Preamble-Less Group Synchronization of Binary Linear Block Codes

Gheorghe Zaharia

Preamble-Less Group Synchronization of Binary Linear Block Codes

Gheorghe ZAHARIA
Université Européenne de Bretagne, INSA, IETR - UMR CNRS 6164, Rennes, France
gheorghe.zaharia@insa-rennes.fr

***Abstract*—This paper addresses the possibility to realize the group synchronization of short linear, binary, block codes without the use of a preamble. The proposed algorithm is based on the cyclic use of the coding relations. The input data are equal probable and generated by a binary, memoryless source. We consider a binary symmetric channel (BSC) with known error probability. It is shown that the reliability of the proposed algorithm depends on the channel error probability and the length of the cumulative sums used for synchronization.**

Keywords: linear block codes • synchronization • coding relations • binary symmetric channel • error probability

I. INTRODUCTION

In several applications as transmission of medical parameters obtained by measurements [1-2], short messages from vehicles to infrastructure in V2I configurations [3] or in the case of several wireless sensor networks (WSN) [4] the amount of data is reduced. In order to obtain a reliable communication, error-correcting codes like linear block codes can be used [5-6]. The main idea of these codes is to add some redundancy by appending to k binary information symbols m parity-check symbols in order to increase the Hamming distance between the resulting code words of length $n=k+m$. These m parity-check symbols are computed as linear combinations of the k information symbols. Thus, the code rate is k/n. The received data must be divided in code words. Generally, this group synchronization can be done using a periodic preamble [7-8]. In order to avoid the reduction of the code rate due to the periodic preamble, for small values of n it is interesting to investigate to possibility to exploit the periodicity of encoding relations of a linear block code (LBC) for the group synchronization. Indeed, in a previous study [9] it has been shown that a LBC can be characterized as an information source with memory if the input data are generated by a memoryless binary source. In [10], the LBCs have been characterized as cyclic Markov chains with a period equal to n. In this paper, we study the performance of the group synchronization based on the period of the cyclic Markov chain corresponding to a given LBC. The encoding process is shown in Fig. 1.

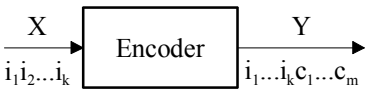

Fig. 1 Encoding process

Generally, for a linear block code, the encoding operation is performed according to relation [5–6]:

$$v = [i_1 i_2 \ ... \ i_k]G \tag{1}$$

where:

v is the code word;

i_j, $j = \overline{1,k}$ are the information symbols;

G is the generator matrix.

The number of rows of the generator matrix G is equal to the number of the information symbols k, while the number of columns is equal to the code word length n.

Since from (1) 2^k distinct code words have to result, the rank of the generator matrix G must be equal to k. This means that, by elementary transformations, the generator matrix can be expressed in the equivalent form:

$$G = \left[I_k \ P \right] \tag{2}$$

where I_k denotes the identity matrix of rank k and P is a k x m matrix with binary elements. If the generator matrix is as in (2), the code is systematic, with the information symbols placed on the first k positions of the code word, that is:

$$v = [i_1 i_2 ... i_k c_1 c_2 ... c_m] \tag{3}$$

where $c_j \in \{0,1\}$, $j = \overline{1,m}$, are the parity-check symbols.

We assume that the binary information symbols are equal probable and provided by the memoryless source $X=\{0, 1\}$. The parity-check symbols can be computes as:

$$\begin{aligned} c_1 &= i_1 p_{11} \oplus i_2 p_{21} \oplus ... \oplus i_k p_{k1} \\ c_2 &= i_1 p_{12} \oplus i_2 p_{22} \oplus ... \oplus i_k p_{k2} \\ &\cdots\cdots\cdots\cdots\cdots\cdots \\ c_m &= i_1 p_{1m} \oplus i_2 p_{2m} \oplus ... \oplus i_k p_{km} \end{aligned} \tag{4}$$

In order to perform the group synchronization even when the received data is absent (*i.e.*, all the information symbols are equal to 0), the proposed algorithm uses the relations:

$$\begin{aligned} c_1 &= \overline{i_1 p_{11} \oplus i_2 p_{21} \oplus ... \oplus i_k p_{k1}} \\ c_2 &= \overline{i_1 p_{12} \oplus i_2 p_{22} \oplus ... \oplus i_k p_{k2}} \\ &\cdots\cdots\cdots\cdots\cdots\cdots \\ c_m &= \overline{i_1 p_{1m} \oplus i_2 p_{2m} \oplus ... \oplus i_k p_{km}} \end{aligned} \tag{5}$$

This inversion of the logical value of the parity-check symbols does not modify the correction properties of the considered LBC.

