Analysis of the Probability of Sync-words in Reed-Solomon Codes

Thokozani Shongwe
Department of Electrical and Electronic Engineering Technology, University of Johannesburg,
P.O. Box 17011, Doornfontein, 2028, Johannesburg, South Africa
Email: tshongwe@uj.ac.za or caltoxs@gmail.com

***Abstract*—Given binary data transmission encoded using a Reed-Solomon (RS) code and employing binary sync-words (markers) for synchronization, we calculate the probability of finding the sync-word in the codewords of the RS code as P_S. We give analytical expressions for calculating P_S, which is applicable to RS codes. Knowledge of P_S can be used to calculate the probability of finding a sync-word that is used as a marker in RS encoded data, P_T. The probability P_T is called the false acquisition probability in the synchonization of RS encoded data.**

***Index Terms*—Reed-Solomon codes, Frame Synchronization, Sync-words.**

I. Introduction

In synchronization, using sync-words, it is desired that the sync-word has a very low probability of occurring in the data being synchronized in order to minimise (or avoid) confusion between the data and the sync-word. The work in [1] gave simulation results of the probability of false acquisition in a Reed-Solomon (RS) codeword. As a follow up on [1], in this paper we give analytical expressions for the probability of finding sync-words in Reed-Solomon (RS) codes (probability of false acquisition). If we consider transmission of RS encoded data which is synchronized using sync-words, these expressions can be used to estimate the probability of false acquisition of the sync-word in RS codes, P_T (the probability of finding the sync-word where it was not inserted in the data). In [2] a method of avoiding specific symbols in RS codes, inspired by [3], was presented. This method in [2] found an application of reducing the propability of false acquisition in RS codes and resulted in the article in [4]. However, the work in [4] focuses on modified RS codes, and does not give analytical expressions for the probability of false acquisition in RS codes.

The reader who wants to get acquainted with the principles of Reed-Solomon encoding is referred to [5] and [6]. When describing RS codes over GF(2^m), where m is a positive integer, we will be using integer symbols to represent elements of GF(2^m) because the integer symbols make it easier to follow operations on the RS codes and also aid presentation. We will, in short, refer to the integer symbols simply as symbols. When the distance properties are not important, as is the case in the rest of the paper, we will refer to the RS codes as $(2^m-1, k)$ RS code (or (n, k) RS code), where m is the number of bits per symbol, k is the dimension of the code and $n = 2^m - 1$ is the codeword length.

II. Probability of a Subsequence of Symbols

Consider an n-symbol sequence S_n, where the symbols are equiprobable and statistically independent. Let the complete set of such symbols making up S_n have a cardinality q, hence the probability of a symbol in S_n is $\mathcal{P} = 1/q$. We now want to find the probability of an L_S-symbol subsequence of consecutive symbols from S_n, where $L_S \leq n$. Denote by P_S, the probability of an L_S-symbol subsequence from S_n. P_S is going to be used later on to estimate the probability of a sync-word in a RS code. For now we give an expression for P_S.

We consider a case where the L_S-symbol subsequences can share symbols. This means that each L_S-symbol subsequence has no unique symbols and shares some of its symbols with the neighbouring L_S-symbol subsequences. The expression for P_S for this scenario is given as

$$P_S \approx \frac{1}{d_{\max}} \sum_{i=1}^{d_{\max}} i \binom{n - L_S + 1}{i} \mathcal{P}^{iL_S} (1 - \mathcal{P}^{L_S})^{(n-L_S+1-i)}, \tag{1}$$

where $d_{\max} = n - L_S + 1$, $\binom{n-L_S+1}{i}$ is combinations and the subscript i indicates the number of L_S-symbol subsequences in S_n. The expression in (1) can be found in [5, pp. 345], which is an approximation used here to estimate the probability of finding an L_S-symbol subsequence in S_n. Our expression in (1) is a good approximation for large n and small L_S, and can be verified through simulations.

