

# A Framework for Adaptive PCR Jitter Correction in MPEG-2 TS Processors

Heitor J. Savino, Eddie B. L. Filho

**Abstract**—Currently, the MPEG-2 Transport Stream (TS) is widely used in many transmission systems, like digital terrestrial television and internet protocol television networks. It is capable of carrying interleaved audio, video, system information, timing and generic data, in an unified transport framework. Regarding receiver synchronization, it employs the Program Clock Reference (PCR), which is a snapshot of the transmitter 27 MHz clock and is used to adapt the local one. However, the PCR may present inaccuracies on its time base information, due to reprocessing or network variable delay. This paper introduces a methodology for adaptive PCR correction, which has the potential to reduce the output jitter due to stream modifications performed by the TS processor, depending on the characteristics of the input TS. Simulation results show that the proposed method is effective and outperforms traditional schemes presented in the literature, by keeping the output jitter level as close as possible to the one at the input.

**Keywords**—PCR correction, MPEG-2, transport stream, semaphore, rate adaptation.

## I. INTRODUCTION

The Transport Stream (TS) is a transport technology described in the MPEG-2 standard [1], which is used for transparent transmission of multimedia data through communication networks, like internet protocol television systems. In order to correctly display video, audio and other data, it employs the Program Clock Reference (PCR), which is just a snapshot of the current state of the transmitter 27 MHz counter and bears all necessary timing information. If it is equal to the current value presented by the receiver hardware, the system is perfectly synchronized; otherwise, a phase-locked loop (PLL), driven by the PCR difference, adapts the local 27 MHz clock [1]. This kind of synchronization procedure is intended for constant-delay systems, like digital terrestrial television networks, in such a way that distances between PCR values are not changed and the PLL is perfectly capable of correcting clock deviations.

However, when the TS is further processed, for instance by including new programs (remultiplexing) or converting from an interface type to another [2], the relative position of PCR values may change, due to the rate adaptation procedure, which may result in inaccuracies of the PCR information, known as jitter [3]. This is given due to the its constant value, which no longer represents difference, in bytes, between the previous one. In order to keep PCR deviation within acceptable

bounds, and avoid compromising the synchronization procedure, a PCR control module is often employed, which tries to compensate for the new PCR locations in the stream.

According to the MPEG-2 standard, the maximum jitter must be around  $\pm 500$  ns [1]; any value beyond this limit may not be compensated by the PLL. Besides, this issue is very important for transport stream decoders, like set-top boxes, because they use the recovered 27MHz clock for creating the video interface sample frequencies [4].

Some authors [5], [6], [7], [8], [9] have devised solutions for the PCR jitter caused by MPEG-2 TS processors, by correcting PCR values to account for changes in their relative distances. Two traditional frameworks may be found in the literature: a set of counters [5], [6], which are associated to programs in the TS and are also loaded with the respective PCR values, and a single counter [7], [8], [9], which provides intermediate time stamps. There is also a more recent variant, proposed by Machmerth and Stoerte [10], in which the 27 MHz counters are replaced by accumulators, driven by the output byte clock.

So far, the existing solutions try to modify PCR values in order to reflect their new locations in the stream, in the same manner for every TS, without taking into account their differences. Besides, none of them addresses a very important source of jitter: the mismatch between rate adaptation and granularity of PCR correction, which is mainly given by the relation among the output rate, the input rate and the 27 MHz clock frequency. As a result, even when the PCR correction is carefully embedded into TS processors, there is an increase in the output jitter.

Based on what was described in the last paragraphs, if the rate adaptation algorithm works cooperatively with the PCR correction module, in an adaptive manner regarding the input TS, the system may be capable of reducing the output jitter. The present paper introduces a new framework that provides joint rate adaptation and PCR correction, in such a way that PCR packets are scheduled to be transmitted when the predicted jitter is small enough.

This paper is organized as follows. Section II describes the existing methods, highlighting their main features. In section III, the new framework for joint rate adaptation and PCR correction is presented, along with its extension for multiple programs. Finally, section IV provides simulation results and section V presents the conclusions of the work.

