Study and Implementation of Complementary Golay Sequences for PAR reduction in OFDM signals

Abstract

In this paper some results of PAR reduction in OFDM signals and error correction capabilities by using Complementary Golay Sequences are presented. This results have been obtained by simulations and later we have implemented this technique in a DSP. Simulation results show a PAR of 3 dB; however, physical results produce a PAR of 6 dB. Also, an algorithm for generation of Golay Base Sequences *on the fly* is proposed.

Key Words

OFDM, Complementary Golay Sequences, PAR

I. Introduction

Today's needs of bandwidth and flexibility are imposing the use of efficient modulations that may be fit to the characteristics of wireless channels. This is one of the reasons why multicarrier modulation techniques are finding growing interest for Wireless Local Area Networks (WLAN). Recent WLAN standards, such as Hiperlan type 2 [1] and IEEE 802.11a [2], have adopted them for transmission of high bit rates in these networks.The choice of OFDM (Orthogonal Frequency Division Multiplexing) is due to its good performance in multipath environments and the number of sub-carriers has been chosen so as to mitigate the indoor channel effects.

Basically, OFDM divides bandwidth into several orthogonal sub-carriers and sends information into these sub-carriers. Nevertheless, one of the main disadvantages of this modulation is its high PAR (Peak-to-Average power Ratio) requiring the use of linear HPAs (High Power Amplifiers) that are very power-inefficient and have an enormous impact on equipment's autonomy.

Mathematically, we can define PAR by:

$$PAR\{x\} = \frac{max|x_\tau|^2}{E\{|x_\tau|^2\}}$$

This is the relationship between the peaks of the signal and its mean.

Since OFDM is a good technique to mitigate multipah effects, it is interesting to try to improve its disadvantage and reduce the high PAR of this kind of signals. There are a lot of techniques for this purpouse like Complementary Golay Sequences [3], Partial Transmit Sequences (PTS) [4], Selective Mapping (SLM)[4], Tone Reservation [5], Clustered Transmission [6] and Orthogonal Pilot Sequences (OPS) [7].

Golay Sequences have been chosen for two reasons. First, with this technique, PAR is limited up to 3 dB, independently of the number of carriers and input data. This is very important, because we know in advance how is the dynamic range for HPA. Second is the error correction capability of these codes that allow us to improve the whole system.

The remaining of the paper is organized as follows. Section II is a brief description of Complementary Golay Sequences and their characteristics. Then, in section III we show the performance of this kind of code. Next, section IV analyses the complexity of the encoder and decoder. In section V we propose an algorithm to generate Golay Base Sequences, in this way, we do not need to spend memory to store them, and in section VI we show results obtained in the implementation of this encoder. Finally, we draw some conclusions.

II. Complementary golay Sequences

Only a few input data sequences produce signals with high PAR. With Golay codes, we generate sequences that once modulated have PAR limited to 3 dB [3].

Two sequences a and b are Complementary Golay Sequences if the sum of their autocorrelations is null except in zero. This is: $C_a(i) + C_b(i) = 0 \; \forall \; i \neq 0$

There are two very interesting relationships between

Complementary Golay Sequences and Reed - Muller codes [3]. First: each of the $m!/2$ cosets of $RM_{2^h(1,m)}$ in $ZRM_{2^h(2,m)}$ having a cosets representative of the form:

$$2^{h-1} \cdot \sum_{k=1}^{m-1} x_{\pi(k)} \cdot x_{\pi(k+1)} \quad (1)$$

comprise one of $2^{h(m+1)}$ Golay sequences over $\mathbb{Z}_{2^h}$ of length 2^m, where $h > 1$, π is a permutation of $\{1 \ldots m\}$ and 2^m is the length of the complementary Golay sequence.

In addition, there is another interesting relationship: any sequence of the form

$$2^{h-1} \cdot \sum_{k=1}^{m-1} x_{\pi(k)} \cdot x_{\pi(k+1)} + \sum_{k=1}^{m} c_k \cdot x_k \quad (2)$$

with $c_k \in \mathbb{Z}_{2^h}$ is a complementary Golay sequence with respect to the others.

With these two results, it is easy to design a block algorithm for coding input sequences into Complementary Golay Sequences (so, PAR of 3 dB) and with the error correction capabilities of Reed-Muller codes.

