

Reconfigurable Adaptive Singular Value Decomposition Engine Design for High-Throughput MIMO-OFDM Systems

Yen-Liang Chen, Cheng-Zhou Zhan, Ting-Jyun Jheng, and An-Yeu (Andy) Wu, *Member, IEEE*

Abstract—Singular value decomposition (SVD) is an optimal method to obtain spatial multiplexing gain in multi-input multi-output (MIMO) channels. However, the high cost of implementation and high decomposing latency of the SVD restricts its usage in current wireless communication applications. In this paper, we present a complete adaptive SVD algorithm and a reconfigurable architecture for high-throughput MIMO-orthogonal frequency division multiplexing systems. There are several proposed architectural design techniques: reconfigurable scheme, division-free adaptive step size scheme, early termination scheme, and data interleaving scheme. The reconfigurable scheme can support all antenna configurations in a MIMO system. The division-free adaptive step size and early termination schemes are used to effectively reduce the decomposing latency and improve hardware utilization. The data interleaving scheme helps to deal with several channel matrices concurrently. Besides, we propose an orthogonal reconstruction scheme to obtain more accurate SVD outputs, and then the system performance will be greatly enhanced. We apply our SVD design to the IEEE 802.11n applications. This design is implemented and fabricated in UMC 90 nm 1P9M CMOS technology. The maximum operating frequency is measured to be at 101.2 MHz, and the corresponding power dissipation is at 125 mW. The core size is 2.17 mm² and the die size occupies 4.93 mm². The chip result shows that the average latency is only 0.33% of the wireless local area network coherence time. Hence, the proposed reconfigurable adaptive SVD engine design is very suitable for high-throughput wireless communication applications.

Index Terms—Adaptive array processing, multi-input multi-output (MIMO), orthogonal frequency division multiplexing (OFDM), reconfigurable architecture, singular value decomposition (SVD).

I. INTRODUCTION

DUE to the rapid evolution of wireless communication and the demand of high data rate for multi-media information access in recent years, single-input single-output transmission has become insufficient for use [1], [2]. Therefore, the research about multi-input multi-output (MIMO) technology becomes an important topic in many advanced wireless communication standards [3]–[5]. The advantage of a MIMO system is

that it exploits the space dimension to improve the system capacity and reliability. However, in a MIMO system, one receiving antenna may suffer from the interference of other transmitting antennas [6], [7]. This makes it hard for the receiver to obtain correct data. By applying the singular value decomposition (SVD) technique [8]–[10], the interference can be totally eliminated. Hence, the throughput and coverage of a MIMO system can be greatly enhanced. From an information-theoretical viewpoint, the use of SVD can be claimed as an optimal solution [11]–[13]. Besides, the advanced wireless local area network (WLAN) standard, IEEE 802.11n [14]–[16], has treated the SVD technique as an optional MIMO signal processing technique to enhance system performance. It is also shown that the application of the SVD technique has the highest throughput compared with other MIMO signal processing techniques in the IEEE 802.11n systems [17]. This indicates that the SVD technique is very important for the MIMO wireless communication systems.

Nowadays, there are several issues in applying the SVD technique to the wireless communication systems. These issues are discussed in detail as follows.

- 1) In many wireless communication standards, a MIMO system is usually combined with orthogonal frequency division multiplexing (OFDM) technology. The SVD engine needs to deal with hundreds of channel matrices of almost all subcarriers before data transmission. Hence, it is important to effectively reduce the total computational complexity.
- 2) In the WLAN environment, the coherence time over which the channel is considered essentially time-invariant is about 0.07 s [17], [18]. This indicates that we should complete the SVD operations of all channel matrices as soon as possible. Otherwise, the SVD results cannot be used for the present channel condition.
- 3) Assume that an MIMO system consists of up to M_T transmitter antennas and M_R receiver antennas. There are possibly $M_R \cdot M_T$ antenna configurations as well as channel matrix sizes. Hence, it is necessary to design a reconfigurable SVD engine for all antenna configurations. For example, in an 802.11n system, the number of transmitter antennas or receiver antennas can be from 1 to 4. The SVD engine should be capable of dealing with 16 antenna configurations.

In recent years, [19] proposed one ASIC realization of the SVD without the need of CSI for WLAN applications.

Manuscript received August 27, 2011; revised January 17, 2012; accepted March 21, 2012. Date of publication May 14, 2012; date of current version March 18, 2013. This work was supported in part by the National Science Council, Taiwan, under Grant NSC 97-2220-E-002-012.

The authors are with the Graduate Institute of Electronics Engineering and Department of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan (e-mail: ben@access.ee.ntu.edu.tw; andywu@cc.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/TVLSI.2012.2195040

However, the chip implements an adaptive blind-tracking $\mathbf{U}\Sigma$ algorithm [20] which is not complete for SVD outputs, and long convergence time is another disadvantage for the high-throughput MIMO-OFDM applications. A matrix decomposition architecture was proposed in [21] according to the Golub-Kahan SVD (GK-SVD) algorithm [22]. It achieves higher processing throughput than [19] with lower hardware cost. Based on the matrix decomposition architecture in [21], a hardware-efficient VLSI architecture was proposed in [23] by modifying the GK-SVD algorithm and using a high-speed Givens rotation design. To increase the processing speed, it only computes \mathbf{V} and Σ which are partial to SVD outputs. Nevertheless, the above-mentioned SVD designs only support 4×4 (four transmitter and four receiver antennas) antenna configuration which is not sufficient for dealing with different antenna configurations.

In this paper, we propose a complete adaptive SVD algorithm, as well as a reconfigurable architecture design, for the high-throughput MIMO-OFDM systems. Some of its key features are listed as follows.

- 1) Adaptive step size scheme, partial update scheme, and subcarrier inherit scheme (SIS) to effectively reduce the decomposing latency and increase the processing throughput.
- 2) Reconfigurable architecture for all antenna configurations in an MIMO system.
- 3) Early termination scheme to improve hardware utilization without losing system performance.
- 4) Data interleaving scheme to deal with several channel matrices simultaneously.
- 5) Orthogonal reconstruction (OR) scheme to enhance the system performance.

We implement the proposed reconfigurable SVD engine for the application of the IEEE 802.11n systems with up to four transmitter antennas and four receiver antennas. This chip is implemented using 90-nm CMOS technology with a core area of 2.17 mm^2 . It can be measured at 101.2 MHz with 125 mW power consumption. As compared with other related works, this chip achieves the highest throughput and power efficiency in the 4×4 SVD operations. In addition, the chip result shows that for an 802.11n system, the average latency of our SVD engine is only 0.33% of the WLAN coherence time. Therefore, the proposed SVD engine is very suitable for the high-throughput MIMO-OFDM systems.

The remainder of this paper is organized as follows. In Section II, we introduce the SVD technique in an MIMO system and review the adaptive blind-tracking $\mathbf{U}\Sigma$ algorithm. In Section III, the proposed complete adaptive SVD algorithm is presented. The proposed architectural design techniques for reconfigurable SVD engine are demonstrated in Section IV. The OR scheme is described in Section V. Section VI demonstrates the simulation and implementation results of the proposed SVD engine. Finally, we conclude this paper in Section VII.

In this paper, the following notation will be adopted. We use boldface capital letters to indicate matrices and boldface lowercase letters to indicate vectors. The letter \mathbf{I} denotes the

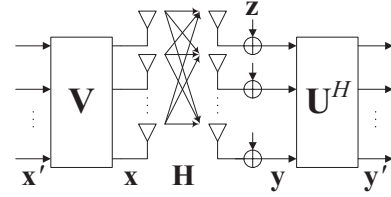


Fig. 1. MIMO system with the SVD technique.

identity matrix. $(\cdot)^H$ denotes the complex conjugate transpose of a vector or matrix. Expression $\text{tr}(\cdot)$ denotes the trace of a matrix, $\|\cdot\|$ denotes the two-norm of a vector, $\mathbf{R}(:, k)$ denotes the k th column of the matrix $\mathbf{R}(\mathbf{a}, \mathbf{b})$, $\|\mathbf{a}\|$ is the Euclidean inner product as $\mathbf{b}^H \mathbf{a}$, $\mathbb{C}^{p \times 1}$ denotes the set of $p \times 1$ complex vectors, and $\mathbb{C}^{p \times q}$ denotes the set of $p \times q$ complex matrices.

II. INTRODUCTION TO SVD TECHNIQUE

A. MIMO System Model and SVD

Consider a MIMO system with N_T transmitter and N_R receiver antennas. The baseband, discrete-time equivalent model is written by $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{z}$, where $\mathbf{H} \in \mathbb{C}^{N_R \times N_T}$ is the complex channel matrix, $\mathbf{z} \in \mathbb{C}^{N_R}$ is the additive white complex Gaussian noise vector, $\mathbf{x} \in \mathbb{C}^{N_T}$ is the transmitted data vector, and $\mathbf{y} \in \mathbb{C}^{N_R}$ is the received data vector. If we decompose the channel matrix \mathbf{H} by the SVD technique, we have

$$\mathbf{H} = \mathbf{U}\Sigma\mathbf{V}^H \quad (1)$$

where \mathbf{U} and \mathbf{V} are an $N_R \times N_R$ left singular matrix and an $N_T \times N_T$ right singular matrix, respectively. Both \mathbf{U} and \mathbf{V} are unitary matrices (i.e., $\mathbf{U}\mathbf{U}^H = \mathbf{I}$ and $\mathbf{V}\mathbf{V}^H = \mathbf{I}$) and Σ is an $N_R \times N_T$ matrix with only real and nonnegative main diagonal entries. The entry (i, i) of Σ denotes the i th largest value σ_i , with $1 \leq i \leq \min(N_R, N_T)$.

