ISBN 978-981-11-0008-6
Proceedings of 2016 6th International Workshop on Computer Science and Engineering (WCSE 2016)
Tokyo, Japan, 17-19 June, 2016, pp. 345-350
doi: 10.18178/wcse.2016.06.054

Threshold-based Finite State Machine for Frame Synchronization in Variable Coding and Modulation System

Yong Sheng[+], Jinshu Chen, Xiaoming Fan

Department of Electronic Engineering, 100084 Tsinghua University, Beijing, China

Abstract. In this paper, we present a robust and real-time frame synchronization scheme for second generation digital video broadcasting (DVB-S2) standard in Variable Coding and Modulation (VCM) system. Variable frame lengths, large frequency offset and extremely low SNRs make it hard to obtain frame headers in VCM. Our proposed scheme is based on differential detector and threshold-based finite state machine (FSM) to achieve fast frame synchronization. Simulation results indicate that our proposed scheme shares similar robustness with existing methods, while has low latency and implementation complexity than others.

Keywords: frame synchronization; DVB-S2; Variable Coding and Modulation (VCM); threshold-based; finite state machine.

1. Introduction

The second generation digital video broadcasting via satellite (DVB-S2) has been developed for a decade. Its distinctive transmission performance attracts much attention. To adapt to different applications, DVB-S2 provides Constant Coding and Modulation (CCM), Variable Coding and Modulation (VCM) and Adaptive Coding and Modulation (ACM) [1]. Many satellite missions are carried in a highly dynamic environment on account of variations of weather, geometry and other interferences. VCM system makes the best use of the link margin by transmitting more information bits when SNR is high, e.g., exploits 8/9 LDPC and 32APSK combination, and throttles down the information rate to assure basic communication performance when SNR is low, e.g., uses 1/2 LDPC and QPSK combination. Compared with CCM, frame synchronization in VCM system suffers from variable frame lengths, besides large frequency offset and extremely low SNRs.

Several algorithms have been studied over the issue of frame synchronization. Frame synchronization consists of acquisition and tracking. Acquisition is the process to obtain information about frame by detector. Choi and Lee proposed a differential detector known as CLD based on the approximate maximum likelihood (ML) criterion [2]. CLD is insensitive to large frequency offset but has high complexity. To decrease complexity, Non-coherent post detection integration (NCPDI) [3] and differential post detection integration (DPDI) [4] are proposed. Differential correlation scheme based on the start of frame (SOF) and physical layer signalling code (PLSC) is recommended in [5] in the trade-off between performance and complexity.

After acquiring differential correlators, frame headers can be tracked by comparing differential correlations with a pre-established threshold in [5]. Peak detector is presented in [6] without the assistance of threshold. Double peak searching scheme is analysed in [7] to improve robustness.

However, few investigations have been made over the issue of frame synchronization in VCM service. Even though VCM provides a magnificent improvement in throughput over CCM, it suffers from variable frame lengths during the detections of physical layer frame (PLFRAME) header. Multi-windows threshold-based scheme is proposed in [8] to cope with variable frame lengths. The scheme is implemented by comparing positions of all candidates whose differential correlations are above threshold. Though the

[+]Corresponding author. Tel.: +8618810918727.
E-mail address: cheng-y13@mails.tsinghua.edu.cn.

performance of this method is satisfactory, it is hard to implement in Field Programmable Gate Array (FPGA) for the difficulty in buffering all candidate positions in three successive windows. In this paper, we propose a finite state machine (FSM) based on threshold with low latency and complementation complexity. MATLAB simulations are carried, demonstrating that the performance of the FSM scheme is pretty satisfactory, even in low SNRs and large frequency offset condition.

The remaining of this paper is organized as follows. Section 2 describes the structure of DVB-S2 physical frame in VCM service. Threshold-based FSM is proposed in section 3. Analysis of the algorithm and performance comparisons are made in section 4. Finally, the conclusion is given in section 5.

