# Minimally modified balanced codes

Schouhamer Immink, Kees A.; Weber, Jos H.

**DOI**
10.1109/TIT.2022.3200136

**Publication date**
2023

**Document Version**
Accepted author manuscript

**Published in**
IEEE Transactions on Information Theory

# Minimally modified balanced codes

Kees A. Schouhamer Immink, Fellow, IEEE and Jos H. Weber, Senior Member, IEEE

***Abstract*—We present and analyze a new construction of bipolar balanced codes where each codeword contains equally many $-1$'s and $+1$'s. The new code is minimally modified as the number of symbol changes made to the source word for translating it into a balanced codeword is as small as possible. The balanced codes feature low redundancy and time complexity. Large look-up tables are avoided.**

***Keywords***– balanced code, constrained code, error propagation, Raney's Lemma.

## I. Introduction

Let $\boldsymbol{x} = (x_1, \ldots, x_n)$, $x_i \in \{-1, 1\}$, be a word of length $n$ with bipolar symbols. The *balance* of a word $\boldsymbol{x}$, denoted by $w(\boldsymbol{x})$, is defined by $w(\boldsymbol{x}) = \sum_{i=1}^{n} x_i$. A word is said to be *balanced* if $w(\boldsymbol{x}) = 0$, $n$ even, i.e., it consists of equal numbers of $-1$'s and $+1$'s. A code is said to be balanced if each word in the code is balanced. Balanced codes have found widespread application in various fields such as data transmission and data storage [1, 2, 3, 4, 5, 6]. Look-up tables for translating source words into balanced codewords and *vice versa* have been applied for small $n$ [7, 8]. Enumeration techniques [9, 10, 11, 12, 13] have been advocated for encoding and decoding balanced words as it achieves the minimum redundancy possible. The complexity of enumerative coding, mainly the coefficients look-up tables, grows with $n^2$, which makes it less practical if complexity is at a premium.

Knuth's implementation of balanced codes [14, 15, 16] is attractive for encoding large source words as its complexity scales linearly with word length $n$, but it requires a redundancy $\log_2 n$, which is, for large $n$, around twice the minimum redundancy of a code comprising the full set of balanced codewords. Modifications of Knuth's generic scheme bridging the gap between the minimum redundancy and that of Knuth's implementation are discussed in Al-Bassam and Bose [17, 18], Tallini, Capocelli, and Bose [19, 20], and Weber and Immink [21, 22].

Knuth's balancing method is handsomely simple: a first segment of the source word is inverted, i.e. flip the symbol sign, for balancing. In addition, a prefix (tag) that uniquely identifies the length of the inverted segment is forwarded to the receiver. A disadvantage of Knuth's method is that the encoder inverts on average $n/4+1$ symbols, which may result in extreme error propagation when the tag is received in error. Implementations having low redundancy, complexity, and error propagation are welcome alternatives to the art.

Kees A. Schouhamer Immink is with Turing Machines Inc, Willemskade 15d, 3016 DK Rotterdam, The Netherlands. E-mail: immink@turingmachines.com.

Jos H. Weber is with Delft University of Technology, Delft, The Netherlands. E-mail: j.h.weber@tudelft.nl.

**Our contributions:** We present a novel method for efficiently translating arbitrary user data into balanced codewords, which is based on *Raney's Lemma* also known as *Cycle Lemma* [23, 24, 25, 26]. As in Knuth's construction, the encoder judiciously inverts a number of source symbols for obtaining the codeword. The proposed code, however, is *minimally modified* as the number of symbol inversions is minimal, which is an attractive virtue for reconstructing the source word when errors are made during transmission. The encoder inverts, on average, approximately $\sqrt{n/2\pi}$, $n \gg 1$, symbols of the source word. Thus, for example, for $n = 1000$ only around twelve symbol inversions are required on average (assuming equiprobable source words).

Information regarding the symbol modifications made to the source word is encoded into a small redundant tag appended to the codeword. We investigate fixed- and variable-length tag schemes. Tags of multiple codewords can be combined so reducing the overall redundancy. The redundancy of a fixed-length tag scheme equals $\log_2(n/2 + 1)$. The average redundancy of a variable-length tag scheme approaches the minimum possible for asymptotically large values of $n$. The (time) complexity of the new balanced encoder and decoder grows linearly with $n$.

We start in Section II with a description of Raney's Lemma. In Section III, we detail the encoding and decoding algorithms. The code's redundancy is discussed in Section IV, and a performance comparison is given in Section V. Section VI furnishes the conclusions of our paper.

## II. Raney's Lemma

We start with two definitions. The $n$ *partial, or running, balances* of the index $i$, $1 \leq i \leq n$, denoted by $s(i, k)$, are defined by

$$s(i,k) = \sum_{j=i}^{i+k-1} x_j, \ 1 \leq i \leq n, \tag{1}$$

where we extend the sequence $\boldsymbol{x}$ by letting $x_{n+p} = x_p$ for $1 \leq p \leq n$. An index $i$ is said to be a *minimal* index of $\boldsymbol{x}$ if and only if all the partial balances are positive, i.e.

$$s(i,k) > 0, \ \ 1 \leq k \leq n. \tag{2}$$

In other words, an index $i$ is a minimal index of $\boldsymbol{x}$ if and only if the partial balances of $x_i, \ldots, x_n, x_1, \ldots, x_{i-1}$ are all positive. Note that it is immediate from (2) that if index $i$ is a minimal index then $x_i = x_{i+1} = 1$.

Define the set of all minimal indexes of $\boldsymbol{x}$ by $\sigma(\boldsymbol{x})$. If $w(\boldsymbol{x}) > 0$ there are, according to *Raney's Lemma* [23, 24], exactly $|\sigma(\boldsymbol{x})| = w(\boldsymbol{x})$ minimal indexes, where $|X|$ denotes the cardinality of a set $X$.

This article has been accepted for publication in IEEE Transactions on Information Theory. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIT.2022.3200136



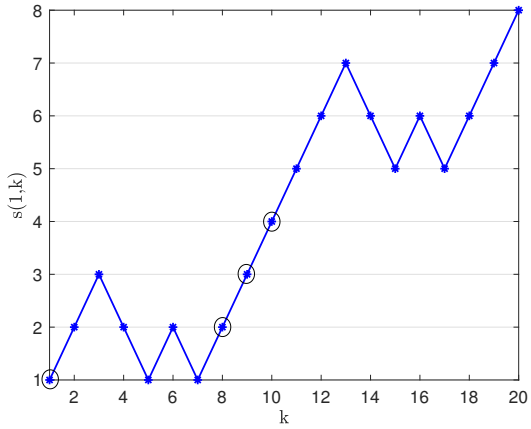


Fig. 1. Partial balance $s(1,k)$ versus $k$ for $n = 10$ and $\boldsymbol{x} = (1,1,1,-1,-1,1,-1,1,1,1)$. The diagram shows $s(1,k)$ in the extended interval $1 \le k \le 2n$, by letting $x_{n+p} = x_p$ for $1 \le p \le n$, which makes it more convenient to peruse the partial balances $s(i,k)$ for any $i$, as explained in the text. The minimal indexes of $\boldsymbol{x}$ are the $k$-values of the points indicated by the circles.

