Sequence Design for Symbol-Asynchronous CDMA with Power or Rate Constraints

Jyothiram Kasturi and Rajesh Sundaresan *Senior Member, IEEE*

***Abstract*— Sequence design and resource allocation for a symbol-asynchronous chip-synchronous code division multiple access (CDMA) system is considered in this paper. A simple lower bound on the minimum sum-power required for a non-oversized system, based on the best achievable for a non-spread system, and an analogous upper bound on the sum rate are first summarised. Subsequently, an algorithm of Sundaresan and Padakandla is shown to achieve the lower bound on minimum sum power (upper bound on sum rate, respectively). Analogous to the synchronous case, by splitting oversized users in a system with processing gain N, a system with no oversized users is easily obtained, and the lower bound on sum power (upper bound on sum rate, respectively) is shown to be achieved by using N orthogonal sequences. The total number of splits is at most $N-1$.**

***Index Terms*— asynchronous, code division multiple access, resource allocation, sum capacity, uplink**

I. INTRODUCTION

In code division multiple access (CDMA) communications, users share the entire bandwidth with each other. The symbol waveform of a user signal is generated by spreading the chip waveform with its signature sequence. Due to CDMA systems' capability of offering high capacity, flexibility, and security, such systems have become the *de facto* standard for the third generation wireless systems and their immediate successors (for example, 1xEV-DO and HSUPA). There have been extensive researches on the impact of the signature sequences on the sum rate achievable under a set of power constraints, and the minimum sum power required to meet a set of rate constraints, in a CDMA system. These researches typically cover synchronous CDMA systems. The uplink of a cellular system is however inherently asynchronous because the users are spread over a large area and different users are received with different delays at the base station. Moreover, the clocks employed by the users are typically asynchronous resulting in timing drift. In this paper, we focus on two types of resource allocation problems for a simplistic asynchronous physical layer model. Specifically, we consider two problems:

- **Problem I**: Given users' power constraints, design signature sequences for users to maximise the sum rate achieved.
- **Problem II**: Given users' rate requirements, design signature sequences for users so that their sum power is minimised.

The authors are with the Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore

This work was supported by the Defence Research and Development Organisation, Government of India, Ministry of Defence, under the DRDO-IISc Programme on Advanced Research in Mathematical Engineering

We will assume that the timing drift is slow enough with respect to the duration of optimisation and frame times. This is typically the case in today's cellular systems supporting high speed data; optimisation is done once every frame, roughly about 2 milliseconds, whereas a clock error of 0.1 parts per million (typical in current mobile phones) leads to a drift of one chip every thousand frames in a 5 MHz bandwidth system.

For the symbol-synchronous CDMA system, Viswanath and Anantharam [1] provide a solution for Problem I, while Guess [2] provides a solution for Problem II. Tropp *et al.* [3] provide finite step algorithms for Problem I with reduced computational complexity and pose the problem within the framework of an inverse singular value problem where the goal is to construct matrices with specified column norms and singular values. Sundaresan and Padakandla [4] recently provided new and unified algorithms for both these problems. Their algorithm has complexity less than those of previous algorithms reported by Viswanath and Anantharam [1], Guess [2], is comparable with that of [3], and is computationally stable.

One useful concept introduced by Viswanath and Anantharam [1] is that of an *oversized* user. A user is oversized if his power constraint or rate requirement is large with respect to the other users. For example, if user 1 is the user with the highest power constraint p_1, he is oversized if $p_1 > p_{tot}/N$, the average power per dimension. A similar condition holds for rate requirements. We say the system is oversized if there exists one or more oversized users. Viswanath and Anantharam [1] show that an oversized system incurs a loss due to spreading, *i.e.*, the sum capacity is strictly smaller than that achievable with no spreading ($N = 1$). It is possible to convert an oversized system into a system with more users such that no user is oversized, if a mild splitting of users and the consequent multi-dimensional signaling is allowed ([4], [2]).

For the symbol-asynchronous chip-synchronous CDMA system, Problem I was solved by Luo *et al.* [5]. In this paper, we solve Problem II for the non-oversized symbol-asynchronous chip-synchronous CDMA system, and moreover, show that the algorithms of Sundaresan and Padakandla [4] achieve the bounds in both problems. As shown in [4], the oversized system can be reduced to a non-oversized system by splitting the oversized users, a concept put forth by Guess [2].

In our abstraction, asynchronism is modeled at the symbol level and not at the chip level. Practical systems however are chip asynchronous and optimisations of such systems have to be done over signature sequences and chip waveform subject to bandwidth constraints. While this is an interesting open problem, we address only the simpler chip-synchronous case in this paper. We also assume that the transmitters and receiver know the delay profiles of the users. The design of optimal sequences when the transmitters do not know the delay profiles of users is of practical interest and still open. In our model, signature sequences take on real values as opposed to taking values from a finite-valued set. Fortunately, practical systems such as the third generation partnership project's (3GPP's) high speed uplink packet access allocate multiple sequences (or codes) to a user. In such a scenario, we will see that the solution reduces to a time-division multiple-access system with signature sequences drawn from the standard basis and therefore the components take on values from the set $\{0, 1\}$. This simplification is achieved because of the admittedly unrealistic and overly simplistic single path channel assumption. Nevertheless, our analysis provides some insight into how resources may be divided and allocated in a cellular system.

