Low Complexity Optimal Joint Detection for Over-Saturated Multiple Access Communications

by

Rachel E. Learned

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 1997

Author .....
Department of Electrical Engineering and Computer Science
Jan 30, 1997

Certified by .....
Alan S. Willsky
Professor
Thesis Supervisor

Accepted by .....
Professor Athur C. Smith
Chairman
Department of Electrical Engineering and Computer Science

Low Complexity Optimal Joint Detection for Over-Saturated Multiple Access Communications

by

Rachel E. Learned

Submitted to the Department of Electrical Engineering and Computer Science
on Jan 30, 1997, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Optimal joint detection for interfering (non-orthogonal) users in a multiple access communication system has, in general, a computational complexity which is exponential in the number of users. For this reason, optimal joint detection has been thought impractical for large numbers of users. A number of *sub-optimal* low complexity joint detectors have been proposed for direct sequence spread spectrum user waveforms which have properties suitable for mobile cellular systems. There are, however, other systems, such as satellite systems, for which other waveforms may be considered. This thesis shows that there are user signature set selections which enable *optimal* joint detection that is *extremely low* in complexity. When a hierarchical cross-correlation structure is imposed on the user waveforms, optimal detection can be achieved with a tree-structured receiver having complexity that is, in typical cases, a *low-order-polynomial* in the number of users. This is a huge savings over the *exponential* complexity needed for the optimal detection of general signals.

Work in recent literature has shown that a hierarchically structured signal set can achieve over-saturation (more users than dimensions) with no growth in required signal-to-noise ratio. The proposed tree detector achieves low complexity optimal joint detection even in this over-saturated case.

In this thesis the optimal one-shot tree joint detector is derived for the special case of all user signatures (including phases) exactly known at the receiver and its behavior in non-ideal conditions is examined via simulation. For the more realistic case of having an unknown or a partially known phase at the receiver the optimal one-shot tree joint weight/phase estimator is derived and its performance is studied through simulations. Three procedures that take advantage of having a sequence of symbol frames are proposed: the optimal joint weight/phase sequence estimator, the multi-frame phase estimate average, and the multi-frame recursive phase estimate.

Thesis Supervisor: Alan S. Willsky
Title: Professor

Acknowledgments

First, and foremost, I wish to thank my thesis advisor, Alan Willsky for providing me with years of guidance and inspiration. His inherent curiosity for engineering research enabled me to adopt a topic beyond his area of interest. I am grateful to Alan for fostering innovation through his enthusiasm to embrace new ideas. My experiences in the Stochastic Systems Group have set a foundation that will benefit me throughout my career.

I would also like to thank my thesis committee; Professor Greg Wornell for suggesting the research area of MA communications and joint detection, and Professor Mitch Trott for fielding my questions throughout my doctoral research.

I am most appreciative of the support and guidance that the people at the MIT Lincoln Laboratory provided. I wish to thank Dr. Vincent Chan, the Communications division leader, Dr. Dave McElroy the assistant division head, and Dr. Bill Greenberg, the Satellite Communications Technology group leader for taking me on as a research assistant. This experience gave me the opportunity to interact with many bright and talented satellite/communications engineers. I am especially grateful to Bill Greenberg and group 64 for welcoming me into the group. I wish to extend my most sincere thanks to Dr. Don Boroson for his insightful enthusiasm and for his never-tiring interest in my work. I am particularly grateful to group 64's computer systems manager, Mark Fishman, for his efforts that continually saved my computer endeavors from ending up in the electronic scrap heap. In addition I wish to thank my friend, Dr. Scott Stadler, who consistently volunteered his time to answer questions and who readily offered the use of his computer for my simulations. I also extend my gratitude to Ed Rolfe who eased the craziness of my last month of thesis preparation by kindly turning over two of his computers for my exclusive use. Additionally, I would like to express my thanks to Dr. Dean Kolba, Lori Jeromin, Dr. Russ Rhodes, and Tom Shake for offering their knowledge about satellite communication systems. I wish to thank Linda Wesley, Mary Anne Roy, and Karen Morgan for getting my papers through clearance and for arranging my conference trips. And finally, I acknowledge Marilyn Semprucci, Al Richard, Tom Moore, Gloria Liias, Bob Figucia, Susie Schmidt, Linda Kukolich, Doug White, Malcom Coley, Dr. Ben Eaves, Steve Bernstein, and Dr. Ed Bucher for their continuing interest and good wishes. I will always look back fondly on my time with group 64 at the MIT Lincoln Laboratory.

I would like to thank the SSG administrative assistant, Ben Halpern, for all the times he offered his help. Additionally, I would like to thank my close friend and office-mate Seema Jaggi for readily and enthusiastically giving her time, from setting up a proof to organizing a wedding. I also wish to thank my friend and co-Lincoln Lab RA Aradhana Narula and my friend and co-SSGer Mike Daniel for their camaraderie. In addition, I wish to thank my mother-in-law Usha Kamal for the many deliveries of homemade dinners during the last months of thesis preparation. I would also like to express special thanks to my mom, Rosalie Learned, whose pride in my success supplies me with unwaivering inspiration. Moreover, it is with warm and sincere appreciation that I extend thanks to my husband, Manny Kamal, for his continued love, friendship, support, and understanding throughout my doctoral studies.

Contents

List of Figures

List of Tables

Chapter 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 fixed 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 waveform design, user packing, detection and estimation, development of low complexity receivers, etc. This thesis 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 ([8, 23, 21, 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.[1] If there is only one user transmitting through an additive white Gaussian noise channel, the detector that maximizes the a posteriori joint probability for the user weights is realized at the receiver by a simple

[1] 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.

matched filter followed by a quantization to the closest weight used in transmission ([7]). If, however, many users were to transmit through the channel the situation can become far more complex.

One MA case in which detection is simple is that in which the user waveforms are orthogonal. In this case, as in the single user case, a matched filter followed by a quantization to the closest weight used in transmission is optimal for each user.[2] The restriction to orthogonal signal sets, however, is often not a satisfactory one.

In particular, 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 time-varying fading, time-varying multipath or frequency dispersion, it is possible to constrain user signals to be orthogonal.[3] Of course, this choice limits the number of users to the dimension of the signal space available for transmission ([18]). "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.[4] 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 be confronted.

The challenge, then, is to design optimal, or very near optimal, detectors for MA systems that employ non-orthogonal signal sets. As discussed in [23], the optimal joint detector for an arbitrary, non-orthogonal signal set has exponential complexity

[2]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.

[3]For example, current MA processing-satellite systems employing narrow beam terrestrial antennas assign each user a disjoint portion of the available frequency spectrum. This is frequency division multiple access (FDMA).

[4]Moreover, as can be seen in the work of Ross and Taylor [16, 15, 17], it is, indeed, possible to design signal sets having more users than dimensions where the minimum inter-decision-point distance resulting from use of this set is the same as that achieved by the orthogonal set. Their design constrains all users to have powers no higher than the users in the orthogonal set. Sections 2.4 and 3.1.1 present the results of Ross and Taylor in greater detail.

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, the convention currently used, even in the case of non-orthogonal users, is a bank of matched filters where for each user the interference from all other users is assumed to be 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 Verdú in [8], the performance loss of the conventional approach, as compared to the optimal, can be significant.[5] This has motivated several researchers ([8, 23, 21, 4]) to consider slightly 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 were developed for the case of pseudo-noise user signals[6] and require the set of user signals to be linearly independent[7]. These suboptimal approaches offer near optimal performance when the signal energy to noise energy ratio is very high for all users.

In contrast, this thesis 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.[8] In particular, the

[5] In particular, for the "near-far" problem (large power variations among interfering users) the conventional detector fails consistently.

[6] The MA detection literature is heavily concentrated on the cellular problem for which code division multiple access (CDMA)—through the use of pseudo-noise user signals—exhibits advantages over orthogonal signals, for example an orthogonal spot is not wasted when a person is not talking. By restricting the user waveforms to be pseudo-noise pulses, orthogonality among users is not possible.

[7] 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.

[8] In contrast to previous work with suboptimal detectors, user signals are not constrained to be pseudo-noise pulses. As will be discussed in Chapter 3, the structure is chosen to aid low complexity detection; this structure would be used as a guideline for the choice or design of the actual waveforms to be transmitted.

class of signal sets considered in this thesis 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. Moreover, in the communication literature, we find that Ross and Taylor ([16, 17]) have developed signal design guidelines that fit $K > N$ users in N dimensions while preserving the minimum distance of a corresponding orthogonal system. The advantageous tree structure is a by-product of their design.

This thesis is organized as follows. Chapter 2 formally states the problem and the general optimal solution. Several existing one-shot detectors are shown, the over-packing of users is motivated, and the difficulty of the general over-saturated problem is explained. Chapter 3 details the specific signature structure used throughout this work. For the special scenario of all user signatures (including phases) exactly known at the receiver, the optimal one-shot tree joint detector is derived, examples are given and the computational complexity is calculated. Violations of the ideal conditions are tested via simulations. Chapter 4 derives the one-shot tree joint detector for the cases of an unknown or a partially known phase at the receiver. Again, examples are given to illustrate the detection procedure and computational complexity is calculated. The complexity of the joint weight/phase estimator can, under some circumstances, grow to beyond practicality, hence, some sub-optimal procedures are developed. Simulation results are reported to show performance of the optimal detectors proposed in this chapter. Chapter 5 derives both optimal and sub-optimal tree weight/phase estimators that would be used with a training sequence. Chapter 6 discusses issues for future work on this topic and offers concluding remarks.

Chapter 2

Background

2.1 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 ([12]), the N-dimensional complex signal space would correspond to $\mathbb{C}^N$ and the multiuser joint detection problem may be stated as follows: for a given set of user waveforms represented in signal space by the set of signal vectors, $\{\mathbf{s}_k\}_1^K$, $\mathbf{s}_k \in \mathbb{C}^N$, the general uncoded detection problem is to compute an estimate of weights, $\mathbf{b}$, from an observation $\mathbf{r} \in \mathbb{C}^N$,

$$\mathbf{r} = \sum_{k=1}^{K} b_k \mathbf{s}_k + \sigma \mathbf{n} = \mathbf{S}\mathbf{b} + \sigma \mathbf{n}, \tag{2.1}$$

where

- K is the number of users.
- $\mathbf{b} \in \Gamma = \{[b_1 \cdots b_K]^T \mid b_i \in P_i\}$, where P_i is some finite set of amplitudes (real or complex) and the b_i's are independent and identically distributed (i.i.d.), uniform. For P_i having M elements, this is M-ary signaling.

- $\mathbf{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_K]$ is an $N \times K$ matrix whose columns are user signal vectors as seen at the receiver.
- $\mathbf{n}$ is a vector of independent, circularly Gaussian, zero mean, unit variance complex random variables.
- σ is the noise standard deviation.

The satellite uplink channel is well modeled by Equation (2.1) where the received waveforms are, simply, translated and attenuated replicas of the transmitted waveforms.[1]

A two dimensional, two user example of a received signal for which the noiseless received signal is denoted by $\mathbf{r}'$,

$$\mathbf{r}' = b_1\mathbf{s}_1 + b_2\mathbf{s}_2,$$

is shown in Figure 2-1. The signature vectors are as shown in Figure 2-1-(a) and (b). If binary phase shift keying (BPSK) is employed, i.e., $b_i \in \{+1, -1\}$, the range of $\mathbf{r}'$, the possible noise free received signal vector, consists of the four points in the real plane, as shown in Figure 2-1-(c). The collection of points that comprise the range of $\mathbf{r}'$ will be referred to as the **received constellation**. Figure 2-1-(d) depicts the probability density for the actual received vector, $\mathbf{r}$, as a density cloud.[2] The density cloud qualitatively shows that the received vector, $\mathbf{r}$, is more likely to be near one of the four points in Figure 2-1-(c) and less likely to be far from the points.

[1]Knowledge of the operations of existing and next generation military satellites that will be referred to throughout this thesis was obtained through discussions with Dr. Don Boroson, the Assistant Group Leader of the Satellite Communications Technology Group at the MIT Lincoln Laboratory. The text on satellite communications by Gagliardi ([5]) is also recommended for more on the modeling and operation of communication satellites.

[2]The probability density function (PDF) for this example is the sum of four appropriately shifted and normalized two-dimensional Gaussian functions. The next section mathematically shows the general PDF for the MA detection problem.

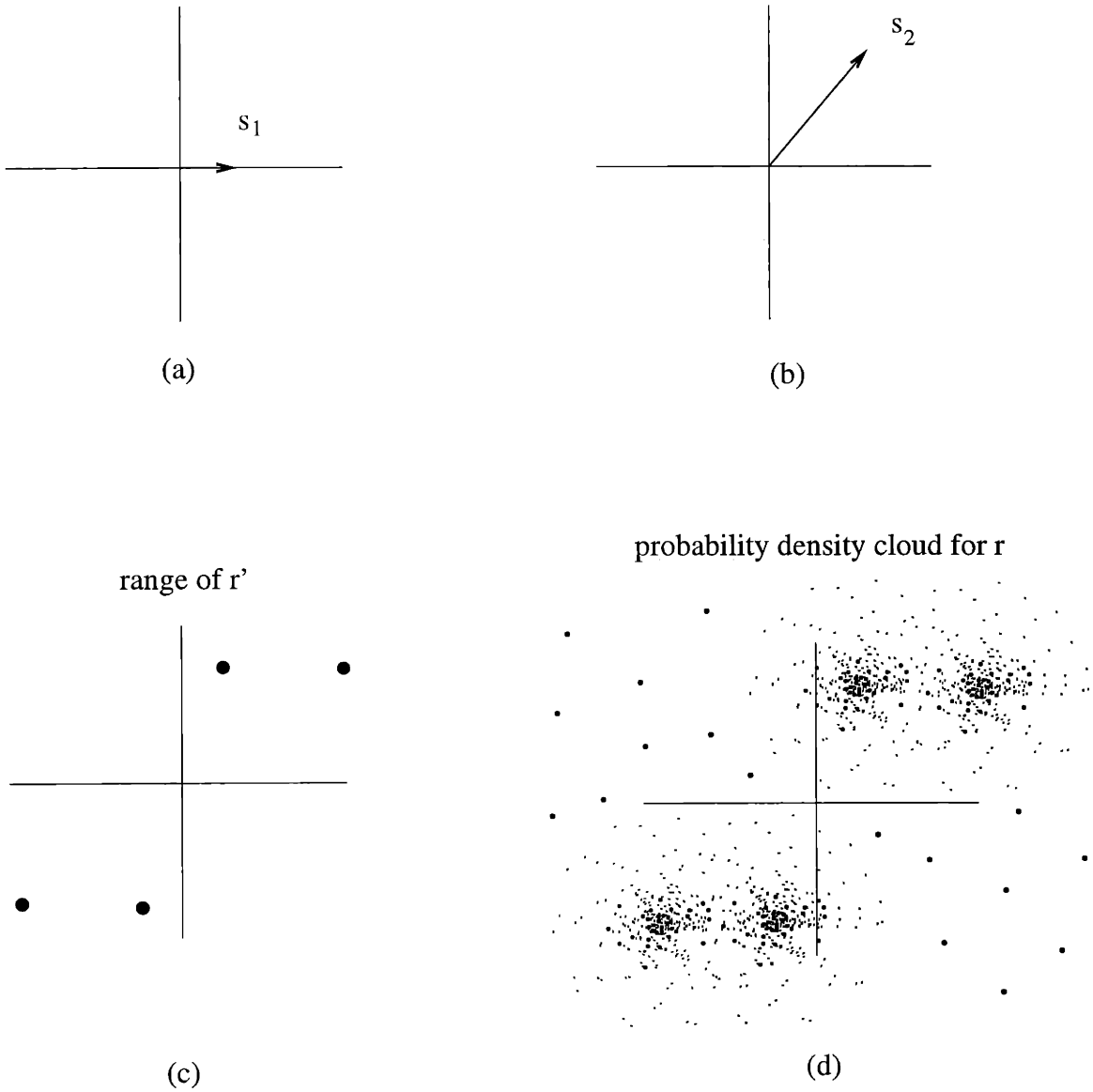


Figure 2-1: Example of two users in two real dimensions. (a) The signature of user 1, $\mathbf{s}_1$, is depicted as a vector pointing along one dimension. (b) User 2 has a signature vector that lies in both dimensions. (c) Set of all possible points, $\mathbf{r}' = b_1\mathbf{s}_1 + b_2\mathbf{s}_2$, $b_1 \in \{+1, -1\}$. (d) Probability density function depicted qualitatively as a density cloud for $\mathbf{r} = \mathbf{r}' + \sigma\mathbf{n}$, $\mathbf{n}$ having zero mean, independently Gaussian elements.

2.2 The General Optimal Solution

In this section, the maximum a posteriori (MAP) joint weight estimator is derived for the problem stated in the previous section. Recall that the MAP estimator results from using a cost function that equally penalizes all possible ways of making an error ([20]). The MAP estimator for $\mathbf{b}$, minimizes the probability of there being an error in *any* of the users' weight estimates. The MAP construction is used throughout this thesis.[3] For this derivation, no special structure among user signature waveforms is assumed.

The MAP joint weight estimator chooses the values for $\mathbf{b} \in \Gamma$ that maximizes the a posteriori probability density function (PDF) for $\mathbf{b}$, given the received vector, $\mathbf{r}$,

$$p_{\mathbf{b}|\mathbf{r}}(\mathbf{b}|\mathbf{r}) = p_{\mathbf{r}|\mathbf{b}}(\mathbf{r}|\mathbf{b})\frac{Prob(\mathbf{b})}{p_{\mathbf{r}}(\mathbf{r})}.$$

For an actual system, $Prob(\mathbf{b})$ is well modeled as being the same for every realization of $\mathbf{b}$. The MAP detector, then, will result in the same estimate found by the maximum likelihood (ML) detector which maximizes $p_{\mathbf{r}|\mathbf{b}}(\mathbf{r}|\mathbf{b})$ over $\mathbf{b}$. We can easily write $p_{\mathbf{r}|\mathbf{b}}(\mathbf{r}|\mathbf{b})$ as

$$p_{\mathbf{r}|\mathbf{b}}(\mathbf{r}|\mathbf{b}) = (\frac{1}{\sqrt{2\pi}\sigma})^{2N} \exp(-\frac{1}{2\sigma^2}||\mathbf{r} - \mathbf{S}\mathbf{b}||^2).$$

The optimal joint weight/phase estimator is given by

$$\hat{\mathbf{b}}(\mathbf{r}) = \arg \max_{\mathbf{b}} p_{\mathbf{r}|\mathbf{b}}(\mathbf{r}|\mathbf{b}), \quad \mathbf{b} \in \Gamma$$

or

$$\hat{\mathbf{b}}(\mathbf{r}) = \arg \max_{\mathbf{b}} \ln[p_{\mathbf{r}|\mathbf{b}}(\mathbf{r}|\mathbf{b})], \quad \mathbf{b} \in \Gamma.$$

[3]Note that a single user in the system may be more interested in the average probability of weight error for each user. A worthwhile extension of the work done in this thesis would be the derivation of detectors that minimize the average probability of making a weight error for *each user*. More is said on this topic in the future work section of Chapter 6.

Discarding constant terms and constant negative multipliers from the function to be maximized gives the optimal joint detector for general signature matrices, $\mathbf{S}$,

$$\hat{\mathbf{b}} = \arg \min_{\mathbf{b}\in\Gamma} ||\mathbf{r} - \mathbf{Sb}||^2. \tag{2.2}$$

Notice that the general optimal joint detector in Equation (2.2) requires a search over all possible realizations of the weight vector $\mathbf{b}$, the number of which is exponential in the number of users. For typical numbers of users, this search is not possible with even the most advanced computers available today.

The ML joint weight/phase estimator may be interpreted as follows. The optimal detector defined by Equation (2.2) finds the point, $\hat{\mathbf{r}}'$, in the range of the random variable $\mathbf{r}' = \mathbf{Sb}$ that is closest[4] to the realization of the random variable $\mathbf{r}$. This closest point, $\hat{\mathbf{r}}'$, lies on one of the points in the range of $\mathbf{r}'$. The value of $\mathbf{b}$ that corresponds to this point constitutes $\hat{\mathbf{b}}$, and the weight estimate is correct, i.e., $\hat{\mathbf{b}} = \mathbf{b}_{true}$, if $\hat{\mathbf{r}}'$ lies on the same point as does $\mathbf{r}'_{true}$.

For the simple example in Figure 2-1, the decision regions for $\hat{\mathbf{b}}$, given any possible value of the received vector, $\mathbf{r}$, are shown in Figure 2-2.

2.3 Existing Joint Detectors for MA Communications

This section briefly describes four detectors, the conventional detector (used in systems of present), and three more proposed in the literature, the decorrelating joint detector ([8, 9]), the decision feedback joint detector ([4]), and the multistage joint detector ([21, 22]). For simplicity, all detectors are shown for BPSK signaling.

Systems of present do not yet employ joint detection. Rather, each user is treated as if it were the only user in the system, where any interference due to the presence

[4]Closest, in terms of Euclidean distance.

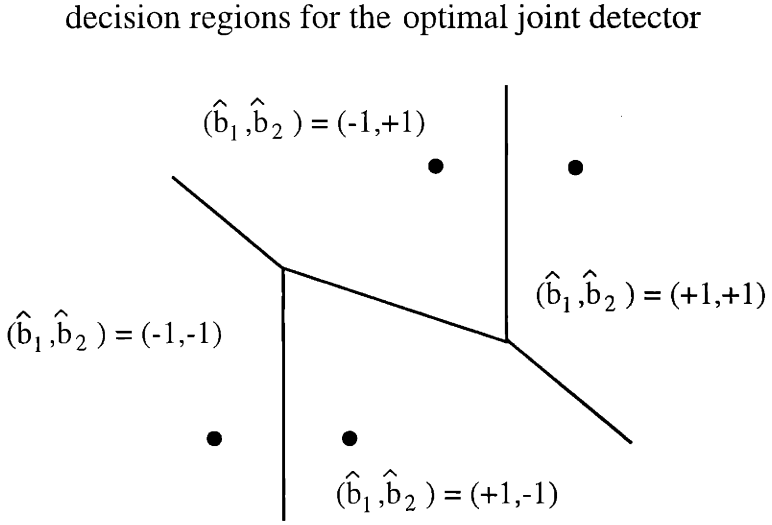


Figure 2-2: Optimal decision regions for the example of two users in Figure 2-1.

of other users is treated as if it were additive white Gaussian noise. The conventional approach to receiver design is to use a matched filter and slicer,[5]

$$\hat{b}_k(\mathbf{r}) = \text{sgn}[\mathbf{s}_k^H \mathbf{r}], \tag{2.3}$$

or

$$\hat{\mathbf{b}}_{conventional}(\mathbf{r}) = \text{sgn}[\mathbf{S}^H \mathbf{r}], \tag{2.4}$$

where sgn represents the signum function and $\hat{b}_k$ denotes the estimated/detected weight of the k^{th} user. Here, $\mathbf{S}^H$ is the complex conjugate transpose, or Hermitian, of $\mathbf{S}$.

The conventional matched filter (2.3) represents the optimal receiver in two cases. In the first case, the user signatures must be orthogonal, e.g., if $\mathbf{s}_i^H \mathbf{s}_j = 0 \quad \forall i \neq j$, the conventional detector of Equation (2.4) is equivalent to the optimal detector of Equation (2.2) for BPSK. Typical orthogonal MA systems keep users from interfering through the frequency division or time division of user transmissions. In the second

[5]This assumes binary PSK, $\mathbf{b} \in \{+1, -1\}$.

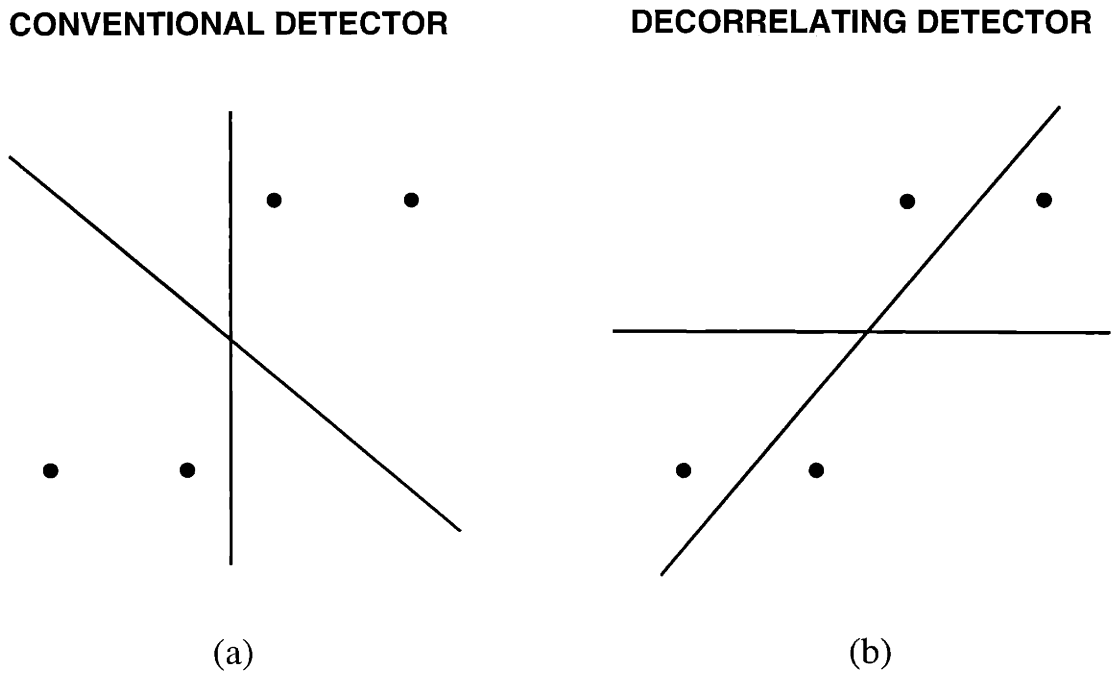


Figure 2-3: (a) Decision regions for the conventional detector for the example of two users in Figure 2-1. (b) Decision regions carved out by the decorrelating detector for the two user example of Figure 2-1.

case, user signatures are designed to look like white Gaussian noise to one another.[6] Specifically, users are assigned pseudo-noise waveforms (or direct sequence spread spectrum (DSSS) signatures) which have low cross correlations with one another. Under this assumption, as users are added to the system the multiple user interference raises the noise level; this, in turn, degrades the matched filter performance. Figure 2-3-(a) shows the decision regions that would be carved out by the conventional detector for the two user example from Figure 2-1. Notice that even in the absence of noise, this detector will consistently fail for the set of user signatures in this example since the cross-correlations between these users is too high to assume that they are orthogonal.

[6] In this case the multiple access interference is well modeled to be additive white Gaussian noise if the receiver has no prior knowledge of the waveforms that comprise the multiple access interference. Since the receiver knows each user's waveform, the conventional detector is not strictly optimal for this second case.

The remaining detectors described in this section were developed for use with DSSS signatures (pseudo-noise waveforms). In this case there is no exploitable structure in the set of user signatures, and correspondingly, $\mathbf{S}$ in Equation (2.1) also has arbitrary structure. In [23], it is shown that the optimal joint detector for an arbitrary signal set has exponential complexity in the number of users, K. The following detectors were motivated by the search for low complexity suboptimal solutions for the MA joint detection problem with DSSS signatures.[7]

The decorrelating linear detector of Lupas and Verdú ([8]) is analogous to the linear zero forcing equalizer used to combat single user inter-symbol interference.[8] The decorrelator is

$$\hat{\mathbf{b}}_{decorrelator}(\mathbf{r}) = \mathrm{sgn}[(\mathbf{S}^*)^{-1}\mathbf{r}]. \tag{2.5}$$

where $(\mathbf{S}^*)^{-1}$ is the complex conjugate inverse of $\mathbf{S}$. In words, for each user, the decorrelator "zeros" out the contribution of the other users, leaving only the portion of the received signal which lies in the subspace which is *not* spanned by the rest of the users' signatures. From the sign of this, possibly very small, remnant of the received signal, the weight of the user is estimated. Figure 2-4-(b) shows the decision regions that would be carved out by the decorrelating detector for the two users from Figure 2-1.

Notice that this detector requires $\mathbf{S}$ to have an inverse. This implies that the user signatures are linearly independent, hence, the number of users must not exceed the number of signal dimensions, i.e., $K \leq N$ must be satisfied.[9]

Another joint detection scheme analogous to one used for single user inter-symbol interference is the decision feedback joint detector of Duel-Hallen ([4]). This detector

[7]These detectors will, of course, work well under certain conditions for MA systems that use waveforms other than DSSS.

[8]A good text for single user communications is [7], by Lee and Messerschmitt.

[9]Of course, replacing $(\mathbf{S}*)^{-1}$ with the complex conjugate pseudo inverse, $(\mathbf{S}^*)^+$, is the natural extension of this detector for the case of $K > N$. This will give very poor performance, however. Some discussion of this is offered in Section 2.5.

is based on solving

$$\mathbf{r} = \mathbf{S}\mathbf{b} \tag{2.6}$$

with **QR** factorization.[10] For a matrix, **S**, having linearly independent columns, a convenient factorization can be found,

$$\mathbf{S} = \mathbf{Q}\mathbf{R},$$

for which **Q** is unitary (orthonormal columns, $\mathbf{Q}^H = \mathbf{Q}^{-1}$) and **R** is upper triangular. Equation (2.6) becomes

$$\mathbf{Q}^H\mathbf{r} = \mathbf{R}\mathbf{b},$$

a compact representation of N equations with N unknowns. The last row or equation has only one unknown, since **R** is upper triangular, and we may read off the solution for the last element of **b**. Next we may substitute this solution into the second to last row or equation (feed it back) to obtain the second to last element of **b**, and so on. The decision feedback MA joint detector differs from this method (also called Gram-Schmidt orthogonalization) only slightly due to the existence of noise and the a priori knowledge, $\mathbf{b} \in \{+1, -1\}$. Including noise gives

$$\mathbf{Q}^H\mathbf{r} = \mathbf{R}\mathbf{b} + \sigma\mathbf{Q}^H\mathbf{n}. \tag{2.7}$$

The weight estimate at each stage of the feedback, starting at the bottom of the matrix equation in (2.7) will have one more step at each level than did the noiseless procedure just described. Specifically, the solution at a given level will be changed to a $+1$ (-1) if the sign of the solution to the equation at that level is positive (negative). Figure 2-4 shows the decision regions carved out by the decision feed back joint detector. This detector will give different results for different ordering of the users (or columns of **S**). Figure 2-4-(a) corresponds to stripping off the user with

[10] See the text by Strang [19] for more details on solving matrix equations.

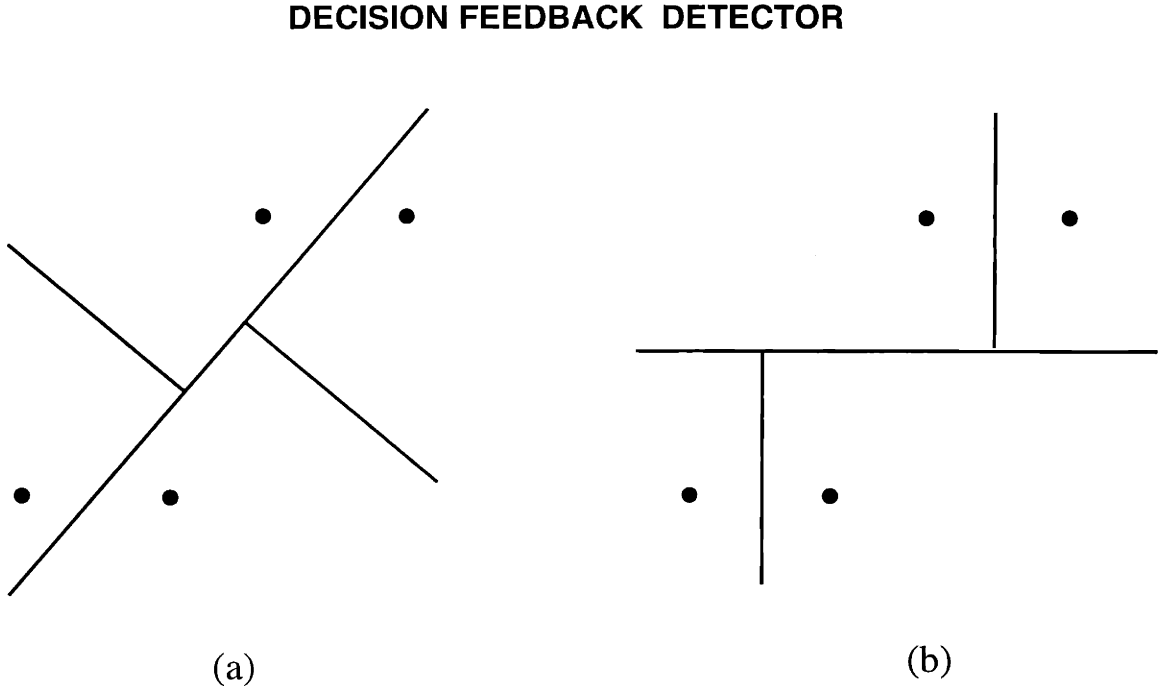


Figure 2-4: Decision regions carved out by the decision feedback detector for two user example of Figure 2-1. (a) Detecting user 1 first. (b) Detecting user 2 first.

the lowest energy first, while Figure 2-4-(b) corresponds to stripping off the user with the highest energy first.

Another approach that has been shown to offer good performance is the multistage joint detector (MJD) developed by Varanasi and Aazhang in [21] and [22]. In this procedure, the weight decisions are achieved after several *iterations*. The iteration defining the MJD is

$$\hat{\mathbf{b}}(m+1) = \text{sgn}[\mathbf{S}^T\mathbf{r} + (\mathbf{E} - \mathbf{S}^T\mathbf{S})\hat{\mathbf{b}}(m)], \tag{2.8}$$

where the energy matrix $\mathbf{E} \triangleq \text{diag}\ (\langle \mathbf{s}_k, \mathbf{s}_k \rangle)_{k=1}^K$. In words, the MJD estimates the interference seen by each user due to the presence of the other users and subtracts it from the output of the matched filter to obtain an estimate of the desired weight. This process is iterated to obtain "better" estimates of the multiuser interference in the hope of improving the estimate of the desired bit.

If the sgn were removed from Equation (2.8), it would be the Jacobi iteration to solve $\mathbf{S}^T\mathbf{r} = \mathbf{S}^T\mathbf{S}\mathbf{b}$ for which $\mathbf{S}^T\mathbf{S} = \mathbf{E} - (\mathbf{E} - \mathbf{S}^T\mathbf{S})$. Recall, that for a Jacobi iteration,

$$\mathbf{S}^T\mathbf{r} = [\mathbf{E} - (\mathbf{E} - \mathbf{S}^T\mathbf{S})]\mathbf{b}$$

is rearranged to give

$$\mathbf{E}\mathbf{b} = \mathbf{S}^T\mathbf{r} + (\mathbf{E} - \mathbf{S}^T\mathbf{S})\mathbf{b},$$

and the $\mathbf{b}$ on the right hand side is replaced with $\mathbf{b}(m)$ while the $\mathbf{b}$ on the left hand side is replaced with $\mathbf{b}(m+1)$. The logic behind the addition of the signum is to take advantage of the prior information on $\mathbf{b}$, namely, $\mathbf{b} \in \{+1, -1\}$.[11] In the presence of noise, the Jacobi iteration is not guaranteed to converge. No figure is given for the decision regions for the MJD.

The common constraint for the methods described in this section is that the user signatures must be linearly independent and the number of users cannot exceed the number of signal space dimensions available.[12] In other words, even if the users are not orthogonal, use of these detectors limits the number of users to that which can be orthogonally fit into the given time/bandwidth signaling space. Alternatively, this thesis is concerned with the operation of MA systems that have *more* users than signal space dimensions, hence, the work in recent literature, although valuable and insightful, is an inappropriate foundation for this thesis. The next section offers some simple examples to motivate the packing of more users than dimensions into a MA communication system.

[11] Since $\mathbf{E}$ is diagonal with positive elements, we may remove it from the left hand side before taking the signum.

[12] Moreover, as these detectors were intended for DSSS waveforms which tend to have small cross-correlations, the performance of these detectors deteriorates as user signatures become more highly correlated.

2.4 Over-saturation of the Signal Space

A communication service provider would, of course, like to provide reliable communications to as many users as possible. In wireless systems of present it is common for users to be turned away due to a lack of signaling spots. The number of spots in current systems is determined, in the case of frequency division or time division multiple access (FDMA or TDMA, respectively), by the number of orthogonal slots available or, in the case of a DSSS (or CDMA) system, by the maximum level of interference noise that can be tolerated ([27, 18]). With the use of the optimal joint detector, more users can be reliably fit into a system than would be possible when assigning users orthogonal signatures.

This over-packing of the signal space is termed **over-saturation**. The simple example in Figure 2-5 shows and example of over-saturation. This figure shows the same user signatures, $\mathbf{s}_1$ and $\mathbf{s}_2$, used in the previous examples, beginning with Figure 2-1. In Figure 2-5-(a) we have allowed a third user, having signature vector $\mathbf{s}_3$, to enter the system. Figure 2-5-(b) shows the received constellation, i.e., the set of possible noiseless received signals, $\mathbf{r}' = b_1\mathbf{s}_1 + b_2\mathbf{s}_2 + b_3\mathbf{s}_3$ for $b_i \in \{+1, -1\}$. Figure 2-5-(c) shows decision regions carved out by the optimal joint detector.

The probability that the optimal detector makes an error is entirely determined by the size, shape and placement of the decision regions relative to the received constellation. When the noise variance is small, the dominating factor in calculating the probability of making an error is the minimum distance between points in the received constellation and decision region boundaries. In Figure 2-5-(d) this minimum distance is denoted as d_{min}. For this example d_{min} is identical to the minimum distance for the optimal detector in Figure 2-2 for *two* users in two dimensions. This means that if a system were operating with the two users having signatures $\mathbf{s}_1$ and $\mathbf{s}_2$ shown in Figure 2-5-(a), with minimal performance degradation, service could simultaneously be offered to a third user by assigning it the signature corresponding to the vector $\mathbf{s}_3$ in Figure 2-5-(a).

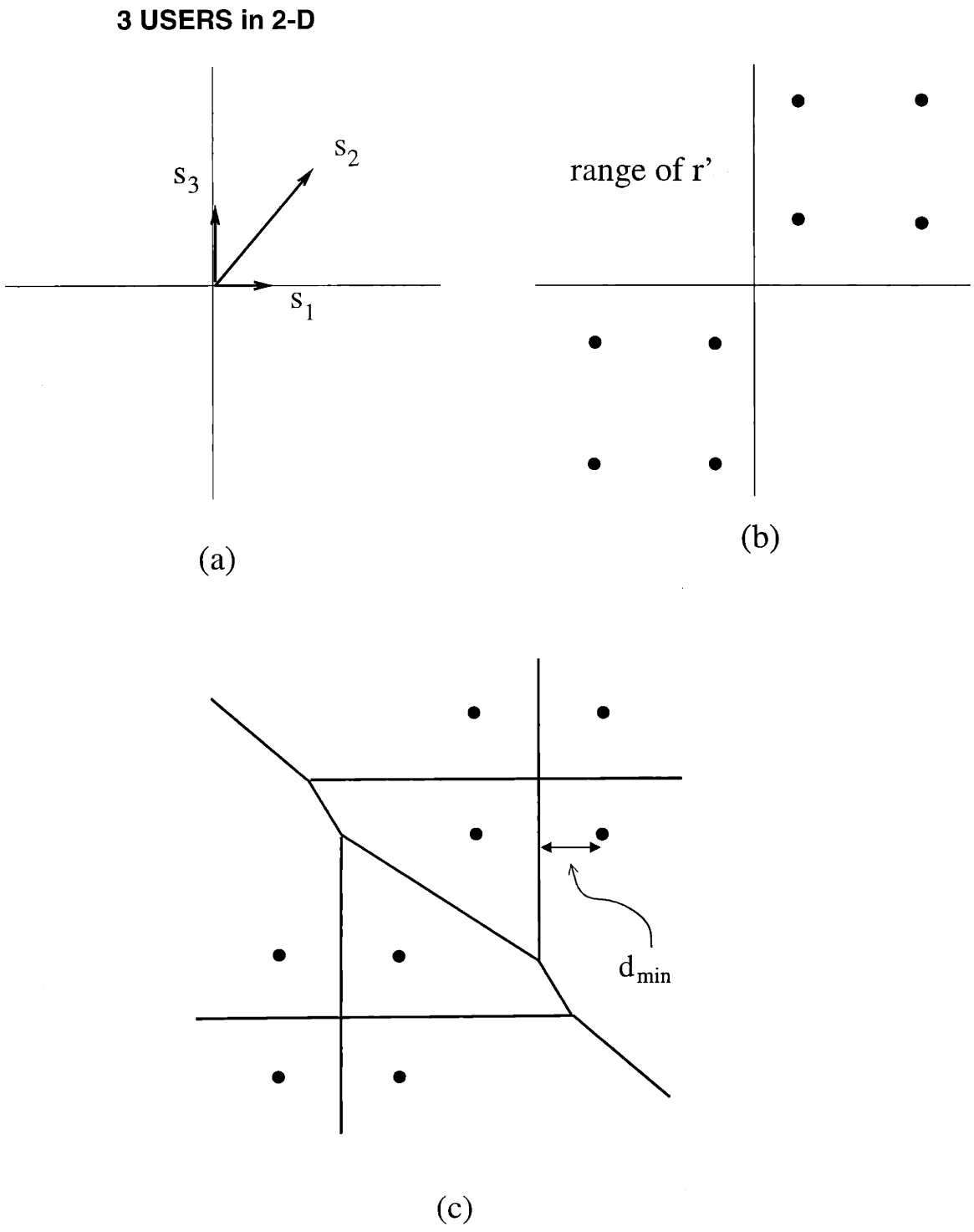


Figure 2-5: (a) Three signatures vectors. (b) The received constellation or range of possible noiseless received signals, $\mathbf{r}' = b_1\mathbf{s}_1 + b_2\mathbf{s}_2 + b_3\mathbf{s}_3$ for $b_i \in \{+1, -1\}$. (c) Decision regions carved out by the optimal joint detector.

In two dimensions, the ability to squeeze in another user arises from the unequal energies of the user signature waveforms. This scenario occurs naturally in practice in a terrestrial wireless system since some users will be closer to the receiver and others will be far away. In a satellite system, some users will have large transmitting antennas powered by comparatively unlimited sources while others may have small portable antennas powered by batteries.[13]

As given by a simple theorem developed by Ross and Taylor in [16, 17], beginning with an orthogonal system of users having equal or different energies, more users can be fit into the system without decreasing the minimum distance compared to that of the received constellation of the original orthogonal set. The number of users is determined by the number of dimensions and the relative powers of the users. One example of equal energy users employing BPSK can fit one additional user of the same energy for every four orthogonal users. Iterative applications of this rule leads to the packing of 21 users in 16 dimensions, 85 users in 64 dimensions, etc. In the limit as the number of dimensions gets large, the number of users can be increase by 33% compared to the orthogonal system. Section 3.1.2 shows some specific examples of the Ross/Taylor sets.

Via simulations, Appendix A shows the performance obtained by the optimal receiver in an MA system employing the over-packed Ross/Taylor sets and compares it to the performance of an MA systems employing orthogonal sets. For a set of equal energy users, the performance degradation due to over-packing is 0.25 dB, on average. This means that for the users in an over-saturated MA system with the Ross/Taylor sets achieve the same bit error rate (BER) as the users in an orthogonal system, the over-packed users must have a signal to noise ratio (SNR) 0.25 dB (or 6%) greater than the SNR for the users in the orthogonal MA system.

From the example in Figure 2-5 and from the results of Ross and Taylor, we see

[13]Large power variations among interfering users is often referred to as the near-far problem. Here, we see that as long as optimal detection is used, the near-far scenario offers an advantage for increasing system throughput.

that if we can use optimal joint detection (which has a performance that is largely dictated by the minimum distance in the received constellation) over-saturation is a viable alternative to orthogonal MA, thus, allowing for a more efficient use of the system resources.

It is important to note, however, that for the general MA problem, optimal joint detection is achieved with procedures having no less than exponential complexity. Moreover, without reliable low complexity detection at the receiver, over-saturation, i.e., throughput gain, is not possible. The next section explains the difficulty of detection for the general over-saturated case.

2.5 Difficulty of General MA Joint Detection

In order to understand the obstacles of developing a low complexity joint detection algorithm the detection problem is described in a geometrical framework. To begin our understanding of the problem we examine its fundamental structure in the absence of noise.

Recall the definition of the set of weight vectors for BPSK

$$\Gamma \triangleq \{[b_1 \dots b_K]^T \mid b_i \in \{+1, -1\} \; \forall i = 1, \cdots, K\}.$$

Geometrically, Γ comprises the vertices of a hypercube of dimension K. For $K > N$ the N-dimensional signature vectors, $\{\mathbf{s}_k\}_1^K$, are linearly *dependent*. This means that the solution, $\mathbf{x}$, to

$$\mathbf{r} = \mathbf{S}\mathbf{x} \tag{2.9}$$

is not unique. By definition of linear dependence, we have

$$\mathbf{S}\alpha = 0$$

for any $\alpha \in \mathcal{N}(\mathbf{S})$, the null-space of $\mathbf{S}$. We may, then, express the solution of Equa-

tion (2.9) as

$$\mathbf{x} = \beta + \alpha,$$

for which $\beta \perp \alpha$ so that

$$\mathbf{r} = \mathbf{S}(\beta + \alpha) = \mathbf{S}\beta. \tag{2.10}$$

The only solutions of interest to the MA problem are the values of $(\beta+\alpha)$ that are contained in the set Γ. In other words, for every $\beta \perp \mathcal{N}(\mathbf{S})$ which solves Equation (2.10) we are interested only in the solutions for which $(\beta + \alpha) \in \Gamma$, where $\alpha \in \mathcal{N}(\mathbf{S})$.

A geometric interpretation of the above discussion follows. We have our set of possible solutions, Γ, the vertices of a K-dimensional hypercube. We separate our solution, $\mathbf{x}$, into two parts, α and β. This corresponds to viewing our vector space, $\mathbb{C}^K$,[14] as the Cartesian product of two complex subspaces, $\mathcal{N}(\mathbf{S})$ and the space which is orthogonal to $\mathcal{N}(\mathbf{S})$, namely, $\mathcal{R}(\mathbf{S}^T)$, the row-space of $\mathbf{S}$. Given the uniquely determined solution, $\beta \perp \mathcal{N}(\mathbf{S})$[15], the general solution must lie in the affine space $\mathcal{W} \triangleq \mathcal{N}(\mathbf{S}) + \beta$. The MA joint detection problem corresponds, geometrically, to finding the point, $\mathbf{x}$, which lies in the intersection of the set Γ and the affine space $\mathcal{W}$.

Our problem of finding the intersection between $\mathcal{W}$ and Γ can be shown for general singular signature matrices, $\mathbf{S}$, to be NP-complete, i.e., to have a solution procedure having a complexity which is, at best, non-polynomial in the number of elements of $\mathbf{x}$ ([11]).[16] No solution which is polynomial in complexity is known to solve the NP-complete problem. In general, sub-optimal attempts to solve Equation (2.9) for an arbitrary singular matrix, $\mathbf{S}$, and for $\mathbf{x}$ from a known discrete set, suffer from possible convergence to local minima.[17]

---

[14]The space $\mathbb{C}^K$ contains all solutions, $\mathbf{x}$.

[15]Alternatively, β may be specified as the minimum length least squares solution to Equation (2.9) where $\beta = \mathbf{S}^+\mathbf{r}$, and $\mathbf{S}^+$ is the pseudoinverse of $\mathbf{S}$.

[16]The noise free problem of over-saturated MA joint detection is exactly the same as the integer programming problem which is known to be NP-complete ([11]).

[17]For example, the decorrelator doesn't make sense for the over-saturated system since extending

The development of low complexity optimal detectors for over-saturated MA communications is the topic of the remaining chapters of this thesis. The approach taken is to recognize that the problem of designing a reliable and realistic MA communications system allows for the joint design of users' signatures and detection procedures. In other words, exercising the control we have over the characteristics (or structure) of $\mathbf{S}$ will make our job easier. As is shown later in this thesis, imposing structure on $\mathbf{S}$ will aid in the design of low complexity optimal detectors for over-saturated MA. The next chapter begins with an illustration of the signal structure that is used throughout this thesis.

---

the decorrelating detector of Equation (2.5) by replacing $\mathbf{S}^{-1}$ with the pseudoinverse, $\mathbf{S}^{+}$, will give an estimate, $\hat{\mathbf{b}}$, that is the projection of β onto the nearest point in Γ. This projection is typically not the point at which Γ and $\mathcal{W}$ intersect. For a more detailed treatment of this geometric interpretation and for an analysis of an alternating projection joint detector, see the paper by Learned, Mallat, Claus, and Willsky ([6]).

Chapter 3

The Multiple Access Tree Detector

In this chapter the signal set structure of interest is described and illustrated. For the ideal case in which each user's signature waveform is entirely known at the receiver, an overview of the hierarchical **one-shot** tree joint detector is given via an example, this low complexity optimal detector is formally derived, and a its computational complexity is calculated. The term "one-shot" is used to specify that the weight estimates for all users are made after observing the received signal over only one symbol duration.[1] Details of the processing procedure of the tree detection algorithm are also given along with a binary example. Via simulations, the tree detector derived for the ideal case of completely known, tree structured signature sets is tested for two non-ideal cases: 1) violation of tree structure, 2) incorrect knowledge of a user's phase at the receiver. In such cases, the tree detector, being sub-optimal, offers surprisingly good performance.

[1] In contrast, to reduce the probability of making errors at the receiver, error correction coding can be used. It can either be implemented separately from the one-shot communication system or it can be paired with detectors that make sequence decisions after collecting several symbols worth of the received signal. This thesis assumes that any decoding is done on the hard decisions made by the one-shot detector.

3.1 The Signal Sets

3.1.1 Signal Vector Set Structure

The geometric structure imposed on the signal vector set[2] is best described by saying that the set of signatures has **tree-structured cross-correlations**. Specifically, **S** will have the desired structure if the signal vectors, the columns of the matrix **S**, can be assigned to the nodes of a tree like the one shown in Figure 3-1. 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 descendant 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 for a considerable amount of flexibility in designing user waveforms. Given a tree, a signal set may be constructed to possess the desired cross-correlation structure. Assume that waveforms at the bottom level of the tree comprise an orthogonal set. An orthogonal set is obtained 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 descendant nodes. Since orthogonal signals have been assigned to the lowest level nodes of the tree, the sets of bottom-level descendants for distinct nodes at the l^{th} level are disjoint, and consequently the signals created at level l are mutually orthogonal.

[2]For ease of notation, the abstract complex baseband signal space representation of real passband signals is used, and, hence, all properties imposed on the signal vectors will also be true for the real waveform counterparts. The signal vector set structure described in this section, therefore, can be viewed as design guidelines for the waveforms that would be used in practice.

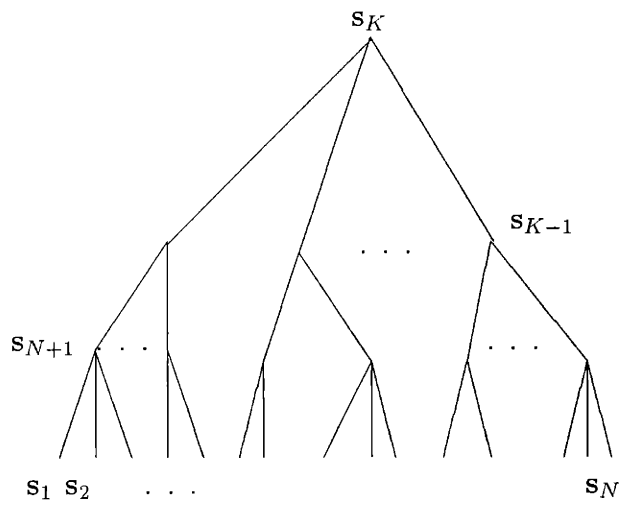


Figure 3-1: This example of a general tree shows the correlation structure needed among signature vectors within the signature set.

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 and nodes of the tree; (b) the specification of *any* orthogonal basis $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_N$, of $\mathbb{C}^N$ which is then assigned to the N nodes on the bottom of the tree;[3] (c) the specification of the weights for each of the linear combinations used to construct signals from their bottom-level descendants; 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 the flexibility to capture linearly independent sets with the desired correlation.[4]

[3]Without loss of generality, we may assume that the bottom level of the tree has exactly N nodes.

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

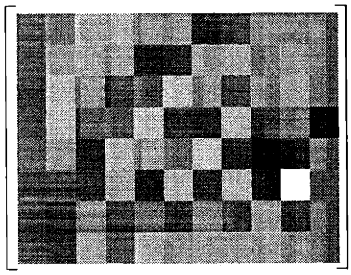
3.1.2 Some Examples of Signature Sets

There are many different signal sets that can be constructed to have the tree structure just described. This section gives 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 ([10, 3]) and one that was introduced in [16, 15, 17] directly in the context of designing signal vector sets for over-saturated MA systems. Note that the following two examples are, simply, two different realizations of requirements (a), (b), and (c), in the above discussion.

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.[5] 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 8×11 image. The values are shown in gray scale where the smallest is denoted by white and the largest is denoted by black.

$\mathbf{S}$ =



Each column of $\mathbf{S}$ is a user signature vector. In order to reveal the tree-structured cross-correlations among user signatures, the absolute values of the elements of $\mathbf{S}^T\mathbf{S}$

[5] For a tutorial treatment of wavelet packets see the paper by Coifman and Wickerhauser ([3]).

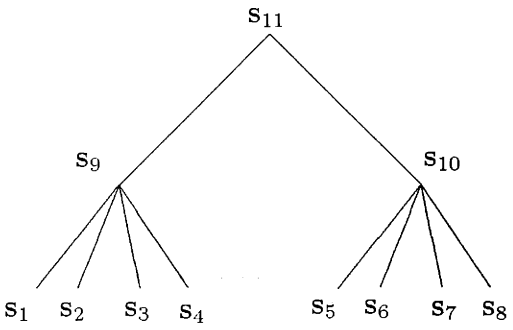
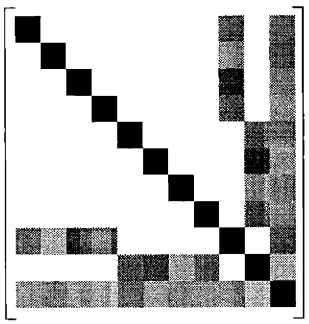


Figure 3-2: Correlation tree for a wavelet packet signature set.

are displayed below, where 0 and 1 are denoted by white and black, respectively.

$\mathbf{S}^T\mathbf{S}$ =



$\mathbf{S}^T\mathbf{S}$ is the matrix of cross-correlations between the received signals of the eleven different users. The wavelet packet signal vector set can be cast onto a tree with three levels as shown in Figure 3-2.

Minimum Distance Sets

Another example is the minimum distance sets developed by Ross and Taylor in [15, 17]. They 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 the vectors $\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_N\}$ is the orthogonal signal set, then there is a specified minimum distance between any two points in the received constellation, i.e., the set $\{\sum_{k=1}^{N} b_k \mathbf{s}_k \mid b_k \in P_k\}$. 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. The reader is referred to [17] for details of their construction.

Ross and Taylor, for antipodal binary modulation, $P_k = \{+1, -1\}$, 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 [17] is briefly repeated below.

$$\mathbf{S} = \begin{bmatrix} & 1/2 & 0 & 0 & 0 & 1/4 \\ & 1/2 & 0 & 0 & 0 & 1/4 \\ & 1/2 & 0 & 0 & 0 & 1/4 \\ & 1/2 & 0 & 0 & 0 & 1/4 \\ & 0 & 1/2 & 0 & 0 & 1/4 \\ & 0 & 1/2 & 0 & 0 & 1/4 \\ & 0 & 1/2 & 0 & 0 & 1/4 \\ \mathbf{I}_{16} & 0 & 1/2 & 0 & 0 & 1/4 \\ & 0 & 0 & 1/2 & 0 & 1/4 \\ & 0 & 0 & 1/2 & 0 & 1/4 \\ & 0 & 0 & 1/2 & 0 & 1/4 \\ & 0 & 0 & 1/2 & 0 & 1/4 \\ & 0 & 0 & 0 & 1/2 & 1/4 \\ & 0 & 0 & 0 & 1/2 & 1/4 \\ & 0 & 0 & 0 & 1/2 & 1/4 \\ & 0 & 0 & 0 & 1/2 & 1/4 \end{bmatrix} \tag{3.1}$$

Here, I_{16} is the 16 dimensional identity matrix. The cross-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.

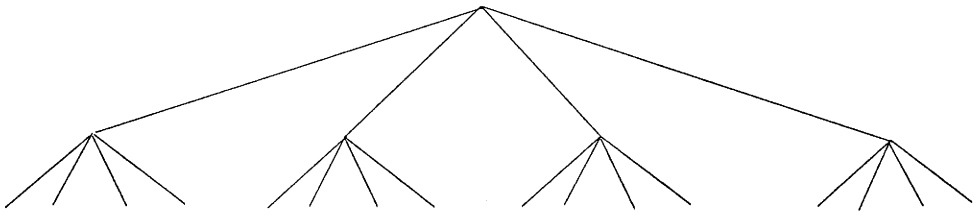
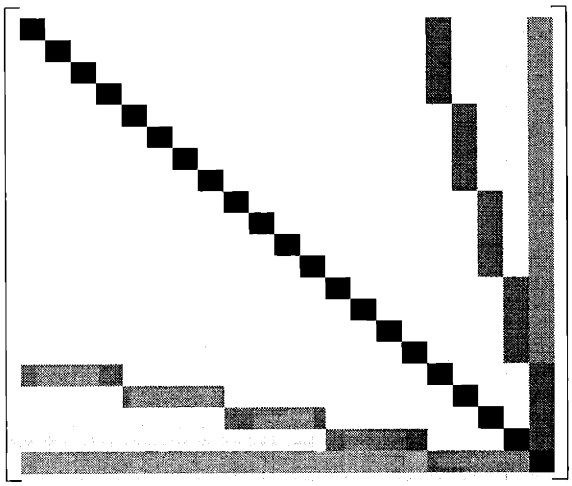


Figure 3-3: Correlation tree for a unit energy minimum distance signature set.

