14th European Signal Processing Conference (EUSIPCO 2006), Florence, Italy, September 4-8, 2006, copyright by EURASIP

BLIND FRAME SYNCHRONISATION FOR BLOCK CODE

S. Houcke and G. Sicot

Dept Signal and Communication. ENST-Bretagne
Technople Brest Iroise - CS 83818 - 29238 Brest Cedex 3 FRANCE
TAMCIC (CNRS 2658) email:{sebastien.houcke}{guillaume.sicot} @enst-bretagne.fr

ABSTRACT

Canal coding can not nowadays be by passed. In order to decode the coded sequence, the receptor has to find the beginning of the codewords. This problem is usually solved by adding periodically to the transmit sequence a frame synchronization sequence. Of course the longer the sequence the better the synchronization but the less the spectral efficiency. We understand clearly the stakes of developing a blind technique that synchronizes before decoding (at high bit error rate) without synchronization sequence. In this article we propose a blind method that allows us to synchronize a block code, we show that it is specially well suited for the LDPC codes and has for those particular code very convincing performance.

1. INTRODUCTION AND NOTATION

Communication and information storage use more and more numerical solutions. Those techniques use a canal code more or less sophisticated. Many works have been done in this field. The principle consists in adding redundant bits to the information bits. This controlled redundancy allows the receptor to detect the presence of errors and eventually to correct them.

Along the existing codes, we focus in this paper on the block codes. They associate to an information word of n_b bits a code word of n_c bits with $n_c > n_b$. The more redundant bits we have, the more powerful the code is. In order to decode, the receptor needs to find the beginning of the codewords. This operation is usually called frame synchronization [1], [2], [3]. The goal of this paper is to present a blind method that allows to synchronize the frame without synchronization bits (blind approach). This method has to work before decoding (at high Bit Error Rate). Developing blind methods has an important economic stake and is a challenging scientific problem.

1.1 Notation

A block encoder is defined by a full-rank generator matrix G that transforms each block of n_b information bits into n_c encoded bits ($n_b < n_c$). Representing the i^{th} information block and the i^{th} encoded block by vectors b_i and y_i, we have: $y_i = b_i G$. y_i is called a codeword. The ratio $r = n_b/n_c$ is called the code rate. The $n_r = n_c - n_b$ redundant bits are computed as the sum modulo 2 of some information bits.

The receiver received the codeword eventually corrupted with errors due to the propagation channel. Let denote by r the received codeword:

$$r = m_c \oplus e$$

where m_c is the codeword, e an error vector of length n_c and $\oplus$ stands for the sum modulo 2. From this observation, the receiver should be able to restore the n_b information bits. The optimal decoding from binary data consists in finding the code word that minimizes the Hamming distance to the observation r.

To the generator matrix G corresponds a parity check matrix H of size $n_r n_c$ such that $GH^T = 0$. The decoding principle consists in computing the syndrome $s(r)$ of the observation:

$$s(r) = rH^T = m_i GH^T \oplus eH^T = eH^T.$$

And for each syndrome, we associate an error word privileging the low weight error word.

Until now, we implicitly assumed that the receptor knows the beginning of the code words. Unfortunately this is usually not the case. Let us denote by $\mathbf{X}$ the received sequence of $\mathbf{Z}$. Because of the propagation channel, $\mathbf{X}$ is a delayed replica of $\mathbf{Z}$ (by t_0 bits corresponding to the propagation delay) that has been passed through a binary symmetric channel. Let us denote p_e the error probability of the channel. Without loss of generality, we assume that the restitution delay t_0 is smaller than the size n_c of a codeword. The goal of frame synchronization is to estimate t_0.

2. PRINCIPLE OF OUR BLIND SYNCHRONIZATION

The redundancy introduced by the code is used to synchronize our receptor. Indeed for a noise free channel, when we are synchronised, all syndromes computed from blocks of size n_c are equal to zero. This is mostly not the case when we are not synchronized. Our method is based on this obvious observation. Let's H_d be an extracted sequence of size

Kn_c from the received sequence $\mathbf{X}$:

$$H_d = \Big[\underbrace{x(d),\ldots,x(d+n_c-1)}_{B_1},x(d+n_c),\ldots,$$
$$\underbrace{x(d+(K-1)n_c),\ldots,x(d+Kn_c-1)}_{B_K}\Big]^T.$$

H_d can be divided into K blocks $(B_i)_{k=1,\ldots,K}$ of size n_c. When $d=t_0$, H_d has exactly K complete codewords. From H_d, we define a vector of syndromes S_d of size Kn_r.

$$S_d = [S_d^{(1)},\ldots,S_d^{(K)}]^T = [S_d(1),\ldots,S_d(Kn_r)]^T$$

where $S_d^{(i)}$ is the syndrome computed from the block B_i of H_d and $S_d(k)$ is the k^{th} element of S_d.

