Kd-Tree Codebook for Limited Feedback CDMA

Kritsada Mamat and Wiroonsak Santipach
Department of Electrical Engineering
Faculty of Engineering, Kasetsart University
Bangkok, Thailand 10900
Email: {g5065253, wiroonsak.s}@ku.ac.th

***Abstract*—We propose a quantization scheme based on a Kd (K dimensional)-tree algorithm for signature sequence in a reverse-link direct sequence (DS)- code division multiple access (CDMA). With a few feedback bits, a receiver quantizes the optimal signature that minimizes interference for a desired user, and relays it to the user via an error-free feedback channel. A user performance depends on the quantization codebook and a number of feedback bits. We show the performance of the proposed Kd-tree codebook with a nearest neighbor criterion and derive the performance approximation. Also we modify the Kd-tree scheme to search for the entry in the codebook, which gives the least interference. Numerical examples show that the Kd-tree codebook performs close to the optimal codebook with only fraction of computational complexity.**

I. Introduction

To increase a user performance in a direct sequence (DS)-code division multiple access (CDMA), a signature sequence for a desired user needs to be optimized for current channel and interference conditions. A receiver, which can estimate channel state information (CSI), can compute the user's optimal signature and sends it to the user via a feedback channel. If the channel is changing slowly, the signature update needs not be frequent and can greatly improve the user performance [1]. A feedback channel is rate-limited and therefore, the user signature optimized at the receiver must be quantized before feeding back to the transmitter, which updates its signature sequence, accordingly. The resulting performance will depend on a quantization scheme used to quantize signature and a feedback rate.

References [1]–[5] have proposed codebook designs, which depend on channel and interference statistics. (In a multi-antenna channel, a signature refers to a set of spatial transmit coefficients.) Assuming that interfering signatures are independent and Gaussian distributed, we proposed a random vector quantization (RVQ) codebook [1], [6] whose entries are independent and isotropically distributed. RVQ is motivated by the fact that the optimal *unquantized* signature is the eigenvector of a interference covariance matrix, corresponding to the minimum eigenvalue. The distribution of the eigenvector is isotropic. In other words, the optimal signature is likely to point in any direction of the signal space. RVQ was shown to be optimal (i.e. minimizing interference) in a large system limit in which processing gain, number of interfering users, and feedback bits all tend to infinity with fixed ratios. Even though optimality is achieved only in a large system regime, RVQ performs close to the optimized codebook for a finite-size system as well [7].

To locate the optimal entry in an RVQ codebook, exhaustive search is required. Since a number of entries in an RVQ codebook grows exponentially with number of feedback bits, search complexity also grows exponentially as well. Reference [8] proposed a low-complexity quantization scheme based on a noncoherent detection technique. Elements for code entries in [8] are constrained to be PAM symbols for real vector entries or QAM symbols for complex ones. A search complexity of this PAM codebook is much less than RVQ and it performs very well for a large feedback. However, its performance is lacking for small number of feedback bits. This dues to a nearest neighbor criterion, which PAM scheme employs to locate the selected entry. We note that the entry that is nearest to the optimal unquantized vector does not necessarily minimize interference power.

In [9], we proposed to organize entries from RVQ codebook into a binary search tree. The entries are clustered in each step by the generalized Lloyd algorithm [10]–[12]. The associated search complexity for this tree-structured RVQ increases *linearly* with feedback bits. However, the performance of this tree-structured codebook is significantly inferior to the original RVQ codebook with exhaustive search. In this paper, we improve the tree-structured RVQ by applying a K-dimensional (Kd)-tree algorithm proposed by [13], [14]. Kd-tree represents a set of points in a K-dimensional space and supports a nearest neighbor query. First, we apply Kd-tree to locate the code entry in RVQ codebook, which is the nearest neighbor (closest in Euclidean distance) to the optimal unquantized vector. The associated performance is approximately equal to that of PAM codebook. We also derive the performance upper bound and show that it is a good approximation for the performance of Kd-tree. Then, we modify the objective of the search algorithm for Kd-tree from Euclidean distance to the optimal signature vector to an interference power. The Kd-tree search with modified objective performs very close to RVQ with exhaustive search and requires fewer orders of magnitude in computational complexity.

