Adaptive MIMO Decision Feedback Equalization for Receivers in Time-Varying Channels

Jihoon Choi[†], Heejung Yu[‡○], and Yong H. Lee[†*]

[†]Division of Electrical Engineering, Korea Advanced Institute of Science and Technology
373-1, Guseong-dong, Yuseong-gu, Daejeon, 305-701, KOREA, e-mail: yohlee@ee.kaist.ac.kr

[‡]Wireless LAN Modem Research Team, Electronics and Telecommunications Research Institute
161, Gajeong-dong, Yuseong-gu, Daejeon, 305-350, KOREA

Abstract— **In an attempt to reduce the computational complexity of the vertical Bell Labs layered space time (V-BLAST) processing for time-varying channels, an efficient adaptive receiver is developed based on the generalized decision feedback equalizer (GDFE) architecture. This receiver updates the filter weight vectors using a recursive least squares (RLS)-based time- and order-update algorithm. In time-varying channels, the proposed adaptive technique can be considerably simpler to implement than the V-BLAST processor with channel tracking, yet its performance is almost comparable to that of the latter.**

I. Introduction

Although the vertical Bell Labs layered space time (V-BLAST) architecture can provide high spectral efficiencies when multiple antennas are used at both the transmitter and the receiver [1]–[2], its application to systems in time-varying channels is difficult because of excessive computational load. Suppose that there are M transmitter and N receiver antennas. If we let $\mathbf{d}(n)$ and $\mathbf{y}(n)$ denote transmitted and received vectors at time n, respectively, then

$$\mathbf{y}(n) = \mathbf{H}(n)\mathbf{d}(n) + \mathbf{u}(n) \qquad (1)$$

where $\mathbf{H}(n)$ is an $N \times M$ channel matrix whose elements represent independent flat fading channels and $\mathbf{u}(n)$ denotes noise. To detect $\mathbf{d}(n)$ from $\mathbf{y}(n)$, the V-BLAST algorithm performs iterative nulling and cancellation: the optimal ordering of detection (or nulling and cancellation) is determined and nulling vectors are computed at each iteration. When the channel matrix $\mathbf{H}(n)$ is time-invariant, the detection ordering and the nulling vectors are fixed for all n. However, for time-varying channels the detection ordering and the nulling vectors need to be updated at each time. In addition, channel parameters should be tracked. These update and tracking operations in time domain require excessive computations. To overcome this difficulty, a simplified policy for update and tracking is proposed in [3]: it updates the V-BLAST detection blockwise and performs interpolation-based channel tracking. This technique exhibits a tradeoff between complexity and performance, and its performance is degraded rapidly as the channel fading rate increases.

In this paper, we develop an alternative technique for detection of multi-input multi-output (MIMO) systems in time-varying channels. The proposed scheme is a data-aided adaptive MIMO decision feedback equalizer (MIMO-DFE) which is based on the generalized DFE (GDFE) architecture [4][1]. At each time, it updates the tap weight vectors using the RLS-based time- and order-update algorithm and determines detection ordering depending on a least squares error criterion. The proposed algorithm does not require explicit channel tracking. As a result, its implementation can be simpler than the V-BLAST detection.

The organization of this paper is as follows: The proposed adaptive MIMO-DFE is derived in Section II and its complexity is compared with the V-BLAST detection in Section III. In Section IV, the convergence of the proposed algorithm is demonstrated in a stationary environment. Finally, Section V compares the BER performance between the proposed MIMO-DFE and the V-BLAST processor.

II. Proposed Adaptive MIMO-DFE

Fig. 1 shows the signal model and the structure of the adaptive MIMO-DFE. M symbols in $\mathbf{d}(n)$ are simultaneously transmitted through M antennas, and received by N antennas to yield the N-dimensional vector $\mathbf{y}(n)$ (see (1)). The vector $\mathbf{y}(n)$ is passed through the feedforward filter that corresponds to the nulling vectors of the V-BLAST detection [4], and then the signals which are already detected are cancelled via decision feedback filtering. The filter tap weight vectors and the order of detection are updated at each time. These updates will be referred to as *time*-update operations. Fig. 2 depicts further details of the adaptive MIMO-DFE. Here $\{\mathbf{w}_{f,i}(n)\}$ are N-dimensional feedforward weight vectors and $\{\mathbf{w}_{b,i}(n)\}$ denote $(i-1)$-dimensional feedback weight vectors. The MIMO-DFE successively detects the M transmitted symbols in $\mathbf{d}(n)$: the detection is started with linear equalization and the order of decision feedback increases with the number of iterations i. The order for detecting these M symbols is determined depending on the sum of squared errors. Let $\mathbf{d}(n) = [d_1(n), d_2(n), \cdots, d_M(n)]^T$; $\hat{d}_{k_i}(n)$ and $\tilde{d}_{k_i}(n)$ denote the i-th detected symbol and the corresponding equalizer output, respectively, where $k_i \in \{1, 2, \cdots, M\}$. Suppose, for the time being, that the order of detection $[k_1, k_2, \cdots, k_M]$ is given. The output of the equalizers can be represented as

$$\tilde{d}_{k_i}(n) = \mathbf{w}_{f,i}^H(n-1)\mathbf{y}(n) + \mathbf{w}_{b,i}^H(n-1)\hat{\mathbf{d}}_{k_{i-1}}(n) \qquad (2)$$

*Corresponding author.

