ECE 273 Convex Optimization Project - Spring 2010

Quasi-maximum likelihood detection via convex optimization with application to asynchronous CDMA system

Bien Aik Tan
Ray Chu

1. INTRODUCTION

Multiuser detection (MUD) is the study of demodulating multiple user signals sharing a common multiple access channel. Multiuser detection for asynchronous code division multiple access (CDMA) system is necessary for acceptable performance. This is because, apart from the usual additive white Gaussian noise (AWGN) present in the channel, there is also cross user interference among the users better known as multiple access interference (MAI). Better performance is achieved if information about the multiple users is used jointly and the multiuser detector exploits this dependence between the users. The maximum-likelihood (ML) multiuser detector is well known to exhibit better bit-error-rate (BER) performance than many other multiuser detectors. Unfortunately, ML detection (MLD) is a nondeterministic polynomial-time hard (NP-hard) problem, for which there is no known algorithm that can find the optimal solution with polynomial-time complexity. That is, it's computational complexity increases exponentially, ($O(2^K)$), with the number of users, K. A polynomial-complexity detection means the running time is bounded by a polynomial function in only K and is denoted, for example, $O(K^2)$ or $O(K \log K)$. A comparison on the growth rates of several typical complexities is shown below.

No of Users	Complexity O()			
K	K log K	K^2	K^3	2^k
10	33	10^2	10^3	10^3
100	664	10^4	10^6	1.27×10^{30}
1000	9965	10^6	10^9	1.07×10^{301}

Table 1 Growth Rates of Some Polynomial Function and an Exponential Function

Hence, it is advantageous to look at polynomial-complexity algorithms for multiuser detection with some assumptions made on the system or the possibility of trading it off with bit error rate (BER) performance. However, the trade-off in BER performance cannot be too great such that the BER performance falls nearer to that of a single user detector. Assumptions can be made to simplify the MLD problem such as constant cross correlations and perfect power control [1], this results in polynomial-complexity optimum multiuser detection with complexity $O(K \log K)$. Another perspective of solving the MLD problem with polynomial-complexity around $O(K^2)$ is found in [2] and [3]. It is based on transforming a {0,1} quadratic programming problem into an equivalent problem of solving a minimum cut in a graph or network (in the realms of graph theory and algorithms). A minimum cut problem can be solved with polynomial-complexity if all the non-source and non-sink incident edge capacities are non-negative [4]. This result translates into designing a set of spreading sequences with the property that the cross correlation between users over each symbol period is non-positive. It relaxes the assumption of constant cross correlation stated previously for polynomial complex algorithm to allowing different cross correlation values as long as it is non-positive.

However, all the above imposed a limited set of codewords for the CDMA chip sequence. This is equivalent to a limited number of users in the MLD problem. It is known that the MLD, also known as a Boolean least square problem, is a non-convex quadratically constraint quadratic programming (QCQP) problem, [5]. Fortunately, a polynomial-time of near-optimal approach using semi-definite programming (SDP), approximately $O(K^{3.5})$, relaxation can be applied to the non-convex QCQP problem. This is elaborated in [5] and [6]. As a result, it further relaxes the cross correlation matrix constraint to only positive semi-definiteness. In this project, a study on SDP relaxation approach to the MLD problem, with a special application to asynchronous and arbitrary correlated CDMA signals will be done.

2. FORMULATING THE PROBLEM

2.1 BASIC ASYNCHRONOUS CDMA

Consider a simple model for an asynchronous CDMA system where only one packet is transmitted by each user at the time [7]. The basic asynchronous CDMA signal of *K* users comprises N bits that are antipodally modulated with spreading sequence waveforms and additive white Gaussian noise. It is expressed mathematically in Equation (1). It should be noted that this signal is continuous-time.

$$y(t) = \sum_{k=1}^{K} A_k \sum_{i=1}^{N} b_k(i) s_k(t - \tau_k - iT) + n(t) \qquad t \in [0, T], 0 \le \tau_k \le T \quad (1)$$

where

- T is the bit interval
- N is the total number of bits transmitted
- τ_k is the k^{th} user delay in an asynchronous setting
- $s_k(t)$ is the deterministic spreading sequence of length M chips assigned to the k^{th} user

$$\| s_k \|^2 = \int_0^T s_k^2(t)dt = 1 \tag{2}$$

- A_k is the received amplitude of the k^{th} user signal. A_k^2 is the energy of the k^{th} user.
- $b_k(\mathrm{i}) \in \{-1,+1\}$ is the antipodal i^{th} bit signal transmitted by the k^{th} user.
- *n(t)* is the white Gaussian noise with zero mean and uniform power spectral density of $\frac{N_0}{2}$.

Equation (2) simply states that the spreading sequence waveform has unit energy over [0,*T*]. Another parameter that is commonly used here is the cross-correlation of the spreading sequence waveforms (see Equation (3)). It quantifies the similarity between two spreading sequence waveforms.

$$\rho_{kl} = \langle s_k, s_l \rangle = \int_0^T s_k(t)s_l(t)dt \tag{3}$$

where

$$|\rho_{kl}| = |\langle s_k, s_l \rangle| \le \| s_k(t) \| \| s_l(t) \| = 1 \tag{4}$$

by the Cauchy Schwarz inequality.

The subscripts k and l denote which user the spreading sequence belongs to. If there are K users, then k and l can vary from 1 to K. When k and l are the same, the cross-correlation value is the energy of the sequence waveform which is equal to one. The cross correlation values are bounded by the limits from –1 to 1. It can be proven by using the property of the Cauchy Schwarz inequality shown in Equation (4) and the fact that all spreading sequence waveforms has unit energy. To display the cross correlation values in a systematic form, it can be organized in the form of a matrix where the diagonal elements are equal to one and is of K by K size. The matrix will be symmetric along the diagonal as ρ_{kl} and ρ_{lk} are the same (see Equation (5)).

$$\mathbf{R} = \begin{bmatrix} \rho_{11} & \rho_{12} & .. & .. & \rho_{1K} \\ \rho_{21} & \rho_{22} & . & . & : \\ : & . & . & . & : \\ : & . & . & . & \rho_{(K-1)K} \\ \rho_{K1} & .. & .. & \rho_{K(K-1)} & \rho_{KK} \end{bmatrix} \quad (5)$$

2.2 DISCRETE TIME ASYNCHRONOUS MODEL

In this section, we convert the continuous-time signal y(t) into the discrete form by correlating it with deterministic signals of spreading sequence waveforms and conventional sampling. This is done by passing y(t) through a bank of matched filters (see Figure 1) each matched to the spreading sequence waveform of different users.