## II. PCR CORRECTION METHODS AVAILABLE IN THE LITERATURE

The jitter is perceived as a mismatch regarding the time interval between two consecutive PCR packets, computed with the respective PCR values, and the same elapsed time calculated with the original TS bit rate. Given that, one can

Heitor J. Savino<sup>‡</sup>, Eddie B. L. Filho<sup>\*†</sup>. <sup>‡</sup> Graduate Program in Electrical Engineering - Federal University of Minas Gerais - UFMG, Av. Antônio Carlos, 6627, Belo Horizonte - MG, 31270-901, Brazil. <sup>\*</sup>Federal University of Amazonas - UFAM, Av. Gen. Rodrigo Octávio Jordão Ramos, 3000, Manaus - AM, 69077-000, Brazil. <sup>†</sup>Science, Technology, and Innovation Center for the Industrial Polo of Manaus - CT-PIM, Rua Salvador, 391, Adrianópolis, Manaus-AM, 69057-040, Brazil. E-mails: heitorjs@ufmg.br, eddie@ctpim.org.br.

note that PCR values in a TS may be corrected with the addition of a variable displacement term, as in

$$PCR_{corr} = PCR + \Delta PCR_{comp}, \quad (1)$$

where  $PCR_{corr}$  is the new PCR value,  $PCR$  is the value from the incoming TS and  $\Delta PCR_{comp}$  is the displacement term, which is normally calculated with the aid of 27 MHz counters. This simple approach is the base of the following traditional methods.

#### A. The dedicated PCR counters method

An MPEG-2 TS may have many programs, with different time bases, which is commonly called a Multiple Program Transport Stream (MPTS) [1]. Given that, an intuitive solution for the PCR correction problem for MPTS would be to use as many 27 MHz counters as available programs [5], [6].

In order to accomplish that, the correction module must identify packets carrying PCR information and extract it; the resulting data is then loaded into respective counter registers. After performing rate adaptation, through the insertion of null packets, PCR data are then scheduled to leave the system, with its output value given by the replacement by the content of allocated counters.

Initially, one can argue that there must be as many counters as programs in the TS; however, different programs may share the same time base [1]. This way, in practical correction modules based on the dedicated counters method, there should be as many counters as different time bases.

#### B. The compensation method

The main distinguishing feature of the compensation method [7], [8] is the use of only one 27 MHz counter, which saves a lot of resources. It is accomplished by the use of two counter states: if a PCR packet is detected, the current counter state ( $Count_{in}$ ) is taken and subtracted from the respective PCR value ( $PCR_I$ ), which is then used for replacing the original PCR data in the adaptation field. When the same packet leaves the system, the PCR stored in the adaptation field is once more changed, by taking the current counter state ( $Count_{out}$ ) and adding it to the stored value. This way, the output value of the PCR is given by the replacement by a new stamp, computed as

$$PCR_I(i) = PCR(i) - Count_{in}(i) \quad (2)$$

$$PCR_{corr}(i) = PCR_I(i) + Count_{out}(i) \quad (3)$$

$$PCR_{corr}(i) = PCR(i) + \Delta PCR_{comp}(i), \quad (4)$$

where  $\Delta PCR_{comp}(i) = Count_{out}(i) - Count_{in}(i)$  is the compensation term for  $PCR(i)$ .

#### C. The accumulators method

In the accumulators method [10], PCR values are corrected based on the current value of an accumulator, which is initialized with the first extracted PCR value of the input TS. At each output-byte clock trigger, the accumulator value is then incremented with an accumulation rate given by

$$accu\_rate = \frac{sys\_clk\_freq}{out\_ts\_rate}, \quad (5)$$

where  $sys\_clk\_freq$  is the system clock, that is, 27 MHz, and  $out\_ts\_rate$  is the output rate, in bytes per second. As the PCR is not refreshed at every incoming PCR value,  $accu\_rate$  must be adjusted, in order to keep the current accumulation as near as possible to the expected values, which is done through

$$\Delta_{accu\_rate} = \frac{PCR_{accu} - PCR_{arr}}{num\_packet} \quad (6)$$

and

$$accu\_rate_{new} = accu\_rate + \gamma \cdot \Delta_{accu\_rate}, \quad (7)$$

where  $PCR_{accu}$  is the current accumulator value,  $PCR_{arr}$  is the PCR that just arrived,  $num\_packet$  is the number of packets since the last  $PCR$ , and  $\gamma$  is the convergence speed.

Indeed, this method employs a different approach, when compared to the dedicated counters or the compensation method, since each PCR value is not directly corrected and instead a different time base is created, which then converges to the original one. Besides, if there is any sudden change in the input TS behavior, there can be some steep variations the output jitter. It is worthy noticing that it is possible to use as many accumulators as programs, as in the dedicated counters method, or only one accumulator, with the same logic used in the compensation method.