[1]A blind adaptive MIMO-DFE is proposed in [5].

0-7803-7757-5/03/$17.00 ©2003 IEEE.

where $\hat{\mathbf{d}}_{k_{i-1}}(n) = [\hat{d}_{k_1}(n), \hat{d}_{k_2}(n), \cdots, \hat{d}_{k_{i-1}}(n)]^T$, and $\mathbf{w}_{b,i}^H(n) = 0$ when $i = 1$. For notational convenience, define

$$\mathbf{w}_{t,i}(n) = \begin{cases} \mathbf{w}_{f,i}(n), & i = 1 \\ [\mathbf{w}_{f,i}^T(n), \mathbf{w}_{b,i}^T(n)]^T, & i = 2, \cdots, M. \end{cases} \quad (3)$$

and

$$\mathbf{y}_{t,i}(n) = \begin{cases} \mathbf{y}(n), & i = 1 \\ [\mathbf{y}^T(n), \hat{\mathbf{d}}_{k_{i-1}}^T(n)]^T, & i = 2, \cdots, M. \end{cases} \quad (4)$$

Then, (2) is rewritten as

$$\tilde{d}_{k_i}(n) = \mathbf{w}_{t,i}^H(n-1)\mathbf{y}_{t,i}(n). \quad (5)$$

The weight vectors $\{\mathbf{w}_{t,i}(n)\}$ are determined by minimizing the following cost function:

$$J_i(n) = \sum_{l=1}^{n} \lambda^{n-l} |\hat{d}_{k_i}(l) - \mathbf{w}_{t,i}^H(l)\mathbf{y}_{t,i}(l)|^2 \quad (6)$$

where λ is the forgetting factor which satisfies $0 < \lambda \leq 1$. The optimal tap weight minimizing $J_i(n)$ is given by

$$\mathbf{w}_{t,i}(n) = \mathbf{\Phi}_i^{-1}(n)\mathbf{z}_{i,k_i}(n) \quad (7)$$

where $\mathbf{\Phi}_i(n)$ is the time-averaged correlation matrix, defined as

$$\mathbf{\Phi}_i(n) = \sum_{l=1}^{n} \lambda^{n-l} \mathbf{y}_{t,i}(l)\mathbf{y}_{t,i}^H(l) \quad (8)$$

and $\mathbf{z}_{i,k_i}(n)$ is the cross correlation vector, defined as

$$\mathbf{z}_{i,j}(n) = \sum_{l=1}^{n} \lambda^{n-l} \mathbf{y}_{t,i}(l)\hat{d}_j^*(l). \quad (9)$$

The optimal weight in (7) can be calculated recursively using the RLS algorithm. Given $\mathbf{\Phi}_i^{-1}(n)$, the RLS recursion is summarized as follows:

$$\mathbf{q}_i(n) = \mathbf{\Phi}_i^{-1}(n-1)\mathbf{y}_{t,i}(n) \quad (10)$$

$$\mathbf{k}_i(n) = \frac{\lambda^{-1}\mathbf{q}_i(n)}{1 + \lambda^{-1}\mathbf{y}_{t,i}^H(n)\mathbf{q}_i(n)} \quad (11)$$

$$\mathbf{\Phi}_i^{-1}(n) = \lambda^{-1}\mathbf{\Phi}_i^{-1}(n-1) - \lambda^{-1}\mathbf{k}_i(n)\mathbf{q}_i^H(n) \quad (12)$$

$$\mathbf{w}_{t,i}(n) = \mathbf{w}_{t,i}(n-1) + \mathbf{k}_i(n)\xi_i^*(n) \quad (13)$$

where $\mathbf{k}_i(n)$ is the gain vector and $\xi_i(n)$ is a priori estimation error defined by

$$\xi_i(n) = \hat{d}_{k_i}(n) - \mathbf{w}_{t,i}^H(n-1)\mathbf{y}_{t,i}(n). \quad (14)$$

