2nd URSI AT-RASC, Gran Canaria, 28 May - 1 June 2018

A Novel Adaptive Golay Correlator Synchronizer for IEEE 802.11ad Indoor mmWave Systems

Ahmed El-Yamany(1),and Markus Petri(1)
(1) IHP, Im Technologiepark 25, 15236 Frankfurt(Oder), Germany, elyamany;petri@ihp-microelectronics.com

Abstract

In this paper, a new design for the synchronizer of an 802.11ad WLAN system is proposed. The proposed design is based upon the well-known Golay Correlator (GC) synchronizer. The GC synchronizer performance is degraded in the low Signal to Noise Ratio (SNR) regime to a fixed threshold. In our proposed scheme, an adaptive algorithm optimizes the threshold of the GC so that it becomes SNR-independent. Our simulation results show that the proposed scheme greatly outperforms the traditional GC based synchronizers in different indoor environments indicating a revolutionary change in the synchronizer design not limited to the IEEE 802.11ad.

1 Introduction

The unlicensed 60 GHz radio frequency (RF) band has drawn a great attraction in the past decade due to the high bandwidth offered and the possibility of achieving multi-gigabit throughput. The IEEE 802.11ad for Wireless Personal Area Network (WLAN) [1] and IEEE 802.15.3c for Wireless Personal Area Network (WPAN) [2] standards are presented to reach data rates over 1 Gbit/s. Both use OFDM for multi-carrier transmission offering high data rates and a great immunity to multi-path fading and low Signal to Noise Ratios (SNR).[3] In previous works [3, 4, 5] different methodologies were used for synchronization. In [3, 5] Golay Sequences were used, with their exceptional auto-correlation properties the approach outperformed most of the other ways. While the results in [5, 6] regarding the synchronizer design were reliable, a problem was revealed when using the Golay based Normalized Auto-Correlator (NAC). The block tends to be sensitive to low SNR values, the root cause was traced back to the fact that it uses a fixed and a predefined threshold for estimating the frame start.

The SNR dependency risks the whole reception process in different ways. With low thresholds -which are most commonly used- the synchronizer mistakenly interprets incoming noise as a start of a signal. This also occupies the system with false data preventing it from detecting the real transmitted signal. When using higher thresholds, the low SNR signals are lost because the NAC's threshold is never trigerred. In this paper, we present an adaptive design solution for the synchronizer with little hardware overhead solving the mentioned synchronization dilemma.

2 The Architecture of the Adaptive Golay Correlator Based Synchronizer

Fig.1 shows the preamble structure as stated in the standard IEEE 802.11ad. It is composed of two main parts; a Short Training Field (STF) and a Channel Estimation Field (CEF). The STF consists of 17 Golay sequences each of 128 samples, with an inverted sequence at the end. The CEF is divided into three parts; two are 512 samples long Golay sequences made from smaller complementary 128 sequences and the last one is an inverted sequence.

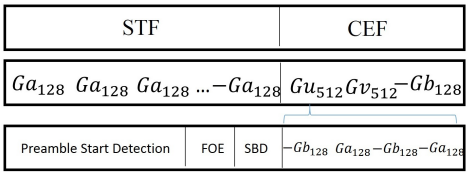

Figure 1. IEEE 802.11ad preamble structure

In [5], a normalized Golay correlator was used to detect the preamble start and to estimate the frequency offset. Finally, it detects the start of the channel estimation sequence through the inverted Golay symbol at the end of Symbol Training Field (STF) as shown in Fig.1. The functionality of the synchronizer is divided into 3 processes, first the synchronizer detects the start of the preamble, then it estimates the frequency offset, corrects it and finally detects the start of the channel estimation part of the preamble.

Every 128 samples of the received signal are first correlated with the preceding 128 samples in a normalized auto-correlation process:

$$NAC[i] = \frac{\sum_{n=0}^{128-1} r_{i-n}.r_{i-n-128}}{\sum_{n=0}^{128-1} r_{i-n}.r_{i-n}} \quad (1)$$

where (r) is the received signal, (i) is the sample index, and (n) is the auto-correlation index. In the ideal case, the output should be 1. But with a noisy received signal:

$$r_k = \sum_{i=0}^{L-1} x_{k-i}.h_i + N_k \quad (2)$$

where (x) is the transmitted signal, (h) is the channel impulse response and (N) is the additive white Gaussian noise,

the output of the correlation is dependent on the noise level, as shown in the following derivation:

$$NAC[i] = \frac{\sum_{n=0}^{128-1} (\tilde{x}_{i-n} + N_{i-n}) \cdot (\tilde{x}_{i-n-128} + N_{i-n-128})}{\sum_{n=0}^{128-1} (\tilde{x}_{i-n} + N_{i-n}) \cdot (\tilde{x}_{i-n} + N_{i-n})} \quad (3)$$

where $\tilde{x}$ is the transmitted signal convoluted with the channel impulse response.