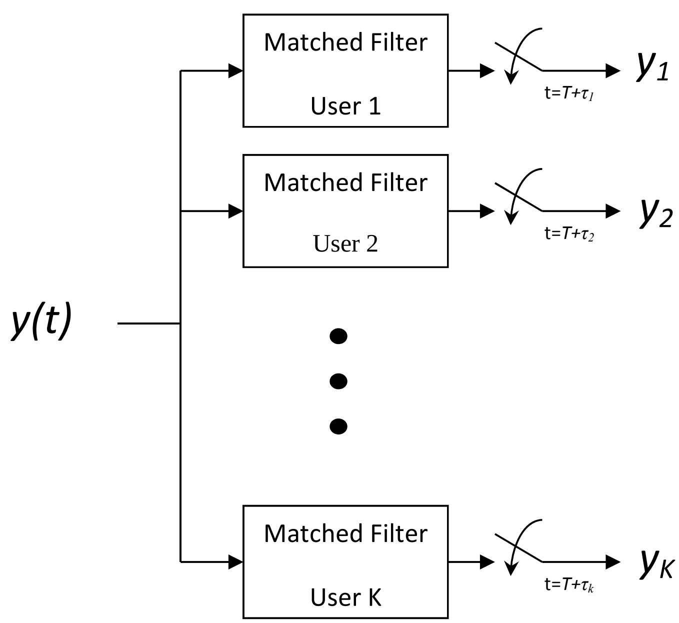

Figure 1 Matched Filtering in Asynchronous CDMA

The matched filter output for the k^{th} user is

$$y_k(i) = \int_{iT+\tau_k}^{(i+1)T+\tau_k} y(t)s_k(t - iT + \tau_k)dt, \qquad 1 \le i \le N \tag{6}$$

Equation (6) can be expressed as a form of linear Gaussian vector shown below.

$$\mathbf{y} = \mathbf{R}_N \mathbf{A}\mathbf{b} + \mathbf{n} \tag{7}$$

where

$$\begin{aligned}
&\mathbf{y} = [\mathbf{y}^{\mathrm{T}}(1), \ldots\ldots\ldots\ldots, \mathbf{y}^{\mathrm{T}}(N)]^{\mathrm{T}} \\
&\mathbf{y}(i) = [y_1(i), \ldots\ldots\ldots\ldots, y_K(i)]^{\mathrm{T}} \\
&\mathbf{R}_N = \begin{bmatrix} \mathbf{R}_a(0) & \mathbf{R}_a^T(1) & \mathbf{0} & .. & .. & \mathbf{0} \\ \mathbf{R}_a(1) & \mathbf{R}_a(0) & \mathbf{R}_a^T(1) & \mathbf{0} & .. & \mathbf{0} \\ : & : & : & : & : & : \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{R}_a(1) & \mathbf{R}_a(0) & \mathbf{R}_a^T(1) \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{R}_a(1) & \mathbf{R}_a(0) \end{bmatrix} \text{cross correlation matrix} \\
&\mathbf{A} = diag[A_{11}, \ldots\ldots\ldots\ldots, A_{1K}, \ldots, A_{N1}, \ldots\ldots\ldots\ldots, A_{NK}] \\
&\mathbf{b} = [\mathbf{b}^{\mathrm{T}}(1), \ldots\ldots\ldots\ldots, \mathbf{b}^{\mathrm{T}}(N)]^{\mathrm{T}} \\
&\mathbf{b}(i) = [b_1(i), \ldots\ldots\ldots\ldots, b_K(i)]^{\mathrm{T}} \\
&\mathbf{n} = [\mathbf{n}^{\mathrm{T}}(1), \ldots\ldots\ldots\ldots, \mathbf{n}^{\mathrm{T}}(N)]^{\mathrm{T}} \\
&\mathbf{n}(i) = [n_1(i), \ldots\ldots\ldots\ldots, n_K(i)]^{\mathrm{T}}
\end{aligned} \tag{8}$$

and $\mathbf{R}_a(m)$ is a K x K matrix with elements

$$R_{kl}(m) = \int_{-\infty}^{\infty} s_k(t - \tau_k)s_k(t - mT + \tau_l)dt, \qquad 1 \le i \le N \tag{9}$$

Also, $\mathbf{n}(i)$ is a zero mean Gaussian random vector with covariance matrix of

$$\mathrm{E}[\mathbf{n}(k)\mathbf{n}^{\mathrm{T}}(j)] = \frac{N_0}{2}\mathbf{R}_a(k - j) \tag{10}$$

2.3 MAXIMUM LIKELIHOOD MULTIUSER DETECTION

The maximum likelihood estimate for asynchronous CDMA can be expressed as

$$\widehat{\mathbf{b}}_{ML} = \underset{\mathbf{b}\in\{-1,+1\}^{NK}}{\arg\max}\, p_{\mathbf{y}|\mathbf{b}}(\mathbf{y} \mid \mathbf{b}) = \underset{\mathbf{b}\in\{-1,+1\}^{NK}}{\arg\max}\left[2\mathbf{y}^{\mathrm{T}}\mathbf{A}\mathbf{b} - \mathbf{b}^{\mathrm{T}}\mathbf{A}\mathbf{R}_N\mathbf{A}\mathbf{b}\right] \tag{11}$$

Because the correlation matrix is now *NK* x *NK* in the asynchronous CDMA case, it has an maximum likelihood estimate order of complexity of $O(2^{NK})$ instead of $O(2^{K})$ in the synchronous CDMA case. Without loss of generality, we can assume N is equal to one to reduce the complexity of the simulation. The order of complexity is now similar to that of synchronous case. However, asynchronous CDMA's correlation matrix can change randomly, due to random delays for each user, when compared to the fixed correlation matrix in synchronous case. Hence, the polynomial complexity discussed in [1] through [4] cannot be used for random cross correlations in asynchronous CDMA. At this point, we have derived a non-convex QCQP problem, specifically known as the Boolean least square problem. In the next section, we will discuss several interpretations of this problem.