**Example** *1:* Let $n = 10$ and $\boldsymbol{x} = (1,1,1,-1,-1,1,-1,1,1,1)$. There are $w(\boldsymbol{x}) = 4$ minimal indexes. Figure 1 illustrates the partial balances, $s(1,k)$, versus $k$. We can check that index $i = 1$ is a minimal index, since all partial balances $s(1,k)$ are positive. Further, note that $s(i,k) = s(1,k+i-1) - s(1,i-1)$, This implies that for any $i$, $1 \le i \le n$ the $s(i,k)$ curve with $1 \le k \le n$ can be obtained from the curve in Figure 1 by considering it from $k = i$ up to $k = i+n-1$ and then shifting this segment $i-1$ units to the left and $s(1,i-1)$ units downwards. Hence it follows that the minimal index set is $\sigma(\boldsymbol{x}) = \{1,8,9,10\}$.

## III. Raney's Lemma-based balanced codes

### A. Antipodal matchings

Ordentlich and Roth [25] pioneered *antipodal matchings* for two-dimensional weight-constrained codes, which are based on Raney's Lemma. Before showing their results, we define the function $\boldsymbol{y} = f(\boldsymbol{x}, S)$, where $S$ is a subset of $\{1,\ldots,n\}$, by

$$y_i = \begin{cases} -x_i, & i \in S, \\ x_i, & i \notin S. \end{cases} \tag{3}$$

Ordentlich and Roth [25] showed that all $n$-bit input words, $\boldsymbol{x}$, of balance $w(\boldsymbol{x}) > 0$ can be converted into $n$-bit output words, $\boldsymbol{y}$, of inverted balance $w(\boldsymbol{y}) = -w(\boldsymbol{x})$ by

$$\boldsymbol{y} = f(\boldsymbol{x}, \sigma(\boldsymbol{x})). \tag{4}$$

In other words, we simply obtain the entries $y_i$ of $\boldsymbol{y}$ by inverting the $+1$'s at all minimal indexes of $\boldsymbol{x}$ to $-1$'s. For the other indexes we simply have $y_i = x_i$. Ordentlich and Roth proved that the above antipodal matchings are bijective mappings, and they presented an efficient (linear-time complexity) algorithm for finding the set of minimal indexes $\sigma(\boldsymbol{x})$, $w(\boldsymbol{x}) > 0$, for all word lengths $n$. They generalized the algorithm to words $\boldsymbol{x}$ with $w(\boldsymbol{x}) < 0$.

At first sight, the above algorithm looks superfluous as purely reversing the sign of a word balance is obviously achieved by inverting *all* symbols of $\boldsymbol{x}$. The algorithm based on Raney's Lemma, however, has the advantage that a minimal plurality of symbols ($+1$'s only if $w(\boldsymbol{x}) > 0$ or $-1$'s only if $w(\boldsymbol{x}) < 0$) is inverted, which is a highly attractive feature for constructing two-dimensional weight-constrained codes as shown in [25]. Below we show that Raney's Lemma can be harnessed to balance codewords with a minimal number of symbol inversions.

### B. Balanced codes

Let $\mathcal{S}_w$ denote the set of $n$-bit words, $\boldsymbol{x}$, whose balance equals $w(\boldsymbol{x}) = w$, that is

$$\mathcal{S}_w = \left\{ \boldsymbol{x} \in \{-1,1\}^n : \sum_{i=1}^{n} x_i = w \right\}. \tag{5}$$

Note that $\mathcal{S}_0$ is the set of balanced words.

Let the (minimal) indexes in $\sigma(\boldsymbol{x})$ be ordered in magnitude, that is $\sigma(\boldsymbol{x}) = \{i_1, i_2, \ldots, i_w\}$, where $i_1 < i_2 < \ldots < i_{w-1} < i_w$. For $w > 0$ we define the mapping $\phi(.)$ between $\boldsymbol{x} \in \mathcal{S}_w$ and the balanced $\boldsymbol{y} = \phi(\boldsymbol{x}) \in \mathcal{S}_0$, where

$$\boldsymbol{y} = \phi(\boldsymbol{x}) = f(\boldsymbol{x}, \{i_1, i_2, \ldots, i_{\frac{w}{2}}\}). \tag{6}$$

Clearly $w(\phi(\boldsymbol{x})) = 0$.

The following lemma shows some important properties, which will be used later. Essential parts have been presented in [25, Prop. 4.6].

*Lemma 1: For* $\boldsymbol{x} \in \{-1,1\}^n$ *with* $w(\boldsymbol{x}) = w > 0$ *and* $\sigma(\boldsymbol{x}) = \{i_1, i_2, \ldots, i_w\}$ *with* $i_1 < i_2 < \ldots < i_{w-1} < i_w < i_{w+1} = i_1 + n$, *it holds for all* $1 \le j \le w$ *and* $i_j + 1 \le v \le i_{j+1} - 1$ *that*

$$(i) \quad \sum_{i=i_j+1}^{v} x_i \ge 0,$$

$$(ii) \quad \sum_{i=i_j+1}^{i_{j+1}-1} x_i = 0,$$

$$(iii) \quad \sum_{i=v}^{i_{j+1}-1} x_i \le 0.$$

**Proof:** (i) Since $i_j$ is a minimal index of $\boldsymbol{x}$, we have

$$\sum_{i=i_j+1}^{v} x_i = \sum_{i=i_j}^{v} x_i - x_{i_j} \ge 1 - 1 = 0. \tag{7}$$

(ii) Note that

$$\begin{aligned} \sum_{j=1}^{w} \sum_{i_j+1}^{i_{j+1}-1} x_i &= \sum_{j=1}^{w} \sum_{i=i_j}^{i_{j+1}-1} x_i - \sum_{j=1}^{w} x_{i_j} \\ &= \sum_{i=1}^{n} x_i - \sum_{j=1}^{w} x_{i_j} = w - w = 0. \end{aligned}$$

This article has been accepted for publication in IEEE Transactions on Information Theory. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIT.2022.3200136



