

A Low-Complexity High-Performance Preprocessing Algorithm for Multiuser Detection using Gold Sequences

Daniel Axehill, Fredrik Gunnarsson, Anders Hansson

Division of Automatic Control

E-mail: daniel@isy.liu.se, fred@isy.liu.se,
hansson@isy.liu.se

31st January 2008

Report no.: LiTH-ISY-R-2840

Submitted to IEEE Transactions on Signal Processing

Address:

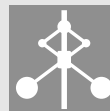
Department of Electrical Engineering

Linköpings universitet

SE-581 83 Linköping, Sweden

WWW: <http://www.control.isy.liu.se>

AUTOMATIC CONTROL
REGLERTEKNIK
LINKÖPINGS UNIVERSITET



Abstract

The optimum multiuser detection problem can be formulated as a maximum likelihood problem, which yields a binary quadratic programming problem to be solved. Generally this problem is \mathcal{NP} -hard and is therefore hard to solve in real-time. In this article, a preprocessing algorithm is presented which makes it possible to detect some or all users optimally for a low computational cost if signature sequences with low cross-correlation, *e.g.*, Gold sequences, are used. The algorithm can be interpreted as, *e.g.*, an adaptive trade-off between parallel interference cancellation and successive interference cancellation. Simulations show that the preprocessing algorithm is able to optimally compute more than 94% of the bits in the problem when the users are time-synchronous, even though the system is heavily loaded and affected by noise. Any remaining bits, not computed by the preprocessing algorithm, can either be computed by a suboptimal detector or an optimal detector. Simulations of the time-synchronous case show that if a suboptimal detector is chosen, the BER rate is significantly reduced compared to using the suboptimal detector alone.

Keywords: Preprocessing, Multiuser Detection, Code Division Multiple Access, Integer Quadratic Programming

A Low-Complexity High-Performance Preprocessing Algorithm for Multiuser Detection using Gold Sequences

Daniel Axehill, Fredrik Gunnarsson and Anders Hansson

Abstract—The optimum multiuser detection problem can be formulated as a maximum likelihood problem, which yields a binary quadratic programming problem to be solved. Generally this problem is \mathcal{NP} -hard and is therefore hard to solve in real-time. In this article, a preprocessing algorithm is presented which makes it possible to detect some or all users optimally for a low computational cost if signature sequences with low cross-correlation, *e.g.*, Gold sequences, are used. The algorithm can be interpreted as, *e.g.*, an adaptive trade-off between parallel interference cancellation and successive interference cancellation. Simulations show that the preprocessing algorithm is able to optimally compute more than 94% of the bits in the problem when the users are time-synchronous, even though the system is heavily loaded and affected by noise. Any remaining bits, not computed by the preprocessing algorithm, can either be computed by a suboptimal detector or an optimal detector. Simulations of the time-synchronous case show that if a suboptimal detector is chosen, the BER rate is significantly reduced compared to using the suboptimal detector alone.

I. INTRODUCTION

Multiuser detection (MUD) is the process to demodulate multiple users sharing a common multi-access channel. A first approach is to demodulate each user independently and treat the signal from other users as additive Gaussian noise, [1]. An improvement to this strategy is to use the known correlation between users in the demodulation process. The performance can be improved even more if the detector makes the most likely decision, which formally is achieved by solving a so-called Maximum Likelihood (ML) problem. When the optimum multiuser detection problem is cast in the form of an ML problem it requires the solution of a so-called Binary Quadratic Programming (BQP) problem. Unfortunately, these problems are generally known to be \mathcal{NP} -hard, [2]. If the signature waveform produces a cross-correlation matrix with some special structures, the problem can sometimes turn out to have lower complexity, [3], [4], [5].

Many contributions to the area of multiuser detection have already been published. The objective has been to find an algorithm which solves the multiuser detection problem in reasonable time in order to make a real-time implementation possible. Previously, this has been done either by restricting the class of possible cross-correlation matrices or by employing some suboptimal procedure. In [3], an algorithm with polynomial complexity has been derived for systems with only negative cross-correlations. A similar requirement on the cross-correlation matrix is found in [4], where the synchronous multiuser detection problem is solved with a polynomial complexity algorithm if the cross-correlations between the users are non-positive. It is also shown that Gold sequences satisfy this condition in the synchronous case. Another paper also dealing with a special class of cross-correlations is the one in [5], where a polynomial complexity algorithm is derived for the case of identical, or a few different, cross-correlations between the users. Thorough work in the field of approximate algorithms for multiuser detection is found in [1]. Several different algorithms, optimal as well as suboptimal, are

presented and evaluated in [6]. The suboptimal algorithm local search is evaluated in [7]. Branch and bound methods are investigated in [8]. Another near optimal approach is presented in [9]. Also the well-known Kalman filter has been applied to the problem. This approach is presented in [10]. Another interesting approach is to use so-called semidefinite relaxations to produce suboptimal solutions to the problem. This idea has been considered in, *e.g.*, [11] and [12].

A different detector approach is based on adaptive thresholding to provide a low-complexity detector for the ML problem, [13], [14]. The resulting algorithm is evaluated at low traffic loads for pseudo-random sequences with promising results. It is also concluded that the algorithm is not expected to perform well at high loads.

In this article, a preprocessing algorithm with polynomial complexity for the BQP problem is derived. A preprocessing algorithm is an algorithm which processes the optimization problem in the step previous to the one when the actual solver is applied. Because the preprocessing algorithm executes in polynomial time and the BQP solver, generally, executes in exponential time, the required CPU time can be reduced if bits can be computed optimally already in the preprocessing step. It became clear that when applied to the MUD problem, the resulting BQP preprocessing algorithm presented in [15] was found to be equivalent to the one presented in [13], [14]. The two major differences and enhancements made in this article compared to [13], [14] is that, first, the algorithm is derived using an optimization point of view of the problem, which results in a general algorithm for BQP problems and a framework which makes it easy to prove optimality for the formulation of the problem solved in the step after preprocessing, and second, that the evaluation is very thoroughly performed in the important case where chip sequences of Gold type are used. Both the BER performance as well as the computational performance is evaluated. Note that the ideas presented in this article also work with other types of sequences with low cross-correlation, and the Gold sequences were chosen as one representative of such sequences. The algorithm is in principle also useful if the favorable cross-correlations from the time-synchronous case have been slightly distorted by, *e.g.*, time-asynchronous users, as long as the resulting cross-correlations between the users are low. Furthermore, it is shown in simulations that when signature sequences with low cross-correlations, like Gold sequences, are used in a synchronous system, the algorithm is useful also at high loads.

