

# Low Complexity Signal Processing and Optimal Joint Detection for Over-Saturated Multiple Access Communications

*A report of the Laboratory of Information and Decision Systems: LIDS-P-2315*

Rachel E. Learned\* Alan S. Willsky\* Don M. Boroson\*\*

Key Words: Multiuser detection, interference rejection, EDICS: SP2.8.1

This paper has also been submitted to the *IEEE Transactions on Signal Processing*, Special Issue on Communications.

December 29, 1995

## Abstract

This paper addresses the problem of uncoded multiple access (MA) joint detection for the case in which user signatures are linearly dependent. The linearly dependent scenario would occur when the number of users in a communication system is increased beyond the dimension of the signal space available for transmission. This “over-saturating” of the signal space can, in principle, be accomplished with minimal impact on system performance, assuming that *optimal* detection can be implemented. The optimal detector for the general over-saturated case has a complexity which is exponential in the number of users. In order to find a joint detector for the MA communication of  $K > N$  users in  $N$  dimensional signal space, we impose a hierarchical cross-correlation structure on the user signature waveforms. This paper develops a tree processing procedure which takes advantage of this structure to give the *optimal* estimate with an extremely low computational complexity. We show this complexity to be (in typical cases) a low-order-polynomial in the number of users. This is an enormous savings in computations over the  $O(2^K)$  computations needed if the signatures did not exhibit any structure.

\* Massachusetts Institute of Technology, email: learned@lids.mit.edu, fax:(617)-981-7687, phone:(617)-253-3816

\*\* MIT Lincoln Laboratory

The work of Rachel Learned and Don Boroson has been sponsored in part by the Department of the Air Force under contract number F19628-95-C-0002. The work of Alan Willsky and Rachel Learned has been supported in part by the Air Force Office of Scientific Research under grant number F49620-95-1-0083 and the Army Research Office under grant number DAAL03-92-G-0115.

The authors wish to thank Professor Sergio Verdú for his most helpful discussion during the early stages of our hierarchical algorithm investigation. We are also thankful to Professor Mitchell Trott for his input on Section 3.2.

# 1 Introduction

Multiple access (MA) communication represents an active area of current research since it is the only means of communication among users in wireless systems such as mobile and cellular terrestrial systems and satellite-based systems. In each of these applications the possibility of many users sharing the available communication channel offers obvious advantages in terms of flexible and cost-efficient use of the channel. In addition, MA also poses a number of challenging research problems including many that fall within the domain of signal processing. This paper investigates one of those challenges, namely, the problem of optimal detection in uncoded MA communications.

The importance and difficulty of the problem of detection in an uncoded MA system has been recognized for some time ([1, 2, 3, 4]). In particular, consider a pulse-amplitude-modulated (PAM) communication system in which each user transmits a distinct waveform, the amplitude of which is modulated by a weight corresponding to the information to be communicated.<sup>1</sup> If there is only one user transmitting through an additive white Gaussian noise channel, optimal detection at the receiver is realized by a simple matched filter followed by a quantization to the closest weight used in transmission [5]. If, however, many users were to transmit through the channel the situation can become far more complex.

## *The Problem*

Time-bandwidth restrictions on any communication system limit the dimension,  $N$ , of the space of possible user waveforms. Adopting the commonly-used vector space framework we identify the  $N$ -dimensional signal space with  $\mathbb{R}^N$  and can then state the multiuser joint detection problem as follows: for a given set of user waveforms represented in signal space by the set of signal vectors,  $\{\mathbf{s}_k\}_1^K$ ,  $\mathbf{s}_k \in \mathbb{R}^N$ , the general uncoded detection problem is to compute an estimate of weights,  $\mathbf{b}$ , from an observation  $\mathbf{r} \in \mathbb{R}^N$

$$\mathbf{r} = \sum_{k=1}^K b_k \mathbf{s}_k + \sigma \mathbf{n} = \mathbf{S} \mathbf{b} + \sigma \mathbf{n}, \quad (1)$$

where

- $K$  is the number of users.
- $\mathbf{b} \in \{[b_1 \cdots b_K]^T \mid b_i \in P_i\}$ , where  $P_i$  is some finite set of real amplitudes and  $b_i$  is iid uniform. For  $P_i$  having  $M$  elements, this is  $M$ -ary PAM.
- $\mathbf{S} = [\mathbf{s}_1, \dots, \mathbf{s}_K]$  is an  $N \times K$  matrix whose columns are user signal vectors as seen at the receiver.
- $\mathbf{n}$  is a real Gaussian vector of mean zero and identity covariance.
- $\sigma$  is the noise standard deviation.

One case in which detection is simple is where the user waveforms are orthogonal. In this case, once again, a matched filter followed by a quantization to the closest weight used in transmission is optimal for each user.<sup>2</sup> The restriction to orthogonal signal sets, however, is often not a satisfactory one for several scenarios.

A first reason for the investigation of MA with non-orthogonal signal sets is motivated by signal variations induced by the channel. For example, in typical terrestrial MA systems the communication channel can introduce signal distortion such as multipath. Clearly, in such a situation there is a need to estimate the time-varying channel. For this paper, we assume that the channel is capable of

---

<sup>1</sup>In binary communications this weight takes on one of two values. Among the most popular methods for binary PAM is binary phase-shift-keying (BPSK) in which the weights are  $\{+1, -1\}$ . For general  $M$ -ary PAM, however,  $M$  possible values are allowed for these weights.

<sup>2</sup>Forcing user transmissions to be orthogonal or nearly orthogonal, even at the expense of inserting wasteful buffer zones in which no user is permitted to transmit, is common practice in systems of present.

being estimated and is, hence, known. Channel distortion, in general, does not allow (either through signal design or through equalization) for the received signal to be written as a weighted sum of orthogonal waveforms plus white Gaussian noise.

A second reason for investigating non-orthogonal signal sets is the difficulty of achieving accurate synchronization among users. Often, the orthogonality of a set of waveforms is a result of careful timing alignment (such as in TDMA). If the relative time shifts in the transmitted signals are not controlled, orthogonality cannot be maintained.

Finally, the assumption of orthogonality among user signals must be abandoned if we are to offer service to more users than orthogonality would allow. In the absence of channel distortion (e.g. MA satellite systems employing narrow beam terrestrial antennas) and timing inadequacies (e.g. systems employing a timing acquisition step), it is possible, in principle, to limit ourselves to orthogonal signal sets. Of course, this choice limits the number of users to the dimension of the signal space available for transmission. “Over-saturating” the signal space with users can, in principle, be accomplished with minimal impact on system performance, assuming that *optimal* detection can be implemented.<sup>3</sup> It is, therefore, desirable to increase the number of users beyond the orthogonal limit in order to enhance both system utilization and throughput. The success of such a system requires that the problem of optimal detection for non-orthogonal signal sets to be confronted.

The challenge, then, is to design optimal detectors for MA systems that employ non-orthogonal signal sets. As discussed in [2], the optimal joint detector for an arbitrary, non-orthogonal signal set has exponential complexity in the number of users,  $K$ . This is a catastrophic increase over the linear complexity of a bank of matched filters, one for each user. Surprisingly, though, the most common method currently used is a bank of matched filters where for each user the interference from all other users is considered as a second source of “noise”. With this type of detection, however, it is understood that the error rate will be higher than that obtained by the computationally complex optimal detector.

Indeed, as argued by Lupas and Verdù in [1], the performance loss of such an approach, as compared to the optimal, can be significant.<sup>4</sup> This has motivated several researchers ([1, 2, 3, 4]) to consider more complex, suboptimal detection algorithms that perform joint detection for all users; better performance than the simple matched filtering approach is achieved with complexity that is at most polynomial in the number of users. These methods, however, require the set of user signals to be linearly independent<sup>5</sup> and work best when the signal energy to noise energy ratio is very high for all users.

In contrast, this paper addresses the problem of finding an *optimal* joint detection algorithm for the case of  $K > N$  users in  $N$ -dimensional signal space that, like the suboptimal detectors, has complexity which is a low-order polynomial in the number of users. The key to devising such a detection algorithm is to choose the set of user waveforms so that an advantageous geometric structure is present.<sup>6</sup> In particular, the class of signal sets we consider has a hierarchical tree structure that allows for a rich variety of possibilities. For example, this desired tree structure is present in signals of considerable current interest in the signal processing literature such as wavelets and wavelet packets. In addition, in the communication literature, we find that Ross and Taylor ([6, 7]) have developed signal sets that fit  $K > N$  users in  $N$  dimensions while preserving the orthogonal minimum distance. The tree hierarchy is a by-product of their design.

