A New Class of Parallel Data Convolutional Codes

Wei Xiang and Steven S. Pietrobon

***Abstract*— We propose a new class of parallel data convolutional codes (PDCCs) in this paper. The PDCC encoders inputs are composed of an original block of data and its interleaved version. A novel single self-iterative soft-in/soft-out *a posteriori* probability (APP) decoder structure is proposed for the decoding of the PDCCs. Simulation results are presented to compare the performance of PDCCs to that of parallel concatenated convolutional codes (PCCCs).**

***Index Terms*—PDCCs, Self-iterative, APP.**

I. Introduction

The original turbo codes proposed by Berrou *et al.* [1] are binary turbo codes in that those codes accept only single binary inputs. The so-called non-binary turbo codes are based on a parallel concatenation of RSC component codes with m inputs ($m \geqslant 2$) [2]. The advantages of non-binary turbo codes include better convergence in iterative decoding, large minimum distances, less sensitivity to puncturing patterns and suboptimum decoding algorithms and reduced latency [2]. Double-binary turbo codes [3] ($m = 2$) usually possess better error-correcting capabilities than binary turbo codes for equivalent implementation complexity and coding rate. This observation led to the use of circular recursive systematic convolutional (CRSC) codes by Berrou *et. al.* [4]. CRSC codes have the advantage of a graceful degradation to increasing coding rate, and is less susceptible to puncturing and suboptimal decoding algorithms [5]. As a consequence, a CRSC code was chosen for the DVB-RCS standard for return channel via satellite [6] as an alternative to concatenated Reed-Solomon (RS) and non-systematic convolutional codes due to their outstanding performance.

Inspired by a paper submitted recently to *Electronic Letters* [7], we propose a new class of parallel data convolutional codes (PDCCs) in this paper. The PDCC encoder inputs are composed of an original block of data and its interleaved version. A novel single self-iterative soft-in/soft-out *a posteriori* probability (APP) decoder structure is proposed for the decoding of the PDCCs.

The remainder of this paper is organised as follows. Section II-A briefly reviews the circular recursive systematic convolutional code adopted in the DVB-RCS standard. A new class of parallel convolutional codes is proposed in Section II-B. Section III discusses the MAP decoding and self-iterative decoding of PDCCs. Section IV is dedicated to simulation results. The concluding remarks are presented in Section V.

II. Parallel Data Convolutional Codes

A. Circular Recursive Systematic Convolutional Codes

Fig. 1 depicts the CRSC code adopted in the DVB-RCS standard [6]. The data sequence to be encoded consists of a block of N bits grouped into M couples ($N = 2M$). The incoming data is first demultiplexed and fed into A and B of a CRSC encoder (the first bit to A, second bit to B and so on). A and B are two systematic bits, whereas Y and W are two parity bits. Since a CRSC encoder has two inputs and four outputs, it can provide an ample set of seven coding rates, *i.e.*, $R = 1/3, 2/5, 1/2, 2/3, 3/4, 4/5$ and $6/7$. These rates are achievable through puncturing the parity bits.

We show the derivation of the parity check equations and its canonical form for the CRSC code adopted by the DVB-RCS standard as shown in Fig. 1.

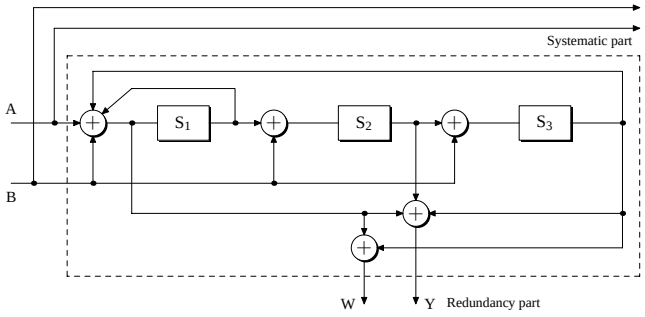

Fig. 1. CRSC component code structure.