We use the first w bits to select the Base Golay Sequence in $ZRM_{2^h}(2, m)$, then we take $m + 1$ groups of h bits each one to build the final code. In this way, we have the encoding algorithm.

We will have therefore $w + h \cdot (m + 1)$ input bits and $2^m \cdot h$ output bits, where m is the code length, h is the modulation depth and w is the number of Golay Base Sequences to use.
Thus, the code rate is:

$$R = \frac{w + h \cdot (m + 1)\ bits}{2^m \cdot h\ bits} \quad (3)$$

where m, h and w are the parameters before cited.

In this equation 3 we can see that code rate decreases exponentially as m increases, so from this point of view, we would use small codes (m small). And, this is less obvious, but h parameter does not affect very much the code rate, so we will have flexibility to modify this parameter without having much effect in code rate.

These Complementary Golay Sequences are only valid for phase modulations, this is, M-PSK modulations. It should be noted that they are not valid for M-QAM.

III. Characteristics and Simulation

This section shows some results obtained by Montecarlo simulations.

A. Channel with Aditive White Gaussian Noise

First, we begin with some results in channels with additive White Gaussian Noise (AWGN). This simulations have been carried out with:

- Same number of sub-carriers as code length ($N = 2^m$)
- Without Frequency Guards
- With Cyclic Prefix
- Additive White Gaussian Noise (AWGN)
- Parameter w at maximum each moment.

PAR is always 3 dB.

We can view in fig. 1 that as we increase the code length (m parameter), we improve BER. This is easy to explain. When m is larger, the errror correction capability of this code is better so, as we increase m, BER decreases.

Also, we can see that there are some E_b/N_0 for which uncoded system is better than coded. This is a block code system so, when the code can correct all errors, the system works properly but when the code is not enough to correct them, the system fails in block and BER increases. This effect is not only for these Golay codes but is very common in FEC (*Forward Error Correction*) codes.

In this fig. 1 we can also see that the slope imcreases as we increase m since, as we have said before, it is a block code, and when it fails, it fails in block and the probability of error is larger.

And also in fig. 1, when modulation depth (h parameter) increases, BER increases too and it would be neccesary to increase code length (m) to compensate for this effect. This is the same as decreasing E_b/N_0. This effect is better shown in fig. 2. This figure was obtained with fixed code length and varying modulation depth (h).

And finally, also fig. 1 shows that if we decrease code rate for the same modulation depth, the performance is better. Obviously the more redundance, the better performance.