3 RE-FORMULATING THE PROBLEM

3.1 SDP RELAXATION OF THE BOOLEAN LS PROBLEM

We can rewrite equation (11) as,

$$\widehat{\mathbf{b}}_{ML} = \arg\min_{\mathbf{b}\in\{-1,+1\}^K}\left[-2\mathbf{y}^{\mathrm{T}}\widetilde{\mathbf{A}}\mathbf{b} + \mathbf{b}^{\mathrm{T}}\widetilde{\mathbf{A}}\mathbf{R}_a\widetilde{\mathbf{A}}\mathbf{b}\right] \tag{12}$$

where

$$\widetilde{\mathbf{A}} = diag[A_1,............., A_K]$$

By substituting $\mathbf{b} = t\mathbf{x}$ into equation (12) where t is a scalar, we can rewrite equation (12) as

$$\begin{aligned} p^* = \min_{\mathbf{x},t}\ & -2\mathbf{y}^{\mathrm{T}}\widetilde{\mathbf{A}}t\mathbf{x} + \mathbf{x}^{\mathrm{T}}\widetilde{\mathbf{A}}\mathbf{R}_a\widetilde{\mathbf{A}}\mathbf{x} \\ \text{s.t.}\ & \mathrm{x}_{\mathrm{i}}^2 = 1,\ \ \mathrm{i} = 1,...,\mathrm{K} \\ & \mathrm{t}^2 = 1 \end{aligned} \tag{13}$$

where p^* is the minimum value of the objective function and $\mathbf{x}^*$ and t^* is the optimal value for the minimum objective function value. Let $\mathbf{w} = [\mathbf{x}^T, t]^T \in \{\pm 1\}^{K+1}$. Then equation (13) can be reformulated as a homogenous QCQP as follows:

$$\begin{aligned} \min_{\mathbf{w}}\ & \mathbf{w}^T\mathbf{Q}\mathbf{w} \\ \text{s.t.}\ & w_{\mathrm{i}}^2 = 1,\ \ \mathrm{i} = 1,..., K+1 \end{aligned} \tag{14}$$

where
$$\mathbf{Q} = \begin{bmatrix} \mathbf{A}\mathbf{R}_a\mathbf{A} & -(\mathbf{A}\mathbf{y}) \\ -(\mathbf{A}\mathbf{y})^T & 0 \end{bmatrix}$$

Let $\mathbf{W} = \mathbf{w}\mathbf{w}^T$. This fundamentally implies that $\mathbf{W} \succeq 0$ and $rank(\mathbf{W}) = 1$. Hence, we can linearize equation (14) and rewrite as

$$\begin{aligned} \min_{\mathbf{y}} \ & \mathrm{Tr}(\mathbf{QW}) \\ \text{s.t.} \ & \mathrm{W}_{\mathrm{ii}} = 1, \ \mathrm{i} = 1, \ldots, \mathrm{K}+1 \\ & \mathbf{W} = \mathbf{w}\mathbf{w}^T \end{aligned} \tag{15}$$

However, we note that the implicit constraint $rank(\mathbf{W}) = 1$ in equation (15) is non convex. The relaxation of equation (15) is obtained by dropping the non-convex $rank(\mathbf{W}) = 1$ constraint and rewrite it in the following SDP:

$$\begin{aligned} \min_{\mathbf{y}} \ & \mathrm{Tr}(\mathbf{QW}) \\ \text{s.t.} \ & \mathrm{W}_{\mathrm{ii}} = 1, \ \mathrm{i} = 1, \ldots, \mathrm{K}+1 \\ & \mathbf{W} \succeq 0 \end{aligned} \tag{16}$$

It is important to note the effect of relaxation. The relaxed SDP gives a lower bound on the optimal objective value, specifically, $p^*_{sdp} \leq p^*_{qcqp}$. It will be evident in Section 3.3 that equation (16) gives the same lower bound as the Lagrangian dual of equation (14) because it can be shown that the SDP relaxation equation (16) is the bi-dual or dual-dual of homogenous QCQP equation (14). Therefore, from a non-convex QCQP problem, through relaxation, a convex SDP problem is formulated. Thus the SDP problem can be efficiently solved by interior point methods in polynomial time [8].

3.2 OBTAINING AN APPROXIMATE RANK-1 SOLUTION VIA GAUSSIAN SAMPLING

In [5, Section 4.2.2], it proposes a Gaussian sampling approach to extract the rank-1 component from the $\mathbf{W}^*$ from the SDP problem that will serve as a good approximate solution to the non-convex QCQP. A probabilistic interpretation of the relaxation equation (16) is to treat $\mathbf{w}^*$ (i.e. $\mathbf{w}^* = \mathbf{E}[\mathbf{w}]$) and $\mathbf{W}^*$ (i.e. $\mathbf{W}^* = \mathbf{E}[\mathbf{w}\mathbf{w}^\mathbf{T}]$) as the first and second moments of a random vector $\mathbf{w}$. It can be shown that

$$\min_{\mathbf{w}} \ \mathbf{E}[\mathbf{w}^T \mathbf{Q} \mathbf{w}] \tag{17}$$
$$\text{s.t.} \ \ \mathbf{E} w_{\mathrm{i}}^2 = 1, \ \ \mathrm{i} = 1, ..., K+1$$

is equivalent to the problem define in equation (14) where we minimize over all possible probability distributions of $\mathbf{w}$. This interpretation suggests another heuristic method for computing suboptimal solutions of (13) based on the result of (16). Let $\mathbf{W}^*$ be the optimum point for (16). We generate a set of random vectors $\zeta_l, l = 1, ..., M_{rand}$ from the Gaussian distribution $N(\mathbf{0}, \mathbf{W}^*)$, and quantize them into the binary vector $\hat{\mathbf{w}}_l = sign(\zeta_l) \in \{\pm 1\}^{K+1}, l = 1, ..., M_{rand}$. A best approximate solution of (14) can be obtained as

$$\hat{\mathbf{w}} = \underset{\{\hat{\mathbf{w}}_l\}, l=1,...,M_{rand}}{\arg\min} \ \hat{\mathbf{w}}_l^{\mathbf{T}} \mathbf{Q} \hat{\mathbf{w}}_l \tag{18}$$

Let $\hat{\mathbf{w}} = [\hat{\mathbf{x}}^T, \hat{t}]^T$. Finally, the best associate approximate solution to (13) is given by $\mathbf{b}^* = \hat{t}\hat{\mathbf{x}}$. It is empirically found that M_{rand} >10 is sufficient to obtain near-optimum approximation performance. Although this random sampling is performed multiple times, it should be noted that it is still polynomial in time.