II. SYNCHRONIZATION ALGORITHM

The main idea of the proposed algorithm is to exploit the periodicity of the encoding relations (5). At the receiver, without errors, we compute:

$$\begin{aligned} s_1 &= i_1 p_{11} \oplus i_2 p_{21} \oplus \ldots \oplus i_k p_{k1} \oplus c_1 \\ s_2 &= i_1 p_{12} \oplus i_2 p_{22} \oplus \ldots \oplus i_k p_{k2} \oplus c_2 \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ s_m &= i_1 p_{1m} \oplus i_2 p_{2m} \oplus \ldots \oplus i_k p_{km} \oplus c_m \end{aligned} \tag{7}$$

where $\oplus$ is the modulo-2 addition. It is simple to verify that in this case all the binary symbols $s_j = 1$ ($j = 1, 2, \ldots, k$), thus:

$$S = \sum_{j=1}^{m} s_j = m. \tag{8}$$

Let us consider a sequence of $2n$-1 successive received bits:

$$r = [r_1\ r_2 \ldots r_{2n-1}] \tag{9}$$

For $d = 1, 2, \ldots, n$ we compute:

$$\begin{aligned} s_1(d) &= r_d p_{11} \oplus r_{d+1} p_{21} \oplus \ldots \oplus r_{d+k-1} p_{k1} \oplus r_{d+k} \\ s_2(d) &= r_d p_{12} \oplus r_{d+1} p_{22} \oplus \ldots \oplus r_{d+k-1} p_{k2} \oplus r_{d+k+1} \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ s_m(d) &= r_d p_{1m} \oplus r_{d+1} p_{2m} \oplus \ldots \oplus r_{d+k-1} p_{km} \oplus r_{d+k+m+1} \end{aligned} \tag{10}$$

If r_d is the first symbol of a received code word, then, in the absence of errors, we have for $d \in \{1, 2, \ldots, n\}$:

$$S(d) = \sum_{j=1}^{m} s_j(d) = m \tag{11}$$

while for $d' \in \{1, 2, \ldots, n\}$ and $d' \neq d$:

$$S(d') = \sum_{j=1}^{m} s_j(d') \leq m. \tag{12}$$

Let us consider the time evolution of these sums. In the absence of errors will have $S(d+jn) = m$ and $S(d'+jn) \leq m$ for every positive integer value j. For several values of j, due to the random generation of the information values, it is possible to obtain $S(d'+jn) < m$. This observation indicates that for a large enough value P_s, the cumulative sums:

$$CS(P_s, d) = \sum_{j=1}^{P_s} S(d) = mP_s \tag{13}$$

and:

$$CS(P_s, d') = \sum_{j=1}^{P_s} S(d') < mP_s \tag{14}$$

will be quite different and the difference:

$$Diff(P_s) = CS(P_s, d) - \max\{CS(P_s, d')\}, \quad d' \neq d \tag{15}$$

will be quite large to allow a reliable determination of d even in the presence of some errors. Thus, the first position r_d of a received code word can be determined. The value P_s depends on the LBC parameters, the error probability p characterizing the transmission channel and the false alarm (false detection) probability. Its value can be determined by simulation.