III. Probability of a Sync-word in a RS Code

In the previous section, an expression for the probability of a subsequence of symbols, in a given sequence, where symbols are generated with a uniform distribution, was given. It was required that the symbols occur independently with a uniform distribution. In this section we show that a subset of the symbols in a RS code behave the same as equiprobable and statistically independent symbols. This enables us to apply the same analysis as in Section II to find the probability of a subsequence of symbols, P_S in a RS code (or a RS codeword). We shall soon show that a sync-word, in relation to a RS code, is made up of subsequences from the same symbols as the RS code, hence we can use the P_S to find the probability of the sync-word in a RS code (which gives the probability of false acquisition on data).

To show that a subset of the symbols in a RS codeword behave the same as equiprobable and statistically independent symbols, we need the following definition.

Definition 1 A vector of n random variables is called k-wise independent if each subset of k of the variables is independent, where $k \leq n$.

In other words, this definition means that if any k (or less than k) random variables out of the n are selected, they form an independent distribution. Definition 1 can be found in different variations in the literature, for example, see [7] and [8].

An (n, k) RS code over GF(q) has its n-symbol *codewords* as vectors, where the symbols are random variables such that any codeword is k-wise independent. The k symbols of a RS codeword form a uniform distribution with a symbol probability of $1/q$. It is easy to see that for an (n, k) RS code, any of the n-symbol codewords is k-wise independent. This follows from the following knowledge about the generator matrix and message vector of a RS code. The symbols in a message vector of length k are uniformly distributed. This message vector results in a codeword of length n, when multiplied by the $k \times n$ generator matrix G. The $k \times n$ generator matrix has rank k, that is, any k columns of the generator matrix are independent. The implication of this is that any k symbols in the codeword are a result of multiplying the message vector with k independent columns of G, hence those k symbols form a uniform distribution.

Throughout this paper we deal with known binary sync-words. To represent the binary sync-words as subsequences, with symbols taken from the same alphabet as the RS code, we break down the sync-words into their symbol equivalents as follows. For a $(2^m - 1, k)$ RS code, sync-words of any number of bits can be represented by the m-bit symbols from GF(2^m). All possible symbol combinations that make up the sync-word are listed, and the probabilities, for each combination found in the RS code, are calculated and summed up to give an estimate of the probability of finding the sync-word in the RS code. The representation of a sync-word by the symbols from GF(q) is illustrated by Example 1. We shall refer to the representation of a sync-word by the symbols from GF(q) as *symbol make-up* of a sync-word.

Example 1 For a $(2^m - 1, k)$ RS code, let $m = 4$, and let the sync-word be a 7-bit Barker sequence $S = \{1110010\}$. This results in the symbol make-up of S as shown in Figure 1. To arrive at the result of Figure 1, S is left shifted one bit at a time, over the three blocks (the rectangles containing bits) of possible symbols, until all possible combinations of symbols making up S are found.

The symbol make-up for S is shown in four groups, A, B, C and D. The groups A and D, each consists of two (2) subsequences of length two, and groups B and C, each consists of 32 subsequences of length three. The symbol X in a block in a group means that the bit can either be a 0 or a 1. As an example, in group A the block with $010X$ means that the four bit sequence there can be either 0100 or 0101 which represents either a 4 or a 5, respectively hence the set $\{4, 5\}$ in that block.

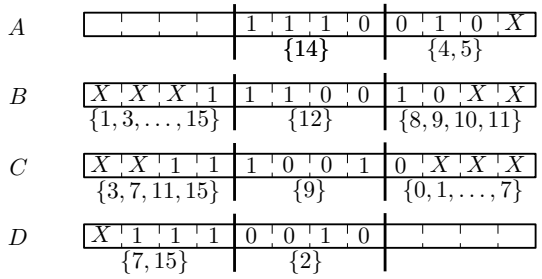

Fig. 1. Symbol make-up of a 7-bit Barker sequence $S = \{1110010\}$.

