



## Improved Multiplierless Architecture for Header Detection in DVB-S2 Standard

Héctor Miyashiro, Emmanuel Boutillon, Christian Roland, Joel Vilca, Daniel Díaz

### ► To cite this version:

Héctor Miyashiro, Emmanuel Boutillon, Christian Roland, Joel Vilca, Daniel Díaz. Improved Multiplierless Architecture for Header Detection in DVB-S2 Standard. IEEE International Workshop on Signal Processing Systems (SIPS'2016), Oct 2016, Dallas, United States. hal-01390473

**HAL Id: hal-01390473**

**<https://hal.science/hal-01390473>**

Submitted on 2 Nov 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Improved Multiplierless Architecture for Header Detection in DVB-S2 Standard

Héctor Miyashiro\*, Emmanuel Boutillon<sup>†</sup>, IEEE Senior Member, Christian Roland<sup>†</sup>, Joel Vilca\* and Daniel Díaz\*

\*INICTEL-UNI, San Borja, Lima, PERÚ

E-mail: hmiyashiro@inictel-uni.edu.pe, vilcapizarrojoel@gmail.com, ddiaz@inictel-uni.edu.pe

<sup>†</sup>Université de Bretagne Sud, Lab-STICC, UMR 6285, 56100 Lorient, FRANCE

E-mail: emmanuel.boutillon@univ-ubs.fr, christian.roland@univ-ubs.fr

**Abstract**—One of the first processing steps in a DVB-S2 signal receiver is the detection of frame's header. Recently, an architecture using only the phase information of the received samples was proposed. In this paper several optimization in algorithm/architecture are proposed, leading to better performance and reduced hardware complexity. For an SNR of -3 dB, the probability of miss detection of the header detector is reduced from 0.7 down to 0.52 for a constant false alarm probability of  $10^{-6}$ .

## I. INTRODUCTION

DVB-S2 is the satellite TV broadcasting standard proposed by ETSI. It is the successor of the popular DVB-S with novel features [1]: ACM (Adaptive Coding and Modulation), LDPC (Low-Density Parity Check) to increase system capacity in order to meet the increasing demand of multimedia data such as HDTV, 3D-HDTV. Data sent according to DVB-S2 standard are structured into frames that receivers need to detect and synchronize with, before performing demodulation and decoding processes. With the aide of a known sequence at the beginning of frames, PL (Physical Layer) header, algorithms can detect the start of frames. The PL header and the PL payload are part of the PL frame as shown below:

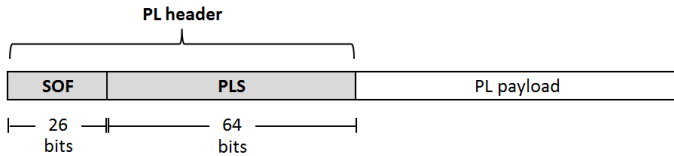


Fig. 1: PL frame structure

The first 26 bits of a PL frame correspond to the SOF (Start Of Frame) field which contains a synchronization sequence *sof* defined by the DVB-S2 standard. The next 64 bits is the PLS (Physical Layer Signaling) field which contains encoded 7 bits of information ( $b_0 b_1 b_2 b_3 b_4 b_5 b_6$ ) that registers the parameters used in the generation of the PL frame indicating: length (64 800, 16 200 bits), structure (presence or absence of pilot symbols), modulation (4 types of modulation) and coding (8 code rates). The encoding process of these 7 bits into the PLS field first multiplies in modulo-2 the first 6 bits of information with the  $6 \times 32$  Hadamard matrix  $\mathbf{H}$ , i.e.,

$$\mathbf{y} = (b_0 \ b_1 \ b_2 \ b_3 \ b_4 \ b_5)_{1 \times 6} \times \mathbf{H}_{6 \times 32},$$

where the Hadamard matrix  $\mathbf{H}$  is a constant matrix that has the following pattern of generation:  $\mathbf{H}(u, v) = (v \gg u)$  for the first 5 rows (element  $\mathbf{H}(u, v)$  equals LSB of binary representation of  $v$  after being shifted right  $u$  positions) and  $\mathbf{H}(5, v) = 1$  for the last row, i.e., the first row contains a periodic sequence of "01", the second row, a periodic sequence of "0011", the third row, a periodic sequence of "00001111" and so on up to the sixth row that is only composed of "1". Then each element of the product vector  $\mathbf{y} = (y_0 \ y_1 \ y_2 \ \dots \ y_{31})$  is xor operated with the bit of information  $b_6$ . The elements and the results are put together in a new vector and scrambled by the fixed sequence *scr* defined by the standard using the xor operation to finally obtain the coded *pls* sequence contained in PLS field, i.e.,

$$\mathbf{pls} = (y_0 \ y_0 \oplus b_6 \ \dots \ y_{31} \ y_{31} \oplus b_6)_{1 \times 64} \oplus \mathbf{scr}_{1 \times 64}.$$

