Binary Variable-to-Fixed Length Balancing Scheme with Simple Encoding/Decoding

Swart, Theo G.; Weber, Jos H.

DOI
10.1109/LCOMM.2018.2865350

Publication date
2018

Document Version
Accepted author manuscript

Published in
IEEE Communications Letters

Binary Variable-to-Fixed Length Balancing Scheme with Simple Encoding/Decoding

Theo G. Swart, *Senior Member, IEEE*, and Jos H. Weber, *Senior Member, IEEE*

***Abstract*—We present a systematic variable-to-fixed (VF) length scheme encoding binary information sequences into binary balanced sequences. The redundancy of the proposed scheme is larger than the redundancy of the best fixed-to-fixed (FF) length schemes in case of long codes, but it is smaller in case of short codes. The biggest advantage comes from the simplicity of the scheme: encoding only requires one to keep track of the sequence weight, while decoding requires only one extremely simple step, irrespective of the sequence length.**

***Index Terms*—Balanced code, constrained code, dc-free code, variable length**

I. Introduction

A binary sequence is said to be balanced if the number of zeroes equals the number of ones. Hence, if we consider the bits as bipolar symbols, i.e., they are taken from the alphabet $\{-1, 1\}$ rather than from $\{0, 1\}$, then a sequence is balanced if the sum of the symbols equals zero. Balanced sequences, sometimes called dc-free codes, have found widespread application in magnetic and optical storage media [1], [2] and in cable communication [3].

The study of simple and efficient algorithms for transforming arbitrary binary sequences into balanced binary sequences has been an active field of research. Look-up tables can be used for short sequences, but this quickly becomes memory inefficient for long sequences. Knuth [4] presented two simple schemes for generating binary balanced sequences that are efficient even for very long sequences. The simple parallel scheme works by complementing the fixed-length information sequence, bit by bit, until balancing is achieved, and then using a fixed-length balanced prefix to convey this balancing point. At the receiver, the balancing point can be obtained from the prefix, and the original information sequence is retrieved by omitting the prefix and inverting the remaining sequence up to the balancing point. The price to pay for this elegancy is a somewhat higher redundancy when compared to that of the full set of all possible balanced sequences.

Ever since the presentation of Knuth's celebrated algorithms, many other binary balancing schemes have been proposed, such as [5]–[8]. Mostly these are based on the ideas of Knuth, offering some improvements with respect to redundancy and/or complexity. As in Knuth's scheme, most prior-art balancing methods convert fixed-length information sequences to fixed-length balanced sequences (FF schemes).

Manuscript received . . .; revised . . ..

T. G. Swart is with the Department of Electrical and Electronic Engineering Science, University of Johannesburg, Auckland Park 2006, South Africa (email: tgswart@uj.ac.za).

J. H. Weber is with the Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, 2628 Delft, The Netherlands (email: j.h.weber@tudelft.nl).

Schouhamer Immink and Weber [9] presented a more efficient balancing scheme, also based on Knuth's algorithm to modify the information. It uses a variable-length prefix to indicate which segment has been inverted. In essence, this is a scheme that converts fixed-length information sequences into variable-length balanced sequences. Steadman and Fair [10] presented a simple construction for converting variable-length information sequences into variable-length constrained sequences, based on the search of partial extensions of a minimal set of codeword lengths. It is shown that dc-free codes close to capacity can be constructed. If dc-free codes with very specific spectral performance are required, then the multicodes of Schouhamer Immink and Patrovics [11] can be used.

We will present a method to convert variable-length information sequences to fixed-length balanced sequences (VF scheme), and show its advantages in terms of decoding and memory requirements, as well as the received sequence being systematic, i.e., the information appears as is without being transformed. Furthermore, we will show that for small lengths the proposed VF scheme has a smaller redundancy than all FF schemes.

In the rest of this letter, let $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ represent a binary (bipolar) sequence of length n with $x_i \in \{-1, 1\}$, n even. The sequence $\boldsymbol{x}$ is balanced if $w = \sum_{i=1}^{n} x_i = 0$, where w represents the weight of the sequence. Since the information sequences will be of variable length, let k represent the average length of these sequences, and let r represent the average redundancy, i.e., $r = n - k$. It is assumed that the information source is memoryless and uniformly distributed.

