CRC-ASSISTED ERROR CORRECTION IN A TRELLIS CODED SYSTEM WITH BIT STUFFING

Raoul Prévost[1,2], *Martial Coulon*[1], *David Bonacci*[2], *Julia LeMaitre*[3], *Jean-Pierre Millerioux*[3] *and Jean-Yves Tourneret*[1]

[1] University of Toulouse, INP-ENSEEIHT/IRIT, 2 rue Charles Camichel, BP 7122, 31071 Toulouse cedex 7, France
[2] TéSA, 14-16 Port Saint-Étienne, 31000 Toulouse, France [3] CNES, 18 Avenue Edouard Belin, 31400 Toulouse, France
{raoul.prevost, david.bonacci}@tesa.prd.fr, {martial.coulon, jean-yves.tourneret}@enseeiht.fr,
{julia.lemaitre, jean-pierre.millerioux}@cnes.fr

ABSTRACT

This paper introduces a new error correction strategy using cyclic redundancy checks (CRC) for a trellis coded system in the presence of bit stuffing. The proposed receiver is designed to simultaneously demodulate, decode and correct the received message in the presence of bit stuffing. It is based on a Viterbi algorithm exploiting the conditional transitions of an appropriate extended trellis. The receiver is evaluated with automatic identification system (AIS) messages constructed with a 16 bit CRC and a Gaussian Minimum Shift Keying (GMSK) modulation. The stuffed bits are inserted after any sequence of five consecutive bits 1 as requested by the AIS recommendation. Simulation results illustrate the algorithm performance in terms of packet error rate. A gain of more than 2.5dB is obtained when compared to the conventional GMSK receiver.

Index Terms— AIS, CRC, trellis codes, error correction, bit-stuffing, Viterbi decoding.

1. INTRODUCTION

The problem addressed in this paper is the demodulation of automatic identification system (AIS) [1] messages received by a satellite. In this context, the noise and interference levels are higher than when received from the earth and message retransmission cannot be requested. It is therefore necessary to develop error correction methods in order to obtain acceptable packet error rates (PER).

Cyclic redundancy checks (CRC) are widely used to detect errors in data transmissions. Many strategies have been proposed to optimize the CRC implementation. Although CRC were originally proposed to detect errors, they can also be used for error correction. An interesting CRC-assisted error correction method uses the difference between the received CRC and the CRC computed from the received data as a syndrome to detect the location of the erroneous bit [2]. This method was initially proposed to correct one error in the received message. It was generalized to correct two erroneous bits of a message in [3]. Keeping the syndrome idea, a correction method for multiple bit errors was studied in [4], where the low confidence bits are considered in priority. A correction strategy based on bit error probability was also proposed in [5]. The idea is to change the bits with high error probability until the received and re-computed CRC's are equal. Finally, a correction algorithm based on convolutional codes (CC) assisted by a CRC was considered in [6]. However, all of these methods fail when the data messages are subject to bit stuffing. This paper studies a CRC-assisted error correction in a trellis coded (TC) system with bit stuffing. Stuffed bits are generally inserted after the CRC calculation in order to avoid confusion between data and the end flag byte contained in each frame, or to create additional transitions. They are used for instance in the universal serial bus (USB), high level data link control (HDLC) or X.25 systems that are useful in many applications. These applications include integrated services digital network (ISDN) and AIS. However, the proposed approach is not restricted to these applications and can be applied to any system subjected to CRC and bit stuffing.

The main contribution of this paper is to propose an error correction strategy using CRC for a TC system in the presence of bit stuffing. The proposed receiver is based on the Viterbi algorithm and is designed to simultaneously demodulate, decode and correct the received messages. Note that the receiver described in this paper resulted in the submission of two patents [7, 8].

The paper is organized as follows. Section 2 presents the CRC concept and the iterative method used by the receiver for its computation. Bit stuffing principles are introduced in section 3. The proposed approach and the detailed algorithm are developed in sections 4 and 5. Simulation results performed on a realistic AIS simulator (developed by the CNES of Toulouse, France) are shown in Section 6. Conclusions are finally reported in Section 7.

2. CYCLIC REDUNDANCY CHECK

2.1. Principle

CRC is a hash function that is commonly used to detect errors in communication data. A fixed-length binary sequence is computed from the transmitted data and is sent jointly with the information bits for protection issues. The receiver computes the CRC with the received data and compares it with the CRC computed with the original data. If both CRC's are not identical, one or more errors are present in the transmitted data. An n-bit CRC applied to data of arbitrary length can detect any single error burst no longer than n bits and can detect a fraction equal to $1 - 2^{-n}$ of longer error bursts.