To describe the suggested header detector, we consider the transmission signal model of DVB-S2 standard. Let  $s(k)$  be the  $k^{th}$  bit of information to be transmitted with a BPSK (Binary Phase-Shift Keying) modulation according to the function  $B(s(k)) = 2s(k) - 1$ , then the received signal  $r(k)$ , assuming that automatic gain control and symbol-time synchronization circuits have been worked properly, can be written as

$$r(k) = B(s(k))e^{j(2\pi\Delta_f T_s k + \phi(k))} + n(k), \quad (1)$$

where  $\Delta_f$  is the carrier frequency offset,  $T_s$  is the symbol period,  $\phi(k)$  is the time-variant phase noise and  $n(k)$  is the AWGN (Additive White Gaussian Noise) with two-sided power spectral density,  $N_o$ . Header detection task should be able to be performed at low signal to noise ratio (SNR = -2.35 dB, minimum required for DVB-S2 standard) and large carrier frequency offset introduced by low cost mass market DVB-S2 receivers, that can be in the order of  $\Delta_f T_s = 0.2$  leading to phase rotation of input symbols in the range between  $-\pi/2.5$  and  $+\pi/2.5$  radians per symbol, i.e., during the reception of the 90 symbols of a PL header, up to a maximum of  $\pm 18$  rotations of input signal can be observed.

The remaining parts of this paper is organized as follows: Section II gives a brief state of the art header detection algorithms. Section III presents the proposed variations to SOF and PLS algorithms presented in [3]. Section IV introduces Global Summation method that combines SOF and PLS detections, and compares its performance to already published solution. Section V presents the improved architecture. Finally, conclusions are given in Section VI.

## II. STATE OF THE ART

In this section, we will first defined the metric to measure performance and then give a rapid survey of header detection algorithms.

### A. Performance evaluation of a header detector

Frame synchronization task is a detection task, i.e., determine in the stream of incoming symbols the actual start of frame. The detector computes a metric to evaluate if the last  $L$  symbols (length of synchronization sequence) correspond to a start of frame or not. The output metric is compared to a threshold to make the decision. The detector can make two types of error: first, to make a false detection (false alarm) and second, to miss a true start of frame (missed detection). Each value of the threshold correspond to a couple  $(P_{MD}, P_{FA})$ , where  $P_{MD}$  is the probability of missed detection and  $P_{FA}$  the probability of false alarm. The Receiver Operating Characteristic (ROC) curve is a 2-dimensional plot used to represent the trade-off between  $P_{MD}$  (y-axis) and  $P_{FA}$  (x-axis) according to different values of the threshold. Comparing the performance between different detector can be simply done using the ROC curve. For a given false alarm rate, the one leading to the lowest missed detection rate is the most efficient. One should note that, in the DVB-S2 case, the false alarm rate should be very low. In fact, with a QPSK modulation, with long frame and no pilots, a start of frame occurs every 32,290 symbols. A false alarm probability greater than  $10^{-5}$  leads to too many dummy starting of synchronization/decoding processes.

### B. Review of existing header detectors

The simplest algorithm for detecting a synchronisation sequence, **sync** of length  $L$ , in a received signal  $r(k)$  is the correlation function  $c(k) = |\sum_{l=0}^{L-1} r^*(l-k)\mathbf{sync}(l)|$ , ( $r^*(k)$  means the complex conjugate of  $r(k)$ ), and locating for which  $k$ ,  $c(k)$  is greater than a predefined threshold. Based on correlation an optimized algorithm was derived through Maximum-Likelihood (ML) estimation in [4], which adds to the correlation function an energy correction term, for accounting random data surrounding **sync**, as

$$\Lambda(k) = |c(k)| - e(k), \quad (2)$$

where  $e(k) = \lambda \sum_{l=0}^{L-1} r^2(l-k)$ ,  $\lambda$  is a constant, providing better performance compared to the simple correlation function, with a little increment in implementation complexity. These algorithms for frame synchronization are simple and effective but have performance degradation as frequency offset

