A Fast Recursive Algorithm for Optimum Sequential Signal Detection in a BLAST System

Jacob Benesty, *Member, IEEE,* Yiteng (Arden) Huang, *Member, IEEE*, and Jingdong Chen, *Member, IEEE*

***Abstract*—Bell Laboratories layered space-time (BLAST) wireless systems are multiple-antenna communication schemes that can achieve very high spectral efficiencies in scattering environments with no increase in bandwidth or transmitted power. The most popular and, by far, the most practical architecture is the so-called vertical BLAST (V-BLAST). The signal detection algorithm of a V-BLAST system is computationally very intensive. If the number of transmitters is M and is equal to the number of receivers, this complexity is proportional to M^4 at each sample time. In this paper, we propose a very simple and efficient algorithm that reduces the complexity by a factor of M.**

***Index Terms*—Antenna array processing, Bell Laboratories layered space-time (BLAST) architecture, multiple-input–multiple-output (MIMO) systems.**

I. Introduction

Telatar [1] and Foschini [2] showed that the multipath wireless channel is capable of huge capacities, provided that the multipath scattering is sufficiently rich and is properly exploited through the use of an appropriate processing architecture and multiple antennas (both at transmission and reception). The original architecture proposed in [2], which is called the diagonal Bell Laboratories layered space-time (D-BLAST), is theoretically capable of approaching the Shannon capacity for multiple transmitters and receivers, but it is very complex to implement. A simplified version known as vertical BLAST (V-BLAST) was proposed in [3] and [5] and can still achieve a substantial portion of that capacity. For example, the authors in [3] have demonstrated, using a laboratory prototype and in an indoor environment, spectral efficiencies of 20–40 b/s/Hz at average signal-to-noise ratios (SNRs) ranging from 24 to 34 dB. In the rest, we will focus on signal detection algorithms in the V-BLAST systems.

In a V-BLAST system, a data stream is split into M uncorrelated substreams, each of which is transmitted by one of the M transmitting antennas. The M substreams are picked up by N receiving antennas after being perturbed by a channel matrix $\mathbf{H}$. The substream signal with the highest SNR is detected first, and this involves the calculation of the pseudo-inverse of $\mathbf{H}$ using the zero-forcing algorithm or the calculation of a minimum mean-square error filter. The effect of the detected symbol as well as the effect of the corresponding channel is subtracted from the N received antennas. This process repeats with the next strongest substream signal among the remaining undetected signals. Thus, this algorithm detects the M symbols in M iterations, and it is proven in [3] that this decoding order is optimal from a performance point of view. However, as will be shown later, the complexity required to achieve this performance is very high, which makes it difficult to be implemented in real-time systems. Hassibi proposed a square root method for V-BLAST signal detection, which reduces the computational complexity by an order of magnitude when the number of antennas is large [4]. Thus far, however, the quite small number of antennas (e.g., four or eight) is more interesting in practice, and a real-time implementation cannot benefit from the square-root method. In this paper, we are going to develop a fast V-BLAST algorithm that is more efficient than the existing methods for any number of transmitting/receiving antennas.

This paper is organized as follows. Section II defines the signal model and gives the channel capacity. In Section III, we explain in detail the V-BLAST algorithm. In Section IV, we show how to derive a fast algorithm for BLAST. Section V evaluates the complexity of different algorithms. Finally, we give our conclusions in Section VI.

II. Signal Model and Channel Capacity

The BLAST architecture is a multiple-input multiple-output (MIMO) channel where a single user uses a communication link comprising M transmitting antennas and N receiving antennas in a flat-fading environment (meaning that the signals are narrow-band). At the receivers, at sample time k, we have

$$\begin{aligned}\mathbf{x}(k) &= \sum_{m=1}^{M} \mathbf{h}_{:m} s_m(k) + \mathbf{w}(k)\\ &= \mathbf{H}\mathbf{s}(k) + \mathbf{w}(k), \quad k = 1, 2, \ldots, K \end{aligned} \tag{1}$$

where we have the equation at the bottom of the next page, which is the N-dimensional received vector

$$\begin{aligned}\mathbf{H} &= \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1M} \\ h_{21} & h_{22} & \cdots & h_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ h_{N1} & h_{N2} & \cdots & h_{NM} \end{bmatrix}\\ &= [\, \mathbf{h}_{:1} \quad \mathbf{h}_{:2} \quad \cdots \quad \mathbf{h}_{:M} \,]\\ &= \begin{bmatrix} \mathbf{h}_{1:}^H \\ \mathbf{h}_{2:}^H \\ \vdots \\ \mathbf{h}_{N:}^H \end{bmatrix}\end{aligned}$$

Manuscript received October 18, 2002; revised January 21, 2003. The associate editor coordinating the review of this paper and approving it for publication was Dr. Martin Haardt.

The authors are with Bell Laboratories, Lucent Technologies, Murray Hill, NJ 07974 USA (e-mail: jb@research.bell-labs.com; arden@research.bell-labs.com; jingdong@research.bell-labs.com).

Digital Object Identifier 10.1109/TSP.2003.812897

1053-587X/03$17.00 © 2003 IEEE

is an $N \times M$ complex matrix assumed to be constant for K symbol periods, vectors $\mathbf{h}_{n:}$ and $\mathbf{h}_{:m}$ are, respectively, of length M and N

$$\mathbf{s}(k) = [\, s_1(k) \quad s_2(k) \quad \cdots \quad s_M(k)\,]^T$$

is the M-dimensional transmitted vector

$$\mathbf{w}(k) = [\, w_1(k) \quad w_2(k) \quad \cdots \quad w_N(k)\,]^T$$

is a zero-mean complex additive white Gaussian noise (AWGN) vector with covariance

$$\begin{aligned}\mathbf{R}_{ww} =& E\{\mathbf{w}(k)\mathbf{w}^H(k)\}\\ =& \sigma_w^2 \mathbf{I}_{N\times N}\end{aligned} \tag{2}$$

and ^{T} and ^{H} denote, respectively, transpose and conjugate transpose of a matrix or a vector.

The transmitted vector $\mathbf{s}(k)$ has a total power P_T. This power is held constant, regardless of the number of transmitting antennas M, and corresponds to the trace of the covariance matrix of the transmitted vector

$$\begin{aligned}P_T =& \text{tr}[\mathbf{R}_{ss}] = \text{Constant}\\ =& \sum_{m=1}^{M} \sigma_{s_m}^2 .\end{aligned} \tag{3}$$

In the rest of this paper, we suppose that all the antennas transmit with the same power

$$\sigma_{s_1}^2 = \sigma_{s_2}^2 = \cdots = \sigma_{s_M}^2 = \sigma_s^2$$

so that

$$P_T = M\sigma_s^2. \tag{4}$$

We now define a parameter that relates P_T and σ_w^2 as

$$\rho = \frac{P_T}{\sigma_w^2}. \tag{5}$$

This parameter corresponds to the average receive SNR per antenna when the average power of channel coefficients is 1 as assumed in the flat-fading channel model.