2.2. CRC Computation

The CRC is computed using a polynomial division. More precisely, a polynomial is formed from the data and is divided by a standardized polynomial called generator polynomial, whose degree equals the length of the CRC plus one. The generator polynomial is selected in order to maximize the error detection capabilities, while minimizing overall collision probabilities. After dividing (modulo two) the

The authors would like to thank the DGA and the CNES for funding.

polynomial associated to the data by the generator polynomial, the quotient of the division is discarded while the remainder (whose degree cannot exceed the degree of the generator polynomial minus one) defines the CRC. Note that zeros are generally added before the remainder to obtain a CRC with fixed length. Finally, it is interesting to mention that the calculation of the modulo of two polynomials can be performed iteratively by initializing the CRC to a standard value and by applying the operations depicted in Fig. 1 to each data bit.

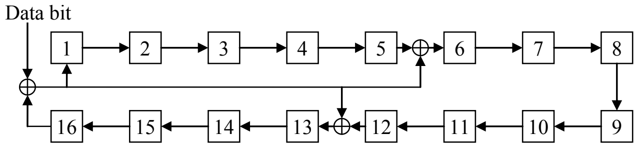

Fig. 1. Example of iterative CRC computation with the generator polynomial $G(x) = x^{16} + x^{12} + x^5 + 1$. $\oplus$ represent XOR and are placed according to the generator polynomial. The numbered boxes contain the derived CRC bits.

2.3. Error detection

To detect errors in data transmission, it is possible to compute the CRC of the received data and to compare it with the transmitted CRC. However, a simpler error detection method is commonly used in hardware implementations. More precisely, when the CRC is transmitted after the data message, the receiver can compute a joint CRC by dividing the data and CRC by a generator polynomial. When there is no error in the received data, the resulting joint CRC equals zero when the received CRC is correct, i.e.,

$$\text{CRC}([\text{Data}, \text{CRC}(\text{Data})]) = 0. \tag{1}$$

3. BIT STUFFING

In data transmission, some non-informative bits can be inserted into the data of interest. These "stuffed bits" can be used for run length limited coding to limit the number of consecutive bits of the same value in order to create additional signal transitions. These additional transitions are inserted to re-synchronize the receiver clock or to avoid specific code words. For instance, the HDLC data transmission protocol uses bit stuffing to avoid the presence of the end frame flag byte (composed of two bits 0 on each side of six consecutive bits 1) in the data. The HDLC bit stuffing is called zero-bit insertion because it inserts an artificial 0 after any sequence of five consecutive bits 1 to ensure that the end frame flag byte never appears in payload data, as shown in Fig. 2. In this paper, it is assumed that the stuffed bit is always a 0 bit, as specified for HDLC and AIS. Note that the proposed method could be easily generalized to the case of any bit stuffing type.

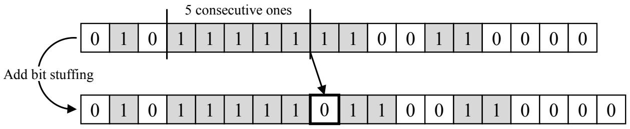

Fig. 2. Principle of bit stuffing in HDLC: a bit 0 bit is inserted after each sequence of five consecutive bits 1.

4. PROPOSED RECEIVER

4.1. General principle

The standard Viterbi algorithm minimizes the square Euclidean distance between the received samples and the estimated symbols defined as

$$d^2 = \sum_{k=1}^{K} \sum_{l=1}^{N_s} |r_k(l) - \boldsymbol{m}_k(l)|^2 \tag{2}$$

where K, N_s, r_k and $\boldsymbol{m}_k$ are the number of received symbols, the number of samples per symbol, the kth received symbol and the kth estimated symbol, respectively.

The proposed algorithm is based on a constrained maximum likelihood estimation minimizing the square Euclidean distance defined in (2) subjected to two constraints: 1) the number of consecutive ones is upper bounded by a maximum value $\overline{P}$ specified by the AIS standard, 2) the CRC satisfies (1). In order to satisfy these constraints, the proposed receiver is based on a trellis composed of extended states formed by a CRC state and a TC state. The trellis is designed so that all paths ending with a final state give a message whose joint CRC is zero, according to (1) (note that the paths corresponding to a non zero CRC do not appear in the trellis). Moreover, the stuffed bits are taken into account by considering special transitions in the extended trellis.