### III. THE NEW FRAMEWORK FOR JOINT RATE ADAPTATION AND PCR CORRECTION

In the proposed scheme, instead of the 27 MHz clocks, the PCR correction module employs two accumulators, the output accumulator, triggered by the output TS rate, and the input accumulator, for compensating for the instant which PCR values are loaded into the output accumulator, in order to minimize the phase difference, in periods of the 27 MHz clock, between the input and output instants of time. In this way, the joint rate adaptation and PCR correction algorithm tries to predict the most favorable moment for sending PCR packets. Even if a MPTS is processed, only one set of accumulators is employed, which is possible due to the use of a semaphore structure [11] for shared access.

#### A. Correcting PCR values: the output accumulator

The proposed methodology uses a different clock source for computing the number of elapsed 27 MHz triggers. This is performed through an accumulator, which is triggered by the output TS rate and initialized with the current PCR value. Its accumulation rate is then given by

$$out\_accu\_rate = \frac{sys\_clk\_freq}{out\_ts\_rate}, \quad (8)$$

where  $out\_accu\_rate$  contains the number of PCR ticks, elapsed in a byte sent through the output interface, and  $out\_ts\_rate$  is the output rate, in bytes per second, represented in float number, for any feasible rate. So, this accumulator contains the PCR value that should have been computed during a period of  $N$  output bytes.

### B. Estimating the mismatch between input and output: the input accumulator

As any incoming PCR value is immediately loaded into the output accumulator and increased by the output byte rate, there is an intrinsic error caused by the phase difference between the input byte and the trigger of the output accumulator. This phase difference may be estimated by an input accumulator, triggered by the input TS clock, with a rate

$$in\_accu\_rate = \frac{out\_ts\_rate}{in\_ts\_rate} \bmod 1, \quad (9)$$

where  $0 \leq in\_accu\_rate < 1$ , given that  $\bmod 1$  gets the fractional part, which contains the phase difference at each incoming byte, in terms of the output byte period, and  $in\_ts\_rate$  is the input rate, in bytes per second. The input accumulator is then increased by taking only its fractional part

$$in\_accu = (in\_accu + in\_accu\_rate) \bmod 1, \quad (10)$$

where  $0 \leq in\_accu < 1$  is the input accumulator value and corresponds to the cumulated phase difference, which is related to the time at which a PCR value arrives.

At the moment a PCR value is loaded into the output accumulator,  $in\_accu$  is taken and then multiplied by  $out\_accu\_rate$  of (8), resulting in the Byte Phase Difference ( $BPD$ ) value, which is a measure of displacement between the last trigger of the output accumulator and the moment at which a PCR packet is loaded into the output accumulator.

### C. The complete PCR correction procedure

If the system must handle an MPTS, one can argue that there must be as many accumulators as different time bases. However, if a shared access structure is employed, it is possible to have only one accumulator. One possible solution is to use the compensation approach, as presented in section II-B; However, even with only one program or PCR values of different time bases far apart from each other, many arithmetic operations are performed, which increases the computation complexity of the whole infrastructure. In the present work, it was decided to use the semaphore controlled infrastructure, introduced by Savino and Filho [11]. Its main advantage is that the majority of correction procedures are performed by direct loading the current PCR into the correction module, without the use of intermediate stamps. The block diagram of the proposed approach is depicted in Fig. 1.

As shown in Fig. 1, two semaphores are used: one for the input and another for the output, both initialized with 0. Every time a packet containing PCR information is sent to the input buffer, the input semaphore is increased; in the same fashion, when a PCR packet leaves the rate adaptation module, the output semaphore is the one to be increased. If the input and output semaphores hold the same value, both are set to 0. When the first PCR packet arrives, its PCR value is extracted and the accumulator is directly loaded with  $PCR(i) - BPD$ . The next packets will then find a semaphore value greater than 0, indicating that the accumulator is already occupied. In that case, instead of using a dedicated accumulator, the system generates a temporary stamp for this packet, based on the current incoming PCR value and the output accumulator, which is given by

$$PCR\_M(i) = PCR(i) - out\_accu\_in(i) - BPD, \quad (11)$$

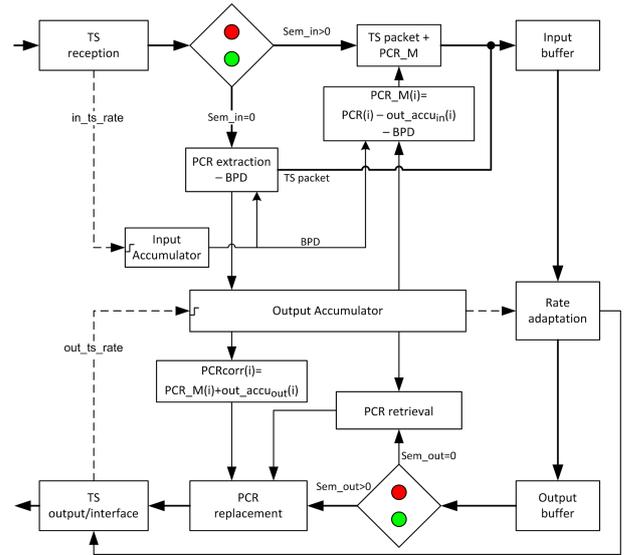


Fig. 1. The block diagram of the semaphore-controlled accumulator method.