Through some mathematic derivation, it is not difficult to show that the parity check equations of the parity output bits Y and W are expressed by (1) and (2) as follows:

$$(1 + D^2 + D^3)A + (1 + D + D^2 + D^3)B + (1 + D + D^3)Y = 0 \quad (1)$$

$$(1 + D^3)A + (1 + D^2)B + (1 + D + D^3)W = 0. \quad (2)$$

Furthermore, the canonical form [8] of the CRSC depicted in Fig. 1 can be derived, which is illustrated in Fig. 2. Refer to [9] for detailed derivation of the canonical form shown in Fig. 2.

B. Parallel Data Convolutional Codes

we propose a new class of parallel data convolutional codes. Fig. 3 depicts a PDCC encoder in its canonical form which adopts the CRSC code described in Section II-A as the constituent convolution code. It is assumed that S'_1 is the MSB (most significant bit) and $S' = 4S'_1 + 2S'_2 + S'_3$.

As depicted in Fig. 3, the block of data sequence to be encoded A and its interleaved version A' constitute two inputs into the encoder. The fact that a PDCC encoder has two parallel

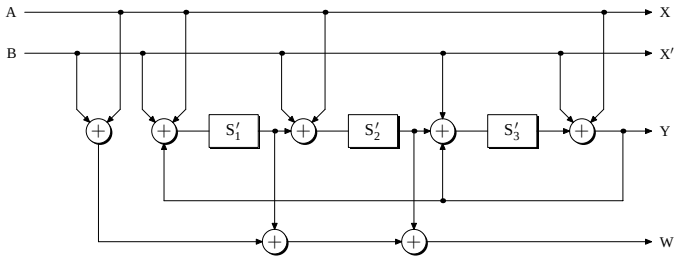

Fig. 2. Canonical systematic convolutional code of coding rate 1/2.

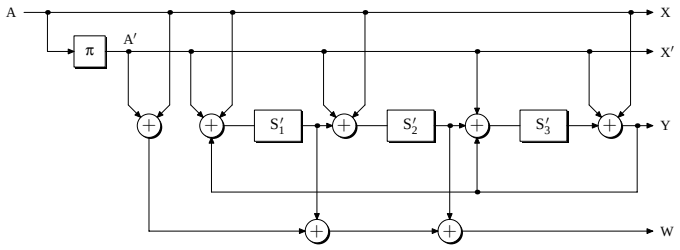

Fig. 3. PDCC encoder.

data inputs is the reason that we name it parallel data convolutional codes. X and X' are two systematic outputs, whereas Y and W are two parity bits. The parity check relationships of Y and W resembling (1) and (2) are given by

$$(1+D^2+D^3)A+(1+D+D^2+D^3)A' + (1+D+D^3)Y=0 \quad (3)$$

$$(1+D^3)A+(1+D^2)A'+(1+D+D^3)W=0. \quad (4)$$

The data stream A and its interleaved version A' are fed into the decoder at the same time. However, A' is decorrelated relative to A due to the presence of the interleaver. For a reasonably good interleaver, like the S-interleaver used in our simulations, this should not adversely affect the performance of the code. The systematic bit X' is not transmitted as X' is the interleaved version of X. Thus, the PDCC encoder shown in Fig. 3 can typically provide a code rate of $1/2$ by transmitting the systematic bit X and the parity bit Y, and a code rate of $1/3$ by transmitting the systematic bit X and the parity bits Y and W. It can also provide other coding rates through puncturing the parity bits Y and W if needed.

It is noted that the idea of self-concatenation in [10] is different from that of PDCCs. For the idea of self-concatenation, data X and its interleaved version X' are joined together and encoded as a single data stream. In other words, the end state of X is the starting state of X'. However, this is quite different from the idea of PDCCs where X and X' are fed into the decoder at the same time.