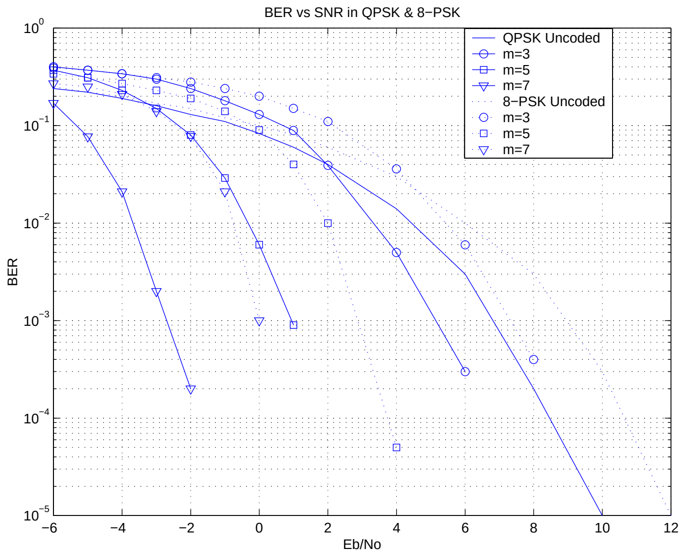

Fig. 1. *Code Rate Comparation between QPSK and 8-PSK*

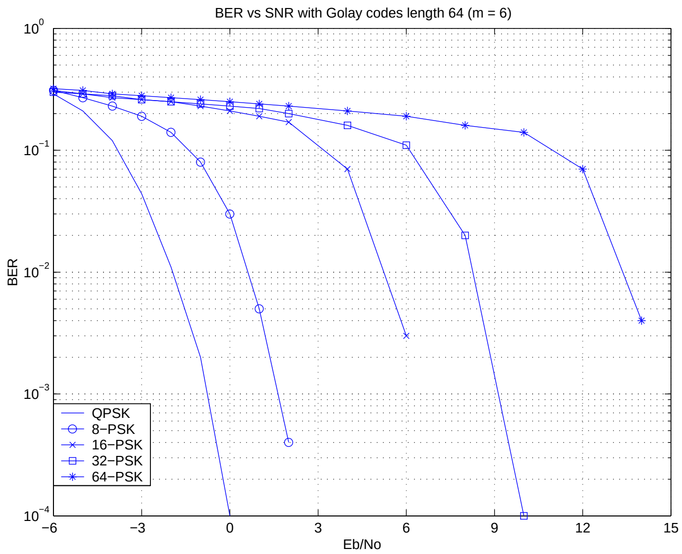

Fig. 2. *Comparation M-PSK (m = 6)*

B. *PAR and Number of Sub-carriers*

In [3] it is demostrated that if we use twice the number of sub-carriers than the code length we obtain PAR of 6 dB. We ask ourselves which rates between the number of sub-carriers and code length are valid for PAR reduction, and we have simulated some environments. With these results, we have more flexibility to design our system.

Simulations have these speciffications:

- Variable number of subcarriers
- Without Guard Frequencies
- With Cyclic Prefix
- With perfect channel (without noise and distortion)
- Parameter w at maximum each moment.

With the obtained data we can say that if we use the same number of sub-carriers as code length or a sub-multiple, PAR obtained is 3 dB, except if the number of sub-carriers is 8; in this case only with $m = 3$ we can obtain a PAR of 3 dB. This is very interesting because we can increase the code length as we want in order to achieve the desired probability of error. But, as we increase the code length (m), we increase the computational load too.

It is true that if we use twice the number of sub-carriers than the code length we obtain PAR of 6 dB, but if we use higher multiples of the code length, PAR is 1 or 2 dB less than in the uncoded system.

From this point of view, when the system has few sub-carriers (64, 128, 256) as in WLAN environments, we will have more flexibility in order to design the system (we will be able to use short codes or long codes for very restrictive BER characteristics), but in systems with large number of sub-carriers (1024, 2048 and so on) codes will only be large or very large, and the system will be more complex.

C. *PAR and Guard Frequencies*

Another point is how PAR reduction depends on Guard Frequencies. Until this moment, all simulations have been run without guard frequencies. Now:

- Number of sub-carriers ($N = 2^m$, $N = 2^{m+1}$ y $N = 2^{m-1}$)
- Variable Guard Frequencies
- With Cyclic Prefix
- Without channel simulation
- Parameter w at maximum each moment.

In this case, results are not conclusive but only tendencies. When we use guard frequencies, Golay Sequences properties are broken, therefore the PAR obtained is not 3 dB but larger; however it is always 2, 3 or 4 dB lower than in the uncoded system. This diference is larger as we increase modulation depth because PAR is more or less constant for one m, independently of the ratio between Guard Frequencies, the number of sub-carriers and h, and in the uncoded system it is not constant with h parameter.

If we focus our atention in the ratio between the number of sub-carriers and guard frequencies, we can conclude that while we hold a small rate (10 %), results are similar.

IV. Algorithm Complexity

In this section, we will analize the complexity of the encoder and the decoder. We will use MAC (*Multiply Add Carry*) operations as a measure, because we will implement them in a DSP (*Digital Signal Processor*). This computacional load will depend on the code parameters (m, h and w), but mainly on m.

A. Encoder Complexity

If we only focus in the encoder algorithm and we supose that previously we have generated all Base Golay Sequences (BGS) and we have them into a lookuptable, we will need $2^m \cdot (m+2)$ MAC operations and 2^m divisions.

For the encoder, complexity will depend exponetially only on m.

B. Decoder Complexity

The decoder, does not always do the same number of operations. We will have therefore one lower bound and one higher bound. The higher bound:

- $2^m \cdot (1 + m \cdot h + m!/2)$ Divisions
- $\left[\left[2^{(m+2)} + m^3 + 2^{m+1} + 2^m \cdot (m+1)\right] \cdot m + 2^m\right] \cdot h + (h+1)$ MAC operations.
- $(2^m + 2) \cdot m + 2^w$ test operations.

