate. The proposed SL-SVD has the property of superlinear-convergence rate and the detailed procedures are described in the following subsections.

A. Initial Stage and Iterative Process

To handle MIMO-OFDM channels with short coherent time or short training sequence, we propose a superlinear-convergence SVD (SL-SVD) algorithm for closed-loop beamforming. From (2), the results of the SVD process consist of singular values and singular vectors. The main idea of the proposed SL-SVD algorithm is to derive the singular vectors prior to singular values. Deriving singular vectors first has a significant advantage that we only have to care about the direction of the singular vector but not the norm.

In the proposed SL-SVD algorithm, we do not compute the decomposition directly. Instead, we derive the direction of the right singular vectors by iterative computation. The convergence rate is enhanced by using the matrix multiplications iteratively. At the same time, we apply the proposed adaptive binary shift mechanism to prevent the growth of the dynamic range during the iterative multiplication. Unlike the traditional power iteration method [4], adaptive method [8] and [6], this work provides higher convergence rate of deriving the results of SVD, and needs only 10-bit precision for the variables during the iterative computation in our simulations in Section V.

To simplify the SVD problem from three unknown matrices, $\mathbf{U}$, $\mathbf{\Sigma}$, and $\mathbf{V}$, to two unknown matrices, we firstly define the initial matrix $\mathbf{P}_1^{(0)}$

$$\mathbf{P}_1^{(0)} = k_{1,0}\cdot\mathbf{H}^{\mathrm{H}}\mathbf{H} = k_{1,0}\cdot\mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^{\mathrm{H}} = k_{1,0}\cdot\sum_{i=1}^{N_t}\sigma_i^2\mathbf{v}_i\mathbf{v}_i^{\mathrm{H}} \quad (8)$$

$\mathbf{P}_i^{(n)}$ and $k_{i,n}$ are the updating matrix and arbitrary non-zero coefficients after the nth iteration of the proposed algorithm for deriving the ith singular vector $\mathbf{v}_i$. The value of the maximum iteration number, n, can be defined in advance. We only have to solve two unknown matrices, $\mathbf{\Sigma}$ and $\mathbf{V}$, in (8).

By applying n times self-multiplication on $\mathbf{P}_1^{(0)}$, we have

$$\begin{aligned}\mathbf{P}_1^{(n)} &= k_{i,n}\cdot\mathbf{P}_1^{(n-1)}\times\mathbf{P}_1^{(n-1)} = k_{i,n}\cdot\mathbf{V}\mathbf{\Sigma}^{2^{n+1}}\mathbf{V}^{\mathrm{H}}\\ &= k_{i,n}\cdot\sum_{i=1}^{N_t}\sigma_i^{2^{n+1}}\mathbf{v}_i\mathbf{v}_i^{\mathrm{H}}\\ &= k_{i,n}\cdot\Sigma_{i=1}^{N_t}\sigma_i^{2^{n+1}}\mathbf{v}_i\mathbf{v}_i^{\mathrm{H}}. \end{aligned}\quad (9)$$

Then, $\mathbf{P}_1^{(n)}$ can be rewritten as

$$\begin{aligned}\mathbf{P}_1^{(n)} &= k_{i,n}\cdot\mathbf{P}_1^{(n-1)}\times\mathbf{P}_1^{(n-1)} = k_{i,n}\cdot\mathbf{V}\mathbf{\Sigma}^{2^{n+1}}\mathbf{V}^{\mathrm{H}}\\ &= k_{i,n}\cdot\sum_{i=1}^{N_t}\sigma_i^{2^{n+1}}\mathbf{v}_i\mathbf{v}_i^{\mathrm{H}} \end{aligned}\quad (10)$$

where n denotes the nth iterative multiplication. The gap between the component, $\sigma_1^{2^{n+1}}\mathbf{v}_1\mathbf{v}_1^{\mathrm{H}}$, corresponding to the greatest singular value, σ_1, is significantly amplified. In other words, the gaps between $\sigma_1^{2^{n+1}}\mathbf{v}_1\mathbf{v}_1^{\mathrm{H}}$ and other components, $\{\sigma_j^{2^{n+1}}\mathbf{v}_j\mathbf{v}_j^{\mathrm{H}}, j \neq 1\}$, corresponding to smaller singular values, are enlarged after each iterative matrix multiplication. The convergence rate defined in [4] is an index to evaluate the convergence speed of the iterative approximation algorithms. and the convergence rate of the proposed SL-SVD is proportional to $O\left(\left|\frac{\sigma_1}{\sigma_2}\right|^{2^{n+1}}\right)$. That is to say, the proposed algorithm has superlinear-convergence rate which is demonstrated in Section IV-A. If n is great enough (chosen to be 4 for 4×4 channel matrices according to the simulation in Section IV), we can have the following approximation:

$$\begin{aligned}\mathbf{P}_1^{(n)} &= \left(\mathbf{P}_1^{(0)}\right)^{2^n}\\ &= \prod_{j=1}^{n}k_{1,n}\cdot\sum_{i=1}^{N_t}\sigma_i^{2^{n+1}}\mathbf{v}_i\mathbf{v}_i^{\mathrm{H}}\\ &\approx \prod_{j=1}^{n}k_{1,n}\cdot\boldsymbol{\sigma}_1^{2^{n+1}}\mathbf{v}_1\mathbf{v}_1^{\mathrm{H}}. \end{aligned}\quad (11)$$

The first column vector of $\mathbf{P}_1^{(n)}$, says $\mathbf{P}_1^{(n)}(:,1)$, can be normalized to be $\tilde{\mathbf{v}}_1$ such that

$$\tilde{\mathbf{v}}_1 = \frac{\mathbf{P}_1^{(n)}(:,1)}{\left\|\mathbf{P}_1^{(n)}(:,1)\right\|_2} \approx \mathrm{e}^{\mathrm{j}\phi}\mathbf{v}_1 \quad (12)$$

where $\tilde{\mathbf{v}}_1$ is the estimated unit vector of $\mathbf{v}_1$ and ϕ is the phase difference between $\tilde{\mathbf{v}}_1$ and $\mathbf{v}_1$.

Since the solutions of the singular vectors are not unique as mentioned in Section II, we show a sufficient condition to check if the derived vectors are the solutions of right singular vectors, $\mathbf{v}_i$, corresponding to the greatest singular value of the current matrix. With a Hermitian matrix $\mathbf{R}$, we say that a vector $\mathbf{g}$ is the singular vector corresponding to the greatest singular value, σ_{max}, in $\mathbf{R}$ if $\mathbf{g}$ has the two properties of

$$\|\mathbf{g}\|_2 = 1, \quad (13)$$

$$\mathbf{g}^{\mathrm{H}}(\mathbf{R})\mathbf{g} = \sigma_{max}^2. \quad (14)$$

Note that the coexistence of the properties of (13) and (14) is a sufficient condition but not a necessary condition.

From (12), (13), and (14), we can derive

$$\|\tilde{\mathbf{v}}_1\|_2 \approx \|\mathrm{e}^{\mathrm{j}\phi}\mathbf{v}_1\|_2 = 1, \quad (15)$$

$$\tilde{\mathbf{v}}_1^{\mathrm{H}}(\mathbf{H}^{\mathrm{H}}\mathbf{H})\tilde{\mathbf{v}}_1 \approx \left(\mathrm{e}^{\mathrm{j}\phi}\mathbf{v}_1\right)^{\mathrm{H}}(\mathbf{H}^{\mathrm{H}}\mathbf{H})(\mathrm{e}^{\mathrm{j}\phi}\mathbf{v}_1) = \sigma_1^2 \quad (16)$$

where $\sigma_1^2 = \sigma_{max}^2$ which is the square of the greatest singular value of the current matrix, $\mathbf{H}^{\mathrm{H}}\mathbf{H}$.

Equations (15) and (16) show that $\tilde{\mathbf{v}}_1$ has the same property to fulfill the sufficient condition of (13) and (14). In other words, any $\tilde{\mathbf{v}}_1$ with only phase difference from $\mathbf{v}_1$ can be treated as a correct estimation of $\mathbf{v}_1$.

B. The Proposed Adaptive Binary Shift (A.B.S.) Mechanism

The column vector $\mathbf{P}_1^{(n)}(:,1)$ is normalized to obtain $\tilde{\mathbf{v}}_1$ at last no matter what magnitude of $\mathbf{P}_1^{(n)}(:,1)$ is. We can apply re-scaling after each self-multiplication in (9). That is, (10) can be modified as

$$\begin{aligned}\mathbf{P}_1^{(n)} &= \left(\prod_{i=1}^{n}k_{1,i}\right)\sum_{i=1}^{N_t}\sigma_i^{2^{n+1}}\mathbf{v}_i\mathbf{v}_i^{\mathrm{H}}\\ &\approx \left(\prod_{i=1}^{n}k_{1,i}\right)\sigma_1^{2^{n+1}}\mathbf{v}_i\mathbf{v}_i^{\mathrm{H}} \end{aligned}\quad (17)$$

where k_i is the rescaling factor in each iteration. Considering the cost-efficient hardware implementations, we choose k_i to be the power of two so that only binary shifts needed in the rescaling operation. At the beginning of the each iteration, let the rescaling factor, k_i, be $2^{k_i'}$, (17) can be formed into

$$\begin{aligned}\mathbf{P}_1^{(n)} &= \left(\prod_{i=1}^{n}2^{k_i'}\right)\sum_{i=1}^{N_t}\sigma_i^{2^{n+1}}\mathbf{v}_i\mathbf{v}_i^{\mathrm{H}}\\ &\approx \left(2^{\Sigma_{i=1}^{n}k_i'}\right)\sigma_1^{2^{n+1}}\mathbf{v}_i\mathbf{v}_i^{\mathrm{H}}. \end{aligned}\quad (18)$$

We will propose an optimal adaptive binary shift mechanism under the situation of fixed bit number precision and no extra

storage overhead. The optimal k_i' chosen in the nth iteration of deriving the kth singular vector is adaptively set to be

$$k_n' = \underset{k_n'}{arg}\left\{2^{b-2} \le 2^{k_n'} \cdot max\left(abs\left(\left(\mathbf{P}_i^{(n-1)}\right)^2\right)_{p,q}\right) < 2^{b-1}, k_n' \in \mathbb{Z}\right\}$$
$$= \underset{k_n'}{arg}\left\{2^{b-2} \le 2^{k_n'} \cdot max\left(\text{diag}\left(\left(\mathbf{P}_i^{(n-1)}\right)^2\right)\right)\right) < 2^{b-1}, k_n' \in \mathbb{Z}\right\} \quad (19)$$

where $abs(\cdot)_{m,n}$ denotes the absolute value of the (p,q)th entry of the indicated matrix and each element in the matrix is described in b-bit real and imaginary parts. The entry with maximum absolute value is definitely located at the diagonal of the semipositive definite matrices. Any binary shift greater or smaller than k_n' will cause overflow or underflow problems.

