Diophantine Index Assignments for Distributed Source Coding

Gerhard Maierbacher   João Barros

Instituto de Telecomunicações
Department of Computer Science, Universidade do Porto,
R. Campo Alegre 1021, 4150-180 Porto, Portugal
Email: {gerhard, barros}@dcc.fc.up.pt

***Abstract*— We consider the design of index assignments for the distributed source coding problem in large-scale sensor networks. Using basic tools from number theory, specifically Diophantine analysis, we provide a framework for constructing cyclic index assignments that have very low complexity yet perform very close to fundamental bounds provided by rate-distortion theory.**

I. Introduction

In distributed sensing scenarios, where correlated sets of data are gathered by a large number of power-restricted sensors, efficient source coding techniques are key towards reducing the required number of transmissions and enabling extended network life-time. Inspired by the seminal work of Slepian and Wolf [10], which characterizes the fundamental limits of separate encoding of correlated sources with arbitrarily small probability of error, several authors have contributed with practical coding solutions for this problem (see e.g. [13] and references therein). For continuous-valued sources subject to common distortion criteria, [2], [5] and [1] proposed different (heuristic) optimization algorithms for the case of two correlated sources. Exploiting the duality between Slepian-Wolf coding and channel coding, [8] and [9] use syndromes of channel codes with appropriate distance properties to produce a class of simple distributed block codes. Similarly, [3] provides an encoding concept based on bit-puncturing relying on the error-correcting capabilities of highly evolved turbo-codes.

Our take is to exploit the symmetries within common source models (e.g. multivariate Gaussian distributions) and elementary properties of integer numbers to devise scalable distributed source codes with very low complexity.

We consider the scenario where N correlated sources $U_1, U_2, \dots, U_N$, with output symbols $u_n \in \mathcal{U}_n$, $n \in \{1, 2, \dots, N\}$, are drawn according to the joint probability distribution $p(u_1, u_2, \dots, u_N)$. The observations are encoded independently on a single-letter basis onto the codewords $W_1, W_2, \dots, W_N$, with realizations $w_n \in \mathcal{W}_n$, $n \in \{1, 2, \dots, N\}$, i.e. each observed symbol is mapped onto a single codeword (see Figure 1). The joint decoder knows $p(u_1, u_2, \dots, u_N)$ produces the estimates $\hat{U}_1, \hat{U}_2, \dots, \hat{U}_N$, with realizations $\hat{u}_n \in \hat{\mathcal{U}}_n \subseteq \mathcal{U}_n$, $n \in \{1, 2, \dots, N\}$, subject to a distortion criterion (to be specified later). Based on this coding scheme, distributed data compression is achieved by choosing codeword alphabets $\mathcal{W}_1, \mathcal{W}_2, \dots, \mathcal{W}_N$ such that $|\mathcal{W}_n| = K_n \leq |\mathcal{U}_n|$, $n \in \{1, 2, \dots, N\}$.

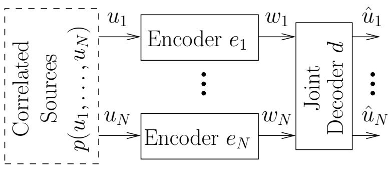


Fig. 1. N correlated sources are independently encoded and jointly decoded. Each transmitter e_n encodes the observed source symbol u_n onto a separate codeword w_n, $n = 1, 2, \dots, N$. The joint decoder d uses the received codewords $w_1, w_2, \dots, w_N$ and its knowledge about the source statistics $p(u_1, u_2, \dots, u_N)$ to jointly form the estimates $\hat{u}_1, \hat{u}_2, \dots, \hat{u}_N$.

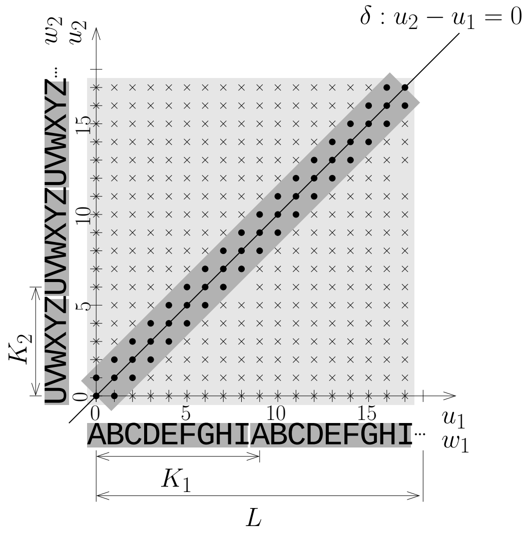


Fig. 2. Example of cyclic encoding. The source symbols u_1, u_2, with $u_1, u_2 \in \mathcal{U} = \{0, 1, \dots, 17\}$, are mapped in a cyclic fashion onto the codewords w_1, w_2, with $w_1 \in \mathcal{W}_1 = \{A, B, C, D, E, F, G, H, I\}$ and $w_2 \in \mathcal{W}_2 = \{U, V, W, X, Y, Z\}$. Codeword pairs located at the positions indicated by dots reappear only at the positions indicated by crosses. Within the shaded area area around the line $\delta : \; u_2 - u_1 = 0$ there are no duplicate codeword pairs.

For encoding, we propose a simple class of deterministic mapping functions that can be viewed as *cyclic* index assignments: Assuming that the symbols of the source alphabet are ordered, all input symbols u_n differing by K_n positions within the initial order are mapped onto the same output codeword w_n, $n \in \{1, 2, \dots, N\}$. Due to the cyclic structure of the resulting mapping, the encoders can be fully characterized by the size of their alphabets K_n. The task at hand is then to jointly find $K_1, K_2, \ldots, K_N$ leading to favorable rate-distortion trade-offs — we shall show that this can be achieved by means of linear Diophantine equations.

