Reducing the Probability of Sync-word False Acquisition with Reed-Solomon Codes

Thokozani Shongwe
Department of Electrical
and Electronic Engineering Science,
University of Johannesburg,
P.O. Box 524, Auckland Park, 2006,
Johannesburg, South Africa
Email: tshongwe@uj.ac.za

A. J. Han Vinck
University of Duisburg-Essen,
Institute for Experimental Mathematics,
Ellernstr. 29, 45326 Essen, Germany
E-mail: vinck@iem.uni-due.de

Hendrik C. Ferreira
Department of Electrical
and Electronic Engineering Science,
University of Johannesburg,
P.O. Box 524, Auckland Park, 2006,
Johannesburg, South Africa
Email: hcferreira@uj.ac.za

***Abstract*—In our previous work we introduced a method for avoiding/excluding some symbols in Reed-Solomon (RS) codes, called *symbol avoidance*. In this paper, we apply the symbol avoidance method in sync-word based synchronization of RS encoded data. With the symbol avoidance method we reduce the probability of the RS encoded data being mistaken for the sync-word used to delimit the start/end of the data. The symbols in the RS code are avoided according to the sync-word used, such that the sync-word has very low probability of being found in the RS codewords, where it was not inserted. Therefore, for different sync-words, different symbols need to be avoided in the RS code. The goal here is to reduce the probability of false acquisition of the sync-word in the RS encoded framed data.**

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

I. Introduction

The conventional frame synchronization technique, employing sync-words to delimit the beginning or end of a frame, is widely deployed in digital systems and has been shown to be a more practical solution to synchronization compared to synchronizable block codes like comma-free codes [1] [2] [3] [4], prefix codes [5] [6] [7] [8], comma codes [9] [10]. In conventional frame synchronization, performance is optimized in two ways. Firstly, is to design good sync-words with good aperiodic auto-correlation functions (ACFs), that is, the sync-word gives a high value only when there is a perfect match. Secondly, to reduce the probability of data being mistaken for the sync-word, longer sync-words are used. However, there is a problem with using long sync-words in channels with high probability of error. Long sync-words are more susceptible to channel errors compared to short ones. There is therefore a need to find ways of frame synchronization that can reap the benefits of reduced probability of mistaking data for a sync-word without significantly increasing the lengths the sync-words.

In this paper, we present results on the reduction of the probability of mistaking data for a sync-word for Reed-Solomon (RS) encoded data. The results show that our method reduces the probability of mistaking data for a sync-word in RS codes without the need to significantly increase the lengths of the sync-words. We will use RS codes together with well known sync-words, but modify the RS codes by avoiding some symbols such that the probability of mistaking data for the sync-words is reduced. In our previous work, we developed a method for avoiding particular symbols in a RS code, such that the new code does not have any of the symbols we avoided. We called the method for avoiding symbols in a RS code, *Symbol Avoidance*. In this paper, we use the symbol avoidance method to avoid symbols in a RS code in order to reduce the probability of falsely acquiring the sync-words in the synchronization of RS encoded data.

Our method of synchronization of RS codes using symbol avoidance and sync-words actually borrows from the ideas of synchronizable block codes (prefix codes and comma codes) and conventional frame synchronization with sync-words. Prefix codes and comma codes require that the sync-word should not appear in the codewords or in overlaps of codewords, which implies a probability of mistaking data for a sync-word is zero in the absence of noise. In our synchronization method, we relax this requirement imposed in prefix and comma codes (that is, the zero probability of mistaking data for a sync-word in the absence of noise) by using symbol avoidance to reduce the probability of mistaking data for a sync-word such that it approaches zero in the absence of noise. In some cases, the symbol avoidance method achieves the zero probability of mistaking data for the sync-word, for individual codewords (not overlapping codewords).