In the proposed SL-SVD algorithm, we first derive the unit singular column vectors, $\mathbf{v}_i$, of $\mathbf{V}$ instead of finding out correct σ_i. The advantage of deriving $\mathbf{v}_i$ first is that we only have to keep the direction of the singular vectors during iterations, which makes us be free from the problem of dynamic range caused by the explosively growing σ_i during the iterative self-multiplications by applying the proposed adaptive binary shift mechanism. This property is suitable for the cost-effective hardware implementations.

C. Deflation

After $\tilde{\mathbf{v}}_1$ is found out, the correlated components of $\tilde{\mathbf{v}}_1$ in $\mathbf{P}_1^{(0)}$ should be eliminated for deriving the next estimated singular vector $\tilde{\mathbf{v}}_2$. The singular vectors $\tilde{\mathbf{v}}_i$'s have two properties as follows:

$$\text{Summation property}: \Sigma_{i=1}^{N_t} \tilde{\mathbf{v}}_i \tilde{\mathbf{v}}_i^{\mathrm{H}} = \mathbf{I}_{N_t} \quad (20)$$
$$\text{and Orthogonal property}: \tilde{\mathbf{v}}_i^{\mathrm{H}} \tilde{\mathbf{v}}_{\mathrm{j}} = 0, \forall i \ne j \quad (21)$$

where $\mathbf{I}_{N_t}$ is an $N_t \times N_t$ identity matrix. By exploiting the two properties, the deflation operation can be performed by

$$\mathbf{P}_2^{(0)} = \left(\mathbf{I}_{N_t} - \tilde{\mathbf{v}}_i \tilde{\mathbf{v}}_i^{\mathrm{H}}\right) \mathbf{P}_1^{(0)} = \left(I_{N_t} - \mathbf{v}_1 \mathbf{v}_1^{\mathrm{H}}\right) \sum_{i=1}^{N_t} \sigma_i^2 \mathbf{v}_i \mathbf{v}_i^{\mathrm{H}} = \sum_{i=2}^{N_t} \sigma_i^2 \mathbf{v}_i \mathbf{v}_i^{\mathrm{H}}. \quad (22)$$

$\mathbf{P}_2^{(1)}$ has no components of σ_1 and $\mathbf{v}_1$ after deflation operation. Without losing generality, the deflation operation can be written as

$$\mathbf{P}_k^{(0)} = \left(\mathbf{I}_{N_t} - \tilde{\mathbf{v}}_{k-1} \tilde{\mathbf{v}}_{k-1}^{\mathrm{H}}\right) \mathbf{P}_{k-1}^{(0)} = \sum_{1=2}^{N_t} \sigma_i^2 \mathbf{v}_i \mathbf{v}_i^{\mathrm{H}}. \quad (23)$$

We can now apply the same self-multiplication process aforementioned to $\mathbf{P}_k^{(0)}$, and $\tilde{\mathbf{v}}_k$ is also found after normalizing one column or row vector of $\mathbf{P}_k^{(n)}$. After deriving $\tilde{\mathbf{v}}_1 \sim \tilde{\mathbf{v}}_{N_t-1}$ by the same iterative process, we then only need to perform Gram-Schmidt (GS) operation [4] to find out the last right singular vector $\tilde{\mathbf{v}}_{N_t}$ due to the orthogonal property. It means the iterative multiplication of deriving the last right singular vector $\tilde{\mathbf{v}}_{N_t}$ is not required, and it reduces the total computational cost.

D. Left Singular Vector and Singular Value Matrix Derivation

After the matrix $\widetilde{\mathbf{V}}$ is derived, we multiply the channel matrix $\mathbf{H}$ with $\widetilde{\mathbf{V}}$

$$\text{let } \mathbf{T} = \mathbf{H}\widetilde{\mathbf{V}} = [\mathbf{H}\tilde{\mathbf{v}}_1\ \mathbf{H}\tilde{\mathbf{v}}_2\ \ldots\ \mathbf{H}\tilde{\mathbf{v}}_{Nt}] = (\mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}})\widetilde{\mathbf{V}} = \widetilde{\mathbf{U}}\widetilde{\mathbf{\Sigma}} \approx \mathbf{U}\mathbf{\Sigma}. \quad (24)$$

Equivalently, the column vectors of $\mathbf{T}, \mathbf{H}\tilde{\mathbf{v}}_i$, can be obtained after deriving each $\tilde{\mathbf{v}}_i$. The estimated singular values and left singular vectors can be derived as

$$\widetilde{\boldsymbol{\sigma}}_i(:,i) = \|\mathbf{T}(:,i)\|_2 \quad (25)$$
$$\widetilde{\mathbf{u}}_i = \widetilde{\mathbf{U}}(:,i) = \frac{\mathbf{T}(:,i)}{\|\mathbf{T}(:,i)\|_2} \quad (26)$$

where $\widetilde{\mathbf{U}}$ and $\widetilde{\mathbf{\Sigma}}$ are the estimated matrix of singular values and the estimated left singular vectors, respectively. By computing the norm of each column in $\mathbf{H} \cdot \mathbf{V}$ and normalizing the column vectors, we then derive $\widetilde{\mathbf{\Sigma}}$ and $\widetilde{\mathbf{U}}$ without any iterative multiplication.

Note that the main computations and storage needed are related to the matrix $\widetilde{\mathbf{V}}$, and only small wordlength required in the iterative multiplication due to the proposed adaptive binary shift mechanism so as to reduce the critical path and the hardware needed at the same time. The computation of $\widetilde{\mathbf{\Sigma}}$ and $\widetilde{\mathbf{U}}$ requires no iterative process and is outside the loop, which indicates we can use greater wordlength to store the values of $\widetilde{\mathbf{\Sigma}}$ and $\widetilde{\mathbf{U}}$ for higher overall accuracy without increasing much hardware overhead or lengthening the critical path.

E. Orthogonality Reconstruction (OR)

In practical hardware implementations, all the elements will be expressed in finite precision. The orthogonal property among singular vectors, column vectors of $\mathbf{U}$ and $\mathbf{V}$, will be corrupted and induce the interferences among transmitted substreams. We will then propose an operation called OR to preserve the most orthogonality. Applying SVD to the channel matrix $\mathbf{H}$, we can learn that

$$\mathbf{\Sigma} = \mathbf{U}^{\mathbf{H}}\mathbf{H}\mathbf{V}. \quad (27)$$

The corruption of orthogonal property among singular vectors will cause nonzero value of off-diagonal entry of diagonal matrix $\mathbf{\Sigma}$. Such nonzero off-diagonal entries will cause interference among each antenna and inaccurate singular values which bring BER degradation. The corruption of orthogonal property among singular vectors should be carefully handled. However, in fixed point design, this property is corrupted by quantization error and inaccurate deflation due to the finite precision. Especially, error propagation induced by deflation stage may cause a fatal error to orthogonal property among singular vectors. Take two singular vectors as an example

$$\tilde{\mathbf{v}}_{\boldsymbol{i}}^{\mathbf{H}} \tilde{\mathbf{v}}_{\boldsymbol{j}} = \varepsilon, \quad i \ne j \quad (28)$$

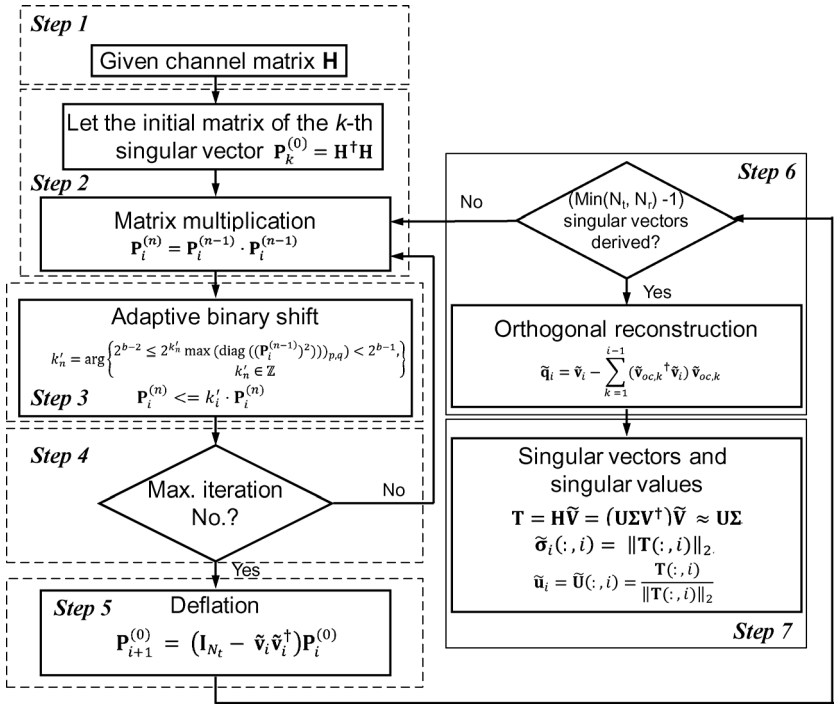


Fig. 1. The flowchart of the proposed superlinear-convergence SVD algorithm.

where $\tilde{\mathbf{v}}_i$ and $\tilde{\mathbf{v}}_j$ are two orthogonal singular vectors. If $\tilde{\mathbf{v}}_i$ and $\tilde{\mathbf{v}}_j$ have perfect orthogonal property, ε should be equal to zero. If the orthogonal property of $\tilde{\mathbf{v}}_i$ and $\tilde{\mathbf{v}}_j$ are destroyed by quantization error, the value of ε is near to the accuracy which fixed point can represent. However, error propagation induced by deflation stage may lead ε become hundreds times of system accuracy. The destruction of orthogonal property among singular vectors caused by quantization error may not be prevented. However, we can use orthogonality reconstruction for fixed point operation to eliminate the destruction caused by deflation stage and improve the performance.