To gain some intuition, consider the simple example with two sources shown in Figure 2, which assumes integer-valued source alphabets $\mathcal{U}_1 = \mathcal{U}_2 = \mathcal{U} = \{0, 1, \ldots, L-1\}$, with $L = 18$. The goal is to construct index assignment functions such that $K_1, K_2 \leq \frac{L}{f}$, in order to reduce the data rate from $R = \log_2(L)$ to $R' = \log_2(\frac{L}{f})$ [bits/symbol]. For *reuse-factor* $f = 2$ choosing $K_1 = 9$ and $K_2 = 6$ leads to favorable code properties as illustrated in Figure 2. Specifically it is worth pointing out that all symbol pairs (u_1, u_2) lying within the shaded area around the line $\delta: \ u_2 - u_1 = 0$ are mapped onto different codeword pairs (w_1, w_2), i.e. there are no duplicate codeword pairs within the shaded area. Assuming that only symbol pairs within the shaded area are likely to be generated by the source (e.g. because of strong correlation between u_1 and u_2) then the decoder can reconstruct the original symbol pairs from the received codeword pairs most of the time without error. It is worth mentioning that the shaded area around the line $\delta: \ u_2 - u_1 = 0$ contains exactly those symbol pairs which are the most probable ones for large classes of source models (e.g. bi-variate Gaussian source with subsequent quantization).

The rest of this paper is organized as follows: Section II presents the system model, provides a general description of the coding scheme and states the fundamental coding problem. Our code construction is presented in Section III, and finally Section IV offers numerical examples and concluding remarks.

II. Problem Setup

A. Notation

We start by introducing our notation. Random variables are always denoted by capital letters, e.g. U, where its realizations are denoted by the corresponding lowercase letters, e.g. u. Vectors are denoted by bold letters and (if not stated differently) assumed to be column vectors, e.g. $\mathbf{u} = (u_1, u_2, \ldots, u_N)^T$. The expression $\mathbf{0}_N = (0, 0, \ldots, 0)^T$ is the length-N zero vector and similarly $\mathbf{1}_N = (1, 1, \ldots, 1)^T$ is the length-N one vector. Index sets are denoted by capital calligraphic letters $\mathcal{N}$, unless otherwise noted, and $|\mathcal{N}|$ denotes the set's cardinality. We follow the convention, that variables indexed by a set denote a set of variables, e.g. if $\mathcal{M} = \{1, 2, 3\}$ then $u_{\mathcal{M}} = \{u_1, u_2, u_3\}$, and use the same concept to define variable vectors, such that $\mathbf{u}_{\mathcal{M}} = (u_1, u_2, u_3)^T$.

B. Source Model and Geometric Definitions

We consider a system-setup with N correlated sources $U_1, U_2, \ldots, U_N$, generating output symbols $u_n \in \mathcal{U}_n$, $n = 1, 2, \ldots, N$. The output symbols are collected in the vector $\mathbf{u} = (u_1, u_2, \ldots, u_N)^T \in \mathcal{U} = \mathcal{U}_1 \times \mathcal{U}_2 \times \ldots \times \mathcal{U}_N$ and we assume that the joint probability density function (PDF) $p(\mathbf{u})$ is known. For digital data processing, we consider the discrete representations of the source symbols $I_1, I_2, \ldots, I_N$, with realizations $i_n \in \mathcal{I}_n$, $n = 1, 2, \ldots, N$. This discrete representation can be obtained e.g. by quantization. We define the vector $\mathbf{i} = (i_1, i_2, \ldots, i_N)^T \in \mathcal{I} = \mathcal{I}_1 \times \mathcal{I}_2 \times \ldots \times \mathcal{I}_N$ to collect the discrete representations and assume that $p(\mathbf{i})$ is the resulting probability mass function (PMF).

Considering the vectors $\mathbf{i} \in \mathcal{I}$ as *points* in the Euclidean space $\mathbb{R}^N$, we specifically consider PMF's $p(\mathbf{i})$ characterized by the fact that all points $\mathbf{i}$ located around some *symmetry axis* $\alpha \in \mathbb{R}^N$ have high probability. This assumption may seem somewhat restrictive, however it is satisfied by several relevant source models. Figure 3 shows an example.

The symmetry axes is expressed in parametric vector form as

$$\alpha(\mathbf{m}, \mathbf{n}): \ \mathbf{x} = \mathbf{m} + t \cdot \mathbf{n}, \tag{1}$$

i.e. a line passing through point $\mathbf{m} = (m_1, m_2, \ldots, m_N)^T$, $\mathbf{m} \in \mathbb{R}^N$, and with direction vector $\mathbf{n} = (n_1, n_2, \ldots, n_N)^T$, $\mathbf{n} \in \mathbb{R}^N$, $\mathbf{n} \neq \mathbf{0}_N$ where $t \in \mathbb{R}$ is an arbitrary parameter.

Specifically, we shall consider the symmetry axis of the form

$$\delta: \ \mathbf{x} = \mathbf{0}_N + t \cdot \mathbf{1}_N, \tag{2}$$

henceforth called *main diagonal*.

To quantify the distance between an arbitrary point $\mathbf{i}$ and δ, we use the minimum Euclidean distance

$$l(\mathbf{i}) = \min_{\mathbf{h} \in \mathbb{R}^N : \mathbf{h} \in \delta} \{||\mathbf{i} - \mathbf{h}||\} \tag{3}$$

and assume that all points $\mathbf{i} \in \mathbb{Z}^N$ with high probability $p(\mathbf{i})$ have a distance less or equal to some *radius* $r \in \mathbb{R}_0^+$.