$\mathbf{S}^T\mathbf{S}$ =



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, as shown in Figure 3-3. The signature vector associated with the top of the correlation tree is the right-most column of $\mathbf{S}$. The first 16 columns are associated with the bottom of the tree. These signature sets were designed for their minimum distance property. The tree hierarchy they possess is a by-product that can be exploited in the optimal detector described in Section 3.4.

The remainder of this thesis is devoted to the development of low complexity, optimal, and near optimal joint detectors for an over-saturated system of tree-structured user signatures. It is important to recognize that for the success of an over-saturated system, further investigation of good ways of over-packing users with tree structured signature sets needs to be done to ensure reliable performance. Though the design and analysis of signature sets is beyond the scope of this thesis, the minimum dis-

tance sets offer a good foundation for the rest of this thesis since they do a good job of packing in more users than dimensions. In the limit as the number of dimensions gets large, the minimum distance sets offer a 33% increase in the number of active users in a MA system relative to that of an orthogonal system with very small increases in bit error rate (BER) per user. Appendix A shows some bit error rate curves comparing the performance of optimal joint detection for orthogonal users, a Ross/Taylor set of 5 users in 4 dimensions, and a Ross/Taylor set of 21 users in 16 dimensions. All simulations in this thesis will be done with the minimum distance sets.

3.2 Notation

Notation is introduced that will be used throughout the remainder of this thesis.

- 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[6]
- $an = \{pn, p^2n, p^3n, \ldots, p^{l-1}n\}$: set of indices corresponding to the ancestor nodes of node n[7]
- cn_i: node index for the i^{th} child of node n
- Kn: number of children of node n
- $cn = \{cn_1, cn_2, \ldots, cn_{Kn}\}$: set of indices corresponding to children of node n
- $dn = \{cn_1, dcn_1, cn_2, dcn_2, \cdots, cn_{Kn}, dcn_{Kn}\}$: set of indices corresponding to the descendant nodes of node n [8]

[6]Note that if node n is at level l, the index of the root node can be denoted by $p^{l-1}n$.

[7]This ordering of ancestors is important and will be useful later.

[8]Note that dn is recursively defined, where dcn_i is the set of descendants of node cn_i. In addition, the ordering of nodes into sub-tree groupings will be useful in later sections.

- $fn = \{n, dn\}$: the family of indices associated with node n[9]

Note that the set of descendants for a node, 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 $\mathbf{S}_{dn}$.

The inner products between the user signals is required for the derivation of the estimator in Section 3.6

$$y_{i,j} = \mathbf{s}_i^T \mathbf{s}_j. \tag{3.2}$$

Extending this definition, the following notation is established:

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

[9]Each set of family indices corresponds to an entire sub-tree having root node corresponding to node n.

3.3 Discussion of Processing

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

$$l_i = \mathbf{s}_i^T \mathbf{r}, \quad i = 1, 2, \ldots, K. \tag{3.3}$$

The calculations of each l_i corresponds to processing the data $\mathbf{r}$ through a filter matched to the signal $\mathbf{s}_i$.[10] Reductions in the calculation of the set $\{l_i\}_1^K$ can be obtained by taking advantage of the exact relationships among user signatures on the tree.

Due to the manner in which the tree-structured signal set was created[11], 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 $\{l_i\}_{i=1}^N$, where $i = 1, \ldots, N$ corresponds to the bottom-level nodes. This will allow for an efficient calculation of $y_{i,j}$ and l_i for $i > N$, the non-bottom-level nodes.

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*.[12] The computational reduction is due to a node having far less children than bottom-level descendants. 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 signal set construction

$$\mathbf{s}_n = \mathbf{S}_{cn}\mathbf{h}_n = \sum_{i=1}^{Kn} h_{n,cn_i}\mathbf{s}_{cn_i} \tag{3.4}$$

[10] As will become clear in Section 3.6, $\{l_i\}_1^K$ is the set of sufficient statistics needed for optimal detection.

[11] Recall that a signature at node n was constructed as a linear combination of its bottom-level descendants.

[12] Note that both the minimum distance and the wavelet/wavelet packet sets can easily be chosen to exhibit this tree-recursive quality.

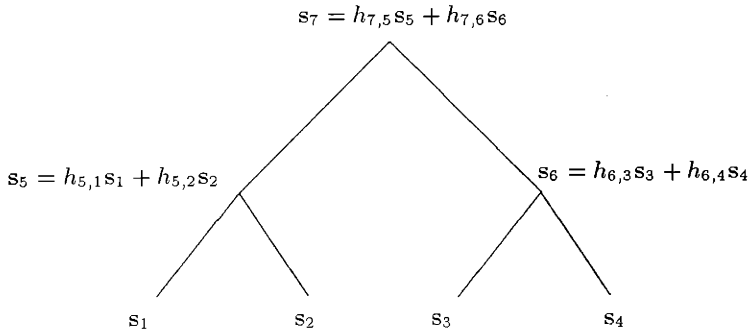


Figure 3-4: An example of a set of tree-structured signature vectors.

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

To illustrate a tree-recursive signal set, consider the signal set shown in Figure 3-4. 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 $\mathbb{C}^4$, 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 $i = 1, \ldots, N$. In particular, from Equation (3.4), 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,

$$l_n = \mathbf{h}_n^T\mathbf{S}_{cn}^T\mathbf{r} = \mathbf{h}_n^T\mathbf{l}_{cn} \tag{3.5}$$

where $\mathbf{l}_{cn}$ is the column vector having elements l_k, $k \in cn$. For our example of Figure 3-4, 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

$$l_5 = h_{5,1}l_1 + h_{5,2}l_2\ , \quad l_6 = h_{6,3}l_3 + h_{6,4}l_4 \tag{3.6}$$

and at the top level of the tree

$$l_7 = h_{7,5}l_5 + h_{7,6}l_6. \tag{3.7}$$

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 ([3]).[13] 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 $l_i = \mathbf{s}_i^T\mathbf{r}$, $i = 1, 2, \ldots, N$ 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.

3.4 The Tree Joint Detection Algorithm

3.4.1 Overview of the Detector

Recall that the optimum joint detector for the problem stated in Equation (2.1) chooses the weight vector estimate, $\hat{\mathbf{b}}$, according to the nearest neighbor or minimum distance rule.

$$\hat{\mathbf{b}} = \arg \min_{\mathbf{b} \in P^K} ||\mathbf{r} - \mathbf{S}\mathbf{b}||^2. \tag{3.8}$$

For ease of discussion, each user is assumed to employ the same M-ary PAM for the remainder of this thesis 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, the columns of $\mathbf{S}$, can achieve optimal detection through an exhaustive search, i.e., the detector performs $M^K - 1$ comparisons to find the best estimate ([23]).

[13] In this case, however, the coefficients in the vectors $\mathbf{h}_n$ are not required to correspond to the filter coefficients used to construct orthogonal wavelets or filter banks. Moreover, a regular structure is not required—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.

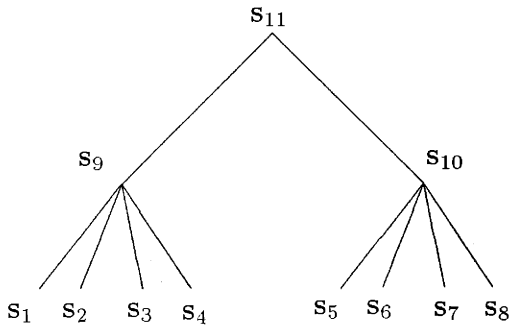


Figure 3-5: An example.

If the signal set has been constructed to have the tree cross-correlation structure described in Section 3.1.1, the optimum detector of Equation (3.8) can be achieved through a tree-structured algorithm that offers a huge reduction in the number of comparisons. In particular, because of this signal set structure, a signature at a given node is correlated with all signatures at its ancestor and descendant nodes and is orthogonal to *all other* signatures on the tree. The weight estimate, $\hat{b}_n$, at a given node, n, will affect the estimates at descendant and ancestor nodes but will not directly affect the other estimates on the tree.

A simple example provides the basic idea. Consider the tree structure in Figure 3-5 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 $\mathbf{s}_5$—$\mathbf{s}_8$ and $\mathbf{s}_{10}$, are coupled with those for $\mathbf{s}_1$-$\mathbf{s}_4$. These estimates can, however, be decouple by looking, instead, at the *conditional* estimates. Specifically, for each possible pair of weight estimates for $\mathbf{s}_9$ and $\mathbf{s}_{11}$, the optimal weight estimates for $\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3, \mathbf{s}_4$ can be independently computed—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 be thought of as producing a **conditional weight estimate table**, i.e., for each possible pair

of choices for the weight estimates for s_9 and s_{11} the optimal weight estimates for s_1—s_4 is known. Similarly, for each pair of possible weight estimates for s_{10} and s_{11} the optimal estimates for s_5—s_8 can be computed. The estimation process may be iterated for s_9: for each possible choice of weight value for its ancestor s_{11} and with knowledge of the just-constructed conditional estimate table for its descendants s_1—s_4, the optimal estimate for s_9 may be computed in a manner decoupled from the analogous computation for s_{10}. This gives conditional estimate tables for s_9 and s_{10} which then can be used to determine the optimal estimate for s_{11} at the top of the tree. Conceptually, once this last estimate is obtained it is a simple matter of successive table look-ups that propagate down the tree to determine the optimal estimates first for s_9 and s_{10} and then for their descendants.

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 descendants. Since each conditional estimate table requires an entry 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 described 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 is 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. The overall complexity, then, is bounded by a very low-order polynomial in K. This is discussed more fully in Section 3.5.

Moreover, while the derivation of the general algorithm given in the next section is most easily explained 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 3.6.

3.4.2 Derivation of Tree Detector

The global cost that must be minimized in Equation (3.8) is

$$F(\mathbf{b}|\mathbf{r}) = ||\mathbf{r} - \mathbf{S}\mathbf{b}||^2. \tag{3.9}$$

In general, $F(\mathbf{b}|\mathbf{r})$ is not separable in weight variables, b_i. Hence, the solution to Equation (3.8) is found by the calculation and comparison of $F(\mathbf{b}|\mathbf{r}), \forall \mathbf{b} \in P^K$.

The introduction of tree-structured cross-correlations transforms the structure of the cost function. The complexity of finding the smallest cost may be reduced by making decisions in stages. The independence of the conditional decisions discussed in the previous section can be seen mathematically. Specifically, the global cost may be separated into independent terms. First, Equation (3.9) may be re-written as

$$F(\mathbf{b}|\mathbf{r}) = \sum_{i=1}^{N} f_i(\mathbf{b}|\mathbf{r}) = \sum_{i=1}^{N} (\mathbf{r}[i] - \mathbf{t}[i])^2 \tag{3.10}$$

where $\mathbf{t} = \mathbf{S}\mathbf{b}$ and $\mathbf{t}[i]$ is the i^{th} element of the vector $\mathbf{t}$. In general, each term $f_i(\mathbf{b}|\mathbf{r})$ is a function of all users' bits.

If the signature set were to exhibit tree-structured cross-correlations, a rotation matrix $\mathbf{S}_R$ may be constructed from the orthogonal basis vectors that reside on the

bottom of the tree,

$$\mathbf{S}_R = [\frac{\mathbf{s}_1}{||\mathbf{s}_1||}\frac{\mathbf{s}_2}{||\mathbf{s}_2||}\cdots\frac{\mathbf{s}_N}{||\mathbf{s}_N||}]. \tag{3.11}$$

Since the cost function of Equation (3.9) is a squared Euclidean distance between a Gaussian random vector, $\mathbf{r}$, and a deterministic vector, $\mathbf{Sb}$, a rotation of this difference vector, $\mathbf{r} - \mathbf{Sb}$, does not change its length or probability distribution. Hence,

$$F(\mathbf{b}|\mathbf{r}) = ||\mathbf{S}_R^T\mathbf{r} - \mathbf{S}_R^T\mathbf{Sb}||^2. \tag{3.12}$$

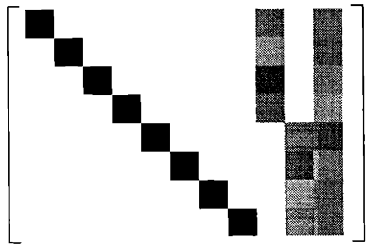
The tree structure is reflected in the position of the zero-valued elements of $\mathbf{S}_R^T\mathbf{S}$. In this form, the partition of $F(\mathbf{b}|\mathbf{r})$ into terms is

$$F(\mathbf{b}|\mathbf{r}) = \sum_{i=1}^{N} \tilde{f}_i(\mathbf{b}|\mathbf{r}) = \sum_{i=1}^{N}(\tilde{\mathbf{r}}[i] - \tilde{\mathbf{t}}[i])^2, \tag{3.13}$$

where $\tilde{\mathbf{r}} = \mathbf{S}_R^T\mathbf{r}$, $\tilde{\mathbf{t}} = \mathbf{S}_R^T\mathbf{Sb}$, and where the indices, $i \in \{1, 2, \ldots, N\}$, correspond to the orthogonal users at the bottom nodes of the tree.

For example, the rotated version of the wavelet packet signal matrix from Section 3.1.2 is

$\mathbf{S}_R^T\mathbf{S}$ =



It is clear that for this example, $\tilde{\mathbf{t}}[i] = [\mathbf{S}_R^T\mathbf{Sb}][i]$ can be written as a linear combination of b_i and the elements of $\mathbf{b}_{ai}$. Here, we have used the ancestor vector notation from Section 3.2.

In general, it follows that the "rotated" cost may be separated into additive terms where each term is a function of only one of the weights $b_i, i \in \{1, 2, \ldots, N\}$, and all of the weights that correspond to its ancestors. Each term, $\tilde{f}_i(\mathbf{b}|\mathbf{r})$ in Equation (3.13) may be explicitly written as $\tilde{f}_i(b_i, \mathbf{b}_{ai}|\mathbf{r})$. Note that for $i, j \in \{1, 2, \ldots, N\}$ and $i \neq j$, $\tilde{f}_i(b_i, \mathbf{b}_{ai}|\mathbf{r})$ and $\tilde{f}_j(b_j, \mathbf{b}_{aj}|\mathbf{r})$ have no common unknown parameters given the values for $\mathbf{b}_{ai}$ and $\mathbf{b}_{aj}$. It follows that the optimal solution may be determined through the optimization of each term conditioned on the values of the weights corresponding to the ancestors of the index, i, of that term.

Since the optimal cost function to be minimized is capable of being decomposed into disjoint parts, the minimization may be formulated as a **shortest path** problem. The twists and turns of the best path correspond to the user weight choices.

As an example, the shortest path construction for the joint estimation of three tree-structured users is shown in Figure 3-6. In the general shortest path problem, we wish to find the path from the root node to the terminal node that has the lowest total cost. The total cost is obtained by adding each path segment cost. The **metric** (or cost) values for each path segment are the terms, $\tilde{f}_j(b_j, \mathbf{b}_{aj}|\mathbf{r})$. The unlabeled path segments in the figure have metric value of zero.

See any text on mathematical programming, [28] for example, for several procedures that can be used to search a node/path graph for the shortest path from its root node to its terminal node. The preferred algorithm would depend upon the available storage and processing resources. Some procedures from which to choose are the "best-first" search (attributed to Dijkstra) [28, 1], or transforming the path/node graph into a transshipment problem so that the transportation simplex procedure may be used [28], or noticing that the node/path graphs for tree structured MA joint detection have the specific stage-by-stage structure needed by the back-to-front procedure of dynamic programming [28, 1]. These and other shortest path search procedures avoid enumerating all possible path segment combinations to arrive at the optimal solution, but with many less operations than needed by the enumeration

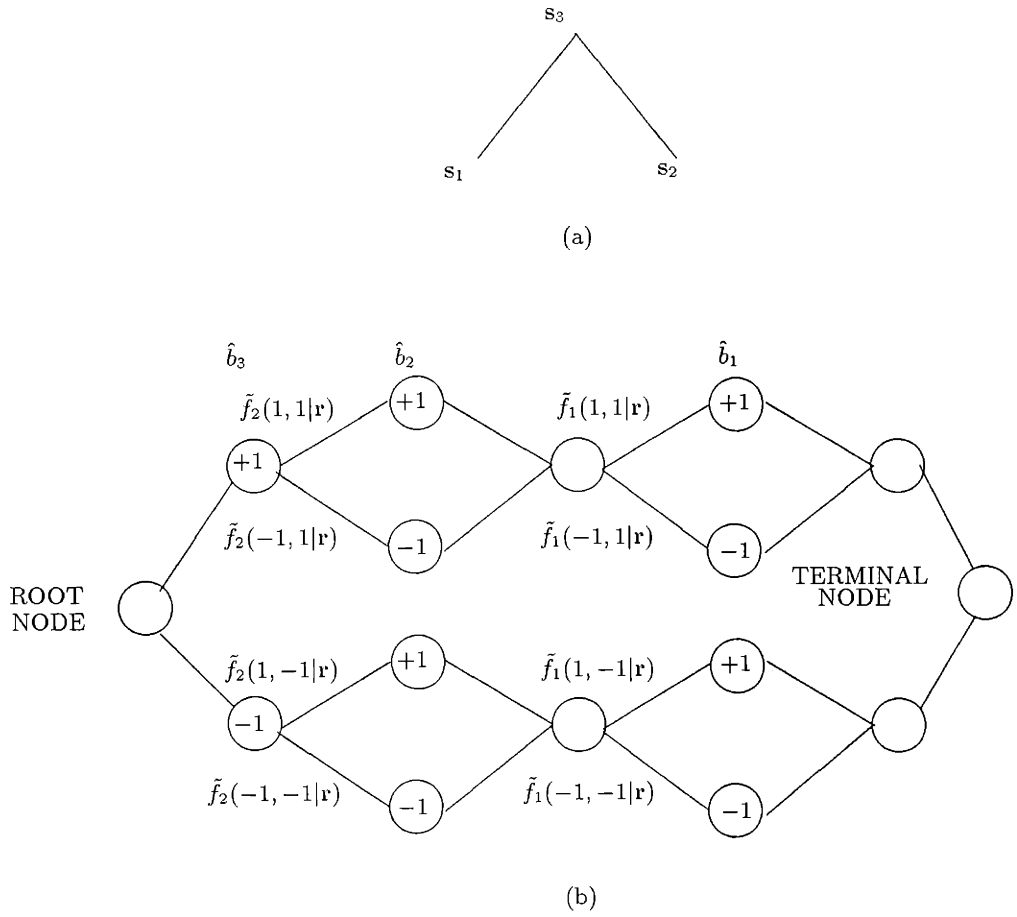


Figure 3-6: (a) Tree structure for signature set of 3 vectors. (b) Node/path graph corresponding to the tree in (a) where the unlabeled branches have metric values of zero. Unlabeled nodes are dummy nodes inserted to reflect the independence between the variables corresponding to either side of the dummy node.

method.

The dynamic program has exactly as many compares as the tree-climbing conditional estimation procedure described in the previous section. The dynamic programming approach can be carried out stage by stage where several cumulative path metrics are constructed and decisions are made at every instance of merging paths.[14] For a system employing an L-level tree and M-ary PAM for all users, at any interior

[14]This type of search is identical to the one used by the Viterbi algorithm for the the node/path graph created by convolutional coding ([26]) or for single user inter-symbol interference ([7]).

stage of the graph, no more than M^L paths need to be retained. Recall that the number of levels in the tree is typically logarithmic in 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.

3.5 Computational Complexity

The conceptual description of the tree detector in Section 3.4.1 and the dynamic programming description in Section 3.4.2 inevitably include many wasteful calculations and storage requirements.[15] Furthermore, it is expected that for each different tree structure and modulation a streamlined implementation is possible. The *order* of complexity of optimal detection may be calculated from 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, the tree is restricted to have exactly Q children emanating from each non-terminal 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. A measure of complexity which is in agreement with the MA joint detection literature is the number of compares, c, needed to perform the detection algorithm;[16] c is stated below and derived shortly.

$$c(N,Q,M) = \frac{M-1}{QM-1}(NQM^{\log_Q N+1} - 1) \tag{3.14}$$

[15]The discussion in Section 3.6 addresses the removal of this redundancy.

[16]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 M 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.

Table 3.1: Computational complexity.

N	K	Exhaustive Detector $c = 2^K$	Tree Detector $c = \frac{8N^{3/2}-1}{7} = O(K^{3/2})$
4	5	32	9
16	21	2.10e+6	73
64	85	3.87e+25	585
256	341	4.4795e+102	4681

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

$$c(N, Q = 4, M = 2) = \frac{8N^{3/2} - 1}{7}. \tag{3.15}$$

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 $O(K^{3/2})$. Table 3.1 shows $c(N, 4, 2)$ for several values of N (and K). Notice that with a typical K of 85 users, an exhaustive joint detector would require computations on the order of 10^{25}, while the optimal tree detector would require only 585 computations. Following the tree-structured signal design guidelines clearly eliminates the computational obstacle of optimal joint detection.

Equation (3.14) is derived 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, n. For a given $\hat{\mathbf{b}}_{an}$ the best of M possible values of $\hat{b}_n$ must be found. 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 tree. It follows that the total number of comparisons needed for the tree algorithm is

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

3.6 Signal Processing for the Optimal Tree Detector

3.6.1 Calculation of the Estimate

As was shown in Section 3.4.2, 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 (3.8) 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 $\mathbf{b}_{an}$;

$$\hat{b}_n(\mathbf{r}|\mathbf{b}_{an}) = \arg \min_{b_n \in P} ||\mathbf{r} - \mathbf{s}_n b_n - \mathbf{S}_{an}\mathbf{b}_{an} - \mathbf{S}_{dn}\hat{\mathbf{b}}_{dn}(\mathbf{r}|b_n, \mathbf{b}_{an})||^2. \tag{3.16}$$

Notation developed in Section 3.2 is used throughout this section. 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 $\{b_n, \mathbf{b}_{an}\}$,

$$\hat{\mathbf{b}}_{dn}(\mathbf{r}|b_n, \mathbf{b}_{an}) = \begin{bmatrix} \hat{\mathbf{b}}_{fcn_1}(\mathbf{r}|b_n, \mathbf{b}_{an}) \\ \hat{\mathbf{b}}_{fcn_2}(\mathbf{r}|b_n, \mathbf{b}_{an}) \\ \vdots \\ \hat{\mathbf{b}}_{fcn_{Kn}}(\mathbf{r}|b_n, \mathbf{b}_{an}) \end{bmatrix}. \tag{3.17}$$

For that same set of $\{b_n\ \mathbf{b}_{an}\}$, $\hat{b}_{cn_i}(\mathbf{r}|b_n\ \mathbf{b}_{an})$ has already been calculated; if the value found for $\hat{b}_{cn_i}(\mathbf{r}|b_n\ \mathbf{b}_{an})$ is ξ, then the sub-vectors on the right hand side of Equation (3.17) are given by

$$\hat{\mathbf{b}}_{fcn_i}(\mathbf{r}|b_n, \mathbf{b}_{an}) = \begin{bmatrix} \hat{b}_{cn_i}(\mathbf{r}|b_n\ \mathbf{b}_{an}) = \xi \\ \hat{\mathbf{b}}_{dcn_i}(\mathbf{r}|b_{cn_i} = \xi,\ b_n, \mathbf{b}_{an}) \end{bmatrix}. \tag{3.18}$$

We examine the argument of the minimization in Equation (3.16) more closely. We may, of course, remove any terms that do not depend on b_n and we can multiply by any positive constant.[17] As a result, some algebra shows that (3.16) is equivalent to

$$\hat{b}_n(\mathbf{r}|\mathbf{b}_{an}) = \arg\max_{b_n \in P} J(b_n|\mathbf{r}, \mathbf{b}_{an}) \tag{3.19}$$

where

$$\begin{aligned} J(b_n|\mathbf{r}, \mathbf{b}_{an}) &= [l_n b_n - \frac{1}{2} b_n^2 y_{n,n}] - b_n \mathbf{y}_{n,an} \mathbf{b}_{an} && (3.20) \\ &\quad + [\mathbf{l}_{dn}^T \hat{\mathbf{b}}_{dn}(\mathbf{r}|b_n, \mathbf{b}_{an}) - \frac{1}{2} \hat{\mathbf{b}}_{dn}^T(\mathbf{r}|b_n, \mathbf{b}_{an}) \mathbf{Y}_{dn,dn} \hat{\mathbf{b}}_{dn}(\mathbf{r}|b_n, \mathbf{b}_{an})] && (3.21) \\ &\quad - b_n \mathbf{y}_{n,dn} \hat{\mathbf{b}}_{dn}(\mathbf{r}|b_n, \mathbf{b}_{an}) && (3.22) \\ &\quad - \mathbf{b}_{an}^T \mathbf{Y}_{an,dn} \hat{\mathbf{b}}_{dn}(\mathbf{r}|b_n, \mathbf{b}_{an}). && (3.23) \end{aligned}$$

Note 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 (3.20) and (3.21).

[17] or multiply by a negative constant and replace minimization by maximization

The first bracketed term $[l_n b_n - \frac{1}{2} b_n^2 y_{n,n}]$ on line (3.20) 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 (3.20), $b_n \mathbf{y}_{n,an} \mathbf{b}_{an}$, represents the interaction between the choice of b_n and the particular hypothesized choices for the ancestor weights. Note that since the values of $\mathbf{b}_{an}$ will be hypothesized, this term can be pre-computed. Line (3.21) represents a counterpart to the bracketed term on line (3.20). Specifically, if all users other than the ones corresponding to the weights $\mathbf{b}_{dn}$ were not present, then

$$[\mathbf{l}_{dn}^T \mathbf{b}_{dn} - \frac{1}{2} \mathbf{b}_{dn}^T \mathbf{Y}_{dn,dn} \mathbf{b}_{dn}] \tag{3.24}$$

represents the decision statistic that would be used to determine the optimal choice of $\mathbf{b}_{dn}$. Since this is *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 (3.22) accounts for the interactions between these descendant decisions and the possible decisions b_n. Likewise, line (3.23) 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 (3.21–3.23) can be computed based on the value of b_n and the calculations that have already been performed at lower levels on the tree.

3.6.2 The Binary Conditional Decision Rule

This section focuses 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 (3.19)[18]. The solution to (3.19) can, therefore, be expressed as[19]

$$\hat{b}_n(\mathbf{r}|\mathbf{b}_{an}) = \text{sgn}\left[\frac{J(+1|\mathbf{r},\mathbf{b}_{an}) - J(-1|\mathbf{r},\mathbf{b}_{an})}{2}\right]. \tag{3.25}$$

Substituting the definition of $J(b_n|\mathbf{r},\mathbf{b}_{an})$ from lines (3.20-3.23) into Equation (3.25) and performing some algebra, (3.25) can be written as

$$\hat{b}_n(\mathbf{r}|\mathbf{b}_{an}) = \text{sgn}\left[l_n - \delta_n(\mathbf{b}_{an}) - \varepsilon_n(\mathbf{r}|\mathbf{b}_{an})\right]. \tag{3.26}$$

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:

$$l_n \underset{\hat{b}_n = -1}{\overset{\hat{b}_n = +1}{\gtrless}} \delta_n(\mathbf{b}_{an}) - \varepsilon_n(\mathbf{r}|\mathbf{b}_{an}). \tag{3.27}$$

The threshold on the right hand side of Equation (3.27) has both a deterministic component reflecting the influence of the hypothesized decisions at ancestor nodes,

$$\delta_n(\mathbf{b}_{an}) = \mathbf{y}_{n,an}\mathbf{b}_{an}, \tag{3.28}$$

and an adaptive component reflecting decision rules already constructed at descendant

[18]For the more general M-ary case, there would be $(M-1)$ comparisons.

[19]Dividing by 2 in (3.25) has no effect on the sign and is included to put the subsequent expressions into a form that can be compared with standard results.

nodes,

$$\begin{aligned}\varepsilon_n(\mathbf{r}|\mathbf{b}_{an}) = \; & \tfrac{1}{2}\mathbf{y}_{n,dn}\left[\hat{\mathbf{b}}_{dn}(\mathbf{r}|+1,\mathbf{b}_{an}) + \hat{\mathbf{b}}_{dn}(\mathbf{r}|-1,\mathbf{b}_{an})\right] \\ & + \tfrac{1}{4}\hat{\mathbf{b}}_{dn}^T(\mathbf{r}|+1,\mathbf{b}_{an})\mathbf{Y}_{dn,dn}\hat{\mathbf{b}}_{dn}(\mathbf{r}|+1,\mathbf{b}_{an}) \\ & - \tfrac{1}{4}\hat{\mathbf{b}}_{dn}^T(\mathbf{r}|-1,\mathbf{b}_{an})\mathbf{Y}_{dn,dn}\hat{\mathbf{b}}_{dn}(\mathbf{r}|-1,\mathbf{b}_{an}) \\ & + \tfrac{1}{2}\left[\mathbf{b}_{an}^T\mathbf{Y}_{an,dn} - \mathbf{l}_{dn}^T\right]\left[\hat{\mathbf{b}}_{dn}(\mathbf{r}|+1,\mathbf{b}_{an}) - \hat{\mathbf{b}}_{dn}(\mathbf{r}|-1,\mathbf{b}_{an})\right].\end{aligned} \tag{3.29}$$

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 (3.27) for each of the hypothesized set of values, $\mathbf{b}_{an}$, correspond to comparing l_n to the fixed threshold given in Equation (3.28). This non-zero threshold represents the adjustment of the test statistic to reflect the interference of users at ancestor nodes.

The calculation of $\varepsilon_n(\mathbf{r}|\mathbf{b}_{an})$, the adaptive portion of each threshold, has a child-separable structure:

$$\varepsilon_n(\mathbf{r}|\mathbf{b}_{an}) = \eta_{cn_1}(\mathbf{r}|\mathbf{b}_{an}) + \eta_{cn_2}(\mathbf{r}|\mathbf{b}_{an}) + \cdots + \eta_{cn_{Kn}}(\mathbf{r}|\mathbf{b}_{an}). \tag{3.30}$$

This is easy to see from the structure of $\mathbf{Y}_{dn,dn}$. Note the grouping structure of $\mathbf{S}_{dn} = [\mathbf{S}_{fcn_1}\ \mathbf{S}_{fcn_2}\ \ldots\ \mathbf{S}_{fcn_{Kn}}]$. That is, $\mathbf{S}_{dn}$ consists of orthogonal sub-matrices, one for each child and its descendants. Furthermore, for each child node, cn_i, we have $\mathbf{S}_{fcn_i} = \{\mathbf{s}_{cn_i}\ \mathbf{S}_{dcn_i}\}$, where $\mathbf{S}_{dcn_i}$ 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 (3.30), 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 ε_n

Table 3.2: Table created at node n for the general binary case.

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

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 (3.30):

$$\begin{aligned}
\eta_n(\mathbf{r}|\mathbf{b}_{apn}) = \ & \tfrac{1}{2}\mathbf{y}_{pn,fn}[\hat{\mathbf{b}}_{fn}(\mathbf{r}|1, \mathbf{b}_{apn}) + \hat{\mathbf{b}}_{fn}(\mathbf{r}|-1, \mathbf{b}_{apn})] \\
& + \tfrac{1}{4}\hat{\mathbf{b}}_{fn}^T(\mathbf{r}|1, \mathbf{b}_{apn})\mathbf{Y}_{fn,fn}\hat{\mathbf{b}}_{fn}(\mathbf{r}|1, \mathbf{b}_{apn}) \\
& - \tfrac{1}{4}\hat{\mathbf{b}}_{fn}^T(\mathbf{r}|-1, \mathbf{b}_{apn})\mathbf{Y}_{fn,fn}\hat{\mathbf{b}}_{fn}(\mathbf{r}|-1, \mathbf{b}_{apn}) \\
& + \tfrac{1}{2}[\mathbf{b}_{apn}^T\mathbf{y}_{apn,fn} - \mathbf{l}_{fn}^T][\hat{\mathbf{b}}_{fn}(\mathbf{r}|1, \mathbf{b}_{apn}) - \hat{\mathbf{b}}_{fn}^T(\mathbf{r}|-1, \mathbf{b}_{apn})].
\end{aligned} \tag{3.31}$$

Implementation of the optimal decision rule may be organized as follows. Starting at the bottom of the tree and progressing to the top, construct augmented decision tables as illustrated in Table 3.2.[20]

At each bottom node n, the conditional optimal decision, $\hat{b}_n(\mathbf{r}|\mathbf{b}_{an})$, is computed

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

by comparing l_n to the pre-computed threshold $\delta_n(\mathbf{b}_{an})$. For each node at this level calculate and store $\eta_n(\mathbf{r}|\mathbf{b}_{an})$ to be used at the next level. 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. 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 $\delta_n(\mathbf{b}_{an}) = 0$; there is a single threshold to be computed from the η's stored at the children of the root node.

3.6.3 A Binary Example

This procedure is illustrated for the simple signal set shown in Figure 3-4. Consider node 1 at the lowest level. In this case the table that is constructed for node 1 is shown in Table 3.3. Note that each value of $\eta_1(\mathbf{r}|b_7)$ depends on the two decisions $\hat{b}_1(\mathbf{r}|+1,\ +1)$ and $\hat{b}_1(\mathbf{r}|-1,\ +1)$,

$$\begin{aligned}\eta_1(\mathbf{r}|+1) = \ & \tfrac{1}{2}y_{5,1}\left[\hat{b}_1(\mathbf{r}|+1,\ +1) + \hat{b}_1(\mathbf{r}|-1,\ +1)\right] \\ & + \tfrac{1}{2}\left[y_{7,1} - l_1\right]\left[\hat{b}_1(\mathbf{r}|+1,\ +1) - \hat{b}_1(\mathbf{r}|-1,\ +1)\right]\end{aligned} \tag{3.32}$$

and

$$\begin{aligned}\eta_1(\mathbf{r}|-1) = \ & \tfrac{1}{2}y_{5,1}[\hat{b}_1(\mathbf{r}|+1,\ -1) + \hat{b}_1(\mathbf{r}|-1,\ -1)] \\ & + \tfrac{1}{2}[-y_{7,1} - l_1][\hat{b}_1(\mathbf{r}|+1,\ -1) - \hat{b}_1(\mathbf{r}|-1,\ -1)].\end{aligned} \tag{3.33}$$

For example, suppose $y_{1,5} = y_{5,1} = 2$ and $y_{1,7} = y_{7,1} = 1$ and $l_1 = \frac{5}{2}$. Table 3.4 shows the values of the η_1's for this case. Similarly, tables are also constructed at the other bottom level nodes, 2–4.

Moving to the next level of the tree, consider node 5. The following conditional

Table 3.3: Table created at node 1 for the example in Figure 3-4.

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

estimates are computed:

$$\hat{b}_5(\mathbf{r}|+1) = \mathrm{sgn}(l_5 - y_{5,7} - \eta_1(\mathbf{r}|+1) - \eta_2(\mathbf{r}|+1)) \tag{3.34}$$

and

$$\hat{b}_5(\mathbf{r}|-1) = \mathrm{sgn}(l_5 + y_{5,7} - \eta_1(\mathbf{r}|-1) - \eta_2(\mathbf{r}|-1)). \tag{3.35}$$

where $\eta_1(\mathbf{r}|+1)$ and $\eta_1(\mathbf{r}|-1)$ are the quantities in Table 3.3 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 part of Table 3.3 and part of the corresponding table for node 2 may be discarded and the remaining information may be consolidated 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 may discard the first row of Table 3.3 since the first row corresponds to choosing $\hat{b}_5 = +1$ when $b_7 = +1$. Similarly, the analogous row of the table for node 2 may be discarded. That is, once the values in Equations (3.34) and (3.35) have been computed the following vectors may be assembled:

$$\hat{\mathbf{b}}_{f5}(\mathbf{r}|+1) = \begin{bmatrix} \hat{b}_5(\mathbf{r}|+1) \\ \hat{b}_1(\mathbf{r}|\hat{b}_5(\mathbf{r}|+1), +1) \\ \hat{b}_2(\mathbf{r}|\hat{b}_5(\mathbf{r}|+1), +1) \end{bmatrix}, \quad \hat{\mathbf{b}}_{f5}(\mathbf{r}|-1) = \begin{bmatrix} \hat{b}_5(\mathbf{r}|-1) \\ \hat{b}_1(\mathbf{r}|\hat{b}_5(\mathbf{r}|-1), -1) \\ \hat{b}_2(\mathbf{r}|\hat{b}_5(\mathbf{r}|-1), -1) \end{bmatrix} \tag{3.36}$$

Table 3.4: 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 3.5: 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	$\hat{\mathbf{b}}_{f5}(\mathbf{r}\|-1)$	

The table residing at node 5 (shown in Table 3.5) may now be constructed.

Since node 7 has no ancestors, a single threshold correction, $\eta_5(\mathbf{r})$, is calculated from Equation (3.31) with $n = 5$ by dropping the last term. The calculation of $\eta_5(\mathbf{r})$ from (3.31) uses the following substitutions:

$$\mathbf{y}_{pn,fn} = \mathbf{y}_{7,f5} = [y_{7,5}\ \ y_{7,1}\ \ y_{7,2}], \quad \mathbf{Y}_{fn,fn} = \mathbf{Y}_{f5,f5} = \begin{bmatrix} y_{5,5} & y_{5,1} & y_{5,2} \\ y_{5,1} & y_{1,1} & 0 \\ y_{5,2} & 0 & y_{2,2} \end{bmatrix}.$$

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

$$\hat{b}_7 = \mathrm{sgn}(l_7 - \eta_5(\mathbf{r}) - \eta_6(\mathbf{r})), \tag{3.37}$$

where $\eta_6(\mathbf{r})$ is computed in an analogous fashion to the computation of $\eta_5(\mathbf{r})$. Once we have $\hat{b}_7$, e.g., $\hat{b}_7 = +1$, the full optimal estimate may be read off the tables for nodes 5 and 6, e.g. $\hat{b}_{f5}(\mathbf{r}|+1)$ and $\hat{b}_{f6}(\mathbf{r}|+1)$.

3.7 Study of Violations of Tree Detector Requirements

The optimal MA tree joint detector has two requirements. First, the user signature waveforms must exhibit tree-structured cross correlations. Without this, the computationally simple tree traversal procedure would not yield the optimal estimate. Second, each of the received user signature waveforms must be completely known at the receiver. With incorrect knowledge of the user signatures, the minimization in Equation (3.8) would not be optimal. This section examines the bit error rate degradation of the tree detector when each of these requirements is violated.

The signature vector sets used for these simulations are based on the equal energy 5 user minimum distance set

$$\mathbf{S} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1/2 \\ 0 & 1 & 0 & 0 & 1/2 \\ 0 & 0 & 1 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 1/2 \end{bmatrix} \tag{3.38}$$

and the equal energy 21 user minimum distance set

$$\mathbf{S}^T = \begin{bmatrix} & & & & & & & & \mathbf{I}_{16} & & & & & & & \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix}. \tag{3.39}$$

Here, I_{16} is the 16 dimensional identity matrix. The trees corresponding to these minimum distance sets are shown in Figure 3-7 and Figure 3-8, respectively.

In an actual system, each user is assigned a signature waveform envelope and a

carrier frequency. The carrier phase is not controllable by the user or by the receiver. To model this phenomenon, prior to the simulations, each user's signature has been multiplied by a unit magnitude complex scalar, arbitrarily chosen.

3.7.1 Experimental Analysis of Performance Under Structure Mismatch

In the case of structure mismatch, a slight violation of the necessary tree-structure is permitted. If, for example, each user's signature at a node of the tree had a non-zero, but known, correlation with it's sibling users at the same level, the tree detector could be used as a sub-optimal low complexity detector. For example, in existing orthogonal FDMA satellite communications systems, user signature waveforms are narrow band signals, lined up side by side in frequency. Due to slight carrier frequency imperfections and timing offsets, user signatures are not strictly orthogonal. A user can typically see -20 to -40 dB of the adjacent channel user's energy.[21] This section shows performance curves resulting from the use of the tree joint detector for user signature sets that are *nearly* tree-structured, just as the FDMA signature sets are *nearly* orthogonal in the satellite system. The tree detector, ignoring this "leakage" into adjacent users, is a suboptimal detector for such a set of signatures.

The equal energy minimum distance sets of 5 users in 4 dimensions and 21 users in 16 dimensions are used for these simulations. Figure 3-9, for the 5 user set, shows the bit error rate averaged over all 5 users. The signatures have -20 dB, -30 dB, and -40 dB cross-correlations with their neighboring users on the tree. In other words, in the 5 user set of Figure 3-7, user 1 has a 0.1 (-20 dB), 0.036 (-30 dB), or 0.01 (-40 dB) cross-correlation with user 2. User 2 has these same cross-correlations with both user 1 and user 3. User 5, at the top of the tree has no neighbors with which

[21] Note that different allocation of channel resources and different signaling schemes result in the wide range of adjacent channel leaks examined in this section. For example, for time division multiple access (TDMA) leaks can be very low and nearly zero. Keeping adjacent channel interference in an FDMA satellite systems down at -30 dB is commonly done ([13]).

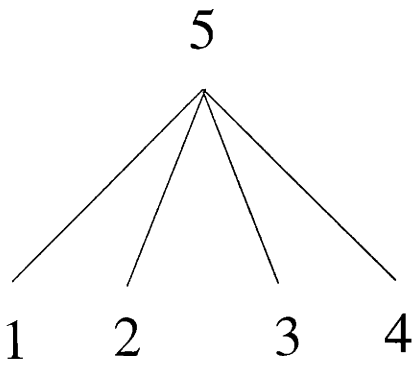


Figure 3-7: Correlation tree for a unit energy minimum distance signature set of 5 users in 4 dimensions.

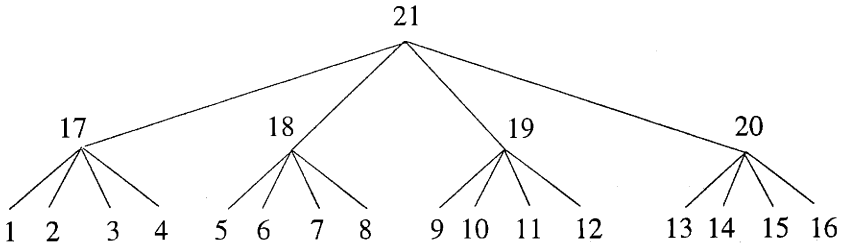


Figure 3-8: Correlation tree for a unit energy minimum distance signature set of 21 users in 16 dimensions.

to interfere. From Figure 3-9 we see negligible performance degradation for leaks of -30 dB and -40 dB and less than 1dB degradation due to the -20 dB leaks. A 1dB performance degradation means that all users in the -20 dB leaky system would need to increase their signal energy to noise energy ratio by 1dB (or by 26%) per bit transmitted to achieve the same bit error rate of the strictly-tree-structured system. The degradations from the strictly tree-structured system to the nearly tree-structured systems are approximately the same as for an orthogonal versus nearly-orthogonal systems for which adjacent channel leaks are treated as noise ([13]).[22]

The average bit error rate curves for the leaky equal energy minimum distance set of 21 users in 16 dimensions are shown in Figure 3-10. For 21 users in 16 dimensions, a -40 dB leak goes virtually unnoticed and a -30 dB leak causes only negligible performance loss. A -20 dB leak in each user results in a 1.25 dB performance loss. In other words, each user in the -20 dB leaky system would need to increase its signal energy to noise energy ratio by 1.25 dB (or 33%) to regain the performance of a strictly tree-structured system. For comparison, a system of nearly orthogonal users experiences a loss of approximately 1 dB (26%) relative to a strictly orthogonal system.[13]

3.7.2 Experimental Analysis of Phase Mismatch

In an actual MA system a user, say user k, is assigned an envelope waveform, $s_k(t)$, by the base station. The user modulates this envelope with a cosine wave having a carrier frequency, w_c, that was also assigned by the base station. The user's oscillator will have a phase, ϕ_k, relative to the pulse, $s_k(t)$, which is not controllable. The user's signature waveform may be represented as

$$s_k(t)\, cos(w_c t + \phi_k).$$

[22]Here, it should be noted that all simulations in this thesis are done with equal energy users. Study of non-ideal scenarios on the tree joint detector for unequal energy users is left for future work.

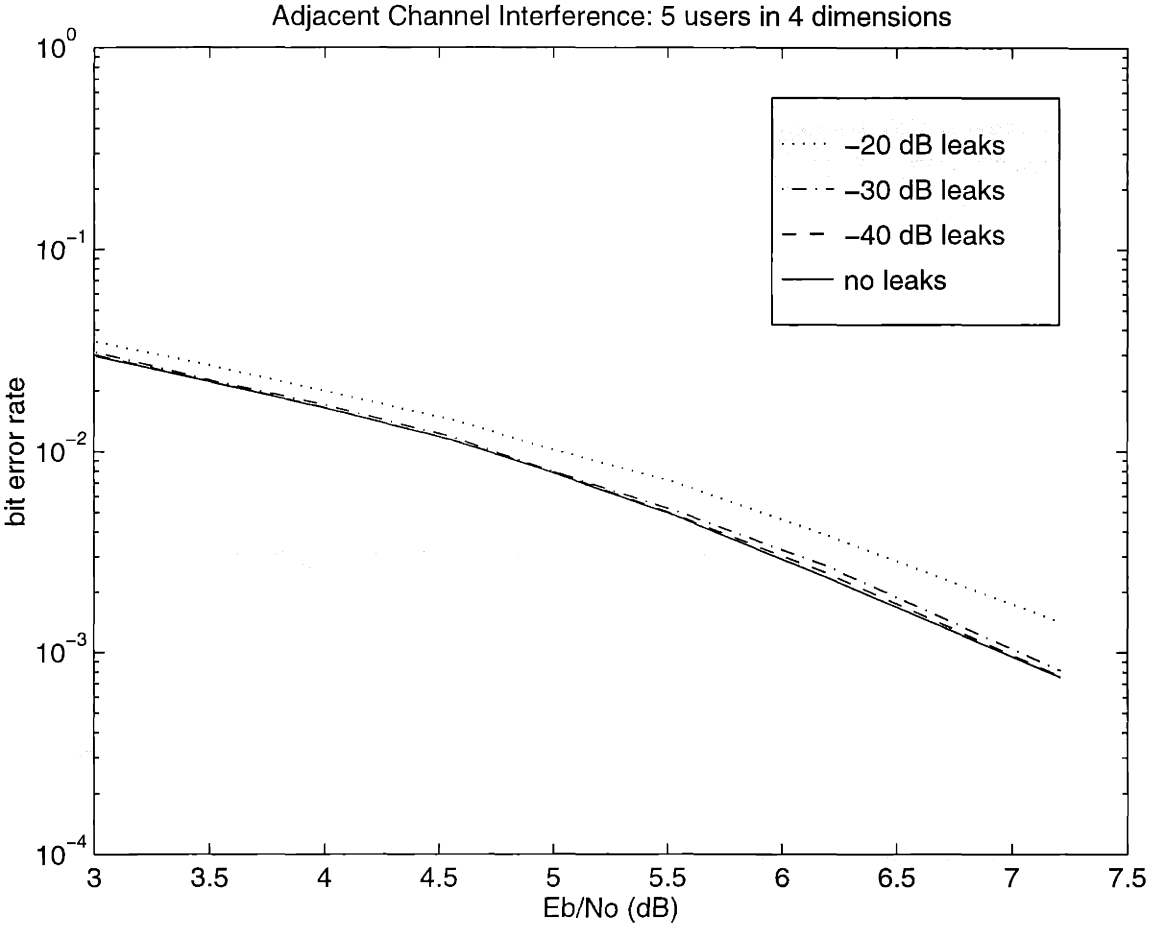


Figure 3-9: Average bit error rate for a minimum distance set of 5 users in 4 dimensions for which sibling users are not strictly orthogonal. Each user is allowed to have a 0.1 (−20 dB), 0.036 (−30 dB), or 0.01 (−40 dB) cross-correlation with its neighboring sibling user on the tree. One standard deviation error bars (not shown) are between 2% and 3% of BER.

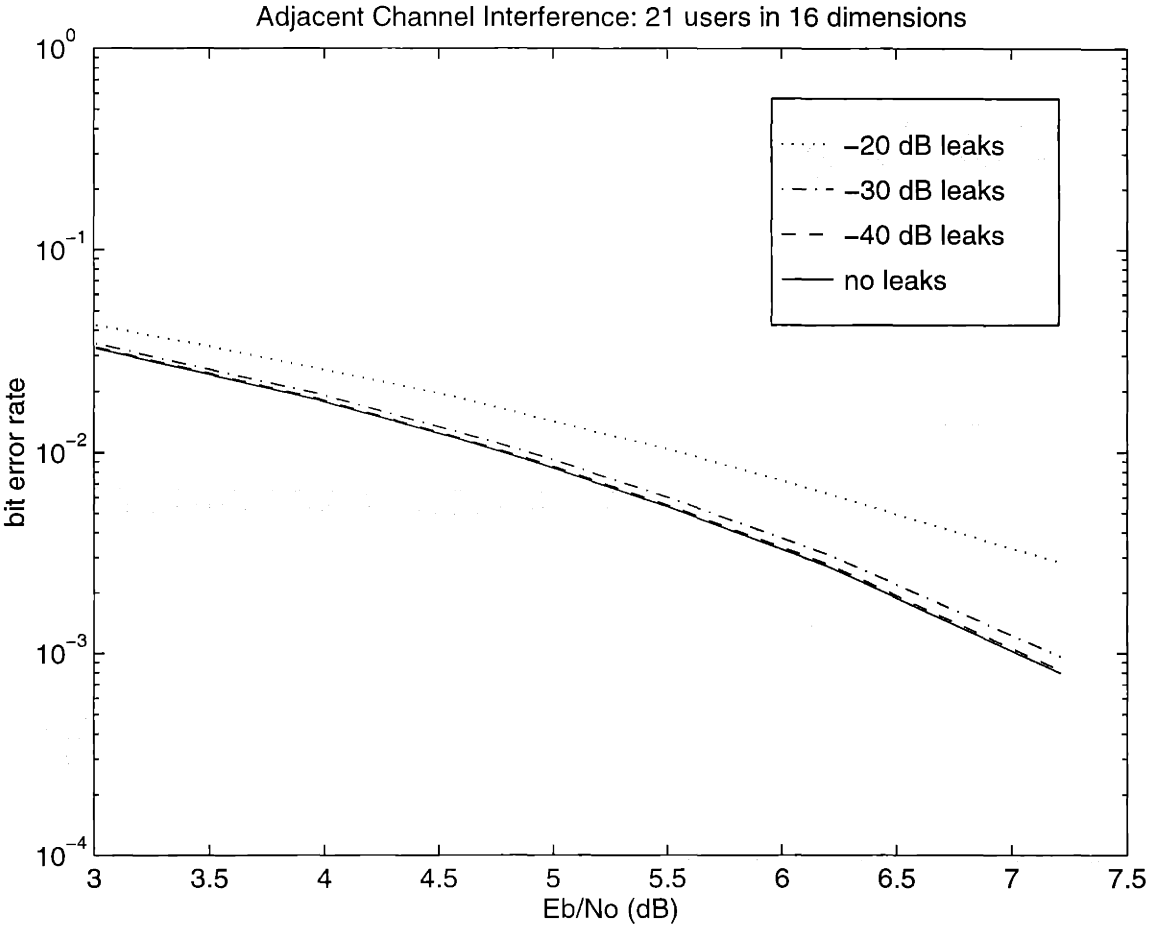


Figure 3-10: Average bit error rate for a minimum distance set of 21 users in 16 dimensions for which sibling users are not strictly orthogonal. Each user is allowed to have a 0.1 (−20 dB), 0.036 (−30 dB), or 0.01 (−40 dB) cross-correlation with its neighboring sibling user on the tree. One standard deviation error bars are not shown; they are between 1% and 1.2% of BER.

In practice, the received power of each user can be estimated to a high degree of accuracy by the satellite. The user's power can be considered to be constant since it will either remain constant throughout transmission or vary very slowly relative to the symbol repetition period ([2]). The phase, on the other hand, is less certain at the receiver. The estimate of the phase will typically have a non-negligible variance. Furthermore, it can drift over many symbol durations. This section examines the effect of having incorrect knowledge of a user's phase at the receiver.[23] Through simulations we find that the tree joint detector (for equal energy minimum distance sets) can tolerate up to a 15° phase error in one user, resulting in negligible increases in bit error rates, and can accommodate up to a 20° phase error in one user with small performance degradations.

An interesting scenario for the examination of phase mismatch is with the minimum distance set of 21 users in 16 dimensions. The 21 user tree is shown in Figure 3-8. Figure 3-11 shows the bit error rate averaged over all 21 users for all users having perfectly known phases at the receiver, and for the case of user 17 having a 10°, 15°, and a 20° phase mismatch while the other users have no mismatch. From the figure we see that a substantial phase error of 20° causes a performance loss of 0.5 dB.[24]

A user in such a system cares more about the worst case bit error rate than it does about the average bit error rate. Figure 3-12 shows several bit error curves for users that are effected by the mismatch of user 17's phase. From Figure 3-12-(a) we see that user 17, the user with phase mismatch, experiences a 0.5 dB loss for the 20° phase mismatch and negligible loss for the 15° and 10° mismatches. As can be seen from Figure 3-12-(f), user 21, the parent of user 17, experiences the most loss at 1dB for the 20° mismatch, 0.5 dB for the 15° mismatch, and nearly unnoticeable loss for the 10° phase mismatch on user 17.

[23] A thorough treatment of incorporating phase estimation into the tree detector is done in Chapter 4.

[24] Each of the users would need to increase their E_b/N_0 by 12% to regain the bit error rate of the corresponding system having no phase mismatch.

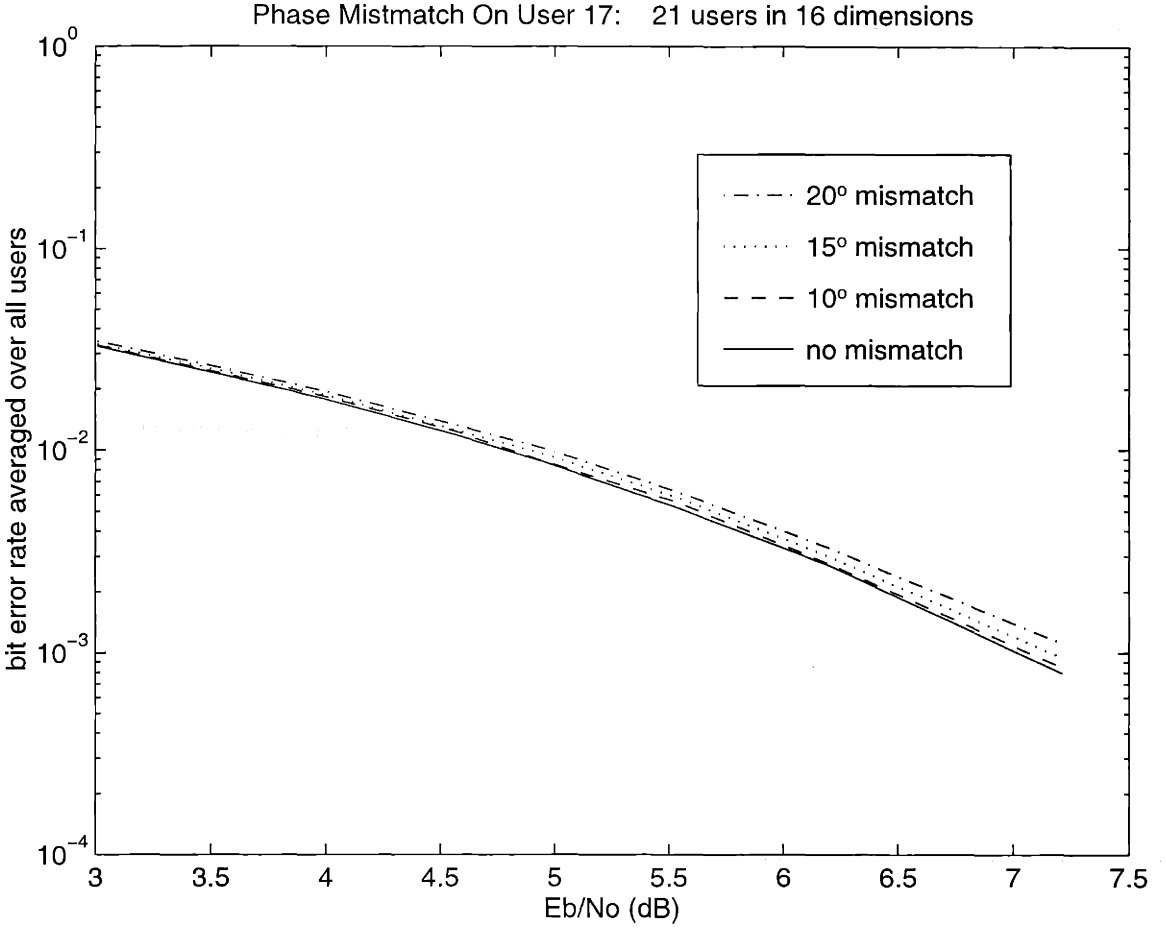


Figure 3-11: Average bit error rate for a set of 21 users in 16 dimensions for which user 17's phase is incorrectly known at the receiver. All other user signature waveforms are entirely known at the receiver. One standard deviation error bars on the BER points in the figure are not shown; they are between 1% and 1.2% of BER.