For orthogonality reconstruction, first we consider the data flow in Fig. 1. Notice that $\tilde{\mathbf{v}}_1$ corresponding to the greatest singular value does not suffer from the errors caused by the deflation operation. While $\mathbf{u_i}$'s, for all $i > 1$ are derived, we eliminate the inaccurate remaining part on previously derived singular vectors by applying Gram-Schmidt process respect to $\tilde{\mathbf{v}}_1 \sim \tilde{\mathbf{v}}_{i-1}$. The operation can be expressed as

$$\widetilde{\mathbf{q}}_1 = \tilde{\mathbf{v}}_1 \tag{29}$$

$$\widetilde{\mathbf{q}}_i = \tilde{\mathbf{v}}_i = \sum_{k=1}^{i-1} \left(\tilde{\mathbf{v}}_{oc,k}^{\mathrm{H}} \tilde{\mathbf{v}}_i\right) \tilde{\mathbf{v}}_{oc,k}, \quad \text{for} \quad i \ge 2, \tag{30}$$

$$\tilde{\mathbf{v}}_{oc,i} = \frac{\widetilde{\mathbf{q}}_i}{\|\widetilde{\mathbf{q}}_i\|_2}. \tag{31}$$

After applying orthogonality reconstruction to all column vectors of $\tilde{\mathbf{V}}$, the most interferences caused by inaccurate deflation process can be avoided. In most cases

$$\|\widetilde{\mathbf{q}}_i\|_2 \approx 1 \tag{32}$$

after previous coarse deflation process so that we can simplify (31) to be

$$\tilde{\mathbf{v}}_{or,i} = \frac{\widetilde{\mathbf{q}}_i}{\|\widetilde{\mathbf{q}}_i\|_2} \approx \widetilde{\mathbf{q}}_i \tag{33}$$

to avoid the division. This simplification will shorten the critical path in the hardware implementations.

For the architecture of orthogonality reconstruction, the classical Gram-Schmidt is adopted. We have notices that modified Gram-Schmidt is numerical stable compared with classical Gram-Schmidt before implementations. The classical Gram-Schmidt has less latency in our hardware implementations, and there is no obvious difference for 4×4 matrices with respect to BER performance compared with that of the ideal floating-point SVD in Section V-C. It is still recommended to use the modified Gram-Schmidt for handling greater sizes of matrices to mitigate the errors induced by finite-precision arithmetic. Take a 4×4 matrix as an example, we want to recover the orthogonal property of the last right singular vector, says $\tilde{\mathbf{v}}_4$, we can derive the orthogonality reconstructed vector $\tilde{\mathbf{v}}_{\mathrm{OR},4}$ as

$$\tilde{\mathbf{v}}_{\mathrm{OR},4} = \frac{\widetilde{\mathbf{q}}_4}{\|\widetilde{\mathbf{q}}_4\|_2} \approx \widetilde{\mathbf{q}}_4 = \tilde{\mathbf{v}}_4 - \sum_{k=1}^{3} \left(\tilde{\mathbf{v}}_{\mathrm{OR},k}^{H} \tilde{\mathbf{v}}_4\right) \tilde{\mathbf{v}}_{\mathrm{OR},k} \tag{34}$$

$$= \left[\tilde{\mathbf{v}}_4\ \tilde{\mathbf{v}}_{OR,1}\ \tilde{\mathbf{v}}_{\mathrm{OR},2}\ \tilde{\mathbf{v}}_{OR,1}\right] \left(\begin{bmatrix} \tilde{\mathbf{v}}_4^H \\ -\tilde{\mathbf{v}}_{\mathrm{OR},1}^H \\ -\tilde{\mathbf{v}}_{\mathrm{OR},2}^H \\ \tilde{\mathbf{v}}_{\mathrm{OR},3}^H \end{bmatrix} [\tilde{\mathbf{v}}_4] \right) \tag{35}$$

where $\tilde{\mathbf{v}}_{\mathrm{OR,k}}$ is the orthogonal compensated kth right singular vector. The Gram-Schmidt can be represented by two successive matrix-vector multipliers based on (35).

F. Algorithm Flow and the Architecture

Fig. 1 shows the flow chart of the proposed SL-SVD algorithm in this paper. The detailed steps will be listed as follows,

- Step 1) Given the complex channel matrix $\mathbf{H}$.
- Step 2) Derive the updating matrix $\mathbf{P}_k^{(0)}$ of the right singular vector corresponding kth singular value and perform the matrix multiplication.
- Step 3) Use adaptive binary shift to approach the desired singular vector under the constraint of wordlength precision.
- Step 4) Check if the set maximum iteration number (chosen to be 4 for the worst case of 4×4 matrices according to the simulation results in Section V) is reached or not. Go back to Step 3 if the condition is not satisfied, or else go to Step 5.
- Step 5) Perform the deflation operation.
- Step 6) Check if all singular vectors are solved or not. If not, go to Step 2, otherwise perform OR operation and go to Step 7. (For a 4×4 matrix, only 3 OR operation is required except for the first singular vector.)
- Step 7) Derive the results of $\widetilde{\mathbf{U}}$, $\widetilde{\mathbf{\Sigma}}$, and $\widetilde{\mathbf{V}}$.

IV. Analysis of the Proposed SL-SVD Algorithm

In this section, we will then analyze the properties of the proposed SL-SVD algorithm.

A. Superlinear-Convergence Property

The speed at which a convergent sequence approaches its limit is called the rate of convergence in numerical analysis. Although the speed of convergence does not give any information about the value of each term in the sequence, it implies that fewer iterations are required to achieve the useful approximation for an iterative algorithm if the speed of convergence is higher.

Suppose that the sequence $\{x_k\}$ convergence to the number ξ. We can evaluate the convergence speed of the sequence by

$$\lim_{k\to\infty} \frac{|x_{k+1}-\xi|}{|x_k-\xi|^q} = \mu \quad \text{with} \quad \mu > 0. \tag{36}$$

The greater q indicates the higher convergence rate. Mathematically speaking, the sequence has sublinear, linear, and superlinear-convergence rate if $q < 1$, $q = 1$, and $q > 1$, respectively. The terms in the convergence equation can be described as

$$x_k = \left|\frac{\sigma_1}{\sigma_2}\right|^{2^{k+1}} \tag{37}$$

$$x_{k+1} = \left(\left|\frac{\sigma_1}{\sigma_2}\right|^{2^{k+1}}\right)^2. \tag{38}$$

The total convergence equation will then be written as

$$\lim_{k\to\infty} \frac{|x_{k+1}|}{|x_k|^2} = \lim_{k\to\infty} \frac{\left(\left|\frac{\sigma_1}{\sigma_2}\right|^{2^{k+1}}\right)^2}{\left(\left|\frac{\sigma_1}{\sigma_2}\right|^{2^{k+1}}\right)^2} = 1 \tag{39}$$

where $\xi = 0$ and $q = 2 > 1$. We can then say that the convergence rate of the proposed SL-SVD algorithm is superlinear.

B. Extension to Smaller Sizes of Channel Matrices

In our proposed SL-SVD algorithm, we can handle an $N_r \times N_t$ complex channel matrix, $\mathbf{H}$. In the situation of the MIMO channel with less transmit or receive antennas, we will then obtain an $N_r' \times N_t'$ channel matrix, $\mathbf{H}'$, where $N_t' \leq N_t$ and $N_r' \leq N_r$. All we have to do is just put matrix $\mathbf{H}'$ into the first $\mathrm{N_t'} \times \mathrm{N_r'}$ entries of the original storage space of $\mathbf{H}$, and fill all other unused entries with zero's then the same iterative operations are executed to obtain the results of SVD. Since we have only N_t' right singular vectors to derive, we can terminate the iterative process after deriving the first $(N_t' - 1)$ singular vectors and start to perform the GS operation to obtain the remaining singular vectors and singular values. In conclusion, the proposed SL-SVD algorithm can be easily applied to any smaller complex channel matrices than the original target matrix size with almost no hardware overhead

$$\mathbf{H} = \begin{bmatrix} h_{1,1} & \cdots & h_{1,N_t'} & \\ \vdots & \ddots & \vdots & \mathbf{0} \\ h_{N_r',1} & \cdots & h_{N_r',N_t\prime} & \\ & \mathbf{0} & & \mathbf{0} \end{bmatrix} \tag{40}$$

$$= [\mathbf{u}_1\ \mathbf{u}_2\ \ldots\ \mathbf{u}_{N_r}] \begin{bmatrix} \sigma_1 & & & \\ & \ddots & & \mathbf{0} \\ & & \sigma_t & \\ & \mathbf{0} & & \mathbf{0} \end{bmatrix} \times [\mathbf{v}_1\ \mathbf{v}_2\ \ldots\ \mathbf{v}_{N_t}]. \tag{41}$$

The column number of channel matrix $\mathbf{H}$ equals to the number of transmit antennas and the row number equals to the number of receive antennas. Take the specification in IEEE 802.11n standard for example, the number of transmit antennas and receive antennas are both defined from 1 to 4. In other words, we may have 16 different sizes of channel matrices in IEEE 802.11n system. Design and implementations of specific SVD engine for each size of channel matrix are hardware inefficient. Therefore, we proposed a scheme extending to smaller sizes of channel matrices based on our SVD algorithm. At beginning, we separate the problem into two classifications. One is that the channel matrix is square definite (1×1, 2×2, 3×3, and 4×4 for IEEE 802.11n standard), and the other is that the channel matrix is nonsquare definite (1×2, 1×3, 1×4, 2×1, 2×3, 2×4, 3×1, 3×2, 3×4, 4×1, 4×2, 4×3, and 4×4 for IEEE 802.11n standard).