For the code design procedure presented in this work, we consider lines parallel to δ. By defining $c_l = m_{l+1} - m_1$, $l = 1, \ldots, N-1$, and $\mathbf{c} = (c_1, c_2, \ldots, c_{N-1})^T \in \mathbb{Z}^{N-1}$ those lines can be fully characterized by the parameter $\mathbf{c}$ and we define the *subdiagonal* at *position* $\mathbf{c}$ as

$$\gamma(\mathbf{c}): \ x_{l+1} - x_1 = c_l, \ l = 1, 2, \ldots, N-1. \tag{4}$$

Since the subdiagonals $\gamma(\mathbf{c})$ are parallel to the main diagonal, all points $\mathbf{i}$ on the same subdiagonal also have the same Euclidean distance to the main diagonal and it can be shown that (3) is equal to

$$L(\mathbf{c}) = \sqrt{\sum_{l=1}^{N-1} c_l^2 - \frac{1}{N}\Big(\sum_{l=1}^{N-1} c_l\Big)^2}. \tag{5}$$

for all those points.

Let $\mathbf{j}(\mathbf{c})$ be an arbitrary point on the subdiagonal $\gamma(\mathbf{c})$, henceforth called the *reference point* of $\gamma(\mathbf{c})$.

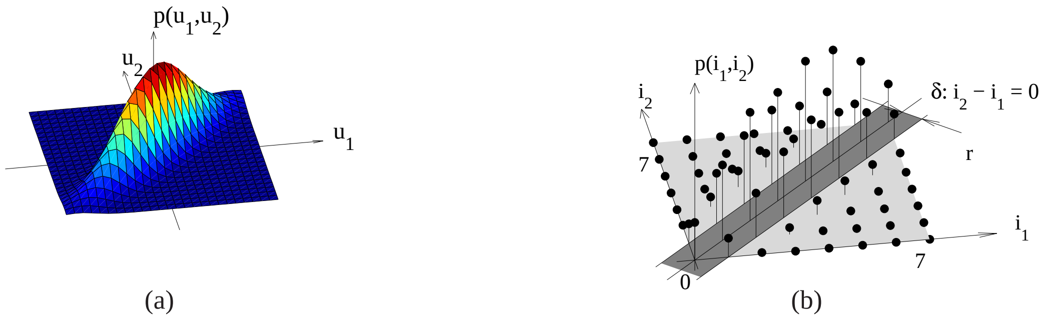
The *segment* on the subdiagonal $\gamma(\mathbf{c})$ of *length* $V(\mathbf{c}) \in \mathbb{Z}$ is defined as $\mathcal{S}(\mathbf{j}(\mathbf{c}), V(\mathbf{c})) = \{\mathbf{i} \in \mathbb{Z}^N : \mathbf{i} = \mathbf{j}(\mathbf{c}) + t \cdot \mathbf{1}_N, t = 0, 1, \ldots, V(\mathbf{c})\}$.

The *cylinder* $\mathcal{C}(r)$ with radius $r \in \mathbb{R}_0^+$ is defined as the union of all diagonal segments $\mathcal{S}(\mathbf{j}(\mathbf{c}), V(\mathbf{c}))$ with a Euclidean distance $L(\mathbf{c}) \leq r$, i.e. $\mathcal{C}(r) = \{\mathcal{S}(\mathbf{j}(\mathbf{c}), V(\mathbf{c})) : L(\mathbf{c}) \leq r\}$.

C. Encoding Scheme

Each source U_n, $n \in \{1, 2, \ldots, N\}$, is processed by a separate encoder, which does not know the source symbols observed by the other encoders.

As shown in Figure 4, in the first stage the observed source symbol u_n is mapped onto the *source index* $i_n \in \mathcal{I}_n$ by the



(a) (b)

Fig. 3. Properties of correlated Gaussian sources: (a) PDF of symbol pairs $\mathbf{u}=(u_1,u_2)^T$ given by a bi-variate Gaussian distribution $p(\mathbf{u})$; (b) PMF of the index pairs $\mathbf{i}=(i_1,i_2)^T$ resulting from $\mathbf{u}$ after (independent) Lloyd-Max quantization with resolution 3 bit. The index pairs $\mathbf{i}$ within the proximity of the symmetry axis $\delta: i_2-i_1=0$, i.e. with a distance smaller or equal r, have high probability $p(\mathbf{i})$.

ranking function $r_n: \mathcal{U}_n \rightarrow \mathcal{I}_n$ such that $i_n = r_n(u_n)$ where r_n is surjective. The ranking function models any form of preprocessing prior to encoding. For example, in the case of continuous-valued sources the ranking function can be seen as a standard scalar quantizer.

After ranking the discrete-valued source indices i_n are mapped onto the codewords $w_n \in \mathcal{W}_n$ by the *index assignment function* $m_n: \mathcal{I}_n \rightarrow \mathcal{W}_n$ such that $w_n = m_n(i_n)$ where m_n is again surjective. In this work, we specifically consider the case where $|\mathcal{W}_n| \leq |\mathcal{I}_n|$, i.e. we have less codewords w_n than source indices i_n, in order to achieve data-compression.

We restrict our attention to a special class of index assignment functions, which we now define. Assuming, that $|\mathcal{I}_n| = L$ and $\mathcal{I}_n = \{0, 1, \ldots, L-1\}$ and that $|\mathcal{W}_n| = K_n$ then *cyclic* index assignment functions are characterized by the fact that all indices $i_n \in \mathcal{I}_n$ congruent modulo K_n are mapped onto the same codeword w_n, i.e. all indices $i_n \in \mathcal{I}_n$ with $i_n (\bmod K_n) = j_n$, for some constant $j_n \in \{0, 1, \ldots, K_n-1\}$, are mapped onto the same codeword $w_n = m_n(j_n)$.

In summary, each encoder operates in a sequential way: The codeword $w_n \in \mathcal{W}_n$ is obtained from the source symbol $u_n \in \mathcal{U}_n$ by the *encoding function* $e_n: \mathcal{U}_n \rightarrow \mathcal{W}_n$, where e_n is given by the composition $e_n = m_n \circ r_n$ such that $w_n = m_n(r_n(u_n)) = e_n(u_n)$.