II. Preliminaries

A. Aperiodic Auto-correlation of Sequences

Since the pioneering work by Barker [11] in 1953 on frame synchronization, most sequences that were used as sync-words were designed based on aperiodic auto-correlation, and some of these sequences are Barker [11], Turyn [12], Willard [13] and Maury and Styles [14] sequences. It should be noted that all these types of sequences are binary sequences. These sequences have "good" aperiodic Auto-Correlation Functions (ACFs) which makes them "optimal" sync-words. By good aperiodic ACF here we mean that the sequence only produces an aperiodic auto-correlation function that has a high *main lobe* when correlated with zero-shifted version of itself, and minimal *side lobes* with all other shifted versions of itself. Also, the sequence is considered optimal if the side lobes of its aperiodic ACF satisfy a particular required minimal value.

While considering the aperiodic ACF as the criterion for designing/searching for optimal sync-words is a good guideline, there are other ways of looking at the problem of optimizing synchronization with sync-words. One such criterion was presented by Scholtz [15], which is consideration of the probability of false acquisition on data (probability of mistaking data for the sync-word), P_{FAD} and probability of true acquisition on the sync-word.

B. Symbol Avoidance in Reed-Solomon Codes

In our previous work [16], we presented a method for avoiding symbols in Reed-Solomon codes, which we called symbol avoidance. In this subsection, we briefly describe the symbol avoidance method.

Let us define a linear RS code as $(n,\ k,\ d)W$ over GF(q), where n is the length, k is the dimension, d is the minimum Hamming distance and q is the size of the field which is power of a prime. From the linear RS code $(n,\ k,\ d)W$ we produce a new code $(n,\ k',\ d')W'$, of length n, dimension k' and minimum Hamming distance d', over an alphabet of size q'. We call the operation by which W' is produced from W, symbol avoidance. This operation is given in simplified form as

$$(n,\ k,\ d)W \rightarrow \left\{\begin{matrix} Symbol \\ Avoidance \\ Operation \end{matrix}\right\} \rightarrow (n,\ k',\ d')W',$$

where $d' \geq d$, $q' < q$, $k' < k$, and $(n,\ k',\ d')W'$ maybe be non-linear. $q' = q - |A|$, where A is a set of elements/symbols to be avoided in $(n,\ k',\ d')W'$.

Note that the key operation of the symbol avoidance method is about adding two codewords of the original RS code W to form a new codeword without the symbols in A. This new codeword then belongs to W' together with all other codewords without the symbols in A. Since W is a linear code, then it always holds that $W' \subseteq W$. According to the definition of linear codes, W' can be non-linear, but its minimum Hamming distance can be the same as that of W or even greater. Next, we expand a bit on the details of the symbol avoidance operation.

The conventional systematic generator matrix of the RS code $(n,\ k\ d)W$, $G = [I_k|P_{n-k}]$ with the symbols taken from a Galois field GF(q), is decomposed into two parts. The first part of G, which is composed of k' rows of G, will be denoted $G^{k'}$. The second part of G, which is composed of r rows of G such that $k = k' + r$, will be denoted G^r.

$$G = \begin{bmatrix} I_{k'} & 0_r^{k'} & |\ P_{n-k}^{k'} \\ 0_{k'}^r & I_r & |\ P_{n-k}^r \end{bmatrix},$$

where $G^{k'} = [I_{k'}\ 0_r^{k'}|P_{n-k}^{k'}]$ and $G^r = [0_{k'}^r\ I_r|P_{n-k}^r]$.

$G^{k'}$ is used to encode a k'-tuple ($M = m_1 m_2 \ldots m_{k'}$, where $m_i \in$ GF(q)), and this results in a codeword $C = MG^{k'}$. G^r encodes an r-tuple ($V = v_1 v_2 \ldots v_r$, where $v_i \in$ GF(q)), resulting in what we call a *control vector* $R = VG^r$. The difference between C and R will be in their usage, otherwise they are both codewords of the RS code, W. The control vectors (collection of the vectors R) are used to control the presence/absence of a particular symbol(s) in each codeword C. When C has symbols we want to avoid (that is, symbols from set A), we add a suitable R to C forming a new codeword without the symbols we want to avoid. Hence, by so doing, adding C and R, we would have avoided the symbols in A. Obviously, not all control vectors will be suitable for successfully avoiding the symbols from A, when added to the codeword C. However, as long as we have at least a control vector that can successfully avoid symbols from A in C, symbol avoidance is possible. For more detailed information on the symbol avoidance procedure, we refer the reader to [16].