After zero columns and zero rows are inserted to the original matrix, the SVD operation data flow is totally the same to the $N \times N$ channel matrix and no need for any extra control unit.

We only need to change the value of d (d is equal to M in this case) in the algorithm.

For a nonsquare $N_r' \times N_t'$ complex channel matrix, we take $N_r' \leq N_t'$ in this case, for example

$$\mathbf{H} = \begin{bmatrix} h_{11} & \dots & h_{1N_t'} & & & \\ \vdots & \ddots & \vdots & 0 & \dots & 0 \\ h_{N_r'1} & \dots & h_{N_r'N_t'} & & & \\ & 0 & & 0 & & \\ & \vdots & & & \ddots & \\ & 0 & & & & 0 \end{bmatrix} \tag{42}$$

$$= \begin{bmatrix} \mathbf{u}_1 \, \mathbf{u}_2 \, \dots \, \mathbf{u}_{N_r'} \end{bmatrix} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_{N_r'} \\ & \mathbf{0} & \end{bmatrix} \times \begin{bmatrix} \mathbf{v}_1 \, \mathbf{v}_2 \, \dots \, \mathbf{v}_{N_t'} \end{bmatrix} \tag{43}$$

where $\mathbf{u}_i$ and $\mathbf{v}_i$ are column vectors of $\mathbf{U}$ and $\mathbf{V}$, respectively. Since $\mathrm{N_r'} \leq \mathrm{N_t'}$ in this case. Then, we define $\mathbf{P}_1^{(1)}$, where

$$\mathbf{P}_1^{(0)} = \mathbf{H}^{\mathrm{H}}\mathbf{H} = \begin{bmatrix} p_{11} & \dots & p_{1N_t'} \\ \vdots & \ddots & \vdots \\ p_{N_t'1} & \dots & p_{N_t'N_t'} \end{bmatrix}. \tag{44}$$

N_t' is smaller than the N_t, which is the dimension of the original target matrix size, and $\mathbf{P}_1^{(1)} \in \mathbb{C}^{N_t' \times N_t'}$ is also a semipositive definite matrix. The proposed algorithm can also be applied to find the singular vectors and corresponding singular values of the smaller channel matrix.

The dimension of $\mathbf{U}$ is greater than that of $\mathbf{V}$ due to $N_r' \leq N_t'$ in this case. After deriving the complete $N_t' \times N_t'$ matrix $\mathbf{V}$, there will be still $(N_r' - N_t')$ unresolved left singular vectors, $\mathbf{u}_i'$s, even after the operation of (22) and (24). To derive these unresolved left singular vectors, we need only to perform GS scheme and normalization operations.

Recalling that singular vector matrix $\mathbf{U}$ is a unitary matrix, the column vectors in matrix $\mathbf{U}$ are orthonormal to each other. Therefore, the remaining vectors can be calculated from applying GS. The remaining left singular vectors can be derived as

$$\text{For} = (N_t' + 1) \ \textit{to} \ N_r'$$

$$\widetilde{\mathbf{w}}_i = \mathbf{e}_i - \sum_{k=1}^{i-1} \left(\tilde{\mathbf{e}}_i^{\mathrm{H}} \widetilde{\mathbf{u}}_k \right) \widetilde{\mathbf{u}}_k \tag{45}$$

$$\widetilde{\mathrm{u}}_i = \frac{\widetilde{\mathbf{w}}_i}{\|\widetilde{\mathbf{w}}_i\|_2} \tag{46}$$

end

where $\mathbf{e}_i$ is orthonormal to $\mathbf{e}_j$ with $i \neq j$, and $\mathbf{e}_i$ is not parallel to $\mathbf{u}_1, \mathbf{u}_2, \dots$, and $\mathbf{u}_{i-1}$. For simplicity, we can define

$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \mathbf{e}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \dots \tag{47}$$

The equations of (34) and (35) show the proposed SL-SVD algorithm can also handle non-square channel matrices. Notice that while N_r' is smaller than N_t', we only need to exchange the roles of $\mathbf{u}_i$ and $\mathbf{v}_i$.

In the case of nonsquare channel matrix, we can also insert zero column vectors and zero row vectors to the channel matrix with smaller size since the generated matrix has the same properties. The channel matrix after inserting zero vectors has the size equal to the maximum square channel matrix case (4×4 for IEEE 802.11n standard for example). Then, we can apply the proposed algorithm to find the desired singular vectors and values.

C. Computational Cost

We now analyze and compare the computational cost of the proposed SL-SVD algorithm with other referenced algorithms. Our proposed algorithm requires a matrix to matrix multiplication in each iteration, which seems to be more complex than other algorithms whereas it is not a fair comparison. The comparison would be fairer if we can compare the overall computational cost after all iterations.

Assume that the maximum iteration number needed for each singular vector is N_{ite} and use one *complex multiplier and adder (CMAC)* to be the basic computational cost unit. $\frac{N_t^3}{2}$ CMACs are required for the initial stage operation. For all the iterations we need about $N_{\mathrm{ite}} \left(\frac{N_t^3}{2} \right) (N_t - 1)$ CMACs for the matrix multiplication of the Hermitian matrix $\mathbf{P}_k^{(n)}$ in the main loop. In addition, we still need $\left(\frac{N_t^3}{2} \right) (N_t - 1)$ and about N_t^3 CMACs for deflation and GS process, respectively. At last, $(N_r \times N_t^2)$ CMACs, few square rooters, and dividers are required for the derivation of $\widetilde{\boldsymbol{U}}$ and $\widetilde{\boldsymbol{\Sigma}}$. The number of N_{ite} set in our algorithm for a 4×4 channel matrix is about 4 at most, which will be shown in Section V. We will then compare the total computational cost and show the hardware architecture here. The overall computational cost of the proposed SL-SVD is listed in (48) and we also compare it with the SVD algorithms in [6]–[9].

For an $N \times N$ matrix, the overall computational cost of each SVD algorithm is listed as follows:

Proposed SL-SVD:

$$N_{\mathrm{ite}} \left(\frac{N^3}{2} \right) (N - 1) + \frac{5N^3}{2} + O(N). \tag{48}$$

SVD in [8]:

$$N_{\mathrm{ite}}' \times (12N + 3) + O(N). \tag{49}$$

SVD in [6], [7]:

$$N_{\mathrm{ite}}'' \times (10N + 3) + O(N). \tag{50}$$

SVD in [9]:

$$N_{\mathrm{ite}}''' \times (20N) + \frac{8}{3}N^3 + 10N^2 + O(N). \tag{51}$$

The computational costs of each iteration in [6]–[8] are derived according to the algorithm flows in their articles. Though the adaptive SVD algorithm proposed in [6]–[8] have less cost

TABLE I
COMPUTATIONAL COST COMPARISON OF SVD ALGORITHMS PER CHANNEL MATRIX

	3 × 3 complex channel matrix	4 × 4 complex channel matrix
SVD algorithm in [8]	18k CMACs	32k CMACs
SVD algorithm in [6][7]	15.6k CMACs	20.4k CMACs
SVD algorithm in [9]	0.4k CMACs	0.8k CMACs
SL-SVD algorithm	0.3 k CMACs	0.6k CMACs

in each iteration and are able to operate without CSI, the numbers of iteration required in [6]–[8], N'_{ite}, and N''_{ite}, are about 600 and 500, respectively. The SVD in [6] and [7] switches to blind-tracking mode after the training phase and makes updates every clock cycle for slowly fading channels. Nevertheless, it may require the training operation frequently for fast changing channels. On the contrary, the proposed scheme completes the SVD with few iterations under all the channel environments.

The SVD in [9] utilized CORDIC with microrotations for Givens rotation. We treat one complete CORDIC operation for complex values as one CMAC, and the overall computational cost is listed in (50). The number of iteration required in [9], N'''_{ite}, is only about 4. Nevertheless, it requires 16-bit precision and 9 microrotations of one complete CORDIC processing. The requirements of longer wordlength and numbers of microrotations may limit the throughput of the SVD processing. The computational cost of each algorithm for decomposing one 3 × 3 or 4 × 4 channel matrix is shown in Table I. The proposed SL-SVD has about 75% computational cost compared with the related works.

D. Analysis of Quantization Error Effect

The proposed SL-SVD algorithm can ideally accomplish the SVD process with infinite numerical precision and iterations in mathematics. While in practical implementations, only finite precision can be used to represent the elements in the matrices in the iterative processing of our algorithm. In reality, different channel noise models and algorithms of channel estimation will also affect the results in various ways. It is not easy to completely evaluate the effects to the results. However, we can still analyze the bound of the singular value errors induced by the quantization noises during the iterative multiplication in the proposed SL-SVD algorithm under the assumption of perfect channel matrix information given here.