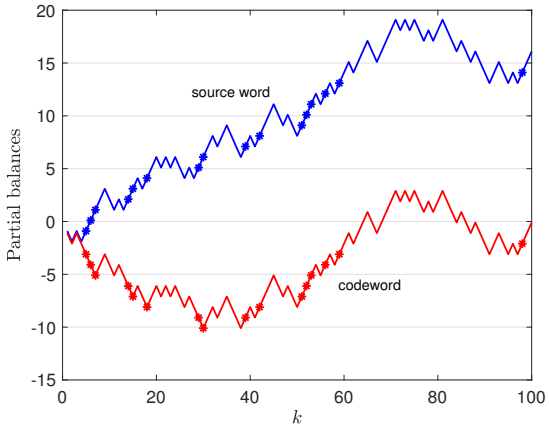


Fig. 2. Partial balances, $s(1,k)$, of a) an arbitrary source word, $\boldsymbol{x}$, and b) that of the balanced codeword $\boldsymbol{y} = \phi(\boldsymbol{x})$ versus index $k$ for $n = 100$; word balance equals $w = w(\boldsymbol{x}) = 16$. The partial balances at the minimal indexes of $\boldsymbol{x}$ are indicated by a '*'.

Since

$$\sum_{i=i_j+1}^{i_{j+1}-1} x_i \geq 0 \ \forall j$$

because of (i), the result follows.
(iii) It follows from (i) and (ii) that

$$\sum_{i=v}^{i_{j+1}-1} x_i = \sum_{i=i_j+1}^{i_{j+1}-1} x_i - \sum_{i=i_j+1}^{v-1} x_i \leq 0 - 0 = 0.$$

■

Note that this lemma deals with sums of symbols at positions in the strings between two (cyclicly) consecutive minimal indexes. In particular, it says that the sum of the symbols in

(i) any head of such string is nonnegative,
(ii) the complete string is equal to zero,
(iii) any tail of such string is nonpositive.

As a visual illustration we have plotted in Figure 2 the partial balances, $s(1,k)$, of a) an arbitrary source word $\boldsymbol{x}$ of length $n = 100$, $w(\boldsymbol{x}) = 16$ and b) the partial balances of the balanced codeword $\boldsymbol{y} = \phi(\boldsymbol{x})$. The partial balances at the minimal indexes of $\boldsymbol{x}$ are indicated by a '*'. The various properties discussed in Lemma 1 can easily be noted. For example, note the unity balance increments between consecutive minimal indexes. The partial balances of the balanced codeword $\boldsymbol{y}$ are indicated by a '*' at the minimal indexes of source word $\boldsymbol{x}$. For the smallest $w/2$ minimal indexes of $\boldsymbol{x}$ we note unity balance decrements between consecutive minimal indexes, while for the largest $w/2$ minimal indexes we note unity balance increments between consecutive minimal indexes.

### C. Encoding

We propose the following encoding rule, denoted by $\boldsymbol{y} = \psi(\boldsymbol{x})$, for translating an $n$-bit source word $\boldsymbol{x}$, $\boldsymbol{x} \in \{-1,1\}^n$, into a balanced $n$-bit codeword $\boldsymbol{y}$, $\boldsymbol{y} \in \mathcal{S}_0$:

$$\boldsymbol{y} = \psi(\boldsymbol{x}) = \begin{cases} \phi(\boldsymbol{x}), & w(\boldsymbol{x}) > 0, \\ -\phi(-\boldsymbol{x}), & w(\boldsymbol{x}) < 0, \\ \boldsymbol{x}, & w(\boldsymbol{x}) = 0, \end{cases} \tag{8}$$

*Input:* The bipolar $n$-bit word $(x_1, \ldots, x_n)$, $x_i \in \{-1, 1\}$.
*Output:* Encoded $n$-bit bipolar word $\boldsymbol{y}$ and tag $w$, i.e. ENC($\boldsymbol{x}$) = $(\boldsymbol{y}, w)$.

begin
let $w = \sum_{i=1}^{n} x_i$
if $w = 0$ $\boldsymbol{y} = \boldsymbol{x}$ halt
if $w < 0$ set $\boldsymbol{x} = -\boldsymbol{x}$ {invert all symbols}
run Algorithm [25, Fig. 6] yielding $\{i_1, \ldots, i_w\}$
for $i \in \{i_1, \ldots, i_{\frac{w}{2}}\}$ set $x_i = -1$
if $w > 0$ set $\boldsymbol{y} = \boldsymbol{x}$
if $w < 0$ set $\boldsymbol{y} = -\boldsymbol{x}$ {invert all symbols}
end.

Fig. 3. Basic encoding algorithm ENC($\boldsymbol{x}$).

where $-\boldsymbol{x}$ denotes $(-x_1, -x_2, \ldots, -x_n)$. Figure 3 shows the basic encoding algorithm ENC($\boldsymbol{x}$). Part of the encoding table, $n = 6$, has been tabulated in Table I; note that for clerical convenience a '0' indicates a '$-1$' symbol. As $\psi(-\boldsymbol{x}) = -\psi(\boldsymbol{x})$ we can easily extend the table. Note that in the mapping

TABLE I
PART OF ENCODING TABLE $\boldsymbol{y} = \psi(\boldsymbol{x})$ FOR $n = 6$. A '0' INDICATES '$-1$'.

| $\boldsymbol{x}$ | $\psi(\boldsymbol{x})$ | $\boldsymbol{x}$ | $\psi(\boldsymbol{x})$ |
|---|---|---|---|
| 000000 | 111000 | 001000 | 101100 |
| 000001 | 110001 | 001001 | 101001 |
| 000010 | 110010 | 001010 | 101010 |
| 000011 | 100011 | 001011 | 001011 |
| 000100 | 110100 | 001100 | 001110 |
| 000101 | 100101 | 001101 | 001101 |
| 000110 | 100110 | 001110 | 001110 |
| 000111 | 000111 | 001111 | 000111 |

$\boldsymbol{y} = \psi(\boldsymbol{x})$ the $|w(\boldsymbol{x})/2|$ rightmost symbols of the codeword $\boldsymbol{y}$ equal those of the source word $\boldsymbol{x}$. We have $\boldsymbol{y} = \psi(\boldsymbol{x}) \implies x_i = y_i$, $i = n - w/2 + 1, \ldots, n$, where $w = w(\boldsymbol{x}) > 0$. By definition of $\boldsymbol{y} = \psi(\boldsymbol{x})$, see (6), (8), and (9), only the symbols are inverted at indexes in $\{i_1, i_2, \ldots, i_{w/2}\}$. We have $\{i_1, i_2, \ldots, i_{w/2}\} \subset \{1, \ldots, n - w(\boldsymbol{x})/2\}$, so that the $w(\boldsymbol{x})/2$ rightmost symbols of $\boldsymbol{x}$ are unchanged. As $w(\boldsymbol{x}) \geq 2$ we have $y_n = x_n$ for all $\boldsymbol{x}$.