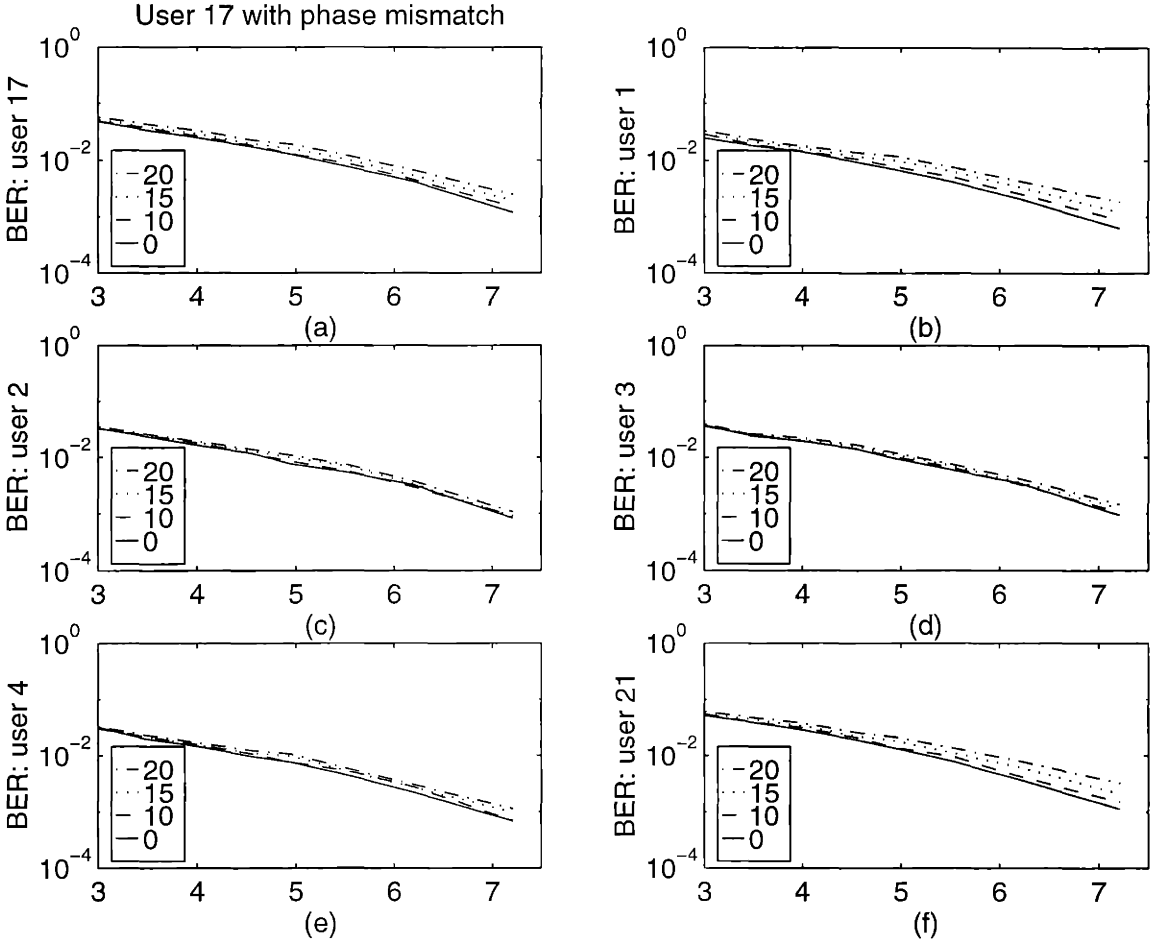


Figure 3-12: Bit error rates for individual users in a set of 21 users in 16 dimensions for which user 17's phase is incorrectly known at the receiver. One standard deviation error bars on the BER curves are not shown; they are between 4.5% and 5.5% of BER. (a) user 17, the user with phase mismatch. (b) user 1, a child of user 17. (c) user 2, a child of user 17. (d) user 3, a child of user 17. (e) user 4, a child of user 17. (f) user 21, the parent of user 17.

Table 3.6: Approximate performance loss seen by users 1, 2, 3, 4, 17 and 21. The loss due to a mismatch on user 17's phase is relative to the performance for the case in which all phases are known.

user number	20°	15°	10°
user 1	0.75 dB	0.5 dB	0.15 dB
user 2	0.15 dB	0 dB	0 dB
user 3	0.25 dB	0.15 dB	0 dB
user 4	0.50 dB	0.50 dB	0 dB
user 17	0.50 dB	0.25 dB	0.15 dB
user 21	1.00 dB	0.50 dB	0.15 dB

From the remaining plots in Figure 3-12 we see that each of the children of user 17 experiences a different degree of loss. In other words, the presence of an unknown phase on user 17 does not cause the same degradation for users 1, 2, 3, and 4. User 1, in Figure 3-12-(b), shows a 0.75 dB loss for a 20° mismatch, a 0.5 dB loss for a 15° mismatch, and less than 0.25 dB loss for a 10° mismatch on user 17's phase. The losses for the rest of the child users are summarized in Table 3.6.

The variation in loss among the children of the unknown-phase user is to be expected since each user signature of the minimum distance set has been multiplied by an arbitrary value of phase to emulate an actual system. This step causes slightly difference performance from user to user. Notice that the 0° BER curve for user 1 in Figure 3-12-(b) is lower than the corresponding curves for users 2 and 3 in Figures 3-12-(c) and (d). Also notice that the 20° curve in Figure 3-12-(b), for user 1, is nearly equivalent to the 20° curve for user 3 in Figure 3-12-(c). Part of the degradation seen by user 1 due to user 17's mismatch in phase is due to the change in the relationships among user signatures resulting from a change in user 17 phase. In other words, the arbitrary assignment of phases for each user gave user 1 a slight advantage; changing this by assigning user 17 a phase that is 20° greater than its original assignment would change the relationship among user signatures, perhaps taking away user 1's original advantage. All other users in the tree exhibit no noticeable performance loss;

bit error curves for these remaining users are not shown here.

Simulations were also run for the same 21 user signature set, but with a phase mismatch in user 21 at the top of the tree. Figure 3-13 shows average bit error curves for user 21 having the phase mismatch. Not shown here, plots of each user's bit error curve revealed similar behavior to that described above for the case of user 17 having the mismatch. User 21, the user with the phase mismatch for this simulation suffered only a 0.5 dB degradation for the 20° case, a negligible degradation for a 15° mismatch, and an almost unnoticeable degradation for the 10° mismatch. The performance loss seen by user 21's children, users 17, 18, 19 , and 20, due to a 20° mismatch in user 21's phase ranged from virtually no loss to approximately 0.75 dB loss. The bottom users experienced the smallest loss, on average, where the worst case loss was only 0.25 dB for the 20° mismatch on user 21's phase.

Figure 3-14 shows the average bit error curves for a corresponding simulation to the previous ones, but with the mismatch in user 1's phase. As might be expected from the low tree-position of user 1, the mismatch on user 1 affects fewer users, hence the degradation in bit error rates is less than that in the previous simulations for which the mismatched phase user is higher up in the tree. Again, plots of each user's bit error curves (not shown here) revealed that only users 1 and 17 were affected by the mismatch on user 1. The siblings to user 1, users 2 and 3 were slightly affected, exhibiting a 0.25 dB performance loss in the 20° mismatch case. All other users, including the root user (the grandparent to user 1) showed no loss, even for the 20° mismatch in user 1's phase.

The more common scenario of having small mismatches in *all* users phases was also simulated. Figure 3-15 shows the average bit error curves for a 5° mismatch, a 10° mismatch, and no mismatch. From Figure 3-15 we see a 0.75 dB (19%) average performance loss for a 10° mismatch in all user phases, and a mere 0.13 dB (3%) loss for a mismatch of 5°. For near lossless performance, then, the tree joint detector requires better than a $\pm 5°$ certainty in the knowledge of each user's phase.

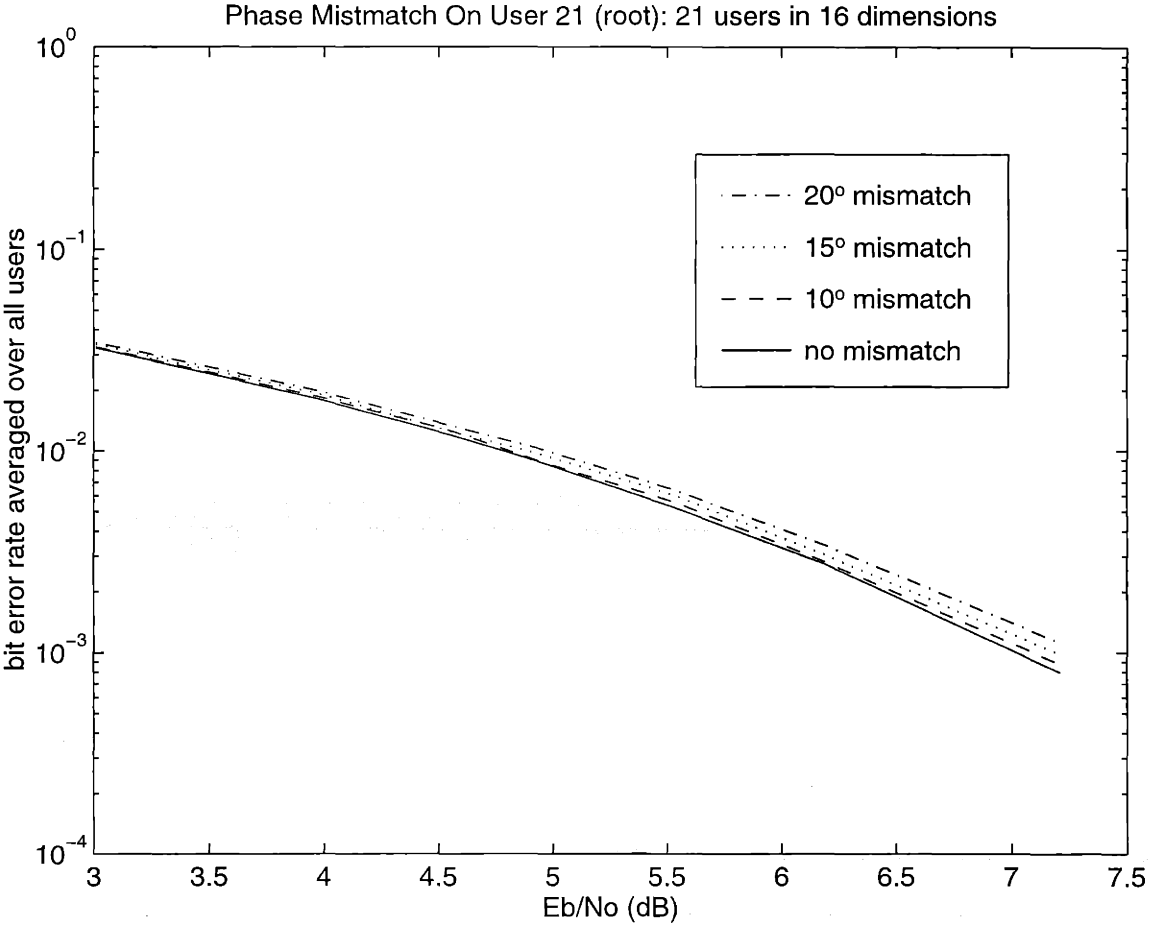


Figure 3-13: Average bit error rate for a set of 21 users in 16 dimensions for which user 21's phase is incorrectly known at the receiver. All other user signature waveforms are entirely known at the receiver. One standard deviation error bars (not shown) are between 1% and 1.2% of BER.

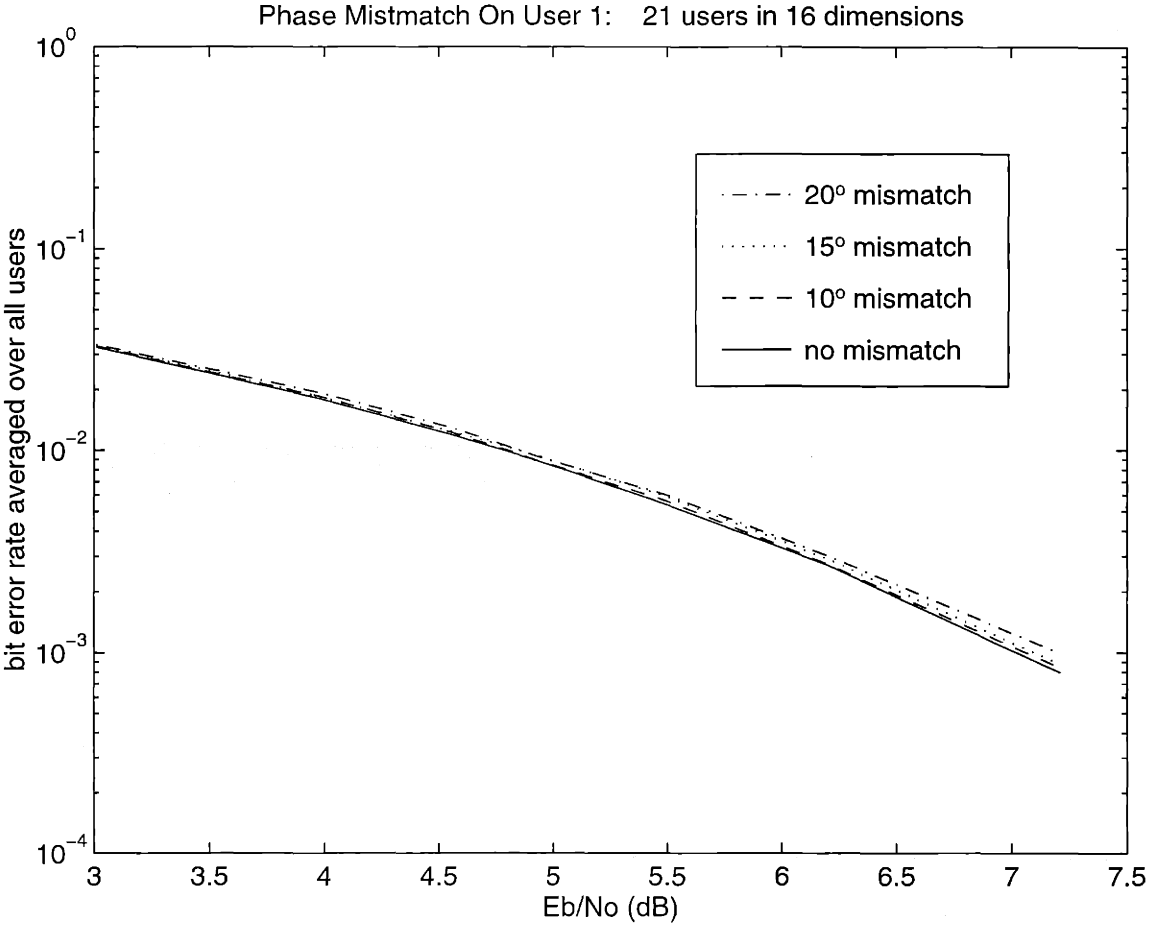


Figure 3-14: Average bit error rate for a set of 21 users in 16 dimensions for which user 1's phase is incorrectly known at the receiver. All other user signature waveforms are entirely known at the receiver. One standard deviation error bars are not shown; they are between 1% and 1.2% BER.

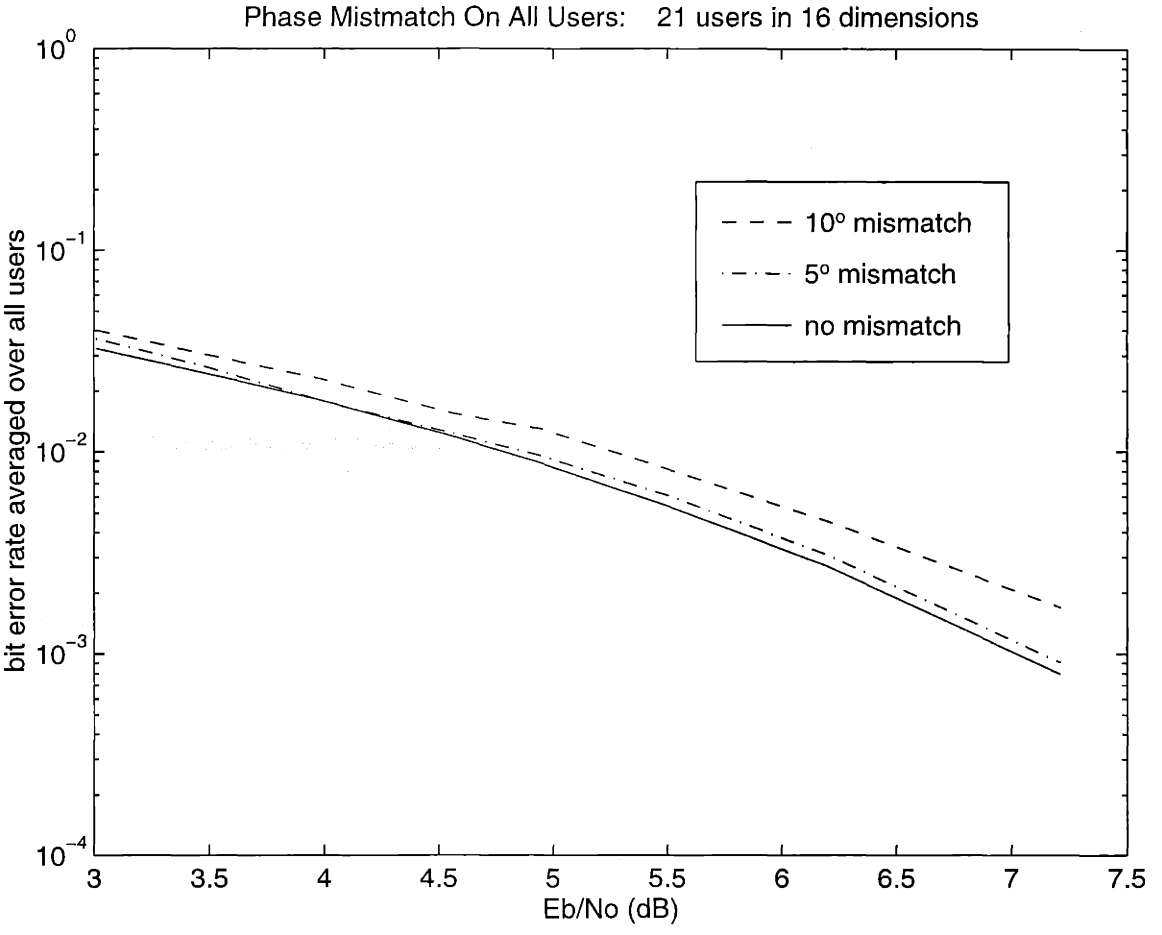


Figure 3-15: Average bit error rate for a set of 21 users in 16 dimensions for which all users' phases are incorrectly known at the receiver. The dashed line corresponds to all users having a 10° phase mismatch, the dash-dot line corresponds to a 5° mismatch, and the solid line corresponds to all users having known phases (no mismatch). The standard deviation on the BER calculations range from 0.5% to 4.4% BER.

In the next chapter, the estimation of an unknown phase is incorporated into the tree detection procedure. As indicated by the results for the case of a small mismatch on all users' phases, once a phase is estimated to within 5°, the estimate can be taken to be true for many subsequent symbol transmissions.[25]

[25]Recall that it is common for there to be a relatively slow drift in a user's phase. The user's phase will need to be re-estimated periodically, but the frequency of the need for a phase estimate update is far lower than the symbol transmission rate.

Chapter 4

One-shot Weight/Phase Estimation

In a MA system, a set of channels can be set aside for interaction between the base station and the users. Specifically, if a new user wishes to obtain a place in the system, it makes a request to the base station over this separate acquisition channel. If the system can accommodate this new user, the base station assigns this user a signature waveform (which might consist of a signature envelope, modulation frequency, and symbol timing). During the acquisition process, the base station refines any timing or power demands it might have on this new user by sending control signals telling the new user to make the necessary adjustments.[1] Although the phase of the new user's transmission can be estimated during acquisition, this phase will not necessarily remain the same once the user begins transmitting at it's assigned modulation frequency. It is possible that the phase of this new user will be unknown to the receiver once communication commences.[2]

Joint detection, in general, requires comparison of the actual received signal with the set of possible received signals. The set of possible received signals is constructed by the detector using replicas of the actual received signature signals that are stored at

[1]This procedure is typical of state of the art military satellite communications. For more information on satellite communications, see [5].

[2]Coherent changeover from the acquisition channel to the transmission channel would require a broad band, continuous-phase transmitter, which may be difficult to implement ([2]).

the receiver. If the set of signature waveforms stored at the receiver do not faithfully replicate the actual received signature waveforms, the performance, in general, of a joint detector is degraded.

The performance tradeoff with varying degrees of phase error was studied for the optimal tree joint detector via simulations in Section 3.7.2. The best performance occurs when the receiver has perfect information about the users' phases. As expected, as the error between the actual and assumed user phases grows, the performance of the tree detector of Chapter 3 degrades. The simulation results of Section 3.7.2 indicate that a phase error of 5° (10°) on all users will cause a performance loss 0.13 dB (0.75 dB) relative to the case in which all users' phases are accurately known by the coherent detector. For tree-structured MA that used the tree detector, it is important, then, to estimate each user's phase with no worse than a 5° accuracy. Once an estimate for a given user is found to meet this requirement, the phase estimate can be taken to be true for subsequent symbol transmissions for this user. This chapter incorporates phase estimation into the tree detector.

In general, there are three conditions for which a communications system may be designed. The first is the condition that the user's phase is known; this condition results in a **coherent** system.[3] The second is the condition that the user's phase is unknown; this condition results in a **non-coherent** system.[4] The third is the condition for which partial phase information is available; this condition results in a **partially coherent** system.[5] The tree detector in Chapter 3 was developed for the

[3]An example of a coherent single user system is one that employs phase shift keying (PSK). If binary signaling is used, a phase of 0 radians corresponds to an information bit of 0 and a phase of π radians corresponds to an information bit of 1. Any system employing PSK must be capable of knowing the phase of the received signal.

[4]An example of a non-coherent single user system is one that employs frequency shift keying (FSK). If binary signaling is used, two disjoint bands of frequency are reserved for a single user. The user transmits in one band to communicate an information bit of 0 and in the other band to communicate an information bit of 1. The detector monitors each band for the energy per symbol time slot to decide which bit is being sent.

[5]If an estimate of a user's phase is fed to the receiver, the receiver should use the estimate, but only to the degree of its accuracy. See the paper by Viterbi, [25], for a detailed derivation of single user partially coherent detection.

case in which all phases were assumed to be known and will, therefore, be referred to as the **coherent tree joint detector** for the remainder of this thesis.

The specific scenario addressed in this chapter is the case in which one user has an unknown or uncertain phase and all other users' phases are known exactly. Focusing on this scenario is motivated by the dynamic nature of a MA communication system. Users drop in and out of a system, one at a time, hence the newest user's phase will need to be estimated upon its entry to the system. The training protocol of MA communication systems of present is also adopted in this thesis. Specifically, a new user will begin with the transmission of a previously agreed upon training sequence of weights so that the receiver can concentrate only on estimating this user's phase. It can reasonably be assumed that an estimate of this new user's phase is found prior to the addition of another user to the system.

This chapter develops low complexity optimal and sub-optimal one-shot algorithms for the non-coherent and partially coherent cases to be used with tree-structured user signature sets.[6] The ultimate goal at the receiver is to achieve a phase estimate that is in error of, at most, 5°. Although the one-shot phase estimation algorithms developed in the chapter are not capable of achieving such accuracy, they will serve as the foundation for several estimation procedures of Chapter 5 that make use of a sequence of symbol transmissions.

In Section 4.1 the **non-coherent weight/phase estimator** is derived for the general case in which no specific structure is present among user signatures. Section 4.2 shows the computational reduction of this estimator realized for signature sets exhibiting tree-structure. An example is also given to illustrate the non-coherent weight/phase estimation procedure which is only a slight variation from the coherent tree joint detector of Chapter 3. Section 4.3 offers a formal derivation of the complexity of the non-coherent tree weight/phase estimator. Section 4.4 develops the

[6] Recall that the term one-shot means that weight and phase estimates are made with data from one symbol duration. In contrast, estimation can be done using data from a sequence of symbols. The sequence weight/phase estimator is developed and analyzed in Chapter 5.

partially coherent weight/phase estimator that incorporates partial information about the new user's phase. If an estimate of the phase is not desired, a **partially coherent weight-only estimator** may be used. This is derived in Section 4.5. Both the non-coherent and partially coherent estimators have a complexity that can range from very low order polynomial in the number of users, to exponential in the number of users, depending upon the position in the tree of the new, uncertain-phase user. As was seen in Section 3.7.2, one low complexity option is to ignore the phase uncertainty and use the coherent tree joint detector with an incorrect value for the phase; this works well given less than a 15° inaccuracy in the phase of one user, only if all other users' phases are known exactly. In Section 4.6.1, this **assumed coherent joint weight detector** is formally presented and intuition is given to explain the simulations results of Section 3.7.2. Since, ultimately, the receiver needs to know all users' phases within a 5° error, in general, use of the AC detector to avoid the complexity of the optimal estimator is not recommended. Section 4.6.2 proposes a sub-optimal version of the optimal phase estimator, namely, the **assumed discrete joint weight/phase estimator**, which achieves low complexity by assuming that the known phase is a *discrete* random variable. The weight error performance and the phase estimate accuracy for the optimal estimators derived in this chapter is explored through simulations in Section 4.7.1 and Section 4.7.2, respectively.

4.1 Optimal Non-coherent Weight/Phase Estimator

In this section the maximum a posteriori (MAP) joint weight/phase estimator is derived for the case in which one user's phase is not known at the receiver. To be consistent with the single user literature, the MA joint estimator derived in this section will be called the **non-coherent (NC) joint weight/phase estimator**. No special structure among user signature waveforms is assumed for the derivation. The

received signal is modeled by the complex vector equation[7]

$$\mathbf{r} = \sum_{k=1}^{K-1} b_k \mathbf{s}_k e^{j\phi_k} + \mathbf{s}_K e^{j\phi_K} + \sigma\mathbf{n} \tag{4.1}$$

where the assigned N-dimensional user signature vectors, $\{\mathbf{s}_k\}_1^K$, are now real. Users $1, 2, \ldots, (K-1)$ have arbitrary, but known, phases, $\{\phi_k\}_1^{K-1}$. The noise vector, $\mathbf{n}$, is complex with elements independently and identically distributed as circularly Gaussian with unit variance. Notice that user K has no weight, i.e., $b_K = +1$. This reflects the protocol of MA communication systems of present, in that a new user joining the system transmits a previously agreed upon training sequence of weights. For this derivation of the one-shot joint weight/phase estimator the training weight is assumed to be $b_K = +1$, without loss of generality. The following, equivalent matrix representation of Equation (4.1) is more convenient:

$$\mathbf{r} = \mathbf{S}\Phi\mathbf{b} + \mathbf{s}_K e^{j\phi_K} + \sigma\mathbf{n}. \tag{4.2}$$

$\mathbf{S}$ now has as its columns the $K-1$ user signature vectors that correspond to the users with known phase. The known phases are collected in $\Phi = diag[\{e^{j\phi_k}\}_1^{K-1}]$. The (K-1)-element information weight vector, $\mathbf{b}$, has a finite number of possible realizations enumerated in the discrete set Γ. For this section in which we assume no knowledge of the true value of user K's phase, ϕ_K is modeled as a uniformly distributed random variable:[8]

$$p_{\phi_K}(\phi) = \frac{1}{2\pi}, \quad -\pi \leq \phi \leq \pi. \tag{4.3}$$

In the absence of noise, the received signal vector is

$$\mathbf{r}' = \mathbf{S}\Phi\mathbf{b} + \mathbf{s}_K e^{j\phi_K}.$$

[7]This model replaces that of Equation (2.1) so that the users' phases may be explicitly expressed.
[8]Note that this is the worst case prior distribution on ϕ_K. Section 4.4 uses a different prior that reflects varying degrees of accuracy of the receiver's knowledge of the new user's phase.

The received constellation, or range of the complex random vector, $\mathbf{r}'$, in the complex hyper-plane, is the union of $|\Gamma|$ rings, one ring for each possible realization of $\mathbf{b}$.

For example, if there were only two users present in one complex dimension,

$$r' = b_1 s_1 e^{j\phi_1} + s_2 e^{j\phi_2},$$

then the range of r' would be two rings in the real and imaginary plane. A two user, binary signaling example is shown in Figure 4-1.[9] In this example, the phase of user 1 is assumed to be known at 0 radians and the signature signal for user 1 is depicted as a complex scalar having a phase of zero and is shown in Figure 4-1-(a). The phase of user 2 is unknown so ϕ_2 is assumed to be uniformly distributed between $-\pi$ and π. The complex scalar signature for user 2, then, is only known up to its length, therefore, it could lie anywhere on the ring in Figure 4-1-(b). For this example, user 1 may transmit a weight of $b_1 = +1$ or $b_1 = -1$. Recall that user 2, the user with the unknown phase, must transmit a training weight of $b_2 = +1$. It follows that the range of r' for this example is the union of two rings as shown in Figure 4-1-(c): one ring centered at $\mathbf{s}_1$ (for $b_1 = +1$, the right ring) and one ring centered at $-\mathbf{s}_1$ (for $b_1 = -1$, the left ring).

The actual received value is

$$r = r' + \sigma n,$$

where n, again, is a complex scalar Gaussian random variable that has identically and independently distributed real and imaginary parts. In other words, n is circularly Gaussian in the complex plane. The range for r is the entire complex plane. The probability density the function for r is illustrated via the 2-D density cloud shown in Figure 4-1-(d). The density cloud qualitatively shows that r is more likely to be

[9] Any illustration of the range of $\mathbf{r}'$ for examples requiring more than 1 complex dimension is not done in this thesis. To show even the simplest example, say, with three users in two complex dimensions, one user's phase unknown, would require at least three real dimensions. Such an example would have a range of $\mathbf{r}'$ comprised of 2-D rings in three dimensions and is too confusing to show. This two user example, however, does offer some intuition that can be extended to higher dimensions.

near a point on the two rings of Figure 4-1-(c) and less likely to be far from the rings.

The general MAP joint weight/phase estimator chooses the values for $\mathbf{b} \in \Gamma$ and $-\pi \leq \phi_K \leq \pi$ which maximize the a posteriori probability density function (PDF) for $\mathbf{b}$ and ϕ_K, given the received vector, $\mathbf{r}$,

$$p_{\mathbf{b},\phi_K|\mathbf{r}}(\mathbf{b},\phi|\mathbf{r}) = p_{\mathbf{r}|\mathbf{b},\phi_K}(\mathbf{r}|\mathbf{b},\phi)\frac{Prob(\mathbf{b})p_{\phi_K}(\phi)}{p_{\mathbf{r}}(\mathbf{r})},$$

where $\mathbf{b}$ and ϕ_K are statistically independent. We are maximizing the above PDF over $\mathbf{b}$ and ϕ_K. Note that $p_{\mathbf{r}}(\mathbf{r})$ is not a function of these parameters. In addition $Prob(\mathbf{b})$ is assumed to be the same for every realization of $\mathbf{b}$, and $p_{\phi_K}(\phi)$ is uniform. Under these conditions, the MAP estimator will give the same estimate found by the maximum likelihood (ML) estimator which maximizes $p_{\mathbf{r}|\mathbf{b},\phi_K}(\mathbf{r}|\mathbf{b},\phi)$ over $\mathbf{b}$ and ϕ_K. We can easily write $p_{\mathbf{r}|\mathbf{b},\phi_K}(\mathbf{r}|\mathbf{b},\phi)$ as

$$p_{\mathbf{r}|\mathbf{b},\phi_K}(\mathbf{r}|\mathbf{b},\phi) = (\frac{1}{\sqrt{2\pi}\sigma})^{2N}\exp(-\frac{1}{2\sigma^2}[||\Re\{D(\mathbf{b},\phi)\}||^2 + ||\Im\{D(\mathbf{b},\phi)\}||^2]). \quad (4.4)$$

The difference, denoted by $D(\mathbf{b},\phi)$, is

$$D(\mathbf{b},\phi) = (\mathbf{r} - \mathbf{S}\Phi\mathbf{b} - \mathbf{s}_K e^{j\phi}), \quad (4.5)$$

where $\Re(\cdot)$ and $\Im(\cdot)$ are the real and imaginary operators. The optimal joint weight/phase estimator is given by

$$\{\hat{\mathbf{b}},\hat{\phi}_K\} = \arg\max_{\mathbf{b},\phi} p_{\mathbf{r}|\mathbf{b},\phi_K}(\mathbf{r}|\mathbf{b},\phi), \quad \mathbf{b} \in \Gamma,\ -\pi \leq \phi \leq \pi \quad (4.6)$$

or

$$\{\hat{\mathbf{b}},\hat{\phi}_K\} = \arg\max_{\mathbf{b},\phi} \ln[p_{\mathbf{r}|\mathbf{b},\phi_K}(\mathbf{r}|\mathbf{b},\phi)], \quad \mathbf{b} \in \Gamma,\ -\pi \leq \phi \leq \pi.$$

Discarding constant terms and constant negative multipliers from the function to be

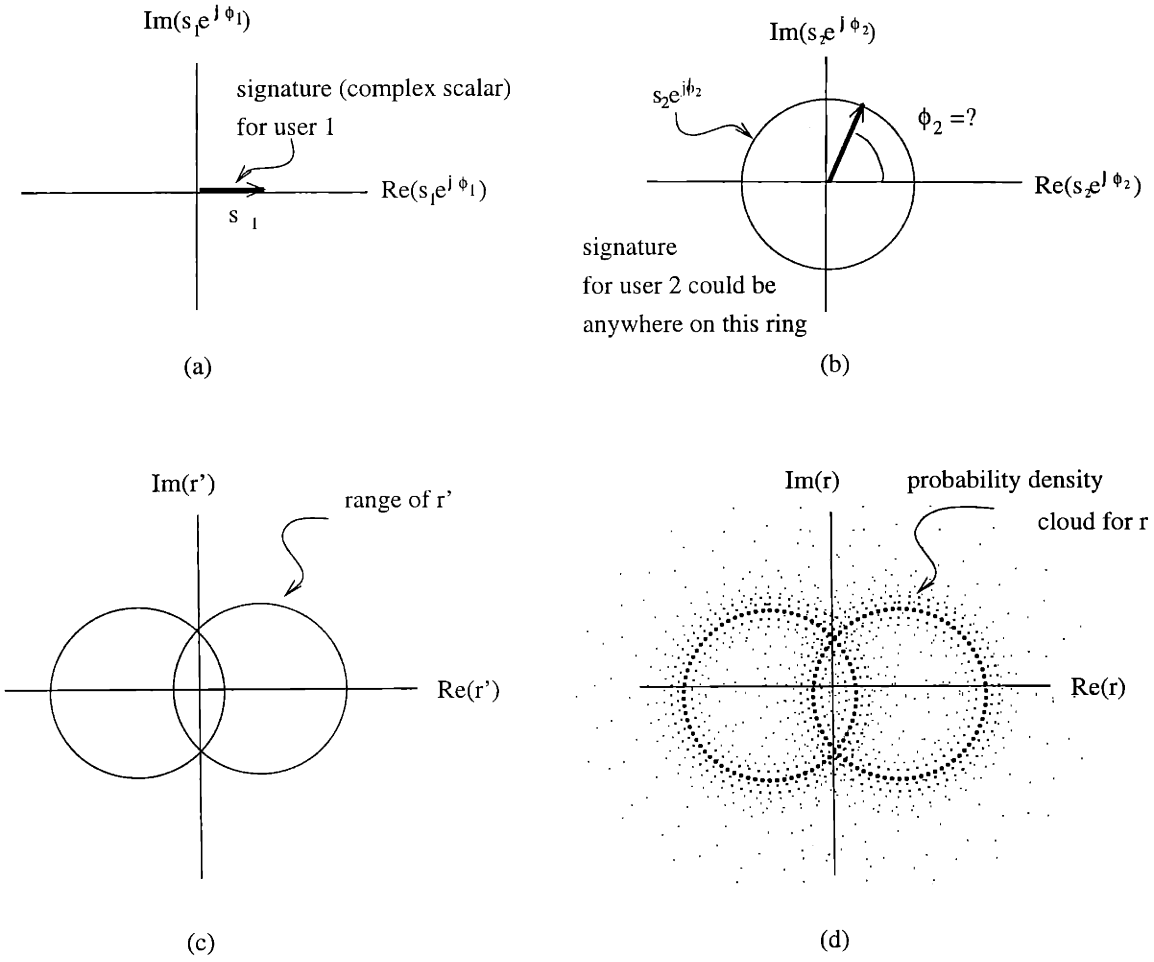


Figure 4-1: Example of two users in one complex dimension. User 1 has a known phase (0 radians) and user 2 has a uniformly distributed phase. (a) The signature of user 1, $\mathbf{s}_1$, is depicted as a complex scalar having only a real part. (b) User 2 has a uniformly distributed phase, hence, its signature (complex scalar) could lie anywhere on the ring. (c) Set of all possible points, $r' = b_1 s_1 + s_2 e^{j\phi_2}$, $b_1 \in \{+1, -1\}$, $\phi_2 \in [-\pi, \pi]$. (d) Probability density function for $r = r' + \sigma n$, n circularly Gaussian, depicted qualitatively as a density cloud.

maximized leads us to the minimization

$$\{\hat{\mathbf{b}}, \hat{\phi}_K\} = \arg \min_{\mathbf{b},\phi} \Lambda_1(\mathbf{b}, \phi|\mathbf{r}), \quad \mathbf{b} \in \Gamma, \ -\pi \leq \phi \leq \pi, \tag{4.7}$$

$$\Lambda_1(\mathbf{b}, \phi|\mathbf{r}) = ||\Re\{D(\mathbf{b}, \phi|\mathbf{r})\}||^2 + ||\Im\{D(\mathbf{b}, \phi|\mathbf{r})\}||^2. \tag{4.8}$$

The ML joint weight/phase estimator may be interpreted as follows. The optimal detector defined by Equations (4.7) and (4.8) finds the point, $\hat{\mathbf{r}}'$, in the range of the random variable $\mathbf{r}' = \mathbf{S}\Phi\mathbf{b} + \mathbf{s}_K e^{j\phi_K}$ that is closest[10] to the realization of the random variable, $\mathbf{r}$. This closest point, $\hat{\mathbf{r}}'$, lies on one of the rings in the range of $\mathbf{r}'$. The value of $\mathbf{b}$ that corresponds to this ring would constitute $\hat{\mathbf{b}}$. Moreover, the weight estimate is correct, i.e., $\hat{\mathbf{b}} = \mathbf{b}_{true}$, if $\hat{\mathbf{r}}'$ lies on the same ring as does $\mathbf{r}'_{true}$. The phase estimate, $\hat{\phi}_K$, is the angle of the complex vector $\hat{\mathbf{r}}' - \mathbf{S}\Phi\hat{\mathbf{b}}$.

The two user binary example in Figure 4-1 can be used to illustrate this procedure carried out by the optimal non-coherent joint estimator. Recall that if there were only two users present in one complex dimension, one user's phase known to be zero and one user's phase unknown, the range of $r' = b_1 s_1 + s_2 e^{j\phi_2}$ would be the two rings in the real and imaginary plane as shown in Figure 4-1-(c). Now, consider as an example of r, the received complex scalar that is shown in Figure 4-2-(a) as a 2-D vector with one dimension corresponding to the real part and one dimension corresponding to the imaginary part. The optimal non-coherent joint estimator for b_1 and ϕ_2 would project r onto the range of r', thus, finding the closest possible transmitted value, $\hat{r}'$, to the received value, r. This projection is shown in Figure 4-2-(b). The estimate $\hat{b}_1$ would correspond to the ring on which $\hat{r}'$ lies. In the example of Figure 4-2, $\hat{r}'$ lies on the left ring. The left ring corresponds to $b_1 = -1$, hence $\hat{b}_1 = -1$. For this simple example, the decision regions for $\hat{b}_1$ given any possible value of the received complex scalar, r, are shown in Figure 4-2-(c). Specifically, if the received signal value, r, were to lie in one of the pink (green) regions, the optimum non-coherent weight/phase estimator

[10] Closest, in terms of Euclidean distance.

would assign $\hat{b}_1 = +1$ ($\hat{b}_1 = -1$). The phase estimate for ϕ_2 is, simply, the phase of $\hat{r}' - \hat{b}_1 s_1$. In this example, $\hat{\phi}_2 = \frac{\pi}{3}$, as shown in Figure 4-2-(d).

The above discussion offered a conceptual interpretation of the optimal non-coherent joint weight/phase estimator. In practice, we must find an algorithm to carry out the detector of Equations (4.7) and (4.8). Substituting Equation (4.5) into Equation (4.8) and expanding gives

$$\begin{aligned}
\Lambda_1(\mathbf{b}, \phi|\mathbf{r}) &= ||\Re\{\mathbf{r} - \mathbf{S}\Phi\mathbf{b}\}||^2 + ||\Im\{\mathbf{r} - \mathbf{S}\Phi\mathbf{b}\}||^2 && (4.9)\\
&\quad + ||\Re\{\mathbf{s}_K e^{j\phi}\}||^2 + ||\Im\{\mathbf{s}_K e^{j\phi}\}||^2 && (4.10)\\
&\quad - 2\Re\{\mathbf{s}_K e^{j\phi}\}^T \Re\{\mathbf{r} - \mathbf{S}\Phi\mathbf{b}\} - 2\Im\{\mathbf{s}_K e^{j\phi}\}^T \Im\{\mathbf{r} - \mathbf{S}\Phi\mathbf{b}\}. && (4.11)
\end{aligned}$$

The term on line (4.9) is

$$F_K(\mathbf{b}|\mathbf{r}) = ||\Re\{\mathbf{r} - \mathbf{S}\Phi\mathbf{b}\}||^2 + ||\Im\{\mathbf{r} - \mathbf{S}\Phi\mathbf{b}\}||^2,$$

or

$$\boxed{F_K(\mathbf{b}|\mathbf{r}) = ||\mathbf{r} - \mathbf{S}\Phi\mathbf{b}||^2} \qquad (4.12)$$

where $F_K(\mathbf{b}|\mathbf{r})$ is, simply, the energy of the residual obtained from subtracting all but the K^{th} received signal from the actual received signal. $F_K(\mathbf{b}|\mathbf{r})$ is a function of $\mathbf{b}$, the value of the weights corresponding to users $1, 2, \ldots, (K-1)$.

The term on line (4.10) is

$$\varepsilon_K = ||\Re\{\mathbf{s}_K e^{j\phi}\}||^2 + ||\Im\{\mathbf{s}_K e^{j\phi}\}||^2 = ||\mathbf{s}_K e^{j\phi}||^2. \qquad (4.13)$$

This is the received energy due to the K^{th} user's transmission.

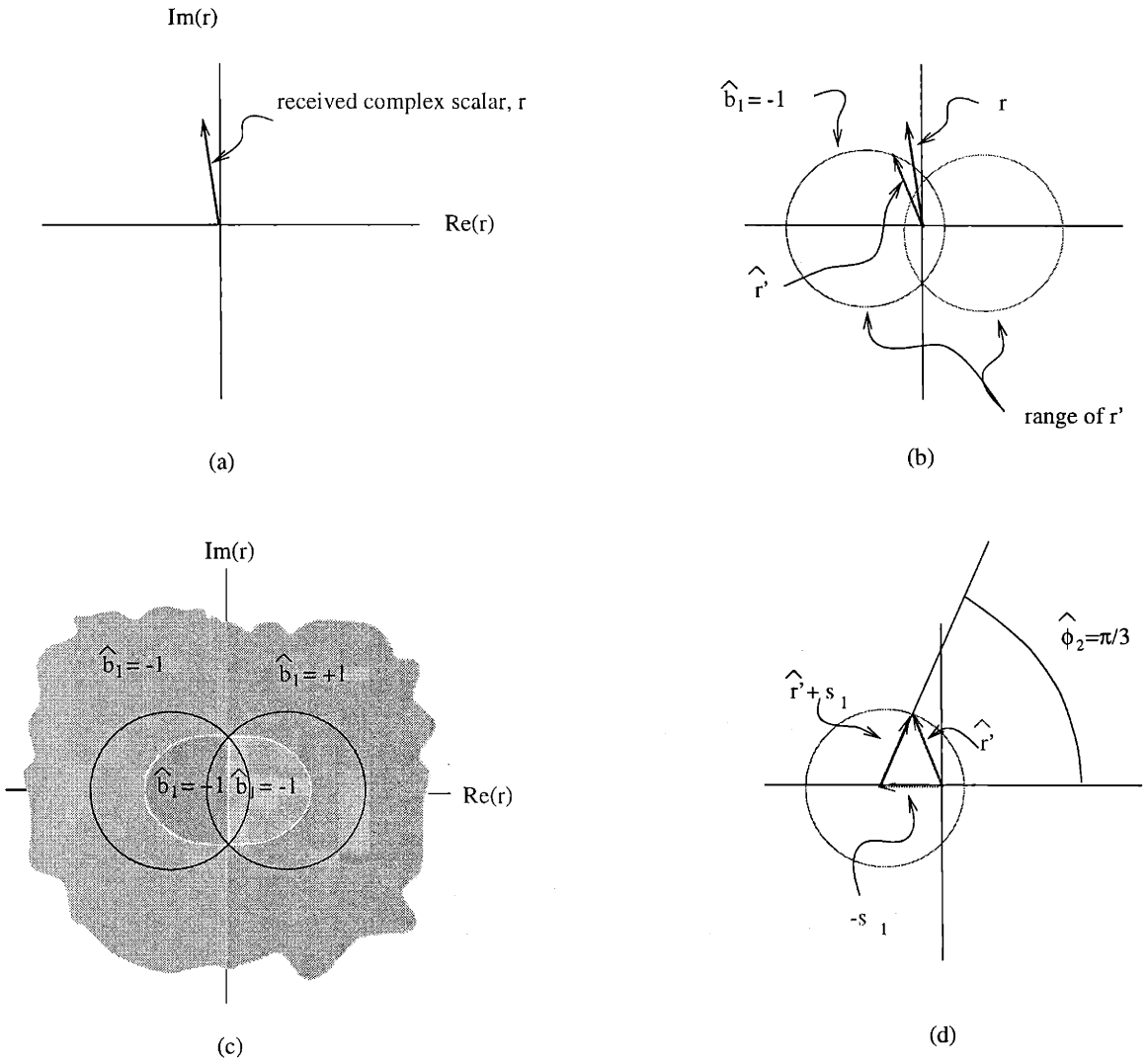


Figure 4-2: Example of two users in one complex dimension. User 1 has a known phase (0 radians) and user 2 has a uniformly distributed phase. See Figure 4-1-(a) and (b). (a) A specific realization of the received signal, r. (b) The received value is projected on to the range of r'. This projection is $\hat{r}'$. The bit estimate is $\hat{b}_1 = -1$ since $\hat{r}'$ is on the left ring. (c) The $\hat{b}_1$ decision regions. For any received signal that falls within the pink (green) regions, the estimate for b_1 will be $+1$ (-1). (c) The phase estimate for this example is the angle of $\hat{r}' - \hat{b}_1 s_1 = \hat{r}' - (-s_1)$. The angle for this example is $\hat{\phi}_2 = \frac{\pi}{3}$.

Finally, we re-write the last two terms on line (4.11) using the two substitutions

$$\Re\{\mathbf{s}_K e^{j\phi}\} = \Re\{\mathbf{s}_K\}\cos\phi - \Im\{\mathbf{s}_K\}\sin\phi,$$

$$\Im\{\mathbf{s}_K e^{j\phi}\} = \Im\{\mathbf{s}_K\}\cos\phi + \Re\{\mathbf{s}_K\}\sin\phi.$$

The last two terms on line (4.11) can be re-written as

$$\begin{aligned} &-2\cos\phi[\Re\{\mathbf{s}_K\}^T\Re\{\mathbf{r}-\mathbf{S}\Phi\mathbf{b}\} + \Im\{\mathbf{s}_K\}^T\Im\{\mathbf{r}-\mathbf{S}\Phi\mathbf{b}\}] \\ &-2\sin\phi[\Re\{\mathbf{s}_K\}^T\Im\{\mathbf{r}-\mathbf{S}\Phi\mathbf{b}\} - \Im\{\mathbf{s}_K\}^T\Re\{\mathbf{r}-\mathbf{S}\Phi\mathbf{b}\}]. \end{aligned} \tag{4.14}$$

Using the identity

$$z_1 z_2^* = \Re\{z_1\}\Re\{z_2\} + \Im\{z_1\}\Im\{z_2\} + j[\Re\{z_2\}\Im\{z_1\} - \Im\{z_2\}\Re\{z_1\}]$$

(4.14) may be re-written as

$$-2\Re\{\mathcal{X}(\mathbf{b}|\mathbf{r})\}\cos\phi - 2\Im\{\mathcal{X}(\mathbf{b}|\mathbf{r})\}\sin\phi,$$

where

$$\boxed{\mathcal{X}(\mathbf{b}|\mathbf{r}) = \mathbf{s}_K^T(\mathbf{r}-\mathbf{S}\Phi\mathbf{b})} \tag{4.15}$$

is a complex scalar.[11] The likelihood function of Equation (4.8) may now be written as

$$\Lambda_1(\mathbf{b},\phi|\mathbf{r}) = F_K(\mathbf{b}|\mathbf{r}) + \varepsilon_K - 2\Re\{\mathcal{X}(\mathbf{b}|\mathbf{r})\}\cos\phi - 2\Im\{\mathcal{X}(\mathbf{b}|\mathbf{r})\}\sin\phi. \tag{4.16}$$

Since ε_K, defined in Equation (4.13), is independent of $\mathbf{b}$ and is the same for all values

[11] Recall that $\mathbf{s}_K$ is real for the MA model used in this chapter.

of ϕ, the ML joint weight/phase estimator becomes

$$\{\hat{\mathbf{b}}, \hat{\phi}_K\} = \arg \min_{\mathbf{b},\phi} \Lambda_2(\mathbf{b}, \phi|\mathbf{r})$$

where

$$\Lambda_2(\mathbf{b}, \phi|\mathbf{r}) = F_K(\mathbf{b}|\mathbf{r}) - 2\Re\{\mathcal{X}(\mathbf{b}|\mathbf{r})\}\cos\phi - 2\Im\{\mathcal{X}(\mathbf{b}|\mathbf{r})\}\sin\phi. \tag{4.17}$$

Since the choices for $\mathbf{b}$ are from a discrete set, it is convenient to perform this minimization in two steps,

$$\{\hat{\mathbf{b}}, \hat{\phi}_K\} = \arg \min_{\mathbf{b}} \min_{\phi} \Lambda_2(\mathbf{b}, \phi|\mathbf{r}).$$

Given the measurement $\mathbf{r}$, an estimate of ϕ is found for every possible value of $\mathbf{b}$. This estimate is denoted by $\hat{\phi}(\mathbf{b}|\mathbf{r})$. The weight vector, $\hat{\mathbf{b}}$, is chosen to equal the the value of $\mathbf{b}$ which gives the smallest $\Lambda_2(\mathbf{b}, \hat{\phi}(\mathbf{b}|\mathbf{r})|\mathbf{r})$. To find $\hat{\phi}(\mathbf{b}|\mathbf{r})$, we set $\frac{\partial}{\partial\phi}\Lambda_2(\mathbf{b}, \phi|\mathbf{r}) = 0$ and solve for ϕ:

$$\frac{\partial}{\partial\phi}\Lambda_2(\mathbf{b}, \phi|\mathbf{r}) = -2\Re\{\mathcal{X}(\mathbf{b}|\mathbf{r})\}\sin\phi + 2\Im\{\mathcal{X}(\mathbf{b}|\mathbf{r})\}\cos\phi = 0,$$

$$\hat{\phi}(\mathbf{b}|\mathbf{r}) = \tan^{-1}\frac{\Im\{\mathcal{X}(\mathbf{b}|\mathbf{r})\}}{\Re\{\mathcal{X}(\mathbf{b}|\mathbf{r})\}}.$$

Calculating an inverse tangent for every possible value of $\mathbf{b}$ may be avoided by substituting the above solution for ϕ into Equation (4.17):

$$\Lambda_3(\mathbf{b}|\mathbf{r}) \;=\; \Lambda_2(\mathbf{b}, \phi = \tan^{-1}\frac{\Im\{\mathcal{X}(\mathbf{b}|\mathbf{r})\}}{\Re\{\mathcal{X}(\mathbf{b}|\mathbf{r})\}}|\mathbf{r})$$

and using the trigonometric identities

$$\cos(\tan^{-1}\frac{a}{b}) = \frac{b}{\sqrt{a^2+b^2}}, \quad \sin(\tan^{-1}\frac{a}{b}) = \frac{a}{\sqrt{a^2+b^2}},$$

to obtain

$$\begin{aligned}\Lambda_3(\mathbf{b}|\mathbf{r}) &= F_K(\mathbf{b}|\mathbf{r}) - 2\Re\{\mathcal{X}(\mathbf{b}|\mathbf{r})\}\cos(\hat{\phi}(\mathbf{b}|\mathbf{r})) - 2\Im\{\mathcal{X}(\mathbf{b}|\mathbf{r})\}\sin(\hat{\phi}(\mathbf{b}|\mathbf{r})) \\ &= F_K(\mathbf{b}|\mathbf{r}) - 2\sqrt{\Re\{\mathcal{X}(\mathbf{b}|\mathbf{r})\}^2 + \Im\{\mathcal{X}(\mathbf{b}|\mathbf{r})\}^2}\end{aligned}$$

or

$$\Lambda_3(\mathbf{b}|\mathbf{r}) = F_K(\mathbf{b}|\mathbf{r}) - 2|\mathcal{X}(\mathbf{b}|\mathbf{r})|,$$

where $|\cdot|$ is the magnitude of the complex scalar argument.

An estimation procedure equivalent to the ML estimator in Equation (4.6) can be performed in two parts. First, find the value of $\mathbf{b}$ such that

$$\hat{\mathbf{b}} = \arg\min_{\mathbf{b}} \Lambda_3(\mathbf{b}|\mathbf{r})$$

or

$$\boxed{\hat{\mathbf{b}} = \arg\min_{\mathbf{b}}[F_K(\mathbf{b}|\mathbf{r}) - 2|\mathcal{X}(\mathbf{b}|\mathbf{r})|]} \tag{4.18}$$

where $F_K(\mathbf{b}|\mathbf{r})$ and $\mathcal{X}(\mathbf{b}|\mathbf{r})$ are defined in Equations (4.12) and (4.15), respectively.

Second, calculate the estimate $\hat{\phi}_K = \hat{\phi}(\hat{\mathbf{b}}|\mathbf{r})$

$$\boxed{\hat{\phi}_K = \tan^{-1}\frac{\Im\{\mathcal{X}(\hat{\mathbf{b}}|\mathbf{r})\}}{\Re\{\mathcal{X}(\hat{\mathbf{b}}|\mathbf{r})\}}} \tag{4.19}$$

Simulation results of the NC weight/phase estimator showing bit error rate curves and phase error histograms are reported in Sections 4.7.1 and 4.7.2, respectively.

4.2 The Estimation Algorithm: An Example

The MAP joint weight/phase estimator is shown in Equations (4.18) and (4.19). In general, each possible realization of $\mathbf{b}$ must be tried. This requires $|\Gamma|-1$ comparisons, where $|\Gamma|$, the number of elements in the set Γ, is exponential in the number of users. If the user signature vectors are tree-structured, a substantial savings in computational complexity of the joint estimator may be realized. As was the case in Chapter 3 for $F(\mathbf{b}|\mathbf{r})$ of Equation (3.9), $F_K(\mathbf{b}|\mathbf{r})$ of Equation (4.12) is separable when conditioned on the weights that correspond to upper levels of the tree. The other term in Equation (4.18), $|\mathcal{X}(\mathbf{b}|\mathbf{r})|$, is not separable by conditioning, but may be significantly reduced, since

$$\mathbf{s}_K^T\mathbf{s}_n = 0 \quad \forall n \notin adK = dK \cup aK.$$

That is, as a by-product of the tree structure, $\mathcal{X}(\mathbf{b}|\mathbf{r})$ is a function of the weights associated with user K's ancestors and descendants and is *independent* of the rest of the weights corresponding to user signatures that are orthogonal to user K's signature. For tree-structured signature sets,

$$\boxed{\mathcal{X}(\mathbf{b}|\mathbf{r}) = \mathcal{X}(\mathbf{b}_{adK}|\mathbf{r}) = \mathbf{s}_K^T(\mathbf{r} - \mathbf{S}_{adK}\Phi_{adK}\mathbf{b}_{adK}).} \tag{4.20}$$

The optimal NC tree weight estimator for tree-structured signatures is

$$\boxed{\hat{\mathbf{b}} = \arg\min_{\mathbf{b}}[F_K(\mathbf{b}|\mathbf{r}) - 2|\mathcal{X}(\mathbf{b}_{adK})|\mathbf{r})|]} \tag{4.21}$$

with $F_K(\mathbf{b}|\mathbf{r})$ defined in Equation (4.12).

As was done in Section 3.4.2, $F_K(\mathbf{b}|\mathbf{r})$ can be decomposed into N conditionally independent terms, corresponding to the N users at the bottom of the tree. This

separation allowed for a tree climbing procedure that created conditional weight estimate tables at each node. For MA with one unknown-phase user, the minimization in Equation (4.21) can also be carried out through a tree climbing procedure of table creations. For this unknown-phase case, however, there are four types of tables. Figure 4-3 pictorially shows the four different types of tables and where they occur on the tree relative to the node at which the unknown-phase user is located. For the example in Figure 4-3, user 12 is the last addition to the system. The phase of user 12 is not known, but the phases of the other users are assumed to have been accurately acquired prior to the addition of user 12. Below, a description of each type of table is given and illustrated by stepping through the example in the figure.

Standard table: Since $\mathcal{X}(\mathbf{b}_{adK}|\mathbf{r})$ is independent of $\mathbf{b}_{\overline{adK}}$ (the weights at nodes n, $\forall n \notin adK$), a **standard table** of conditional weight estimates may be constructed at all nodes n, $\forall n \notin adK$. The estimate that is stored in the standard table at node n for a specific realization of $\mathbf{b}_{an}$ is found from

$$\hat{b}_n(\mathbf{r}|\mathbf{b}_{an}) = \arg \min_{b_n} ||\mathbf{r} - \mathbf{s}_n b_n e^{j\phi_n} - \mathbf{S}_{an}\Phi_{an}\mathbf{b}_{an} - \mathbf{S}_{dn}\Phi_{dn}\hat{\mathbf{b}}_{dn}(\mathbf{r}|b_n, \mathbf{b}_{an})||^2,$$

where $\hat{\mathbf{b}}_{dn}(\mathbf{r}|b_n, \mathbf{b}_{an})$ is defined in Equations (3.17) and (3.18). As was described in more detail in Section 3.6.1, the weight estimates, $\hat{\mathbf{b}}_{dn}(\mathbf{r}|b_n, \mathbf{b}_{an})$, can easily be obtained from a succession of simple standard table look-ups at the descendent nodes of node n.

