EFM CODING: SQUEEZING THE LAST BITS

Kees A. Schouhamer Immink
Philips Research Laboratories
5656 AA Eindhoven, The Netherlands

Abstract—**Runlength-limited (RLL) codes have found widespread usage in optical and magnetic recording products. Specifically, the RLL codes EFM and its successor, EFMPlus, are used in the Compact Disc (CD) and the Digital Versatile Disc (DVD), respectively. EFMPlus offers a 6% increase in storage capacity with respect to EFM. The work reports on the feasibility and limits of EFM-like codes that offer an even larger capacity. To this end, we will provide an overview of the various limiting factors, such as runlength constraint, dc-content, and code complexity, and outline their relative effect on the code rate. In the second part of the article we will show how the performance predicted by the tenets of information theory can be realized in practice. A worked example of a code whose rate is 7.5% larger than EFMPlus, namely a rate 256/476, ($d = 2, k = 15$) code, showing a 13 dB attenuation at $f_b = 10^{-3}$, will be given to illustrate the theory.**

I. Introduction

INTRODUCED more than a decade ago, the Compact Disc (CD), has become a successful medium for the distribution and storage of digital information. In the Compact Disc system, the EFM code is used for transforming the digital audio bit stream into a sequence of binary symbols, called *channel bits,* which are suitable for storage on the disc [1]. Not only is the EFM coding scheme useful for the Compact Disc for which it has been designed, but it has also been extensively employed in a wide variety of digital audio players and home-storage products such as CD-ROM, CD-I, and MiniDisc.

In the international standard of the successor to the CD, the DVD, a new code, called *EFMPlus* [2], having properties similar to those of EFM, was adopted. The reason for adopting an 'EFM-like' code was that we can build on the expertise in critical electronics, such as data detection circuitry and servo mechanisms, gained since the introduction of the CD. The strategy of EFMPlus is much more refined and more powerful than the original EFM. The complexity of EFMPlus is also significantly greater. The rate of EFMPlus, i.e. the quotient of the numbers of bits entering and leaving the encoder, is 8:16. It is therefore capable of recording 6 % more user information than is possible with EFM whose rate is 8:17.

The introduction of EFMPlus immediately raises the engineering question as to whether a redesign of EFMPlus might offer another step for increasing the storage capacity. Or, in more general terms, how much room is there in the realms of coding theory? The aim of the study reported here is to determine the potential of candidate 'EFM-like' codes having a larger rate than EFMPlus without compromising the suppression of low-frequency components (lf-content). To this end, the combined effect of limiting factors such as lf-content and timing recovery will be computed. These parameters are very critical as the code's lf-content constitutes an added noise signal, which has a detrimental effect on the reliability of the servo systems and the timing recovery.

We will start with a brief description of EFM and EFMPlus in Section II. Thereafter, we briefly discuss the information theoretical background of our investigations in Section III. In the remaining sections, we will pay special attention to practical implementations of codes whose performance is close to that promised by the tenets of information theory.

II. EFM and EFMPlus

EFM and its successor EFMPlus are members of the family of *dc-free runlength-limited (RLL) codes.* The number of sequential like symbols in a (binary) sequence is known as a *runlength.* A runlength-limited sequence is a sequence of binary symbols characterized by two parameters, $T_{\min} = (d+1)$ and $T_{\max} = (k+1)$, which stipulate the minimum and maximum runlength, respectively, that may occur in the sequence. The parameter d controls intersymbol interference when the sequence is transmitted over a bandwidth-limited channel. The maximum runlength parameter k ensures adequate frequency of transitions for synchronization of the read clock. The reasons why EFM suppresses the low-frequency components are twofold. First, the servo systems for track following and focusing are controlled by low-frequency signals, so that low-frequency components of the information signal could interfere with the servo-systems. Second, low-frequency disturbances resulting from fingerprints on the disc can be filtered out without distorting the data signal itself.