III. ALGORITHM SIMULATION FOR H(7,4)

We consider the well-known binary (7,4) Hamming code with $k = 4$ information symbols and $m = 3$ parity-check symbols. This code can correct one error. In order to simplify the simulation program, a systematic version of this code is used. Thus, for a given code word:

$$v = [v_1\ v_2\ v_3\ v_4\ v_5\ v_6\ v_7] \tag{16}$$

we consider the first $k = 4$ bits as information symbols, while the last $m = 3$ bits as parity-check symbols given by:

$$\begin{aligned} v_5 &= v_1 \oplus v_2 \oplus v_3 \\ v_6 &= v_1 \oplus v_2 \oplus v_4 \\ v_7 &= v_1 \oplus v_3 \oplus v_4 \end{aligned} \tag{17}$$

Based on these relations, the generator matrix of this code can be obtained:

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \tag{18}$$

As explained in introduction, we prefer to inverse the coding relations (17):

$$\begin{aligned} v_5 &= \overline{v_1 \oplus v_2 \oplus v_3} \\ v_6 &= \overline{v_1 \oplus v_2 \oplus v_4} \\ v_7 &= \overline{v_1 \oplus v_3 \oplus v_4} \end{aligned} \tag{19}$$

For this code, $d = 1, 2, \ldots, 7$. The symbols $s_j(d)$ are:

$$\begin{aligned} s_1(d) &= v_d \oplus v_{d+1} \oplus v_{d+2} \oplus v_{d+4} \\ s_2(d) &= v_d \oplus v_{d+1} \oplus v_{d+3} \oplus v_{d+5} \\ s_3(d) &= v_d \oplus v_{d+2} \oplus v_{d+3} \oplus v_{d+6} \end{aligned} \tag{20}$$

For the simulation, we consider a noisy channel with the error probability $p = 10^{-2}$. In fact, for each code word v it is possible to obtain an error vector

$$e = [e_1\ e_2\ e_3\ e_4\ e_5\ e_6\ e_7] \tag{21}$$

The received code word v_r is calculated as the module-2 sum between the transmitted vector v and the error vector e:

$$v_r = v \oplus e \tag{22}$$

Each symbol of the error vector e is generated with the error probability p. This can be done using a uniform random variable $u \in [0, 1[$. If $0 \leq u \leq 1\text{-}p$, then we consider $e_j = 0$, else $e_j = 1$.

The results shown in Fig. 2 are obtained with $P_s = 24$. The number of code words simulated is $N = 10^5$. The frame considered for simulation began with a code word. Therefore, the decision $d = 1$ indicates that that the first received bit r_1 is the start of a code word. During the reception, there was no decision modification, so there were no missing bits and the group synchronization was performed without errors.

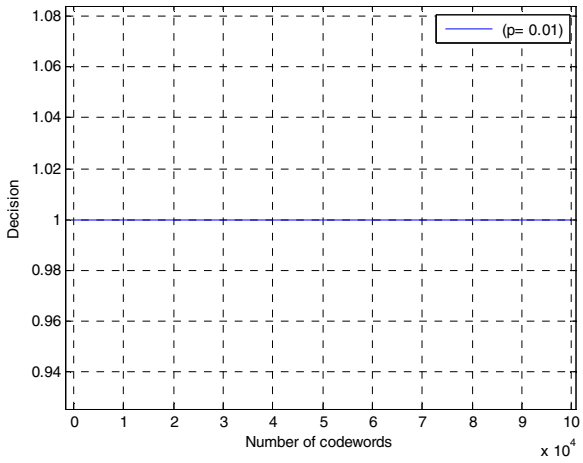

Fig. 2 H(7,4): Evolution of decision d for $p = 10^{-2}$

Fig. 3 presents the time evolution of the difference $Diff(P_s)$ computed with (13) and (24). This difference is sufficient large to take each time the good decision. However, the capacity of this algorithm to take good decisions is limited.

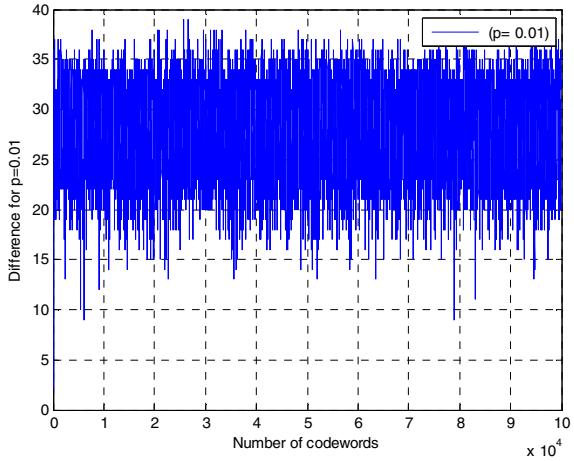

Fig. 3 Evolution of the difference $Diff(P_s)$

For example, for $p = 4.10^{-2}$, as shown in Fig. 4, this algorithm has some wrong decisions.