Now, Figure 1 can be interpreted as follows. For a given group, selecting a symbol in each block beginning from the leftmost block up to the rightmost block, will result in the bit sequence S. A block without a symbol has no effect since it has no symbol or bits. The list of all the possible combinations of symbols that can result in the bit sequence S is:

$$A = \begin{bmatrix} 14 & 4; & 14 & 5 \end{bmatrix},$$

$$B = \begin{bmatrix} 1 & 12 & 8; & 1 & 12 & 9; & 1 & 12 & 10; & 1 & 12 & 11; \\ 3 & 12 & 8; & 3 & 12 & 9; & 3 & 12 & 10; & 3 & 12 & 11; \\ 5 & 12 & 8; & 5 & 12 & 9; & 5 & 12 & 10; & 5 & 12 & 11; \\ 7 & 12 & 8; & 7 & 12 & 9; & 7 & 12 & 10; & 7 & 12 & 11; \\ 9 & 12 & 8; & 9 & 12 & 9; & 9 & 12 & 10; & 9 & 12 & 11; \\ 11 & 12 & 8; & 11 & 12 & 9; & 11 & 12 & 10; & 11 & 12 & 11; \\ 13 & 12 & 8; & 13 & 12 & 9; & 13 & 12 & 10; & 13 & 12 & 11; \\ 15 & 12 & 8; & 15 & 12 & 9; & 15 & 12 & 10; & 15 & 12 & 11 \end{bmatrix},$$

$$C = \begin{bmatrix} 3 & 9 & 0; & 7 & 9 & 0; & 11 & 9 & 0; & 15 & 9 & 0; \\ 3 & 9 & 1; & 7 & 9 & 1; & 11 & 9 & 1; & 15 & 9 & 1; \\ 3 & 9 & 2; & 7 & 9 & 2; & 11 & 9 & 2; & 15 & 9 & 2; \\ 3 & 9 & 3; & 7 & 9 & 3; & 11 & 9 & 3; & 15 & 9 & 3; \\ 3 & 9 & 4; & 7 & 9 & 4; & 11 & 9 & 4; & 15 & 9 & 4; \\ 3 & 9 & 5; & 7 & 9 & 5; & 11 & 9 & 5; & 15 & 9 & 5; \\ 3 & 9 & 6; & 7 & 9 & 6; & 11 & 9 & 6; & 15 & 9 & 6; \\ 3 & 9 & 7; & 7 & 9 & 7; & 11 & 9 & 7; & 15 & 9 & 7 \end{bmatrix},$$

$$D = \begin{bmatrix} 7 & 2; & 15 & 2 \end{bmatrix}. \qquad \square$$

Proposition 1 *Given a sync-word of N bits and any $(2^m - 1, k)$ RS code, if the N bits of the sync-word are grouped into m bits to form m-bit integer symbols, there will be a maximum of:*

1) $(m - r + 1)2^{m-r}$ *sequences of* $\lceil N/m \rceil$ *integer symbols*
2) $(r - 1)2^{2m-r}$ *sequences of* $\lceil N/m \rceil + 1$ *integer symbols, where r is the remainder of N/m, $r = 1, 2, \ldots m$. Note that* $(r = 0) \equiv (r = m)$. $\square$

The total probability of a sequence of N bits is then given by

$$\begin{aligned} P_T &= (m - r + 1)2^{m-r} P_S(L_S = \lceil N/m \rceil) \\ &\quad + (r - 1)2^{2m-r} P_S(L_S = \lceil N/m \rceil + 1), \end{aligned} \quad (2)$$

where $P_S(x)$ denotes that P_S is a function of x.

PROOF To give a sketch of the proof of Proposition 1 we will occasionally refer to Example 1 to enhance understanding.

Take any row of N bits and partition them in blocks of m bits as was done in Figure 1. Each whole number of m bits in a block can form a decimal integer. Let the number of the remainder of the bits which are less than m be denoted r, where $r = 0, 1, 2, \ldots m-1$.