where  $out\_accu\_in$  is the output accumulator content, when the PCR packet arrives. As soon as the first packet is ready to leave, it will find an output semaphore equal to 0, meaning that the corrected PCR associated to it must be directly retrieved from the accumulator. When the other packets leave the system, the semaphore value greater than 0 will indicate that the complete correction procedure must be carried out with

$$PCR_{corr}(i) = PCR\_M(i) + out\_accu\_out(i), \quad (12)$$

where  $out\_accu\_out$  is the output accumulator content, when the PCR packet needs to leave the system.

### D. The integrated PCR correction and rate adaptation

When the rate adapter needs to send a PCR packet, it computes the  $PCR_{corr}$  value that should be written if the packet is sent now or some time in the future. This way, it generates a vector of possible PCR values for the  $n$  future packets, called *holding\_vector*, with  $n$  varying from 0 (send the PCR packet now) to a maximum number given by

$$max\_hold = \frac{(buff\_size - pack\_n)}{PCR\_n} \cdot \frac{output\_TS\_rate}{input\_TS\_rate}, \quad (13)$$

where  $buff\_size$ ,  $pack\_n$ , and  $PCR\_n$  are the size of the rate adaptation buffer, the number of stored packets, and the number of packets containing PCR values, currently in the buffer, respectively.

After creating the *holding\_vector* array, the rate adapter will look for the position  $n$ , throughout the vector, with minimal fractional part, and then send  $n$  null packets before the current PCR packet.

## IV. SIMULATION RESULTS

Simulations for two likely scenarios were carried out. In the first one, 30 seconds of a TS containing 2 programs, with zero PCR jitter, were processed by two versions of a SPI to SPI converter [2]: one implemented with the proposed joint rate adaptation and PCR correction scheme and other with

the compensation method. The bit rate of the multiplex is about 32.507936 Mbps, with 15.52 Mbps for the first program and 600 Kbps for the second one. Results regarding the first program were analyzed, in order to evaluate the jitter at the output of the system. The output rate was set to 43 Mbps, which is accomplished, in the rate adaptation module, by inserting null packets throughout the original TS.

In Fig. 2, one can notice that the jitter inserted by the proposed scheme is below 7ns, which is less than 2% of the maximum permitted value, that is, 500ns [1]. However, the compensation-method based implementation provided a maximum output jitter smaller than 40ns, which is a much higher result.

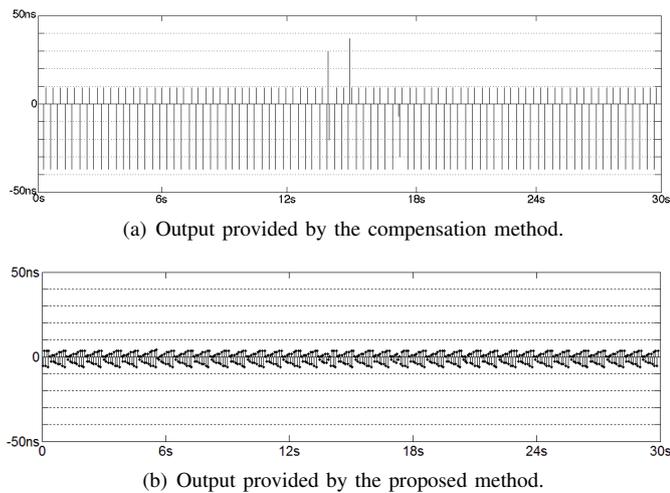


Fig. 2. Simulation results for the Program 1 of the first TS.

In the second setup, 30 seconds of a TS with 14 programs and 8 different time bases were processed by a rate converter with the proposed method. The bit rate of the multiplex is about 24.1Mbps, and results regarding the first program, which presents a bit rate of 1.85Mbps, are shown in Fig. 3. The output stream is configured with a bit rate of 43Mbps.

During the simulation using the presented MPTS, the evolution of semaphore parameters presented a maximum value of 4, meaning that, without the proposed shared access structure, at least four accumulators would be necessary.

In Fig. 3, one can notice that the output jitter is stable for the chosen programs and the difference is almost non perceived for the scale of 500 ns, given that the jitter inserted is around 5 ns.