For each n, $\{\mathbf{w}_{t,i}(n), i = 1, 2, \cdots, M\}$ can be evaluated by the time-update equations in (10)–(13). In this case, the weight $\mathbf{w}_{t,i}(n)$ of the i-th DFE is updated irrespective of the other DFEs. An alternative way of updating the weights is perform an *order*-update which utilizes the relation between the correlation matrix of the i-th DFE, $\mathbf{\Phi}_i(n)$, and that of the $(i+1)$-th DFE, $\mathbf{\Phi}_{i+1}(n)$. From (8), $\mathbf{\Phi}_{i+1}(n)$ is expressed as

$$\mathbf{\Phi}_{i+1}(n) = \begin{bmatrix} \mathbf{\Phi}_i(n) & \mathbf{z}_{i,k_i}(n) \\ \mathbf{z}_{i,k_i}^H(n) & \alpha_{k_i}(n) \end{bmatrix} \quad (15)$$

where

$$\alpha_j(n) = \sum_{l=1}^{n} \lambda^{n-l} |\hat{d}_j(l)|^2. \quad (16)$$

Using the matrix inversion lemma, $\mathbf{\Phi}_{i+1}^{-1}(n)$ is written as

$$\mathbf{\Phi}_{i+1}^{-1}(n) = \begin{bmatrix} \mathbf{\Phi}_i^{-1}(n) + c_i(n)\mathbf{w}_{t,i}(n)\mathbf{w}_{t,i}^H(n) & -c_i(n)\mathbf{w}_{t,i}(n) \\ -c_i(n)\mathbf{w}_{t,i}^H(n) & c_i(n) \end{bmatrix} \quad (17)$$

where

$$c_i(n) = \frac{1}{\alpha_{k_i}(n) - \mathbf{z}_{i,k_i}^H(n)\mathbf{w}_{t,i}(n)}. \quad (18)$$

The relation between $\mathbf{y}_{t,i}(n)$ in (4) and $\mathbf{y}_{t,i+1}(n)$ is expressed as

$$\mathbf{y}_{t,i+1}(n) = [\mathbf{y}_{t,i}^T(n), \hat{d}_{k_i}(n)]^T. \quad (19)$$

Using (17) and (19) in (10), the following recursive equation for $\mathbf{q}_i(n)$ is obtained:

$$\mathbf{q}_{i+1}(n) = \begin{bmatrix} \mathbf{q}_i(n) \\ 0 \end{bmatrix} + c_i(n-1)\xi_i(n) \begin{bmatrix} -\mathbf{w}_{t,i}(n-1) \\ 1 \end{bmatrix}. \quad (20)$$

Note that this order-update equation is considerably simpler to implement than the time-update equation in (10). Furthermore, the order-update does not require the update of $\mathbf{\Phi}_i^{-1}(n)$ in (12). Given $\mathbf{q}_1(n) = \mathbf{\Phi}_1^{-1}(n-1)\mathbf{y}(n)$, $\{\mathbf{w}_{t,i}(n), i = 1, 2, \cdots, M\}$ can be efficiently calculated using (20), (11), and (13).

It should be pointed out that the order-update equations in (17) and (20) have been derived under the assumption that the detection ordering $[k_1, k_2, \cdots, k_M]$ is known. In practice, the detection ordering is unknown and should be updated at each time n. Next, we incorporate the process for determining the detection order with the order-update for calculating the weight.

To determine the detection order k_i for the i-th DFE, the least squares error (LSE) criterion in (6) is evaluated for all candidate symbols. Then the symbol associated with the minimum LSE is selected. To be specific, define

$$\mathcal{E}_{i,j}(n) = \sum_{l=1}^{n} \lambda^{n-l} |d_j(l) - \mathbf{w}_{t,i}^H(l)\mathbf{y}_{t,i}(l)|^2 \quad (21)$$

where $j \in S_i = \{1, 2, \cdots, M\} - \{k_1, \cdots, k_{i-1}\}$ (S_i is a set encompassing all indices of the symbols to be detected from the i-th iteration). The detection order k_i is given by

$$k_i = arg \min_j \mathcal{E}_{i,j}(n). \quad (22)$$

After some calculation, (21) is rewritten as

$$\mathcal{E}_{i,j}(n) = \alpha_j(n) - \mathbf{v}_{i,j}^H(n)\mathbf{z}_{i,j}(n) \tag{23}$$

where $\alpha_j(n)$ and $\mathbf{z}_{i,j}(n)$ are defined in (16) and (9), respectively, and

$$\mathbf{v}_{i,j}(n) = \mathbf{\Phi}_i^{-1}(n)\mathbf{z}_{i,j}(n). \tag{24}$$

Note that $\mathbf{v}_{i,j}(n)$ in (24) becomes the same as $\mathbf{w}_{t,i}(n)$ in (7) if j is replaced with k_i. This means that $\{\mathbf{v}_{i,j}(n), j \in S_i\}$ is a set of candidate weight vectors which includes the true weight $\mathbf{w}_{t,i}(n)$. The candidate weights $\{\mathbf{v}_{i,j}(n)\}$ can also be obtained via an order-update process. Consider the definition of $\{\mathbf{v}_{i+1,j}(n)\}$:

$$\mathbf{v}_{i+1,j}(n) = \mathbf{\Phi}_{i+1}^{-1}(n)\mathbf{z}_{i+1,j}(n). \tag{25}$$