II. System Model

We consider a discrete-time reverse-link DS-CDMA with K users and processing gain N. Assuming an ideal nonfading

This work was supported by the 2008 Telecommunications Research and Industrial Development Institute (TRIDI) scholarship and the Commission on Higher Education and Thailand Research Fund under grant MRG5080174.

channel, the $N \times 1$ received vector is given by

$$\boldsymbol{r} = \sum_{k=1}^{K} \boldsymbol{s}_k b_k + \boldsymbol{n} \tag{1}$$

where $\boldsymbol{s}_k$ is the $N \times 1$ signature vector for user k whose element is independent and Gaussian distributed with zero mean and variance $1/N$, b_k is the transmitted symbol for user k with zero mean and unit variance, and $\boldsymbol{n}$ is the $N \times 1$ additive white Gaussian noise vector with zero mean and covariance $\sigma_n^2 \boldsymbol{I}$ where $\boldsymbol{I}$ is an identity matrix.

We assume that a receiver for user 1 is a linear matched filter given by

$$\boldsymbol{c}_1 = \boldsymbol{s}_1 \tag{2}$$

and the associated output signal-to-interference plus noise ratio (SINR) for user 1 is given by [15]

$$\gamma_1 = \frac{1}{\boldsymbol{s}_1^\dagger \boldsymbol{S}_1 \boldsymbol{S}_1^\dagger \boldsymbol{s}_1 + \sigma_n^2} \tag{3}$$

where † denotes Hermitian transpose and $\boldsymbol{S}_1$ is the $N \times (K-1)$ signature matrix whose columns are the interfering signatures $\{\boldsymbol{s}_2, \ldots, \boldsymbol{s}_K\}$. The matched filter was shown to perform worse than other more complex receivers, e.g. linear MMSE receiver [15]. However, its performance can be improved significantly when it is combined with a feedback scheme with a few feedback bits [1].

As we can see from (3), the user performance is a function of signature $\boldsymbol{s}_1$. By adapting $\boldsymbol{s}_1$ to minimize the interference power given by $\boldsymbol{s}_1^\dagger \boldsymbol{S}_1 \boldsymbol{S}_1^\dagger \boldsymbol{s}_1$, we can improve the performance. Given the interfering signature matrix $\boldsymbol{S}_1$, the optimal $\boldsymbol{s}_1$ is the eigenvector of the interference covariance matrix $\boldsymbol{S}_1 \boldsymbol{S}_1^\dagger$ corresponding to the smallest eigenvalue. In a typical wireless setting, the receiver can estimate the interference covariance during training and thus, can compute the optimal $\boldsymbol{s}_1$. With B bits, the receiver can quantize $\boldsymbol{s}_1$ and relay the quantized signature to user 1 via an error-free feedback channel. Then, user 1 adjusts its signature accordingly. Here we implicitly assume that channel is changing slowly and hence, a feedback from the receiver is meaningful.

A singular value decomposition gives

$$\boldsymbol{S}_1 \boldsymbol{S}_1^\dagger = \sum_{i=1}^{N} \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^\dagger \tag{4}$$

where $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$ are the ordered eigenvalues and $\boldsymbol{u}_i$ is the corresponding ith eigenvector. With B feedback bits, the quantization codebook

$$\mathcal{V} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{2^B}\} \tag{5}$$

where $\boldsymbol{v}_j$ is the $N \times 1$ codebook entry with $\|\boldsymbol{v}_j\| = 1$ and number of entries is 2^B. The optimal *unquantized* signature that minimizes the interference for user 1 is $\boldsymbol{u}_1$. We can quantize $\boldsymbol{u}_1$ with a nearest neighbor objective and the quantized signature is given by

$$\bar{\boldsymbol{s}}_1 = \arg\min_{\boldsymbol{v}_j \in \mathcal{V}} \|\boldsymbol{u}_1 - \boldsymbol{v}_j\|^2 = \arg\max_{\boldsymbol{v}_j \in \mathcal{V}} \boldsymbol{u}_1^\dagger \boldsymbol{v}_j, \tag{6}$$

which follows from the fact both $\|\boldsymbol{v}_j\| = \|\boldsymbol{u}_1\| = 1$. This is a classical vector quantization problem to which there are many solutions [10] (see references therein). If the objective is changed to minimizing the angle between $\boldsymbol{u}_1$ and $\boldsymbol{v}_j$ denoted by ϕ_j, then the quantized vector is given by

$$\tilde{\boldsymbol{s}}_1 = \arg\max_{\boldsymbol{v}_j \in \mathcal{V}} \cos^2 \phi_j = \arg\max_{\boldsymbol{v}_j \in \mathcal{V}} (\boldsymbol{u}_1^\dagger \boldsymbol{v}_j)^2. \tag{7}$$

With statistics of the eigenvector, we can apply the Lloyd-Max iterative algorithm [11], [12] to obtain the optimal $\mathcal{V}$, which minimizes the expected Euclidean distance between $\boldsymbol{u}_1$ and $\bar{\boldsymbol{s}}_1$ or the angle between $\boldsymbol{u}_1$ and $\tilde{s}_1$. However, either $\bar{\boldsymbol{s}}_1$ or $\tilde{s}_1$ may not necessarily minimize the interference power.