In the example of Figure 4-3, $\overline{ad(12)} = \{1, 2, 3, 4, 5, 6, 9, 10, 11, 13\}$. Weight estimates of $\mathbf{b}_{\overline{ad(12)}}$ are decoupled from the estimate of user 12's phase and the weight estimates for users 7 and 8 by conditioning on the weights of users 14 and 15. Hence, the nodes $\mathbf{b}_{\overline{ad(12)}}$ are treated exactly as if all phases in the tree were known. Specifically, the standard tables of conditional weight estimates that were created for the coherent tree detector of Section 3.4.1 may be constructed for nodes $\overline{ad(12)} = \{1, 2, 3, 4, 5, 6, 9, 10, 11, 13\}$, but not for nodes $ad(12) = \{7, 8, 14, 15\}$.

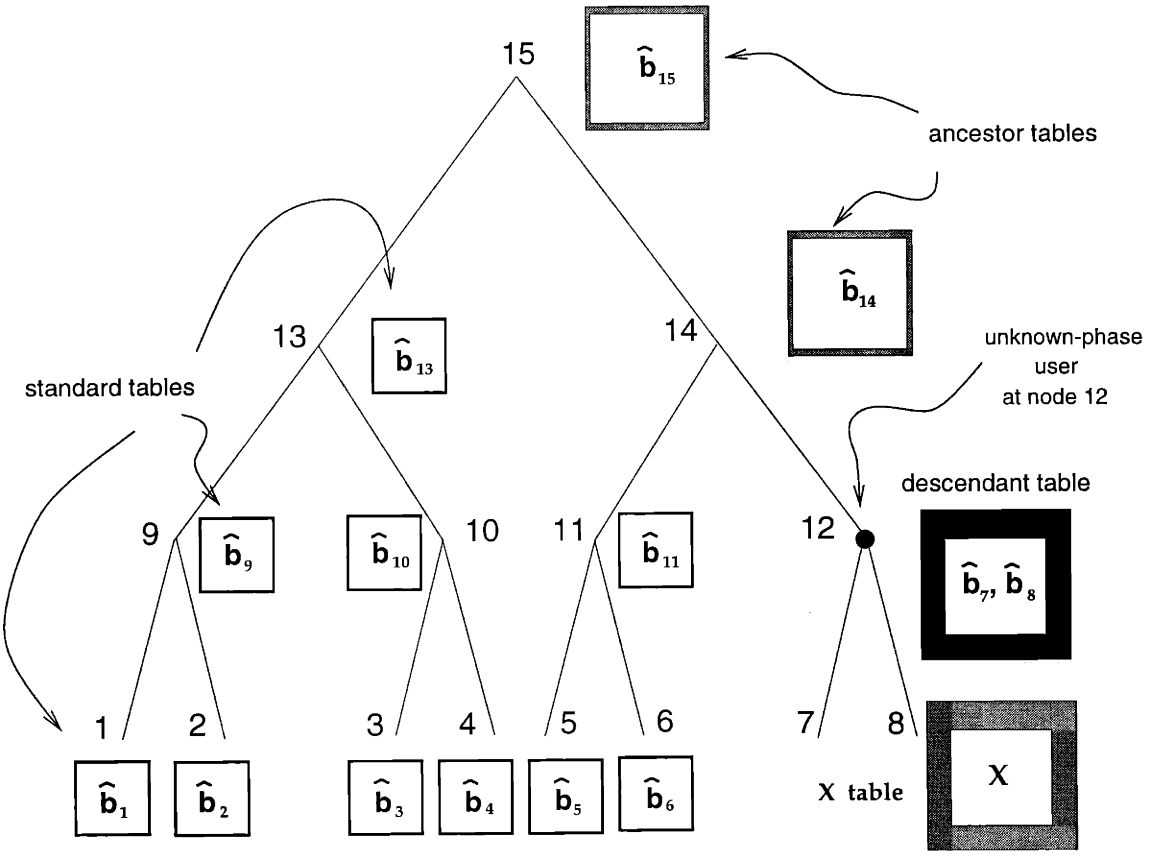


Figure 4-3: Example of the non-coherent weight/phase estimator and the tables it constructs.

Table 4.1: **Node 1: standard table.** See Figure 4-3 for the full tree illustration and the position of node 1. The blanks would be filled by calculations of Equation (4.22).

b_9, b_{13}, b_{15}	$\hat{b}_1(\mathbf{r}\|b_9, b_{13}, b_{15})$
$+1\ +1\ +1$	
$+1\ +1\ -1$	
$+1\ -1\ +1$	
$+1\ -1\ -1$	
$-1\ +1\ +1$	
$-1\ +1\ -1$	
$-1\ -1\ +1$	
$-1\ -1\ -1$	

Tables 4.1, 4.2, and 4.3 show a binary example of the standard tables that would be created at nodes 1, 9, and 13, respectively. Without loss of generality, assume all known phases are zero, i.e., $\Phi = \mathbf{I}$. The entries of Table 4.1 are calculated by

$$\hat{b}_1(\mathbf{r}|b_9, b_{13}, b_{15}) = \arg \min_{b_1} ||\mathbf{r} - \mathbf{s}_1 b_1 - \mathbf{s}_9 b_9 - \mathbf{s}_{13} b_{13} - \mathbf{s}_{15} b_{15}||^2, \tag{4.22}$$

where each entry of the table is indexed by $\{b_9, b_{13}, b_{15}\}$. The tables at nodes 2, 3, 4, 5 and 6 would be constructed in the same fashion as the table at node 1.

The entries of the standard table created at node 9, shown in Table 4.2, are calculated by

$$\hat{b}_9(\mathbf{r}|b_{13}, b_{15}) = \arg \min_{b_9} ||\mathbf{r} - \mathbf{s}_9 b_9 - \mathbf{s}_{13} b_{13} - \mathbf{s}_{15} b_{15} - \sum_{i=1}^{2} \mathbf{s}_i \hat{b}_i(\mathbf{r}|b_9 b_{13} b_{15})||^2, \tag{4.23}$$

where each entry of the table is indexed by $\{b_{13}, b_{15}\}$. The estimates $\hat{b}_i(\mathbf{r}|b_9, b_{13}, b_{15})$ for $i = 1, 2$ have been stored in the standard tables just created at nodes 1 and 2, and are indexed by $\{b_9, b_{13}, b_{15}\}$. The standard tables at nodes 10 and 11 would be constructed in the same fashion as was the standard table at node 9.

The standard table at node 13 of Figure 4-3 is shown in Table 4.3. An entry of

Table 4.2: **Node 9: standard table.** See Figure 4-3 for the position of node 9 in the tree. The blanks would be filled by calculations of Equation (4.23).

b_{13}, b_{15}	$\hat{b}_9(\mathbf{r}\|b_{13}, b_{15})$
$+1\ +1$	
$+1\ -1$	
$-1\ +1$	
$-1\ -1$	

this table, in the row indexed by a specific realization of b_{15}, would be calculated from

$$\begin{aligned}\hat{b}_{13}(\mathbf{r}|b_{15}) \quad = \arg\ \min_{b_{13}} \|\ & \mathbf{r} - \mathbf{s}_{13}b_{13} - \mathbf{s}_{15}b_{15} \\ & - \textstyle\sum_{i=9}^{10} \mathbf{s}_i \hat{b}_i(\mathbf{r}|b_{13}, b_{15}) \\ & - \textstyle\sum_{k=1}^{4} \mathbf{s}_k \hat{b}_k(\mathbf{r}|\hat{b}_{pk}(\mathbf{r}|b_{13}, b_{15}), b_{13}, b_{15})\ \|^2.\end{aligned} \tag{4.24}$$

The first summation above requires the estimates $\hat{b}_i(\mathbf{r}|b_{13}, b_{15})$, $i = 9, 10$. Recall that these estimates have been stored in the previously constructed standard tables for nodes 9 and 10 and are indexed by $\{b_{13}, b_{15}\}$. The second summation in Equation (4.24) requires the four estimates denoted by $\hat{b}_k(\mathbf{r}|\hat{b}_{pk}(\mathbf{r}|b_{13}, b_{15}), b_{13}, b_{15})$ for $k = 1, 2, 3, 4$. These estimates have also been stored in the previously constructed standard tables at nodes 1, 2, 3, and 4. Each of these four estimates is indexed by three values: the specific value of b_{15} denoting which row of the standard table at node 13 that we are currently being calculated; the specific value of b_{13} that we are testing; and the estimate $\hat{b}_{pk}(\mathbf{r}|b_{13}, b_{15})$. Recall that pk denotes the parent node of node k. In this example, for $k = 1$, $pk = 9$, i.e., the parent of node 1 is node 9. For $k = 1$, we need to look up $\hat{b}_1(\mathbf{r}|\hat{b}_9(\mathbf{r}|b_{13}, b_{15}), b_{13}, b_{15})$; for this, we need the value of $\hat{b}_9(\mathbf{r}|b_{13}, b_{15})$. This value has been stored in the standard table that was constructed at node 9; it is indexed by $\{b_{13}, b_{15}\}$. For $k = 2$, $pk = 9$; given the realization of $\{b_{13}, b_{15}\}$, and the value from the standard standard table at node 9 for $\hat{b}_9(\mathbf{r}|b_{13}, b_{15})$ we look up the value for $\hat{b}_2(\mathbf{r}|\hat{b}_9(\mathbf{r}|b_{13}, b_{15}), b_{13}, b_{15})$ from the standard table at node 2. For $k = 3$,

Table 4.3: **Node 13: standard table.** See Figure 4-3 for the position of node 13 in the tree. The blanks would be filled by calculations of Equation (4.24).

b_{15}	$\hat{b}_{13}(\mathbf{r}\|b_{15})$
$+1$	
-1	

$pk = 10$; given the same realization of $\{b_{13}, b_{15}\}$, we use the value from the standard table at node 10 for $\hat{b}_{10}(\mathbf{r}|b_{13}, b_{15})$ to look up the value for $\hat{b}_3(\mathbf{r}|\hat{b}_{10}(\mathbf{r}|b_{13}, b_{15}), b_{13}, b_{15})$ from the standard table at node 3. Similarly, we find $\hat{b}_4(\mathbf{r}|\hat{b}_{10}(\mathbf{r}|b_{13}, b_{15}), b_{13}, b_{15})$.

$\mathcal{X}$-table: Due to the existence of the unknown phase at node K of the tree, the weight estimates corresponding to nodes dK cannot be decoupled through conditioning. In other words, since there are infinitely many possibilities for the phase at node K, weight estimates at nodes dK cannot be conditioned upon ϕ_K. Instead, a single, large table, the **$\mathcal{X}$-table**, may be constructed. For each possible realization of $\mathbf{b}_{adK}$, this table has one entry, namely the complex scalar resulting from the calculation of Equation (4.20).

This is seen in the example of Figure 4-3. Specifically, since $\mathbf{s}_7$ and $\mathbf{s}_8$ are correlated with $\mathbf{s}_{12}$, the decisions on weight estimates for users 7 and 8 are dependent upon the phase of user 12's transmission. Since this phase is a continuous random variable, the standard tables cannot be created for nodes 7 and 8 since there are infinitely many values of ϕ_{12} upon which to condition. We can, however, create an $\mathcal{X}$-table. Specifically, for each possible realization of $\mathbf{b}_{ad(12)} = [b_7, b_8, b_{14}, b_{15}]$ we find

$$\mathcal{X}(\mathbf{b}_{ad(12)}|\mathbf{r}) = \mathbf{s}_{12}^T(\mathbf{r} - \mathbf{S}_{ad(12)}\mathbf{b}_{ad(12)}). \tag{4.25}$$

Table 4.4 gives an example of the $\mathcal{X}$-table that would be placed at the bottom of the tree in Figure 4-3. Note that the number of entries in the $\mathcal{X}$-table is exponential in the number of ancestors *and* descendants of the unknown-phase user. Since the users

Table 4.4: $\mathcal{X}$-**table** depicted in Figure 4-3. For each realization of $\mathbf{b}_{ad(12)} = [b_7, b_8, b_{14}, b_{15}]^T$ in the left column of this table, a corresponding value for $\mathcal{X}(\mathbf{b}_{ad(12)}|\mathbf{r})$, calculated by Equation (4.25), may be entered into the right column of this table.

b_7, b_8, b_{14}, b_{15}	$\mathcal{X}(b_7, b_8, b_{14}, b_{15}\|\mathbf{r})$
$+1\ +1\ +1\ +1$	
$+1\ +1\ +1\ -1$	
$+1\ +1\ -1\ +1$	
$+1\ +1\ -1\ -1$	
$+1\ -1\ +1\ +1$	
$\vdots$	$\vdots$
$-1\ -1\ +1\ -1$	
$-1\ -1\ -1\ -1$	

are employing binary signaling in this example, this $\mathcal{X}$-table has $2^4 = 16$ entries.

Descendant table: The next table that can be constructed is a table of joint conditional weight estimates for the descendants of node K. This table is most conveniently placed at node K (node 12 in the example of Figure 4-3). The joint estimate that would be entered into the **descendant table**, for a specific realization of $\mathbf{b}_{aK}$, is found from

$$\hat{\mathbf{b}}_{dK}(\mathbf{r}|\mathbf{b}_{aK}) = \arg \min_{\mathbf{b}_{dK}} ||\mathbf{r} - \mathbf{S}_{aK}\Phi_{aK}\mathbf{b}_{aK} - \mathbf{S}_{dK}\Phi_{dK}\mathbf{b}_{dK}||^2 - 2|\mathcal{X}(\mathbf{b}_{aK}, \mathbf{b}_{dK}|\mathbf{r})|.$$

Note that the values of the conditional estimates of $\mathbf{b}_{\overline{adK}}$ stored in the standard tables are independent of $\mathbf{b}_{dK}$ and, thus, do not appear in the minimization equation for the creation of the descendant table. This is due to the conditionally separable property of $F_K(\mathbf{b}|\mathbf{r})$ that was shown in Section 3.4.2.

In the example of Figure 4-3, the descendant table is constructed at node 12 where, for each hypothesized pair of values of $\{b_{14}, b_{15}\}$, the joint conditional estimates of b_7 and b_8 are optimally determined by choosing those values which correspond to the

Table 4.5: **Node 12: descendant table.** See Figure 4-3. For each realization of $\mathbf{b}_{a(12)} = [b_{14}, b_{15}]^T$ in the left column of this table, a corresponding pair of values for $\{\hat{b}_7(\mathbf{r}|b_{14}, b_{15}), \hat{b}_8(\mathbf{r}|b_{14}, b_{15})\}$, calculated by Equation (4.26), may be entered into the right column of this table.

b_{14}, b_{15}	$\hat{b}_7(\mathbf{r}\|b_{14}, b_{15})\ \hat{b}_8(\mathbf{r}\|b_{14}, b_{15})$
$+1\ +1$	
$+1\ -1$	
$-1\ +1$	
$-1\ -1$	

minimization

$$\begin{bmatrix} \hat{b}_7(\mathbf{r}|b_{14}, b_{15}) \\ \hat{b}_8(\mathbf{r}|b_{14}, b_{15}) \end{bmatrix} = \arg\min_{b_7, b_8} ||\mathbf{r} - \mathbf{s}_7 b_7 - \mathbf{s}_8 b_8 - \mathbf{s}_{14} b_{14} - \mathbf{s}_{15} b_{15}||^2 - 2|\mathcal{X}(b_7, b_8, b_{14}, b_{15}|\mathbf{r})|. \tag{4.26}$$

Recall that $\mathcal{X}(b_7, b_8, b_{14}, b_{15}|\mathbf{r})$ has already been calculated and is stored in the $\mathcal{X}$-table at the bottom of the tree. An example of the descendant table that would be created at node 12 is shown in Table 4.5. The computational complexity for the descendant table is exponential in the number of ancestors *and* descendants of the unknown-phase user. In this example there are $2^{|d(12)|} - 1 = 2^4 - 1 = 15$ comparisons needed to find a single entry in the table. There are $2^{|a(12)|} = 2^2 = 4$ entries in the table. It follows that there are a total of $2^{|a(12)|}(2^{|d(12)|} - 1) = 60$ comparisons needed to fill the descendant table.

Ancestor tables: The estimation algorithm is completed by successively creating tables for the remainder of the tree, at the nodes aK, starting first, with node pK and ending with the root node. At a node $i \in aK$, an ancestor table is constructed by calculating a conditional estimate for every possible realization of $\mathbf{b}_{ai}$ from

$$\hat{\mathbf{b}}_i(\mathbf{r}|\mathbf{b}_{ai}) = \arg\ \min_{b_i} \mathcal{F}_i(b_i, \mathbf{b}_{ai}|\mathbf{r}) - 2|\mathcal{X}(\hat{\mathbf{b}}_{dK}(\mathbf{r}|\hat{\mathbf{b}}_\vartheta(\mathbf{r}|b_i\mathbf{b}_{ai})), \hat{\mathbf{b}}_\vartheta(\mathbf{r}|b_i\mathbf{b}_{ai}), b_i, \mathbf{b}_{ai}|\mathbf{r})|$$

where

$$\begin{aligned}\mathcal{F}_i(b_i, \mathbf{b}_{ai}|\mathbf{r}) \quad = || \quad & \mathbf{r} - \mathbf{s}_i e^{j\phi_i} b_i - \mathbf{S}_{ai}\Phi_{ai}\mathbf{b}_{ai} \\ & - \mathbf{S}_{dK}\Phi_{dK}\hat{\mathbf{b}}_{dK}(\mathbf{r}|\hat{\mathbf{b}}_{\vartheta}(\mathbf{r}|b_i\mathbf{b}_{ai}), b_i, \mathbf{b}_{ai}) \\ & - \mathbf{S}_{\zeta}\Phi_{\zeta}\hat{\mathbf{b}}_{\zeta}(\mathbf{r}|b_i\mathbf{b}_{ai}) \quad ||^2\end{aligned}$$

where $\vartheta = aK\backslash\{i, ai\}$, i.e., the set of ancestor nodes to node K that are not in the set of nodes $\{i, ai\}$. Note that $\hat{\mathbf{b}}_{\vartheta}(\mathbf{r}|b_i\mathbf{b}_{ai})$ can be retrieved from successive table look-ups of ancestor tables previously constructed at the nodes below node i and above node K. Recall that $\hat{\mathbf{b}}_{dK}(\mathbf{r}|\hat{\mathbf{b}}_{\vartheta}(\mathbf{r}|b_i\mathbf{b}_{ai}))$ can be obtained by a simple look-up in the descendant table at node K. In the last term of $\mathcal{F}_i(b_i, \mathbf{b}_{ai}|\mathbf{r})$, $\zeta = di\backslash dK$, i.e., the set of descendant nodes of node i that are not descendants of node K. The conditional weight estimates for these nodes are collected together in $\hat{\mathbf{b}}_{\zeta}(\mathbf{r}|b_i\mathbf{b}_{ai})$ and may be obtained from simple successive look-ups of either the standard tables or just constructed ancestor tables.

To illustrate the construction of the ancestor tables, consider node 14 in Figure 4-3. The ancestor table for node 14 has two entries, one corresponding to $b_{15} = +1$ and one corresponding to $b_{15} = -1$. Each entry is found by fixing the value for b_{15} and trying both possible values for b_{14}. The conditional estimate for b_{14} is the one that minimizes

$$\hat{b}_{14}(\mathbf{r}|b_{15}) = \arg\min_{b_{14}} \mathcal{F}_{14}(b_{14}, b_{15}|\mathbf{r}) - 2|\mathcal{X}(\hat{\mathbf{b}}_{7,8}(\mathbf{r}|b_{14}, b_{15}), b_{14}, b_{15}|\mathbf{r})|,$$

where

$$\mathcal{F}_{14}(b_{14}, b_{15}|\mathbf{r}) = ||\mathbf{r} - \mathbf{s}_{14}b_{14} - \mathbf{s}_{15}b_{15} - \sum_{i=7}^{8} \mathbf{s}_i\hat{b}_i(\mathbf{r}|b_{14}, b_{15}) - \sum_{k\in\zeta} \mathbf{s}_k\hat{b}_k(\mathbf{r}|b_{14}, b_{15})||^2. \quad (4.27)$$

In this case, for node 14, $\zeta = \{11, 5, 6\}$. The estimates $\hat{b}_7(\mathbf{r}|b_{14}, b_{15})$ and $\hat{b}_8(\mathbf{r}|b_{14}, b_{15})$ are needed in the first summation on the right hand side of Equation (4.27); these conditional estimates have been stored in the descendant table that sits at node 12. The estimates $\hat{b}_k(\mathbf{r}|b_{14}, b_{15}), k \in \zeta = \{11, 5, 6\}$ are needed in the second summation on

the right hand side of Equation (4.27). First, for the fixed value of b_{15} and the value of b_{14} that we are testing, we retrieve the estimate $\hat{b}_{11}(\mathbf{r}|b_{14}, b_{15})$ from the standard table at node 11. Next, we need the estimates from the lowest level of the tree: $\hat{b}_5(\mathbf{r}|\hat{b}_{11}(\mathbf{r}|b_{14}, b_{15}), b_{14}, b_{15})$ and $\hat{b}_6(\mathbf{r}|\hat{b}_{11}(\mathbf{r}|b_{14}, b_{15}), b_{14}, b_{15})$. These have been stored in the standard tables at nodes 5 and 6. The computational complexity of creating an ancestor table is nearly the same as that needed to create a standard table since the number of comparisons needed for each table entry is exponential in the number of *ancestors* in both cases.

The estimate for b_{15} is obtained by minimizing

$$\hat{b}_{15}(\mathbf{r}) = \arg\min_{b_{15}} \mathcal{F}_{15}(b_{15}|\mathbf{r}) - 2|\mathcal{X}(\hat{\mathbf{b}}_{7,8}(\mathbf{r}|\hat{b}_{14}(\mathbf{r}|b_{15}), b_{15}), \hat{b}_{14}(\mathbf{r}|b_{15}), b_{15}|\mathbf{r})|,$$

$$\mathcal{F}_{15}(b_{15}|\mathbf{r}) = ||\mathbf{r} - \mathbf{s}_{15}b_{15} - \sum_{i=7}^{8} \mathbf{s}_i \hat{b}_i(\mathbf{r}|b_{14}, b_{15}) - \sum_{k\in\zeta} \mathbf{s}_k \hat{b}_k(\mathbf{r}|b_{14}, b_{15})||^2,$$

where, in this case for node 15, $\zeta = \{13, 14, 9, 10, 11, 1, 2, 3, 4, 5, 6\}$. Obtaining the estimates at these nodes may be done with successive table-lookups as described above for the calculation of the ancestor table at node 14.

Conceptually, once the estimate of b_{15} is obtained, it is a simple matter of successive table look-ups that propagate down the tree to determine the optimal estimates, first, for users 13 and 14, and then for their children, users 9, 10, 11, and then for 7 and 8, then, finally, down to nodes 1, 2, 3, 4, 5, and 6 at the bottom. To complete the optimal NC joint weight/phase estimation procedure, the optimal estimate of user 12's phase is calculated from Equation (4.19), using the optimal estimates for users 7, 8, 14 and 15

$$\hat{\phi}_{12} = \tan^{-1} \frac{\Im\{\mathcal{X}(\hat{\mathbf{b}}_{ad(12)}|\mathbf{r})\}}{\Re\{\mathcal{X}(\hat{\mathbf{b}}_{ad(12)}|\mathbf{r})\}}.$$

As this example illustrates, the NC tree weight/phase estimation algorithm takes advantage of the signal set structure, sweeping up the tree creating a standard table of conditional weight estimates at each node that is neither an ancestor nor descendant

of the unknown-phase user. For the rest of the nodes in the tree, the algorithm begins at the node corresponding to the unknown phase, creating the $\mathcal{X}$-table, the entries of which are a function of the values of the weights corresponding to the ancestors and descendants of the unknown-phase user. Also at this node, the descendant table of conditional joint estimates is created. The algorithm then continues to sweep up through the ancestors of the unknown-phase user, creating ancestor tables of conditional weight estimates.

A rough lower bound on the complexity of this procedure is easily determined: the introduction of a single unknown phase at a node K of the tree requires the calculation of two large tables, the $\mathcal{X}$-table and the descendant table, each having a computational complexity on the order of $M^{(|adK|)}$, where $|adK|$ is the number of ancestors and descendants to node K, and each user employs M-ary signaling. In general, the joint weight/phase estimation procedure for tree-structured signature sets is of lowest complexity if the user having the unknown phase is at the bottom of the tree since $|adK|$ for typical tree structures is minimized if K denotes a lowest level node. The computational complexity is calculated exactly in the next section.

4.3 Computational Complexity

For simplicity, assume all $K-1$ users employ M-ary signaling, i.e., $\mathbf{b} \in \Gamma$, where, $|\Gamma| = M^{(K-1)}$. The K^{th} user transmits a weight of $+1$, and its phase is to be estimated. Also, the tree is restricted to be of uniform composition in that there are exactly Q children emanating from each node.[12] Furthermore, there are a total of L levels, and at the bottom there are N nodes, hence, $N = Q^{L-1}$ and $L = \log_Q N + 1$. As was done in Section 3.5 we count the number of compares, c, needed to perform the joint weight/phase estimation algorithm. The calculation of c is done in two parts, one for each section of the tree as color coded in the example tree shown in Figure 4-4. Let

[12]A tree having Q children emanating from each node will be called a Q-tree.

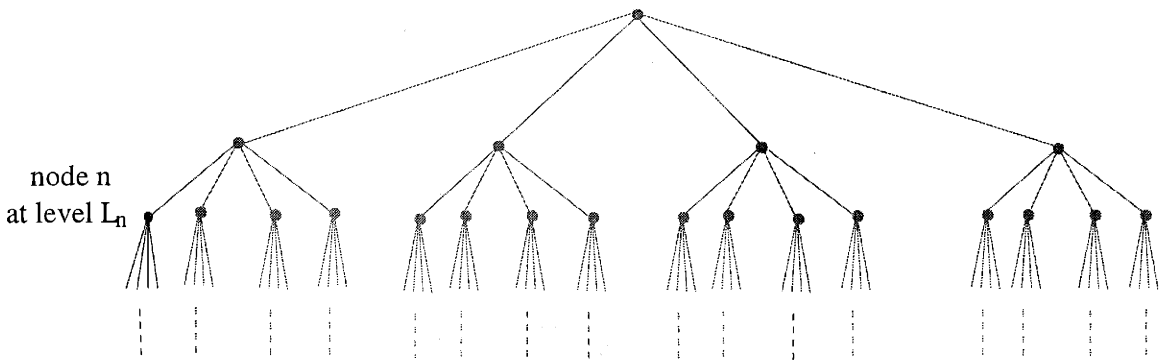


Figure 4-4: A quad tree partitioned into two sections by color.

node n, at level L_n, correspond to the user having the unknown phase.

Green section of tree: The algorithm creates a conditional bit estimate table for each node in this green section of the tree.[13] The green section comprises all non-descendant nodes to node n. It is easiest to find the computational complexity corresponding to the green section of the tree by calculating the computational complexity of filling the entire tree with standard tables, and then subtracting out the complexity of filling the red portion of the tree with standard tables.

Let $c_C(L, Q, M)$ denote the complexity of the coherent tree detector.[14] Calculating $c_C(L, Q, M)$ was done in Section 3.5. Briefly, for each node i at level l, we must do $M - 1$ compares to find the table entry for a single realization of $\mathbf{b}_{ai}$. There are Q^{l-1} nodes at level l. There are M^{l-1} realizations of $\mathbf{b}_{ai}$ for each node, i, at level l. Translating the above sentences into an equation gives

$$c_C(L, Q, M) = \sum_{l=1}^{L} (M-1) Q^{l-1} M^{l-1}.$$

[13]The standard tables and the ancestor tables created in the green section of the tree require the same number of compares for each entry and are considered computationally equivalent.

[14]Recall that this was found in Section 3.5 as a function of N, the number of dimensions. Here, it is shown as a function of L.

In Section 3.5, this was shown to be

$$c_C(L,Q,M) = \frac{M-1}{QM-1}[(QM)^L - 1], \tag{4.28}$$

with $L = \log_Q N + 1$.

The notation $c'_C(L, L_n, Q, M)$ will be used to denote the number of computations needed to fill the red portion of the tree with standard tables. The calculation of $c'_C(L, L_n, Q, M)$ follows. For each node i in the red portion of the tree at level l, $M-1$ compares are needed to find the table entry for a single realization of $\mathbf{b}_{ai}$. There are Q^{l-L_n} nodes in the red section at level $l \geq L_n$. There are M^{l-1} realizations of $\mathbf{b}_{ai}$ for each node, i, at level l. We have

$$\begin{aligned} c'_C(L, L_n, Q, M) &= \sum_{l=L_n}^{L} Q^{l-L_n}(M-1)M^{l-1} \\ &= \frac{M-1}{Q^{L_n}M}\frac{(QM)^{L+1} - (QM)^{L_n}}{QM-1}, \end{aligned}$$

which is re-written as

$$c'_C(L, L_n, Q, M) = \frac{M-1}{QM-1}M^{L_n-1}[(QM)^{L-L_n+1} - 1]. \tag{4.29}$$

Notice that when $L_n = L$, $c'_C(L, L_n, Q, M) = (M-1)M^{L-1}$; this is just the complexity of making a single table at the bottom level.

The computational complexity corresponding to the green section of the tree is

$$c_{green}(L, L_n, Q, M) = c_C(L, Q, M) - c'_C(L, L_n, Q, M). \tag{4.30}$$

Red section of tree: The red section requires the construction of two large tables, the $\mathcal{X}$-table and the descendant table. For the $\mathcal{X}$-table we calculate one value of $\mathcal{X}(\mathbf{b}_{an}, \mathbf{b}_{dn}|\mathbf{r})$ for each possible realization of $[\mathbf{b}_{an}^T, \mathbf{b}_{dn}^T]$. This table, then, will have

$M^{|an|+|dn|}$ entries, where $|an|$ and $|dn|$ are the number of ancestors and descendants to node n, respectively.

For each entry of the descendant table we are finding the best joint estimate of the weights $\{b_k\}_{k\in dn}$, hence, we must do $M^{|dn|}-1$ compares for every possible realization of $\mathbf{b}_{ai}$. There are $M^{|an|}$ possible realizations of $\mathbf{b}_{ai}$. The number of compares needed to create the descendant table is $M^{|an|}(M^{|dn|}-1)$. The complexity corresponding to the red section of the tree is

$$\begin{aligned} c_{red}(L, L_n, Q, M) &= M^{|an|+|dn|} + M^{|an|}(M^{|dn|}-1) \\ &= M^{|an|}(2M^{|dn|}-1). \end{aligned}$$

Noting that

$$\begin{aligned} |an| &= L_n - 1 \\ |dn| &= \textstyle\sum_{j=L_n+1}^{L} Q^{j-L_n} \\ &= \tfrac{Q}{Q-1}\left(\tfrac{Q^{L-1}}{Q^{L_n-1}} - 1\right) \end{aligned}$$

gives

$$c_{red}(L, L_n, Q, M) = M^{L_n-1}[2(M^{\frac{Q}{Q-1}})^{\frac{Q^{L-1}}{Q^{L_n-1}}-1} - 1]. \tag{4.31}$$

Notice the dependence of $c_{red}(L, L_n, Q, M)$ on L_n, the level at which the new unknown-phase user is placed. When $L_n = L$ (when the unknown-phase user is at the bottom of the tree), $c_{red}(L, L_n, Q, M) = M^{L-1}$, as we would expect. Hence, if the new user is at the bottom of the tree, the only calculations done for the red section of the tree correspond to the creation of the $\mathcal{X}$-table. This table requires one entry for each possible realization of the ancestor weights, and a bottom node has M^{L-1} ancestor nodes.

Total complexity for the non-coherent tree estimator: We combine the computational complexities calculated for the two parts of the tree to find the complexity of the

non-coherent tree estimator

$$\begin{aligned} c_{NC}(L, L_n, Q, M) &= c_{green}(L, L_n, Q, M) + c_{red}(L, L_n, Q, M) \\ &= c_C(L, Q, M) - c'_C(L, L_n, Q, M) + c_{red}(L, L_n, Q, M). \end{aligned}$$

We may also write

$$c_{NC}(L, L_n, Q, M) = c_C(L, Q, M) + c_\Delta(L, L_n, Q, M).$$

The difference between $c_C(L, Q, M)$, the complexity of the coherent tree joint detector of Chapter 3, and $c_{NC}(L, L_n, Q, M)$, the complexity of the non-coherent tree joint estimator (for the case of a single user at level L_n having an unknown phase), is

$$\begin{aligned} c_\Delta(L, L_n, M, Q) &= c_{red}(L, L_n, Q, M) - c'_C(L, L_n, Q, M) \\ &= M^{L_n-1}[2(M^{\frac{Q}{Q-1}})^{\frac{Q^{L-1}}{Q^{L_n-1}}-1} - 1] - \tfrac{M-1}{QM-1} M^{L_n-1}[(QM)^{L-L_n+1} - 1]. \end{aligned} \tag{4.32}$$

When $L_n = L$,

$$c_\Delta(L, L, Q, M) = M^{L-1} - (M-1)M^{L-1}$$

and we see a small *decrease* in complexity in the non-coherent case relative to the coherent case. This apparent decrease is an artifact of the different means used to measure the complexity of the $\mathcal{X}$-table and the standard table. The operations counted for the construction of a standard table is the number of compares. Since the $\mathcal{X}$-table is constructed without compares, its complexity is measured as the number of rows in the table.

Table 4.6: Computational complexity of tree algorithm for binary signaling ($M = 2$) and a quad tree ($Q = 4$). Recall that L is the total number of levels in the tree, K is the total number of users (nodes on the tree), and L_n is the level at which the unknown-phase user sits on the tree.

L	K	$c_{NC}(L_n = L)$	$c_{NC}(L_n = L-1)$	$c_{NC}(L_n = L-2)$	$c_{NC}(L_n = L-3)$
3	21	73	117	2.0972e+06	
4	85	585	673	4.1947e+06	3.8686e+25
5	341	4681	4857	8.3930e+06	7.7371e+25

At the other extreme, when $L_n = 1$, we have

$$c_{\Delta}(L, 1, Q, M) = 2(M\frac{Q}{Q-1})^{Q^{L-1}-1} - 1 - \frac{M-1}{QM-1}[(QM)^L - 1],$$

where $Q^{L-1} = N$. As we would expect, in the case where the new unknown-phase user is at the top of the tree, $c_{\Delta}(L, 1, Q, M)$ is exponential in the number of dimensions, N. Recall that for typical trees, N is a linear function of K, the number of users.

Table 4.6 shows $c_{NC}(L, L_n, Q, M)$ for different values of L and L_n for $Q = 4$ and $M = 2$. Notice that for $L_n \leq L - 3$, the total complexity is above 10^{25} which is computationally impractical.

To avoid the high complexity of having an unknown-phase user high up in the tree, a sub-optimal estimation procedure which saves on computational complexity is proposed in Section 4.6.2.

Alternatively, this complexity could be alleviated by designing the the system to accommodate dynamic re-assignment of new users. To avoid ballooning complexity with the addition of new, unknown-phase users, we would prefer to put a new user at a lower level of the tree. Once its phase has been learned accurately, we may move it to another spot in the tree, as long as this move preserves phase. With the state of the art signal synthesizers, this requirement for coherent changeover from one signal to another may be possible but has not yet been implemented ([2]). If such a coherent changeover is possible, then, perhaps, a user's phase can be learned during

acquisition and a coherent changeover can be made from the acquisition signal to the transmission signal within the tree. For this last scenario, the newest user's phase, then, would be known within some standard deviation. Although this might alleviate the need for the NC tree joint weight/phase estimator, the phase estimate might not be within a desired accuracy and might need to be re-estimated. The next section derives the tree joint weight/phase estimator for the case of partial phase information.

4.4 Optimal Partially Coherent Weight/Phase Estimator

A communication system will, for some portion of its operation, have some information about the phase of a user. Often this information is the result of some estimation procedure such as that just described in the previous section. Other times, the phase of a user might drift, hence the new phase is not entirely unknown. In the presence of phase inaccuracies, the systems has two choices of operation: 1) require the user with the phase inaccuracy to transmit a known, training weight and estimate this user's phase only; 2) allow the user to transmit an unknown **information weight** (e.g., $b_i \in \{+1, -1\}$) and estimate both the phase and the weight for this user. This section derives the MAP weight estimator that would correspond to the receiver having complete knowledge for all but one user's signature waveforms and having partial information about the phase of one user. For this section *all* users are allowed to send information weights. Restricting the uncertain-phase user to send a known training weight results in a trivial simplification of the detector derived for the general case of allowing all users to send information weights.

To reflect the partial knowledge of user K's phase, ϕ_K is assumed to be a continuous random variable with a non-uniform distribution that corresponds to the accuracy of the phase knowledge at the receiver. To be consistent with the single user literature ([25]), the MA joint weight/phase estimator derived for this case will be called the

optimal **partially coherent (PC) joint weight/phase estimator.**[15]

There are two slight differences between the partially coherent estimator that will be derived in this section and the non-coherent estimator that was derived in Section 4.1.[16] First, in the partially coherent scenario, *all* users are assumed to be transmitting information, therefore, the model for the received signal includes the weight b_K,

$$\mathbf{r} = \mathbf{S}\Phi\mathbf{b} + b_K\mathbf{s}_K e^{j\phi_K} + \sigma\mathbf{n}. \tag{4.33}$$

Without loss of generality, we may think of user K's presumed phase as being zero radians and the random error in our knowledge of this phase as ϕ_K. Second, the prior distribution on user K's phase in the non-coherent scenario was uniform, for the partially coherent scenario, a reasonable prior is

$$p_{\phi_K}(\phi) = \frac{e^{\alpha\cos\phi}}{2\pi I_o(\alpha)}, \tag{4.34}$$

where $I_o(\alpha)$ is the modified Bessel function of zeroth order. This exponential cosine PDF has been shown by Viterbi ([24]) to be the PDF for the MAP phase estimate of the single user phase-locked loop. Figure 4-5 shows this PDF for several values of α. The parameter α allows the phase to range from a uniform random variable, when $\alpha = 0$, to a deterministic variable, when $\alpha = \infty$.

We wish to find the values of $\mathbf{b}$, b_K and ϕ that maximize the a posteriori probability[17]

$$p_{\mathbf{b},b_K,\phi_K|\mathbf{r}}(\mathbf{b}, b_K, \phi|\mathbf{r}) = p_{\mathbf{r}|\mathbf{b},b_K,\phi_K}(\mathbf{r}|\mathbf{b}, b_K, \phi)p_{\phi_K}(\phi)\frac{Prob(\mathbf{b}, b_K)}{p_{\mathbf{r}}(\mathbf{r})}.$$

[15]For a thorough treatment of PC single user detection see the paper by Viterbi, [25].

[16]Note that for large phase uncertainties, the estimator derived in this section reduces to the non-coherent joint weight/phase estimator. See Appendix B.

[17]The derivation of this section is very much like that done in Section 4.1 for the NC joint weight/phase estimator. The only differences for the PC case is that we must also estimate user K's weight (b_K) and that the prior distribution on ϕ_K is not uniform. The reader may skip over the details of this derivation and pick up with the final form of the PC joint weight/phase estimator shown in Equations (4.40), (4.41) and (4.42).

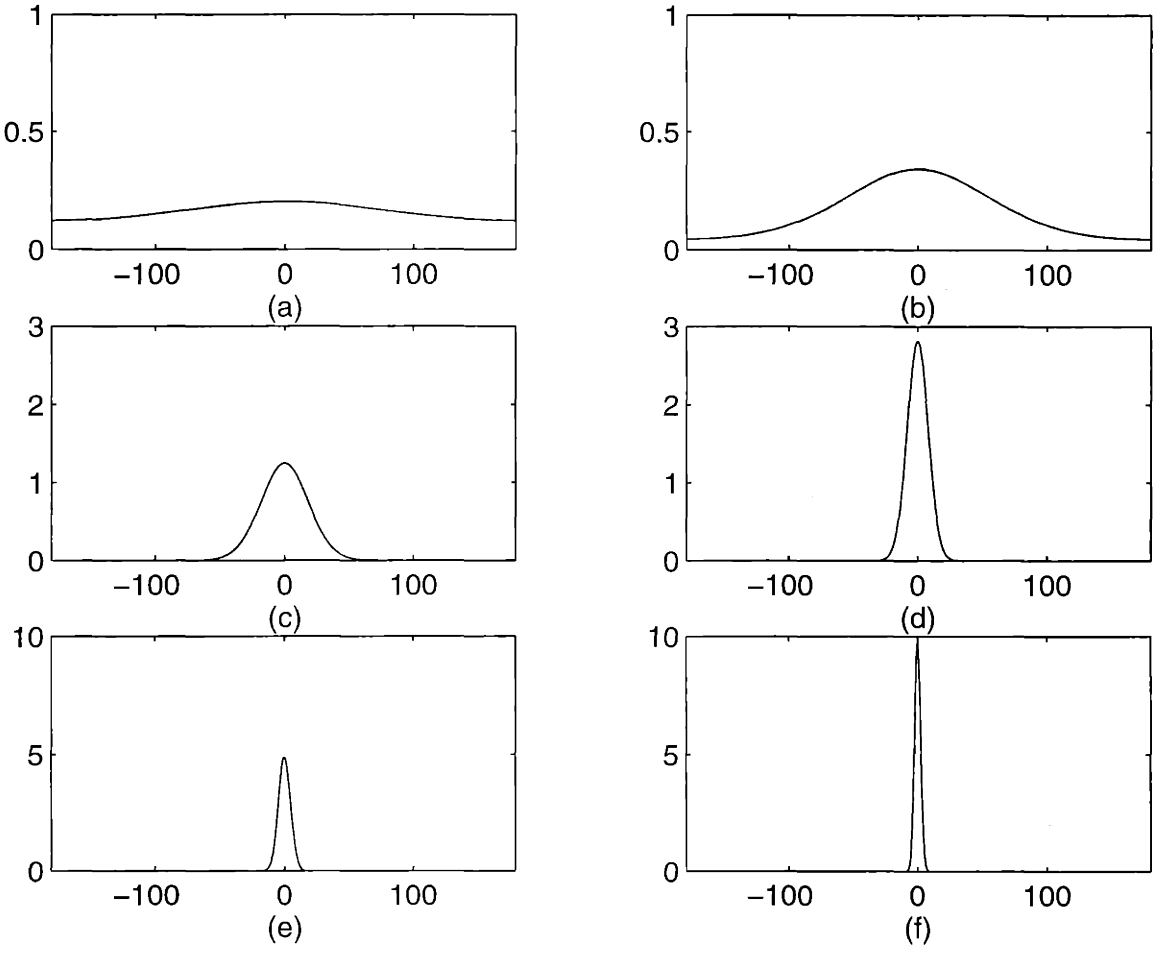


Figure 4-5: The exponential cosine PDF is assumed to be the distribution for ϕ_K, where $p_{\phi_K}(\phi) = \frac{e^{\alpha \cos \phi}}{2\pi I_o(\alpha)}$, $-180^o \leq \phi_K < 180^o$. This PDF is shown for six different values of α. (a) $\alpha = 0.25$. Note that as α approaches zero the PDF approaches a uniform distribution. (b) $\alpha = 1$ (c) $\alpha = 10$ (d) $\alpha = 50$ (e) $\alpha = 150$ (f) $\alpha = 600$ Note that as α approaches infinity the PDF approaches a Dirac.

Since $p_{\mathbf{r}}(\mathbf{r})$ is independent of $\mathbf{b}$, b_K and ϕ and $Prob(\mathbf{b}, b_K)$ is equally likely for all realizations of the weights, the MAP partially coherent joint weight/phase estimator is

$$\{\hat{\mathbf{b}}, \hat{b}_K, \hat{\phi}_K\} = \arg \max_{\{\mathbf{b}, b_K, \phi\}} \Omega_1(\mathbf{b}, b_K, \phi|\mathbf{r}),$$

where

$$\Omega_1(\mathbf{b}, b_K, \phi|\mathbf{r}) = p_{\mathbf{r}|\mathbf{b}, b_K, \phi_K}(\mathbf{r}|\mathbf{b}, b_K, \phi) p_{\phi_K}(\phi),$$

and

$$p_{\mathbf{r}|\mathbf{b}, b_K, \phi_K}(\mathbf{r}|\mathbf{b}, b_K, \phi) p_{\phi_K}(\phi) = \frac{e^{-\frac{1}{2\sigma^2}[||\Re\{d(\mathbf{b}, b_K, \phi)\}||^2 + ||\Im\{d(\mathbf{b}, b_K, \phi)\}||^2]}}{(\sqrt{2\pi}\sigma)^{2N}} \frac{e^{\alpha \cos \phi}}{2\pi I_o(\alpha)}, \quad (4.35)$$

where

$$d(\mathbf{b}, b_K, \phi) = (\mathbf{r} - \mathbf{S}\Phi\mathbf{b} - b_K \mathbf{s}_K e^{j\phi}).$$

We take the log of Equation (4.35) and discard any constant terms to obtain

$$\Omega_2(\mathbf{b}, b_K, \phi|\mathbf{r}) = -\frac{1}{2\sigma^2}[||\Re\{d(\mathbf{b}, b_K, \phi)\}||^2 + ||\Im\{d(\mathbf{b}, b_K, \phi)\}||^2] + \alpha \cos \phi.$$

The same substitutions and algebra used to reduce Equation (4.8) to Equation (4.16) will give

$$\begin{aligned} \Omega_2(\mathbf{b}, b_K, \phi|\mathbf{r}) &= -\tfrac{F_K(\mathbf{b}|\mathbf{r})}{\sigma^2} - \tfrac{\varepsilon_K(b_K)}{\sigma^2} \\ &\quad + \tfrac{2}{\sigma^2}\Re\{\mathcal{Y}(\mathbf{b}, b_K|\mathbf{r})\} \cos \phi + \tfrac{2}{\sigma^2}\Im\{\mathcal{Y}(\mathbf{b}, b_K|\mathbf{r})\} \sin \phi \quad (4.36) \\ &\quad + \alpha \cos \phi, \end{aligned}$$

where $F_K(\mathbf{b}|\mathbf{r})$ is defined in Equation (4.12) and

$$\boxed{\varepsilon_K(b_K) = ||b_K \mathbf{s}_K e^{j\phi}||^2 = |b_k|\,||\mathbf{s}_K||^2} \tag{4.37}$$

and

$$\boxed{\mathcal{Y}(\mathbf{b}, b_K|\mathbf{r}) = b_K^* \mathbf{s}_K^T(\mathbf{r} - \mathbf{S}\Phi\mathbf{b}).} \tag{4.38}$$

The MAP joint weight/phase estimate may be found in two steps:

$$\{\hat{\mathbf{b}}, \hat{b}_K, \hat{\phi}_K\} = \arg \max_{\{\mathbf{b}, b_K\}} \max_{\phi} \Omega_2(\mathbf{b}, b_K, \phi|\mathbf{r}).$$

First, set

$$\frac{\partial}{\partial \phi}\Omega_2(\mathbf{b}, b_K, \phi|\mathbf{r}) = 0$$

to find

$$\hat{\phi}_K(\mathbf{b}, b_K|\mathbf{r}) = \tan^{-1} \frac{\frac{2}{\sigma^2}\Im\{\mathcal{Y}(\mathbf{b}, b_K|\mathbf{r})\}}{\frac{2}{\sigma^2}\Re\{\mathcal{Y}(\mathbf{b}, b_K|\mathbf{r})\} + \alpha}. \tag{4.39}$$

The parameter α does exactly what we expect, namely, when α is large, $\hat{\phi}_K(\mathbf{b}, b_K|\mathbf{r})$ approaches zero, i.e., our knowledge of user K's phase is nearly perfect. When α is small, $\hat{\phi}_K(\mathbf{b}, b_K|\mathbf{r})$ approaches the estimator of Equation (4.19) which assumes no knowledge of user K's phase.

Second, we plug Equation (4.39) into Equation (4.36) and multiply by -1 to get

$$\Omega_3(\mathbf{b}, b_K|\mathbf{r}) = -\Omega_2(\mathbf{b}, b_K, \phi = \hat{\phi}_K(\mathbf{b}, b_K|\mathbf{r})|\mathbf{r})$$

or

$$\Omega_3(\mathbf{b}, b_K|\mathbf{r}) = \frac{F_K(\mathbf{b}|\mathbf{r})}{\sigma^2} + \frac{\varepsilon_K(b_K)}{\sigma^2} - \sqrt{(\frac{2}{\sigma^2}\Re\{\mathcal{Y}(\mathbf{b}, b_K|\mathbf{r})\} + \alpha)^2 + (\frac{2}{\sigma^2}\Im\{\mathcal{Y}(\mathbf{b}, b_K|\mathbf{r})\})^2}.$$

Noting that α is real and

$$\Re\{\frac{2}{\sigma^2}\mathcal{Y}(\mathbf{b}, b_K|\mathbf{r}) + \alpha\}^2 + \Im\{\frac{2}{\sigma^2}\mathcal{Y}(\mathbf{b}, b_K|\mathbf{r}) + \alpha\}^2 = |\frac{2}{\sigma^2}\mathcal{Y}(\mathbf{b}, b_K|\mathbf{r}) + \alpha|^2,$$

gives

$$\boxed{\Omega_3(\mathbf{b}, b_K|\mathbf{r}) = \frac{F_K(\mathbf{b}|\mathbf{r})}{\sigma^2} + \frac{\varepsilon_K(b_K)}{\sigma^2} - |\frac{2}{\sigma^2}\mathcal{Y}(\mathbf{b}, b_K|\mathbf{r}) + \alpha|.} \tag{4.40}$$

The PC joint weight/phase estimation procedure would first solve

$$\boxed{\{\hat{\mathbf{b}}, \hat{b}_K\} = \arg\min_{\{\mathbf{b}, b_K\}} \Omega_3(\mathbf{b}, b_K|\mathbf{r})} \tag{4.41}$$

and then plug these weight estimates into Equation (4.39) to get

$$\boxed{\hat{\phi}_K = \tan^{-1} \frac{\Im\{\mathcal{Y}(\hat{\mathbf{b}}, \hat{b}_K|\mathbf{r})\}}{\Re\{\mathcal{Y}(\hat{\mathbf{b}}, \hat{b}_K|\mathbf{r})\} + \alpha\frac{\sigma^2}{2}}.} \tag{4.42}$$

This detector is, not surprisingly, very similar to the joint weight phase estimator derived in Section 4.1. There are two slight differences. First, the NC joint weight/phase estimator assumed that $b_K = +1$. The PC joint weight/phase estimator assumes b_K to be a discrete complex valued random variable, hence the K^{th} user's energy, $\varepsilon_K(b_K)$, must be taken into account by the estimator.[18] Second, the

[18] If $b_K \in \{e^{\theta}|\theta = \theta_1, \ldots, \theta_M\}$ (this is uniform envelope signaling) then $\varepsilon_K(b_K)$ would be indepen-

PC joint weight/phase estimator, by definition, assumes a non-uniform prior PDF on ϕ_K, hence the phase accuracy parameter, α, must be taken into account by the estimator.

A qualitative example of the binary weight decision regions carved out by the PC weight/phase estimator is shown in Figure 4-6. The signature signal for user 1 is completely known at the receiver and is depicted in Figure 4-6-(a) as a complex scalar having only a real part. The signature for user 2 is depicted in Figure 4-6-(b) and has a phase which is known to be $\frac{\pi}{3}$ within a phase error. This phase error is represented as the arc in the figure and is characterized by the PDF of Equation (4.34) with $\alpha = 10$. The PDF for this phase error is shown in Figure 4-5-(c). With non-negligible probability, user 2's signature could lie anywhere along the 100° arc shown in Figure 4-6-(b).[19] The length of this arc exactly corresponds to the range of ϕ centered underneath the lobe of the PDF (shown in Figure 4-5-(c) for which $\alpha = 10$) that holds 99% of the probability. If each user is transmitting its signature signal with either a +1 weight or a -1 weight, the collection of all possible noiseless received points, $r' = b_1 s_1 + b_2 s_2 e^{j\phi_2}$, is shown in Figure 4-6-(c). The noiseless received point could lie anywhere on the four arcs shown in Figure 4-6-(c). The optimal PC weight estimator would draw decision boundaries similar to the ones shown in Figure 4-6-(d).[20] For example, if any received point, $r = r' + \sigma n$, falls in the upper left region marked $(-1, +1)$, the PC joint weight/phase estimator would decide that $\hat{b}_1 = -1$ and $\hat{b}_2 = +1$. Figures 4-7-(a) and 4-7-(b) show the binary weight decision regions if the 99% probability arc for user 2's phase was decreased to 45° and 30° by increasing α to 50 and 150, respectively.

If user signatures were to exhibit tree structure, the PC joint estimator reduces

---

dent of b_K and could be dropped.

[19]Note that a random phase having the PDF shown in Figure 4-6-(b) for $\alpha = 10$ will, with 99% (95%) probability, fall within an arc of 100° (74.5°).

[20]The boundaries shown in the figure have been drawn free hand and have not been mathematically calculated. The purpose of this figure is to offer intuition on the PC estimator, hence, performing the difficult calculations needed to obtain the mathematically exact boundary regions is not necessary.

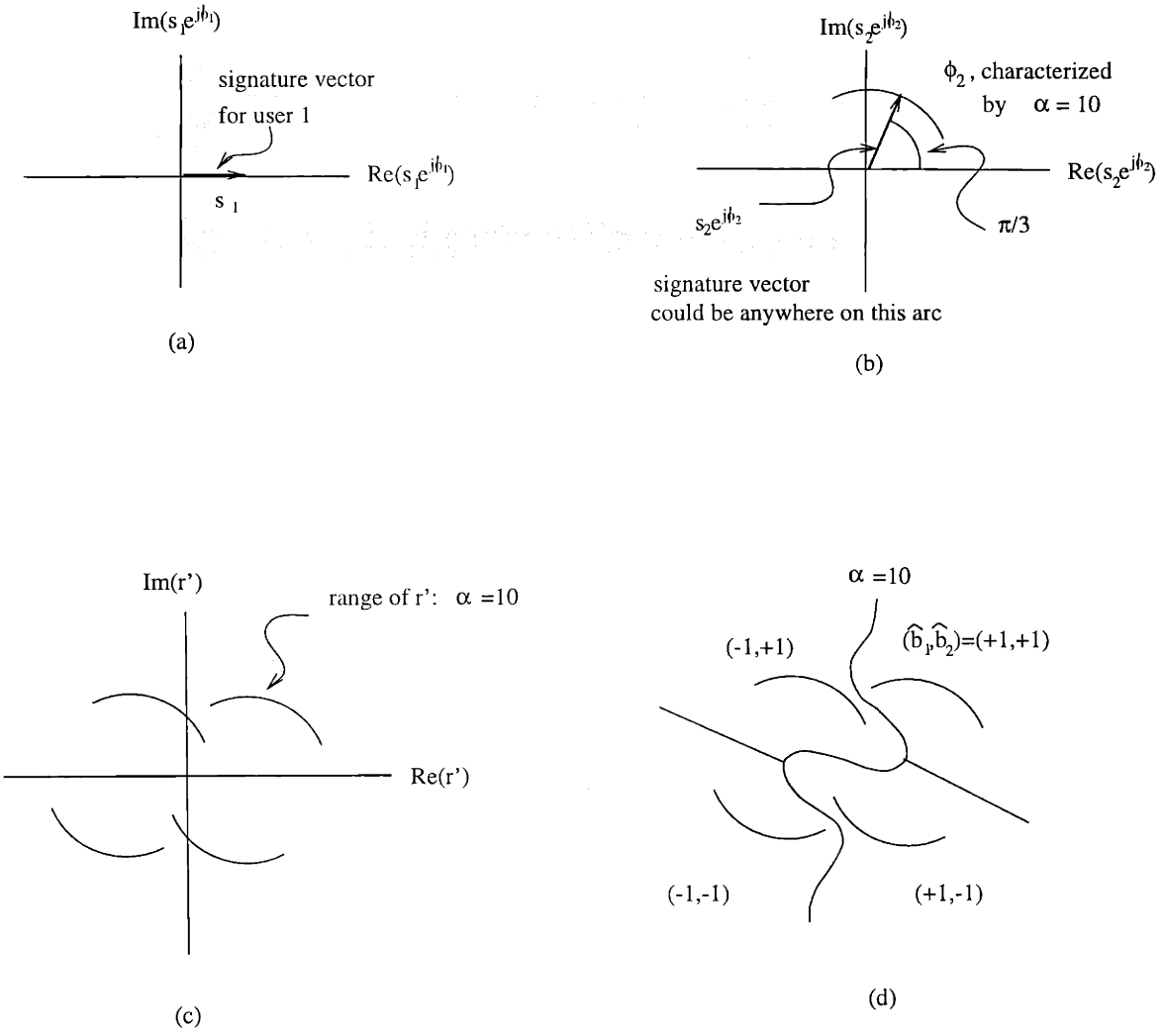


Figure 4-6: (a) The signature for user 1 is shown as complex scalar having a non-zero real part and zero imaginary part. (b) The signature signal for user 2 is known within a phase error. This phase error is characterized by representing the signature of user 2 as a complex scalar with an unknown phase having the PDF of Equation (4.34) with $\alpha = 10$. User 2's signature (complex scalar) could lie anywhere along the 100° arc with 99% probability. (c) The range of $r' = b_1 s_1 + b_2 s_2 e^{j\phi_2}$. This is the collection of all possible values for r'. (d) The binary weight decision boundaries that would be carved out by the optimal PC joint weight/phase estimator are super-imposed on the range of r'. For example, if $r = r' + \sigma n$ were to lie in the upper right region marked $(\hat{b}_1, \hat{b}_2) = (+1, +1)$, the PC detector would decide that r' must have come from the upper right arc, corresponding to $b_1 = +1$ and $b_2 = +1$.