increases. Solutions to this problem have been published and are mostly variations of the algorithm presented in [5] that was derived by modifying the procedure used for obtaining the ML estimation in [6]. The algorithm is as follows

$$\Lambda(k) = \sum_{i=1}^{L-1} \{|d_i(k)|^2 - \varepsilon_i(k)\}, \quad (3)$$

where  $d_i(k)$ , referred to as the differential correlation term, is computed as  $d_i(k) = \sum_{l=0}^{L-1-i} r(k-l)r^*(k-l-i)\mathbf{sync}^*(l)\mathbf{sync}(l-i)$  and  $\varepsilon_i(k)$ , which represents an energy correction term of the received sequence, is defined as  $\varepsilon_i(k) = \sum_{l=0}^{L-1-i} |r(k-l)|^2 |r(k-l-i)|^2$ . Different variations of (3) exist leading to different algorithms that provide gain over the others, e.g., in [7] an energy-corrected approach was proposed by introducing a multiplicative weight factor  $\alpha$  to the square root of the energy correction term  $\varepsilon_i(k)$  and dropping the square of  $d_i(k)$  i.e.,  $\Lambda(k) = \sum_{i=1}^{L-1} \{|d_i(k)| - \alpha \sqrt{\varepsilon_i(k)}\}$ . Another variation proposed uses the vector sum instead of summing their magnitudes i.e.,  $\Lambda(k) = |\sum_{i=1}^M d_i(k)| - \beta \sqrt{\sum_{i=1}^M \varepsilon_i(k)}$ , where  $M$  is a constant that needs to be chosen such that  $M\Delta_f T_s < 0.5$ . In [8] was proposed a decision variable,  $\Lambda(k) = \sum_{i=1}^{N-1} |d_i(k)|$  which becomes to be a simplification of (3) wherein only the differential correlation was taken with dropped square. This variable was derived from a post-detection integration (PDI) approach. In [2], the Hadamard structure of the PLS field is fully exploited and combined with the SOF field to achieve better detection performance. In [3] a hardware implementation of a method called JOINT that combines SOF and PLS detectors is proposed: a detection is declared when both SOF and PLS detectors agree. The extra cost of hardware induced by the PLS detector allows reducing the  $P_{MD}$  for more than 45% compared to SOF detectors for a  $P_{FA} = 10^{-6}$  and  $E_s/N_o = -3$  dB. This article proposes to replace the JOINT method (described in details in the following section) by a GLOBAL method, involving coherent summation of the outputs of SOF and PLS filters before decision.

## III. VARIATIONS OF PROPOSED SOF AND PLS DETECTORS

In this section variations of original SOF and PLS detectors presented in [3] are proposed for improving performance and reduce implementation complexity.

### A. SOF detector

Original SOF detector algorithm  $R_o(k)$  performs a differential correlation for every symbol of the input signal  $r(k)$  with the synchronization sequence **sof** in the polar domain, using only the phase information, i.e.,  $\theta(k) = r(k)/|r(k)|$ , this way multiplications are performed as additions reducing implementation complexity. SOF detection is achieved when  $R_o(k)$  is larger than a predetermined threshold  $T_{SOF}$ . The algorithm is computed as

$$R_o(k) = \sum_{i=1}^{25} |n_i(k)|^2, \quad (4)$$

where  $n_i(k)$  is given by

$$n_i(k) = \sum_{l=0}^{25-i} e^{j(\theta(k-l) - \theta(k-l-i))} g_i(l), \quad (5)$$

and the coefficients  $g_i(l)_{l=0,1,\dots,(25-i)}$  of the Finite Impulse Response (FIR) filter  $G_i$  are given by

$$g_i(l) = B(\text{sof}(l))B(\text{sof}(l-i)). \quad (6)$$

In (4), in terms of implementation complexity the computation of  $R_o(k)$  requires extensive operations (additions, multiplications) because exponentiation is involved (i.e.,  $R_o(k)$  takes the square of the modulus). Considering this, variation #1 of SOF algorithm, represented by  $R_1(k) = \sum_{i=1}^{25} |n_i(k)|$  reduces hardware utilization because it only takes the modulus of  $n_i(k)$ , requiring fewer operations and also fewer number of bits for representing big numbers resulted from squared modulus summation. There can be further simplification if we change the set of  $i$  values used in  $R_1(k)$ . In order to derive a simpler implementation, variation #2 given as  $R_2(k) = \sum_{i=1,2,4,8,16} |n_i(k)|$  takes only five values of  $n_i(k)$ , instead of twenty-five, for index  $i$  equal to the power of 2. In this way, part of the architecture of  $R_2(k)$  can be shared with the PLS detector due to the same values of  $i$  chosen.