Because of the nature of the proposed algorithm, deriving the singular vectors prior to singular values, we will first derive the error bound of the perturbation of the singular vectors. Second, we will then show how the perturbation in singular vectors affects the singular values, and also derive the error bound of the derived singular values. We employ the index of *relative singular value error (RSVE)* to evaluate the performance degradation. We will then prove that the RSVE in our proposed SL-SVD algorithm caused by the quantization effect is theoretically bounded. In the critical case, as shown in Fig. 2, there is an error angle θ between the ideal singular vector $\boldsymbol{v}_1$ and the estimated singular vector $\widetilde{\boldsymbol{v}}_1$. The angular difference will cause the resulted singular values derived at the last stage to be affected by numerical errors.

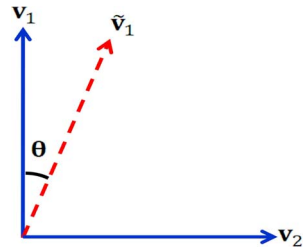


Fig. 2. A difference angle between the ideal singular vector $\mathbf{v}_1$ and the estimated singular vector $\tilde{\mathbf{v}}_1$.

We define the channel matrix to be an $N \times N$ complex matrix, the energy of the maximum quantization error of each number to be E_{num}, and $\mathrm{E} = \sqrt{2}\mathrm{N}\mathrm{E}_{\text{num}}$. We will then prove that the relative singular value error is bounded under the effect of quantization errors in practical hardware implementations. The RSVE can be expressed as (52), And then we utilize the concept in matrix perturbation theory [10] to continue the proof.

$$
\begin{aligned}
&\text{RSVE} \\
&= \frac{\sigma_1^2 - \widehat{\sigma}_1^2}{\sigma_1^2} = \frac{\sigma_1^2\left(\sigma_1^2\cos^2\theta + \sigma_2^2\sin^2\theta\right)}{\sigma_1^2} \\
&= 1 - \left(\cos^2\theta + \frac{\sigma_2^2}{\sigma_1^2}\sin^2\theta\right) = \left(1 - \frac{\sigma_2^2}{\sigma_1^2}\right)\sin^2\theta \\
&= \left(1 - \frac{\sigma_2^2}{\sigma_1^2}\right)\left(\frac{\left|\widehat{\mathbf{v}}_1^{\mathrm{H}}\mathbf{v}_2\right|}{\|\widehat{\mathbf{v}}_1\|_2\|\mathbf{v}_2\|_2}\right)^2 \\
&= \left(1 - \frac{\sigma_2^2}{\sigma_1^2}\right)\left(\frac{\left(\sigma_1^2 + 2\mathrm{E}^2 + \sigma_2^2\right)(\sigma_1 + \sigma_2)\mathrm{E}}{\left(\sigma_1^2 + \mathrm{E}^2\right)^2 - \left(\sigma_2^2\right)^2 + (\sigma_1 + \sigma_2)^2\mathrm{E}^2}\right)^2 \\
&= \left(1 - \frac{\sigma_2^2}{\sigma_1^2}\right)\left(\frac{(\sigma_1 + \sigma_2)\mathrm{E}}{\frac{\left(\sigma_1^2 + \mathrm{E}^2\right)^2 - \left(\sigma_2^2\right)^2}{\left(\sigma_1^2 + \mathrm{E}^2 + \sigma_2^2\right)} + \frac{(\sigma_1 + \sigma_2)^2\mathrm{E}^2}{\sigma_1^2 + \mathrm{E}^2 + \sigma_2^2}}\right)^2 \\
&= \left(1 - \frac{\sigma_2^2}{\sigma_1^2}\right)\left(\frac{(\sigma_1 + \sigma_2)\mathrm{E}}{\left(\sigma_1^2 - \sigma_2^2 + \mathrm{E}^2\right) + \frac{(\sigma_1 + \sigma_2)^2\mathrm{E}^2}{\sigma_1^2 + \mathrm{E}^2 + \sigma_2^2}}\right)^2 \\
&< \left(1 - \frac{\sigma_2^2}{\sigma_1^2}\right)\left(\frac{(\sigma_1 + \sigma_2)\mathrm{E}}{\sigma_1^2 - \sigma_2^2}\right)^2 \\
&= \begin{cases} \left(1 - \frac{\sigma_2^2}{\sigma_1^2}\right)\frac{\mathrm{E}^2}{(\sigma_1 - \sigma_2)^2}, & \text{if } \frac{\mathrm{E}^2}{(\sigma_1 - \sigma_2)^2} \le 1 \\ \left(1 - \frac{\sigma_2^2}{\sigma_1^2}\right), & \text{if } \frac{\mathrm{E}^2}{(\sigma_1 - \sigma_2)^2} > 1. \end{cases}
\end{aligned}
\tag{52}
$$

At the end of (52), there are two cases to be concerned. We will prove the relative error is bounded under both two cases.

Case 1)

$$
\begin{aligned}
&\frac{\mathrm{E}^2}{(\sigma_1 - \sigma_2)^2} \le 1 \\
&\quad \therefore \sigma_2 \le \sigma_1 - \mathrm{E}\left(1 - \frac{\sigma_2^2}{\sigma_1^2}\right)\frac{\mathrm{E}^2}{(\sigma_1 - \sigma_2)^2} \\
&\quad \le \left(1 - \frac{(\sigma_1 - \mathrm{E})^2}{\sigma_1^2}\right)\frac{\mathrm{E}^2}{(\sigma_1 - (\sigma_1 - \mathrm{E}))^2} \\
&\quad = \frac{2\sigma_1\mathrm{E} - \mathrm{E}^2}{\sigma_1^2} \\
&\quad = \frac{2\sqrt{2}\sigma_1 N E_{\text{num}} - 2N^2E_{\text{num}}^2}{\sigma_1^2}.
\end{aligned}
\tag{53}
$$

Case 2)

$$\therefore \frac{\mathrm{E}^2}{(\sigma_1-\sigma_2)^2} > 1$$
$$\therefore \sigma_2 > \sigma_1 - \mathrm{E}$$
$$\left(1-\frac{\sigma_2^2}{\sigma_1^2}\right) < \left(1-\frac{(\sigma_1-\mathrm{E})^2}{\sigma_1^2}\right) = \frac{2\sigma_1\mathrm{E}-\mathrm{E}^2}{\sigma_1^2}$$
$$= \frac{2\sqrt{2}\sigma_1 N E_{\mathrm{num}} - 2N^2E_{\mathrm{num}}^2}{\sigma_1^2}.$$

As shown in (52) and (53), the relative singular value error is bounded by

$$\mathrm{RSVE} < \frac{2\sqrt{2}\sigma_1 N E_{\mathrm{num}} - 2N^2E_{\mathrm{num}}^2}{\sigma_1^2}. \quad (54)$$

V. Simulation Results

In this section, we will show the simulations of convergence rate and the BER performance in the IEEE 802.11n system. The performance with OR mechanism, different fixed-point precision, and coherent time will also be shown.

A. Superlinear-Convergence Rate

In this section, we introduce an index, normalized mean square error (NMSE) [8], which is defined as

$$\mathbf{NMSE} = \frac{\left\|\tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^{\mathrm{H}} - \mathbf{H}\right\|_F^2}{\|\mathbf{H}\|_F^2} \quad (55)$$

where $\|\cdot\|_{\mathrm{F}}$ symbolizes the *Frobenius norm* [4] of a matrix. In this simulation under different channel matrices, the better SVD algorithm with better estimation has lower NMSE. In other words, if we have a perfect estimation of the SVD results, the value of NMSE would be zero.

In Fig. 3, we use 4×4 complex channel matrices and evaluate the NMSE performance. Deriving each right singular vector, we use different iteration number and wordlength precision for computation. The proposed SL-SVD algorithm needs only 1–4 iterations for the performance to converge. We need about 4 iterations for each singular vector to converge, and the NMSE in convergence region is as low as -48 dB with 10-bit precision. The parallelized two-sided Jacobi SVD algorithm in [11] is also compared in Fig. 3 due to its feature of requiring small numbers of iteration for small matrices. The SVD algorithm in [11] can deal with all singular values and vectors at the same time. The equivalent iteration number required for each singular vector is derived by scaling down the total amount of iteration required by 4 for the 4×4 channel matrices for fairness. The operation of one complete Jacobi rotation on both sides is treated as one iteration, and the parallel architecture mentioned in [11] is considered. Compared to it, the proposed SL-SVD still has faster convergence rate and better NMSE performance. In Fig. 4, we set the channel matrices to be 3×2 asymmetric complex matrices and the SL-SVD algorithm still works. The convergence rate in these cases is even faster than that in the 4×4 cases due to the rank deficiency, and only 1–2 iterations needed of deriving each right singular vector.

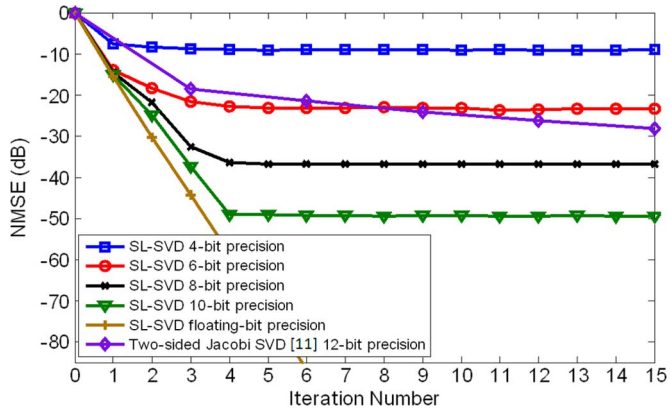


Fig. 3. Convergence performance of the NMSE of the proposed SL-SVD algorithm with 4×4 complex channel matrices.