The order-update of $\mathbf{\Phi}_{i+1}^{-1}(n)$ is shown in (17), and the remaining problem is to find the update process for $\mathbf{z}_{i+1,j}(n)$. In this case, $\mathbf{z}_{i+1,j}(n)$ can be obtained from $\mathbf{z}_{i,j}(n)$ because at the i-th iteration, $\mathbf{z}_{i,j}(n)$ is evaluated for all $j \in S_i$. On the other hand, when the detection ordering $[k_1, k_2, \cdots, k_M]$ is given as in the previous case for deriving (20), only $\mathbf{z}_{i,k_i}(n)$ is calculated at the i-th iteration – this prevents $\mathbf{z}_{i,k_i}(n)$ from direct order-update; instead, $\mathbf{q}_i(n)$ in (10) is updated. To proceed, the elements of the $(N+i-1)$-dimensional vector $\mathbf{z}_{i,j}(n)$ are denoted as

$$\mathbf{z}_{i,j}(n) = [g_{1,j}(n), \cdots, g_{N,j}(n), g_{N+k_1,j}(n), \cdots, g_{N+k_{i-1},j}(n)]^T \tag{26}$$

where

$$g_{i,j}(n) = \begin{cases} \sum_{l=1}^{n} \lambda^{n-l} y_i(n) d_j^*(n), & 1 \le i \le N \\ \sum_{l=1}^{n} \lambda^{n-l} d_{i-N}(n) d_j^*(n), & N+1 \le i \le N+M \end{cases} \tag{27}$$

Using the new notations, $\mathbf{z}_{i+1,j}(n)$ can be expressed as

$$\mathbf{z}_{i+1,j}(n) = [\mathbf{z}_{i,j}^T(n), g_{N+k_i,j}(n)]^T. \tag{28}$$

In addition, it is straightforward to see that $\alpha_j(n)$ in (16) is equal to $g_{N+j,j}(n)$. Thus, the LSE in (23) can be rewritten as

$$\mathcal{E}_{i,j}(n) = g_{N+j,j}(n) - \mathbf{v}_{i,j}^H(n)\mathbf{z}_{i,j}(n). \tag{29}$$

By extending the definition of $\mathbf{z}_{i,j}(n)$ in (9), we define a $(N+M)$-by-M cross-correlation matrix

$$\mathbf{G}(n) = \sum_{l=1}^{n} \lambda^{n-l} [\mathbf{y}^T(n), \hat{\mathbf{d}}^T(n)]^T \hat{\mathbf{d}}^H(n). \tag{30}$$

Then, it can be shown through some calculation that $g_{i,j}(n)$ in (27) is the (i,j)-th entry of $\mathbf{G}(n)$. Therefore, the order-update of $\mathbf{z}_{i,j}(n)$ in (28) can be achieved once $\mathbf{G}(n)$ in (30) is known. Evaluation of $\mathbf{G}(n)$ requires the detected symbol vector $\hat{\mathbf{d}}(n)$. To obtain $\hat{\mathbf{d}}(n)$ prior to updating $\mathbf{G}(n)$, we use the tap weight vectors and detection ordering of the previous time $(n-1)$ for detecting the current symbols in $\mathbf{d}(n)$. Specifically, for the detection order $[k_1, k_2, \cdots, k_M]$ obtained at time $(n-1)$,

$$\tilde{d}_{k_i}(n) = \mathbf{w}_{t,i}^H(n-1)\mathbf{y}_{t,i}(n) \tag{31}$$

and

$$\hat{d}_{k_i}(n) = \text{decision}\{\tilde{d}_{k_i}(n)\}. \tag{32}$$

After evaluating $\hat{d}_{k_i}(n)$ for all i, $\hat{\mathbf{d}}(n) = [\hat{d}_1(n), \hat{d}_2(n), \cdots, \hat{d}_M(n)]^T$ can be obtained by rearranging $\{\hat{d}_{k_i}(n)\}$, and $\mathbf{G}(n)$ can be calculated. Given $\mathbf{G}(n)$, use of (17) and (28) in (25) results in

$$\mathbf{v}_{i+1,j}(n) = \mathbf{\Phi}_{i+1}^{-1}(n)\mathbf{z}_{i+1,j}(n)$$

$$= \begin{bmatrix} \mathbf{v}_{i,j}(n) \\ 0 \end{bmatrix} + \frac{g_{N+k_i,j}(n) - \mathbf{w}_{t,i}^H(n)\mathbf{z}_{i,j}(n)}{\mathcal{E}_{i,k_i}(n)} \begin{bmatrix} -\mathbf{w}_{t,i}(n) \\ 1 \end{bmatrix} \tag{33}$$

where $j \in S_{i+1} = S_i - \{k_i\}$. The proposed adaptation procedure is summarized below and will be referred to as Algorithm 1.