Let $\mathbf{x}' \in \mathbb{C}^{N_T}$ be the symbol vector such that $\mathbf{x} = \mathbf{V}\mathbf{x}'$ and the received signal \mathbf{y} is multiplied by \mathbf{U}^H as shown in Fig. 1. The channel between \mathbf{x}' and \mathbf{y}' can be written as

$$\mathbf{y}' = \mathbf{U}^H \mathbf{y} = \mathbf{U}^H (\mathbf{H}\mathbf{x} + \mathbf{z}) = \mathbf{U}^H \mathbf{H}\mathbf{V}\mathbf{x}' + \mathbf{z}' = \Sigma\mathbf{x}' + \mathbf{z}'. \quad (2)$$

Note that the distribution of \mathbf{z}' is invariant since \mathbf{U} is unitary. The MIMO channel can be treated as $d = \min(N_R, N_T)$ independent parallel Gaussian subchannels. The i th subchannel has the gain being σ_i . Hence, the transmitter can send independent data streams across these parallel subchannels without any interference from an antenna. Note that the values $\sigma_1, \sigma_2, \dots, \sigma_d$ are called the singular values of \mathbf{H} . The column vectors of \mathbf{V} (i.e., $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{N_T}$) are the right singular vectors of \mathbf{H} , and the column vectors of \mathbf{U} (i.e., $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{N_R}$) are the left singular vectors of \mathbf{H} .

B. Review of Adaptive Blind-Tracking $\mathbf{U}\Sigma$ Algorithm

In [20], the authors proposed an adaptive blind-tracking algorithm for \mathbf{U} and Σ as shown in Algorithm 1. n denotes the discrete time index. Without loss of generality, we omit the time index n in this subsection for simplicity. The received signal \mathbf{y} is used to estimate the autocorrelation matrix. Hence is the estimated autocorrelation matrix of $\mathbf{K}_y = E[\mathbf{y}\mathbf{y}^H]$.

Algorithm 1 Pseudo-code of adaptive blind-tracking U Σ algorithm [20]

```

Receive  $\mathbf{y}(n+1)$ 
Update covariance matrix
 $\hat{\mathbf{K}}_y(n+1) = (1-\beta)\hat{\mathbf{K}}_y(n) + \beta\mathbf{y}(n+1)\mathbf{y}(n+1)^H$ 
 $\mathbf{R}_1 = \hat{\mathbf{K}}_y(n+1)$ 
for  $i = 1:d$ ,
    Update the  $i$ th pair
     $\mathbf{w}_i(n+1) = \mathbf{w}_i(n) + \mu_i(\mathbf{R}_i - \lambda_i(n)\mathbf{I})\mathbf{w}_i(n)$ 
     $\lambda_i(n+1) = \mathbf{w}_i(n+1)^H \mathbf{w}_i(n+1)$ 
    Apply deflation
     $\mathbf{R}_{i+1} = \mathbf{R}_i - \mathbf{w}_i(n+1)\mathbf{w}_i(n+1)^H$ 
end

```

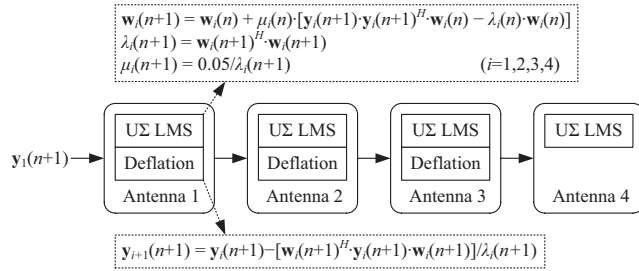


Fig. 2. Adaptive blind-tracking U Σ algorithm for 4 \times 4 antenna system [19].

Hence $\hat{\mathbf{K}}_y$ is the estimated autocorrelation matrix of \mathbf{K}_y . β is the forgetting factor and its choice depends on the stationary degree of the channel. d is the number of useful subchannels. The algorithm is meant to perform LMS-based estimation to find the pair $(\mathbf{w}_i, \lambda_i)$. The step size μ_i controls the convergence speed and accuracy. The deflation process cancels the information of the pair $(\mathbf{w}_i, \lambda_i)$ for the estimation of next pair $(\mathbf{w}_{i+1}, \lambda_{i+1})$. The blind-tracking and deflation process continues until all pairs are estimated. The singular pairs (\mathbf{u}_i, σ_i) of channel matrix \mathbf{H} can be derived by the use of the pairs $(\mathbf{w}_i, \lambda_i)$ as follows:

$$\sigma_i = \sqrt{\lambda_i}, \quad \mathbf{u}_i = \frac{\mathbf{w}_i}{\sqrt{\lambda_i}}, \quad i = 1, 2, \dots, d. \quad (3)$$

The adaptive blind-tracking U Σ algorithm for 4 \times 4 antenna system as shown in Fig. 2 was implemented in [19]. The forgetting factor β is set to 1. Hence, the autocorrelation matrix is estimated by using the instantaneous received signals only. This reduces the computational complexity at the expense of additional square root and division. The step size is adaptively adjusted as $0.05/\lambda_i$.

III. PROPOSED ADAPTIVE SVD ALGORITHM

In many MIMO OFDM-based communication standards, the channel matrix \mathbf{H} can be obtained through channel estimation [23]–[25]. With this additional information, we propose a complete adaptive SVD algorithm for high-throughput MIMO OFDM-based applications. The BER performance may be affected by imperfect channel estimation, \mathbf{H} , and the degradation discussed in referenced works [23]–[25] about the channel estimation which is beyond the scope of this paper.

A. Derivation of Matrix \mathbf{R}_1

In Algorithm 1, the positive semidefinite matrix \mathbf{R}_1 is estimated by a moving average of the recent received signal vectors. In many MIMO OFDM-based standards, the channel matrix \mathbf{H} is already known by channel estimation. Therefore, we can utilize the information to evaluate accurate \mathbf{R}_1

$$\mathbf{R}_1 = \begin{cases} \mathbf{H}^H \mathbf{H}, & N_R \geq N_T \\ \mathbf{H} \mathbf{H}^H, & N_R < N_T. \end{cases} \quad (4)$$

With this definition of \mathbf{R}_1 , we can still use the same update and deflation process to find the pairs $(\mathbf{w}_i, \lambda_i)$ sequentially. In the i th update process, we have

$$\begin{aligned} \mathbf{w}_i(n+1) &= \mathbf{w}_i(n) + \mu_i(\mathbf{R}_i - \lambda_i(n)\mathbf{I})\mathbf{w}_i(n) \\ \lambda_i(n+1) &= \mathbf{w}_i(n+1)^H \mathbf{w}_i(n+1), \quad i = 1, 2, \dots, (d-1) \end{aligned} \quad (5)$$

where d is $\min(N_R, N_T)$, and the i th deflation process is given by

$$\mathbf{R}_{i+1} = \mathbf{R}_i - \mathbf{w}_i(n+1)\mathbf{w}_i(n+1)^H, \quad i = 1, 2, \dots, (d-1). \quad (6)$$

After convergence, we have \mathbf{w}_i and λ_i with $1 \leq i \leq (d-1)$. We can derive the singular values and the corresponding singular vectors of \mathbf{H} by using the pairs $(\mathbf{w}_i, \lambda_i)$. Since it is possible to have $N_R \geq N_T$ or $N_R < N_T$, there are two cases to be considered. For the case when $N_R \geq N_T$, we have

$$\sigma_i = \sqrt{\lambda_i}, \quad \mathbf{v}_i = \frac{\mathbf{w}_i}{\sqrt{\lambda_i}}, \quad \mathbf{u}_i = \frac{\mathbf{H}\mathbf{v}_i}{\sigma_i}, \quad i = 1, 2, \dots, (d-1). \quad (7)$$

On the other hand, when $N_R < N_T$, we only need to interchange \mathbf{v}_i with \mathbf{u}_i , and \mathbf{H} is changed to \mathbf{H}^H in (7).

B. Partial Update Scheme

In Algorithm 1, \mathbf{w}_d and λ_d are derived by applying the update operation. From our observation, after the $(d-1)$ -time deflation, the positive semi-definite matrix \mathbf{R}_d can be expressed as

$$\mathbf{R}_d = \mathbf{w}_d \mathbf{w}_d^H. \quad (8)$$

Hence, the update operation for \mathbf{w}_d and λ_d is unnecessary. We can directly find the d th singular value and the corresponding singular vectors by some simple operations. For the case of $N_R \geq N_T$, we get

$$\sigma_d = \sqrt{\text{tr}(\mathbf{R}_d)}, \quad \mathbf{v}_d = \frac{\mathbf{R}_d(:, 1)}{\|\mathbf{R}_d(:, 1)\|}, \quad \mathbf{u}_d = \frac{\mathbf{H}\mathbf{v}_d}{\sigma_d}. \quad (9)$$

On the other hand, when $N_R < N_T$, we only need to interchange \mathbf{v}_d with \mathbf{u}_d , and \mathbf{H} is changed to \mathbf{H}^H in (9). The advantage of applying partial update is to effectively reduce the decomposing latency.

C. Adaptive Step Size Scheme

The step size μ_i is an important parameter for the convergence speed and stability of the algorithms. As mentioned in the Appendix, the objective function is a quartic function which is complicated (also mentioned in [19]) to derive the exact bound of the step size. We derive a loose bound by approximating the objective function from a quartic function

to quadratic function in Appendix. We have derived a convergence region and a near-optimal step size as follows:

$$0 < \mu_i < \frac{1}{\lambda_i} \quad (10)$$

and

$$\mu_i = \frac{2}{3\lambda_i - \lambda_{i+1}} > \frac{2}{3\lambda_i}. \quad (11)$$

Hence, fixed step size is inefficient and not robust for all kinds of channel matrices. In [19], the step size is adaptively adjusted as $0.05/\lambda_i(n)$ which is too small for fast convergence purposes. Therefore, for the goal of fast and stable convergence, the proposed adaptive step size is given by

$$\mu_i(n) = \frac{a}{\lambda_i(n)} \quad (12)$$

where a is a scaling factor. From (11), we suggest that the value of a could be 0.75 or 0.5 for hardware-friendly implementation.

D. SIS

In (5), we have to give the initial values of $\{\mathbf{w}(n)\}_{d-1}^{i=1}$ for each update process. Although the update equation in (5) surely converges with arbitrary initial values, choosing good initial values can help to speed up the update processes. We denote $\mathbf{w}_i(0)$ and $\mathbf{w}_i(\infty)$ as the initial and converged values of $\mathbf{w}_i(n)$. In a wireless MIMO-OFDM system, since two adjacent subcarriers often have similar channel matrices, one subcarrier's converged information is useful to its adjacent subcarrier. Therefore, if one subcarrier's $\{\mathbf{w}(\infty)\}_{d-1}^{i=1}$ is obtained, we can take the converged values as its adjacent subcarrier's initial values of $\{\mathbf{w}(n)\}_{d-1}^{i=1}$. It should be noted that pilot and null subcarriers will be skipped since they do not need SVD operations.