As in the encoder, the number of operations depends expontially on m, but also on h and w. And the number of operations is larger than in the encoder case.

If we compare with the complexity of the FFT using radix2 algoritm, we can see that:

$$((N/4+2)\cdot 7 + 32 + log_2 N \cdot (15 + 4\cdot(N/2-2))$$

$$+N/4 + 6\cdot N)\ cycles$$

In fig. 3 we can see the evolution of the number of cycles depending on parameters.

Actually, DSPs can execute several millions of instructions per second, so this algorithm seems viable.

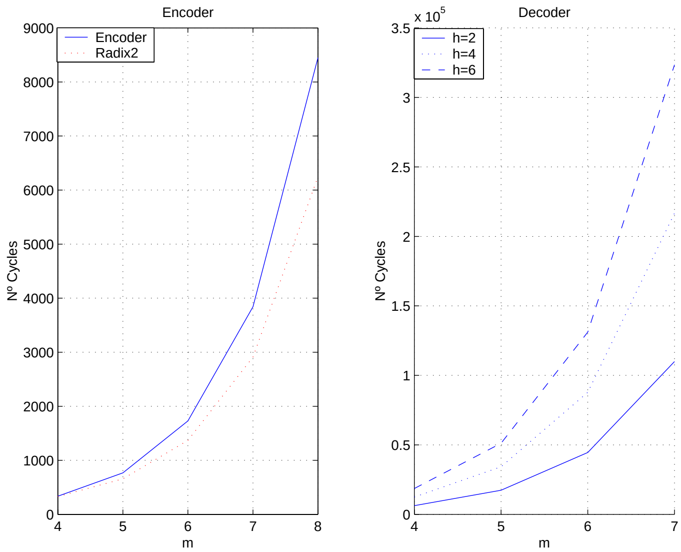

Fig. 3. *Number cycles in DSP for implementation*

V. Generation of Base Sequences

The maximum number of BGS (2^w) is depending on code length ($2^w \leq m!/2$). This number is very large and increases exponentially with m. The memory needed to store all these BGS may be very large.

One way to solve this problem is to limit the number of BGS, but the code rate depends on this w. The advantage of this method is that we can analyse all the base codes and find the best.

Another way is to try to generate these sequences *on the fly* when it is needed. Looking at equation 2, we can see that, if we are able to generate a specific permutation, it is easy to build BGS.

Basically, there are two methods to generate all the permutations: iterative and lexicographic algorithms. Lexicographic algorithms generate all the permutations with a factorial relationship. We can use this in order to design an algorithm to generate a specific permutation. This algorithm is as follows:

Algorithm to Calculate Permutations

1. Array Initialization *list* with all elements in lexicographic order.
2. Array initialization *permutation* vide, $aux = n$, $act = N$ and $i = 1$, where n is number of permutation and N total number of elements.
3. If $n \ mod \ act! = 0$, array *permutation* is completed with inverse of *list* and jump to 7 else jump to 4.
4. $res = \frac{aux}{(act-1)!}$. If integer part of res is 0 and $res > 1$, $res = res - 1$.
5. Add to *permutacion* element in *list* in position integer part of res. Delete this element from *list*.
6. $act = act - 1$, aux is multiplied by the fractionary part of res by $act!$. $i = i + 1$ and back to 3.
7. Array *permutation* is the desired permutation.

With this algorithm we can generate any permutation without the need to generate the rest.

However, not all possible permutations ($m!$) are valid, only half of the generated codes are valid. Symmetrical permutation generate the same code.

It is needed therefore to know how the permutations are distributed with respect to their symmetrical. This is not easy, but first $\sum_{i=1}^{m-1}(m-i)!$ permutations generate first $\sum_{i=1}^{m-1}(m-i)!$ codes, and this number of codes are enough for our purpouse.