Algorithm 1

Step 1. Initialization:

$$\begin{aligned} n &= 1 \\ k_i &= i, \text{ for all } i \\ \mathbf{\Phi}_1^{-1}(0) &= \delta^{-1}\mathbf{I}, \ \mathbf{G}(0) = \mathbf{0} \\ \mathbf{w}_{f,i}(0) &= \mathbf{1}, \ \mathbf{w}_{b,i}(0) = \mathbf{0}, \ \mathbf{v}_{1,i}(0) = \mathbf{0}, \text{ for all } i \end{aligned}$$

where δ is a small positive constant.

Step 2. Iterative equalization and decision: for $i = 1, 2, \cdots, M$, calculate $\{\hat{d}_{k_i}(n)\}$ using (31) and (32).

Step 3. Updating the cross-correlation matrix $\mathbf{G}(n)$ in (30).

Step 4. Updating the weight vectors and detection ordering:

(i) Computation of $\{\mathbf{v}_{1,j}(n), j = 1, 2, \cdots, M\}$ by the time-update equations:

$$\mathbf{q}_1(n) = \mathbf{\Phi}_1^{-1}(n-1)\mathbf{y}(n) \tag{34}$$

$$\mathbf{k}_1(n) = \frac{\lambda^{-1}\mathbf{q}_1(n)}{1 + \lambda^{-1}\mathbf{y}^H(n)\mathbf{q}_1(n)} \tag{35}$$

$$\mathbf{\Phi}_1^{-1}(n) = \lambda^{-1}\mathbf{\Phi}_1^{-1}(n-1) - \lambda^{-1}\mathbf{k}_1(n)\mathbf{q}_1^H(n) \tag{36}$$

$$\mathbf{v}_{1,j}(n) = \mathbf{v}_{1,j}(n-1) + \mathbf{k}_1(n)(d_j(n) - \mathbf{v}_{1,j}^H(n-1)\mathbf{y}(n))^* \tag{37}$$

(ii) For $i = 1, 2, \cdots, M-1$:

- Obtain $\mathbf{z}_{i,j}(n)$ in (26).
- Evaluate the LSEs $\mathcal{E}_{i,j}(n)$ and k_i in (29) and (22), respectively.
- $\mathbf{w}_{t,i}(n) = \mathbf{v}_{i,k_i}(n)$.
- Evaluate $\mathbf{v}_{i+1,j}(n)$ using (33).

Step 5. Recursion: $n = n + 1$ and go to Step 2.

During a training period in which the true symbol vector $\mathbf{d}(n)$ is known, Step 2 is unnecessary. When the channel is slowly varying, it may not be necessary to update detection ordering at each time. In fact, our simulation results, for slowly varying channels, showed that neighboring detection orderings from Algorithm 1 are often identical. The fact that detection orderings for $\mathbf{d}(n-1)$ and $\mathbf{d}(n)$ are the same indicates that the detection ordering is known when calculating $\mathbf{w}_{t,i}(n)$; and thus the weight $\mathbf{w}_{t,i}(n)$ can be efficiently updated using (20), (11), and (13), instead of Step 5 of Algorithm 1. These observations lead to the following adaptation procedure, called Algorithm 2, that sparsely updates the detection ordering.

Algorithm 2

This algorithm is identical to Algorithm 1 with the exception that Step 4-(ii) in Algorithm 1 is replaced with the following:

(ii) If n is a multiple of γ which is a positive integer, then follow Step 4-(ii) of Algorithm 1. Otherwise, for $i = 1, 2, \cdots, M-1$, evaluate $\mathbf{w}_{t,i}(n)$ using (20), (11), and (13).

Algorithm 2 updates the detection ordering at every γ symbol period, and it becomes Algorithm 1 when $\gamma = 1$. Steps 3 and 4-(i) of Algorithm 1 are included in Algorithm 2 for updating the detection order.

III. Complexity Comparison

This section compares complexities for implementing the proposed algorithms and the V-BLAST method with RLS tracking.

One of the most efficient algorithm for the V-BLAST is the one in [7] that performs QR decomposition, inversion of an upper triangular matrix, reordering, and triangularization of reordered matrices. When $M = N$, these operations need $\frac{7}{3}M^3$ complex multiplications and $\frac{5}{3}M^3$ complex additions. The RLS channel tracking requires $O(M^2)$ operations. Table I(a) compares the complexities of Algorithm 1 and the corresponding V-BLAST based method (the detection order is updated at each time). Both the methods require $O(M^3)$ operations but the proposed algorithm provides some savings in computation: for example, when $M = N = 8(4)$, the number of complex multiplications is reduced by 25.3%(14.0%). In the case of Algorithm 2 where the detection order is updated at every γ symbols, additional savings can be made. Table I(b) lists the number of operations for this case. Note that both the methods become simpler to implement; however, in this case, the proposed Algorithm 2 needs $O(M^2)$ operations when $\gamma \geq M$, while the V-BLAST still requires $O(M^3)$ operations. Therefore, the computational savings of Algorithm 2 over the V-BLAST can be more significant. For example, when $M = N = 8(4)$ and $\gamma = 8$, the number of complex multiplications is reduced by 36.3%(14.8%).