We shall sometimes need to consider the *global index assignment function* corresponding to $m: \mathcal{I} \rightarrow \mathcal{W}$ such that $\mathbf{w} = m(\mathbf{i}) = (m_1(i_1), m_2(i_2), \ldots, m_N(i_N))^T$ for the *index vector* $\mathbf{i} = (i_1, i_2, \ldots, i_N)^T$.

The *data rate* for transmitting the codeword w_n to the decoder is defined as $R_n = \lceil \log_2(K_n) \rceil$ [bit/codeword].

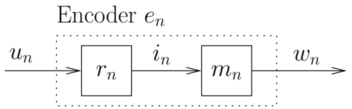


Fig. 4. Two-Stage Encoder. The observed source symbols u_n are encoded one-to-one in a sequential fashion onto the codewords w_n. In the first stage, the discrete source index i_n is obtained from u_n by the ranking function r_n and w_n is obtained subsequently by the index assignment function m_n. The encoding function is thus given by $e_n = m_n \circ r_n$.

D. Decoding and Design Criteria

After error-free transmission, the decoder uses the received *codeword vector* $\mathbf{w} = (w_1, w_2, \ldots, w_N)^T \in \mathcal{W} = \mathcal{W}_1 \times \ldots \times \mathcal{W}_N$ and its knowledge about the joint statistics to form estimates $\hat{\mathbf{u}} = (\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_N,) \in \hat{\mathcal{U}} = \hat{\mathcal{U}}_1 \times \hat{\mathcal{U}}_2 \times \ldots \times \hat{\mathcal{U}}_N$. The decoding function is defined as $d: \mathcal{W} \rightarrow \hat{\mathcal{U}}$ such that $\hat{\mathbf{u}} = d(\mathbf{w})$ minimizes the chosen fidelity criterion. Particular cases shall be considered in Section IV.

The set of index vectors $\mathbf{i} \in \mathcal{I}$ encoded onto a certain codeword vector $\mathbf{w}$ is defined as

$$\mathcal{Q}(\mathbf{w}, \mathcal{I}) = \{\mathbf{i} \in \mathcal{I} : m(\mathbf{i}) = \mathbf{w}\}. \tag{6}$$

We say that $\mathcal{Q}(\mathbf{w}, \mathcal{I})$ is the *originating set* for $\mathbf{w}$. Let $\mathcal{A} \subseteq \mathcal{I}$ be an arbitrary subset of index vectors in which we are interested and call it the *admissible* set. The *decoded set* $\mathcal{D}(\mathbf{w}, \mathcal{I})$ for a certain codeword vector $\mathbf{w}$ is then defined as the as the set of all index vectors $\mathbf{i} \in \mathcal{I}$ that are in the admissible set $\mathcal{A}$ as well as the originating set $\mathcal{Q}(\mathbf{w}, \mathcal{I})$ for $\mathbf{w}$, i.e.

$$\mathcal{D}(\mathbf{w}, \mathcal{I}) = \mathcal{Q}(\mathbf{w}, \mathcal{I}) \cap \mathcal{A}. \tag{7}$$

We say that a certain index vector $\mathbf{i}$ is *decodable*, if it can be recovered from the resulting codeword vector $\mathbf{w}$ with zero-error and, likewise, we say that the set of index vectors $\mathcal{P} \subseteq \mathcal{A}$ is decodable, if all $\mathbf{i} \in \mathcal{P}$ are decodable.

In the following, the index vector $\mathbf{i}$ shall also be referred to as *index tuple* in cases where the term *vector* could lead to misconceptions and, equally, we shall refer to the codeword vector $\mathbf{w}$ as the *codeword tuple*.

E. Problem Statement

Considering the cylinder $\mathcal{C}(r)$ as admissible set $\mathcal{A}$, the goal of our distributed source coding problem is to (jointly) design the cyclic index assignments such that the cylinder $\mathcal{C}(r)$ is decodable while (at the same time) the radius r is maximized. This is to be achieved by an appropriate choice of $\mathbf{K} = (K_1, K_2, \ldots, K_N)^T$.

III. Code Design

Considering the characteristic properties of the cyclic index assignments, it is possible to provide the mathematical means for code analysis giving rise to efficient, non-heuristic design.

The following Lemma is useful when considering code design under the decodability criterion:

Lemma 1: (Zero-Error Decodability)
(a) If the decoded set $\mathcal{D}(\mathbf{w}, \mathcal{I})$ does not have more than a single member for all codeword tuples $\mathbf{w}$ resulting from an index tuple $\mathbf{i}$ within the admissible set $\mathcal{A} \subseteq \mathcal{I}$, then $\mathcal{A}$ is decodable.
(b) A sufficient condition for the requirement in (a) is that all index tuples $\mathbf{i}$ within the admissible set $\mathcal{A}$ are encoded on different codeword tuples $\mathbf{w}$.

Proof: For lack of space, please refer to [7]. ■

According to Lemma 1, it can be ensured that the admissible set $\mathcal{A}$ is decodable, if all index tuples $\mathbf{i}$ contained are encoded onto different codeword tuples $\mathbf{w}$. Since this work aims at providing a source optimized design concept, specifically the cylinder $\mathcal{C}(r)$ is considered as admissible set $\mathcal{A}$. The goal is to ensure that $\mathcal{C}(r)$ is decodable. For the code design it will prove useful to abstract the cylinder by a collection of diagonal segments as described in Section II-B.

We say that an index tuple (point) $\mathbf{i} \in \mathcal{I}$ lies on a certain subdiagonal if it verifies the diagonal's equation (4) and it lies on a certain diagonal segment if it additionally verifies the segment's defining property. Furthermore, we say that a codeword tuple $\mathbf{w} \in \mathcal{W}$ lies on a certain subdiagonal if there is an index tuple $\mathbf{i} \in \mathcal{I}$ on the diagonal that is encoded onto $\mathbf{w}$. The same holds for the diagonal segment accordingly.