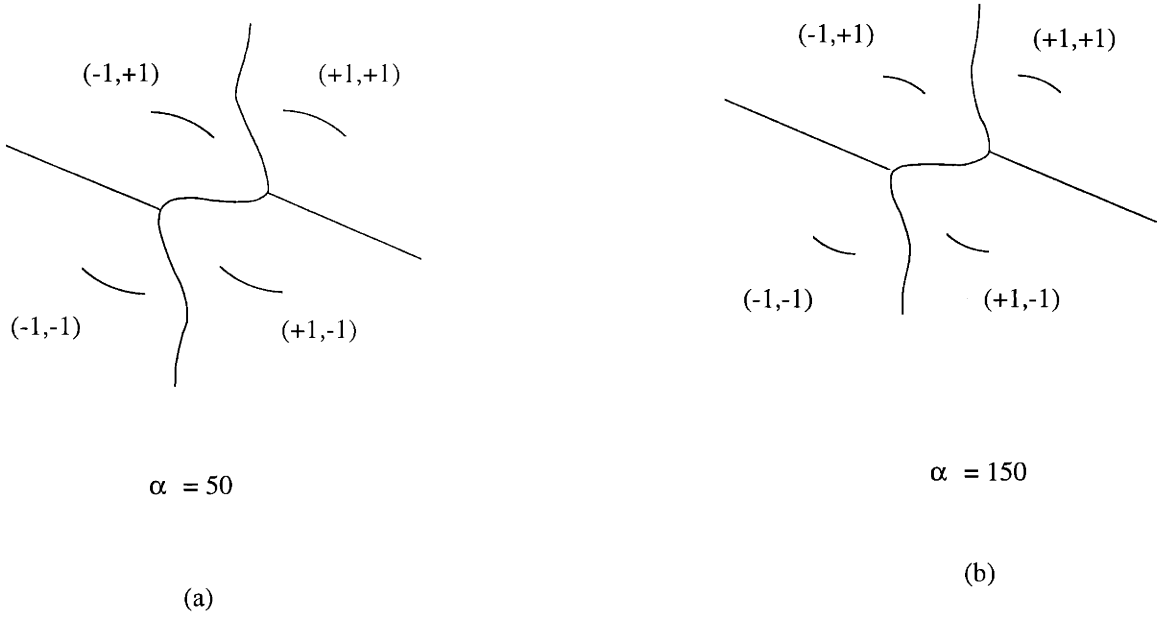


Figure 4-7: Binary weight decision boundaries that would be carved out by the optimal PC joint weight/phase estimator if α in Figure 4-6 were changed. The arc lengths shown are the 99% probability regions corresponding to each value of α. (a)$\alpha = 50$ (45° arc) (b) $\alpha = 150$ (30° arc). See Figure 4-5 for the PDF corresponding to these values of α.

in exactly the same manner as does the NC joint estimator for uniformly distributed phase shown in Section 4.2.[21] It would sweep up the tree creating a standard table at each node that is neither an ancestor nor descendant of the uncertain-phase user. The only trivial difference between the NC tree procedure and the PC tree procedure is the replacement of the $\mathcal{X}$-table with a table of $\mathcal{Y}(\mathbf{b}_{adK}, b_K|\mathbf{r})$, or a **$\mathcal{Y}$-table**. Just as with $\mathcal{X}(\mathbf{b}_{adK}|\mathbf{r})$ for the NC joint weight/phase estimator, $\mathcal{Y}(\mathbf{b}_{adK}, b_K|\mathbf{r})$ is a function of the values of the weights corresponding to it and its ancestors and descendants. The $\mathcal{Y}$-table has a number of entries which is exponential in the number of ancestors and descendants *plus one*, since we are now estimating b_K. Also at this node, the algorithm creates a descendant table of conditional estimates $[\hat{\mathbf{b}}_{dK}, b_K]$. This requires a number of compares that is also exponential in the number of ancestors and descendants plus one. The algorithm proceeds exactly as it did for the NC tree procedure where an ancestor table is created at each node that is an ancestor to the node corresponding to the uncertain-phase user.

The computational complexity is virtually the same as that derived in Section 4.3. The only difference is due to the estimation of the uncertain-phase user's weight (the NC joint weight/phase estimator required this weight to be fixed).[22] The resulting trivial change in complexity does not change the order of the computational complexity of the algorithm as a function of K, the number of users.

Simulation results of the PC weight/phase estimator are reported in Sections 4.7.1 and 4.7.2.

---

[21] Note that the two user example given above does not show a tree structured set of user signatures. For any full tree, no two users would lie entirely in the same complex plane, hence, no two rings would intersect. The close proximity of arcs in Figures 4-6 and 4-7 is not typical of tree structured signature sets.

[22] The complexity of the PC joint weight/phase estimator can be obtained by replacing the M^{L_n-1} in the first term of Equation (4.31) with M^{L_n} and adding $(M-1)M^{L_n}$ to Equation (4.29). If the receiver knew the value of b_K, the complexity of the NC joint weight/phase estimator and the PC joint weight/phase estimator would be exactly the same.

4.5 Optimal PC Joint Weight-only Estimation

Partially coherent detection may also be done without estimating user K's phase.[23] The PC weight estimator finds the value of $[\mathbf{b}, b_K]$ to maximize

$$p_{\mathbf{b},b_K|\mathbf{r}}(\mathbf{b}, b_K|\mathbf{r}) = \int_{-\pi}^{\pi} p_{\mathbf{b},b_K,\phi_K|\mathbf{r}}(\mathbf{b}, b_K, \phi|\mathbf{r})d\phi.$$

Since

$$p_{\mathbf{b},b_K|\mathbf{r}}(\mathbf{b}, b_K|\mathbf{r}) = p_{\mathbf{r}|\mathbf{b},b_K}(\mathbf{r}|\mathbf{b}, b_K)\frac{Prob(\mathbf{b}, b_K)}{p(\mathbf{r})},$$

and $p(\mathbf{r})$ is independent of $\mathbf{b}$ and b_K, and $Prob(\mathbf{b}, b_K)$ is equally likely for all realizations of the weights, the MAP joint weight PC estimator is

$$\{\hat{\mathbf{b}}, \hat{b}_K\} = \arg \max_{\{\mathbf{b},b_K\}} \Upsilon_1(\mathbf{b}, b_K|\mathbf{r}),$$

where

$$\Upsilon_1(\mathbf{b}, b_K|\mathbf{r}) = \int_{-\pi}^{\pi} p_{\mathbf{r}|\mathbf{b},b_K,\phi_K}(\mathbf{r}|\mathbf{b}, b_K, \phi)p_{\phi_K}(\phi)d\phi.$$

and $p_{\mathbf{r}|\mathbf{b},b_K,\phi_K}(\mathbf{r}|\mathbf{b}, b_K, \phi)p_{\phi_K}(\phi)$ is defined in Equation (4.35).

$$\Upsilon_1(\mathbf{b}, b_K|\mathbf{r}) = Ce^{\frac{-(F_K(\mathbf{b}|\mathbf{r})+\varepsilon_K(b_K))}{\sigma^2}} \int_{-\pi}^{\pi} e^{(\frac{2}{\sigma^2}\Re\{\mathcal{Y}(\mathbf{b},b_K|\mathbf{r})\}+\alpha)\cos\phi+\frac{2}{\sigma^2}\Im\{\mathcal{Y}(\mathbf{b},b_K|\mathbf{r})\}\sin\phi}d\phi,$$

where C holds all the multipliers that are independent of $\mathbf{b}$ and b_K. See Equations (4.12), (4.13), and (4.38) for definitions of $F_K(\mathbf{b}|\mathbf{r})$, $\varepsilon_K(b_K)$ and $\mathcal{Y}(\mathbf{b}, b_K|\mathbf{r})$. Recognizing that the integral above is a modified Bessel function, we can write

$$\Upsilon_1(\mathbf{b}, b_K|\mathbf{r}) = Ce^{\frac{-[F_K(\mathbf{b}|\mathbf{r})+\varepsilon_K(b_K)]}{\sigma^2}} I_o(\sqrt{(\frac{2}{\sigma^2}\Re\{\mathcal{Y}(\mathbf{b}, b_K|\mathbf{r})\} + \alpha)^2 + (\frac{2}{\sigma^2}\Im\{\mathcal{Y}(\mathbf{b}, b_K|\mathbf{r})\})^2}).$$

[23] Note that for very small phase uncertainties, this estimator reduces to the coherent joint weight/phase estimator. See Appendix B.

Maximizing $\Upsilon_1(\mathbf{b}, b_K|\mathbf{r})$ over $[\mathbf{b}, b_K]$ will result in the same weight estimates as minimizing its negative log, hence the PC weight-only estimator is

$$\boxed{\{\hat{\mathbf{b}}, \hat{b}_K\} = \arg\min_{\{\mathbf{b}, b_K\}} \Upsilon_2(\mathbf{b}, b_K|\mathbf{r})} \tag{4.43}$$

where

$$\boxed{\Upsilon_2(\mathbf{b}, b_K|\mathbf{r}) = \frac{F_K(\mathbf{b}|\mathbf{r}) + \varepsilon_K(b_K)}{\sigma^2} - \ln\, I_o(\tfrac{2}{\sigma^2}|\mathcal{Y}(\mathbf{b}, b_K|\mathbf{r}) + \alpha|)} \tag{4.44}$$

where we used

$$\Re\{\mathcal{Y}(\mathbf{b}, b_K|\mathbf{r}) + \alpha\}^2 + \Im\{\mathcal{Y}(\mathbf{b}, b_K|\mathbf{r}) + \alpha\}^2 = |\mathcal{Y}(\mathbf{b}, b_K|\mathbf{r}) + \alpha|^2.$$

It is easy to see that both the PC weight-only estimator above and the PC weight/phase estimator in Equations (4.41) and (4.40) have the same form. Both, therefore, exhibit the same reduction when user signatures are tree-structured. The computational complexity of the PC weight-only estimator, then, is at least that of the PC weight/phase estimator. The estimation procedures are the same, except where the PC weight/phase estimator calculates

$$\sqrt{(\frac{2}{\sigma^2}\Re\{\mathcal{Y}(\mathbf{b}, b_K|\mathbf{r})\} + \alpha)^2 + (\frac{2}{\sigma^2}\Im\{\mathcal{Y}(\mathbf{b}, b_K|\mathbf{r})\})^2},$$

the PC weight-only estimator does the same calculation, but must do one extra step. The PC weight-only estimator must use this value as the argument to a modified Bessel function; this requires a look-up in a Bessel table. To create a single entry at an ancestor node to the uncertain-phase user, the PC weight-only estimator must do M Bessel look-ups. Since the number of ancestors of the uncertain-phase user is typically low, this additional computational complexity will be negligible. If a

system is low on storage space for its detection algorithm, then the PC weight/phase estimator would be preferable to the PC weight-only estimator.

4.6 Sub-optimal Lower Complexity Alternatives

As found in Section 4.3, both the non-coherent and partially coherent estimators have a complexity that can range from very low order polynomial in the number of users, to a complexity which is impractical, depending upon the position of the uncertain-phase user in the tree. If the tree position of the uncertain-phase user cannot be controlled to keep the complexity low, there are two low complexity alternatives:

1. If, at the receiver, a user's phase is known within a small error, no phase estimation is needed and the coherent detector may be used with slightly incorrect phase information.

2. If the phase uncertainty cannot be ignored, a sub-optimal, lower complexity version of the optimal joint weight/phase estimator may be used.

This section discusses both options of avoiding the high complexity of optimal joint weight/phase estimation.

As was seen in Section 3.7.2, ignoring phase inaccuracies works well under some conditions. Section 4.6.1 formally presents the **assumed coherent weight estimator** that was used for the simulations of Section 3.7.2 in which phase uncertainty was ignored and the coherent tree joint detector was used with an incorrect value for the phase.[24] In the process of developing this detector, intuition is given to explain the simulations results of Section 3.7.2.

Section 4.6.2 proposes a sub-optimal version of the optimal joint weight/phase estimator which is much lower in complexity than its optimal counterpart. The low

[24]Recall that the complexity for the coherent tree joint detector is typically low order polynomial in the number of users.

complexity version makes it possible for an uncertain-phase user to be near the top of the tree. This sub-optimal approach assumes that the random phase is from a discrete set of possible phases to allow it choose a weight vector estimate with low complexity, although the final phase estimate is not from the discrete set. This **assumed discrete joint weight/phase estimator** is developed and its complexity is calculated.

4.6.1 Assumed Coherent Weight Estimator

Section 3.7.2 empirically found that for the equal energy minimum distance signature sets, a phase error in one user of 20° resulted in a notable performance loss (up to 1 dB, for some users in the system).[25] A phase mismatch for one user of 10° to 15° resulted in virtually no performance loss. This section formally presents the **assumed coherent (AC) weight estimator** and, in the process, discusses the intuition behind its behavior.

The AC weight estimator finds the combination of valid weights, $[\mathbf{b}, b_k]$ that minimizes the distance between the received signal vector, $\mathbf{r}$, and the *assumed* possible aggregate signal vector $\mathbf{r}' = \mathbf{S}\Phi\mathbf{b} + b_K \mathbf{s}_K e^{j\tilde{\phi}_K}$, where $\tilde{\phi}_K$ is *assumed* to be the accurate value for user K's phase. The AC weight estimator is

$$\{\hat{\mathbf{b}}, \hat{b}_K\} = \arg \min_{\{\mathbf{b}, b_K\}} ||\mathbf{r} - \mathbf{S}\Phi\mathbf{b} + b_K \mathbf{s}_K e^{j\tilde{\phi}_K}||^2. \tag{4.45}$$

The only case in which this detector is optimal is if perfect knowledge of user K's phase were communicated to the receiver. Since this is, strictly, never the case, the AC weight estimator of Equation (4.45) is always sub-optimal. As was shown in Section 4.4, the PC weight/phase estimator is the optimal estimator for ϕ_K random with the PDF of Equation (4.34).

There are conditions under which the difference between the sub-optimal AC

[25] Losses are with respect to the performance of a corresponding system for which all users' phases are correctly known at the receiver.

weight estimator and the optimal PC weight estimator range from tolerable to negligible. This concept can be illustrated with an example. Consider, again, the two user signatures shown in Figures 4-6-(a) and 4-6-(b). Assume that our partial knowledge of user 2's signature is due to some estimation procedure such as the NC estimator described in Section 4.1. Assume that the output of this procedure was an estimate for user 2's phase, $\hat{\phi}_2 = \frac{\pi}{3}$. This estimate is not entirely accurate; Figure 4-6-(b) conveys this inaccuracy by allowing user 2's signature to lie anywhere along the arc. The AC weight estimator would ignore the possibility of the estimate being in error and would set $\tilde{\phi}_2 = \hat{\phi}_2 = \frac{\pi}{3}$. Figures 4-8-(a) and 4-8-(b) show the known signature for user 1 and the assumed signature for user 2, respectively. Each user is allowed to transmit an information weight, of either a $+1$ or -1, therefore the received constellation, or set of assumed possible received points, $r' = b_1 s_1 + b_2 s_2 e^{j\frac{\pi}{3}}$, is the collection of four points shown in Figure 4-8-(c). Assuming this four point received constellation is correct, the optimal coherent joint weight estimator would carve out the decision regions shown in Figure 4-8-(d). The AC joint weight estimator, then, uses the decision regions of Figure 4-8-(d) in place of the optimal decision regions that would be carved out by the PC joint weight/phase estimator shown in Figure 4-6-(d). The decision region boundaries of the AC joint weight estimator are not a function of α, hence, they remain constant with respect to the uncertainty of the estimate. If the uncertainty is large, the replacement of the optimal decision regions with the sub-optimal regions of will yield unsatisfactory weight error performance. These two decision regions are superimposed in Figure 4-9-(a) for $\alpha = 10$. Notice that the AC decision region boundaries shown as dashed lines are an approximation to the optimal PC decision boundaries shown as solid lines. As α is increased and the uncertainty arc is shortened, the differences between the AC decision boundaries and the optimal PC decision boundaries is decreased. Figures 4-9-(b),(c), and (d) show both the PC and AC boundaries for $\alpha = 50, 150, 600$, respectively.

As can be concluded from the above example, the performance of the sub-optimal

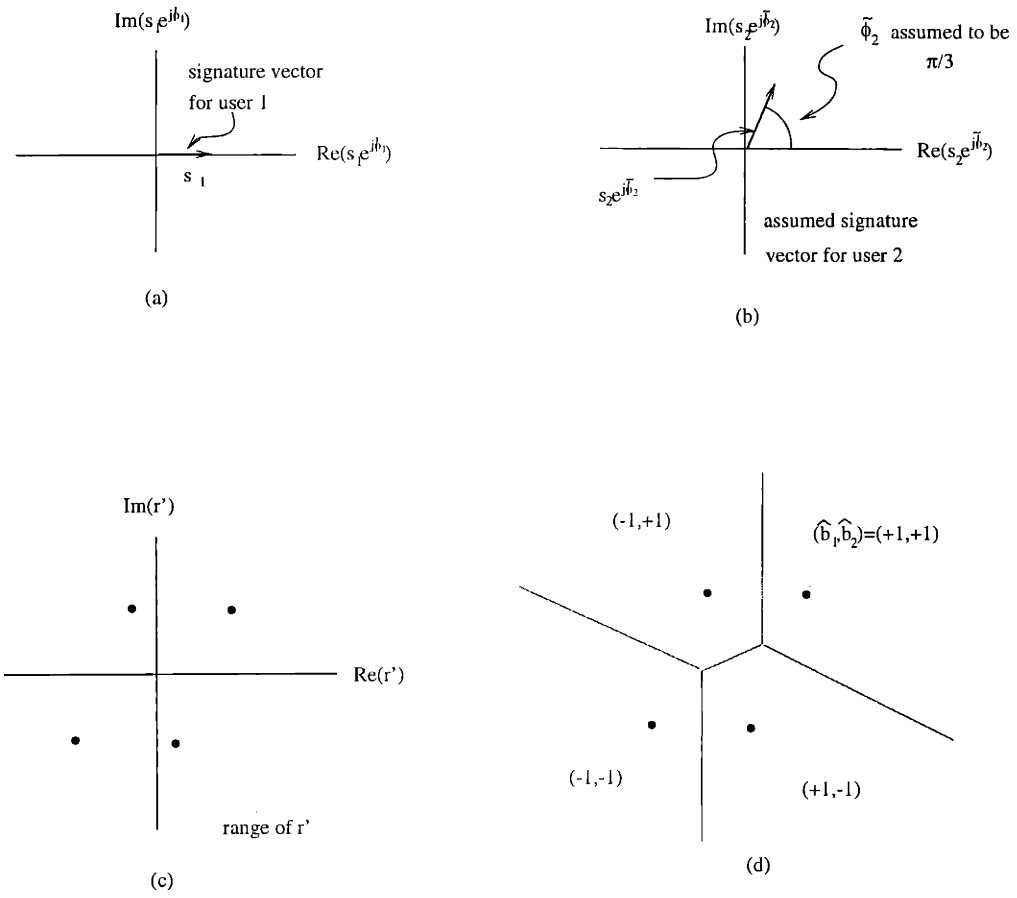


Figure 4-8: An example of the AC joint weight estimator. (a) Complex scalar signature for user 1. (b) Assumed complex scalar signature for user 2. The signature is actually known within a phase error, but the AC joint weight detector ignores this uncertainty. (c) The range of $r' = b_1 s_1 + b_2 s_2 e^{j\frac{\pi}{3}}$. This is the collection of all possible values for r' if the phase were actually $\frac{\pi}{3}$. (d) Binary weight decision boundaries that would be carved out by the sub-optimal AC joint weight estimator. For example, if $r = r' + \sigma n$ were to lie in the region marked $(\hat{b}_1, \hat{b}_2) = (+1, +1)$, the AC detector would decide that r' must have been the upper right point, corresponding to $b_1 = +1$ and $b_2 = +1$.

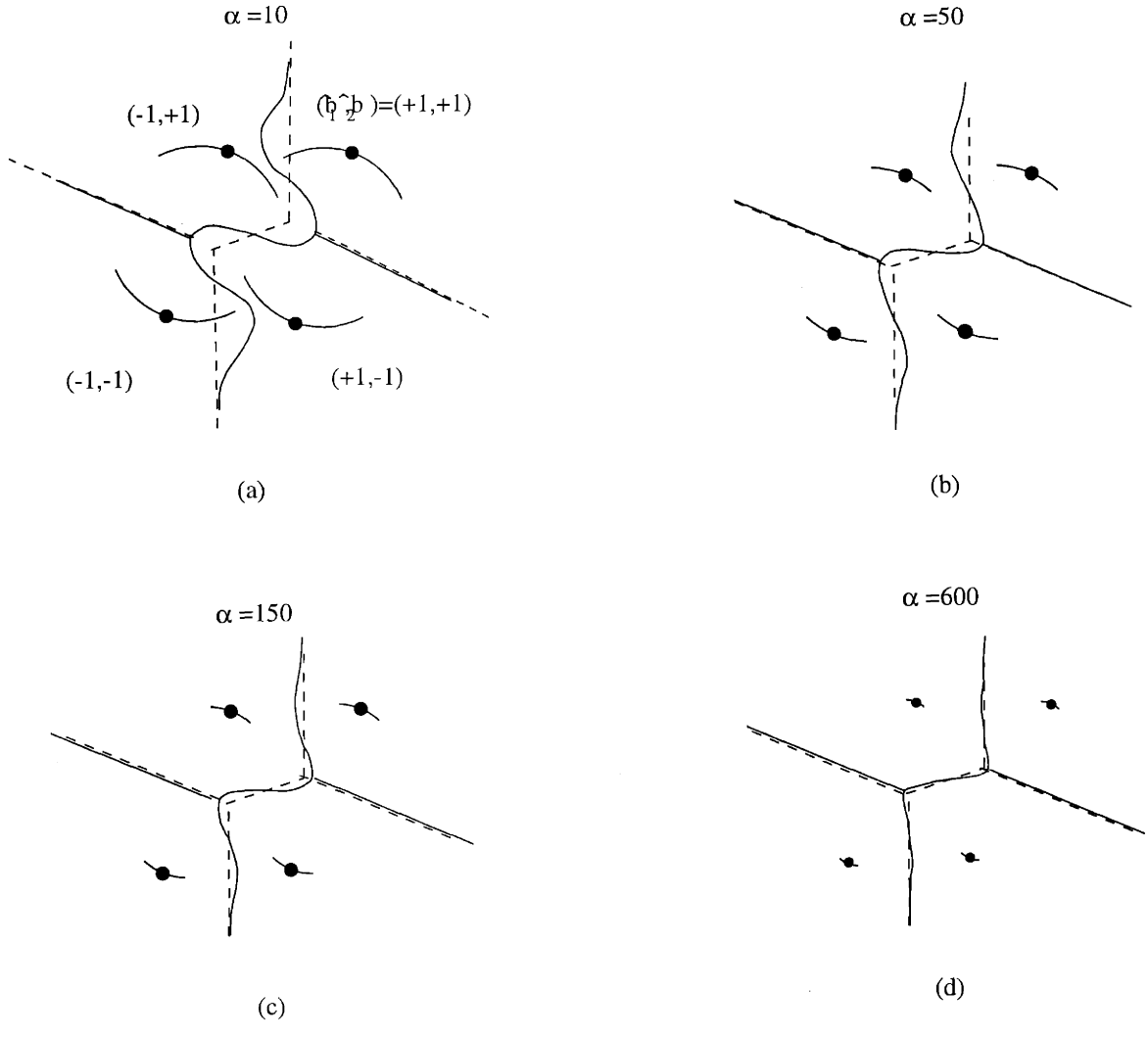


Figure 4-9: Decision boundaries for AC (PC) weight estimator shown as a dashed (solid) line. The PC weight estimator is optimal. The set of all possible aggregate transmitted vectors is indicated by the four arcs in each figure, while the set of assumed possible transmitted vectors is indicated by the four points in each figure. The accuracy parameter (α) is increased in each successive figure, causing the arcs to shrink from figure to figure: (a) $\alpha = 10$ (100° arc) (b) $\alpha = 50$ (45° arc) (c) $\alpha = 150$ (30° arc) (c) $\alpha = 600$ (10° arc)

AC joint weight phase estimator compared to that of the optimal PC joint weight/phase estimator depends upon how well straight line decision boundaries can approximate the optimal decision boundaries of the PC estimator. The two user example, however, does not show a tree structured set of user signatures. In general, the range of $\mathbf{r}'$ for a tree-structured signature set would consist of arcs in N complex dimensions, where no set of four arcs would lie on the same complex plane. This would result in a more favorable spacing between arcs. Unfortunately, examples in more than 1 complex dimension are too difficult to illustrate in a figure.

4.6.2 Assumed Discrete Weight/Phase Estimator

As we learned from the simulations of Section 3.7.2, to maintain near optimal performance, the coherent joint detector requires a $\pm 5°$ phase accuracy in its knowledge of each user's phase. If the information concerning user K's phase is of this accuracy, there is no need to estimate user K's phase in addition to finding the joint weight estimate. The receiver, then, would employ the AC joint weight estimator described in the previous section, using the partial phase information as if it were complete. If, on the other hand, the information on user K's phase is not of this accuracy (i.e., if the phase is likely to be off by more than 5°), the receiver must jointly estimate the user weights and estimate user K's phase. The only options for joint weight/phase estimation given up to this point have been the optimal non-coherent and partially coherent joint weight/phase estimators of Sections 4.1 and 4.4. If the uncertain-phase user is at a level in the tree that will cause these optimal estimators to be computationally impractical, a sub-optimal alternative must be found.

This section presents a sub-optimal joint weight/phase estimator which assumes the uncertain phase to be from a discrete set. Since the uncertain phase is not a discrete random variable, the **assumed discrete (AD) joint weight/phase estimator** is not optimal. This section proposes the AD joint weight/phase estimator and gives its computational complexity.

The AD estimator replaces the continuous random variable, ϕ_n with a discrete random variable θ_n for which $\theta_n \in \Gamma_{\theta_n}$, and $|\Gamma_{\theta_n}| = M_{\theta_n}$. If $b_n \in \Gamma_n$, i.e., the actual weight of user n is chosen from the discrete set of complex scalars, Γ_n, then the **compound weight** for user n may be denoted by a_n for which $a_n \in \{b_n e^{j\theta_n} | b_n \in \Gamma_n, \ \theta_n \in \Gamma_{\theta_n}\}$. The AD joint weight/phase estimator, then, is equivalent to the tree joint detector of Chapter 3 that assumes the weight of user n to be this compound weight, a_n, having a total number of possibilities equal to $M_{\theta_n} M_n$, where $|\Gamma_n| = M_n$.

If ϕ_n is completely unknown, a training weight is sent, i.e., $b_n = +1$, and the AD joint weight/phase estimator discretizes the continuous range of the unknown phase $[-\pi, \pi]$ into a set of evenly spaced phases, $\Gamma_{\theta_n} = \{i\frac{2\pi}{M_{\theta_n}} | i = 0, 1, 2, ...(M_{\theta_n} - 1)\}$.

See Figure 4-10 for a two user binary signaling example. User 1's signature vector is shown in Figure 4-10-(a). User 2's signature vector is shown in Figure 4-10-(b). The phase of user 2's vector is not known, hence, s_2 could lie anywhere on the ring shown in Figure 4-10-(b). The noiseless received vector is $r' = b_1 s_1 + s_2 e^{j\phi_2}$, $b_1 \in \{+1, -1\}$ and $-\pi \leq \phi_2 < \pi$. The range of r' is shown in Figure 4-10-(c); the transmitted aggregate, r', could lie anywhere on one of the two rings shown in the figure. In this example, the AD estimator would assume that user 2's phase is a discrete random variable taking on any one of the 12 values in $\Gamma_{\theta_2} = \{i\frac{2\pi}{12} | i = 0, 2, 3, ...11\}$ with equal probability. The assumed range of r', then, would be the 24-point subset of the actual range of r' as shown in Figure 4-10-(d).

The AD estimator returns an estimate for $\mathbf{b}$ and θ_n. The AD estimate for θ_n, chosen from the set Γ_{θ_n}, can be refined by replacing it with

$$\hat{\phi}_n = \tan^{-1} \frac{\Im\{\mathcal{X}(\hat{\mathbf{b}}|\mathbf{r})\}}{\Re\{\mathcal{X}(\hat{\mathbf{b}}|\mathbf{r})\}},$$

where $\mathcal{X}(\hat{\mathbf{b}}|\mathbf{r})$ was defined in Equation (4.15).

Similarly, if the receiver has partial information about ϕ_n, the AD joint weight/phase estimator would discretize the continuous range on ϕ_n. Recall that this range can be

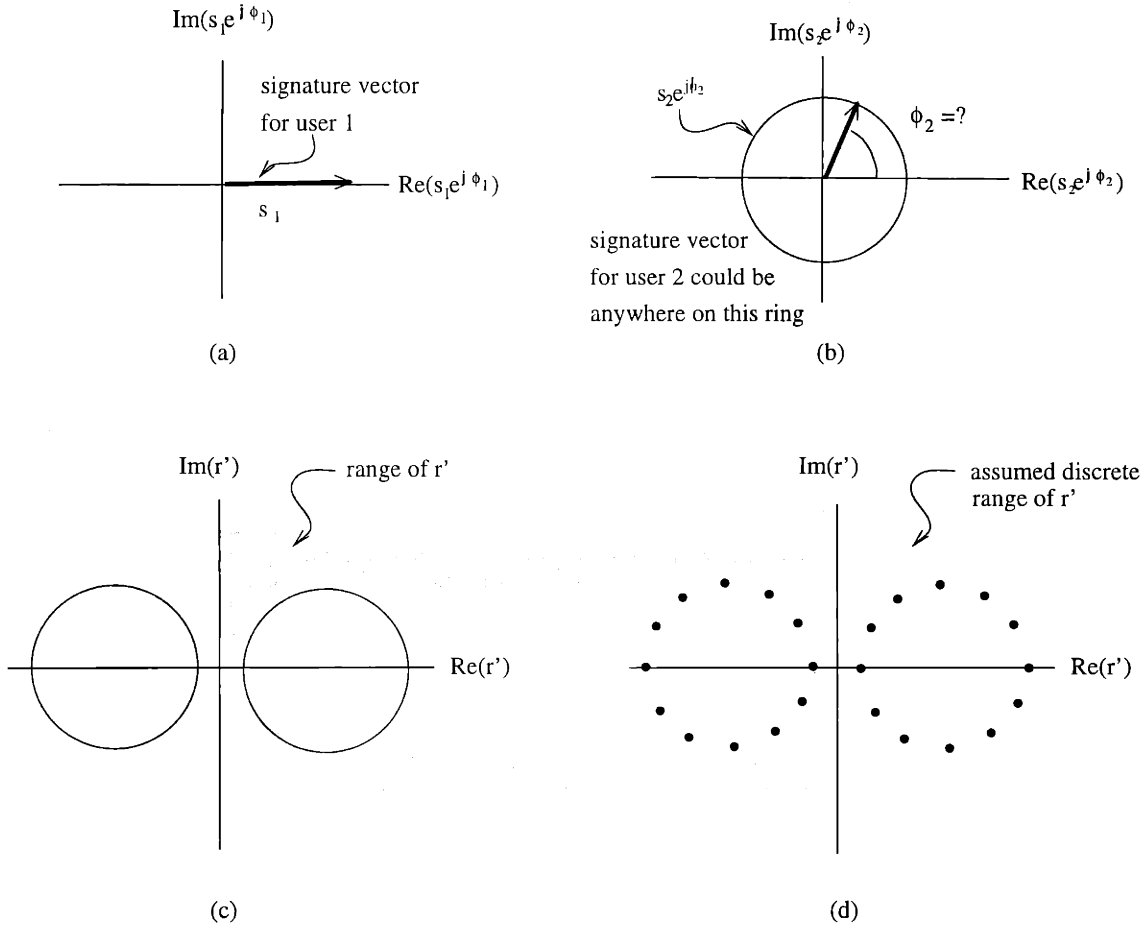


Figure 4-10: Example of two users in one complex dimension. User 1 has a known phase (0 radians) and user 2 has a uniformly distributed phase. (a) Signature vector of user 1, $\mathbf{s}_1$. (b) User 2 has a uniformly distributed phase, hence, its signature vector could lie anywhere on the ring. (c) Set of all possible points, $r' = b_1 s_1 + s_2 e^{j\phi_2}$, $b_1 \in \{+1, -1\}$, $\phi_2 \in [-\pi, \pi]$. (d) The *assumed discrete* range of r' when M_{θ_2}, the number of discrete possibilities for user 2's phase is chosen to be 12.

defined to be the values of ϕ that fall within the 99% probability range of $p_{\phi_n}(\phi) = \frac{e^{\alpha\cos(\phi-m_\phi)}}{2\pi I_o(\alpha)}$, where m_ϕ is the mean of the random phase ϕ_n.[26] If this range were denoted as $[\phi_{low}, \phi_{high}]$, the assumed discrete range would be $\Gamma_{\theta_n} = \{\phi_{low} + i\frac{(\phi_{high}-\phi_{low})}{M_{\theta_n}} | i = 0, 2, 3, ...(M_{\theta_n} - 1)\}$. The AD estimator returns an estimate for $[\mathbf{b}, b_n]$ and θ_n.[27] As was possible above, with the NC version of the AD joint weight/phase estimator, refinement of the phase estimate can be made with only one additional computation. The AD estimate $\hat{\theta}_n$ chosen from the set Γ_{θ_n} can be replaced with

$$\hat{\phi}_n = \tan^{-1}\frac{\Im\{\mathcal{Y}(\hat{\mathbf{b}}, \hat{b}_n|\mathbf{r})\}}{\Re\{\mathcal{Y}(\hat{\mathbf{b}}, \hat{b}_n|\mathbf{r}) + \frac{\sigma^2}{2}\alpha\}},$$

where $\mathcal{Y}(\mathbf{b}, b_n|\mathbf{r})$ was defined in Equation (4.38).

The total complexity of this procedure is easily found by noting that this procedure is the same as for the coherent tree joint detector of Chapter 3. The only difference between the AD procedure and the coherent procedure in which ϕ_n is known is that the size of each of the standard tables created at nodes $\{n, dn\}$ is multiplied by M_{θ_n}. This is the result of replacing $b_n \in \Gamma_n$ with $a_n \in \{b_n e^{j\theta_n} | b_n \in \Gamma_n, \quad \theta_n \in \Gamma_{\theta_n}\}$ in the coherent tree joint detector. The total complexity of the AD joint detector for which the unknown-phase user is at level L_n in a Q-tree having a total of L levels is

$$\begin{aligned} c_{AD}(L, L_n, Q, M, M_{\theta_n}) &= c_C(L, Q, M) - c'_C(L, L_n, Q, M) + M_{\theta_n} c'_C(L, L_n, Q, M) + 1 \\ &= c_C(L, Q, M) + (M_{\theta_n} - 1)c'_C(L, L_n, Q, M) + 1, \end{aligned} \tag{4.46}$$

where $c_C(L, Q, M)$ is the complexity of the coherent tree detector and is shown in Equation (4.28). Defined in Equation (4.29), $c'_C(L, L_n, Q, M)$ is the number of computations needed to fill the set of nodes $\{n, dn\}$ with standard tables, assuming user n

[26] Recall that the accuracy is reflected in the parameter α.

[27] The estimate, $[\hat{\mathbf{b}}, \hat{b}_n]$, corresponds to the arc that is believed to contain the actual transmitted aggregate signal.

has only M possible weights from which to choose. Similarly, $(M_{\theta_n} - 1)c'_C(L, L_n, Q, M)$ is the number of computations needed to fill the set of nodes $\{n, dn\}$ with standard tables if user n has $M_{\theta_n}M$ possible compound weights from which to choose. The one additional computation is due to the phase refinement calculation.

Since

$$c'_C(L, L_n, Q, M) \leq c_C(L, Q, M),$$

the complexity of the AD procedure is very loosely upper bounded by

$$c_{AD}(L, L_n, Q, M, M_{\theta_n}) < M_{\theta_n} c_C(L, Q, M) + 1.$$

Recall that in Section 3.5 $c_C(L, Q, M) = O(K^p)$, p small, K the total number of users. An example was given for $Q = 4$ that showed $c_C(L, Q, M) = O(K^{\frac{3}{2}})$. From this, we may conclude for typical trees that c_{AD} is, at most, $O(M_{\theta_n}K^p)$, p small. Hence, the suboptimal AD joint weight/phase estimator can be of very low complexity, for many practical values of M_{θ_n}.

Substituting Equations (4.28) and (4.29) into Equation (4.46) yields

$$c_{AD}(L, L_n, Q, M, M_{\theta_n}) = \frac{M-1}{QM-1}[(QM)^L - 1 + (M_{\theta_n} - 1)M^{L_n-1}((QM)^{L-L_n+1} - 1)]. \tag{4.47}$$

For $M_{\theta_n} = 10, 18$, and 24 the complexity, c_{AD}, is shown in Tables 4.7, 4.8, and 4.9 for various values of L_n and L. The assumed discrete points on the ring for user n lie 36° apart for $M_{\theta_n} = 10$, 20° apart for $M_{\theta_n} = 18$, and 15° apart for $M_{\theta_n} = 24$. Notice that none of the complexities in the tables are impractical, even for values of $M_{\theta_n} = 24$.[28] In particular, for $L_n = L - 3$, the complexity of the AD estimator is on the order of $O(10^4)$ in all three tables, whereas, the PC estimator for this same L_n and L would be $O(10^{25})$. Refer to Table 4.6 for the complexity of the PC estimator

[28]Note that if points on the ring are 15° apart, no point in the true range or $\mathbf{r}'$ is more than 7.5° from a point in the assumed discrete range of $\mathbf{r}'$.

for various values of L_n and L.

With a 15° separation between discrete phases, the actual phase is guaranteed to be within 7.5° of one of the discretized phases. It is important to note that the phase estimate given by the assumed-discrete weight/phase estimator will not, in general, give a phase estimate that is one of the discrete phase points. Recall that this estimator uses the set of discrete phases to assist in finding the weight estimates for the known-phase users. Once these are found, the phase estimation procedure is identical to the single user phase estimator.

Simulations of the assumed-discrete weight/phase estimator, including calculation of the phase estimate error standard deviation, are not done in this thesis. Since the discretization can be changed to bring the discrete phase separation down to a very small amount, this estimator is anticipated to be capable of giving very near optimal performance. In other words, if a level of discretization does not give near optimal performance, further reduction of the discrete phase separation may be done, at a cost of a relatively low increase in complexity.

Table 4.7: Computational complexity of the AD tree algorithm for binary signaling ($M = 2$) and a quad tree ($Q = 4$). Recall that L is the total number of levels in the tree, K is the total number of users (nodes on the tree), and L_n is the level at which the unknown-phase user sits on the tree. For this table, $M_{\theta_n} = 10$ (a point separation of 36°).

L	K	$c_{AD}(L_n = L)$	$c_{AD}(L_n = L-1)$	$c_{AD}(L_n = L-2)$	$c_{AD}(L_n = L-3)$
3	21	110	236	731	
4	85	658	910	1900	5851
5	341	4826	5330	7310	15212

Table 4.8: Computational complexity of the AD weight estimator for $M_{\theta_n} = 18$ (point separation of 20°).

L	K	$c_{AD}(L_n = L)$	$c_{AD}(L_n = L-1)$	$c_{AD}(L_n = L-2)$	$c_{AD}(L_n = L-3)$
3	21	142	380	1315	
4	85	722	1198	3068	10531
5	341	4954	5906	9646	24572

Table 4.9: Computational complexity of the AD weight estimator for $M_{\theta_n} = 24$ (point separation of 15°).

L	K	$c_{AD}(L_n = L)$	$c_{AD}(L_n = L-1)$	$c_{AD}(L_n = L-2)$	$c_{AD}(L_n = L-3)$
3	21	166	488	1753	
4	85	770	1414	3944	14041
5	341	5050	6338	11398	31592

4.7 Experimental Analysis

Performance prediction for the optimal joint weight/phase estimators (NC and PC estimators) described in Sections 4.1 and 4.4 and the suboptimal joint weight/phase estimator (AD) described in Section 4.6.2 is a complex task since the probability of making a weight error and the accuracy of the phase estimate are coupled. In lieu of deterministic analyses, this section reports the results of several simulations. These experimental results offer estimates for the probability of making a weight error[29] and for the probability distribution of the error on the phase estimate.

This section examines the performance of the one-shot optimal non-coherent (NC) and partially coherent (PC) tree weight/phase estimators from Sections 4.1 and 4.4. As we have learned from the simulations in Section 3.7.2, the ultimate goal at the receiver is to achieve a phase estimate that is in error of, at most, 5°. Although the one-shot phase estimation algorithms developed in this chapter are not capable of achieving such accuracy, they will serve as the engine for estimation procedures of Chapter 5 that make use of a sequence of symbol transmissions. It is important, then, to study the capabilities of the one-shot estimators several different degrees of prior phase uncertainty. The results of this section will be the impetus for some of the estimators developed in Chapter 5.

The non-coherent and partially coherent estimators can be thought of as being the same. Recall that the non-coherent case of having no prior knowledge of a user's phase can be represented with the partially coherent model by setting the accuracy parameter, α, to zero. Repeated below is the PDF used by the partially coherent

[29]Note that the detectors were derived to maximize the probability of making no weight errors but it is more meaningful to show simulation results in terms of bit error rate (BER) curves since BER curves are the standard method for performance evaluation in the communication literature. Each point in a BER curve is an estimate of the probability of making a weight error for a single user.

estimator to model the receiver's prior knowledge of a user's phase.

$$p_{\phi_K}(\phi) = \frac{e^{\alpha \cos \phi}}{2\pi I_o(\alpha)}, \tag{4.48}$$

The simulations are done with the 21 user minimum distance set of Ross and Taylor ([17]), repeated below.

$$\mathbf{S} = \begin{bmatrix} & 1/2 & 0 & 0 & 0 & 1/4 \\ & 1/2 & 0 & 0 & 0 & 1/4 \\ & 1/2 & 0 & 0 & 0 & 1/4 \\ & 1/2 & 0 & 0 & 0 & 1/4 \\ & 0 & 1/2 & 0 & 0 & 1/4 \\ & 0 & 1/2 & 0 & 0 & 1/4 \\ & 0 & 1/2 & 0 & 0 & 1/4 \\ \mathbf{I}_{16} & 0 & 1/2 & 0 & 0 & 1/4 \\ & 0 & 0 & 1/2 & 0 & 1/4 \\ & 0 & 0 & 1/2 & 0 & 1/4 \\ & 0 & 0 & 1/2 & 0 & 1/4 \\ & 0 & 0 & 1/2 & 0 & 1/4 \\ & 0 & 0 & 0 & 1/2 & 1/4 \\ & 0 & 0 & 0 & 1/2 & 1/4 \\ & 0 & 0 & 0 & 1/2 & 1/4 \\ & 0 & 0 & 0 & 1/2 & 1/4 \end{bmatrix} \tag{4.49}$$

Recall that I_{16} is the 16 dimensional identity matrix. The tree corresponding to this minimum distance set is shown in Figure 4-11.

Recall that in an actual system, each user is assigned a signature waveform envelope and a carrier frequency. The carrier phase is not controllable by the user or by the receiver. To model this phenomenon, the signature matrix above has been

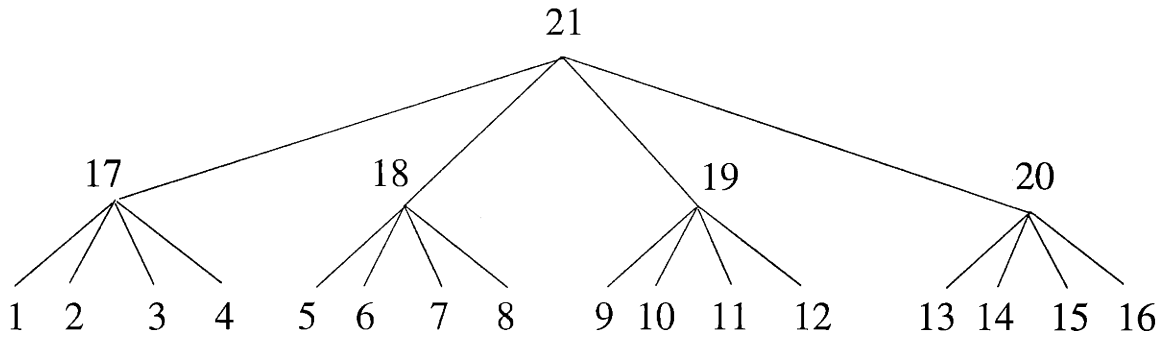


Figure 4-11: Correlation tree for a unit energy minimum distance signature set of 21 users in 16 dimensions.

altered by multiplying each column by an arbitrarily chosen unit magnitude complex scalar; the columns of this phase-altered matrix comprise the signature set used for all simulations in this section.

An MA system with an additive white Gaussian noise channel supporting 21 users with signature waveforms that correspond to the set just described is simulated. User 17 has an unknown or partially known phase at the receiver. The remaining 20 user signatures are completely and correctly known at the receiver.

The optimal detector is examined for the non-coherent case and for two different scenarios of the partially coherent case. For the non-coherent case, user 17 transmits a training weight, namely, $b_{17} = +1$. The partially coherent case has been simulated for two scenarios: 1) the receiver has knowledge of user 17's phase within $\pm 18.7°$, 2) the receiver has knowledge of user 17's phase within $\pm 8.4°$. For the $\pm 18.7°$ case, user 17 can, of course, send a training weight; alternatively, user 17 does not need to send a training weight.

We are interested in the examination of both the training and non-training cases since results for these cases will determine the phase uncertainty at which a user may reliably transmit an **information weight**. We use the term information weight since this weight bears information, i.e., the user chooses this weight from its set of M possible weights and the receiver does not know, a priori, which of the M weights is being transmitted. Hence, for the $\pm 18.7°$ case, both the non-training and training

cases are tried. For the $\pm 8.14°$ cases no training weight is used.

The joint weight/phase estimators, then, must account for this partial phase knowledge by adjusting the α parameter accordingly. See the estimator equations in (4.41) and (4.42). Recall that the detector has assumed the prior distribution of Equation (4.48), the exponential cosine, for the PDF of the uncertain phase. For the non-coherent case, $\alpha = 0$.

For the exponential cosine PDF having an accuracy parameter, α, and a standard deviation, $\pm\sigma_\alpha$, the phase will fall within $\pm\sigma_\alpha$ with 68.5% of the probability. Moreover, the phase will fall within $\pm 2\sigma_\alpha$ ($\pm 3\sigma_\alpha$) with 95% (99.7%) of the probability. The standard deviation corresponding to $\alpha = 10$ ($\alpha = 50$) is $\sigma_{10} = 18.7°$ ($\sigma_{10} = 8.14°$).[30]

The PC detector requires the choice of a value for α. If the receiver has knowledge of a user's phase to within $\pm x°$, the value for α chosen for the simulations in this thesis corresponds to $\sigma_\alpha = x$.

The three *uniform* prior distributions used for the creation of the *actual phase* of user 17 for the simulations are shown in Figure 4-12. The first is the uniform distribution of Figure 4-12-(a) for which user 17's phase is equally likely to take on any value from $-180°$ to $+180°$. The second and third are the uniform distributions shown in Figures 4-12-(b) and (c) for which user 17's phase is equally likely to take on any value between $-18.7°$ and $+18.7°$ or between $-8.14°$ and $+8.14°$, respectively.

For the simulations done with the $\pm 180°$ prior, the NC joint weight/phase estimator of Section 4.1 was used (this corresponds to $\alpha_{180°} = 0$). For the simulations done with the $\pm 18.7°$ ($\pm 180°$) uniform prior, the PC joint weight/phase estimator, having a value of $\alpha_{18.7°} = 10$ ($\alpha_{8.14°} = 50$), was used.

[30] Viterbi, in [24], gives formulas for finding σ_α and Prob($|\phi| < x$). These formulas contain an infinite series. An extremely good approximation has been used for the calculations of σ_α and Prob($|\phi| < x$) in this thesis by carrying out the first 40 terms of the series.

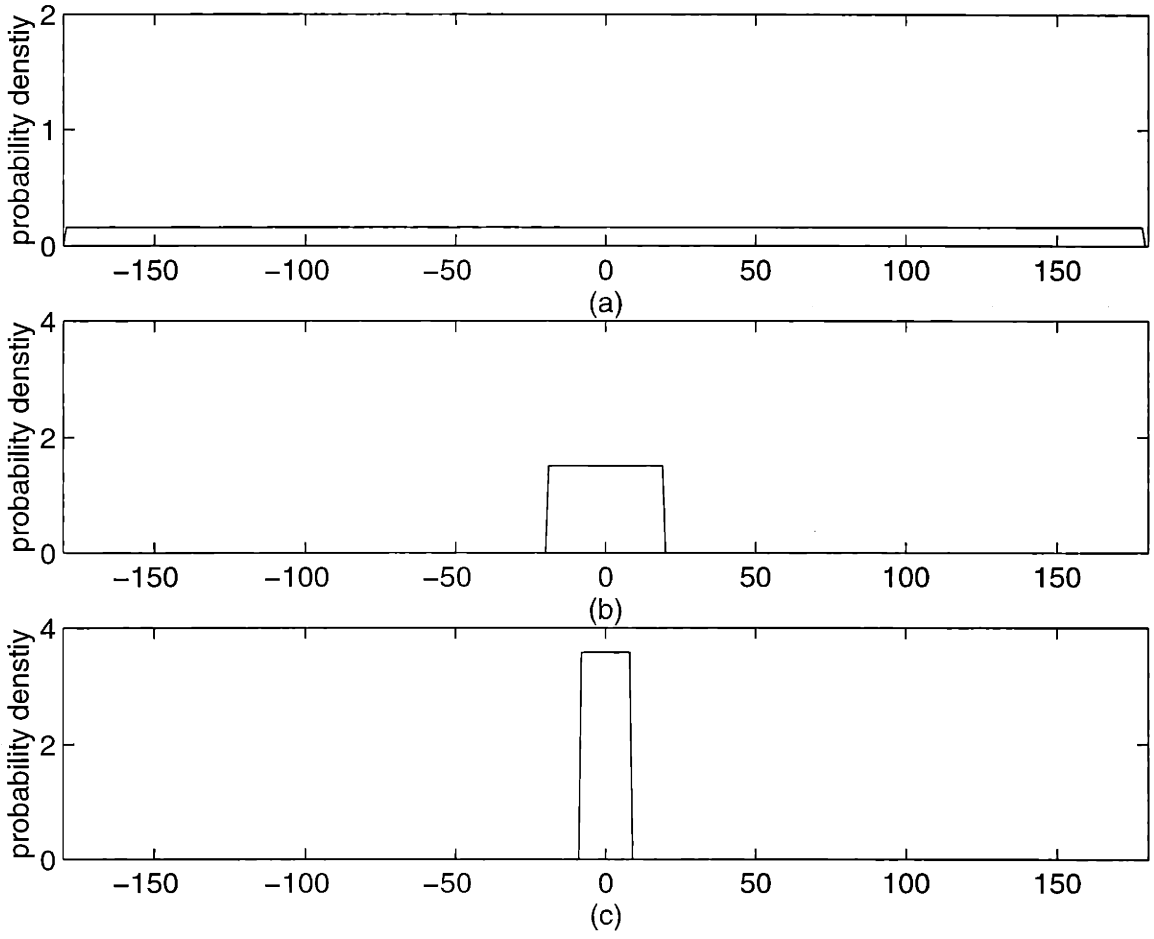


Figure 4-12: (a) Uniform PDF: range = $[-180°, +180°]$. (b) Uniform PDF: range = $[-18.7°, +18.7°]$. (c) Uniform PDF: range = $[-8.4°, +8.4°]$.

4.7.1 Weight Error Analysis

This section shows two sets of bit error rate (BER) curves. The first set of curves is the result of having user 17 send a known training weight. The second set is the result of allowing user 17 to send an information weight, i.e., no training on user 17.

Uncertain-Phase-User Sends a Training Weight

All simulations reported in this part have been run with a training weight for user 17. Figure 4-13 shows three average bit error rate curves. The solid curve is shown as a reference and is the BER for the optimal coherent weight detector, i.e. all users' phases are correctly known at the receiver. The dashed curve is for the non-coherent weight/phase estimator[31] where the actual phase is drawn from the uniform distribution in Figure 4-12-(a) ($\pm 180°$). The dash-dot curve is for the PC estimator for which $\alpha = 10$ where the actual phase is drawn from the uniform distribution in Figure 4-12-(b) ($\pm 18.7°$). Error bars on the BER curves are not shown; the standard deviation of each BER point in the figure is approximately (0.015)(BER).

From the figure we see a 0.25 dB difference between the solid and dashed curves. This tell us to expect an average performance loss of 0.25 dB (6%)[32] from the introduction of a new user having an unknown phase.[33] When the new user has a partially known phase (within $\pm 18.7°$), no performance loss is seen on average.

We are interested in examining the BER curves for several of the individual users in the tree. The users that experience loss from the presence of the unknown-phase user at node 17 are the users at nodes 1, 2, 3, and 4 (the children of node 17) and 21 (the parent of user 17). Figure 4-14 shows individual BER curves corresponding to

[31] Or, equivalently, the optimal PC estimator having α set to 0.

[32] Recall that a 0.25 dB (6%) loss in performance means that all users of the affected system would need to increase their signal to noise ratios (E_b/N_o) by 0.25 dB or 6% to regain the performance seen when all phases were known.

[33] Concrete conclusions are, of course, limited to the specific signal set and case tested in these simulations. These results, however, indicate general trends to be expected for tree-structured MA systems.

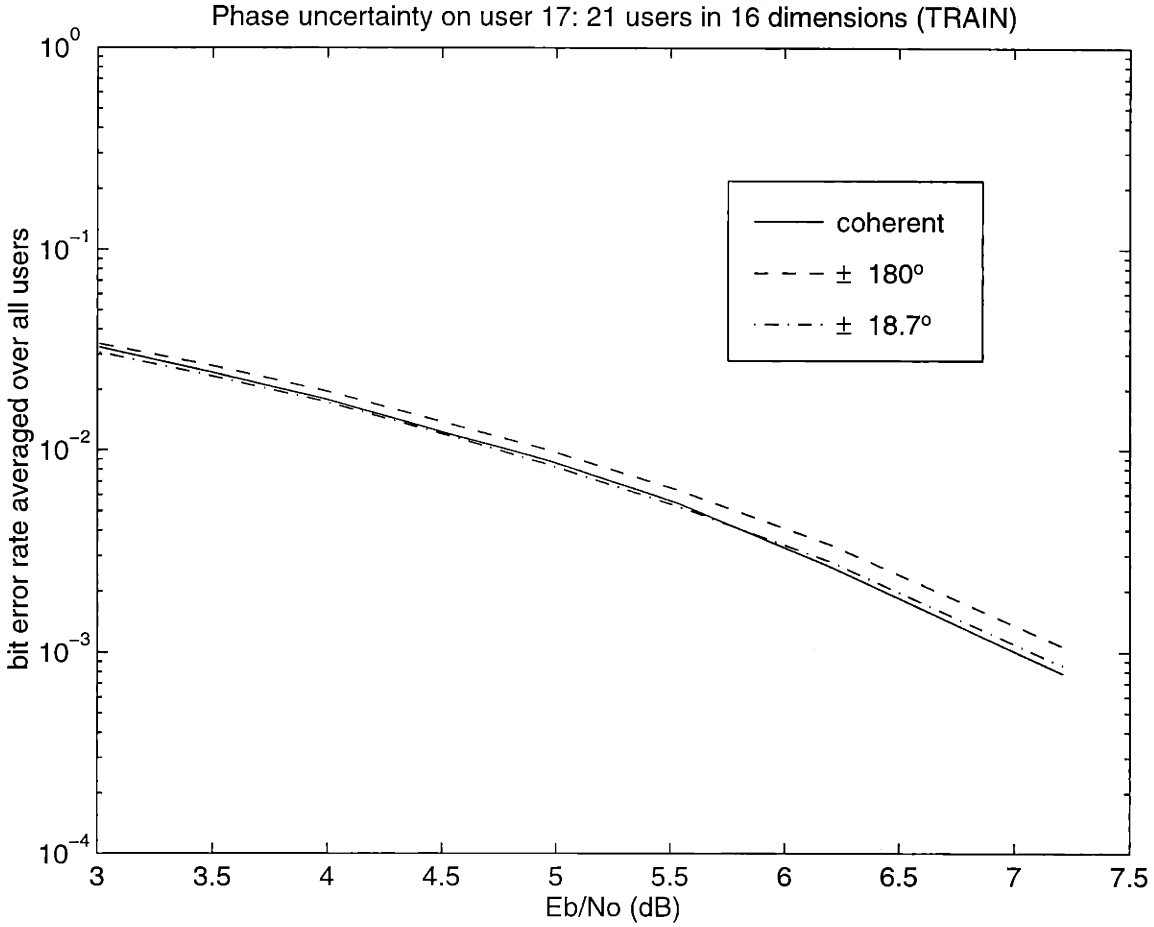


Figure 4-13: TRAINING WEIGHT USED FOR USER 17. Average bit error rate for a set of 21 users in 16 dimensions. User 17's phase is either known (solid line), unknown (dashed line), or known within $\pm 18.7°$ (dash-dot line). Error bars for the standard deviation of the BER points are not shown; they are are between 1.0% and 1.5% of the BER.

the same three curves shown in Figure 4-13 for each child user.

