



Run Length Limited Error Control Codes Derived from Reed Solomon Codes

Peter Farkaš^{1,2} · Martin Rakús¹

Accepted: 2 December 2023 / Published online: 20 December 2023
© The Author(s) 2023

Abstract

Reed Solomon codes were standardized for numerous wireless communication systems. Most practical Reed Solomon codes belong to non-binary linear block codes defined over finite fields with characteristic two. Each linear block code contains one codeword composed of all zeros. The concatenation of this and also other codewords can lead to long, theoretically even infinite, runs of equal symbols. Such long runs do not support synchronization in wireless receivers and therefore are unwanted. In this paper it is shown that extended and some appropriately shortened Reed Solomon codes constructed over finite fields with characteristic two can be transformed into Run Length Limited Reed Solomon codes. The presented method, if applicable, allows for doing it without inserting additional redundancy. Another advantage is that after the transformation, if some round conditions are fulfilled, the decoding does not have to be rebuilt.

Keywords Reed Solomon codes · Run length limited · Modifier · Redundancy · Extended · Shortened · Parity check matrix

JEL Classification E.4 · H.4.3 · C.2.5

Mathematics Subject Classification 94A05 · 94A99 · 94B60

Peter Farkaš and Martin Rakús has equally contributed to this work.

✉ Martin Rakús
martin.rakus@stuba.sk

Peter Farkaš
p.farkas@ieee.org

¹ Institute of Multimedia ICT, Faculty of Electrical Engineering and Information Technology, Ilkovičova 3, 812 19 Bratislava, Slovakia

² Institute of Applied Informatics, Faculty of Informatics Pan-European University, Tematínska 10, 851 05 Bratislava, Slovakia

1 Introduction

Error control codes (ECCs) and Run Length Limited (RLL) codes are used for two different goals in communications systems. ECCs are used to eliminate or significantly suppress the negative influence of errors on transmitted or stored information [1, 2]. This goal is achieved by adding some redundancy to the coded sequence of transmitted information expressed usually by bits. The coded sequence is denoted as a codeword in the case of convolutional codes or a chain of codewords in the case of block codes. The RLL codes are used in order to exclude long runs of equal symbols from the coded sequence [3]. These long runs can cause the signals which are transmitting the information to not have the required level of synchronization supportive properties. The RLL codes also have to add some redundancy to the coded sequence which is transmitted in order to implement the required RLL properties in it.

In many wireless communications systems both ECCs and RLL codes are used together. As was already noted, both contribute to the increase of redundancy which is introduced into the coded sequence. This leads to a natural question if it is necessary to add the redundancy twice and if it would be not possible to achieve error control properties and RLL properties by only a single addition of redundancy. Another problem arises when both mentioned types of codes are used in the same transmission or storage system, namely how to position the codes in cascade. In other words, which code should be the outer code and which the inner code. The reason is that in general the RLL codes are constructed with the assumption that their encoder and decoder communicate via an error free channel. The approximately error free channel is obtained from the real channel with errors and other imparities using the ECCs. Taking into account these facts one can conclude that the ECC has to be an inner code and the RLL the outer code. However the ECC is not constructed with the goal to have RLL properties. This would suggest that the opposite concatenation should be used, namely the RLL code should be applied as the inner code and the ECC as the outer one.

Fortunately there exists a class of codes which can help to overcome the dilemma with ordering the ECCs and RLL codes. The combined RLL-ECCs codes allow it. One approach to obtain such codes consists of transforming an original ECC into an RLL-ECC using modifiers. It is illustrated in Fig. 1. It was used in [4] in order to construct RLL-ECCs from the standard 5 G LDPC codes. Recently it was also shown in [5] and [6] respectively, that standard CCSDS and convolutional codes could be transformed into RLL-ECCs.

2 Basic Related Backgrounds on RS Codes

Reed Solomon (RS) codes, which have been known for more than eight decades, are actually used in many wireless communications systems. For example they are standardized in DVB standards and in CCSDS specifications. Huge knowledge regarding them has been accumulated since their discovery. Never-the-less, they are still a focus for researchers and practitioners [7–14].

There are numerous approaches regarding how to define the Reed Solomon codes. RS codes belong to linear block codes and can be described as cyclic codes, which belong to a subset of non-binary BCH codes. They could be defined over finite fields $GF(q)$ and also over infinite fields with real or complex numbers as elements. However, the most

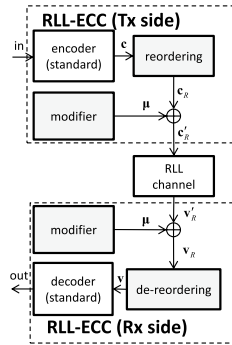


Fig. 1 On the transmitting side (Tx side) the \mathbf{c} , \mathbf{c}_R , and \mathbf{c}'_R denote a codeword from the original binary code, the codeword with reordered coordinates, modifier and the codeword after summing the previous one with the modifier respectively. On the Receiving side (Rx side) the \mathbf{v}'_R , $\boldsymbol{\mu}$, \mathbf{v}_R and \mathbf{v} denote the reordered received vector, modifier, reordered received vector after summing it with the modifier and the received respectively. The summing of vectors is defined over binary finite field $GF(2)$

useful RS codes are defined over finite fields with characteristic two. In other words, these most popular RS codes are defined over $GF(q)$, where $q = 2^m$; $m \in \mathcal{N}^+$, $m > 1$. The reason is that at the present time communications and memory systems are digital and a binary expression of symbols is the most natural, convenient and practical in such systems.