Section II provides the CDMA channel models upon which this article is based, and Section III describes preprocessing for the general BQP problem. The resulting preprocessing is applied to CDMA in Section IV, followed by extensive numerical evaluations in Section V. Finally, Section VI provides some conclusive remarks.

II. CDMA CHANNEL MODELS

In this section, the synchronous and asynchronous CDMA channel models used in the article are presented. The notation has been taken from [16], where also a more thorough description of the models can be found. In the case when the users are time-synchronous, the received waveform $y(t)$ is modeled as a K -user channel consisting of the sum of K antipodally modulated synchronous signature

The research has been supported by the Swedish Research Council for Engineering Sciences under contract Nr. 621-2002-3822.

The authors are with the Division of Automatic Control, Linköpings universitet, SE-581 83 Linköping, Sweden, (e-mail: {daniel.fred,hansson}@isy.liu.se)

waveforms embedded in additive white Gaussian noise, *i.e.*, as

$$y(t) = \sum_{k=1}^K A_k b_k s_k(t) + \sigma n(t), t \in [0, T] \quad (1)$$

where A_k is the received amplitude of the signal from user k , $b_k \in \{-1, +1\}$ is the data bit transmitted by user k , $s_k(t)$ is the deterministic signature waveform assigned to user k , normalized to have unit energy, and $n(t)$ is white Gaussian noise with unit power spectral density. Furthermore, σ is the channel noise variance and T is the inverse of the data rate.

The similarity of different signature waveforms $s_k(t)$ is expressed in terms of the cross-correlation defined by

$$\rho_{ij} = \int_0^T s_i(t) s_j(t) dt \quad (2)$$

The normalized cross-correlation matrix $R = \{\rho_{ij}\}$ has ones in the diagonal and is symmetric non-negative definite, [16]. If $\rho_{ij} = 0$ whenever $i \neq j$, the signature sequences are orthogonal. In this work, non-orthogonal sequences of Gold type have been used and these usually give low cross-correlation for all possible non-zero offsets. Another common choice of sequences with these properties is Kasami sequences.

In the case when the users are time-asynchronous, a frame of length $2M + 1$ symbols is considered. The signature sequences used in this work are user-specific, and reused repeatedly for each new bit sent by a specific user. The CDMA channel model is also in this case adopted from [16] and can be written as

$$y(t) = \sum_{k=1}^K \sum_{i=-M}^M A_k b_k [i] s_k(t - iT - \tau_k) + \sigma n(t) \quad (3)$$

where $b_k[i]$, $i = -M, \dots, M$ represent the bits sent by user k during the frame under consideration, and τ_k is the user-specific time offset. Assume, without any loss of generality, that the users are labeled by their time of arrival, *i.e.*, that $\tau_1 \leq \tau_2 \leq \tau_3 \leq \dots \leq \tau_K$. If $k < l$, the elements in the cross-correlation matrix can be written as

$$\begin{aligned} \rho_{kl} &= \int_{\tau_l - \tau_k}^T s_k(t) s_l(t - \tau_l + \tau_k) dt \\ \rho_{lk} &= \int_0^{\tau_l - \tau_k} s_k(t) s_l(t + T - \tau_l + \tau_k) dt \end{aligned} \quad (4)$$

where $0 \leq \tau_l - \tau_k \leq T$. If it is assumed that the intersymbol interference is limited to occur for symbols immediately adjacent in time, the complete cross-correlation matrix R is in this case built-up as a block-Toeplitz matrix with blocks $R[0]$ along the main diagonal and cross-correlations $R[1]^T$ and $R[1]$, repeated above and below the main block diagonal, respectively, where,

$$R_{jk}[0] = \begin{cases} 1, & \text{if } j = k \\ \rho_{jk}, & \text{if } j < k \\ \rho_{kj}, & \text{if } j > k \end{cases}, \quad R_{jk}[1] = \begin{cases} 0, & \text{if } j \geq k \\ \rho_{kj}, & \text{if } j < k \end{cases} \quad (5)$$

According to [16], the bit-sequence most likely sent by the users in the synchronous case are given by the solution $\hat{b} = [\hat{b}_1, \hat{b}_2, \dots, \hat{b}_n]^T$ to the optimization problem

$$\max_{\hat{b}} \exp \left(-\frac{1}{2\sigma^2} \int_0^T \left(y(t) - \sum_{k=1}^K \hat{b}_k A_k s_k(t) \right)^2 dt \right) \quad (6)$$

The asynchronous case follows analogously, but the notation becomes slightly more complex. Note that, the variables in the optimization

problem in (6) are the bits to be estimated. This optimization problem can be rewritten as an equivalent minimization problem

$$\min_{\hat{b}} \frac{1}{2} \hat{b}^T H \hat{b} - y^T A^T \hat{b} \quad (7)$$

where $\hat{b} \in \{-1, +1\}^K$, $A = \text{diag}(A_1, A_2, \dots, A_K)$, $H = ARA$ and $y = [y_1, y_2, \dots, y_K]^T$, where y_i is the output from matched filter i . After a variable substitution, this problem can be identified as a 0/1-BQP problem.