2. Structure of DVB-S2 Physical Layer Frame

2.1. Physical layer frame structure

According to DVB-S2, Physical Layer Frame (PLFRAME) is shown in Fig. 1. Physical Layer Header (PLHEADER) occupies one SLOT (90 symbols). PLHEADER consists of two parts: SOF and PLSC. SOF shall correspond to the 26-symbols fixed pseudo-random pattern (18D2E82), assisting in frame synchronization and carrier recovery. PLSC is a binary code of 64 symbols, which is the result of the encoding of 7 signaling bits using a Reed-Muller code. These 7 bits consists of 5 bits of MODCOD to identify the coding rate and modulation and 2 bits of TYPE to identify the Forward Error Correction Frame (FECFRAME) length (64800 bits or 16200 bits) and the presence or absence of pilots. Pilot blocks (36 symbols) could insert every 16 SLOTs to make receiver synchronization and carrier recovery easier [8]. SOF and PLSC shall be utilized in differential detector in section 3.

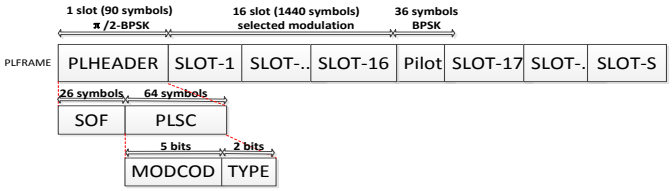

Fig. 1: Structure of physical layer frame

2.2. Variable PLFRAME lengths

According to DVB-S2, Forward Error Correction (FEC) Encoding consists of BCH outer code and LDPC inner code. Depending on different application, the length of FECFRAME shall be 64800 bits or 16200 bits. The FECFRAME shall be serial-to-parallel converted according to mapping rules, forming complex FECFRAME (XFECFRAME). Modulation set contains QPSK, 8PSK, 16APSK and 32APSK, and their corresponding modulation efficiency η_{MOD} are 2, 3, 4, and 5. The length of PLFRAME depends on the length of FECFRAME, modulation type and pilot inserting status.

Different frames could adopt different MODCODs, while all SLOTs (90 symbols per one SLOT) in one frame share the same MODCOD. $L_{FECFRAME}$ is the length of frame after FEC process, which is 64800 bits for normal FECFRAME or 16200 bits for short FECFRAME. L_{header} is the length of PLHEADER, which equals 90 symbols. *Pilot_Insert* is a Boolean variable, which is true for pilot presence and false for pilot absence. L_{pilot} is 36 symbols for one pilot block. S is the number of SLOTs, and int{.} is the integer function. For example, when FECFRAME is a sequence of 64800 bits, modulation type is QPSK, and pilot blocks are inserted, then S is 360 and $L_{PLFRAME}$ is 33282.

$$S = (L_{FECFRAME} / \eta_{MOD}) / 90 \tag{1}$$

$$L_{PLFRAME} = L_{FECFRAME} / \eta_{MOD} + L_{header} + Pilot\_Insert \bullet \left[L_{pilot} \bullet \text{int}\{(S-1)/16\} \right] \tag{2}$$

Capturing all the case as equations (1-2), physical layer frame lengths can be obtained in ascending order as follows:

Frame Length Set = {3330, 3402, 4140, 4212, 5490, 5598, 8190, 8370, 13050, 13338, 16290, 16686, 21690, 22194, 32490, 33282}. (3)

3. Proposed Frame Synchronization Scheme

Generally speaking, frame synchronization is composed of acquisition and tracking. Acquisition is the phase to calculate differential correlation, implementing by differential correlator. After obtaining differential correlations, header tracking shall determine the accurate location of PLHEADER, with the assistance threshold-based FSM, which is the innovation of this paper.

3.1. Acquisition: differential detector

To work smoothly in the presence of large frequency offset and extremely low SNR, ETSI recommends to correlate on both the SOF and PLSC differentially [5]. The differential correlation $\wedge$ can be obtained as

$$\wedge = \left| \sum_{k=1}^{25} r_k r_{k+1}^* C_{SOF\,k}^* + \sum_{m=1}^{45} R_{2m-1} R_{2m}^* C_{PLSC\,m}^* \right| \tag{4}$$

where r denotes the received symbol of SOF, R denotes the received symbol of PLSC, and the star mark denotes complex conjugate.

C_{SOF} and C_{PLSC} denote the corresponding differential coefficients obtained for SOF and PLSC respectively with index k and m, which can be obtained by

$$C_{SOF\,k} = s_k s_{k+1}^* = \pm j, k = 1, 2, ..., 24, 25, \tag{5}$$

$$C_{PLSC\,m} = s_{2m-1} s_{2m}^* = \pm j, m = 1, 2, ..., 44, 45, \tag{6}$$

where s represents the transmitted PLHEADER symbol.