Further, we will suppose that the infinite family of original RS codes is defined as a family of linear cyclic codes with the following generator polynomial:

$$g(x) = (x + \alpha^j)(x + \alpha^{j+1}) \dots (x + \alpha^{j+2t-1}) \tag{1}$$

where α is a primitive element of the finite field $GF(2^m)$; $m \in \{2, 3, 4, \dots\}$ and $j \in \{0, 1, 0 \dots, q - 2\}$. In practice, often $j \in \{0, 1\}$ is chosen.

Another useful description of such RS codes is via their parity check matrix:

$$\mathbf{H}_1 = \begin{bmatrix} 1 & \alpha^j & \alpha^{2j} & \alpha^{3j} & \dots & \alpha^{(q-2)j} \\ 1 & \alpha^{j+1} & \alpha^{2(j+1)} & \alpha^{3(j+1)} & \dots & \alpha^{(q-2)(j+1)} \\ 1 & \alpha^{j+2} & \alpha^{2(j+2)} & \alpha^{3(j+2)} & \dots & \alpha^{(q-2)(j+2)} \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & \alpha^{j+2t-1} & \alpha^{2(j+2t-1)} & \alpha^{3(j+2t-1)} & \dots & \alpha^{(q-2)(j+2t-1)} \end{bmatrix} \tag{2}$$

For example, after choosing $j = 0$ we get:

$$\mathbf{H}_2 = \begin{bmatrix} 1 & \alpha^0 & \alpha^0 & \alpha^0 & \dots & \alpha^0 \\ 1 & \alpha^1 & \alpha^2 & \alpha^3 & \dots & \alpha^{(q-2)} \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & \dots & \alpha^{2(q-2)} \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & \alpha^{2t-1} & \alpha^{2(2t-1)} & \alpha^{3(2t-1)} & \dots & \alpha^{(q-2)(2t-1)} \end{bmatrix} \tag{3}$$

From the parity check matrices given in (2) or (3), the basic parameters of the RS codes could be derived, namely the codeword length n , (which is equal to the number of columns in them), number of information symbols k in each codeword (because the number of rows in them is equal to $n - k$) and also the code distance d_m , which is for the original RS codes equal to the number of rows plus one in (2) or (3). Usually these parameters for linear

block codes are given as a triple $[n, k, d_m]$ in literature. Different tables containing bounds on these parameters of the best known codes exist [15, 16]. The original Reed Solomon codes belong to the maximum distance separable codes (MDS) and therefore have maximal possible code distance $d_m = n - k + 1$ for the given n and k .

In [17] it was shown that the RS codes defined over $GF(2^\chi)$, where $\chi \geq 3$ is an odd integer, with the following parity check matrix:

$$\mathbf{H}_3 = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & \alpha^1 & \alpha^2 & \dots & \alpha^{(2\chi-2)} & 0 & 1 & 0 & 0 & 0 \\ 1 & \alpha^2 & \alpha^4 & \dots & \alpha^{2(2\chi-2)} & 0 & 0 & 1 & 0 & 0 \\ 1 & \alpha^3 & \alpha^6 & \dots & \alpha^{3(2\chi-2)} & 0 & 0 & 0 & 1 & 0 \\ 1 & \alpha^4 & \alpha^8 & \dots & \alpha^{4(2\chi-2)} & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

are five times extended original RS codes. The obtained RS codes are $[q + 4, q - 1, 5]$ - codes. These codes are not MDS codes, but they are so-called almost MDS codes. They are also reaching the upper bound on d_m for the given n and k [15, 16]. However the once extended RS codes are MDS codes [18]. In case that $j = 1$, they are defined by the following parity check matrix:

$$\mathbf{H}_4 = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 & 1 \\ 1 & \alpha^1 & \alpha^2 & \dots & \alpha^{(q-2)} & 0 \\ 1 & \alpha^2 & \alpha^4 & \dots & \alpha^{2(q-2)} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha^{(2l-1)} & \alpha^{2(2l-1)} & \dots & \alpha^{(2l-1)(q-2)} & 0 \end{bmatrix} \tag{5}$$

The parity check matrices of RS codes defined over $GF(2^m)$ could be expressed over $GF(2)$ using the so-called binary image. Each element from $GF(2^m)$ can be expressed not only as a power of a primitive element, but also as a binary vector or power of the companion matrix \mathbf{A} . This matrix \mathbf{A} can be obtained from the primitive polynomial $p(x)$ which is used for obtaining the extended field $GF(2^m)$ from $GF(2)$:

$$p(x) = x^m + p_{m-1}x^{m-1} + \dots + p_1x + p_0 \tag{6}$$

where $p_i \in GF(2)$. Then the companion matrix \mathbf{A} can be expressed as:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & \dots & 0 & p_0 \\ 1 & 0 & \dots & 0 & p_1 \\ 0 & 1 & \dots & 0 & p_2 \\ \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & p_{m-1} \end{bmatrix} \tag{7}$$

For each RS code defined over $GF(2^m)$ the binary image of its parity check matrix can be obtained by substituting each element from $GF(2^m)$ in it by its companion matrix.