The receiver is able to uniquely recover $\boldsymbol{x}$ from the received (balanced) $\boldsymbol{y} = \psi(\boldsymbol{x})$ if we add a tag to the sent $\boldsymbol{y}$ that uniquely identifies the balance of the source word $\boldsymbol{x}$. A tag can be sent separately as a pre- or postfix or we may combine multiple tags to form a large tag data word. The code redundancy is discussed in Section IV.

### D. Decoding

The decoder uniquely retrieves a facsimile $\boldsymbol{x}'$ of the original source word, $\boldsymbol{x}$, from the received (balanced) $\boldsymbol{y} = \psi(\boldsymbol{x})$ and tag associated with the balance of the source word, $w(\boldsymbol{x})$. Figure 4 shows a description of the basic decoding algorithm. Note that the decoder (time) complexity grows linearly with word length $n$. The next theorem shows that

*Input:* The integer $w = w(\boldsymbol{x}) \in \{-n, -n+2, \ldots, n\}$, and the bipolar $n$-bit balanced word $(y_1, \ldots, y_n) = \psi(\boldsymbol{x}) \in \mathcal{S}_0$, $y_i \in \{-1, 1\}$.
*Output:* Decoded $n$-bit bipolar word DEC($\boldsymbol{y}, w$) = $\boldsymbol{x}'$.

*Initialize:*
if $w = 0$ $\boldsymbol{x}' = \boldsymbol{y}$ halt;
if $w < 0$ $\boldsymbol{v} = -\boldsymbol{y}$ {invert all symbols}
if $w > 0$ $\boldsymbol{v} = \boldsymbol{y}$
set $w_2 = \text{abs}(\frac{w}{2})$

begin
let $z_i = \sum_{j=1}^{i} v_j, \forall i = 1, \ldots, n$
let $m = \min\{z_i\}$
let $i'_j = \min\{i : z_i = m + w_2 - j\}$ $\forall j = 1, \ldots, w_2$
let $\boldsymbol{v}' = f(\boldsymbol{v}, \{i'_1, \ldots, i'_{w_2}\})$
if $w > 0$ $\boldsymbol{x}' = \boldsymbol{v}'$
if $w < 0$ $\boldsymbol{x}' = -\boldsymbol{v}'$ {invert all symbols}
end.

Fig. 4. Basic decoding algorithm DEC($\boldsymbol{y}, w$).

the decoding algorithm is correct, that is, DEC(ENC($\boldsymbol{x}$)) = $\boldsymbol{x}$.

*Theorem 1: For any $\boldsymbol{x} \in \{-1, 1\}^n$, it holds that DEC(ENC($\boldsymbol{x}$)) = $\boldsymbol{x}$ .*

**Proof:** We show that the decoding algorithm, shown in Figure 4, with input $(\psi(\boldsymbol{x}), w(\boldsymbol{x}))$ is correct and generates the original source word $\boldsymbol{x}$ as an output. From the encoding and decoding procedures, this is trivially true if $w(\boldsymbol{x}) = 0$, while correctness of the $w(\boldsymbol{x}) > 0$ case implies that it is also true for the $w(\boldsymbol{x}) < 0$ case. Hence, we further assume that $w = w(\boldsymbol{x}) > 0$. Note that

$$v_i = \begin{cases} -x_i = -1 & \text{if } i \in \{i_1, \ldots, i_{\frac{w}{2}}\}, \\ x_i = 1 & \text{if } i \in \{i_{\frac{w}{2}+1}, \ldots, i_w\}, \\ x_i & \text{otherwise.} \end{cases} \tag{9}$$

Let $a$ be the sum of the first $i_1 - 1$ entries of $\boldsymbol{v}$, i.e.,

$$a = z_{i_1-1} = \sum_{i=1}^{i_1-1} v_i = \sum_{i=1}^{i_1-1} x_i. \tag{10}$$

It follows from Lemma 1 (ii) and (9) that

$$z_{i_j} = a - j, \ \forall j \in \{1, 2, \ldots, \frac{w}{2}\}. \tag{11}$$

Furthermore we have

$$z_i \geq \begin{cases} a - \frac{w}{2} & \forall i \in \{i_{\frac{w}{2}}, \ldots, n\}, \\ a - j & \forall j \in \{1, \ldots, \frac{w}{2} - 1\}, i \in \{i_j, \ldots, i_{j+1} - 1\}, \\ a & \forall i \in \{1, \ldots, i_1 - 1\}, \end{cases} \tag{12}$$

where the first two inequalities follow from (9), (11), and Lemma 1 (i), while the third inequality follows from the fact that $z_i < a$ would imply with (9) and (10) that

$$\sum_{j=i+1}^{i_1-1} x_j = \sum_{j=i+1}^{i_1-1} v_j = z_{i_1-1} - z_i > a - a = 0,$$

which contradicts Lemma 1 (iii). Hence, (11) and (12) give that $m = a - \frac{w}{2}$ and that for any $j \in \{1, \ldots, \frac{w}{2}\}$ the smallest $i$ such that $z_i = m + \frac{w}{2} - j = a - j$ is $i = i_j$, and thus that $i'_j = i_j$. In conclusion, the decoder output satisfies

$$\boldsymbol{x}' = \boldsymbol{v}' = f(\boldsymbol{v}, \{i'_1, \ldots, i'_{\frac{w}{2}}\}) = f(\boldsymbol{y}, \{i_1, \ldots, i_{\frac{w}{2}}\}) = \boldsymbol{x}.$$

■

## IV. Redundancy

The number of balanced codewords of length $n$ equals

$$|\mathcal{S}_0| = \binom{n}{\frac{n}{2}}, \tag{13}$$

and thus the minimum redundancy of balanced codewords of length $n$, denoted by $H_0$, is

$$H_0 = n - \log_2 |\mathcal{S}_0| = n - \log_2 \binom{n}{\frac{n}{2}}. \tag{14}$$

For asymptotically large $n$ we have the approximation [14]

$$H_0 \approx \frac{1}{2} \log_2 n + 0.326, \ n \gg 1. \tag{15}$$

The redundancy of the new code is governed by the amount of data required to recover the balance $w(\boldsymbol{x})$ of the source word $\boldsymbol{x}$. The balance $w(\boldsymbol{x}) \in \{-n, -n+2, \ldots, n-2, n\}$ so that for the simplest fixed-length tag scheme, the redundancy is $\log_2(n+1)$. The next theorem will help to reduce the redundancy.

