Hamming Distance Spectrum of DAC Codes for Equiprobable Binary Sources

Yong Fang, *Member, IEEE*, Vladimir Stankovic, *Senior Member, IEEE*, Samuel Cheng, *Member, IEEE*, and En-hui Yang, *Fellow, IEEE*

***Abstract*—Distributed Arithmetic Coding (DAC) is an effective technique for implementing Slepian-Wolf coding (SWC). It has been shown that a DAC code partitions source space into unequal-size codebooks, so that the overall performance of DAC codes depends on the cardinality and structure of these codebooks. The problem of DAC codebook cardinality has been solved by the so-called Codebook Cardinality Spectrum (CCS). This paper extends the previous work on CCS by studying the problem of DAC codebook structure. We define Hamming Distance Spectrum (HDS) to describe DAC codebook structure and propose a mathematical method to calculate the HDS of DAC codes. The theoretical analyses are verified by experimental results.**

***Index Terms*—Distributed source coding, Slepian-Wolf coding, distributed arithmetic coding, Hamming distance spectrum, codebook cardinality spectrum.**

I. Introduction

ARITHMETIC coding (AC) [1] is an effective method for data compression that works by mapping each source sequence onto a half-open interval $[l, h)$, where $0 \leq l < h < 1$. Though the principle of AC codes is rather simple, a major technical problem when putting AC codec into practice is that one has to use infinite-precision real numbers to represent l and h, which is impossible for a digital circuit. Fortunately, there is a canonical implementation in [2] that represents l and h with finite-precision integers and utilizes some scaling rules to solve the problems of *renormalization* and *underflow* that are caused by finite-precision operations. An alternative solution to the complexity and precision problems in the AC codec is to use *quasi-AC* (QAC) codes, which can be seen as a reduced-precision version of AC codes [3].

As other variable-length codes, AC codes suffer from error propagation when the bitstream is conveyed over noisy channels. This problem can be solved by reserving a forbidden interval in $[0, 1)$ for error detection [4] and running *maximum a posteriori* (MAP) decoding for error correction [5]. For QAC codes, state models can be defined and used in a straightforward manner for MAP or soft decoding [6]. Such solutions are known as *joint source-channel AC* (JSCAC). Forbidden interval reservation is not the only solution to this problem, *e.g.*, [7] achieves the same goal by inserting segment markers at fixed positions of bitstreams. Besides hard markers, the soft synchronization mechanism is also a powerful option for the JSCAC which allows controlling the trade-off between redundancy and resilience [8]. To predict and evaluate the effectiveness of the JSCAC, [9] provides an analytical tool to derive the distance spectrum of the JSCAC and proposes an algorithm to compute the free distance of the JSCAC.

Recently, AC codes also find their application to lossless *distributed source coding* (DSC), or *Slepian-Wolf coding* (SWC) [10], which has traditionally been implemented with channel codes, *e.g.*, turbo codes [11] and *low-density parity-check* (LDPC) codes [12], [13], [14]. Such solutions are known as *distributed AC* (DAC) codes. In fact, DAC codes are dual codes of JSCAC codes, so they can be realized by either interval overlapping [15], [16], [17] or bitstream puncturing [18], [19], [20]. Naturally, DAC codes can be combined with JSCAC codes to obtain the so-called *distributed JSCAC* (DJSCAC) codes, which allow the coexistence of overlapped and forbidden intervals to realize data compression and error correction simultaneously [21].

Since the emergence of DAC codes, a lot of work has been done to verify the *coding efficiency* of DAC codes [16]. An important finding is that the residual errors of DAC codes cannot be removed by increasing code rate and/or length [16]. Thus, it is better to quantitatively measure the *coding efficiency* of DAC codes in terms of *frame-error-rate* (FER) or *symbol-error-rate* (SER) at a given code rate. Moreover, it is shown that at least for short code length, DAC codes outperform LDPC-based SWC codes with acceptable *decoder complexity* [16].

However, the above results are heuristic and lack strict theoretical analyses. To obtain an illuminating insight into the *coding efficiency* and *decoder complexity* of DAC codes, the concept of *spectrum* was introduced, and the following findings were reported in [22], [23], [24]:

- A DAC code partitions source space into unequal-size codebooks whose cardinalities are proportional to the so-called *initial spectrum* [23]. According to this finding, we can draw the following conclusion: For a DAC code with initial spectrum $f_0(u)$ (see Sect. III for a formal definition of the initial spectrum), its total rate loss to SWC limit [10] will tend to a constant $\int_0^1 f_0(u) \log_2 f_0(u) du$ as code

This work was supported by the National Science Foundation of China (grant nos. 61271280 and 61377011), the Program for New Century Excellent Talents in University of China (grant no. NCET-13-0481), Provincial Foundation for Youth Nova of Science and Technology of Shaanxi, China (grant no. 2014KJXX-41), and the Fundamental Research Fund for the Central Universities of China (grant nos. 2014YQ001 and QN2013086).

Y. Fang is with the College of Information Engineering, Northwest A&F University, Yangling, Shaanxi 712100, China (email: yfang79@gmail.com). V. Stankovic is with the Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, UK (email: vladimir.stankovic@strath.ac.uk). S. Cheng is with the School of Electrical and Computer Engineering, University of Oklahoma, Tulsa, OK (email: samuel.cheng@ou.edu). E.-H. Yang is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (email: ehyang@uwaterloo.ca). The corresponding author is Y. Fang.

length goes to infinity, and hence the per-symbol rate loss will vanish as code length increases [24].
- DAC spectrum will become uniformly distributed as the decoding proceeds, which implies that 1-away (in Hamming distance) codewords in each codebook cannot be removed by increasing code length [24]. Further, a loose lower bound of decoding error probability is given as $\epsilon(2-2^R)$, where ϵ is the crossover probability between source and *side information* (SI), and R is code rate [24].
- Two techniques can be used to improve the coding efficiency of DAC codes [24]. First, the *permutation* technique can remove those closely-packed (in Hamming distance) codewords in each codebook. Second, the *weighted branching* technique can reduce the mis-pruning risk of proper paths during the decoding.

Besides the above advances, the authors of [25] also noticed the existence of 1-away (in Hamming distance) codewords in each DAC codebook and proposed the *distributed block arithmetic coding* (DBAC) to solve this problem.

In summary, *the problem of deducing DAC codebook cardinality has been solved, but we still know very little about DAC codebook structure*. The only thing we know about the latter problem is that 1-away (in Hamming distance) codewords in each DAC codebook almost always exist [24]. Obviously, to analyze the coding efficiency of DAC codes, more knowledge about DAC codebook structure is necessary. Motivated by this problem, this paper introduces the concept of *Hamming distance spectrum* (HDS), which is essentially proportional to the average number of d-away (in Hamming distance) codeword-pairs inside each DAC codebook. We denote the HDS by $\psi_{n,R}(d)$, a function *with respect to* (w.r.t.) inter-codeword Hamming distance $d \in \{0, \cdots, n\}$ that is parameterized by code length n and rate R, and propose a mathematical method to calculate $\psi_{n,R}(d)$. Equipped with the HDS, it may be possible to calculate the FER and SER of DAC codes. Notice that to distinguish from the HDS, the spectrum defined in [22], [23], [24] will be formally referred to as *codebook cardinality spectrum* (CCS).

The rest of this paper is arranged as follows. Section II describes the encoding procedure of DAC codes. Section III briefly reviews the previous work on the CCS. Section IV defines the HDS for DAC codes and gives an example to illustrate how to calculate it by exhaustive enumeration. Section V develops a mathematical method to calculate $\psi_{n,R}(1)$, which is then generalized to $\psi_{n,R}(d)$ for $d \geq 2$ in Sect. VI. Two implementation issues during calculating $\psi_{n,R}(d)$, *i.e.*, complexity and convergency, are discussed in Sects. VII and VIII, respectively. Experimental results are presented in Sect. IX to verify the correctness of the proposed method. Finally, Sect. X concludes this paper.

Source Model Following [22], [23], [24], this paper restricts the research scope to equiprobable binary sources. The reason is that the tackled issue is difficult, thus we have to begin with the simplest but non-trivial source model to simplify the analysis and make many hard problems tractable. Note that, the concepts proposed in this paper (and previous work [22], [23], [24]) cannot easily be extended to nonuniform sources, because in contrast to *uniform* sources, for *nonuniform* sources, the DAC behaves like a *source* code rather than a *channel* code, making it very difficult to build the concepts of codebook and space partitioning.

Notation This paper will adopt the notations defined in [26], which are also used in [24]. We use X to denote a random variable and $f(X)$ to denote a function of X. Correspondingly, we use $x \in \mathcal{X}$ to denote a realization of X, where $\mathcal{X}$ is the alphabet of X, and $f(x)$ to denote a function of x. We use $X^n \triangleq (X_0, \cdots, X_{n-1})$ to denote the tuple of n random variables and $x^n \triangleq (x_0, \cdots, x_{n-1})$ to denote a realization of X^n. We use 0^n to denote the tuple of n consecutive 0s, while the meaning of 1^n is similar. We define $[i:j) \triangleq \{i, \cdots, j\}$ and $(i:j) \triangleq \{(i+1), \cdots, (j-1)\}$, while the meanings of $[i:j)$ and $(i:j]$ are similar. Further, we define $q^{[i:j)} \triangleq (q^i, \cdots, q^{j-1})$ and the meanings of $q^{[i:j]}$, $q^{(i:j)}$, and $q^{(i:j]}$ are similar. For brevity, the crossover probability between source and SI is abbreviated to *source-SI crossover probability* and denoted by ϵ. Moreover, we use q to denote the *length of enlarged intervals* (the same as [22] and [23], while different from [24]). The operation of $|\cdot|$ may denote the *absolute value of a number*, the *cardinality of a set*, or the *length of an interval*, depending on the operand. The dot product of x^n and y^n is denoted by $\langle x^n, y^n \rangle$, and the Hamming distance between x^n and y^n is denoted by $d_H(x^n, y^n)$. We use $\{(l,h]+\Delta\}$ and $\{\xi(l,h]\}$ to denote the *interval shifting* and *interval scaling* operations, respectively, *i.e.*,

$$\begin{cases} \{(l,h]+\Delta\} \triangleq (l+\Delta, h+\Delta] \\ \{\xi(l,h]\} \triangleq (\xi l, \xi h] \end{cases}. \tag{1}$$

The clip function $\max(0,\cdot)$ is abbreviated to $(\cdot)^+$, *i.e.*, $(\cdot)^+ \triangleq \max(0,\cdot)$.

II. Review of DAC Encoding

Let Y^n be a tuple of n independent and uniformly-distributed (i.u.d.) binary random variables with $Y_i \sim p(y) = 0.5$, where $y \in \mathbb{B} \triangleq \{0,1\}$ and $i \in [0:n)$. Let X^n be another tuple of n i.u.d. binary random variables with $X_i|\{Y_i = y\} \sim p(x|y)$ for $x \in \mathbb{B}$. The correlation between X^n and Y^n is modeled as a virtual *binary symmetric channel* (BSC) with crossover probability $p(0|1) = p(1|0) = \epsilon$. According to the Slepian-Wolf theorem [10], if only Y^n is available at the decoder, lossless recovery of X^n will be possible at rates $R \geq H(X|Y) = H_b(\epsilon)$ *bits per symbol* (bps), where $H_b(\cdot)$ denotes the *binary entropy function* (BEF), no matter whether Y^n is available at the encoder or not.

To compress X^n, the rate-R, where $0 < R < 1$, DAC encoder iteratively maps source symbols onto partially-overlapped intervals $[0, q)$ and $[(1-q), 1)$, where $q \triangleq 2^{-R}$ [15], [16]. Let $[L_i, H_i)$ be the interval after coding X^i. It is easy to show that $[L_0, H_0) = [0, 1)$ and $(H_i - L_i) = q^i = 2^{-iR}$ [24]. Therefore, we only need to trace either L_i or H_i. It is usually more convenient to trace L_i. As shown in [24], $L_i = l(X^i)$, where

$$l(X^i) \triangleq (1-q)\langle q^{[0:i)}, X^i \rangle. \tag{2}$$

Since the length of the *final interval* after coding X^n is always q^n, it can be uniquely identified by $\lceil -\log_2 q^n \rceil = \lceil nR \rceil$ bits. To obtain the bitstream of X^n, we scale L_n to get $S \triangleq 2^{\lceil nR \rceil} L_n$. It is easy to see $S = s(X^n) \triangleq 2^{\lceil nR \rceil} l(X^n)$. The final interval $[L_n, L_n + q^n)$ is now mapped onto $[S, S + 2^{\lceil nR \rceil - nR})$, which will be referred to as *scaled final interval*. An important problem is: What is the range of S? This problem can be solved by considering the following two extreme cases: If $X^n = 0^n$, then $[L_n, H_n) = [0, q^n)$; and if $X^n = 1^n$, then $[L_n, H_n) = [(1-q^n), 1)$. Hence, $L_n \in [0, (1-q^n)]$ and further $S \in [0, (2^{\lceil nR \rceil} - 2^{\lceil nR \rceil - nR})]$. Then we calculate $\lceil S \rceil$. Because $(\lceil nR \rceil - nR) \in [0, 1)$, we have $2^{\lceil nR \rceil - nR} \in [1, 2)$ and further $\lceil 2^{\lceil nR \rceil} - 2^{\lceil nR \rceil - nR} \rceil = (2^{\lceil nR \rceil} - 1)$. Therefore, $\lceil S \rceil \in [0 : 2^{\lceil nR \rceil})$, implying that $\lceil S \rceil$ can be binarized into a string of $\lceil nR \rceil$ bits, which is just the DAC bitstream of X^n.