### B. PLS detector

Even though **pls** is not a fixed sequence, it is generated from fixed elements: Hadamard matrix **H** and the scrambling sequence **scr** that encode the frame's information. These elements are used for performing PLS detection as proposed in [3] where the original PLS detector algorithm  $T_o(k)$  is compared with a threshold  $T_{PLS}$  to determine if **pls** is detected ( $T_o(k) > T_{PLS}$ ) or not and is computed as

$$T_o(k) = \sum_{i=1,2,4,8,16,32} |m_i(k)|^2, \quad (7)$$

where  $m_i(k)$  is given by

$$m_i(k) = \sum_{l=0}^{63-i} e^{j(\theta(k-l) - \theta(k-l-i))} h_i(l), \quad (8)$$

the phase information of  $r(k)$  is represented by  $\theta(k)$ , the coefficients  $h_i(j)_{j=0,1,\dots,(63-i)}$  of the FIR filter  $H_i$  are given by

$$h_i(l) = B(\text{scr}(l))B(\text{scr}(l-i))(1 - (l >> \log_2(i))), \quad (9)$$

where **scr** is the binary scrambling sequence and  $(l >> \log_2(i))$  is the LSB of binary representation of  $l$  after being shifted right  $\log_2(i)$  positions. As in (4), the same simplification applies here to (7), thus variation #1 of PLS detector algorithm can be computed as  $T_1(k) = \sum_{i=1,2,4,8,16,32} |m_i(k)|$ , which only takes the modulus value of  $m_i(k)$  instead of its square.

### C. Performance of proposed variations

In this subsection, we show the performances of the original and different proposed variations of SOF and PLS detectors presented above using the ROC curve Fig. 2 at  $E_s/N_0 = -3$  dB (the DVB-S2 limit is -2.35 dB) and a normalized frequency offset  $\Delta_f T_s = 0.1$ . The modulation is a BPSK.

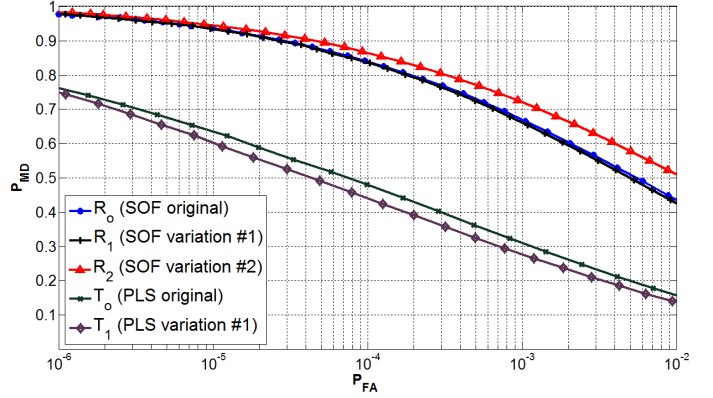


Fig. 2: ROC curves for SOF and PLS detectors at SNR = -3 dB

For SOF detector's algorithms,  $R_1(k)$  has a similar performance as  $R_o(k)$  but with a simpler implementation, achieving this only by taking the modulus of  $n_i(k)$  instead of its square value. In the case of  $R_2(k)$  algorithm shows also a similar performance as  $R_o(k)$  for  $P_{FA} < 10^{-5}$  with a difference of  $P_{MD} < 0.01$  but a little worse performance for  $P_{FA} > 10^{-5}$  but still have simpler implementation compared to both  $R_o(k)$  and  $R_1(k)$ . Taking only the modulus of  $m_i(k)$  for PLS detection,  $T_1(k)$  leads to an increase in performance reducing the probability of missed detection  $P_{MD}$  up to 0.04 compared with  $T_o(k)$  and also reducing implementation complexity.

