PERFORMANCE AND COMPLEXITY TRADE OFF IN CDMA MULTIUSER COMMUNICATIONS

Mohammed Nafie and Ahmed H. Tewfik

Dept. of Electrical Engineering, University of Minnesota
Minneapolis, MN 55455
mnafie,tewfik@ece.umn.edu

ABSTRACT

In this paper, we describe a new tree-based CDMA receiver that can optimally trade complexity for detection performance. It yields the detector with the best detection performance for a given desired complexity level. Alternatively, it yields the lowest complexity receiver for any given desired detection performance. We describe a technique for designing receivers with linear complexity (including the optimal linear detector and decorrelator). We then explain how we can increase performance at the expense of a minimal increase in complexity. We show that as complexity increases to the level of that of the optimal receiver, our design approach automatically produces the optimal receiver. We also explain how our approach can be used with a minimum-mean-square-error design criterion and coded CDMA transmission. Finally, we illustrate with several examples the superiority of the receivers designed with our approach and discuss their advantages.

1. INTRODUCTION

Detection performance in CDMA communications is limited by the interference of *multiple* users. The effect of this interference on detection performance depends on the users' signatures, and the detector used in the receiver. Matched filter receivers have low complexity but sacrifice detection performance. On the other hand, Optimal multiuser receivers are infeasible because their complexity is exponential in the number of bits per chip. Here, we propose a novel CDMA receiver design approach that optimally trades detection performance for complexity. For any desired detector complexity, our design approach yields the best performing detector. It also produces the lowest complexity receiver for any desired detection performance. This design approach uses the maxmin technique that we described in [1] for constructing low complexity tree-based M-ary hypothesis testing procedures.

For simplicity, we shall focus in this paper on synchronous CDMA detection. In such a case, the powers of users are known a priori and our detector has the following main advantages. When used to design detectors with a complexity proportional to the number of bits per chip, i.e., equivalent to that of the matched filter or the decorrelating receiver, our approach produces a non-linear detector that is asymptotically superior to the optimal linear detector of [2]. Specifically, it has higher asymptotic efficiency for all users after user 1, with user 1 having asymptotic efficiency equal to the optimal linear receiver. Note that, for certain ratios between user powers, our detector is equivalent to the decision feedback detector in [3], while at other ratios, our detector outperforms that detector. In fact, the detector with a complexity equal to the matched filter receiver that our design technique produces, is to the optimal linear detector what the decision feedback detector is to the decorrelating detector. Another main advantage of our design approach is that it can produce the optimal linear detector, or the decorrelating receiver when certain conditions are added to the maxmin optimization. Perhaps, the most important advantage of this proposed detector design approach is its ability to interpolate between the performance of the best-known linear complexity receiver, and the optimal infeasible exponential detector. Finally detectors designed with our technique are a natural choice for coded CDMA.

The rest of the paper is organized as follows. In the next section, we will briefly explain our low complexity M-Hypotheses tree detection algorithm originally presented in [1]. We will also show how the CDMA problem can be posed as an M-Hypotheses detection problem where M is exponential in the number of users. In Section 3, we explain how to design the optimal CDMA detector with a complexity equal to that of the matched filter receiver. We will show that the optimal detector can be applied to any set of signatures with arbitrary cross correlations. It can also be applied to dependent signature sequences (a case where the decision feedback detector does not exist), making it an excellent candidate for detection of oversaturated CDMA communications. We also show that by imposing certain conditions on our maxmin optimization problem, we can obtain the optimum linear detector, or the decorrelating detector. As a byproduct of our design procedure, we give a different derivation of the optimal linear detector. This leads to an easier optimization technique for deriving this detector as compared to the technique given in [2]. We then take our detector one step further and show how to obtain even better performance by increasing the complexity in steps. At complexity equal to the optimal receiver complexity, our receiver converges to this optimal detector. Next, we discuss in Section 4 how the design approach can be used with a Minimum mean squared error receiver design criterion. In the final section, we show how our technique performs in coded CDMA communications.

2. LOW COMPLEXITY M-HYPOTHESIS DETECTION

In [1] we presented a progressive refinement approach to M-ary detection problems. This approach can lead to a logarithmic reduction in the complexity of the detector. Ideally, we would hope to group the hypotheses in two large groups and subsequently split these two sets in two recursively. Unfortunately, binary partitions cannot always capture the exact boundary between two groups of hypotheses. Therefore, we cannot group the hypotheses in disjoint groupings if we wish to achieve a detection performance close to optimal. On the other hand, we want to minimize overlap between the groupings to minimize the number of comparisons required to

This work was partially supported by AFOSR under grant F49620-96-1-0028 and ONR under grant N/N00014-92-J-1678

make a decision. Hence, our problem then is one of approximating the partitions of the decision plane with the *minimal* number of binary partitions, or equivalently of designing a tree of minimal depth.