Figure 1 represents three different sequences of H_d : for $d=0$, $d=1$ and $d=t_0$ where the size of H_d is fixed to $K=3$.

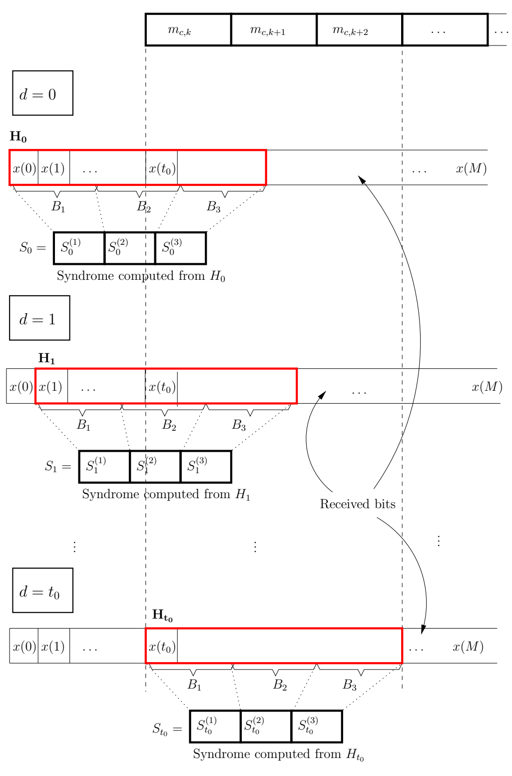

Figure 1: Blind synchronization principle

From S_d, we define two different functions, the first one $\phi_1(d)$ corresponds to the number of syndrome equal to zero in S_d :

$$\phi_1(d) = \text{ Card}\left\{i\in\{1,\ldots,K\}/S_d^{(i)}=[0,\ldots,0]\right\}.$$

We estimate the synchronized position as the one that maximizes $\phi_1(d)$. Notice that for a small window H_d, for example $K=1$, and for cyclic codes (like BCH), if we have a syndrome equal to zero at position d, we have a probability of 0.5 to have a syndrome null at position $d+1$. Thus the cyclic property of the code has dramatic influences on our synchronization procedure. Nevertheless this can be easily solved by interleaving the emitted sequence by an interleaver of the size of the synchronization windows. Indeed the interleaver destroys the cyclic property and our method has then much better performance as shown in the simulation section.

Unfortunately, this synchronization method based on ϕ_1 does not work for long code words. In presence of errors, $\phi_1(d)$ is a Bernoulli variable with parameter p_1 for the synchronized position and p_2 for the other positions. It is easily seen that :

$$p_1 = (1-p_e)^{n_c} + \sum_{i=2}^{2^{n_i}} p_e^{d_i}(1-p_e)^{n_c-d_i}$$

with d_i the hamming distance between the emitted code word and the i^{th} code word. For p_2 we have : $p_2 = \frac{2^{n_i}}{2^{n_c}}$. Of course the performance of this method depends on the BER and the length of a code word: for a fix code rate, we have :

$$\lim_{n_c\to\infty} p_1 = \lim_{n_c\to\infty} p_2 = 0.$$

Which means that for long codewords we are not able to dissociate the synchronized position from the others.

Another way to take advantage of S_d is to compute $\phi_2(d)$ as the number of elements equal to zero in the vector S_d:

$$\phi_2(d) = \sum_{k=1}^{K(n_c-n_b)} S_d(k)$$

and to estimate the synchronized position as the one that minimizes ϕ_2

$$\hat{t}_0 = \text{ArgMin}_{d=0,\ldots,n_c-1}\phi_2(d).$$

2.1 property of ϕ_2

For $d\neq t_0$, assuming that the bits are uniformly distributed, we have:

$$P[S_d(k)=1] = \frac{1}{2} \quad \text{for } d\neq t_0.$$

For $d=t_0$, $P[S_{t_0}(k)=1]$ depends on the number of ”ones” in the column k of H^T and of the BER. Assuming that column k of the matrix H^T has u_k ”ones”, we have :

$$P[S_{t_0}(k)=1] = \sum_{l=0}^{\lfloor\frac{u_k}{2}\rfloor-1}\binom{u_k}{2l+1}p_e^{2l+1}(1-p_e)^{u_k-2l-1}.$$

Figure 2 represents the evolution of $P[S_{t_0}(k)=1]$ versus u for different values of p_e. The function $u \rightarrow P[S_{t_0}(k)=1]$ tends to $\frac{1}{2}$ when u grows. It grows faster for bigger p_e. Thus the synchronisation procedure should be more efficient as soon as the difference between $(P[S_d(k)=1])_{d \neq t_0}$ and $P[S_{t_0}(k)=1]$ is big enough. This is the case for LDPC codes [4] [5]. Note that this condition does not depend on the size of the code word as this is the case for the synchronisation procedure based on the maximisation of ϕ_1.