This letter is organized as follows. In Section II we present the encoding and decoding of the new variable-to-fixed length balancing scheme. Section III studies the redundancy of the proposed scheme, while Section IV provides comparisons with fixed-to-fixed length balancing schemes. Section V concludes this letter.

II. Variable-to-Fixed (VF) Length Balancing

We illustrate the idea using $n = 6$, along with the weight diagram in Fig. 1, where the normal edges going to the left represent a -1 bit and the bold edges going to the right represent a 1 bit. For the bit in the first position we have a free choice, i.e., -1 or 1. The same applies to bits in the second and third positions. However, should the information sequence be $(-1, -1, -1)$ or $(1, 1, 1)$, it is obvious that the fourth bit, and in fact the fifth and sixth bits, must be constrained to achieve balancing, resulting in $(-1, -1, -1, 1, 1, 1)$ and $(1, 1, 1, -1, -1, -1)$, respectively. Thus in these cases only the first three bits represent information.

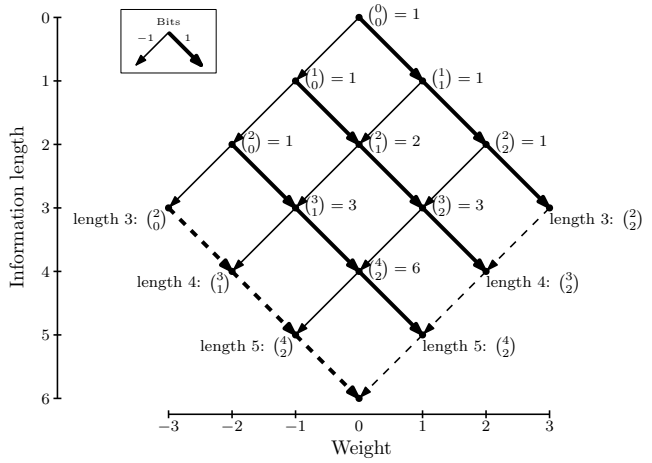


Fig. 1. Example of weight diagram and information lengths for $n = 6$

It is obvious that once the bottom edges of the weight diagram are reached, that one is forced to follow the dashed paths to reach the balanced state. By keeping track of the weight of the sequence as the weight diagram is being traversed, it can be determined whether the next bit will be an information bit or a constrained bit. All possible $n = 6$ balanced sequences, with the underlined bits representing the information bits, are:

$$\begin{array}{ll}
(\underline{-1,-1,-1},1,1,1), & (\underline{1,1,1},-1,-1,-1),\\
(\underline{-1,-1,1,-1},1,1), & (\underline{1,1,-1,1},-1,-1),\\
(\underline{-1,-1,1,1,-1},1), & (\underline{1,1,-1,-1,1},-1),\\
(\underline{-1,-1,1,1,1},-1), & (\underline{1,1,-1,-1,-1},1),\\
(\underline{-1,1,-1,-1},1,1), & (\underline{1,-1,1,1},-1,-1),\\
(\underline{-1,1,-1,1,-1},1), & (\underline{1,-1,1,-1,1},-1),\\
(\underline{-1,1,-1,1,1},-1), & (\underline{1,-1,1,-1,-1},1),\\
(\underline{-1,1,1,-1,-1},1), & (\underline{1,-1,-1,1,1},-1),\\
(\underline{-1,1,1,-1,1},-1), & (\underline{1,-1,-1,1,-1},1),\\
(\underline{-1,1,1,1,-1},-1), & (\underline{1,-1,-1,-1},1,1).
\end{array} \tag{1}$$

One can easily see how the weight diagram of Fig. 1 can be expanded to account for any sequence length, by extending the weight and length axes. Based on this, we next present the encoding/decoding algorithms.