EXAMPLE

The one-time extended RS code $[8, 6, 3]$ defined in $GF(8)$ generated by the primitive polynomial $p(x) = x^3 + x + 1$ has the following parity check matrix:

$$\mathbf{H}_{[8,6,3]} = \begin{bmatrix} \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^1 & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 & \alpha^0 \end{bmatrix} \tag{8}$$

The element of $GF(8)$ expressed as binary vectors, powers of primitive element α and as powers of the accompanying matrix respectively are:

$$000 \leftrightarrow \alpha^{-\infty} \leftrightarrow \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (9)$$

$$001 \leftrightarrow \alpha^0 \leftrightarrow \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (10)$$

$$010 \leftrightarrow \alpha^1 \leftrightarrow \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (11)$$

$$100 \leftrightarrow \alpha^2 \leftrightarrow \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \quad (12)$$

$$011 \leftrightarrow \alpha^3 \leftrightarrow \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \quad (13)$$

$$110 \leftrightarrow \alpha^4 \leftrightarrow \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad (14)$$

$$111 \leftrightarrow \alpha^5 \leftrightarrow \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} \quad (15)$$

$$101 \leftrightarrow \alpha^6 \leftrightarrow \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \quad (16)$$

Substituting the accompanying matrices (9)-(16) into (8) the following binary image of (8) could be obtained:

$$\mathbf{H}_{[8,6,3]} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (17)$$

The binary image of the codeword is then defined as:

$$\mathbf{c} \cdot \mathbf{H}^T = \mathbf{0} \quad (18)$$

3 RLL-ECCs Obtained from Selected Reed Solomon Codes

In this section a technique will be proposed for obtaining RLL-ECCs from selected RS codes. It exploits their binary images and the specific property of rows in parity check matrices used for their construction. This property was proven in [19] as a Lemma 1. For the convenience of the reader here a simple rewording of it follows. Parity check matrix property necessary for RLL-ECC construction (*Prop.*):

Let the linear binary block code be defined by parity check matrix \mathbf{H} . If in any row from \mathbf{H} there is a set S of ones with even cardinality then adding to each codeword $\mathbf{c} \in C$ a binary vector $\boldsymbol{\mu}$ which has an odd number of ones in positions corresponding to the set S will transform the code C into a coset code C' . For each codeword $\mathbf{c}' \in C'$ it is valid that it does not have all symbols in the positions corresponding to set S identical. In other words not equal to all ones or all zeros.

This property, if found in more rows, can be used for each such row in case that the sets of ones in these rows are disjoint.

Reordering of coordinates in codewords from $\mathbf{c} \in C$ can help to get even better RLL properties than using only the addition of the modifier to the original codewords. Further, we will denote \mathbf{c}_R and $\boldsymbol{\mu}$ the reordered codewords $\mathbf{c} \in C$ and modifier respectively.

In practice, the above-described property could be used as follows. After finding as many sets with even numbers of ones in disjoint positions in different rows as possible, and after appropriate reordering, modifier $\boldsymbol{\mu}$ could be fixed, which causes the run lengths of equal symbols in codewords to be minimal. Then k information bits which have to be protected can be encoded into codeword $\mathbf{c} \in C$, which then has to be reordered into codeword \mathbf{c}_R . Before this codeword is stored or transmitted via constrained channel, the modifier $\boldsymbol{\mu}$ has to be added to it:

$$\mathbf{c}'_R = \mathbf{c}_R + \boldsymbol{\mu} \quad (19)$$

The method of obtaining an RLL-ECC using the reordering and addition of modifier is illustrated from a high-level perspective in Fig. 1. On the receiving side the influence of the modifier $\boldsymbol{\mu}$ is canceled by the addition of the same modifier in a synchronized manner to the received vector \mathbf{v}'_R .

The main problem which has to be overcome when designing RLL-ECC codes from RS codes is that the RS codes are non-binary. Therefore the *Prop.*, which is explicitly valid only for binary linear block codes, cannot be used directly. Therefore the first step which has to be done before the method based on using a modifier could be applied is to get binary representation of the parity check matrices and of RS code codewords. This could be achieved for RS codes defined over finite fields with characteristic two. The analysis will be done in the following sub-sections for selected RS codes.

(a) One-time extended Reed Solomon codes

The parity check matrix of one-time extended RS codes (5) has the following binary image:

$$\mathbf{H}_b = \begin{bmatrix} \mathbf{A}^0 & \mathbf{A}^0 & & \mathbf{A}^0 & & \dots & \mathbf{A}^0 & & \mathbf{A}^0 \\ \mathbf{A}^0 & \mathbf{A}^1 & & \mathbf{A}^2 & & \dots & \mathbf{A}^{(q-2)(j+1)} & & \mathbf{0} \\ \vdots & \vdots & & \vdots & & \vdots & \vdots & & \vdots \\ \mathbf{A}^0 & \mathbf{A}^{(j+2t-1)} & & \mathbf{A}^{2(j+2t-1)} & & \dots & \mathbf{A}^{(q-2)(j+2t-1)} & & \mathbf{0} \end{bmatrix} \tag{20}$$

The binary image in (20) contains $m \times 2t$ binary rows and $m \times q$ binary columns. The matrix (20) is composed of $2t$ rows and $q = 2^m$ columns of submatrices with dimensions $m \times m$. These rows and columns of submatrices will be denoted as macro-rows and macro-columns respectively. It is obvious that the one-time extended RS code in which $j = 0$ has a binary image expressed by a matrix having in the top macro row \mathbf{h}_j only submatrices \mathbf{A}^0 with dimensions $m \times m$.

$$\mathbf{h}_j = [\mathbf{A}^0 \ \mathbf{A}^0 \ \dots \ \mathbf{A}^0] \tag{21}$$

The binary image of \mathbf{h}_j in more detail is expressed in (22).

$$\mathbf{h}_j = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 & 1 & 0 & \dots & 0 & 0 & & 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 & 0 & 1 & \dots & 0 & 0 & & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \dots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 & 0 & 0 & \dots & 1 & 0 & & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & \dots & 0 & 1 & 0 & 0 & \dots & 0 & 1 & & 0 & 0 & \dots & 0 & 1 \end{bmatrix} \tag{22}$$