An original information sequence for wireless transmission is demultiplexed into M data sequences $s_m(k), m = 1, \ldots, M$ (called substreams), and each one of them is sent through a transmitting antenna. These M substreams are assumed to be uncorrelated, which implies that the covariance matrix of the transmitted vector $\mathbf{s}(k)$ is diagonal:

$$\begin{aligned}\mathbf{R}_{ss} =& E\{\mathbf{s}(k)\mathbf{s}^H(k)\}\\ =& \sigma_s^2 \mathbf{I}_{M\times M}.\end{aligned} \tag{6}$$

We also suppose the following:

- $N \geq M$.
- $\mathbf{H}$ has full column rank, i.e., rank$[\mathbf{H}] = M$.

Here, we assume that the transmitter has no knowledge of the channel. In this case, the mutual information between the inputs and outputs of the (M, N) flat-fading channel is given by the familiar formula [1], [2]

$$\begin{aligned}C =& \log_2 \left[\det\left(\mathbf{I}_{N\times N} + \frac{\rho}{M}\mathbf{H}\mathbf{H}^H\right)\right] \quad \left[\frac{\text{bps}}{\text{Hz}}\right]\\ =& \log_2 \left[\det\left(\mathbf{I}_{M\times M} + \frac{\rho}{M}\mathbf{H}^H\mathbf{H}\right)\right].\end{aligned} \tag{7}$$

One very important observation that can be made from (7) is that, for rich scattering channels (meaning that the elements of the channel matrix are independent of one another), the MIMO channel capacity grows roughly proportionally to M [6].

III. V-BLAST Algorithm

In order to detect the transmitted symbols at the receivers, the complex channel matrix $\mathbf{H}$ needs to be known. In practice, $\mathbf{H}$ is identified by sending a training sequence (known at the reception) at the beginning of each burst [7]. The length of this burst is equal to $K = K_1 + K_2$ symbols, where the K_1 symbols are used for training, and the K_2 symbols are the data information. The propagation coefficients are assumed to be constant during a whole burst, after which they change to new independent random values, which they maintain for another K symbols, and so on. Since channel estimation is out of the scope of the present paper, in the remainder, we will make no distinction between $\mathbf{H}$ and its estimate.

The first step of the V-BLAST algorithm [3] makes use of the pseudo-inverse of the channel matrix $\mathbf{H}$ or the minimum mean-square error (MMSE) filter $\mathbf{G}$.

Define the error vector signal at time k between the input $\mathbf{s}(k)$ and its estimate

$$\begin{aligned}\mathbf{e}(k) =& \mathbf{s}(k) - \mathbf{y}(k)\\ =& \mathbf{s}(k) - \mathbf{G}^H\mathbf{x}(k).\end{aligned} \tag{8}$$

Now, let us define the error criterion

$$\begin{aligned}J =& E\{\mathbf{e}^H(k)\mathbf{e}(k)\}\\ =& \text{tr}\left[E\{\mathbf{e}(k)\mathbf{e}^H(k)\}\right].\end{aligned} \tag{9}$$

The minimization of (9) leads to the Wiener-Hopf equation

$$\mathbf{G}^H\mathbf{R}_{xx} = \mathbf{R}_{sx} \tag{10}$$

where

$$\mathbf{R}_{xx} = E\{\mathbf{x}(k)\mathbf{x}^H(k)\} \tag{11}$$

is the output signal covariance matrix, and

$$\mathbf{R}_{sx} = E\{\mathbf{s}(k)\mathbf{x}^H(k)\} \tag{12}$$

is the cross-correlation matrix between the input and output signals.

$$\begin{aligned}\mathbf{x}(k) =& [\, x_1(k) \quad x_2(k) \quad \cdots \quad x_N(k)\,]^T\\ =& [\, \mathbf{h}_{1:}^H\mathbf{s}(k) + w_1(k) \quad \mathbf{h}_{2:}^H\mathbf{s}(k) + w_2(k) \quad \cdots \quad \mathbf{h}_{N:}^H\mathbf{s}(k) + w_N(k)\,]^T\end{aligned}$$

From (10), we find that the MMSE filter is

$$\mathbf{G} = \left[\mathbf{H}\mathbf{H}^H + \alpha \mathbf{I}_{N\times N}\right]^{-1}\mathbf{H} \tag{13}$$

where

$$\alpha = \frac{\sigma_w^2}{\sigma_s^2}. \tag{14}$$

It can easily be seen that (13) is equivalent to

$$\begin{aligned}\mathbf{G} =& \mathbf{H}\left[\mathbf{H}^H\mathbf{H} + \alpha \mathbf{I}_{M\times M}\right]^{-1}\\ =& \mathbf{H}\mathbf{Q}.\end{aligned} \tag{15}$$

The second form [see (15)] is more useful and more efficient in practice since $M \leq N$ and the size of the matrix to invert in (15) is smaller or equal than the size of the matrix to invert in (13).

Instead of the MMSE filter, we can use directly the pseudo-inverse of $\mathbf{H}$, which is

$$\mathbf{G}_{\text{PI}} = \mathbf{H}\left[\mathbf{H}^H\mathbf{H}\right]^{-1}. \tag{16}$$

The only difference between the expressions $\mathbf{G}$ and $\mathbf{G}_{\text{PI}}$ is that the first one is "regularized" by a diagonal matrix $\alpha\mathbf{I}_{M\times M}$, whereas the second one is not. This regularization introduces a bias, but (15) gives a much more reliable result than (16) when the matrix $\mathbf{H}^H\mathbf{H}$ is ill-conditioned and the estimation of the channel is noisy. In practice, depending on the condition number of the matrix $\mathbf{H}^H\mathbf{H}$, we can take a different value for α than the one given in (14). For example, if this condition number is very high and the SNR is also high, it will be better to take a higher value for α. Thus, the MMSE filter can be seen as a biased pseudo-inverse of $\mathbf{H}$.

In the V-BLAST algorithm, the detection of the symbols $s_m(k)$ is done in M iterations. The order in which the components of $\mathbf{s}(k)$ are detected is important to the overall performance of the system. Let the ordered set

$$\mathcal{S} = \{p_1, p_2, \ldots, p_M\} \tag{17}$$

be a permutation of the integers $1, 2, \ldots, M$ specifying the order in which components of the transmitted symbol vector $\mathbf{s}(k)$ are extracted. The first iteration, which is also the initialization, is performed in three steps (as well as the other iterations).

Step 1) Using the MMSE filter or the pseudo-inverse, we compute

$$\mathbf{y}(k) = \mathbf{G}^H\mathbf{x}(k). \tag{18}$$

Step 2) The element of $\mathbf{y}(k)$ with the highest SNR is detected. This element is associated with the smallest diagonal entry of $\mathbf{Q}$ for the MMSE filter (as explained in the next section) or the column of $\mathbf{G}_{\text{PI}}$ having the smallest norm for the pseudo-inverse (zero-forcing) [3]. If such a column is p_1, we get

$$\hat{s}_{p_1}(k) = \mathcal{Q}\left[y_{p_1}(k)\right] \tag{19}$$

with $\mathcal{Q}[\cdot]$ indicating the slicing or quantization procedure according to the constellation in use.