II. System Model and Bounds

We consider a direct-sequence code division multiple access (DS-CDMA) system where all K users transmit to a single base-station and assume that the users signals are symbol-asynchronous but chip-synchronous. The processing gain is N and each symbol lasts N chips. The power, rate, and unit-energy signature sequence for user k are denoted by p_k, r_k, and s_k, respectively. We will assume that $s_k \in \mathbb{R}^N$, and that $s_k^t s_k = 1$.

Let the symbol delay for user k be τ_k, where τ_k takes on values from the set $\{0, 1, \cdots, N-1\}$. This indicates that the beginning of a symbol is delayed from a fixed reference chip by τ_k chips. The system designer is assumed to have knowledge of these delays. This is possible if these values change slowly with time in comparison to the optimisation duration.

Consider a frame consisting of $2M+1$ symbols for all K users with the ith symbol of user k given by $b_k[i]$, where $1 \le k \le K$, and $-M \le i \le M$. We assume no specific constellation and let $b_k[i] \in \mathbb{R}$. If a power constraint is specified, as in Problem I, then $\mathbb{E}\left[b_k^2[i]\right] \le Np_k$. The starting point of a user's symbol, relative to the reference, is τ_k chips. Thus, in a set of N chips starting from the reference, at most two symbols appear. Let $y(i) \in \mathbb{R}^N$ be the ith vector of received values with N components, starting from the reference. We can write $y(i)$ as

$$y(i) = \sum_{k=1}^{K} \Big([s_{k,N-\tau_k+1}, \cdots, s_{k,N}, 0, \cdots, 0]^t \, b_k[i-1] + [0, \cdots, 0, s_{k,1}, \cdots, s_{k,N-\tau_k}]^t \, b_k[i] \Big) + w(i), \quad (1)$$

where $s_k = [s_{k,1}, s_{k,2}, \cdots, s_{k,N}]^t$, and $w(i)$ is a zero-mean white Gaussian noise process taking values in $\mathbb{R}^N$ with covariance matrix I_N.

In Problem I, we are given a set of power constraints $p = (p_1, p_2, \cdots, p_K)$ energy-units/chip *i.e.*, $\mathbb{E}\left[b_k^2[i]\right] \le Np_k$. For a given set of signature sequences, let $r = (r_1, r_2, \cdots, r_K)$ be the rate assignment that is achievable on the symbol-asynchronous chip-synchronous Gaussian multiple access channel. The goal is to assign such a set of signature sequences and rates to the users so that the sum rate $r_{tot} \triangleq \sum_{k=1}^{K} r_k$ achievable is maximised.

In Problem II, we are given a set of rate requirements $r = (r_1, r_2, \cdots, r_K)$ nats/chip for the K users, and the goal is to assign signature sequences and power to users so that the sum power at the receiver $p_{tot} \triangleq \sum_{k=1}^{K} p_k$ is minimised, while ensuring reliable transmission at the specified rates.

We first show that sum power is lower bounded in Problem II and sum rate is upper bounded in Problem I as follows.

Proposition 1: The maximum sum rate achievable on a symbol-asynchronous chip-synchronous CDMA system with power constraints $(p_1, \cdots, p_K)$ is upper bounded by $\frac{1}{2}\log(1+p_{tot})$ nats/chip.

Analogously, the minimum sum of received powers to meet a specified rate constraint $(r_1, \cdots, r_K)$ is lower bounded by $\exp\{2r_{tot}\} - 1$ energy-units/chip.

Proof: We simply map a strategy on this asynchronous system with processing gain N to that on a synchronous system with processing gain 1. Consider a code of length M on the system with processing gain N. A user transmits his code in MN chips. User k begins a new code symbol τ_k chips after the reference, where $0 \le \tau_k < N$. Place a prefix of τ_k zeros prior to this user's first symbol, and suffix $N - \tau_k$ zeros after all $2M+1$ code symbol transmissions. Doing the same for all users, we have converted the M-length code on the asynchronous system with processing gain N into a code of length $MN+N$ on a chip-synchronous uplink multiple access system (MAC) with processing gain 1. Since $\lim_{M\to\infty} \frac{MN+N}{MN} = 1$, the sum rate of the asynchronous code (in bits per chip) can be no larger than the sum capacity of the synchronous unit-processing gain system which is $(1/2)\log(1+p_{tot})$.

The same code transformation technique shows that in a system with rate constraints, to support a sum rate of r_{tot}, the powers should be such that $r_{tot} \le (1/2)\log(1+p_{tot})$ and therefore $p_{tot} \ge \exp\{2r_{tot}\} - 1$. ■

III. Applicability of Algorithms

A. The Algorithms

We now reproduce here the algorithms of Sundaresan and Padakandla [4] and highlight the essential properties needed in our proof. The algorithms work in scenarios with no oversized users, *i.e.*,

$$Nx_k \le x_{tot}, \text{ for every } k = 1, 2, \cdots, N,$$

where $x = r$ or p as the case may be. The interpretation is that the rate or power per user does not exceed the average per dimension. If it does for a particular user, such a user is deemed an oversized one.