Length of Scaled Final Interval For simplicity, we will no longer consider the case $\lceil nR \rceil > nR$ in the following. The reason is: If $\lceil nR \rceil > nR$, we can always re-encode X^n at rate $R' = \lceil nR \rceil / n$, which will produce a bitstream with exactly the same length. Therefore, in the rest of this paper, $S = q^{-n} L_n$ and

$$s(X^n) = q^{-n} l(X^n) = (1-q)\langle q^{[0:n)-n}, X^n \rangle. \tag{3}$$

It is easy to know $S \in [0, (2^{nR} - 1)]$ and $\lceil S \rceil \in [0 : 2^{nR})$. The scaled final interval after coding X^n is always $[S, S+1)$, *i.e.*, the length of the scaled final interval is always 1.

Illustration of DAC Encoding An illustration to better understand DAC encoding is shown in Fig. 1. As shown in Fig. 1, if we take each codeword $x^n \in \mathbb{B}^n$ as a *ball*, DAC encoding is equivalent to putting 2^n *balls* into 2^{nR} *bins* according to $s(x^n)$. The rule is: If $s(x^n) = 0$, x^n is put into the 0-th bin; otherwise if $s(x^n) \in ((m-1), m]$, where $m \in [1 : 2^{nR})$, x^n is put into the m-th bin (cf. Fig. 1).

III. Review on Codebook Cardinality Spectrum

In this section, we briefly review the main results on CCS [22], [23], [24]. As shown in [24], a $(2^{nR}, n)$ binary DAC code is defined as

- an encoder $m : \mathbb{B}^n \to [0 : 2^{nR})$ that assigns index $m \in [0 : 2^{nR})$ to each source sequence $x^n \in \mathbb{B}^n$, and
- a decoder $\hat{x}^n : [0 : 2^{nR}) \to \mathbb{B}^n \cup \{e\}$ that assigns an estimate $\hat{x}^n \in \mathbb{B}^n$ or an error message e to each index $m \in [0 : 2^{nR})$.

The DAC encoding is in fact a many-to-one nonlinear mapping $\mathbb{B}^n \to [0 : 2^{nR})$, which unequally partitions source space $\mathbb{B}^n$ into 2^{nR} codebooks. Let $\mathcal{C}_m$, where $m \in [0 : 2^{nR})$, be the m-th codebook. If $x^n \in \mathcal{C}_m$, then $\lceil s(x^n) \rceil = m$ and

$$s(x^n) \in ((m-1), m] \cap [0, (2^{nR} - 1)]. \tag{4}$$

Especially, if $x^n \in \mathcal{C}_0$, $s(x^n) \equiv 0$; otherwise, $s(x^n) \in ((m-1), m]$. Since $l(x^n) = q^n s(x^n)$,

$$l(x^n) \in ((m-1)q^n, mq^n] \cap [0, (1-q^n)]. \tag{5}$$

Especially, if $x^n \in \mathcal{C}_0$, $l(x^n) \equiv 0$; otherwise, $l(x^n) \in ((m-1)q^n, mq^n]$.

An important property of DAC codebooks is *cardinality*, which is determined in [22], [23], [24] by defining the so-called *initial spectrum*.

Initial Spectrum Let $X^\infty \triangleq (X_0, X_1, \cdots)$ and $q^{[0:\infty)} \triangleq (q^0, q^1, \cdots)$. Both L_∞ and H_∞ will converge to the following continuous random variable

$$U_0 \triangleq (1-q)\langle q^{[0:\infty)}, X^\infty \rangle, \tag{6}$$

whose *probability density function* (pdf) $f_0(u)$ is called the *initial spectrum* [22], [23], [24].

According to the definition of $f_0(u)$, for $m \in [1 : 2^{nR})$, the cardinality of $\mathcal{C}_m$ is proportional to the integral of $f_0(u)$ over $((m-1)q^n, mq^n]$ in the asymptotic sense, *i.e.*, as $n \to \infty$,

$$|\mathcal{C}_m| \to 2^n \int_{(m-1)q^n}^{mq^n} f_0(u) du. \tag{7}$$

For n sufficiently large, the interval $((m-1)q^n, mq^n]$ will be so short that $f_0(u)$ almost holds constant in $((m-1)q^n, mq^n]$. Thus $|\mathcal{C}_m| \to f_0(mq^n) 2^{n(1-R)}$ as $n \to \infty$, *i.e.*, $|\mathcal{C}_m|$ is proportional to $f_0(mq^n)$ in the asymptotic sense. For this reason, we will call $f_0(u)$ the *codebook cardinality spectrum* (CCS) from now on. For equiprobable binary sources, $\Pr\{X^n = x^n\} \equiv 2^{-n}$ for all $x^n \in \mathbb{B}^n$. Thus, for all $x^n \in \mathcal{C}_m$, $\Pr\{X^n = x^n | X^n \in \mathcal{C}_m\} \equiv 1/|\mathcal{C}_m|$, and for all $m \in [0 : 2^{nR})$, $\Pr\{\lceil s(X^n) \rceil = m\} = |\mathcal{C}_m| 2^{-n}$.

The 0-th Codebook Because $\lceil s(x^n) \rceil = 0$ only if $x^n = 0^n$, $\mathcal{C}_0$ has one and only one codeword 0^n in any case, and its cardinality is always 1, *i.e.*, $|\mathcal{C}_0| \equiv 1$.

Conditional CCS The pdf of U_0 given $X_j = b \in \mathbb{B}$ is called the *conditional CCS* given $X_j = b$, and denoted by $f_{0,j}(u|b)$ [27].

Though DAC codebook *cardinality* has been well studied, very little about DAC codebook *structure* is known up to now. The only thing we know is that for a $(2^{nR}, n)$ binary DAC code, as $n \to \infty$, the proportion of twin leaf nodes in the decoding tree will tend to $(2 - 2^R)$. Thus, the decoding error probability is lower bounded by $\epsilon(2 - 2^R)$, where ϵ is the source-SI crossover probability [24].

IV. Hamming Distance Spectrum

Just as channel codes, it is rather intuitive that another very important property of DAC codes is how far (in Hamming distance) the codewords in each codebook keep away from each other. Therefore, we will below define the *Hamming distance spectrum* (HDS) to measure quantitatively the distribution of inter-codeword Hamming distances within each DAC codebook, which will be helpful for understanding DAC codebook structure.

A. Definition of Hamming Distance Spectrum

Codeword HDS The HDS w.r.t. codeword X^n is defined as

$$k_d(X^n) \triangleq \left| \left\{ \tilde{X}^n : \lceil s(X^n) \rceil = \lceil s(\tilde{X}^n) \rceil \text{ and } d_H(X^n, \tilde{X}^n) = d \right\} \right|. \tag{8}$$

Fig. 1. Explanation of DAC encoding with *ball binning*. The horizontal axis is $s(x^n)$.

In plain words, $k_d(X^n)$ is the number of codewords $\tilde{X}^n$ in codebook $\lceil s(X^n)\rceil = m$ that are d-away (in Hamming distance) from X^n. It is easy to see $d \in [0:n]$ and $0 \le k_d(X^n) \le \binom{n}{d}$. If we define $k_0(X^n) = 1$, then $\sum_{d=0}^{n} k_d(X^n) = |\mathcal{C}_m|$.

Codebook HDS The HDS of the m-th codebook is defined as

$$\phi_m(d) \triangleq E[k_d(X^n)|X^n \in \mathcal{C}_m]. \tag{9}$$

Code HDS The HDS of the $(2^{nR}, n)$ DAC code is defined as

$$\psi_{n,R}(d) \triangleq E[k_d(X^n)]. \tag{10}$$

It is easy to see that $\phi_m(d)$ is proportional to the number of d-away *codeword-pairs* within the m-th codebook and similarly, $\psi_{n,R}(d)$ is proportional to the average number of d-away *codeword-pairs* within all codebooks of the $(2^{nR}, n)$ DAC code.

Asymptotic Code HDS The *asymptotic HDS* of the rate-R DAC code is defined as

$$\lambda_R(d) \triangleq \lim_{n\to\infty} \psi_{n,R}(d). \tag{11}$$

B. Calculating HDS by Exhaustive Enumeration

In practice, $\psi_{n,R}(d)$ can be calculated by exhaustive enumeration. Let us first consider $\phi_m(d)$. For equiprobable binary sources,

$$\begin{aligned}\phi_m(d) &= \sum_{x^n\in\mathcal{C}_m} \Pr\{X^n = x^n | X^n \in \mathcal{C}_m\} k_d(x^n) \\ &= (1/|\mathcal{C}_m|) \sum_{x^n\in\mathcal{C}_m} k_d(x^n),\end{aligned} \tag{12}$$

where $k_d(x^n)$ is a realization of $k_d(X^n)$ [23]. Further, we can obtain

$$\begin{aligned}\psi_{n,R}(d) &= \sum_{m=0}^{2^{nR}-1} \Pr\{\lceil s(X^n)\rceil = m\}\phi_m(d) \\ &= 2^{-n} \sum_{m=0}^{2^{nR}-1} \sum_{x^n\in\mathcal{C}_m} k_d(x^n).\end{aligned} \tag{13}$$

Convexity of Sum-of-HDS Since $\sum_{d=0}^{n} k_d(x^n) \equiv |\mathcal{C}_m|$ for all $x^n \in \mathcal{C}_m$, we have [24]

$$\begin{aligned}\sum_{d=0}^{n} \psi_{n,R}(d) &= 2^{-n} \sum_{m=0}^{2^{nR}-1} |\mathcal{C}_m|^2 \\ &\to 2^{n(1-R)} \int_0^1 f_0^2(u)du,\end{aligned} \tag{14}$$

as $n \to \infty$. Hence,

$$\Gamma_n \triangleq \frac{\sum_{d=0}^{n} \psi_{n,R}(d)}{2^{n(1-R)}} \to \int_0^1 f_0^2(u)du \ge 1. \tag{15}$$

After expansion, we have $\Gamma_n = \prod_{i=0}^{n-1} \gamma_i$, where γ_i is the level-i expansion factor that is defined as the ratio of the number of level-$(i+1)$ nodes to that of level-i nodes in the DAC decoding tree [23]. Apparently, Γ_∞ is a nonnegative and convex function in $f_0(u)$, which takes the minumum value 1 only when $f_0(u)$ is uniform over $[0,1)$. Similarly, we have

$$\sum_{d=0}^{n} \psi_{n,R}(d) \ge 2^{n(1-R)} \tag{16}$$

and the equality holds only if $|\mathcal{C}_m| \equiv 2^{n(1-R)}$, *i.e.*, source space $\mathbb{B}^n$ is equally partitioned into 2^{nR} codebooks of cardinality $2^{n(1-R)}$.

C. Example of Hamming Distance Spectrum

To illustrate the concept of HDS, we give an example to show how to calculate $\psi_{n,R}(d)$ for $n = 4$ and $R = 0.5$. The source space $\mathbb{B}^n$ contains $2^n = 16$ codewords and is partitioned into $2^{nR} = 4$ codebooks. We list all codewords of the source space in Tab. I. For each codeword x^n, $s(x^n)$ (the lower bound of the scaled final interval) and m (the corresponding codebook index) are included in Tab. I, where different codebooks are marked with different colors for clarity. We also plot the positions of $s(x^n)$ for all codewords x^n in Fig. 2. It can be seen that $|\mathcal{C}_0| = 1$, $|\mathcal{C}_1| = 4$, $|\mathcal{C}_2| = 7$, and $|\mathcal{C}_3| = 4$. We list the HDS of each codeword in Tab. I. After a simple calculation, we obtain the HDS of each codebook and the code HDS, as shown in Tab. II. It is easy to verify $\Gamma_n = 5.125/4 > 1$.

