ITERATIVE MAXIMUM LIKELIHOOD DECODING OF
GENERALIZED SPACE-TIME BLOCK CODES
A. Lee Swindlehurst
Dept. of Elec. & Comp. Engineering, Brigham Young University, Provo, UT, USA, 84602
e-mail: swindle@ee.byu.edu

ABSTRACT
An iterative algorithm for ML detection of a certain class of space-time codes is presented. Codes within this general framework employ redundant linear precoding, and include as special cases many recently proposed algorithms. A number of codes within this framework possess a special property that allows for decoupled ML symbol detection, which greatly reduces complexity. This property is exploited to derive an iterative minimizer of the ML criterion that alternates between estimating the transmitted data in one step, and the interference statistics and channel in the next. Training data is used to initialize the algorithm.

INTRODUCTION
The advantages of using multiple antennas at both the transmit and receive ends of a wireless communications link have recently been noted. A number of space-time codes have been proposed that exploit the potential for increased throughput and diversity that such systems offer. As shown in this paper, many of these codes can be unified within a general framework based on redundant linear precoding, where different linearly transformed versions of the same data sequence are broadcast from each transmit antenna. This framework is general enough to include the coding schemes described in [1]-[9] as special cases, among others. Codes within the proposed framework are referred to as Generalized Space-Time Block Codes (GSTBCs).

In this paper, the special linear structure of GSTBCs is exploited for efficient Maximum Likelihood (ML) detection and estimation of the transmitted data sequence, the (flat-fading) channel coefficients, and the covariance matrix of the noise and interference. Sufficient training data is assumed to be available to obtain an initial estimate of the channel and the noise statistics, after which the algorithm iterates between estimating these parameters and detecting the symbols in the transmitted block of data. In [10], it was shown that for Alamouti-type codes [1, 5]), the symbol detection step can be decoupled into a series of scalar decisions, which greatly reduces algorithm complexity. In this paper, it is shown that this simplification results for many other codes within the GSTBC framework.

GENERALIZED SPACE-TIME BLOCK CODES
Assume a single-user transmit array with \( K > 1 \) elements, a receive array with \( M \) elements, and a flat-fading channel. Assuming \( N_x = N_t + N \) symbol-rate samples are taken from the array, the following model results:

\[
\mathbf{X} = \mathbf{H}[\mathbf{S}_t \mathbf{S}(\mathbf{u})] + \mathbf{N},
\]

where \( \mathbf{X} \) is an \( M \times N_x \) matrix of received data, \( \mathbf{H} \) is the \( M \times K \) channel matrix, \( \mathbf{N} \) is additive noise, \( \mathbf{S}_t \) is \( K \times N_t \) and contains training data, and \( \mathbf{S}(\mathbf{u}) \) is \( K \times N \) and depends on an \( N_u \) element vector of unknown symbols \( \mathbf{u} \). To describe the dependence of \( \mathbf{S}(\mathbf{u}) \) on \( \mathbf{u} \), let the \( N \times 1 \) vector \( \mathbf{s}_k(\mathbf{u}) \) represent the data transmitted from antenna \( k \), with real and imaginary parts \( \mathbf{s}_{k,r}(\mathbf{u}) \), \( \mathbf{s}_{k,i}(\mathbf{u}) \), respectively. Defining \( \tilde{\mathbf{s}}_k^T(\mathbf{u}) = [\mathbf{s}_{k,r}(\mathbf{u}) \quad \mathbf{s}_{k,i}(\mathbf{u})]^T \), \( \mathbf{S}(\mathbf{u}) \) is decomposed as

\[
\begin{bmatrix}
\text{real } \{ \mathbf{S}^T(\mathbf{u}) \} \\
\text{imag } \{ \mathbf{S}^T(\mathbf{u}) \}
\end{bmatrix} =
\begin{bmatrix}
\mathbf{s}_{1,r}(\mathbf{u}) & \cdots & \mathbf{s}_{K,r}(\mathbf{u}) \\
\mathbf{s}_{1,i}(\mathbf{u}) & \cdots & \mathbf{s}_{K,i}(\mathbf{u})
\end{bmatrix} =
\begin{bmatrix}
\tilde{\mathbf{s}}_1(\mathbf{u}) & \cdots & \tilde{\mathbf{s}}_K(\mathbf{u})
\end{bmatrix}.
\]

Each column of this matrix is assumed to be encoded in the following way:

\[
\tilde{\mathbf{s}}_k(\mathbf{u}) = \mathcal{U}_k \tilde{\mathbf{u}}, \quad k = 1, \cdots, \begin{bmatrix}
\text{real}(\mathbf{u}) \\
\text{imag}(\mathbf{u})
\end{bmatrix},
\]