### IV. GLOBAL SUMMATION OF SOF/PLS DETECTOR

Performance of header detection in DVB-S2 standard can be further improved by using different methods that combine SOF and PLS detection algorithms. Here we propose a Global Summation method, that will be called from now GLOBAL, which computes a new variable of decision  $G_{R,T}(k)$  that depends on which SOF (o:  $R_o$ , 1:  $R_1$ , 2:  $R_2$ ) and PLS (o:  $T_o$ , 1:  $T_1$ ) detection algorithms are used. This variable is then compared to a threshold  $T_{Global}$  to determine header detection. GLOBAL is given as

$$G_{R,T}(k) = \sum_{i \geq 1} p_i(k), \quad (10)$$

where  $p_i(k)$  is given by

$$p_i(k) = \max(|n_i(k') + m_i(k)|, |n_i(k') - m_i(k)|). \quad (11)$$

In (11),  $n_i(k')$  and  $m_i(k)$  are variables from SOF and PLS detection algorithms respectively and are calculated using (5) and (8). SOF detector algorithm is computed at time  $k' = k - 64$  in order to be synchronized with PLS detector algorithm.

In GLOBAL the only difference when using either original or variation algorithms is in the values of  $i$  taken for  $n_i(k')$ , that depends on which SOF algorithm is used (i.e.,  $R_o(k)$ :  $i \in [1 : 25]$  or  $R_1(k)$ ,  $R_2(k)$ :  $i \in \{1, 2, 4, 8, 16\}$ ). Considering the algorithms presented in section III, we apply GLOBAL and evaluate their performance at  $E_s/N_0 = -3$  dB and  $\Delta_f T_s = 0.1$ :

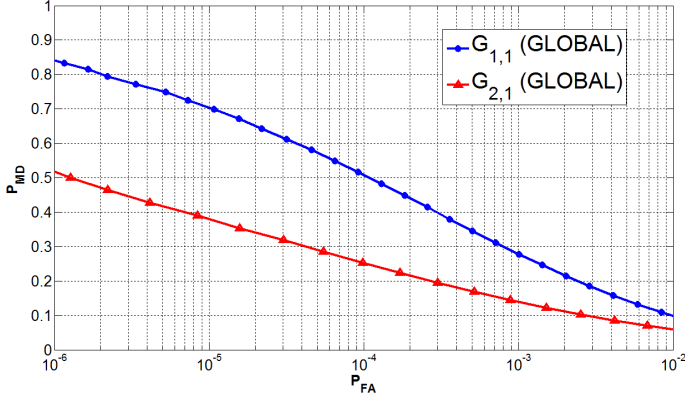


Fig. 3: ROC curves for Global Summation SOF/PLS detectors at SNR = -3 dB

In Fig. 2 although  $R_2$  algorithm presents worse performance than  $R_1$ , in GLOBAL with  $T_1$  ( $G_{2,1}$ ) presents better performance than the GLOBAL between  $R_1$  and  $T_1$  ( $G_{1,1}$ ) as shown in Fig. 3. The comparison between different header detectors is shown in Fig. 4. Compared detectors use different methods for combining SOF and PLS detection algorithms (i.e., JOINT, SINGLE and GLOBAL methods). JOINT method ( $J_{o,o}$ ) achieve header detection by comparing if both  $R_o(k')$  and  $T_o(k)$  are above two chosen thresholds  $T_{SOF}$  and  $T_{PLS}$  respectively. Detector using Single Summation method called SINGLE ( $S_{2,1}$ ), determines header detection by summing  $R_2(k')$  and  $T_1(k)$ , then compares the sum with a threshold  $T_{Single}$ . The ROC curve is obtained for  $E_s/N_o = -3$  dB and  $\Delta_f T_s = 0.1$ .

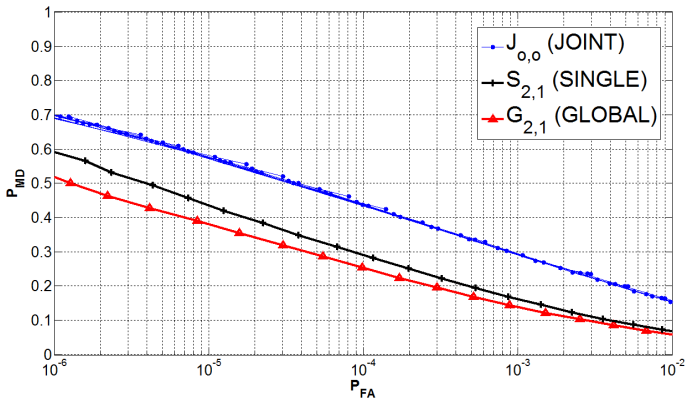


Fig. 4: ROC curves for header detectors at SNR = -3 dB

GLOBAL method presented here,  $G_{2,1}$ , reduces the  $P_{MD}$  in 0.18 for a  $P_{FA} = 10^{-6}$  compared to  $J_{o,o}$ , method presented

in [3]. Apart from improving performance, reducing the  $P_{MD}$  for a given  $P_{FA}$ , ( $G_{2,1}$ ) leads to a very simple implementation.