V. Mathematical Calculation of $\psi_{n,R}(1)$

For a large n, it is difficult to calculate code HDS $\psi_{n,R}(d)$ through exhaustive enumeration in Subsect. IV-C because it needs the HDS $k_d(x^n)$ of all 2^n codewords. To get around it, we propose below a mathematical method that is able to obtain $\psi_{n,R}(d)$ directly in the absence of $k_d(x^n)$. The procedure of the proposed method is still very time-consuming for large d. Nevertheless, this is usually enough in practice because the decoding failure of DAC codes is caused mainly by closely-packed (in Hamming distance) codewords within each codebook. For clarity, we first use the simplest case $d = 1$ to illustrate the principle of the developed method in

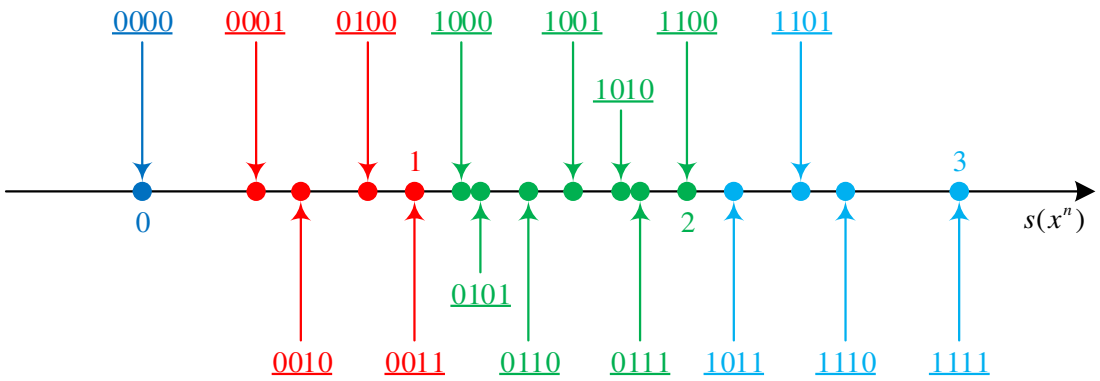


Fig. 2. Example for illustrating the mapping of x^n and $s(x^n)$, where $n = 4$ and $R = 0.5$. Each node at the horizontal axis denotes the position of $s(x^n)$ corresponding to codeword x^n. Different codebooks are marked with different colors.

TABLE I
EXAMPLE OF CODEWORD HDS

x^n	$s(x^n)$	m	$k_0(x^n)$	$k_1(x^n)$	$k_2(x^n)$	$k_3(x^n)$	$k_4(x^n)$
0000	0.0000	0	1	0	0	0	0
0001	0.4142	1	1	1	2	0	0
0010	0.5858	1	1	1	2	0	0
0011	1.0000	1	1	2	0	1	0
0100	0.8284	1	1	0	2	1	0
0101	1.2426	2	1	1	3	1	1
0110	1.4142	2	1	1	3	1	1
0111	1.8284	2	1	2	0	3	1
1000	1.1716	2	1	3	0	2	1
1001	1.5858	2	1	1	3	1	1
1010	1.7574	2	1	1	3	1	1
1011	2.1716	3	1	1	2	0	0
1100	2.0000	2	1	1	4	1	0
1101	2.4142	3	1	1	2	0	0
1110	2.5858	3	1	1	2	0	0
1111	3.0000	3	1	3	0	0	0
Sum	—	—	16	20	28	12	6

TABLE II
EXAMPLE OF CODEBOOK HDS AND CODE HDS

Term	$d=0$	$d=1$	$d=2$	$d=3$	$d=4$	Sum
$\phi_0(d)$	1	0	0	0	0	1
$\phi_1(d)$	1	4/4	6/4	2/4	0	4
$\phi_2(d)$	1	10/7	16/7	10/7	6/7	7
$\phi_3(d)$	1	6/4	6/4	0	0	4
$\psi_{n,R}(d)$	1	20/16	28/16	12/16	6/16	5.125

this section and then extend it to the general case $d \geq 2$ in the next section. The core idea of our proposed method is to expand $\psi_{n,R}(d)$ as the sum of multiple tractable terms (called *atoms* below). To achieve this goal, we define the following important concept.

XOR Pattern We refer to $Z^n = (X^n \oplus \tilde{X}^n)$ as the *XOR pattern* between X^n and $\tilde{X}^n$, where X^n and $\tilde{X}^n$ are two binary vectors.

A. Expansion of $\psi_{n,R}(1)$ as Sum-of-Atoms

Given $d_H(X^n, \tilde{X}^n) = 1$, there are $\binom{n}{1} = n$ different XOR patterns between X^n and $\tilde{X}^n$, which must take the form of $z^n(j) \triangleq (0^j, 1, 0^{n-j-1})$, where $j \in [0:n)$. Let $z_i(j)$ be the i-th element of $z^n(j)$, then $z_j(j) = 1$ and $z_i(j) = 0$ for all other $i \neq j$. We define

$$k_1^{(j)}(X^n) \triangleq \left|\left\{\tilde{X}^n : \lceil s(\tilde{X}^n)\rceil = \lceil s(X^n)\rceil \text{ and } (X^n \oplus \tilde{X}^n) = z^n(j)\right\}\right|. \tag{17}$$

In plain words, $k_1^{(j)}(X^n)$ is the number of codewords $\tilde{X}^n$ in codebook $\lceil s(X^n)\rceil = m$ that satisfy $(X^n \oplus \tilde{X}^n) = z^n(j)$. It is easy to see $k_1^{(j)}(X^n) = 0$ or 1, and $\sum_{j=0}^{n-1} k_1^{(j)}(X^n) = k_1(X^n)$, where $k_d(X^n)$ is the codeword HDS of X^n (see Subsect. IV-A). With the help of XOR patterns, we can expand $\psi_{n,R}(1)$ as $\psi_{n,R}(1) = \sum_{j=0}^{n-1} \omega_j$, where $\omega_j \triangleq E[k_1^{(j)}(X^n)]$. We refer to ω_j as *molecule*, which can be further expanded as

$$\begin{aligned} \omega_j &= \Pr\{X_j = 0\}\beta_j(0) + \Pr\{X_j = 1\}\beta_j(1) \\ &= (1/2)\left(\beta_j(0) + \beta_j(1)\right), \end{aligned} \tag{18}$$

where $\beta_j(b) \triangleq E[k_1^{(j)}(X^n)|X_j = b]$ for $b \in \mathbb{B}$. Similarly, we refer to $\beta_j(b)$ as *atom*. In this way, we expand $\psi_{n,R}(1)$ as the sum of n molecules, each of which is the average of two atoms. The problem finally boils down to calculating atoms $\beta_j(b)$ for all $j \in [0:n)$ and $b \in \mathbb{B}$.

B. Definition of Risky Interval

Before calculating $\beta_j(b)$, we need to introduce the concept of *risky interval*. From (3), it is easy to see that given $(X^n \oplus \tilde{X}^n) = z^n(j)$,

$$s(\tilde{X}^n) = \begin{cases} s(X^n) + (1-q)q^{j-n}, & \text{if } X_j = 0 \\ s(X^n) - (1-q)q^{j-n}, & \text{if } X_j = 1 \end{cases}, \tag{19}$$

which can be abbreviated to $s(\tilde{X}^n) = s(X^n) + \tau_j(b)$, where $b \in \mathbb{B}$ is the value of X_j and $\tau_j(b) \triangleq (1-q)(-1)^b q^{j-n}$. For $m \in [1 : 2^{nR})$, notice the following two points:

- if $\lceil s(X^n) \rceil = m$, then $s(X^n) \in ((m-1), m]$;
- if $\lceil s(\tilde{X}^n) \rceil = m$, then $s(\tilde{X}^n) \in ((m-1), m]$ and $s(X^n) \in \{((m-1), m] - \tau_j(b)\}$, where $\{((m-1), m] - \tau_j(b)\}$ denotes a shifted version of $((m-1), m]$ (refer to the *Notation* part of Sect. I for the definition of *interval shifting* operation).

Clearly, given $X_j = b$ and $\lceil s(X^n) \rceil = m \in [1 : 2^{nR})$, the necessary and sufficient condition for the existence of a binary vector $\tilde{X}^n$ in the m-th codebook satisfying $(\tilde{X}^n \oplus X^n) = z^n(j)$ is $s(X^n) \in \mathcal{I}_{m,j}^{(b)}$, where

$$\mathcal{I}_{m,j}^{(b)} \triangleq \{((m-1), m] - \tau_j(b)\} \cap ((m-1), m]. \tag{20}$$

Conversely, once $s(X^n)$ falls into $\mathcal{I}_{m,j}^{(b)}$, there must exist a binary vector $\tilde{X}^n$ in the m-th codebook that satisfies $(X^n \oplus \tilde{X}^n) = z^n(j)$. Let

$$\begin{cases} \delta_j^-(b) \triangleq \min\left(1, \left(|\tau_j(b)| - \tau_j(b)\right)/2\right) \\ \delta_j^+(b) \triangleq \min\left(1, \left(|\tau_j(b)| + \tau_j(b)\right)/2\right) \end{cases}, \tag{21}$$

where $|\tau_j(b)|$ is the absolute value of $\tau_j(b)$. Then (20) can be rewritten as

$$\mathcal{I}_{m,j}^{(b)} = \left((m-1) + \delta_j^-(b), m - \delta_j^+(b)\right]. \tag{22}$$

where $m \in [1 : 2^{nR})$, $j \in [0 : n)$, and $b \in \mathbb{B}$. We refer to $\mathcal{I}_{m,j}^{(b)}$ as a *risky interval*. It is easy to know $\mathcal{I}_{m,j}^{(b)} = \emptyset$ if $|\tau_j(b)| \geq 1$.

The 0-th Risky Interval Because $\mathcal{C}_0$ contains only one codeword 0^n, $\mathcal{I}_{m,j}^{(b)}$ is meaningless for $m = 0$. Thus, we will ignore $\mathcal{I}_{0,j}^{(b)}$ in the following discussion.

Length of Risky Interval Let $|\mathcal{I}_{m,j}^{(b)}|$ be the length of $\mathcal{I}_{m,j}^{(b)}$ and $(\cdot)^+ \triangleq \max(0, \cdot)$, then

$$|\mathcal{I}_{m,j}^{(b)}| = (1 - |\tau_j(b)|)^+ = (1 - (1-q)q^{j-n})^+. \tag{23}$$

Obviously, $|\mathcal{I}_{m,j}^{(b)}| \in [0, 1)$, $|\mathcal{I}_{m,j}^{(0)}| = |\mathcal{I}_{m,j}^{(1)}|$, and $|\mathcal{I}_{1,j}^{(b)}| = \cdots = |\mathcal{I}_{2^{nR}-1,j}^{(b)}|$. In addition, $|\mathcal{I}_{m,j}^{(b)}|$ is a nondecreasing function w.r.t. j, *i.e.*, $0 \leq |\mathcal{I}_{m,0}^{(b)}| \leq \cdots \leq |\mathcal{I}_{m,n-1}^{(b)}| < 1$.

Example of Risky Interval Let $n = 4$ and $R = 0.5$, then $m \in [0 : 2^{nR}) = \{0, 1, 2, 3\}$, $j \in [0 : n) = \{0, 1, 2, 3\}$, and $q = 1/\sqrt{2}$. It is easy to obtain

$$\begin{aligned}(|\tau_0(b)|, |\tau_1(b)|, |\tau_2(b)|, |\tau_3(b)|) = \\ (1.1716, 0.8284, 0.5858, 0.4142).\end{aligned} \tag{24}$$

The risky intervals $\mathcal{I}_{m,j}^{(b)}$ for all $m \in \{1, 2, 3\}$, $j \in \{0, 1, 2, 3\}$, and $b \in \mathbb{B}$ are listed in Tab. III. For clarity, the relative relationship of $\mathcal{I}_{m,j}^{(b)}$ for different j and b is illustrated by Fig. 3. It is easy to see that $\mathcal{I}_{m,j}^{(b)} \subset \mathcal{I}_{m,j'}^{(b)}$ for $j < j'$. It can be seen that because $|\tau_0(b)| > 1$, $\mathcal{I}_{m,0}^{(b)} = \emptyset$ for all $m \in \{1, 2, 3\}$. In addition, we can find that $|\mathcal{I}_{m,j}^{(b)}|$ is indeed nondecreasing w.r.t. j.