Under EFM rules, the data bits are translated eight at a time into fourteen channel bits, with a minimum runlength parameter $d = 2$ and a maximum runlength parameter $k = 10$ channel bits (this means at least 2 and at most 10 successive 'zeros' between successive 'ones'). Three bits, called *merging bits,* are used to ensure that the runlength conditions continue to be satisfied when the codewords are cascaded. If we do this, we still retain a large measure of freedom in the choice of the merging bits. This freedom is used for minimizing the low-frequency content of the signal. The measure of the low-frequency content is the *running digital sum* (RDS), which is the difference between the totals of pit and land lengths accumulated from the beginning of the disc. The encoder now opts for the merging combination that makes the RDS at the end of the second codeword as close to zero as possible. EFM employs a single look-up table with a simple merging rule for concatenating the codewords. EFMPlus, however, is more complex as it uses four tables [2]. The power spectral density of EFMPlus has been computed with the aid of a computer program which simulated the encoder algorithm. Results are plotted in Figure 1.

III. Limiting Factors

After having described the state of the art in the previous section, we will now focus on the feasibility of alternative and preferably 'superior' EFM-like schemes. Three parameters play

Manuscript received June 13, 1997

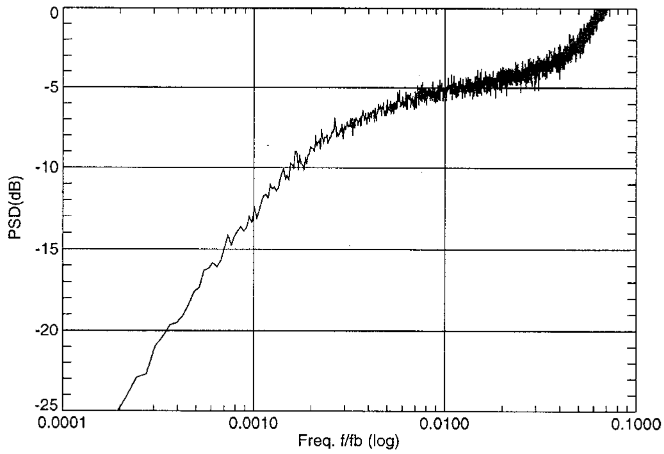

Fig. 1. Spectrum of EFMPlus.

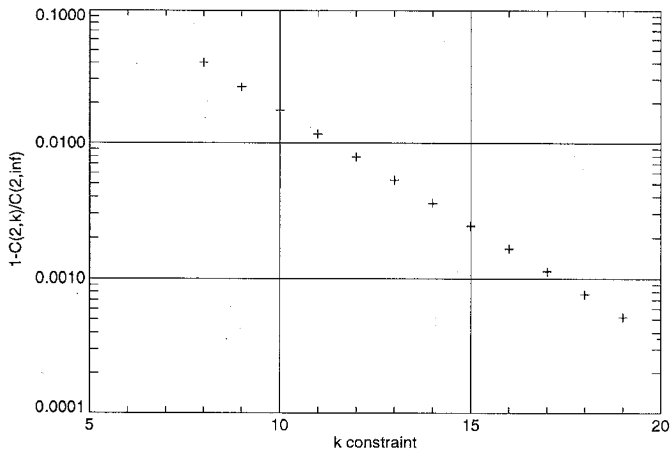

Fig. 2. The relative efficiency $1 - C(2,k)/C(2,\infty)$ versus k-constraint. $C(2,\infty) = 0.551463$.

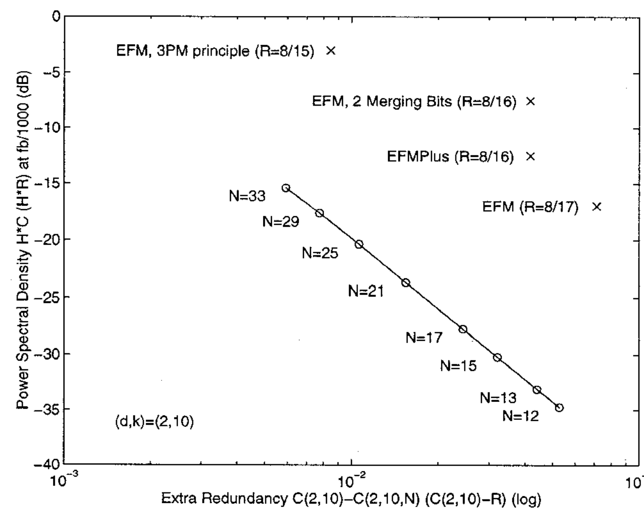

Fig. 3. Spectral density at $f_b = 10^{-3}$ of four ($d = 2, k = 10$) codes, including EFM and EFMPlus. The series of points connected by a straight line implies the upper bound to the lf-suppression. Taken from Braun and Janssen [3].