Let $A_k \triangleq A_{k-1} + Np_k s_k s_k^t$, for $k = 1, \cdots, K$, and set $A_0 = I_N$. The matrix A_k is of size $N \times N$ and represents the covariance matrix after the first k users are added into the system. The algorithm builds a sequence of eigenvalues of A_k:

$$\lambda^{(k)} \triangleq \left(\lambda_1^{(k)}, \cdots, \lambda_N^{(k)}\right) \triangleq \text{eig}\left(A_k\right),$$

for $k=1,\cdots,K$, where the vector of eigenvalues is arranged in decreasing order,

$$\lambda_1^{(k)} \geq \lambda_2^{(k)} \geq \cdots \geq \lambda_N^{(k)}.$$

The algorithm proceeds in such a way that adjacent sets of eigenvalues $\lambda^{(k)}$ and $\lambda^{(k+1)}$ satisfy the interlacing inequality,

$$\lambda_1^{(k+1)} \geq \lambda_1^{(k)} \geq \lambda_2^{(k+1)} \geq \lambda_2^{(k)} \geq \cdots \geq \lambda_N^{(k+1)} \geq \lambda_N^{(k)}. \quad (2)$$

Moreover, if energy is poured into a dimension, then all the previous dimensions are completely filled up to a specified maximum level $\lambda_{\max}$, *i.e.*, the corresponding eigenvalues with lower indices have attained their maximum target $\lambda_{\max}$. The interlacing inequality constraint ensures that an s_k can be found after each step, and the filling up of a dimension before subsequent dimensions are poured ensures that there is always space to pour a new user's energy without affecting the reachability of the optimal target set $\lambda^{(K)}$.

The following algorithms assume the existence of a subroutine $c(A,\lambda)$ that takes in a matrix A, a set of target eigenvalues λ so that λ and eig(A) satisfy the interlacing inequality. It then outputs a vector c such that eig$(A+cc^t)=\lambda$. The operation of this subroutine was discussed in [4] and is summarised immediately after the description of the algorithms.

Algorithm 2 ([4]): Problem I

- **Initialisation**: Set $\lambda_n^{(k)} \leftarrow 1$ for $k=0,1,\cdots,K$ and $n=1,\cdots,N$. Set the user index $k \leftarrow 1$, dimension $n \leftarrow 1$, $\lambda_{\max} \leftarrow 1+p_{tot}$, and $A_0 \leftarrow I_N$.
- **Step 1**: If $k>K$, stop.
- **Step 2 (a)**: If $\lambda_n^{(k-1)}+Np_k<\lambda_{\max}$, then set $\lambda_n^{(k)} \leftarrow \lambda_n^{(k-1)}+Np_k$ and go to **Step 3**.
- **Step 2 (b)**: If $\lambda_n^{(k-1)}+Np_k=\lambda_{\max}$, then set $\lambda_n^{(j)} \leftarrow \lambda_{\max}$ for $j=k,\cdots,K$. Also set $n \leftarrow n+1$ and go to **Step 3**.
- **Step 2 (c)**: If $\lambda_n^{(k-1)}+Np_k>\lambda_{\max}$, then set $\lambda_n^{(j)} \leftarrow \lambda_{\max}$ for $j=k,\cdots,K$, and $\lambda_{n+1}^{(k)} \leftarrow 1+\lambda_n^{(k-1)}+Np_k-\lambda_{\max}$. Also set $n \leftarrow n+1$.
- **Step 3**: Identify the vector $c_k=c\left(A_{k-1},\lambda^{(k)}\right)$. Then set $s_k \leftarrow c_k/||c_k||$, $r_k \leftarrow \frac{1}{2N}\log|A_k|-\frac{1}{2N}\log|A_{k-1}|$. This provides the sequence and rate for user k. Finally, set $A_k \leftarrow A_{k-1}+c_kc_k^t$, $k \leftarrow k+1$, and go to **Step 1**.

□

Remarks: User ordering does not matter in achieving the bounds. It is also easy to check via induction that

$$\text{trace } A_k = N + N\sum_{j=1}^{k} p_j, \quad \text{for } k=0,1,\cdots,K.$$

Algorithm 3 ([4]): Problem II

- **Initialisation**: Set $\lambda_n^{(k)} \leftarrow 1$ for all $k=0,1,\cdots,K$ and $n=1,\cdots,N$. Set the user index $k \leftarrow 1$, dimension $n \leftarrow 1$, $\lambda_{\max} \leftarrow \exp\{2r_{tot}\}$, and $A_0 \leftarrow I_N$.
- **Step 1**: If $k>K$, stop.
- **Step 2 (a)**: If $\lambda_n^{(k-1)}\cdot\exp\{2Nr_k\}<\lambda_{\max}$, then set $\lambda_n^{(k)} \leftarrow \lambda_n^{(k-1)}\cdot\exp\{2Nr_k\}$ and go to **Step 3**.
- **Step 2 (b)**: If $\lambda_n^{(k-1)}\cdot\exp\{2Nr_k\}=\lambda_{\max}$, then set $\lambda_n^{(j)} \leftarrow \lambda_{\max}$ for $j=k,\cdots,K$. Set $n \leftarrow n+1$ and go to **Step 3**.
- **Step 2 (c)**: If $\lambda_n^{(k-1)}\cdot\exp\{2Nr_k\}>\lambda_{\max}$, then set $\lambda_n^{(j)} \leftarrow \lambda_{\max}$ for $j=k,\cdots,K$, and $\lambda_{n+1}^{(k)} \leftarrow \lambda_n^{(k-1)}\cdot\exp\{2Nr_k\}/\lambda_{\max}$. Also set $n \leftarrow n+1$.
- **Step 3**: Identify the vector $c_k=c\left(A_{k-1},\lambda^{(k)}\right)$. Then set $s_k \leftarrow c_k/||c_k||$, $p_k \leftarrow \left(c_k^tc_k\right)/N$. This provides the sequence and power for user k. Finally, set $A_k \leftarrow A_{k-1}+c_kc_k^t$, $k \leftarrow k+1$, and go to **Step 1**.