We can conclude that the cylinder $\mathcal{C}(r)$ is decodable if there are no duplicate codeword tuples $\mathbf{w}$ on any diagonal segment or any pair of diagonal segments contained in the cylinder.

We shall now present the mathematical tools used to evaluate this.

A. Diophantine Analysis

Let $\mathbf{w}^{(1)} \in \mathcal{W}$ be the codeword tuple which should be located on a certain subdiagonal $\gamma(\mathbf{c})$.

Given an arbitrary reference point $\mathbf{j}(\mathbf{c}) = (j_1(\mathbf{c}), j_2(\mathbf{c}), \ldots, j_N(\mathbf{c}))^T$ on the considered subdiagonal $\gamma(\mathbf{c})$, it is easy to see that any index tuple $\mathbf{i}$ lying on $\gamma(\mathbf{c})$ can be expressed relative to the reference point such that $\mathbf{i} = \mathbf{j}(\mathbf{c}) + s \cdot \mathbf{1}_N$ and, equally, that any index i_n can be expressed relative to the reference point such that $i_n = j_n(\mathbf{c}) + s_n$, $n = 1, 2, \ldots, N$.

In the remainder of this paper we shall refer to s as the *position of the index tuple* $\mathbf{i}$ relative to $\mathbf{j}(\mathbf{c})$ and, equally, we shall refer to s_n as the *position of the index* i_n relative to $j_n(\mathbf{c})$, $n = 1, 2, \ldots, N$. Furthermore, if s is the position of $\mathbf{i}$ and $m(\mathbf{i}) = \mathbf{w}$, then s is also the *position of the codeword tuple* $\mathbf{w}$ and, equally, if s_n is the position of i_n and $m_n(i_n) = w_n$, then s_n is also the *position of the codeword* w_n, $n = 1, 2, \ldots, N$.

Let $K = \operatorname{lcm}(K_1, K_2, \ldots, K_N)$ be the *lowest common multiple* [12] of $K_1, K_2, \ldots, K_N$, choose $i_n \in \mathcal{I}_n$ such that $m(i_n) = w_n^{(1)}$ and set $s_n = i_n - j_n(\mathbf{c})$, $n = 1, 2, \ldots, N$.

The following theorem provides us with the means to determine whether and where the target codeword tuple $\mathbf{w}^{(1)}$ lies on the subdiagonal $\gamma(\mathbf{c})$ with reference point $\mathbf{j}(\mathbf{c})$:

Lemma 2: (Codeword Positions on Diagonal)
(a) The codeword tuple $\mathbf{w}^{(1)}$ lies on $\gamma(\mathbf{c})$, iff the Diophantine equation

$$\Phi(\mathbf{K}, \mathbf{s}): \; a_1 K_1 + s_1 = a_2 K_2 + s_2 = \ldots = a_N K_N + s_N \quad (8)$$

has an integer solution, i.e. there exist $\mathbf{a}^{(0)} = (a_1^{(0)}, a_2^{(0)}, \ldots, a_N^{(0)})^T \in \mathbb{Z}^N$, called the *particular solution*, such that choosing $a_n = a_n^{(0)}$, $n = 1, 2, \ldots, N$, verifies the equation. The position of the index tuple $\mathbf{i}$ encoded onto $\mathbf{w}^{(1)}$ is then given by $s^{(0)} = a_n^{(0)} K_n + s_n$, $n \in \{1, 2, \ldots, N\}$.
(b) If there exists a particular solution to (8), then there is an infinite number of integral solutions. The positions of the index tuples $\mathbf{i}$ encoded onto $\mathbf{w}^{(1)}$ are then given by $s = aK + s^{(0)}$ where a runs through all integers.

Proof: For details, please see [7]. ■

This gives us the means for mathematical code analysis and forms the basis for the design procedure described ahead.

B. Design Procedure

We propose an iterative algorithm to determine if there are duplicate codeword tuples $\mathbf{w}$ within the cylinder $\mathcal{C}(r)$. Its main principle is to determine for all possible pairs of diagonals $\gamma^{(1)} = \gamma(\mathbf{c}^{(1)})$ and $\gamma^{(2)} = \gamma(\mathbf{c}^{(2)})$ at positions $\mathbf{c}^{(1)}, \mathbf{c}^{(2)} \in \mathbb{Z}^{N-1}$ with a distance $L^{(1)} = L(\mathbf{c}^{(1)}), L^{(2)} = L(\mathbf{c}^{(2)}) \leq r$ if there are duplicate codeword tuples on the corresponding diagonal segments. Let $\gamma^{(1)}$ and $\gamma^{(2)}$ be called the *start* and *destination* diagonal, respectively. The duplicity is evaluated by choosing one particular codeword tuple on the start diagonal and finding its position on the destination diagonal. Specifically, we consider the codeword tuple $\mathbf{w}^{(1)} = m(\mathbf{j}^{(1)})$ corresponding to the reference point $\mathbf{j}^{(1)} = \mathbf{j}(\mathbf{c}^{(1)})$ on the start diagonal and we employ a Diophantine equation as described in Lemma 2 to determine the position of $\mathbf{w}^{(1)}$ on the destination diagonal. Using the obtained solution for this particular codeword tuple, we can infer if there are any duplicate codeword tuples within the entire scope of the start and destination diagonal segments.