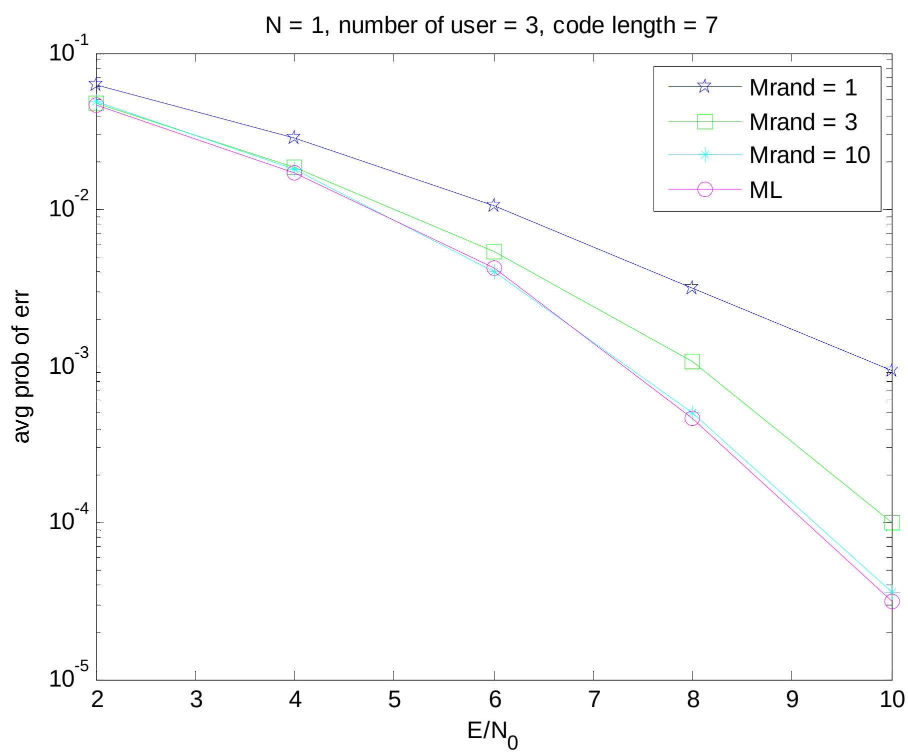

Figure 2 BER simulation of SDP Detector and ML Detector versus SNR in Decibels for various $\mathbf{M_{rand}}$

3.3 LAGRANGE DUAL and BI-DUAL ANALYSIS

The dual problem of (14) can be obtained as:

$$\begin{aligned} g(\mathbf{v}) &= \min_{\mathbf{v}} \mathbf{w}^T\mathbf{Q}\mathbf{w} + \sum_{i=1}^{K+1} v_i\left(w_i^2 - 1\right) \\ &= \min_{\mathbf{v}} \mathbf{w}^T\mathbf{Q}\mathbf{w} + \mathbf{w}^T diag(\mathbf{v})\mathbf{w} - \mathbf{1}^T\mathbf{v} \\ &= \min_{\mathbf{v}} \mathbf{w}^T\left(\mathbf{Q} + diag(\mathbf{v})\right)\mathbf{w} - \mathbf{1}^T\mathbf{v} \\ &= \begin{cases} -\mathbf{1}^T\mathbf{v} & \left(\mathbf{Q} + diag(\mathbf{v})\right) \succeq 0 \\ -\infty & \text{otherwise} \end{cases} \end{aligned} \tag{19}$$

$$\begin{aligned} d^* &= \max_{\mathbf{v}}\ g(\mathbf{v}) \\ &= \max_{\mathbf{v}}\ -\mathbf{1}^T\mathbf{v} \\ &\quad \text{s.t.}\ \ \mathbf{Q} + diag(\mathbf{v}) \succeq \mathbf{0} \end{aligned} \tag{20}$$

The Lagrange dual function of (20) can be obtained as follow:

$$\begin{aligned} g(\Gamma) &= \max_{v} -\mathbf{1}^T\mathbf{v} + \mathrm{Tr}(\mathbf{\Gamma}(\mathbf{Q} + diag(\mathbf{v}))) \\ &= \max_{v} -\mathbf{1}^T\mathbf{v} + \mathrm{Tr}(\mathbf{\Gamma} diag(\mathbf{v})) + \mathrm{Tr}(\mathbf{\Gamma}\mathbf{Q}) \\ &= \max_{v} \sum_{i=1}^{K+1} (\Gamma_{ii} - 1)v_i + \mathrm{Tr}(\mathbf{\Gamma}\mathbf{Q}) \\ &= \begin{cases} \mathrm{Tr}(\mathbf{\Gamma}\mathbf{Q}) & \mathit{for}\ \Gamma_{ii} = 1, i = 1,\ldots,K+1 \\ \infty & \text{otherwise} \end{cases} \end{aligned} \tag{21}$$

From (21), one can formulate the dual problem of (20) as

$$\begin{aligned} &\min_{\mathbf{y}}\ \mathrm{Tr}(\mathbf{Q}\mathbf{\Gamma}) \\ &\text{s.t.}\ \ \Gamma_{ii} = 1,\ \ i = 1,\ldots,K+1 \\ &\qquad \mathbf{\Gamma} \succeq \mathbf{0} \end{aligned} \tag{22}$$

Comparing equation (22) with equation (16), we see that equation (16) is actually the bi-dual of equation (14), which, according to duality theorem, always forms a lower bound of the primal problem. Also, because a dual problem is always convex, we can say equation (22) is a convex relaxation of equation (14). Furthermore, if the optimum point $\mathbf{\Gamma}^*$ has rank one, it is also the optimum point for the primal problem.

4 COMPARING WITH OTHER SUBOPTIMAL DETECTORS

In this section, we consider the unconstrained relaxation (UR) of the original problem, where detectors derived from this are known to be suboptimal detectors. It will be illustrated, both analytically and through simulation that the unconstrained relaxation is a further relaxation of the SDP relaxation method. Therefore, it is expected that the SDP-detector will perform better than these sub-optimal detectors. Consider the following unconstrained relaxation:

$$\tilde{\mathbf{b}}_{UR} = \arg\min_{\mathbf{b}\in \mathrm{R}^K} \left[-2(\mathbf{Ay})^{\mathrm{T}}\mathbf{b} + \mathbf{b}^{\mathrm{T}}\mathbf{Hb}\right] + \gamma\|\mathbf{b}\|^2 \tag{23}$$

where $\mathbf{H} = \mathbf{ARA}$, $\gamma\|\mathbf{b}\|^2$ is the penalty function with $\gamma \geq 0$. The reason for choosing such a penalty function is to implicitly constrain the magnitude of $\mathbf{b}$ while maintaining the least-square nature of the relaxation problem. The major advantage of using UR is the availability of a closed-form solution. Assuming that $\mathbf{H} + \gamma\,\mathbf{I}$ is invertible, the solution to (23) is given by

$$\tilde{\mathbf{b}}_{UR} = (\mathbf{H} + \gamma\,\mathbf{I})^{-1}\mathbf{Ay} \tag{24}$$