4.2. Trellis construction

The CRC computation can be performed iteratively, as described in section 2.2. It is therefore possible to initialize the CRC with bits 1 or bits 0, depending on the standard CRC used, and update it for each received bit. The intermediate CRC calculations can be seen as states in a trellis. These CRC states are interconnected by transitions linking a CRC state to a new CRC state obtained after updating the previous CRC with one bit and the method shown in Fig. 1.

The trellis proposed in our algorithm consists of extended states composed of a CRC state and a TC state. For example, if the TC state α is followed by a TC state β (resp. TC state γ) when the bit 0 (resp. the bit 1) is transmitted and a CRC state A is followed by the CRC state B (resp. C) when the bit 0 (resp. the bit 1) is transmitted, then the extended state $(A;\alpha)$ is followed by the extended state $(B;\beta)$ in the transmission of the bit 0 (resp. the extended state $(C;\gamma)$ in the transmission of the bit 1). In (3), the integer k refers to the number of the received symbol.

$$\begin{array}{ccccc} \texttt{CRC state} & & \texttt{TC state} & & \texttt{Extended state} \\ \overset{k}{A} \xrightarrow{0} \overset{k+1}{B} & \& & \overset{k}{\alpha} \xrightarrow{0} \overset{k+1}{\beta} & \Rightarrow & \overset{k}{(A;\alpha)} \xrightarrow{0} \overset{k+1}{(B;\beta)} \\ A \xrightarrow{1} C & & \alpha \xrightarrow{1} \gamma & & (A;\alpha) \xrightarrow{1} (C;\gamma) \end{array} \tag{3}$$

As in the Viterbi algorithm, for the received symbol k, the distance $\Gamma[k,(A;\alpha)]$ is defined as the distance between the received signal and the sequence of k symbols coming to the extended state $(A;\alpha)$ at time k. Moreover, $R[k,(A;\alpha)]$ denotes the last bit received leading to the extended state $(A;\alpha)$. Finally, the transition variable $\Gamma_{\textit{trans}}[k,(A;\alpha),t]$ is defined as the sum of $\Gamma[k,(A;\alpha)]$ and the distance between the received symbol at time $k+1$ and the symbol coming from the extended state $(A;\alpha)$ containing the bit t.

4.3. Modified Viterbi algorithm for bit stuffing

To consider the presence of bit stuffing, we propose to introduce special transitions in the extended trellis (i.e., the trellis formed by the extended states). These transitions are used when a stuffed bit is received, and are defined in order to take into account the stuffed bit for the TC state evolution, while the CRC state remains the same. Note that the CRC state should not change when a stuffed bit is received since the CRC is computed before the bit stuffing. The different transitions are illustrated in (4) where the "bit" *SB* represents a stuffed bit 0, since one considers in this paper only zero-bit insertion.

$$\begin{array}{ccccc} \texttt{CRC state} & & \texttt{TC state} & & \texttt{Extended state} \\ \overset{k}{A} \xrightarrow{0} \overset{k+1}{B} & & \overset{k}{\alpha} \xrightarrow{0} \overset{k+1}{\beta} & & \overset{k}{(A;\alpha)} \xrightarrow{0} \overset{k+1}{(B;\beta)} \\ A \xrightarrow{1} C & \& & \alpha \xrightarrow{1} \gamma & \Rightarrow & (A;\alpha) \xrightarrow{1} (C;\gamma) \\ A \xrightarrow{SB} A & & \alpha \xrightarrow{SB} \beta & & (A;\alpha) \xrightarrow{SB} (A;\beta) \end{array} \quad (4)$$

SB transitions are only used when a stuffed bit is received. To know if such a bit is received, a state variable $P[k,(A;\alpha)]$ is attached to each state. This variable indicates the number of consecutive bits 1 received before reaching the extended state $(A;\alpha)$ at time k. If this variable is equal to a given maximum value $\overline{P}$ ($\overline{P}$ is set to 5 for HDLC or AIS, corresponding to the zero-bit insertion illustrated in Fig. 2), a stuffed bit is detected and the only possible transition from $(A;\alpha)$ is the special transition $(A;\alpha) \xrightarrow{SB} (A;\beta)$. After this special transition, $P[k+1,(A;\beta)]$ takes the value 0 since the received bit is in this case the stuffed bit, equal to 0. This procedure is exemplified in (5), where $\not\longrightarrow$ represents an impossible transition (an information bit cannot be a "bit" *SB*, and a stuffed bit can only be a "bit" *SB*). In this case, an infinite distance is assigned to an impossible transition.