There exist a number of suboptimal detectors applicable to the multiuser detection problem. The simplest one is the conventional detector, [6],

$$\hat{b} = \text{sign}(y) \quad (8)$$

where disturbances from other users are treated as noise. This detector does not use any floating-point operations (flops). A simple detector using knowledge of the cross-correlation between different users is the decorrelating detector, [6],

$$\hat{b} = \text{sign}(H^{-1}Ay) \quad (9)$$

The number of flops for computing computing $H^{-1}Ay$ grows as $\mathcal{O}(K^3)$, [17].

For what follows, the channel models in (1) and in (3) are used for all computational results to be presented in this article. Furthermore, the received amplitudes from different users A_k are assumed equal to one for all users, *i.e.*, the uplink is subject to power control. The signature waveforms used are all of Gold type of length 127.

III. PREPROCESSING FOR THE BQP PROBLEM

Most algorithms used when solving BQP problems either focus on producing approximative solutions or only on handling various special cases of the general problem, [18]. The algorithm to be presented in this article belongs to the latter type of algorithms and it is applicable to BQP problems having dominating diagonal elements compared to the off-diagonal elements. In [15], the algorithm has previously been successfully applied to Model Predictive Control (MPC) for systems including binary variables. Some approximative heuristic algorithms can be found in, *e.g.*, [2], [19], [20], and [21].

In this article, a BQP problem in the form

$$\mathcal{P} : \begin{cases} \min_x & \frac{1}{2} x^T H x + f^T x \\ \text{subject to} & x \in \{0, 1\}^{n_b} \end{cases} \quad (10)$$

is studied, where $H \in \mathbb{R}^{n_b \times n_b}$ is symmetric. The algorithm presented in this section makes it possible to speed up the solution of BQP problems having large diagonal elements compared to the non-diagonal elements. For each binary variable the algorithm delivers one out of three possible results: 1 is the optimal value, 0 is the optimal value, or nothing can be said for sure. Some of the basic BQP optimization ideas used in this work can be found in, *e.g.*, [22]. In this reference, the ideas are not used to build a preprocessing algorithm, but to find all optimal solutions to the BQP problem. Before the main theorem is given, the following definition is introduced

$$H = H_d + H^+ + H^- \quad (11)$$

where

$$\begin{aligned} H_{d,ij} &= \begin{cases} H_{ij}, & i = j \\ 0, & i \neq j \end{cases} \\ H_{ij}^+ &= \max(0, H_{ij} - H_{d,ij}) \\ H_{ij}^- &= \min(0, H_{ij} - H_{d,ij}) \end{aligned} \quad (12)$$

The preprocessing algorithm is built upon the following theorem:

Theorem 1: For a BQP problem of type \mathcal{P} , an optimal value of one or more components x_i can be found in polynomial time if for some $i \in \{1, \dots, n_b\}$ any of the following conditions is satisfied

$$(i) : \quad H_{ii} \geq -2f_i - 2 \sum_{j=1}^{n_b} H_{ij}^-$$

$$(ii) : \quad H_{ii} \leq -2f_i - 2 \sum_{j=1}^{n_b} H_{ij}^+$$

If any of the conditions (i) or (ii) is satisfied for a certain value of i , an optimal value of x_i is given by

$$x_i = \begin{cases} 0, & \text{if (i) holds} \\ 1, & \text{if (ii) holds} \end{cases}$$

Proof: See [15]. A simple flop count shows that the computational complexity for each test (i) or (ii) grows as $\mathcal{O}(n_b)$. Hence, performing the tests for all variables gives a computational complexity of $\mathcal{O}(n_b^2)$, i.e., polynomial complexity in n_b . ■

IV. APPLICATION OF THE RESULTS TO CDMA

In this section, it is shown how to apply the preprocessing algorithm derived in Section III to the optimization problem in (7) and how the result can be interpreted.

A. Using Preprocessing

In order to be able to apply the preprocessing algorithm, the optimization problem in (7) has to be rewritten in the BQP form \mathcal{P} . Note especially the domain of the optimization variable x . In order to get an optimization problem with binary variables the following variable substitution is performed

$$\hat{b} = 2\bar{b} - \mathbf{1} \quad (13)$$

where $\bar{b} \in \{0, 1\}^K$, $\hat{b} \in \{-1, +1\}^K$ and $\mathbf{1}$ denotes a column vector with all elements equal to one. Using (13), neglecting constant terms and dividing by 4, the objective function in (7) can be rewritten as

$$\frac{1}{2} \bar{b}^T H \bar{b} + \bar{f}^T \bar{b} \quad (14)$$

where

$$\bar{f} = -\frac{1}{2} H \mathbf{1} - \frac{1}{2} A y \quad (15)$$

The problem is now in the form \mathcal{P} , on which preprocessing can be performed.

B. Using the Algorithm in Multiuser Detection

In this section it is described how the algorithm can be used in multiuser detection. It is first shown how it works in its simplest form where the conditions in Theorem 1 are applied once for each user. Later it is shown how the performance of the algorithm can be significantly improved by applying the algorithm iteratively. Finally, it is described how any remaining bits not computed by the preprocessing algorithm can be computed by either an optimal or a suboptimal algorithm. Detailed algorithm descriptions can be found in [23], and in [13].

1) *Non-Iterated Implementation:* In its simplest form the algorithm consists of applying the conditions in Theorem 1 once for each user. As previously discussed in the proof of Theorem 1, the computational complexity of one such preprocessing iteration (test the conditions once for each user) grows as $\mathcal{O}(K^2)$. Using the notation in the multiuser detection problem, the two conditions (i) and (ii) in the corollary can after simplification be written as

$$\begin{cases} A_i y_i \leq -\sum_{j \neq i} |H_{ij}| & (i) \\ A_i y_i \geq \sum_{j \neq i} |H_{ij}| & (ii) \end{cases} \quad (16)$$

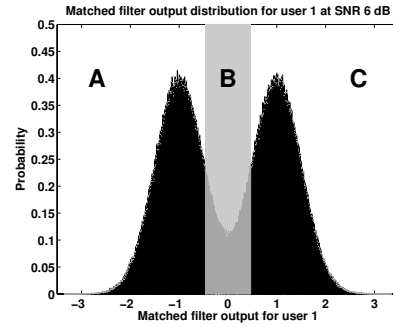


Figure 1. This plot shows the distribution for the matched filter output for user 1 in an example with 60-users and SNR 6 dB for this user. The gray region **B** illustrates the region where user 1 cannot be detected in a single preprocessing iteration. The computational result shows that in this example, the probability of detection of user 1 in the first preprocessing iteration is 0.856.