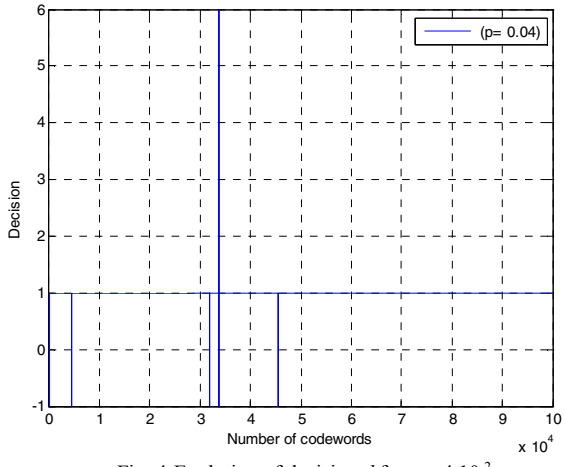

Fig. 4 Evolution of decision d for $p = 4.10^{-2}$

The value $d = 6$ shows that for a bad enough channel, it is possible to obtain the maximum value of the cumulative sums $CS(d)$ for a wrong value of d (as shown before, because the received vector r starts with the first bit of a code word, the right decision for these simulations is $d = 1$).

It is also possible to observe in Fig. 4 several values $d = -1$. This indicates that the maximum value of the cumulative sums $CS(d)$, with $d = 1, 2, ..., n$ is obtained for at least 2 different values of d. In this case, the decision cannot be correctly taken. However, it is possible to remark that the values $d = -1$ are isolated (before and after a value $d = -1$ one can observe good decisions $d = 1$). This remark suggests an improvement of the basic synchronisation algorithm.

IV. ALGORITHM SIMULATION FOR H(15,11)

In order to determine the performance of the proposed algorithm, a longer code is conserved for simulation: the Hamming code with a length $n = 15$ and $k = 11$ information symbols. As for the previous code, this code can correct just one error. Let consider:

$$v = [v_1\, v_2\, ...,\, v_{15}] \tag{21}$$

the code word with the information symbols placed on the first $k = 11$ positions. The last $m = 4$ parity-check symbols can be computed as:

$$\begin{aligned} v_{12} &= v_1 \oplus v_2 \oplus v_4 \oplus v_5 \oplus v_7 \oplus v_9 \oplus v_{11} \\ v_{13} &= v_1 \oplus v_3 \oplus v_4 \oplus v_6 \oplus v_7 \oplus v_{10} \oplus v_{11} \\ v_{14} &= v_2 \oplus v_3 \oplus v_4 \oplus v_8 \oplus v_9 \oplus v_{10} \oplus v_{11} \\ v_{15} &= v_2 \oplus v_6 \oplus v_7 \oplus v_8 \oplus v_9 \oplus v_{10} \oplus v_{11} \end{aligned} \tag{22}$$

As explained in introduction, we prefer to inverse the coding relations (22):

$$\begin{aligned} v_{12} &= \overline{v_1 \oplus v_2 \oplus v_4 \oplus v_5 \oplus v_7 \oplus v_9 \oplus v_{11}} \\ v_{13} &= \overline{v_1 \oplus v_3 \oplus v_4 \oplus v_6 \oplus v_7 \oplus v_{10} \oplus v_{11}} \\ v_{14} &= \overline{v_2 \oplus v_3 \oplus v_4 \oplus v_8 \oplus v_9 \oplus v_{10} \oplus v_{11}} \\ v_{15} &= \overline{v_2 \oplus v_6 \oplus v_7 \oplus v_8 \oplus v_9 \oplus v_{10} \oplus v_{11}} \end{aligned} \tag{23}$$