In the next section the signal set structure of interest is described and illustrated and notation is

---

<sup>3</sup>Moreover, as can be seen in the work of Ross and Taylor [6, 7], it is, indeed, possible to design signal sets having more users than dimensions where the inter-decision-point distance resulting from use of this set is the same as that achieved by an orthogonal set. Their design constrains all users to have powers no higher than the users in the orthogonal set. Section 2.1 summarizes the results of Ross and Taylor in greater detail.

<sup>4</sup>In particular, for the “near-far” problem (large power variations among users) the conventional detector fails consistently.

<sup>5</sup>Although some of these detection algorithms may be applied in the linearly dependent case, they were not intended for the over-saturated problem and, therefore, give very poor performance.

<sup>6</sup>This implies that the transmitters have the flexibility to generate signals with the desired structure.

established. In Section 3 an overview of the hierarchical tree joint detector is given via an example and a formal derivation of the low complexity optimal detector is done along with a calculation of its computational complexity. Section 4 details the processing procedure of the tree detection algorithm and steps through a binary example. The paper is concluded in Section 5.

## 2 The Signal Sets

### 2.1 Signal Vector Set Structure

The geometric structure imposed on the signal vector set<sup>7</sup> is best described by saying that the set of signatures has *tree-structured cross-correlations*. Specifically,  $\mathbf{S}$  will have the desired structure if the signal vectors can be assigned to the nodes of a tree like the one shown in Figure 1. The tree pictorially conveys the following required relationships among user signal vectors.

- Each vector at a given level of the tree is orthogonal to all other vectors at that level.
- A signal vector is correlated only with its ancestor vectors (parent, grandparent, etc.) and its descendent vectors (children, grandchildren, etc.).

Both linearly dependent and linearly independent sets of signature vectors may be created to have tree-structured cross-correlations. The detector detailed in this paper finds the optimal solution for both cases.

The constraint of tree-structured cross-correlations, while very particular, actually allows a considerable amount of flexibility in designing user waveforms. Given a tree, we may construct a signal set that possesses the desired cross-correlation structure. Assume that waveforms at the bottom level of the tree comprise an orthogonal set. We obtain an orthogonal set at any level, i.e. the  $l^{th}$  level, by constructing a signal at each node at this level as a linear combination of the signals at its bottom-most descendent nodes. Since we have a tree with orthogonal signals assigned to the lowest level nodes, the sets of descendents for distinct nodes at the  $l^{th}$  level are disjoint, and consequently the signals created at level  $l$  are mutually orthogonal.

It follows that the general construction of a signal set with tree-structured cross-correlations requires (a) the specification of the tree structure (i.e. the number of levels,  $L$ , and the parent-child relations for all levels of the tree; (b) the specification of *any* orthogonal basis  $\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_N$ , of  $\mathbb{R}^N$  which we then assign to the  $N$  nodes on the finest scale of the tree;<sup>8</sup> (c) the specification of the weights for each of the linear combinations used to construct signals from their bottom-level descendents; and possibly (d) the deletion of signals at any of the nodes. This formulation allows for considerable flexibility in designing the signal set since *any* choices that satisfy (a)-(d) will lead to the desired geometric structure on the signal set. Note also that (d) provides us with the flexibility to capture linearly independent sets with the desired correlation.<sup>9</sup>

### 2.2 Some Examples of Signature Sets

As we have indicated, there are many different signal sets that can be constructed to have the tree structure we have just described. In this section we illustrate examples of two particular choices of signal vector sets, one of which involves signals of considerable current interest in the signal processing community, namely, wavelets and wavelet packets ([8, 9]) and one that was introduced in [6] directly in the context of designing signal vector sets for over-saturated MA systems.

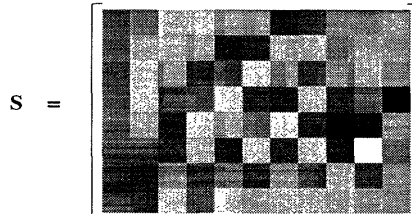
<sup>7</sup>For ease of notation, we use the abstract signal space representation of real signals, and, hence, all properties imposed on the signal vectors will also be true for the real waveform counterparts.

<sup>8</sup>Without loss of generality, we assume that the bottom level of the tree has exactly  $N$  nodes.

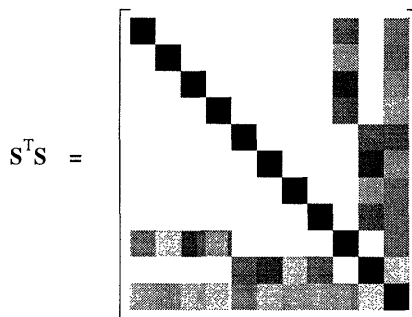
<sup>9</sup>For simplicity, however, (and since we wish to emphasize the applicability of our methods to the over-saturated case) we will assume that our tree is *full*, i.e. that there is a user signature at each node on the tree. The extension of our low complexity optimal detection scheme to the case in which there are fewer users is immediate.

### Wavelet Packet Sets

Wavelet and wavelet packet waveforms may be generated from a tree-structured procedure in which subspaces (generated by sets of orthogonal signals) are decomposed into Cartesian products of orthogonal lower-dimensional subspaces.<sup>10</sup> The result is a wavelet or wavelet packet dictionary consisting of an over-complete set of basis functions. A discrete wavelet packet dictionary offers a rich set of signal vectors from which to select many tree-structured sets. An example of a discrete wavelet packet signal set is shown below as an intensity matrix where each element of the matrix is shown as a pixel in the  $8 \times 11$  image. The values are shown in gray scale where the smallest is denoted by white and the largest is denoted by black.



In order to reveal the tree-structured cross-correlations, the absolute values of the elements of  $\mathbf{S}^T \mathbf{S}$  are displayed below, where 0 and 1 are denoted by white and black, respectively.



The wavelet packet signal vector set can be cast onto a tree with three levels as shown in Figure 2.

### Minimum Distance Sets

Another example is the minimum distance sets developed by Ross and Taylor in [7]. Ross and Taylor begin with  $N$  orthogonal users in  $N$  dimensions. The set of possible received points based on an  $M$ -ary PAM MA system with an orthogonal set of signal vectors has associated with it a minimum distance. That is, if  $\{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_N\}$  is the orthogonal signal set, then there is a specified minimum distance between any two points in the set  $\{\sum_{i=1}^N b_i \mathbf{s}_i \mid b_i \in P\}$ . Since the distance between the elements in this set are directly related to the probability that the optimal detector makes an error, maintaining a specified minimum distance is desirable. Ross and Taylor devise a method for adding additional, energy-constrained, linearly dependent users so that the minimum Euclidean distance between received points is preserved. We refer the reader to [7] for details of their construction.

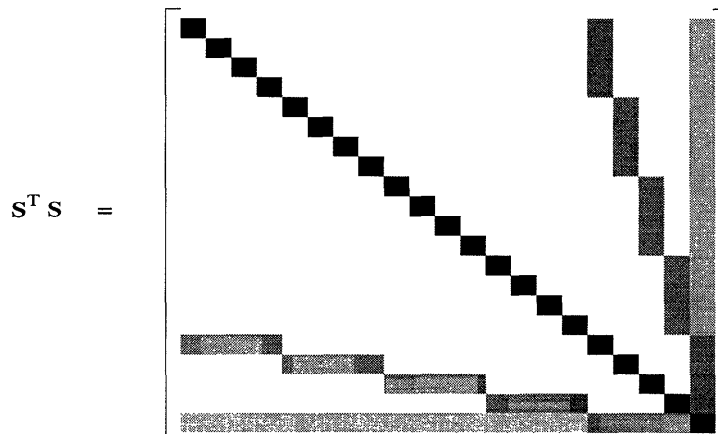
Ross and Taylor, for antipodal binary modulation,  $P = \{+1, -1\}$ , fit  $\frac{4}{3}N - \frac{1}{3}$  unit energy signal vectors into  $N$  dimensions where  $N$  must be a power of 4. A specific example detailed in [7] is briefly

<sup>10</sup>For a tutorial treatment of wavelet packets see the paper by Coifman and Wickerhauser [9].

repeated below.

$$\mathbf{S} = \begin{bmatrix} 1/2 & 0 & 0 & 0 & 1/4 \\ 1/2 & 0 & 0 & 0 & 1/4 \\ 1/2 & 0 & 0 & 0 & 1/4 \\ 1/2 & 0 & 0 & 0 & 1/4 \\ 0 & 1/2 & 0 & 0 & 1/4 \\ 0 & 1/2 & 0 & 0 & 1/4 \\ 0 & 1/2 & 0 & 0 & 1/4 \\ \mathbf{I}_{16} & 0 & 1/2 & 0 & 1/4 \\ 0 & 0 & 1/2 & 0 & 1/4 \\ 0 & 0 & 1/2 & 0 & 1/4 \\ 0 & 0 & 1/2 & 0 & 1/4 \\ 0 & 0 & 0 & 1/2 & 1/4 \\ 0 & 0 & 0 & 1/2 & 1/4 \\ 0 & 0 & 0 & 1/2 & 1/4 \\ 0 & 0 & 0 & 1/2 & 1/4 \\ 0 & 0 & 0 & 1/2 & 1/4 \end{bmatrix} \quad (2)$$

Here,  $\mathbf{I}_{16}$  is the 16 dimensional identity matrix. The correlation matrix,  $\mathbf{S}^T \mathbf{S}$ , is given below, again, as an intensity plot with 0 and 1 corresponding to white and black, respectively.