And it can be shown that

$$\hat{\mathbf{b}}_{UR} = sign((\mathbf{H} + \gamma\,\mathbf{I})^{-1}\mathbf{Ay}) = sign((\mathbf{R} + \gamma\,\mathbf{A}^{-2})^{-1}\mathbf{y}) \tag{25}$$

To see how the penalized UR method is related to some of the existing suboptimal detectors, we consider the outputs of three well-known linear detectors: the matched filter (MF) detector, the decorrelator (DC), and the LMMSE detector, which are given, respectively, by

$$\hat{\mathbf{b}}_{MF} = sign(\mathbf{y}) \tag{26}$$

$$\hat{\mathbf{b}}_{DC} = sign(\mathbf{R}^{-1}\mathbf{y}) \tag{27}$$

$$\hat{\mathbf{b}}_{LMMSE} = sign((\mathbf{R} + \frac{N_0}{2}\mathbf{A}^{-2})^{-1}\mathbf{y}) \tag{28}$$

Comparing equation (25) with equation (26) through (28), the equivalence between the UR detectors can be clearly seen. For $\gamma = 0$, $\hat{\mathbf{b}}_{UR} = \hat{\mathbf{b}}_{DC}$. If γ is chosen to be $\frac{N_0}{2}$, then $\hat{\mathbf{b}}_{UR} = \hat{\mathbf{b}}_{LMMSE}$. Finally, if we choose $\gamma >> \max E_k$, the approximate solution $\hat{\mathbf{b}}_{UR}$ approaches

$sign((\gamma\mathbf{A}^{-2})^{-1}\mathbf{y}) = sign(\mathbf{y}) = \hat{\mathbf{b}}_{MF}$. Next, consider the tighter problem with bound constrain than equation (23):

$$\begin{aligned} \tilde{\mathbf{b}}_{BR} &= \arg\min_{\substack{-d \le \mathbf{b}_i \le d \\ i=1,\dots,K}} \mathbf{b}^{\mathrm{T}}\mathbf{H}\mathbf{b} - 2(\mathbf{A}\mathbf{y})^{\mathrm{T}}\mathbf{b} + \gamma\|\mathbf{b}\|^2 \\ \hat{\mathbf{b}}_{BR} &= sign(\tilde{\mathbf{b}}_{BR}) \end{aligned} \tag{29}$$

where d ≥ 1.

By following the same procedure in above section, (29) can be rewritten as

$$\begin{aligned} h^* &= \min_{\substack{-d \le \mathbf{b}_i \le d \\ i=1,\dots,K}} \mathbf{b}^{\mathrm{T}}(\mathbf{H} + \gamma\,\mathbf{I})\mathbf{b} - 2(\mathbf{A}\mathbf{y})^{\mathrm{T}}\mathbf{b} \\ &= \min_{\substack{-d \le x_i \le d \\ i=1,\dots,K \\ x_{K+1}^2 = 1}} \mathbf{x}^{\mathbf{T}}\tilde{\mathbf{Q}}\mathbf{x} \\ &= \min_{\mathbf{X},\mathbf{x}} Tr(\mathbf{X}\tilde{\mathbf{Q}}) \\ &\quad s.t.\ \ \mathbf{X} = \mathbf{x}\mathbf{x}^T \\ &\qquad\quad X_{ii} \le d^2 \\ &\qquad\quad X_{K+1,K+1} = 1 \end{aligned} \tag{30}$$

where $\tilde{\mathbf{Q}} = \begin{bmatrix} \mathbf{H} + \gamma\,\mathbf{I} & -(\mathbf{A}\mathbf{y}) \\ -(\mathbf{A}\mathbf{y})^T & 0 \end{bmatrix}$.

We relax equation (30) as

$$\begin{aligned} l^* &= \min_{\mathbf{X}} Tr(\mathbf{X}\tilde{\mathbf{Q}}) \\ &\quad s.t.\ \ \mathbf{X} \succeq \mathbf{0} \\ &\qquad\quad X_{ii} \le d^2 \\ &\qquad\quad X_{K+1,K+1} = 1 \end{aligned} \tag{31}$$

Now we will prove (30) and (31) are equivalent problem. First, because (31) is the relaxation of (30),

$$l^* \le h^* \tag{32}$$

Let $\mathbf{X} = \begin{bmatrix} \mathbf{Z} & \mathbf{z} \\ \mathbf{z} & 1 \end{bmatrix}$, (31) can be reformulated as

$$l^* = \min_{\mathbf{Z},\mathbf{z}} Tr(\mathbf{Z}(\mathbf{H}+\gamma\,\mathbf{I})) - 2\mathbf{z}^T(\mathbf{Ay}) \tag{33}$$
$$s.t.\ \mathbf{Z}-\mathbf{zz}^T \succeq \mathbf{0}$$
$$Z_{ii} \le d^2\ , i=1,...,K$$

Since $\mathbf{Z}-\mathbf{zz}^T \succeq \mathbf{0}$ and $\mathbf{H}+\gamma\,\mathbf{I} \succeq \mathbf{0}$, we have

$$Tr((\mathbf{Z}-\mathbf{zz}^T)(\mathbf{H}+\gamma\,\mathbf{I})) \ge 0 \quad \Rightarrow \quad Tr(\mathbf{Z}(\mathbf{H}+\gamma\,\mathbf{I})) \ge \mathbf{z}^T(\mathbf{H}+\gamma\,\mathbf{I})\mathbf{z} \tag{34}$$

for any feasible $\mathbf{Z}$ and $\mathbf{z}$.

And from $\mathbf{Z}-\mathbf{zz}^T \succeq \mathbf{0}$ *and* $Z_{ii} \le d^2\ , i=1,...,K$, we have $z_i \le d^2\ , i=1,...,K$ (35).