Step 3) Assuming that $\hat{s}_{p_1}(k) = s_{p_1}(k)$, we cancel $s_{p_1}(k)$ from the received vector $\mathbf{x}(k)$, resulting in a modified received vector

$$\begin{aligned}\mathbf{x}_2(k) =& \mathbf{x}(k) - \hat{s}_{p_1}(k)\mathbf{h}_{:p_1}\\ =& \sum_{m\neq p_1} \mathbf{h}_{:m}s_m(k) + \mathbf{w}(k)\\ =& \mathbf{H}_{M-1}\mathbf{s}_{M-1}(k) + \mathbf{w}(k)\end{aligned} \tag{20}$$

where $\mathbf{H}_{M-1}$ is an $N \times (M-1)$ matrix derived from $\mathbf{H}$ by removing its p_1th column, and $\mathbf{s}_{M-1}(k)$ is a vector of length $M-1$ obtained from $\mathbf{s}(k)$ by removing its p_1th component.

Steps 1–3 are then performed for components $p_2, \ldots, p_M$ by operating in turn on the progression of modified received vectors $\mathbf{x}_2(k), \ldots, \mathbf{x}_M(k)$. Note that at the mth iteration, we will obtain the $N \times (M-m)$ matrix $\mathbf{H}_{M-m}$, which can be derived from $\mathbf{H}$ by removing m of its columns: $p_1, \ldots, p_m$. As shown in [3], this ordering (choosing the best SNR at each iteration in the detection process) is optimal among all possible orderings.

Since the MMSE filter is more advantageous than the zero-forcing algorithm from a performance point of view, we will focus on only the MMSE implementation of the V-BLAST in the rest of this paper. Table I summarizes the V-BLAST algorithm using the MMSE filter.

IV. Fast V-BLAST Algorithm

The arithmetic complexity of the V-BLAST algorithm is very high. The complexity of computing the inverse of an $M \times M$ matrix is approximately in the order of M^3. In addition, the matrix $\mathbf{G}$ is the product of a rectangular matrix of size $N \times M$ and a square matrix of size $M \times M$, and the complexity of such a product is proportional to NM^2 at each iteration. The algorithm requires M iterations; therefore, the overall complexity is in $\mathcal{O}(NM^3)$ for each sample time k, even if the matrices are deflated by 1 at each iteration. A more detailed complexity evaluation will be given in Section V.

Here, the matrix $\mathbf{G}$ is not computed directly. Recall that

$$\mathbf{G} = \mathbf{H}\mathbf{R}^{-1} \tag{21}$$

where

$$\mathbf{R} = \mathbf{H}^H\mathbf{H} + \alpha\mathbf{I}_{M\times M}. \tag{22}$$

The covariance matrix of the error signal $\mathbf{e}(k) = \mathbf{s}(k) - \mathbf{y}(k)$ is

$$\begin{aligned}\mathbf{R}_{ee} =& E\{\mathbf{e}(k)\mathbf{e}^H(k)\}\\ =& \sigma_w^2\mathbf{R}^{-1}\\ =& \sigma_w^2\mathbf{Q}.\end{aligned} \tag{23}$$

Clearly, the element of $\mathbf{y}(k)$ with the highest SNR is the one with the smallest error variance so that

$$p_1 = \arg\min_m q_{mm} \tag{24}$$

where q_{mm} are the diagonal elements of the matrix $\mathbf{Q} = \mathbf{R}^{-1}$.

TABLE I
V-BLAST ALGORITHM USING THE MMSE FILTER

```
Initialization:  $\mathbf{x}_1(k) = \mathbf{x}(k)$,  $\mathbf{H}_M = \mathbf{H} = \left[\, \mathbf{h}_{M,:1} \;\; \mathbf{h}_{M,:2} \;\; \cdots \;\; \mathbf{h}_{M,:M} \,\right]$
                 $\mathbf{Q}_M = \left[\, \mathbf{q}_{M,:1} \;\; \mathbf{q}_{M,:2} \;\; \cdots \;\; \mathbf{q}_{M,:M} \,\right] = \left[\mathbf{H}_M^H \mathbf{H}_M + \alpha \mathbf{I}_{M\times M}\right]^{-1}$
                 $\mathbf{f}(k) = [\, 1 \;\; 2 \;\; \cdots \;\; M \,]^T$
                 $l_1 = \arg\min_i q_{M,ii}$,   $p_1 = f_{l_1}(k)$
                 $y_{p_1}(k) = \mathbf{q}_{M,:l_1}^H \mathbf{H}_M^H \mathbf{x}_1(k)$
                 Move the $l_1$-th entry of vector $\mathbf{f}(k)$ to the end
                 $\hat{s}_{p_1}(k) = \mathcal{Q}\left[y_{p_1}(k)\right]$
Recursion:       for $m = 1, 2, ..., M-1$
                   (a) $\mathbf{x}_{m+1}(k) = \mathbf{x}_m(k) - \hat{s}_{p_m}(k)\mathbf{h}_{M,:l_m}$
                   (b) Determine $\mathbf{H}_{M-m}$ by removing the $l_m$-th column of $\mathbf{H}_{M-m+1}$
                   (c) $\mathbf{Q}_{M-m} = \left[\mathbf{H}_{M-m}^H \mathbf{H}_{M-m} + \alpha \mathbf{I}_{(M-m)\times(M-m)}\right]^{-1}$
                   (d) $l_{m+1} = \arg\min_i q_{M-m,ii}$,   $p_{m+1} = f_{l_{m+1}}(k)$
                   (e) $y_{p_{m+1}}(k) = \mathbf{q}_{M-m,:l_{m+1}}^H \mathbf{H}_{M-m}^H \mathbf{x}_{m+1}(k)$
                   (f) Move the $l_1$-th entry of vector $\mathbf{f}(k)$ to the position behind
                       the $(M-m)$-th entry
                   (g) $\hat{s}_{p_{m+1}}(k) = \mathcal{Q}\left[y_{p_{m+1}}(k)\right]$
Solutions:       The estimates of the transmitted signals: $[\, \hat{s}_{p_1}(k) \;\; \hat{s}_{p_2}(k) \;\; \cdots \;\; \hat{s}_{p_M}(k) \,]^T$
                 The decoding order: $\mathbf{f}(k) = [\, p_M \;\; p_{M-1} \;\; \cdots \;\; p_1 \,]^T$
```

The matrix $\mathbf{R}$ can be rewritten as follows:

$$\mathbf{R} = \sum_{n=1}^{N} \mathbf{h}_{n:}\mathbf{h}_{n:}^H + \alpha \mathbf{I}_{M\times M} \tag{25}$$

which means that $\mathbf{R}$ can be computed recursively in N iterations as

$$\begin{aligned}\mathbf{R}_{[l]} &= \sum_{n=1}^{l} \mathbf{h}_{n:}\mathbf{h}_{n:}^H + \alpha \mathbf{I}_{M\times M} \\ &= \mathbf{R}_{[l-1]} + \mathbf{h}_{l:}\mathbf{h}_{l:}^H \end{aligned} \tag{26}$$

and

$$\mathbf{R}_{[N]} = \mathbf{R}, \;\; \mathbf{R}_0 = \alpha \mathbf{I}_{M\times M}. \tag{27}$$