III. Self-iterative Decoding of PDCCs

A. MAP Decoding of PDCCs

The key difference between the MAP algorithm for PDCCs and the MAP algorithm presented in [11] is that the PDCC encoder has two input bits and four output bits, including two systematic bits A, A' and two parity bits Y, W. The MAP algorithm described in [11], however, is applicable to the soft decoding of rate 1/2 systematic convolution codes which have one input bit and two output bits, including one systematic bit and one parity bit. Fig. 4 illustrates the trellis diagram of the PDCC presented in Fig. 3. It is an 8-state trellis. The numbers shown on the left of each state represent a format of systematic bits/parity bits, *i.e.*, AA'/YW. The numbers from top to bottom correspond to the state transitions diverging from each state from top to bottom.

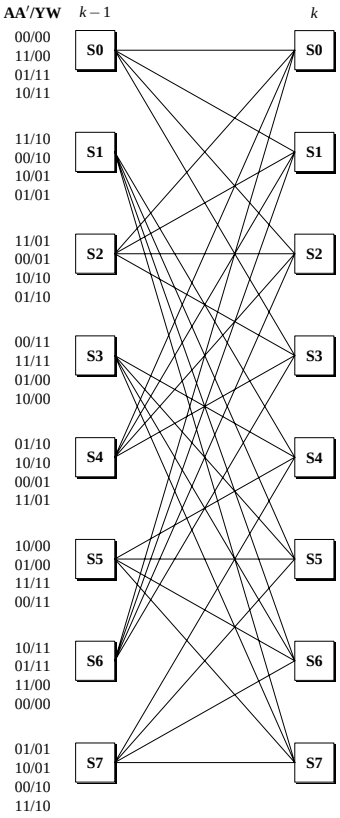

Fig. 4. PDCC's 8-state trellis diagram.

Assume that the outputs of the PDCC encoder depicted in Fig. 3 at time index k are the systematic bit A_k, and the parity bits Y_k and W_k. These outputs are BPSK modulated and transmitted through an AWGN channel. At the receiver end, the received symbols at time index k are defined as

$$R_{A_k}=(1-2A_k)+n_k^1 \quad (5)$$

$$R_{Y_k}=(1-2Y_k)+n_k^2 \quad (6)$$

$$R_{W_k}=(1-2W_k)+n_k^3, \quad (7)$$

with n_k^1, n_k^2 and n_k^3 being three independent normally distributed Gaussian random variables with variance σ^2. A'_k, the interleaved version of the received symbol A_k, is obtained by interleaving A_k at the receiver end.

As shown in Fig. 4, each branch in the PDCC trellis diagram is associated with two input bits and four output bits. Therefore, the branch metric $\gamma_k^{i,m}$, which denotes the branch exiting from

$S_k = m$ with $A_k = i$, can be expressed as

$$\begin{aligned}\gamma_k^{i,m} &= \frac{\xi_k^i\,\xi_k'^j}{2^3\sqrt{2\pi}\sigma}\exp\left(-\frac{1}{2\sigma^2}\big(R_{A_k}-(1-2A_k)\big)^2\right)dR_{A_k}\\ &\quad\cdot\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{1}{2\sigma^2}\big(R_{A'_k}-(1-2A'_k)\big)^2\right)dR_{A'_k}\\ &\quad\cdot\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{1}{2\sigma^2}\big(R_{Y_k}-(1-2Y_k)\big)^2\right)dR_{Y_k}\\ &\quad\cdot\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{1}{2\sigma^2}\big(R_{W_k}-(1-2W_k)\big)^2\right)dR_{W_k}\\ &= \chi_k\,\xi_k^i\,\xi_k'^j\exp\Big(-L_c\big(R_{A_k}A_k+R_{A'_k}A'_k+R_{Y_k}Y_k\\ &\quad+R_{W_k}W_k\big)\Big),\end{aligned}\tag{8}$$

where χ_k is a constant, $\xi_k^i = Pr(A_k = i)$, $\xi_k'^j = Pr(A'_k = j)$, $L_c = 2/\sigma^2$, and dR_{A_k}, $dR_{A'_k}$, dR_{Y_k} and dR_{W_k} are differentials of R_{A_k}, $R_{A'_k}$, R_{Y_k} and R_{W_k}.