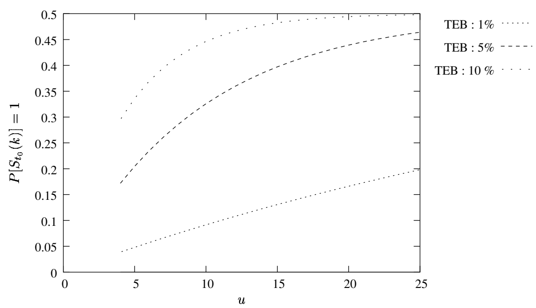

Figure 2: $P[S_{t_0}(k)=1]$ versus u for different values of p_e.

Nevertheless this difference between $\{P[S_d(k)=1]\}_{d \neq t_0}$ and $P[S_{t_0}(k)=1]$ is not enough to guarantee a good performance of the synchronization procedure based on the minimization of ϕ_2. Indeed, to be relevant, the elements of S_d have to be independent.

Figure 3 presents the probability that element k and element l of a syndrome are independent. To compute this probability, we assumed that all the lines of H have the same number u of "ones" and that the position of the "one" in the lines of H are uniformly chosen. The smaller u, the more independent the elements of the syndrome. Once again the LDPC codes are good candidates for our synchronization procedure based on ϕ_2.

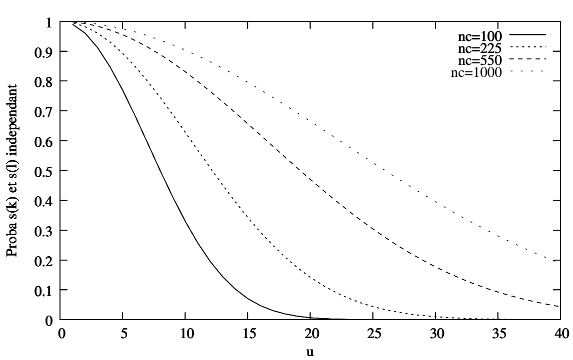

Figure 3: Probability that element k and element l of a syndrome are independent

Assuming that the element of the syndromes are independent and that all lines of H have the same number of "ones", then $\phi_2(d)$ follows a Bernoulli law of parameter $\frac{1}{2}$ for $d \neq t_0$ and of parameter $P[S_{t_0}(k)=1]$ for $d=t_0$. Using this model, we may choose a threshold in order to avoid an exhaustive search on the position d. Synchronization position is estimated as the first one that verifies $\phi_2(d)<\beta$. This solution is suboptimal compared to the exhaustive search but it has a lower computational cost. Because of the lack of space, we won't discuss anymore on this subject.

3. SIMULATION

In this section, we illustrate the behavior and the performance of our algorithm. First of all, we consider the systematic $(15,11,03)$ hamming block code. Figure 4 represents the probability of frame synchronization versus the size of the sliding synchronization window for different values of p_e. These curves are obtained by 3000 Monte Carlo trials where,sq for each trial, the coded sequence, the errors and the propagation delay are chosen randomly. For this simulation, the emitted coded sequence is not interleaved. Therefore the synchronization mismatch is due to the cyclic property of the code. Figure 5 is obtained for a coded sequence interleaved by a pseudo random interleaver that has a size equal to the size of the synchronization window. Performance is much better. However, the Hamming $(15,11,03)$ code has $u=8$ "ones" in each line of H therefore the performance of the method based on ϕ_2 gives bad performance. As the code word is small, the probability to have an error in the code word is also small, therefore the synchronization based on ϕ_1 is good.

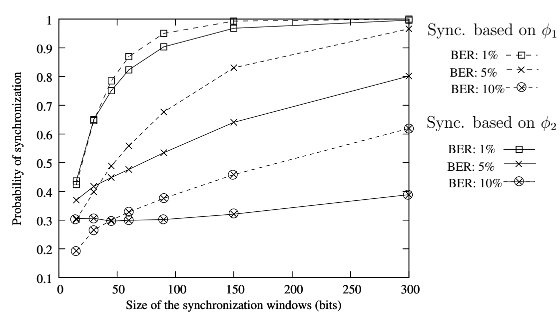

Figure 4: Code $(15,11)$ not interleaved

To illustrate the behavior of our method, we tried another Hamming code: $(15,07,05)$. This code has a code rate inferior to the Hamming code: $(15,11,03)$, thus the syndrome is longer, this may help the method based on ϕ_2 as soon as the syndrome element are independent. This hamming code has the property to have $u=4$ or $u=6$ in the lines of H. As shown in section 2.1 this favours

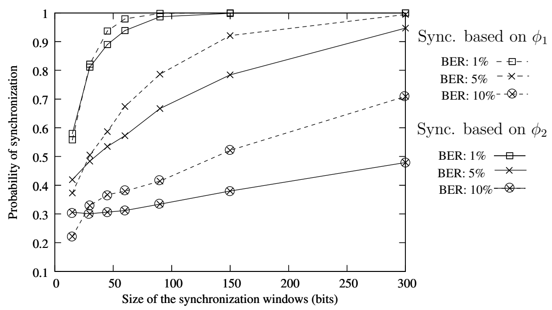

Figure 5: Code $(15,11)$ interleaved

the synchronization procedure based on ϕ_2. Simulations confirm our claims (see fig. 5 and 6).