Using the Sherman–Morrison formula, $\mathbf{Q}$ can also be computed recursively as

$$\mathbf{Q}_{[l]} = \mathbf{Q}_{[l-1]} - \frac{\mathbf{Q}_{[l-1]}\mathbf{h}_{l:}\mathbf{h}_{l:}^H \mathbf{Q}_{[l-1]}}{1 + \mathbf{h}_{l:}^H \mathbf{Q}_{[l-1]}\mathbf{h}_{l:}}. \tag{28}$$

With the initialization $\mathbf{Q}_0 = (1/\alpha)\mathbf{I}_{M\times M}$, we obtain $\mathbf{Q}_{[N]} = \left[\mathbf{H}^H\mathbf{H} + \alpha\mathbf{I}_{M\times M}\right]^{-1}$, which is $\mathbf{Q}_M$ of dimension $M \times M$. Note that if we start the process at iteration $M+1$ with the initialization $\mathbf{Q}_{[M]} = \sum_{n=1}^{M} \mathbf{h}_{n:}\mathbf{h}_{n:}^H$, we obtain $\mathbf{Q}_{[N]} = \left[\mathbf{H}^H\mathbf{H}\right]^{-1}$. Before going further, it is important to comment on (28). Indeed, it is well known that the computation of any recursion introduces numerical instabilities because of the finite precision of the processor units. This instability occurs only after a very large number of iterations. Fortunately in this application, the number of iterations to compute $\mathbf{Q}$ is limited by the number of receiving antennas (N), which is rather small; therefore, in principle, we should not expect any particular problem here. In any case, the numerical stability can be improved by increasing α at the initialization. Furthermore, as it will become clearer in the following, we can use any method to compute $\mathbf{Q}$ and still have a very efficient algorithm.

In the proposed algorithm, $\mathbf{Q}_{[N]}$ is computed only once at the first iteration using (28). The complexity to compute $\mathbf{Q}_{[N]}$ is in $\mathcal{O}(NM^2)$. Once $\mathbf{Q}_{[N]}$ is computed, it is easy to determine p_1 from (24). Continuing the process for this first iteration, the input estimate is computed as follows:

$$y_{p_1}(k) = \sum_{m=1}^{M} q_{p_1 m}\mathbf{h}_{:m}^H \mathbf{x}(k) \tag{29}$$

and

$$\hat{s}_{p_1}(k) = \mathcal{Q}\left[y_{p_1}(k)\right]. \tag{30}$$

The last step (Step 3) is the same as the one for the V-BLAST algorithm.

For the following iterations, the process is different. We show that the matrix $\mathbf{Q}$ can be deflated recursively. We have

$$\begin{aligned}&\mathbf{Q}_{[N]}\\ &=\mathbf{Q} = \left[\mathbf{H}^H\mathbf{H} + \alpha\mathbf{I}_{M\times M}\right]^{-1} = \mathbf{R}^{-1}\\ &= \begin{bmatrix} \mathbf{h}_{:1}^H\mathbf{h}_{:1}+\alpha & \mathbf{h}_{:1}^H\mathbf{h}_{:2} & \cdots & \mathbf{h}_{:1}^H\mathbf{h}_{:M} \\ \mathbf{h}_{:2}^H\mathbf{h}_{:1} & \mathbf{h}_{:2}^H\mathbf{h}_{:2}+\alpha & \cdots & \mathbf{h}_{:2}^H\mathbf{h}_{:M} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{h}_{:M}^H\mathbf{h}_{:1} & \mathbf{h}_{:M}^H\mathbf{h}_{:2} & \cdots & \mathbf{h}_{:M}^H\mathbf{h}_{:M}+\alpha \end{bmatrix}^{-1}.\end{aligned} \tag{31}$$

After p_1 corresponding to the element $y_{p_1}(k)$ with the smallest variance is determined, we can interchange the p_1th and Mth entries of the transmitted signal $\mathbf{s}(k)$ such that the Mth signal is currently the best estimate. Of course, the indices of the transmitted signals will be tracked after the reordering. Accordingly, the p_1th and Mth columns of the channel matrix $\mathbf{H}$ should be interchanged, which can be easily done by post-multiplying $\mathbf{H}$ with a permutation matrix $\mathbf{P}_{p_1M}$, which is given by

$$\mathbf{P}_{p_1M} = \left[\begin{array}{cccccccc} 1 & 0 & \cdots & & & & \cdots & 0 \\ 0 & \ddots & & & & & & \vdots \\ \vdots & & 1 & & & & & \\ & & & 0 & \cdots & & & 1 \\ & & & \vdots & 1 & & & 0 \\ & & & & & \ddots & & \vdots \\ & & & & & & 1 & 0 \\ 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & 0 \\ & & & \uparrow & & & & \uparrow \\ & & & p_1 & & & & M \end{array}\right]_{M\times M}.$$

Since

$$\begin{aligned}&\left(\mathbf{HP}_{p_1M}\right)^H\left(\mathbf{HP}_{p_1M}\right) + \alpha\mathbf{I}_{M\times M}\\ &\qquad = \mathbf{P}_{p_1M}\left(\mathbf{H}^H\mathbf{H} + \alpha\mathbf{I}_{M\times M}\right)\mathbf{P}_{p_1M},\end{aligned} \tag{32}$$

it follows that the rows and columns p_1 and M of the matrix $\mathbf{R}$ should be permuted. Equivalently, we can permute the rows and columns p_1 and M of the matrix $\mathbf{Q}$, which can be easily seen from

$$\left(\mathbf{P}_{p_1M}\mathbf{R}\mathbf{P}_{p_1M}\right)^{-1} = \mathbf{P}_{p_1M}\mathbf{R}^{-1}\mathbf{P}_{p_1M} = \mathbf{P}_{p_1M}\mathbf{Q}\mathbf{P}_{p_1M}. \tag{33}$$

For easy presentation and without much confusion, we still use $\mathbf{Q}_M$ to denote the matrix after permutation, i.e., $\mathbf{Q}_M \overset{\triangle}{=} \mathbf{P}_{p_1M}\mathbf{Q}\mathbf{P}_{p_1M}$. These permutations will allow us to remove the effect of the channel $\mathbf{h}_{:p_1}$ easily. In this case, we have

$$\begin{aligned}\mathbf{Q}_M &= \begin{bmatrix} \mathbf{h}_{:1}^H\mathbf{h}_{:1}+\alpha & \mathbf{h}_{:1}^H\mathbf{h}_{:2} & \cdots & \mathbf{h}_{:1}^H\mathbf{h}_{:p_1} \\ \mathbf{h}_{:2}^H\mathbf{h}_{:1} & \mathbf{h}_{:2}^H\mathbf{h}_{:2}+\alpha & \cdots & \mathbf{h}_{:2}^H\mathbf{h}_{:p_1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{h}_{:p_1}^H\mathbf{h}_{:1} & \mathbf{h}_{:p_1}^H\mathbf{h}_{:2} & \cdots & \mathbf{h}_{:p_1}^H\mathbf{h}_{:p_1}+\alpha \end{bmatrix}^{-1}\\ &= \begin{bmatrix} \mathbf{R}_{M-1} & \mathbf{v}_{M-1} \\ \mathbf{v}_{M-1}^H & \beta_{p_1} \end{bmatrix}^{-1}\end{aligned} \tag{34}$$

where

$$\begin{aligned}\beta_{p_1} &= \mathbf{h}_{:p_1}^H\mathbf{h}_{:p_1} + \alpha\\ \mathbf{v}_{M-1} &= \left[\mathbf{h}_{:1}^H\mathbf{h}_{:p_1} \quad \mathbf{h}_{:2}^H\mathbf{h}_{:p_1} \quad \cdots \quad \mathbf{h}_{:M-1}^H\mathbf{h}_{:p_1}\right]^T\end{aligned}$$

and

$$\mathbf{R}_{M-1} = \mathbf{H}_{M-1}^H\mathbf{H}_{M-1} + \alpha\mathbf{I}_{(M-1)\times(M-1)}.$$