Now, assume that we receive $v_i + noise$, $1 \leq i \leq M$, where the noise is white Gaussian noise, and v_i is the ith possible transmitted N dimensional vector. We want to divide these M vectors into 2 groups and choose a representative for each group. Assume these representatives are g_1 and g_2. To decide which group is more likely we would compare $r^T g_1$ and $r^T g_2$. Or, we can compare $r^T(g_1 - g_2)$ to a certain threshold. Let $g = g_1 - g_2$. Therefore, we should select a unit norm N dimensional vector, g, that solves

$$\max_{g,\ th} \min_{v_i} abs(g^T v_i - th) \tag{1}$$

and then divide the vectors into two groups: those that have correlation with g larger than the threshold, th, and those with less than the threshold correlation. We then continue building the tree by repeating the above procedure at each node, till we reach the leaf nodes with only one hypothesis associated with them.

Now, in CDMA with K users, assume that we form a vector, $\vec{b_i}$, formed of a certain possible combination of all the user bits. The received signal $\vec{r}$ can be written as $\vec{r} = v_i + noise$, where

$$v_i = \sum_{k=1}^{K} b_i(k) * A_k * \vec{s_k} \tag{2}$$

where $b_i(k), A_k, \vec{s_k}$ are, respectively, the transmitted bit, amplitude, and signature of user k. We form the M hypotheses as the possible noiseless transmissions using all possible combinations of user bits.

3. PROPOSED DETECTOR

3.1. Designing Detectors with Linear Complexity

We shall use the terminology "linear complexity" to mean a complexity equivalent to that of the matched filter receiver, that is a complexity of one vector multiplication per bit. In this section we will show how to design such an optimal detector. We will also show that by adding some constraints to our optimization problem we can get the optimal linear detector, or the decorrelating detector. At each node of the tree, we want to select the unit norm vector g that solves (1). This problem can be written in the form of a constrained minimization problem [4]. This problem may lead us to local minimas, and so we choose to pre assign vectors to the two groups, the group belonging to the left branch and the group belonging to the right branch. Assume that the preassigned groups are called v_l and v_r. Therefore, our optimization problem can now be written as

$$\max_{g,\ th} \min_{v_i} [(g^T v_{i \in l} - th),\ (th - g^T v_{i \in r})] \tag{3}$$

where vectors $v_{i \in l}$ are the ones chosen to be in group 1, and $v_{i \in r}$ are the vectors chosen to be in group 2. By some manipulation of the equivalent minimization problem, this maxmin problem can be shown to be convex.

Dividing the vectors into two groups can be done in a variety of ways. For example, we can use one of the local minimas from (1), and then re-optimize. Or we can decide that we want to divide the vectors into 2 equal halves, and solve the optimization problems for all the possible divisions of the vectors. This means that for M vectors, we have to solve this optimization problem $\frac{M!}{2 * \frac{M}{2}! * \frac{M}{2}!}$ times. This is a large number for large M, but notice that this is done off line, and therefore in some cases it can be affordable. Also, we can choose to assign all the vectors corresponding to user 1 transmitting a '1' to group 1, and those corresponding to a '-1' to group 2. Therefore, j will take values $1, 2, \ldots \frac{M}{2}$, and k values $\frac{M}{2}+1, \frac{M}{2}+2, \ldots .. M$. On the next node we can do this division based on user 2 transmission, and so on. Doing this gives us the feedback detector of [3], with the difference that our detector at each stage converges to the best linear detector and not the decorrelating detector. Let us assume that each user is transmitting binary bits. Our tree detector will be as shown in Fig. 1. At each level of the tree a user bit will be declared. Notice that because of the symmetry between nodes at any given level of the tree, the vector g will be the same for the nodes at the same level, while the threshold will be different from a node to the other. We will call this vector, at level j, $\vec{z_j}$. We will also call the tree designed using this approach a "natural" tree.