Our symbol avoidance method was inspired by the work of Solomon [17] on alphabet codes and fields of computation, where he showed the following. Given an alphabet (Q) and a field of computation (F) which is prime or power of a prime, where the size of Q is less than that of F, it is possible to form a code over Q from another code over F with the field of computation of the encoding procedure being F. Other related work include [18], [19], [20].

Note: when describing RS codes, we will be using integer symbols to represent elements of GF(q), 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. We will sometimes refer to the RS codes as follows: RS codes without symbol avoidance–W and RS codes with symbol avoidance–W'. 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, where m is the number of bits per symbol.

III. Probability of a Sync-word in a RS Code with and without Symbol Avoidance

In this section, we present numerical results for the probabilities of sync-words in the codewords of the RS codes with symbol avoidance (W'), as well as the corresponding codes W for comparison purposes. We find the probability of the sync-word, of a given number of bits, occurring in a $(2^m - 1, k)$ RS code, given symbols to avoid. These probabilities are then used to indicate which symbols are best to avoid in order to reduce the probability of false acquisition on data, P_{FAD}, for a given sync-word. We use the P_{FAD} of the RS code without symbol avoidance (W) as the benchmark for comparison against the P_{FAD} of W'. All results presented here are for binary sync-words by Maury and Styles [14]. Since the binary sync-words can be very long, for ease of presentation, we will write them in their octal representation.

In the simulations, all the codewords of the RS code are searched for the sync-word *individually*, in the absence of noise. The codewords are the data, but the sync-word itself is not included at the end/beginning of each codeword because

TABLE I

RESULTS OF THE P_{FAD} FOR CORRESPONDING AVOIDED SYMBOLS, FOR BINARY SYNC-WORDS OF LENGTHS $7 - 11$ WRITTEN IN THEIR OCTAL FORMAT. A $(2^3 - 1, 3)$ RS CODE WAS USED FOR THE SYMBOL AVOIDANCE, $|A| = 1$ AND $r = 1$.

(a) Binary sync-word 130, length 7

Probability of False Acquisition	Avoided Symbols
6.80E-03	0
6.80E-03	2
6.80E-03	4
8.16E-03	3
8.16E-03	5
8.16E-03	7
9.52E-03	1
1.09E-02	6

(b) Binary sync-word 270, length 8

Probability of False Acquisition	Avoided Symbols
0	6
1.46E-03	3
1.46E-03	5
2.92E-03	1
2.92E-03	7
4.37E-03	4
5.83E-03	0
5.83E-03	2

(c) Binary sync-word 560, length 9

Probability of False Acquisition	Avoided Symbols
0	3
0	6
1.57E-03	5
3.14E-03	0
3.14E-03	1
3.14E-03	4
3.14E-03	7
6.28E-03	2

(d) Binary sync-word 1560, length 10

Probability of False Acquisition	Avoided Symbols
0	3
0	6
1.70E-03	1
1.70E-03	4
1.70E-03	5
1.70E-03	7
3.40E-03	0
5.10E-03	2

(e) Binary sync-word 2670, length 11

Probability of False Acquisition	Avoided Symbols
0	3
0	5
0	6
1.86E-03	1
1.86E-03	4
1.86E-03	7
3.71E-03	0
5.57E-03	2

we are only interested in the probability of false acquisition on data. Searching the codewords individually for the sync-word means that codewords are treated independently, such that there is no overlap of bits from different codewords.