A. Encoding

The first $\frac{n}{2}$ bits are information bits, as the sequences up to this point in the weight diagram need not be constrained. For the next $\frac{n}{2}$ bits it needs to be determined, according to the weight of the preceding bits, whether we are on the bottom edge of the weight diagram. More formally, the encoding algorithm is described as:

1) Take $\frac{n}{2}$ information bits from the source and set these as $(x_1, x_2, \ldots, x_{\frac{n}{2}})$.
2) Set $j = \frac{n}{2} + 1$.
3) If $|\sum_{i=1}^{j-1} x_i| \leq n - j$, then choose x_j to be a (new) information bit, else set $x_i = -x_{j-1}$ for $i = j, j+1, \ldots, n$ and end the algorithm.
4) Set $j \leftarrow j + 1$ and return to Step 3.

It is easy to check that the resulting $\boldsymbol{x}$ is indeed balanced. In order to illustrate the variable-to-fixed length nature of the encoder, we consider as an example a stream of source symbols starting with

$$1, 1, -1, 1, 1, 1, 1, -1, 1, \ldots$$

and a block length of six. Then the first codeword generated is

$$\boldsymbol{x}_1 = (1, 1, -1, 1, -1, -1),$$

representing the first four information symbols, the second codeword is

$$\boldsymbol{x}_2 = (1, 1, 1, -1, -1, -1),$$

representing the next three information symbols, etc. Note that, for the encoding process, the only memory requirement is space to store the sequence and the calculated weight.

B. Decoding

Decoding consists of one extremely simple step: drop the entire last run of (identical) bits from the received n-bit sequence. Hence, if the received balanced sequence is $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$, then the resulting information sequence is $(x_1, x_2, \ldots, x_g)$, with g being the largest integer from the set $\{n/2, n/2+1, \ldots, n-1\}$ such that $x_g \neq x_n$. For example, if $n = 6$ and $\boldsymbol{x} = (1, 1, -1, 1, -1, -1)$, then $g = 4$ and the information sequence is $(1, 1, -1, 1)$. This is further illustrated in (1), where for all sequences the (underlined) information bits will remain once the last run of bits is discarded. Note that, for the decoding process, there are no memory requirements, except for the received sequence itself, and no calculations are needed.

III. Redundancy

To determine the average information length, and thus also the average redundancy, we first consider again the $n = 6$ example in Fig. 1. The number of sequences at different points is related to Pascal's triangle. For length 0 we have one sequence, the empty sequence, for length 1 we have one sequence of weight -1, i.e., (-1), and one sequence of weight 1, i.e., (1), for length 2 we have one sequence of weight -2, i.e., $(-1, -1)$, one sequence of weight 2, i.e., $(1, 1)$, and two sequences of weight 0, i.e., $(-1, 1)$ and $(1, -1)$, and so forth. The labels on the bottom nodes indicate the number of information bits in the sequences up to that point, as well as the number of these sequences. So, for example, there are $\binom{3}{1} + \binom{3}{2} = 3 + 3 = 6$ sequences that contain 4 information bits, each occurring with probability 2^{-4}. Since there are further 2 sequences with 3 information bits and probability 2^{-3} and 12 information sequences with 5 information bits and probability 2^{-5}, we find that the average number of information bits in a coded sequence is

$$2 \times 3 \times 2^{-3} + 6 \times 4 \times 2^{-4} + 12 \times 5 \times 2^{-5} = 33/8 = 4.125,$$

and thus the average redundancy is $6 - 33/8 = 15/8 = 1.875$.