The algorithm can be summarized in the following steps:

```
1: Select a start diagonal $\gamma^{(1)}$ and a destination diagonal $\gamma^{(2)}$
   by choosing $\mathbf{c}^{(1)}, \mathbf{c}^{(2)} \in \mathbb{Z}^{N-1}$ such that $L^{(1)}, L^{(2)} \leq r$.
2: Look-up the corresponding reference points $\mathbf{j}^{(1)}$, $\mathbf{j}^{(2)}$ and
   the segment lengths $V^{(1)} = V(\mathbf{c}^{(1)})$, $V^{(2)} = V(\mathbf{c}^{(2)})$.
3: Calculate the parameters of the Diophantine equation
   $\Phi(\mathbf{K}, \mathbf{s})$ to search for the codeword tuple $\mathbf{w}^{(1)} = m(\mathbf{j}^{(1)})$
   on the destination diagonal $\gamma^{(2)}$ with reference point
   $\mathbf{j}^{(2)} = \mathbf{j}(\mathbf{c}^{(2)})$:
       $s_n \leftarrow j_n^{(1)} - j_n^{(2)}$ for $n = 1, 2, \ldots, N$.
4: Solve Diophantine equation $\Phi(\mathbf{K}, \mathbf{s})$ for $\mathbf{K} =
   (K_1, \ldots, K_N)^T$ and $\mathbf{s} = (s_1, s_2, \ldots, s_N)^T$ and
   determine (if it can be found) a particular solution
   $\mathbf{a}^{(0)} = (a_1^{(0)}, a_2^{(0)}, \ldots, a_N^{(0)})^T$. Calculate the corresponding
   position $s^{(0)}$:
       $s^{(0)} \leftarrow a_n^{(0)} K_n + s_n$ for any $n \in \{1, 2, \ldots, N\}$
5: Use $s^{(0)}$ to decide if there are duplicate codeword tuples
   within the entire scope of the start and destination diagonal
   segments of length $V^{(1)}$ and $V^{(2)}$.
6: Repeat for all possible diagonal pairs and break if there
   are duplicate codeword tuples.
```

In the following, we shall explain the single steps within the algorithm in more detail. Since the proofs are rather lengthy and technical, we omit the details and refer to [7].

Step 1 and 6: A pair of diagonals $\gamma^{(1)}$ and $\gamma^{(2)}$ is chosen by selecting $\mathbf{c}^{(1)}, \mathbf{c}^{(2)} \in \mathbb{Z}^{N-1}$ such that $L^{(1)}, L^{(2)} \leq r$. It is possible to formulate a systematic search algorithm to efficiently *construct* all diagonals pairs fullfilling above criteria. This will be discussed in [7].

Step 2: When $\mathbf{c}^{(1)}, \mathbf{c}^{(2)}$ are selected, the corresponding reference points $\mathbf{j}^{(1)}$ and $\mathbf{j}^{(2)}$ as well as the segment lengths $V^{(1)}$ and $V^{(2)}$ can be derived by a simple table look-up or be calculated online, depending on the setup.

Step 3: To initialize the Diophantine equation $\Phi(\mathbf{K}, \mathbf{s})$ the parameters $\mathbf{K}$ and $\mathbf{s}$ are required. $\mathbf{K}$ is known from the cyclic index assignments to be tested. $\mathbf{s}$ can be derived by setting $s_n = j_n^{(1)} - j_n^{(2)}$ for $n = 1, 2, \ldots, N$, see [7] for details.

Step 4: For the case of $N = 2$ the Diophantine equation $\Phi(\mathbf{K}, \mathbf{s})$ can be solved using techniques from number theory.

Let $\mathbf{K}_{\{k,l\}} = (K_k, K_l)^T$ and $\mathbf{s}_{\{k,l\}} = (s_k, s_l)^T$, $k \neq l \in \{1, 2, \ldots, N\}$, be the parameters to the linear Diophantine equation $a_k K_k + s_k = a_l K_l + s_l$ which shall be denoted as $\Phi(\mathbf{K}_{\{k,l\}}, \mathbf{s}_{\{k,l\}})$. Furthermore, let the *greatest common divisor* [12] of K_k and K_l be denoted as $\gcd(K_k, K_l)$.

Proposition 1: (Solutions for $N = 2$)

The Diophantine equation $\Phi(\mathbf{K}_{\{k,l\}}, \mathbf{s}_{\{k,l\}})$ has an integral solution, iff the greatest common divisor $\gcd(K_k, K_l)$ divides (without remainder) $s_k - s_l$. A particular solution $\mathbf{a}_{\{k,l\}}^{(0)} = (a_k^{(0)}, a_l^{(0)})^T$ is then given by

$$a_k^{(0)} = -x \frac{s_k - s_l}{\gcd(K_k, K_l)} \text{ and } a_l^{(0)} = +y \frac{s_k - s_l}{\gcd(K_k, K_l)} \quad (9)$$

where $\{x, y\} \in \mathbb{Z}$ are chosen to fullfill $xK_k + yK_l = \gcd(K_k, K_l)$.

Notice, that because of Bézout's identity the existence of the integers x and y is guaranteed which can be derived using the extended Euclidean algorithm.

For the case of $N > 2$ the following result is useful:

Let $\mathbf{K}$ and $\mathbf{s}$ be the parameters to the Diophantine equation $\Phi(\mathbf{K}, \mathbf{s})$. Let $\mathcal{M} \subseteq \{1, 2, \ldots, N\}$ and let $\mathbf{K}_\mathcal{M}$ and $\mathbf{s}_\mathcal{M}$ be the parameters to the Diophantine equation $\Phi(\mathbf{K}_\mathcal{M}, \mathbf{s}_\mathcal{M})$.

Proposition 2: (Existence of Solutions for $N > 2$)

(a) The equation $\Phi(\mathbf{K}, \mathbf{s})$ can only have a solution, if $\Phi(\mathbf{K}_\mathcal{M}, \mathbf{s}_\mathcal{M})$ has a solution.

(b) The vector $\mathbf{a} = (a_1, a_2, \ldots, a_N)^T \in \mathbb{Z}^N$ can only be a solution to $\Phi(\mathbf{K}, \mathbf{s})$, if $\mathbf{a}_\mathcal{M}$ is a solution to $\Phi(\mathbf{K}_\mathcal{M}, \mathbf{s}_\mathcal{M})$.