Tables I and II show results for a $(2^3 - 1, 3)$ RS code, for various binary sync-words with avoided symbols. For each subtable, we show the probabilities of a sync-word occurring in the RS code, under the column "Probability of False Acquisition", and the corresponding avoided symbol(s), under the column "Avoided Symbols". The rows are arranged in ascending order according to the "Probability of False Acquisition". The results were found by avoiding the symbol(s) in the second column and then searching for the sync-word to find its probability of being in the RS code. The limitation to the length of sync-words tested in the results was when all avoided symbols give the lower bound on the probability of false acquisition, $P_{\text{FAD}} = 0$. Therefore all tables not included in the results are those where all the avoided symbols give $P_{\text{FAD}} = 0$.

In Table I only one symbol is avoided for the corresponding RS codes, where $r = 1$. We went further to avoid two symbol, where possible for $r = 1$, in an attempt to achieve $P_{\text{FAD}} = 0$ at shorter sync-word lengths. The results for two avoided symbols are shown in Table II.

Looking at Table I and Table II we can see that some avoided symbols begin to give the desired result of $P_{\text{FAD}} = 0$ at sync-words of lengths 8. This means that avoiding one symbol is good enough to give us the desired result.

A major problem with finding these tables (Tables I and II) is the amount of computer processing power needed to generate all the possibilities of avoided symbols, especially when avoiding more than one symbol. It is also not that practical to present all combinations of avoided symbols as this will result in very long tables. For example, for $m = 6$, if $|A| = 2$ (two symbols avoided), the table will have ${}^{2^m}C_{|A|} = 2016$ rows. For most of our synchronization purposes, we only need to choose from the best performing avoided symbols (the ones at the top of the table). If the avoided symbols have the same P_{FAD}, any can be chosen. To reduce the lengths of very long

TABLE II
RESULTS OF THE P_{FAD} FOR CORRESPONDING AVOIDED SYMBOLS, FOR BINARY SYNC-WORDS OF LENGTHS $7-11$ WRITTEN IN THEIR OCTAL FORMAT. A $(2^3-1,3)$ RS CODE WAS USED FOR THE SYMBOL AVOIDANCE, $|A|=2$ AND $r=1$.

(a) Binary sync-word 130, length 7

Probability of False Acquisition	Avoided Symbols	
1.85E-03	0	5
3.70E-03	0	1
3.70E-03	0	2
3.70E-03	0	3
3.70E-03	0	4
3.70E-03	0	6
3.70E-03	1	3
3.70E-03	1	4
3.70E-03	3	6
3.70E-03	4	6
5.56E-03	1	7
5.56E-03	2	4
5.56E-03	3	5
7.41E-03	1	5
7.41E-03	1	6
7.41E-03	2	3
7.41E-03	2	5
7.41E-03	3	4
7.41E-03	4	5
7.41E-03	4	7
9.26E-03	1	2
9.26E-03	2	6
9.26E-03	2	7
9.26E-03	3	7
9.26E-03	5	6
9.26E-03	6	7
1.11E-02	0	7
1.11E-02	5	7

(b) Binary sync-word 270, length 8

Probability of False Acquisition	Avoided Symbols	
0	1	3
0	1	5
0	1	6
0	2	3
0	2	6
0	3	5
0	3	6
0	4	5
0	4	6
0	5	6
0	6	7
1.98E-03	0	1
1.98E-03	0	6
1.98E-03	3	4
1.98E-03	5	7
3.97E-03	0	3
3.97E-03	0	4
3.97E-03	2	5
3.97E-03	3	7
5.95E-03	0	2
5.95E-03	0	5
5.95E-03	1	2
5.95E-03	1	4
7.94E-03	0	7
7.94E-03	2	4
7.94E-03	4	7
9.92E-03	1	7
1.19E-02	2	7

(c) Binary sync-word 560, length 9

Probability of False Acquisition	Avoided Symbols	
0	0	1
0	0	3
0	0	4
0	0	6
0	1	3
0	1	5
0	1	6
0	2	3
0	2	6
0	3	4
0	3	5
0	3	6
0	3	7
0	4	5
0	4	6
0	5	6
0	6	7
2.14E-03	5	7
4.27E-03	0	7
4.27E-03	2	5
6.41E-03	0	2
6.41E-03	0	5
6.41E-03	1	2
6.41E-03	1	4
6.41E-03	4	7
8.55E-03	1	7
8.55E-03	2	4
1.28E-02	2	7