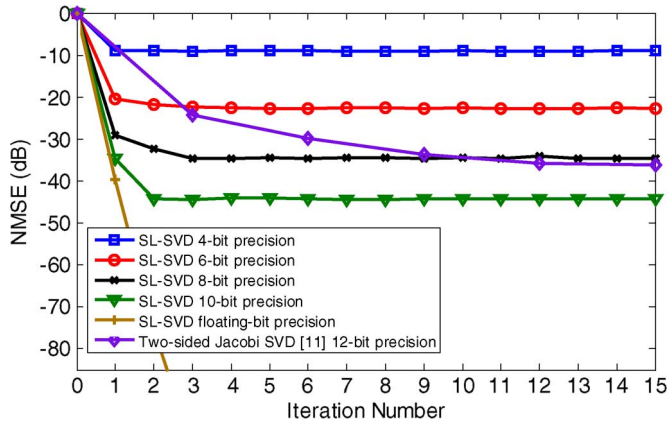


Fig. 4. Convergence performance of the NMSE of the proposed SL-SVD algorithm with 3×2 complex channel matrices.

B. Fixed-Point Simulation

We also evaluate the performance under a practical MIMO-OFDM application, IEEE 802.11n. The results of the proposed SL-SVD engine applied to IEEE 802.11n standard PHY is presented in bit error rate (BER).

- Target application: IEEE 802.11n WLAN
- Beamforming technique: SVD
- AWGN, Ch. E (nLOS) channels [12], 4 spatial streams (4×4 channel matrix)
- Time invariant channel is used
- Perfect channel state information is assumed
- 4 QAM, 16 QAM, and 64 QAM is considered
- MIMO-OFDM system with 128 tones [13]
- Code rate $\frac{1}{2}$ convolutional code with constraint length 7, generator polynomials [133 171] [13]
- Block interleaving is used
- Zero-forcing detector

In Fig. 5, BPSK-modulated signals are exploited to evaluate the BER performance versus different signal-to-noise ratios (SNR) with different fixed-point precisions without any channel coding. The channel matrices are set to be 4×4 complex matrices. The unitary matrix $\widetilde{\mathbf{V}}$ is fed back to transmitter for precoding and the other unitary matrix $\widetilde{\mathbf{U}}$ is used for post-coding at the receiver. Different precision in Fig. 5 indicates

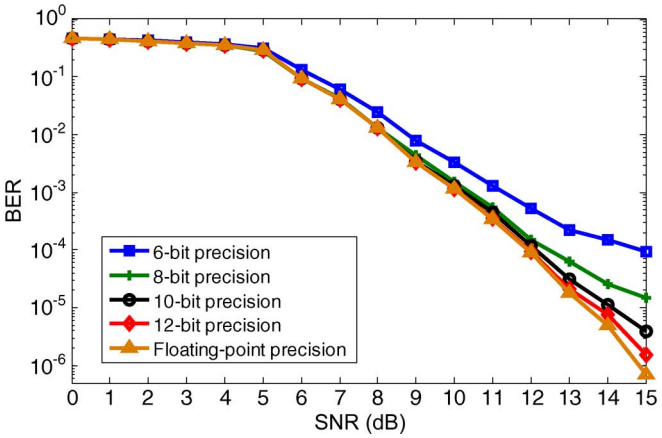


Fig. 5. BER performance of the proposed SL-SVD algorithm with 6, 8, 10, and 12-bit precision with 4 × 4 complex channel matrices and QPSK signals.

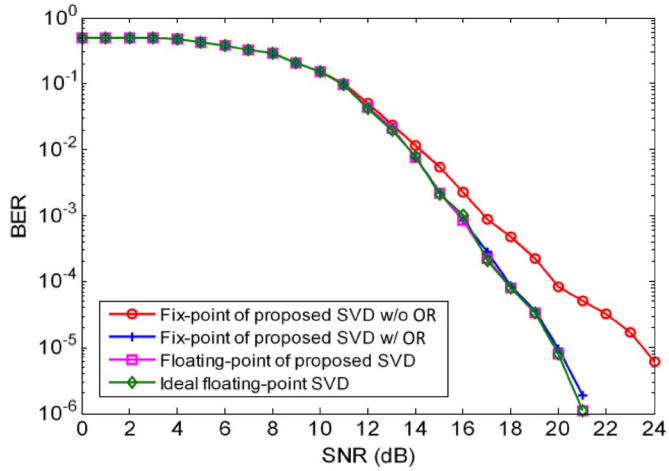


Fig. 6. IEEE 802.11n PHY layer BER versus SNR with 16 QAM.

the precision for the iterative processes and the storage of the right singular vectors, which is the critical path of the SVD engine. We can use more precision to calculate and store the singular values σ_i's, and left singular vectors $\mathbf{u}_i$'s, which can be put outside the critical path, and we use 30 bits here to store the singular values and left singular vectors.

As shown in Fig. 5, the proposed SL-SVD has no noticeable performance degradation with 10- and 12-bit precision. In the situation of lower SNR, the BER performances with 10 and 12 bits are almost as good as that with the floating-point precision. Under the conditions of higher SNR, the SL-SVD algorithm with 10-bit precision still works well.

C. Simulation of Orthogonal Reconstruction

Simulation results are shown in Fig. 6 are the performance of the proposed SL-SVD algorithm with Viterbi decoder defined in the IEEE 802.11n specification. There are four lines indicating the performance of ideal floating-point SVD, floating-point, fixed-point (10-bit precision) without OR, and fixed-point (10-bit precision) with OR scheme of the proposed SL-SVD. Simulation shows that the proposed SL-SVD algorithm with floating-point has the same BER performance compare to ideal SVD with floating-point. It also shows that with proper word-length, 10-bit precision in this case, our SL-SVD engine without OR scheme can work under IEEE 802.11n standard physical layer with less than 0.2 dB performance lose at 10^{-5} BER compared with ideal SVD.

D. Simulation of Different Channel Coherent Time

In the following simulations, we will compare the performances of different beamforming algorithms in the IEEE 802.11n system with different channels of various coherent times. In Figs. 7 and 8 with 10 and 1 ms coherent time, we can see that the proposed beamforming algorithm has apparently better performance than that of the referenced algorithms.

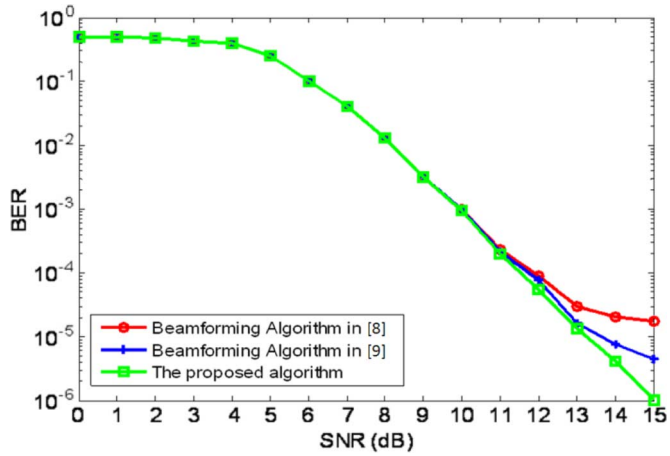


Fig. 7. IEEE 802.11n PHY layer BER versus SNR with 10 ms coherent time.

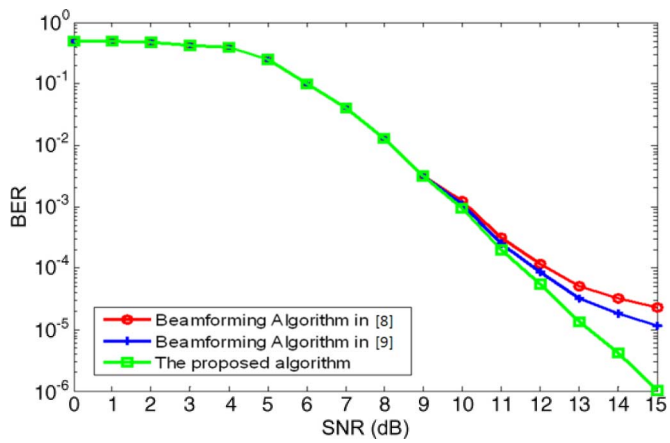


Fig. 8. IEEE 802.11n PHY layer BER versus SNR with 1 ms coherent time.

VI. Architecture Design

The overall architecture of the hardware design is shown in Fig. 9. It is mainly composed of four parts: 1) matrix array multiplication for iterative multiplication and deflation; 2) matrix-vector multiplier for orthogonal reconstruction; 3) pipelined vector normalization for deriving singular values and vectors; and 4) specific control circuits and storages of right singular vectors before or after OR operation. We can derive the desired singular values, left, and right singular vectors after the proposed iterative processing.