□

Remark: Once again, it should be easy to see that at each step,

$$|A_k| = \exp\left\{2N\sum_{j=1}^{k} r_j\right\} \text{ for } k=0,1,\cdots,K, \quad (3)$$

where $|\cdot|$ denotes the determinant of the argument.

We now make some remarks about the subroutine $c(\cdot,\cdot)$. Sundaresan and Padakandla [4] show the sufficiency of identifying and storing the orthogonal matrices

$$U_k = \left[u_1^{(k)} \ \cdots \ u_N^{(k)}\right]$$

that diagonalize A_k, and the eigenvalues $\lambda^{(k)}$. Note that U_0 can be taken as any arbitrary orthogonal matrix since $A_0=I_N$, the identity matrix. Computation of $c\left(A_{k-1},\lambda^{(k)}\right)$ utilizes only U_{k-1}, $\lambda^{(k-1)}$, and $\lambda^{(k)}$, and is done as follows. If only one eigenvalue changes as in Steps 2(a) or 2(b), then

$$U_k = U_{k-1}, \quad (4)$$

and

$$c_k = \left(\sqrt{\lambda_n^{(k)}-\lambda_n^{(k-1)}}\right)u_n^{(k)}, \quad (5)$$

where n is the index before it is updated to $n+1$ in case of Step 2(b).

For Step 2(c) exactly two eigenvalues change in going from A_{k-1} to A_k. So the two matrices share $N-2$ eigenvectors and exactly two eigenvectors change. These are computed via a rotation in the $(n,n+1)$st plane as follows:

$$\left[u_n^{(k)} \ u_{n+1}^{(k)}\right] = \left[u_n^{(k-1)} \ u_{n+1}^{(k-1)}\right] \begin{bmatrix} \alpha & -\beta \\ \beta & \alpha \end{bmatrix}, \quad (6)$$

where

$$\alpha = \sqrt{\frac{\left(\hat{\lambda}_n-\lambda_{n+1}\right)\left(\lambda_n-\hat{\lambda}_{n+1}\right)}{\left(\hat{\lambda}_n-\hat{\lambda}_{n+1}\right)\left(\lambda_n-\lambda_{n+1}\right)}}, \quad (7)$$

$$\beta = \sqrt{1-\alpha^2} = \sqrt{\frac{\left(\hat{\lambda}_{n+1}-\lambda_{n+1}\right)\left(\hat{\lambda}_n-\lambda_n\right)}{\left(\hat{\lambda}_n-\hat{\lambda}_{n+1}\right)\left(\lambda_n-\lambda_{n+1}\right)}}, \quad (8)$$

with $\lambda \triangleq \lambda^{(k-1)}$ and $\hat{\lambda} \triangleq \lambda^{(k)}$ for simplicity. Furthermore,

$$c_k = y_n \ u_n^{(k-1)} + y_{n+1} \ u_{n+1}^{(k-1)}, \quad (9)$$

where

$$y_n = \sqrt{\frac{\left|\lambda_n - \hat{\lambda}_n\right| \left|\lambda_n - \hat{\lambda}_{n+1}\right|}{|\lambda_n - \lambda_{n+1}|}}, \quad (10)$$

$$y_{n+1} = \sqrt{\frac{\left|\lambda_{n+1} - \hat{\lambda}_n\right| \left|\lambda_{n+1} - \hat{\lambda}_{n+1}\right|}{|\lambda_n - \lambda_{n+1}|}}. \quad (11)$$

In all three cases, we have

$$p_k = \hat{\lambda}_{n+1} + \hat{\lambda}_n - (\lambda_{n+1} + \lambda_n),$$
$$r_k = \frac{1}{2N} \log \left(\frac{\hat{\lambda}_{n+1} \hat{\lambda}_n}{\lambda_{n+1} \lambda_n} \right).$$

As further remarked in [4], the algorithms' complexity is only $O(KN)$ floating point operations.

We will later see that these algorithms are applicable even in the symbol-asynchronous case provided we start with $U_0 = I_N$ and use an appropriate time labeling as described in the next subsection.

B. Time Labeling

The goal of choosing a good time labeling is to choose a good reference where users with delay τ_k relative to this reference have $s_{k,n} = 0$ for $n = N - \tau_k + 1, \cdots, N$. This property is beneficial because the first term in (1) is zero and the system reduces to a symbol-synchronous system. This is not valid for any arbitrary time reference, and therefore a good choice has to be made. The idea was first used by Luo *et al.* [6] in the context of decision feedback detection and subsequently exploited by Luo *et al.* in [5]. Our approach is essentially the same as that of Luo *et al.* in [5] except for a change in indexing.

Let $\tau_1, \cdots, \tau_K$ be the delays in chips with respect to an arbitrary reference. Define group G_j for $j = 1, \cdots, N$, to be the set of all the users whose symbol delay is $N - j$ chips, *i.e.,*

$$G_j = \{k \mid 1 \le k \le K \text{ and } \tau_k = N - j\},$$

for $j = 1, 2, \cdots, N$. We will denote this choice of reference, also called *time labeling,* by the ordering $T = [G_1, G_2, \cdots, G_N]$. See Fig. 1 for a depiction of the groups. Any other choice of reference can be represented by a cyclic permutation of the time labeling $[G_1, G_2, \cdots, G_N]$. The groups $G_1, G_2, \cdots, G_N$ partition the set $\{1, 2, \cdots, K\}$. As an example, the time labeling $\hat{T} = [G_2, \cdots, G_N, G_1]$ indicates that the reference is delayed by 1 chip unit.