1) It is easy to see that this partitioning of the row of N bits in blocks of m bits will result in $\lceil N/m \rceil$ blocks, including the block of remainder bits. The $m-r$ empty spaces (denoted by X in Figure 1, where $X \in \{0,1\}$) in the incomplete block will result in 2^{m-r} possible integers. Let the number of such rows/groups which result in $\lceil N/m \rceil$ blocks be α, then there will be a total of $\alpha 2^{m-r}$ possible integer sequences of length $\lceil N/m \rceil$ which can form the N-bit sequence. We will soon find α.
2) Now if we begin shifting the bits partitioned in blocks either to the left or right, we begin to see a spill-over of the bits which create one extra partially occupied block such that there will be $\lceil N/m \rceil + 1$ blocks as shown by Groups B and C in Figure 1. If we view the spill-over bits as coming from the r bits, we can see that the r bits are now shared between two incomplete block, the total number of empty spaces in those blocks will now be $m + (m-r)$, hence giving a total of $2^m \times 2^{m-r} = 2^{2m-r}$ possible integers. This extra partially occupied block only occurs when $r > 1$. Let the number of such rows/groups which result in $\lceil N/m \rceil + 1$ blocks be β, then there will be a total of $\beta 2^{2m-r}$ possible integer sequences of length $\lceil N/m \rceil + 1$ which can form the N-bit sequence. We will find β, as well as α next.
3) Shifting the partitioned bits will result in a maximum of m groups. Therefore $\alpha + \beta = m$. We need to find α or β. The groups with $0, 1, 2 \ldots m-r$ empty spaces are therefore $m-r+1$ in total, and those are the groups with $\lceil N/m \rceil$ blocks. This therefore means that $\alpha = m-r+1$. Then, $\beta = m - \alpha = m - (m-r+1) = r-1$.
4) There is a special case of $r = 0$, that is when there are no remainder bits in the division of N by m. For this case, $r = 0$ is equivalent to $r = m$ because the last block of the $\lceil N/m \rceil$ will be completely filled with m bits. Hence $r = 1, 2, \ldots m$. ■

Having shown how a binary sync-word can be represented by the symbols from GF(q), same as the RS code of interest, it can easily be shown how P_S can be used to calculate the probability of a sync-word in a RS code, P_T. The following example illustrates this.

Example 2 Using the symbol make-up of the $N = 7$ bits Barker sequence, $S = \{1110010\}$ in Example 1, let the $(2^m - 1, k)$ RS code have $m = 4$ and $k = 3$. The RS code will be over GF(2^m) of size $q = 2^m = 16$, and length $n = 2^m - 1 = 15$. The probability of a symbol in the RS code is then $\mathcal{P} = 1/q = 1/16$.

As mentioned in Example 1, the groups A and D, each consists of two (2) subsequences of length two, and groups B and C, each consists of 32 subsequences of length three. This can be related to Proposition 1 as follows: The remainder of N/m is $r = 2$.

1) Groups A and D: there are $(m-r+1)2^{m-r} = (4-3+1)2^{4-3} = 4$ sequences of length, $L_S = \lceil N/m \rceil = \lceil 7/4 \rceil = 2$.
2) Groups B and C: there are $(r-1)2^{2m-r} = (3-1)2^{2\times 4-3} = 64$ sequences of length, $L_S = \lceil N/m \rceil = \lceil 7/4 \rceil + 1 = 2 + 1 = 3$

The probability of the $N = 7$ bits Barker sequence $S = \{1110010\}$ in the $(15, 3)$ RS code is then the probability of the group A and D sequences plus the probability of the group B and C sequences, and is given by the expression in (2) as

$$P_T = 4P_S(L_S = 2) + 64P_S(L_S = 3), \tag{3}$$

where

$$\begin{aligned} P_S(L_S = 2) &= \frac{1}{14}\sum_{i=1}^{14} i \binom{15+i-1}{i} \mathcal{P}^{2i}(1-\mathcal{P}^2)^{(14-2+1-i)} \\ &= 4.2 \times 10^{-3} \end{aligned}$$

and

$$\begin{aligned} P_S(L_S = 3) &= \frac{1}{13}\sum_{i=1}^{13} i \binom{15+i-1}{i} \mathcal{P}^{3i}(1-\mathcal{P}^3)^{(14-3+1-i)} \\ &= 3 \times 10^{-4}, \end{aligned}$$

therefore

$$\begin{aligned} P_T &= 4P_S(L_S = 2) + 64P_S(L_S = 3) \\ &= 4 \times 4.2 \times 10^{-3} + 64 \times 3 \times 10^{-4} \\ &= 0.035. \end{aligned}$$