E. Gram-Schmidt Scheme for Nonsquare Matrix

Generally speaking, for an $N_R \times N_T$ channel matrix, we need to find d singular values, N_R left singular vectors, and N_T right singular vectors, where $d = \min(N_R, N_T)$. After applying the above schemes, we can find d singular values, d left singular vectors, and d right singular vectors. If the channel matrix is square, it means that $d = N_R = N_T$. Therefore, we can find all singular values and singular vectors. But for the case of nonsquare channel matrix, assume that $N_R > N_T$, we have $d = N_T$, there are still $(N_R - N_T)$ unsolved left singular vectors (i.e., $\mathbf{u}_{N_T+1}, \mathbf{u}_{N_T+2}, \dots, \mathbf{u}_{N_R}$) after applying the above schemes. On the other hand, when $N_R < N_T$, there are $(N_T - N_R)$ unsolved right singular vectors (i.e., $\mathbf{v}_{N_R+1}, \mathbf{v}_{N_R+2}, \dots, \mathbf{v}_{N_T}$). Note that both the cases are similar. To find these remaining vectors, recall that \mathbf{U} and \mathbf{V} are the unitary matrices, the column vectors in \mathbf{U} or \mathbf{V} are orthonormal to each other. That is

$$\langle \mathbf{u}_i, \mathbf{u}_j \rangle = 0, \quad \forall i \neq j \quad (13)$$

and

$$\langle \mathbf{v}_i, \mathbf{v}_j \rangle = 0, \quad \forall i \neq j. \quad (14)$$

Algorithm 2 Pseudo-code of the proposed adaptive SVD algorithm

Given \mathbf{H}, N_R, N_T
 $\mathbf{R}_1 = \begin{cases} \mathbf{H}^H \mathbf{H}, & N_R \geq N_T \\ \mathbf{H} \mathbf{H}^H, & N_R < N_T \end{cases}$
 $d = \min(N_R, N_T)$
 1) Update and Deflation
 for $i = 1 : (d-1)$,
 Initial setting
 $\mathbf{w}_i(0) = \text{adjacent subcarrier's } \mathbf{w}_i(\infty)$
 $\lambda_i(0) = \mathbf{w}_i(0)^H \mathbf{w}_i(0)$
 $\mu_i(0) = a / \lambda_i(0)$
 Update the i -th pair
 $\mathbf{w}_i(n+1) = \mathbf{w}_i(n) + \mu_i(n)(\mathbf{R}_i - \lambda_i(n)\mathbf{I})\mathbf{w}_i(n)$
 $\lambda_i(n+1) = \mathbf{w}_i(n+1)^H \mathbf{w}_i(n+1)$
 $\mu_i(n+1) = a / \lambda_i(n+1)$
 Apply deflation
 $\mathbf{R}_{i+1} = \mathbf{R}_i - \mathbf{w}_i(n+1)\mathbf{w}_i(n+1)^H$
 end
 2) Derivation of $\mathbf{u}_i, \mathbf{v}_i, \sigma_i, i = 1, 2, \dots, (d-1)$
 if $N_R \geq N_T$
 $\sigma_i = \sqrt{\lambda_i}, \mathbf{v}_i = \mathbf{w}_i / \sqrt{\lambda_i}, \mathbf{u}_i = \mathbf{H} \mathbf{v}_i / \sigma_i$
 else
 $\sigma_i = \sqrt{\lambda_i}, \mathbf{u}_i = \mathbf{w}_i / \sqrt{\lambda_i}, \mathbf{v}_i = \mathbf{H}^H \mathbf{u}_i / \sigma_i$
 end
 3) Partial Update for $\mathbf{u}_d, \mathbf{v}_d, \sigma_d$
 if $N_R \geq N_T$
 $\sigma_d = \sqrt{\text{tr}(\mathbf{R}_d)}, \mathbf{v}_d = \mathbf{R}_d(:,1) / \|\mathbf{R}_d(:,1)\|, \mathbf{u}_d = \mathbf{H} \mathbf{v}_d / \sigma_d$
 else
 $\sigma_d = \sqrt{\text{tr}(\mathbf{R}_d)}, \mathbf{u}_d = \mathbf{R}_d(:,1) / \|\mathbf{R}_d(:,1)\|, \mathbf{v}_d = \mathbf{H}^H \mathbf{u}_d / \sigma_d$
 end
 4) Gram-Schmidt for remaining singular vectors
 if $N_R \geq N_T$
 for $k = 1 : (N_R - N_T)$
 $\mathbf{w}_{d+k} = \mathbf{e}_k - \sum_{i=1}^{d+k-1} \langle \mathbf{e}_k, \mathbf{u}_i \rangle \cdot \mathbf{u}_i$
 $\mathbf{u}_{d+k} = \frac{\mathbf{w}_{d+k}}{\|\mathbf{w}_{d+k}\|}$
 end
 else
 for $k = 1 : (N_T - N_R)$
 $\mathbf{w}_{d+k} = \mathbf{e}_k - \sum_{i=1}^{d+k-1} \langle \mathbf{e}_k, \mathbf{v}_i \rangle \cdot \mathbf{v}_i$
 $\mathbf{v}_{d+k} = \frac{\mathbf{w}_{d+k}}{\|\mathbf{w}_{d+k}\|}$
 end
 end
 end

Therefore, the remaining vectors can be obtained by applying the Gram-Schmidt technique [8]. First, we consider the case of $N_R > N_T$. After applying the above schemes, we already have $\mathbf{u}_1, \mathbf{u}_2, \dots$, and \mathbf{u}_{N_T} . Then the remaining left singular vectors can be obtained by

$$\begin{aligned} \mathbf{w}_{d+k} &= \mathbf{e}_k - \sum_{i=1}^{d+k-1} \langle \mathbf{e}_k, \mathbf{u}_i \rangle \cdot \mathbf{u}_i \\ \mathbf{u}_{d+k} &= \frac{\mathbf{w}_{d+k}}{\|\mathbf{w}_{d+k}\|}, \quad k = 1, 2, \dots, (N_R - N_T) \end{aligned} \quad (15)$$

where \mathbf{e}_k is orthonormal to \mathbf{e}_j with $k \neq j$, and \mathbf{e}_k is unequal to \mathbf{u}_i with $1 \leq i \leq (d+k-1)$. Note that for the case of $N_R < N_T$, we only need to replace \mathbf{u}_i with \mathbf{v}_i and to interchange N_R with N_T in (15). The proposed adaptive SVD algorithm is summarized in Algorithm 2.

IV. ARCHITECTURE DESIGN OF PROPOSED SVD ENGINE

The block diagram of the proposed reconfigurable adaptive SVD engine is depicted in Fig. 3. There are two single-port SRAM banks in the memory module, and four $16 \text{ entries} \times 80$ bits memory banks in the \mathbf{H} buffers. The detailed word length

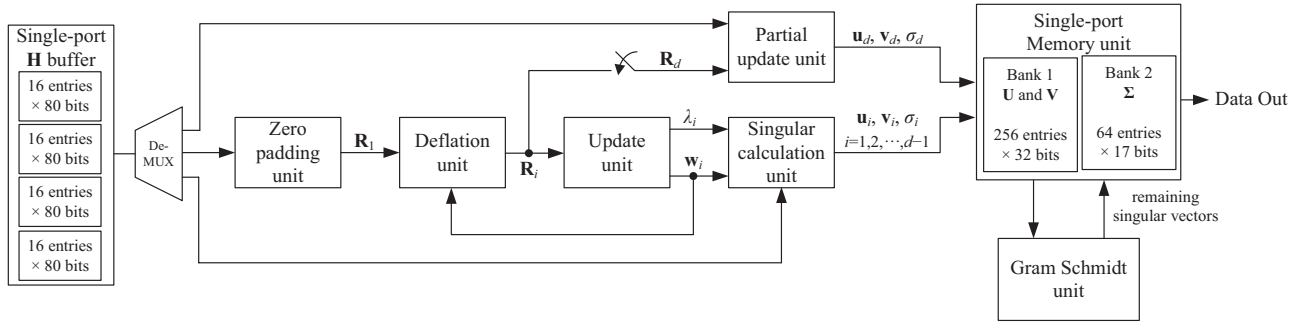


Fig. 3. Block diagram of the proposed reconfigurable adaptive SVD engine.

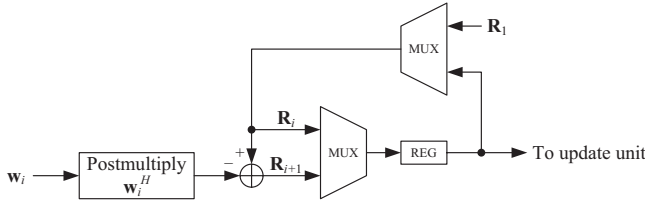


Fig. 4. Block diagram of deflation unit.

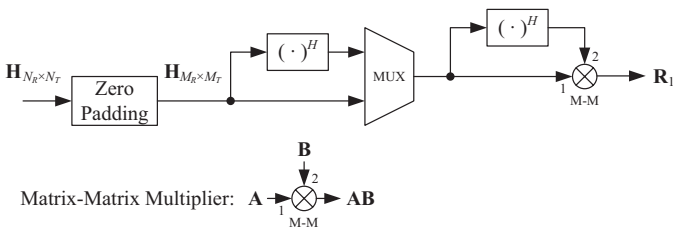


Fig. 5. Block diagram of zero padding unit.

consideration of the architecture and memory banks will be discussed in Section VI-D. It consists of six functional units which are zero padding unit, deflation unit, update unit, singular calculation unit, partial update unit, and simplified Gram-Schmidt unit. We could implement deflation unit directly and the block diagram of deflation unit derived from (6) is shown in Fig. 4. The register REG is used to store all entries of the positive semi-definite matrix. In the first update process, $\mathbf{R}_i = \mathbf{R}_1$. After the first update process, \mathbf{R}_{i+1} is derived from \mathbf{R}_i . In the remainder of this section, each unit will be described in more detail.

A. Reconfigurable Design for Different Size of Channel Matrix

In a MIMO system, assume that the maximum number of transmitter and receiver antennas is M_R and M_T , respectively. This means that we have possibly $M_R \times M_T$ different sizes of channel matrices (i.e., $1 \times 1, 1 \times 2, \dots, M_R \times M_T$). Therefore, we propose a reconfigurable scheme to support all antenna configurations.