For a given rate- or power-tuple denoted generically by the vector $x = (x_1, x_2, \cdots, x_K)$, and for a subset $G \subset \{1, 2, \cdots, K\}$ we let the aggregate sum rate or sum power to be

$$x_G \triangleq \sum_{k \in G} x_k.$$

Following the proof of [5, Theorem 5] we show the existence of a good time labeling. This labeling will have the property that users in G_j have signature sequences with zeros in positions $j + 1, \cdots, N$, effectively resulting in a symbol-synchronous system (see Fig. 1).

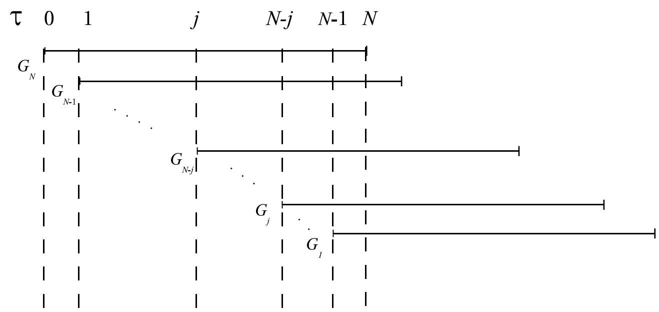

Fig. 1. A snapshot of symbol asynchronous CDMA system showing different groups

Theorem 4 ([5]): Given a vector $x \in \mathbb{R}_+^K$, there exists a time labeling $\hat{T} = \left[\hat{G_1}, \hat{G_2}, \cdots, \hat{G_N}\right]$, such that for all g satisfying $1 \le g \le N$, we have

$$\sum_{j=1}^{g} x_{\hat{G}_j} \le \frac{g}{N} \cdot x_{tot}. \quad (12)$$

Proof: The proof is identical to the proof of [5, Theorem 5] except for a change in indexing. It is provided in the Appendix for completeness. Note that the theorem holds even for systems with oversized users. ■

Henceforth we may assume that the labeling $[G_1, \cdots, G_N]$ satisfies (12). The intuition is now as follows. The particular choice of time labeling ensures that as users are allocated rate (or power) one after another in the order of the groups in the time labeling, all the users in groups up to and including g do not need more than g dimensions. Consequently, $g+1$ and beyond carry no energy and the signature sequence values for these locations are zeros. This is true for every group index g. This is proved formally in the next section.

C. Applicability of algorithms

We now state and prove the main result of this paper.

Theorem 5: Consider a symbol-asynchronous system of K non-oversized users. For the time labeling satisfying (12), index the users in group G_j as

$$\sum_{i=1}^{j-1} |G_i| + 1, \sum_{i=1}^{j-1} |G_i| + 2, \cdots, \sum_{i=1}^{j} |G_i|, \quad (13)$$

thus leading to an indexing of all K users. After setting $U_0 = I_N$, an execution of Algorithms 2 or 3 results in sequences for the symbol-synchronous system with the property that users in group G_j have zeros in locations $j+1, j+2, \cdots, N$, for $j = 1, 2, \cdots, N$.

Furthermore, there is a mapping from this set of sequences on the symbol-synchronous system to a set of sequences on the symbol-asynchronous system such that the resulting sequences are optimal for the symbol-asynchronous system, *i.e.,* they achieve the sum power lower bound or the sum rate upper bound given in Proposition 1.

Proof: As remarked earlier, the algorithms pour a user's energy in the dimension with the lowest possible index n

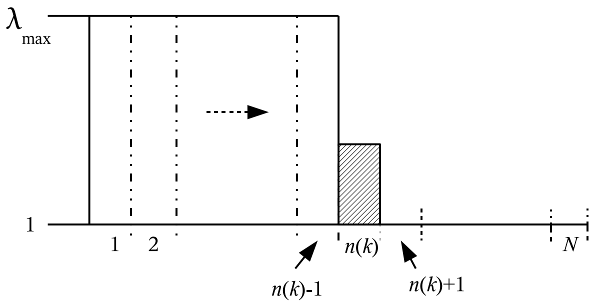

Fig. 2. Illustration of power across dimensions after the k^{th} user is added.

subject to the condition that the eigenvalues do not exceed $\lambda_{\max}$. See Fig. 2 for an illustration of the algorithm. Let $n(k)$ be defined as (see once again Fig. 2)

$$n(k) \triangleq \max\left\{n \mid 1 \leq n \leq N \text{ and } \lambda_n^{(k)} > 1\right\}, \qquad (14)$$

the dimension with the highest index containing energy from user k. We first claim that

$$n(k) = \left\lceil \frac{N\sum_{j=1}^{k} x_j}{x_{tot}} \right\rceil. \qquad (15)$$

We show that this is the case for both algorithms (with x replaced by p or r, as the case may be). For Algorithm 2,

$$\begin{aligned} N \cdot \sum_{j=1}^{k} p_j &= \text{trace } A_k - N \\ &= \sum_{i=1}^{N} \lambda_i^{(k)} - N \\ &= (n(k)-1)\cdot(\lambda_{\max}-1) + \lambda_{n(k)}^{(k)} - 1 \qquad (16) \\ &= (n(k)-1)\cdot p_{tot} + \lambda_{n(k)}^{(k)} - 1, \qquad (17) \end{aligned}$$

where (16) follows from the fact that $n(k)-1$ eigenvalues have attained their maximum value $\lambda_{\max}$. From (14), we have

$$0 < \lambda_{n(k)}^{(k)} - 1 \leq \lambda_{\max} - 1 = p_{tot}.$$

Substitution of these two inequalities in (17) yields

$$(n(k)-1)\cdot p_{tot} < N\sum_{j=1}^{k} p_j \leq n(k)\cdot p_{tot},$$

which substantiates the claim in (15) for Algorithm 2.