It can easily be shown that

$$\mathbf{Q}_M = \begin{bmatrix} \mathbf{T}_{M-1}^{-1} & -\frac{\mathbf{T}_{M-1}^{-1}\mathbf{v}_{M-1}}{\beta_{p_1}} \\ -\frac{\mathbf{v}_{M-1}^H\mathbf{T}_{M-1}^{-1}}{\beta_{p_1}} & \frac{1}{\beta_{p_1}} + \frac{\mathbf{v}_{M-1}^H\mathbf{T}_{M-1}^{-1}\mathbf{v}_{M-1}}{\beta_{p_1}^2} \end{bmatrix} \tag{35}$$

where

$$\mathbf{T}_{M-1} = \mathbf{R}_{M-1} - \frac{\mathbf{v}_{M-1}\mathbf{v}_{M-1}^H}{\beta_{p_1}} \tag{36}$$

is the Schur complement of β_{p_1} in $\mathbf{Q}_M^{-1}$. Furthermore, from (36), we deduce that

$$\mathbf{R}_{M-1}^{-1} = \mathbf{Q}_{M-1} = \left[\mathbf{T}_{M-1} + \frac{\mathbf{v}_{M-1}\mathbf{v}_{M-1}^H}{\beta_{p_1}}\right]^{-1} \tag{37}$$

and using the Sherman–Morrison formula, we obtain

$$\mathbf{Q}_{M-1} = \mathbf{T}_{M-1}^{-1} - \frac{\mathbf{T}_{M-1}^{-1}\mathbf{v}_{M-1}\mathbf{v}_{M-1}^H\mathbf{T}_{M-1}^{-1}}{\beta_{p_1} + \mathbf{v}_{M-1}^H\mathbf{T}_{M-1}^{-1}\mathbf{v}_{M-1}}. \tag{38}$$

Clearly, (38) shows that the matrix $\mathbf{Q}$ can be deflated recursively in $\mathcal{O}(M^2)$ at each iteration. In the general case, we have

$$\mathbf{R}_{M-m+1} = \begin{bmatrix} \mathbf{R}_{M-m} & \mathbf{v}_{M-m} \\ \mathbf{v}_{M-m}^H & \beta_{p_m} \end{bmatrix} \tag{39}$$

$$\mathbf{Q}_{M-m+1} = \begin{bmatrix} \mathbf{T}_{M-m}^{-1} & -\frac{\mathbf{T}_{M-m}^{-1}\mathbf{v}_{M-m}}{\beta_{p_m}} \\ -\frac{\mathbf{v}_{M-m}^H\mathbf{T}_{M-m}^{-1}}{\beta_{p_m}} & \frac{1}{\beta_{p_m}} + \frac{\mathbf{v}_{M-m}^H\mathbf{T}_{M-m}^{-1}\mathbf{v}_{M-m}}{\beta_{p_m}^2} \end{bmatrix} \tag{40}$$

$$\mathbf{Q}_{M-m} = \mathbf{T}_{M-m}^{-1} - \frac{\mathbf{T}_{M-m}^{-1}\mathbf{v}_{M-m}\mathbf{v}_{M-m}^H\mathbf{T}_{M-m}^{-1}}{\beta_{p_m} + \mathbf{v}_{M-m}^H\mathbf{T}_{M-m}^{-1}\mathbf{v}_{M-m}}. \tag{41}$$

Note that $\mathbf{R}_{M-m}$ is not computed but rather easily determined from $\mathbf{R}_{M+1-m}$ by removing its last row and column. Only $\mathbf{R}_M = \mathbf{R}$ is calculated at the first iteration. Similarly, $\mathbf{T}_{M-m}^{-1}$ is obtained without additional calculation. Table II summarizes the proposed fast V-BLAST algorithm. The complexity of this algorithm is in $\mathcal{O}(NM^2 + M^3)$. For $N = M$, the complexity is reduced by a factor of M compared with the V-BLAST algorithm.

V. Complexity Evaluation

We now look at the computational complexity of the proposed fast V-BLAST algorithm and compare it with the traditional V-BLAST and the square-root algorithms [4]. Since the transmitted and received signals as well as the channel matrix are complex, all processing is conducted on complex values. Therefore, unless otherwise specified, multiplications, divisions, and additions refer to complex operations throughout this section.

TABLE II
FAST V-BLAST ALGORITHM

```
Initialization: $\mathbf{x}_1(k) = \mathbf{x}(k)$
    Compute $\mathbf{R} = \mathbf{R}_M$ and $\mathbf{Q} = \mathbf{Q}_M$ recursively using (26) and (28)
    $\mathbf{f}(k) = [\,1 \;\; 2 \;\; \cdots \;\; M\,]^T$
    $l_1 = \arg\min_i q_{M,ii}, \quad p_1 = f_{l_1}(k)$
    $y_{p_1}(k) = \sum_{i=1}^{M} q_{M,l_1 i}\mathbf{h}_{:i}^H \mathbf{x}_1(k)$
    Interchange the entries $l_1$ and $M$ of the vector $\mathbf{f}(k)$
    $\hat{s}_{p_1}(k) = \mathcal{Q}\,[y_{p_1}(k)]$
Recursion: for $m = 1, 2, ..., M-1$
    (a) $\mathbf{x}_{m+1}(k) = \mathbf{x}_m(k) - \hat{s}_{p_m}(k)\mathbf{h}_{:p_m}$
    (b) Permute the rows and columns $l_m$ and $M-m+1$ of $\mathbf{R}_{M+1-m}$
    (c) Permute the rows and columns $l_m$ and $M-m+1$ of $\mathbf{Q}_{M+1-m}$
    (d) Determine $\mathbf{R}_{M-m}, \mathbf{v}_{M-m},$ and $\beta_{p_m}$ from $\mathbf{R}_{M+1-m}$
    (e) Determine $\mathbf{T}_{M-m}^{-1}$ from $\mathbf{Q}_{M+1-m}$ by removing its last row and column
    (f) Compute $\mathbf{Q}_{M-m}$ recursively using (41)
    (g) $l_{m+1} = \arg\min_i q_{M-m,ii}, \quad p_{m+1} = f_{l_{m+1}}(k)$
    (h) $y_{p_{m+1}}(k) = \sum_{i=1}^{M-m} q_{M-m,l_{m+1}i}\mathbf{h}_{:f_i(k)}^H \mathbf{x}_{m+1}(k)$
    (i) Interchange the entries $l_{m+1}$ and $M-m$ of the vector $\mathbf{f}(k)$
    (j) $\hat{s}_{p_{m+1}}(k) = \mathcal{Q}\,[y_{p_{m+1}}(k)]$
Solutions: The estimates of the transmitted signals: $[\,\hat{s}_{p_1}(k) \;\; \hat{s}_{p_2}(k) \;\; \cdots \;\; \hat{s}_{p_M}(k)\,]^T$
    The decoding order: $\mathbf{f}(k) = [\,p_M \;\; p_{M-1} \;\; \cdots \;\; p_1\,]^T$
```

The computational complexity of the traditional V-BLAST method can be evaluated as follows.