## V. ARCHITECTURE OF GLOBAL SUMMATION OF SOF/PLS DETECTOR

The overall architecture of the proposed header detector is presented in Fig. 5. At time  $k$ , the input data is the complex sample  $r(k)$ . The first step is to compute the phase  $\theta(k)$  of  $r(k)$  by a CORDIC (COordinate Rotation Digital Computer) [11]. For a hardware implementation,  $\theta(k)$  is represented in fixed point precision on  $n_\theta$  bits as  $\theta_q(k)$  using

$$\theta_q(k) = \lfloor \theta(k)/(2\pi) \times 2^{n_\theta} \rfloor, \quad (12)$$

where  $\lfloor x \rfloor$  represents the integer smaller or equal to  $x$ . The determination of  $n_\theta$  is a trade-off between performance and hardware complexity. In order to evaluate the minimum required precision, we have simulated header detection using  $G_{2,1}$  with  $\theta(k)$  quantized respectively on  $n_\theta = 3, 4, 5$  bits and also in floating point precision. The ROC curves Fig. 6 at  $E_s/N_0 = -2.35$  dB shows that  $n_\theta = 4$  is enough to have almost optimal detection performance ( $P_{MD}$  rises only from 0.3 to 0.33 for  $P_{FA} = 10^{-6}$ ).

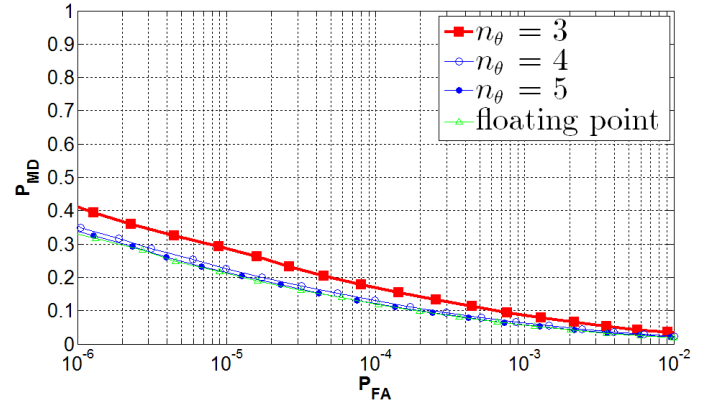


Fig. 6: ROC curves for  $G_{2,1}$  with  $\theta_q(k)$  at SNR = -2.35 dB

Once  $\theta_q(k)$  computed, the delayed versions  $\theta_q(k-i)$  for  $i \in \{1, 2, 4, 8, 16, 32\}$  are obtained by a chain of delay elements (FIFO registers), as shown in Fig. 5. The next operation is the computation of  $\theta_q(k) - \theta_q(k-i)$ , since  $\theta_q$  represents angles, this operation is done modulo  $2^{n_\theta}$ , i.e., the result is also on  $n_\theta = 4$  bits. The next step is the exponentiation of  $j(\theta_q(k) - \theta_q(k-i))$ , noted  $e^{j\theta}$  in Fig. 5, to go back to the Cartesian Coordinates. Simulation result (not shown in the paper) shows that  $n_c = 3$  bits are enough to code in 2's complement format the real and imaginary parts of  $e^{j(\theta_q(k) - \theta_q(k-i))}$ . The explicit mapping between angle and Cartesian Coordinates is given in Fig. 7. This operation can be implemented either by a small ROM (address of 4 bits, codewords of size  $3 + 3 = 6$  bits), or by a simple combinational logic. Complex samples  $e^{j(\theta_q(k) - \theta_q(k-i))}$  are then stored in Shift Registers (SR) of length  $64-i$  to be able to compute in parallel (8) to obtain  $m_i(k)$  and also  $n(k-38)$  (see (5)) to