To minimize the interference power, the receiver selects

$$\hat{\boldsymbol{s}}_1 = \arg\min_{\boldsymbol{v}_j \in \mathcal{V}} \boldsymbol{v}_j^\dagger \boldsymbol{S}_1 \boldsymbol{S}_1^\dagger \boldsymbol{v}_j \tag{8}$$

for given $\boldsymbol{S}_1$. If the interfering signatures are i.i.d., we have shown that RVQ codebook, which contains independent and isotropically distributed vectors is optimal in a large system limit [1], [6]. The large system limit refers to the limit in which N, K, and B goes to infinity with a fixed normalized load $\bar{K} = K/N$ and normalized feedback bits $\bar{B} = B/N$. RVQ codebook is simple to construct. However, RVQ codebook and many codebooks proposed in the literature require exhaustive search to locate the optimal entry. This can pose a serious problem on complexity if B is large. Next we describe our proposed quantization codebook, which demands much less computational complexity, but sacrifices minimal performance.

III. Kd-Tree Codebook

We organize code entries in an RVQ codebook using Kd-tree algorithm proposed by [13], [14]. The algorithm produces an unbalanced binary search tree by clustering the code entries by one dimension at a time in each step. Steps to create Kd-tree is shown in Algorithms 1 and 2. We start at the root node with all 2^B entries from RVQ codebook $\mathcal{V} = \{\boldsymbol{v}_j\}$. We find a median for the *first* elements for all vectors and select the element that is the closest to the median. We call that element the *pivot*. Then vectors are divided into two groups by comparing whether the first element is greater or less than the pivot. We store the vector whose first element of each vector is the pivot at the root node and create the left- and right-child nodes with two groups of vectors. We move to the left-child node and find the pivot for the *second*-dimension elements of the vectors in that node. We divide the group of vectors according to the new pivot. Each time we move down the tree, we operate on the next dimension. We iterate until all nodes contain one code entry and obtain a binary search tree with 2^B nodes at the end. Building the tree is relatively fast since we examine one dimension at a time. Also it can be done offline and thus does not add burden on the transmitter-receiver pair.

A. Finding Nearest Neighbor

Given $\boldsymbol{u}_1$, Kd-tree search shown in Algorithms 3 and 4 [13], [14] will locate the entry that is the nearest neighbor to $\boldsymbol{u}_1$.

Algorithm 1 Kd-TREE construction [13], [14]

Start with RVQ codebook with 2^B N-dimensional vectors.
Store all entries at the root of a binary tree.
Set $n = 1$ where n is the index of element in an N-dimensional vector.
kd-tree = build_tree(root node, n)
return kd-tree

Algorithm 2 function: kd-tree = build_tree(node, n) [13], [14]

if Node contains only one vector. **then**
  return
else
  if $n > N$ **then**
    $n = 1$
  end if
  Sort codebook entries by the nth dimension.
  Select the median as pivot element.
  Use pivot to divide vectors in two groups.
  $n = n + 1$
  build_tree(left-child node, n)
  build_tree(right-child node, n)
end if

Algorithm 3 starts at the root and move down to the left- or right-child nodes by comparing the element in $\boldsymbol{u}_1$ in specified dimension with the pivot of the present node. Once we reach the leaf node, we have the candidate. Algorithm 4 makes certain that the candidate is the nearest neighbor by comparing the distance of the candidate and $\boldsymbol{u}_1$ with that of other nodes and $\boldsymbol{u}_1$. We remark that the selected entry may not minimize the interference and is the suboptimal entry.