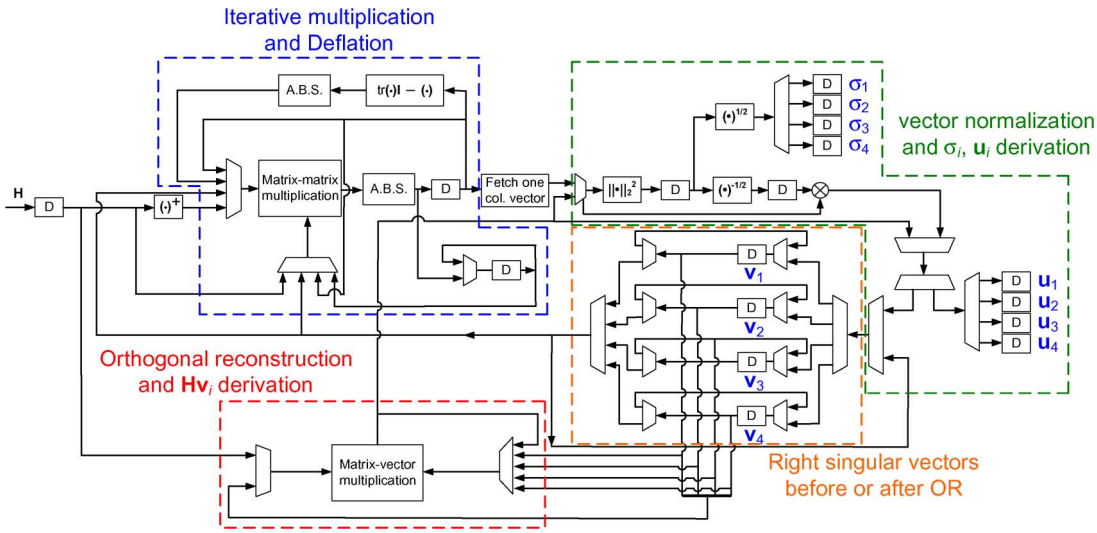


Fig. 9. The architecture of the proposed SL-SVD algorithm.

In Fig. 10(a), the matrix-matrix multipliers are designed for the matrix multiplication. The inputs are two matrices and the output is an upper triangular matrix due to its Hermitian property so that the iterative multiplication cost can be reduced by half without performance degradation. For an $N \times N$ complex matrix, only $\frac{N^3}{2\ \mathrm{CMACs}}$ are required in the matrix-matrix multiplication block. In addition to the iterative multiplication, the deflation operation can also be executed with these multipliers. The function of A.B.S is designed to solve the problem of the exponentially growing values in the matrix during iterative multiplication. As shown in (19), a delicate binary shift is applied to the whole matrix after each iteration according to the magnitudes of the diagonal elements. The A.B.S. block is simplified to be multiplexers and XOR gates only.

In Fig. 10(b), the matrix-vector multipliers can be utilized in the phase of orthogonal reconstruction by Gram-Schmidt process and computation of $\mathbf{Hv}_i$. As described in (35), two cycles are required to obtain the results of OR operation. In Fig. 10(c), the pipelined vector normalization can be decomposed to be: square of the vector 2-norm, inverse square root, square root, and vector scaling. The digit-by-digit calculation and digit recurrence algorithm in [27] are adopted for implementing the square root and inverse square root operations, respectively. This block can be used to obtain the normalized left and right singular vectors. The singular values can also be derived with the square root function. The straightforward implementations of inverse square root and square root functions are applied in our design, and the equivalent gate counts are about 9 and 0.8 k, respectively. These two function blocks are hardware expensive and occupy about 6% area over the entire design. Although straight implementations for inverse square root and square root functions are employed in this work, the CORDIC operation is feasible to mitigate the cost of the square root function. The storages of left singular vectors before or after OR operation is shown in Fig. 10(d). With dedicate task arrangement, the storages of the right singular vectors can be outputted for OR operation or $\mathbf{Hv}_i$ computation. The fine-tuned results of right singular vectors can also be stored after OR operation.

VII. Implementation Results and Comparisons

The postlayout analysis of the proposed SL-SVD engine is obtained by using Verilog HDL codes synthesized with the standard cell library of UMC 90 nm 1P9M Low-K process in a core size 0.48 mm^2 at 182–MHz operating frequency. The power consumption is evaluated with Synopsys Prime Power in 4×4 antenna mode. Chip features are summarized in Table II. To meet the specification of IEEE 802.11n standard, the proposed SL-SVD engine can support 16 antenna modes.

For the application to IEEE 802.11n standard, we use the SVD engine to serially decompose all the channel matrices all subcarriers. Chip results show that the latency of our SL-SVD engine for 128 subcarrier MIMO-OFDM system is about 0.3% of WLAN coherence time [14] to prevent time-varying channel.

For comparison, we define two indexes. First, the throughput is defined by the number of channel matrices the SVD engine can deal with within a second

$$\mathrm{Throughput} = \frac{\mathrm{Number\ of\ processed\ matrices}}{\mathrm{time(sec)}}. \tag{56}$$

Then, the system area efficiency is defined by

$$\mathrm{Normalized\ Area\ efficiency} = \frac{\mathrm{Throughput}}{\mathrm{Equivalent\ gate\ count}}. \tag{57}$$

Table III shows the comparisons of the proposed SL-SVD engine with other works. All the numerical data in Table III are excerpted from the referenced papers or calculated according to the given data. The work in [6] and [7] proposed an SVD engine which does not need channel estimation for general WLAN, and it employees an iterative division-free Newton-Raphson method to execute the division operation. The work in [6] only supports the 4×4 antenna mode for wireless local area network application. In the IEEE 802.11n systems, the complex channel matrix is given with channel estimation. Only 0.14 μs is required for

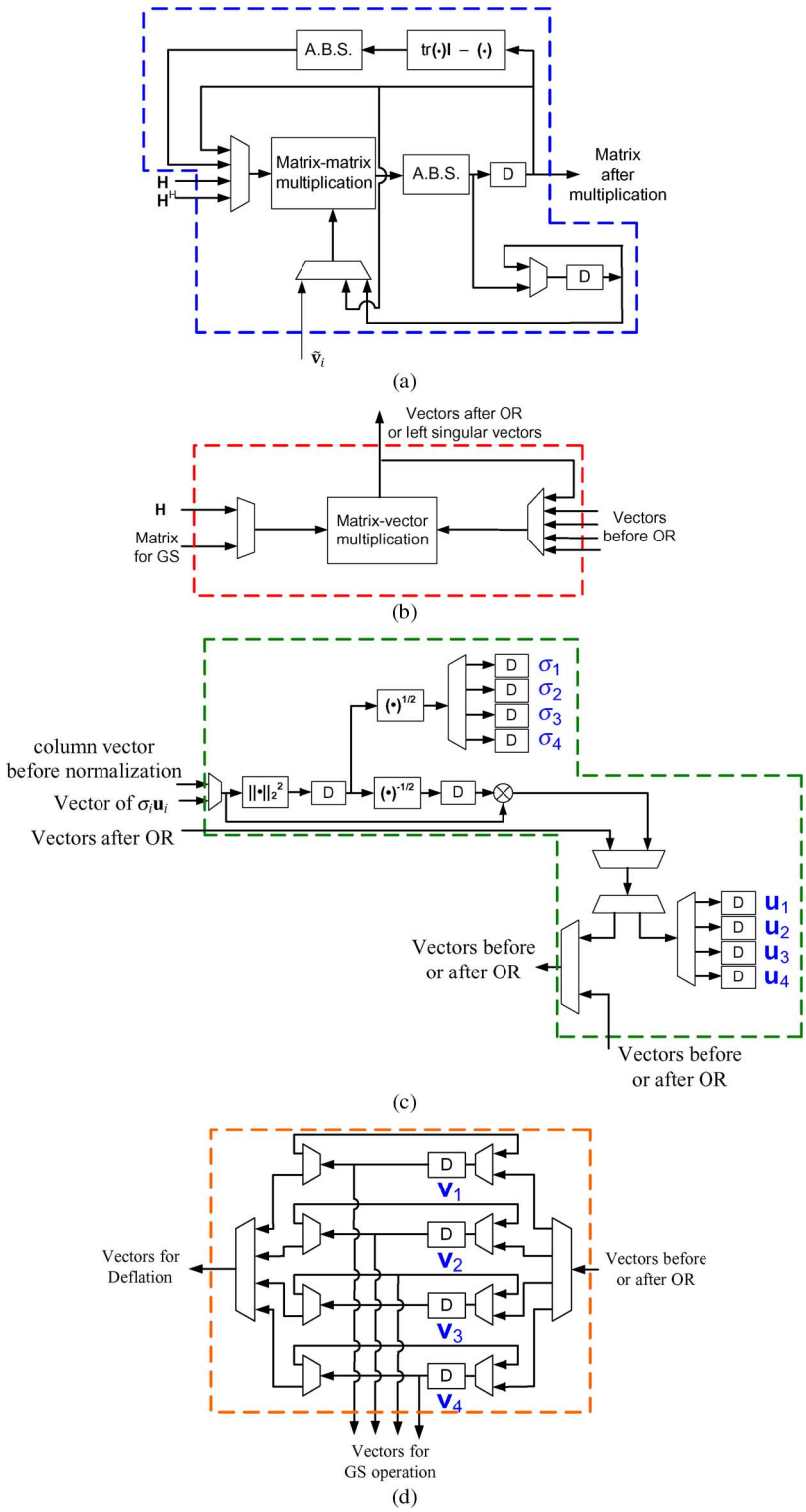


Fig. 10. (a) Matrix-matrix multiplication and A.B.S. (b) Matrix-vector multiplication. (c) Pipelined vector normalization. (d) Storages of right singular vectors before or after OR operation.

the SVD of one complex channel matrix owing to the superlinear-convergence property of proposed SL-SVD. In successive matrix processing, the equivalent processing time required for each matrix can even be reduced to 90 ns. The normalized area efficiency is five times better than the referenced works due to the properties of low computational cost and insensitivity to the dynamic range problem. The prototype design can be also extended to different antenna sets.