Generalizing this reasoning, we can determine the average information length k and the average redundancy r for any block length n. Actually, this is equivalent to solving (a slightly modified version of) Banach's matchbox problem [12] using

properties of the negative binomial distribution. The results are that

$$k = \sum_{l=\frac{n}{2}}^{n-1} \frac{l}{2^l}\left[\binom{l-1}{l-\frac{n}{2}} + \binom{l-1}{\frac{n}{2}-1}\right] = \sum_{l=\frac{n}{2}}^{n-1} \frac{l}{2^{l-1}}\binom{l-1}{\frac{n}{2}-1}$$

and

$$r = \sum_{j=1}^{\frac{n}{2}} \frac{j}{2^{n-j-1}}\binom{n-j-1}{\frac{n}{2}-1} = \frac{n-1}{2^{n-2}}\binom{n-2}{\frac{n}{2}-1}. \tag{2}$$

For very long sequences, we can can use Stirling's formula, i.e., $m! \approx \sqrt{2\pi m}\left(\frac{m}{e}\right)^m$, to approximate the last binomial coefficient in (2) as

$$\frac{2^{n-2}}{\sqrt{\frac{\pi}{2}(n-2)}},$$

giving

$$r \approx \frac{n-1}{\sqrt{\pi(\frac{n}{2}-1)}}.$$

Hence, the redundancy of our scheme is $O(\sqrt{n})$.

IV. Comparisons

From a redundancy perspective, the best fixed-length code consist of all balanced sequences of the length n under consideration, with all sequences being equiprobable. So for instance, in the case of $n = 6$, we have the 20 sequences listed in (1), ignoring the underlining, each with probability $\frac{1}{20}$, leading to a redundancy of $6 - \log_2 20 = 1.68$. In general, the full set of balanced sequences has redundancy

$$r^+ = n - \log_2 \binom{n}{n/2}. \tag{3}$$

However, encoding/decoding cannot be established by a simple mapping in case of a binary uniform source, since the cardinality of this set is not a power of two.

This leads to a more practical scheme where a number (specifically a power of two) of sequences are chosen from the full balanced set. For the $n = 6$ example, we choose 16 of the 20 sequences from (1), again ignoring the underlining, and therefore information words of length 4 can be mapped one-to-one to these chosen balanced sequences, leading to a redundancy of $6-4 = 2$. Then in general, the best FF balanced code of length n consists of a subset of the full set of balanced sequences of length n, where the size of the subset is 2^{k^*} with

$$k^* = \left\lfloor \log_2 \binom{n}{n/2} \right\rfloor.$$

Encoding and decoding can be established through look-up tables, one-to-one mapping information sequences of length k^* to balanced sequences of length n. We refer to this scheme as FF*. Its redundancy is

$$r^* = n - k^* = n - \left\lfloor \log_2 \binom{n}{n/2} \right\rfloor. \tag{4}$$

Again using Stirling's formula gives the well-known result that this redundancy is $O(\log n)$. Also for practical FF methods not requiring large look-up tables, like Knuth's and its variants, the redundancy is only $O(\log n)$. In contrast, recall that the proposed VF method has a redundancy of $O(\sqrt{n})$. Hence, although its relative redundancy r/n goes to zero when n gets large, it has a poor redundancy compared to FF methods for long codes.

TABLE I
Redundancies of the full balanced set, the best FF scheme and the proposed VF scheme for small block lengths

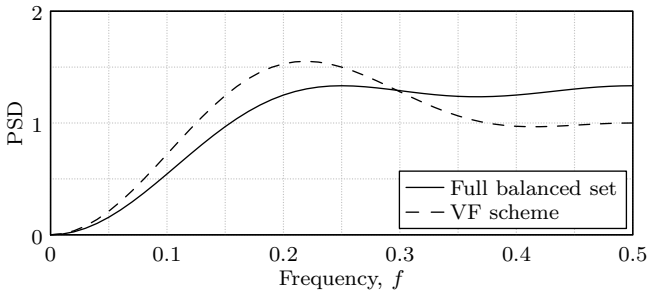
n	Full set	FF*	VF
4	1.42	2	1.50
6	1.68	2	1.88
8	1.87	2	2.19
10	2.02	3	2.46
12	2.15	3	2.71
14	2.26	3	2.93
16	2.35	3	3.14
18	2.43	3	3.34
20	2.50	3	3.52

On the other hand, for some very short codes ($n = 4, 6, 10, 12, 14$), the VF method is doing better than any FF method, as shown in Table I, where the Full, FF* and VF entries follow from (3), (4) and (2), respectively. We believe this to be an important observation, since for most applications where the balancing property is relevant, block lengths will be small, as argued next.