As a demonstration of an oversaturated communication system, four random signatures of length 3 were given to four users. Notice that the detector of [3] is not defined here. We built two different trees. The first one was the "natural" tree described above. Another tree was built by trying different divisions at each node, and-for each node- choosing the division that maximizes the distance between the two clusters. The results are shown in Fig. 2. It is obvious that by optimizing the tree in that case, the probability of error was decreased substantially. This shows that in certain cases, especially those that the "natural" tree suffers from a large difference in error from the optimal probability of error, optimizing the tree can gain us a large part of this difference.

To construct the optimal linear detector using our tree approach, we design a "natural" tree described above. In a "natural" tree, deciding on user k transmitted bit depends entirely on which cluster we chose at level k of the tree. If, at each node of our tree, we want the probability of choosing the wrong cluster to be independent on the previous bit decisions, we have to make the vector that differentiates between the two clusters, $\vec{z_k}$, and the threshold the same for all the nodes at a certain level. Due to the symmetry of the vectors, we can choose the threshold to be zero, and solve the maxmin problem for g only. Notice that the above procedure applies to whether the vectors are dependent or not.

Similarly, we can derive the decorrelating detector by identifying a "natural" tree. Since, the decorrelating detector "decorrelates" the signatures of the different users, the vector $\vec{z_k}$ at level k should be orthogonal to all other user's signatures. Therefore, to make our tree detector equivalent to the decorrelating detector, we add this orthogonality constraint to our maxmin problem.

3.2. Designing Higher Complexity Optimal Detectors

The way we defined our clusters in this paper till this point, does not allow overlapping of clusters. As shown in [1], allowing vectors to fall in both clusters would enhance the performance of our detector, while at the same time increase the complexity of bit detection. We can re-define our optimization problem to allow vectors to fall in both cluster instead in just one. Let v_l correspond to the vectors in the left branch of the current node, and v_r correspond to the right branch. Allowing repetition means that the sets v_l and v_r intersect. Let v_{ll} to be those vectors in the left branch but not in the right branch, and v_{rr} to be those in the right branch and not the left branch. Now, we solve the maxmin problem using only these "exclusive" vectors.

Once more, we have a variety of ways to choose these partitions. We can choose the number of vectors we are allowing to be repeated, and then try all possible combinations. We can start with the linear complexity tree, and then take those vectors in the right cluster and closer to the left cluster, and those in the right cluster and closer to the left cluster, and place them in both clusters. We also might choose, at the first node, those vectors corresponding to users 1 and 2 sending a '1' and a '1' to be in a cluster, and those corresponding to users 1 and 2 sending a '-1' and a '1' to be in the other cluster. On the next node, we can divide the vectors corresponding to a '1' and a '-1' and those corresponding to a '-1' and a '-1'. After these 2 divisions, we are either decided on user '1' or we need another comparison. We can choose the third comparison such that we are left with a decision for both users 1 and 2.

Our Experiments indicate that the following few guidelines always enhance performance. One should always choose user 1 to be the user with the largest power. Also, choose user 2 to be the one that has the largest correlation with user 1. Also, try to keep the v_i's that are separated by a small distance in the same cluster. If a certain v_i is close to many other v_i's allow it to repeat in several clusters until the leaf node.

Fig. 3 shows the probability of error at various noise standard deviations for a 4-user CDMA system that uses pseudonoise sequences, and are assumed to be of equal power. We use 2 trees that use either 5 or 6 multiplications per 4 bits. The trees are not shown here due to lack of space. Fig. 4 gives a complexity-probability of error curve at a certain signal to noise ratio.

We can also implement the optimal multiuser detector using our technique. In particular, we will make every node differentiate between only 2 vectors. The two clusters from the first node will contain $M-1$ vectors each, with $M-2$ vectors repeated in both clusters. At the next level, we repeat $M-3$ vectors, and so on. We will hence have a tree with $M-1$ levels, so our complexity for binary K users is 2^K-1, which is the exponential complexity of the optimal detector. It is clear that the performance is independent on which nodes differentiate between which vectors, and that to differentiate between vectors v_r and v_l, we multiply by $z = v_r - v_l$ and compare to $\frac{1}{2}(v_l^T v_l - v_r^T v_r)$.