□

It should be noted that if any number of symbols in the sync-word symbol make-up exceed the dimension of the $(2^m - 1, k)$ RS code, i.e. $L_S > k$, the analytical expression for P_S cannot be applied as the L_S symbols will no longer be following a uniform distribution. This is therefore a limitation to our analytical expression for P_S. The next task is then to state when this limitation occurs. Given a binary sync-word of length N and a $(2^m - 1, k)$ RS code, we want to find the largest L_S in the symbol make up. This largest L_S should not exceed k, for validity of the analytical expression for P_S to be guaranteed.

Theorem 1 *Given a sequence of N bits and any $(2^m - 1, k)$ RS code, the largest value of N for which the sequence can be grouped into m-bit symbols such that the symbols are k-wise independent in relation to the RS code is*

$$N \leq (k-1)m + 1. \tag{4}$$

□

Remark: P_S is valid as long as the m-bit symbols, of the sequence of length N bits, do not exceed k. These symbols will then behave as equiprobable and statistically independent in the $(2^m - 1, k)$ RS code, hence satisfying Definition 1.

PROOF Let us prove the validity of (4) as follows. We need to put a sequence of N bits into groups of m bits, where m bits make up a symbol, i.e. a group here represents a symbol. We can put the N bits into $\lceil N/m \rceil$ groups (symbols), and the groups are completely filled with bits if N is a multiple of m. However, since we perform a shifting of the bits to find all the possible symbols making up the N-bit sequence, we need one extra empty group to which we can shift the bits. The empty group can also hold m bits, therefore as we shift bits from previous groups, the same number of bits shifted out of last group get shifted into the extra group. We can never need more that $\lceil N/m \rceil + 1$ groups (symbols) because when all the m bits from the last group get filled into the extra group, the last group ceases to exist and the situation gets back to $\lceil N/m \rceil$ groups. So, the only time we need $\lceil N/m \rceil + 1$ groups is when there are some bits in the last group as well as some bits in the extra group. There is one special case to consider. This is the case when there is only one bit in the last group. It can be realised that in this case, there is never a situation when there are some bits in the last group as well as in the extra group. This is because when the one bit in the last group gets shifted out of the last group, the last group ceases to exist and gets created again by the bit in the first group. Therefore, there will not be an extra group in this case. Hence, the number of necessary groups, to contain the bits, never exceeds $\lceil N/m \rceil$. Therefore, the analytical expression for P_S is guaranteed to be valid if

$$k \geq \left\lceil \frac{N}{m} \right\rceil + 1. \tag{5}$$

To include the special case of one bit in the last group, we can modify the expression in (5) by removing one bit from the N bits and obtain the expression

$$k \geq \left\lceil \frac{N-1}{m} \right\rceil + 1. \tag{6}$$

By rearranging the expression in (6) and removing the ceiling operator, we get (4). ∎

TABLE I
COMPARISON OF P_T FROM (2) AND P_T FROM SIMULATED RESULTS (P'_T), FOR BINARY SYNC-WORDS OF LENGTHS $7-11$, WRITTEN IN THEIR OCTAL FORMAT. A $(2^4-1,3)$ RS CODE WAS USED.