Let $\mathbf{Z}^*$ *and* $\mathbf{z}^*$ be the optimum points for (33). From (34) and (35), we have

$$\begin{aligned} l^* &= Tr(\mathbf{Z}^*(\mathbf{H}+\gamma\mathbf{I})) - 2\mathbf{z}^{*T}(\mathbf{Ay}) \\ &\ge \mathbf{z}^{*T}(\mathbf{H}+\gamma\mathbf{I})\mathbf{z}^* - 2\mathbf{z}^{*T}(\mathbf{Ay}) \\ &\ge \min_{\substack{-d\le z_i\le d\\ i=1,...,K}} \mathbf{z}^{\mathrm{T}}(\mathbf{H}+\gamma\mathbf{I})\mathbf{z} - 2(\mathbf{Ay})^{\mathrm{T}}\mathbf{z} = h^* \end{aligned} \tag{36}$$

From (32) and (36), we conclude $l^* = h^*$

Therefore, from the first two lines of (36), it can seen that

$$\begin{aligned} & Tr(\mathbf{Z}^*(\mathbf{H}+\gamma\mathbf{I})) = \mathbf{z}^{*T}(\mathbf{H}+\gamma\mathbf{I})\mathbf{z}^* = Tr(\mathbf{z}^*\mathbf{z}^{*T}(\mathbf{H}+\gamma\mathbf{I})) \\ & \Rightarrow Tr((\mathbf{Z}^*-\mathbf{z}^*\mathbf{z}^{*T})(\mathbf{H}+\gamma\mathbf{I})) = 0 \\ & \Rightarrow (\mathbf{Z}^*-\mathbf{z}^*\mathbf{z}^{*T})(\mathbf{H}+\gamma\mathbf{I}) = 0 \\ & \Rightarrow (\mathbf{Z}^*-\mathbf{z}^*\mathbf{z}^{*T}) = 0 \\ & \Rightarrow \mathbf{Z}^* = \mathbf{z}^*\mathbf{z}^{*T} \end{aligned} \tag{37}$$

Therefore, we conclude (30) and (31) are equivalent problem. Comparing (31) with the SDP relaxation problem (16), due to problem (14) is non-sensitive to the diagonal element of $\mathbf{H}$, we can reformulate (16) as

$$\min_{\mathbf{y}}\ \mathrm{Tr}(\tilde{\mathbf{Q}}\mathbf{Y}) \tag{38}$$
$$\text{s.t.}\ \ \mathrm{Y_{ii}}=1,\ \ \mathrm{i}=1,...,\mathrm{K}+1$$
$$\mathbf{Y}=\mathbf{yy}^T$$

where $\tilde{\mathbf{Q}}$ is defined in (30). Compare (38) with (31), we can conclude the SDP relaxation problem is tighter than problem (30). Therefore, it is expected the performance of SDP relaxation detector is better than the other suboptimal detectors listed in (26)-(28). The simulation result is shown in Figure 3. It shows that even though DC, LMMSE and MF are polynomial in complexity, $O(K^3)$, it sacrifices or traded it off from the BER performance. Figure 3 also shows that the SDP performed better than these sub-optimal detectors. For example at 10^{-3} BER, the SDP/ML is 1dB better than the LMMSE/DC detectors. The MF detector performs the worst as it is a single user detector in a multiuser setting. That is, the MAI is not cancelled off in the MF detector.

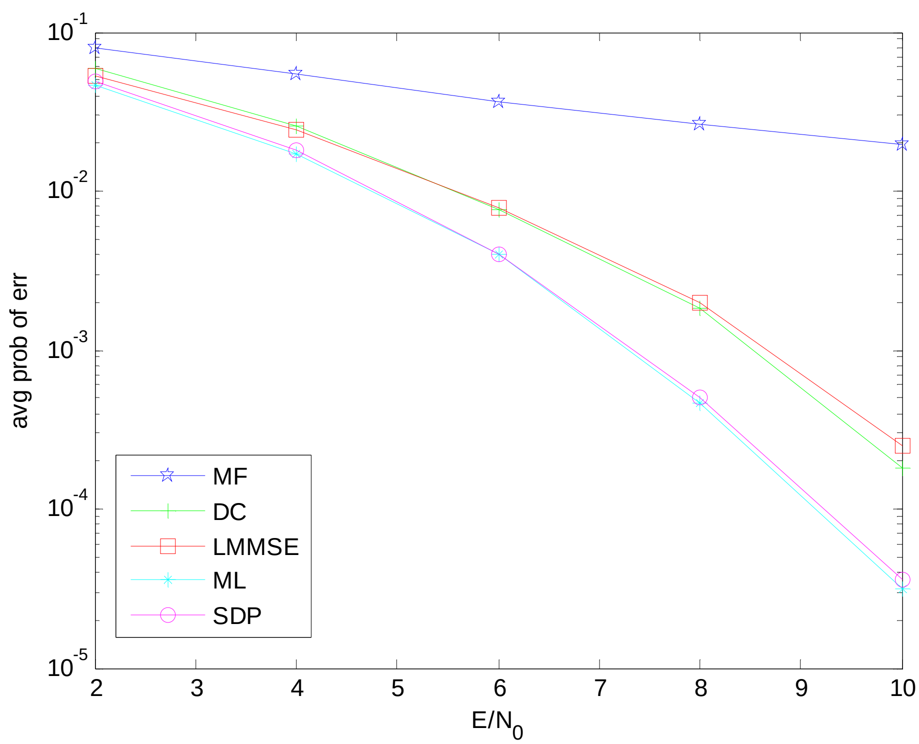

Figure 3 Comparison of SDP, ML and UR detectors

5 DISCUSSION/CONCLUSION

In this project, we have considered various polynomial-complexity algorithms for the asynchronous CDMA multiuser detection problem where the optimum detector is the ML detector. However, the MLD is known for its exponential complexity as it is a NP hard problem. Interestingly, prior to the application of convex optimization to the MLD problem, some of the polynomial algorithms proposed could achieve optimum performance but places strict constraints on the correlation matrix. These are done through clever linear algebra manipulations and transforming the quadratic problem to a minimum cut problem in graph theory. Other polynomial complexity detectors such as the MF, DC and LMMSE detectors traded it off from the BER performance, hence they are known to be sub-optimal detectors. However, it is noted that the asynchronous CDMA problem is a non-convex QCQP problem, specifically known as the Boolean LS problem. Through rank-1 relaxation, the non-convex QCQP problem can be converted into a convex SDP problem. The optimum solution of the SDP problem can then be used to approximate the rank-1 solution of the original QCQP problem. Some simulation has been performed to assess the efficacy of the SDP relaxation approach and it is found that the SDP relaxation method is highly effective, reliable and efficient, yielding excellent near-optimal results. We approximate the rank-1 solution of the original problem through Gaussian sampling and found that only 10 randomizations or more are required to produce near-optimum results. Lastly, we showed that unconstrained relaxation (UR) is a further relaxation of the SDP relaxation method. Hence, the SDP-detector is expected to perform better than these UR based sub-optimal detectors.