The structure of  $\mathbf{S}^T \mathbf{S}$  reveals that this minimum distance set of signature vectors may be cast onto a quad tree for which 4 children emanate from each parent node. The signature vector associated with the top of our correlation tree is the right-most column of  $\mathbf{S}$ . The first 16 columns would be associated with the bottom of the tree. These signature sets were designed in [6] for their minimum distance property. The tree hierarchy they possess is a by-product that, happily, can be exploited in the optimal detector we describe.

### 2.3 Notation

We introduce notation that will be used throughout the remainder of this paper.

- $n$ : node index
- $pn$ : index of the parent to node  $n$
- $p^m n$ : index of the ancestor to node  $n$  that is  $m$  levels above  $n$ .<sup>11</sup>

<sup>11</sup>Note that if node  $n$  is at level  $l$ , the index of the root node can be denoted by  $p^{l-1}n$ .

- $an = \{pn, p^2n, p^3n, \dots, p^{l-1}n\}$ : set of indices corresponding to the ancestor nodes of node  $n$ <sup>12</sup>
- $cn_i$ : node index for the  $i^{\text{th}}$  child of node  $n$
- $Kn$ : number of children of node  $n$
- $cn = \{cn_1, cn_2, \dots, cn_{Kn}\}$ : set of indices corresponding to children of node  $n$
- $dn = \{cn_1, dcn_1, cn_2, dcn_2, \dots, cn_{Kn}, dcn_{Kn}\}$ : set of indices corresponding to the descendent nodes of node  $n$ <sup>13</sup>
- $fn = \{n, dn\}$ : the family of indices associated with node  $n$ <sup>14</sup>

Note that the set of descendents for  $n$  at the lowest level of the tree is empty, i.e.  $dn = \emptyset$ , Likewise, the set of ancestors for the root node is empty, i.e.  $an = \emptyset$ .

Using the above tree index notation, the weight estimate and signature vector associated with a node,  $n$ , of the tree may be denoted by  $\hat{b}_n$  and  $\mathbf{s}_n$ , respectively. Collect the weight estimates and signature vectors of all ancestors of node  $n$  into a column vector,  $\hat{\mathbf{b}}_{an}$ , and corresponding signature matrix,  $\mathbf{S}_{an}$ , respectively. Here, the columns of  $\mathbf{S}_{an}$  are the signature vectors,  $\mathbf{s}_i, i \in an$ . Similarly defined are  $\hat{b}_{dn}$  and  $\mathbf{S}_{dn}$ .

For the derivation of the estimator in Section 4 we will require the inner products between the user signals

$$y_{i,j} = \mathbf{s}_i^T \mathbf{s}_j \quad (3)$$

Extending this definition, we establish the following:

- $y_{i,an} = \mathbf{s}_i^T \mathbf{S}_{an}$ : row vector of inner products between a signal and its ancestor signals
- $y_{i,dn} = \mathbf{s}_i^T \mathbf{S}_{dn}$ : row vector of inner products between a signal and its descendent signals
- $\mathbf{Y}_{an,dn} = \mathbf{S}_{an}^T \mathbf{S}_{dn}$ : matrix of inner products between ancestor and descendent signals
- $\mathbf{Y}_{dn,dn} = \mathbf{S}_{dn}^T \mathbf{S}_{dn}$ : matrix of inner products among the group of descendent signals

The first key signal processing operation in our algorithm will be the calculation of the following set of coefficients from the received signal  $\mathbf{r}$ :

$$l_i = \mathbf{s}_i^T \mathbf{r}, \quad i = 1, 2, \dots, K \quad (4)$$

The calculations of each  $l_i$  corresponds to processing the data  $\mathbf{r}$  through a filter matched to the signal  $\mathbf{s}_i$ .

Due to the manner in which we create our tree-structured signal set, both the inner products and the matched filter outputs for nodes above the bottom level can be easily calculated from the sets  $\{y_{i,i}\}_{i=1}^N$  and  $\{l_i\}_{i=1}^N$ , respectively. This will allow for an efficient calculation of these quantities.

If we were to adopt a tree-recursive construction for our tree-correlated sets we would realize further simplification in calculating the sets of  $y_{i,j}$  and  $l_i$ . A tree-recursive set requires the signal,  $\mathbf{s}_n$ , at a node  $n$  to be a linear combination of its children, only.<sup>15</sup> If we define  $\mathbf{S}_{cn}$  to be the signal vector matrix for the signals that lie at the children nodes of node  $n$ , we require the following signal set construction

$$\mathbf{s}_n = \mathbf{S}_{cn} \mathbf{h}_n = \sum_{i=1}^{Kn} h_{n,cn_i} \mathbf{s}_{cn_i} \quad (5)$$

where the elements of  $\mathbf{h}_n^T = [h_{n,cn_1} \ h_{n,cn_2} \ \dots \ h_{n,cn_{Kn}}]$  are known but arbitrary.

<sup>12</sup>This ordering of ancestors is important and will be useful later.

<sup>13</sup>Note that  $dn$  is recursively defined, where  $dcn_i$  is the set of descendents of node  $cn_i$ . In addition, the ordering of nodes into sub-tree groupings will be useful in later sections.

<sup>14</sup>Each set of family indices corresponds to an entire sub-tree having root node corresponding to node  $n$ .

<sup>15</sup>Note that both the minimum distance and the wavelet/wavelet packet sets can easily be chosen to exhibit this tree-recursive quality.

To illustrate a tree-recursive signal set, consider the signal set shown in Figure 3. This signal set comprises 7 users in 4 dimensions. The four signals,  $\{\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3, \mathbf{s}_4\}$ , at the bottom of the tree form an orthogonal basis for  $\mathbb{R}^4$ , and the three upper-level signals are as shown in the figure.

The inner products,  $y_{n,i}$ , of recursive tree-structured signals can be quickly computed from  $\mathbf{h}_n$  and  $\|\mathbf{s}_i\|^2 = y_{i,i}$  for  $i = 1, \dots, N$ . In particular, from Equation (5), since the signal  $\mathbf{s}_n$  at node  $n$  is expressed in terms of the signals at its children, we have  $y_{n,i} = \mathbf{h}^T \mathbf{S}_{cn}^T \mathbf{s}_i = \mathbf{h}_n^T \mathbf{y}_{cn,i}$ . It follows that the  $y_{i,j}$  can be calculated in a pyramidal fashion from the bottom of the tree to the top.

Due to a tree-recursive creation of the signal set, the matched filter outputs for nodes above the bottom level can also be calculated in a pyramidal fashion. Specifically,

$$l_n = \mathbf{h}_n^T \mathbf{S}_{cn}^T \mathbf{r} = \mathbf{h}_n^T \mathbf{l}_{cn} \quad (6)$$

where  $\mathbf{l}_{cn}$  is the column vector having elements  $l_k$ ,  $k \in cn$ . For our example of Figure 3, we calculate  $l_1, l_2, l_3$ , and  $l_4$  at the bottom level of the tree. Then, at the next level of the tree we compute

$$l_5 = h_{5,1}l_1 + h_{5,2}l_2, \quad l_6 = h_{6,3}l_3 + h_{6,4}l_4 \quad (7)$$

and at the top level of the tree

$$l_7 = h_{7,5}l_5 + h_{7,6}l_6. \quad (8)$$

The structure of these calculation is reminiscent of the structure of the calculations involved in computing wavelet or wavelet packet coefficients at a sequence of scales ([9]).<sup>16</sup> It follows that the computational complexity of determining the  $l_i$ 's is comparable to that of a wavelet transform: for an  $N$ -dimensional signal space with  $K$  users there are  $N$  matched filter outputs  $l_i = \mathbf{s}_i^T \mathbf{r}$ ,  $i = 1, 2, \dots, N$  to be calculated at the bottom level of the tree followed by, at most,  $(K - 1)$  additional multiplies and adds to compute  $l_i$  for the remainder of the nodes on the tree.