For Algorithm 3, observe that

$$\begin{aligned} \exp\left\{2N\sum_{j=1}^{k} r_j\right\} &= |A_k| \\ &= \prod_{i=1}^{N} \lambda_i^{(k)} \\ &= (\lambda_{\max})^{n(k)-1}\cdot\lambda_{n(k)}^{(k)} \\ &= \exp\left\{(n(k)-1)\cdot 2r_{tot}\right\}\cdot\lambda_{n(k)}^{(k)}. \end{aligned} \qquad (18)$$

Substitution of the two obvious inequalities

$$1 < \lambda_{n(k)}^{(k)} \leq \lambda_{\max} = \exp\{2r_{tot}\}$$

in (18) yields

$$\begin{aligned} \exp\left\{(n(k)-1)\cdot 2r_{tot}\right\} &< \exp\left\{2N\sum_{j=1}^{k} r_j\right\} \\ &\leq \exp\left\{n(k)\cdot 2r_{tot}\right\}, \end{aligned}$$

and thus the claim in (15) is verified for Algorithm 3 after taking logarithms.

The lemma below shows that the sequence for user k spans only the first $n(k)$ dimensions. More precisely,

Lemma 6: Let c_k be the output of the subroutine $c(\cdot,\cdot)$, a scaled version of the sequence for user k. Let U_k be the orthogonal matrix of size $N \times N$ that diagonalises A_k. Then

$$c_k \in \text{span}\left\{e_1, \cdots, e_{n(k)}\right\}, \qquad (19)$$

where $e_i \in \mathbb{R}^N$ is the standard basis vector with 1 in the ith position and zeros elsewhere, and

$$U_k = \begin{bmatrix} B_k & \bigcirc \\ \bigcirc & I_{N-n(k)} \end{bmatrix}, \qquad (20)$$

where B_k is a real-valued submatrix of size $n(k)\times n(k)$.

Proof: We use the notation $U_k = \left[u_1^{(k)}, \cdots, u_N^{(k)}\right]$. The proof is by induction on k. For $k=1$, because there are no oversized users, we clearly have $n(1)=1$. Upon addition of user 1, only one eigenvalue will change, $\lambda_1^{(1)}$, and because we begin with $U_0 = I_N$, we have

$$c_1 = \left(\sqrt{\lambda_1^{(1)} - 1}\right) e_1 \in \text{span}\{e_1\}$$

Furthermore, $U_1 = U_0$, and therefore the lemma is true for $k=1$ with $n(1)=1$ and $B_1 = [1]$.

Assume now that the lemma is true for $k-1$. We may therefore write

$$U_{k-1} = \begin{bmatrix} B_{k-1} & \bigcirc \\ \bigcirc & I_{N-n(k-1)} \end{bmatrix}. \qquad (21)$$

As user k is not oversized, clearly $n(k)$ is either $n(k-1)$ or $n(k-1)+1$. Since column $n(k-1)$ of U_{k-1}, *i.e.* $u_{n(k-1)}^{(k-1)}$, is the column corresponding to the last column of B_{k-1}, we have

$$u_{n(k-1)}^{(k-1)} \in \text{span}\left\{e_1, \cdots, e_{n(k-1)}\right\}, \qquad (22)$$

because column $n(k-1)$ of U_{k-1} has only the first $n(k-1)$ terms possibly nonzero. If $n(k) = n(k-1)$, then only one eigenvalue changes, and by (4) and (5) $U_k = U_{k-1}$,

$$\begin{aligned} c_k &= \left(\sqrt{\lambda_{n(k)}^{(k)} - \lambda_{n(k-1)}^{(k-1)}}\right)\cdot u_{n(k-1)}^{(k-1)} \\ &\in \text{span}\{e_1, \cdots, e_{n(k-1)}\}, \end{aligned}$$

and the lemma holds. If $n(k) = n(k-1)+1$, then from (6) it is easy to see that

$$U_k = U_{k-1}\cdot\begin{bmatrix} I_{n(k-1)-1} & & \bigcirc \\ & \begin{bmatrix} \alpha & \beta \\ \beta & -\alpha \end{bmatrix} & \\ \bigcirc & & I_{N-n(k)} \end{bmatrix}, \qquad (23)$$

where α and β are as in (7) and (8). From (9) and after defining $n \triangleq n(k)$, we get

$$c_k = y_{n-1} \cdot u_{n-1}^{(k-1)} + y_n \cdot u_n^{(k-1)}, \qquad (24)$$

where y_{n-1} and y_n are scalars as defined in (10) and (11). From (22), the fact that $u_n^{(k-1)} = e_n$, and (24), it is clear that (19) is verified. By multiplying the right side of (23) it is easy to verify that U_k is of the form (20) with B_k of size $n \times n$. This concludes the proof of the lemma. ■

We have thus verified that c_k and therefore s_k is in the span of the standard basis vectors $\{e_1, \cdots, e_{n(k)}\}$. Consequently, s_k has only zeros in locations $n(k)+1, \cdots, N$.