Algorithm 3 Kd-tree search for the nearest neighbor

Start at the root of a Kd-tree codebook with $\boldsymbol{u}_1$.
Set $n = 1$ where n is the index of element in an N-dimensional vector.
while Not a leaf node. **do**
  if $n > N$ **then**
    $n = 1$
  end if
  $\boldsymbol{p}$ = current node's vector
  if $p_n > u_{1,n}$ **then**
    Move to the left-child node.
  else
    Move to the right-child node.
  end if
  $n = n + 1$
end while
$\boldsymbol{s}$ = vector from leaf node
$D = \|\boldsymbol{s} - \boldsymbol{u}_1\|$
$n = 1$
$\boldsymbol{s}$ = check_distance(root node, n, $\boldsymbol{u}_1$, D, $\boldsymbol{s}$)
return $\boldsymbol{s}$

Algorithm 4 function: $\boldsymbol{s}$ = check_distance(node, n, $\boldsymbol{u}_1$, D, $\boldsymbol{s}$)

if A leaf node **then**
  $\boldsymbol{s}_c$ = vector of a leaf node
  $D_c = \|\boldsymbol{s}_c - \boldsymbol{u}_1\|$
  if $D_c < D$ **then**
    $D = D_c$
    $\boldsymbol{s} = \boldsymbol{s}_c$
    return $\boldsymbol{s}$
  end if
end if
$D_p = p_n^2 - u_{1,n}^2$
if $D_p < D$ **then**
  if $n > N$ **then**
    $n = 1$
  end if
  if $p_n > u_{1,n}$ **then**
    $n = n + 1$
    check_distance(left-child node, n, $\boldsymbol{u}_1$, D, $\boldsymbol{s}$)
  else
    $n = n + 1$
    check_distance(right-child node, n, $\boldsymbol{u}_1$, D, $\boldsymbol{s}$)
  end if
else
  $n = n + 1$
  check_distance(left-child node, n, $\boldsymbol{u}_1$, D, $\boldsymbol{s}$)
  check_distance(right-child node, n, $\boldsymbol{u}_1$, D, $\boldsymbol{s}$)
end if

Performance of the Kd-tree codebook is difficult to evaluate analytically. Thus, we look to approximate the performance instead. We will approximate Kd-tree search with closest-in-angle search. The interference power with RVQ codebook with closest-in-angle search is given by

$$\tilde{I}_{\mathrm{rvq}} = \tilde{\boldsymbol{s}}_1^\dagger \boldsymbol{S}_1 \boldsymbol{S}_1^\dagger \tilde{\boldsymbol{s}}_1 \quad (9)$$

$$= \lambda_1 (\tilde{\boldsymbol{s}}_1^\dagger \boldsymbol{u}_1)^2 + \sum_{i=2}^{N} \lambda_i (\tilde{\boldsymbol{s}}_1^\dagger \boldsymbol{u}_i)^2 \quad (10)$$

where $\tilde{\boldsymbol{s}}_1$ is the signature for user 1 that is closest in angle to $\boldsymbol{u}_1$ given by (7) and the second equation is obtained by substituting (4). Finding expectation for $\tilde{I}_{\mathrm{rvq}}$ for finite N, K, and B is not tractable due to very complicated distribution expressions. However, as $(N, K, B) \to \infty$, an eigenvalue distribution of $\boldsymbol{S}_1 \boldsymbol{S}_1^\dagger$ converges to a deterministic function and the smallest eigenvalue λ_1 converges to [16]

$$\lambda_1 \longrightarrow \lambda_{\min}^{\infty} = \begin{cases} 0 & : \ 0 \le \bar{K} \le 1 \\ (1 - \frac{1}{\sqrt{\bar{K}}})^2 & : \ \bar{K} > 1 \end{cases} . \quad (11)$$

We have shown in [17] that as $(N, K, B) \to \infty$

$$(\tilde{\boldsymbol{s}}_1^\dagger \boldsymbol{u}_1)^2 \to 1 - 2^{-2\bar{B}} \quad (12)$$

almost surely. The large system $(\tilde{\boldsymbol{s}}_1^\dagger \boldsymbol{u}_1)^2$ increases with $\bar{B}$. With infinite feedback ($\bar{B} = \infty$), we can relay $\boldsymbol{u}_1$ unquantized and $(\tilde{\boldsymbol{s}}_1^\dagger \boldsymbol{u}_1)^2 \to 1$. On the other hand, with no feedback, $\tilde{\boldsymbol{s}}$ is randomly selected and $(\tilde{\boldsymbol{s}}_1^\dagger \boldsymbol{u}_1)^2 \to 0$.