The forward state metric α_k^m at time k and state m can be shown as

$$\alpha_k^m = \sum_{i=0}^{3}\gamma_{k-1}^{i,b(i,m)}\alpha_{k-1}^{b(i,m)},\tag{9}$$

where $b(i,m)$ denotes the backward state whose next state is m given input i at the previous time. Likewise, the backward state metric β_k^m at time k and state m can be expressed as

$$\beta_k^m = \sum_{i=0}^{3}\gamma_k^{i,m}\beta_{k+1}^{f(i,m)},\tag{10}$$

where $f(i,m)$ denotes the forward state given current input i and state m.

The likelihood ratio λ_k associated with each decoded bit A_k is defined as

$$\begin{aligned}\lambda_k &= \frac{\Pr(A_k=1|R_A,R_Y,R_W)}{\Pr(A_k=0|R_A,R_Y,R_W)}\\ &= \frac{\sum_m\sum_{i=2}^{3}\alpha_k^m\,\gamma_k^{i,m}\beta_{k+1}^{f(i,m)}}{\sum_m\sum_{i=0}^{1}\alpha_k^m\,\gamma_k^{i,m}\beta_{k+1}^{f(i,m)}},\end{aligned}\tag{11}$$

where $i = 0,1,2,3$ corresponds to A, A' inputs of 00, 01, 10 and 11. Similarly, the log-likelihood ratio λ'_k of the interleaved bit A'_k can be written as

$$\begin{aligned}\lambda'_k &= \frac{\Pr(A'_k=1|R_A,R_Y,R_W)}{\Pr(A'_k=0|R_A,R_Y,R_W)}\\ &= \frac{\sum_m\sum_{i=1,3}\alpha_k^m\,\gamma_k^{i,m}\beta_{k+1}^{f(i,m)}}{\sum_m\sum_{i=0,2}\alpha_k^m\,\gamma_k^{i,m}\beta_{k+1}^{f(i,m)}}.\end{aligned}\tag{12}$$

Decisions on decoded bits $\widehat{A}_k$ are then made by the PDCC MAP decoder by comparing λ_k to a threshold equal to one.

B. Self-iterative Decoding

The novelty of decoding the PDCCs lies in self-iterative decoding. The self-iterative PDCC decoder operates like a normal MAP decoder except it feeds the extrinsic outputs after interleaving or deinterleaving back as *a priori* inputs. Fig. 5 shows a schematic of a self-iterative PDCC decoder.

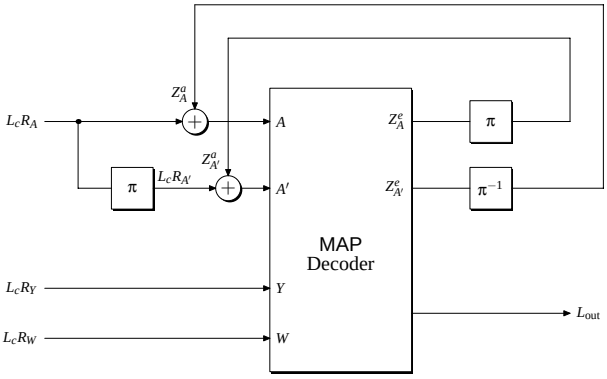

Fig. 5. Self-iterative PDCC decoder.