a primary role in this feasibility study, namely

- maximum runlength or k-constraint,
- low-frequency content,
- encoder and decoder complexity.

In the next subsections, we will compute or estimate the effects of the above parameters on the code's rate and performance.

A. *Maximum runlength*

The maximum runlength constraint, k, is imposed to restrict the maximum time between two consecutive transitions in the recorded signal. The $k = 10$ constraint, used in EFM and EFMPlus, could be relaxed to improve the code rate. The maximum rate of any code given the imposed d and k constraints, called *capacity* and denoted by $C(d,k)$, is [1]

$$C(d,k) = \log_2 \lambda, \quad (1)$$

where λ is the largest real root of the characteristic equation

$$z^{k+2} - z^{k+1} - z^{k-d+1} + 1 = 0. \quad (2)$$

Figure 2 shows the effect of the k-constraint on the capacity $C(2,k)$. Clearly, an increase of $k = 10$ to $k = 15$ improves the capacity by approximately 1 percent. This saving in rate offers a small but serviceable increase in storage capacity as compared with its predecessors EFM and EFMPlus. A further increase of k can bring at most 0.2 percent increase in capacity. This diminishing return is not worth the effort.

B. *Low-frequency contents*

The specification of the low-frequency content of a code is not a simple matter. The spectral density curves plotted in Figure 1 reveal that at the (very) low-frequency end the curves are parabolic, i.e. the PSD function can be approximated by

$$PSD(f_b) \approx A f_b^2, f_b << 1, \quad (3)$$

where the constant A is independent of the source bit frequency f_b. In the literature, the lf-content is usually measured in terms of the *cut-off frequency*, ω_0, which measures the width of spectral notch. Here, in contrast, we opt for measuring the spectral density at a relatively low frequency, namely at $f_b = 10^{-3}$, where eq. (3) is valid for the range of parameters of interest. It has the advantage relative to the ω_0 criterion that its engineering significance is immediately clear and that the spectral density at other (low) frequencies can easily be computed by invoking eq. (3). A disadvantage is that it can only be assessed with the aid of computer simulations.

The relationship between the low-frequency properties of (d,k) sequences and the channel capacity was studied by Braun and Janssen [3]. They estimated upper bounds to the performance of any code. Figure 3 surveys the key findings of their study. We observe the spectral density at $f_b = 10^{-3}$ of four ($d = 2, k = 10$) codes, including EFM and EFMPlus, and of a series of points connected by a straight line, which imply the upper bound to the lf-suppression. Code implementations for the given redundancy and lf-suppression below this line are not possible. A number of interesting conclusions can be drawn

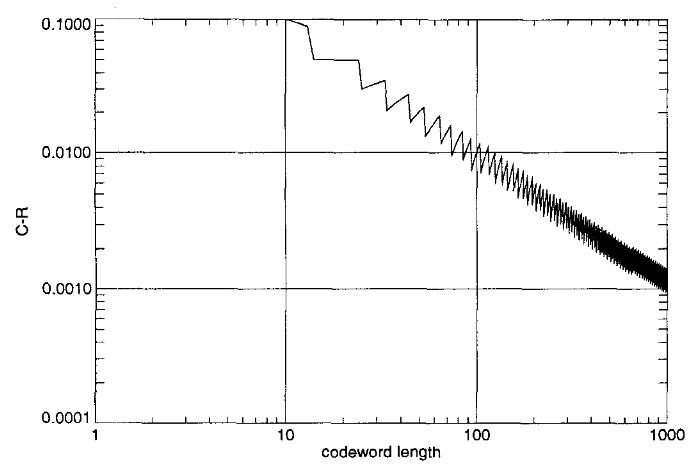

Fig. 4. Redundancy $C-R$ versus codeword length n. The runlength parameters are $d=2$ and $k=15$. The channel capacity is $C(2,15)=0.5501$.