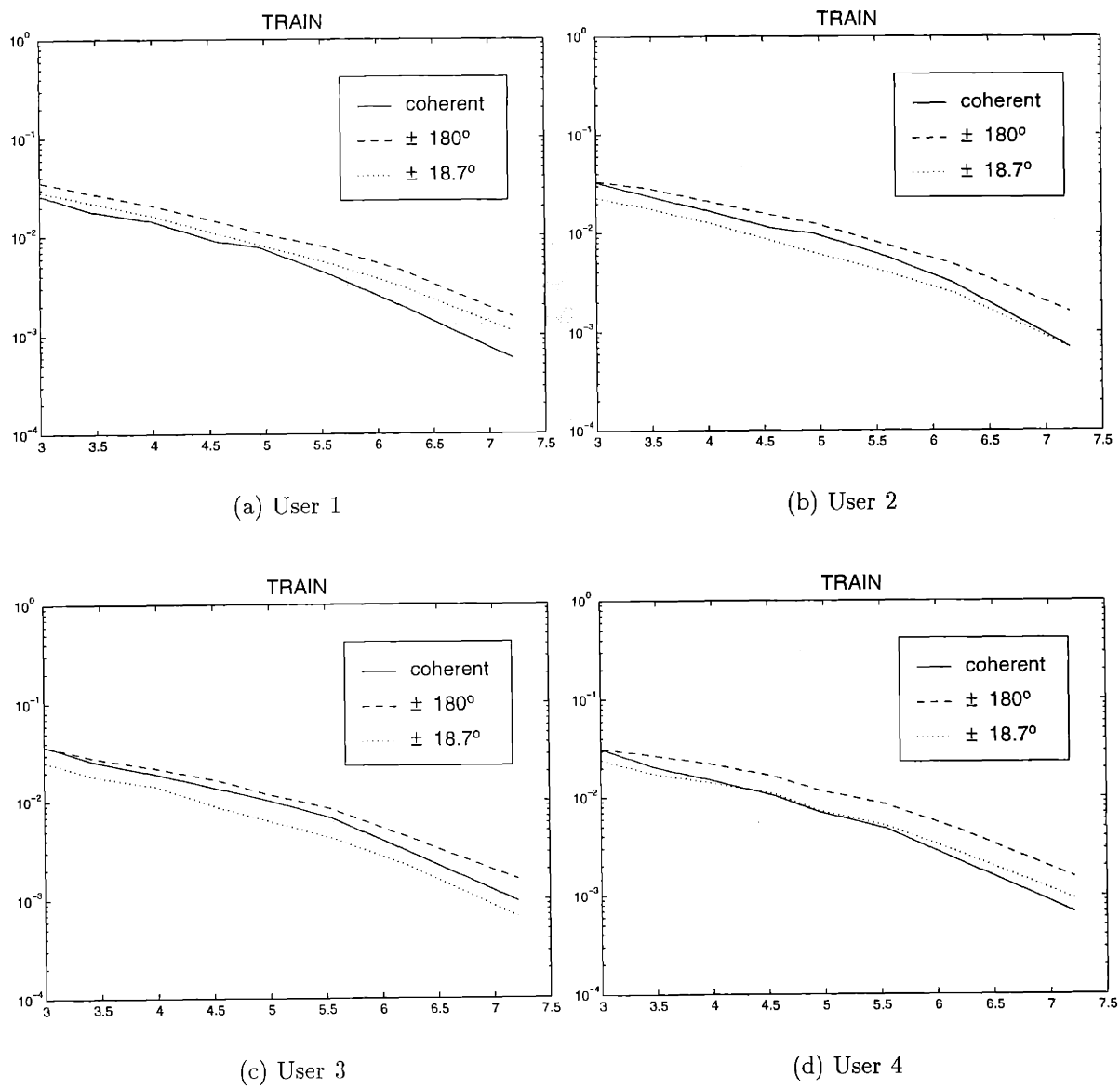
Notice that among the children users, there is at most at 0.75 dB (19%) loss in the BER curve for user 17 having an unknown phase ($\pm 180°$ uncertainty) relative to the BER curve for all users having known phases. For $\pm 18.7°$ uncertainty on user 17's phase, the children of user 17 experience a worst case loss of 0.35 dB (7%).

Figure 4-15 shows individual BER curves for user 21 (the parent of 17) and users 18, 19, and 20 (the siblings of user 17). Notice that the existence of the unknown-phase user at node 17 causes the user at node 21 (the parent of 17) to experience a performance loss of approximately 0.75 dB (19%) for the $\pm 180°$ uncertainty on user 17's phase and 0.5 dB (12%) for the $\pm 18.7°$ uncertainty. Also for the $\pm 180°$ uncertainty on user 17's phase, the sibling users, users 18 and 19 experience no loss, while the other sibling user, user 20, experiences a 0.15 (3.5%) dB loss.

It is interesting to compare the performance of the PC joint weight phase estimator for the $\pm 18.7°$ uncertainty on user 17's phase with the performance of the AC (assumed coherent) weight-only detector having a 20° mismatch on user 17's phase. Recall that in Section 3.7.2, the AC detector was run for the 21 user minimum distance set; the AC detector assumed a value for user 17's phase that was off by 20°. The worst case performance loss, at 1 dB (26%), due to this phase mismatch, is seen by user 21, while the average performance loss is 0.5 dB (12%).

In contrast, the simulations done for this section do not ignore the mismatch. Instead, for an expected mismatch in user 17's phase of up to $\pm 18.7°$, the receiver jointly estimates user 17's phase and the weights for the other 20 users in the system. In doing this, the worst case loss is 0.5 dB (12%) (for user 21), moreover, the average loss is, virtually, 0 dB (0%). This tells us that using a training weight on user 17 and jointly estimating its phase along with the weights of the other users will virtually eradicate performance loss for phase mismatches of up to 20°.

The results of this and the next section are summarized in Tables 4.10 and 4.11.



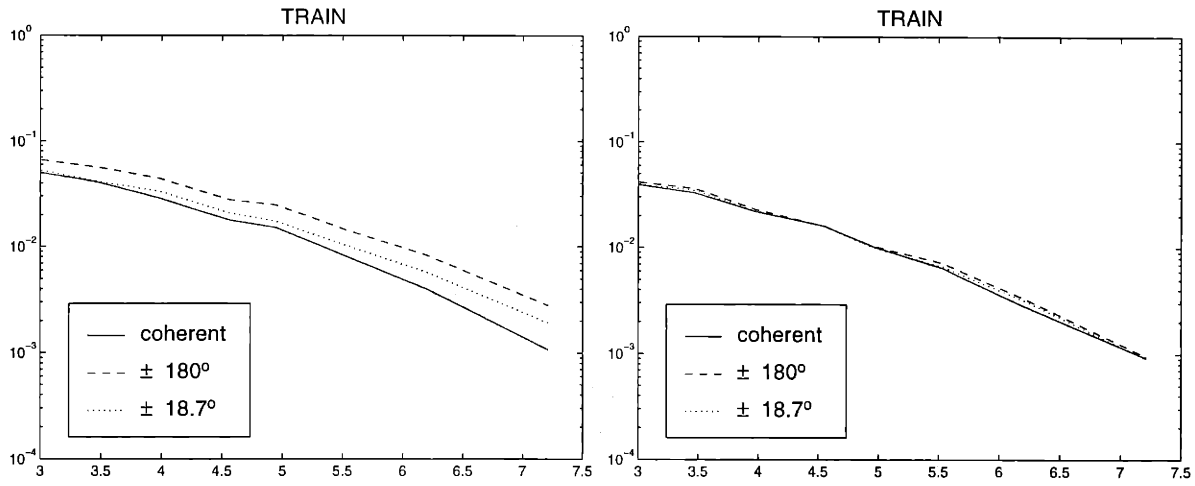
(a) User 1

(b) User 2

(c) User 3

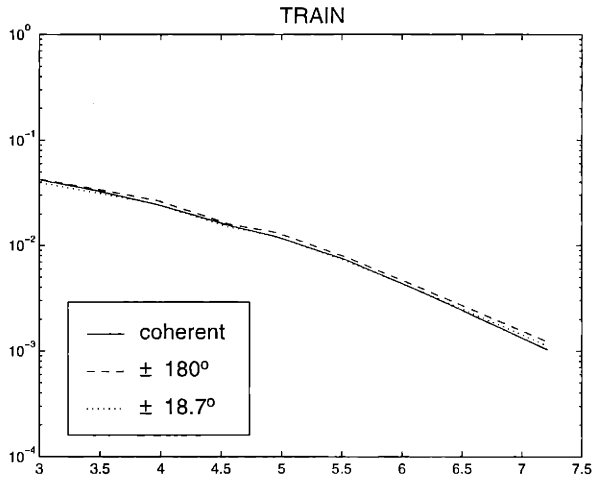
(d) User 4

Figure 4-14: TRAINING WEIGHT USED FOR USER 17. Bit error rates for individual users in a set of 21 users in 16 dimensions for which user 17's phase is uncertain. The solid curve corresponds to no phase uncertainties, the dashed curve corresponds to complete phase uncertainty ($\pm 180°$), and the dotted curve corresponds to partial phase uncertainty ($\pm 18.7°$). Error bars are not shown; the standard deviation of each BER point ranges from 4.1% to 7.7% of the BER. (a) user 1, a child of user 17. (b) user 2, a child of user 17. (c) user 3, a child of user 17. (d) user 4, a child of user 17. 17.

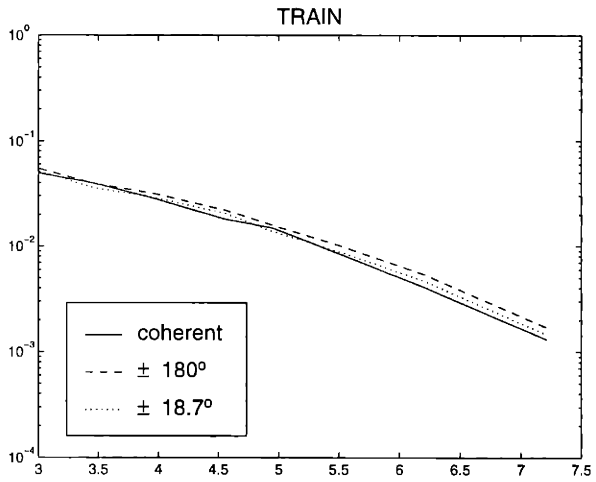


(a) User 21

(b) User 18



(c) User 19



(d) User 20

Figure 4-15: TRAINING WEIGHT USED FOR USER 17. Bit error rates for individual users for a set of 21 users in 16 dimensions for which user 17's phase is uncertain. The solid curve corresponds to no phase uncertainties, the dashed curve corresponds to complete phase uncertainty ($\pm 180°$), and the dotted curve corresponds to partial phase uncertainty ($\pm 18.7°$). The standard deviation of each BER point ranges from 3.0% to 6.7% of BER. (a) user 21, the parent of user 17. (b) user 18, a sibling of user 17. (c) user 19, a sibling of user 17. (d) user 20, a sibling of user 17.

Table 4.10: Loss in average BER when user 17's phase is unknown/uncertain relative to when it is known.

TRAINING		NO TRAINING	
phase uncertainty		phase uncertainty	
±180°	±18.7°	±18.7°	±8.14°
0.25 dB	0 dB	0.25 dB	0 dB

Table 4.11: Maximum loss for single user BER when user 17's phase is unknown/uncertain relative to when it is known.

TRAINING		NO TRAINING	
phase uncertainty		phase uncertainty	
±180°	±18.7°	±18.7°	±8.14°
0.75 dB	0.5 dB	0.5 dB	0 dB

Uncertain-Phase-User Does Not Send a Training Weight

In the following simulations user 17 does not send a training weight, i.e., it sends an information weight. Figure 4-16 compares the average performance loss between training and not training when the uncertainty is within ±18.7°. We see that the two curves are very close. This tells us that if the phase uncertainty is less than ±18.7°, user 17 does not need to send a training weight and that we might as well let user 17 transmit an information weight.

Figure 4-17 shows three average bit error rate curves. The solid curve, again, corresponds to the optimal joint detector for which all phases are known. The dashed curve is for the optimal PC estimator for which $\alpha = 10$ where the actual phase is drawn from the uniform distribution in Figure 4-12-(b) (±18.7°). The dash-dot curve is for the optimal PC estimator for which $\alpha = 50$ where the actual phase is drawn from the uniform distribution in Figure 4-12-(c) (±8.14°). Error bars on the BER curves are not shown; the standard deviation of each BER point in the figure is approximately 0.015 of the BER points.

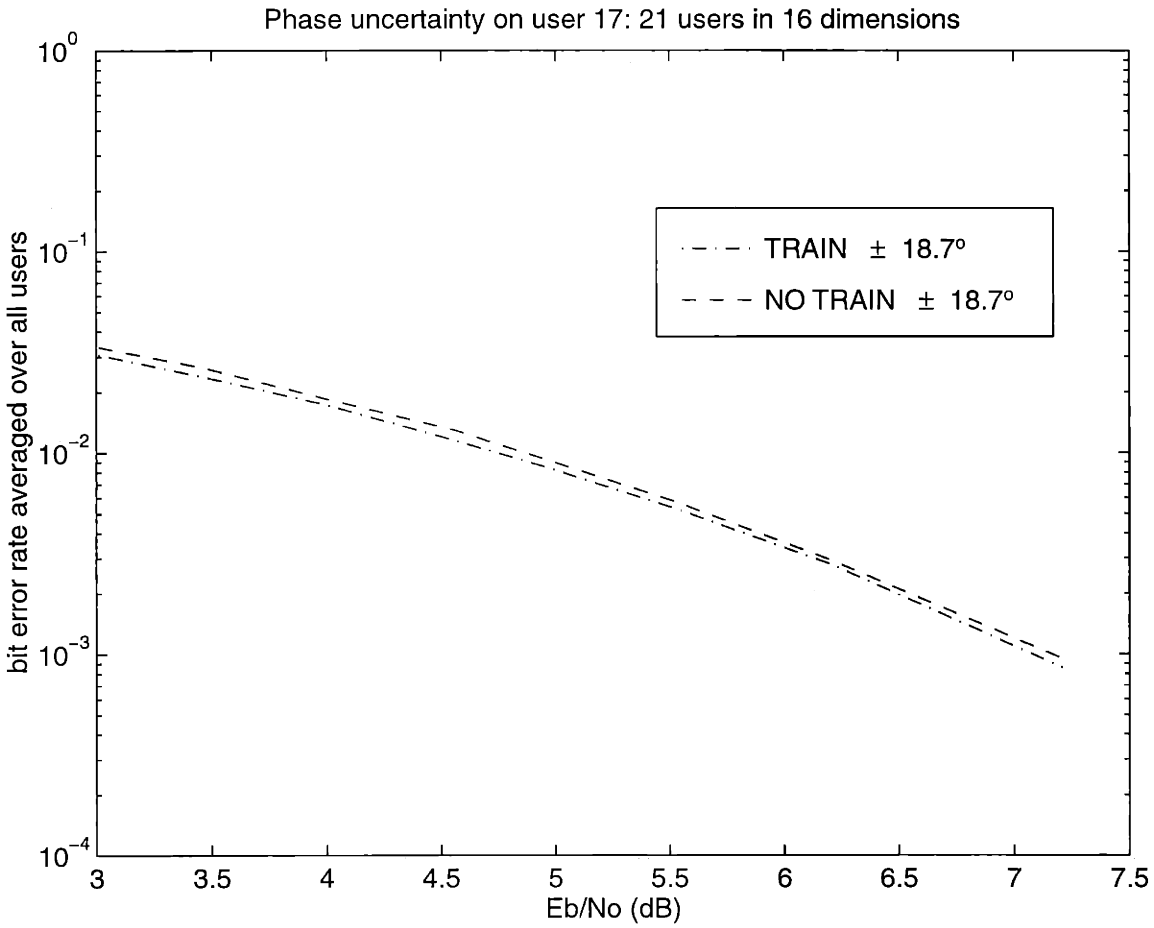


Figure 4-16: TRAINING VS NO TRAINING. Average bit error rate for 21 users in 16 dimensions. User 17's phase is known within ±18.7°. User 17 sends either an information weight (dashed line) or a training weight (dash-dot line). The standard deviations on the BER points are between 1.0% and 1.4% of the BER.

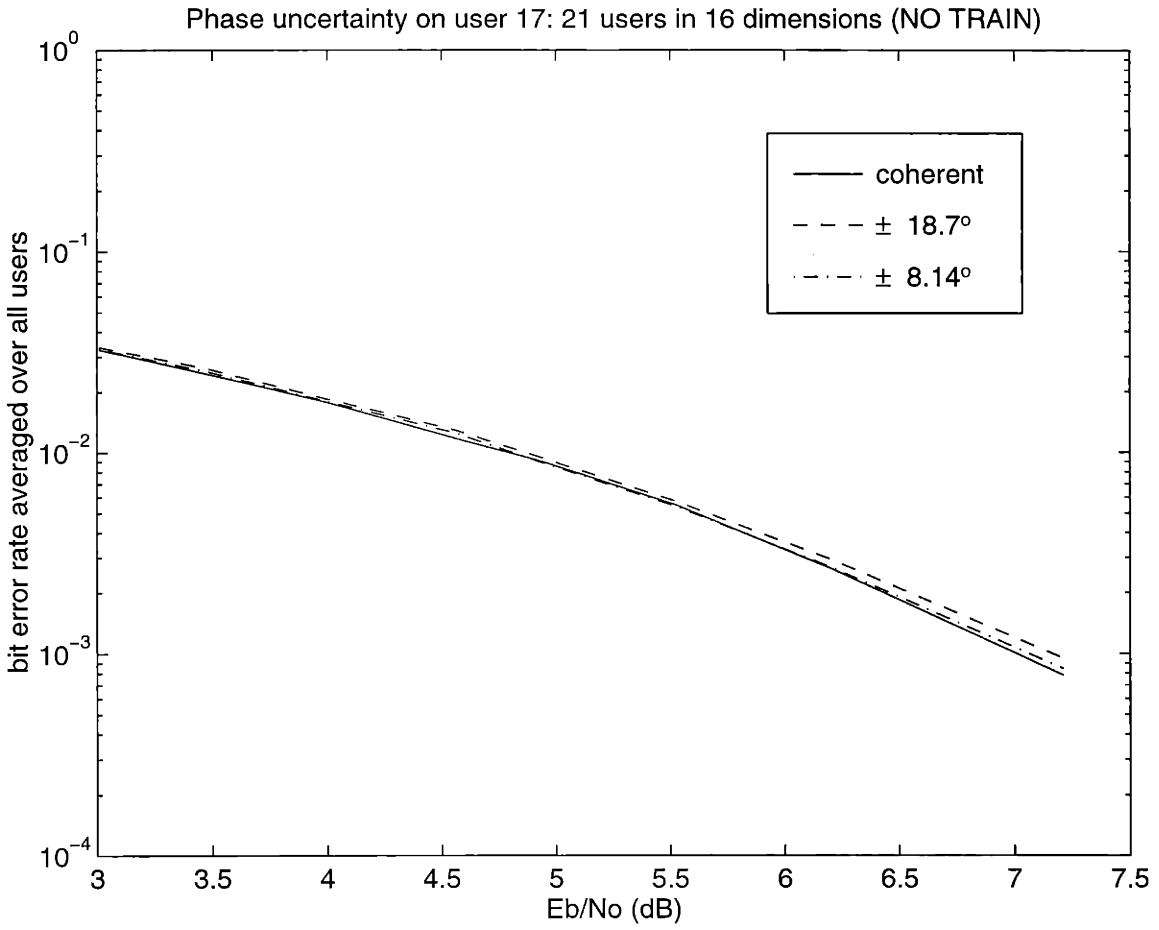


Figure 4-17: NO TRAINING. Average bit error rate for a set of 21 users in 16 dimensions. User 17's phase is either known (solid line), known within $\pm 18.7°$ (dashed line), or known within $\pm 8.14°$ (dash-dot line). The standard deviation on the BER points ranges from 1% and 1.5% of BER.

From Figure 4-17 we see less than 0.1 dB difference between the solid and dashed curves. This tell us to expect an average performance loss of only 0.1 dB (2.5%) when we go from all phases known to a $\pm 18.7°$ uncertainty on user 17's phase. Recall that all users send information weights, including user 17. When the new user has a small phase uncertainty of only $\pm 8.14°$, no performance loss is seen.

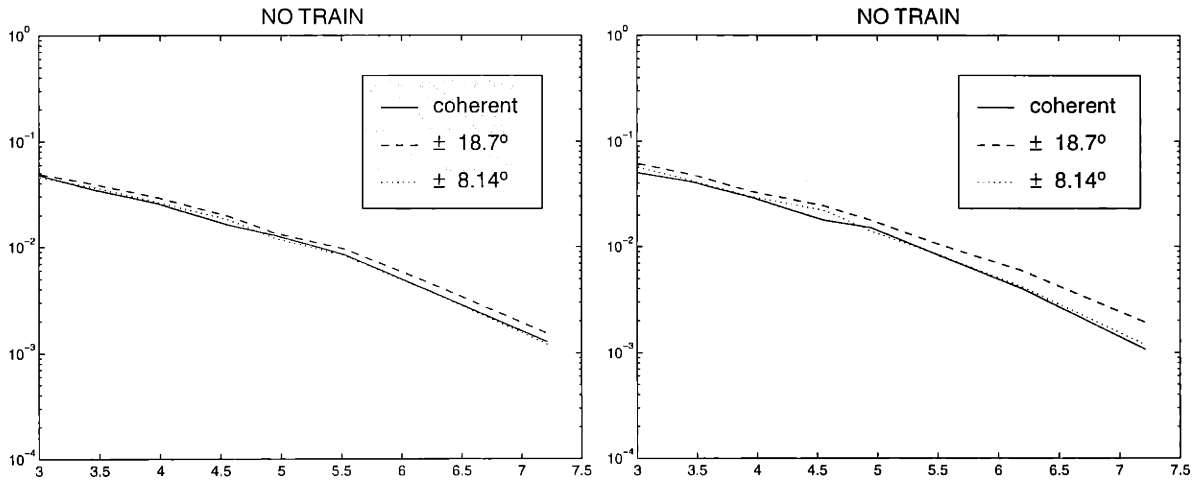
Also for this simulation, BER curves are shown for the individual users in the tree that experience loss due the presence of the unknown-phase user at node 17. From Figure 4-18-(a) we see that user 17 experiences virtually no loss when its phase is known within $\pm 18.7°$.[34] For the $\pm 8.14°$ uncertainty on user 17's phase, no users experience loss. Notice in Figure 4-18-(b) that a loss of 0.25 dB (6%) is experienced by user 21, the parent of user 17. From Figures 4-19-(a) and (d) we see, for the $\pm 18.7°$ uncertainty, a 0.5 dB (12%) loss for user 1 and a 0.2 dB (5%) loss for user 4. The other children of user 17 are untouched by the $\pm 18.7°$ uncertainty in user 17's phase.

Again, it is interesting to compare the performance between the AC detector from Section 3.7.2, that ignores a 20° mismatch, and the corresponding PC optimal joint weight/phase estimator. Recall that training is not used in either case. Comparison of Figure 3-11 in Section 3.7.2 with Figure 4-17 indicates that if the receiver ignores a phase uncertainty of 20°, it will do 0.25 dB (6%) *worse* than if it accounts for the phase uncertainty by using the optimal PC estimator with the appropriate value for α.

4.7.2 Phase Error Analysis

Section 3.7.2 in Chapter 3 found that a 5° phase error on all users in a 21 user unit energy minimum distance set caused only 0.13 dB (3%) degradation in the performance of the coherent detector. This section investigates the ability of the one-shot

[34]Note that these BER calculations have standard deviations ranging of 3.5% to 8.3% of the BER, hence, the apparent degradations in the figures might be partially due to inaccuracies in the BER curves.



(a) User 17 (b) User 21

Figure 4-18: NO TRAINING. Bit error rates for individual users for a set of 21 users in 16 dimensions for which user 17's phase is uncertain. The solid curve corresponds to no phase uncertainties, the dashed curve corresponds to a partial phase uncertainty of $\pm 18.7°$ and the dotted curve corresponds to a partial phase uncertainty of $\pm 8.14°$. Error bars are not shown; the standard deviation of the BER points range from 3.5% to 8.3% of BER. (a) user 17 (the user with the uncertain phase). (b) user 21, parent of user 17.

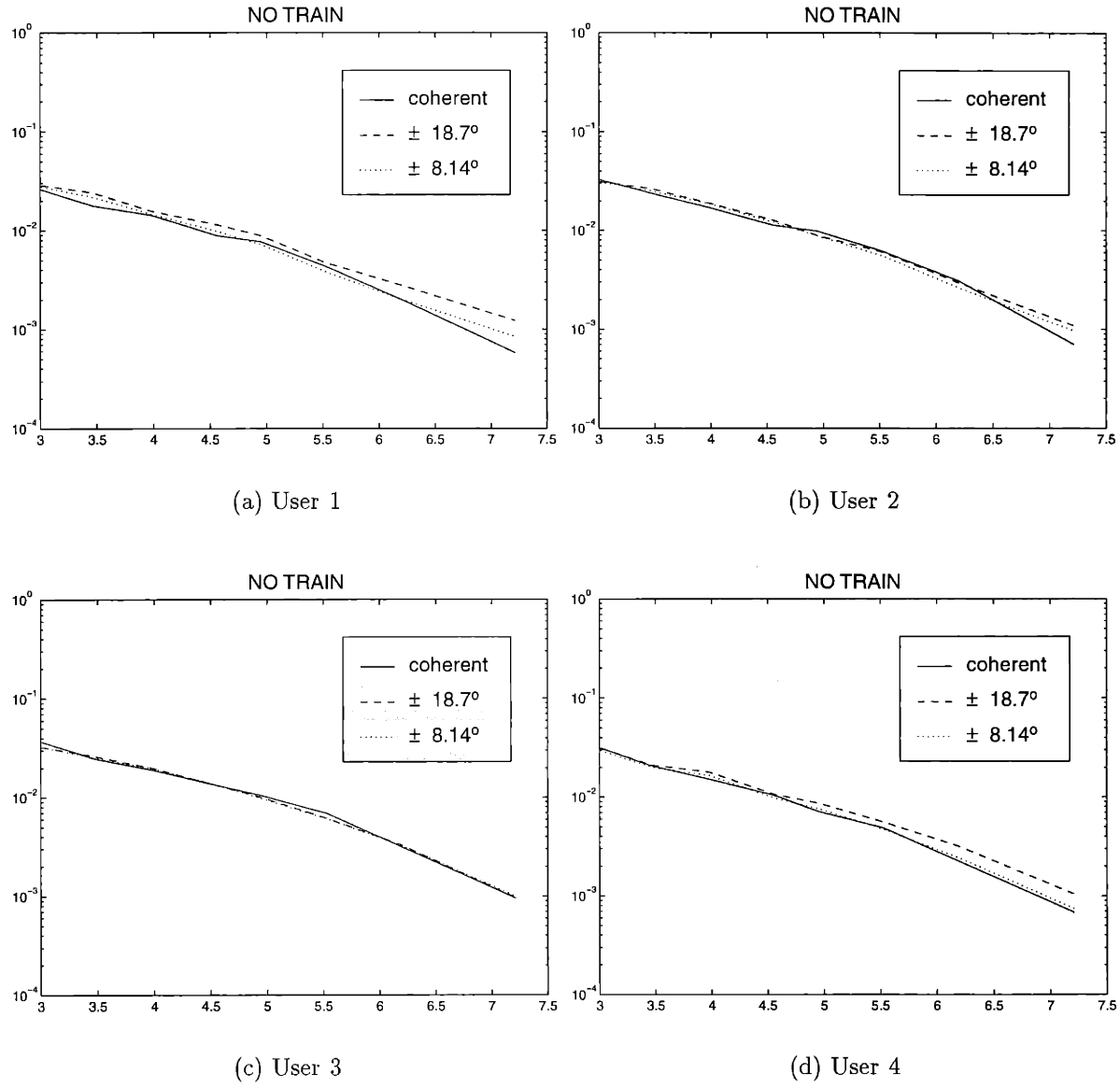


Figure 4-19: NO TRAINING. Bit error rates for individual users for a set of 21 users in 16 dimensions for which user 17's phase is uncertain. The solid curve corresponds to no phase uncertainties, the dashed curve corresponds to a partial phase uncertainty of $\pm 18.7°$ and the dotted curve corresponds to a partial phase uncertainty of $\pm 8.14°$. Error bars are not shown; the standard deviation of the BER points range from 3.5% to 8.3% of BER. (a) user 1, a child of user 17. (b) user 2, a child of user 17. (c) user 3, a child of user 17. (d) user 4, a child of user 17.

optimal PC joint weight/phase estimator to return a phase estimate that will meet this need of the coherent tree joint detector. Phase information from the simulations of the previous section are presented. Specifically, the probability distribution function for the one-shot phase estimate error is approximated by histograms made from simulation results where an exponential cosine PDF is fit to each histogram.

Recall that for the the exponential cosine PDF having an accuracy parameter, α, and a standard deviation, σ_α, the phase will fall within $\pm\sigma_\alpha$ with 68.5% of the probability. The phase will fall within $\pm 2\sigma_\alpha$ ($\pm 3\sigma_\alpha$) with 95% (99.7%) of the probability.

Recall from the previous section that a 21 user unit energy minimum distance set is used in all simulations. User 17's phase was either unknown or partially known at the receiver. Two scenarios were simulated: 1) user 17 sends a training weight; 2) user 17 does not send a training weight, i.e., it sends an information weight.

Uncertain-Phase-User Sends a Training Weight

Recall that two simulations were run in which user 17 sends a known training weight. One simulation was run with an uncertainty for user 17's phase of $\pm 180°$ and the other simulation was run with a phase uncertainty of $\pm 18.7°$. Normalized histograms were created with the phase error results from these simulations.

For the $\pm 180°$ uncertainty, Figure 4-20 shows, for several pertinent values of the signal to noise ratio,[35] these normalized histograms for the error in the phase estimates given by the optimal non-coherent joint weight/phase estimator. It is interesting to note that the histograms in Figure 4-20 closely resemble the exponential cosine function which is the PDF for the error in the phase locked loop phase estimate for the *single* user system ([24]). The exponential cosine function is given in Equation (4.48).[36] The histograms in Figure 4-20 exhibit, in addition to a main lobe

[35]These four values of the SNR are most pertinent since at these values the BER is between 10^{-2} and 10^{-3}. Communications systems typically require one-shot detectors to give bit error rates of between 10^{-2} and 10^{-3} so that the overall bit error rate of the coded system will be less than 10^{-5} ([2]).

[36]The exponential cosine PDF (for values of α greater than 5) resembles the Gaussian PDF with

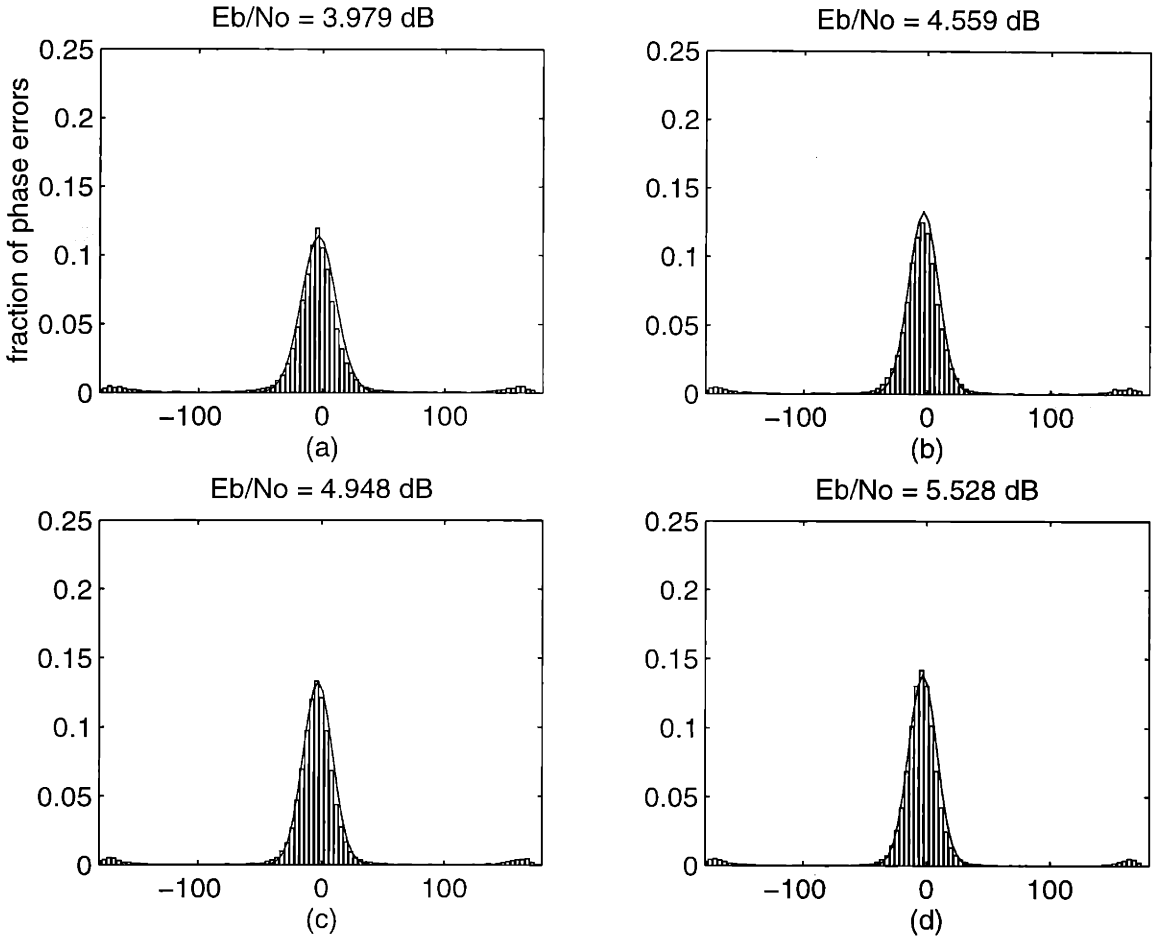


Figure 4-20: TRAINING WEIGHT USED. UNIFORM PRIOR (±180°). This figure shows normalized histograms of the phase error for user 17's phase estimate given by the optimal non-coherent joint weight/phase detector. User 17 sends a known weight. The solid curve is the exponential cosine probability mass function having a value for α that gives the best fit to the histograms. (a) E_b/N_o = 3.979 dB, $\alpha_{\hat{\phi}}$=17, sample stadev($\hat{\phi}$) = 43.8°, effective stadev($\hat{\phi}$) = 17.3°. (b) E_b/N_o = 4.559 dB, $\alpha_{\hat{\phi}}$=23, sample stadev($\hat{\phi}$) = 41.9°, effective stadev($\hat{\phi}$) = 15.6°. (c) E_b/N_o = 4.948 dB, $\alpha_{\hat{\phi}}$=23, sample stadev($\hat{\phi}$) = 41.1°, effective stadev($\hat{\phi}$) = 14.9°. (d) E_b/N_o = 5.528 dB, $\alpha_{\hat{\phi}}$=25, sample stadev($\hat{\phi}$) = 39.7°, effective stadev($\hat{\phi}$) = 13.5°.

centered at zero, two small lobes centered at $\pm 175°$, respectively. The outlier lobes, together, contain from 4% to 8.2% of the probability, depending on SNR. The outlier probability will be denoted by p. For each SNR, p was estimated as the fraction of the phase errors greater than $\pm 160°$ made by the optimal joint weight/phase estimator run for the simulations.

The existence of other lobes for the PDF of the phase error of the optimal one-shot joint weight/phase estimator is to be expected since, if the detector makes weight errors, it has, in effect, chosen the wrong "ring". This will, of course, give a phase estimate that is, possibly, quite far from the actual phase.[37] The placement of these additional small lobes is dependent on the configuration of hypothesized rings that correspond to the received signal constellation. Hence, outlier probability will not, for general tree-structured signature sets, be centered around $\pm 175°$.

The one-shot estimate error ϕ_e, then, is well described by

$$p_{\phi_e|\mathbf{r}}(\phi_e|\mathbf{r}) = (1-p)\frac{e^{\alpha\cos(\phi_e)}}{2\pi I_o(\alpha)} + pu(\phi), \tag{4.50}$$

where $u(\phi)$ is the uniform distribution with a range $[-180°, 180°]$. Of course, this is not an exact model, but it is sufficient to illustrate the behavior of the one-shot phase estimate that we wish to acknowledge.

The sample variance was calculated for the phase estimate errors found by the simulation. The square root of this sample variance, or sample standard deviation, for the SNR's of interest was was found to be between 34° and 48°, depending on the signal to noise ratio. Given that the histograms are multi-modal with one dominant mode, a more informative way of measuring the quality of the one-shot phase estimate given by the non-coherent joint weight/phase estimator is to assign an *effective standard deviation* which quantifies the main lobe (the portion of the histograms

tails that go to zero more quickly.

[37]Recall the ring detection interpretation of the optimal joint weight/phase estimator given in Section 4.1.

Table 4.12: TRAINING: UNIFORM PRIOR ($\pm 180°$). User 17 sends a known training weight. Prior distribution on user 17's actual phase is uniform between $\pm 180°$. For each value of the signal to noise ratio, the following quantities have been calculated: sample standard deviation of the phase errors in the estimate (stadev($\hat{\phi}$)), the outlier probability (p), the effective sample standard deviation (effective stadev($\hat{\phi}$)), and the value of α ($\alpha_{\hat{\phi}}$) that gives the best fit of the exponential cosine PDF to the main lobe of the histogram of phase errors.

E_b/N_o dB	stadev($\hat{\phi}$)	p	effective stadev($\hat{\phi}$)	$\alpha_{\hat{\phi}}$
3.014	47.8°	.082	19.8°	15
3.468	46.0°	.075	18.8°	15
3.979	43.8°	.068	17.3°	17
4.559	41.9°	.061	15.6°	23
4.948	41.1°	.058	14.9°	23
5.528	39.7°	.053	13.5°	25
6.200	37.2°	.046	12.0°	25
7.210	34.7°	.040	10.3°	30

that contain $(1 - p)$ or 0.918 to 0.960 of the probability, depending on the SNR). The one-shot phase estimate, then, was found to have an effective standard deviation ranging from 10° to 20° where an occasional outlier error will occur with a 0.04 to 0.08 probability, depending on the SNR.

Furthermore, an exponential cosine probability mass function was empirically fit to each histogram in the figure. The value for α that gave a good fit to the histograms ranged from 17 to 20. See Table 4.12 for the specific values of the standard deviation, denoted by stadev(ϕ), the outlier probability, denoted by p, the effective standard deviation, denoted by effective stadev(ϕ), and the value for α, denoted by $\alpha_{\hat{\phi}}$, that gives the best fit of the exponential cosine PDF to each histogram.

We know that for an exponential cosine PDF the phase will fall within $\pm 2\sigma_\alpha$ with 95% of the probability. Although the histograms in Figure 4-20 are not an exact fit with the exponential cosine PDF we may conclude that, if the outlier probability is denoted by p, the error in the one-shot optimal phase estimate will be within twice the effective standard deviation (as given in Table 4.12) with $(1 - p)$ probability.

For example, at an SNR of 5.528 dB, the estimate of user 17's phase given by the optimal one-shot non-coherent joint weight/phase estimator will, with approximately 0.947 probability, fall within $\pm 27°$, while, with 0.53 probability, the estimate will be in error of $\pm 175°$. This probability analysis of the histograms is quantified further in Chapter 5.

Recall from the results of Section 3.7.2 that the coherent joint detector requires an accuracy in all users phases of better than 5°. As we would expect, the one-shot non-coherent joint weight/phase estimator does not supply an estimate of this quality. Hence, we might think to allow user 17 to transmit two or more known training weights, collect the two or more independently obtained optimal phase estimates, and average them. Moreover, given that we anticipate, with small probability, an outlier estimate, we may first discard any estimates believed to be outliers before calculating the average. In Chapter 5, this notion of combining two or more successive non-coherent phase estimates is addressed in detail.

Simulation results were also obtained for the partially coherent case for which user 17 sent a known training weight. Figure 4-21 shows the histograms for this case. Notice that the histograms have only one lobe. As was determined from examination of the BER curves in Figure 4-13, having partial information on user 17's phase within $\pm 18.7°$ resulted in no performance loss. This BER result coincides with an absence of the outlier probability in the phase estimate. Phase estimate results are summarized in Table 4.13. Since there are no outliers the effective sample standard deviation is the same as the actual standard deviation and the outlier probability, p, is 0.

From the simulations with an $\pm 18.7°$ uncertainty on user 17's phase, we have found that the optimal partially coherent joint weight/phase estimator gives a phase estimate that is within $2(\text{stadev}(\hat{\phi}))= 18°$ of the true phase, with 95% probability. Recall that in Section 3.7.2 it was determined that any estimate of user's phase must be within a 5° error, hence, even with this partial information on user 17's phase ($\pm 18.7°$ uncertainty), the one-shot optimal joint weight/phase estimator (with

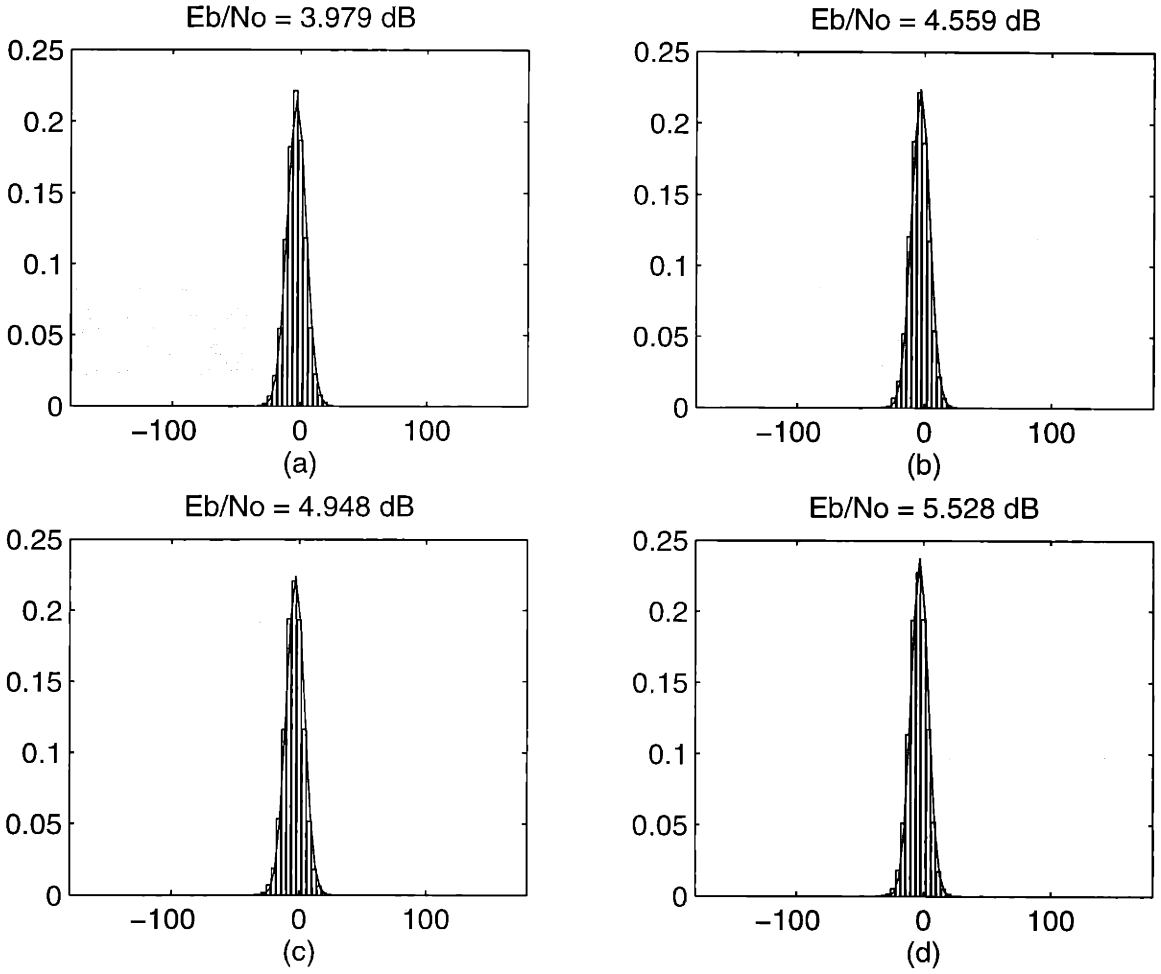


Figure 4-21: TRAINING WEIGHT USED. UNIFORM PRIOR (±18.7°). This figure shows normalized histograms of the phase error for user 17's phase estimate given by the optimal partially coherent joint weight/phase detector. User 17 sends a known weight. The solid curve is the exponential cosine probability mass function having a value for alpha that gives the best fit to the histograms. (a) $E_b/N_o = 3.979$ dB, $\alpha_{\hat{\phi}}$= 60, sample stadev($\hat{\phi}$) = 8.6°. (b) $E_b/N_o = 4.559$ dB, $\alpha_{\hat{\phi}}$= 65, sample stadev($\hat{\phi}$) = 8.14°. (c) $E_b/N_o = 4.948$ dB, $\alpha_{\hat{\phi}}$= 65, sample stadev($\hat{\phi}$) = 8.14°. (d) $E_b/N_o = 5.528$ dB, $\alpha_{\hat{\phi}}$= 70, sample stadev($\hat{\phi}$) = 8.2°.

Table 4.13: TRAINING: UNIFORM PRIOR ($\pm 18.7°$). User 17 sends a known training weight. The prior distribution on user 17's actual phase is uniform between $\pm 18.7°$. For each value of the signal to noise ratio, the sample standard deviation of the error in the phase estimate (stadev($\hat{\phi}$)) has been calculated and value of α ($\alpha_{\hat{\phi}}$) that gives the best fit of the exponential cosine PDF to the histogram of phase errors has been approximated.

E_b/N_o dB	stadev($\hat{\phi}$)	$\alpha_{\hat{\phi}}$
3.014	8.6°	60
3.468	8.7°	60
3.979	8.6°	60
4.559	8.4°	65
4.948	8.4°	65
5.528	8.2°	70
6.200	8.0°	70
7.210	7.7°	75

training) will not achieve a phase estimate of high enough quality to be used as if it were true for the detection of subsequent weights.

From the phase results presented up to this point, we might think to construct a multi-symbol recursive phase estimator. For example, the user initially joins the system and its phase is completely unknown to the detector. One training weight is transmitted; the non-coherent optimal joint detector gives a phase estimate ($\hat{\phi}(1)$) having an effective standard deviation of around 14°. In addition, this phase estimate, $\hat{\phi}(1)$, is incorporated into the uncertain phase user's signature vector, as known at the receiver. Another training weight is sent. This time, the partially coherent joint detector is used (having an appropriate value of α to reflect the partial information obtained by the previous estimate of the phase). Now, we have a *refined* estimate of the phase, $\hat{\phi}(2)$, with stadev($\hat{\phi}(2)$) $< 14°$. Again, this phase estimate, $\hat{\phi}(2)$, is incorporated into the uncertain phase user's signature vector, as known at the receiver. The process may be repeated until the refined estimate is of the appropriate quality. Chapter 5 formally presents this recursive phase refinement procedure along with some

Table 4.14: NO TRAINING : UNIFORM ±18.7°. User 17 sends and information weight. Prior distribution on user 17's actual phase is uniform between ±18.7°. For each value of the signal to noise ratio, the sample standard deviation of the phase error (stadev($\hat{\phi}$)) has been calculated and the value of α ($\alpha_{\hat{\phi}}$) that gives the best fit of the exponential cosine PDF to the histogram of phase errors has been approximated.

E_b/N_o dB	stadev($\hat{\phi}$)	$\alpha_{\hat{\phi}}$
3.014	8.8°	55
3.468	8.8°	60
3.979	8.6°	60
4.559	8.4°	60
4.948	8.4°	65
5.528	8.3°	70
6.200	8.0°	70
7.210	7.7°	75

other phase estimation algorithms that take advantage of having several symbol's worth of information.

Uncertain-Phase-User Does Not Send a Training Weight

From the simulations of Section 3.7.2 it was shown to be reasonable (for the 21 user unit energy minimum set) to allow a user with an uncertain phase to send an information weight as long as the phase uncertainty is within ±20°. For the results reported in the following paragraphs, user 17 does not use training, i.e., user 17 sends an information weight. The uniform priors on user 17's actual phase are shown in Figures 4-12-(b) and (c), where the ranges of uncertainty on user 17's phase are ±18.7° and ±8.4°.

Tables 4.14 and 4.15 summarize the simulations results for both the ±18.7° and the ±8.4° uncertainty cases. From comparison of Table 4.13 for which user 17 sends a *known training weight* with Table 4.14 for which user 17 *does not send a known training weight*, we see that the phase estimates in both cases are of the same quality. Specifically, allowing user 17 to send an information weight does not jeopardize the

Table 4.15: NO TRAINING : UNIFORM $\pm 8.4°$. User 17 sends a known weight. Prior distribution on user 17's actual phase is uniform between $\pm 8.4°$. For each value of the signal to noise ratio, the sample standard deviation of the phase error (stadev($\hat{\phi}$)) has been calculated and the value of α ($\alpha_{\hat{\phi}}$) that gives the best fit of the exponential cosine PDF to the histogram of phase errors has been approximated.

E_b/N_o dB	stadev($\hat{\phi}$)	$\alpha_{\hat{\phi}}$
3.014	5.1°	300
3.468	5.1°	300
3.979	5.1°	300
4.559	5.1°	300
4.948	5.1°	300
5.528	5.2°	300
6.200	5.2°	300
7.210	5.2°	300

estimate of its phase. The phase estimate, in either case, will be in error of less than 18° with 95% probability.

Since we ultimately wish to achieve an estimate having $2\sigma_{\hat{\phi}} \leq 5°$, results are also shown for the case in which the uncertainty is only slightly greater than what is needed by the coherent joint detector. The prior uncertainty for this simulation was $\pm 8.4°$. The simulation results for this case, shown in Table 4.15, indicate that only a small improvement in a single user's phase uncertainty is obtained by the one-shot the optimal partially coherent weight/phase estimator. Specifically, we see that the standard deviation on the phase estimate is approximately 5°. Recall that this means that with 95% probability, the phase estimate is within twice the standard deviation, or within $\pm 10°$.

It is clear from these estimates that the one-shot optimal NC joint weight/phase estimator (for prior uncertainty of $\pm 180°$) is not capable of giving a estimate with the needed standard deviation of 2.5°. If partial information is obtained, the PC joint weight/phase joint estimator is capable of giving a much improved phase estimate, where the variance of this phase estimate depends, of course, on the range of the prior

uncertainty. Chapter 5 discusses phase estimation from a *sequence* of symbols in a tree-structured MA system, where the results of this section are used to motivate two procedures of successive phase estimation over several symbol transmissions.

Chapter 5

Sequence Weight/Phase Estimation

The accuracy of the phase estimate can be improved by using observations of the received signal for more than one symbol frame. The user having the unknown phase would transmit a known training *sequence* of T weights. The optimal training sequence estimator would collect the received signal over T successive symbol transmissions, after which it would jointly estimate the phase of the unknown-phase user and all of the $(K-1)T$ weights for the rest of the users.[1] In this chapter the optimal **joint weight/phase sequence estimator** for tree structured sets with one user having an unknown phase is derived and its complexity is calculated. Since for moderate T the complexity of the optimal sequence estimator turns out to be prohibitive for many systems, two alternative multi-frame procedures are proposed, one calculates a **multi-frame phase estimate average** and the other calculates a **recursive phase estimate**. Both make use of T successive frames to give a high quality phase estimate with negligible increase in complexity relative to the one-shot

[1] For the derivation of the optimal sequence estimator, coding that might be employed by each user is not taken into account. I.e., it is assumed that all possible combinations of weights in a T-tuple of user weights is equally likely.

joint weight/phase estimator.

5.1 Optimal Weight/Phase Sequence Estimation

5.1.1 Derivation of the Estimator

We allow T frames to be collected. Let us re-write the model for the received signal to reflect the frame number, i.

$$\mathbf{r}(i) = \mathbf{S}\Phi\mathbf{b}(i) + b_K(i)\mathbf{s}_K e^{j(\phi_K+\phi)} + \sigma\mathbf{n}(i), \quad i = 1, \cdots, T$$

The new user, user K, is required to transmit a training sequence that is known both to the transmitter and receiver. Without loss of generality, it is assumed that $\mathbf{b}_K(i) = 1$ for $i = 1, 2, \cdots, T$.

The received vectors and the noise vectors may be stacked into two $TN \times 1$ column vectors.

$$\mathbf{r}_T = \begin{bmatrix} \mathbf{r}(1) \\ \mathbf{r}(2) \\ \vdots \\ \mathbf{r}(T) \end{bmatrix}, \quad \mathbf{n}_T = \begin{bmatrix} \mathbf{n}(1) \\ \mathbf{n}(2) \\ \vdots \\ \mathbf{n}(T) \end{bmatrix}$$

Likewise, the weight vectors may be stacked into a $T(K-1) \times 1$ column vector.

$$\mathbf{b}_T = \begin{bmatrix} \mathbf{b}(1) \\ \mathbf{b}(2) \\ \vdots \\ \mathbf{b}(T) \end{bmatrix}$$

The received signal model, then, may be written as

$$\mathbf{r}_T = \mathbf{S}_T\Phi_T\mathbf{b}_T + \mathbf{s}_{K_T}e^{j\phi_K} + \sigma\mathbf{n}_T, \tag{5.1}$$

where $\mathbf{S}_T$ is $TN \times T(K-1)$

$$\mathbf{S}_T = \begin{bmatrix} \mathbf{S} & & & \\ & \mathbf{S} & & \\ & & \ddots & \\ & & & \mathbf{S} \end{bmatrix},$$

$\mathbf{s}_{K_T}$ is $TN \times 1$

$$\mathbf{s}_{K_T} = \begin{bmatrix} \mathbf{s}_K \\ \mathbf{s}_K \\ \vdots \\ \mathbf{s}_K \end{bmatrix},$$

and Φ_T is $T(K-1) \times T(K-1)$

$$\Phi_T = \begin{bmatrix} \Phi & & & \\ & \Phi & & \\ & & \ddots & \\ & & & \Phi \end{bmatrix}.$$

In Section 4.1, the optimal weight/phase estimator was derived for the model of Equation (4.2), i.e. for $T = 1$. Equation (5.1), above, has the same form as Equation (4.2), thus, the optimal joint weight/phase *sequence* estimator may simply be written directly from the optimal joint weight/phase *one-shot* estimator shown in Equations (4.18) and (4.19).

$$\hat{\mathbf{b}}_T = \arg\min_{\mathbf{b}_T} [\, F_K(\mathbf{b}_T|\mathbf{r}_T) - 2|\mathcal{X}(\mathbf{b}_T|\mathbf{r}_T)|] \tag{5.2}$$

$$\hat{\phi}_K = \tan^{-1} \frac{\Im\{\mathcal{X}(\hat{\mathbf{b}}_T|\mathbf{r}_T)\}}{\Re\{\mathcal{X}(\hat{\mathbf{b}}_T|\mathbf{r}_T)\}}$$

Recall the definition

$$F_K(\mathbf{b}_T|\mathbf{r}_T) = ||\mathbf{r}_T - \mathbf{S}_T\Phi_T\mathbf{b}_T||^2.$$

Due to the block diagonal structure of $\mathbf{S}_T$ and Φ_T, $F_K(\mathbf{b}_T|\mathbf{r}_T)$ is separable by time frame, i.

$$F_K(\mathbf{b}_T|\mathbf{r}_T) = \sum_{i=1}^{T} F_K(\mathbf{b}(i)|\mathbf{r}(i))$$

This separation means that the calculation of $F_K(\mathbf{b}_T|\mathbf{r}_T)$ can be done independently from symbol frame to symbol frame. Moreover, as was the case in Chapters 3 and 4, for user signature sets exhibiting the tree structure, each term, $F_K(\mathbf{b}(i)|\mathbf{r}(i)$, is further separable through the conditioning upon weight values corresponding to upper levels of the tree. As was done in the one-shot estimator, standard tables may be constructed for nodes that are neither ancestors nor descendants of the unknown-phase user.

As each frame, $\mathbf{r}(i)$, is received, the algorithm proceeds as in the one-shot case for the rest of the tree that corresponds to non-descendants and non-ancestors of unknown-phase user. For example, if the tree of Figure 4-3 were used, at each time frame, standard tables would be created for nodes $\{1, 2, 3, 4, 5, 6, 9, 10, 11, 13\}$.

As was the case in Section 4.2, $\mathcal{X}(\mathbf{b}_T|\mathbf{r}_T)$ may be reduced to be a function of the weights of the ancestor and descendant users of node K, $\mathbf{b}_{adK_T}$. We have

$$\mathcal{X}(\mathbf{b}_T|\mathbf{r}_T) = \mathcal{X}(\mathbf{b}_{adK_T}|\mathbf{r}_T) = \mathbf{s}_{K_T}^T(\mathbf{r}_T - \mathbf{S}_{adK_T}\Phi_{adK_T}\mathbf{b}_{adK_T}),$$

where

$$\mathbf{b}_{adK_T} = \begin{bmatrix} \mathbf{b}_{aK}(1) \\ \mathbf{b}_{aK}(2) \\ \vdots \\ \mathbf{b}_{aK}(T) \\ \mathbf{b}_{dK}(1) \\ \mathbf{b}_{dK}(2) \\ \vdots \\ \mathbf{b}_{dK}(T) \end{bmatrix},$$

$$\mathbf{S}_{adK_T} = \begin{bmatrix} \mathbf{S}_{aK} & & & \mathbf{S}_{dK} & & \\ & \ddots & & & \ddots & \\ & & \mathbf{S}_{aK} & & & \mathbf{S}_{dK} \end{bmatrix},$$

and

$$\Phi_{adK_T} = \begin{bmatrix} \Phi_{aK} & & & & & \\ & \ddots & & & & \\ & & \Phi_{aK} & & & \\ & & & \Phi_{dK} & & \\ & & & & \ddots & \\ & & & & & \Phi_{dK} \end{bmatrix}.$$

The sequence estimator in Equation (5.2) requires the calculation of

$$|\mathcal{X}(\mathbf{b}_{adK_T}|\mathbf{r}_T)| = |\mathbf{s}_{K_T}^T(\mathbf{r}_T - \mathbf{S}_{adK_T}\Phi_{adk_T}\mathbf{b}_{adK_T})|$$

and cannot be separated into a term for each time frame, i. The calculation of $\mathcal{X}(\mathbf{b}_{adK_T}|\mathbf{r}_T)$, however, may be done in a cumulative manner, frame by frame. Since

$\mathbf{S}_{adK_T}$ and Φ_{adK_T} are block diagonal, we may write

$$\mathcal{X}(\mathbf{b}_{adK_T}|\mathbf{r}_T) = \sum_{i=1}^{T} \mathcal{X}(\mathbf{b}_{adK}(i)|\mathbf{r}(i)).$$

It follows that the sequence weight/phase estimator of Equation (5.2) can be written to indicate time frame separation.

$$\hat{\mathbf{b}}_T = \arg \min_{\{\mathbf{b}(i)\}_{i=1}^{T}} \sum_{i=1}^{T} F_K(\mathbf{b}(i)|\mathbf{r}(i)) - 2 \left| \sum_{i=1}^{T} \mathcal{X}(\mathbf{b}_{adK}(i)|\mathbf{r}(i)) \right| \tag{5.3}$$

$$\hat{\phi}_K = \tan^{-1} \frac{\Im\{\mathcal{X}(\hat{\mathbf{b}}_{adK_T}|\mathbf{r}_T)\}}{\Re\{\mathcal{X}(\hat{\mathbf{b}}_{adK_T}|\mathbf{r}_T)\}} \tag{5.4}$$

The sequence weight/phase estimation procedure is broken into three parts.