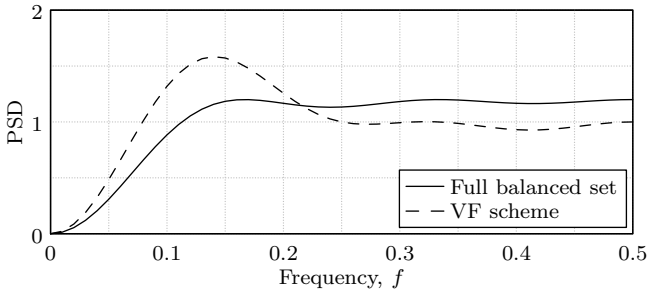
It is often desired to keep the running digital sum (RDS) $\sum_{i=1}^{h} x_i$ of a bipolar sequence close to zero. This can be established by the balancing property, since it leads to an RDS of exactly zero at $h = n$. However, along the way, the RDS may be large for long sequences, with the balanced length-n sequence $(1, \ldots, 1, -1, \ldots, -1)$ as an extreme example, leading to an RDS of $n/2$ at $h = n/2$. Note that there are $n + 1$ RDS values in principle, ranging from $-n/2$ to $n/2$. This span may not be acceptable for large values of n. A possible strategy to keep the RDS small in a long string is to parse it into short subsequences (of either fixed or variable length), and to map each subsequence to a balanced sequence of a small fixed length n. Cascading many of such short balanced sequences results in a long sequence in which the number of RDS levels, $n + 1$, is small in comparison with the total length (a large multiple of n). Since for small values of n the redundancy of our VF balancing scheme is smaller than for any FF balancing scheme of the same length, this gives an important advantage of the new method in the context of achieving limited RDS through balancing.

Furthermore, it is important to note that our new VF scheme has the benefit of being systematic, in the sense that the information appears as an unchanged integral part of the code sequence, which is not the case for the mentioned FF methods. In fact, the best-case scenario for an FF scheme to be systematic would require a redundancy of $r = k$, i.e., r is $O(n)$, since an information sequence of k 1's can only be balanced by the same number of -1's. So also in this respect the VF length scheme significantly improves on the prior art.

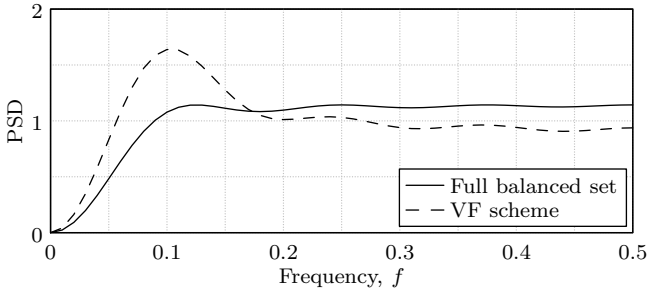
Fig. 2 shows the power spectral densities for the full balanced set and for our VF scheme. Although both codes consist of exactly the same sequences, as we have seen the probabilities for sequences in either code differs, resulting in the VF scheme having a narrower spectral null at dc.



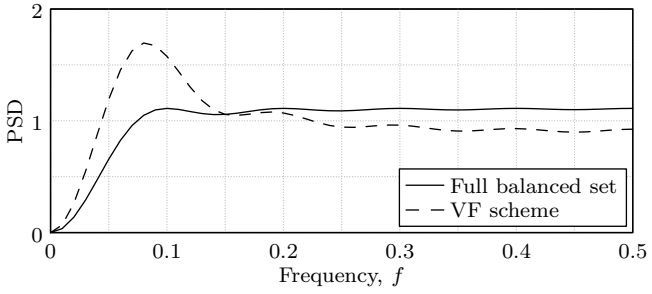
(a) $n = 4$



(b) $n = 6$



(c) $n = 8$



(d) $n = 10$

Fig. 2. Power spectral densities for balanced sequences of various lengths

Finally, as mentioned when introducing the encoding and decoding procedures, the VF scheme has hardly any time and space complexity, e.g., no look-up tables are needed. Such tables are commonly used for encoding and decoding the FF* codes, which have minimum redundancy among all FF schemes. Since their size grows exponentially with the block length n, the FF* scheme with look-up tables is infeasible for large block lengths. In such cases, the Knuth scheme or one of its many variants are good efficient alternatives in case of large values of n. However, if extremely low complexity is the goal, one may prefer the proposed VF scheme.