from this figure. For the given lf-content, the rate of EFMPlus is approximately 9-10% above the theoretical bound. Measuring along the vertical axis, we conclude that, for the given rate 8/16, an improvement of the lf-content by at most 20 dB is possible. Extra suppression of lf-content is not the primary aim of our study. In the next section, we will report on our attempts to improve the rate of the code without sacrificing the lf-content.

IV. How long is long enough?

The practicable limits on coding performance are not set by the theoretical bound described above but arise from the difficulty of building the encoders and decoders. Codes with a codeword length shorter than, say, 20 cannot provide the degree of freedom required for an efficient and flexible trade-off between the various relevant parameters. Codes having (very) long codewords, say 200-1000, offer far greater flexibility. Encoding and decoding such huge codes with the aid of look-up tables is obviously out of the question. There are, however, practical techniques at our disposal for the translation part of such large codes (see next section).

Block codes, such as EFM, seem to be ideally suited for usage with long words. In this format user information is translated into a codeword which complies with the prescribed d and k constraints, and the codewords start and end with at most $k-d$ consecutive 'zeros'. A bridge of d 'merging' bits suffices to guarantee that both the d and k constraints can be preserved during the catenation process of two consecutive codewords.

The size and efficiency of the codes can be found by computing the number of constrained sequences as a function of the codeword length. As an example, we have computed $C(d=2, k=15)-R$ as a function of the codeword length n. Results are shown in Figure 4. Clearly, a codeword length of approximately $n=256$ suffices to approach capacity to within 0.2-0.5%. The encoding and decoding massiveness can be solved by using a technique called *enumeration*. The enumerative coding technique makes it possible to translate source words into codewords and vice versa by invoking an algorithmic procedure rather than performing the translation with a look-up table (see next section). The second drawback of the use of long codewords is the risk of extreme error propagation. Single channel bit errors may result in error propagation which could corrupt the entire data in the decoded word, and, of course, the longer the codeword the greater the number of data symbols affected. With a new technique massiv error propagation is avoided by reversing the conventional hierarchy of the error control code and the constrained code [4]. Before dealing with the techniques for suppressing the low-frequency components of RLL codes, we will describe in the next section how encoding and decoding of these very long codes can be done in a practical manner.

V. Enumerative coding

The encoding and decoding of large codes is impossible, if we could not employ enumerative coding techniques. Enumerative coding makes it possible to translate source words into codewords and vice versa by invoking an algorithm rather than performing the translation with a look-up table. In [5], a method is described that requires storage capacity of a bank of n integer coefficients. The algorithm itself is conceptually very simple. The operations required for encoding and decoding are simple (binary) additions, subtractions, and comparisons, which can easily be implemented. A feature of enumerative coding is that encoding and decoding can be done with the same hardware. This is particularly attractive for recorders, which are usually equipped with both an encoder and decoder.

Denote by $N^0(i)$ the number of dk-constrained sequences of a length i whose first element equals '1'. We define the quantity $a_j(\boldsymbol{x})$ as the length of the trailing zero-run of the sub-vector $(x_1,\ldots,x_{j-1})$, if it is not the all-zero sequence. Or,

$$a_j(\boldsymbol{x}) = \begin{cases} \min\{(j-i-1): & \\ 1 \le i < j, x_i = 1\} & \text{if } j>1 \text{ and} \\ & (x_1,\ldots,x_{j-1}) \ne (0,\ldots,0) \\ d & \text{otherwise.} \end{cases}$$

The decoding operation translates a (dk) sequence $\boldsymbol{x}$ of length n into an integer $I(\boldsymbol{x})$ by invoking

$$I(\boldsymbol{x}) = \sum_{j=1}^{n} \delta_j(\boldsymbol{x}) N^0(n-j+1),$$

where

$$\delta_j(\boldsymbol{x}) = \begin{cases} 1 & \text{if } x_j = 0 \text{ and } a_j(\boldsymbol{x}) \ge d; \\ 0 & \text{otherwise.} \end{cases}$$

In words, the translation is done by summing the weight of each 'zero' of $\boldsymbol{x}$ that is either contained in the leading zero-run or if it is preceded by at least d 'zeros'.