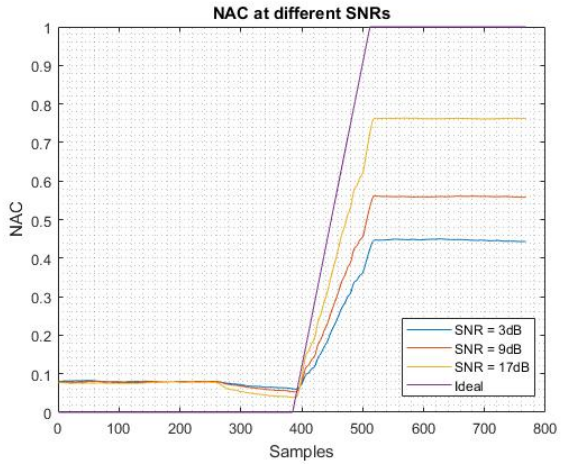

Figure 2. SNR-dependency for different reception scenarios

In Fig. 2, we can observe how the NAC varies in different reception situations. Two perfectly matched Golay sequences would result in a constant one. If 512 zeros are added in front of the Golay sequences, the result is the curve labeled "'ideal"'. The output of the NAC linearly increase for 128 samples until it saturates at "1". The other curves show the practical cases, where the received signal and the preceeding zeros are noisy. The NAC result now is merely dependent on the SNR level. Therefore, the detection of the frame start would depend on the predefined used threshold.

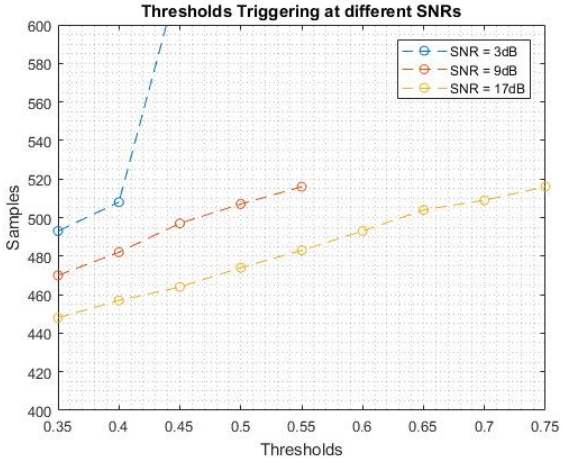

Figure 3. Thresholds Triggering at Different SNRs

Fig. 3 shows the threshold triggering for different SNR values. As we see there are different saturation levels for the NAC, at SNR = 3dB, it reaches around 0.4, 0.55 at SNR = 9dB and finally around 0.75 at SNR = 17dB. The results clearly points to the fact that the higher the SNR, the NAC curve should approach the ideal case as shown in table I.

Table 1. GC-Synchronizer THRESHOLDS TO SNR LEVEL MAPPING

Nearest Threshold	SNR Level
0.35 - 0.45	SNR < 3 dB
0.5 -0.65	SNR < 9 dB
0.7 -0.8	SNR > 9 dB

Our proposal is provided to maintain a more reliable result in the detection process. In Fig. 4, there is a block diagram for the proposed synchronizer architecture. The proposed architecture takes advantage of what we call "transition region" where the NAC output starts to increase till it reaches saturation at the ideal frame start. In the IEEE 802.11ad case, it takes 128 samples, since the GS is repeated every 128 sample. The transition region is where most of the thresholds should be triggered, making it easier to detect when using multiple thresholds.

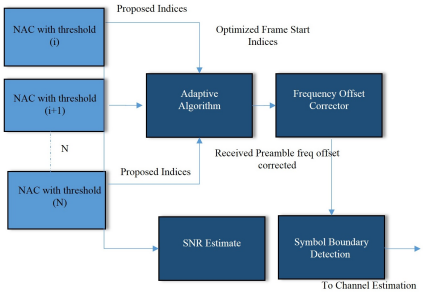

Figure 4. Proposed Adaptive Synchronizer Block Diagram

3 The Adaptive Synchronizer Algorithm

The proposed algorithm uses N thresholds to get the most probable start of the preamble. N depends on the range of the SNR to be covered. In our simulations, we used 10 equidistant thresholds to get the most probable preamble start. The adaptive algorithm measures the "compactness" of the indices given by each threshold. The compactness is a parameter defined to measure the most number of resultant indices within a defined interval –transition region-.