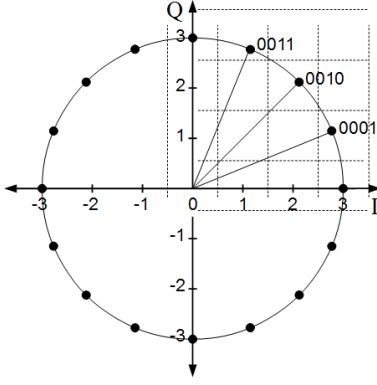


Fig. 7: Mapping of  $\theta_q(k)$  to real and imaginary parts

take profit as late as possible of the available data in the SR. Then,  $n_i(k - 38)$  is sent to a FIFO of size 26 to obtained  $n_i(k - 64) = n_i(k')$  used in (11) to compute  $p_i(k)$ . The hardware details for computing the real part (the architecture for the imaginary part is identical) of  $m_i(k)$  and  $n_i(k')$  is shown in Fig. 8, where  $I_i(k) = \mathcal{R}(e^{j(\theta_q(k) - \theta_q(k-i))})$  is the real part of  $e^{j(\theta_q(k) - \theta_q(k-i))}$ . One should note that the part of the hardware use to compute  $m_i(k)$  and  $n_i(k - 38)$  can be shared to further save hardware (in fact, both computation involve multiplication with 0, 1 or -1 coefficients and a final summation). Since the number of non-null elements in the filter are smaller or equal to 32, additional  $\log_2(32) = 5$  bits are required to encode the result with full precision, i.e., the real and imaginary parts of  $m_i(k)$  and  $n_i(k')$  are coded on  $n_c + 5 = 8$  bits. In Fig. 5 is shown the filter architecture  $H_i$  and  $G_i$ , inside the dashed rectangle, integrated with the overall header detector. This filter architecture only shows the real part of the complex processing to compute  $m_i(k)$  and  $n_i(k')$  (the architecture for the imaginary part is the same). Finally,  $p_i(k)$  for  $i \in \{1, 2, 4, 8, 16, 32\}$  are computed from  $m_i(k)$  and  $n_i(k')$ . Then, the final adder performs (10) to compute the variable of decision  $G_{2,1}(k)$ .  $G_{2,1}(k)$  is

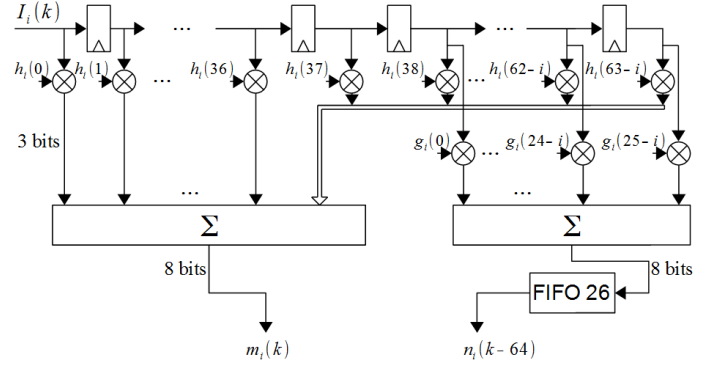


Fig. 8: Hardware implementation of  $G_i$  and  $H_i$  filters using SR and FIFO for real part

then compare with the threshold  $T_{Global}$  to determine header detection or not (not shown in Fig. 5). Finally, the critical path in Fig. 5) is rather long. Since the architecture does not contain loop, it can be pipelined as needed to achieve the targeted clock frequency. The computation for obtaining an output sample  $G_{2,1}(k)$  requires 6 additions (for performing  $\theta(k) + (-\theta(k-i))$ ), 6 accesses to ROM (for angle to Cartesian Coordinates conversion). In the filtering process,  $G_i$  filter requires  $\sum_{i=1,2,4,8,16} 25 - i = 94$  complex additions,  $H_i$  filter  $31 \times 6 = 186$  complex additions ( $G_i$  has 32 non-zero coefficients taking only values -1 or 1). Finally for global summation are required  $2 \times 5 = 10$  complex additions,  $2 \times 5 + 1 = 11$  additions (1 addition per modulus calculation) and 5 additions for the final summation. Table I compares and summarizes operations needed to compute the decision variable of Joint and Global Summation detectors.