From (13) and (16), it follows that the optimal choice of \hat{b}_i is given by

$$\hat{b}_i^* = \begin{cases} -1, & \text{if (i) holds} \\ 1, & \text{if (ii) holds} \end{cases} \quad (17)$$

Note that these conditions are equivalent to the ones given in [13]. Furthermore, note that the conditions in (16) for a certain bit i are independent of the value of any other undetected bits. Hence, if a decision can be made, it is made independently of the value of any so far undetected bits. If (16) is investigated, it can be realized that a necessary property of the cross-correlation matrix to be able to successfully use the algorithm, is that the signature sequences give low cross-correlations between different users. This is typically the case for, e.g., Gold sequences.

In the non-iterated implementation, the detector can be interpreted as a threshold detector. Considering, e.g., user 1, the distribution of the output from the matched filter for this user is illustrated in Figure 1. In the example, 60 users are sharing the channel and the SNR for user 1 is 6 dB. If the matched filter output gets outside the gray region **B**, the bit sent by the user considered can be detected *optimally* by the preprocessing algorithm in a *single* iteration. Note that the size of the threshold has been chosen by the algorithm in order to be able to *guarantee* optimality. The algorithm can also be related to the conventional detector in (8). In region **A** and **C**, the output from the preprocessing algorithm and the output from the conventional detector coincide. There is however an extremely important difference; the output from the preprocessing algorithm is guaranteed to be optimal. Note that the fact that the output from the two detectors often coincide is not a weakness of the preprocessing algorithm but rather a proof that the conventional detector often, but not always, make optimal decisions. Compared to previous optimal low complexity methods presented in [3], [4] and [5], the algorithm presented in this article does not introduce any requirements on the sign of the cross-correlations or that the cross-correlations between users are equal. On the other hand, the preprocessing algorithm cannot, in general, detect all users. In most cases the algorithm can detect some of the users in the first iteration, but there is also often some users which cannot be detected in the first iteration. In the iterated implementation of the algorithm, the algorithm is applied repeatedly and bits detected optimally in previous iterations are used to relax the conditions for previously unsolved bits.

2) *Iterated Implementation:* When the algorithm is applied iteratively, the number of users possible to detect by the preprocessing algorithm is significantly increased. As long as at least one user

could be detected in the previous run, it is possible to start over again and try to compute the remaining ones. In principle, this is done by inserting the already computed variables in the objective function of the BQP problem and formulating a new BQP problem in the remaining variables on which it is possible to run preprocessing. This is described in detail in the next section. This procedure is in this text referred to as the iterated implementation and is further described in [23]. Note that even in the worst case the number of iterations is limited to K since the algorithm is aborted if less than one user was detected in the last iteration. Hence, the worst case computational complexity for the iterated implementation of the preprocessing algorithm grows as $\mathcal{O}(K^3)$. Furthermore, note that it is possible, *e.g.*, to be able to satisfy time demands in a real-time system, to abort the algorithm before the K steps have completed and to solve any remaining variables suboptimally. Hence, it is easy to make a dynamic trade-off between BER performance and computational time. This has been further investigated in [14].

To gain performance compared to the conventional detector it is necessary that information from previous iterations contracts region **B** and either extends region **A** to the right of the origin or extends region **C** to the left of the origin. That this actually occurs in practice will be shown in Section V where the proposed algorithm outperforms the conventional detector.

When the iterated implementation is used, the algorithm can be interpreted as an adaptive trade-off between Parallel Interference Cancellation (PIC) and Successive Interference Cancellation (SIC). In SIC, only one user is detected and cancelled in each iteration, which means that the number of iterations equals the number of users to be detected. On the contrary, in PIC all users are detected in every iteration and subsequently cancelled. Uncertain detection reduces cancellation performance, and therefore, only a fraction of the regenerated signals are cancelled, known as partial PIC. Hence, in SIC only one user is detected per iteration even though more users can be detected with confidence, while in partial PIC all users are detected, also the ones with less confidence, in each iteration. In the proposed scheme, all previously undetected users that can be detected with confidence, but no more, are detected in each iteration. More information on PIC and SIC can be found in, *e.g.*, [24].

3) *The Step After Preprocessing:* When the preprocessing algorithm has terminated, another detector has to be applied to detect any remaining bits. Consider an optimization problem in the form in (7). Without any loss of generality, order the elements in \hat{b} in two parts, and denote the parts \hat{b}_c and \hat{b}_r where the first part is possible to compute by preprocessing and the latter part is not. The problem can be rewritten in the new variables as

$$\min_{\hat{b}_c, \hat{b}_r} \frac{1}{2} \hat{b}_c^T H_{cc} \hat{b}_c - y_c^T A_{cc}^T \hat{b}_c + \hat{b}_c^T H_{cr} \hat{b}_r + \frac{1}{2} \hat{b}_r^T H_{rr} \hat{b}_r - y_r^T A_{rr}^T \hat{b}_r \quad (18)$$

where subindices c and r have been used to denote the components in \hat{b} that can and cannot be computed by preprocessing, respectively. These subindices are also used, analogously, to index out the corresponding parts in A , R , H and y .

