AN EFFICIENT SQUARE-ROOT ALGORITHM FOR BLAST

Babak Hassibi

Mathematics of Communications Research
Bell Labs, Lucent Technologies, Murray Hill, NJ 07974

ABSTRACT

Bell Labs Layered Space-Time (BLAST) is a scheme for transmitting information over a rich-scattering wireless environment using multiple receive and transmit antennas. The main computational bottleneck in the BLAST algorithm is a "nulling and cancelation" step, where the optimal ordering for the sequential estimation and detection of the received signals is determined. To reduce the computational cost of BLAST, in this paper we develop an efficient square-root algorithm for the nulling and cancellation step. The main features of the algorithm include *efficiency*: the computational cost is reduced by $0.7M$, where M is the number of transmit antennas, and *numerical stability*: the algorithm is division-free and uses only orthogonal transformations. In a 14 antenna system designed for transmission of 1 Mbit/sec over a 30 kHz channel, the nulling and cancellation computation is reduced from 190 MFlops/sec to 19 MFlops/sec, with the overall computations being reduced from 220 MFlops/sec to 49 MFlops/sec. The numerical stability of the algorithm also make it attractive for implementation in fixed-point (rather than floating-point) architectures.

1. INTRODUCTION

In theory multiple transmit and receive antennas can greatly increase the capacity, as well as significantly lower the probability of error, of a wireless communications link. One practical scheme for for transmittng information over a flat-fading, rich-scattering, wireless environment is Bell Labs Layered Space-Time (BLAST) [1]. BLAST has the potential to increase the capacity of the wireless link by a factor of M (where M is the minimum of the number of receive and transmit antennas), and is applicable for fixed wireless access (as in a wireless LAN).

Since multi-antenna communications allows for information transmission at very high rates, a major issue of concern is to keep the computational complexity of the decoding algorithm within reasonable bounds. Thus, maximum-likelihood decoding, for example, is clearly beyond question since it typically requires a search over the prohibitively large set of all possible transmitted signals. BLAST alleviates this problem by employing a "divide and conqure" decoding strategy: at each time instant, rather than jointly decoding the signals from all the transmit antennas, BLAST first decodes the "strongest" signal, then cancels the effect of this strongest transmit signal from each of the received signals, and then proceeds to decode the "strongest" of the remaining transmit signals, and so on. This detection scheme is referred to as sequential *nulling and cancellation*. The optimal detection order is from the strongest to the weakest signal, since this minimizes the propagation of errors from one step of detection to the next.

It turns out that the main computational bottelneck in the BLAST algorithm is the step where the optimal ordering for the sequential estimation and detection of the transmitted signals, as well as the corresponding so-called *nuling vectors*, is determined. Current implementations of BLAST devote over 80% of the total computational cost to this step. This high computational cost limits the scope of applications that admit inexpensive real-time solutions. Moreover, when the number of transmit and receive antennas is large, the repeated pseudoinverse computations that BLAST requires can lead to numerical instability, and so a numerically robust and stable algorithm is desired.

In an attempt to reduce the computational complexity of BLAST, in this paper we develop an efficient square-root algorithm for the nulling-vector optimal-ordering step. The main features of the algorithm include *efficiency*: the computational cost is reduced by an order of magnitude, effectively from $O(M^3)$ to $O(M^4)$, and *numerical stability*: the algorithm is division-free and uses only orthogonal transformations.

The remainder of the paper is organized as follows. Sec. 2 describes the basic model for BLAST and outlines the nulling and cancellation idea. The optimal detection order and the nulling vectors are obtained from the repeated computation of the pseudoinverses of certain deflated subchannel matrices. Sec. 3 gives the main result of this paper. It is first shown how it is possible to obtain the optimal ordering and all the nulling vectors by implicitly computing a *single* pseudoinverse — thereby leading to an efficient algorithm. Then it is shown how all the quantities of interest can be computed using a division-free algorithm that employs only orthogonal transformations — thereby leading to a numerically stable algorithm. The paper is concluded with Sec. 4.