The encoding operation, i.e. given the source word I find the corresponding $\boldsymbol{x}$, is described by the following program:

$\hat{I} := I$, $a := d$;
for $j=1$ **to** n **do**
  if $\hat{I} \ge N^0(n-j+1)$ **and** $a \ge d$
    then $x_j := 0$, $\hat{I} := \hat{I} - N^0(n-j+1)$
    else if $a < d$
      then $x_j := 0$

else $x_j := 1, a := -1;$
$a := a + 1;$
end for

VI. DC-CONTROL

Guided scrambling (randomizing) introduced by Fair *et al.* [6] is a promising and flexible technique for suppressing the low-frequency components (dc-control). Guided scrambling is a member of a class of coding schemes, called *multi-mode* code. In multi-mode codes, each source word x can be represented by a member of a (selection) set consisting of L, $L > 1$, codewords. The encoder opts for transmitting the specific codeword that minimizes the low-frequency spectral contents of the encoded sequence. Of course, the selection mechanism can also be based on other relevant properties. Two key elements should be chosen judiciously: (a) the mapping between the source words and their corresponding selection sets, and (b) the criterion used to select the "best" word. The spectral performance of the code greatly depends on both issues.

Examples of multi-mode codes are codes using the guided scrambling algorithm presented by Fair *et al.* [6] and the randomization using a RS code described by Kunisa *et al.* [7].

We propose to implement the dc-control as an intrinsic part of the enumerative coding scheme. Essentially, r redundant bits, which are a part of the input of the channel encoder, can be freely chosen to optimize the lf-content of the codeword transmitted. In this format, the channel encoder input comprises the r dc-control bits plus the m user bits. The $r + m$ input word is translated by the (d, k) encoder into the n-bit codeword. The selection set of size $L = 2^r$ is found with the aid of the following procedure.

1. In the first step, called *augmenting*, the source word x is preceded by all the possible binary sequences of length r to produce the set $\mathcal{B}_x = \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_L\}$. Hence:
$$\boldsymbol{b}_1 = (0, 0, \ldots, 0, x_1, \ldots, x_m), \ \ldots \ ,$$
$$\boldsymbol{b}_L = (1, 1, \ldots, 1, x_1, \ldots, x_m).$$
2. The selection set $\mathcal{C}_x = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_L\}$ is obtained by translating all vectors in $\mathcal{B}_x$ using the rate $(r + m)/n$, (d, k) code. The (d, k) code translates each vector $\boldsymbol{b} = (b_1, \ldots, b_{m+r}) \in \mathcal{B}_x$ into $\boldsymbol{c} = (c_1, \ldots, c_n) = f(\boldsymbol{b}) \in \mathcal{C}_x$ using the code book in force.
3. The "best" codeword in $\mathcal{C}_x$ is selected for transmission.
4. The inverse operation $\boldsymbol{b} = f^{-1}(\boldsymbol{c})$ is executed at the receiver's site. The source word is simply found after deleting the first r redundant bits.

In our context, the selection set can be extended even further. First, as in conventional EFM, the merging bits can be used to control the lf-content. Second, the number of words available is usually not an exact power of two. The excess codewords, as in EFMPlus, can be used as alternative channel representations.