The inputs to the decoder are the soft outputs of a noisy channel L_cR_A, L_cR_Y and L_cR_W, respectively. The decoder reconstructs $L_cR'_A$ by interleaving L_cR_A. The idea of the self-iterative decoding comes from the fact that R'_A is the interleaved version of R_A, so that the extrinsic information of R_A can be fed back as the *a priori* information for R'_A after interleaving and the extrinsic information of R'_A can be fed back as the *a priori* information for R_A after deinterleaving.

We denote the *a priori* information of R_A and R'_A by Z_A^a and $Z_{A'}^a$, while the extrinsic information of R_A and R'_A are denoted by Z_A^e and $Z_{A'}^e$, respectively. The self-iterative MAP decoder computes the APP of the information bit A. The LLR output of the decoder can be expressed as

$$L_{\text{out}} = L_cR_A + L_cR_{A'} + Z_A^a + Z_A^e + Z_{A'}^a + Z_{A'}^e.\tag{13}$$

The self-iterative PDCC decoder proceeds as follows. At the first decoding iteration, Z_A^a and $Z_{A'}^a$ are initialised to zero. For the subsequent iterations, Z_A^e is interleaved and fed back as the *a priori* information for A', *i.e.*, $Z_{A'}^a = \pi(Z_A^e)$ where $\pi(\cdot)$ denotes an interleaving mapping. Likewise, $Z_{A'}^e$ is deinterleaved and fed back as the *a priori* information for A, *i.e.*, $Z_A^a = \pi^{-1}(Z_{A'}^e)$ where $\pi^{-1}(\cdot)$ denotes a deinterleaving mapping. At the final iteration, the decoder delivers the log-likelihood output L_{out}. The self-iterative decoding process can be clearly seen from the two feed back connections between Z_A^e and $Z_{A'}^a$, and $Z_{A'}^e$ and Z_A^a in Fig. 5.

IV. Simulation Results

In this section, we compare the performance of the proposed PDCC to a parallel concatenated CRSC coding system used in the DVB-RCS standard and present simulation results. The system functional block diagram is graphically shown in Fig. 6.

The horizontal and vertical constituent codes used in Fig. 6 are the same as the CRSC presented in Fig. 1. The incoming information data sequence consists of a block of N bits

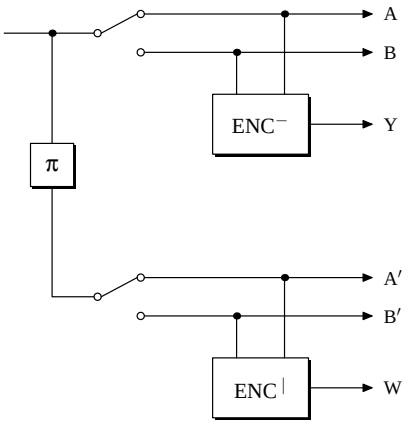

Fig. 6. Parallel concatenated CRSC encoder.

grouped into M couples ($N = 2M$). The horizontal constituent encoder is fed with the information bits in the natural order of the data, whereas the vertical constituent encoder is fed with the same information bits in an interleaved order of the data. A' and B' bits are not transmitted since they can be reconstructed by interleaving the A and B bits at the receiver side. The parallel concatenated CRSC encoder can provide seven code rates as defined in the DVB-RCS standard, *i.e.*, $R = 1/3,\ 2/5,\ 1/2,\ 2/3,\ 3/4,\ 4/5,\ 6/7$ [6]. The decoder structure is similar to that of turbo codes [1] except that the MAP decoders for a rate 1/2 systematic convolutional code are replaced by the MAP decoders for the CRSC described in Section II-A.

Simulations were conducted to compare the performance for the proposed PDCC system shown in Fig. 3 to the parallel concatenated CRSC system shown in Fig. 6. The simulation configurations are as follows. An S-type interleaver [12] is adopted with S equal to 47. Randomly generated data of length $8K$ (8192) bits is used for both systems. The channel coding rate for both systems is 1/2. Simulation results are presented in Fig. 7, where the PCCC curve refers to the performance of the parallel concatenated CRSC system.