*Theorem 2: Let $\boldsymbol{y} \in \mathcal{S}_0$, $z_i = \sum_{j=1}^{i} y_j$, for $i = 1, \ldots, n$, $z_{\min} = \min\{z_i\}$, and $z_{\max} = \max\{z_i\}$. Then it holds that*

$$|\{\boldsymbol{x} \in \mathcal{S}_w : \psi(\boldsymbol{x}) = \boldsymbol{y}\}| = \begin{cases} 1 & \text{if } w \in \{-2z_{\max}, \\ & \quad -2z_{\max} + 2, \ldots, -2z_{\min}\}, \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

**Proof:** From Theorem 1 it follows that the mapping ENC($\boldsymbol{x}$) from $\{-1, 1\}^n$ to $\mathcal{S}_0 \times \{-n, -n+2, \ldots, n\}$ is injective. Hence, for each $w \in \{-n, -n+2, \ldots, n\}$, there is at most one word $\boldsymbol{x} \in \mathcal{S}_w$ for which $\psi(\boldsymbol{x}) = \boldsymbol{y}$. In items (i)-(v) below we investigate for which values of $w$ such a word $\boldsymbol{x}$ exists. Define $i'_j = \min\{i : z_i = z_{\min} + w/2 - j\}$, $j = 1, \ldots, w/2$, and observe that $\boldsymbol{y} \in \mathcal{S}_0$ implies

$$z_{\min} \leq 0 \leq z_{\max}.$$

(i) If $w = 0$, then there does exist an $\boldsymbol{x} \in \mathcal{S}_w$ such that $\psi(\boldsymbol{x}) = \boldsymbol{y}$, namely $\boldsymbol{x} = \boldsymbol{y}$, which immediately follows from (8).

(ii) If $w \in \{2, 4, \ldots, -2z_{\min}\}$ then there does exist an $\boldsymbol{x} \in \mathcal{S}_w$ such that $\psi(\boldsymbol{x}) = \boldsymbol{y}$, namely $\boldsymbol{x} = f(\boldsymbol{y}, \{i'_1, \ldots, i'_{w/2}\})$. This can be checked as follows. Note that $z_{i'_j} = z_{\min} + w/2 - j < 0$ and $x_{i'_j} = -y_{i'_j} = 1$ for $j = 1, 2, \ldots, w/2$, while $x_i = y_i$ for all indexes $i \neq i'_j$. On the one hand, observe that any $i$ with $i'_j < i < i'_{j+1}$, $j \in \{0, 1, \ldots, w/2 - 1\}$, $i'_0 = 0$, is not a minimal index of $\boldsymbol{x}$, since

$$\sum_{m=i}^{i'_{j+1}-1} x_m = \sum_{m=i}^{i'_{j+1}-1} y_m = z_{i'_{j+1}-1} - z_{i-1} \leq 0.$$

On the other hand, any $i'_j$, $j = 1, \ldots, w/2$, is a minimal index of $\boldsymbol{x}$, since for all $k \in \{1, 2, \ldots, n\}$ it holds that

$$\sum_{i=i'_j}^{i'_j+k-1} x_i \geq \sum_{i=i'_j}^{i'_j+k-1} y_i + 2b \geq -b + 2b = b \geq 1,$$

where $b = |\{m \in \{j, j+1, \ldots, w/2\} : i'_j \leq i'_m \leq i'_j + k - 1\}|$. In conclusion, $i'_1, \ldots, i'_{w/2}$ are the $w/2$ smallest minimal indexes of $\boldsymbol{x}$, and thus $\psi(\boldsymbol{x}) = f(\boldsymbol{x}, \{i'_1, \ldots, i'_{w/2}\}) = \boldsymbol{y}$.

(iii) If $w \in \{-2z_{\min} + 2, -2z_{\min} + 4, \ldots, n\}$, then there is no $\boldsymbol{x} \in \mathcal{S}_w$ for which $\psi(\boldsymbol{x}) = \boldsymbol{y}$, as we will show next. Suppose there does exist such $\boldsymbol{x}$. Let the $w/2$ smallest minimal indexes of $\boldsymbol{x}$ be $i_1, \ldots, i_{w/2}$. Since $\boldsymbol{y} = f(\boldsymbol{x}, \{i_1, \ldots, i_{w/2}\})$ and $\boldsymbol{x} = f(\boldsymbol{y}, \{i'_1, \ldots, i'_{w/2}\})$, it follows that $i_j = i'_j \ \forall j$. Hence, we obtain the contradiction

$$z_{i_{\frac{w}{2}}} = \sum_{i=1}^{i_1-1} y_i + \sum_{i=i_1}^{i_{\frac{w}{2}}} y_i = \sum_{i=1}^{i_1-1} x_i - \frac{w}{2} \leq -\frac{w}{2} < z_{\min},$$

where the first inequality follows from Lemma 1 (iii) and the second from the fact that $w > -2z_{\min}$.

(iv) If $w \in \{-n, -n+2, \ldots, -2z_{\max} - 2\}$, then there is no $\boldsymbol{x} \in \mathcal{S}_w$ for which $\psi(\boldsymbol{x}) = \boldsymbol{y}$, which can be shown in a similar way as (iii).

(v) If $w \in \{-2z_{\max}, -2z_{\max} + 2, \ldots, -2\}$, then there exists an $\boldsymbol{x} \in \mathcal{S}_w$ such that $\psi(\boldsymbol{x}) = \boldsymbol{y}$, which can be shown in a similar way as (ii).

■

Define

$$N(\boldsymbol{y}) = z_{\max} - z_{\min} + 1, \tag{17}$$

where $N(\boldsymbol{y})$ is called the *balance span* of $\boldsymbol{y}$. Let $r(\boldsymbol{y})$ denote the number of distinct source words $\boldsymbol{x} \in \{-1, 1\}^n$ that map to $\boldsymbol{y} \in \mathcal{S}_0$, that is

$$r(\boldsymbol{y}) = |\{\boldsymbol{x} \in \{-1, 1\}^n : \boldsymbol{y} = \psi(\boldsymbol{x})\}|, \ \boldsymbol{y} \in \mathcal{S}_0. \tag{18}$$

*Corollary 1: For all* $\boldsymbol{y} \in \mathcal{S}_0$ , it holds that

$$r(\boldsymbol{y}) = N(\boldsymbol{y}).$$

**Proof:** This result immediately follows from Theorem 2 by counting the number of $w$ for which $|\{\boldsymbol{x} \in \mathcal{S}_w : \psi(\boldsymbol{x}) = \boldsymbol{y}\}| = 1$. ■

### A. Fixed-length (FL) tag scheme

The tag length of a scheme with a fixed-length tag depends on the maximum value of $r(\boldsymbol{y})$, and for a variable-length scheme it depends on the distribution of $r(\boldsymbol{y})$. We easily find that $2 \leq r(\boldsymbol{y}) \leq n/2 + 1$. Note that the codeword denoted by $\boldsymbol{y}_1$ that starts with $n/2$ -1's and ends with $n/2$ +1's (and the $n-1$ circular shifts of $\boldsymbol{y}_1$) has the largest number of source words that map on it, namely the $n/2+1$ words, $\boldsymbol{x}$, that start with $p$, $p = 0, 1, \ldots, n/2$, -1's and end with $n-p$ +1's.