TABLE I.

Required number of operations per symbol duration.

	Complex Multiplication	Complex Addition
V-BLAST Detector	$\frac{7}{3}M^3$ $+5M^2$	$\frac{5}{3}M^3$ $+4M^2$
Proposed Algorithm 1	$\frac{4}{3}M^3$ $+7M^2$	$\frac{4}{3}M^3$ $+5M^2$

(a) Detection orderings are updated at each time.

	Complex Multiplication	Complex Addition
V-BLAST Detector	$\frac{1}{\gamma}(\frac{7}{3}M^3 + 5M^2)+$ $\frac{\gamma-1}{\gamma}(\frac{5}{6}M^3 + \frac{11}{2}M^2)$	$\frac{1}{\gamma}(\frac{5}{3}M^3 + 4M^2)+$ $\frac{\gamma-1}{\gamma}(\frac{5}{6}M^3 + 4M^2)$
Proposed Algorithm 2	$\frac{1}{\gamma}(\frac{4}{3}M^3 + 7M^2)$ $+\frac{\gamma-1}{\gamma}(\frac{15}{2}M^2)$	$\frac{1}{\gamma}(\frac{4}{3}M^3 + 5M^2)$ $+\frac{\gamma-1}{\gamma}(5M^2)$

(b) Detection orderings are updated at every γ symbols.

IV. Convergence Analysis

Our aim is to demonstrate the convergence of Algorithm 1 when the channel is time-invariant. However, directly proving the convergence is formidable, because of the detection order that is updated at each time. Therefore, we take an indirect approach based on the observations stated below:

- For time-invariant channels, the optimal detection order is fixed for all n.
- Assuming that the detection order of Algorithm 2 converges to the optimal order, Algorithm 1 approaches the RLS algorithm in (10)–(13).

In what follows, we first analyze the convergence of the RLS algorithm in (10)–(13), adopting the analysis in [6]. Then the convergence of Algorithm 1 is indirectly demonstrated by comparing the analytical results with the corresponding simulation results for Algorithm 1.

A. Convergence of the RLS Algorithm

To proceed with the analysis, the estimation error, $e_{o,i}(n)$ of the i-th ideal equalizer, is defined by

$$e_{o,i}(n) = d_{k_i}(n) - \mathbf{w}_{o,i}^H \mathbf{y}_{t,i}(n) \tag{38}$$

where $\mathbf{w}_{o,i}$ is the optimal weight vector which is constant in a stationary environment. It is assumed that $e_{o,i}(n)$ is white with zero mean. Its variance σ_i^2 is written as

$$\begin{aligned} \sigma_i^2 &= E[|d_{k_i}(n) - \mathbf{w}_{o,i}^H \mathbf{y}_{t,i}(n)|^2] \\ &= E[|d_{k_i}(n)|^2](1 - [\mathbf{h}_{k_i}^H, \mathbf{0}_{(i-1)}^H]\mathbf{w}_{o,i}) \end{aligned} \tag{39}$$

where $\mathbf{h}_j$ is the j-th column of a fixed channel matrix $\mathbf{H}$ and $\mathbf{0}_j$ represents the j-dimensional null vector. Adopting the analysis in [6], it can be shown that: For $\lambda = 1$,

- $E[\epsilon_i^H(n)\epsilon_i(n)] = \dfrac{\sigma_i^2}{n-(M+i)}\mathrm{tr}[\mathbf{R}_i^{-1}], n \geq M+i-1$ (40)

where $\epsilon_i(n)$ is the weight error vector given by $\epsilon_i(n) = \mathbf{w}_{t,i}(n) - \mathbf{w}_{o,i}$ and

$$\mathbf{R}_i = E[\mathbf{y}_{t,i}(n)\mathbf{y}_{t,i}^H(n)] = \begin{bmatrix} \mathbf{H}\mathbf{H}^H + \sigma^2\mathbf{I}_N & \mathbf{H}_i \\ \mathbf{H}_i^H & \mathbf{I}_{i-1} \end{bmatrix}. \quad (41)$$

Here $\mathbf{H}_i = [\mathbf{h}_{k_1}, \mathbf{h}_{k_2}, \cdots, \mathbf{h}_{k_{i-1}}]$, $\mathbf{h}_i$ is defined in (39), and $\mathbf{I}_j$ is the $(j \times j)$ identity matrix.

- $E[|\xi_i(n)|^2] = \sigma_i^2 + \frac{M\sigma_i^2}{n-(M+i+1)}, n \geq M+i+2 \quad (42)$

where $\xi_i(n)$ is a priori estimation error defined as $\xi_i(n) = d_{k_i}(n) - \mathbf{w}_{t,i}^H(n-1)\mathbf{y}_{t,i}(n)$.