6 REFERENCES

1. Schlegel, C. and A. Grant, *Polynomial complexity optimal detection of certain multiple-access systems.* IEEE Transactions on Information Theory, Sept. 2000. **46**(6): p. 2246 -2248.
2. Sankaran, C. and A. Ephremides, *Solving a Class of Optimum Multiuser Detection Problems with Polynomial Complexity.* IEEE Transactions on Information Theory, 1998. **44**(5): p. 1958-1961.
3. Ulukus, S. and R.D. Yates, *Optimum Multiuser Detection is Tractable for Synchronous CDMA Systems Using M-Sequences.* IEEE Communications Letters, 1998. **2**(4): p. 89-91.
4. Motwani, R. and P. Raghavan, *Randomized Algorithms*. . 1995, New York: Cambridge University Press.
5. Palomar, D.P. and Y.C. Eldar, *Convex Optimization in Signal Processing and Communications*. 2010, New York: Cambridge University Press.
6. Ma, W.-K., et al., *Quasi-Maximum-Likelihood Multiuser Detection Using Semi-Definite Relaxation With Application to Synchronous CDMA.* IEEE Trans. on Signal Processing, 2002. **50**(4): p. 912-922.
7. Proakis, J.G., *Digital Communications*. 4th ed. 2001, New York: Mcgraw Hill.
8. Helmberg, C., et al., *An Interior-point Method for Semidefinite Programming.* SIAM Journal of Optimization, 1996. **6**(2): p. 342-361.

7 MATLAB CODES

Section 3 Matlab Codes

```
clc;clear;close all

No_user=3;  %no of users
N=7;        %no of chips or spreading factor
dlen=10000;     %data length
c = zeros(No_user,N); % code
Eb=1;                   %Energy per bit
SNRdb=2:2:10;          %string of SNR in dB
No_SNRs=length(SNRdb);  %Find out the length of SNRdb string
err=zeros(1,No_user);   %no of errors bits
uber = zeros(No_SNRs,No_user); % ber for each user vs given SNR
dlen_cnt=0;                     %no of dlen sent
Mrand = 100;  % number of random variable generated
% assign spreading sequence for each user
c(1,:) = [-1 -1 -1 1 1 -1 1];
c(2,:) = [1 1 -1 -1 -1 1 -1];
c(3,:) = [-1 1 1 -1 -1 -1 1];
%diagonal matrix A for the matrix of received amplitude--------------------
-----------------------
A=zeros(1,No_user);
A(:)=1;
A=diag(A);

% simulation
for snr_cnt=1:No_SNRs
    dlen_cnt=0;                  %counter for no of time the while loop
below till error bit>100
    err(:)=0;                    %error bit counter for K users
    while ((sum(err)/No_user < 100) | (dlen_cnt<10000))
        SNR=(10^(SNRdb(snr_cnt)/10)); %Signal to noise ratio
        sd=sqrt((Eb/SNR)/2);          % standard deviation
        b = 2*randint(No_user,1)-1;       % information bits
        P = partCorgen(c);              % partial cross-correlation matrix
        n = sd*mvnrnd(zeros(1,No_user),P);%(R_half)*(sd*randn(No_user,1)); %
noise with correlation matrix N0/2 * R
        y = P*A*b + n';                 % test stat
        % Multiuser Detection using conventional SDP
        %temp = A*P*R_inv;
        Q = [A*P*A , -A*y ; -(A*y)' , 0];
        cvx_begin sdp quiet
            cvx_solver sdpt3
            variable Y(No_user+1,No_user+1) symmetric
            minimize ( trace(Q*Y) )
            diag(Y) == 1;
            Y >= 0;
        cvx_end
        if min(eig(Y) < 0) % make sure Y is positive semidefinite
            continue;
        end

        for i = 1:Mrand
            y_hat = (sign(mvnrnd(zeros(1,No_user+1),Y)))';
            yQy = y_hat'*Q*y_hat;
```

```
            if i==1
                yQy_min = yQy;
                y_star = y_hat;
            else    if yQy < yQy_min
                    yQy_min = yQy;
                    y_star = y_hat;
                    end
            end
        end % for i = 1:Mrand
        x_star = y_star(end)*y_star(1:end-1); % x_star is the estimate of b
        err=err + (b~=x_star)'; % count for number of error
        dlen_cnt=dlen_cnt+1;
    end % while
    uber(snr_cnt,:) = err/dlen_cnt; % ber for each user
    sber_SDP = sum(uber,2)/No_user; % avg ber
end %for snr_cnt=1:No_SNRs
semilogy(SNRdb,sber_SDP,'b-p');
xlabel('E/N_0');ylabel('avg prob of err');

clc;clear;close all

No_user=3;  %no of users
N=7;        %no of chips or spreading factor
dlen=10000;     %data length
c = zeros(No_user,N); % code
Eb=1;                   %Energy per bit
SNRdb=2:2:6;           %string of SNR in dB
No_SNRs=length(SNRdb);  %Find out the length of SNRdb string
err_SDP=zeros(1,No_user);   %no of errors bits
err_ML=zeros(1,No_user);    %no of errors bits
uber_SDP = zeros(No_SNRs,No_user); % ber for each user vs given SNR
uber_ML = zeros(No_SNRs,No_user); % ber for each user vs given SNR
dlen_cnt=0;                 %no of dlen sent
Mrand = 100;  % number of random variable generated
% assign spreading sequence for each user
c(1,:) = [-1 -1 -1 1 1 -1 1];
c(2,:) = [1 1 -1 -1 -1 1 -1];
c(3,:) = [-1 1 1 -1 -1 -1 1];
%diagonal matrix A for the matrix of received amplitude--------------------
-----------------------
A=zeros(1,No_user);
A(:)=1;
A=diag(A);
%pb is a permutation of data bits for k no of users = 2^k ; k by 2^k matrix
pb=zeros(No_user,2^No_user);
toggle=-1;
for count=1:No_user
   lenpb=2^(No_user-count);
   counter=0;
   for count2=1:2^No_user
      pb(count,count2)=toggle;
        counter=counter+1;
      if counter==lenpb
         toggle=-toggle;
         counter=0;
      end
   end
end
% simulation
for snr_cnt=1:No_SNRs
```

```
    dlen_cnt=0;                  %counter for no of time the while loop
below till error bit>100
    err_SDP(:)=0;                    %error bit counter for K users
    err_ML(:)=0;
    while ((sum(err_ML)/No_user < 2) | (dlen_cnt<10000))
        SNR=(10^(SNRdb(snr_cnt)/10)); %Signal to noise ratio
        sd=sqrt((Eb/SNR)/2);          % standard deviation
        b = 2*randint(No_user,1)-1;       % information bits
        P = partCorgen(c);              % partial cross-correlation matrix
        n = sd*mvnrnd(zeros(1,No_user),P);%(R_half)*(sd*randn(No_user,1)); %
noise with correlation matrix N0/2 * R
        y = P*A*b + n';                    % test stat
        % Multiuser Detection using SDP
        Q = [A*P*A , -A*y ; -(A*y)' , 0];
        cvx_begin sdp quiet
            cvx_solver sdpt3
            variable Y(No_user+1,No_user+1) symmetric
            minimize ( trace(Q*Y) )
            diag(Y) == 1;
            Y >= 0;
        cvx_end
        if min(eig(Y) < 0) % make sure Y is positive semidefinite
            continue;
        end
        yQy_min = inf;
        for i = 1:Mrand
            y_hat = (sign(mvnrnd(zeros(1,No_user+1),Y)))';
            yQy = y_hat'*Q*y_hat;
            if yQy < yQy_min
                yQy_min = yQy;
                y_star = y_hat;
            end
        end % for i = 1:Mrand
        x_star = y_star(end)*y_star(1:end-1); % x_star is the estimate of b
        err_SDP=err_SDP + (b~=x_star)'; % count for number of error
        % Multiuser Detection using ML
        for pb_ptr=1:(2^No_user)
            Bml=2*(A*y)'*pb(:,pb_ptr)-pb(:,pb_ptr)'*A*P*A*pb(:,pb_ptr);
            if pb_ptr==1
               Bml_max=Bml;
                pb_max=pb_ptr;
            else    if Bml>Bml_max
                    Bml_max=Bml;
                    pb_max=pb_ptr;
                  end
            end
         end
        br=pb(:,pb_max); % br is the estimate of b
        err_ML=err_ML + (b~=br)';% count for number of error
        dlen_cnt=dlen_cnt+1;
    end % while
    uber_ML(snr_cnt,:) = err_ML/dlen_cnt; % ber for each user
    uber_SDP(snr_cnt,:) = err_SDP/dlen_cnt; % ber for each user

end %for snr_cnt=1:No_SNRs
sber_ML = sum(uber_ML,2)/No_user; % avg ber
sber_SDP = sum(uber_SDP,2)/No_user; % avg ber
semilogy(SNRdb,sber_SDP,'b-p',SNRdb,sber_ML,'g-o');
legend('SDP,ML');
xlabel('E/N_0');ylabel('avg prob of err');
```

```
function [P]=partCorgen(c)
% c is the No_user by N matrix containing spreading sequence for each user
% P is the partial correlation matrix
[no_user,N] = size(c);
tau = randint(1,no_user-1,N);
tau = [0,sort(tau)];
P = ones(no_user);
for i = 1:no_user
    for j = i+1:no_user
        P(i,j) = 1/N*(c(i,tau(j)-tau(i)+1:end)*c(j,1:N - (tau(j)-tau(i)))');
    end
end
P = P + P' -1;
return
```