### 3 The Tree Joint Detection Algorithm

#### 3.1 Overview of the Detector

The optimum joint detector for the problem stated in Equation (1) chooses the weight vector estimate,  $\hat{\mathbf{b}}$ , according to the nearest neighbor or minimum distance rule.

$$\hat{\mathbf{b}} = \arg \min_{\mathbf{b} \in P^K} \|\mathbf{r} - \mathbf{S}\mathbf{b}\|^2 \quad (9)$$

For ease of discussion, we assume each user employs the same  $M$ -ary PAM for the remainder of this paper where  $b_i \in P$ ,  $\forall i$ ; this assumption is not essential to the operation of the tree algorithm. An MA system employing an arbitrary set of signal vectors,  $\mathbf{S}$ , can achieve the optimal detection of the above detector through an exhaustive search, i.e. the detector needs to perform  $M^K - 1$  comparisons to find the best estimate ([2]).

If the signal set has been constructed to have the tree cross-correlation structure described in Section 2.1, the optimum detector of Equation (9) can be achieved through a tree-structured algorithm that offers a huge reduction in the number of comparisons. In particular, because of this structure, a signature at a given node is correlated with *all* signatures at its ancestor and descendent nodes and is orthogonal to all other signatures on the tree. The weight estimate,  $\hat{b}_n$ , at a given node,  $n$ , therefore, will effect the estimates at descendent and ancestor nodes but will not effect the other estimates on the tree.

<sup>16</sup>In our case, however, we do not require the coefficients in the vectors  $\mathbf{h}_n$  to correspond to the filter coefficients used to construct orthogonal wavelets or filter banks. Moreover, we do not require a regular structure – e.g. different parent nodes at the same level of the tree may have a different coefficients in their  $\mathbf{h}_n$  vectors and may also have different numbers of children.



A simple example provides the basic idea. Consider the tree structure in Figure 2 and consider, first, the choice of the weight estimates for users 1 through 4 having signal vectors  $\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3, \mathbf{s}_4$ . These vectors are mutually orthogonal and are also orthogonal to  $\mathbf{s}_5, \mathbf{s}_6, \mathbf{s}_7, \mathbf{s}_8$  and  $\mathbf{s}_{10}$  but not to  $\mathbf{s}_9$  and  $\mathbf{s}_{11}$ . Since  $\mathbf{s}_5$ - $\mathbf{s}_8$  and  $\mathbf{s}_{10}$  are also correlated with  $\mathbf{s}_{11}$ , the decisions on weight estimates for these users are not decoupled from those for  $\mathbf{s}_1$ - $\mathbf{s}_4$ . We can, however, decouple these by looking, instead, at the *conditional* estimates. Specifically, for each possible pair of weight estimates for  $\mathbf{s}_9$  and  $\mathbf{s}_{11}$ , we can compute the optimal weight estimates for  $\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3, \mathbf{s}_4$  independently – i.e. the problem to be solved for each of these weight estimates is decoupled not only from the other three but also from the weight estimates corresponding to  $\mathbf{s}_5$ - $\mathbf{s}_8$  and  $\mathbf{s}_{10}$ . The result of this calculation for  $\mathbf{s}_1$ - $\mathbf{s}_4$  can then be thought of as producing a *conditional weight estimate table*, i.e. for each possible pair of choices for the weight estimates for  $\mathbf{s}_9$  and  $\mathbf{s}_{11}$  we now know the optimal weight estimates for  $\mathbf{s}_1$ - $\mathbf{s}_4$ . Similarly, for each pair of possible weight estimates for  $\mathbf{s}_{10}$  and  $\mathbf{s}_{11}$  we can compute the optimal estimates for  $\mathbf{s}_5$ - $\mathbf{s}_8$ . If we now consider  $\mathbf{s}_9$ , we can iterate this process: for each possible choice of weight value for its ancestor  $\mathbf{s}_{11}$  and with knowledge of the just-constructed conditional estimate table for its descendents  $\mathbf{s}_1$ - $\mathbf{s}_4$ , we can compute the optimal estimate for  $\mathbf{s}_9$  in a manner decoupled from the analogous computation for  $\mathbf{s}_{10}$ . This gives conditional estimate tables for  $\mathbf{s}_9$  and  $\mathbf{s}_{10}$  which then can be used to determine the optimal estimate for  $\mathbf{s}_{11}$  at the top of the tree. Conceptually, once we have this estimate it is a simple matter of successive table look-ups as we propagate down the tree to determine the optimal estimates first for  $\mathbf{s}_9$  and  $\mathbf{s}_{10}$  and then for their descendents.

As this simple example illustrates, the tree detection algorithm takes advantage of the tree structure and sweeps through the tree from bottom to top, creating a conditional weight estimate table at each node. The table of decisions at a given node is conditioned on weight decisions of the ancestors and is a function of weight decisions of the descendents. Since each conditional estimate table requires entries for each possible combination of weights at all ancestor nodes, the number of computations needed to create a table and the size of the table is exponential in the number of ancestors (since if there are  $l$  ancestors there are  $M^l$  possible sets of weight values for these ancestors). This complexity decreases exponentially as  $l$  decreases, i.e. as the algorithm moves from the bottom to the top of the tree the number of decisions made at each level decreases exponentially until there is only one decision associated with the top node of the tree. The full weight vector estimate for all user weights is a by-product of the last decision at the top of the tree.

While the complexity of the procedure as we have described it to this point is exponential in the number of levels in the tree (which bounds the number of ancestors of each node), the actual algorithm complexity is, in fact, extremely modest. If the tree were of uniform construction, i.e. if there are  $Q$  children emanating from each node, the number of levels of the tree is *logarithmic* in the number of users,  $K$ . Hence, the overall complexity is, then, bounded by a very low-order polynomial in  $K$ . This is discussed more fully in Section 3.3.

Moreover, while the derivation of the general algorithm given in the next section is most easily explained (and its optimality proved) in terms of conditional estimate tables, it is actually possible to use the structure of the signal sets to simplify the required on-line processing. The details of the calculation of the estimates are given in Section 4.

### 3.2 Derivation of Tree Detector

The global cost that must be minimized in Equation (9) is

$$F(\mathbf{r}, \mathbf{b}) = \|\mathbf{r} - \mathbf{S}\mathbf{b}\|^2. \quad (10)$$

In general,  $F(\mathbf{r}, \mathbf{b})$  is not separable by weight variables,  $b_i$ . Hence, the solution to Equation (9) is found by the calculation and comparison of  $F(\mathbf{r}, \mathbf{b}), \forall \mathbf{b} \in P^K$ .

The introduction of tree-structured cross-correlations transforms the structure of the cost function. We may reduce complexity of finding the smallest cost by making decisions in stages. The



For a system employing an  $L$ -level and  $M$ -ary PAM for all users, at any interior stage of the graph we must retain no more than  $M^L$  paths. Recall that the number of levels in the tree is likely to be a logarithmic function of the number of users, hence the number of paths to keep track of at any given stage of the dynamic program is a low order polynomial in the number of users.

### 3.3 Computational Complexity

The conceptual description of the tree detector in Section 3.1 and the dynamic programming description in Section 3.2 inevitably include many wasteful calculations and storage requirements. The discussion in Section 4 addresses the removal of this redundancy. Furthermore, it is expected that for each different tree structure and modulation a streamlined implementation is possible. Since we are concerned with the *order* of complexity of optimal detection for this paper, we calculate the complexity of the conceptually simplest, although admittedly inefficient, version of the tree detector. The resulting complexity of this “inefficient” version leads to an upper bound on tree detector complexity that is extremely low.

For simplicity of calculation of computational complexity, we restrict the tree to be of uniform composition in that there are exactly  $Q$  children emanating from each node. Recall that  $N$  is the number of signal space dimensions available (the number of nodes at the bottom of the tree) and  $M$  is the number of levels that can be modulated by each user. We measure complexity as the number of compares,  $c$ , needed to perform the tree algorithm;<sup>19</sup>  $c$  is stated below and derived later in this section.

$$c(N, Q, M) = \frac{M-1}{QM-1} NQM^{\log_Q N+1} - 1 \quad (15)$$

For example, if a system were to employ antipodal modulation,  $P = \{+1, -1\}$ ,  $M = 2$ , and signal sets having quad-tree ( $Q = 4$ ) structure such as the minimum distance waveform sets of Ross and Taylor, the number of comparisons needed for the tree detector estimate is

$$c(N, Q = 4, M = 2) = \frac{8N^{3/2} - 1}{7}. \quad (16)$$

The computational complexity is polynomial in the number of dimensions. The number of users,  $K$ , in this special case is  $K = \frac{4}{3}N - \frac{1}{3}$ , hence, the tree detector is also polynomial in the number of users, resulting in a computational complexity of  $O(K^{3/2})$ .