- Using the traditional V-BLAST algorithm, we need to directly invert the complex matrix
$$\mathbf{R}_{M-m} = \mathbf{H}_{M-m}^H \mathbf{H}_{M-m} + \alpha \mathbf{I}_{(M-m)\times(M-m)} \tag{42}$$
with dimensions $(M-m)\times(M-m)$ at the mth step of the recursion. With Gauss–Jordan, computing such an inverse requires $(M-m)^3$ multiplications/divisions and $(M-m)^3 - 2(M-m)^2 + (M-m)$ additions. In spite of the efficiency of the Gauss–Jordan method, it is barely used especially in a fixed-point implementation because of its poor stability. The most numerically stable way to compute $\mathbf{R}_{M-m}^{-1}$ is via singular value decomposition (SVD) of $\mathbf{H}_{M-m}$, which is given by
$$\mathbf{H}_{M-m} = \mathbf{U}_{M-m}\boldsymbol{\Sigma}_{M-m}\mathbf{V}_{M-m}^H \tag{43}$$
where $\mathbf{U}_{M-m} \in \mathbb{C}^{N\times N}$ and $\mathbf{V}_{M-m} \in \mathbb{C}^{(M-m)\times(M-m)}$ are unitary matrices, and $\boldsymbol{\Sigma}_{M-m} \in \mathbb{R}^{N\times(M-m)}$ has the form
$$\boldsymbol{\Sigma}_{M-m} = \text{diag}(\lambda_{M-m,1},\ \lambda_{M-m,2}, \ldots, \lambda_{M-m,M-m}).$$
Substituting (43) into (42) yields
$$\mathbf{R}_{M-m} = \mathbf{V}_{M-m}\left(\boldsymbol{\Sigma}_{M-m,c}^2 + \alpha\mathbf{I}_{(M-m)\times(M-m)}\right)\mathbf{V}_{M-m}^H \tag{44}$$
where
$$\boldsymbol{\Sigma}_{M-m,c}^2 = \text{diag}\left(\lambda_{M-m,1}^2, \lambda_{M-m,2}^2, \ldots, \lambda_{M-m,M-m}^2\right) \in \mathbb{R}^{(M-m)\times(M-m)}.$$
Taking the inverse of (44) produces
$$\mathbf{Q}_{M-m} = \mathbf{V}_{M-m}\left(\boldsymbol{\Sigma}_{M-m,c}^2 + \alpha\mathbf{I}_{(M-m)\times(M-m)}\right)^{-1}\mathbf{V}_{M-m}^H = \sum_{i=1}^{M-m}\gamma_{M-m,i}\mathbf{v}_{M-m,:i}\mathbf{v}_{M-m,:i}^H \tag{45}$$
where
$$\gamma_{M-m,i} = \frac{1}{(\lambda_{M-m,i}^2 + \alpha)}$$
and $\mathbf{v}_{M-m,:i}$ is the ith column vector of matrix $\mathbf{V}_{M-m}$.
Using the Golub–Reinsch algorithm, the complexity of performing an SVD of $\mathbf{H}_{M-m}$ to compute only $\boldsymbol{\Sigma}_{M-m}$ and $\mathbf{V}_{M-m}$ is $4N(M-m)^2 + 8(M-m)^3$, which implies approximately equal numbers of multiplications and

  additions [8]. After $\mathbf{\Sigma}_{M-m}$ and $\mathbf{V}_{M-m}$ are determined, forming $\mathbf{Q}_{M-m}$ according to (45) requires $(M-m)^3 + (M-m)^2$ multiplications and $(M-m)^3 - (M-m)^2$ additions. Therefore, computing $\mathbf{Q}_{M-m}$ needs $9(M-m)^3 + (4N+1)(M-m)^2$ multiplications and $9(M-m)^3 + (4N-1)(M-m)^2$ additions.

- In order to compute $y_{p_{m+1}}$ following Step (e) in Table I, $(M-m)(N+1)$ multiplications and $(M-m)(N+1)-1$ additions are necessary.
- Nulling out the effect of source s_{p_m} on the received signal requires N multiplications and N additions.

Collecting these results, since there is a one-step initialization and $(M-1)$ steps in the recursion with the traditional V-BLAST algorithm, the total number of multiplications is as shown in the first equation at the bottom of the page, and the total number of additions is as shown in the second equation at the bottom of the page. If the numbers of transmitting and receiving antennas are the same, i.e., $M = N$, then the total numbers of multiplications and additions are $(43/12)M^4 + (22/3)M^3 + \mathcal{O}\left(M^2\right)$ and $(43/12)M^4 + (20/3)M^3 + \mathcal{O}\left(M^2\right)$, respectively.

In the square-root algorithm for V-BLAST decoding, the square-root matrix $\mathbf{Q}_{M-m}^{1/2}$ of $\mathbf{Q}_{M-m}$ is recursively computed by using Householder transformations. Applying a Householder transformation to a given matrix with respect to one of its column/row vector requires equal numbers of multiplications and additions. As given in [4], the square-root algorithm requires $(2/3)M^3 + 7M^2N + 2MN^2 + \mathcal{O}\left(M^2 + MN\right)$ multiplications and additions. If $M = N$, then these numbers turn to $(29/3)M^3 + \mathcal{O}\left(M^2\right)$. Indeed, square-root operations were omitted in the evaluation.

Let us compute the computational complexity of the proposed fast V-BLAST algorithm:

- In the initialization, we need to determine $\mathbf{R}_M$, $\mathbf{Q}_M$, and $y_{p_1}(k)$. The computational cost of these operations are given as follows.
  - Determining $\mathbf{R}_M$ using the recursive method has no computational advantage over the direct matrix multiplication. The numbers of necessary multiplications and additions are $MN(M+1)/2$ and $(M^2N + MN - M^2)/2$, respectively.
  - On the other hand, it is efficient to compute $\mathbf{Q}_M$ recursively using (28). At each step of the recursion, at least $2\left(M^2 + M\right)$ multiplications and $\left(3M^2 + M\right)/2$ additions are necessary. Since there are N steps to determine $\mathbf{Q}_M$, the total numbers of multiplications and additions are $2MN(M+1)$ and $MN(3M+1)/2$, respectively.
  - Computing $y_{p_1}(k)$ according to (29) requires $M(N+1)$ multiplications and $(MN-1)$ additions. Thus, the initialization takes $(5M^2N/2 + 7MN/2 + M)$ multiplications and $(2M^2N + 2MN - M^2/2 - 1)$ additions.
- Consider the mth ($m = 1, 2, \ldots, M-1$) step of the recursion.
  - It takes N multiplications and N additions to null out the effect of the source s_{p_m} on the received signals.
  - Computing Q_{M-m} using (41) needs $2(M-m)(M-m+1)$ multiplications and $(M-m)(3M-3m+1)/2$ additions.
  - To estimate $y_{p_{m+1}}(k)$ according to Step (h) in Table II, $(M-m)(N+1)$ multiplications and $(MN - mN - 1)$ additions are necessary.