TABLE III
EXAMPLE OF RISKY INTERVALS

Term	$j=0$	$j=1$	$j=2$	$j=3$
$\mathcal{I}_{1,j}^{(0)}$	$\emptyset$	$(0, 0.1716]$	$(0, 0.4142]$	$(0, 0.5858]$
$\mathcal{I}_{2,j}^{(0)}$	$\emptyset$	$(1, 1.1716]$	$(1, 1.4142]$	$(1, 1.5858]$
$\mathcal{I}_{3,j}^{(0)}$	$\emptyset$	$(2, 2.1716]$	$(2, 2.4142]$	$(2, 2.5858]$
$\mathcal{I}_{1,j}^{(1)}$	$\emptyset$	$(0.8284, 1]$	$(0.5858, 1]$	$(0.4142, 1]$
$\mathcal{I}_{2,j}^{(1)}$	$\emptyset$	$(1.8284, 2]$	$(1.5858, 2]$	$(1.4142, 2]$
$\mathcal{I}_{3,j}^{(1)}$	$\emptyset$	$(2.8284, 3]$	$(2.5858, 3]$	$(2.4142, 3]$
$\lvert\mathcal{I}_{m,j}^{(b)}\rvert$	0	0.1716	0.4142	0.5858

C. Link between Atom and Risky Interval

According to the definitions of $\beta_j(b)$ and $\mathcal{I}_{m,j}^{(b)}$, we can easily link them as

$$\beta_j(b) = \sum_{m=0}^{2^{nR}-1} \Pr\{\lceil s(X^n) \rceil = m | X_j = b\} p(\mathcal{I}_{m,j}^{(b)} | m, b), \tag{25}$$

where

$$p(\mathcal{I}_{m,j}^{(b)} | m, b) \triangleq \Pr\{s(X^n) \in \mathcal{I}_{m,j}^{(b)} | \lceil s(X^n) \rceil = m, X_j = b\}. \tag{26}$$

It is easy to see that in the asymptotic sense, *i.e.*, as $n \to \infty$,

$$p(\mathcal{I}_{m,j}^{(b)} | m, b) \to \frac{\int_{\{q^n \mathcal{I}_{m,j}^{(b)}\}} f_{0,j}(u|b) du}{\int_{(m-1)q^n}^{mq^n} f_{0,j}(u|b) du}, \tag{27}$$

where $f_{0,j}(u|b)$ is the conditional CCS given $X_j = b$ (see Sect. III) and $\{q^n \mathcal{I}_{m,j}^{(b)}\}$ denotes a scaled version of $\mathcal{I}_{m,j}^{(b)}$ (see the *Notation* part of Sect. I for the definition of *interval scaling* operation). As n increases, $((m-1)q^n, mq^n]$ will converge to a real number. Hence, for n sufficiently large, $f_{0,j}(u|b)$ will be approximately uniform over $((m-1)q^n, mq^n]$ and

$$\begin{aligned} p(\mathcal{I}_{m,j}^{(b)} | m, b) &\to \frac{|q^n \mathcal{I}_{m,j}^{(b)}|}{|((m-1)q^n, mq^n]|} = |\mathcal{I}_{m,j}^{(b)}| \\ &= (1 - (1-q)q^{j-n})^+, \end{aligned} \tag{28}$$

where $j \in [0 : n)$. Equivalently,

$$p(\mathcal{I}_{m,n-j}^{(b)} | m, b) \to (1 - (1-q)q^{-j})^+, \tag{29}$$

where $j \in [1 : n]$. It can be seen that $p(\mathcal{I}_{m,n-j}^{(b)} | m, b)$ keeps the same for all $m \in [1 : 2^{nR})$, thus for n sufficiently large, $\beta_{n-j}(b) \to (1 - (1-q)q^{-j})^+$. It is easy to know that $\beta_{n-j}(b) \in [0, 1)$, $\beta_{n-j}(0) = \beta_{n-j}(1)$, and $\beta_{n-j}(b)$ is a nonincreasing function w.r.t. j.

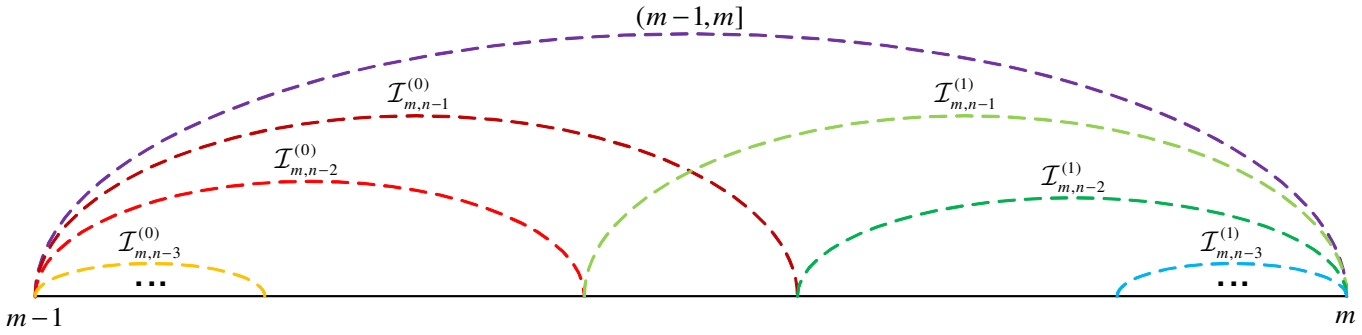


Fig. 3. Illustration of risky interval $\mathcal{I}^{(b)}_{m,j}$ for $q=1/\sqrt{2}$. The horizontal axis is $s(x^n)$.

D. Calculation of Code HDS

After knowing atoms, we can obtain molecules $\omega_{n-j} \to (1-(1-q)q^{-j})^+$, where $j \in [1:n]$. In turn, we can obtain the code HDS as below

$$\psi_{n,R}(1) \to \sum_{j=1}^{n} (1-(1-q)q^{-j})^+. \tag{30}$$

Finally, we can obtain the asymptotic code HDS as below

$$\lambda_R(1) = \sum_{j=1}^{\infty} (1-(1-q)q^{-j})^+. \tag{31}$$

VI. Mathematical Calculation of $\psi_{n,R}(d)$ for $d \geq 2$

One can easily extend the method developed in Sect. V to the general case $d \geq 2$. This section will first expand $\psi_{n,R}(d)$ as the sum of *atoms*, then define the *risky interval* to calculate atoms, and finally give the expression for $\psi_{n,R}(d)$.

A. Expansion of $\psi_{n,R}(d)$ as Sum-of-Atoms

Given $d_H(X^n,\tilde{X}^n) = d$, there are $\binom{n}{d}$ different XOR patterns between X^n and $\tilde{X}^n$. Let $\boldsymbol{j} \triangleq (j_1,\cdots,j_d)$, where $0 \leq j_1 < \cdots < j_d < n$. The XOR pattern between X^n and $\tilde{X}^n$ must take the form of

$$z^n(\boldsymbol{j}) \triangleq (0^{j_1},1,0^{j_2-j_1-1},\cdots,0^{j_d-j_{d-1}-1},1,0^{n-j_d-1}). \tag{32}$$

In other words, $z_{j_1}(\boldsymbol{j}) = \cdots = z_{j_d}(\boldsymbol{j}) = 1$ and $z_i(\boldsymbol{j}) = 0$ for other $i \notin \boldsymbol{j}$, where $z_i(\boldsymbol{j})$ denotes the i-th element of $z^n(\boldsymbol{j})$. Beginning with $j_1 \in [0:(n-d)]$, we can obtain $j_{d'} \in (j_{d'-1} : n-d+d')$ for $d' \in [2:d]$ by recursion. Let us define

$$k_d^{(\boldsymbol{j})}(X^n) \triangleq \left|\left\{\tilde{X}^n : \lceil s(\tilde{X}^n)\rceil = \lceil s(X^n)\rceil \text{ and } (X^n \oplus \tilde{X}^n) = z^n(\boldsymbol{j})\right\}\right| \tag{33}$$

and $\omega_{\boldsymbol{j}} \triangleq E[k_d^{(\boldsymbol{j})}(X^n)]$. Then we can expand $\psi_{n,R}(d)$ as

$$\psi_{n,R}(d) = \sum_{j_1=0}^{n-d} \cdots \sum_{j_d=j_{d-1}+1}^{n-1} \omega_{\boldsymbol{j}}. \tag{34}$$

Let $X_{\boldsymbol{j}} \triangleq (X_{j_1},\cdots,X_{j_d})$ and $\boldsymbol{b} \triangleq (b_1,\cdots,b_d) \in \mathbb{B}^d$, then we can further expand $\omega_{\boldsymbol{j}}$ as $\omega_{\boldsymbol{j}} = 2^{-d}\sum_{\boldsymbol{b}=0^d}^{1^d} \beta_{\boldsymbol{j}}(\boldsymbol{b})$, where $\beta_{\boldsymbol{j}}(\boldsymbol{b}) \triangleq E[k_d^{(\boldsymbol{j})}(X^n)|X_{\boldsymbol{j}} = \boldsymbol{b}]$.

B. Length of Risky Interval

According to (3), given $(X^n \oplus \tilde{X}^n) = z^n(\boldsymbol{j})$, we have $s(\tilde{X}^n) = s(X^n) + \tau_{\boldsymbol{j}}(\boldsymbol{b})$, where $\boldsymbol{b} \in \mathbb{B}^d$ is the value of $X_{\boldsymbol{j}}$ and

$$\tau_{\boldsymbol{j}}(\boldsymbol{b}) \triangleq (1-q)\sum_{d'=1}^{d} (-1)^{b_{d'}} q^{j_{d'}-n}. \tag{35}$$

Given $X_{\boldsymbol{j}} = \boldsymbol{b}$ and $\lceil s(X^n)\rceil = m \in [1:2^{nR})$, the necessary and sufficient condition for the existence of a binary vector $\tilde{X}^n$ in the m-th codebook satisfying $(\tilde{X}^n \oplus X^n) = z^n(\boldsymbol{j})$ is $s(X^n) \in \mathcal{I}^{(\boldsymbol{b})}_{m,\boldsymbol{j}}$, where

$$\mathcal{I}^{(\boldsymbol{b})}_{m,\boldsymbol{j}} \triangleq \{((m-1),m] - \tau_{\boldsymbol{j}}(\boldsymbol{b})\} \cap ((m-1),m]. \tag{36}$$

Let us define

$$\begin{cases} \delta^-_{\boldsymbol{j}}(\boldsymbol{b}) \triangleq \min\left(1, \left(|\tau_{\boldsymbol{j}}(\boldsymbol{b})| - \tau_{\boldsymbol{j}}(\boldsymbol{b})\right)/2\right) \\ \delta^+_{\boldsymbol{j}}(\boldsymbol{b}) \triangleq \min\left(1, \left(|\tau_{\boldsymbol{j}}(\boldsymbol{b})| + \tau_{\boldsymbol{j}}(\boldsymbol{b})\right)/2\right) \end{cases}. \tag{37}$$

It is easy to obtain the risky interval

$$\mathcal{I}^{(\boldsymbol{b})}_{m,\boldsymbol{j}} = \left((m-1) + \delta^-_{\boldsymbol{j}}(\boldsymbol{b}), m - \delta^+_{\boldsymbol{j}}(\boldsymbol{b})\right]. \tag{38}$$

Obviously, $\mathcal{I}^{(\boldsymbol{b})}_{m,\boldsymbol{j}} = \emptyset$ if $|\tau_{\boldsymbol{j}}(\boldsymbol{b})| \geq 1$. The length of $\mathcal{I}^{(\boldsymbol{b})}_{m,\boldsymbol{j}}$ is $|\mathcal{I}^{(\boldsymbol{b})}_{m,\boldsymbol{j}}| = (1-|\tau_{\boldsymbol{j}}(\boldsymbol{b})|)^+$.

C. Link between Atom and Risky Interval

According to the definitions of $\beta_{\boldsymbol{j}}(\boldsymbol{b})$ and $\mathcal{I}^{(\boldsymbol{b})}_{m,\boldsymbol{j}}$, we have

$$\beta_{\boldsymbol{j}}(\boldsymbol{b}) = \sum_{m=0}^{2^{nR}-1} \Pr\{\lceil s(X^n)\rceil = m | X_{\boldsymbol{j}} = \boldsymbol{b}\} p(\mathcal{I}^{(\boldsymbol{b})}_{m,\boldsymbol{j}} | m, \boldsymbol{b}), \tag{39}$$

where

$$p(\mathcal{I}^{(\boldsymbol{b})}_{m,\boldsymbol{j}} | m, \boldsymbol{b}) \triangleq \Pr\{s(X^n) \in \mathcal{I}^{(\boldsymbol{b})}_{m,\boldsymbol{j}} | \lceil s(X^n)\rceil = m, X_{\boldsymbol{j}} = \boldsymbol{b}\}. \tag{40}$$

In the asymptotic sense,

$$p(\mathcal{I}^{(\boldsymbol{b})}_{m,\boldsymbol{j}} | m, \boldsymbol{b}) \to |\mathcal{I}^{(\boldsymbol{b})}_{m,\boldsymbol{j}}| = (1-|\tau_{\boldsymbol{j}}(\boldsymbol{b})|)^+. \tag{41}$$

Let $(n-\boldsymbol{j}) \triangleq (n-j_1,\cdots,n-j_d)$ for $1 \leq j_1 < \cdots < j_d \leq n$, then (41) is equivalent to

$$p(\mathcal{I}^{(\boldsymbol{b})}_{m,n-\boldsymbol{j}} | m, \boldsymbol{b}) \to \left(1-(1-q)\,|\rho_{\boldsymbol{j}}(\boldsymbol{b})|\right)^+, \tag{42}$$

where

$$\rho_{\boldsymbol{j}}(\boldsymbol{b}) \triangleq \sum_{d'=1}^{d} (-1)^{b_{d'}} q^{-j_{d'}}. \tag{43}$$

Therefore, for n sufficiently large,

$$\beta_{n-\boldsymbol{j}}(\boldsymbol{b}) \to \left(1-(1-q)\left|\rho_{\boldsymbol{j}}(\boldsymbol{b})\right|\right)^{+}. \tag{44}$$

After a simple deduction, we obtain the tight ranges of $j_{d'}$ in (43) as follows: $j_1 \in [1:(n-d+1)]$ and $j_{d'} \in (j_{d'-1}:(n-d+d')]$ for $d' \in [2:d]$.