For this code, $d = 1, 2,...,15$. The symbols $s_l(d)$ are:

$$\begin{aligned} s_1(d) &= v_d \oplus v_{d+1} \oplus v_{d+3} \oplus v_{d+4} \oplus v_{d+6} \oplus v_{d+9} \oplus v_{d+10} \oplus v_{d+11} \\ s_2(d) &= v_d \oplus v_{d+2} \oplus v_{d+3} \oplus v_{d+5} \oplus v_{d+6} \oplus v_{d+9} \oplus v_{d+10} \oplus v_{d+12} \\ s_3(d) &= v_{d+1} \oplus v_{d+2} \oplus v_{d+3} \oplus v_{d+7} \oplus v_{d+8} \oplus v_{d+9} \oplus v_{d+10} \oplus v_{d+13} \\ s_4(d) &= v_{d+4} \oplus v_{d+5} \oplus v_{d+6} \oplus v_{d+7} \oplus v_{d+8} \oplus v_{d+9} \oplus v_{d+10} \oplus v_{d+14} \end{aligned} \tag{24}$$

For the simulation, we consider a noisy channel with the error probability $p = 10^{-2}$ and $P_s = 28$. The number of code words simulated is $N = 10^5$. The frame considered for simulation began with a code word. One can observe in Fig. 5 a good result obtained with $P_s = 28$. However, the first decisions are not correct. This is an expected result, because the cumulative sums given by (13) are computed with few terms $S(d)$. However, a small number of terms $S(d)$ (Fig. 6 indicates 2 terms) are sufficient to take a good decision.

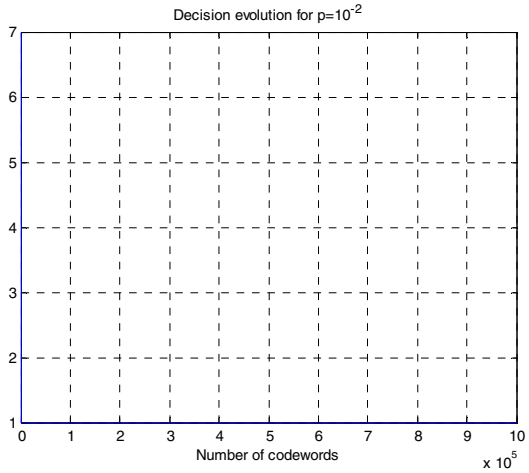

Fig. 5 H(15,11): Evolution of decision d for $p = 10^{-2}$

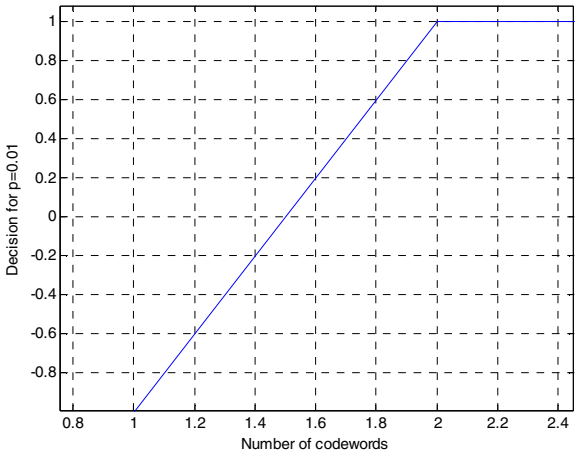

Fig. 6 H(15,11): Start of the evolution of decision d for $p = 10^{-2}$

Hence, after a transient regime, the group synchronization is realized without errors.

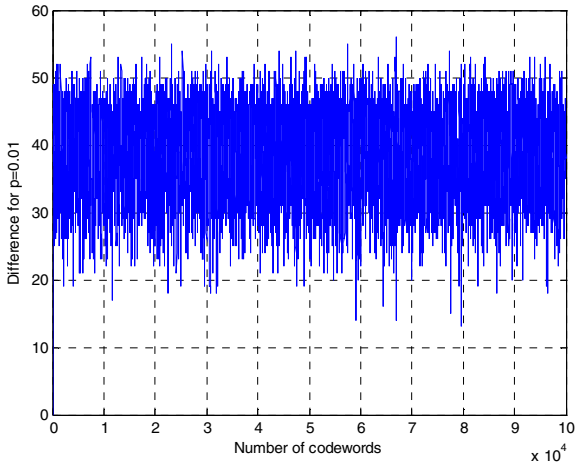

Fig. 7 H(15,11): Evolution of the difference $Diff(P_s)$

Fig. 7 presents the time evolution of the difference $Diff(P_s)$ computed with (25). One can observe that this difference is sufficient large to take each time the good decision. This is possible after a short transient regime (some few code words).