Let $g(k)$ be the index of the group to which user k belongs *i.e.*, $g(k) = g$ if and only if $k \in G_g$. From (15) and the fact that the time labeling satisfies (12), we have

$$n(k) = \left\lceil \frac{N \sum_{i=1}^{k} x_i}{x_{tot}} \right\rceil \leq \left\lceil \frac{N \sum_{i=1}^{g(k)} x_{G_i}}{x_{tot}} \right\rceil \leq g(k),$$

where the last inequality follows because the inequality holds without the ceiling function due to the time ordering, and $g(k)$ is an integer. Consequently, s_k has only zeros in locations $g(k)+1, \cdots, N$, and the first part of the main theorem is proved. The upper bound on sum rate or lower bound on sum power is achieved on the symbol-synchronous system by virtue of the property of the algorithms for non-oversized systems.

We still have to show how to map these to a set of sequences on the asynchronous system. But this is easily done via a time-reversal technique. Referring to Fig. 1, we see that users in G_1 last only one chip, users in G_2 two chips, and so on. We simply define $(s'_{k,1}, s'_{k,2}, \cdots, s'_{k,g}, 0, \cdots, 0)$ as $(s_{k,g}, s_{k,g-1}, \cdots, s_{k,1}, 0, \cdots, 0)$. Indeed, this is equivalent to looking at the N chips time-reversed starting from the reference, and we have converted the symbol-asynchronous system into a symbol-synchronous system. This concludes the proof that the algorithm results in an optimal allocation. ■

Sequences which achieve the sum rate upper bound have been named Generalised Asynchronous Welch Bound Equality (GAWBE) sequences by Luo *et al.* in [5].

IV. Multi-Dimensional Signaling and Sufficiency of N Orthogonal Sequences

Let the time labeling satisfy (12). Furthermore index the users as given in (13). Define $X_k \triangleq \sum_{j=1}^{k} x_k$, the cumulative constraint, for $k = 1, 2, \cdots, K$. We say that a *symmetric sum partition* (SSP) exists for this time labeling and indexing if we can find integers $k_1, k_2, \cdots, k_N$ such that $1 \leq k_1 < k_2 < \cdots < k_N = N$, and $X_{k_g} = (g/N)x_{tot}$, for $g = 1, \cdots, N$. In other words, there is a partition such that the partial sum up to k_g correctly fills g dimensions, for $g = 1, 2, \cdots, N$. This definition is more restrictive than the definition given by Sundaresan and Padakandla in [4] because the definition is based on a specific user ordering; however, the specialisation is necessary to handle symbol-asynchronism.

If an SSP exists, it is clear that during the execution of either of the algorithms, Step 2(c) is never entered. Consequently, $U_k = I_N$ for $k = 0, 1, \cdots, K$, and the sequence matrix is composed of only $e_i, \; i = 1, \cdots, N$; time-division multiplexing with shared time-slots is therefore sufficient for optimality. Users sharing a time-slot are decoded via the successive interference cancelation technique.

If an SSP does not exist, it is easy to engineer one by splitting users that straddle two dimensions, *i.e.*, those that require an execution of Step 2(c). If the original system has no oversized users, at most $N-1$ of these users get split into exactly two virtual users each. The resulting enlarged vector x has an SSP. However, such split users pour their energy into two time-slots instead of one, and therefore use two-dimensional signaling.

If there are oversized users in the system, a time labeling satisfying (12) can still be found since Theorem 4 holds for this case as well. We then index the users as in (13), look at the cumulative constraint X_k for $k = 1, \cdots, K$, and divide those users k whose inclusion makes X_k exceed $(g/N)x_{tot}$, for $g = 1, \cdots, N-1$. These users are then split (perhaps multiple times) to obtain an enlarged x that has an SSP. Note that there are at most $N-1$ splittings. The resulting sequence matrix for the enlarged system is simply composed of elements from the standard basis.

The above remarks are easy to verify and their proofs omitted.

V. Concluding Remarks

The work in this paper was motivated by a need to abstract certain imperfections in the physical layer into a model suitable enough for higher layer optimisation. Towards this end, we modeled asynchronism inherent in uplink multiple access systems. Our model is the same as that of Luo *et al.* [5]. The model has certain drawbacks - for example, the assumption of chip-synchronism - but provides some insights to the system designer for optimal use of resources. In particular, we looked at two optimisation problems. In the first problem our goal was to maximise the achievable sum rate subject to a power constraint. In the second, the goal was to minimise the received sum power subject to meeting a rate constraint. Sum rate is a good measure to optimise if all users pay the same rupees per successfully transmitted bit. It maximises the revenue collected by the service provider. The total received power is also a good quantity to minimise because it is indicative of interference at a neighbour cell when users are placed at random (uniform) within the cell. It also represents battery utilisation of all users in a collective fashion. The delays are assumed to be static over the optimisation interval, a realistic assumption since clock drifts result in a shift of a chip roughly once every 1000 frames, where a frame is roughly 2 milliseconds and is considered one optimisation unit.

We then showed that the finite-step algorithms proposed by Sundaresan and Padakandla [4] can be utilised to design optimal sequences and allocate rates or powers. The time labeling idea of [6] provided a good time reference; if we ordered users in the decreasing order of delays relative to this time reference, the cumulative requirements (rate or power) satisfied a boundedness condition given precisely in (12). We then showed that the algorithms of [4] led to a design with at least as many trailing zeros as the delay from the chosen reference. We then recognised this as a sequence assignment on the symbol-asynchronous system. The upshot is that we converted the symbol-asynchronous system into a symbol-synchronous system without suffering a penalty in our measures of performance. The last step is similar to that of Luo *et al.*, but the sequence design algorithms studied in this paper are simpler, use fewer computations, and are computationally stable.

We then remarked as in [4] that splitting will result in sequences taken from the standard basis. So as to remain optimal, however, the ordering of users and the initial choice U_0 turned out to be important.