D. Calculation of Code HDS

After knowing atoms, we can obtain molecules as below

$$\omega_{n-\boldsymbol{j}} \to 2^{-d} \sum_{\boldsymbol{b}=0^d}^{1^d} (1-(1-q)|\rho_{\boldsymbol{j}}(\boldsymbol{b})|)^{+}, \tag{45}$$

where $1 \le j_1 < \cdots < j_d \le n$. In turn, we can obtain the code HDS as below

$$\psi_{n,R}(d) \to 2^{-d} \sum_{j_1=1}^{n-d+1} \cdots \sum_{j_d=j_{d-1}+1}^{n} \sum_{\boldsymbol{b}=0^d}^{1^d} (1-(1-q)|\rho_{\boldsymbol{j}}(\boldsymbol{b})|)^{+}. \tag{46}$$

Finally, we can obtain the asymptotic code HDS as below

$$\lambda_R(d) = 2^{-d} \sum_{j_1=1}^{\infty} \cdots \sum_{j_d=j_{d-1}+1}^{\infty} \sum_{\boldsymbol{b}=0^d}^{1^d} (1-(1-q)|\rho_{\boldsymbol{j}}(\boldsymbol{b})|)^{+}. \tag{47}$$

VII. Complexity of Calculating DAC HDS

The complexity of (46) is $O(\binom{n}{d}2^d)$. To reduce the complexity, we exploit the fact $\rho_{\boldsymbol{j}}(\boldsymbol{b}) = -\rho_{\boldsymbol{j}}(1^d \oplus \boldsymbol{b})$ to obtain

$$\psi_{n,R}(d) = 2^{1-d} \sum_{j_1=1}^{n-d+1} \cdots \sum_{j_d=j_{d-1}+1}^{n} \sum_{\boldsymbol{b}=0^d}^{(0,1^{d-1})} (1-(1-q)|\rho_{\boldsymbol{j}}(\boldsymbol{b})|)^{+}. \tag{48}$$

Therefore, in the following, the leading bit of $\boldsymbol{b}$ will always be 0 without explicit declaration. Though the complexity of (46) is now reduced to $O(\binom{n}{d}2^{d-1})$, it is still unacceptable for large n and d. Thus the proposed method is feasible only for small n and d.

The complexity of (48) can further be reduced by swapping the order of summations

$$\psi_{n,R}(d) = 2^{1-d} \sum_{\boldsymbol{b}=0^d}^{(0,1^{d-1})} \theta(\boldsymbol{b}), \tag{49}$$

where $\theta(\boldsymbol{b}) \triangleq \sum_{j_d=d}^{n} \eta(\boldsymbol{b}, j_d)$ and further

$$\eta(\boldsymbol{b}, j_d) \triangleq \sum_{j_{d-1}=d-1}^{j_d-1} \cdots \sum_{j_1=1}^{j_2-1} (1-(1-q)|\rho_{\boldsymbol{j}}(\boldsymbol{b})|)^{+}. \tag{50}$$

The complexity of $\theta(\boldsymbol{b})$ is $O(\binom{n}{d})$, still high for large n and d. However, we find that in some special cases, $\eta(\boldsymbol{b}, j_d) \equiv 0$ for all $j_d > J(\boldsymbol{b})$, where $J(\boldsymbol{b})$ is an integer totally depending on q while unrelated to n. Thus, we can obtain $\theta(\boldsymbol{b}) = \sum_{j_d=d}^{J(\boldsymbol{b})} \eta(\boldsymbol{b}, j_d)$, whose complexity is $O(\binom{J(\boldsymbol{b})}{d})$. For $n \gg J(\boldsymbol{b})$, the complexity of $\theta(\boldsymbol{b})$ will be significantly reduced.

The trick for finding $J(\boldsymbol{b})$ is to make $|\rho_{\boldsymbol{j}}(\boldsymbol{b})| \ge (1-q)^{-1}$ for all $j_d > J(\boldsymbol{b})$. However, up to now, this problem is solved only for the special case that there is no more than one 1 in $\boldsymbol{b}$, while still remains open for the general case. To facilitate the description, we divide the case that there is no more than one 1 in $\boldsymbol{b}$ into three subcases:

- $\boldsymbol{b}$ is an all-0 vector, *i.e.*, $\boldsymbol{b} = 0^d$;
- there is only one 0 before the 1 in $\boldsymbol{b}$, *i.e.*, $\boldsymbol{b} = (0,1,0^{d-2})$;
- there are two or more 0s before the 1 in $\boldsymbol{b}$, *i.e.*, $\boldsymbol{b} = (0^a,1,0^{d-a-1})$, where $a \ge 2$.

We list the expressions of $J(\boldsymbol{b})$ and $\theta(\boldsymbol{b})$ in the above three subcases in Tab. IV, while the detailed deductions are placed in the Appendix.

Below we will give some examples of $J(\boldsymbol{b})$ and $\theta(\boldsymbol{b})$ in special cases by looking up Tab. IV and discuss the existence of $J(\boldsymbol{b})$ in general cases. Afterwards, we will propose an approximation of $\psi_{n,R}(d)$ for large n and d, and finally justify the practical values of (46).

A. Examples in Special Cases

1) Examples when $\boldsymbol{b} = 0^d$: For $d = 1$, by looking up Tab. IV, we can obtain

$$\begin{cases} J(0) = \lfloor \log_q(1-q) \rfloor \\ \theta(0) = \sum_{j=1}^{J(0)} \left(1-(1-q)q^{-j}\right) \end{cases}. \tag{51}$$

For $d = 2$, by looking up Tab. IV, we can obtain

$$\begin{cases} J(0^2) = \lfloor \log_q(1-q) - \log_q(2-q^{-1}) \rfloor \\ \theta(0^2) = \sum_{j_2=2}^{J(0^2)} \sum_{j_1=1}^{j_2-1} \left(1-(1-q)(q^{-j_2}+q^{-j_1})\right) \end{cases}. \tag{52}$$

2) Examples when $\boldsymbol{b} = (0,1,0^{d-2})$: For $d = 2$, by looking up Tab. IV, we can obtain

$$\begin{cases} J(01) = \lfloor 2\log_q(1-q) \rfloor \\ \theta(01) = \sum_{j_2=2}^{J(01)} \sum_{j_1=1}^{j_2-1} \left(1-(1-q)(q^{-j_2}-q^{-j_1})\right) \end{cases}. \tag{53}$$

For $d = 3$, by looking up Tab. IV, we can obtain

$$\begin{cases} J(010) = \lfloor 2\log_q(1-q) - \log_q(2-q^{-1}) \rfloor \\ \theta(010) = \sum_{j_3=3}^{J(010)} \sum_{j_2=2}^{j_3-1} \sum_{j_1=1}^{j_2-1} \left(1-(1-q)(q^{-j_3}-q^{-j_2}+q^{-j_1})\right) \end{cases}. \tag{54}$$

3) Examples when $\boldsymbol{b} = (0^a,1,0^{d-a-1})$ and $a \ge 2$: For $d = 3$ and $a = 2$, by looking up Tab. IV, we can obtain

$$\begin{cases} J(001) = \lfloor \log_q(1-q) - \log_q(2q^{-1}-q^{-2}) \rfloor \\ \theta(001) = \sum_{j_3=3}^{J(001)} \sum_{j_2=2}^{j_3-1} \sum_{j_1=1}^{j_2-1} \left(1-(1-q)(q^{-j_3}+q^{-j_2}-q^{-j_1})\right) \end{cases}. \tag{55}$$

TABLE IV
EXAMPLE OF $J(\boldsymbol{b})$ AND $\theta(\boldsymbol{b})$

Term	Expression
$J(0^d)$	$\left\lfloor \log_q(1-q) - \log_q(2-q^{1-d}) \right\rfloor$
$J(010^{d-2})$	$\left\lfloor 2\log_q(1-q) - \log_q(2-q^{2-d}) \right\rfloor$
$J(0^a10^{d-a-1})$	$\left\lfloor -\log_q\left((2-q^{1-d})(1-q)^{-1} + 2q^{a-d}\right) \right\rfloor$
$\theta(0^d)$	$\sum_{j_d=d}^{J(0^d)} \sum_{j_{d-1}=d-1}^{j_d-1} \cdots \sum_{j_1=1}^{j_2-1} \left(1-(1-q)\sum_{d'=1}^{d} q^{-j_{d'}}\right)$
$\theta(010^{d-2})$	$\sum_{j_d=d}^{J(010^{d-2})} \sum_{j_{d-1}=d-1}^{j_d-1} \cdots \sum_{j_1=1}^{j_2-1} \left(1-(1-q)\left(\sum_{d'=1}^{d} q^{-j_{d'}} - 2q^{-j_{d-1}}\right)\right)$
$\theta(0^a10^{d-a-1})$	$\sum_{j_d=d}^{J(0^a10^{d-a-1})} \sum_{j_{d-1}=d-1}^{j_d-1} \cdots \sum_{j_1=1}^{j_2-1} \left(1-(1-q)\left(\sum_{d'=1}^{d} q^{-j_{d'}} - 2q^{-j_{a+1}}\right)\right)$

B. Discussions in General Cases

As shown above, if the leading bit of $\boldsymbol{b}$ is 0 and $\boldsymbol{b}$ contains no more than one 1, there will exist an integer $J(\boldsymbol{b})$ unrelated to n such that $\eta(\boldsymbol{b}, j_d) \equiv 0$ for all $j_d > J(\boldsymbol{b})$. The secret hidden behind it is that $\rho_{\boldsymbol{j}}(\boldsymbol{b})$ is always positive in this case. On the contrary, if there are more than one 1s in $\boldsymbol{b}$, the positiveness of $\rho_{\boldsymbol{j}}(\boldsymbol{b})$ cannot be guaranteed so that it is unknown whether there still exists an integer $J(\boldsymbol{b})$ unrelated to n such that $\eta(\boldsymbol{b}, j_d) \equiv 0$ for all $j_d > J(\boldsymbol{b})$. Through many experiments, we find that $\eta(\boldsymbol{b}, j_d)$ usually tends to zero as j_d increases. Thus, in most cases, there exists an integer $J(\boldsymbol{b})$ unrelated to n such that $\eta(\boldsymbol{b}, j_d) \equiv 0$ for all $j_d > J(\boldsymbol{b})$. However, we are not able to prove this conjecture. Note that, if $J(\boldsymbol{b})$ exists, the two codewords X^n and $\tilde{X}^n$ belonging to the same codebook differ from each other only in the last $J(\boldsymbol{b})$ symbols.

C. Approximation of $\psi_{n,R}(d)$ for large n and d

The complexity of calculating $\psi_{n,R}(d)$ by (46) is unacceptable for large n and d, so we give below a simple method to calculate the approximation of $\psi_{n,R}(d)$ for large n and d. Let X^n and $\tilde{X}^n$ be two binary sequences belonging to the same codebook. As $d_H(X^n, \tilde{X}^n) = d$ increases, X^n and $\tilde{X}^n$ will become less correlated. Thus, for a large d, X^n and $\tilde{X}^n$ can be taken as two binary sequences that are independently drawn from $\mathbb{B}^n$. This means: For d sufficiently large, $\psi_{n,R}(d)$ can be well approximated by the scaled combination formula

$$\psi_{n,R}(d) \approx \binom{n}{d} \left(\int_0^1 f_0^2(u)du\right) 2^{n(1-R)}, \tag{56}$$

where $\left(\int_0^1 f_0^2(u)du\right) 2^{n(1-R)}$ is in fact the average codebook cardinality [24].

D. Practical Values of the Proposed Method

To obtain a complete HDS by (46), one must try all possible source sequences $x^n \in \mathbb{B}^n$ and all possible XOR patterns $z^n \in \mathbb{B}^n$. Actually, by the Monte-Carlo method, one may obtain an approximate HDS much faster. Naturally, one may ask: Does the proposed method make sense in practice? Our answer is YES. There are two reasons for this answer.