(d) Binary sync-word 1560, length 10

Probability of False Acquisition	Avoided Symbols	
0	0	1
0	0	3
0	0	4
0	0	6
0	1	3
0	1	5
0	1	6
0	2	3
0	2	6
0	3	4
0	3	5
0	3	6
0	3	7
0	4	5
0	4	6
0	5	6
0	6	7
2.31E-03	5	7
4.63E-03	0	7
4.63E-03	1	2
4.63E-03	1	4
4.63E-03	2	5
4.63E-03	4	7
6.94E-03	0	2
6.94E-03	0	5
6.94E-03	1	7
6.94E-03	2	4
1.16E-02	2	7

(e) Binary sync-word 2670, length 11

Probability of False Acquisition	Avoided Symbols	
0	0	1
0	0	3
0	0	4
0	0	6
0	1	3
0	1	5
0	1	6
0	2	3
0	2	6
0	3	4
0	3	5
0	3	6
0	3	7
0	4	5
0	4	6
0	5	6
0	6	7
2.53E-03	5	7
5.05E-03	0	7
5.05E-03	1	2
5.05E-03	1	4
5.05E-03	2	5
5.05E-03	4	7
7.58E-03	0	2
7.58E-03	0	5
7.58E-03	1	7
7.58E-03	2	4
1.26E-02	2	7

tables, only a few of the avoided symbols giving the best performance is tabulated and presented. Now, when $|A| = 2$, we use the avoided symbols giving the best performance from the results of $|A| = 1$ to form the tables for $|A| = 2$ as follows. From simulation results we found that it is enough to consider the top two symbols for the $|A| = 1$ results to make the table for $|A| = 2$, for the same RS code. Therefore, the shaded symbols in the subtables for $|A| = 2$ in Table II are the top two symbols in the corresponding subtables for $|A| = 1$ in Table I.

TABLE III

COMPARISON OF THE BEST P_{FAD} FOR W' AND P_{FAD} FOR W, FOR BINARY SYNC-WORDS OF LENGTHS $7-11$, WRITTEN IN THEIR OCTAL FORMAT. A $(2^3-1,3)$ RS CODE.

Binary Sync-words	Length, N	P_{FAD} for W	P_{FAD} for W'
130	7	7.81E-03	1.85E-03
270	8	3.91E-03	0
560	9	1.95E-03	0
1560	10	1.95E-03	0
2670	11	1.95E-03	0

Table III gives a summary of the results of the P_{FAD} for the $(2^3-1,3)$ RS code with symbol avoidance (summary of Tables I and II), compared with the P_{FAD} for the $(2^3-1,3)$ RS code without symbol avoidance. Table III gives an indication of the amount of improvement on the P_{FAD} in the code W' from the code W.

For other RS codes with symbol avoidance (W'), instead of showing the complete tables (for example, Tables I and II), we only show the comparison of the best P_{FAD} for codes W' and the P_{FAD} for corresponding codes W, as shown in Table III. Such results are shown in Tables IV, V and VI, for $(2^4-1,3)$ RS code, $(2^3-1,5)$ RS code and $(2^4-1,5)$ RS code, respectively.

TABLE IV

COMPARISON OF THE BEST P_{FAD} FOR W' AND P_{FAD} FOR W, FOR BINARY SYNC-WORDS OF LENGTHS $7-11$, WRITTEN IN THEIR OCTAL FORMAT. A $(2^4-1,3)$ RS CODE.