With $p = 2.10^{-2}$, these differences are smaller. Therefore, sometimes, some synchronization decisions are wrong. In this case, a larger value $P_r = 30$ is sufficient to obtain a good synchronization (as given in Fig. 5).

IV. CONCLUSION

A new and original non-preamble group synchronization based on coding relations has been proposed in this paper. With a convenient choice of length of the cumulative sums given by the parameter P_r, this algorithm gives good results even for quite bad transmission channels (error probability of 10^{-2}). The main advantage is the reliability of the decisions of the proposed algorithm without using a preamble and, therefore, without further reduction of the coding rate. However, for practical applications, this algorithm is convenient only for short code words. If the length of the code word increases, the number of electronic circuits needed for the implementation becomes quite large. However, for the actual FPGAs or ASICs, this can be easily realized.

The future work will focus on the convenient choice of the parameter P_r as a function of p and n for a given low value of the miss detection probability. It will be also interesting to confirm the simulation results by theoretical calculations. The theoretical values of the synchronization miss-detection and false-alarm probabilities must also be considered.

REFERENCES

[1] D. Obeid, S. Sadek, G. Zaharia, and G. El Zein, "Non-Contact Heartbeat Detection at 2.4, 5.8 and 60 GHz: A Comparative Study," *Microwave Opt. Technol. Lett.*, vol. 51, no. 3, pp. 666-669, March 2009.

[2] D. Obeid, S. Sadek, G. Zaharia, and G. El Zein, "Multi-tunable Microwave System for Touch-less Heartbeat Detection and Heart Rate Variability Extraction," *Microwave Opt. Technol. Lett.*, vol. 52, no. 1, pp. 192-198, Jan. 2010.

[3] H. Kdouh, G. Zaharia, C. Brousseau, G. El Zein, G. Grunfelder, "ZigBee-Based Sensor Network for Shipboard Environments", *Proc. of 10th IEEE ISSCS 2011,* 29 June - 1 July 2011.

[4] O. Berder, P. Quemerais, O. Sentieys, J. Astier, T. Nguyen, J. Menard, G. Le Mestre, Y. Le Roux, Y. Kokar, G. Zaharia, R. Benzerga, X. Castel, M. Himdi, G. El Zein, S. Jegou, P. Cosquer, and M. Bernard, "Cooperative communications between vehicles and intelligent road signs," in *Proc. 8th Int. Conf. ITST*, Oct. 2008, pp. 121–126.

[5] A. Neubauer, J. Freudenberger, V. Kühn, "Coding Theory", Wiley, 2007

[6] R. H. Morelos-Zaragoza, "The Art of Error Correcting Coding", Wiley, 2006

[7] R. A. Pacheco and D. Hatzinakos, "Analysis of a frame synchronization method using periodic preamble for OFDM based WLANs", *Proc. of IEEE PIMRC*, Barcelona, 5-8 Sept. 2004, Vol. 3, pp. 1911-1915

[8] L. Rakotondrainibe, Y. Kokar, G. Zaharia, G. Grunfelder, G. El Zein, "Performance Analysis of a 60 GHz near gigabit system for WPAN applications", *Proc. of IEEE PIMRC*, Istanbul, 2010, pp. 1038-1043

[9] V. Munteanu, D. Tarniceriu, G. Zaharia, "Analysis of linear block codes as sources with memory", *Advances in Electrical and Computer Engineering*, Vol. 10, No. 4, 2010, pp. 77-80

[10] G. Zaharia, V. Munteanu, D. Tarniceriu, "Characterization of linear block codes as cyclic sources with memory", *Proc. of IEEE ISSCS 2011,* 29 June - 1 July 2011