We now derive Equation (15) by counting the number of comparisons needed to execute the tree algorithm. Some facts used in the complexity calculation follow:

- Each node at level  $l$  has  $l - 1$  ancestor nodes.
- There are  $Q^{l-1}$  nodes at level  $l$  of the tree.
- The tree has a total of  $L$  levels (counting the top as level 1).
- There are  $N = Q^{L-1}$  nodes at the bottom of the tree, thus,  $L = \log_Q N + 1$ .

The algorithm creates a conditional bit estimate table for each node. For a given  $\hat{\mathbf{b}}_{an}$  we must choose the best of  $M$  possible values of  $\hat{b}_n$ . This requires  $M - 1$  comparisons for a single configuration of  $\hat{\mathbf{b}}_{an}$ . Since there are  $l - 1$  ancestors of node  $n$ , there are  $M^{l-1}$  possible configurations of  $\hat{\mathbf{b}}_{an}$ . The tree detector, therefore, creates a single table at level  $l$ , node  $n$ , with  $(M - 1)M^{l-1}$  comparisons. There are  $Q^{l-1}$  tables needed for level  $l$  of the tree and there are a total of  $\log_Q N + 1$  levels in the

---

<sup>19</sup>Counting the number of comparisons is equivalent to counting number of tentative decisions that must be made. Without computational optimization of the algorithm, each decision requires the computation of two metrics. Each metric requires several adds and subtracts. To find the order of the complexity of the tree algorithm, it is sufficient to count the number of compares.

tree. It follows that the total number of comparisons needed for the tree algorithm is

$$\begin{aligned}
c(N, Q, M) &= \sum_{l=1}^{\log_Q N+1} Q^{l-1} (M-1) M^{l-1} \\
&= (M-1) \frac{(QM)^{\log_Q N+1} - 1}{QM - 1} \\
&= \frac{(M-1)}{(QM-1)} (NQ M^{\log_Q N+1} - 1)
\end{aligned}$$

## 4 Signal Processing for the Optimal Tree Detector

### 4.1 Calculation of the Estimate

As was shown in Section 3.2, each weight estimate is related to the weight estimates corresponding to its ancestors and descendants. Mathematically, this dependence is revealed as the reduction of the general optimal estimator of Equation (9) to the tree-structured optimal estimator below. For each node,  $n$ , of the tree calculate the following estimate conditioned on the value of the set  $\mathbf{b}_{an}$

$$\hat{b}_n(\mathbf{r}|\mathbf{b}_{an}) = \arg \min_{b_n \in P} \|\mathbf{r} - \mathbf{s}_n b_n - \mathbf{S}_{an} \mathbf{b}_{an} - \mathbf{S}_{dn} \hat{\mathbf{b}}_{dn}(\mathbf{r}|b_n, \mathbf{b}_{an})\|^2, \quad (17)$$

where we need the vector of numbers  $\hat{\mathbf{b}}_{dn}(\mathbf{r}|b_n, \mathbf{b}_{an})$ . The set of estimates for all descendants of node  $n$  has already been calculated in the previous steps of the algorithm. Hence,  $\hat{\mathbf{b}}_{dn}(\mathbf{r}|b_n, \mathbf{b}_{an})$  is best defined recursively. For a given set of values for  $\{b_n, \mathbf{b}_{an}\}$ , we have

$$\hat{\mathbf{b}}_{dn}(\mathbf{r}|b_n, \mathbf{b}_{an}) = \begin{bmatrix} \hat{b}_{fc_{n_1}}(\mathbf{r}|b_n, \mathbf{b}_{an}) \\ \hat{b}_{fc_{n_2}}(\mathbf{r}|b_n, \mathbf{b}_{an}) \\ \vdots \\ \hat{b}_{fc_{n_{K_n}}}(\mathbf{r}|b_n, \mathbf{b}_{an}) \end{bmatrix}. \quad (18)$$

For that same set of  $\{b_n, \mathbf{b}_{an}\}$  we have already calculated  $\hat{b}_{cn_i}(\mathbf{r}|b_n, \mathbf{b}_{an})$ ; if the value found for  $\hat{b}_{cn_i}(\mathbf{r}|b_n, \mathbf{b}_{an})$  is  $\xi$ , then the sub-vector on the right hand side of Equation (18) are given by

$$\hat{b}_{fc_{n_i}}(\mathbf{r}|b_n, \mathbf{b}_{an}) = \begin{bmatrix} \hat{b}_{cn_i}(\mathbf{r}|b_n, \mathbf{b}_{an}) = \xi \\ \hat{\mathbf{b}}_{dc_{n_i}}(\mathbf{r}|b_{cn_i} = \xi, b_n, \mathbf{b}_{an}) \end{bmatrix}. \quad (19)$$

We examine the argument of the minimization in Equation (17) more closely. We can, of course, remove any terms that do not depend on  $b_n$  and we can multiply by any positive constant.<sup>20</sup> As a result, some algebra shows that (17) is equivalent to

$$\hat{b}_n(\mathbf{r}|\mathbf{b}_{an}) = \arg \max_{b_n \in P} J(b_n|\mathbf{r}, \mathbf{b}_{an}) \quad (20)$$

where

$$J(b_n|\mathbf{r}, \mathbf{b}_{an}) = [l_n b_n - \frac{1}{2} b_n^2 y_{n,n}] - b_n \mathbf{y}_{n,an} \mathbf{b}_{an} \quad (21)$$

$$+ [\mathbf{l}_{dn}^T \hat{\mathbf{b}}_{dn}(\mathbf{r}|b_n, \mathbf{b}_{an}) - \frac{1}{2} \hat{\mathbf{b}}_{dn}^T(\mathbf{r}|b_n, \mathbf{b}_{an}) \mathbf{Y}_{dn,dn} \hat{\mathbf{b}}_{dn}(\mathbf{r}|b_n, \mathbf{b}_{an})] \quad (22)$$

$$- b_n \mathbf{y}_{n,dn} \hat{\mathbf{b}}_{dn}(\mathbf{r}|b_n, \mathbf{b}_{an}) \quad (23)$$

$$- \mathbf{b}_{an}^T \mathbf{Y}_{an,dn} \hat{\mathbf{b}}_{dn}(\mathbf{r}|b_n, \mathbf{b}_{an}) \quad (24)$$

<sup>20</sup>or multiply by a negative constant and replace minimization by maximization

From this, we see that the only explicit processing of the data  $\mathbf{r}$  is for the terms  $l_n = \mathbf{s}_n^T \mathbf{r}$  and  $\mathbf{l}_{dn} = \mathbf{S}_{dn}^T \mathbf{r}$  on lines (21) and (22).

The first bracketed term  $[l_n b_n - \frac{1}{2} b_n^2 y_{n,n}]$  on line (21) corresponds exactly to the decision statistic that would be used to choose  $b_n$  if there were no other users to consider or if all other users had orthogonal signals. The remaining terms, then, represent the *adjustments* of this decision statistic to reflect the impact of the non-orthogonality in the signals. The last term on line (21),  $b_n \mathbf{y}_{n,an} \mathbf{b}_{an}$ , represents the interaction between the choice of  $b_n$  and the particular hypothesized choices for the ancestor weights. Note that since we are hypothesizing the values of  $\mathbf{b}_{an}$  this term can be precomputed. Line (22) represents a counterpart to the bracketed term on line (21). Specifically, if all users other than the ones corresponding to the weights  $\mathbf{b}_{dn}$  were not present, then

$$[\mathbf{l}_{dn}^T \mathbf{b}_{dn} - \frac{1}{2} \mathbf{b}_{dn}^T \mathbf{Y}_{dn,dn} \mathbf{b}_{dn}] \quad (25)$$

represents the decision statistic that would be used to determine the optimal choice of  $\mathbf{b}_{dn}$ . Since this *not* the case, this term incorporates the decisions  $\hat{\mathbf{b}}_{dn}(\mathbf{r}|b_n, \mathbf{b}_{an})$  conditioned on the value of  $\{b_n, \mathbf{b}_{an}\}$ . Line (23) accounts for the interactions between these descendant decisions and the possible decisions  $b_n$ . Likewise, line (24) accounts for the interactions between the descendant decisions and the hypothesized decisions  $\mathbf{b}_{an}$  at the ancestors of node  $n$ . Thus, all of the quantities needed in the last three lines (22–24) can be computed based on the value of  $b_n$  and the calculations that have already been performed at lower levels on the tree.