Hence, in the recursion, the number of multiplications is

$$\sum_{m=1}^{M-1} \left[N + 2(M-m)(M-m+1) + (M-m)(N+1)\right] = \frac{2}{3}M^3 + \frac{1}{2}M^2N + \frac{1}{2}M^2 + \frac{1}{2}MN - \frac{7}{6}M - N$$

and the number of additions is

$$\sum_{m=1}^{M-1} \left[N + \frac{(M-m)(3M-3m+1)}{2} + (MN - mN - 1)\right] = \frac{1}{2}M^3 + \frac{1}{2}M^2N - \frac{1}{2}M^2 + \frac{1}{2}MN - M - N + 1.$$

$$\begin{aligned}&\sum_{m=0}^{M-1} \left[9(M-m)^3 + (4N+1)(M-m)^2 + (M-m)(N+1)\right] + \sum_{m=1}^{M-1} N\\ =&\frac{9}{4}M^4 + \frac{4}{3}M^3N + \frac{29}{6}M^3 + \frac{5}{2}M^2N + \frac{13}{4}M^2 + \frac{13}{6}MN + \frac{2}{3}M - N\\ =&\frac{9}{4}M^4 + \frac{4}{3}M^3N + \frac{29}{6}M^3 + \frac{5}{2}M^2N + \mathcal{O}\left(M^2 + MN\right)\end{aligned}$$

$$\begin{aligned}&\sum_{m=0}^{M-1} \left[9(M-m)^3 + (4N-1)(M-m)^2 + (M-m)(N+1) - 1\right] + \sum_{m=1}^{M-1} N\\ =&\frac{9}{4}M^4 + \frac{4}{3}M^3N + \frac{25}{6}M^3 + \frac{5}{2}M^2N + \frac{9}{4}M^2 + \frac{13}{6}MN - \frac{2}{3}M - N\\ =&\frac{9}{4}M^4 + \frac{4}{3}M^3N + \frac{25}{6}M^3 + \frac{5}{2}M^2N + \mathcal{O}\left(M^2 + MN\right).\end{aligned}$$

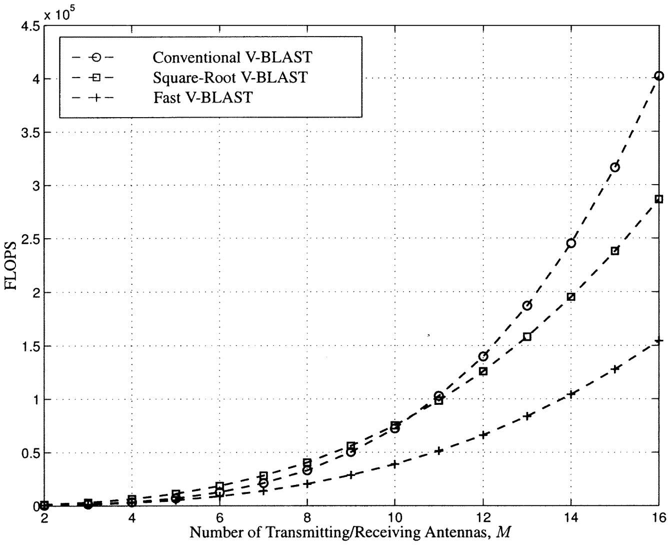


Fig. 1. Comparison of computational complexity among the V-BLAST, the square-root V-BLAST, and the proposed fast V-BLAST algorithms for different numbers of antennas.

Summing up the complexity in the initialization and recursion, we get the total number of multiplications

$$\frac{2}{3}M^3 + 3M^2N + \frac{1}{2}M^2 + 4MN - \frac{1}{6}M - N$$
$$= \frac{2}{3}M^3 + 3M^2N + \mathcal{O}\left(M^2 + MN\right)$$

and the total number of additions

$$\frac{1}{2}M^3 + \frac{5}{2}M^2N - M^2 + \frac{5}{2}MN - M - N$$
$$= \frac{1}{2}M^3 + \frac{5}{2}M^2N + \mathcal{O}\left(M^2 + MN\right).$$

If $M = N$, then the proposed fast V-BLAST algorithm requires $(11/3)M^3 + \mathcal{O}\left(M^2\right)$ multiplications and $3M^3 + \mathcal{O}\left(M^2\right)$ additions. Therefore, the speedups of the proposed algorithm over the traditional V-BLAST in the number of multiplications and additions are $43M/44 + 2 \approx M + 2$ and $43M/36 + 20/9 \approx 1.2M + 2.2$, respectively. Compared with the square-root algorithm, the proposed algorithm is also more efficient, and the speedups in the number of multiplications and additions are $29/11 \approx 2.6$ and $29/6 \approx 4.8$, respectively.

Note that one complex multiplication/division takes six floating-point operations (flops) and one complex addition/subtraction needs two flops. Therefore, the flop counts of the traditional V-BLAST and the square-root algorithms are approximately $(43/42)M + 43/21 \approx M + 2$ times and $58/21 \approx 2.76$ times, respectively, more than that of the proposed algorithm in the case of $M = N$.

In order to justify the complexity analysis presented in this section, we carried out some numerical experiments to count the floating-point operations (flops) per data sample for the studied V-BLAST algorithms for different numbers of transmitting/receiving antennas. It is well known that for a common floating-point implementation of an algorithm, the flops dominate the calculation, and the number of flops is a consistent measure of the algorithm's computational complexity, independent of what machine it runs on. Although the absolute number of flops for the studied algorithms are not particularly meaningful, their relative values illustrate the great efficiency of the proposed fast V-BLAST method. The result is shown in Fig. 1. As can be clearly seen, the square-root V-BLAST algorithm is more efficient than the traditional V-BLAST only when the number of antennas is large, in particular when M is greater than 10, but so far, only four or eight antennas are interesting in practice, and the square-root V-BLAST algorithm is not advantageous. The proposed fast V-BLAST has the least flops for all numbers of transmitting/receiving antennas.