Next, we analyze each term in the sum in (10). Since $\boldsymbol{u}_1$ are orthogonal to all $\boldsymbol{u}_i$, as B increases, $\tilde{\boldsymbol{s}}$ and $\boldsymbol{u}_i$ is getting closer be perpendicular. We can show that

Lemma 1: For large N with fixed $\bar{K}$ and $\bar{B}$,

$$(\tilde{\boldsymbol{s}}_1^\dagger \boldsymbol{u}_i)^2 = \frac{1}{N} 2^{-2\bar{B}} + \mathcal{O}(\frac{1}{N^2}). \tag{13}$$

A core of the proof is to derive the distribution of $(\tilde{\boldsymbol{s}}_1^\dagger \boldsymbol{u}_i)^2$, which is equivalent to a partial surface of the N-dimensional unit hypersphere. We substitute (13) into the sum on the right-hand side of (10) and obtain

$$\sum_{i=1}^{N} \lambda_i (\tilde{\boldsymbol{s}}_1^\dagger \boldsymbol{u}_i)^2 = \frac{1}{N} \sum_{i=1}^{N} \lambda_i 2^{-2\bar{B}} + \mathcal{O}(\frac{1}{N}). \tag{14}$$

As $(N, K) \to \infty$, it can be shown that [16]

$$\frac{1}{N} \sum_{i=1}^{N} \lambda_i \to \bar{K}. \tag{15}$$

Applying Lemma 1 and combining (10)–(15), we have

Theorem 1: As $(N, K, B) \to \infty$ with fixed $\bar{K}$ and $\bar{B}$,

$$\tilde{I}_{\text{rvq}} \to \tilde{I}_{\text{rvq}}^{\infty} \tag{16}$$

$$= \begin{cases} \bar{K} 2^{-2\bar{B}} & : 0 \le \bar{K} \le 1 \\ (1 - \frac{1}{\sqrt{\bar{K}}})^2 (1 - 2^{-2\bar{B}}) + \bar{K} 2^{-2\bar{B}} & : \bar{K} > 1 \end{cases} \tag{17}$$

almost surely.

Hence, the associated SINR for RVQ that quantizes $\boldsymbol{u}_1$ is given by

$$\tilde{\gamma}_{\text{rvq}}^{\infty} = \frac{1}{\tilde{I}_{\text{rvq}}^{\infty} + \sigma_n^2}. \tag{18}$$

Thus, performance of Kd-tree is approximated by the result in Theorem 1.

B. Minimizing Interference Power

We modify the objective function in the search algorithm in Algorithms 3 and 4 to be the interference power for a given interfering signature matrix $\boldsymbol{S}_1$. The modified search shown in Algorithms 5 and 6 is in the same spirit as the earlier search algorithm. At each step, we look for the entry, which gives the interference power, which is closest to the minimum.

These algorithms however are not guaranteed to find the entry that minimizes the interference power since the tree we used is constructed with the nearest neighbor criterion. The solution will be suboptimal to the exhaustive search. However, as we will observe from simulation results, this Kd-tree with modified search performs very close to RVQ with exhaustive search.

Again, analyzing the performance of this scheme exactly is not tractable. So, we look to analyze the performance of RVQ. Reference [18] has shown that

$$\hat{I}_{\text{rvq}} = \hat{\boldsymbol{s}}^\dagger \boldsymbol{S}_1 \boldsymbol{S}_1^\dagger \hat{\boldsymbol{s}} \to \hat{I}_{\text{rvq}}^{\infty} \tag{19}$$

and $\hat{I}_{\text{rvq}}^{\infty}$ is a function of $\bar{K}$ and $\bar{B}$, which can be used to approximate the performance of Kd-tree.

Algorithm 5 Kd-tree search for the minimum interference

```
Start at the root of a Kd-Tree codebook with interfering signature matrix S_1.
Set n = 1 where n is the index of element in an N-dimensional vector.
while Not a leaf node do
    if n > N then
        n = 1
    end if
    p_L^(n) = [0 0 ... p_{L,n+1} ... 0 0]^† where p_L is the vector of the left-child node.
    p_R^(n) = [0 0 ... p_{R,n+1} ... 0 0]^† where p_R is the vector of the right-child node.
    if p_L^(n)† S_1 S_1^† p_L^(n) < p_R^(n)† S_1 S_1^† p_R^(n) then
        Move to the left child node.
    else
        Move to the right child node.
    end if
    n = n + 1
end while
s = vector from leaf node.
I = s^† S_1 S_1^† s
n = 1
s = check_interference(root node, n, S_1, I, s)
return s
```

IV. Numerical Results

Fig. 1 shows the SINR for user 1 versus normalized feedback bits $\bar{B}$ for loads $\bar{K} = 0.5$ and $\bar{K} = 1.2$, and SNR $= 10$ dB. Simulation results shown are for RVQ and Kd-tree codebooks with closest in angle and nearest neighbor criterions, respectively. As expected, SINR increases with $\bar{B}$. User 1 can achieve 7 dB and 4 dB with only one feedback bit for $\bar{K} = 0.5$ and $\bar{K} = 1.2$ as compared to 3 dB and -1 dB for zero feedback. Also shown is the asymptotic performance for RVQ with closest in angle criterion with $(N, K, B) \to \infty$. We can see that the large system limit approximates the performance of a small-size system very well. We expect a gap between simulation and large system results to narrow as N increases.