This work was supported by the U. S. National Science Foundation under Wireless Initiative Grant CCR-9979452 and Information Technology Research Grant CCR-0081476.
where $\mathcal{U}_k$ is a full-rank, real-valued, $2N \times 2N_u$ linear precoder. While splitting the data into real and imaginary parts provides the most general framework, it is convenient to use the more compact notation that results when $\mathcal{U}_k$ has the following form:

$$\mathcal{U}_k = \begin{bmatrix} \text{real}(\mathbf{U}_k) & -\text{imag}(\mathbf{U}_k) \\ \text{imag}(\mathbf{U}_k) & \text{real}(\mathbf{U}_k) \end{bmatrix}$$  \hspace{1cm} (3)$$

for a $N \times N_u$ complex matrix $\mathbf{U}_k$. Equation (2) then becomes $s_k(\mathbf{u}) = \mathbf{U}_k \mathbf{u}$. Signals that obey (2) are referred to Generalized Space-Time Block Codes (GSTBCs) [11]. The GSTBC framework described above is very general, and encompasses many types of popular codes:

**Example 1** – The $K=2$ STBC of [1] satisfies (2) with $N = N_u$, $\mathcal{U}_1 = I_{2N}$, and

$$\mathcal{U}_2 = \begin{bmatrix} \mathbf{I}_N \otimes \mathbf{J} & 0 \\ 0 & -\mathbf{I}_N \otimes \mathbf{J} \end{bmatrix}, \quad \mathbf{J} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}.$$  \hspace{1cm} (4)$$

where $\mathbf{I}_n$ indicates an $n \times n$ identity matrix. Similar transformations exist for the STBCs with larger values of $K$ described in [5].

**Example 2** – The code described in [8] combines the structure of the $K=2$ STBC together with the use of two $\frac{N}{2} \times \frac{N}{2}$ “sub-precoders” $\mathbf{C}_1$ and $\mathbf{C}_2$, where $N > N_u$. When cast in the framework of (2), this method results in

$$\mathcal{U}_1 = \begin{bmatrix} \text{real}(\mathbf{C}_1) & 0 & -\text{imag}(\mathbf{C}_1) & 0 \\ 0 & -\text{real}(\mathbf{C}_2) & 0 & \text{imag}(\mathbf{C}_2) \\ \text{imag}(\mathbf{C}_1) & 0 & \text{real}(\mathbf{C}_1) & 0 \\ 0 & \text{imag}(\mathbf{C}_2) & 0 & \text{real}(\mathbf{C}_2) \end{bmatrix}, \quad \mathcal{U}_2 = \begin{bmatrix} \mathbf{J} \otimes \mathbf{I}_\frac{N}{2} & 0 \\ 0 & -\mathbf{J} \otimes \mathbf{I}_\frac{N}{2} \end{bmatrix} \mathcal{U}_1,$$

where $\mathbf{J}$ is defined in (4).

**Example 3** – The so-called linear dispersion (LD) codes of [9] satisfy (2) as well. For these codes,

$$\mathbf{S}^T = \sum_{q=1}^{N_u} \alpha_q \mathbf{A}_q + j \beta_q \mathbf{B}_q, \hspace{1cm} (5)$$

where $\alpha_q$ and $\beta_q$ are real scalars, and $\mathbf{A}_q, \mathbf{B}_q$ are $N \times K$ complex matrices. The LD approach is equivalent to (2) with the definitions

$$\mathcal{U}_k = \begin{bmatrix} \text{real}(\mathbf{A}_k) & -\text{imag}(\mathbf{B}_k) \\ \text{imag}(\mathbf{A}_k) & \text{real}(\mathbf{B}_k) \end{bmatrix}, \quad \mathbf{A}_k = [\mathbf{A}_1(:,k) \cdots \mathbf{A}_{N_u}(:,k)] \quad \mathbf{B}_k = [\mathbf{B}_1(:,k) \cdots \mathbf{B}_{N_u}(:,k)],$$

where $(:,k)$ denotes the $k^{th}$ column of the associated matrix.

**Example 4** – For the code described in [3], $\mathbf{S}$ is circulant and Hankel, so (3) applies with $N = N_u$, and

$$\mathbf{U}_k = \begin{bmatrix} 0 & \mathbf{I}_{N-1} \\ 1 & 0 \end{bmatrix}^{k-1}.$$  

**Example 5** – For the modulation-induced code of [4], $\mathbf{U}_k$ is $N \times N$ and chosen to be diagonal.

### Iterative Maximum Likelihood Decoding

Assume that the columns of the noise matrix $\mathbf{N}$ can be modeled as independent circular complex Gaussian random vectors with (spatial) covariance $\mathbf{Q}$. Under this assumption, the log-likelihood of $\mathbf{X}$ conditioned on $\mathbf{H}, \mathbf{Q}$, and $\mathbf{u}$ is given by

$$L = -N_x \log \det(\mathbf{Q}) - \text{Tr}(\mathbf{Q}^{-1}(\mathbf{X} - \mathbf{H}\mathbf{S})(\mathbf{X} - \mathbf{H}\mathbf{S})^*) \hspace{1cm} (6)$$