As depicted in the figure, the performance of the PDCC is very close to that of the parallel concatenated CRSC at low E_b/N_0 up to 0.6 dB. However, PDCC performs at least 0.2 dB worse than the parallel concatenated CRSC at low BERs. We conjecture the relatively poor performance of PDCCs at low BER may be due to the self-terminating phenomena with one input bit. Fig. 8 illustrates the trellis terminating properties of the PCCCs (turbo codes) and PDCCs. For the PCCCs, an error bit could cause the trellis path to divert from the two all-zero paths as shown in Fig.8–(a). The same bit is interleaved and fed into the second encoder. That bit would not cause the diverted trellis path to re-emerge earlier. On the other hand, for the PDCCs, an error causes a diversion from the all-zero trellis path. The same bit is interleaved and then fed into the same PDCC encoder. That bit could cause an earlier trellis remerge as shown in Fig.8–(b). The decoding complexity of PDCCs is similar to that of the PCCCs. This is because the PDCC decoder consists of one constituent MAP decoder which decodes one bit in each iteration, whereas the PCCC decoder consists of two constituent MAP decoders which decode two bits in each iteration.

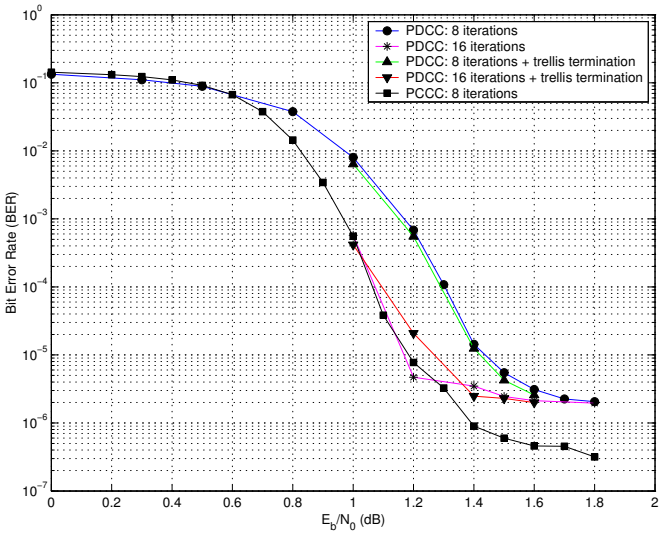

Fig. 7. Simulation results.

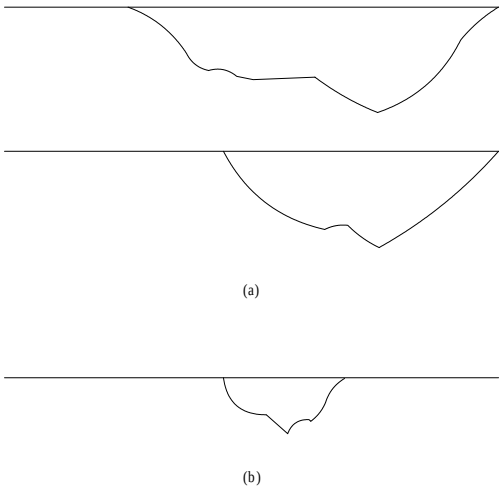

Fig. 8. Self-terminating property of (a) PCCC; (b) PDCC.

A way of overcoming the self-terminating property may be by designing the interleaver so as to avoid self-terminations. For example, if $A_k = 1$, with zeros elsewhere except at $A'_{k+3} = 1$, then we would have a self-terminating sequence going through states $S_0 \rightarrow S_3 \rightarrow S_4 \rightarrow S_0$. Thus, we could design the trellis such that if the interleaver maps position i to j, then $(i - j)$ mod 3 is not equal to zero. This will remove many self-terminating sequences and hopefully lower the error floor. Another possibility is designing the convolutional code so that these period 3 terminations do not occur. It may be possible to increase the self-termination length to $2^\nu - 1 = 7$, which is the maximum that can be expected with a primitive divisor polynomial. These longer lengths should be easier to design out of the interleaver.