Since it in an optimization problem always is possible to first minimize over some of the variables and then over the remaining ones, [17], it is here possible to first minimize over \hat{b}_c and then over \hat{b}_r . The optimal solution \hat{b}_c^* is then parameterized in \hat{b}_r , *i.e.*, the solution to the first optimization problem is in general a function $\hat{b}_c^*(\hat{b}_r)$, which can be hard to explicitly formulate in an integer optimization problem like this one. However, from the discussion below the equation in (17) it is clear that the preprocessing algorithm finds the optimal value of those variables that have an optimal value which is *independent* of the value of the remaining variables. Hence, \hat{b}_c^* is not dependent on \hat{b}_r , *i.e.*, $\hat{b}_c^*(\hat{b}_r) = \hat{b}_c^*$. After the solution \hat{b}_c^*

from the preprocessing algorithm has been inserted into the problem in (18), the remaining optimization problem in \hat{b}_r can be reformulated as

$$\min_{\hat{b}_r} \frac{1}{2} \hat{b}_r^T H_{rr} \hat{b}_r + \left(H_{cr}^T \hat{b}_c^* - A_{rr}^T y_r \right)^T \hat{b}_r \quad (19)$$

This is a new optimization problem in BQP form. Depending on the time available, the algorithm used after preprocessing may be chosen to produce either optimal solutions or suboptimal solutions. If an optimal solution is desired, existing optimization software like the commercial state-of-the-art branch and cut solver CPLEX, [25], or the freely available branch and bound solver `miqp.m`, [26], can be used. If the dimension of \hat{b}_r is low, which it often is after preprocessing, it is actually tractable to solve the problem in (19) by explicit enumeration of all solutions. If suboptimal solutions are considered sufficient, possible choices are to apply one of the detectors in (8) or (9). An important question to answer is to which MUD problem, *i.e.*, which \tilde{y} , \tilde{A} and \tilde{R} that defines the new problem, a detector should be applied in order to guarantee a jointly optimal decision in $\left[\hat{b}_c^T \ \hat{b}_r^T \right]^T$. This can be done by identifying the problem in (19) to be in the form in (7). The result after identification is

$$\tilde{y} = y_r - R_{cr}^T A_{cc} \hat{b}_c^*, \quad \tilde{A} = A_{rr}, \quad \tilde{R} = R_{rr} \quad (20)$$

Note that, this result also serves as a theoretical motivation for equation (15) in [13]. Consequently, an optimal detector working on the problem defined by the parameters in (20) will provide an optimal solution \hat{b}_r^* to the optimization problem in (19), and hence the total solution $\left[\hat{b}_c^T \ \hat{b}_r^T \right]^T$ will be jointly optimal. This holds independently of if \hat{b}_r^* is found as the solution to the optimization problem in (19) or if it is found as the optimally detected bit-sequence to an MUD problem with the parameters in (20).

Now, the choice of \tilde{y} in (20) will be slightly more investigated. Using the CDMA models presented in Section II, the matched filter output can be expressed as

$$y = \begin{bmatrix} y_c \\ y_r \end{bmatrix} = RAb + n = \begin{bmatrix} R_{cc} A_{cc} b_c + R_{cr} A_{rr} b_r + n_c \\ R_{cr}^T A_{cc} b_c + R_{rr} A_{rr} b_r + n_r \end{bmatrix} \quad (21)$$

which means that

$$\begin{aligned} \tilde{y} &= y_r - R_{cr}^T A_{cc} \hat{b}_c^* = R_{rr} A_{rr} \hat{b}_r + R_{cr}^T A_{cc} (b_c - \hat{b}_c^*) + n_r \\ &\triangleq R_{rr} A_{rr} \hat{b}_r + \tilde{n}_r \end{aligned} \quad (22)$$

where b_c and n_r denote the true values of the remaining bit-sequence and the noise in the output from the matched filters, respectively, and $\tilde{n}_r \in \mathcal{N} \left(R_{cr}^T A_{cc} (b_c - \hat{b}_c^*), \sigma^2 R_{rr} \right)$. The term $-R_{cr}^T A_{cc} \hat{b}_c^*$ has intuitively the function of cancelling the part of y_r resulting from bits that already have been detected in step one. However, it is not always true that $\hat{b}_c^* = b_c$ even though \hat{b}_c^* is optimal. In [13], there is a discussion about that errors made in the first step will affect the detector in the second step and as a result there is a risk that the detector in the second step might make "additional erroneous decisions". However, note that this offset is *necessary* in order for the detector in the second step to make a decision which makes the entire decision $\left[\hat{b}_c^T \ \hat{b}_r^T \right]^T$ *jointly optimal*. It would be possible to assume that $\hat{b}_c^* = b_c$ and work on a reduced MUD system in the second step with measurements y_r and correlation R_{rr} , but this setup would result in a decision which might not be jointly optimal when considered as one large decision.

V. NUMERICAL EXPERIMENTS

In this section, the preprocessing algorithm is applied to the multiuser detection problem as described in Section IV and its

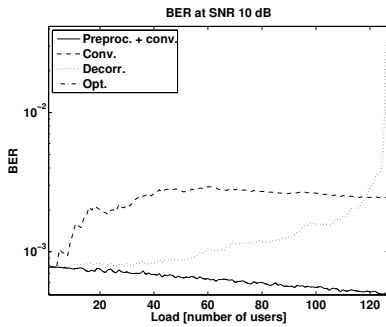


Figure 2. The plot shows the BER of the different detectors as a function of the load when Gold sequences of length 127 are used. It can be seen that the multiuser detector implemented by the preprocessing algorithm combined with the conventional detector gives the lowest BER. Note that, the BER performance of the proposed detector is almost indistinguishable from the BER performance of the optimal detector.

performance is evaluated using Monte Carlo simulations. The Signal to Noise Ratio (SNR) used in the simulations is computed after the matched filters. Hence, there are disturbances originating both from the noise $n(t)$ and from the cross-correlation from other users.