where $\mathbf{S} = [\mathbf{S}_t \mathbf{S}(\mathbf{u})]$ and $(\cdot)^*$ denotes the complex conjugate transpose. If $\mathbf{u}$ were known, the likelihood function could easily be maximized with respect to $\mathbf{H}$ and $\mathbf{Q}$ to obtain

$$\hat{\mathbf{H}} = \mathbf{X}\mathbf{S}^*(\mathbf{S}\mathbf{S}^*)^{-1} \quad \hat{\mathbf{Q}} = \frac{1}{N_x} \mathbf{X}\mathbf{P}^\dagger \mathbf{S}^* \mathbf{X}^*,$$  \hspace{1cm} (7)
where $P_{S'}^{\perp} = I - S^* (S S^*)^{-1} S$. On the other hand, if it is assumed that $H$ and $Q$ are known, and that each symbol in the vector $u$ is drawn from a $c$-element finite alphabet $A$, then $u$ is estimated from

$$
\hat{u} = \arg \min_{u \in \mathbb{A}^c} \text{Tr}(Q^{-1} (X_u - HS(u)) (X_u - HS(u))^*)
$$

(8)

where $X_u$ is formed from the last $N_u$ columns of $X$.

For large $N_u$, direct minimization of (8) over all possible $c^{N_u}$ outcomes is prohibitively expensive. However, due to the linear structure of the space-time codes assumed, a considerable savings is achieved if the $S(u)S^*(u)$ term in (8) is independent of $u$. It was shown in [10] that this is true for the Alamouti code [1] and its generalizations [5], but it also holds for other codes from the GSTBC family as well, including those described in [6, 7, 8] (with unitary sub-precoders) and [4]. It is also approximately true for other codes such as [3] where $u$ results in a matrix $S(u)$ with nearly orthogonal rows. If $S(u)S^*(u)$ is independent of $u$, then (8) can be decoupled into $N_u$ scalar detection problems of the form:

$$
\hat{u}(n) = \arg \max_{u(n) \in \mathbb{A}} u_r(n)z(n) + u_i(n)z(n + N)
$$

(9)

where $u(n)$ is the $n$th element of $u$, $u_r(n)$ and $u_i(n)$ are its real and imaginary parts, and the vector $z$ is defined by

$$
z = \text{Re} \left\{ \sum_{k=1}^{K} H_{k}^T \left[ \begin{array}{c} I_{N_u} \\ jI_{N_u} \end{array} \right] X_u^* Q^{-1} h_k \right\}
$$

(10)

where $h_k$ is the $k$th column of $H$.

The derivations above suggest the following iterative algorithm for obtaining the maximum likelihood estimates of $H$, $Q$, and $u$:

1. Obtain an initial estimate of $H$ and $Q$, using either estimates from the previous block of data, or by exploiting the training data present in the current frame (i.e., substitute $S_t$ for $S$ in (7)).
2. Perform the decoupled symbol detection of (9) with $H$ and $Q$ replaced by their most recent estimates.
3. Use the detected symbols $\hat{u}$ from step 2 to form $\hat{S} = [S_t \ S(\hat{u})]$, and replace $S$ with $\hat{S}$ in (7) to update the channel and noise covariance estimates.
4. Repeat steps 2 and 3 until there is no change in the symbol decisions, or until some pre-determined stopping criterion is reached.

As with other alternating-projection approaches, the likelihood function is non-decreasing with each step, and the algorithm is guaranteed to converge to at least a local maximum of (6).

A SIMULATION EXAMPLE

This example considers a case with one user, $M = 3$ receive antennas, $K = 2$ transmit antennas, a block of $N = N_u = 40$ transmitted data symbols, and $N_t = 5$ training symbols ($N_e = 45$). The diagonal space-time code of [4] was implemented, with the diagonal elements of $U_1$ and $U_2$ chosen randomly from the unit circle. Unit-amplitude QPSK symbols were generated for both the training and unknown data, and the elements of the channel and noise matrices were zero-mean, circular complex Gaussian random variables, with variances chosen to achieve the desired SNR. The noise was chosen to be spatially white, although this information was not used by the algorithm. The SNR is defined by $\sigma_n^2 / \sigma_e^2$, where $\sigma_e^2$ and $\sigma_n^2$ are the variances of the elements of $H$ and $N$ respectively. In each trial, a new random $H, N$, and $u$ are generated, and used to create the data matrix $X$. Figure 1 shows the symbol error rate (SER) versus SNR achieved by the proposed algorithm using only the training data, and then for the first three ML iterations (further iterations did not improve performance). Essentially all of the improvement comes after only two iterations, and results in a reduction of $7$ dB in the SNR required to achieve a nominal SER of $10^{-3}$.

REFERENCES

Figure 1: Performance of Diagonal Coding versus SNR for First Three Algorithm Iterations.