First, the decoding failures of DAC codes come mainly from closely-packed (in Hamming distance) codewords in each codebook (cf. Fig. 7(a) in Subsect. IX-D), so it is unnecessary to calculate the exact value of $\psi_{n,R}(d)$ for large d by (46). Though the complexity of computing $\psi_{n,R}(d)$ by (46) is rather high for large d, it is very low for small d (much faster than the Monte-Carlo method). Hence, the proposed method is useful in practice.

Second, the proposed method may be used to compute the FER of DAC codes. Let y^n be the SI available only at the decoder and $\hat{x}^n$ be the recovered version of x^n. If we assume that x^{n-1} is known at the decoder, then $\hat{x}^{n-1} = x^{n-1}$. Let $\Pr\{e\}$ be the FER and $\Pr\{e|x^{n-1}\}$ be the conditional FER given x^{n-1} known at the decoder, then $\Pr\{e|x^{n-1}\} = \Pr\{\hat{x}_{n-1} \neq x_{n-1}\}$ and $\Pr\{e|x^{n-1}\} < \Pr\{e\}$. Given $x^n \in \mathcal{C}_m$, $c^n = x^n \oplus (0^{n-1}, 1)$ may or may not belong to $\mathcal{C}_m$. If $c^n \notin \mathcal{C}_m$, the decoding will always be correct, regardless of y_{n-1}; otherwise, *i.e.*, if $c^n \in \mathcal{C}_m$, the correctness of the decoding purely depends on y_{n-1}. Therefore,

$$\Pr\{e|x^{n-1}\} = \Pr\{c^n \in \mathcal{C}_m\} \cdot \Pr\{y_{n-1} \neq x_{n-1}\}. \tag{57}$$

It is easy to obtain

$$\Pr\{c^n \in \mathcal{C}_m\} = \Pr\{s(x^n) \in \mathcal{I}_{m,n-1}^{(b)}\}. \tag{58}$$

Let $\Pr\{X \neq Y\} = \epsilon$. By (29), we can obtain $\Pr\{e|x^{n-1}\} \to (2-2^R)\epsilon$ as $n \to \infty$, which is just the lower bound given in [24]. The above method can be extended to more complex cases, *i.e.*, all but the last $n' > 1$ symbols of each sequence are known at the decoder. When $n' = n$, $\Pr\{e|x^{n-n'}\} = \Pr\{e\}$. Actually, the residual errors of DAC codes happen mainly at sequence tails (cf. Fig. 7(b) in Subsect. IX-D), so $\Pr\{e|x^{n-n'}\}$ may be very close to $\Pr\{e\}$ for $n' \ll n$. Therefore, $\Pr\{e|x^{n-n'}\}$, where $n' \ll n$, may be taken as an approximation of $\Pr\{e\}$ and the complexity of computing the FER of DAC codes is significantly reduced.

VIII. CONVERGENCY OF DAC HDS

Another interesting question regarding the DAC HDS is: Will $\psi_{n,R}(d)$ finally converge to a finite value for any finite d as n goes to infinity, *i.e.*, $\lambda_R(d) < \infty$ for $d < \infty$? The answer

in general is NO. However, we will show below that for $d = 1$ and 2, the answer is YES.

In the case of $d = 1$, according to the analysis in Subsect. VII-A, we can obtain

$$\lambda_R(1) = \sum_{j=1}^{J(0)} \left(1 - (1-q)q^{-j}\right), \tag{59}$$

where $J(0)$ is given by (51). Apparently, $\lambda_R(1) < \infty$, *i.e.*, $\lambda_R(d)$ converges for $d = 1$.

In the case of $d = 2$, according to the analysis in Subsect. VII-A, we can obtain

$$\lambda_R(2) = \frac{1}{2}\left(\sum_{j_2=2}^{J(00)}\sum_{j_1=1}^{j_2-1}\left(1-(1-q)(q^{-j_2}+q^{-j_1})\right)+ \sum_{j_2=2}^{J(01)}\sum_{j_1=1}^{j_2-1}\left(1-(1-q)(q^{-j_2}-q^{-j_1})\right)\right), \tag{60}$$

where $J(00)$ and $J(01)$ are given by (52) and (53), respectively. Apparently, $\lambda_R(2) < \infty$, *i.e.*, $\lambda_R(d)$ converges for $d = 2$.

The convergency of $\psi_{n,R}(d)$ for $d \geq 3$ is unknown. However, we have found that $\psi_{n,R}(d)$ may not converge in some cases. For example, if $d = 3$ and $q = (\sqrt{5}-1)/2$, it is easy to verify $(q^{-j_3} - q^{-(j_3-1)} - q^{-(j_3-2)}) \equiv 0$ and thus

$$(1-(1-q)|q^{-j_3} - q^{-(j_3-1)} - q^{-(j_3-2)}|)^+ \equiv 1. \tag{61}$$

Therefore,

$$\begin{aligned}\theta(011) &= \sum_{j_3=3}^{n}\sum_{j_2=2}^{j_3-1}\sum_{j_1=1}^{j_2-1}(1-(1-q)|q^{-j_3}-q^{-j_2}-q^{-j_1}|)^+ \\ &> \sum_{j_3=3}^{n}(1-(1-q)|q^{-j_3}-q^{-(j_3-1)}-q^{-(j_3-2)}|)^+ \\ &= n-2. \end{aligned}\tag{62}$$

As n goes to infinity, $\theta(011)$ will tend to infinity, *i.e.*, $\psi_{n,R}(d)$ does not converge for $d = 3$ if $q = (\sqrt{5}-1)/2$.

IX. Experimental Results

We implemented (46) in MATLAB to calculate the theoretical values of $\psi_{n,R}(d)$ and the DAC codec in C language to obtain the empirical values of $\psi_{n,R}(d)$. Since DAC HDS does not depend on SI, we ignored SI in implementation for simplicity. We first generated a length-n equiprobable binary sequence as the source and compressed it by the DAC encoder. Then, the DAC decoder parsed the bitstream through a depth-first full search that was implemented by a recursive function. For each $d \in [0:n]$, the decoder counted the number of paths that were d-away (in Hamming distance) from the source. For fairness, 10^3 trials were run and the average number of paths that were d-away from the source was output as the empirical value of $\psi_{n,R}(d)$. The precision of the used DAC codec was 32-bit. Four experiments were conducted to study the properties of DAC codes from different aspects.

TABLE V
Comparison of Theoretical and Empirical Values of Γ_n

R	2/6	3/6	4/6	5/6
Theoretical Γ_n	1.6094	1.3046	1.1394	1.1677
Empirical Γ_n	1.6121	1.3071	1.1384	1.1675
$\int_0^1 f^2(u)du$	1.6147	1.3047	1.1340	1.1541

A. Correctness Verification of (46)

The aim of the first experiment is to verify the correctness of (46), which was achieved by comparing the theoretical values of $\psi_{n,R}(d)$ with its empirical values. Some results for code length $n = 12$ are presented in Fig. 4. Notice that since $\psi_{n,R}(0) \equiv 1$, it is not plotted in Fig. 4. We tested four different code rates: $R = 2/6$, $3/6$, $4/6$, and $5/6$, but only the results for $R = 2/6$ and $5/6$ are included in Fig. 4 for conciseness. It can be seen that the theoretical values of $\psi_{n,R}(d)$ coincide with its empirical values perfectly, which confirms the correctness of (46). Similar results are obtained for different values of n and R.

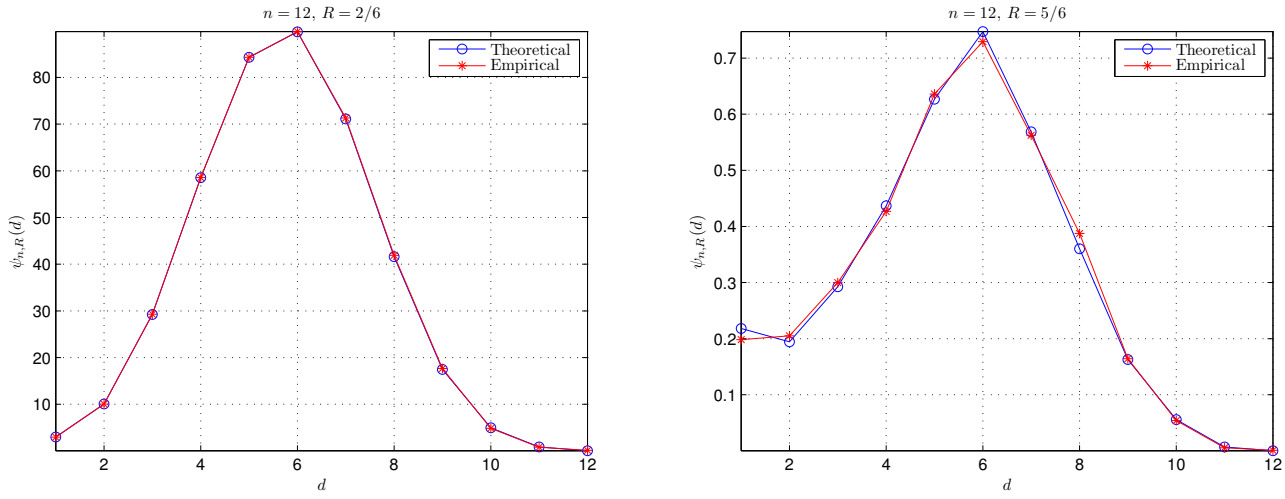
In addition, we calculated the theoretical values of Γ_n using (15) and its empirical values by experiments. The results are listed in Tab. V, where the numerical values of $\int_0^1 f^2(u)du$, which were obtained through the numerical algorithm in [23], are also included. It can be seen that the theoretical values of Γ_n, the empirical values of Γ_n, and the numerical values of $\int_0^1 f^2(u)du$ are very close to each other. These findings also confirm the correctness of (46).

B. Convergency of DAC HDS

The aim of the second experiment is to study the convergency of DAC HDS, which was achieved by trying different code lengths n ranged from 12 to 28. Both theoretical and empirical values of $\psi_{n,R}(d)$ are plotted in Fig. 5. Considering the computational complexity, only the results of $d = 1$, 2, and 3 are included in Fig. 5. To show how code rate R impacts the convergency of DAC HDS, two special code rates $R = 0.5$ and $\log_2[(\sqrt{5}-1)/2]$ were tried. From Fig. 5, it can be seen that the theoretical values of $\psi_{n,R}(d)$ coincide with its empirical values perfectly. For $d = 1$ and 2, $\psi_{n,R}(d)$ remains constant as code length n increases, *i.e.*, $\psi_{n,R}(d)$ converges as n goes to infinity. For $d = 3$, $\psi_{n,R}(d)$ remains constant when code rate $R = 0.5$ while grows continuously when $R = \log_2[(\sqrt{5}-1)/2]$ as n increases. Therefore, the convergency of $\psi_{n,R}(d)$ for $d \geq 3$ depends on code rate: At some code rates, $\psi_{n,R}(d)$ may tend to infinity as code length increases, *i.e.*, does not converge. This property of DAC codes is very different from that of random codes because according to the *law of large numbers* (LLN), for any $d < \infty$, $\psi_{n,R}(d)$ of random codes will tend to 0 as code length goes to infinity.

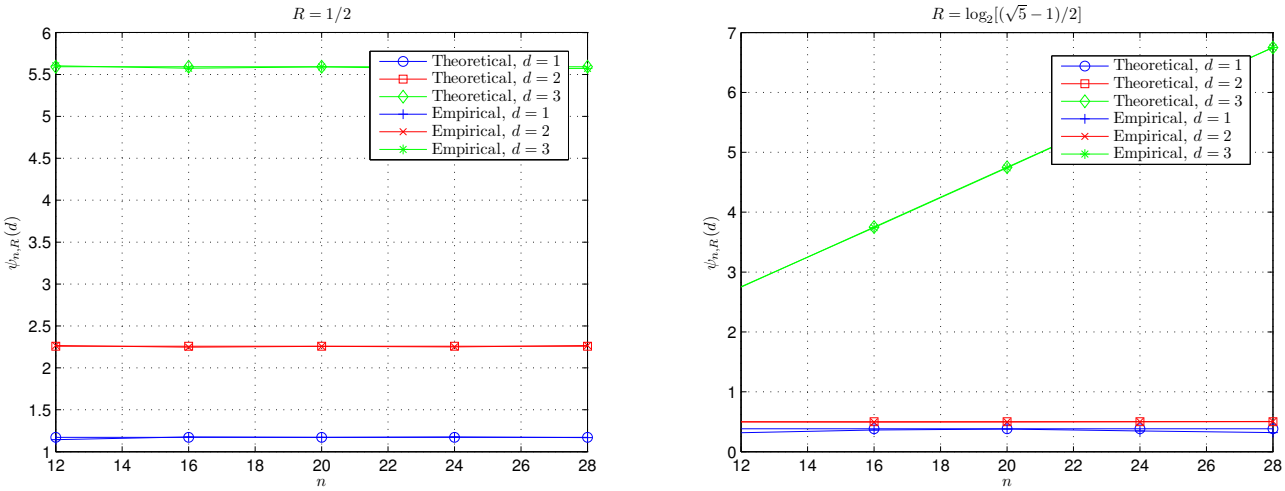
C. Comparison of DAC Codes with Other Codes

The aim of the third experiment is to compare DAC codes with random codes and some practical channel codes. For code length $n = 24$, the empirical HDS of DAC codes and the theoretical HDS of random codes are given in Fig. 6. For random codes, the inter-codeword Hamming distances within



(a) (b)

Fig. 4. Correctness verification of (46) for $n = 12$. (a) $R = 2/6$. (b) $R = 5/6$.