Binary Sync-words	Length, N	P_{FAD} for W	P_{FAD} for W'
130	7	7.81E-03	4.35E-03
270	8	3.91E-03	2.12E-03
560	9	1.95E-03	1.96E-04
1560	10	9.77E-04	0
2670	11	4.88E-04	0

TABLE V

COMPARISON OF THE BEST P_{FAD} FOR W' AND P_{FAD} FOR W, FOR BINARY SYNC-WORDS OF LENGTHS $7-16$, WRITTEN IN THEIR OCTAL FORMAT. A $(2^3-1,5)$ RS CODE.

Binary Sync-words	Length, N	P_{FAD} for W	P_{FAD} for W'
130	7	7.81E-03	3.40E-03
270	8	3.91E-03	1.10E-03
560	9	1.95E-03	0
1560	10	9.77E-04	0
2670	11	4.88E-04	0
6540	12	2.44E-04	0
16540	13	1.22E-04	0
34640	14	6.10E-05	0
73120	15	3.05E-05	0
165620	16	3.05E-05	0

TABLE VI

COMPARISON OF THE BEST P_{FAD} FOR W' AND P_{FAD} FOR W, FOR BINARY SYNC-WORDS OF LENGTHS $7-20$, WRITTEN IN THEIR OCTAL FORMAT. A $(2^4-1,5)$ RS CODE.

Binary Sync-words	Length, N	P_{FAD} for W	P_{FAD} for W'
130	7	7.81E-03	6.27E-03
270	8	3.91E-03	3.23E-03
560	9	1.95E-03	1.44E-03
1560	10	9.77E-04	7.36E-04
2670	11	4.88E-04	2.53E-04
6540	12	2.44E-04	1.21E-04
16540	13	1.22E-04	6.13E-05
34640	14	6.10E-05	2.90E-05
73120	15	3.05E-05	2.23E-05
165620	16	1.53E-05	9.61E-06
363240	17	7.63E-06	4.49E-06
746500	18	3.81E-06	4.59E-07
1746240	19	1.91E-06	4.70E-07
3557040	20	9.53E-07	0

The results in Tables IV, V and VI also show the amount of improvement on the P_{FAD} in the code W' from the code

W. The improvement in P_{FAD} due to symbol avoidance has to be weighed against the loss in efficiency (reduced code rate) incurred from W to W'. To demonstrate the efficiencies of the RS codes with and without symbol avoidance, in terms of redundant bits we use the following expressions. The redundancy, in bits, of a $(2^m-1,k)$ RS code is given by $R = m(2^m-1-k)$. We denote by R_X the redundancy of the $(2^m-1,k)$ RS code including an X-bit sync-word, such that $R_X = R + X$,

$$R_X = m(2^m-1) - mk + X.$$

For the same $(2^m-1,k)$ RS code with symbol avoidance, W', we denote the redundancy including a Y-bit sync-word by R_Y, such that

$$R_Y = m(2^m-1) - \lfloor (k-r)\log_2(2^m - |A|) \rfloor + Y.$$

The parameter r is the number of rows no longer used to encode information in the generator matrix of the RS code, and $|A|$ is the number of symbols to be avoided from the set of 2^m symbols.

As an example, in Table V, the code W' requires only $Y = 9$ bits to achieve $P_{\text{FAD}} = 0$, which means

$$R_Y = 3(2^3-1) - \lfloor (5-1)\log_2(2^3-2) \rfloor + 9 = 20 \text{ bits},$$

and the code W still does not achieve $P_{\text{FAD}} = 0$ even with $X = 16$ bits, and the redundancy is

$$R_X = 3(2^3-1) - 3 \times 5 + 16 = 22 \text{ bits}.$$

IV. Conclusion

We have shown a way of reducing the probability of false acquisition for Reed-Solomon encoded data, where sync-words are used to mark the start or end of a frame. We used a method called symbol avoidance, which was applied to Reed-Solomon codes to remove specific symbols in the codewords. The removal of the symbols in the codewords reduced the probability of mistaking a portion of the codewords for the sync-word used in the synchronization. We mainly presented results comparing the probabilities of false acquisition for Reed-Solomon codes with and without symbol avoidance applied. The results showed an improvement in the probability of false acquisition when symbol avoidance is applied to the Reed-Solomon codes. However, this improvement needs to be carefully weighed against the loss in efficiency incurred in applying symbol avoidance. This is similar to the trade-off between efficiency and error correcting capabilities of a code in coding theory.