1) *Zero Padding Scheme for Square and Nonsquare Channel Matrix:* The maximum size of channel matrix is $M_R \times M_T$ in a MIMO system. Hence, it is intuitive to design an SVD engine to support the maximum channel size. For the smaller channel matrix, we can extend it to the maximum-size channel matrix by inserting zeros. If the size of a given matrix is

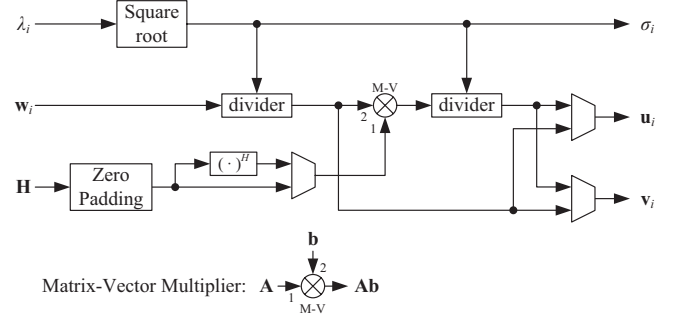


Fig. 6. Block diagram of singular calculation unit.

$N_R \times N_T$, the extended channel matrix is

$$\mathbf{H}_{\text{extended}} = \begin{bmatrix} \mathbf{H}_{N_R \times N_T} & \mathbf{0}_{N_R \times (M_T - N_T)} \\ \mathbf{0}_{(M_R - N_R) \times N_T} & \mathbf{0}_{(M_R - N_R) \times (M_T - N_T)} \end{bmatrix}_{M_R \times M_T}. \quad (16)$$

After extending the original channel matrix by inserting zeros, the SVD operation of the original channel is exactly the same as that of the maximum-size channel matrix. The extended channel shown in the referenced works [19], [21], and [22] support the antenna configurations after some modifications based on their own SVD algorithms. Note that the value of d in Algorithm 2 depends on the size of the original channel matrix. Therefore, d is still equal to $\min(N_R, N_T)$. Fig. 5 shows the block diagram of zero padding unit. A given channel matrix $\mathbf{H}_{N_R \times N_T}$ is extended to $\mathbf{H}_{M_R \times M_T}$ by inserting zeros, and the multiplexer is used to construct the positive semi-definite matrix \mathbf{R}_1 based on (4). We also apply the zero padding scheme to singular calculation unit and partial update unit. According to (7), Fig. 6 illustrates the architecture of singular calculation unit. Three multiplexers is used to consider two cases of $N_R \geq N_T$ and $N_R < N_T$. We employ (9) to realize partial update unit as shown in Fig. 7.

2) *Simplified Gram-Schmidt Scheme for Nonsquare Channel Matrix:* In (15), we apply the Gram-Schmidt technique to find the remaining vectors for the case of $N_R > N_T$. Due to the fact that the entries of a channel matrix, as well as the entries of its singular vectors, are always complex-valued, we can define \mathbf{e}_k as a unit vector with the k th entry being 1. With this setting, we can rewrite (15) into a more simplified

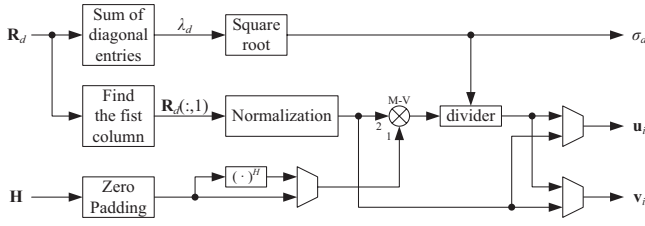


Fig. 7. Block diagram of partial update unit.

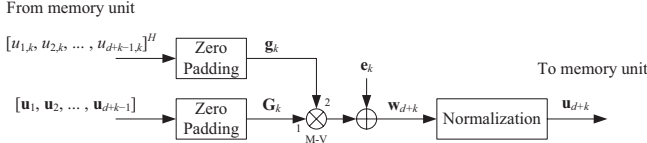
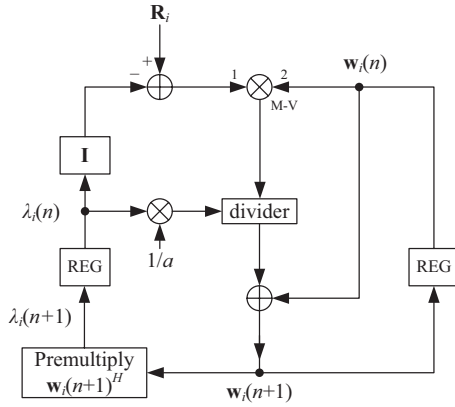
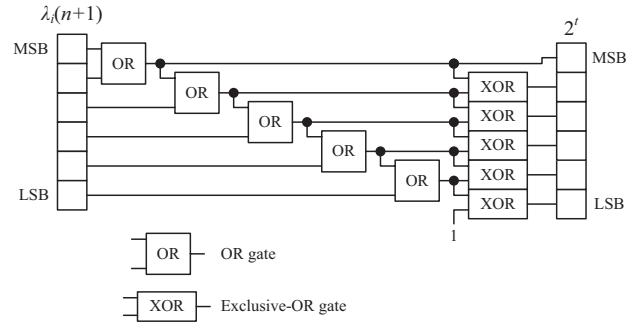
Fig. 8. Block diagram of Gram-Schmidt unit for the case of $N_R > N_T$.

Fig. 9. Block diagram of the original update unit.

form

$$\begin{aligned}
 \mathbf{w}_{d+k} &= \mathbf{e}_k - \sum_{i=1}^{d+k-1} u_{i,k}^* \cdot \mathbf{u}_i \\
 &= \mathbf{e}_k - [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{d+k-1}] [u_{1,k}, u_{2,k}, \dots, u_{d+k-1,k}]^H \\
 &= \mathbf{e}_k - \mathbf{G}_k \mathbf{g}_k \\
 \mathbf{u}_{d+k} &= \frac{\mathbf{w}_{d+k}}{\|\mathbf{w}_{d+k}\|}, \quad k = 1, 2, \dots, (N_R - N_T)
 \end{aligned} \quad (17)$$

where $u_{i,k}$ means the k -th element of \mathbf{u}_i . Note that for the case of $N_R < N_T$, we only need to replace \mathbf{u}_i with \mathbf{v}_i and to interchange N_R with N_T in (17). After this simplification, it is easier to implement a reconfigurable Gram-Schmidt design for different sizes of channel matrices. We can choose the maximum size of \mathbf{e}_k , \mathbf{G}_k , and \mathbf{g}_k in advance. In an $M_R \times M_T$ MIMO system, the maximum size of \mathbf{e}_k , \mathbf{G}_k , and \mathbf{g}_k is $L \times 1$, $L \times (L - 1)$, and $(L - 1) \times 1$ respectively, where $L = \max(M_R, M_T)$. For smaller-size antenna configurations, we just need to insert zeros in \mathbf{e}_k , \mathbf{G}_k , and \mathbf{g}_k . The block diagram of Gram-Schmidt unit is shown in Fig. 8 for the case of $N_R > N_T$. Gram-Schmidt unit needs to be executed $(N_R - N_T)$ times to find all remaining singular vectors. Note that the computational complexity of simplified Gram-Schmidt scheme is greatly smaller than that of original Gram-Schmidt algorithm.

Fig. 10. Mapping circuit that transforms $\lambda_i(n)$ into a number of powers of two.

B. Architectural Design of Update Unit

The main computational time of our SVD architecture is in the update unit. Fig. 9 shows the block diagram of the original update unit based on (5) and (12). For the architectural design of the update unit, we propose three schemes to reduce the decomposing latency and enhance the hardware utilization.

1) *Division-Free Adaptive Step Size Scheme*: In order to achieve fast convergent purpose, the step size $\mu_i(n)$ is adaptively adjusted with $\lambda_i(n)$. Obviously, in Fig. 9, there is a division at every iteration in the update unit. This will slow down the operating speed. For this reason, we propose a division-free adaptive step size scheme to avoid the division in the update operation. Due to the property of the step size [9], we do not need to calculate the exact value of $\mu_i(n)$. From (12), the step size is in inverse proportion to $\lambda_i(n)$, if we transform $\lambda_i(n)$ into a number of powers of 2, which is the nearest to and greater than $\lambda_i(n)$. Hence, the new step size can be expressed as

$$\mu'_i(n) = \frac{a}{2^t} \quad (18)$$

where t is an integer, and its value depends on the word-length of $\lambda_i(n)$. Since the new step size is a number of the power of 2, a shift operation can be substituted for a division at every iteration in the update operation. Fig. 10 shows the mapping circuit that transforms $\lambda_i(n)$ into a number of the power of 2, and the block diagram of the update unit with division-free adaptive step size scheme is shown in Fig. 11. Also note that

$$0 < \mu'_i(n) \leq \mu_i(n). \quad (19)$$

The stability of convergence is still guaranteed. Although the number of converged iterations increases, the required time at every iteration can be reduced effectively. Hence, the overall latency is reduced.

2) *Early Termination Scheme*: In (5), the correction vector for $\mathbf{w}_i(n + 1)$ is given by

$$\Delta \mathbf{w}_i(n) = \mu_i (\mathbf{R} - \lambda_i(n) \mathbf{I}) \mathbf{w}_i(n). \quad (20)$$

For a floating-point view, $\Delta \mathbf{w}_i(n)$ is always nonzero. However, for a fixed-point implementation, if every entry of $\Delta \mathbf{w}_i(n)$ satisfies the following condition:

$$\Delta w_{i,k}(n) < 2^{-(\text{Fractional Length of } \mathbf{w}_i(n))} \quad (21)$$

where $\Delta w_{i,k}(n)$ is the k th element of $\Delta \mathbf{w}_i(n)$. Then $\Delta \mathbf{w}_i(n)$ can be considered as a vector with all elements being zeros

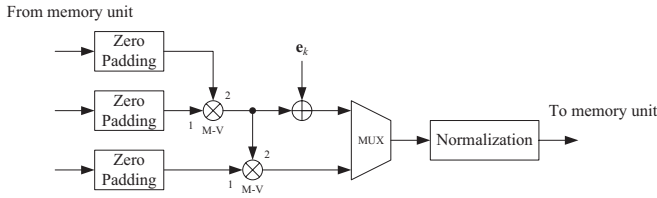


Fig. 14. Block diagram of Gram-Schmidt unit for nonsquare matrix and OR.