A. Synchronous Case

In the first simulation, the joint Bit Error Rate (BER) is compared for a multiuser detector implemented using a combination of the preprocessing algorithm and the conventional detector, the conventional detector itself in (8), the decorrelating detector in (9) and the optimal detector found by solving the optimization problem in (7) optimally. When the preprocessing algorithm is combined with a conventional detector, any variables not computed by the preprocessing algorithm is computed by the suboptimal detector (8) as described in Section IV-B3. Hence, if the preprocessing algorithm does not solve all variables, some parts of the solution might be suboptimal. To be able to obtain an optimal solution, the preprocessing algorithm is combined with CPLEX, where CPLEX is used to solve any remaining variables not possible to compute by preprocessing. The algorithms were compared for the loads 1 to 127 users. In the simulation, Gold sequences of length 127 were used and the SNR for user 1 was chosen to be 10 dB. The SNRs for the other users are then given by the choice of signature sequences and the noise variance. However, in this case, it is close to 10 dB for all users. To get a sufficiently smooth plot in a reasonable time, the number of Monte Carlo simulations used was adjusted in several steps from 10^7 for low loads to 10^5 for high loads. In each Monte Carlo simulation, a new noise realization was used and a new random bit was assigned to each user. The result from this test can be found in Figure 2. The conclusion from this test is that the BER performance of the detector built on a combination of preprocessing and the conventional detector is better than the performance of the conventional detector used alone or the decorrelating detector. In this test, the BER performance of the proposed detector is almost indistinguishable from the BER performance of the optimal detector.

In Figure 3 it can be seen that for a wide span of loads and SNRs, the preprocessing algorithm computes nearly all variables in average. Therefore, for those combinations of load and SNR, the remaining variables not computed by preprocessing do not alter the BER performance significantly even though they are computed suboptimally. In Figure 3 it can be seen that the number of variables computed by preprocessing decreases for high loads in combination with low SNR. Somewhat counterintuitive, the number of computed variables also decreases slightly for high loads in combination with

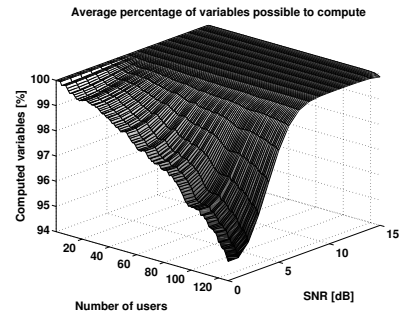


Figure 3. This plot shows how many percents of the variables that were computed by preprocessing for loads between 1 and 127 users in combination of SNRs between 0 to 15 dB. The result shows that even at 127 users and at an SNR as low as 0 dB, the preprocessing algorithm only fails to compute about 5% of the total number of variables.

high SNR. The effect of this degradation in detection performance will be discussed in coming experiments. The conclusion drawn from the simulation is that the preprocessing algorithm can be expected to compute almost all variables for a large span of combinations of SNRs and loads. However, for large loads in combination with low SNRs, the number of variables computed by preprocessing decreases.

In Figure 4 the BER is evaluated for different values of SNR in the range from 0 dB to 15 dB. In the comparison, five different detectors are used; a detector formed by the combination of the preprocessing algorithm and the conventional detector, a combination of the preprocessing algorithm and the decorrelating detector, the conventional detector, the decorrelating detector and finally the optimal detector. Note that, this experiment is also an excellent practical example of the computational performance of the algorithm. Actually, without the computational performance of the preprocessing algorithm, combined with the computational performance of CPLEX, it would have been impossible to perform such rigorous comparisons (a huge number of Monte Carlo simulations are used even at high loads) in a reasonable time where different suboptimal detectors are compared with the *optimal* detector. As expected, the BER performances of the conventional detector in (8) and the decorrelating in (9) are significantly worse compared to the other detectors, specifically for high values of SNR. The BER performance of the detectors built on preprocessing is so close to the optimal one that they cannot be visually distinguished from each other in this plot when SNR is 14 dB or below. Above 14 dB, the detector built on a combination of the preprocessing algorithm and the conventional detector shows a slightly higher BER than the other detectors. The behavior seems to be a consequence of the decrease of variables computed by preprocessing for a combination of high loads and high SNRs shown in Figure 3. This does not occur for loads below 117 users, and can be avoided also for high loads by combining the preprocessing algorithm with the decorrelating detector instead of the conventional detector for high SNRs.

The computational complexities for the different detectors have been previously theoretically discussed. An example of computational times in practice are shown in Figure 5. The conventional detector in (8) is not shown in the plot because its computational time is negligible in comparison with the other two. The conclusion drawn is that the computational complexity for the preprocessing algorithm grows similarly, or slower, compared to the one for the decorrelating detector in (9). Since there are several different ways to implement the different detectors, the computational times presented in Figure 5 should only be considered as guidelines to relate the performance of the different algorithms. For example, the matrix

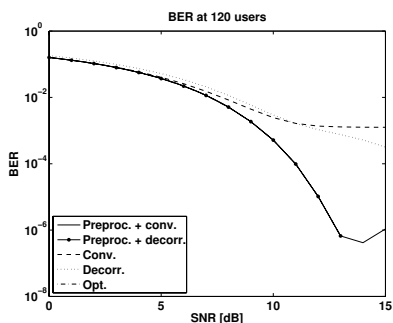


Figure 4. In this plot the BER is examined when SNR varies from 0 dB to 15 dB. The result is that at higher SNRs, the conventional and decorrelating detectors cannot offer BERs as low as the preprocessing based detectors. For SNRs below 13 dB, the resulting BERs from the preprocessing based detectors are that close to the optimal one that the BER curves cannot be visually separated.