1. For each time frame, i, calculate the complex scalar $\mathcal{X}(\mathbf{b}_{adK}(i)|\mathbf{r}(i))$ for each possible realization of $\mathbf{b}_{adK}(i)$ and each measurement vector $\mathbf{r}(i)$, and store in a table. This table will be referred to as the **hyper-$\mathcal{X}$-table**; an example is shown in Table 5.1. Each column corresponds to a particular time frame, i. Each row corresponds to a particular realization of $\mathbf{b}_{adK}$; there are $M^{|adK|}$ rows, where $|adK|$ is the number of ancestors and descendants to node K.

2. For each time frame, i, create a standard table of conditional weight estimates for the non-ancestor/descendant nodes of user K. This part of the procedure is identical to the processing done for this same portion of the tree in the one-shot estimation procedure. We must, however, leave the rest of the tree un-processed until the last frame, $\mathbf{r}(T)$, is received.

3. After the last frame is observed, we may process the rest of the tree corresponding to the ancestors and descendants of user K. Specifically, we may calculate

Table 5.1: Hyper-$\mathcal{X}$-table created for part 1 of sequence weight/phase estimation algorithm.

$\mathbf{b}_{adK}$	$\mathcal{X}(\mathbf{b}_{adK}(1)\|\mathbf{r}(1))$	$\cdots$	$\mathcal{X}(\mathbf{b}_{adK}(T)\|\mathbf{r}(T))$
1 1 1 $\cdots$ 1			
-1 1 1 $\cdots$ 1			
1 -1 1 $\cdots$ 1			
$\vdots$	$\vdots$	$\vdots$	$\vdots$
-1 -1 -1 $\cdots$ -1			

the joint weight estimate of $\mathbf{b}_{adK_T}$ using

$$\hat{\mathbf{b}}_{adK_T} = \arg \min_{\{\mathbf{b}_{adK}(i)\}_{i=1}^{T}} \sum_{i=1}^{T} F_K(\mathbf{b}_{adK}(i), \hat{\mathbf{b}}_{rest}(i|\mathbf{r}(i), \mathbf{b}_{aK}(i)) - 2|\sum_{i=1}^{T} \mathcal{X}(\mathbf{b}_{adK}(i)|\mathbf{r}(i))|$$

where $\hat{\mathbf{b}}_{rest}(i|\mathbf{r}(i), \mathbf{b}_{aK}(i))$ is found in the standard tables at the non-ancestor/descendant nodes that were previously constructed in step 2, above. Equivalently, we may do the above joint estimate in two parts, as was done with the one-shot non-coherent estimator. See the discussion that follows this list of steps.

4. As was the case for the one-shot joint weight/phase estimator, the decision made at the top of the trees finalizes all decisions made all lower levels in all T trees. Conceptually, the top decisions (at level 1 of the T trees) are used to look up decisions in the tables at level 2. The level 2 and level 1 decisions, together, are used to look up decisions at level 3, and so on, until the lowest level of the tree is reached.

Conceptually, we may achieve part 3 of the algorithm as if we were climbing up a single tree, but instead of estimating a *scalar* weight at each node, we are estimating a *vector* weight at each node. First at hyper-node K (the collection of node-K's in the T trees), we create the **hyper-descendant table** of estimates of $\mathbf{b}_{dK_T}$ conditioned upon

Table 5.2: Hyper-descendant table created for part 3 of sequence weight/phase estimation algorithm.

$\mathbf{b}_{aK_T}$	$\hat{\mathbf{b}}_{dK_T}(\mathbf{r}_T \mid \mathbf{b}_{aK_T})$
1 1 $\cdots$ 1	
-1 1 $\cdots$ 1	
1 -1 $\cdots$ 1	
$\vdots$	$\vdots$
-1 -1 $\cdots$ -1	

Table 5.3: Hyper-ancestor table created for part 3 of sequence weight/phase estimation algorithm.

$\mathbf{b}_{aj_T}$	$\hat{\mathbf{b}}_{j_T}(\mathbf{r}_T \mid \mathbf{b}_{aj_T}, \hat{\mathbf{b}}_{rest}(i \mid \mathbf{r}_T, \mathbf{b}_{aj_T}), \mathbf{b}_{dK_T}(\mathbf{r}_T \mid \mathbf{b}_{aK_T}))$
1 1 $\cdots$ 1	
-1 1 $\cdots$ 1	
$\vdots$	$\vdots$
-1 -1 $\cdots$ -1	

the **hyper-ancestor vector**, $\mathbf{b}_{aK_T}$ like the one shown in Table 5.2.[2] The **hyper-descendant vector**, $\mathbf{b}_{dK_T}$, has $T|dK|$ elements and the hyper-ancestor vector has $T|aK|$ elements. The hyper-descendant table, then, has one entry for each of the $M^{T|aK|}$ possible realizations of $\mathbf{b}_{aK_T}$. Each entry is chosen from the $M^{T|dK|}$ possible realizations of $\mathbf{b}_{dK_T}$.[3] The remainder of the hyper-tree may be climbed. At a node j which is an ancestor to user K, replace b_j in the joint weight/phase one-shot estimator with $\mathbf{b}_{j_T} = [b_j(1)\ b_n(2)\ \cdots\ b_j(T)]$. Recall that the one-shot estimator would create an ancestor table at node j where a single table entry would be chosen from M possible values of b_j. If we treat this portion of the sequence estimator as if it were the corresponding portion of a one-shot estimator having *vector* weights instead of scalar weights, then a single table entry must be chosen from M^T possible realizations of $\mathbf{b}_{j_T} = [b_j(1)\ b_j(2)\ \cdots\ b_j(T)]$. See the example hyper-ancestor table in Table 5.3.

5.1.2 Computational Complexity

Consider Figure 5-1. The figure shows three trees ($T = 3$), one for each time frame, $t = 1, 2, 3$. The unknown phase user is at node n. The pink section of the trees corresponds to nodes that are neither descendants nor ancestors of node n. The black section of the tree corresponds to the descendant nodes of node n and the blue section of the tree corresponds to the ancestor nodes of node n. From the description of the sequence algorithm in the previous section, we know that the calculations for the pink sections of each of the T trees being jointly processed is exactly the same as if we were performing the one-shot weight/phase estimation algorithm on T independent trees. The black and blue sections of each of the T trees, however, must be processed together since they are coupled by the unknown, but constant phase at node n of each of the T trees. It follows from the description of part

[2]Recall that a motivation for creating a descendant table on its own is that it doesn't require pulling values from the standard tables at the non-ancestor/descendant nodes as does the creation of ancestor tables.

[3]Creation of this table would require a complexity of $M^{T|aK|}(M^{T|dK|} - 1)$.

3 of the estimation procedure in Section 5.1.1 that the complexity of the sequence estimator corresponding to the black and blue portions of the trees is equivalent to the complexity of the one-shot algorithm, but for M^T-ary signaling. The computational complexity, c_S, of the sequence weight/phase estimator is

$$c_S(L, L_n, Q, M, T) = c_{pink}(L, L_n, Q, M, T) + c_{black}(L, L_n, Q, M, T) + c_{blue}(L_n, M, T),$$

where $c_{pink}(L, L_n, Q, M)$, $c_{black}(L, L_n, Q, M)$ and $c_{blue}(L, L_n, Q, M)$ are each derived below.

Pink sections of trees: The joint weight/phase estimation procedure in the pink section of the trees is exactly the same as doing T one-shot procedures. Recall that the green section of the tree in Figure 4-4 is just the pink *and* blue sections of the tree in Figure 5-1 and $c_{green}(L, L_n, Q, M)$ was calculated in Section 4.3. Before multiplying by T we must subtract out the complexity of making standard tables for the blue section of a tree. This complexity is just

$$\begin{aligned} c_{standard-blue}(M, L_n) &= \textstyle\sum_{k=1}^{L_n+1}(M-1)M^{l-1} \\ &= M^{L_n-1} - 1 \end{aligned}$$

and

$$\begin{aligned} c_{pink}(L, Q, M) &= T[c_{green}(L, L_n, Q, M) - c_{standard-blue}(M, L_n)] \\ &= \tfrac{T(M-1)}{QM-1}[(QM)^L - 2 - M^{L_n-1}[(QM)^{L-L_n+1} - 2]] \end{aligned} \tag{5.5}$$

Black sections of trees: In the black sections of the T trees, we must calculate the hyper-$\mathcal{X}$-table and the hyper-descendant table. The hyper-$\mathcal{X}$-table, as shown in Table 5.1, is constructed with $TM^{|adn|}$ calculations. That is, for each time frame, $t = 1, 2, \cdots, T$, we may calculate one column of the table; each column has $M^{|adn|}$ entries.

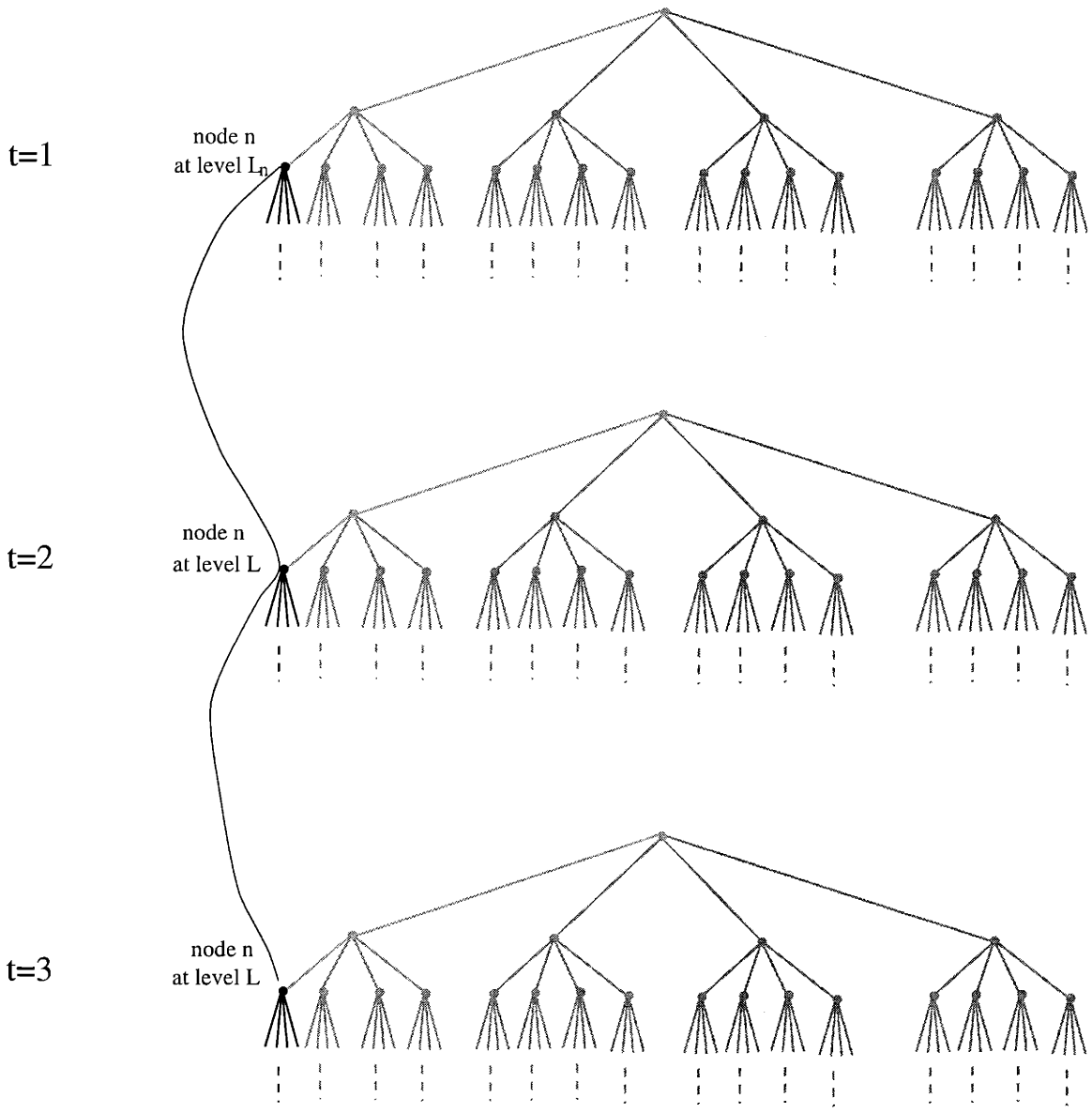


Figure 5-1: Three quad trees, each partitioned into three sections by color.

The hyper-descendant table as shown in Table 5.2 is also needed for the black section. This construction requires $[(M^T)^{|dn|} - 1](M^T)^{|an|}$ compares. That is, each entry, $\hat{\mathbf{b}}_{dn_T}(\mathbf{r}_T|\mathbf{b}_{an_T})$, is the choice of one possible realization from a total of $(M^T)^{|dn|}$ possible realizations of $\mathbf{b}_{dn_T}$; there are $(M^T)^{|an|}$ entries in the table since there are $(M^T)^{|an|}$ possible realizations of $\mathbf{b}_{an_T}$.

Noting that

$$\begin{aligned} |an| &= L_n - 1 \\ |dn| &= \textstyle\sum_{j=L_n+1}^{L} Q^{j-L_n} \\ &= \frac{Q}{Q-1}\left(\frac{Q^{L-1}}{Q^{L_n-1}} - 1\right) \end{aligned}$$

gives

$$c_{black}(L, L_n, Q, M, T) = TM^{L_n-1} + [(M^T)^{\frac{Q}{Q-1}(\frac{Q^{L-1}}{Q^{L_n-1}}-1)} - 1](M^T)^{L_n-1}. \tag{5.6}$$

Blue sections of trees: Construction of a hyper-ancestor table at a node j in the blue section of the tree requires $[(M^T) - 1](M^T)^{|aj|}$ compares. An example table is given in Table5.3. Each entry, $\hat{\mathbf{b}}_{j_T}(\mathbf{r}_T|\mathbf{b}_{aj_T})$, is the choice of one possible realization from a total of M^T possible realizations of the vector weight $\mathbf{b}_{j_T} = [b_j(1), \cdots, b_j(T)]$; this choice requires $M^T - 1$ compares. There are $(M^T)^{|aj|}$ entries in the table since there are $(M^T)^{|aj|}$ possible realizations of $\mathbf{b}_{aj_T}$. Summing the complexities of making one hyper-ancestor table at each level, noting that $|aj| = l - 1$ for node j at level l gives

$$\begin{aligned} c_{blue}(Ln, M, T) &= \textstyle\sum_{k=L_n+1}^{L}[(M^T) - 1](M^T)^{l-1} \\ &= (M^T)^{L_n-1} - 1 \end{aligned} \tag{5.7}$$

Total Complexity: The total complexity is, simply the aggregate of Equations (5.5), (5.6), and (5.7) and is not explicitly shown here. If the unknown-phase user is any-

where except at the bottom of the tree the total complexity of the optimal sequence joint weight/phase estimator is dominated by c_{black}, namely, the complexity required to create the hyper-descendant table is exponential in the number of ancestors *and* descendants times T; this becomes prohibitive for an unknown-phase user having several descendants. If, alternatively, the unknown-phase user is at the bottom of the tree, then the total complexity is dominated by c_{blue}, the complexity needed to create the hyper-ancestor tables; this complexity is exponential in the number of ancestors times T and can become impractical for relatively small values of T if a tree has 4 or more levels.

Table 5.4 shows, for values of T ranging from 2 to 9, $c_S(L, L_n, Q, M, T)$ for $Q = 4$, $M = 2$, $L = 4$ and $L_n = 4, 3$, and 2. The realistically reasonable complexities given in the table are are shown in bold. Notice that even with complexity constraints on the order of 10^5 comparisons per user per weight decision, it is possible to use the optimal sequence detector for $L_n = L$ (where the newest user is at the bottom of the tree) for a value of T as high as 6 and, possibly, 7. If the new, unknown-phase user is one level up from the bottom, T can only be as high as 3.

Due to the fast growing complexity of the optimal joint weight/phase sequence estimator with the number of symbols, T, two alternative procedures of negligible increase in complexity over the one-shot joint weight/phase estimator are proposed in the next section.

5.2 Multi-frame Phase Estimation

This section develops two methods for using the training sequence to improve the phase estimate relative to the one-shot scenario. Both methods proposed in this section require only a *negligible increase* in complexity compared to that of the one-shot procedure. The goal for each of these procedures is to produce a phase estimate

Table 5.4: Computational complexity, c_S, of sequence estimation algorithm for $M = 2$, and $Q = 4$, $L = 4$ total levels, $N = 64$ dimensions , and the number of users, $K = 85$.

T	$c_s(Ln = L)$	$c_s(Ln = L-1)$	$c_s(Ln = L-2)$
2	**1215**	**5191**	4.40e+13
3	**2239**	**2.64e+05**	9.22e+18
4	**6399**	1.68e+07	1.93e+25
5	**3.56e+4**	1.07e+09	4.06e+31
6	**2.66e+05**	6.87e+10	8.51e+37
7	**2.10e+06**	4.40e+12	1.78e+44
8	1.69e+07	2.81e+14	3.74e+50
9	1.34e+08	1.80e+16	7.85e+56

that has a standard deviation of 2.5°.[4]

In Section 5.2.1, the **multi-frame phase estimate average** is proposed. This estimator collects T successive one-shot phase estimates and averages them. The value of T needed by this estimator to achieve an estimate with a standard deviation of 2.5° is on the order of 30. In Section 4.7.2 it was found that with small probability, the one-shot non-coherent estimator will give an erroneous phase estimate. The **robust multi-frame phase estimate average** is proposed to take outlier estimates into account. The robust estimate, simply, discards possible outliers prior to the calculation of the average.

In Section 4.7.1, results from the simulations with the 21 unit energy minimum distance set showed that the only users to experience loss in their BER's due to the existence of the unknown-phase user were the children and parent of that user. The losses on the parent and children ranged from 0.25 dB to 0.75 dB. Note that in order to calculate the T-frame phase estimate average, the estimator must wait for the arrival of T phase estimates. During these T symbol frames, the affected users must

[4]Recall that from Section 3.7.2 it was found from simulations that the coherent tree joint detector can withstand a phase mismatch of up to 5° on each user. If the receiver error in a user's phase has a standard deviation of 2.5°, then with 0.95 probability the mismatch is within ±5°.

withstand a higher BER.

The affected users might not be able to tolerate the higher BER for more than a few successive frames. Section 5.2.2 proposes the **recursive phase estimate** to alleviate long exposure of affected users to higher BER's. Briefly, this recursive scheme makes an interim phase estimate after only five symbol frames. The partial information in this interim estimate is used as the seed for a recursive phase estimation procedure. The recursive estimation procedure continues by using the information from the estimate made at one frame to help in making the estimate at the next frame, until an estimate having a standard deviation of 2.5° is found. In a specific example at an SNR of 5.528 dB, this method significantly reduces the BER for the affected users after only five frames, and returns a phase estimate having a 2.5° standard deviation after only 7 frames.

5.2.1 Multi-frame Average

The MAP two-frame estimator is derived and shown to be a simple average of two successive, but independently calculated, one-shot optimal phase estimates. The **multi-frame estimate average**, in which multiple one-shot phase estimates are averaged, is proposed and analyzed. Given that each one-shot optimal phase estimate has some non-zero probability of being an outlier, a **robust multi-frame estimate average** is proposed in which possible outliers are discarded prior to averaging. The number of total estimates that need to be collected and the maximum number to be discarded is calculated.

If we are interested only in estimating one user's phase from $\mathbf{r}(1)$ and $\mathbf{r}(2)$, we may choose the value of this user's phase, ϕ, to maximize the following conditional probability

$$p_{\phi|\mathbf{r}(1),\mathbf{r}(2)}(\phi|\mathbf{r}(1)\mathbf{r}(2)) = p_{\mathbf{r}(1),\mathbf{r}(2)|\phi}(\mathbf{r}(1),\mathbf{r}(2)|\phi)\,\frac{p_{\phi}(\phi)}{p_{\mathbf{r}(1),\mathbf{r}(2)}(\mathbf{r}(1),\mathbf{r}(2))}. \tag{5.8}$$

Such an estimate would be the MAP phase-only estimate which, compared to the estimator derived in Section 5.1.1, averages out the role of the user weights.[5]

Given the value for user K's phase, the received signals from two disjoint frames, $\mathbf{r}(1)$ and $\mathbf{r}(2)$, are statistically independent. We may, then, re-write Equation (5.8) as

$$p_{\phi|\mathbf{r}(1),\mathbf{r}(2)}(\phi|\mathbf{r}(1)\mathbf{r}(2)) = p_{\mathbf{r}(1)|\phi}(\mathbf{r}(1)|\phi)p_{\mathbf{r}(2)|\phi}(\mathbf{r}(2)|\phi)\,\frac{p_\phi(\phi)}{p_{\mathbf{r}(1),\mathbf{r}(2)}(\mathbf{r}(1),\mathbf{r}(2))} \tag{5.9}$$

Furthermore, the denominator in Equation (5.9) is, simply, a normalization constant. Thus, when viewed as a function of ϕ, we see that

$$\begin{aligned} p_{\phi|\mathbf{r}(1),\mathbf{r}(2)}(\phi|\mathbf{r}(1),\mathbf{r}(2)) &\propto p_{\mathbf{r}(1)|\phi}(\mathbf{r}(1)|\phi)p_\phi(\phi)\;\; p_{\mathbf{r}(1)|\phi}(\mathbf{r}(2)|\phi)p_\phi(\phi) \\ &= p_{\phi|\mathbf{r}(1)}(\phi|\mathbf{r}(1))\; p_{\phi|\mathbf{r}(2)}(\phi|\mathbf{r}(2)). \end{aligned} \tag{5.10}$$

Recall that user K's phase is modeled to be random with a uniform PDF, thus, $p_\phi(\phi)$ is a constant. The simulation results in Section 4.7.2 found that for a single time frame, i, and for a uniform ($\pm 180°$) prior PDF on ϕ, $p_{\phi|\mathbf{r}(i)}(\phi|\mathbf{r}(i))$ is multi-modal having one dominant lobe and, possibly, several very small lobes. The existence of these small lobes is ignored for this derivation, and the PDF, $p_{\phi|\mathbf{r}(i)}(\phi|\mathbf{r}(i))$, may be reasonably approximated by the exponential cosine[6]

$$p_{\phi|\mathbf{r}(i)}(\phi|\mathbf{r}(i)) = \frac{e^{\alpha_{one\text{-}shot}\cos(\phi-\hat{\phi}(i))}}{2\pi I_o(\alpha_{one\text{-}shot})},\; i = 1,2, \tag{5.11}$$

where the mean, $\hat{\phi}(i)$, is the one-shot estimate from the non-coherent joint weight/phase estimator. Note that in order for this model to be valid, $\hat{\phi}(i)$ cannot be an outlier. By modeling $p_{\phi|\mathbf{r}(i)}(\phi|\mathbf{r}(i))$ with the PDF in (5.11) we are assuming zero probability of

[5]This estimate is optimal, in the sense that it is the Bayesian estimator that minimizes the risk defined with the uniform cost function, $C(\hat{\phi}-\phi) = 1$ if $\hat{\phi} = \phi$ and 0 otherwise ([20]).

[6]This exponential cosine is exactly equal to the conditional probability we are seeking if user K was the only user in the system ([24, 25]).

an outlier estimate. The standard deviation, $\sigma_{one\text{-}shot}$, can be empirically determined from simulations. From the standard deviation, we may find the value for $\alpha_{one\text{-}shot}$ by using the relation

$$\sigma(\alpha) = \left[\frac{\pi^2}{3} + \sum_{n=1}^{\infty} \frac{(-1)^n I_n(\alpha)}{n^2 I_o(\alpha)}\right]^{1/2}. \tag{5.12}$$

derived by Viterbi in [24]. Specifically, from Equation (5.12) a table of $\{\alpha, \sigma(\alpha)\}$ pairs may be constructed and used to find $\alpha_{one\text{-}shot}$ from $\sigma_{one\text{-}shot}$. In Section 4.7.2, the simulation run at $E_b/N_o = 5.528$ dB, found the effective standard deviation[7] of the one-shot non-coherent phase estimate, $\sigma_{one\text{-}shot}$, to be 13.5°; the corresponding $\alpha_{one\text{-}shot}$ is 18.56.

The substitution of Equation (5.11) into Equation (5.10) gives

$$\begin{aligned} p_{\phi|\mathbf{r}(1),\mathbf{r}(2)}(\phi|\mathbf{r}(1),\mathbf{r}(2)) &\propto \frac{1}{4\pi^2 I_o(\alpha_{one\text{-}shot})^2} \exp(\alpha_{one\text{-}shot}[\cos(\phi - \hat{\phi}(1)) + \cos(\phi - \hat{\phi}(2))]) \\ &= \frac{1}{4\pi^2 I_o(\alpha_{one\text{-}shot})^2} \exp(2\alpha_{one\text{-}shot} \cos(\tfrac{\hat{\phi}(1)-\hat{\phi}(2)}{2})[\cos(\phi - \tfrac{\hat{\phi}(1)+\hat{\phi}(2)}{2})]) \end{aligned} \tag{5.13}$$

where the identity

$$\cos(a) + \cos(b) = 2\cos(\frac{a+b}{2})\cos(\frac{a-b}{2}) \tag{5.14}$$

was used.

Before maximizing the function on the right hand side of Equation (5.13) we take the log and discard all constants. The MAP two-frame phase estimator is

$$\hat{\phi}(1,2) = \arg\max_{\phi} \cos(\phi - \frac{\hat{\phi}(1) + \hat{\phi}(2)}{2}). \tag{5.15}$$

The value of ϕ we are seeking sets the argument of the cosine in Equation (5.15) to zero, hence, the MAP two-frame phase estimate is a simple average of two indepen-

[7] Recall that the effective standard deviation best described the error histogram if we are to ignore the outlier lobes.

dently calculated phase estimates, one for each frame.

$$\boxed{\hat{\phi}(1,2) = \tfrac{1}{2}(\hat{\phi}(1) + \hat{\phi}(2))} \tag{5.16}$$

From the form of Equation (5.13), we know that the PDF for the error probability of the two-frame phase estimate is of the form

$$p_{(\hat{\phi}(1,2)-\phi_{true})}(\phi_e) = C \exp[2\alpha_{one\text{-}shot} \cos(\frac{\hat{\phi}(1) - \hat{\phi}(2)}{2}) \cos(\phi_e)], \tag{5.17}$$

where ϕ_e is the error in the estimate and C is a normalization constant. Recall that in an exponential cosine function, the accuracy parameter multiplies the $\cos(\phi_e)$ above. The accuracy parameter for the two-frame estimate average, $\hat{\phi}(1,2)$, is

$$\alpha_{\hat{\phi}(1,2)} = 2\alpha_{one\text{-}shot} \cos(\frac{\hat{\phi}(1) - \hat{\phi}(2)}{2}). \tag{5.18}$$

Recall that the one-shot optimal phase estimate, $\hat{\phi}(i)$, is a random variable since it is a function of the noisy received signal vectors, $\mathbf{r}(1)$ and $\mathbf{r}(2)$. This means that the accuracy parameter for the two-frame estimate average is also a random variable. From Equation (5.18) we see that an upper bound is,

$$\alpha_{\hat{\phi}(1,2)} \leq 2\alpha_{one\text{-}shot}. \tag{5.19}$$

This tells us that accuracy value of the two-frame estimate average is no more than twice that of the one-shot optimal non-coherent phase estimate.

Recognizing that the estimate $\hat{\phi}(1,2)$, shown in Equation (5.16), has an exponential cosine PDF allows us to use the accuracy parameter, $\alpha_{\hat{\phi}(1,2)}$, found in Equation (5.18), to calculate the standard deviation of the two-frame estimate average. Specifically, the conditional standard deviation would be found by substituting $\alpha_{\hat{\phi}(1,2)}$ for

α in Equation (5.12). This α-dependent standard deviation is denoted by $\sigma(\alpha)$, and is defined in terms of variance as

$$\sigma^2(\alpha_{\hat{\phi}(1,2)}) = E[(\phi - \hat{\phi}(1,2))^2 | \mathbf{r}(1), \mathbf{r}(2)], \tag{5.20}$$

where $E[\cdot]$ denotes the expectation.

This conditional standard deviation, $\sigma(\alpha_{\hat{\phi}(1,2)})$, is a function of the random vectors $\mathbf{r}(1)$ and $\mathbf{r}(2)$, hence, it is also a random variable. We may, however, calculate a lower bound on the conditional standard deviation. Recall that an upper bound was found for $\alpha_{\hat{\phi}(1,2)}$ in Equation (5.19), namely, $\alpha_{\hat{\phi}(1,2)} \leq 2\alpha_{one\text{-}shot}$. Also recall that the accuracy parameter, α, and its corresponding standard deviation, $\sigma(\alpha)$, are inversely related. From Equation (5.12) and from the *upper bound* on $\alpha_{\hat{\phi}(1,2)}$, we may calculated a *lower bound* on $\sigma(\alpha_{\hat{\phi}(1,2)})$. Specifically

$$\sigma(2\alpha_{one\text{-}shot}) \leq \sigma(\alpha_{\hat{\phi}(1,2)}), \tag{5.21}$$

where $\sigma(2\alpha_{one\text{-}shot})$ is found by substituting $\alpha = 2\alpha_{one\text{-}shot}$ in Equation (5.12). Equation (5.21) tells us that the standard deviation of the two-frame estimate average is no less than that corresponding to a doubling of the accuracy parameter of the one-shot non-coherent phase estimate.

Alternatively, we may consider the variance, $E[(\phi - \hat{\phi}(1,2))^2]$, which, in contrast to to Equation (5.20), is not conditional on the measurements $\mathbf{r}(1)$ and $\mathbf{r}(2)$. Recall that $\hat{\phi}(1,2)$ in Equation (5.16) is, simply, the average of two independent identically distributed random variables, $\hat{\phi}(1)$ and $\hat{\phi}(2)$, namely, $\hat{\phi}(1,2) = \frac{1}{2}[\hat{\phi}(1) + \hat{\phi}(2)]$. Let us use $\sigma_{ave}(2)$ to denote the standard deviation of $\hat{\phi}(1,2)$. In terms of variance, we have

$$\sigma^2_{ave}(2) = E[(\phi - \hat{\phi}(1,2))^2]. \tag{5.22}$$

Contrasting this with Equation (5.20), we see that $\sigma^2_{ave}(2)$ is the average variance in

the error $\phi - \hat{\phi}(1,2)$ (i.e. averaged over $\mathbf{r}(1)$ and $\mathbf{r}(2)$). Since the two independent random variables, $\hat{\phi}(1)$ and $\hat{\phi}(2)$, have the same standard deviation, $\sigma_{one\text{-}shot}$, we have

$$\sigma_{ave}(2) = \frac{\sigma_{one\text{-}shot}}{\sqrt{2}}. \tag{5.23}$$

Consider, for example, the two-frame phase estimate average from two sequential one-shot optimal phase estimates obtained by the simulation in Section 4.7.2 for $E_b/N_o = 5.528$ dB. As reported in Table 4.12, the effective standard deviation for each one-shot estimate is $\sigma_{one\text{-}shot} = 13.5°$. The standard deviation of the two-frame phase estimate average, assuming neither of the one-shot estimates is an outlier, is

$$\sigma_{ave}(2) = 13.5°/\sqrt{2} = 9.5°.$$

Recall that the coherent joint weight phase estimator needs an estimate with standard deviation no greater than 2.5°. Therefore, $T = 2$ is insufficient and we must extend the phase estimate average to accommodate $T > 2$.

The MAP T-frame estimate would be found by maximizing

$$p_{\phi|\mathbf{r}(1),\cdots,\mathbf{r}(T)}(\phi|\mathbf{r}(1),\cdots,\mathbf{r}(T)) \propto \prod_{i=1}^{T} \exp[\alpha_{one\text{-}shot}\cos(\phi - \hat{\phi}(i))], \tag{5.24}$$

over ϕ. Equivalently,

$$\hat{\phi}(1,\cdots,T) = \arg \max_{\phi} \ \alpha_{one\text{-}shot} \sum_{i=1}^{T} \cos(\phi - \hat{\phi}(i)).$$

From the identity in Equation (5.14), we may write

$$\begin{aligned} \Psi(\phi) &= \alpha_{one\text{-}shot} \textstyle\sum_{i=1}^{T} \cos(\phi - \hat{\phi}(i)) \\ &= 2\alpha_{one\text{-}shot} \textstyle\sum_{i=1}^{T/2} \cos(\phi - \frac{\hat{\phi}(i)+\hat{\phi}(i+1)}{2})\gamma(i) \end{aligned} \tag{5.25}$$

where,

$$\gamma(i) = \cos(\frac{\hat{\phi}(i) - \hat{\phi}(i+1)}{2}),$$

where T is assumed to be a power of 2. Further reduction of Equation (5.25) cannot be done with trigonometric identities since $\gamma(i)$ is different for every i. Furthermore, $\gamma(i)$ is a random variable. Direct derivation of an estimator from Equation (5.25) is not done in this thesis.

Alternatively, we may, view the one-shot phase estimate at frame i, $\hat{\phi}(i)$, as a noisy measurement of the true phase, ϕ_{true},

$$\hat{\phi}(i) = \phi_{true} + \epsilon_i, \tag{5.26}$$

where the phase measurement noise at frame i, ϵ_i, is a zero mean random variable with standard deviation, $\sigma_{one\text{-}shot}$. Furthermore, ϵ_i is independent and identically distributed for all i. Given multiple independent noisy measurements of ϕ_{true}, a good estimate is the average of the measurements. The T-frame estimate average, then, is proposed to be

$$\boxed{\hat{\phi}(1, \cdots, T) = \frac{1}{T} \sum_{j=1}^{T} \hat{\phi}(j)} \tag{5.27}$$

The standard deviation is, simply,

$$\boxed{\sigma_{ave}(1, \cdots, T) = \sigma_{one\text{-}shot}/\sqrt{T}} \tag{5.28}$$

Recall that we need a standard deviation of $\pm 2.5°$. This requires a number of training frames,

$$T = (\sigma_{one\text{-}shot})^2/(2.5)^2.$$

For the example of the simulation in Section 4.7.2 for which $E_b/N_o = 5.528$ dB, $\sigma_{one\text{-}shot} = 13.5°$, and we would need to average

$$T = (13.5)^2/(2.5)^2 \approx 30 \tag{5.29}$$

one-shot optimal, independently calculated, non-coherent phase estimates. Notice that with a decrease in the SNR, this number rises significantly. For example, at $E_b/N_o = 4.6$ dB, $\sigma_{one\text{-}shot} = 15.6°$ and $T \approx 40$.[8]

The derivation up to this point did not take into account the multi-modal nature of the actual PDF of the one-shot phase estimate, $p_{\phi|\mathbf{r}(i)}(\phi|\mathbf{r}(i))$. By assuming the one-shot phase estimate to have the exponential cosine PDF (which does not produce outliers), all results found up to this point are for the case in which outliers are not a part of the average. We now formally recognize the possibility of having an outlier estimate among those to be averaged. With a probability of p, the estimate is an outlier, and, hence, is useless. We may calculate a robust average by discarding any of the estimates that are believed to be outliers.

Before writing the procedure for the **robust multi-frame estimate average**, we must determine which of the estimates to keep and which to discard. Whether or not an estimate is an outlier can, of course, be formally decided via a hypothesis test. This degree of sophistication is not needed here. A simpler approach is to pre-determine the total number of estimates that will be collected, T_T, and the maximum number that will be discarded, T_D, and, simply, discard up to T_D estimates that are in disagreement with the majority. The robust T_T-frame estimate average would be found by averaging the remaining estimates.

For example, the smallest value for T_T that will allow for a majority is $T_T = 3$. In this case, the only reasonable possibility for T_D is 1. The robust three-frame estimate

[8]Note that during these T frames, the non-coherent joint weight phase estimator gives a degraded BER for some users in the system. This means that the affected users must withstand T successive frames of degraded performance. This problem is dealt with in Section 5.2.2.

average is found by collecting three sequential, but independently calculated, one-shot phase estimates, discarding up to one that is in disagreement with the other two, and averaging the remaining two (or three, if no estimate was deemed to be an outlier). Denote the robust three-frame estimate average by

$$\hat{\phi}(3 \setminus 1)$$

to indicate that 3 one-shot estimates are collected and at most 1 is discarded.

This procedure for the robust three-frame estimate average will give an erroneous estimate if more than one of the three one-shot estimates is an outlier. Therefore, we wish to calculate the probability that the estimate is good, i.e. only non-outliers were averaged, when $T_T = 3$ and $T_D = 1$. Denoting this probability by $P_g(T_T, T_D) = P_g(3,1)$, we have

$$P_g(3,1) = \begin{pmatrix} 3 \\ 2 \end{pmatrix} (1-p)^2 p + \begin{pmatrix} 3 \\ 3 \end{pmatrix} (1-p)^3, \tag{5.30}$$

where p is the probability that a one-shot estimate is an outlier. The probability, $P_g(3,1)$, then, is the probability that no more than one of the three one-shot estimates is an outlier.

As was found from simulations reported in Section 4.7.2, $0.04 \leq p \leq 0.082$, depending on the SNR. For the case in which $E_b/N_o = 5.528$ dB, $p = 0.053$, and the probability of the robust three-frame estimate average being good is $P_g(3,1) = 0.9919$. If a 2.5° standard deviation can be obtained by averaging only two one-shot non-outlier estimates, then we would use this robust three-frame estimate average only if we are willing to risk getting one bad three-frame estimate in every 100.

In general, collecting T_T one-shot non-coherent phase estimates, discarding up to T_D, and averaging the rest, will be called the robust T_T-frame phase estimate averaging procedure. The **robust T_T-frame phase estimate average** will be denoted

by

$$\hat{\phi}(T_T \setminus T_D)$$

to indicate that T_T one-shot estimates are collected and up to T_D are discarded.

The general formula for the probability that the T_T-frame robust estimate average is free of outliers is

$$P_g(T_T, T_D) = \sum_{i=0}^{T_D} \begin{pmatrix} T_T \\ (T_T - i) \end{pmatrix} (1-p)^{(T_T - i)} p^i. \tag{5.31}$$

If, for example, we set $T_D = 2$ (discard up to 2 estimates), then the smallest possible choice for T_T is 5.[9] The **robust 5-frame estimate average**,

$$\hat{\phi}(5 \setminus 2)$$

is found by collecting $T_T = 5$ one-shot estimates, discarding up to $T_D = 2$ possible outliers, and averaging the rest. Also for $E_b/N_o = 5.528$ dB and $p = 0.053$, the probability that the robust 5-frame estimate average is good is $P_g(5, 2) = 0.9986$. In other words, one in every 714 robust 5-frame estimate averages is expected to be bad.

At an SNR of 5.528 dB, we found, in Equation (5.29), that we needed to average 30 one-shot phase estimates to achieve a standard deviation of 2.5°. If we require a 0.9986 probability that the robust estimate average of at least 30 estimates does not include an outlier, then $T_D = 6$ and $T_T = 36$.[10] In other words to achieve a rate of only 1 bad robust estimate average in every 714, we calculate a **robust 36-frame estimate average**. Specifically, we need to collect 36 one-shot optimal phase estimates and discard up to 6 that could be outliers, before calculating the average.

Note the computation of the robust 36-frame estimate average requires the receiver

[9] Recall that we discard an estimate if it is in disagreement with the majority.

[10] If $T_T = 35$ and $T_D = 5$, $P_g(35, 5) = 0.9956$ which is lower than the requirement of 0.9986, for this example. Trying $T_T = 36$ and $T_D = 6$ gives $P_g(36, 6) = 0.9902$ which more than meets the requirement.

to use the non-coherent joint weight phase estimator for 36 successive frames. This means that any users that are affected by the presence of the unknown-phase user are subject to a higher BER for 36 successive frames. This problem is discussed in the next section.

5.2.2 Recursive Refinement

From the BER curves for the simulations in Section 4.7.1 we found that when a new user with an unknown phase joins the system, the children and parent of this user each experience an increase in the probability that their one-shot weight estimate is in error. For example, at $E_b/N_o = 5.528$ dB, it was found that for the 21 user minimum distance set, an unknown phase on user 17 causes the BERs for users 1 and 21 to nearly double. Specifically, user 1 has a coherent BER of 4.2×10^{-3} and a non-coherent BER of 7.8×10^{-3}. Likewise, user 21 has a coherent BER of 7.9×10^{-3} and a non-coherent BER of 14.4×10^{-3}. Such losses could be significant for many systems. The simple averaging procedure described in the previous section, then, will subject these affected users to noticeably higher BER for the full duration of the training sequence.

This section offers an alternative to the multi-frame estimate average of Section 5.2.1. The **recursive phase estimate** is proposed to alleviate the long exposure of affected users to higher BER's by using the information from the estimate made at one frame to help in making the estimate at the next frame. This method significantly reduces the BER for the affected users after only a small number frames.

For some MA systems designed to combat fading channels or bursty error channels, a succession of 36 BER-degraded frames will not cause a problem since the system is prepared for much worse.[11] In such a case there is no need for the recursive phase

[11] One way to combat the increase in BER over T_T frames is for each user to choose an error correction code to accommodate the worst case BER. Another option is to choose a code that can accommodate the worse case "fade" caused by the dropping in of a new user. Another coding related option is to use interleaving so that the lower quality one-shot estimates do not occur in succession

estimate. On the other hand, for systems with relatively fading-free and bursty-free channels, the incorporation of long codes, interleaving, or other coding related remedies might be wasteful;[12] in such cases it is beneficial to use the recursive phase estimate so that the BER can be restored in as few frames as possible.

The **recursive phase estimation procedure** is motivated by the behavior of the one-shot partially coherent joint weight/phase estimator. Recall from the derivation of the one-shot PC estimator in Section 4.4 that this estimator assumes a value for the phase of the uncertain-phase user. Denoting the assumed phase as $\phi_{assumed}$ and the error in this phase as ϕ_e allows us to write

$$\phi_e = \phi_{assumed} - \phi_{true}. \tag{5.32}$$

Also recall that the probability distribution on the assumed phase error is the familiar exponential cosine PDF

$$p_\phi(\phi_e) = \frac{e^{\alpha \cos(\phi_e)}}{2\pi I_o(\alpha)}, \tag{5.33}$$

where the value of the accuracy parameter, α, corresponds to a value of $\sigma(\alpha)$, the standard deviation of the error in the assumed phase. The PC estimator having partial knowledge characterized by Equations (5.32) and (5.33) produces a phase estimate, $\hat{\phi}$. The error in this estimate has a standard deviation that is less than the the error in the assumed phase.

Since, in this chapter, we are interested in taking advantage of several successive symbol frames, consider the PC estimator at end of the i^{th} frame. The assumed phase at the end of frame i will be denoted by $\phi_{assumed}(i)$; the error in this assumed phase will be denoted by $\phi_e(i)$. The partial information at the end of symbol frame i is

in the de-interleaved bit stream. For tutorial treatment of these techniques, see any text on digital communications, such as [12].

[12]Recall that a user is only affected when a nearby ancestor or descendant spot on the tree is taken by a new unknown-phase user. For many MA systems, the drop in rate for related users would be a small fraction of the length of time a user accesses the system.

characterized by

$$\phi_{assumed}(i) = \phi_{true} + \phi_e(i), \tag{5.34}$$

and

$$p_\phi(\phi_e(i)) = \frac{e^{\alpha_i \cos(\phi_e(i))}}{2\pi I_o(\alpha_i)}, \tag{5.35}$$

where the value of α_i, the accuracy parameter at the end of frame i, corresponds to a value of $\sigma_i = \sigma(\alpha_i)$, the standard deviation of the error in the assumed phase at the end of frame i.

At frame $i+1$, $\mathbf{r}(i+1)$ is received and the PC estimator creates a phase estimate that will be denoted by

$$\hat{\phi}(i+1|\sigma_i).$$

The above notation indicates that the partial information used in creating this estimate is characterized by the standard deviation, σ_i. The error in this estimate is

$$\phi_e(i+1|\sigma_i) = \hat{\phi}(i+1|\sigma_i) - \phi_{true} \tag{5.36}$$

and its PDF is

$$p_\phi(\phi_e(i+1|\sigma_i)) = \frac{e^{\alpha_{i+1} \cos(\phi_e(i+1|\sigma_i))}}{2\pi I_o(\alpha_{i+1})}. \tag{5.37}$$

The value for α_{i+1} is the accuracy parameter that corresponds to the standard deviation, σ_{i+1}, for this estimate.

The value of the standard deviation of the one-shot PC phase estimate can be determined from simulations. Specifically, the value of the standard deviation, σ_{i+1}, of the estimate $\hat{\phi}(i+1|\sigma_i)$, that results from the PC estimator having prior information characterized by the standard deviation, σ_i, can be found from simulations for each E_b/N_o.

Table 5.5 shows several $\{\sigma_i, \sigma_{i+1}\}$ pairs for $E_b/N_o = 5.528$ dB. The table has three columns. The left column shows the standard deviation in the assumed knowledge of the user's phase at the end of symbol frame i, denoted by σ_i. The center column

Table 5.5: Simulation results at $E_b/N_o = 5.528$ dB for the 21 unit energy minimum distance set for which user 17 has an uncertain phase. This table shown the standard deviation, σ_{i+1}, of the phase estimate given by the PC one-shot estimator having partial information characterize by the exponential cosine with accuracy parameter α_i (or standard deviation σ_i).

σ_i	$\alpha(\sigma_i)$	σ_{i+1}
⋮	⋮	⋮
7.79°	55.63	4.00°
6.75°	74.80	3.52°
6.04°	93.01	3.21°
⋮	⋮	⋮
4.00°	205.28	2.29°
3.52°	264.29	2.03°
3.21°	319.51	1.85°
⋮	⋮	⋮

shows the accuracy parameter, $\alpha(\sigma_i)$, corresponding to σ_i. The right column shows the value of σ_{i+1}, the standard deviation of the estimate that would be given by the PC estimator having partial knowledge characterized by the parameters in the left and center columns.

The values in Table 5.5 were found from simulations using the 21 unit energy minimum distance set for which all but one user's phase is known. To find σ_{i+1} in each row of Table 5.5 simulations were run in which user 17's phase was drawn randomly from the distribution shown in Equation (5.35) where α_i is taken from the center column of the table. For example see the first row in Table 5.5. At 5.528 dB, the PC one-shot estimator having partial knowledge characterized by $\sigma_i = 7.79°$, or $\alpha_i = 55.63$, will give a one-shot phase estimate, $\hat{\phi}(i+1|\sigma_i)$, that has a standard deviation of $\sigma_{i+1} = 4.00°$.

We may, of course, use the estimate, $\hat{\phi}(i+1|\sigma_i)$, to improve the partial information

at the receiver. To prepare for the next frame we set

$$\phi_{assumed}(i+1) = \hat{\phi}(i+1|\sigma_i),$$

where this new partial information is characterized by the standard deviation $\sigma_{i+1} = 4.00°$. By looking for the row in Table 5.5 that is indexed (in the left most column) by the standard deviation of 4.00°, we find that the accuracy parameter must be set to $\alpha_{i+1} = 205.28$. The PC estimator, at the next frame, uses this updated partial information to give an estimate,

$$\hat{\phi}(i+2|\sigma_{i+1})$$

with a corresponding value of $\sigma_{i+2} = 2.29°$.

The full recursive estimation procedure is illustrated by the flow chart shown in Figure 5-2. Begin with the system operating with all user phases known. In this case, as shown in block 1 of the flow chart, the coherent joint detector is used. At the next symbol frame the receiver would check to see if a new user has entered the system, as done in block 2. If there is no new user, the coherent detector is used. If there is a new user, the procedure progresses to block 3.

The new user begins transmitting known training weights. All other users continue to transmit as usual. The receiver uses the optimal one-shot non-coherent joint weight/phase estimator for five successive symbol frames. The five independently calculated phase estimates for the new user are collected, $\{\hat{\phi}(1), \hat{\phi}(2), \cdots, \hat{\phi}(5)\}$. The procedure progresses to block 4.

From the five one-shot phase estimates, a **seed estimate** is calculated as the robust 5-frame estimate average

$$\hat{\phi}_0 = \hat{\phi}(5 \setminus 2) = \frac{1}{(5 - T_d)} \sum_{i \in \Theta} \hat{\phi}(i), \tag{5.38}$$

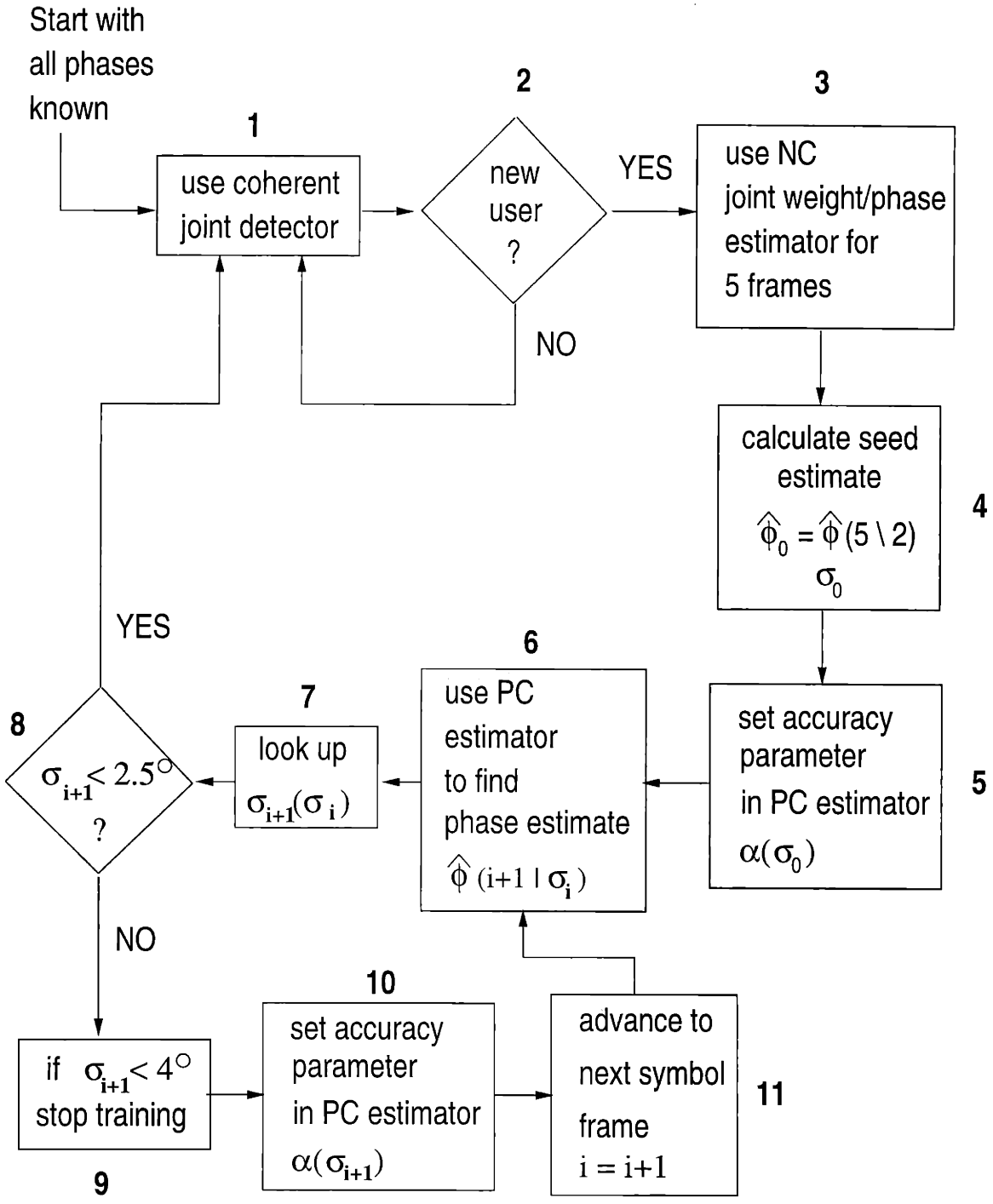


Figure 5-2: Flow chart illustrating the recursive phase estimation procedure.

where the actual number of estimates that are discarded is denoted by T_d which is, at most, 2, and Θ is the set of T_d indices denoting the non-outlier estimates. From the previous section we know that at $E_b/N_o = 5.528$ dB, $p = 0.053$, and the probability that this robust 5-frame estimate contains no outliers is $P_g(5 \setminus 2) = 0.9986$. The standard deviation of this seed estimate is calculated to be

$$\sigma_0 = \sigma_{ave}(5 - T_d) = \sigma_{\hat{\phi}(i)}/\sqrt{5 - T_d}. \tag{5.39}$$

The procedure progresses to block 5.

The seed estimate is incorporated into the unknown-phase user's signature vector, as assumed by the receiver.

$$\phi_{assumed}(0) = \hat{\phi}_0 \tag{5.40}$$

Now, the user has some partial knowledge for the user's phase. The value of α in the estimator must be set to reflect this partial knowledge for the next frame. The standard deviation, σ_0, represents the uncertainty on the assumed value of the user's phase. From this, the accuracy parameter

$$\alpha_0 = \alpha(\sigma_0)$$

is determined.[13] The procedure progresses to block 6.

Now, the receiver uses the optimal one-shot partially coherent joint weight/phase estimator on symbol frame 1. The phase estimate

$$\hat{\phi}(1|\sigma_0) \tag{5.41}$$

is produced. The procedure progresses to block 7 where the standard deviation, σ_1, of the estimate, $\hat{\phi}(1|\sigma_0)$ is found from a table lookup. Recall that a table of triplets,

[13]Recall that we may use Viterbi's formula, shown in Equation (5.12), to create a table of $\{\sigma, \alpha(\sigma)\}$ pairs.

$\{\sigma_i, \alpha_i, \sigma_{i+1}\}$, such as Table 5.5, may be constructed for each value of $\mathbf{E}_b/N_o$ and may be stored at the receiver. The procedure progresses to block 8.

The receiver must check to see if the estimate $\hat{\phi}(1|\sigma_0)$ is of sufficient quality to use as if it were true by the coherent detector on subsequent symbol frames. Specifically, if the standard deviation, σ_1, of this estimate is no more than 2.5°, the new user may send information weights in subsequent symbol frames and the receiver uses the coherent joint detector; the procedure progresses back to block 1. If, on the other hand, the standard deviation, σ_1, is greater than 2.5°, the recursive estimation procedure is not finished and we progress to block 9.

If the standard deviation of the phase estimate is 4° or less, the unknown-phase user may stop sending known training weights and may begin to send information weights in subsequent frames. As was found from simulations in Section 4.7.1, the BER given by the PC joint weight/phase detector with an 8° uncertainty in one user's phase (all users send information weights) is virtually identical to the BER given by the coherent detector for which all user phases are known exactly. A standard deviation of 4° on phase uncertainty translates to a phase uncertainty of at most 8° with 0.95 probability. The procedure progresses to block 10.

The receiver incorporates the new partial information from the most recent estimate by setting

$$\phi_{assumed}(1) = \hat{\phi}(1|\sigma_0).$$

The new accuracy parameter is determined from the standard deviation of the most recent phase estimate,

$$\alpha_1 = \alpha(\sigma_1),$$

where the deterministic relation in Equation (5.12) is used. The procedure progresses to block 11.

The receiver advances to the next symbol frame. The procedure progresses to block 6.

The receiver uses the optimal one-shot partially coherent joint weight/phase estimator on symbol frame 2. The phase estimate

$$\hat{\phi}(2|\sigma_1) \tag{5.42}$$

is produced. The procedure progresses to block 7 where the standard deviation, σ_2, of the estimate, $\hat{\phi}(2|\sigma_1)$ is found from a table lookup. The procedure progresses to block 8. The recursion of blocks 9, 10, 11, 6, and 7 continues until a recursive phase estimate having a standard deviation of less than 2.5° is found.

To illustrate the quality of successive phase estimates found by the recursive estimation procedure, a specific example using simulated data is explained. The 21 user unit energy minimum distance set was used. User 17's phase is completely unknown at the receiver. All other user phases are correctly known at the receiver. The signal to noise ratio for this simulation is 5.528 dB.

The first step is to collected five one-shot optimal non-coherent phase estimates. Up to two outlier estimates may be discarded. If there are two outliers, the seed estimate will have a standard deviation, $\sigma_0 = 7.79°$. If there is one outlier, the seed estimate will have a standard deviation, $\sigma_0 = 6.75°$. If there are no outliers, the seed estimate will have a standard deviation, $\sigma_0 = 6.04°$. Since there are three possible outcomes for the seed estimate, there are three possible outcomes of the recursive estimation procedure. Table 5.6 shows each possible outcome as a progression of standard deviation values.

The first row of Table 5.6 shows the simulation results if there were two outlier estimates, i.e. if $T_d = 2$. The recursive phase estimation procedure follows. User 17 joins the system. The receiver has no prior knowledge of user 17's phase. User 17 begins sending known training weights. The receiver uses the optimal one-shot non-coherent joint weight/phase estimator for five successive frames and five independent one-shot phase estimates are collected. Two of these estimates are determined to be

Table 5.6: An example of the recursive estimation procedure for a simulation at 5.525 dB.