## 4.2 The Binary Conditional Decision Rule

We focus on the binary antipodal signaling case, i.e. when  $P = \{+1, -1\}$ . For each choice of  $\mathbf{b}_{an}$  there is only one comparison to make for the minimization of Equation (20)<sup>21</sup>. The solution to (20) can, therefore, be expressed as<sup>22</sup>

$$\hat{b}_n(\mathbf{r}|\mathbf{b}_{an}) = \text{sgn} \left[ \frac{J(+1|\mathbf{r}, \mathbf{b}_{an}) - J(-1|\mathbf{r}, \mathbf{b}_{an})}{2} \right] \quad (26)$$

Substituting the definition of  $J(b_n|\mathbf{r}, \mathbf{b}_{an})$  from lines (21–24) into Equation (26) and performing some algebra, we find that (26) can be written as

$$\hat{b}_n(\mathbf{r}|\mathbf{b}_{an}) = \text{sgn} [l_n - \delta_n(\mathbf{b}_{an}) - \varepsilon_n(\mathbf{r}|\mathbf{b}_{an})]. \quad (27)$$

The conditional decision rule at node  $n$  for each choice of ancestor bit vectors,  $\mathbf{b}_{an}$ , corresponds to comparing the matched filter output,  $l_n$ , to a threshold.

$$l_n \begin{cases} \hat{b}_1 = +1 \\ \hat{b}_1 \leq \\ \hat{b}_1 = -1 \end{cases} \delta_n(\mathbf{b}_{an}) - \varepsilon_n(\mathbf{r}|\mathbf{b}_{an}) \quad (28)$$

The threshold on the right hand side of Equation (28) has both a deterministic component reflecting the influence of the hypothesized decisions at ancestor nodes

$$\delta_n(\mathbf{b}_{an}) = \mathbf{y}_{n,an} \mathbf{b}_{an} \quad (29)$$

and an adaptive component reflecting decision rules already constructed at descendent nodes

$$\begin{aligned} \varepsilon_n(\mathbf{r}|\mathbf{b}_{an}) = & \frac{1}{2} \mathbf{y}_{n,dn} \left[ \hat{\mathbf{b}}_{dn}(\mathbf{r} + 1, \mathbf{b}_{an}) + \hat{\mathbf{b}}_{dn}(\mathbf{r} - 1, \mathbf{b}_{an}) \right] \\ & + \frac{1}{4} \left[ \hat{\mathbf{b}}_{dn}^T(\mathbf{r} + 1, \mathbf{b}_{an}) \mathbf{Y}_{dn,dn} \hat{\mathbf{b}}_{dn}(\mathbf{r} + 1, \mathbf{b}_{an}) - \hat{\mathbf{b}}_{dn}^T(\mathbf{r} - 1, \mathbf{b}_{an}) \mathbf{Y}_{dn,dn} \hat{\mathbf{b}}_{dn}(\mathbf{r} - 1, \mathbf{b}_{an}) \right] \\ & + \frac{1}{2} \left[ \mathbf{b}_{an}^T \mathbf{Y}_{an,dn} - \mathbf{l}_{dn}^T \right] \left[ \hat{\mathbf{b}}_{dn}(\mathbf{r} + 1, \mathbf{b}_{an}) - \hat{\mathbf{b}}_{dn}(\mathbf{r} - 1, \mathbf{b}_{an}) \right]. \end{aligned} \quad (30)$$

<sup>21</sup>For the more general  $M$ -ary case, there would be  $(M - 1)$  comparisons.

<sup>22</sup>Dividing by 2 in (26) has no effect on the sign and is included to put the subsequent expressions into a form that can be compared with standard results.

In particular, note that for nodes at the bottom level of the tree, there are *no* descendants and, consequently,  $\varepsilon_n(\mathbf{r}|\mathbf{b}_{an}) = 0$ . Hence, at the lowest level of the tree, the decision rules in Equation (28) for each of the hypothesized set of values,  $\mathbf{b}_{an}$ , correspond to comparing  $l_n$  to the fixed threshold given in Equation (29). This non-zero threshold represents the adjustment of the test statistic to reflect the effect of users at ancestor nodes.

The calculations of  $\varepsilon_n(\mathbf{r}|\mathbf{b}_{an})$ , the adaptive portions of each threshold, have a child-separable structure.

$$\varepsilon_n(\mathbf{r}|\mathbf{b}_{an}) = \eta_{cn_1}(\mathbf{r}|\mathbf{b}_{an}) + \eta_{cn_2}(\mathbf{r}|\mathbf{b}_{an}) + \cdots + \eta_{cn_{K_n}}(\mathbf{r}|\mathbf{b}_{an}) \quad (31)$$

This is easy to see from the structure of  $\mathbf{Y}_{dn,dn}$ . Note the grouping structure of  $\mathbf{S}_{dn} = [\mathbf{S}_{fcn_1} \mathbf{S}_{fcn_2} \cdots \mathbf{S}_{fcn_{K_n}}]$ . That is,  $\mathbf{S}_{dn}$  consists of orthogonal sub-matrices, one for each child and its descendants. Furthermore, for each child node,  $\mathbf{S}_{fcn_i} = \{\mathbf{S}_{cni} \mathbf{S}_{dcni}\}$ ,  $\mathbf{S}_{dcni}$  consists of orthogonal sub-matrices. It is this nesting of orthogonal sub-matrices that gives  $\mathbf{Y}_{dn,dn}$  a nested block diagonal structure that leads to the separation of  $\varepsilon_n(\mathbf{r}|\mathbf{b}_{an})$  into terms.

In (31), the term  $\eta_{cn_i}(\mathbf{r}|\mathbf{b}_{an})$  represents the contribution of the  $i^{th}$  child of node  $n$  to the adaptive threshold at node  $n$ . Hence, the calculation of the adjustment  $\varepsilon_n$  may be done in parts, one for each child of node  $n$ . We use the family notation,  $fn = \{n, dn\}$  in showing the formula for the terms of Equation (31)

$$\begin{aligned} \eta_n(\mathbf{r}|\mathbf{b}_{apn}) = & \frac{1}{2} \mathbf{y}_{pn,fn} [\hat{\mathbf{b}}_{fn}(\mathbf{r}|1, \mathbf{b}_{apn}) + \hat{\mathbf{b}}_{fn}(\mathbf{r}|-1, \mathbf{b}_{apn})] \\ & + \frac{1}{4} [\hat{\mathbf{b}}_{fn}^T(\mathbf{r}|1, \mathbf{b}_{apn}) \mathbf{Y}_{fn,fn} \hat{\mathbf{b}}_{fn}(\mathbf{r}|1, \mathbf{b}_{apn})] - \hat{\mathbf{b}}_{fn}^T(\mathbf{r}|-1, \mathbf{b}_{apn}) \mathbf{Y}_{fn,fn} \hat{\mathbf{b}}_{fn}(\mathbf{r}|-1, \mathbf{b}_{apn}) \\ & + \frac{1}{2} [\mathbf{b}_{apn}^T \mathbf{y}_{apn,fn} - \mathbf{1}_{fn}^T] [\hat{\mathbf{b}}_{fn}(\mathbf{r}|1, \mathbf{b}_{apn}) - \hat{\mathbf{b}}_{fn}(\mathbf{r}|-1, \mathbf{b}_{apn})]. \end{aligned} \quad (32)$$

We may organize the implementation of the optimal decision rule as follows. Starting at the bottom of the tree and progressing to the top, we construct augmented decision tables as illustrated in Table 1.<sup>23</sup>

At each bottom node  $n$ , we compute the conditional optimal decision  $\hat{b}_n(\mathbf{r}|\mathbf{b}_{an})$  by comparing  $l_n$  to the precomputed threshold  $\delta_n(\mathbf{b}_{an})$ . For each node at this level we may calculate and store  $\eta_n(\mathbf{r}|\mathbf{b}_{an})$  to be used at the next level. We move to the parent node,  $pn$ , calculating the threshold for this node by starting with the pre-computable portion,  $\delta_{pn}(\mathbf{b}_{apn})$ , and adding to it the adjustments,  $\eta_n(\mathbf{r}|\mathbf{b}_{an})$ , from each of the children of node  $pn$ . We compare  $l_{pn}$  to this threshold to make a decision.

For the root node,  $n$ , corresponding to the top level of the tree  $an = \emptyset$  and  $\delta_n(\mathbf{b}_{an}) = 0$ , hence, there is a single threshold to be computed from the  $\eta$ 's stored at the children of the root node.