Figs. 2 and 3 show the performance of RVQ with closest in angle, and Kd-tree and PAM with nearest neighbor. All three performs approximately the same for a given feedback except when $\bar{B} < 1$, PAM codebook is not valid. We note that for $\bar{B}$ between 1 and 2, PAM performs a bit worse than the other two do. In Fig. 3, we show a number of inner product computations required for all three codebooks to locate the selected entries for a given SINR. We see that RVQ with exhaustive search requires significantly larger number of computations. For SINR at 7 dB, the complexity of RVQ is four orders of magnitude larger than that of Kd-tree and PAM. For low to moderate SINR, PAM codebook is the least complex while for the other range, it is unclear. Due to finite storage and computational power, we do not have simulation

Algorithm 6 function: $\boldsymbol{s}$ = check_interference(node, n, $\boldsymbol{S}_1$, I, $\boldsymbol{s}$)

if Leaf node. **then**
  $\boldsymbol{s}_c =$ vector from leaf node.
  $I_c = \boldsymbol{s}_c^\dagger \boldsymbol{S}_1 \boldsymbol{S}_1^\dagger \boldsymbol{s}_c$
  if $I_c < I$ **then**
    $I = I_c$
    $\boldsymbol{s} = \boldsymbol{s}_c$
    return $\boldsymbol{s}$
  end if
end if
$\boldsymbol{p}_c^{(n)} = [0 \quad 0 \ \dots \ p_{c,n} \ \dots \ 0 \quad 0]^\dagger$ where $\boldsymbol{p}_c$ is the vector of the current node.
$I_p = \boldsymbol{p}_c^{(n)\dagger} \boldsymbol{S}_1 \boldsymbol{S}_1^\dagger \boldsymbol{p}_c^{(n)}$
if $I_p < I$ **then**
  if $n > N$ **then**
    $n = 1$
  end if
  if $\boldsymbol{p}_L^{(n)\dagger} \boldsymbol{S}_1 \boldsymbol{S}_1^\dagger \boldsymbol{p}_L^{(n)} < \boldsymbol{p}_R^{(n)\dagger} \boldsymbol{S}_1 \boldsymbol{S}_1^\dagger \boldsymbol{p}_R^{(n)}$ **then**
    $n = n + 1$
    check_interference(left-child node, n, $\boldsymbol{S}_1$, I, $\boldsymbol{s}$)
  else
    $n = n + 1$
    check_interference(right-child node, n, $\boldsymbol{S}_1$, I, $\boldsymbol{s}$)
  end if
else
  $n = n + 1$
  check_interference(left-child node, n, $\boldsymbol{S}_1$, I, $\boldsymbol{s}$)
  check_interference(right-child node, n, $\boldsymbol{S}_1$, I, $\boldsymbol{s}$)
end if

results for larger $\bar{B}$ for both RVQ and Kd-tree. For large feedback, Kd-tree needs significantly larger storage for code entries while PAM does not have this issue.

In Fig. 4, we have SINR from three quantization schemes: RVQ and Kd-tree with minimizing interference and PAM with nearest neighbor for $\bar{K} = 0.5$ and SNR $= 10$ dB. We see that Kd-tree performs almost the same as RVQ does while PAM with nearest neighbor performs significantly worse for smaller $\bar{B}$. At $\bar{B} = 1$, a gap between quantizing to minimize interference and to find the nearest neighbor is as large as 5 dB. We also show the large system performance for RVQ derived by [18], which seems to approximate performance of a finite-size system very closely.

We also examine computational complexity of each quantization schemes with Fig. 5. The PAM codebook, which finds the nearest neighbor to the optimal eigenvector is the least complex. However, from Fig. 4, its performance is the worst for a given feedback bits. So, what codebook to use depends on the system need and requirement.

V. Conclusions

We have presented different quantization codebooks to quantize the user signature in reverse-link CDMA. Our main focus is Kd-tree codebook, which was motivated by the need to lessen search complexity. Kd-tree codebook was shown to perform close to RVQ with significantly less search complexity. We also derived the performance approximation for Kd-tree with nearest neighbor and showed that it is a good indicator for the actual performance. Numerical examples show tradeoff between performance and complexity for different codebooks. RVQ with exhaustive performs best for a given feedback while PAM is the least complex. Kd-tree is a compromise of the two schemes.