Nevertheless, the idea of self-iterative decoding could be useful in some cases, *e.g.*, space-time coding. We could combine

the two trellises of a space-time code into a super trellis. Subsequently, self-iterative decoding could be applied to such a trellis with non modulo-2 operation.

V. Conclusions

In this paper, a new class of parallel data convolutional codes is presented. The PDCC encoder takes two parallel data inputs, with one being the original data and the other being the interleaved data. The PDCC decoder has an innovative self-iterative decoding structure. Unlike a turbo decoder where the extrinsic output of one MAP decoder is passed on to the other as the *a priori* input, the PDCC decoder operates like a normal MAP decoder but feeds the extrinsic outputs back as its own interleaved *a priori* inputs.

The performance of PDCCs was compared to that of a parallel concatenated CRSC, equivalent to the one used in the DVB-RCS standard. The two schemes performed close to each other at low SNRs, however the parallel concatenated CRSC outperformed PDCCs by at least 0.2 dB at low BERs. Also, the PDCC has a higher error floor than the PCCC. We conjecture this is due to the self-terminating property of PDCCs with single bit errors. Although the performance of PDCCs is not encouraging, the idea of self-iterative decoding is worth exploring and can be applied to some codes like space-time codes.

References

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: turbo-codes," in *Proc. IEEE Int. Conf. Commun. (ICC'93)*, Geneva, Switzerland, May 1993, pp. 1064–1070.

[2] C. Berror, M. Jézéquel, C. Douillard, and S. Kerouédan, "The advantages of non-binary turbo codes," in *Proc. IEEE Information Theory Workshop (ITW'01)*, Cairns, Australia, Sept. 2001, pp. 61–63.

[3] C. Berrou and M. Jezequel, "Non-binary convolutional codes for turbo coding," *Electron. Lett.*, vol. 35, no. 1, pp. 39–40, Jan. 1999.

[4] C. Berrou, C. Douillard, and M. Jezequel, "Multiple parallel concatenation of circular recursive convolutional (CRSC) codes," *Annals of Telecommunications*, vol. 54, no. 3-4, pp. 166–172, Mar.-Apr. 1999.

[5] M. R. Soleymani, Y. Gao, and U. Vilaipornsawai, *Turbo Coding for Satellite and Wireless Communications*. Norwell, MA: Kluwer Academic Publishers, 2002.

[6] ETSI EN 301 790, *Digital Video Broadcasting (DVB); Interaction channel for satellite distribution systems*. ETSI reference EN 301 790, v.1.3.1, Mar. 2003.

[7] Y.-H. Li and D. Li, "Two-dimensional log-MAP decoding algorithm and its applications," *Submitted to Electron. Lett.*, Mar. 2003.

[8] G. D. Forney, "Convolutional codes I: Algebraic structure," *IEEE Trans. Inform. Theory*, vol. IT-16, pp. 720–738, Nov. 1970.

[9] W. Xiang, "Joint source-channel coding for image transmission and related topics," Ph.D. dissertation, University of South Australia, Adelaide, Australia, 2003.

[10] C. Berrou, "Some clinical aspects of turbo codes," in *Proc. Int. Symp. on Turbo Codes & Related Topics*, Brest, France, Sept. 1997, pp. 26–31.

[11] S. S. Pietrobon, "Implementation and performance of a Turbo/MAP decoder," *Int. J. Satellite Commun.*, vol. 16, pp. 23–46, Jan.-Feb. 1998.

[12] D. Divsalar and F. Pollara, "Multiple turbo codes for deep-space communications," in *JPL TDA Progress Report*, vol. 42-121, May 1995, pp. 66–77.