### 4.3 A Binary Example

Let us illustrate this procedure for the simple signal set shown in Figure 3. Consider node 1 at the lowest level. In this case the table that is constructed for node 1 is shown in Table 2. Note that each value of  $\eta_1(\mathbf{r}|b_7)$  depends on the two decisions  $\hat{b}_1(\mathbf{r}|+1, +1)$  and  $\hat{b}_1(\mathbf{r}|-1, +1)$ .

$$\eta_1(\mathbf{r}|+1) = \frac{1}{2} y_{5,1} [\hat{b}_1(\mathbf{r}|+1, +1) + \hat{b}_1(\mathbf{r}|-1, +1)] + \frac{1}{2} [y_{7,1} - l_1] [\hat{b}_1(\mathbf{r}|+1, +1) - \hat{b}_1(\mathbf{r}|-1, +1)] \quad (33)$$

and

$$\eta_1(\mathbf{r}|-1) = \frac{1}{2} y_{5,1} [\hat{b}_1(\mathbf{r}|+1, -1) + \hat{b}_1(\mathbf{r}|-1, -1)] + \frac{1}{2} [-y_{7,1} - l_1] [\hat{b}_1(\mathbf{r}|+1, -1) - \hat{b}_1(\mathbf{r}|-1, -1)] \quad (34)$$

For example, suppose  $y_{1,5} = y_{5,1} = 2$  and  $y_{1,7} = y_{7,1} = 1$  and  $l_1 = \frac{5}{2}$ . In this case we can calculate the decisions and the corresponding values of the  $\eta_1$ 's. This has been done in Table 3. Similarly, tables are also constructed at the other bottom level nodes, 2-4.

<sup>23</sup>Notice that there are half as many values of  $\eta_n$  in the table as there are values of  $\hat{b}_n$ . Since there is one value of  $\eta_n$  for each value of  $\mathbf{b}_{apn}$ , we organize the values of  $\mathbf{b}_{an}$  into pairs corresponding to  $[\pm 1, \mathbf{b}_{apn}]$ .

Moving to the next level of the tree, consider node 5. In this case we compute the following conditional estimates.

$$\hat{b}_5(\mathbf{r} + 1) = \text{sgn}(l_5 - y_{5,7} - \eta_1(\mathbf{r} + 1) - \eta_2(\mathbf{r} + 1)) \quad (35)$$

and

$$\hat{b}_5(\mathbf{r} - 1) = \text{sgn}(l_5 + y_{5,7} - \eta_1(\mathbf{r} - 1) - \eta_2(\mathbf{r} - 1)) \quad (36)$$

where  $\eta_1(\mathbf{r} + 1)$  and  $\eta_1(\mathbf{r} - 1)$  are the quantities in Table 2 for node 1. Similarly,  $\eta_2(\mathbf{r} + 1)$  and  $\eta_2(\mathbf{r} - 1)$  are the corresponding quantities that would be in the table for node 2. At this point note that we can *discard* part of Table 2 and part of the corresponding table for node 2 and consolidate the remaining information into a single table for node 5. Specifically, suppose that  $\hat{b}_5(\mathbf{r} + 1) = -1$ . This implies that the best choice for  $b_5$  is  $-1$  if  $b_7 = +1$ . We can discard the first row of Table 2 since this corresponds to choosing  $\hat{b}_5 = +1$  when  $b_7 = +1$ . Similarly, we can discard the analogous row of the table for node 2. That is, once we have computed the values in Equations (35) and (36) we can assemble the following vectors.

$$\hat{\mathbf{b}}_{f_5}(\mathbf{r} + 1) = \begin{bmatrix} \hat{b}_5(\mathbf{r} + 1) \\ \hat{b}_1(\mathbf{r}|\hat{b}_5(\mathbf{r} + 1), +1) \\ \hat{b}_2(\mathbf{r}|\hat{b}_5(\mathbf{r} + 1), +1) \end{bmatrix}, \quad \hat{\mathbf{b}}_{f_5}(\mathbf{r} - 1) = \begin{bmatrix} \hat{b}_5(\mathbf{r} - 1) \\ \hat{b}_1(\mathbf{r}|\hat{b}_5(\mathbf{r} - 1), -1) \\ \hat{b}_2(\mathbf{r}|\hat{b}_5(\mathbf{r} - 1), -1) \end{bmatrix} \quad (37)$$

We may now construct Table 4 which would reside at node 5.

Since node 7 has no ancestors, a single threshold correction,  $\eta_5(\mathbf{r})$ , is calculated from Equation (32) with  $n = 5$  by dropping the last term since  $ap_5 = a_7 = \emptyset$ . The calculation of  $\eta_5(\mathbf{r})$  from (32) would use the following substitutions.

$$\mathbf{y}_{pn,fn} = \mathbf{y}_{7,f_5} = [y_{7,5} \ y_{7,1} \ y_{7,2}], \quad \mathbf{Y}_{fn,fn} = \mathbf{Y}_{f_5,f_5} = \begin{bmatrix} y_{5,5} & y_{5,1} & y_{5,2} \\ y_{5,1} & y_{1,1} & 0 \\ y_{5,2} & 0 & y_{2,2} \end{bmatrix}$$

Finally, at the top of the tree, since  $an = \emptyset$ ,  $\delta_7 = 0$ , and the optimal decision rule at node 7 is

$$\hat{b}_7 = \text{sgn}(l_7 - \eta_5(\mathbf{r}) - \eta_6(\mathbf{r})) \quad (38)$$

where  $\eta_6(\mathbf{r})$  is computed in an analogous fashion through the construction of the table for node 6. Once we have  $\hat{b}_7$ , e.g.  $\hat{b}_7 = +1$ , we immediately have the full optimal estimate by picking off the appropriate elements of the tables for nodes 5 and 6, e.g.  $\hat{b}_{f_5}(\mathbf{r} + 1)$  and  $\hat{b}_{f_6}(\mathbf{r} + 1)$ .

## 5 Conclusion

In this paper we examine the problem of uncoded multiple access (MA) joint detection for the case in which user signatures are not orthogonal. Specifically, we are interested in the “over-saturated” scenario that would occur when the number of users in a communication system is increased beyond the dimension of the signal space available for transmission. This over-saturating of the signal space with users can, in principle, be accomplished with minimal impact on system performance, assuming that *optimal* detection can be implemented. The optimal detector for the general over-saturated case has a complexity which is exponential in the number of users. In this paper, we find a low complexity optimal algorithm for this case through the imposition of a hierarchical cross-correlation structure on the user signature waveforms. The tree-structure is explained and the tree-detector is derived. This tree processing procedure takes advantage of the cross-correlation structure that has been imposed; the algorithm is pyramidal in that estimates and thresholds calculated at a given level of the tree are either discarded or used in the calculation of estimates at the next higher level of the tree.

The tree detector gives the *optimal* estimate with an extremely low computational complexity. The complexity is derived and for typical cases of interest it is bound by a low-order-polynomial in the number of users. This is an enormous savings in computations over the  $O(M^K)^{24}$  computations needed if the signatures did not exhibit any structure.

For the development of the detector in this paper we have assumed that the set of user signals seen at the receiver exhibit the needed structure. This assumption implies the following: (1) the channel response,  $H(\cdot)$  is capable of being estimated and the set of assigned users signals,  $\mathbf{x}_k$ , leads to the set of received signals,  $\mathbf{s}_k = H(\mathbf{x}_k)$ , that exhibit the needed tree-structure, and (2) perfect the synchronization of user transmissions is possible. The future work of the authors includes the relaxing of these idealizations. The lifting of assumptions leads to two avenues of study (1) waveforms that retain tree structure in the presence of timing inaccuracies or multipath, (2) examine robustness of our detector to small violations of signal set structure.

## References

- [1] R. Lupas and S. Verdú. “Linear multiuser detectors for synchronous code-division multiple-access channels”. *IEEE Trans. Inform. Theory*, 35:123–136, Jan. 1989.
- [2] S. Verdú. “Computational complexity of optimum multiuser detection”. In *Algorithmica*. Springer-Verlag, 1989.
- [3] M. Varanasi and B. Aazhang. “Multistage detection in asynchronous code-division multiple-access communications”. *IEEE Trans. on Comm.*, 38, Apr. 1990.
- [4] A. Duel-Hallen. “Decision-feedback multiuser detector for synchronous code-division multiple access channel”. *IEEE Trans. on Comm.*, 41:285–290, Feb. 1993.
- [5] E. A. Lee and D. G. Messerschmitt. *Digital Signal Communication*. Kluwer Academic Publishers, Boston, 2 edition, 1994.
- [6] J. A. F. Ross and D. P. Taylor. “Vector assignment scheme for M+N users in N-dimensional global additive channel”. *Electronics Letters*, 28:1634–6, Aug. 1992.
- [7] J. A. F. Ross and D. P. Taylor. “Multiuser signaling in the symbol-synchronous AWGN channel”. *IEEE Trans. Inform. Theory*, 41:1174–8, Jul. 1995.
- [8] S. Mallat. “A theory for multiresolution signal decomposition: the wavelet representation”. *IEEE Trans. on Pattern Anal. and Mach. Intell.*, PAMI-11:674–693, Jul. 1989.
- [9] R. Coifman and M. V. Wickerhauser. “Entropy-based algorithms for best basis selection”. *IEEE Trans. Inform. Theory*, 38(2):713–718, Mar. 1992.
- [10] A. J. Viterbi. “Error bound for convolutional codes and an asymptotically optimum decoding algorithm”. *IEEE Trans. Inform. Theory*, 13:260–269, April 1967.