We can straightforwardly extend our model to include fading and other channel characteristics. We expect to see similar performance behaviors for all three codebooks. Here we assume that the receiver knows signatures of other users (or CSI) perfectly. The performance will degrade with the estimation of CSI or interference information.

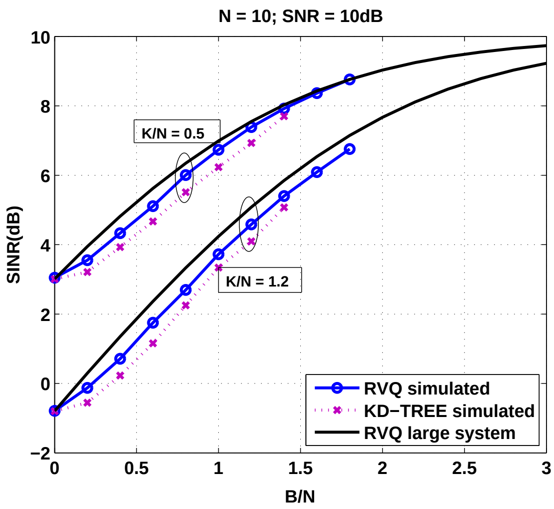

Fig. 1. SINR versus normalized feedback bits are shown for RVQ with closest in angle criterion and Kd-tree with nearest neighbor criterion.

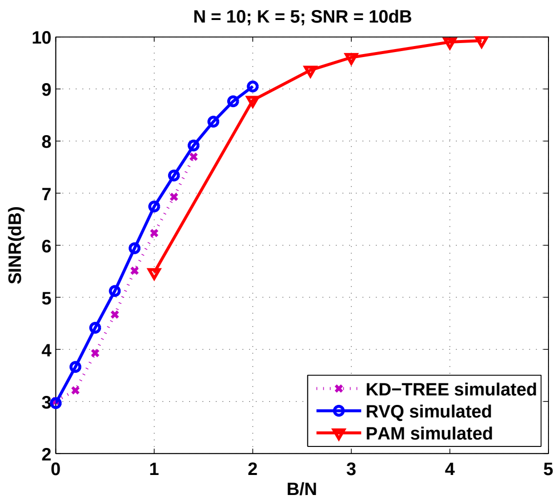

Fig. 2. SINR versus normalized feedback bits are shown for RVQ with closest in angle criterion, and Kd-tree and PAM with nearest neighbor criterion.

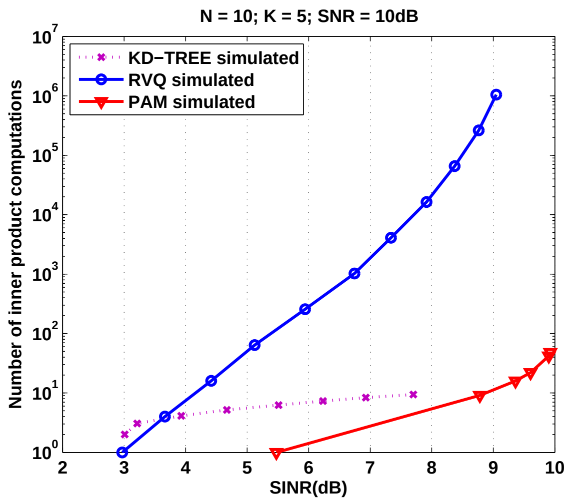

Fig. 3. A number of inner product computations is displayed for RVQ with closest in angle criterion, and Kd-tree and PAM with nearest neighbor criterion for a given SINR.

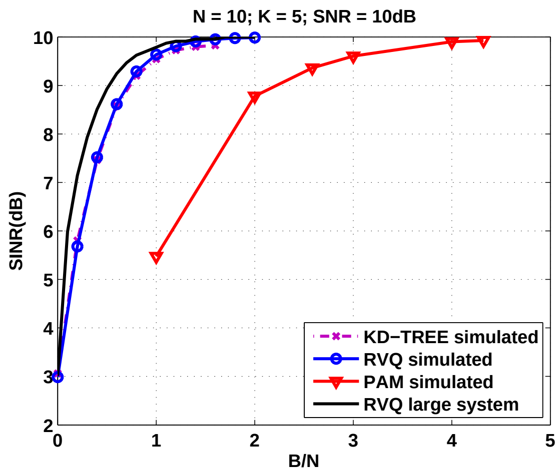

Fig. 4. SINR versus normalized feedback bits are shown for RVQ and Kd-tree with minimizing interference, and PAM with nearest neighbor criterion.