When we are encoding, we take first w bits to select BGS, with $2^w \leq m!/2$. At the begining, this was a problem, but now it plays in our favour, because if we take $m = 5$, we have $5!/2 = 60$ possible codes, but we can only use $w = 5$ ($2^5 = 32$), and $\sum_{i=1}^{4}(5-i)! = 33$, so we do not loose anything using our algorithm. Next table shows some examples:

m	$m!/2$	w_{max}	$\sum_{i=1}^{m-1}(m-i)!$	w_{max}	**Loss**
5	60	5	33	5	0
6	360	8	153	7	1
7	2520	11	873	9	2
8	20160	14	5913	12	2
9	181440	17	46233	15	2
10	1814400	20	409113	18	2
11	19958400	24	4037913	21	3
12	239500800	27	43954713	25	2

TABLE I
LOSS COMPARISON TABLE

The loss in code rate is not very important, because we loose only two or three bits.

If we focus in complexity, this algorithm is not very heavy. It needs $2^m + 13 \cdot m$ MAC operations and $2 \cdot m$ test operations. In fig. 4 we can see the diferences between the number of cycles if we use the algorithm with and without calculating the permutations. In decoding, the diferences are larger.

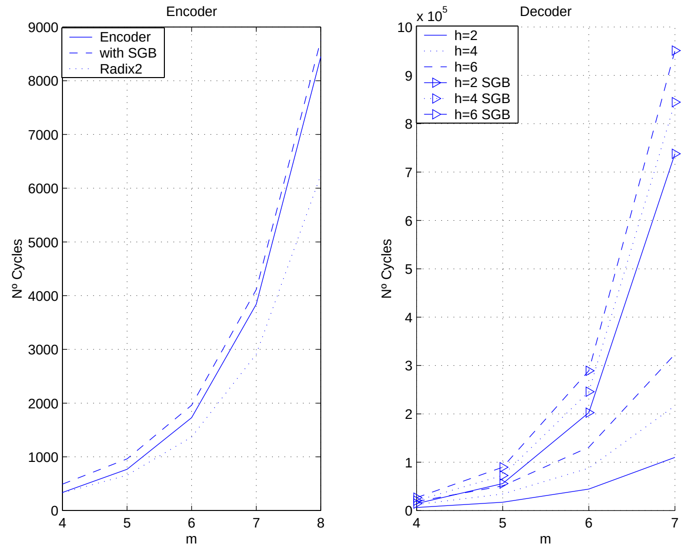

Fig. 4. *Number of cycles with Generation*

Another solution is a hybrid between two methods, to store $1/2$, $1/4$ and so on of all permutations that we want and generate the rest when they are needed. In this way, we reduce complexity.

VI. IMPLEMENTATION

We have implemented this kind of encoder/decoder in a DSP to check whether theoretical characteristics are true or not.

We have used TMS320C6201 DSP from *Texas Instruments* and some converters in order to obtain output physical signals. We have scaled by 100 all the usual values of OFDM parameters [8], thus $N = 32$, BW of 6.25 kHz and carrier frequency of 62.5 kHz.

In fig. 5 we can see the uncoded OFDM output signal. This signal is between 0 and 4.7 V aprox. and in fig. 6 the output coded OFDM signal is shown. We can see clearly that there is a reduction in the peaks of the signal.