The mean-square value of $\xi_i(n)$ yields a learning curve for the RLS algorithm. Next, the theoretical results in (40) and (42) are compared with the corresponding simulation results for Algorithm 1.

B. Comparison between RLS Theory and Behavior of Algorithm 1

The received signal vector $\mathbf{y}(n)$ was generated under the following assumptions: the elements of the channel matrix $\mathbf{H}$ were i.i.d. complex Gaussian with unit power and zero mean; $\mathbf{H}$ was fixed during a burst; $\mathbf{d}(n)$ was composed of Hadamard sequences of length 128; and $\mathbf{u}(n)$ was i.i.d. complex Gaussian noise with zero mean. The number of antennas was set at $M = 4$ and $N = 4$; the forgetting factor $\lambda = 1$; and $E_b/N_o = 15$ dB. In the simulation, the empirical MSE values were obtained through 200 independent trials of Algorithm 1. The optimal detection ordering for $\mathbf{H}$ was obtained and used for evaluating the theoretical values in (40) and (42).

Fig. 3 compares the theoretical MSE values for RLS weight vectors in (40) with the experimental MSEs in the weights of Algorithm 1. The figure shows a good agreement between the RLS theory and the experiment with Algorithm 1, when $n > 20$. The learning curve corresponding to $E[|\xi_2(n)|^2]$ is shown in Fig. 4. The theoretical MSE in (42) reached the steady-state after about 14 iterations $(=2\times(M+N-1))$, and was close to that of the experimental MSE of Algorithm 1. The results in Figs. 3 and 4 indicate that Algorithm 1 acted like the bank of RLS algorithms after about 20 iterations and that the detection orders of Algorithm 1 converges to the optimal detection order.

V. Simulation Results

In this section, we compare the BER performance between the V-BLAST detectors and the proposed adaptive MIMO-DFE in time-varying channels. In the simulation, the following parameters were assumed: QPSK was used; the channels were independent Rayleigh flat fading; and $\mathbf{u}(n)$ was i.i.d. complex Gaussian noise with zero mean. The vectors $\{\mathbf{d}(n)\}$ were grouped into frames consisting of 160 vectors, where the first 32 vectors were training vectors. The number of antennas was set at $M = 4$ and $N = 4$. The MMSE detection was used for V-BLAST receivers and the sub-block size was 4 for the receiver in [3]. For the MMSE detection, the channel was estimated through the RLS algorithm and the noise variance was obtained by

$$\hat{\sigma}^2(n) = \frac{1}{N}\sum_{k=1}^{n}\lambda^{n-k}(1-\lambda) \cdot\{\mathbf{y}(k) - \hat{\mathbf{H}}(k)\hat{\mathbf{d}}(k)\}^H\{\mathbf{y}(k) - \hat{\mathbf{H}}(k)\hat{\mathbf{d}}(k)\}. \quad (43)$$

Fig. 5 compares the bit error rate (BER) performances of the proposed and existing techniques when f_dT(normalized Doppler frequency) $= 5 \cdot 10^{-4}$, and $\lambda = 0.95$ which was the optimal value for the f_dT value. The proposed receiver performed comparable to the V-BLAST processor with channel tracking, while outperforming the scheme in [3] and the V-BLAST receiver without channel tracking. The BER values of the V-BLAST with known channels, which is impractical, provided a performance bound. Fig. 6 shows the performance of the proposed algorithm when the detection ordering is periodically updated. When the period of detection order-update γ was less than 24 symbol time, the performance loss by the sparse ordering was less than 1 dB at BER$=10^{-4}$.

VI. Conclusion

Adaptive algorithms for the MIMO-DFE were proposed for receivers in time-varying channels. The proposed receivers update the weight vectors through time- and order-update operations, and adaptively determine the detection ordering depending on a LSE criterion. In time-varying channels, the proposed techniques ware simpler to implement than the V-BLAST processor with RLS channel tracking, yet performed comparable to the latter. Extending the proposed algorithms to frequency selective fading channels remains to be investigated.

References

[1] P. W. Wolniansky, G. J. Foschini, G. D. Golden, and R. A. Valenzuela, "V-BLAST: An architecture for realizing very high data rates over the rich-scattering wireless channel," *Proc. IEEE ISSSE98*, pp. 295–299, Pisa, Italy, Sept. 1998.

[2] G. J. Foschini, G. D. Golden, R. A. Valenzuela, and P. W. Wolniansky, "Simplified processing for high spectral efficiency wireless communication employing multi-element arrays," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 1841–1851, Nov. 1999.

[3] Q. Sun and D. C. Cox, "Training-based channel estimation for continuous flat fading BLAST," *Proc. IEEE ICC2002*, pp. 325–329, Helsinki, Finland, June 2002.

[4] G. Ginis and J. M. Cioffi, "On the relation between V-BLAST and the GDFE," *IEEE Commun. Letters*, vol. 5, pp. 364–366, Sept. 2001.