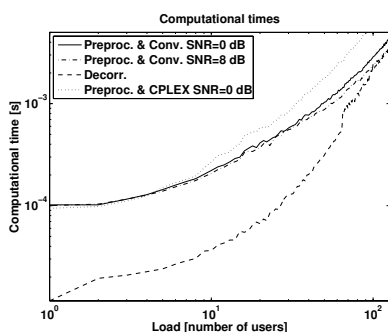


Figure 5. In this plot, the computational times to detect 127 users by the detector using the preprocessing algorithm, first in combination with the conventional detector, and second in combination with CPLEX, are compared with the computational times for the decorrelating detector. The conventional detector has significantly lower computational time and it has therefore been excluded from this plot. The conclusion is that the computational complexity of the preprocessing algorithm grows as, or slower than, the computational complexity of the decorrelating detector.

inversion performed in (9) is in Matlab implemented much more efficiently than the m-code implementation of the preprocessing algorithm. By implementing the preprocessing algorithm in, *e.g.*, C, a significant reduction of the computational time is expected. If the result in this experiment is combined with the result from previous experiments, the conclusion for the synchronous case is that the preprocessing algorithm combined with the conventional detector can produce a near optimal BER at a computational time comparable to the one of the decorrelating detector. Furthermore, in this case it is probably not worth the extra amount of computational time needed to compute the optimal solution, since that even preprocessing in combination with the conventional detector often gives similar BER performance. Anyhow, it is a good example how computational time can be decreased with an order of magnitude if the structure of the problem is used. The tests of the computational times were performed on a computer with two processors of the type Dual Core AMD Opteron 270 sharing 4GB RAM (the code was not written to utilize multiple cores) running CentOS release 4.6 (Final) Kernel 2.6.9-55.ELsmp and MATLAB 7.2.0.294.

From an optimization point of view it is interesting to investigate how far the suboptimal detectors are from being optimal. This is illustrated for 100 users in Figure 6, which shows the relative suboptimality here computed as $\frac{f_{subopt} - f^*}{|f^*|}$, where f_{subopt} denotes the objective function value for a suboptimal detector and f^* denotes

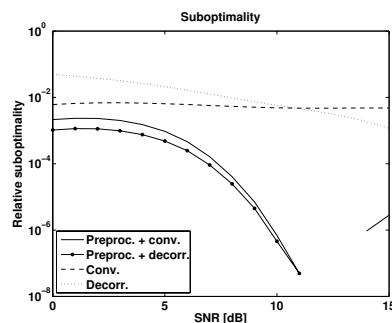


Figure 6. This plot shows the relative suboptimality for the different detectors computed as $\frac{f_{subopt} - f^*}{|f^*|}$, where f_{subopt} denotes the objective function value for a suboptimal detector and f^* denotes the optimal objective function value. Note that the detector is in fact optimal if the line vanishes from the plot. The line fragment from the combination of preprocessing and the conventional detector visible in the figure between 14 dB and 15 dB originates from the slight drop in detection performance of the preprocessing algorithm for a combination of high loads and large SNRs as shown in Figure 3.

the optimal objective function value. From the figure it can be seen that up to 13 dB, the combination of the preprocessing algorithm and the conventional detector gives relatively a significantly lower objective function value compared to the conventional or decorrelating detectors used alone. Above 13 dB, the performance of this combination decreases slightly, which can be explained by the drop in the by preprocessing detected number of users for a combination of high load and SNR as shown in Figure 3. To get near optimal performance also in this region, the preprocessing algorithm can be combined with the decorrelating detector.

B. Asynchronous Case

In the asynchronous case, the user are no longer assumed to be synchronized in time, *i.e.*, the channel model in (3) is used. The different offsets τ_i for different users are uniformly distributed between 0 and an upper value. Three experiments where this upper value have been chosen to 1, 5 and 64 chips have been considered. The result is presented in Figure 7 for the case when SNR is 5 dB. The conclusion from this experiment is that the cross-correlation between the users is rather dependent on the offsets of the users, and the benefits of the preprocessing algorithm tends to decrease as the maximum possible offset is increased. This is a consequence of the fact that fewer and fewer bits are computed in the preprocessing step as the maximum offset is increased. For low values of this offset, the algorithm is still usefull, while for larger values, the algorithm is only useful for very low loads.

VI. CONCLUSIONS

In this work, a preprocessing algorithm for BQP problems has been successfully applied to the multiuser detection problem when signature sequences of Gold type have been used. The preprocessing algorithm is able to detect some or all bits. These bits are detected optimally and any remaining bits can be detected either optimally or suboptimally by another algorithm. Numerical experiments have shown that if Gold sequences are used, more than 94.7% of the bits are computed by the preprocessing algorithm in the synchronous case. Furthermore, simulations have shown that the preprocessing algorithm combined with a suboptimal algorithm outperforms the suboptimal algorithm used alone in terms of BER and the resulting BER is often very close to the optimal one. Moreover, the computational complexity of the preprocessing algorithm is similar to

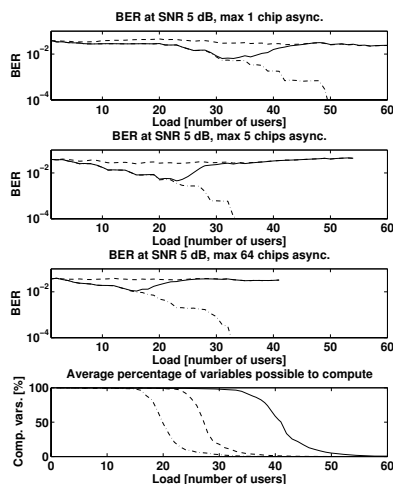


Figure 7. The three uppermost plots show the BER in an asynchronous system. These plots show three cases where different possible worst-case offsets τ_i among the users have been considered. The offsets are uniformly distributed between zero and this value. The BER of the conventional detector is dashed, the conventional detector in combination with preprocessing is solid, and the optimal is dash-dotted. In the bottommost plot, the number of variables computed by preprocessing is shown for the three different cases of maximum offset. Dash-dotted, dashed and solid lines represent a maximum offset of 64, 5 and 1 chips, respectively. In the cases where the offset is 5 and 64 chips, the experiment is aborted at a load lower than 60 users because it is not possible in these cases to maintain an SNR of 5 dB at high loads.

the one of the simple suboptimal decorrelating detector. The result in this work shows that it can be very advantageous to use the proposed preprocessing algorithm for multiuser detection problems where the cross-correlation between different users is low. Compared to several other suboptimal detectors, often a better BER performance can be expected. The benefits of using the proposed preprocessing algorithm decreases in the asynchronous case. In that case, the preprocessing algorithm is still able to compute a significant number of bits at low loads, but fewer and fewer with increased load. If users are almost synchronous, the preprocessing benefits are still evident, while in the worst case scenario, where users can be misaligned up to half a symbol, the benefits are minor already at moderate loads.