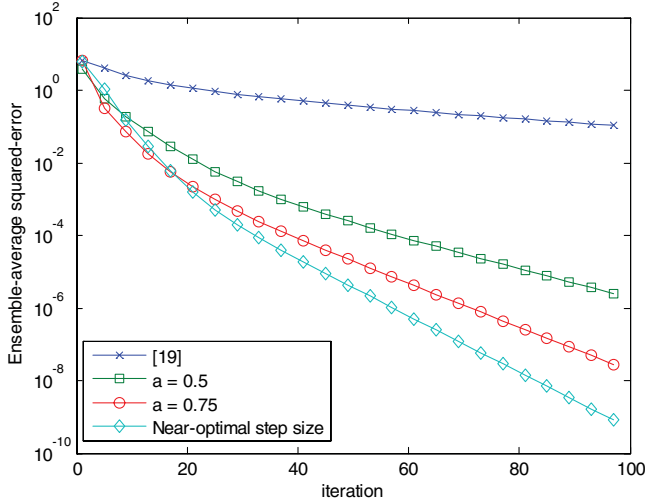


Fig. 15. Convergence rate of different step sizes in the first update process.

only need to replace \mathbf{u}_i with \mathbf{v}_i in (25). After applying OR to all singular vectors, most interference caused by the inaccurate deflation processes can be eliminated.

For the architecture of OR, we have to modify Gram-Schmidt unit in Fig. 8. We rewrite the second equation in (25) into a more compacted form

$$\hat{\mathbf{u}}_i = [\mathbf{u}_i \ \mathbf{u}_{Or,1} \ \cdots \ \mathbf{u}_{Or,i-1}] \begin{pmatrix} \mathbf{u}_i^H \\ -\mathbf{u}_{Or,1}^H \\ \vdots \\ -\mathbf{u}_{Or,i-1}^H \end{pmatrix} \mathbf{u}_i. \quad (26)$$

The operation in (26) can be executed by two successive matrix-vector multipliers. Based on (17), (25), and (26), Fig. 14 shows the block diagram of Gram-Schmidt unit with some modification. The multiplexer is used for considering two cases of nonsquare channel matrix and orthogonal reconstruction. Compared with Figs. 4, 8, 9, 11, 13, and 14 are structures with data interleaving scheme for throughput enhancement in hardware consideration.

VI. PERFORMANCE EVALUATION AND IMPLEMENTATION RESULTS

A. Convergence Rate of Different Adaptive Step Sizes

Using larger step size may cause unstable problem, and the system may fail due to the nonconvergence problem. Therefore, we have derived a near-optimal adaptive step size in (10) and (11) according to the Appendix. The proposed adaptive step size makes the iterative updating have both fast and stable convergence. We compare four step sizes:

TABLE I

AVERAGED REQUIRED ITERATIONS IN UPDATING EACH PAIR $(\mathbf{w}_i, \lambda_i)$ FOR $\mu_i(n) = 0.5/\lambda_i(n)$ WITH AND WITHOUT THE SIS

$\mu_i(n) = 0.5/\lambda_i(n)$	$(\mathbf{w}_1, \lambda_1)$	$(\mathbf{w}_2, \lambda_2)$	$(\mathbf{w}_3, \lambda_3)$	Total iterations	Savings
SIS excluded	26.5	19.7	14.1	60.3	-
SIS included	20.4	16.3	12.1	48.8	19.1%

TABLE II

AVERAGED REQUIRED ITERATIONS IN UPDATING EACH PAIR $(\mathbf{w}_i, \lambda_i)$ FOR $\mu_i(n) = 0.75/\lambda_i(n)$ WITH AND WITHOUT THE SIS

$\mu_i(n) = 0.75/\lambda_i(n)$	$(\mathbf{w}_1, \lambda_1)$	$(\mathbf{w}_2, \lambda_2)$	$(\mathbf{w}_3, \lambda_3)$	Total iterations	Savings
SIS excluded	18.8	15.2	13.4	47.4	-
SIS included	14.4	12.3	10.5	37.2	21.5%

1) $\mu_i(n) = 0.05/\lambda_i(n)$ in [19]; 2) the proposed $\mu_i(n) = 0.5/\lambda_i(n)$; 3) the proposed $\mu_i(n) = 0.75/\lambda_i(n)$; and 4) the near-optimal step size in (11). Assume that the entries of a channel matrix \mathbf{H} are independent and identically distributed (i.i.d.) according to $\mathcal{CN}(0, 1)$, where $\mathcal{CN}(0, 1)$ denotes the complex Gaussian distribution with independent real and imaginary parts distributed according to $\mathcal{N}(0, 1)$. We define the instantaneous error $e(n)$ as

$$e(n) = \|\mathbf{w}_1(n) - \mathbf{w}_{1,opt}\| \quad (27)$$

where $\mathbf{w}_{1,opt}$ is the optimal vector of \mathbf{H} in the first update process. We only consider the first update process since the subsequent update processes have similar results. Fig. 15 compares the convergence rate of different step sizes over 1000 independent channel realizations. The proposed adaptive step size is not only guaranteed to have stable convergence rate but also much faster than the step size $0.05/\lambda_i(n)$ in [19]. Note that at the early stage of total iterations, the proposed step size has faster convergence rate than the near-optimal step size. It is reasonable since the near-optimal step size has optimal convergence speed only when the current vector is close to the optimal vector.

B. Effect of the SIS

We apply the proposed reconfigurable adaptive SVD engine to the IEEE 802.11n applications. To determine the word-lengths in our design, we performed extensive floating point simulation and dynamic range analysis. We list the word-lengths of some key signals used in the fixed-point simulation and chip implementation are shown in the form (integer, fractional). The word-length of real part or imaginary part of each entry of \mathbf{H} , \mathbf{R}_i , \mathbf{w}_i , λ_i , \mathbf{u}_i , \mathbf{v}_i , and σ_i is (3, 7), (6, 14), (4, 12), (6, 26), (1, 7), (1, 7), and (4, 13), respectively.

To observe the effect of the SIS, we consider the channel model E [17], [27] in a 128-subcarrier 4×4 system. Assume that the division-free adaptive step size with the early termination scheme is applied. When the SIS has been included and

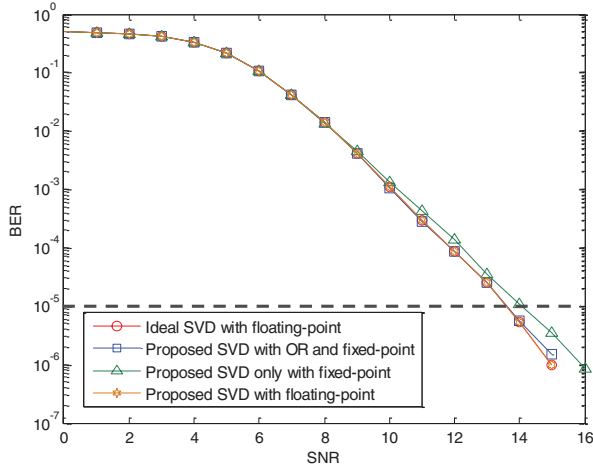


Fig. 16. System performance comparison at 4-QAM.

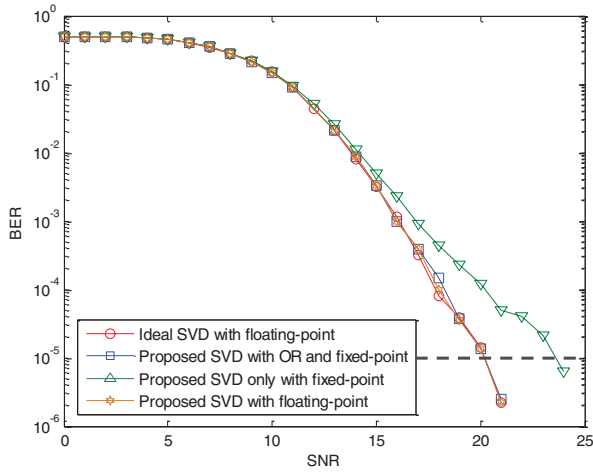


Fig. 17. System performance comparison at 16-QAM.

excluded, Tables I and II show the averaged required iteration number in updating each pair $(\mathbf{w}_i, \lambda_i)$ for $\mu_i(n) = 0.5/\lambda_i(n)$ and $\mu_i(n) = 0.75/\lambda_i(n)$, respectively. Note that with the partial update scheme, updating the last pair $(\mathbf{w}_4, \lambda_4)$ is unnecessary. As this shows, utilizing the SIS has the significant effect of reducing the total iterations by 19.1% and 21.5% for $\mu_i(n) = 0.5/\lambda_i(n)$ and $\mu_i(n) = 0.75/\lambda_i(n)$, respectively.

C. System Simulation

Before the system simulation, we have to determine the maximum iteration number in the update process. In Tables I and II, the first update process requires more iterations. If $\mu_i(n) = 0.5/\lambda_i(n)$, the mean and the standard deviation of the required iteration numbers in the first update process are 26.5 and 9.5. Therefore, in order to guarantee that almost all pairs $(\mathbf{w}_i, \lambda_i)$ are converged, we choose the maximum iteration number in each update process as 64 which is roughly equal to the sum of the mean and the four-times standard deviation. Then, the proposed SVD engine is applied to the IEEE 802.11n PHY system [17]. The performance metric is bit error rate (BER). The simulation environment settings are listed as follows.

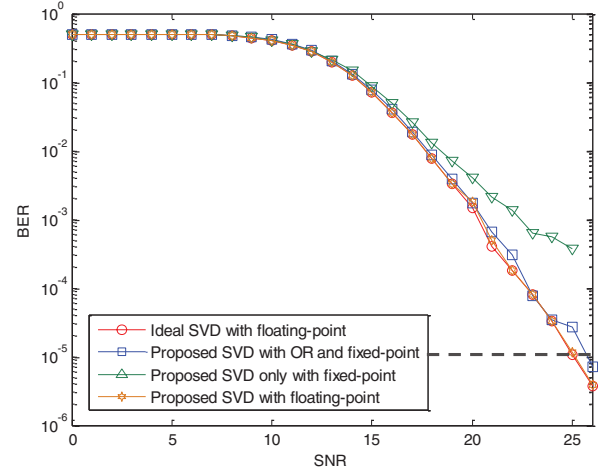


Fig. 18. System performance comparison at 64-QAM.