4. MINIMUM MEAN SQUARED ERROR DETECTOR

Minimum mean squared error detector can be used for CDMA multiuser communications [5] [6]. By writing all the possible M hypotheses, we can also implement the minimum mean squared error detector in a tree structure. We can define the MMSE detector as the vector $\vec{x}$ which when multiplied by the received vector, minimizes the mean of the squared error between the transmitted bit and the result of this multiplication. The mean is taken over all possible user transmissions, i.e. over all possible values of v_i for $i = 1...M$ and over all possible values of the noise. Therefore we can write the problem as

$$\min_{\vec{x}} E_{pdf(noise),v_i}(\pm 1 - \vec{x} * \vec{r})^2 \tag{4}$$

where the +1 is for those v_i corresponding to user 1 sending a 1, and the -1 for those vectors corresponding to user 1 sending a -1 and $\vec{x}$ is an N-dimensional vector. Also, $\vec{r} = v_i + \vec{n}$, where $\vec{n}$ is a Gaussian noise vector. This definition will allow us to extend the MMSE detector to operate with higher complexity than K and come closer to the optimal detector.

Specifically, we can use any of the methods used in building larger trees. We divide the vectors into any two overlapping clusters. In that case, the outputs of both clusters might not be equally spaced from zero due to the DC bias in them. To account for this we have to subtract this DC term from the output. Therefore, we pad a -1 to the end of the vectors v_i. Call these extended vectors v_i^e. Therefore, our problem would be as in (4), but $\vec{x}$ is now an $(N+1)$-dimensional vector. Also, $\vec{r} = v_i^e + \vec{n}$, where $\vec{n}$ is a vector whose first N components are the Gaussian noise vector, and its last element is zero.

An advantage of the MMSE approach over the maxmin approach is the ability to find the detector adaptively using the Least Mean Squares (LMS) approach [7]. However, a small adaptation has to be done to the LMS to make it suitable for the cases where there is a DC bias. As in the none adaptive version, we extend the filter $\vec{x}$ by an extra component. When filtering the received vector we append to it a -1.

The adaptation rule is $\vec{x}_{t+1} = \vec{x}_t - \mu * e_t * \vec{r}_t$ where e_t is the current error, and is equal to $\pm 1 - \vec{x}_t^T \vec{r}_t$. Notice that the adaptation is not done on each bit that is sent. Instead, it is done when the transmitted vector lies in one cluster but not the other. It can be done through training or it can be decision directed. The last element in $\vec{x}$ can be updated using the previous expression. Alternatively, it can be adjusted to cancel the DC bias.

As an example consider the case of 4 user CDMA, and let us cluster the vectors such that the vectors corresponding to user 1 sending a 1, and user 2 sending a 1 lie in one cluster, and those corresponding to user 1 sending a -1 and user 2 sending a 1 in the other cluster. Fig. 5 shows how the tree MMSE can track changes in amplitude. Here, we start with $\vec{x}_0$ corresponding to equal amplitude users, then assume the users have random amplitudes. As in any LMS algorithm, particular care has to be used in choosing the adaptation factor μ so that the algorithm does not get trapped in local minima. This method also works with unknown signatures.

5. ERROR CORRECTION

Using block error correction decoding in multiuser communications seems particularly suited to our approach in detection. Our hypotheses will be only the possible outputs, and hence we can easily use a joint approach for detection and error correction. Other schemes allow all the possible binary combinations in detection, then decode in a later stage. Recently, in [8], the decoding was done on each user's bits before going on to detect the next users, but in detecting the user bits, no use of the error correction code characteristics were utilized.

Assume we use a (n,k) block code, and we have a K user CDMA system, where each user uses a particular pseudonoise signature of length p. To decode and detect jointly, we will have to look at a frame of length n bits. We will assume we have $n*K$ users, where each user uses a $n*p$ length signature. Each user corresponds to a particular user sending a particular coded bit. So, each of these expanded signatures will be zeros everywhere, except for p consecutive chips where it is equal to the original signature of one of the users. Using these signatures, we could generate a 2^{n*p} hypotheses, and solve it in our usual way, and then do decoding. If we did that, and built our tree such that each user bits were detected before going on to detect next user bits, we get the same detector as in [8]. But, we can also, through our knowledge of the error correction code codebook, throw away all the impossible sequences, and hence, end up with only 2^{k*p} hypotheses. Building a tree of complexity $k*p$ now will allow us to jointly detect and decode while making some use of the power of our error correction code while using only a complexity of 1 per uncoded user bit, instead of 1 per coded bit as in current techniques. We can also build larger trees which have higher complexity, while decreasing our probability of error.

We will demonstrate this with an example. Assume we have a 2-user CDMA system. Assume that each user uses a (7,4) Hamming code [9]. Fig. 6 compares the probability of error using a tree of complexity equal to 7 with the feedback technique explained in

[8]. Our technique out performs this technique. Notice that what we calculated for the feedback technique is the result of their first stage. We could always get better results by adding subsequent stages, but if the initial decision is more reliable, as by using our detector, the results after any stage will also be better. Each of the two detectors use 7 matched filter operations per 4 uncoded bits of each user.