3.2. Tracking: finite state machine

FSM consists of four states: search, check, lock and protect, as shown in Fig. 2. In the following part, Diff-cor stands for differential correlation obtained from differential detector for description simplicity. The description of each state is listed below:

- Search state: synchronizer stays in search state by default when it start, and it searches for a quasi-synchronous PLHEADER symbol by symbol. Quasi-synchronous PLHEADER is the symbol whose Diff-cor is higher than the pre-determined threshold. Once a quasi-synchronous PLHEADER is acquired, FSM moves towards check state.
- Check state: check state is defined to make the synchronizer more robust through decreasing the probability of false alarm. Check state contains M sub-check states, ranging check1 to checkM. M can be any non-negative integer. Verification is the process to compare Diff-cor with threshold, and if the Diff-cor in current detection position is above threshold, verification is deemed as successful. If M verifications are successful, then FSM goes to the lock state. Otherwise, the synchronizer returns to search state to start a new round of searching.
- Lock state: lock state is the state that synchronizer comes to a steady synchronization status. If Diff-cor in current detection position is above threshold, the synchronizer maintains in lock state, otherwise, it moves to protect state.
- Protect state: protect state is used to decrease the probability of miss detection of true PLHEADERs. Protect state contains N sub-protect states ranging from protect1 to protectN. N can be any non-negative integer.

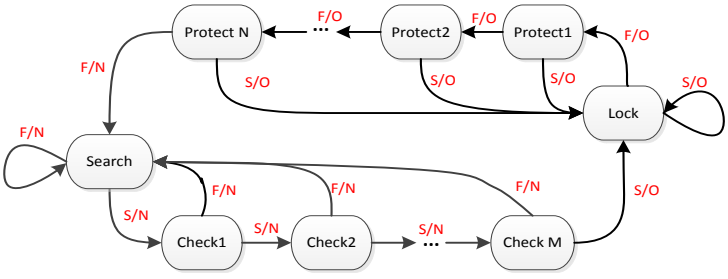

Fig. 2: Flow graph of the FSM

FSM is a Mealy machine, taking differential correlations as input and the position pulse of PLHEADERs as output. In Fig. 2, every arc is labelled with "input/output". The left of slash is the result of the comparison between Diff-cor of current position with threshold. The result can be S, meaning that Diff-cor is above threshold, or F, representing the opposite. The right part of the slash is output, determined by current state and comparison result. Output also composes of two types: N means none to output while O represents to output the pulse of position as PLHEADER. Threshold is the pivotal element in balancing the probability of false alarm (PFA) and the probability of miss detection (PMD).

As mentioned above, FSM transits among four states as Fig. 2 shown, taking the comparison result between Diff-cor in current detection position with threshold as input and declaration of frame synchronization as output. The computation of current detection position is crucial in designing our proposed algorithm. There are two detection position in FSM: previous detection position pos_{pre} and current detection position pos_{cur}. Two positions have the following relationship,

$$pos_{cur} = pos_{pre} + offset \tag{7}$$

In search state, FSM needs to search the Quasi-synchronous PLHEADER symbol by symbol. In other states, *offset* equals fixed frame length in CCM service, while it shall change under different conditions in VCM service because of changeable frame lengths. As shown in Tab 2, receiver has no prior information about frame length in Check1 state, so *offset* needs to take every possible value in Frame Length Set in equation (3). Once *offset* meets the requirement, FSM stores *offset* into *register* for the utilization in other states.

Table 1. Computation of *offset*

State	Search	Check1	Check2,… , CheckN	Lock	Protect
CCM	1 symbol	Fixed known frame length			
VCM	1 symbol	Iterates *offset* through Frame Length Set: 1. $pos_{cur} = pos_{pre} + offset$ 2. if Diff-cor(pos_{cur}) > threshold *register* = *offset*; break;	*offset* = *register*		

Example is given in Fig. 3. There are four steps to show the working principle. FSM starts with Search state and transits to Check1 state when a successful comparison is made as step (1) shows. In Check1 state, every possible offset needs to be checked as step (2-3). Once Diff-cor is above threshold as step (3), offset can be stored, and FSM transits to Check2 or Lock state.