Section 4 Matlab Codes

```
clc;clear;close all

No_user=3;  %no of users
N=7;        %no of chips or spreading factor
c = zeros(No_user,N); % code
Eb=1;                   %Energy per bit
SNRdb=2:2:10;          %string of SNR in dB
No_SNRs=length(SNRdb);  %Find out the length of SNRdb string
err_MF=zeros(1,No_user);    %no of errors bits
err_DC=zeros(1,No_user);    %no of errors bits
err_LMMSE=zeros(1,No_user); %no of errors bits
uber_DC = zeros(No_SNRs,No_user); % ber for each user vs given SNR
uber_MF = zeros(No_SNRs,No_user); % ber for each user vs given SNR
uber_LMMSE = zeros(No_SNRs,No_user); % ber for each user vs given SNR
dlen_cnt=0;                 %no of dlen sent

% assign spreading sequence for each user
c(1,:) = [-1 -1 -1 1 1 -1 1];
c(2,:) = [1 1 -1 -1 -1 1 -1];
c(3,:) = [-1 1 1 -1 -1 -1 1];
%diagonal matrix A for the matrix of received amplitude--------------------
-----------------------
A=zeros(1,No_user);
A(:)=1;
A=diag(A);

% simulation
for snr_cnt=1:No_SNRs
    dlen_cnt=0;                 %counter for no of time the while loop
below till error bit>100
    err_MF(:)=0;                    %error bit counter for K users
    err_DC(:)=0;
    err_LMMSE(:)=0;
    while ((sum(err_LMMSE)/No_user < 10) | (dlen_cnt<10000))

        SNR=(10^(SNRdb(snr_cnt)/10)); %Signal to noise ratio
        sd=sqrt((Eb/SNR)/2);          % standard deviation
        b = 2*randint(No_user,1)-1;       % information bits
        P = partCorgen(c);              % partial cross-correlation matrix
        n = sd*mvnrnd(zeros(1,No_user),P);%(R_half)*(sd*randn(No_user,1)); %
noise with correlation matrix N0/2 * R
        y = P*A*b + n';                  % test stat
```

```
        % Multiuser Detection using MF
        y_MF = sign(y); %  y_MF is the estimate of b
        err_MF = err_MF + (y_MF~=b)';
        % Multiuser Detection using DC
        y_DC = sign(inv(P*A)*y);
        err_DC = err_DC + (y_DC~=b)';
        % Multiuser Detection using LMMSE
        y_LMMSE = sign(inv(P+sd*eye(size(P)))*y);
        err_LMMSE = err_LMMSE + (y_LMMSE~=b)';
        dlen_cnt=dlen_cnt+1;
    end % while
    uber_MF(snr_cnt,:) = err_MF/dlen_cnt; % ber for each user
    uber_DC(snr_cnt,:) = err_DC/dlen_cnt; % ber for each user
    uber_LMMSE(snr_cnt,:) = err_LMMSE/dlen_cnt; % ber for each user
end %for snr_cnt=1:No_SNRs
sber_MF = sum(uber_MF,2)/No_user; % avg ber
sber_DC = sum(uber_DC,2)/No_user; % avg ber
sber_LMMSE = sum(uber_LMMSE,2)/No_user; % avg ber
semilogy(SNRdb,sber_MF,'b-p',SNRdb,sber_DC ,'g-o',SNRdb,sber_LMMSE ,'r-s');
legend('MF','DC','LMMSE');
xlabel('E/N_0');ylabel('avg prob of err');
```