(a) (b)

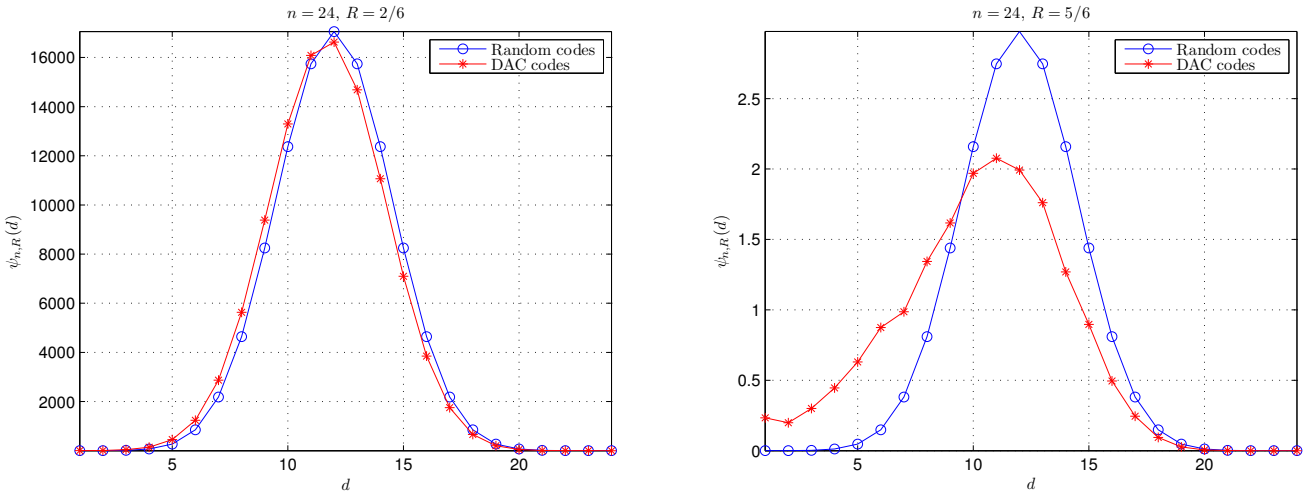
Fig. 5. Convergency of DAC HDS. (a) $R = 0.5$. (b) $R = \log_2[(\sqrt{5}-1)/2]$.

each codebook obey the binomial distribution, so $\psi_{n,R}(d)$ can be calculated by (56). It can be seen that at low rates, the HDS of DAC codes is similar to that of random codes, while at high rates, the HDS of DAC codes is different from that of random codes. Similar results are also obtained for different values of n and R.

For turbo codes, the Fano algorithm was modified to compute the HDS in [28], where some examples of selected turbo codes with short interleaving were given. For a rate-0.5 turbo code based on the $(7, 5)$ *recursive systematic convolutional* (RSC) code with 8×8 nonuniform interleaving, the HDS is $w(8) = 0.34$, $w(9) = 1.5$, $w(10) = 0.63$, $w(11) = 0.12$, and $w(12) = 1.71$. The *minimum Hamming distance* (MHD) is 8.

For LDPC codes, the *nearest nonzero codeword search* (NNCS) algorithm was proposed to find the MHD and multiplicity in [29], where the results for some well-known LDPC codes were reported. For the $(3, 6)$-regular $(504, 252)$ and $(1008, 504)$ MacKay codes [30], the MHDs are 20 and 34, and the multiplicities are 2 and 1, respectively. For the p-11 Margulis code [31], the MHD is 40 and the multiplicity is 66. For the $(13, 5)$ and $(17, 5)$ Ramanujan-Margulis codes [32], the MHDs are 14 and 24, and the multiplicities are 2184 and 204, respectively.

From the above results, we can find that compared to random codes, turbo codes, and LDPC codes, the main drawback of DAC codes is that the MHD is almost always 1, regardless of code length n and rate R. This is because for $d < \infty$, $\psi_{n,R}(d)$ of DAC codes does not converge to 0 as code length n goes to infinity (refer to Sect. VIII for the examples of $d = 1$ and 2). On the contrary, the MHDs of turbo codes and LDPC codes are far larger than 1. Even for random codes, as code length n goes to infinity, the MHD will gradually tend



(a) (b)

Fig. 6. Comparison of the HDS of DAC codes with that of random codes for $n = 24$. (a) $R = 2/6$. (b) $R = 5/6$.

to infinity because $\psi_{n,R}(d)$ will tend to zero for any $d < \infty$ according to the LLN. In addition, it can be seen that $\psi_{n,R}(d)$ of DAC codes is greater than that of random codes for small d while smaller than that of random codes for large d, implying that the HDS of DAC codes is inferior to that of random codes (especially at high rates). Based on these results, it can be concluded that the *pure* DAC codes (mapping all symbols of each sequence onto overlapped intervals) should be worse than random codes, turbo codes, and LDPC codes.

D. Properties of Decoding Errors

Let us first study the property of frame errors. In Fig. 7(a), we plot the occurrence rate of d-errors frames, where $n = 64$ and $R = 0.5$. For clarity, the occurrence rate of error-free frames is not included in Fig. 7(a). It can be seen that most of erroneous frames include only very few erroneous symbols, implying that DAC decoding errors are caused mainly by closely-packed codewords within the same codebook.

Next we study the property of symbol errors. It is pointed out in Subsect. VII-B that the two codewords belonging to the same codebook differ mainly in the last few symbols, so symbol errors should happen mainly at the tail of each frame. To verify this point, we investigate how the error probability of each symbol is impacted by its position in a frame. For $n = 64$ and $R = 0.5$, the individual SER of each symbol versus its index i is plotted in Fig. 7(b). It can be seen that symbol errors are not uniformly distributed over $i \in [0 : n)$ and most of symbol errors do happen at the last few symbols of each frame, as reported in [15], [16].

The above phenomena inspire some improvements of DAC codes, *e.g.*, permutating codewords in each codebook [24], mapping the last few symbols of each sequence onto non-overlapped intervals [15], [16], [27], *etc.* After these improvements, DAC codes may outperform LDPC codes and turbo codes, especially for short sequences [15], [16], [24], [27]. These improvements are all based on the principle of refining the HDS of DAC codes by sparsifying the last few levels of DAC decoding trees. In such a way, closely-packed codewords in each codebook can be removed, *i.e.*, $\psi_{n,R}(d) \to 0$ for small d.

X. Conclusion

The analysis of DAC codes is an interesting and challenging task. By extending our previous work, this paper makes another step forward. We define *Hamming distance spectrum* to describe how DAC codebooks are constructed and propose a method to calculate DAC HDS by the mathematical means. The correctness of the proposed method is verified by experiments. We also show how DAC codes can be improved from the viewpoint of HDS.

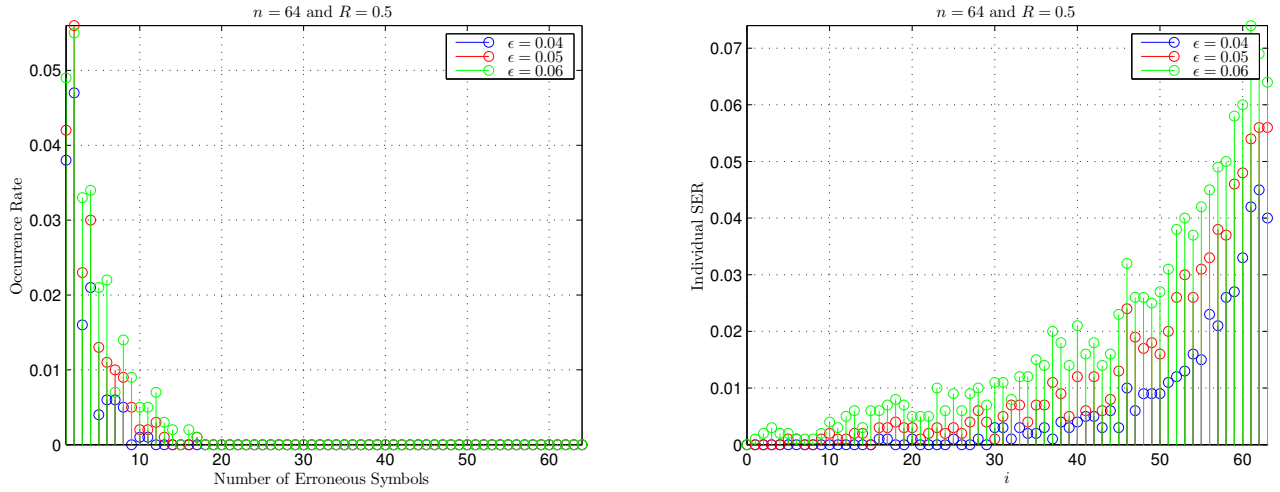
Despite the above advances, many important questions regarding DAC HDS, *e.g.*, complexity, convergency, *etc.*, still remain open. First, under which conditions will the HDS converge as code length increases? Second, if the HDS does converge as code length increases, can we find some way to calculate the HDS with linear or near-linear complexity? Third, if the HDS does not converge, how fast will it grow as code length increases? These difficult problems will be tackled in the future. Finally, another big challenge is the generalization of DAC HDS to nonuniform binary sources and even non-binary sources.

Appendix

A. Deduction of $J(\boldsymbol{b})$

1) Special Case of $\boldsymbol{b} = 0^d$: In this case, since $j_{d'} \geq d'$ for all $d' \in [1 : d]$, we have

$$\rho_{\boldsymbol{j}}(\boldsymbol{b}) = \sum_{d'=1}^{d} q^{-j_{d'}} \geq q^{-j_d} + \left(\sum_{d'=1}^{d-1} q^{-d'} \right) > 0. \qquad (63)$$



(a) (b)

Fig. 7. Properties of DAC decoding errors, where $n = 64$ and $R = 0.5$. (a) Occurrence rate of d-errors frames, where d means the number of erroneous symbols in a recovered frame. (b) Individual SER versus symbol position.

A necessary condition for $|\rho_{\boldsymbol{j}}(\boldsymbol{b})| < (1-q)^{-1}$ is

$$\begin{aligned} q^{-j_d} &< (1-q)^{-1} - \left(\sum_{d'=1}^{d-1} q^{-d'}\right) \\ &= (2-q^{1-d})(1-q)^{-1}, \end{aligned} \tag{64}$$

which is followed by $j_d < \log_q(1-q) - \log_q(2-q^{1-d})$.

2) Special Case of $\boldsymbol{b} = (0,1,0^{d-2})$: In this case, since $j_{d'} \geq d'$ for all $d' \in [1:d]$, we have

$$\begin{aligned} \rho_{\boldsymbol{j}}(\boldsymbol{b}) &= \left(\sum_{d'=1}^{d} q^{-j_{d'}}\right) - 2q^{-j_{d-1}} \\ &\geq q^{-j_d} - q^{-(j_d-1)} + \left(\sum_{d'=1}^{d-2} q^{-d'}\right) > 0. \end{aligned} \tag{65}$$

A necessary condition for $|\rho_{\boldsymbol{j}}(\boldsymbol{b})| < (1-q)^{-1}$ is

$$\begin{aligned} q^{-j_d}(1-q) &< (1-q)^{-1} - \left(\sum_{d'=1}^{d-2} q^{-d'}\right) \\ &= (2-q^{2-d})(1-q)^{-1}, \end{aligned} \tag{66}$$

which is followed by $j_d < 2\log_q(1-q) - \log_q(2-q^{2-d})$.

3) Special Case of $\boldsymbol{b} = (0^a,1,0^{d-a-1})$ for $a \geq 2$ and $d \geq 3$: In this case, since $j_{d'} \geq d'$ for all $d' \in [1:d]$, we have

$$\begin{aligned} \rho_{\boldsymbol{j}}(\boldsymbol{b}) &= \left(\sum_{d'=1}^{d} q^{-j_{d'}}\right) - 2q^{-j_{d-a}} \\ &\geq q^{-j_d} + \left(\sum_{d'=1}^{d-1} q^{-d'}\right) - 2q^{a-d} > 0. \end{aligned} \tag{67}$$

A necessary condition for $|\rho_{\boldsymbol{j}}(\boldsymbol{b})| < (1-q)^{-1}$ is

$$\begin{aligned} q^{-j_d} &< (1-q)^{-1} - \left(\sum_{d'=1}^{d-1} q^{-d'}\right) + 2q^{a-d} \\ &= (2-q^{1-d})(1-q)^{-1} + 2q^{a-d}, \end{aligned} \tag{68}$$

which is followed by $j_d < -\log_q\left((2-q^{1-d})(1-q)^{-1} + 2q^{a-d}\right)$.