2. THE BASIC IDEA OF BLAST

Consider the setting of Fig. 1 where we have M signals impinging on an array of N ($N \geq M$) receivers via a *rich scattering* flat-fading environment. The signals may either come from an array of transmit antennas (as in BLAST) or from M independent transmit antennas (as in the uplink of a wireless LAN).

The assumption of a flat-fading environment essentially means that the signals are narrow-band, so that we may

0-7803-6293-4/00/$10.00 ©2000 IEEE.

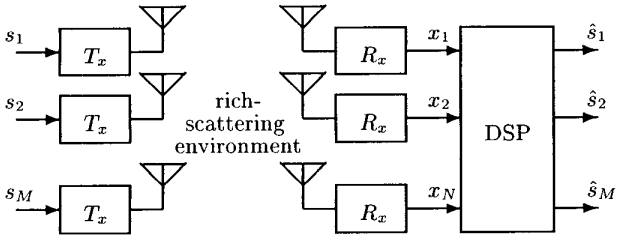

Figure 1: The basic model for BLAST.

assume a channel that is not frequency-selective. In this case, the relationship between the transmit and received signals at each time instant may be written as:

$$x = Hs + v, \quad x = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}, \; s = \begin{bmatrix} s_1 \\ \vdots \\ s_M \end{bmatrix} \tag{1}$$

where $H \in \mathcal{C}^{N \times M}$ is the channel matrix and $v \in \mathcal{C}^N$ represents spatially and temporally additive white noise. The assumption that the channel is rich-scattering essentially means that the elements of the channel matrix (when viewed as random variables) are independent of one another.

Finally, it is also convenient to partition the channel matrix into its rows and columns as follows:

$$H = \begin{bmatrix} H_1 \\ \vdots \\ H_N \end{bmatrix} = \begin{bmatrix} \underline{h}_1 & \cdots & \underline{h}_M \end{bmatrix}.$$

2.1. The Procedure

In BLAST information is transmitted in bursts of length $L_T + L_P$, where L_T denotes the length of the training sequence and L_P denotes the length of the payload. Detecting the payload signals consists of three main steps:

1. *Estimate the channel matrix* using the training sequence.
2. *Determine the optimal detecting order and the MMSE nulling vectors.*
3. *Successive nulling and cancellation:*
   (a) obtain the least-mean-squares estimate of the "strongest" transmit signal.
   (b) slice the least-mean-squares estimate to the nearest value in the signal constellation.
   (c) cancel the effect of the sliced strongest transmit signal from the vector of received signals and return to step (3a).

We now focus on step 2 of the above procedure.

2.2. Optimal Ordering and MMSE Nulling Vectors

Assume that the signal s and additive noise v are zero-mean, spatially white, uncorrelated random vectors with variances $\frac{1}{\alpha} I_M$ and I_N, respectively ($\frac{1}{\alpha}$ is the SNR):

$$Ess^* = \frac{1}{\alpha} I_M, \quad Evv^* = I_N, \quad Esv^* = 0$$

The linear least-mean-squares estimate of s, given the observations $x = Hs + v$, is

$$\hat{s} = (\alpha I + H^* H)^{-1} H^* x = \begin{bmatrix} H \\ \sqrt{\alpha} I_M \end{bmatrix}^\dagger \begin{bmatrix} x \\ 0 \end{bmatrix},$$

where $\dagger$ represents the *pseudo-inverse*. Denoting the first N columns of the pseudo-invserse by $H_\alpha^\dagger$, and the i-th row of $H_\alpha^\dagger$ by $H_{\alpha,i}^\dagger$, we have

$$\hat{s} = H_\alpha^\dagger x \quad \text{and} \quad \hat{s}_i = H_{\alpha,i}^\dagger x. \tag{2}$$

$H_{\alpha,i}$ is referred to as the i-th *MMSE nulling vector*. The covariance matrix for the estimation error $s - \hat{s}$ is readily seen to be

$$E\,(s - \hat{s})\,(s - \hat{s})^* = (\alpha I + H^* H)^{-1} \triangleq P,$$

or, using the pseudo-inverse:

$$P = \begin{bmatrix} H \\ \sqrt{\alpha} I_M \end{bmatrix}^\dagger \left(\begin{bmatrix} H \\ \sqrt{\alpha} I_M \end{bmatrix}^\dagger \right)^* . \tag{3}$$

Clearly, the "strongest" signal among the entries of s will be the one with the smallest error covariance, *i.e.*, the one for which P_{ii} is the smallest.