- 1) AWGN, Ch E (nLOS) channels [27], four spatial streams.
- 2) Assume perfect channel state information is obtained.
- 3) MIMO Technique: SVD.
- 4) Signal constellation: 4-QAM, 16-QAM, and 64-QAM.
- 5) FFT (IFFT) size: 128.
- 6) Code rate 1/2 convolutional code with constraint length 7, generator polynomials [133 171] [28].
- 7) Block interleaving is used.

The simulation result is shown in Figs. 16–18. Our target BER is 10^{-5} . In the floating-point view, the proposed SVD design has no performance loss compared with the ideal SVD. Without orthogonal compensation, the proposed SVD fixed-point design only works well at 4-QAM with a performance loss of 0.4 dB compared with the ideal SVD. If OR is applied to our SVD fixed-point design, there is no performance loss at 4-QAM and 16-QAM. Besides, in the signal constellation of 64-QAM, our SVD fixed-point design has little performance loss of 0.6dB compared with the ideal SVD.

D. Chip Implementation

For a baseline design, we adopt $\mu_i(n) = 0.5/\lambda_i(n)$ and the SIS is not applied. The memory banks of the channel matrices and SVD results are describe as follow. Assume 10-bit precision of each real or imaginary number in the channel matrix \mathbf{H} is given, 320 bits are required for storing one 4×4 complex matrix. To avoid memory access collision, total storages of 16 channel matrices are divided into 4 single-port memory banks. The columns of one channel matrix are stored in 4 different memory banks so that we are able to access one complete channel matrix per cycle. In summary, 416 entries \times 80 bits memory banks are required as channel matrix storage in our design.

There are two memory banks in the memory unit in Fig. 3 to store the elements of \mathbf{U} , \mathbf{V} , and Σ . Bank 1 is designed for \mathbf{U} and \mathbf{V} . We use 16-bit precision for each element in \mathbf{U} and \mathbf{V} . Two elements are stored in each entry of memory bank 1. Total entries required for bank 1 is 256, 16 elements \times 16 matrices, and the overall size is 256 entries \times 32 bits. Bank 2 is designed for storing Σ , the singular values, and the wordlength

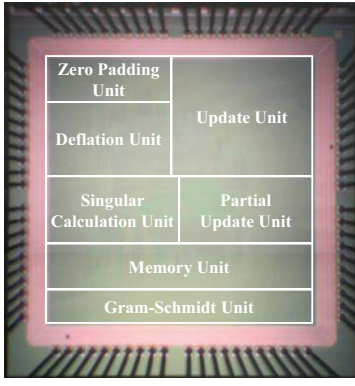


Fig. 19. Die photo of the proposed reconfigurable adaptive SVD engine design.

TABLE III
CHIP SUMMARY

Technology	UMC 90nm 1P9M Low-K Process
IO/core V_{DD}	3.3V/1.0V
Core area	1.475 mm \times 1.475 mm
Die area	2.22 mm \times 2.22 mm
Gate count	543.9k
Frequency	101.2 MHz (max)
Power consumption	125mW @101.2 MHz

of each singular value is 17 bits. Total entries required for bank 2 is 64, 4 elements \times 16 matrices, and the overall size is 64 entries \times 17 bits. In addition, the matrix-to-matrix multiplication is performed by the matrix-to-vector multiplier in 4 cycles.

The chip is fabricated in UMC 90 nm 1P9M Low-K CMOS technology and measured with Tektronix pattern generator TLA 715 and logic analyzer TLA 5203. Fig. 19 shows the die photo of the fabricated chip design. The chip feature is summarized in Table III. The core size is 1.475 mm \times 1.475 mm. The number of total gate counts is 543.9k. The die size is 2.22 mm \times 2.22 mm giving a total area of 4.93 mm². The maximum operating frequency is measured 101.2 MHz and the total power consumption is measured 125 mW for the 4 \times 4 SVD operations. In order to consider reduction of power consumption, we can reduce the core supply voltage to 0.65V as shown in Fig. 20. The corresponding maximum operating frequency and power consumption are 43.48 MHz and 22.1 mW, respectively.

For comparison, we use two performance indices. First, the throughput is defined by the number of channel matrices that the SVD engine can deal with per second

$$\text{Throughput} = \frac{\text{Number of processed channel matrices}}{\text{Time (s)}}. \quad (28)$$

In the worst updating cases of proposed SVD operation, there are 64 iterations for each singular pair updating without early termination scheme and do not have to update the last singular pair. We need 64 iteration per singular pair \times (4-1) singular pairs \times 16 matrices = 3072 cycles, and extra 308 cycles for other operations. The equivalent throughput is

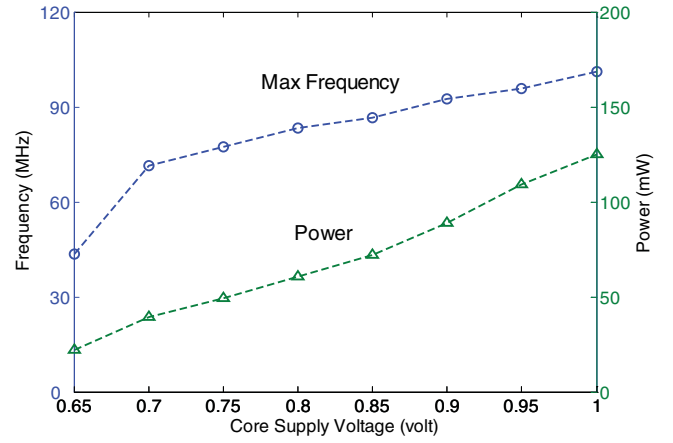


Fig. 20. Measured frequency and power of the chip design.

TABLE IV
COMPARISON TABLE

	JSSC'07 [19]	ACSSC'07 [27]	ISCAS'08 [28]	This Work
Support Antenna Configurations	4 \times 4	4 \times 4	4 \times 4	1 \times 1, 1 \times 2, 1 \times 3, 1 \times 4, 2 \times 1, 2 \times 2, 2 \times 3, 2 \times 4, 3 \times 1, 3 \times 2, 3 \times 3, 3 \times 4, 4 \times 1, 4 \times 2, 4 \times 3, 4 \times 4.
SVD	U and Σ	U, Σ , and V	V, and Σ	U, Σ , and V
Technology	90 nm	180 nm	180 nm	90 nm
Core Size	3.61 mm ²	0.41 mm ²	0.41 mm ²	2.17 mm ²
Gate Count	980k	42.3k	42.3k	543.9k
Frequency	100 MHz	133 MHz	149 MHz	101.2 MHz
Power	34mW@0.4 V	160mW@1.8 V	N/A	125 mW@1.0V
Throughput	50k	86.4k	303k	479.1k ^{*b}
Power Efficiency	1.47	3.50	N/A	3.83

*^b Consider 4 \times 4 SVD operation.

The power consumptions of the proposed SVD with 1 \times 1, 2 \times 2 and 3 \times 3 matrices are about 12mW, 33mW and 74mW, respectively.

derived as $16/[3380 \text{ cycles} \times (1/101.2 \text{ MHz})] = 479.05\text{k-matrices/sec}$. For 16 mxm channel matrices, the total cycles required are $[16 \times 64 \times (m-1) + 308]$ cycles. There are 308, 1332, 2356, and 3380 cycles required when processing 16 1 \times 1, 2 \times 2, 3 \times 3, and 4 \times 4 matrices respectively. In other words, the equivalent throughputs are 5.3 M, 1.2 M, 687 k, and 479 k matrices/sec for 1 \times 1, 2 \times 2, 3 \times 3, and 4 \times 4 matrices, respectively.

Then the power efficiency can be expressed as

$$\text{Power Efficiency} = \frac{\text{Throughput (k)}}{\text{Power consumption (mW)}}. \quad (29)$$

The technology scaling of power from 180 nm@1.8 V to 90 nm@1.0 V is given by $P_{90} = P_{180} \times (C_{90}/C_{180}) \times (V_{90}/V_{180})^2 = P_{180} \times 0.5 \times (1.0/1.8)^2 = P_{180} \times 0.1543$. The proposed reconfigurable adaptive SVD engine design is compared with other designs as shown in Table IV. An SVD chip without the need of CSI was proposed in [19]. The block-type pilots are utilized in the IEEE 802.11n systems for training symbol-based channel estimation of each subcarrier. The least-square and minimum-mean-square-error techniques [30] are widely used for channel estimation when training symbols are available. The complexity is fairly low owing to no matrix inversion required in channel estimation with pre-defined orthogonal training sets [17]. SVD in [19] only supports the 4 \times 4 antenna system and implements the U Σ algorithm

which is not complete for SVD. The overall computational complexity of the SVD in [19] is proportional to the iteration number required which is about 500. By applying the proposed adaptive step size and partial update schemes in our proposed design, the iteration number required per matrix in our design is $3380/16 \approx 212$ at most. In addition, the average iteration number can be further reduced by 20% with the proposed SIS as shown in Tables I and II. An improved design of [21] can be considered as [23], but it only computes \mathbf{V} and Σ which are partial of SVD outputs. Our design is able to handle 16×4 channel matrices at the same time.

Compared with other related works, only our work can support all antenna configurations in a MIMO system. Among all designs, our SVD chip has the highest throughput and power efficiency in the 4×4 SVD operations. In addition, the chip result shows that in an 802.11n system with 128 subcarriers, the average latency of our SVD chip is only 0.33% of the WLAN coherence time. Therefore, our SVD engine design is very suitable for high-throughput wireless communication applications.