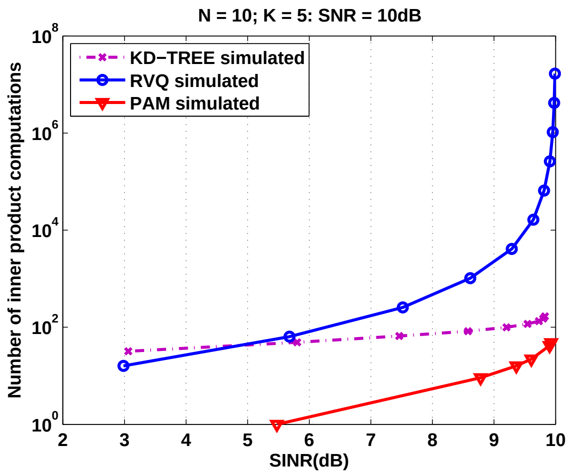

Fig. 5. A number of inner product computations is displayed for RVQ and Kd-tree with minimizing interference and PAM with nearest neighbor criterion for a given SINR.

REFERENCES

[1] W. Santipach and M. L. Honig, "Signature optimization for CDMA with limited feedback," *IEEE Trans. Inf. Theory*, vol. 51, no. 10, pp. 3475–3492, Oct. 2005.

[2] D. J. Love and R. W. Heath, Jr., "Grassmannian beamforming for multiple-input multiple-output wireless systems," *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2735–2745, Oct. 2003.

[3] K. K. Mukkavilli, A. Sabharwal, E. Erkip, and B. Aazhang, "On beamforming with finite rate feedback in multiple antenna systems," *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2562–2579, Oct. 2003.

[4] A. Narula, M. J. Lopez, M. D. Trott, and G. W. Wornell, "Efficient use of side information in multiple antenna data transmission over fading channels," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 8, pp. 1423–1436, Oct. 1998.

[5] J. C. Roh and B. D. Rao, "Transmit beamforming in multiple-antenna systems with finite rate feedback: A VQ-based approach," *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 1101–1112, Mar. 2006.

[6] W. Santipach and M. L. Honig, "Signature optimization for DS-CDMA with limited feedback," in *Proc. IEEE Int. Symp. on Spread-Spectrum Tech. and Appl. (ISSSTA)*, Prague, Czech Republic, Sep. 2002, pp. 180–184.

[7] D. J. Love, R. W. Heath, Jr., W. Santipach, and M. L. Honig, "What is the value of limited feedback for MIMO channels?" *IEEE Commun. Mag.*, vol. 42, no. 10, pp. 54–59, Oct. 2004.

[8] D. J. Ryan, I. V. L. Clarkson, I. B. Collings, D. Guo, and M. L. Honig, "Qam and psk codebooks for low-complexity limited feedback mimo beamforming," *IEEE Trans. Commun.*, vol. 57, no. 4, pp. 1184–1196, Apr. 2009.

[9] K. Mamat and W. Santipach, "Multiuser signature quantization with tree-structured codebook in DS-CDMA," in *Proc. ECTI-CON*, vol. 2, Pattaya, Thailand, May 2009, pp. 870–873.

[10] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Springer, 1991.

[11] J. Max, "Quantizing for minimum distortion," *IEEE Trans. Inf. Theory*, vol. 6, pp. 7–12, Mar. 1960.

[12] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982.

[13] J. L. Bentley, "Multidimisional divide and conquer," *Communications of the ACM*, vol. 22, no. 4, pp. 214–229, 1980.

[14] ——, "K-d trees for semidynamic point sets," in *Proc. the Sixth Annual Symposium on Computational Geometry*, Berkley, California, 1990, pp. 187 – 197.

[15] S. Verdú and S. Shamai, "Spectral efficiency of CDMA with random spreading," *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 622–640, Mar. 1999.

[16] A. M. Tulino and S. Verdú, "Random matrix theory and wireless communications," *Foundations and Trends in Communications and Information Theory*, vol. 1, no. 1, pp. 1–182, 2004.

[17] W. Santipach and M. L. Honig, "Capacity of a multiple-antenna fading channel with a quantized precoding matrix," *IEEE Trans. Inf. Theory*, vol. 55, no. 3, pp. 1218–1234, Mar. 2009.

[18] W. Dai, Y. Liu, and B. Rider, "The effect of finite rate feedback on CDMA signature optimization and MIMO beamforming vector selection," *IEEE Trans. Inf. Theory*, vol. 55, no. 8, pp. 3651–3669, Aug. 2009.