[5] G. Ginis and Y. Bar-Ness, and John M. Cioffi, "Blind adaptive MIMO decision feedback equalization using Givens rotations," *Proc. IEEE ICC2002*, pp. 59–63, Helsinki, Finland, June 2002.

[6] S. Haykin, *Adaptive filter theory*, Prentice-Hall, 1996, 3rd edition.

[7] W. Zha and S. D. Blostein, "Modified decorrelating decision-feedback detection of BLAST space-time system," *Proc. IEEE ICC2002*, pp. 335–339, Helsinki, Finland, June 2002.

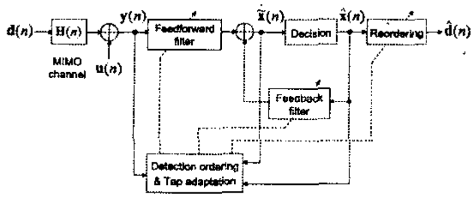

Fig. 1. MIMO-DFE structure.

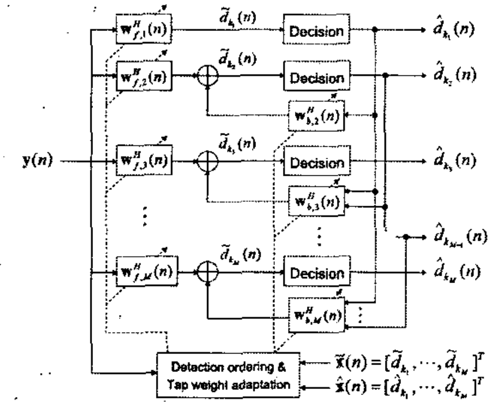

Fig. 2. Detailed architecture of MIMO-DFE.

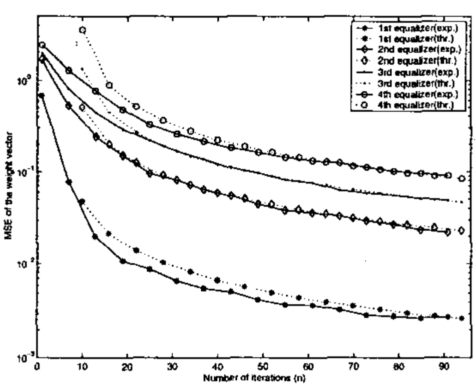

Fig. 3. MSE of tap weight vectors when $M = 4$, $N = 4$, $E_b/N_o = 15$ dB.

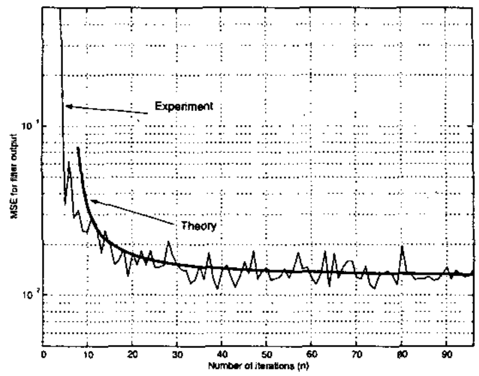

Fig. 4. Learning curves for the 2nd equalizer of Algorithm 1 when $M = 4$, $N = 4$, $E_b/N_o = 15$ dB.

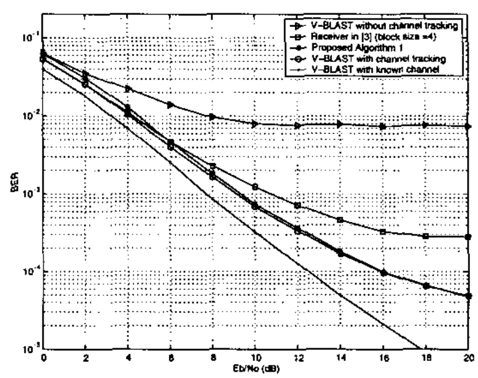

Fig. 5. Comparison of BER performances when $M = 4$, $N = 4$, and $f_dT = 0.0005$.

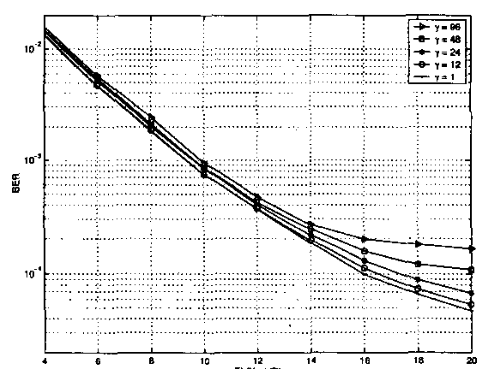

Fig. 6. BER performances of Algorithm 1 and 2 when $M = 4$, $N = 4$, and $f_dT = 0.0005$ (Algorithm 2 becomes Algorithm 1 when $\gamma = 1$).