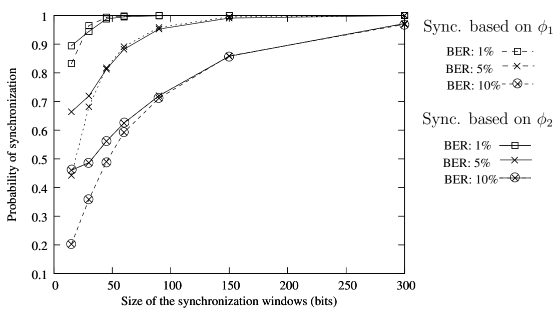

Figure 6: Code $(15,07)$ interleaved

At last, we compare the synchronization performance for a long Hamming code ($n_c = 511$) with an equivalent LDPC code having 4 ”ones” on each line of H. We mean by equivalent a LDPC code that has the same code rate and the same length. Figure 7 presents the probability of synchronization versus the BER of the channel for a sliding window of size n_c.

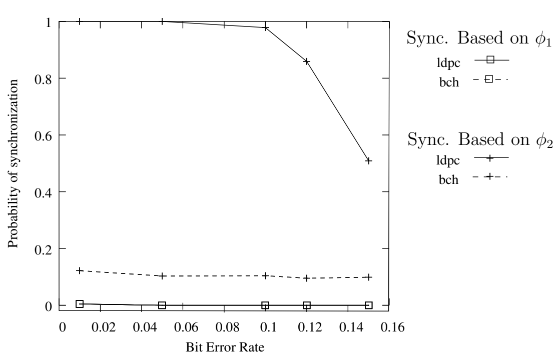

Figure 7: Performance of synchronization for LDPC code

3000 Monte Carlo Trials are run where for each trial the coded sequence, the errors and the propagation delay are chosen randomly. Our synchronization scheme based on ϕ_2 has very convincing performance: for a BER of 10% we are able to synchronize in 97,8% of the cases. Let's have a look at the distribution of the value of $\phi_2(d)$. Figure 8 represents the histogram of ϕ_2 for the synchronized position and for a non synchronized position at a BER of 10%. To minimize the computational cost

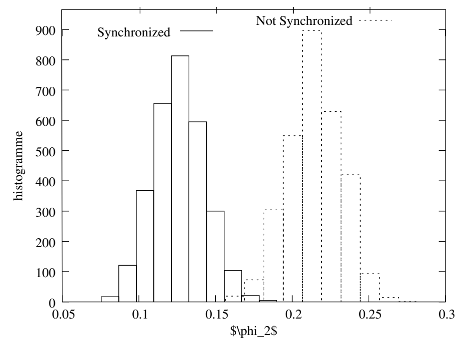

Figure 8: Histogram of $\phi_2(d)$ for $d = t_0$ and $d \neq t_0$ - BER:10%

of the synchronization procedure, we may compare the value of $\phi_2(d)$ and choose the synchronization position as soon as ϕ_2 is less than a fixed threshold.

4. CONCLUSION

We propose a blind synchronization procedure adapted to block codes that has very convincing performance for LDPC code. This method is simple and is working before decoding at high Bit Error Rate. It does not need any synchronization sequence which allows to increase the spectral efficiency. Note that the existed synchronization sequence can be replaced by a more powerful code (introducing more redundant bits) which may lead to an increase of the performance of our synchronization procedure.

REFERENCES

[1] J. L. Massey, “Optimum frame synchronization,” *IEEE Trans. on Communications*, vol. 20, pp. 115–119, April 1972.

[2] R. A. Scholtz, “Frame synchronization techniques,” *IEEE Trans. on Communications*, vol. 28, no. 8, pp. 1204–1212, August 1980.

[3] M. Mostofa and K. Howlader, “Decoder-assisted channel estimation and frame synchonization of turbo coded system,” *VTV*, pp. 1452–1456, 2002.

[4] R.G. Gallager, “Low-density parity-check codes,” *IRE Trans. Inform. Theory*, vol. 8, pp. 21–28, January 1962.

[5] D.J.C. MacKay and R.M. Neal, “Near shonnon-limit performance of low-density parity-check codes,” *Electronics Letters*, vol. 32, pp. 1645–1646, August 1996.