In order to effectively enhance the throughput, we can use larger adaptive step size and apply the SIS to our SVD engine. First, we replace $\mu_i(n) = 0.5/\lambda_i(n)$ with $\mu_i(n) = 0.75/\lambda_i(n)$. This costs four additional complex adders in hardware implementation. Second, by applying the SIS, the registers in the update state can hold the converged values of the previous subcarriers until their adjacent subcarriers' channel information comes. Hence, additional multiplexers are required in hardware implementation. If $\mu_i(n) = 0.75/\lambda_i(n)$ and the SIS is applied, the mean and the standard deviation of the required iteration numbers in the first update process are 14.4 and 5.6, respectively. Therefore, we can choose the maximum iteration number in each update process as 36 which is roughly equal to the sum of the mean and four times the standard deviation. Note that the SVD operations for the first 16 subcarriers, the maximum iteration number in each update process, should be bigger since no additional information could speed up the convergence time. With this scenario, the throughput of our SVD engine can be enhanced to 850 k with little extra hardware cost.

We used the clock gating scheme to turn off the unused multipliers with smaller channel matrices. The power consumption is not directly related to the operating cycles, but related to the executed operation per cycle in average. The main operation in the proposed SVD algorithm is matrix-to-vector multiplication whose complexity is proportional to N^2 , where N is the length of the vector. Owing to the leakage power and other common operations in different matrix sizes, the power consumption of $1 \times 1 \sim 4 \times 4$ matrices are 12, 33, 74, and 125 mW, respectively. The corresponding power consumptions of processing nonsquare matrices are close to that of square matrices with size of $\min(\text{row}, \text{col.})$, where *row* and *col.* are the numbers of rows and columns of the channel matrices.

In summary, a reconfigurable SVD for different antenna sets and deriving all singular vectors is required for the application to IEEE 802.11n systems. The throughput requirement is also high. Compared with the referenced work in [19], our SVD engine is able to achieve the goals mentioned above. For

the throughput consideration, we proposed the adaptive step size, partial update scheme, and SIS to accelerate the overall processing. The throughput and power efficiency is about 9 times and 2.6 times than that in [19], respectively. The throughput improvement with SIS is about 20% as shown in Tables I and III. The proposed design with OR scheme is able to be 4 dB better at least compared with the design without OR scheme as shown in Figs. 17 and 18.

VII. CONCLUSION

This paper presented a reconfigurable adaptive SVD engine design for MIMO-OFDM systems. The proposed architectural design techniques can lower the computational complexity, effectively reduce the decomposing latency, and support all antenna configurations in a MIMO system. These design strategies enable the use of SVD to be effectively applied to the high-throughput wireless communication applications. Our SVD engine is implemented in UMC 90-nm CMOS technology for the application of IEEE 802.11n systems with 16 antenna configurations. The proposed SVD engine achieves a higher throughput rate than that of other related works. Moreover, the chip result shows that for an 802.11n system, the average latency of our SVD engine is only 0.33% of the WLAN coherence time. Therefore, the proposed SVD engine is very suitable for the high-throughput MIMO-OFDM applications.

APPENDIX

We show the detailed derivations of (10) and (11). Assume that the matrix $\mathbf{R} \in \mathbb{C}^{d \times d}$ is a positive semi-definite matrix. The eigenvalue decomposition of \mathbf{R} can be expressed as

$$\begin{aligned} \mathbf{R} &= \mathbf{U} \mathbf{\Lambda} \mathbf{U}^H \\ &= [\mathbf{u}_1 \ \mathbf{u}_2 \ \cdots \ \mathbf{u}_d] \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_d \end{bmatrix} [\mathbf{u}_1 \ \mathbf{u}_2 \ \cdots \ \mathbf{u}_d]^H \end{aligned} \quad (\text{A.1})$$

where $\mathbf{\Lambda}$ is a $d \times d$ matrix with only real and nonnegative main diagonal entries. The entry (i, i) of $\mathbf{\Lambda}$ denotes the i th largest eigenvalue λ_i , with $i = 1, 2, \dots, d$. The i th column vector \mathbf{u}_i in \mathbf{U} is called the i th eigenvector corresponding to the i th largest eigenvalue λ_i .

Consider the objective function $J(\mathbf{w})$

$$J(\mathbf{w}) = \frac{1}{2} \mathbf{w}^H \mathbf{R} \mathbf{w} - \frac{1}{4} (\mathbf{w}^H \mathbf{w})^2. \quad (\text{A.2})$$

In [20], the authors have proved that all the stationary point of $J(\mathbf{w})$ are eigenvectors of \mathbf{R} with magnitude being the square root of the corresponding eigenvalue of \mathbf{R} . Besides, if the dominant eigen pair is of multiplicity one, the dominant eigen pair is the global maximum point of $J(\mathbf{w})$.

Taking the gradient of $J(\mathbf{w})$, we have

$$\nabla_{\mathbf{w}} J(\mathbf{w}) = \mathbf{R} \mathbf{w} - (\mathbf{w}^H \mathbf{w}) \mathbf{w}. \quad (\text{A.3})$$

To maximize the objective function $J(\mathbf{w})$, it is straightforward to apply the steepest-descent techniques [9]. The updated formula is given by

$$\begin{aligned}\mathbf{w}(n+1) &= \mathbf{w}(n) + \mu \cdot \nabla_{\mathbf{w}} J(\mathbf{w}) \\ &= \mathbf{w}(n) + \mu \left(\mathbf{R}\mathbf{w}(n) - (\mathbf{w}(n)^H \mathbf{w}(n))\mathbf{w}(n) \right) \\ &= \mathbf{w}(n) + \mu \left(\mathbf{R} - (\mathbf{w}(n)^H \mathbf{w}(n))\mathbf{I} \right) \mathbf{w}(n) \quad (\text{A.4})\end{aligned}$$

where μ is the step size. Generally speaking, the value of the step size directly impacts the convergence speed, stability, and accuracy of the adaptive algorithms. Since the objective function $J(\mathbf{w})$ is a fourth-order function in \mathbf{w} , the analysis of the step size is complicated. Hence, we will give a loose bound by approximating $J(\mathbf{w})$ to a quadratic function around the optimal point $\sqrt{\lambda_1}\mathbf{u}_1$. By invoking the second-order Taylor series expansion of $J(\mathbf{w})$ around the optimal point $\sqrt{\lambda_1}\mathbf{u}_1$, $J(\mathbf{w})$ can be approximated by

$$\begin{aligned}\hat{J}(\mathbf{w}) &= J(\sqrt{\lambda_1}\mathbf{u}_1) + \frac{1}{2}(\mathbf{w} - \sqrt{\lambda_1}\mathbf{u}_1)^H \nabla_{\mathbf{w}}^2 J(\sqrt{\lambda_1}\mathbf{u}_1) \\ &\quad \times (\mathbf{w} - \sqrt{\lambda_1}\mathbf{u}_1) \quad (\text{A.5})\end{aligned}$$

where is the Hessian of $J(\mathbf{w})$ which can be expressed as

$$\nabla_{\mathbf{w}}^2 J(\mathbf{w}) = \mathbf{R} - (\mathbf{w}^H \mathbf{w})\mathbf{I} - 2\mathbf{w}\mathbf{w}^H. \quad (\text{A.6})$$

By substituting the optimal point $\sqrt{\lambda_1}\mathbf{u}_1$ into (A.6), we obtain

$$\begin{aligned}\nabla_{\mathbf{w}}^2 J(\sqrt{\lambda_1}\mathbf{u}_1) &= \mathbf{U}\mathbf{\Lambda}\mathbf{U}^H - \lambda_1\mathbf{U}\mathbf{U}^H - 2\lambda_1\mathbf{u}_1\mathbf{u}_1^H \\ &= \mathbf{U} \left(\mathbf{\Lambda} - \lambda_1\mathbf{I} - 2 \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \right) \mathbf{U}^H \\ &= -\mathbf{U} \begin{bmatrix} 2\lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_1 - \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_1 - \lambda_d \end{bmatrix} \mathbf{U}^H \\ &= -\mathbf{U}\mathbf{T}\mathbf{U}^H. \quad (\text{A.7})\end{aligned}$$

By employing (A.5) and (A.7), the updated equation around the optimal point can be expressed as

$$\begin{aligned}\mathbf{w}(n+1) &= \mathbf{w}(n) + \mu \cdot \nabla_{\mathbf{w}} \hat{J}(\mathbf{w}), \\ &= \mathbf{w}(n) + \mu \cdot \nabla_{\mathbf{w}}^2 J(\sqrt{\lambda_1}\mathbf{u}_1)(\mathbf{w}(n) - \sqrt{\lambda_1}\mathbf{u}_1) \\ &= \mathbf{w}(n) - \mu\mathbf{U}\mathbf{T}\mathbf{U}^H(\mathbf{w}(n) - \sqrt{\lambda_1}\mathbf{u}_1). \quad (\text{A.8})\end{aligned}$$

We define the error vector at time n as

$$\mathbf{e}(n) = \mathbf{w}(n) - \sqrt{\lambda_1}\mathbf{u}_1. \quad (\text{A.9})$$

By using (A.9), (A.8) can be rewritten as

$$\mathbf{e}(n+1) = (\mathbf{I} - \mu\mathbf{U}\mathbf{T}\mathbf{U}^H)\mathbf{e}(n) = \mathbf{U}(\mathbf{I} - \mu\mathbf{T})\mathbf{U}^H\mathbf{e}(n). \quad (\text{A.10})$$

Pre-multiplying both sides of (A.10) by \mathbf{U}^H and using the property of the unitary matrix that \mathbf{U}^H equals the inverse of \mathbf{U} , we have

$$\begin{aligned}\mathbf{U}^H\mathbf{e}(n+1) &= \mathbf{U}^H\mathbf{U}(\mathbf{I} - \mu\mathbf{T})\mathbf{U}^H\mathbf{e}(n) \\ &= (\mathbf{I} - \mu\mathbf{T})\mathbf{U}^H\mathbf{e}(n). \quad (\text{A.11})\end{aligned}$$

We now define a new set of coordinates as follows:

$$\mathbf{c}(n) = \mathbf{U}^H\mathbf{e}(n). \quad (\text{A.12})$$

Accordingly, we may rewrite (A.11) in the transformed form

$$\mathbf{c}(n+1) = (\mathbf{I} - \mu\mathbf{T})\mathbf{c}(n). \quad (\text{A.13})$$

The initial value of $\mathbf{c}(n)$ equals

$$\mathbf{c}(0) = \mathbf{U}^H(\mathbf{w}(0) - \sqrt{\lambda_1}\mathbf{u}_1). \quad (\text{A.14})$$

For the k th entry of the vector $\mathbf{c}(n)$, we have

$$c_k(n+1) = (1 - \mu t_k)c_k(n), \quad k = 1, 2, \dots, d \quad (\text{A.15})$$

where t_k is the k th diagonal entry of \mathbf{T} . (A.15) is a homogeneous difference equation of the first order. Assume that $c_k(n)$ has the initial value $c_k(0)$, (A.15) can be rewritten as

$$c_k(n) = (1 - \mu t_k)^n c_k(0), \quad k = 1, 2, \dots, d. \quad (\text{A.16})$$

Since all the diagonal values of \mathbf{T} are positive and real, the response $c_k(n)$ will not have no oscillations. In addition, (A.16) represents a geometric series with a geometric ratio equal to $1 - \mu t_k$. For stability or convergence of the adaptive algorithm, the magnitude of this geometric ratio must be less than 1 for all k . That is

$$-1 < 1 - \mu t_k < 1, \quad k = 1, 2, \dots, d. \quad (\text{A.17})$$

Therefore, the necessary and sufficient condition for the stability or convergence of the adaptive algorithm is that the step size μ satisfies the following condition:

$$0 < \mu < \frac{2}{t_{\max}} \quad (\text{A.18})$$

where t_{\max} is the maximal diagonal entry of \mathbf{T} which is given by

$$t_{\max} = 2\lambda_1. \quad (\text{A.19})$$

By substituting (A.19) into (A.18), we have

$$0 < \mu < \frac{1}{\lambda_1}. \quad (\text{A.20})$$

Hence, (A.20) provides a useful bound for the stability or convergence of the adaptive algorithm.