T_d	σ_0	σ_1	σ_2	total number	number of training
2	7.79°	4.00°	2.29°	$5+2=7$	6
1	6.75°	3.52°	2.03°	$5+2=7$	6
0	6.04°	3.21°	1.85°	$5+2=7$	6

outliers; the remaining three are averaged to make the seed estimate, $\hat{\phi}_0$. This seed estimate has a standard deviation, $\sigma_0 = 7.79°$. The receiver sets $\phi_{assumed}(0) = \hat{\phi}_0$. This partial information on user 17's phase is characterized by a standard deviation of 7.79°. This standard deviation corresponds to an accuracy parameter of $\alpha_0 = 55.63$. The receiver uses the optimal PC joint weight/phase estimator on the next symbol frame and produces the phase estimate $\hat{\phi}(1|7.79°)$. This estimate is determined from Table 5.5 to have a standard deviation, $\sigma_1 = 4.00°$. This standard deviation is greater than 2.5°, so the recursion must continue. This standard deviation is small enough, however, to allow user 17 to stop sending training weights and to start sending information weights at the next frame. The receiver prepares for the next frame by updating its partial information: $\phi_{assumed}(1) = \hat{\phi}(1|7.79°)$ and $\alpha_1 = \alpha(4.00°) = 205.28$. The optimal PC joint weight/phase estimator is used (knowing that user 17 sends an information weight) to produce the phase estimate $\hat{\phi}(2|4.00°)$. From Table 5.5 we find that the standard deviation of this estimate is $\sigma_2 = 2.29°$. This standard deviation is less than 2.5° so the recursive phase estimation procedure is terminated and $\hat{\phi}(2|4.00°)$ is taken to be true. The coherent joint detector is used for subsequent symbol frames.

Although the other two possible progressions of the recursive estimation procedure will result in slightly different values for σ_0, σ_1 and σ_2, we see from Table 5.5 that in all three cases, a total of 7 symbol frames were used. In addition, only 5 of those frames required application of the one-shot non-coherent joint weight/phase estimator. Recall that for $E_b/N_o = 5.528$ dB, Section 5.2.1 found that the robust phase

estimate average having a standard deviation of 2.5° would require the application of the one-shot non-coherent joint weight/phase estimator for 36 successive frames. If the recursive phase estimator is used, affected users must withstand a degraded BER for only 7 frames. Moreover, the highest degradation is during the first 5 frames during which the one-shot non-coherent joint weight/phase estimator is used.

We are interested in the progression of BER for the most affected users during the 7 frames of the recursive estimation procedure for the example at $E_b/N_o = 5.528$ dB. Since there are three possible outcomes for the seed estimate, the BER progression is determined to be the weighted average of the three. Specifically, the BER at a frame i, given that T_d outliers were found, is denoted by $BER(i|T_d)$. The weighted averaged BER at frame i is given by

$$BER(i) = (P_2\ BER(i|2) + P_1\ BER(i|1) + P_0\ BER(i|0))/(P_2 + P_1 + P_0), \quad (5.43)$$

where the probability of there being two outliers in the collection of five one-shot non-coherent phase estimates is

$$P_2 = \begin{pmatrix} 5 \\ 2 \end{pmatrix} p^2(1-p)^3,$$

the probability of there being 1 outlier is

$$P_1 = 5p(1-p)^4,$$

and probability of there being no outliers is

$$P_0 = (1-p)^5.$$

Recall that p is the probability that a one-shot non-coherent phase estimate is an outlier. At $E_b/N_o = 5.528$ dB $p = 0.053$.

The weighted average $BER(i)$ is shown for user 1 in Figure 5-3. A total of eight symbol frames are shown. The simulation was done for the 21 user unit energy minimum distance set with an $E_b/N_o = 5.528$ dB. The first symbol frame shows the BER for the case in which all user phases are known. After the first symbol frame, user 17 drops out to be replaced by a new user. At the second symbol frame, the phase of the new user 17 is unknown at the receiver. For symbol frames 2 through 6 user 17 sends a known training weight and the receiver uses the optimal one-shot non-coherent joint weight/phase estimator. Notice that the BER during these frames has nearly doubled. At the end of the 6^{th} frame the seed estimate is calculated. This partial information is set in the estimator and the one-shot PC joint weight/phase estimator is used on frame 7. Notice that in frame 7 the BER for user 1 is nearly restored to its coherent value (the BER experienced in frame 1). In frame 8 user 17 sends an information weight and user 1's BER is restored to its coherent value. The standard deviation of the BER points found in this simulation range from 4.3% to 5.9% of the BER.

Figure 5-4 shows the BER for user 21 for same progression of symbol frames as described above for user 1. The standard deviation of the BER points found in this simulation range from 3.2% to 4.2% of the BER. Notice that the same behavior found for user 1 exists for user 21. Specifically, after five frames of a nearly doubled BER during the use of the one-shot non-coherent joint weight/phase estimator, user 21's BER is nearly restored by the 6^{th} frame of the recursive estimation procedure.

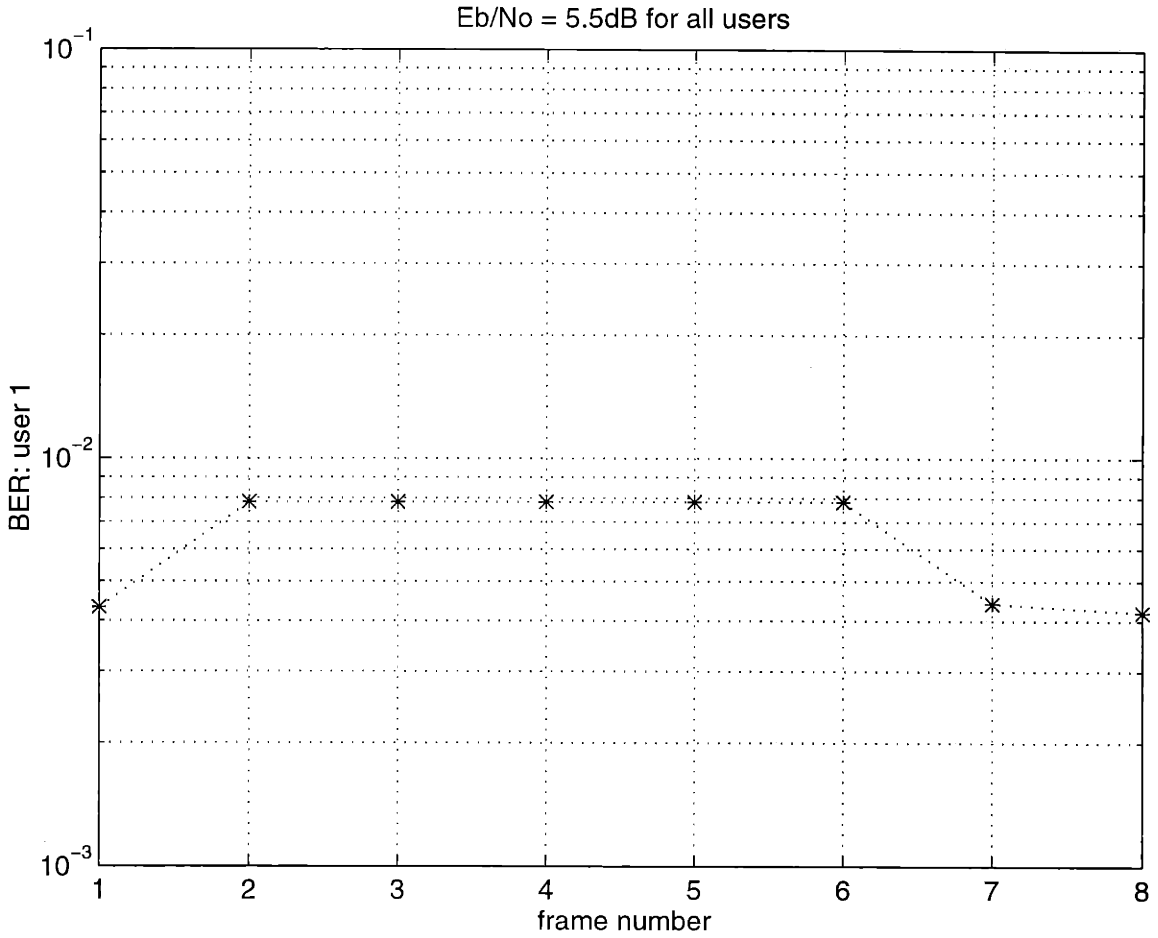


Figure 5-3: User 1's BER progression during the recursive phase estimation procedure on user 17. Error bars are not shown. The standard deviation of the BER points ranges from 4.3% to 5.9%.

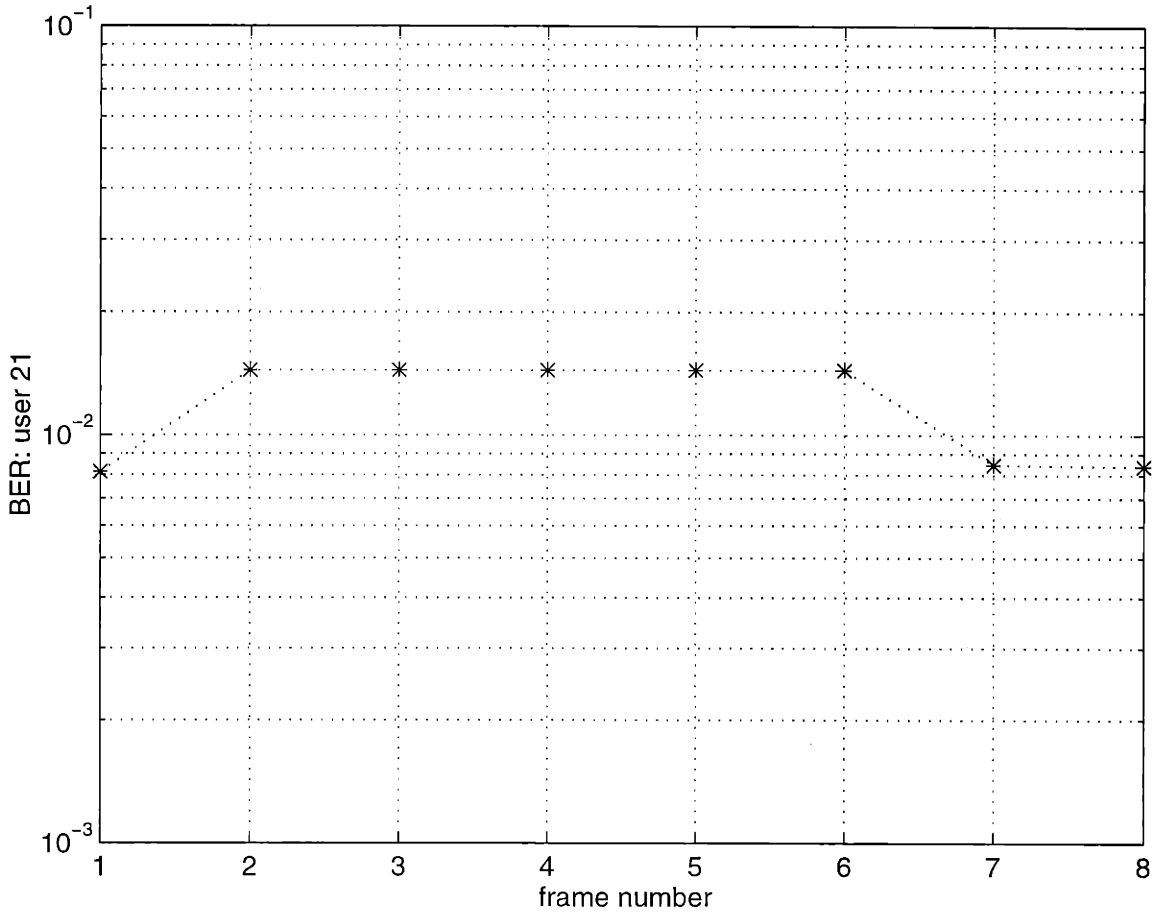


Figure 5-4: User 21's BER progression during the recursive phase estimation procedure on user 17. Error bars are not shown. The standard deviation of the BER points ranges from 3.2% to 4.2%.

Chapter 6

Conclusion and Future Directions

This thesis examines the problem of uncoded multiple access (MA) joint detection for the case in which user signatures are not orthogonal.[1] Specifically, if a system is to offer service to more users than is possible with FDMA and TDMA systems, users will not be orthogonal ([18]). The primary obstacle in a non-orthogonal MA system is the complexity of detection; in general, the optimal detector has a complexity which is exponential in the number of users ([23]).

The approach taken in this thesis is to recognize that the problem of designing a reliable and realistic MA communications system allows for the joint design of users' signatures and detection procedures. In other words, exercising the control we have over the characteristics (or structure) of the set of user signature waveforms will make the job of finding ways for low complexity detection easier.

In this thesis, a tree-structure is used to give an advantage to the detection algorithm. Specifically, the user waveforms must have tree-structured interference. The tree structure is used as a signal design guideline that is expected to be easily achieved for many scenarios in which orthogonal signaling is possible.

For strictly tree-structured signals the one-shot optimal tree detector is derived

[1]The signaling methods assumed in this thesis are M-ary pulse amplitude modulation, M-ary phase shift keying, or a hybrid between the two.

for the special ideal case of the receiver having perfect knowledge of each of the user waveforms, including each user's phase.[2] Briefly, each user is assigned to a node on the tree; the detection algorithm takes advantage of the tree structure by sweeping through the tree from bottom to top. At each node of the tree, an intermediate weight decision is made conditioned on the weight decisions of all the users that sit at ancestor nodes to the current node. The top decision finalizes all the intermediate decisions made below it.

The tree detector gives the *optimal* estimate with an extremely *low* computational complexity. An upper bound on the complexity was found to be, for typical cases of interest, a low-order-polynomial in the number of users, e.g. $O(K^p)$, K the number of users and p small. This is an enormous savings in computations over the $O(M^K)$ computations needed if the signatures did not exhibit any structure.[3] Indeed, with a typical K of 85 users employing binary signaling ($M = 2$), an exhaustive joint detector would require computations on the order of 10^{25}, while the optimal tree detector would require only 585 computations.

Since the tree detector is optimal for *any* set of tree-structured signatures, its use with the minimum distance sets proposed by Ross and Taylor in [17] allows, in principle, for an over-saturated MA system that has comparable performance *and* computational complexity as the corresponding orthogonal MA system supporting fewer users. The detection result of this thesis has lifted a major computational obstacle, thus, opening up the area of over-saturated communications for more research. Some ideas for future work are given in Section 6.1.

Numerous performance questions were explored via simulation. All simulations in this thesis were done with the Ross and Taylor minimum distance sets for which each user employs binary phase shift keying. Left for future work is to test the Ross/Taylor

[2]The term one-shot means that the detector uses only one symbol frame from which to calculate its estimate. In contrast, a sequence detector collects several symbol frames in order to do joint detection on all weights in the sequence. Sequence detection for the purpose of phase estimation was also addressed in this thesis and is summarized shortly.

[3]Here, it is assumed that all users employ M-ary signaling.

sets with other popular signaling schemes such as quadrature phase shift keying. Each user's signature signal, prior to running simulations was arbitrarily assigned a phase. This was done to reflect the lack of phase control in an actual system. Results from simulations done for this thesis, then, reflect a typical set of minimum distance users. Left for future work is the exploration of worst case phase assignment on the minimum distance sets.

In an actual system, strict orthogonality is often not possible; for many MA systems, *near* orthogonality is the best that can be achieved in practice. Since strict tree structure is expected to be possible only for systems capable of keeping users strictly orthogonal, analysis of the tree joint detector was done with *nearly tree-structured* users. Namely, each user's transmission was allowed to interfere with it's neighboring user's transmission, much like FDMA users "leak" into adjacent channels. A simulation with 21 unit energy users in 16 dimensions found that a -40 dB adjacent channel leak by each user goes virtually unnoticed, a -30 dB leak causes only negligible performance loss, and a -20 dB leak results in a 1.25 dB performance loss. These results mirror those for an FDMA system experiencing the same degree of leakage ([13]). In other words, the tree joint detector which is optimal if users are strictly tree-structured experiences the same performance loss in the face of adjacent channel leaks as do current systems using a bank of matched filters followed by a slicer. Recall that for the strictly orthogonal system, a bank of matched filters is optimal ([7]).

Joint detection, in general, requires comparison of the actual received signal with the set of possible received signals. The set of possible received signals is constructed by the detector using replicas of the actual received signature signals (including phases) that are stored at the receiver. Hence, each time a new user joins the MA system, it's phase must be estimated so that it can be used for the detection of subsequent symbols. Any estimate will, of course, have some error variance. Ultimately, then, the tree joint detector must be able to hold up under small degrees of phase mismatch between the actual phase of a user and the phase as known at the receiver.

Again, a 21 user in 16 dimension tree-structured MA system was simulated. For the case of one user having a phase mismatch, the majority of other users in the system experienced no loss. A few users (the users assigned to parent and child nodes of the mismatched-phase user) experienced the following losses: negligible loss (0.0 dB to 0.15 dB) for a 10° mismatch; small loss (0.15 dB to 0.50 dB) for a mismatch of 15°; and notable loss (0.50 dB to 1.00 dB) for a mismatch of 20°, which might prove to be tolerable for some systems if, perhaps, the 20° mismatch persists only for a few symbols.

The more common scenario of having small mismatches in *all* users phases was also simulated. A 0.75 dB average performance loss was found for a 10° mismatch in all user phases in a 21 user set, and a mere 0.13 dB loss was found for a mismatch of 5°. For near lossless performance, then, the tree joint detector requires approximately ±5° certainty in the knowledge of each user's phase.

It is important, then, for a multiple access joint detection scheme to incorporate low complexity phase estimation. The estimation of a user's phase, therefore, was incorporated into the tree detection procedure. The optimal joint weight/phase tree-structured estimation algorithm was derived for one user having an unknown or a partially known phase. Via simulations the one-shot optimal non-coherent joint weight phase estimator was examined for the case of one user having a completely unknown phase; this users sends a training weight. The standard deviation of the phase estimate was found to be between 34° and 48°, depending on the signal to noise ratio. It is interesting to note, however, that the probability density function (PDF) for the error in the phase estimate (approximated by simulation results) closely resembles the exponential cosine function which is the PDF for the error in the phase locked loop phase estimate for the *single* user system ([24]). This PDF, for the conditions of the simulations in this thesis, resembles the Gaussian PDF. The PDF for the MA system exhibited, in addition to a main lobe centered at zero, two small lobes centered at ±175°, respectively. The outlier lobes, together, contained from 4% to

8.2% of the probability, depending on SNR. A more intuitive way of measuring the quality of the one-shot phase estimate given by the non-coherent joint weight/phase estimator is to assign an "effective standard deviation" which quantifies the main lobe containing 91.8% to 96% of the probability. Hence, the one-shot phase estimate was found to have an effective standard deviation ranging from 10° to 20° where an occasional outlier error will occur with a 0.04 to 0.08 probability.

In addition to returning a phase estimate, the non-coherent joint weight/phase estimator finds an estimate for each of the users' weights. Simulation results for the case in which one user's phase is unknown showed a 0.1 dB (2.5%) average performance loss in BER relative to the case in which all users' phases are known accurately. The greatest loss is 0.75 dB (19%) and is experienced by the parent and a child to the unknown-phase user. The other users experience virtually no loss.

The optimal joint weight/phase estimator was also simulated for the case of the receiver having *partial* phase knowledge of one of the user's phases while having perfect knowledge for the rest of the users' phases. Having knowledge of a single user's phase to within ±18.7° allowed for much better phase estimates. Moreover, the approximated PDF for the error on the phase estimate given partial knowledge had no outlier lobes. The actual standard deviations on the phase estimates ranged from 7.7° to 9°, depending on the SNR. The quality of the phase estimates (given a prior uncertainty of ±18.7°) for SNR's ranging from 3 dB to 7.2 dB did not depend on whether or not a training weight was used. For example, at an SNR of 5.528 dB, the standard deviation in phase estimate error was found to be 8.2° (training) and 8.3° (no training). The simulation was repeated for a prior uncertainty of ±8.14°. The phase estimate for this case had a standard deviation of 5°. Recall that this approaches the accuracy needed before the tree joint detector can use the phase estimate as if it were true for the detection of subsequent weights.[4]

[4]A required accuracy of 5° means that the mismatch between a user's actual phase and that phase as known at the receiver must be no greater than 5°. A 5° *standard deviation* means that with 0.68 probability, the estimate error will be no greater than 5°. Ultimately, we need an estimate with a

The average loss in BER for the partially coherent cases was found to range from 0 dB (no loss) to 0.25 dB (6% loss). No losses were found when a training weight was used, even when the prior phase uncertainty was as high as $\pm 18.7°$. The no-training case for the $\pm 18.7°$ uncertainty gave a greatest loss of 0.5 dB (12%), experienced by a child of the unknown-phase user. Also for this same case, the parent of the unknown-phase user experienced a 0.25 dB (6%) loss; the other users experienced no loss for the partially coherent case.

The tree-structured optimal joint weight/phase estimator has a complexity that can range from very low order polynomial in the number of users to exponential in the number of users, depending upon the position in the tree of the uncertain-phase user. For many scenarios, the complexity remains low. Returning to the example of 85 binary users in 64 dimensions, we find that if the unknown-phase user were at the bottom of the tree, there is no increase in complexity relative to the 585 operations needed for the case of all users' phases known. If the unknown-phase user were one level (two levels) up from the bottom, the joint weight/phase estimator requires 673 (4,194,700) total operations. If this unknown-phase user is at the top of the tree, however, the complexity balloons to $O(10^{25})$.

One sub-optimal low complexity option for one-shot joint weight/phase estimation was proposed. The sub-optimal version of the weight/phase joint detector simply assumes that the user's phase is from a *discrete* set of possible phases. The complexity of this sub-optimal detector remains reasonable even for relatively fine discretization of the phase. For example, even for a set of 24 equally spaced phases for which there is a 15° separation, the complexity remains low. Again, the example of 85 users in 64 dimensions offers an illustration. As noted above, the worst case for this scenario would be to put the unknown-phase user at the top of the tree. The "assumed-discrete" joint weight/phase estimator brings the optimal estimators complexity of $O(10^{25})$ down to a manageable 31,592 total operations needed to find the phase

standard deviation of 2.5° so that with 0.95 probability this estimate will be within $2 \times 2.5° = 5°$.

estimate for the unknown-phase user and a weight estimate for each of the other 84 users. Note that with this 15° separation between discrete phases, the actual phase is guaranteed to be within 7.5° of one of the discrete phase points. It is important to note that the phase estimate given by the assumed-discrete weight/phase estimator will not, in general, be one of the discrete phase points. This estimator uses the set of discrete phases to assist in finding the weight estimates for the known-phase users. Once these are found, the phase estimation procedure is identical to the single user phase estimator. Since the discretization can be changed to bring the discrete phase separation down, this estimator is anticipated to be capable of giving very near optimal performance.[5]

Recall that an accuracy is needed within ±5° before the tree joint detector can use a phase estimate as if it were true. This means that we need a phase estimate with a standard deviation of 2.5°.[6] The one-shot non-coherent phase estimate is incapable of giving such an estimate. Three procedures for using a sequence of transmissions was proposed. First, the optimal joint weight/phase sequence estimator was derived and was found to have low complexity only for a small number of special cases. Second, a multi-frame estimate average was proposed that requires virtually no additional computation above the one-shot optimal joint weight/phase estimation. Briefly, T successive, independently calculated, one-shot phase estimates are made. Any estimates believed to be outliers are discarded. The remaining estimates are averaged to create the multi-frame estimate average. For this phase estimate to achieve a 2.5° standard deviation in error (at an SNR of 5.528 dB) it would need at least 36 one-shot estimates. That is, at least 36 transmission frames will go by during which the unknown phase user sends a known weight, also during which a parent and child users will experience BER losses of up to 0.75 dB. Third, a recursive refinement phase estimator was proposed. The recursive estimator collects only five successive

[5]In other words, if there is a need to improve performance, further reduction of the discrete phase separation may be done, at a cost of a relatively low increase in complexity.

[6]With 0.95 probability the estimate will be in error of less than 5°.

one-shot optimal phase estimates, discards up to two possible outliers, and averages the remaining phase estimates to create an estimate with a standard deviation (at an SNR of 5.528 dB) ranging from 6.04° to 7.79°.[7] The recursive estimator continues by noting that it now has partial information on the unknown-user's phase. The partially coherent joint weight phase estimator is used for the next few frames, after each, the accuracy is adjusted to reflect the improvement in the phase estimate. To achieve a phase estimate having standard deviation of 2.5° at 5.528 dB, the recursive phase estimator needs five successive non-coherent phase estimates and two partially coherent phase estimates. That is, seven transmission frames are needed to create the phase estimate. Moreover, only five of those frames employ the non-coherent joint weight/phase estimator; this means that a parent and child users will experience BER losses of 0.75 dB for only five frames. In the last two frames the loss in BER for these affected users is negligible.

6.1 Ideas for Future Work

This thesis, along with the work of Ross and Taylor in [17] just begins to study over-saturated MA communications. This section touches on some ideas for future work in this area. Specifically, comments can be separated into three categories: 1) the study of tree-structured estimation algorithm issues, 2) the study of tree-structured signal set design, and 3) options other than tree-structured approaches.

First, estimator issues are discussed. This thesis examined the problem of estimating one of the user's phases while all other users' phases were accurately known by the receiver. In an actual system, the receiver will have obtained its information on each user's phase from some estimation procedure, hence, the knowledge of each user's phase will be in error, even if by only a small amount. When all users have a small phase mismatch, the estimate of the phase for a new user entering the system

[7]The exact value of the standard deviation depends upon the number of outliers that were found.

will be affected. From the many examples of this thesis it is expected that the phase mismatches on users which are ancestors and descendants to the user having the unknown phase will contribute most to the error in estimating the unknown phase. Simulations exploring this scenario are planned for future work.

Here, it should be pointed out that the optimal detectors proposed in this thesis were derived to minimize the probability of making an error for one or more of the users. That is, an error in only one user's weight is considered to have the same cost as an error in many of the users' weights. Alternatively, it would be interesting to determine if the same type of algorithmic simplification seen in this thesis is found for other optimal detectors that minimize other criteria. Some meaningful minimization criteria follow: 1) minimize the probability of making an error in *each* user's weight, 2) minimize the *average* probability of making an error in a user's weight, 3) minimize the highest probability of error experience by a user in the tree.

It is important to study other sources of user signature mismatch, no matter how slight. Recall that any joint detection scheme relies on the subtraction of a hypothesized received signal from the actual received signal. The accumulation of different signal parameter mismatches, even if each mismatch is considered to be harmless in orthogonal systems, might prove to be detrimental to an MA system employing joint detection if ignored. An understanding the effect of the receiver having imperfect knowledge of user powers and/or symbol timings on phase and weight estimation is needed in order to set goals for power and timing accuracies.

Furthermore, this thesis assumed that once a phase estimate was obtained it remained constant. From the simulations reported in this thesis, small BER degradation resulted from all users having a 5° phase mismatch with the receiver. Recall that a 10° mismatch in every user caused a 0.75 dB average loss. If each user's phase is estimated to within a $\pm 5°$ error, it will be important to keep user phases from drifting more than another 5°. The work in the final two chapters of this thesis offers a foundation for a low complexity extension of phase estimation to phase tracking.

The development of a low complexity procedure to track the phase drift of all users in an over-saturated system is a logical direction for future work.[8]

Second, signal design issues are discussed. Once a structure is shown to offer an advantage for low complexity detection for an MA scenario of interest, user packing according to this structure needs to be studied. Ross and Taylor have begun the work in this area with the introduction of the minimum distance sets for an ideal case of all user powers and signaling schemes known prior to the arrangement of users into a tree-structured set. For this case, Ross and Taylor have found, possibly, the best way of packing in users if the measure is to be minimum distance. Perhaps there are other measures that will work well for the packing of tree-structured users.

In an actual system, however, users drop in and out. A challenging problem, then would be to determine good procedures for the dynamic allocation of users having realistic properties such as rate and power differences. An example illustrates the difficulty of the dynamic allocation problem. If an initial group of users were to be assigned according to the static method of Ross and Taylor, problems will arise once a user drops out since another user will, inevitably, wish to join the system. The Ross/Taylor set has been tightly packed according to the specifications of the original users (including the user that just dropped out). This means that if the new, prospective user, is incapable of exactly replicating the signature (power, frequency, envelope, etc.) of the user that dropped out, allowing the new user to join the system could cause the entire system to fail. Since the received constellation was carefully designed for the original set of users, allowing a user with an arbitrary power to replace one of the original users, for example, will cause an arbitrary alteration of the received constellation, and the minimum distance may be significantly reduced.

One idea for the problem of dynamic allocation of tree-structured users is to design signature sets with the variations among users in mind. It is expected that a good set

[8]Another tracking problem for some systems is that of tracking user power variations, hence, incorporation of the estimation of user powers will expand the utility of the tree joint detector.

for an actual system will not be as tightly packed as for a statically assigned user set. Furthermore, if accurate probability models can be constructed to describe the arrival process of users to a system, where the model would capture the type of user (priority level, range of capable powers, signaling scheme used, duration of access, etc.) the time of day, the inter-arrival times, etc., then a set of tree-structured signatures can be designed to minimize a cost. Perhaps the cost will correspond to the minimization of the number of users that will not be accepted into the system after a first request. As we can see, the problem of dynamic allocation is two-fold: on the one hand we wish to create a signature set that will give the best performance with the optimal detector, on the other hand we must be sure that users are not turned away when the system has empty slots.

Another interesting problem that falls under the heading of signal design is the design of channel or scenario specific tree-structured waveforms. For example, a channel may be plagued by a narrow band jammer. A tree structure, then, cannot be achieved with a static FDMA-based set of waveforms since this is susceptible to jamming. Current systems combat jammers by the spreading of each user over many frequencies. Two possibilities to achieve spreading are either frequency hopping or direct sequence spreading. Is it possible to create a set of tree-structured waveforms based on either of these anti-jam signaling methods? Perhaps a form of hopping can be accommodated on the nodes of the tree. Another scenario occurring in many terrestrial wireless systems is multipath; this problem is partly dealt with through waveform design. The different paths can be more easily recovered if direct sequence spread spectrum waveforms are used. Chirp waveforms are also useful for the recovery of multiple time delayed receptions. Again, the problem of designing waveforms that maintain tree structure in the face of multipath is important if over-saturating with tree structure is to be possible for such systems.

Third, the options of other approaches for over-saturated MA is discussed. Although the tree structure was used throughout this thesis offered promising results,

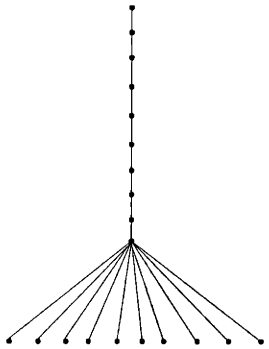


Figure 6-1: Tree corresponding to a signature set for which there are two groups, and orthogonal group of users (corresponding to the bottom of the tree) and a group of remaining users (corresponding to the upper parts of the tree.

it is expected that other structures can be developed to result in low complexity optimal detection algorithms. For some scenarios, any practical problems that might be discovered with tree-structured MA might be solved by the use of another structure. One extension of the tree-structure is to use any set of signatures, denoted by the signature matrix $\mathbf{S}$, that can be separated into two groups of signatures, each denoted by a matrix,

$$\mathbf{S} = [\mathbf{S}_A \, \mathbf{S}_B],$$

where the columns of $\mathbf{S}_A$ are orthogonal. The matrix, $\mathbf{S}_A$, then, corresponds to the largest set of orthogonal users that is possible among the entire set of users. The other users, then, are grouped into set B, having as their signatures the columns of $\mathbf{S}_B$. This would result in tree structure like the one shown in Figure 6-1 where the users corresponding to $\mathbf{S}_A$ would be assigned to the bottom of the tree and the users corresponding to $\mathbf{S}_B$ would be assigned, in any arbitrary order, the nodes along the upper part of the tree. The tree joint detector could, of course, be used at the receiver.

Another, different method for over-saturated MA is that of rate splitting proposed by Rimoldi and Urbanke in [14]. This notion is based on the information theoretic principle that a user can be reliably detected while treating a second user's inter-

ference as noise as long as the first user is transmitting information at a rate at or lower than the channel capacity that would be determined with the incorporation of the second user as noise. The topic of one-shot joint detection with the use of tree structure, addressed in this thesis, and the topic of onion peeling with the use of rate splitting are at opposite ends of the communications theory/information theory spectrum. It would be helpful to the design of an actual over-saturated system to understand tradeoffs between coding/onion peeling and one shot joint detection for typical system scenarios. Although error correction coding is used in most MA systems, the literature on the topic of MA joint detection is focused on one-shot detectors. The explicit incorporation of coding into the tree-joint detector appears to be a difficult and challenging problem, hence, the exploration of bridging the gap between information theoretic approaches and the one-shot detection approaches would be useful.

Appendix A

Analysis of the Minimum Distance Sets

This appendix shows bit error rate curves obtained via simulations. The curves compare the performance of optimal joint detection for orthogonal users, a Ross/Taylor set of 5 users in 4 dimensions, and a Ross/Taylor set of 21 users in 16 dimensions. The two trees, used for these simulations are shown in Figures A-1 and A-2.

In an actual system, each user is assigned a signature waveform envelope and a carrier frequency. The carrier phase is not controllable by the user or by the receiver. To model this phenomenon, prior to the simulations, each user's signature has been multiplied by a unit magnitude complex scalar, arbitrarily chosen.

The average bit error rate curve for a set of orthogonal users, a minimum distance

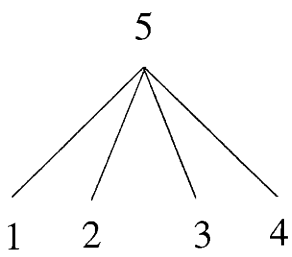


Figure A-1: Correlation tree for a unit energy minimum distance signature set of 5 users in 4 dimensions.

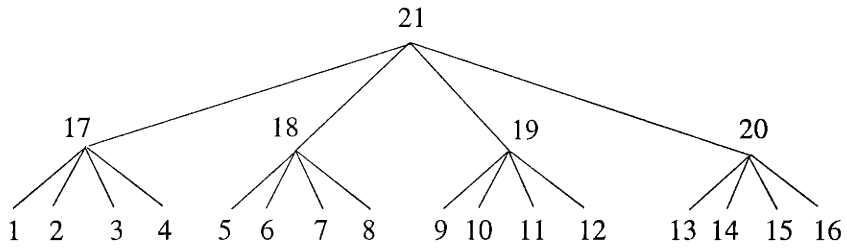


Figure A-2: Correlation tree for a unit energy minimum distance signature set of 21 users in 16 dimensions.

set of 5 user in 4 dimensions, and a minimum distance set of 21 users in 16 dimensions is shown in Figure A-3. The one standard deviation error bars for these curves are not shown in the figure; they are approximately 2.5% of the BER for the case of the orthogonal users and the case of 5 users in 4 dimensions and 1% of the BER for the case of 21 users in 16 dimensions. From the figure we may conclude that more users can be fit into an orthogonal system with a small performance loss of less than 0.5 dB. A 0.5dB performance degradation means that all users in the over-saturated system would need to increase their signal energy to noise energy ratio per bit transmission by 0.5dB (or by 12%) to achieve the same bit error rate of the orthogonal system. For most applications this is considered to be a small loss relative to the 25% and 31% increase in users.

For the set of 5 users in 4 dimensions, the bit error curves are shown for each user in Figure A-4 having error bars between 5% and 6.5%. Notice that the performance of user 5, the user which spans all 4 dimensions, is a bit worse than the performance of the other users. This is explained by the arrangement of possible received points. Briefly, if an error is made, the optimal detector will most likely be confusing one point for its neighboring point. For the minimum distance sets of equal energy users, confusing two neighboring points will result in an error in the estimate of user 5's weight more often that it will for the estimate of the other users' weights.[1]

[1]This behavior can be seen with 3 users in 2 dimensions in which the first two users are orthogonal and of equal energy and the third user spans both dimensions and has an energy a bit smaller than that of the first two users.

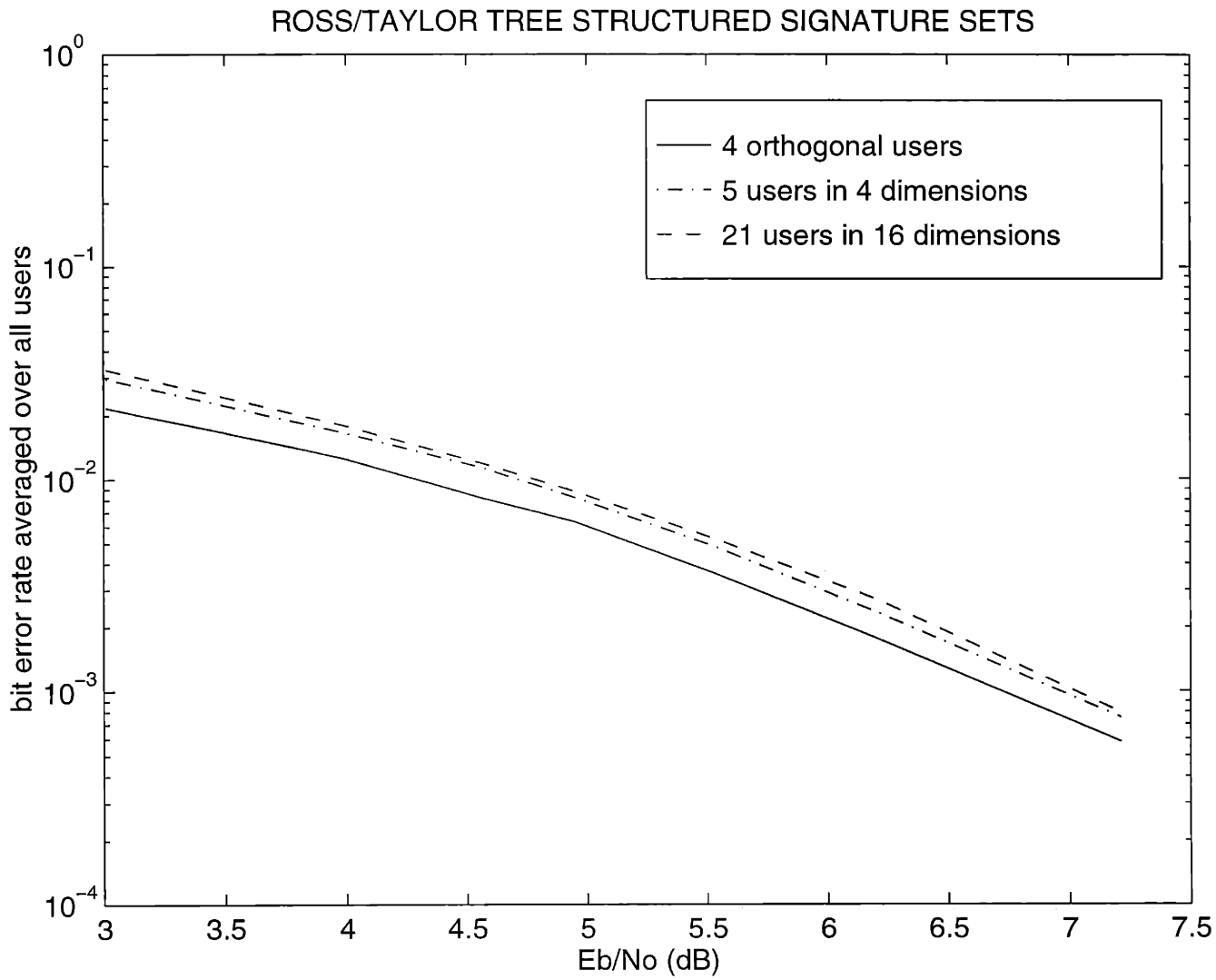


Figure A-3: Average bit error rate for a set of orthogonal users, minimum distance set of 5 user in 4 dimensions, and a minimum distance set of 21 users in 16 dimensions. Error bars are not shown but would be between 1% and 2.5%.

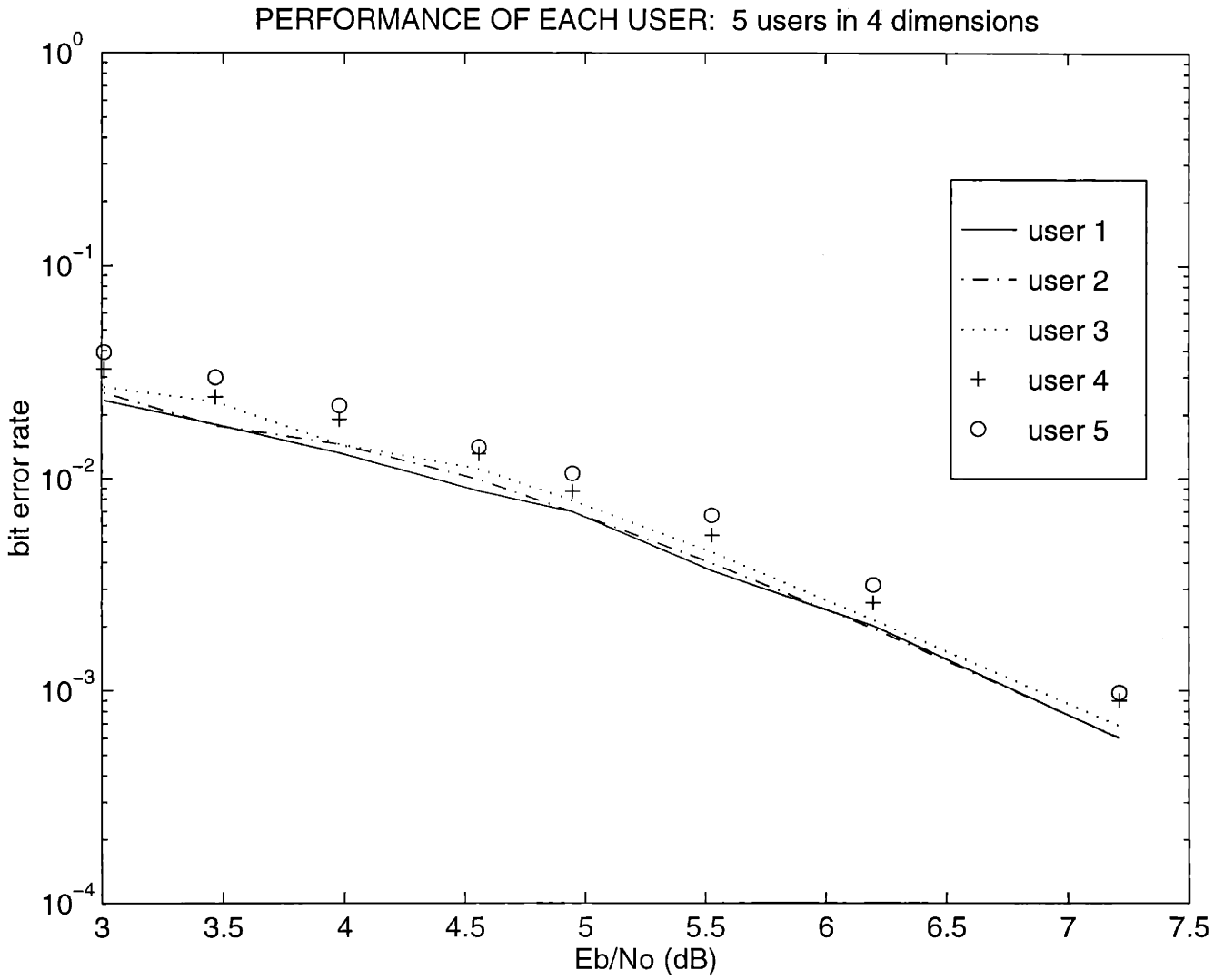


Figure A-4: Fit error rates for each user in a minimum distance set of 5 user in 4 dimensions. Error bars are not shown but would be between 5% and 6.5%.

Appendix B

Limiting Cases of Phase Accuracy

The accuracy of the phase is parameterized by α in Equation (4.34). We examine the case in which α approaches zero. As the accuracy in the phase estimate degrades, we expect the partially coherent weight/phase estimator shown in Equations (4.41) and (4.42) to reduce to the non-coherent weight/phase estimator shown in Equations (4.18) and (4.19).

CLAIM

Assuming $b_K = +1$, as was required by the non-coherent joint weight/phase estimator derived in Section 4.1,

$$\arg \min_{\{\mathbf{b}, b_K=1\}} \lim_{\alpha \to 0} \Omega_3(\mathbf{b}, b_K = 1|\mathbf{r}) = \arg \min_{\mathbf{b}} F_K(\mathbf{b}|\mathbf{r}) - 2|\mathcal{X}(\mathbf{b}|\mathbf{r})|$$

and

$$\lim_{\alpha \to 0} \tan^{-1} \frac{\Im\{\mathcal{Y}(\hat{\mathbf{b}}, \hat{b}_K = 1|\mathbf{r})\}}{\Re\{\mathcal{Y}(\hat{\mathbf{b}}, \hat{b}_K = 1|\mathbf{r})\} + \alpha \frac{\sigma^2}{2}} = \tan^{-1} \frac{\Im\{\mathcal{X}(\hat{\mathbf{b}}|\mathbf{r})\}}{\Re\{\mathcal{X}(\hat{\mathbf{b}}|\mathbf{r})\}},$$

where $\Omega_3(\mathbf{b}, b_K|\mathbf{r})$ is defined in Equation (4.40) and the PC phase estimate is defined in Equation (4.42). The non-coherent weight and phase estimates are defined in Equations (4.18) and (4.19).

PROOF

Taking the limit

$$\lim_{\alpha\to 0}\Omega_3(\mathbf{b}, b_K = 1|\mathbf{r}) = \lim_{\alpha\to 0}\frac{F_K(\mathbf{b}|\mathbf{r}) + \varepsilon_K(b_K = 1)}{\sigma^2} - \frac{2}{\sigma^2}|\mathcal{Y}(\mathbf{b}, b_K|\mathbf{r}) + \frac{\sigma^2}{2}\alpha|$$

we simply substitute $\alpha = 0$ to find

$$\lim_{\alpha\to 0}\Omega_3(\mathbf{b}, b_K = 1|\mathbf{r}) = \frac{F_K(\mathbf{b}|\mathbf{r}) + \varepsilon_K(b_K = 1)}{\sigma^2} - \frac{2}{\sigma^2}|\mathcal{Y}(\mathbf{b}, b_K|\mathbf{r})|$$

or

$$\lim_{\alpha\to 0}\Omega_3(\mathbf{b}, b_K = 1|\mathbf{r}) = \frac{1}{\sigma^2}[F_K(\mathbf{b}|\mathbf{r}) + \varepsilon_K(b_K = 1) - 2|\mathcal{Y}(\mathbf{b}, 1|\mathbf{r})|]. \qquad \text{(B.1)}$$

It is easy to see from Equations (4.38) and (4.15) that

$$\mathcal{Y}(\mathbf{b}, b_K|\mathbf{r}) = b_K\mathcal{X}(\mathbf{b}|\mathbf{r}),$$

hence, we may substitute $\mathcal{Y}(\mathbf{b}, b_K = 1|\mathbf{r}) = \mathcal{X}(\mathbf{b}|\mathbf{r})$ into Equation (B.1) to obtain

$$\arg\min_{\mathbf{b}}\lim_{\alpha\to 0}\Omega_3(\mathbf{b}, b_K = 1|\mathbf{r}) = \arg\min_{\mathbf{b}}\frac{1}{\sigma^2}[F_K(\mathbf{b}|\mathbf{r}) + \varepsilon_K(b_K = 1) - 2|\mathcal{X}(\mathbf{b}, 1|\mathbf{r})].$$

Finally, we may drop any terms that are independent of $\mathbf{b}$ and we may normalize by any constant multipliers so that we may write

$$\arg\min_{\mathbf{b}}\lim_{\alpha\to 0}\Omega_3(\mathbf{b}, b_K = 1|\mathbf{r}) = \arg\min_{\mathbf{b}}F_K(\mathbf{b}|\mathbf{r}) - 2|\mathcal{X}(\mathbf{b}, 1|\mathbf{r}).\square$$

Next, we examine the case as α gets very large, i.e. as the phase becomes highly accurate. In this case we expect the PC joint weight-only estimator to equal the coherent joint weight estimator in Equation (3.8).

CLAIM

$$\arg \min_{\{\mathbf{b},b_K\}} \lim_{\alpha\to\infty} \Upsilon_2(\mathbf{b},b_K|\mathbf{r}) = \arg \min_{\{\mathbf{b},b_K\}} ||\mathbf{r} - \mathbf{S}\Phi\mathbf{b} - b_K\mathbf{s}_K e^{j\phi_K}||^2$$

PROOF

We take the limit of the likelihood function, $\Upsilon_2(\mathbf{b},b_K|\mathbf{r})$, as α approaches infinity.

$$\arg \min_{\{\mathbf{b},b_K\}} \lim_{\alpha\to\infty} \Upsilon_2(\mathbf{b},b_K|\mathbf{r}) = \arg \min_{\{\mathbf{b},b_K\}} \lim_{\alpha\to\infty} [\Upsilon_2(\mathbf{b},b_K|\mathbf{r}) + \ln\, I_o(\alpha)]$$

Adding the constant term $\ln I_o(\alpha)$, above, does not change the estimate since this term is independent of $[\mathbf{b},b_K]$. We take the limit

$$\lim_{\alpha\to\infty} [\Upsilon_2(\mathbf{b},b_K|\mathbf{r}) + \ln\, I_o(\alpha)] = \frac{F_K(\mathbf{b}|\mathbf{r}) + \varepsilon_K(b_K)}{\sigma^2} - \lim_{\alpha\to\infty} \ln \Psi(\mathbf{b},b_K,\alpha), \qquad \text{(B.2)}$$

where

$$\ln \Psi(\mathbf{b},b_K,\alpha) = \ln\, I_o\Big(\sqrt{(\frac{2}{\sigma^2}\Re\{\mathcal{Y}(\mathbf{b},b_K|\mathbf{r})\} + \alpha)^2 + (\frac{2}{\sigma^2}\Im\{\mathcal{Y}(\mathbf{b},b_K|\mathbf{r})\})^2}\,\Big) - \ln\, I_o(\alpha). \qquad \text{(B.3)}$$

Since

$$\lim_{x\to x_o} \ln(I_o(x)) = \ln(\lim_{x\to x_o} I_o(x)),$$

we wish to find

$$\lim_{\alpha\to\infty} I_o\Big(\sqrt{(\frac{2}{\sigma^2}\Re\{\alpha + \mathcal{Y}(\mathbf{b},b_K|\mathbf{r})\})^2 + (\frac{2}{\sigma^2}\Im\{\mathcal{Y}(\mathbf{b},b_K|\mathbf{r})\})^2}\,\Big) \qquad \text{(B.4)}$$

and

$$\lim_{\alpha\to\infty} I_o(\alpha). \qquad \text{(B.5)}$$

For large a, $\sqrt{(a+b)^2+c^2} \cong a+b$. Also, for large x, $I_o(x) \cong \frac{e^x}{\sqrt{2\pi x}}$. Using these

two approximations, we can reduce (B.4) to

$$\lim_{\alpha\to\infty} \frac{e^{\alpha+\frac{2}{\sigma^2}\Re\{\mathcal{Y}(\mathbf{b},b_K|\mathbf{r})\}}}{\sqrt{2\pi(\alpha+\frac{2}{\sigma^2}\Re\{\mathcal{Y}(\mathbf{b},b_K|\mathbf{r})\})}} \tag{B.6}$$

and (B.5) to

$$\lim_{\alpha\to\infty} \frac{e^{\alpha}}{\sqrt{2\pi\alpha}}. \tag{B.7}$$

Using (B.6) and (B.7) in Equation (B.3) gives

$$\begin{aligned}\lim_{\alpha\to\infty}\Psi(\mathbf{b},b_K,\alpha) &= \lim_{\alpha\to\infty}\frac{e^{\alpha+\frac{2}{\sigma^2}\Re\{\mathcal{Y}(\mathbf{b},b_K|\mathbf{r})\}}/\sqrt{2\pi(\alpha+\frac{2}{\sigma^2}\Re\{\mathcal{Y}(\mathbf{b},b_K|\mathbf{r})\})}}{e^{\alpha}/\sqrt{2\pi\alpha}} \\ &= \lim_{\alpha\to\infty} e^{\frac{2}{\sigma^2}\Re\{\mathcal{Y}(\mathbf{b},b_K|\mathbf{r})\}}\sqrt{\frac{\alpha}{\alpha+\frac{2}{\sigma^2}\Re\{\mathcal{Y}(\mathbf{b},b_K|\mathbf{r})\}}} = e^{\frac{2}{\sigma^2}\Re\{\mathcal{Y}(\mathbf{b},b_K|\mathbf{r})\}}.\end{aligned} \tag{B.8}$$

Substituting Equation (B.8) into Equation (B.2) gives

$$\begin{aligned}\lim_{\alpha\to\infty}\Upsilon_2(\mathbf{b},b_K|\mathbf{r})+\ln I_o(\alpha) &= \frac{F_K(\mathbf{b}|\mathbf{r})+\varepsilon_K(b_K)}{\sigma^2}-\frac{2}{\sigma^2}\Re\{\mathcal{Y}(\mathbf{b},b_K|\mathbf{r})\} \\ &= \frac{1}{\sigma^2}||\mathbf{r}-\mathbf{S}\Phi\mathbf{b}-b_K\mathbf{s}_K e^{j\phi_K}||^2\end{aligned} \tag{B.9}$$

which is the nearest neighbor likelihood function of the ML detector discussed in Section 3.4.1. □

Bibliography

[1] D. P. Bertsekas. *Dynamic programming: deterministic and stochastic models.* Prentice-Hall, Inc., 1987.

[2] Conversations with Dr. Don Boroson, Assistant Group Leader of the MIT Lincoln Laboratory, Satellite Communicaitons Technology Group, Lexington, MA.

[3] R. Coifman and M. V. Wickerhauser. "Entropy-based algorithms for best basis selection". *IEEE Trans. Inform. Theory*, 38(2), Mar. 1992.

[4] A. Duel-Hallen. "Decision-feedback multiuser detector for synchronous code-division multiple access channel". *IEEE Trans. on Comm.*, 41, Feb. 1993.

[5] R. M. Gagliardi. *Satellite Communications Systems.* Van Nostrand, New York, 2 edition, 1991.

[6] R. E. Learned, S. Mallat, B. Claus, and A. S. Willsky. "Geometric Interpretation of Multiaccess Joint Detection and the Alternating Projection Algorithm". In *Proc. IEEE Conf. Acoust., Speech, Signal Processing*, May 1995.

[7] E. A. Lee and D. G. Messerschmitt. *Digital Signal Communication.* Kluwer Academic Publishers, Boston, 2 edition, 1994.

[8] R. Lupas and S. Verdú. "Linear multiuser detectors for synchronous code-division multiple-access channels". *IEEE Trans. Inform. Theory*, 35, Jan. 1989.

[9] R. Lupas and S. Verdú. "Near-far resistance of multiuser detectors in asynchronous channels". *IEEE Trans. on Comm.*, 38:496–508, Apr. 1990.

[10] S. Mallat. "A theory for multiresolution signal decomposition: the wavelet representation". *IEEE Trans. on Pattern Anal. and Mach. Intell.*, PAMI-11, Jul. 1989.

[11] C. H. Papadimitriou and K. Steiglitz. *Combinatorial optimization, algorithms and complexity.* Prentice-Hall, Inc., 1982.

[12] J. G. Proakis. *Digital Communications.* McGraw-Hill, 1989.

[13] Conversations with Dr. Russell Rhodes, Senior Staff in the MIT Lincoln Laboratory, Satellite Communicaitons Technology Group, Lexington, MA.

[14] B. Rimoldi and R. Urbanke. "A rate-splitting approach to the gaussian multiple-access channel". *IEEE Trans. Inform. Theory*, 42, March 1996.

[15] J. A. F. Ross. *"Multiple-User Communications with Non-Orthogonal Signalling".* PhD thesis, McMaster University, Hamilton, Ontario, April 1994.

[16] J. A. F. Ross and D. P. Taylor. "Vector assignment scheme for M+N users in N-dimensional global additive channel". *Electronics Letters*, 28, Aug. 1992.

[17] J. A. F. Ross and D. P. Taylor. "Multiuser signaling in the symbol-synchronous AWGN channel". *IEEE Trans. Inform. Theory*, 41, Jul. 1995.

[18] M. K. Simon, J. K. Omura, R. A. Scholtz, and B. K. Levitt. *Spread spectrum communication handbook.* McGraw-Hill, Inc., New York, 1994.

[19] G. Strang. *Linear algebra and its applications.* Harcourt Brace Jovanovich, Inc., 1988.

[20] H. L. Van Trees. *Detection, estimation, and modulation theory: Part I.* John Wiley and Sons, New York, 1968.

[21] M. Varanasi and B. Aazhang. "Multistage detection in asynchronous code-division multiple-access communications". *IEEE Trans. on Comm.*, 38, Apr. 1990.

[22] M. Varanasi and B. Aazhang. "Optimally near-far resistant multiuser detection in differentially coherent synchronous channels". *IEEE Trans. Inform. Theory*, 37:1006–1018, Jul. 1991.

[23] S. Verdú. "Computational complexity of optimum multiuser detection". In *Algorithmica.* Springer-Verlag., 1989.

[24] A. J. Viterbi. "Phase-locked loop dynamics in the presence of noise by Fokker-Plank techniques". *Proc. IEEE*, December 1963.

[25] A. J. Viterbi. "Optimal detection and signal selection for partially coherent binary communication". *IEEE Trans. Inform. Theory*, April 1965.

[26] A. J. Viterbi. "Error bound for convolutional codes and an asymtotically optimum decoding algorithm". *IEEE Trans. Inform. Theory*, 13, April 1967.

[27] A. J. Viterbi. *CDMA: Principles of spread spectrum communication.* Addison Wesley, Massachusetts, 1995.

[28] W. L. Winston. *Introduction to mathematical programming: applications and algorithms.* Duxbury Press, 1995.