VII. SIMULATION RESULTS

The dc-control and the enumerative coding, as discussed in the previous sections, were realized in software. In the results shown the fully fledged dc-control was implemented. Results of simulations in which the redundancy r/n is held fixed are shown in Figure 5. The curves show that the lf-content improves with

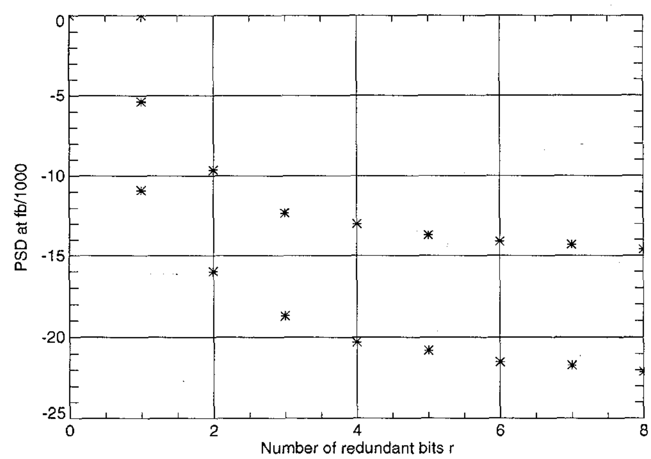

Fig. 5. Power spectral density at 10^{-3} of the data bit frequency of codes with r redundant bits for dc-control. The maximum runlength is $k = 10$. The upper curve shows results for a fixed redundancy $r/n = 0.01$, and the lower curve shows results for a redundancy $r/n = 0.02$.

TABLE I
REDUNDANCY AND SIZE OF SELECTION SET OF CODES WITH PSD($f_b = 10^{-3}$) = -14 DB.

Redundancy r/n	2^r
0.02	4
0.015	8
0.01	128

increasing number of redundant bits r (and thus an exponential increase in the size of the selection sets). The gains in performance diminish as the number of redundant bits increases. Clearly, increasing the number of redundant bits above say $r = 5$ is not worth the effort. The maximum runlength of the RLL codes used was fixed at $k = 10$. When the maximum runlength is set to $k = 15$ we noted that the performance worsened slightly (1 dB). From the results shown in the diagram we deduce that we have a number of options for designing codes with the same lf-content as EFMPlus, i.e. -14 dB at $f_b = 10^{-3}$ (see Figure 1). These design options are listed in Table I. The table offers a good overview of the engineering trade-offs. Of course, many other intermediate values than those shown in the table are possible.

VIII. A NEW EFM-LIKE CODE

Following the preliminary computations described in the previous section, we worked a few dedicated examples of code design. The spectral performance of such an example of a worked design is plotted in Figure 6. Basically, the code is a rate 261/476, $(d = 2, k = 15)$ code in which 5 bits are used for dc-control (selection set has size $2^5 = 32$) and the remaining 256 are source input. The net code rate is therefore 256/476 = 0.5378, which is 7.5 % larger than the rate of EFMPlus. The spectral density at $f_b = 10^{-3}$ is a comfortable -13 dB. The theory developed in the previous sections makes it possible to quantify the various elements of any possible further savings in the code

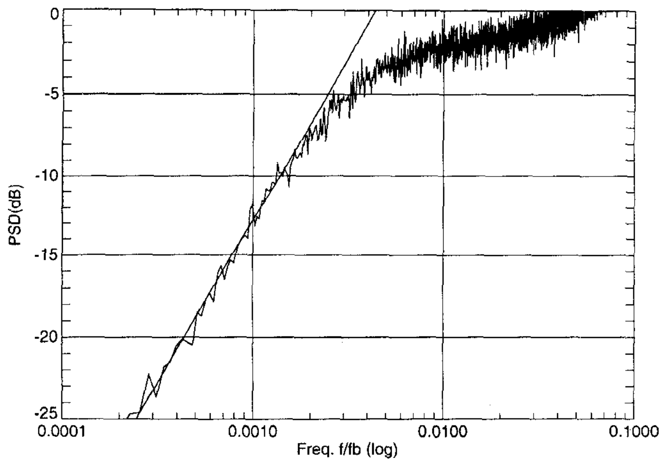

Fig. 6. Power density function of rate (256+5)/476, ($d = 2, k = 15$) code. Five out of the 261 bits are used for dc-control. The straight line drawn is found after a least mean squares fit of the experimental data at the low-frequency end.

rate. The basic code of rate R = 261/476, ($d = 2, k = 15$) is extremely efficient, as can be concluded from $C(2,15) - R$ = 0.0018. A doubling of the codeword length (and a more than proportional increase of the hardware required) can improve the rate by approximately 0.001 (see Figure 4). A relaxation of the k constraint can increase the rate with at most 0.0025 (see Figure 2). The dc-control part of the code takes only 5/256 ≈ 1.95% overhead.