As a direct consequence of Proposition 2, a methodology to construct the solutions to $\Phi(\mathbf{K}, \mathbf{s})$ can be formulated based on a hierarchical strategy:

Using Lemma 2 together with Proposition 2, we observe that $\Phi(\mathbf{K}, \mathbf{s})$ can only have a solution at position $s^{(0)}$, if $\Phi(\mathbf{K}_{\{k,l\}}, \mathbf{s}_{\{k,l\}})$ has a solution at a position $s_{kl}^{(0)}$, which is equal to $s^{(0)}$. It is easy to show that all solutions to $\Phi(\mathbf{K}_{\{k,l\}}, \mathbf{s}_{\{k,l\}})$ can then be expressed in the form $s_{kl} = a_{kl} K_{kl} + s_{kl}^{(0)}$, where $K_{kl} = \operatorname{lcm}(K_k, K_l)$ and a_{kl} runs through all integers. Using this representation, we are able to replace the equation $\Phi(\mathbf{K}_{\{k,l\}}, \mathbf{s}_{\{k,l\}})$ contained in $\Phi(\mathbf{K}, \mathbf{s})$ by the single expression $a_{kl} K_{kl} + s_{kl}^{(0)}$ leading to a new Diophantine equation with a reduced number of equalities. The same principle can then be applied to the newly created equation, a process which can be repeated, until only a single expression of type $aK + s^{(0)}$ remains, fully describing the solutions to the overall equation $\Phi(\mathbf{K}, \mathbf{s})$.

Step 5: Knowing the position $s^{(0)}$, it can be fully tested if there are duplicate codeword tuples on the start and destination diagonal segments of length $V^{(1)}$ and $V^{(2)}$, respectively. This can be done by employing following proposition:

Proposition 3: (Duplicity Test for Diagonal Segments)

There are duplicate codeword tuples on the start and destination segment, iff $s^{(0)}(\mathrm{mod}(K))$ lies within the *test interval* $\mathcal{T}(V^{(1)}, V^{(2)}) = \{t \in \mathbb{Z} : (0 \leq t \leq V^{(2)}) \vee (K - V^{(1)} \leq t < K)\}$.

C. Suboptimal Code Construction for $N > 2$

Using Proposition 2, we know that the Diophantine equation $\Phi(\mathbf{K}, \mathbf{s})$ can only have a solution, if $\Phi(\mathbf{K}_{\{1,2\}}, \mathbf{s}_{\{1,2\}})$ also has a solution. This can be exploited for providing a suboptimal code design procedure, suitable for $N > 2$.

Assume that $\mathbf{K}_{\{1,2\}}$ is optimized such that there are no duplicate codeword tuples $\mathbf{w}_{\{1,2\}}$ on all diagonal segments with a minimum Euclidean distance $L(\mathbf{c}_{\{1,2\}})$ less or equal to some arbitrary value r, then, after choosing $\mathbf{K} = (K_1, K_2, K_2, \ldots, K_2)^T$, there are also no duplicate codeword tuples $\mathbf{w}$ on all diagonal segments with $L(\mathbf{c}) \leq r'$ where r' is a function of r. This property can be exploited for a low-complexity design for large N. The functional relation between r and r' shall be characterized in detail in [7], however, as shown at the end of this paper, simulation results reveal, that this suboptimal design procedure leads to codes with favorable properties.

D. Geometric Code Design

The solutions to the Diophantine equation $\Phi(\mathbf{K}, \mathbf{s})$ can be represented by a rectangular point lattice. Using the geometric interpretation of this lattice, it is possible to formalize, based on geometric considerations, dependencies between $\mathbf{K}$ and properties of the cylinder $\mathcal{C}(r)$ giving rise to an analytic code design, please refer to [7] for details.

IV. Performance Evaluation and Conclusions

To underline the effectiveness and efficiency of our low-complexity coding strategy, we present numerical performance results for the quadratic Gaussian CEO Problem [11]. Let u_0 be the output of a continuous-valued Gaussian source U_0. For $n = 1, 2, ..., N$ let u_n denote noisy observations of u_0 corrupted by additive noise samples such that $u_n = u_0 + n_n$ where the noise samples n_n are generated by Gaussian noise processes N_n statistically independent over n. The observations u_n are encoded and transmitted by independently operating encoders indexed by $n = 1, 2, ..., N$.

In the following, we consider scenarios of with $N = 2$ and 3 encoders. The source process is Gaussian distributed $\mathcal{N}(\mu_0, \sigma_0^2)$ with mean $\mu_0 = 0$ and variance $\sigma_0^2 = 1$. The noise processes are also Gaussian distributed $\mathcal{N}(\eta_n, \lambda_n^2)$ with mean