6. REFERENCES

[1] M. Nafie and A. Tewfik, “Reduced Complexity M-ary Hypotheses Testing in Wireless Communications,” ICASSP, Seattle, WA, May 1998.

[2] R. Lupas and S. Verdu, ”Linear multiuser detectors for synchronous code-division multiple-access channels,” IEEE Trans. Info. Theory, vol. 35, pp. 123-36, Jan. 1989.

[3] A. Duel-Hallen, “Decorrelating Decision-Feedback Multiuser Detector for Synchronous Code-Division Multiple-Access Channel,” IEEE Trans. Commun., vol. 43, pp. 421-34, Feb.-April 1995.

[4] P. Gill, W. Murray, and M. Wright, *Practical Optimization*, Academic Press, CA, 1981.

[5] S. Verdu, University of Minnesota EE 8220 Class Notes, Spring 1995.

[6] U. Madhow and M. Honig, “MMSE Interference Suppression for Direct Sequence Spread Spectrum CDMA,” IEEE Trans. Comm., Vol. 42, pp. 3178-3188, Dec. 1994.

[7] S. Haykin, *Adaptive Filters*, Prentice Hall, NJ, 1991.

[8] R. Learned and A. Singer, “Coding Enhanced Joint Detection for Multiple Access Communications,” ICASSP, Seattle, WA, May 1998.

[9] S. Lin and D. Costello, *Error Control Coding*, Prentice Hall, NJ, 1982.

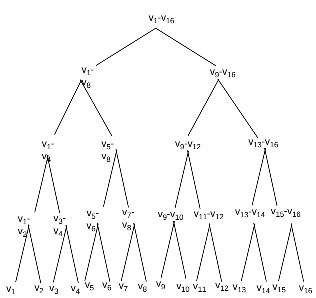

Figure 1: “Natural” tree for 4 users

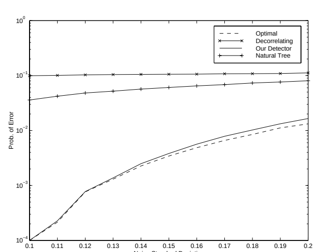

Figure 2: Probability of Error for four users using length 3 PN sequences

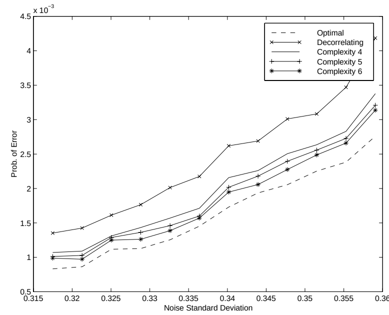

Figure 3: Probability of Error at various complexities

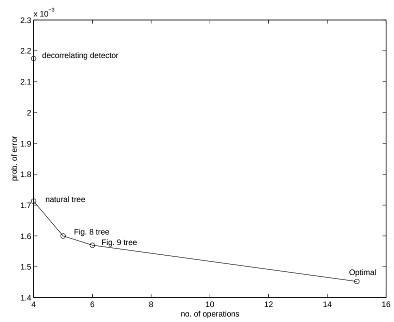

Figure 4: Complexity-Probability of Error Trade-off

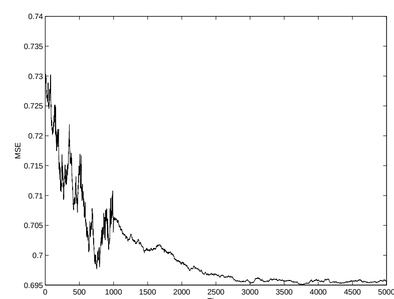

Figure 5: Mean squared error at each time instant for unknown user amplitudes

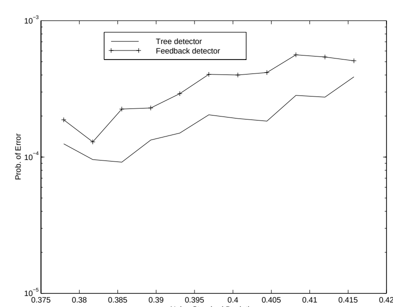

Figure 6: Probability of Error for a 2 user CDMA with each user using a (7,4) Hamming code-complexity=1 matched filter operation per coded bit