The decoder must be able to distinguish between at most $n/2+1$ source words that map on the received word, which makes it possible to reduce the tag length to $\log_2(n/2+1)$. To do so, the encoder first computes $\boldsymbol{y} = \psi(\boldsymbol{x})$ using the encoding algorithm, see Figure 3, and subsequently it computes $r(\boldsymbol{y})$. Using $r(\boldsymbol{y})$, the value $w(\boldsymbol{x})$ is uniquely encoded into the $n/2+1$ possible tag values, so that the decoder can uniquely recover $w(\boldsymbol{x})$ from the tag and $\boldsymbol{y}$. The redundancy of this scheme equals $\log_2(n/2+1)$.

### B. Variable-length (VL) tag scheme

The average redundancy of a VL tag scheme is less than that of the above fixed-length tag scheme. As the distribution of $r(\boldsymbol{y})$ is the same as that of Knuth's code, we follow [22] for the computation of the redundancy of the VL tag scheme. The number of balanced words $\boldsymbol{y}$ of length $n$ with $r(\boldsymbol{y}) = u$, denoted by $P(u, n)$, $2 \leq u \leq n/2+1$, is given by [22]

$$P(u, n) = D(u, n) - 2D(u-1, n) + D(u-2, n), \tag{19}$$

where

$$D(u, n) = 2^n \sum_{i=1}^{u} \cos^n \frac{\pi i}{u+1}. \tag{20}$$

The above expression is surprising as $D(u, n)$ is integer valued. Using a result by Merca [27] we may translate (20) into a summation of binomial coefficients

$$D(u, n) = (u+1) \sum_{k=-v}^{v} \binom{n}{\frac{n}{2} + k(u+1)} - 2^n, \tag{21}$$

where $v = \lfloor n/(2u+2) \rfloor$. The redundancy of the VL tag scheme, denoted by $H$, equals [22]

$$H = 2^{-n} \sum_{u=2}^{n/2+1} uP(u, n) \log_2 u. \tag{22}$$

The redundancy $H$ has been computed in [22, Table II] for selected values of $n \leq 2^{13}$. For $n = 2^{13}$, we find $H - H_0 \approx 0.033$. Eq. (19) is ill-conditioned as $P(u, n)$ is the difference between two much larger quantities. We were not able to obtain results of (22) for asymptotically large $n$, see also [2].

## V. Performance comparison

In this section, we discuss the number of modifications to a source word that are made by the prior art Knuth code [14] and the newly developed code. We start with the new method.

### A. New method

The probability, denoted by $Pr_1(\ell)$, that $\ell = |w(\boldsymbol{x})|/2$, $0 \leq \ell \leq n/2$, symbols of $\boldsymbol{x}$ are inverted to obtain $\psi(\boldsymbol{x})$ equals (assuming equiprobable source words)

$$Pr_1(\ell) = \begin{cases} \frac{1}{2^n}\binom{n}{\frac{n}{2}}, & \ell = 0, \\ \frac{1}{2^{n-1}}\binom{n}{\frac{n}{2}+\ell}, & 1 \leq \ell \leq \frac{n}{2}. \end{cases} \tag{23}$$

The average number of symbol inversions, denoted by $\bar{\ell}_1$, equals

$$\bar{\ell}_1 = \sum_{\ell=1}^{\frac{n}{2}} \ell Pr_1(\ell). \tag{24}$$

This article has been accepted for publication in IEEE Transactions on Information Theory. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIT.2022.3200136



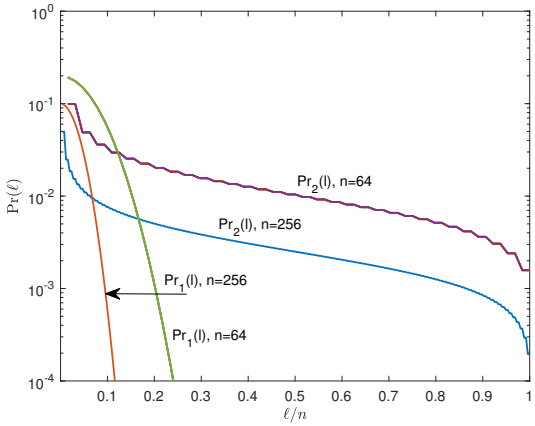


Fig. 5. Distributions $Pr_1(\ell)$ and $Pr_2(\ell)$ versus the (relative) number of symbol inversions $\ell/n$. Word length $n = 64$ and $n = 256$.

For large $n$, we obtain by using the well-known Gaussian approximation to the binomial coefficients,

$$\bar{\ell}_1 \approx \sqrt{\frac{n}{2\pi}}, \quad n \gg 1. \tag{25}$$

### B. Knuth's method

Knuth [14] presented a simple scheme for balancing large codewords. Let $\boldsymbol{x}$ be the $n$-bit ($n$ even) source word of bipolar symbols, $x_i \in \{-1, 1\}$. Knuth showed that there is a *balancing index*, $\ell$, such that

$$-\sum_{i=1}^{\ell} x_i + \sum_{i=\ell+1}^{n} x_i = 0, \ n \ \text{even}. \tag{26}$$

In other words, by inverting a first segment of $\ell$ symbols any word $\boldsymbol{x}$ of even length can be balanced. Note that the balancing index $\ell$ is not unique [21]. We assume here that the encoder selects the smallest balancing index from the set of balancing indexes. The distribution of the number of symbol inversions, $\ell$, for obtaining the balanced word in Knuth's scheme, denoted by $Pr_2(\ell)$, $1 \le \ell \le n$ ($1 \le j \le n/2$), has been computed by Weber and Immink [21] (assuming equiprobable source words)

$$\begin{aligned} Pr_2(2j) &= Pr_2(2j-1) \\ &= \frac{n-2j+1}{n2^{n-2}} \binom{2(j-1)}{j-1} \binom{n-2j}{\frac{n}{2}-j}. \end{aligned}$$