Binary Sync-words	Length, N	P_T	P'_T
130	7	0.0351	0.0312
270	8	0.0178	0.0156
560	9	0.0090	0.0078

TABLE II
COMPARISON OF P_T FROM (2) AND P_T FROM SIMULATED RESULTS (P'_T), FOR BINARY SYNC-WORDS OF LENGTHS $7-14$, WRITTEN IN THEIR OCTAL FORMAT. A $(2^5-1,3)$ RS CODE WAS USED.

Binary Sync-words	Length, N	P_T	P'_T
130	7	0.0407	0.0391
270	8	0.0205	0.0196
560	9	0.0103	0.0098
1560	10	0.0052	0.0049
2670	11	0.0026	0.0024

TABLE III
COMPARISON OF P_T FROM (2) AND P_T FROM SIMULATED RESULTS (P'_T), FOR BINARY SYNC-WORDS OF LENGTHS $7-16$, WRITTEN IN THEIR OCTAL FORMAT. A $(2^6-1,3)$ RS CODE WAS USED.

Binary Sync-words	Length, N	P_T	P'_T
130	7	0.0477	0.0469
270	8	0.0239	0.0235
560	9	0.0120	0.0117
1560	10	0.0060	0.0059
2670	11	0.0030	0.0029
6540	12	0.0015	0.0015
16540	13	0.000756	0.000732

IV. RESULTS

To prove the validity of the expressions in (1) and (2) we present the following results. Let the P_T from simulated results be denoted P'_T. The results in Tables I, II and III are for $(2^4-1,3)$ RS code, $(2^5-1,3)$ RS code and $(2^6-1,3)$ RS code, respectively. In the simulations, each binary sync-word was searched for in all the codewords of the $(2^m-1,k)$ RS code using a "sliding window", and a count of the found sync-words was divided by all the places searched, and then multiplied by m to obtain P'_T in symbol representation.

The results in Tables I, II and III show the that P_T is very close to P'_T, hence validating our expression for P_T in (2).

V. CONCLUSION

We gave analytical expressions for the probability of finding a subsequence in a sequence. These expressions were applied to Reed-Solomon codes to find the probability of false acquisition on data given a sync-word (in its symbol format), P_T. Such knowledge of the probability of a sync-word occurring in a RS code can be used to evaluate the performance of synchronization. The symbol make-up can serve as guideline on which symbols to avoid in a RS code in order to improve the performance of synchronization.

REFERENCES

[1] T. Shongwe, A. J. H. Vinck and H. C. Ferreira, "Application of Symbol Avoidance in Reed-Solomon Codes to Improve their Synchronization," *Telecommunication Systems*, vol. 63, no. 1, pp. 77-88, Sept. 2016.

[2] T. Shongwe and A. J. H. Vinck, "Reed-Solomon Code Symbol Avoidance," *SAIEE Africa Research Journal*, vol. 105, no. 1, pp. 13–19, Mar. 2014.

[3] G. Solomon, "A note on alphabet codes and fields of computation," *Information and Control*, vol. 25, no. 4, pp. 395–398, Aug. 1974.

[4] T. Shongwe, A. J. H. Vinck and H. C. Ferreira, "Reducing the Probability of Sync-word False Acquisition with Reed-Solomon Codes," in *Proceedings of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, Vancouver, Canada, Aug. 27–26, 2013, pp. 159–164.

[5] B. Sklar, *Digital Communications: Fundamentals and Applications*. Prentice Hall Inc., 1988.

[6] S. Lin and D. J. Costello Jr., *Error Control Coding: Fundamentals and Applications*. Prentice Hall Inc., 1983.

[7] H. Karloff and Y. Mansour, "On construction of k-wise independent random variables," in *Proceedings of the Twenty-sixth Annual ACM Symposium on Theory of Computing*, Montreal, Quebec, Canada, May 23–25, 1994, pp. 564–573.

[8] N. Alon, A. Andoni, T. Kaufman, K. Matulef, R. Rubinfeld, and N. Xie, "Testing k-wise and almost k-wise independence," in *Proceedings of the Thirty-ninth Annual ACM Symposium on Theory of Computing*, San Diego, California, USA, June 11–13, 2007, pp. 496–505.