V. Conclusion

In this letter we presented a variable-to-fixed length scheme for balancing arbitrary binary information. The encoding/decoding is extremely simple and efficient, irrespective of the sequence length. Advantages include the coded sequences being systematic and the decoder having no processing or memory requirements, except for storing the sequence itself. The price to be paid for all these virtues is a higher redundancy than prior-art codes in case of large lengths: $O(\sqrt{n})$ versus $O(\log n)$. However, for very short codes, the redundancy of the proposed VF codes is smaller than the redundancy of any FF code. This is of particular importance for applications where the balancing property is imposed in order to keep the running digital sum low at all sequence stages.

Sequences close to the outer edge of the weight diagram with long runs of the same bit, such as $(-1, \ldots, -1, 1, \ldots, 1)$ or similar, although balanced, are best avoided in practice. These are also the sequences that contain the least information in our VF length scheme. Thus, as future work it would be interesting to investigate whether using a constrained code, such as a run-length limited code, as input could be beneficial with respect to the average redundancy.

Naturally the variable-length technique used will introduce insertion/deletion errors into the decoded information, when channel errors are considered, possibly resulting in undesirable error propagation. Further future work would include an analysis of the error propagation in this scenario, and methods to mitigate against this, be it modifications to the proposed scheme or introducing error correction coding.

Acknowledgment

This work is based on research supported in part by the National Research Foundation of South Africa (Grant Number 109543).

References

[1] K. A. Schouhamer Immink, *Codes for Mass Data Storage Systems*, 2nd ed. Eindhoven, The Netherlands: Shannon Foundation Publishers, 2004.
[2] B. Vasic and E. M. Kurtas, *Coding and Signal Processing for Magnetic Recording Systems*. Boca Raton, FL, USA: CRC Press, 2005.
[3] K. W. Cattermole, "Principles of digital line coding," *Int. J. Electron.*, vol. 55, no. 1, pp. 3–33, 1983.
[4] D. E. Knuth, "Efficient balanced codes," *IEEE Trans. Inf. Theory*, vol. 32, no. 1, pp. 51–53, Jan. 1986.
[5] S. Al-Bassam and B. Bose, "On balanced codes," *IEEE Trans. Inform. Theory*, vol. 36, no. 2, pp. 406–408, Mar. 1990.
[6] L. G. Tallini, R. M. Capocelli and B. Bose, "Design of some new efficient balanced codes," *IEEE Trans. Inform. Theory*, vol. 42, no. 3, pp. 790–802, May 1996.
[7] L. G. Tallini and B. Bose, "Balanced codes with parallel encoding and decoding," *IEEE Trans. Comput.*, vol. 48, no. 8, pp. 794–814 Aug. 1999.
[8] J. H. Weber and K. A. Schouhamer Immink, "Knuth's balanced codes revisited," *IEEE Trans. Inform. Theory*, vol. 56, no. 4, pp. 1673–1679, Apr. 2010.
[9] K. A. Schouhamer Immink and J. H. Weber, "Very efficient balanced codes," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 2, pp. 188–192, Feb. 2010.
[10] A. Steadman and I. J. Fair, "Variable-length constrained sequence codes," *IEEE Commun. Let.*, vol. 17, no. 1, pp. 139–142, Jan. 2013.
[11] K. A. Schouhamer Immink and L. Patrovics, "Performance assessment of DC-free multimode codes," *IEEE Trans. Commun.*, vol. 45, no. 3, pp. 293–299, Mar. 1997.
[12] W. Feller, *An Introduction to Probability Theory and its Applications*, vol. 1, 3rd ed. John Wiley & Sons, 1968.