Suppose that the order of the entries of s are arranged such that the strongest signal is the M-th entry. Then we can independently slice $\hat{s}_M = H_{\alpha,M}^\dagger x$ to obtain a fairly accurate estimate of the signal s_M. If we denote the sliced value by s_M (*i.e.*, if we assume correct detection), then we can use s_M to improve our estimate of the remaining $M-1$ signals. Thus treating s_M as a known quantity, we obtain the following reduced order problem:

$$x - \underline{h}_M s_M = H^{(M-1)} s^{(M-1)} + v, \tag{4}$$

where we have defined the deflated channel matrix, $H^{(M-1)}$, and the reduced signal vector, $s^{(M-1)}$, as

$$H^{(M-1)} = \begin{bmatrix} \underline{h}_1 & \cdots & \underline{h}_{M-1} \end{bmatrix} \text{ and } s^{(M-1)} = \begin{bmatrix} s_1 \\ \vdots \\ s_{M-1} \end{bmatrix}.$$

The solution to (4) clearly requires us to compute the pseudo-inverse of $H_\alpha^{(M-1)}$ and the corresponding error covariance matrix $(\alpha I + H^{(M-1)*} H^{(M-1)})^{-1} \triangleq P^{(M-1)}$.

2.2.1. Basic Algorithm

The basic idea can thus be summarized as follows:

1. Find $P = (\alpha I + H^* H)^{-1}$ and $H_\alpha^\dagger$.
2. Find the smallest diagonal entry of P and reorder the entries of s so that the smallest diagonal entry is the last (M-th) one.
3. Form the least-mean-squares estimate $\hat{s}_M = H_{\alpha,M}^\dagger x$.
4. Obtain s_M (via slicing) from $\hat{s}_M = H_{\alpha,M}^\dagger x$.
5. Cancel the effect of s_M and consider the *reduced-order* problem (4).
6. Continue to find $P^{(M-1)}$ and $H_\alpha^{(M-1)\dagger}$, and so on.

2.3. Computational Complexity

The computational complexity of each step of the method just described can be given as follows.

- Channel estimation: $2MN\log_2 L_T$ (assuming the training sequence is obtained from an FFT matrix.)
- Determining the nulling vectors and optimal ordering: This requires the computation of M pseudoinverses, one for each deflated subchannel matrix. The most numerically stable way to compute the pseudoinverse is via the singular value decomposition. For this scheme, using [2], the computational complexity can be shown to be

$$N^2M^2 + 2NM^3 + \frac{15}{4}M^4.$$

- Processing the payload: $2MNL_P$.

When $M = N$ these complexities simplify to $2M^2\log_2 L_T$, $\frac{27}{4}M^4$ and $2M^2L_P$, respectively. Thus the complexity of the optimal ordering and nulling vector computation grows as the fourth power of the number of transmit antennas.

To gain some perspective on the relative computational complexities of the above three steps, let us consider an application that targets the transmission of 1Mb/s of data over a 30khz wireless channel. Such a target can be achieved using a transmission rate of 24.3 ksymbol/sec, 16-QAM modulation, $L_T = 32$, $L_P = 100$, and $M = N = 14$ antennas. The resulting relative computational complexities are given in the table below.

	Flops/burst	MegaFlops/s	%
channel estimation	7,840	1.44	0.65
nulling and ordering	1,036,000	190.8	86.3
payload processing	156,800	28.9	13.1
TOTAL	1,200,000	221.2	100

The dominant portion, nearly %90, of the computation involves determining the nulling vectors and optimal ordering. The question that begs itself, therefore, is whether this computation be reduced in a numerically stable way?