We need only few numbers of iteration to complete SVD process due to the property of superlinear-convergence rate of the proposed SL-SVD. The SL-SVD is division-free and only multiplication operation is introduced in each iteration. The

TABLE II
POSTLAYOUT ANALYSIS

The proposed SL-SVD engine	
Support Antenna Mode	4 × 4, 4 × 3, 4 × 2, 4 × 1, 3 × 4, 3 × 3, 3 × 2, 3 × 1, 2 × 4, 2 × 3, 2 × 2, 2 × 1, 1 × 4, 1 × 3, 1 × 2, 1 × 1
Technology	UMC 90nm 1P9M Low-K Process
Core area	0.69mm x 0.69mm (0.48mm^2)
Gate Count	120k
Frequency	182MHz
Power	18mW@182MHz

TABLE III
COMPARISONS OF THE SVD ENGINES

	2006 Adaptive SVD [7]		2008 GK-SVD [9]	Proposed SL-SVD
Support Tx-Rx Sets	4 × 4	4 × 4	4 × 4	*** 1 × 1 ~ 4 × 4
Technology	90nm	90nm	0.18um	90nm
Gate Count	900k	900k	42.3k	120k
Core Area(mm^2)	3.81 *(4.57)	3.81 *(4.57)	0.41	0.48
Frequency (MHz)	**500	100	149	182
Throughput (Channel-matrices/sec)	125k	25k	303k	7M (11.3M)
SVD time per matrix (μs)	8	40	3.3	0.14 (0.09)
Power (mW)	N/A	34	N/A	18
Normalized Area Efficiency ((Tech/90nm)^2*Throughput/Area)	27.4k	5.5k	3M	14.6M

* Ref. [7] claims that it requires 30% more area to derive the complete SVD results

** Ref. [7] claims that the operating frequency can be up to 500 MHz in simulation

*** Extension to Tx/Rx sets: (4 × 4, 4 × 3, 4 × 2, 4 × 1, 3 × 4, 3 × 3, 3 × 2, 3 × 1, 2 × 4, 2 × 3, 2 × 2, 2 × 1, 1 × 4, 1 × 3, 1 × 2, 1 × 1)

A.B.S. and orthogonality reconstruction (OR) are also utilized for updating and vector correction, so that we can use only 10-bit precision in our design. That is why our design is area and power efficient.

VIII. CONCLUSION

In this paper, we propose a superlinear-convergence rate SVD algorithm. The algorithm can obtain the SVD results of the complex MIMO-OFDM channel matrices about 25 times faster than other referenced algorithms. The superlinear-convergence speed makes this algorithm suitable for the channels with short coherent time. Moreover, the SL-SVD engine can be extended to decompose the $N_t \times N_r$ or smaller channel matrices with little hardware overhead. The total computational cost is low owing to the superlinear-convergence rate. A hardware implementation with 90 nm technology is also presented. The chip has the feature of 0.48 mm^2 core area, 18 mW power consumption, being able to handling 7 M-channel-matrices/s, and can be extended to deal with different transmit and receive antenna sets.

REFERENCES

[1] A. Goldsmith, S. A. Jafar, N. Jindal, and S. Vishwanath, "Capacity limits of MIMO channels," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 5, pp. 684–702, Jun. 2003.

[2] Specification of IEEE 802.11n physical layer [Online]. Available: http://www.ieee802.org/11/

[3] D. J. Love and R. W. Heath, "Limited feedback unitary precoding for spatial multiplexing systems," *IEEE Trans. Inf. Theory*, vol. 51, no. 8, pp. 2967–2976, Aug. 2005.

[4] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, MD: The Johns Hopkins Univ., 1996.

[5] C. Michalke, M. Stege, F. Schäfer, and G. Fettweis, "Efficient tracking of eigenspaces and its application to eigenbeamforming," in *Proc. IEEE Pers., Indoor, Mobile Radio Commun. (PIMRC)*, 2003, vol. 3, pp. 2847–2851.

[6] A. S. Y. Poon, D. N. C. Tse, and R. W. Brodersen, "An adaptive multiantenna transceiver for slowly flat fading channels," *IEEE Trans. Commun.*, pp. 1820–1827, Nov. 2003.

[7] D. Markovic, R. W. Brodersen, and B. Nikolic, "A 70 GOPS, 34 mW multi-carrier MIMO chip in 3.5 mm^2," in *Proc. IEEE Symp. VLSI Circuits*, 2006, pp. 158–159.

[8] T. J. Willink, "Efficient adaptive SVD algorithm for MIMO applications," *IEEE Trans. Signal Process.*, pp. 615–622, Feb. 2008.

[9] C. Senning, C. Studer, P. Luethi, and W. Fichtner, "Hardware-efficient steering matrix computation architecture for MIMO communication systems," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2008, pp. 304–307.

[10] G. W. Stewart and J. G. Sun, *Matrix Perturbation Theory*, 2nd ed. Boston, MA: Academic, 1990.

[11] N. D. Hemkumar and J. R. Cavallaro, "A systolic VLSI architecture for complex SVD," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 1992, vol. 3, pp. 1061–1064.

[12] M. Clark, *IEEE 802.11a WLAN Model*. New York: Mathworks, Inc., Jun. 2003 [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=3540&objectType=file

[13] *Joint Proposal: High Throughput Extension to the 802.11 Standard*, PHY Doc.: IEEE 802.11-05/1102r4 [Online]. Available: http://www.ieee802.org/11/Doc-Files/05/11-05-1102-04-000n-joint-proposal-phy-specification.doc

[14] T. K. Paul and T. Ogunfunmi, "Wireless LAN comes of age: Understanding the IEEE 802.11n amendment," *IEEE Circuits Syst. Mag.*, vol. 8, no. 1, pp. 28–54, 1st Quarter, 2008.

[15] D. Markovic, B. Nikolic, and R. W. Brodersen, "Power and area minimization for multidimensional signal processing," *IEEE J. Solid-State Circuits*, vol. 42, no. 4, pp. 922–934, Apr. 2007.

[16] N. Seshadri and J. H. Winters, "Two signaling schemes for improving the error performance of frequency-division-duplex (FDD) transmission systems using transmitter antenna diversity," *Int. J. Wireless Inf. Netw.*, vol. 1, no. 1, Jan. 1994.

[17] S. M. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 16, pp. 1451–1458, Oct. 1998.

[18] G. J. Foschini and M. J. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," *Wireless Pers. Commun.*, vol. 6, no. 3, pp. 311–335, Mar. 1998.

[19] H. Sampath, S. Talwar, J. Tellado, V. Erceg, and A. Paulraj, "A fourth-generation MIMO-OFDM: Broadband wireless system: Design, performance, and field trial results," *IEEE Commun. Mag.*, vol. 40, no. 9, pp. 143–149, Sep. 2002.

[20] E. Telatar, "Capacity of multiantenna Gaussian channels," AT&T-Bell Labs Internal Tech. Memo, 1995.

[21] G. G. Raleigh and J. M. Cioffi, "Spatio-temporal coding for wireless communication," *IEEE Trans. Commun.*, vol. 46, pp. 357–366, Mar. 1998.

[22] D. J. Love and R. W. Heath, Jr., "Equal gain transmission in multiple-input multiple-output wireless systems," *IEEE Trans. Commun.*, vol. 51, no. 7, pp. 1102–1110, Jul. 2003.

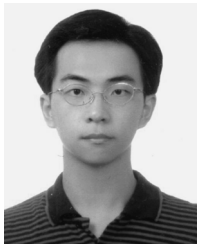
[23] J. H. Winters, "On the capacity of radio communication systems with diversity in a Rayleigh fading environment," *IEEE J. Sel. Areas Commun.*, vol. SAC-5, pp. 871–878, Jun. 1987.

[24] IEEE P802.11n/D3.00, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.

[25] S. Haykin, *Adaptive Filter Theory*. Upper Saddle River, NJ: Prentice-Hall, 1991.

[26] K. K. Parhi, *VLSI Digital Signal Processing System*. New York: Wiley, 1999.

[27] S. Majerski, "Square-rooting algorithms for high-speed digital circuits," *IEEE Trans. Comput.*, vol. C-34, pp. 724–733, Aug. 1985.



Cheng-Zhou Zhan was born in Taiwan in 1983. He received the B.S. and M.S. degrees from National Taiwan University, Taipei, Taiwan, in 2005 and 2007, respectively, both in electrical engineering.

He is currently working toward the Ph.D. degree at the Graduate Institute of Electronics Engineering, National Taiwan University, Taipei, Taiwan, His research interests are in the areas of VLSI implementation of DSP algorithms, digital communication systems, and ultrasonic signal processing.

Yen-Liang Chen (S'07) received the B.S. degree in communication engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2005.

He is currently working toward the Ph.D. degree at the Graduate Institute of Electronics Engineering, National Taiwan University, Taipei, Taiwan. His research interests are in the areas of VLSI implementation of DSP algorithms, adaptive filtering, reconfigurable architecture, and digital communication systems.

An-Yeu (Andy) Wu (S'91–M'96) received the B.S. degree from National Taiwan University, Taipei, Taiwan, in 1987, and the M.S. and Ph.D. degrees from the University of Maryland, College Park, in 1992 and 1995, respectively, all in electrical engineering.

From August 1995 to July 1996, he was a Member of Technical Staff (MTS) at AT&T Bell Laboratories, Murray Hill, NJ, working on high-speed transmission IC designs. From 1996 to July 2000, he was with the Electrical Engineering Department, National Central University, Taiwan. In August 2000, he joined the faculty of the Department of Electrical Engineering and the Graduate Institute of Electronics Engineering, National Taiwan University, where he is currently a Professor. His research interests include low-power/high-performance VLSI architectures for DSP and communication applications, adaptive/multirate signal processing, reconfigurable broadband access systems and architectures, and SoC platform for software/hardware codesign.

Dr. Wu was a recipient of the A-class Research Award from the National Science Council for four times. He has served on many Technical Program Committees of IEEE international conferences, and was an Associate Editor of the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION SYSTEMS.