References

[1] S. W. Golomb, B. Gordon, and L. R. Welch, "Comma-free codes," *Canadian Journal of Mathematics*, vol. 10, no. 2, pp. 202–209, 1958.

[2] B. H. Jiggs, "Recent results in comma-free codes," *Canadian Journal of Mathematics*, vol. 15, pp. 178–187, 1963.

[3] W. L. Eastman, "On the construction of comma-free codes," *IEEE Transactions on Information Theory*, vol. 11, no. 2, pp. 263–267, Apr. 1965.

[4] W. B. Kendall and I. S. Reed, "Path invariant comma-free codes," *IEEE Transactions on Information Theory*, vol. 8, no. 6, pp. 350–355, Oct. 1962.

[5] E. N. Gilbert, "Synchronization of binary messages," *IEEE Transactions on Information Theory*, vol. 6, no. 4, pp. 470–477, Sept. 1960.

[6] L. J. Guibas and A. M. Odlyzko, "Maximal prefix-synchronized codes," *SIAM Journal of Applied Mathematics*, vol. 35, no. 2, pp. 401–418, Sept. 1978.

[7] A. J. van Wijngaarden and H. Morita, "Extended prefix synchronization codes," in *Proceedings of the 1995 IEEE International Symposium on Information Theory*, Whistler, BC, Canada, 1995, p. 465.

[8] H. Morita, A. J. van Wijngaarden, and A. J. H. Vinck, "On the construction of maximal prefix-synchronized codes," *IEEE Transactions on Information Theory*, vol. 42, no. 6, pp. 2158–2166, Nov. 1996.

[9] W. B. Kendall, "Optimum synchronizing words for fixed word-length code dictionaries," in *Space Program Summary*. Pasadena, California: Jet Propulsion Laboratory, California Institute of Technology 4, 1964, p. 37.

[10] J. J. Stiffler, "Instantaneously synchronizable block code dictionaries," in *Space Program Summary*. Pasadena, California: Jet Propulsion Laboratory, California Institute of Technology 4, 1965, p. 268.

[11] R. H. Barker, "Group synchronizing of binary digital systems," in *Communication Theory*. W. Jackson. Ed. New York: Academic-Butterworth, 1953.

[12] R. Turyn, "Sequences with small correlation," in *Error Correcting Codes*. H. B. Mann, Ed. New York: Wiley, 1968, pp. 195–228.

[13] M. W. Willard, "Optimum code patterns for pcm synchronization," in *Proceedings of the 1962 National Telemetering Conference*, 1962.

[14] J. L. Maury and F. Styles, "Development of optimum frame synchronization codes for goddard space flight center pcm telemetry standards," in *Proceedings of the 1965 National Telemetering Conference*, 1965.

[15] R. A. Scholtz, "Frame synchronization techniques," *IEEE Transactions on Communications*, vol. 28, no. 8, pp. 1782–1789, Aug. 1980.

[16] T. Shongwe and A. J. H. Vinck, "Reed-solomon code symbol avoidance," *SAIEE Africa Research Journal*, submitted.

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

[18] ——, "Non-linear, non-binary cyclic group codes," in *Proceedings of the 1993 IEEE International Symposium on Information Theory*, San Antonio, TX, USA, 1993, pp. 192–192.

[19] M. Hattori, R. J. McEliece, and G. Solomon, "Subspace subcodes of Reed-Solomon codes," *IEEE Transactions on Information Theory*, vol. 44, no. 5, pp. 1861–1880, 1998.

[20] A. Urivskiy, "On subset subcodes of linear codes," in *Problems of Redundancy in Information and Control Systems (RED), 2012 XIII International Symposium on*, Moscow, Russia, 2012, pp. 89–92.