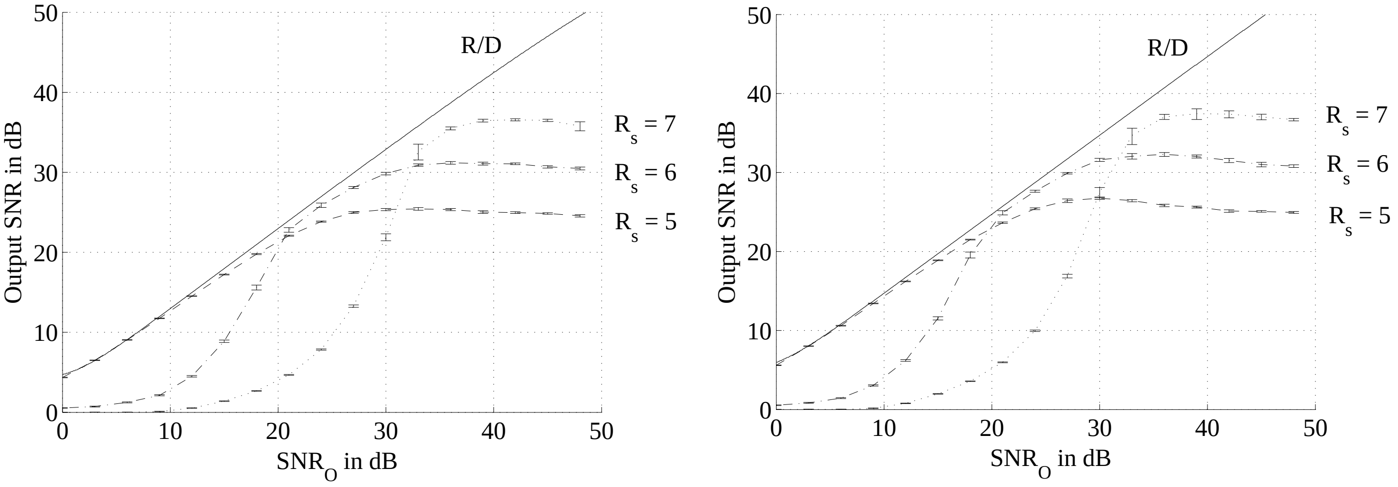


Fig. 5. Simulation results for the CEO problem with $N = 2$ encoders (left) and $N = 3$ encoders (right). The performance for a source rate $R_s = 5, 6$ and 7 [bit/codeword] is compared to the theoretical R/D bound.

$\eta_n = 0$ and variance $\lambda_n^2 = \lambda$, $n = 1, 2, \ldots, N$. We define the SNR in the observation as

$$\mathrm{SNR_O} = 10 \cdot \log_{10}\left(\frac{\sigma_0^2}{\lambda^2}\right) \text{ in dB.}$$

For each encoder n: The ranking function r_n is based on uniform quantization, optimized to minimize for minimum mean squared error in the source observations u_n. We consider the symmetric case with identical quantizer resolution, i.e. $|\mathcal{I}_n| = L$ and choose, depending on the considered setup, a resolution of $L = 32, 64$ and 128 levels for quantization, corresponding to a *source rate* of $R_s = 5, 6$ and 7 [bit/sample]. The mapping function m_n is chosen such that a constant data rate of $R = 5$ [bit/codeword] for all quantizer setups and all encoders is achieved. In the case of $N = 2$ encoders the cyclic index assignments were optimized based on the algorithm described in Section III-B and for the case of $N = 3$ encoders based on the suboptimal design presented in Section III-C.

The optimality criterion of interest is the mean squared error $\mathrm{E}\{||\mathbf{U} - \hat{\mathbf{U}}||^2\}$ and we use a decoding function d based on conditional mean estimation as presented in [4].

To evaluate the performance of our coding strategies, we measure the output SNR for U_0, as given by

$$\text{Output SNR} = 10 \cdot \log_{10}\left(\frac{u_0^2}{(u_0 - \hat{u}_0)^2}\right) \text{ in dB,}$$

versus the $\mathrm{SNR_O}$. The results are compared it with the (sum) rate-distortion function, offered by [6], which presents an upper bound found to be tight for noise processes with identical variance.

Figure 5 illustrates the performance of the system without compression by cyclic index assignments ($R_s = 5$ [bit/sample]) and the performance obtained when index assignments are employed ($R_s = 6$ and 7 [bit/sample]), in comparison to the theoretical limit given by the sum rate-distortion function (R/D) computed according to [6]. The numerical results were obtained by simulations implemented in Matlab R14. The curves show the performance of the whole system after simulating 100000 realizations of U_0 and the vertical bars show the 95% confidence interval. The numerical results show that, over a wide range of correlation values, our low-complexity, Diophantine index assignment techniques lead to significant performance gains over standard scalar quantization and come close to the theoretical optimum for the considered scenario.

References

[1] David Rebollo-Monedero, Rui Zhang and Bernd Girod. Design of Optimal Quantizers for Distributed Source Coding. In *Proceedings of the Data Compression Conference (DCC03)*, 2003.

[2] T. J. Flynn and R. M. Gray. Encoding of Correlated Observations. *IEEE Trans. Inform. Theory*, IT-33(6):773–787, 1987.

[3] J. Garcia-Frias and Y. Zhao. Compression of Correlated Binary Sources Using Turbo Codes. *IEEE Communications Letters*, pages 417–419, 2001.

[4] Gerhard Maierbacher, João Barros. Low-Complexity Coding for the CEO Problem with many Encoders. In *Twenty-sixt Syposium on Information Theory in the Benelux*, Brussels, Belgium, 2005.

[5] Jean Cardinal and Gilles Van Assche. Joint Entropy-Constrained Multiterminal Quantization. In *Proceedings of the International Symposium on Information Theory*, Lausanne, Switzerland, 2002.

[6] Jun Chen, Xin Zhang, Toby Berger, Stephen B. Wicker. An upper bound on the sum-rate distortion function and its corresponding rate allocation schemes for the CEO problem. *Special Issue of JSAC, On Fundamental Performance of Wireless Sensor Networks*, May 2004.

[7] G. Maierbacher and J. Barros. Diophantine distributed source coding. Manuscript under preparation.

[8] S. S. Pradhan and K. Ramchandran. Distributed Source Coding Using Syndromes (DISCUS): Design and Construction. In *Proc. IEEE Data Compression Conf. (DCC)*, Snowbird, UT, 1999.

[9] S. Sandeep Pradhan and Kannan Ramchandran. Generalized Coset Codes for Distributed Binning. *IEEE Trans. Inform. Theory*, 51:3457–3474, 2005.

[10] D. Slepian and J. K. Wolf. Noiseless Coding of Correlated Information Sources. *IEEE Trans. Inform. Theory*, IT-19(4):471–480, 1973.

[11] T. Berger, Z. Zhang and H. Viswanathan. The CEO problem. *IEEE Trans. Inform. Theory*, 42:887–902, 1996.

[12] James J. Tattersall. *Elementary Number Theory in Nine Chapters*. Cambridge University Press, 2nd edition, 2005.

[13] Z. Xiong, A. D. Liveris and S. Cheng. Distributed Source Coding for Sensor Networks. *IEEE Signal Processing Magazine*, 09 2004.