2.4. Objectives

In coming up with an alternative algorithm the following objectives appear to be natural:

- *Cost efficiency*: Is it possible to find $H_\alpha^{(M-1)\dagger}$ and $P^{(M-1)}$ from $H_\alpha^\dagger$ and P, without having to "re-solve" the reduced-order problem all over again?
- *Numerical stability and robustness*:
  - Avoid "squaring" things (forming H^*H, for example, is undesirable).
  - Avoid "inverting" things (inverting H^*H to obtain P is undesirable).
  - Make as much use as possible of *unitary* transformations.

In what follows, we will propose a square-root implementation of the algorithm that meets the above goals.

3. A SQUARE-ROOT ALGORITHM

In order to avoid squaring H, let us begin with the QR decomposition of the augmented channel matrix

$$\begin{bmatrix} H \\ \sqrt{\alpha} I_M \end{bmatrix} = QR = \begin{bmatrix} Q_\alpha \\ Q_2 \end{bmatrix} R, \tag{5}$$

where Q is an $(N + M) \times M$ matrix with orthonormal columns, and R is $M \times M$ and nonsingular. It is not hard to see that

$$P^{1/2} = R^{-1} \quad \text{and} \quad H_\alpha^\dagger = P^{1/2}Q_\alpha^* \tag{6}$$

where $P^{1/2}P^{*/2} = P$. Thus, given $P^{1/2}$ and Q_α, we can compute both the pseudoinverse and the error covariance matrix. However, before addressing the question of how best to compute $P^{1/2}$ and Q_α, let us focus on:

1. How to find the smallest diagonal entry of P?
2. How to find the square-root factor of $P^{(M-1)}$ from $P^{1/2}$?
3. How to find the nulling vectors?

The answers follow.

Claim 1: *The diagonal entries of P are simply the squared lengths of the rows of $P^{1/2}$.*

Claim 2: *Reorder the entries of s so that the M-th diagonal entry of P is the smallest. Consider any unitary transformation Σ that rotates (or reflects) the M-th row of $P^{1/2}$ to lie along the direction of the M-th unit vector. In other words,*

$$P^{1/2}\Sigma = \begin{bmatrix} P^{(M-1)/2} & P_M^{(M-1)/2} \\ 0 & p_M^{1/2} \end{bmatrix}, \tag{7}$$

where $p_M^{1/2}$ is a scalar. Then $P^{(M-1)/2}$ is a square-root of $P^{(M-1)}$.

Claim 3: *Suppose that we have repeated the steps of the above two claims until $P^{1/2}$ is transformed to an upper triangular matrix. Moreover, let $\underline{q}_{\alpha,i}$, $i = 1, \ldots, M$ denote the resulting columns of Q_α, i.e.,*

$$Q_\alpha = \begin{bmatrix} \underline{q}_{\alpha,1} & \cdots & \underline{q}_{\alpha,M} \end{bmatrix}.$$

Then the nulling vectors for the signals s_1 to s_M are given by:

$$H_{\alpha,i}^\dagger = p_i^{1/2}\underline{q}_{\alpha,i}^*, \tag{8}$$

where $p_i^{1/2}$ denotes the i-th diagonal entry of $P^{1/2}$.

In conclusion, once $P^{1/2}$ and Q_α are computed, there is no need to recompute them for the deflated channel matrix $H^{(M-1)}$. All the information we need is already in $P^{1/2}$ and Q_α.

3.1. Computing $P^{1/2}$ and Q_α

One possible way to compute $P^{1/2}$ and Q_α is via the QR decomposition (5). However, this is undesirable since it requires us to invert R to obtain $P^{1/2}$. To avoid inversions, we introduce the following algorithm which can be regarded as an extension of the traditional square-root filtering algorithm to the problem at hand (see, e.g., [3]).