Observing (22) it is obvious that in the case of a one-time extended RS code the number of ones in each row of \mathbf{h}_j will be even:

$$w_j(S) = 2^m \tag{23}$$

By reordering (permutation) the columns in (22):

$$\mathbf{h}_{IR} = \begin{bmatrix} 1 & 1 & \dots & 1 & 0 & 0 & \dots & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 1 & 1 & \dots & 1 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & \dots & 1 & 1 & \dots & 1 \end{bmatrix} \tag{24}$$

Let's denote E , the number of type S intervals fulfilling Prop. in (24):

$$E = m \tag{25}$$

For each of these intervals the cardinality is equal to the number of submatrices \mathbf{A}^0 in \mathbf{h}_j , which is identical to $w_j(S)$ given by (23) or simply:

$$|S| = 2^m \tag{26}$$

Within these intervals there is the possibility to flip an odd number of binary symbols by adding the modifier in order to ensure that not all symbols in that interval will be the same. The union of all these intervals contains all column indexes of macro-rows given by (22) or (24) and also of the corresponding parity check matrices. As a consequence, the maximal run lengths in RLL-ECCs obtained by the presented method from one-time extended RS codes over $GF(2^m)$ will be given by:

$$RLL_{max} = 2^{m+1} - 2 \tag{27}$$

The reason is obvious by observing the illustration in Fig. 2.

Fig. 2 Illustration of RLL_{MAX} set by (27)

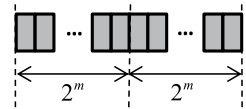
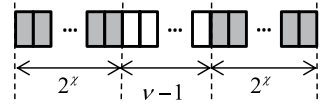


Fig. 3 Illustration of RLL_{MAX} set by (32)



(b) Three and five-times extended Reed Solomon codes

For a ν -times extended RS code $\nu > 1$ and $\nu \in \{3, 5\}$, the first macro-row \mathbf{h}_{II} in parity check matrix (4) will contain $\nu - 1$ submatrices \mathbf{Z} of all zeros with dimensions $m \times m$.

$$\mathbf{h}_{II} = [\mathbf{A}^0 \ \mathbf{A}^0 \ \dots \ \mathbf{A}^0 \ \mathbf{Z} \ \dots \ \mathbf{Z}] \tag{28}$$

The Hamming weight of each row in (28) will be even (29).

$$w_H(S) = 2^\chi \tag{29}$$

where χ is an odd integer.

The number of type S intervals fulfilling Prop. in (28):

$$E = \chi \tag{30}$$

For each of these intervals the cardinality is equal to the number of submatrices \mathbf{A}^0 in \mathbf{h}_{II} , which is identical to $w_H(S)$ given by (29) or simply:

$$|S| = 2^\chi \tag{31}$$

where $\chi \geq 3$ is an odd integer. In these intervals one may flip an odd number of binary symbols by adding the modifier in order to ensure that not all symbols in that interval will be the same. The union of all these intervals contains altogether $m \cdot 2^\chi$ column indexes of macro-row given by (28) and also of the corresponding parity check matrix. For the more times-extended RS codes it is necessary to use such reordering so that the zeros from submatrices denoted as \mathbf{Z} in (28) will be divided and placed evenly between the interval denoted as S in order to get the minimal value of RLL_{MAX} . After doing it, the maximal run lengths in RLL ECCs obtained by the presented method from ν -times extended RS codes $\nu \in \{3, 5\}$ over $GF(2^\chi)$ will be given by:

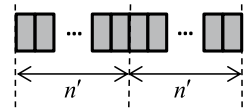
$$RLL_{MAX} = 2^\chi + \nu - 3 \tag{32}$$

It is illustrated in Fig. 3.

(c) Shortened Reed Solomon codes defined by generator matrix with primitive element as a root

If the original RS code given by (3) has α^0 as a first root, the number of ones in each row of the first macro-row from top \mathbf{h}_I in the parity check matrix will be odd. However, by shortening it by σ symbols, where σ is an odd integer, it is possible to achieve that the first macro-row \mathbf{h}_{III}

Fig. 4 Illustration of RLL_{MAX} set by (34), $n' = 2^{(m-2)} + 1 - \sigma$



in the shortened RS code will have an even number of ones in each row as well. In this case the number of ones in each set S will be:

$$w_{III}(S) = 2^{(m-2)} + 1 - \sigma \quad (33)$$

and

$$RLL_{MAX} = 2(2^m - \sigma - 2) \quad (34)$$

in the obtained RLL-ECCs. The reason is illustrated in Fig. 4.

4 On Some Properties of the Obtained RLL-ECCs

In this section some notes on the properties of the obtained RLL-ECCs will be made.

The first note is that the presented method will not change the basic parameters $[n, k, d_m]$ of the original RS codes from which the RLL-ECC construction starts. In other words, the addition of the modifier and the reordering of the symbols in the codeword will not change $[n, k, d_m]$. It is well known that this is valid for any linear block code. The immediate consequence is that the redundancy of the original RS codes will not be increased by applying the method which transforms the original RS codes to RLL-ECCs. From the first note one can also assume that the error correcting capability of the obtained RLL ECCs will be the same as the original codes have. However the error correcting capabilities of ECCs depend also on the method which is used for their decoding. Consequently this assumption is correct only if some round conditions are valid and only for some decoding methods.

Therefore the second note is that if some round conditions are valid the error correcting capabilities of the obtained RLL-ECCs are identical to the original RS codes and shortened original RS codes for selected decoding methods without the need to change them. This second note needs some more explanation than the fact connected with the first note.