The results of this paper are applicable in a wireless scenarios with slow and frequency-flat fading. These assumptions validate a constant channel assumption within the optimisation period. User delay constraints are such that users will have to transmit every symbol period, and are therefore always on. Channel and timing offset information were assumed available. It would be of practical interest to relax the assumptions on perfect channel and timing offset knowledge, and on frequency-flat fading.

APPENDIX I
PROOF OF THEOREM 4

Proof: Consider a time labeling $T_1 = [G_1, G_2, \cdots, G_N]$ that does not satisfy (12). We can find a minimum index $g_{\min}(T_1)$ satisfying $1 \le g_{\min}(T_1) < N$, such that (12) is satisfied for all $g < g_{\min}(T_1)$, and

$$\sum_{j=1}^{g_{\min}(T_1)} x_{G_j} > g_{\min}(T_1) \cdot \frac{x_{tot}}{N} \tag{25}$$

We can also find the maximum index $g_{\max}(T_1)$, such that

$$\sum_{j=1}^{g_{\max}(T_1)} x_{G_j} > g_{\max}(T_1) \cdot \frac{x_{tot}}{N} \tag{26}$$

Clearly, $g_{\max}(T_1) < N$ since (12) holds with equality for $g = N$.

The idea of the proof is to begin with an unsatisfactory time labeling that does not satisfy (12) and construct a new time labeling that satisfies (12). To this purpose, it is sufficient to get a time labeling T_2 from the time labeling T_1 such that $g_{\min}(T_2) > g_{\min}(T_1)$. As the sequence $g_{\min}$ of the obtained sequence of time labelings is monotonically increasing, and because an unsatisfactory time labeling T has $g_{\min}(T) < N$, the construction has to end in a finite number of iterations at a satisfactory time labeling.

By assumption,

$$\sum_{j=1}^{g_{\max}(T_1)+m} x_{G_j} \le (g_{\max}(T_1) + m) \cdot \frac{x_{tot}}{N}, \tag{27}$$

for $1 \le m \le N - g_{\max}(T_1)$. From (26) and (27)

$$\sum_{j=g_{\max}(T_1)+1}^{g_{\max}(T_1)+m} x_{G_j} \le m \cdot \frac{x_{tot}}{N}, \quad 1 \le m \le N - g_{\max}(T_1) \tag{28}$$

Applying backward rotation $N - g_{\max}(T_1)$ times to the time labeling T_1, we get the new time labeling

$$T_2 = \left[G_{g_{\max}(T_1)+1}, \cdots, G_N, G_1, \cdots, G_{g_{\max}(T_1)}\right].$$

Renaming the groups $T_2 = \left[\hat{G_1}, \cdots, \hat{G_N}\right]$, (28) may be rewritten in the new time labeling as

$$\sum_{j=1}^{m} x_{\hat{G_j}} \le m \cdot \frac{x_{tot}}{N}, \quad 1 \le m \le N - g_{\max}(T_1) \tag{29}$$

Further, for $m = N - g_{\max}(T_1) + 1$ to $m = N - g_{\max}(T_1) + g_{\min}(T_1) - 1$

$$\begin{aligned}\sum_{j=1}^{m} x_{\hat{G_j}} &= \sum_{j=1}^{N-g_{\max}(T_1)} x_{\hat{G_j}} + \sum_{j=N-g_{\max}(T_1)+1}^{m} x_{\hat{G_j}} \\ &\le (N - g_{\max}(T_1)) \frac{x_{tot}}{N} \\ &\quad + (m - (N - g_{\max}(T_1))) \frac{x_{tot}}{N} \\ &= m \frac{x_{tot}}{N},\end{aligned} \tag{30}$$

where (30) follows from (29) and an application of (12) for $g < g_{\min}(T_1)$ which in turn follows from the definition of $g_{\min}(T_1)$. Consequently,

$$g_{\min}(T_2) \ge N - g_{\max}(T_1) + g_{\min}(T_1) > g_{\min}(T_1),$$

and the proof is complete. ■

REFERENCES

[1] P. Viswanath and V. Anantharam, "Optimal sequences and sum capacity of synchronous CDMA systems," *IEEE Trans. Inform. Theory*, vol. 45, no. 6, pp. 1984 – 1991, Sep. 1999.

[2] T. Guess, "CDMA with power control and sequence design: The capacity region with and without multidimensional signaling," *IEEE Trans. Inform. Theory*, vol. 50, no. 11, pp. 2604 – 2619, Nov. 2004.

[3] J. A. Tropp, I. S. Dhillon, and J. R. W. Heath, "Finite-step algorithms for constructing optimal CDMA signature sequences," *IEEE Trans. Inform. Theory*, vol. 50, no. 11, pp. 2916–2921, Nov. 2004.

[4] R.Sundaresan and A.Padakandla, "The size of optimal sequence sets for synchronous CDMA systems," *submitted to the IEEE Transactions on Inform. Theory*, http://arxiv.org/abs/cs.IT/0606015, Jun. 2006.

[5] J. Luo, S. Ulukus, and A. Ephremides, "Optimal sequences and sum capacity of symbol asynchronous CDMA systems," *IEEE Trans. Inform. Theory*, vol. 51, no. 8, pp. 2760–2769, Aug. 2005.

[6] J.Luo, K.Pattipati, P.Willett, and F.Hasegawa, "Optimal user ordering and time labeling for decision feedback detection in asynchronous CDMA," *IEEE Trans. Commun.*, vol. 51, no. 11, pp. 1754 – 1757, Nov. 2003.