VI. Conclusions

A general V-BLAST system with M transmitting antennas and N receiving antennas was studied, and an efficient algorithm with low computational complexity was developed for optimum sequential nulling and cancellation detection scheme. The proposed algorithm avoids directly inverting a matrix and finds the minimum mean-square error filter coefficients via induction with the help of the inverse of a block partitioned matrix and the Sherman–Morrison formula. Compared with $\mathcal{O}\left(M^4 + M^3N\right)$ complex operations that are required to determine the optimum detection order and estimate the transmitted signals using the traditional direct matrix-inversion method, the proposed algorithm requires $\mathcal{O}\left(M^3 + M^2N\right)$ operations, which is a complexity reduction by a factor of M. To be exact, when $N = M$, the proposed algorithm reduces the flop counts over the traditional method by a factor of $M + 2$. A comparison between the proposed algorithm and the square-root method whose computational complexity is also on the order of $\mathcal{O}\left(M^3 + M^2N\right)$ reveals that the proposed algorithm is still more efficient with a speedup of 2.76 in flops when $M = N$. This paper addressed the computational complexity of V-BLAST algorithms in a rather simple flat-fading environment. Further work will extend to the spatio-temporal BLAST systems with more practical and more general frequency-selective wireless channels.

References

[1] I. E. Telatar, "Capacity of multi-antenna Gaussian channels," Bell Labs., Murray Hill, NJ, 1995.

[2] G. J. Foschini, "Layered space-time architecture for wireless communication in a fading environment using multi-element antennas," *Bell Labs Tech. J.*, vol. 1, no. 2, pp. 41–59, 1996.

[3] 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," in *Proc. ISSSE*, 1998, pp. 295–300.

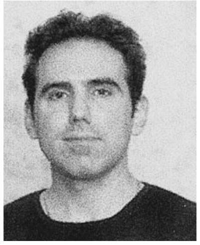
[4] B. Hassibi, "A fast square-root implementation for BLAST," in *Conf. Rec. Thirty-Fourth Asilomar Conf. Signals, Syst. Comput.*, 2000, pp. 1255–1259.

[5] G. D. Golden, C. J. Foschini, R. A. Valenzuela, and P. W. Wolniansky, "Detection algorithm and initial laboratory results using V-BLAST space-time communication architecture," *Electron. Lett.*, vol. 35, no. 1, pp. 14–16, Jan. 1999.

[6] C. B. Papadias, Multiple Antenna Transceivers for Wireless Communications: Capacity Considerations, J. G. Proakis, Ed. New York: Wiley Encyclopedia of Telecommunications, to be published.

[7] T. L. Marzetta, "BLAST training: Estimating the channel characteristics for high capacity space-time wireless," in *Proc. 37th Annual Allerton Conf. Commun., Contr., Comput.*, Monticello, IL, Sept. 1999.

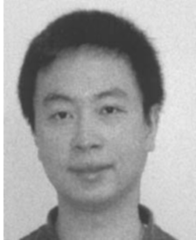
[8] G. H. Golub and C. F. Van Loan, *Matrix Computations*, third ed. Baltimore, MD: Johns Hopkins Univ. Press, 1996.



Jacob Benesty (M'92) was born in Marrakech, Morocco, in 1963. He received the Masters degree in microwaves from Pierre & Marie Curie University, Paris, France, in 1987 and the Ph.D. degree in control and signal processing from Orsay University, Paris, in April 1991.

During his Ph.D. program (from November 1989 to April 1991), he worked on adaptive filters and fast algorithms at the Centre National d'Etudes des Telecommunications (CNET), Paris. From January 1994 to July 1995, he worked at Telecom Paris on multichannel adaptive filters and acoustic echo cancellation. He joined Bell Labs, Lucent Technologies (formerly AT&T), Murray Hill, NJ, in October 1995, first as a Consultant and then as a Member of Technical Staff. Since this date, he has been working on stereophonic acoustic echo cancellation, adaptive filters, source localization, robust network echo cancellation, and blind deconvolution. He co-authored the book *Advances in Network and Acoustic Echo Cancellation* (Berlin, Germany: Springer-Verlag, 2001). He is also a co-editor/co-author of the books *Adaptive Signal Processing: Applications to Real-World Problems* (Berlin, Germany: Springer-Verlag, 2003) and *Acoustic Signal Processing for Telecommunication* (Boston, MA: Kluwer, 2000).

Dr. Benesty received the 2001 Best Paper Award from the IEEE Signal Processing Society. He was the co-chair of the 1999 International Workshop on Acoustic Echo and Noise Control.



Yiteng (Arden) Huang (S'97–M'01) received the B.S. degree from the Tsinghua University, Beijing, China, in 1994 and the M.S. and Ph.D. degrees from the Georgia Institute of Technology (Georgia Tech), Atlanta, in 1998 and 2001, respectively, all in electrical and computer engineering.

During his doctoral studies from 1998 to 2001, he was a research assistant with the Center of Signal and Image Processing, Georgia Tech, and was a teaching assistant with the School of Electrical and Computer Engineering. In the summers from 1998 to 2000, he worked with Bell Laboratories, Murray Hill, NJ, where he engaged in research on passive acoustic source localization with microphone arrays. Upon graduation, he joined Bell Laboratories as a Member of Technical Staff in March 2001. His current research interests are in adaptive filtering, multichannel signal processing, source localization, microphone arrays for hands-free telecommunication, statistical signal processing, and wireless communications. He is a co-editor/co-author of the book *Adaptive Signal Processing: Applications to Real-World Problems* (Berlin, Germany: Springer-Verlag, 2003).

Dr. Huang is currently an associate editor of the IEEE SIGNAL PROCESSING LETTERS. He received the 2002 Young Author Best Paper Award from the IEEE Signal Processing Society, the 2000–2001 Outstanding Graduate Teaching Assistant Award from the School Electrical and Computer Engineering, Georgia Tech, the 2000 Outstanding Research Award from the Center of Signal and Image Processing, Georgia Tech, and the 1997–1998 Colonel Oscar P. Cleaver Outstanding Graduate Student Award from the School of Electrical and Computer Engineering, Georgia Tech.

Jingdong Chen (M'99) received the B.S. degree in electrical engineering and the M.S. degree in array signal processing from the Northwestern Polytechnic University, Xian, China, in 1993 and 1995, respectively, and the Ph.D. degree in pattern recognition and intelligence control from the Chinese Academy of Sciences, Beijing, China, in 1998.

His Ph.D. research focused on speech recognition in noisy environments. He studied and proposed several techniques covering speech enhancement and hidden Markov model adaptation by signal transformation. From 1998 to 1999, he was with ATR Interpreting Telecommunications Research Laboratories, Kyoto, Japan, where he conducted research on speech synthesis, speech analysis, as well as objective measurements for evaluating speech synthesis. He then joined the Griffith University, Brisbane, Australia, as a research fellow, where he engaged in research in robust speech recognition, signal processing, and discriminative feature representation. From 2000 to 2001, he was with ATR Spoken Language Translation Research Laboratories, Kyoto, where he conducted research in robust speech recognition and speech enhancement. He joined Bell Laboratories, Murray Hill, NJ, as a Member of Technical Staff in July 2001. His current research interests include adaptive signal processing, speech enhancement, adaptive noise/echo cancellation, microphone array signal processing, signal separation, and source localization.

Dr. Chen is the recipient of a 1998–1999 research grant from the Japan Key Technology Center and the 1996–1998 President's Award from the Chinese Academy of Sciences.