The basic practical restriction is that the method for derivation of binary RLL-ECCs from non-binary Reed Solomon codes implies that it can be useful only in cases where the binary system is used for transmission. Such a system is in [21] modeled using a Binary Symmetric Channel (BSC) or system which uses coherent binary phase shift keying (BPSK) modulation over Additive White Gaussian Noise (AWGN) channel. In [21] it was shown that hard and soft decoding of binary Low Density Parity Check Codes (LDPC) do not have to be modified when the proposed method is used in order to construct RLL-LDPC codes and also that error correcting capabilities of the RLL-LDPC codes will stay the same as for the original LDPC codes. The soft decoding method for which it is valid and which was analyzed in [21] is the well-known and most often used belief propagation (BP) decoding operating with log-likelihood ratios (LLRs).

The natural question arises if the results from [21] could be applied also for RLL-ECCs obtained from selected RS codes. In other words if the original hard or soft decoders of RS codes can stay intact after the proposed method for construction of RLL-ECCs from RS codes is used and if the method will not have an influence on the error correcting capabilities of

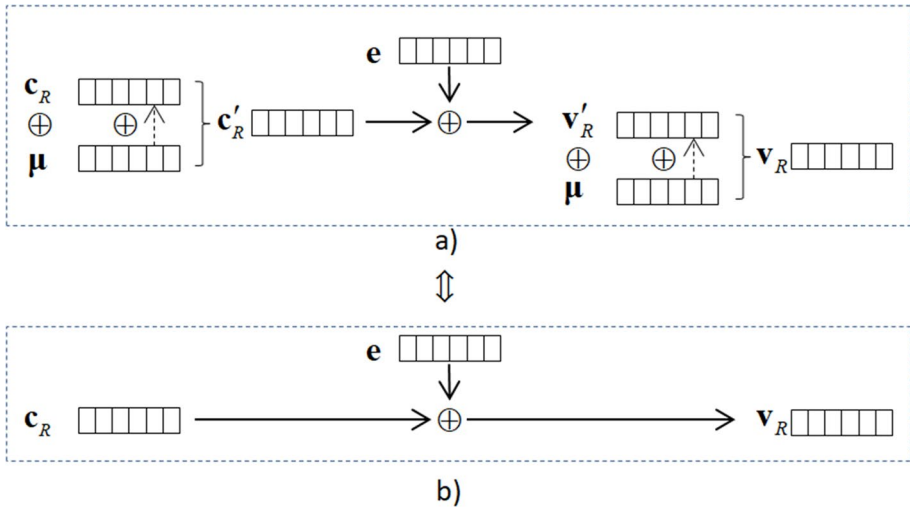


Fig. 5 Illustration of why the modifier has no influence on hard decision decoding and why the hard decision decoding algorithm does not have to be modified in cases where the described method for obtaining RLL ECCs from RS codes is used

the constructed RLL-ECCs. As will be obvious in the same round conditions as in [21] the answer is positive for selected decoding methods.

In case the transmission is modeled using BSC the reasons why the decoding methods do not have to be modified and also that the double addition of a modifier will not have any influence on the error correcting capabilities of the RLL-ECCs are trivial. Namely the addition of the modifier at the input to the BSC and at the output of the BSC cancel out therefore it is obvious that they do not have any influence on the decoding. This is valid for any hard decoding method.

The part concerning the addition of modifiers is illustrated in Fig. 5. In Fig. 5a the re-ordered codeword c_R is modified by the modifier μ :

$$c'_R = c_R + \mu \tag{35}$$

c'_R is the codeword from the RLL-ECC which is transmitted through the BSC. The errors which can occur in the BSC are modeled by the error vector e . The received vector v'_R

$$v'_R = c'_R + e \tag{36}$$

After the second addition of the modifier

$$v'_R + \mu = c'_R + e + \mu = c_R + e + \mu + \mu = v_R \tag{37}$$

From (35)–(37) it is obvious that the models depicted in Figs. 5a and b are equivalent from the point of their inputs and outputs.

In cases of soft decision decoding methods the situation deserves more explanatory text. It has to be mentioned that soft decoding of RS codes presented a challenge for a long time. However because the RS codes have so many applications, there was lot of research work on this challenge and today multiple soft decision decoding methods are known. Therefore, it is

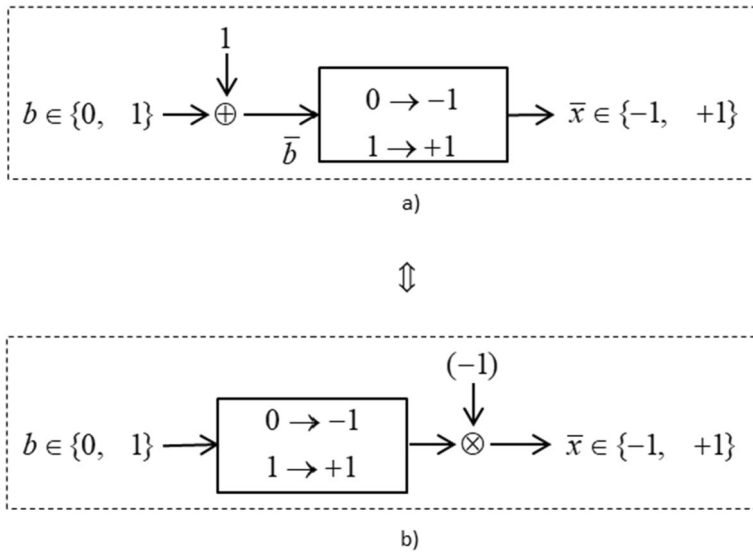


Fig. 6 Illustration of two equivalent models representing the influence of a single 1 from the modifier on one symbol on the transmitting side

appropriate to start with basic categorization of the methods which are known at the present time for soft decoding of RS codes defined over $GF(2^m)$. These soft decoding methods could be categorized into two classes. The first class is the so called symbol-level decoding and the second class the so called bit-level decoding. In the first class the most frequently used methods were proposed in [22] and [23]. In [24] it is argued that the results presented in most previous publications are suggesting that the second class, namely bit-level decoding is superior to the symbol level concerning error correcting capabilities.