REFERENCES

[1] J. Rissanen, "Generalized Kraft inequality and arithmetic coding," *IBM J. Research & Development*, vol. 20, no. 3, pp. 198–203, May 1976.

[2] I. Witten, R. Neal, and J. Cleary, "Arithmetic coding for data compression," *Commun. of the ACM*, vol. 30, no. 6, pp. 520–540, Jun. 1987.

[3] P. Howard and J. Vitter, "Practical implementations of arithmetic coding," in: *Image and Text Compression*, J. A. Storer, ed., pp. 85–112, Kluwer Academic Publishers, Boston, Mass, USA, 1992.

[4] C. Boyd, J. Cleary, S. Irvine, I. Rinsma-Melchert, and I. Witten, "Integrating error detection into arithmetic coding," *IEEE Trans. Commun.*, vol. 45, no. 1, pp. 1–3, Jan. 1997.

[5] B. Pettijohn, M. Hoffman, and K. Sayood, "Joint source/channel coding using arithmetic codes", *IEEE Trans. Commun.*, vol. 49, no. 5, pp. 826–836, May 2001.

[6] T. Guionnet and C. Guillemot, "Soft and joint source-channel decoding of quasi-arithmetic codes," *EURASIP J. Applied Signal Process.*, no. 3, pp. 393–411, Mar. 2004.

[7] I. Sodagar, B. Chai, and J. Wus, "A new error resilience technique for image compression using arithmetic coding," in: *Proc. IEEE ICASSP*, pp. 2127–2130, Istanbul, Turkey, Jun. 2000.

[8] T. Guionnet and C. Guillemot, "Soft decoding and synchronization of arithmetic codes: Application to image transmission over noisy channels," *IEEE Trans. Image Process.*, vol. 12, no. 12, pp. 1599–1609, Dec. 2003.

[9] S. Ben-Jamaa, C. Weidmann, and M. Kieffer, "Analytical tools for optimizing the error correction performance of arithmetic codes," *IEEE Trans. Commun.*, vol. 56, no. 9, pp. 1458–1468, Sep. 2008.

[10] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inform. Theory*, vol. 19, no. 4, pp. 471–480, Jul. 1973.

[11] J. Garcia-Frias and Y. Zhao, "Compression of correlated binary sources using turbo codes," *IEEE Commun. Lett.*, vol. 5, no. 10, pp. 417–419, Oct. 2001.

[12] A. Liveris, Z. Xiong, and C. Georghiades, "Compression of binary sources with side information at the decoder using LDPC codes," *IEEE Commun. Lett.*, vol. 6, no. 10, pp. 440–442, Oct. 2002.

[13] Y. Fang, "LDPC-based lossless compression of nonstationary binary sources using sliding-window belief propagation," *IEEE Trans. Commun.*, vol. 60, no. 11, pp. 3161–3166, Nov. 2012.

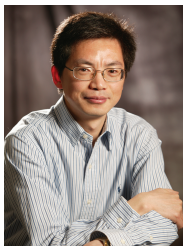
[14] Y. Fang, "Asymmetric Slepian-Wolf coding of nonstationarily-correlated M-ary sources with sliding-window belief propagation," *IEEE Trans. Commun.*, vol. 61, no. 12, pp. 5114–5124, Dec. 2013.

[15] M. Grangetto, E. Magli, and G. Olmo, "Distributed arithmetic coding," *IEEE Commun. Lett.*, vol. 11, no. 11, pp. 883–885, Nov. 2007.
[16] M. Grangetto, E. Magli, and G. Olmo, "Distributed arithmetic coding for the Slepian-Wolf problem," *IEEE Trans. Signal Process.*, vol. 57, no. 6, pp. 2245–2257, Jun. 2009.
[17] X. Artigas, S. Malinowski, C. Guillemot, and L. Torres, "Overlapped quasi-arithmetic codes for distributed video coding," in: *Proc. IEEE ICIP*, 2007, vol. II, pp. 9–12.
[18] S. Malinowski, X. Artigas, C. Guillemot, and L. Torres, "Distributed coding using punctured quasi-arithmetic codes for memory and memoryless sources," *IEEE Trans. Signal Process.*, vol. 57, no. 10, pp. 4154–4158, Oct. 2009.
[19] X. Chen and D. Taubman, "Distributed source coding based on punctured conditional arithmetic codes," in: *Proc. IEEE ICIP*, pp. 3713–3716, Sep. 2010.
[20] X. Chen and D. Taubman, "Coupled distributed arithmetic coding," in: *Proc. IEEE ICIP*, pp. 341–344, Sep. 2011.
[21] M. Grangetto, E. Magli, and G. Olmo, "Distributed joint source-channel arithmetic coding," in: *Proc. IEEE ICIP*, 2010, pp. 3717–3720.
[22] Y. Fang, "Distribution of distributed arithmetic codewords for equiprobable binary sources," *IEEE Signal Process. Lett.*, vol. 16, no. 12, pp. 1079–1082, Dec. 2009.
[23] Y. Fang, "DAC spectrum of binary sources with equally-likely symbols," *IEEE Trans. Commun.*, vol. 61, no. 4, pp. 1584–1594, Apr. 2013.
[24] Y. Fang and L. Chen, "Improved binary DAC codec with spectrum for equiprobable sources," *IEEE Trans. Commun.*, vol. 62, no. 1, pp. 256–268, Jan. 2014.
[25] J. Zhou, K. Wong, and J. Chen, "Distributed block arithmetic coding for equiprobable sources," *IEEE Sensors Journal*, vol. 13, no. 7, pp. 2750–2756, Jul. 2013.
[26] A. Gamal and Y. Kim, "Network information theory," Cambridge University Press, 2012.
[27] Y. Fang, V. Stankovic, S. Cheng, and E.-H. Yang, "Depth-first decoding of distributed arithmetic codes for uniform binary sources," submitted to *IEEE Trans. Commun.*
[28] R. Podemski, W. Holubowicz, C. Berrou, and G Battail, "Hamming distance spectra of turbo-codes," *Annals of Telecommunications*, vol. 50, no. 9–10, pp. 790–797, Sep.-Oct., 1995.
[29] X. Hu, M. Fossorier, and E. Eleftheriou, "On the computation of the minimum distance of low-density parity-check codes," in: *Proc. IEEE Int'l Conf. Commun.*, vol. 2, pp. 767–771, Jun. 2004.
[30] D. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 45, no. 3, pp. 399–431, Mar. 1999.
[31] G. Margulis, "Explicit constructions of graphs without short cycles and low density codes," *Combinatorica*, vol. 2, no. 1, pp. 71–78, Jan. 1982.
[32] J. Rosenthal and P. Vontobel, "Construction of LDPC codes based on Ramanujan graphs and ideas from Margulis," in: *Proc. 38th Annual Allerton Conf. Commun., Computing, and Control*, Monticello, IL, Oct. 2000.

Yong Fang received his BEng, MEng, and PhD from Xidian University, Xi'an China, in 2000, 2003, and 2005, respectively, all in signal processing. Then, he was a post-doctoral fellow for one year with Northwestern Polytechnical University, Xi'an China. From 2007 to 2008, he joined Hanyang University, Seoul Korea, as a research professor. He is currently a full professor with Northwest A&F University, Shaanxi Yangling, China. He had long experiences in hardware system development, *e.g.*, FPGA-based (Xilinx Vertex series) video codec design, DSP-based (TI C64 series) video surveilliance system, *etc.*. His research interests include distributed source coding, joint source-channel coding, network information theory, and image/video coding, processing, and transmission.

Vladimir Stankovic (M03-SM10) received the Dr.-Ing. (Ph.D.) degree from the University of Leipzig, Leipzig, Germany in 2003. From 2003 to 2006, he was with Texas A&M University, College Station, first as Research Associate and then as a Research Assistant Professor. From 2006 to 2007 he was with Lancaster University. Since 2007, he has been with the Dept. Electronic and Electrical Engineering at University of Strathclyde, Glasgow, where he is currently a Reader. He has co-authored 4 book chapters and over 160 peer-reviewed research papers. He was an IET TPN Vision and Imaging Executive Team member, Associate Editor of IEEE Communications Letters, member of IEEE Communications Review Board, and Technical Program Committee co-chair of Eusipco- 2012. Currently, he is Associate Editor of IEEE Transactions on Image Processing, IEEE Transactions on Communications, and Elsevier Signal Processing: Image Communication. His research interests include source/channel/network coding, user-experience driven image processing and communications and energy disaggregation.

Samuel Cheng received the B.S. degree in Electrical and Electronic Engineering from the University of Hong Kong, and the M.Phil. degree in Physics and the M.S. degree in Electrical Engineering from Hong Kong University of Science and Technology and the University of Hawaii, Honolulu, respectively. He received the Ph.D. degree in Electrical Engineering from Texas A&M University in 2004. He worked in Microsoft Asia, China, and Panasonic Technologies Company, New Jersey, in the areas of texture compression and digital watermarking during the summers of 2000 and 2001. In 2004, he joined Advanced Digital Imaging Research, a research company based near Houston, Texas, as a Research Engineer to perform biomedical imaging research and was promoted to Senior Research Engineer the next year. Since 2006, he joined the School of Electrical and Computer Engineering at the University of Oklahoma and is currently an associate professor. He has been awarded six US patents in miscellaneous areas of signal processing. He is a senior member of IEEE and a member of ACM. His research interests include information theory, image/signal processing, and pattern recognition.



En-hui Yang (M'97-SM'00-F'08) received the B.S. degree in applied mathematics from Huaqiao University, Quanzhou, China, and Ph.D. degree in mathematics from Nankai University, Tianjin, China, in 1986 and 1991, respectively. Since June 1997, he has been with the Department of Electrical and Computer Engineering, University of Waterloo, ON, Canada, where he is currently a Professor and Canada Research Chair in information theory and multimedia compression, and the founding Director of the Leitch-University of Waterloo multimedia communications lab. He held a Visiting Professor position at the Chinese University of Hong Kong, Hong Kong, from September 2003 to June 2004; positions of Research Associate and Visiting Scientist at the University of Minnesota, Minneapolis-St. Paul, the University of Bielefeld, Bielefeld, Germany, and the University of Southern California, Los Angeles, from January 1993 to May 1997; and a faculty position (first as an Assistant Professor and then an Associate Professor) at Nankai University, Tianjin, China, from 1991 to 1992. A Co-Founder of SlipStream Data Inc. (now a subsidiary of BlackBerry), he currently also serves as an Executive Council Member of China Overseas Exchange Association, an Overseas Advisor for the Overseas Chinese Affairs Office of the City of Shanghai, and a director of Board of Trustees of Huaqiao University, China, and serves on the Overseas Expert Advisory Committee for the Overseas Chinese Affairs Office of the State Council of China. His current research interests are: multimedia compression, multimedia transmission, digital communications, information theory, source and channel coding, image and video coding, image and video understanding and management, big data analytics, and information security. Dr. Yang is a recipient of several awards and honors, a partial list of which includes the prestigious Inaugural Premier's Catalyst Award in 2007 for the Innovator of the Year; the 2007 Ernest C. Manning Award of Distinction, one of the Canada's most prestigious innovation prizes; the 2013 CPAC Professional Achievement Award; the 2014 IEEE Information Theory Society Padovani Lecture; and the 2014 FCCP Education Foundation Award of Merit. He has exemplified research excellence in both theory and practice. Products based on his early inventions and commercialized by his previous company, SlipStream, received the 2006 Ontario Global Traders Provincial Award. With over 210 papers and more than 200 patents/patent applications worldwide, his research work has benefited people over 170 countries through commercialized products, video coding open sources, and video coding standards. He is a Fellow of the Canadian Academy of Engineering and a Fellow of the Royal Society of Canada: the Academies of Arts, Humanities and Sciences of Canada. He served, among many other roles, as a review panel member for the International Council for Science; a General Co-Chair of the 2008 IEEE International Symposium on Information Theory; an Associate Editor for IEEE Transactions on Information Theory; a Technical Program Vice-Chair of the 2006 IEEE International Conference on Multimedia & Expo (ICME); the Chair of the award committee for the 2004 Canadian Award in Telecommunications; a Co-Editor of the 2004 Special Issue of the IEEE Transactions on Information Theory; a Co-Chair of the 2003 US National Science Foundation (NSF) workshop on the interface of Information Theory and Computer Science; and a Co-Chair of the 2003 Canadian Workshop on Information Theory.