$$\begin{array}{ccc} \texttt{Information bit} & & \texttt{Stuffed bit} \\ \underset{P=3}{\overset{k}{(A;\alpha)}} \xrightarrow{0} \underset{P=0}{\overset{k+1}{(B;\beta)}} & & \underset{P=5}{\overset{k}{(A;\alpha)}} \overset{0}{\not\longrightarrow} \overset{k+1}{(B;\beta)} \\ \underset{P=3}{(A;\alpha)} \xrightarrow{1} \underset{P=4}{(C;\gamma)} & & \underset{P=5}{(A;\alpha)} \overset{1}{\not\longrightarrow} (C;\gamma) \\ \underset{P=3}{(A;\alpha)} \overset{SB}{\not\longrightarrow} (A;\beta) & & \underset{P=5}{(A;\alpha)} \xrightarrow{SB} \underset{P=0}{(A;\beta)} \end{array} \quad (5)$$

In addition to the variable P, another state variable $S[k,(A;\alpha)]$ is associated to each state and represents the number of stuffed bits received before reaching the state. It allows one to know the number of informative bits in the received frame. Similarly to the definition of $\Gamma_{trans}[k,(A;\alpha),t]$, the transition variables $P_{\text{trans}}[k,(A;\alpha),t]$ and $S_{\text{trans}}[k,(A;\alpha),t]$ are attached to each transition and represent the evolution of the state variables P and S when the bit t has been received (note that t can take the values 0, 1 or *SB*).

4.4. Final state decision

To choose the path in the trellis, it is important to know its final state. According to (1) the final CRC state must be 0. However, the final TC state $\boldsymbol{\theta}_{TC}^{f}$ is unknown. Moreover, due to the possible presence of stuffed bits, the number of actual received symbols K in the message is also unknown. The estimated values $\widehat{K}$ and $\widehat{\boldsymbol{\theta}}_{TC}^{f}$ are defined as the values of K and $\boldsymbol{\theta}_{TC}^{f}$ which jointly minimize the distance $\Gamma[K,(0;\boldsymbol{\theta}_{TC}^{f})]$ for any possible value of K and any value of the final TC state $\boldsymbol{\theta}_{TC}^{f}$, i.e.,

$$\left(\widehat{K},\widehat{\boldsymbol{\theta}}_{TC}^{f}\right) = \underset{K,\boldsymbol{\theta}_{TC}^{f}}{\arg\min}\, \Gamma[K,(0;\boldsymbol{\theta}_{TC}^{f})] \quad (6)$$

subject to

$$\begin{array}{c} S_{\min} \leq S[K,(0;\boldsymbol{\theta}_{TC}^{f})] \leq S_{\max} \\ N_{\min} \leq K - S[K,(0;\boldsymbol{\theta}_{TC}^{f})] \leq N_{\max}, \end{array} \quad (7)$$

where the bounds $N_{\min}$ and $N_{\max}$ are defined as the minimum and the maximum number of information bits, and $S_{\min}$ and $S_{\max}$ are defined as the minimum and the maximum number of stuffed bits. These bounds are specified by the application standard (e.g., $N_{\min} = N_{\max} = 184$, $S_{\min} = 0$ and $S_{\max} = 4$ in the AIS). $\boldsymbol{\theta}_{TC}^{f}$ can take any value of the TC state, and K can take any value between the lower and upper bounds $K_{\min} = N_{\min}+S_{\min}$ and $K_{\max} = N_{\max}+S_{\max}$, defined as the minimum and the maximum number of received bits, respectively. The constraints (7) ensure that the number of information bits and stuffed bits are in agreement with their minimum and maximum specified by the standard application.

5. OPERATIONAL DETAILS

This section summarizes the different steps of the proposed algorithm.