The average number of symbol inversions of Knuth's scheme, denoted by $\bar{\ell}_2$, simply equals, see Appendix,

$$\bar{\ell}_2 = \sum_{\ell=1}^{n} \ell Pr_2(\ell) = \frac{n}{4} + 1. \tag{27}$$

### C. Comparison of the two methods

Figure 5 shows two examples of the distributions $Pr_1(\ell)$ and $Pr_2(\ell)$ versus the relative number of symbols inversions $\ell/n$ for word lengths $n = 64$ and $n = 256$. We may notice that the distribution of Knuth's method, $Pr_2(\ell)$, is much wider than that of the new method, $Pr_1(\ell)$, which has a direct effect on the average number of inversions (bit changes) made. For example, for a codeword length $n = 1000$ around 12 symbol inversions are required on average per codeword for the new scheme. Knuth's code requires, on average, for the same codeword length, $n = 1000$, around 250 symbol inversions for translating source words into codewords.

## VI. Conclusions

We have presented a novel method for efficiently translating arbitrary user data into balanced codewords. The new code is minimally modified as the number of symbol changes made to the source word for translating it into a balanced codeword is minimal. The encoder inverts, on average, approximately $\sqrt{n/2\pi}$, $n \gg 1$, symbols of the source word, where $n$ denotes the source word length; the other code symbols being equal to the source symbols. The redundancy of the new method using a fixed-length tag is $\log_2(n/2+1)$. Large look-up tables for encoding and decoding are avoided. The (time) complexity of the new balanced encoder and decoder grows linearly with source word length $n$ for asymptotically large values of $n$.

## VII. Appendix

Let for $1 \le j \le \frac{n}{2}$

$$\begin{aligned} Pr_2(2j) &= Pr_2(2j-1) \\ &= \frac{n-2j+1}{n2^{n-2}} \binom{2(j-1)}{j-1} \binom{n-2j}{\frac{n}{2}-j}. \end{aligned} \tag{28}$$

*Theorem 3:*

$$\bar{\ell}_2 = \sum_{i=1}^{n} i Pr_2(i) = \frac{n}{4} + 1. \tag{29}$$

**Proof:** We simply find, combining $Pr_2(2i)$ and $Pr_2(2i-1)$,

$$\sum_{i=1}^{n} i Pr_2(i) = \sum_{i=1}^{\frac{n}{2}} (4i-1) Pr_2(2i). \tag{30}$$

Since $Pr_2(i)$ is a probability mass function, we have

$$\sum_{i=1}^{\frac{n}{2}} Pr_2(2i) = \frac{1}{2}, \tag{31}$$

and we obtain

$$\bar{\ell}_2 = 4 \sum_{i=1}^{\frac{n}{2}} i Pr_2(2i) - \frac{1}{2}. \tag{32}$$

Define the moments

$$m_k(n) = \sum_{j=1}^{\frac{n}{2}} j^k \binom{2(j-1)}{j-1} \binom{n-2j}{\frac{n}{2}-j}, \ k = 0, 1, 2, \tag{33}$$

then substituting into (32) yields

$$\bar{\ell}_2 = \frac{4(n+1)}{n2^{n-2}} m_1(n) - \frac{8}{n2^{n-2}} m_2(n) - \frac{1}{2}. \tag{34}$$

In the literature [24, pp. 187], we find

$$m_0(n) = 2^{n-2}. \tag{35}$$

As, see (28), (31), and (33),

$$\frac{n+1}{n2^{n-2}}m_0(n) - \frac{2}{n2^{n-2}}m_1(n) = \frac{1}{2}, \tag{36}$$

we obtain

$$m_1(n) = 2^{n-4}(n+2). \tag{37}$$

The zeroth and second moments, $m_0(n)$ and $m_2(n)$, are the autoconvolution of the sequence $\binom{2i}{i}$ and $i\binom{2i}{i}$, $i = 1, 2, \ldots$, respectively. The generating function of the autoconvolution is obtained by squaring the original generating function as presented in [24]. Due to space limitations, we omit the details, and summarize the result:

$$m_2(n) = 2^{n-7}(3n^2 + 6n + 8). \tag{38}$$

Substituting (37) and (38) into (34) proves the theorem. ■

## ACKNOWLEDGEMENT

The authors are indebted to dr. A.J.E.M. (Guido) Janssen for his assistance with the proof of Theorem 3 given in the Appendix.

## REFERENCES

[1] A. R. Calderbank, "The art of signaling: fifty years of coding theory," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2561-2595, Oct. 1998, doi: 10.1109/18.720549.

[2] D. T. Dao, H. M. Kiah, and T. T. Nguyen, "Average Redundancy of Variable-Length Balancing Schemes à la Knuth," ArXiv: 2204.13831, April 2022.

[3] O. P. Babalola and V. Balyan, 'Efficient Channel Coding for Dimmable Visible Light Communications System," *IEEE Access,* vol. 8, pp. 215100-215106, 2020, doi: 10.1109/ACCESS.2020.3041431.

[4] J. N. Franklin and J. R. Pierce, "Spectra and Efficiency of Binary Codes without DC," *IEEE Transactions on Communications,* vol. COM-20, no. 6, pp. 1182-1184, Dec. 1972, doi: 10.1109/TCOM.1972.1091308.

[5] F. Chang, W. Hu, D. Lee and C. Yu, "Design and implementation of anti low-frequency noise in visible light communications," 2017 International Conference on Applied System Innovation (ICASI), Sapporo, 2017, pp. 1536-1538, doi: 10.1109/ICASI.2017.7988219.

[6] K. A. S. Immink and K. Cai, "Properties and Constructions of Constrained Codes for DNA-based Data Storage," *IEEE Access,* vol. 8, pp. 49523-49531, 2020, doi: 10.1109/ACCESS.2020.2980036.

[7] A. X. Widmer and P. A. Franaszek, "A Dc-balanced, Partitioned-Block, 8B/10B Transmission Code," *IBM J. Res. Develop.,* vol. 27, no. 5, pp. 440-451, Sept. 1983, doi: 10.1147/rd.275.0440.

[8] C. N. Yang and D. J. Lee, "Some new efficient second-order spectral-null codes with small lookup tables," *IEEE Transactions on Computers,* vol. 55, no. 7, pp. 924-927, July 2006, doi: 10.1109/TC.2006.111.