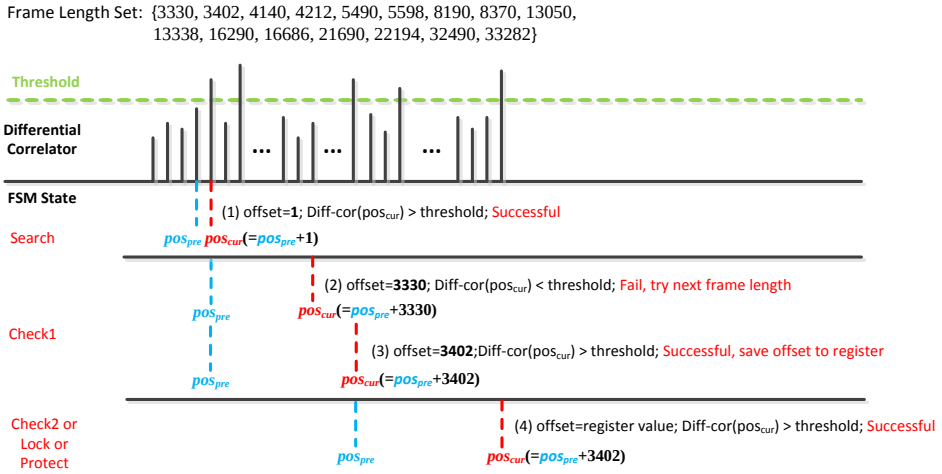

Fig. 3: Example of FSM.

4. Analysis of the Algorithm

4.1. Relationship between parameters and performance

When improperly setting of M, synchronizer may be in false lock frequently in low SNRs; when improperly setting of N, synchronizer shall fall into loss of lock easily in low SNRs. The bigger M and N are

set, the smaller PMD and PFA shall be, but this comes at a cost in implementation complexity. According to MATLAB simulations, N and M are suggested to equal 3 to balance complexity and performance, and in this situation, normalized threshold can range from 0.6 to 0.8 to make PMA and PFA as small as possible.

4.2. Performance comparison of different algorithms

As mentioned before, frame synchronization consists of acquisition and tracking. Acquisition is the phase to obtain Diff-cor. Tracking is to find out PLHEADER according to Diff-cor, which is the concern of this paper. Tracking algorithms mainly include Peak Search Algorithm [6], Double Peak Search Algorithm [7], Multi-windows Threshold-based Algorithm [8] and our proposed FSM Algorithm. Their main procedures are shown in Table 2.

Table 2. Main Procedure of different tracking algorithms

Algorithms	Main Procedure
Peak Search	Iterates window length through Frame Length Set: 1. Select maximum peak Diff-cor in window 2. Do post verification 3. If result is sufficient strength, break;
Double Peak Search	Iterates window length through Frame Length Set: 1. Repeat in 3 successive windows Add maximum and sub-maximum peak Diff-cor to corresponding candidate set in window 2. Compare 6 peak position in 3 candidate sets. 3. If a pair is of same, break;
Multi-windows Threshold-based	Iterates window length through Frame Length Set: 1. Repeat in 3 successive windows Add all position whose Diff-cor is above threshold to corresponding candidate set in window 2. Compare all position in 3 candidate sets. 3. If a pair is of same, break;
Threshold-based FSM	As description in section 3

Experiments are done to investigate performance of different methods. With 20% frequency offset of transmission rate, the performance comparison of different algorithms at different SNRs are shown in Fig. 4. N and M are set as 3, and threshold is set as 0.78 according to the analysis of previous section. Our proposed threshold-based FSM algorithm shares the similar frame synchronization probability with multi-windows threshold-based algorithm, which outperforms peak search algorithm and double peak search algorithm in wide range.