The bit level soft decision decoding methods for RS codes defined over $GF(2^m)$ are based on the usage of Tanner graphs obtained from binary images of the parity check matrices of these RS codes. They can also use LLR values for operation. Therefore they are compatible with RLL-ECCs obtained by using the method presented in this paper which is also based on such binary images of parity check matrices of RS codes. Moreover the arguments used in [21] are valid for these methods as well. Probably the most popular example of bit level soft decision decoding of RS codes was proposed in [25]. However any soft decision bit level decoding algorithm can be used without change for RLL-ECCs which can operate with LLR values. Examples are the decoding methods proposed in [25–28].

For the convenience of the reader Figs. 6 and 7 are included which illustrate the elimination of the influence of the modifier on LLR values in cases where the coherent BPSK and AWGN channel is used for modelling the transmission of RLL ECCs.

In Fig. 6 it is illustrated that the influence on one binary symbol $b \in \{0, 1\}$ by adding modulo 2 or in other words XOR-ing it with one from a modifier and following mapping the result on symbol $x \in \{+1, -1\}$ Fig. 6a can be equivalently modeled as first mapping $b \in \{0, 1\}$ on $x \in \{+1, -1\}$ and then multiplying the result with -1 Fig. 6b.

In Fig. 7 the overall situation is depicted which illustrates that the influence of the 1 from the modifier on the LLR connected with the corresponding symbol is eliminated by multiplying the LLR by -1 .

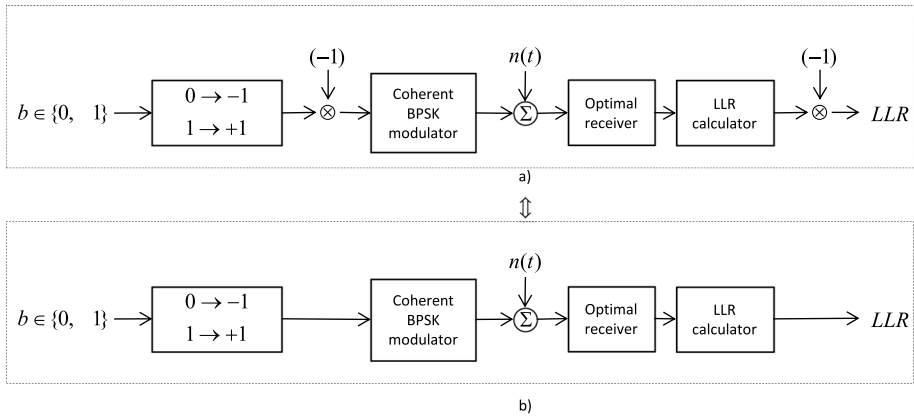


Fig. 7 Illustration of how the influence of the modifier on the LLR can be eliminated in the receiver via multiplication by -1 . (The optimal receiver is composed of a matched filter and sampler)

The following argumentation is an extract from [21]. Let's choose the following mapping between the data bits and the coherent BPSK modulation:

$$0 \rightarrow s_1(t) = \sqrt{E} \cos(2\pi ft) \tag{38}$$

$$1 \rightarrow s_2(t) = -\sqrt{E} \cos(2\pi ft) \tag{39}$$

Using this mapping in the original system (without using a modifier) the LLR is:

$$L(z/s) = \ln \frac{p(z/0)}{p(z/1)} = \ln \frac{\frac{1}{\sigma_0 \sqrt{2\pi}} \exp \left[-\frac{1}{2} \left(\frac{z+\sqrt{E}}{\sigma_0} \right)^2 \right]}{\frac{1}{\sigma_0 \sqrt{2\pi}} \exp \left[-\frac{1}{2} \left(\frac{z-\sqrt{E}}{\sigma_0} \right)^2 \right]} = -\frac{2\sqrt{E}}{\sigma_0^2} z \tag{40}$$

z , $p(z/b)$ and σ_0 are the sample at the output of the optimal receiver, the likelihood of the sample (conditional pdf that the sample has value z if bit b was transmitted) and the dispersion of the AWGN on the output of the optimal receiver respectively.

If the modifier inverts the transmitted bit (which will cause exchanged the mapping given by (38) and (39)) the transmitted bit the LLR becomes

$$L(z/s) = \ln \frac{p(z/0)}{p(z/1)} = \ln \frac{\frac{1}{\sigma_0 \sqrt{2\pi}} \exp \left[-\frac{1}{2} \left(\frac{z-\sqrt{E}}{\sigma_0} \right)^2 \right]}{\frac{1}{\sigma_0 \sqrt{2\pi}} \exp \left[-\frac{1}{2} \left(\frac{z+\sqrt{E}}{\sigma_0} \right)^2 \right]} = +\frac{2\sqrt{E}}{\sigma_0^2} z \tag{41}$$

This confirms that the multiplication by (-1) in the receiver will eliminate the influence of the modifier in the receiver.