## VI. CONCLUSIONS

A greatly improve in performance, in terms of ROC curves compared with state of the art detectors, has been achieved by Global Summation of SOF/PLS header detector thanks to GLOBAL (Global Summation method) that combines SOF

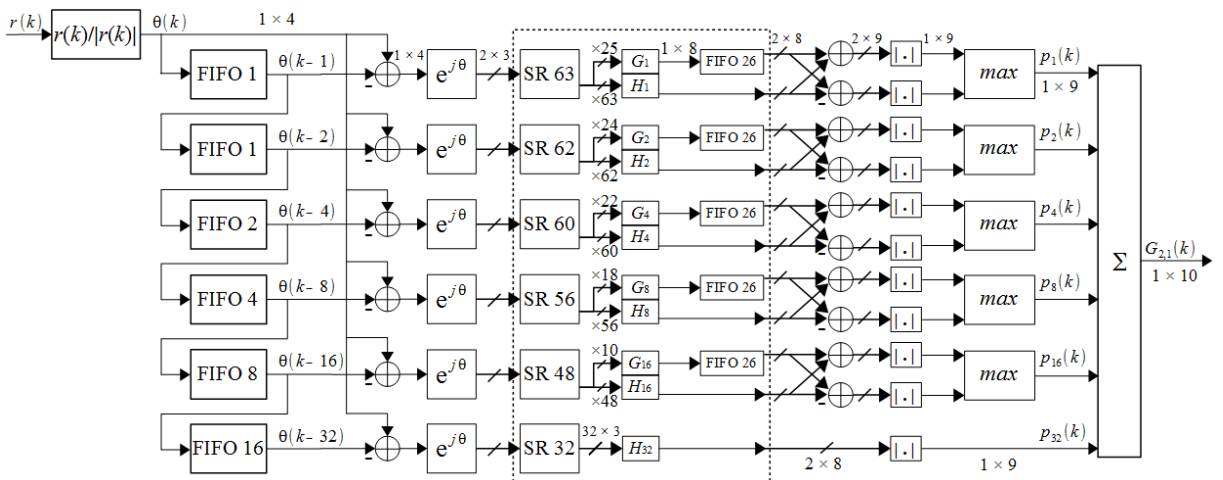


Fig. 5: Serial Architecture of Global Summation of SOF/PLS detector

| Operation        | JOINT | GLOBAL |
|------------------|-------|--------|
| access to ROM    | 25    | 6      |
| addition         | 54    | 22     |
| complex addition | 486   | 290    |
| exponentiation   | 31    | 0      |

TABLE I: Complexity estimation

and PLS detections. Also lower implementation complexity, compared with Joint detector [3], has been derived from this detector due to the simplifications proposed on SOF and PLS algorithms.

#### REFERENCES

- [1] ETSI EN 302 307, v. 1.1.2 *Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering and other broadband satellite applications*, June 2006.
- [2] X. Giraud, A. Duverdier, "Procede de reception de trames d'un flux numerique", Patent PCT number PCT/FR2006/002313, oct. 2005.
- [3] E. Boutillon; P. Kim, C. Roland, "Efficient multiplierless architecture for frame synchronization in DVB-S2 standard," Signal Processing Systems (SiPS), 2011 IEEE Workshop on , vol., no., pp.163,167, 4-7 Oct. 2011
- [4] J. L. Massey, "Optimum frame synchronization," *IEEE Trans. Commun.*, Vol. 20, no. 2, pp.115-119, 1972.
- [5] Z. Y. Choi, and Y. H. Lee, "Frame synchronization in the presence of frequency offset," *IEEE Trans. on Commun.*, Vol. 50, no. 7, pp.1162-1065, Jul. 2002.
- [6] J. A. Gansman, M. P. Fitz, and J. V. Krogmeier, "Optimum and sub-optimum frame synchronization for pilot-symbol-assisted modulation," *IEEE Trans. Commun.*, Vol. 45, pp.1327-1337, Oct. 1997
- [7] D.-U. Lee, P. Kim, and W. Sung, "Robust frame synchronization for Low Signal-to-Noise Ratio Channels Using Energy-Corrected Differential Correlation," *Eurasip Journal on Wireless Communications and Networking*, Vol. 2009.
- [8] P. Kim, et al, "Robust frame synchronization for the DVB-S2 system with large frequency offsets," *Wiley Int. Jour. of Satellite Communications and Networking*, Vol. 27, no. 1, pp.35-52, Jan-Feb 2009.
- [9] F. Sun, Y. Jiang, L. Lee, "Frame synchronization and pilot structure for second generation DVB via satellites," *Wiley Int. Journal of Satellite Communications and Networking*, Vol. 22, no. 3, pp.319-339, May-Jun 2004.
- [10] P. Kim, et al, "Enhanced frame synchronization for the DVB-S2 system under a large of frequency offset," in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC'07)*, , pp.1183-1187, Mar. 2007.
- [11] J. E. Volder, "The CORDIC Trigonometric Computing Technique", IRE Transactions on Electronic Computers, Sept. 1959.