The algorithm is defined into two main stages, first it gets the indices that were triggered within the transition region -the most compacted indices within a 128 interval-, then it gets the highest compactness rate within all the triggered thresholds within the transition region. The first step rules out both the indices triggered by very low thresholds in high SNRs or noise with high correlation peaks and also, the indices triggered by high thresholds in low SNRs which are

usually far away from the exact preamble start. The second step of determining the highest compactness rate within the transition region is merely to get the highest accuracy, since slope usually changes prior to the saturation. This slope change can be detected through the increase of the compactness of the detected frame start indices just before saturation.

There are some remarks regarding the threshold selection to assure more accuracy when getting the exact preamble start. First, the thresholds should be equidistant from each other. Furthermore, thresholds should only cover the effective NAC range. E.g. 0.1 is not a good threshold value as it will only be hit by noise in a real environment, and 0.95 is a very high threshold which will only be hit under perfect circumstances. If used, the indices of those thresholds are usually discarded during phase I, nevertheless they will consume hardware resources.

3.1 Phase I: Transition Region Detection

Phase I is used to get the maximum number of triggered indices within 128 samples -transition region-. At first, the difference between the first triggered threshold and the last one is checked against the transition region length -128 samples in this case-.

$$(|P_n - P_{N-i}|) < N_s, i = 0 : (N-1), n = 0 : (N-1) \quad (4)$$

where P is the vector that contains the indices triggered by the thresholds, n,i are for sweeping the vector P, N is the size of P, and N_s is the length of the transition region. The next step is counting the set of indices triggered within the limit and taking the set with the maximum number of indices. If there are multiple sets with the same number of indices, their variances are determined with the following equation and the one with the highest variance is chosen.

$$\sigma^2 = \sum_{i=0}^{N-1} (\bar{S} - S_i)^2 \quad (5)$$

where S is the vector containing the indices within the limit, the first index is subtracted from the other indices and N is the length of the vector S. The reason for choosing the least variance within the limit is that the thresholds should be triggered all over the transition region, the more equidistant they are, the more they approach the ideal case (see Fig. 2).

3.2 Phase II: Preamble Start Calculation

Phase II gets the exact preamble start from the set of indices defining the transition region. Since the transition region is not ideal, the thresholds tend to get triggered closer to each other as we approach the saturation. This change in the slope of the transition region can be detected through measuring the compactness of the indices and calculating the most compacted set with the most number of indices within.

First we calculate the distance vectors; these vectors contain the distance between the indices triggered with different thresholds as described in the following equation:

$$D_k^l = |I_i - I_{i-l}|; l = 1 : \frac{L}{2}, k = L - l, i = 1 : \frac{L}{2} - l \quad (6)$$

where D is the vector containing the distances for the indices, k is the index and l is the step used for the distances. l is limited with L, which is the length of I, the vector that contains the indices triggered in the transition region. The order indicates how many indices are in this set, we are trying to find the most compacted set of indices in the transition region.

The next step is to get the minimum of the calculated vectors, indicating the most compacted set of indices in each order. To compare the distances calculated for each set, the following equation is used:

$$V_l = \frac{d_l - E\{D_k^l\}}{E\{D_k^l\}} \quad (7)$$

where d is the min of vector D of order l and V is the spread of the deviation from the mean. We use this equation to get the highest deviation from the mean. Since the higher the deviation, the more significant the calculated difference, consequently the more the compactness is.

The algorithm gets a precise preamble start without a significant increase in hardware overhead, since most of the processing units are simple modules like subtracters or comparators. Also, the additional latency is negligible in a hardware implementation.

4 Performance Results

The proposed synchronization was integrated in an IEEE 802.11ad compliant 60 GHz high data rate OFDM communication system. The transmitted signal is first 5/8 Low Density Parity Check (LPDC) coded, then the signal is QPSK modulated. The preamble is $\pi/2$ BPSK modulated. Both data and preamble are mapped to sub-carriers in a 512 FFT OFDM modulator. To simulate the system performance, the TGad conference room channel model with the NLOS scenario from [7] was applied. Both stations are on the same horizontal level and are 2 meters apart. Furthermore the transmitted signal was exposed to phase and frequency offsets, in addition to phase noise.

The results in Fig. 5 shows the spread of the detected frame start index for 1000 simulation runs. The figure clearly shows that the highest percentage of indices is found around the exact start "512". It also shows the increased performance with higher SNRs.