---

<sup>24</sup>Here, it is assumed that  $M$ -ary PAM is used by the each of the  $K$  users.



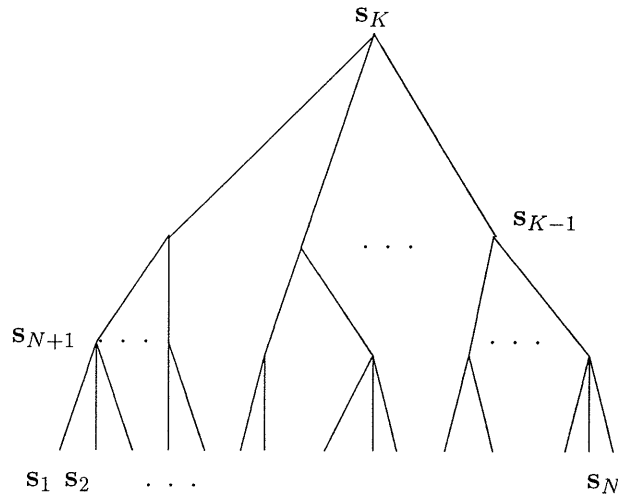


Figure 1: This example of a general tree shows the correlation structure needed among signature vectors within the signature set.

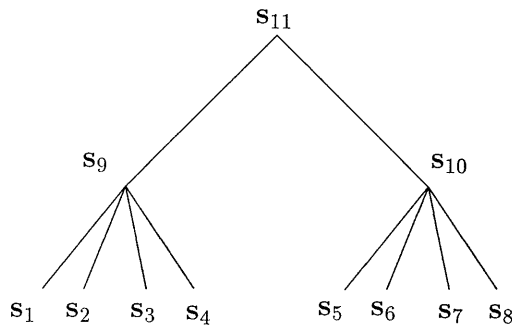


Figure 2: Correlation tree for a wavelet packet signature set.

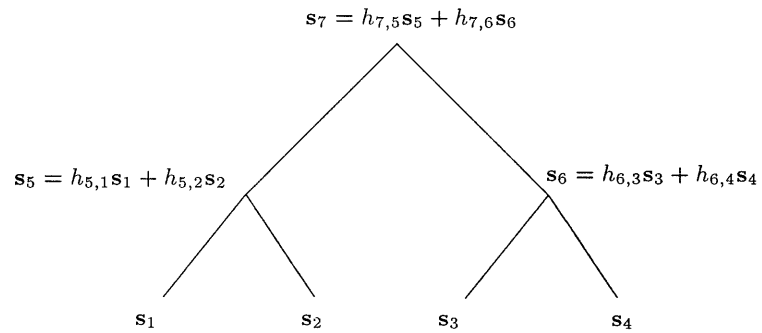
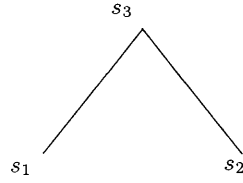
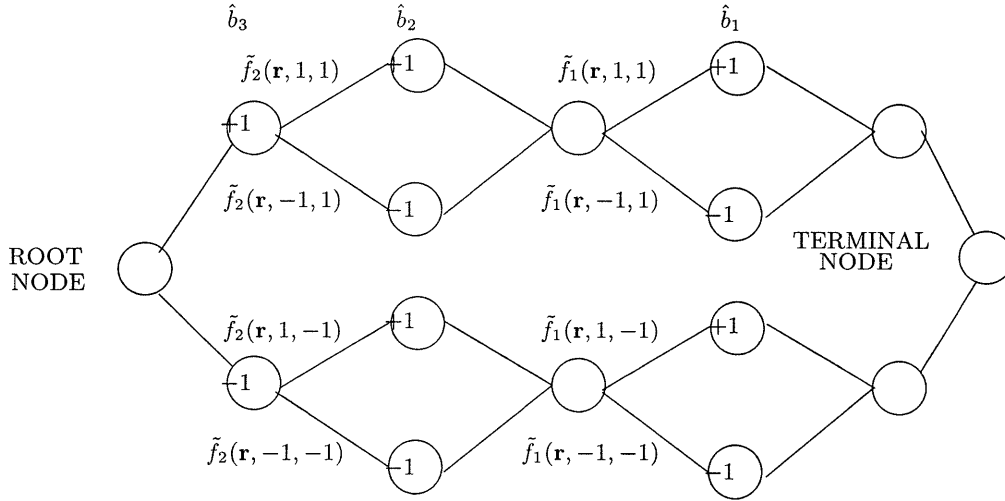


Figure 3: This is an example of a set of tree-structured signature vectors.



(a)



(b)

Figure 4: (a) Tree structure for signature set of 3 vectors. (b) Node/path graph corresponding to tree in (a) where the unlabeled branches have metric values of zero. Unlabeled nodes are dummy nodes inserted to reflect the independence between the variables corresponding to either side of the dummy node.

Table 1: Table created at node  $n$  for the general binary case.

$\mathbf{b}_{an}^T = [b_{pn}b_{p^2n}\cdots b_{p^{(l-1)}n}]$	$\hat{b}_{fn}(\mathbf{r} \mathbf{b}_{an})$	$\eta_n(\mathbf{r} \mathbf{b}_{an}) = \eta_n(\mathbf{r} b_{p^2n}, b_{p^3n}, \cdots, b_{p^{(l-1)}n})$
$[+1, +1, +1, \cdots, +1]$	$\hat{b}_{fn}(\mathbf{r} +1, +1, +1, \cdots, +1)$	$\eta_n(\mathbf{r} +1, +1, \cdots, +1)$
$[-1, +1, +1, \cdots, +1]$	$\hat{b}_{fn}(\mathbf{r} -1, +1, +1, \cdots, +1)$	
$[+1, -1, +1, \cdots, +1]$	$\hat{b}_{fn}(\mathbf{r} +1, -1, +1, \cdots, +1)$	$\eta_n(\mathbf{r} -1, +1, \cdots, +1)$
$[-1, -1, +1, \cdots, +1]$	$\hat{b}_{fn}(\mathbf{r} -1, -1, +1, \cdots, +1)$	
$\vdots$	$\vdots$	$\vdots$

Table 2: Table created at node 1 for the example in Figure 3.

$\mathbf{b}_{a1}^T = [b_5 b_7]$	$\hat{b}_1(\mathbf{r} \mathbf{b}_{a1})$	$\eta_1(\mathbf{r} b_7)$
[+1 +1]	$\hat{b}_1(\mathbf{r} + 1, +1) = \text{sgn}(l_1 - y_{1,5} - y_{1,7})$	$\eta_1(\mathbf{r} + 1)$
[-1 +1]	$\hat{b}_1(\mathbf{r} - 1, +1) = \text{sgn}(l_1 + y_{1,5} - y_{1,7})$	
[+1 -1]	$\hat{b}_1(\mathbf{r} + 1, -1) = \text{sgn}(l_1 - y_{1,5} + y_{1,7})$	$\eta_1(\mathbf{r} - 1)$
[-1 -1]	$\hat{b}_1(\mathbf{r} - 1, -1) = \text{sgn}(l_1 + y_{1,5} + y_{1,7})$	

Table 3: Specific instance of the table at node 1 of our example.

$\mathbf{b}_{a1}^T = [b_5 b_7]$	$\hat{b}_1(\mathbf{r} \mathbf{b}_{a1})$	$\eta_1(\mathbf{r} b_7)$
[+1 +1]	-1	3/2
[-1 +1]	1	
[+1 -1]	1	2
[-1 -1]	1	

Table 4: Table created at node 5 in our example.

$\mathbf{b}_{a5}^T = b_7$	$\hat{b}_5(\mathbf{r} b_7)$	$\eta_5(\mathbf{r})$
+1	$\hat{\mathbf{b}}_{f5}(\mathbf{r} + 1)$	$\eta_5(\mathbf{r})$
-1	$\hat{\mathbf{b}}_{f5}(\mathbf{r} - 1)$	