[9] T. M. Cover, "Enumerative Source Coding," *IEEE Transactions on Information Theory,* vol. IT-19, no. 1, pp. 73-77, Jan. 1973, doi: 10.1109/TIT.1973.1054929.

[10] V. Braun and K. A. S. Immink, "An Enumerative Coding Technique for DC-free Runlength-Limited Sequences," *IEEE Transactions on Communications,* vol. 48, no. 12, pp. 2024-2031, Dec. 2000, doi: 10.1109/26.891213.

[11] J. P. M. Schalkwijk, "An Algorithm for Source Coding," *IEEE Transactions on Information Theory,* vol. IT-18, no. 3, pp. 395-399, May 1972, doi: 10.1109/TIT.1972.1054832.

[12] Y. Xin and I. J. Fair, "Algorithms to Enumerate Codewords for $DC^2$-Constrained Channels," *IEEE Transactions on Information Theory,* vol. IT-47, no. 7, pp. 3020-3025, Nov. 2001, doi: 10.1109/18.959281.

[13] A. Hareedy, B. Dabak, and R. Calderbank, "The Secret Arithmetic of Patterns: A General Method for Designing Constrained Codes Based on Lexicographic Indexing," *IEEE Transactions on Information Theory*, 2022. doi: 10.1109/TIT.2022.3170692.

[14] D. E. Knuth, "Efficient Balanced Codes," *IEEE Transactions on Information Theory,* vol. IT-32, no. 1, pp. 51-53, Jan. 1986, doi: 10.1109/TIT.1986.1057136

[15] H. D. L. Hollmann and K. A. S. Immink, "Performance of efficient balanced codes," *IEEE Transactions on Information Theory,* vol. IT-37, no. 3, pp. 913-918, May 1991, doi: 10.1109/18.79961.

[16] F. Paluncic, B. T. Maharaj, and H. C. Ferreira, "Variable- and Fixed-Length Balanced Runlength-Limited Codes Based on a Knuth-Like Balancing Method," *IEEE Transactions on on Information Theory,* vol. IT-65, no. 11, pp. 7045-7066, Nov. 2019, doi: 10.1109/TIT.2019.2914205.

[17] S. Al-Bassam and B. Bose, "On Balanced Codes," *IEEE Transactions on Information Theory,* vol. IT-36, no. 2, pp. 406-408, March 1990, doi: 10.1109/18.52490.

[18] S. Al-Bassam and B. Bose, "Design of Efficient Balanced Codes," *IEEE Transactions on Computers,* vol. 43, pp. 362-365, March 1994, doi: 10.1109/12.272436.

[19] L. G. Tallini, R. M. Capocelli, and B. Bose, "Design of some new efficient balanced bodes," *IEEE Transactions on Information Theory,* vol. IT-42, no. 3, pp. 790-802, May 1996, doi: 10.1109/18.490545.

[20] L. G. Tallini and B. Bose, "Balanced codes with parallel encoding and decoding," *IEEE Transactions on Computers,* vol. 48, no. 8, pp. 794-814, Aug. 1999, doi: 10.1109/12.795122.

[21] J. H. Weber and K. A. S. Immink, "Knuth's Balanced Codes Revisited," *IEEE Transactions on Information Theory,* vol. IT-56, no. 4, pp. 1673-1679, April 2010, doi: 10.1109/TIT.2010.2040868.

[22] K. A. S. Immink and J. H. Weber, "Very Efficient Balanced Codes," *IEEE Journal on Selected Areas of Communications,* vol. 28, no. 2, pp. 188-192, Feb. 2010, doi: 10.1109/JSAC.2010.100207.

[23] G. Raney, "Functional Composition Patterns and Power Series Reversion," *Transactions of the American Mathematical Society,* vol. 94. pp. 441-451, 1960.

[24] R. L. Graham, D. E. Knuth, and O. Patashnik, *Concrete mathematics: A Foundation for Computer Science* (2nd Edition), Addison-Wesley, ISBN-13: 978-0201558029, 2018.

[25] E. Ordentlich and R. M. Roth, "Low Complexity Two-Dimensional Weight-Constrained Codes", *IEEE Transactions on Information Theory,* vol. 58, no. 6, pp. 3892-3899, June 2012, doi: 10.1109/TIT.2012.2190380.

[26] T. T. Nguyen, K. Cai, K. A. S. Immink and Y. M. Chee, "Efficient Design of Capacity-Approaching Two-Dimensional Weight-Constrained Codes," 2021 IEEE International Symposium on Information Theory (ISIT), pp. 2930-2935, 2021, doi: 10.1109/ISIT45174.2021.9517970.

[27] M. Merca, "A note on cosine power series," *Journal of Integer Sequences*, vol. 15, no. 5, Article 15.5.3, MR2942751, 2012.

**Kees A. Schouhamer Immink** (M'81-SM'86-F'90) founded Turing Machines Inc. in 1998, an innovative start-up focused on novel signal processing for DNA-based storage, where he currently holds the position of president. He was from 1994 till 2014 an adjunct professor at the Institute for Experimental Mathematics, Essen-Duisburg University, Germany.

He contributed to digital video, audio, and data recording products including Compact Disc, CD-ROM, DCC, DVD, and Blu-ray Disc. He received the 2017 IEEE Medal of Honor, a Knighthood in 2000, a personal Emmy award in 2004, the 1999 AES Gold Medal, the 2004 SMPTE Progress Medal, the 2014 Eduard Rhein Prize for Technology, and the 2015 IET Faraday Medal. He received the Golden Jubilee Award for Technological Innovation by the IEEE Information Theory Society in 1998. He was inducted into the Consumer Electronics Hall of Fame, elected into the Royal Netherlands Academy of Sciences and the (US) National Academy of Engineering. He received an honorary doctorate from the University of Johannesburg in 2014. He served the profession as President of the Audio Engineering Society inc., New York, in 2003.

**Jos H. Weber** (S'87-M'90-SM'00) was born in Schiedam, The Netherlands, in 1961. He received the M.Sc. (in mathematics,

with honors), Ph.D., and MBT (Master of Business Telecommunications) degrees from Delft University of Technology, Delft, The Netherlands, in 1985, 1989, and 1996, respectively.

Since 1985 he has been with the Delft University of Technology. Currently, he is an associate professor at the Department of Applied Mathematics. He was the chairman of the Werkgemeenschap voor Informatie- en Communicatietheorie from 2006 until 2021. He is the secretary of the IEEE Benelux Chapter on Information Theory since 2008. He was a visiting researcher at the University of California (Davis, CA, USA), the Tokyo Institute of Technology (Japan), the University of Johannesburg (South Africa), EPFL (Switzerland), and SUTD (Singapore). His main research interests are in the area of channel coding.