In this section it was shown that in case of a BSC the influence of reordering and addition of a modifier are eliminated in the receiver fully by a second addition of the modifier and de-reordering. In cases of soft decision decoding there are selected decoding methods for the RLL ECCs obtained by the method presented in this paper for which there will not

be any influence on the performance of the decoding when compared with the original RS codes if the communication system is modeled using an AWGN channel and coherent BPSK modulation. It is just required to invert the LLR values in the receiver corresponding to binary symbols inverted in the transmitter by the ones in the modifier and then reorder the symbols before the RLL values are input into the soft decision decoder [21]. For example the soft decision decoding methods proposed in [27, 28] can be used in this case.

5 Conclusion

In this paper it was shown that some RS codes could be transformed into RLL-RS codes using the presented method and that the maximal run length in the obtained codes could be given analytically. It introduces no additional redundancy and it is not necessary to develop new encoding and decoding procedures for the obtained RLL-ECCs codes if some round conditions and appropriate decoding methods are used. It is not excluded that other RS codes could also be transformed into RLL-ECCs. This is an open question and further research will be needed in order to answer it.

Acknowledgements This work was supported by the Slovak Research and Development Agency under Contract no. APVV-19-0436.

Author Contributions The first author, Prof. Peter Farkaš, DrSc. made substantial contribution by discovering the method used for construction of RLL RS codes from some original extended and shortened RS codes. He wrote the first version of the manuscript and also the second version after first round of review. He also revised critically the final manuscript and approved the version to be published. He agree to be accountable for all aspects of the work in ensuring that questions related to the accuracy or integrity of any part of the work are appropriately investigated and resolved. The second (corresponding) author Assoc. Prof. Martin Rakús, PhD. contributed by discovering and describing the infinite family of five times extended Reed Solomon (RS) codes over certain finite fields and showing that their code distance is equal to the upper bound on it in tables [15] and [16] on best linear block codes. He also revised critically the final manuscript and approved the version to be published. He agree to be accountable for all aspects of the work in ensuring that questions related to the accuracy or integrity of any part of the work are appropriately investigated and resolved.

Funding Open access funding provided by The Ministry of Education, Science, Research and Sport of the Slovak Republic in cooperation with Centre for Scientific and Technical Information of the Slovak Republic. This work was funded by: Slovak Research and Development Agency under Contract No. APVV-19-0436.

Data Availability The authors declare that no DATASETS were used for the research which resulted in construction of RLL RS codes and which is described in this manuscript.

Code Availability Not applicable.

Declarations

Conflict of interest The authors declare that they have no conflict of interest nor competing interests.

Ethical approval The authors made their best to adhere to all Ethics & Disclosures guidelines specified in Wireless Personal Communication Journal and Committee on Public Ethics.

Consent to participate All authors achieved consent to participate in preparation of this manuscript.

Consent for publication All authors achieved consent to publish this manuscript in WPC.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Lin, S., & Li, J. (2021). *Fundamentals of classic and modern error-correcting codes*. Cambridge University Press.
2. Savicente, E. (2019). *Understanding error control coding*. Springer.
3. Imminck, K. A. S. (2004). *Codes for mass data storage systems* (2nd ed.). Eindhoven.
4. Farkaš, P., Rakús, M., & Páleník, T. (2021). A new technique for incorporating RLL properties into 5G LDPC codes without additional redundancy. *Wireless Personal Communication*, 119(1), 749–762.
5. Farkaš, P., & Rakús, M. (2023). Adding RLL properties to four CCSDS LDPC codes without increasing their redundancy. *Computing and Informatics*, 42(1), 157–197.
6. Farkaš, P., Divinec, L., & Rakús, M. (2022). Run length limited CCSDS convolutional codes for optical communications. *Journal of Electrical Engineering*, 73(6), 432–436. <https://doi.org/10.2478/jee-2022-0059>
7. Chen, J., Li, Z., Zhou, R., Su, L., & Wang, N. (2023). Advanced elastic Reed-Solomon codes for erasure-coded key value stores. *Early Access*. <https://doi.org/10.1109/JIOT.2023.3299574>
8. Liu, X., Zhang, W., Chang, Y., & Liu, Y. (2023). A novel concatenation decoding of Reed-Solomon codes with SPC product codes. *IEEE Signal Processing Letters*, 30, 808–812. <https://doi.org/10.1109/LSP.2023.3291653>
9. Liu, H., Wei, H., Wachter-Zeh, A., & Schwartz, M. (2023). Linearized Reed-Solomon codes with support-constrained generator matrix, Saint-Malo, France. In *IEEE Information Theory Workshop (ITW)*. <https://doi.org/10.1109/ITW55543.2023.10161635>
10. Yu, L., Lin, S.-J., Hou, H., & Li, Z. (2023). Reed-Solomon coding algorithms based on Reed-Muller transform for any number of parities. *Early Access*. <https://doi.org/10.1109/TC.2023.3262922>
11. Con, R., Shpilka, A., & Tamo, I. (2023). Reed Solomon codes against adversarial insertions and deletions. *IEEE Transactions on Information Theory*, 69(5), 2991–3000. <https://doi.org/10.1109/TIT.2023.3237711>
12. Liu, L., Chi, L., Huang, S.-Y., & Yeh, K.-Y. (2023). Compiler of Reed-Solomon codec for 400-Gb/s IEEE 802.3bs standard. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 42(8), 2776–2780. <https://doi.org/10.1109/TCAD.2022.3232994>
13. Chen, B., Ling, S., & Liu, H. (2023). Hulls of Reed-Solomon codes via algebraic geometry codes. *IEEE Transactions on Information Theory*, 69(2), 1005–1014. <https://doi.org/10.1109/TIT.2022.3211887>
14. Goldberg, E., Shangguan, C., & Tamo, I. (2023). List-decoding and list-recovery of Reed-Solomon codes beyond the Johnson radius for every rate. *IEEE Transactions on Information Theory*, 69(4), 2261–2268. <https://doi.org/10.1109/TIT.2022.3222877>
15. Grassl, M. (2022). Linear block codes: <https://codetables.de>, Accessed in August 2023
16. Department of Mathematics, U.o.S. (2015). Generate parameter table for linear codes: <https://mint.sbg.ac.at>, accessed in August 2023
17. Rakús, M., Farkaš, P., Páleník, T., & Daniš, A. (2019). Five times extended Reed-Solomon codes applicable in memory storage systems. *IEEE Letters of the Computer Society*, 2(2), 9–11. <https://doi.org/10.1109/LOCS.2019.2911517>
18. MacWilliams, F. J., & Sloane, N. J. A. (1977). *The theory of error-correcting codes*. North-Holland.
19. Farkaš, P., & Schindler, F. (2017). Run length limited error control codes construction based on one control matrix property. *Journal of Electrical Engineering*, 68(4), 322–324.
20. Lee, H. C., Huang, G. X., Wang, C. H., & Ueng, Y. L. (2015). Iterative soft-decision decoding of reed-solomon codes using informed dynamic scheduling, Hong Kong. In *2015 IEEE International Symposium on Information Theory (ISIT)*. <https://doi.org/10.1109/ISIT.2015.7282989>