REFERENCES

- [1] P. H. Tan, "Multiuser detection in CDMA — combinatorial optimization methods," Licentiate's Thesis, Chalmers University of Technology, Nov. 2001.
- [2] K. Katayama and H. Narihisa, "Performance of simulated annealing-based heuristic for the unconstrained binary quadratic programming problem," *European Journal of Operational Research*, vol. 134, no. 1, pp. 103–119, Oct. 2001.
- [3] S. Ulukus and R. D. Yates, "Optimum multiuser detection is tractable for synchronous CDMA systems using M -sequences," *IEEE Communications Letters*, vol. 2, no. 4, pp. 89–91, Apr. 1998.
- [4] C. Sankaran and A. Ephremides, "Solving a class of optimum multiuser detection problems with polynomial complexity," *IEEE Transactions on Information Theory*, vol. 44, no. 5, pp. 1958–1961, Sep. 1998.
- [5] C. Schlegel and A. Grant, "Polynomial complexity optimal detection of certain multiple-access systems," *IEEE Transactions on Information Theory*, vol. 46, no. 6, pp. 2246–2248, Sep. 2000.
- [6] F. Hasegawa, J. Luo, K. R. Pattipati, P. Willett, and D. Pham, "Speed and accuracy comparison of techniques for multiuser detection in synchronous CDMA," *IEEE Transactions on Communications*, vol. 52, no. 4, pp. 540–545, Apr. 2004.
- [7] G. He, A. Kot, and T. Qi, "Decorrelator-based neighbour-searching multiuser detection in CDMA systems," *Electronics Letters*, vol. 32, no. 25, pp. 2307–2308, Dec. 1996.
- [8] J. Luo, K. R. Pattipati, P. Willett, and G. M. Levchuk, "Fast optimal and suboptimal any-time algorithms for CDMA multiuser detection based on branch and bound," *IEEE Transactions on Communications*, vol. 52, no. 4, pp. 632–642, Apr. 2004.
- [9] Y.-L. Li and Y. Lee, "A novel low-complexity near-ML multiuser detector for DS-SS and MC-SS systems," in *Proceedings of GLOBECOM'02 - IEEE Global Telecommunications Conference*, vol. 1, Nov. 2002, pp. 493–498.
- [10] T. J. Lim, L. K. Rasmussen, and H. Sugimoto, "An asynchronous multiuser CDMA detector based on the Kalman filter," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 9, pp. 1711–1722, Dec. 1998.
- [11] W. K. Ma, T. N. Davidson, K. Wong, Z. Q. Luo, and P. Ching, "Quasi-maximum-likelihood multiuser detection using semi-definite relaxation," *IEEE Transactions on Signal Processing*, vol. 50, no. 4, pp. 912–922, 2002.
- [12] J. Dahl, B. H. Fleury, and L. Vandenberghe, "Approximate maximum-likelihood estimation using semidefinite programming," in *IEEE International Conference on Acoustics, Speech, and Signal Processing 2003*, vol. 6, Apr. 2003, pp. VI–721–724.
- [13] P. Ödling, H. B. Eriksson, and P. O. Börjesson, "Making MLSD decisions by thresholding the matched filter output," *IEEE Transactions on Communications*, vol. 48, no. 2, pp. 324–332, Feb. 2000.
- [14] R. Nilsson, F. Sjöberg, O. Edfors, P. Ödling, H. Eriksson, S. K. Wilson, and P. O. Börjesson, "A low complexity threshold detector making MLSD decisions in a multiuser environment," in *Proceedings of the 48th IEEE Vehicular Technology Conference*, May 1998, pp. 333–337.
- [15] D. Axehill and A. Hansson, "A preprocessing algorithm for MIQP solvers with applications to MPC," in *Proceedings of the 43rd IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas, Dec. 2004, pp. 2497–2502.
- [16] S. Verdú, *Multiuser Detection*. Cambridge University Press, 1998.
- [17] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [18] M. Garey and D. Johnson, *Computers and Intractability: A guide to the Theory of NP-Completeness*. Freeman, 1979.
- [19] J. E. Beasley, "Heuristic algorithms for the unconstrained binary quadratic programming problem," Management School, Imperial College, UK, Tech. Rep., Dec. 1998.
- [20] P. Merz and B. Freisleben, "Greedy and local search heuristics for unconstrained binary quadratic programming," *Journal of Heuristics*, vol. 8, no. 2, pp. 197–213, Mar. 2002.
- [21] F. Glover, B. Alidaee, C. Rego, and G. Kochenberger, "One-pass heuristics for large-scale unconstrained binary quadratic problems," *European Journal of Operational Research*, vol. 137, no. 2, pp. 272–287, Mar. 2002.
- [22] P. L. Hammer and S. Rudeanu, *Boolean Methods in Operations Research and Related Areas*. Springer-Verlag, 1968.
- [23] D. Axehill, "Applications of integer quadratic programming in control and communication," Licentiate's Thesis, Linköpings universitet, 2005. [Online]. Available: <http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-5263>
- [24] J. G. Andrews, "Interference cancellation for cellular systems: A contemporary overview," *IEEE Wireless Communications Magazine*, pp. 19–29, Apr. 2005.
- [25] "ILOG CPLEX," Website, URL: <http://www.ilog.com/products/cplex>, accessed June 10, 2007.
- [26] A. Bemporad and D. Mignone, "A Matlab function for solving mixed integer quadratic programs version 1.02 user guide," Institut für Automatik, ETH, Tech. Rep., 2000.