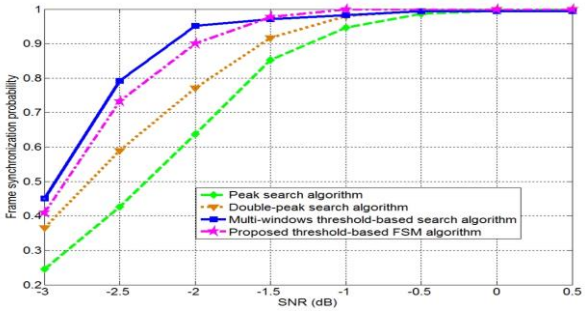

Fig. 4: Frame synchronization probability versus SNR of different algorithms

Table 3 presents comparisons of different tracking algorithms, including algorithm delay, comparison times and resource consumption. Len_{window} denotes the length of search window. Peak Search algorithm has a delay of $1*Len_{window}$ symbols because of waiting for all symbols to come in the search window before selecting maximum peak Diff-cor. FSM is a real-time algorithm for it can determine whether frame synchronization is achieved based on the current state of FSM and the Diff-cor of current position. As described in Table 2, all methods do not involve complex operation other than basic control flow and comparisons, so the number of comparison times can be an evaluation factor. The third column in Table 3 shows comparison times in a frame, indicating that FSM only needs 1 comparison in pre_{cur} position.

Resource consumption is also another important factor in performance comparison. Our FSM scheme only buffer four variables while other three methods have to take up storage space for recording search window endpoints and multiple peak positions.

Table 3. Comparisons of different tracking algorithms

Algorithms	Delay (symbol)	Comparison Times	Resource Consumptions
Peak Search	$1*Len_{window}$	Len_{window} -1 comparisons to find peak Diff-cor	windows left & right endpoints + 1 peak position
Double Peak Search	$3*Len_{window}$	$3*2*(Len_{window} - 1)$ comparisons to find double peak Diff-cor + 12 comparisons to find the pair	windows left & right endpoints + 3*2 peak positions
Multi-windows Threshold-based	$3*Len_{window}$	$3*Len_{window}$ comparisons to find satisfying Diff-cor + comparisons between 3 candidate sets; Associate with threshold setting	windows left & right endpoints + 3*(all symbols whose Diff-cor is above threshold); Associate with threshold setting
Threshold-based FSM	No delay, Real-time	1 comparison in pre_{cur} position	pre_{cur} position + pre_{pre} position + *offset* register + *state* register

Compared with existing approaches, our proposed FSM scheme is easy to implement in FPGA because of its small comparison times and less resource consumption. Moreover, real-time and satisfactory performance at low SNRs make FSM scheme stand out among all these methods in VCM service which requires fast and robust frame synchronization.

5. Conclusion

In this paper, frame synchronization of DVB-S2 in VCM service is studied. Variability of frame length makes it hard to achieve fast synchronization in VCM system, accompanied with low SNRs and large frequency offset. Threshold-based finite state machine possesses robustness, less time delay, small resource consumption and implementation complexity than existing approaches.

6. References

[1] Digital video broadcasting (DVB); second generation framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering and other broadband satellite applications. ETSI EN 302 307-1 V1.4.1, 2014.11.

[2] Choi Z Y, Lee Y H. Frame synchronization in the presence of frequency offset. Communications, IEEE Transactions on, 2002, 50(7): 1062-1065.

[3] Corazza G E, Pedone R, Villanti M. Frame acquisition for continuous and discontinuous transmission in the forward link of satellite systems. International Journal of Satellite Communications and Networking, 2006. 24(2): pp. 185-201.

[4] Kim P, Corazza G E, Pedone R, et al. Enhanced frame synchronization for DVB-S2 system under a large of frequency offset. In Wireless Communications and Networking Conference, 2007. WCNC 2007. IEEE, 2007: 1183-1187.

[5] Digital video broadcasting (DVB); User guidelines for the second generation system for broadcasting, interactive services, news gathering and other broad-band satellite applications (DVB-S2), European Telecommunications Standards Institute (ETSI), TR 102 376, 2005.

[6] Sun F W, Jiang Y, Lee L N. Frame synchronization and pilot structure for second generation DVB via satellites. International journal of satellite communications and networking, 2004, 22(3): 319-339.

[7] Zhang Y, Chen X, Fan W, et al. Robust and reliable frame synchronization method for dvbs2 system. In Wireless Telecommunications Symposium (WTS), 2010, pp. 1-5.

[8] Sheng Y, Chen J S, Fan X M, Multi-windows Threshold-based Frame Synchronization in Variable Coded Modulation Service, 2016 8th International Conference on Computer and Automation Engineering, ICCAE 2016.