21. Farkaš, P., & Páleník, T. (2022). On soft decoding of some binary RLL-transmission codes in systems with coherent BPSK modulation, Visegrád, Hungary. *Cybernetics & Informatics (K &I)*. <https://doi.org/10.1109/KI55792.2022.9925949>
22. Koetter, K. R., & Vardy, A. (2003). Algebraic soft-decision decoding of Reed-Solomon codes. *IEEE Transactions on Information Theory*, 49(11), 2809–2825.
23. Oatall, O., Ogundile, Y., & Genga, Y. (2015). Symbol level iterative soft decision decoder for Reed-Solomon codes based on parity-check equations. *Electronics Letters*, 51(17), 1332–1333.
24. Jiang, J., & Narayanan, K. R. (2006). Iterative soft-input soft-output decoding of Reed-Solomon codes by adapting the parity-check matrix. *IEEE Transactions on Information Theory*, 52(8), 3746–3756. <https://doi.org/10.1109/JSAC.2005.862399>
25. El-Khamy, M., & McEliece, R. J. (2006). Iterative algebraic soft-decision list decoding of Reed-Solomon codes. *IEEE Journal on Selected Areas in Communications*, 24(3), 481–490. <https://doi.org/10.1109/JSAC.2005.862399>
26. Scholl, S., Haider, S. K., & Wehn, N. (2016). An efficient soft decision Reed-Solomon decoder for moderate throughput, Lemesos, Cyprus. In *18th Mediterranean Electrotechnical Conference (MELECON)*. <https://doi.org/10.1109/TCOMM.2020.3047085>
27. Jing, Y., Zhang, W., Wang, H., Chang, Y., & Liu, Y. (2022). Improved adaptive belief propagation decoding of Reed-Solomon codes With SPC codes. *IEEE Communications Letters*, 26(7), 1464–1468. <https://doi.org/10.1109/LCOMM.2022.3167989>
28. Deng, L., Liu, Z., Guan, Y. L., Liu, X., Aslam, C. A., Yu, X., & Shi, Z. (2021). Perturbed adaptive belief propagation decoding for high-density parity-check codes. *IEEE Transactions on Communications*, 69(4), 2065–2079. <https://doi.org/10.1109/TCOMM.2020.3047085>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Peter Farkaš is with Institute of Multimedia ICT, Slovak University of Technology (STU) and also with Institute of Applied Informatics, Pan-European University in Bratislava as a Professor. From 2002 till 2007 he was Visiting Professor at Kingston University, UK and senior researcher at SIEMENS PSE. In 2003 SIEMENS named him VIP for his innovations and patents. In 2004 he was awarded with the Werner von Siemens Excellence Award for research results on two-dimensional Complete Complementary Codes. From 2008 to 2009 he worked also as Consultant for SANDBRIDGE Tech. (USA). He was responsible leader of a team from STU in projects funded by the European Community under the 5FP and 6FP “Information Society Technologies” Programs: NEXWAY IST -2001-37944 (Network of Excellence in Wireless Applications and technology) and CRUISE (Creating Ubiquitous Intelligent Sensing Environments) FP6 IST-2005- 4-027738, (2006-2007). His research interests include Error Control Coding, Communications Theory and sequences for CDMA. He published 3 chapters in books 45 papers in reviewed scientific journals

and about 98 papers in international conferences. He is author or coauthor of 7 patents. He is and was serving in TPC of about 60 international conferences and presented 12 invited lectures. As an IEEE volunteer, he was serving in IEEE Czechoslovakia Section Executive Committee in different positions from 1992 to 2014 and from 2005 to 2006 he served as a chair of Conference Coordinator Subcommittee in IEEE Region 8. He organized IEEE R8 Conferences EUROCON'2001, SympoTIC'03, 05, 06 and Winter school on Coding and Information Theory 2005.



Martin Rakús studied radio electronics and graduated from the Slovak University of Technology in 2001. In 2004 he received his PhD from the Slovak University of Technology and in 2020 he became Associate Professor at the same institute. Since 1995 he has been with the Institute of Multimedia Information and Communication Technology, faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava, Slovakia. His primary research interests are error control coding and digital communication systems. He is a member of the IEEE.