Claim 4: *$P^{1/2}$ and Q_α can be computed using the following recursion, initialized with $P^{1/2}_{|0} = \frac{1}{\sqrt{\alpha}} I_M$ and $Q_0 = 0_{N\times M}$,*

$$\begin{bmatrix} 1 & H_i P^{1/2}_{|i-1} \\ 0 & P^{1/2}_{|i-1} \\ -e_i & Q_{i-1} \end{bmatrix} \Theta_i = \begin{bmatrix} r^{1/2}_{e,i} & 0 \\ \bar{K}_{p,i} & P^{1/2}_{|i} \\ A_i & Q_i \end{bmatrix}, \qquad (9)$$

where e_i is the i-th unit vector of dimension N (i.e., it is an $N \times 1$ vector of all zeros except for the i-th entry, which is unity), and Θ_i is any unitary transformation that transforms the first row of the pre-array to lie along the direction of the first unit row vector. After N steps the algorithm yields the desired quantities via:

$$P^{1/2} = P^{1/2}_{|N} \quad \text{and} \quad Q_\alpha = Q_N. \qquad (10)$$

3.2. Description of Algorithm

We can now summarize the algorithm.

1. Compute $P^{1/2}$ and Q_α by propagating the square-root algorithm of Claim 4.
2. Find the minimum length row of $P^{1/2}$ and permute it to be the last (Mth) row. Permute s accordingly.
3. Find a unitary Σ such that $P^{1/2}\Sigma$ is block upper triangular:
$$P^{1/2}\Sigma = \begin{bmatrix} P^{(M-1)/2} & P^{(M-1)/2}_M \\ 0 & p^{1/2}_M \end{bmatrix}.$$
4. Update Q to $Q\Sigma$.
5. The nuling vector for the M-th signal is given by $p^{1/2}_M \underline{q}^*_M$, where $\underline{q}_M$ is the M-th row of Q.
6. Go back to step 3, but now with $P^{(M-1)/2}$ and $Q^{(M-1)}$ (the first $M-1$ columns of Q).

3.2.1. Remarks

The above algorithm satisfies our objectives of cost-efficiency:

- we have avoided computing the pseudo-inverse (or QR decomposition) for each deflated subchannel matrix. The resulting computational complexity can be shown to be
$$\frac{2}{3}M^3 + 7NM^2 + 2N^2M.$$
When $M = N$, the computational complexity is thus reduced from $27M^4/4$ to $29M^3/3$, i.e., roughly by a factor of $0.7M$.

and numerical stability and robustness:

- we have avoided squaring any of the quantities.
- we have avoided computing inverses (and even scalar divisions) altogether.
- we have used unitary transformations as much as possible.

Returning to our target application (1 Mbit/s, 30 kHz channel, 24.3 ksymbol/sec, 16-QAM, $M = N = 14$, $L_T = 32$, $L_P = 100$), the resulting relative computational complexities are given in the table below.

	Flops/burst	MegaFlops/s	%
channel estimation	7,840	1.44	2.9
nulling and ordering	106,100	19.5	39.1
payload processing	156,800	28.9	58.0
TOTAL	270,400	49.8	100

Thus the computation for the nulling vector and optimal ordering step has been reduced from 190 MFlops/s to 19 MFlops/s, and the total computation has been reduced from 221 MFlops/s to 50 MFlops/s.

4. CONCLUSION AND FINAL REMARKS

In this paper we developed an efficient square-root algorithm for BLAST which offers an order of magnitude of savings in the computational complexity compared to earlier methods.

The prominent component of the algorithm is the use of unitary transformations (the Θ_i and the Σ) which introduce zeros in prescribed entries of given row vectors. These can be performed by either using a Householder reflection, or a sequence of Givens rotations (see, e.g., [2, 3]. In hardware, the sequence of Givens rotations can be implemented using division-free methods, such as the CORDIC method. They can also by parallelized by means of a systolic-array-type architecture. Moreover, the algorithm can be generalized to take account of updates to the the channel matrix. This could result in changes to the optimal ordering for estimating the signals.

Finally, the savings in computational complexity, as well as the numerical stability, of the algorithm suggest that it may be possible to implement the algorithm using a single commercial fixed-point DSP processor.

REFERENCES

[1] G.J. Foschini and M.J. Gans. On limits of wireless communications in a fading environment when using multiple antennas. *Wireless Personal Communications*, 6(3):311, 1998.

[2] G.H. Golub and C.F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, MD, 3rd edition, 1996.

[3] T. Kailath, A.H. Sayed, and B. Hassibi. *Linear Estimation.* Prentice-Hall, Englewood Cliffs, NJ, 1999.