It is worth noticing that the system can only be adapted to an output rate lower than the input rate if the remultiplexer filters some program or remove some null packets. Indeed, the minimum output rate is given by the data rate of the programs to be transmitted.

## V. CONCLUSIONS

A joint rate adaptation and PCR correction scheme was presented, in order to correct the PCR jitter caused by the rate adaptation module, during a rate conversion process. The proposed method introduces less jitter, when compared to traditional schemes, where rate adaptation and PCR correction operate independently. The system complexity is higher than

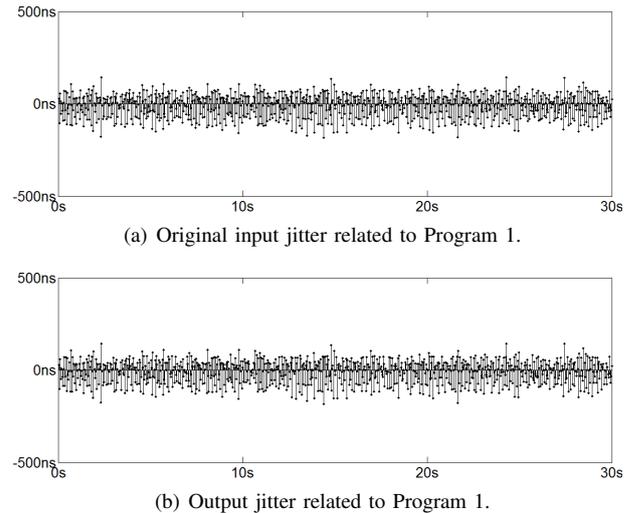


Fig. 3. Simulation results for the first program of the second TS.

the average figure presented by the traditional methods, due to the floating point operations, however, the results show that it is effective and, for instance, can be employed on professional equipments, where a precise output time base is needed.

## REFERÊNCIAS

- [1] ISO/IEC 13818-1: Information technology - Generic coding of moving pictures and associated audio information - Part 1: Systems. 3rd ed. Geneva, Switzerland: 2007.
- [2] ETSI EN 50083-9: Cable distribution systems for television, sound signals and interactive multimedia signals; Part 9: Interfaces for CATV/SMATV Headends and Similar Professional Equipment for DVB/MPEG-2 Transport Streams, 2003.
- [3] C. Tryfonas, and A. Varma, "Timestamping Schemes for MPEG-2 Systems Layer and Their Effect on Receiver Clock Recovery", *IEEE Transactions on Multimedia*, Vol. 1, No. 3, pp. 251-263, September 1999.
- [4] W. Fischer, "Digital Video and Audio Broadcasting Technology: A Practical Engineering Guide. 3rd ed. Berlin, Germany: Springer, 2010.
- [5] S. I. Lee, S. B. Cho, J. H. Kim, H. H. Jeon, and D. G. Oh, "Implementation of MPEG-2 TS Remultiplexer and Data Transport Unit for HDTV Satellite Broadcasting", *IEEE Transactions on Consumer Electronics*, Vol. 43, No. 3, pp. 324-329, August 1997.
- [6] Y. He, J. Zhou, and Y. Zhou, "Counter-Set Based PCR Jitter Correction Method for DVB-T System", in *proc. of the International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 2940-2943, Shanghai, September 2007.
- [7] W. Xingdong, Y. Songyu, and L. Longfei, "Implementation of MPEG-2 transport stream remultiplexer for DTV broadcasting", *IEEE Transactions on Consumer Electronics*, Vol. 48, No. 2, pp. 329-334, May 2002.
- [8] L. Longfei, Y. Songyu, and W. Xingdong, "Implementation of a New MPEG-2 Transport Stream Processor for Digital Television Broadcasting", *IEEE Transactions on Broadcasting*, Vol. 48, No. 4, pp. 348-352, December 2002.
- [9] Y. He, J. Zhou, and Y. Zhou, "Implementation of TS De-multiplexer with FPGA in DVB-IP Gateway for Network TV", presented at the *2005 IEEE 7th Workshop on Multimedia Signal Processing*, Shanghai, China, October 2005.
- [10] M. Machmerth, and C. Stoerte, "Accumulator based PCR restamping", in *proc. of the International Symposium on Consumer Electronics*, pp. 863-865, Kyoto, Japan, May 2009.
- [11] H. J. Savino, and E. B. L. Filho, "Um Contador Controlado por Semáforos Para a Correção de PCR", in *proc. of the Brazilian Symposium on Telecommunications*, Brasília, Brazil, Sep. 2012 (In portuguese).