In Fig.6 the performance of the adaptive synchronizer is compared with the Golay based correlator synchronizer

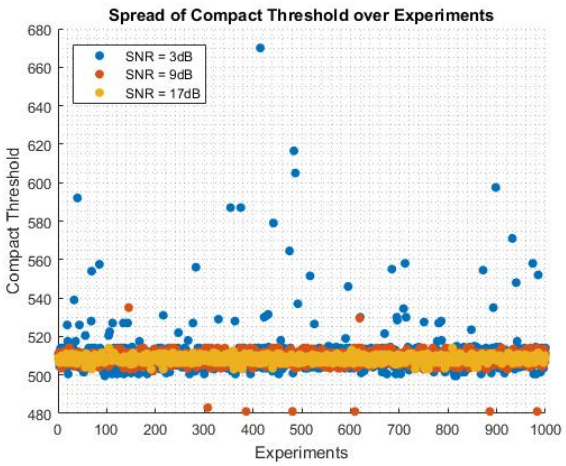

Figure 5. Spread of frame start detection

from [5].The comparison with the GBC synchronizer seems reasonable due to the pre-published results in [5, **?**, 6], indicating the favoring of the Golay based synchronizers. The figure shows the percentage of precise detections. All detected start indices with a tolerance of +/-10 samples from the exact start are defined as successful. It is clear how the adaptive synchronizer outperforms both instances used for comparison, whether using a high threshold or low threshold. The successful detection rate for GC synchronizers with a fixed threshold is nearly zero for low SNRs and a high threshold, increasing up to 100% for higher SNRs. For a lower threshold, the successful detection rate starts with moderate values at lower SNRs and decreases to 0% at higher SNRs. When using the proposed adaptive synchronizer with several thresholds, we see a nearly stabilized performance starting with 95% successful detection rate at 3 dB, increasing to nearly 100% precise detection at 17 dB.

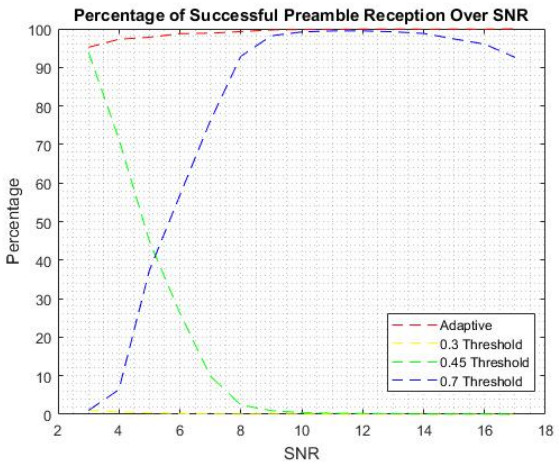

Figure 6. Percentage of Successful Preamble Detection for both adaptive and fixed thresholds Synchronizers

5 Conclusion and Future Work

This paper presents a novel synchronizer design based on Golay complementary sequences. With a little hardware overhead, the performance of the whole system is remarkably improved. Furthermore, with such synchronizer, the whole system gains more flexibility against dynamic changes in the received signal SNR, allowing it to be more reliable. Future work is to use the knowledge acquired from the adaptive synchronizer at different SNRs to adjust the performance of the whole inner receiver optimizing the overall throughput, and increasing the performance.

References

[1] IEEE Standard for Information technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 3: Enhancements for Very High Throughput in the 60 GHz Band.

[2] IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 15.3: Amendment 2: Millimeter-wave-based Alternative Physical Layer Extension.

[3] Nicholas Preyss and Andreas Burg. Digital synchronization for symbol-spaced IEEE802.11ad Gigabit mmWave systems. In *2015 IEEE International Conference on Electronics Circuits, and Systems (ICECS)*. Institute of Electrical and Electronics Engineers (IEEE), dec 2015.

[4] Umberto Mengali. *Synchronization techniques for digital receivers*. Springer Science & Business Media, 2013.

[5] Wei-Chang Liu, Ting-Chen Wei, Ya-Shiue Huang, Ching-Da Chan, and Shyh-Jye Jou. All-Digital Synchronization for SC/OFDM Mode of IEEE 802.15.3c and IEEE 802.11ad. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 62(2):545–553, feb 2015.

[6] Chun-Yi Liu, Meng-Siou Sie, Edmund Wen Jen Leong, Yu-Cheng Yao, Chih-Wei Jen, Wei-Chang Liu, Chih-Feng Wu, and Shyh-Jye Jou. Dual-mode all-digital baseband receiver with a feed-forward and shared-memory architecture for dual-standard over 60 GHz NLOS channel. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 64(3):608–618, mar 2017.

[7] A. Maltsev, R. Maslennikov, A. Lomayev, A. Sevastyanov, and A. Khoryaev. Statistical Channel Model for 60 GHz WLAN Systems in Conference Room Environment. *Radioengineering*, 20(2):409–422, June 2011.