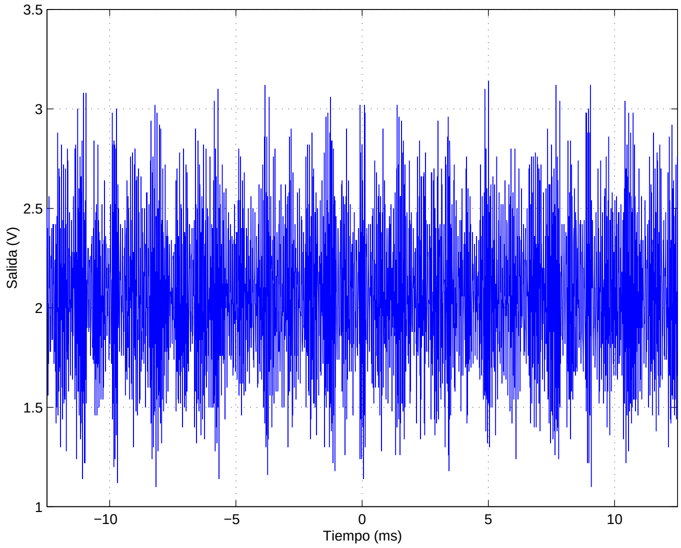

Fig. 5. *Output OFDM. Uncoded*

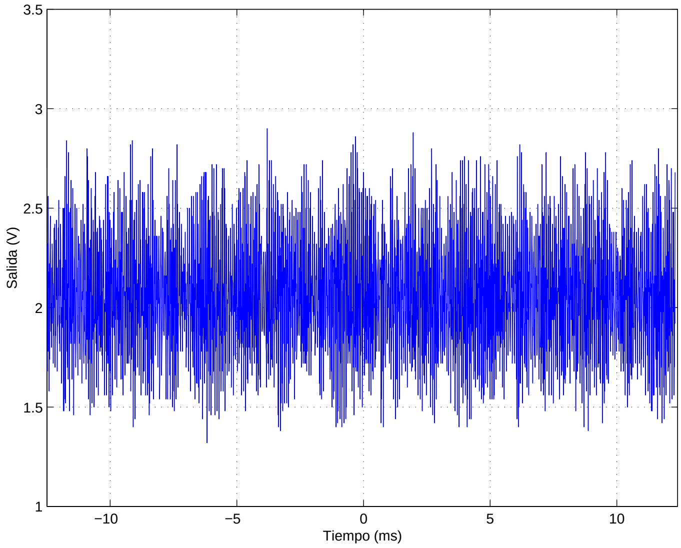

Fig. 6. *Output OFDM. Coded*

PAR of uncoded signal is nearly 9 dB and PAR of coded signal is nearly 6 dB. There is a difference between simulations and implementations of 3 dB [9] because in simulations we use low pass equivalent signal representation and in implementation we use the real signal, quantified and with noise.

VII. Conclusions

The PAR of OFDM signal has been reduced by using Complementary Golay Sequences. In simulations, PAR is limited to 3 dB whereas in physical implementation PAR es limited to 6 dB. However, this PAR reduction is only valid for OFDM signals without guard frequencies. With guard frequencies, the reduction is not so good. Another constraint is that these Complementary Golay Sequences are not valid for use with M-QAM schemes that are more efficient than M-PSK. However the PAR reduction can compensate the efficiency loss.

Also, we have improved the probability of error of this kind of signals by using the error correction capabilities of Reed-Muller codes. We can fix the probability of error and find Golay encoder parameters in order to achieve this desired BER. (m and h). We can use Golay codes alone for PAR reduction and error correction, nevertheless it is a good idea to use them with an outer code to improve the performance.

It should be emphasized too that we have designed an algorithm to generate BGS *on the fly*. In this way it is not needed to use memory to store all base sequences, although it is better for small values of w.

REFERENCES

[1] ETSI101. Broadband radio access networks (bran); hiperlan type 2; physical (phy) layer. (101 475 V1.1.1), 2000.

[2] IEEE802. Wireless lan medium access control (mac) and physical layer (phy) specifications: High-speed physical layer in the 5 ghz band. 1999.

[3] J. A. Davis and J. Jedwab. "Peak-to-Mean Power Control in OFDM, Golay Complementary Sequences and Reed-Muller Codes". Dec. 97.

[4] L.Cimini Jr and N.R. Sollenberger. "Peak-to-Aberage Power Ratio Reduction of an OFDM Signal Using Partial Transmit Sequences". *Icc*, 1999.

[5] J. Tellado and J. Cioffi. "Peak Power Reduction for Multicarrier Transmission". *Globecomm*, Nov 1998.

[6] L.J. Cimini Jr, B. Daneshrad, and N.R. Sollenberger. "Clustered OFDM with Transmitter Diversity and Coding". *Globecom*, 1996.

[7] M. J. Fernández-Getino, J. M. Páez-Borrallo, and O. Edfors. "Orthogonal Pilot Sequences for Peak-to-Average Power Reduction in OFDM". *Procedings of IEEE VTC*, 2001.

[8] M. Lobeira, A. García, R. Torres, and J. L. García. "Parameter estimation and indoor channel modelling at 17 GHz for OFDM-based broadband WLAN". *Procedings of IST Mobile Summit*, 2000.

[9] J. Tellado. *"Peak-to-Average Power Reduction"*. PhD thesis, Standford University, September 1999.