- **Initialization**: Denote as $(A_0;\alpha_0)$ the initial state of the trellis. A_0 is initialized according to the application standard. For instance, since AIS uses a CRC-16, A_0 is initialized to $2^{16}-1$. When α_0 is unknown, all values of the distances $\Gamma[0,(A_0;\alpha_0)]$ are set to 0. When $\alpha_0 = \alpha_0^*$ is known, one simply sets $\Gamma[0,(A_0;\alpha_0^*)] = 0$.
- **Computation of transition variables**: When a P state variable equals $\overline{P}$, the next transition can only be an *SB* transition. In order to avoid impossible transitions, $\Gamma_{trans}[k,(A;\alpha),t]$ is set to ∞ for $t = 0$ and $t = 1$ where k is the current symbol number and $(A;\alpha)$ is the considered state. Moreover, $\Gamma_{trans}[k,(A;\alpha),SB]$ is set to $\Gamma[k,(A;\alpha)]$ plus the distance between the current received symbol and the symbol carrying an *SB*. Conversely, when the P state variable does not equal $\overline{P}$, $\Gamma_{trans}[k,(A;\alpha),t]$ is adjusted as in the classical Viterbi algorithm for $t = 0,1$ and is set to ∞ for $t = SB$. Moreover, $P_{trans}[k,(A;\alpha),t]$ is set to 0 for $t = 0$ and $t = SB$, and is set to $P[k,(A;\alpha)]+1$ for $t = 1$. Finally, $S_{trans}[k,(A;\alpha),t]$ is set to $S[k,(A;\alpha)]$ for $t = 0,1$, and to $S[k,(A;\alpha)]+1$ for $t = SB$.
- **Computation of state variables**: The selected transition leading to a state is the one with minimal Γ_{trans} among those who can precede this state. The state variables Γ, P and S are set to the values of the variables Γ_{trans}, P_{trans} and S_{trans} of the selected transition whereas the state variable R is set to the bit carried by the selected transition.
- **Path reading**: The resulting sequence is read by following the path through the trellis starting from the final state. The previous states are selected by reading the last received bit in the state variable R of each state.

6. SIMULATIONS

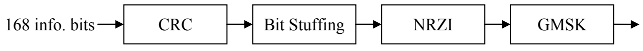

Fig. 3. Example of transmitter model used for simulations.

This section presents some simulation results obtained for the AIS whose transmitter model is illustrated in Fig. 3. The data messages have a fixed length of 168 bits to which the CRC-16 is concatenated. After inserting stuffed bits, the frame is encoded with Non Return to Zero Inverted (NRZI) and modulated in GMSK with bandwidth-bit-time (BT) product parameter of 0.4 and truncation value of 3 (note that these parameters are known by the receiver). These simulations

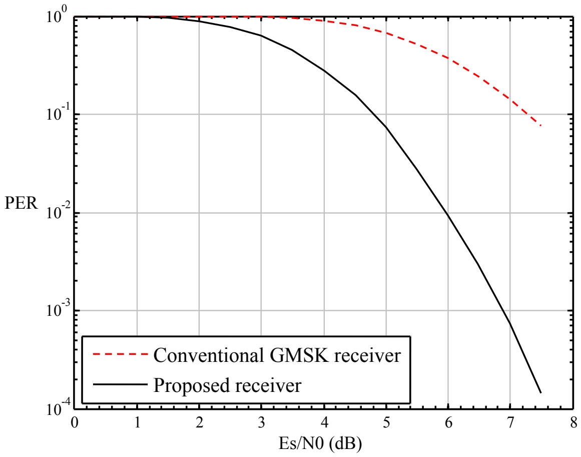

Fig. 4. Proposed receiver compared with the conventional GMSK receiver in Packet Error Rate.

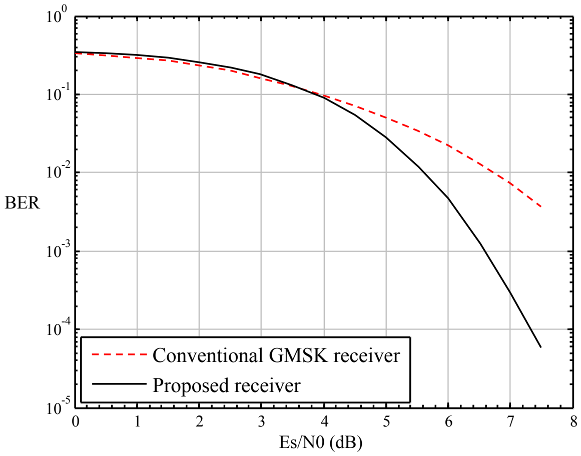

Fig. 5. Proposed receiver compared with the conventional GMSK receiver in Bit Error Rate.