To analyze the convergence speed of the adaptive algorithm, we define a time constant τ_k as the number of iterations required for $c_k(n)$ to decay to $1/e$ of its initial value $c_k(0)$, that is

$$c_k(n) = e^{-\frac{n}{\tau_k}} c_k(0), \quad k = 1, 2, \dots, d. \quad (\text{A.21})$$

From (A.16) and (A.21), the time constant τ_k can be expressed as

$$\tau_k = \frac{-1}{\ln |1 - \mu t_k|}, \quad k = 1, 2, \dots, d. \quad (\text{A.22})$$

Note that the time constant τ_k is the function of the step size μ . The first time constant τ_1 has the following properties:

$$\begin{cases} D_{\mu}\tau_1 > 0, & \text{if } 0 < \mu < \frac{1}{2\lambda_1} \\ D_{\mu}\tau_1 < 0, & \text{if } \frac{1}{2\lambda_1} < \mu < \frac{1}{\lambda_1} \end{cases} \quad (\text{A.23})$$

where is the derivative of τ_1 with respect to μ . According to (A.23), we know that τ_1 is a convex-like function in the

convergence region. In addition, other time constants have the following properties:

$$D_\mu \tau_k < 0, \text{ if } 0 < \mu < \frac{1}{\lambda_1} \quad (\text{A.24})$$

and

$$\tau_k \geq \tau_{k+1} \quad (\text{A.25})$$

where $k = 2, 3, \dots, d$. According to (A.24), $\{\tau_k\}_d^{k=2}$ are the decreasing curves in the convergence region. From (A.25), given a value of μ , the maximal time constant is either τ_1 or τ_2 . Therefore, we have to find a good step size to minimize the maximal time constant. This condition occurred at $\tau_1 = \tau_2$, that is

$$\frac{-1}{\ln|1 - \mu t_1|} = \frac{-1}{\ln|1 - \mu t_2|}. \quad (\text{A.26})$$

By employing (A.7) and (A.26) and solving for μ , we have

$$\mu_{\text{opt}} = \frac{2}{3\lambda_1 - \lambda_2}. \quad (\text{A.27})$$

It should be noted that μ_{opt} is a near-optimal step size since in (A.5) is an approximate function to describe $J(\mathbf{w})$ around the optimal point. As a result, (A.27) provides a good guideline in choosing a proper step size of the adaptive algorithm.

REFERENCES

- [1] 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," in *Proc. IEEE 43rd Veh. Technol. Conf.*, May 1993, pp. 508–511.
- [2] S. M. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 8, pp. 1451–1458, Oct. 1998.
- [3] 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.
- [4] J. H. Winters, J. Salz, and R. D. Gitlin, "The impact of antenna diversity on the capacity of wireless communication systems," *IEEE Trans. Commun.*, vol. 42, no. 234, pp. 1740–1751, Feb.–Apr. 1994.
- [5] 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.
- [6] I. E. Telatar, "Capacity of multi-antenna Gaussian channels," *Eur. Trans. Telecommun.*, vol. 10, no. 6, pp. 585–595, 1999.
- [7] G. G. Raleigh and J. M. Cioffi, "Spatio-temporal coding for wireless communication," *IEEE Trans. Commun.*, vol. 46, no. 3, pp. 357–366, Mar. 1998.
- [8] G.W. Stewart, *Introduction to Matrix Computations*. New York: Academic, 1973.
- [9] S. Haykin, *Adaptive Filter Theory*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1991.
- [10] F. Deprettere, *SVD and Signal Processing: Algorithms, Analysis and Applications*. Amsterdam, The Netherlands: Elsevier, 1988.
- [11] J. Laurila, K. Kopsa, R. Schurhuber, and E. Bonek, "Semi-blind separation and detection of co-channel signals," in *Proc. IEEE Int. Conf. Commun.*, vol. 1. Jun. 1999, pp. 17–22.
- [12] 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.
- [13] J. Ha, A. N. Mody, J. H. Sung, J. R. Barry, S. W. McLaughlin, and G. L. Stuber, "LDPC coded OFDM with alamouti/SVD diversity technique," *Wireless Personal Commun.*, vol. 23, no. 1, pp. 183–194, Oct. 2002.
- [14] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Standard P802.11n/D3.00, 2007.
- [15] R. Van Nee, V. K. Jones, G. Awater, A. Van Zelst, J. Gardner, and G. Steele, "The 802.11n MIMO-OFDM standard for wireless LAN and beyond," *Wireless Personal Commun.*, vol. 37, nos. 3–4, pp. 445–453, Jun. 2006.
- [16] Y. Xiao, "IEEE 802.11n: Enhancements for higher throughput in wireless LANs," *IEEE Wireless Commun.*, vol. 12, no. 6, pp. 82–91, Dec. 2005.
- [17] 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, Jan. 2008.
- [18] T. S. Rappaport, *Wireless Communications: Principle and Practice*, 1st ed. Englewood Cliffs, NJ: Prentice-Hall, 1996.
- [19] 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.
- [20] A. Poon, D. Tse, and R. W. Brodersen, "An adaptive multiantenna transceiver for slowly flat fading channels," *IEEE Trans. Commun.*, vol. 51, no. 11, pp. 1820–1827, Nov. 2003.
- [21] Y. G. Li, J. H. Winters, and N. R. Sollenberger, "MIMO-OFDM for wireless communications: Signal detection with enhanced channel estimation," *IEEE Trans. Commun.*, vol. 50, no. 9, pp. 1471–1477, Sep. 2002.
- [22] H. Minn and N. Al-Dhahir, "Optimal training signals for MIMO OFDM channel estimation," *IEEE Trans. Wireless Commun.*, vol. 5, no. 5, pp. 1158–1168, May 2006.
- [23] T. D. Chiueh and P. Y. Tsai, *OFDM Baseband Receiver Design for Wireless Communications*. New York: Wiley, 2007.
- [24] K. K. Parhi, *VLSI Digital Signal Processing Systems*. New York: Wiley, 1999.
- [25] M. Clark. (2003 Jun.). *IEEE 802.11a WLAN Model*. Mathworks, Inc., Natick, MA [Online]. Available: <http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=3540&objectType=file>
- [26] *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-physpecification.Doc>
- [27] C. Studer, P. Blösch, P. Friendli, and A. Burg, "Matrix decomposition architecture for MIMO systems: Design and implementation trade-offs," in *Proc. 41st Asilomar Conf. Signals, Syst., Comput.*, Nov. 2007, pp. 1986–1990.
- [28] C. Senning, C. Studer, P. Luethi, and W. Fichtner, "Hardware-efficient steering matrix computation architecture for MIMO communication system," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2008, pp. 304–307.
- [29] G. H. Golub and C. F. V. Loan, *Matrix Computations*, 3rd ed. Baltimore, MD: The Johns Hopkins Univ. Press, 1996.
- [30] Y. S. Cho, J. Kim, W. Y. Yang, and C. G. Kang, *MIMO-OFDM Wireless Communications with MATLAB*, New York: Wiley, 2010.



Yen-Liang Chen received the B.S. degree in communication engineering from National Chiao Tung University, Hsinchu, Taiwan, and the Ph.D. degree in electronic engineering from National Taiwan University, Taipei, Taiwan, in 2005 and 2011, respectively.

He is currently serving military duty in Taiwan. His current research interests include VLSI implementation of digital signal processing algorithms, adaptive filtering, reconfigurable architecture, and digital communication systems.



Cheng-Zhou Zhan received the B.S. and M.S. degrees in electronic engineering from National Taiwan University, Taipei, Taiwan, in 2005 and 2007, respectively, where he is currently pursuing the Ph.D. degree in electronic engineering.

His current research interests include the design of VLSI architectures and circuits for digital signal processing and communication systems.



Ting-Jyun Jheng received the B.S. degree from National Chiao Tung University, Hsinchu, Taiwan, and the M.S. degree from National Taiwan University, Taipei, Taiwan, in 2007 and 2009, respectively, both in electronic engineering.

He is currently an Engineer with MediaTek Inc., Hsinchu. His current research interests include the design of VLSI architectures and circuits for digital signal processing and 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.

He was a Technical Staff Member with AT&T Bell Laboratories, Murray Hill, NJ, from August 1995 to July 1996, working on high-speed transmission integrated circuit designs. From 1996 to 2000, he was with the Electrical Engineering Department, National Central University, Taoyuan, Taiwan. In

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 current research interests include low-power/high-performance VLSI architectures for digital signal processing and communication applications, adaptive/multi-rate signal processing, reconfigurable broadband access systems and architectures, and SoC platform for software/hardware co-design.

Dr. Wu was the recipient of the A-class Research Award from the National Science Council four times. He has served on many technical program committees of IEEE international conferences. He was an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART II: EXPRESS BRIEFS and the IEEE TRANSACTIONS ON SIGNAL PROCESSING.