IX. Conclusions

We have provided a survey of the state of the art of EFM-like codes and their properties. The paper has established the extent of the potential for improving the rate of an EFM-like code whilst not compromising its lf-content. It has been shown that infusion of two enabling technologies, namely enumerative coding and guided scrambling, allows very large dc-free RLL codes to be designed whose performance is very close to the one predicted by the tenets of information theory. The rates of the new codes are only 1-2% below the theoretical bound, which implies a 7-8% gain in capacity with respect to the industry standard EFMPlus code. The large codes described are very flexible, because, during the design stage, both the lf-content and the rate of the code can easily be traded. A detailed example of a code whose rate is 7.5% larger than EFMPlus, namely a rate 256/476 (=0.5378), ($d = 2, k = 15$) code, shows a 13 dB attenuation at $f_b = 10^{-3}$. Massive error propagation is solved by reversing the order of application of the modulation and error correcting Reed-Solomon codes.

References

[1] K.A.S. Immink, *Coding Techniques for Digital Recorders,* Prentice-Hall International (UK) Ltd., Englewood Cliffs, New Jersey, 1991.

[2] K.A.S. Immink, 'The Digital Versatile Disc (DVD): System Requirements and Channel Coding', *SMPTE Journal,* vol. 105, pp. 483-489, no. 8, Aug. 1996.

[3] V. Braun and A.J.E.M. Janssen, 'On the Low-frequency Suppression Performance of DC-free Runlength-limited Modulation Codes', *IEEE Trans. Consumer Electr.,* vol. CE-42, pp. 939-945, no. 4, Nov. 1996.

[4] K.A.S. Immink, 'Code configuration for avoiding error propagation', *Electronics Letters,* vol. 32, No. 24, pp. 2191-2192, Nov. 1996.

[5] L. Patrovics and K.A.S. Immink, 'Encoding of $dklr$-sequences using one weight set', *IEEE Trans. Inform. Theory,* vol. IT-42, no. 5, pp. 1553-1554, Sept. 1996.

[6] I.J. Fair, W.D. Gover, W.A. Krzymien, and R.I. MacDonald, 'Guided Scrambling: A New Line Coding Technique for High Bit Rate Fiber Optic Transmission Systems', *IEEE Trans. Commun.,* vol. COM-39, no. 2, pp. 289-297, Feb. 1991.

[7] A. Kunisa, S. Takahashi, and N. Itoh, 'Digital Modulation Method for Recordable Digital Video Disc', *IEEE Trans. Consumer Electr.,* vol. 42, pp. 820-825, Aug. 1996.

Kees A. Schouhamer Immink received the M.S. and Ph.D degrees from the Eindhoven University of Technology. Immink joined the Philips' Research Labs, Eindhoven, in 1968, and currently holds the position of Research Fellow. In addition, he is guest professor at the Institute of Experimental Mathematics, Essen University, Germany. He contributed to the design and development of a variety of digital audio and video recorders such as the Compact Disc, Compact Disc Video, R-DAT, DCC, and DVD. Immink holds 35 US patents and has written books and numerous papers in the field of coding techniques for digital recorders. He is a Fellow of the AES, SMPTE, IEE, IEEE, and a member of the Royal Netherlands Academy of Sciences. His pioneering contributions to the digital audio and video revolution have earned him numerous awards such as the AES Silver Medal, the IEE J.J. Thomson Medal, the SMPTE Poniatoff Gold Medal for Technical Excellence, and the IEEE Masaru Ibuka Consumer Electronics Award. He is the chairman of the IEEE Benelux Chapter on Consumer Electronics, a governor of the AES and the IEEE Information Theory Society.