assume a perfect demodulation with ideal carrier and timing recoveries. In this system model, NRZI coding and GMSK modulation constitute the TC. The generator polynomial for CRC computation is $G(x) = x^{16} + x^{12} + x^{5} + 1$ (specified by the AIS recommendation) and an additive white Gaussian noise (AWGN) channel is used as in satellite communications. Note that the methods [2]–[6] mentioned in the introduction cannot be applied in the presence of bit stuffing. Thus, our receiver cannot be compared to those methods. Instead, we consider a receiver based on an conventional coherent GMSK demodulator based on the Viterbi algorithm. NRZI decoding is performed as follows: two consecutive identical bits give a bit 1 and two consecutive different bits give a bit 0. Moreover, the bits following a sequence of five bits 1 are deleted in order to remove the stuffed bits.

Fig. 4 shows a gain of more than 2.5dB when compared to the reference receiver for a target PER of 0.1. This gain is even better when the target PER decreases. The results depicted in Fig. 5 show that the proposed receiver also outperforms the conventional GMSK receiver in terms of bit error rate (BER).

It is known that the BER corresponding to erroneous packets denoted as EBER can be determined using the following relation

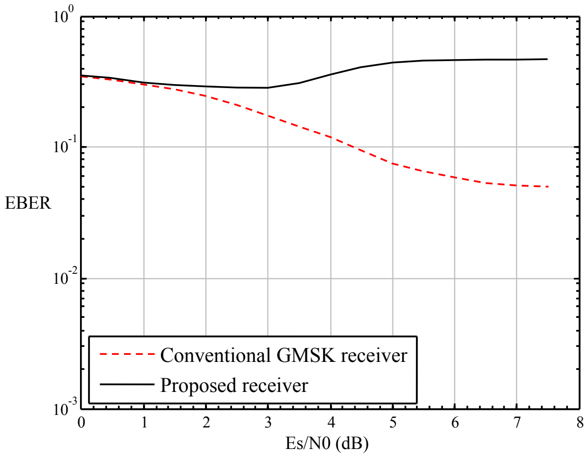

Fig. 6. Proposed receiver compared with the conventional GMSK receiver in Bit Error Rate in erroneous packets.

$$EBER = \frac{BER}{PER}. \qquad (8)$$

As shown in Fig. 6, the EBER is much higher for the proposed receiver than for the conventional GMSK receiver. This allows the use of data consistency methods to determine if a message contains errors.

7. CONCLUSION

This paper studied a new error correction strategy using cyclic redundancy checks (CRC) for a trellis coded system in the presence of bit stuffing. This correction system allows all the redundancies present in each message to be considered. A new type of error correction method was also introduced to compensate bit stuffing that was included between CRC computation and trellis coding. Simulation results illustrated the algorithm performance in terms of packet error rate. A gain of more than 2.5dB was obtained when compared to the conventional GMSK receiver for a packet error rate $PER = 0.1$. The extension of the proposed approach to the multi-user case is currently under investigation.

8. REFERENCES

[1] Recommendation ITU-R M.1371, "Technical characteristics for a universal automatic identification system using time division multiple access in the VHF maritime mobile band." ITU, 2001.

[2] B. McDaniel, "An algorithm for error correcting cyclic redundance checks," *C/C++ Users Journal*, p. 6, 2003.

[3] S. Babaie, A. K. Zadeh, S. H. Es-hagi, and N. J. Navimipour, "Double bits error correction using CRC method," in *Proc. Int. Conf. Semantics, Knowledge and Grid*, no. 5, 2009, pp. 254–257.

[4] C. Shi-yi and L. Yu-bai, "Error correcting cyclic redundancy checks based on confidence declaration," in *Proc. ITS Telecommunications*, no. 6, 2006, pp. 511–514.

[5] Y. Zhang and Q. Yuan, "A multiple bits error correction method based on cyclic redundancy check codes," *ICSP Signal Processing*, no. 9, pp. 1808–1810, 2008.

[6] R. Wang, W. Zhao, and G. B. Giannakis, "CRC-assisted error correction in a convolutionally coded system," *IEEE Trans. Comm.*, vol. 56, no. 11, pp. 1807–1815, 2008.

[7] R. Prévost, D. Bonacci, M. Coulon, J. LeMaitre, J.-P. Millerioux, and J.-Y. Tourneret, "Multi-encodage error correction with extended trellis," Patent Pending.

[8] ——, "A Viterbi algorithm with conditional transitions," Patent Pending.