The Consultative Committee for Space Data Systems

Recommendation for Space Data System Standards

TM SYNCHRONIZATION AND CHANNEL CODING

ECSS MODIFIED VERSION

RECOMMENDED STANDARD

CCSDS 131.0-B-3

BLUE BOOK
September 2017

3 CONVOLUTIONAL CODING

3.1 OVERVIEW

The basic convolutional code is a rate (*r*) 1/2, constraint-length (*K*) 7 transparent code which is well suited for channels with predominantly Gaussian noise. This code is defined in 3.3. When this code is punctured according to 3.4, higher code rates may be achieved although with lower error correcting performance.

Puncturing allows a single code rate of either 2/3, 3/4, 5/6 or 7/8 to be selected. The four different puncturing schemes allow selection of the most appropriate level of error correction and symbol rate for a given service or data rate.

3.2 GENERAL

3.2.1 ATTACHED SYNC MARKER

The Attached Sync Marker used with convolutional code shall be the 32-bit pattern specified in 9.2, and it shall always be inserted before performing convolutional encoding.

3.2.2 DATA RANDOMIZATION

The pseudo-randomizer defined in section 10 shall be used unless the system designer verifies that the concerns identified in the note below are resolved by other means.

NOTE – An inverter is specified with the basic convolutional code to assure sufficient bit transitions to keep receiver symbol synchronizers in lock, when used with Binary Phase Shift Keying (BPSK) modulation. Sufficient bit transitions cannot be guaranteed by the inverter alone if some multiplexing schemes are used, e.g., with Quadrature Phase Shift Keying (QPSK) modulation, or if a punctured convolutional code is used. There are also data patterns for which convolutional code synchronization cannot be determined. The pseudo-randomizer is also used to aid signal acquisition and to mitigate spectral lines in the transmitted signal.

3.2.3 FRAME VALIDATION

When TM or AOS Transfer Frames are used, the Frame Error Control Field (FECF) specified in references [1] and [2] shall be used to validate the Transfer Frame, unless the convolutional code is concatenated with an outer Reed-Solomon code (see section 4).

NOTE – If the decoder's correction capability is exceeded, undetected bursts of errors may appear in the output.

3.2.4 QUANTIZATION

Soft bit decisions with at least three-bit quantization shall be used for the decoder~~should be used whenever constraints (such as complexity of decoder) permit~~.

3.3 BASIC CONVOLUTIONAL CODE SPECIFICATION

3.3.1 The basic convolutional code shall be the non-systematic code with the following characteristics:

(1) Nomenclature: Convolutional code with maximum-likelihood decoding.

(2) Code rate (*r*): 1/2 bit per symbol.

(3) Constraint length (*K*): 7 bits.

(4) Connection vectors: G1 = 1111001 (171 octal); G2= 1011011(133 octal).

(5) Symbol inversion: On output path of G2.

NOTE – An encoder block diagram is shown in figure 3-1. When a single encoder is used, G2 inversion provides no benefit to data randomization when even-order modulations higher than BPSK are used. G2 inversion does provide value when coding is done after channel splitting and with separate encoders on each channel.

3.3.2 The output symbol sequence shall be: $C_1(1), \overline{C_2(1)}, C_1(2), \overline{C_2(2)}. . .$

3.3.3 When suppressed-carrier modulation systems are used:

a) Non-Return-to-Zero-Mark (NRZ-M) or Non-Return-to-Zero-Level (NRZ-L) may be used as a modulating waveform.

b) If the user contemplates differential encoding, i.e., conversion of his modulating waveform from NRZ-L to NRZ-M, such conversion should be performed at the input to the convolutional encoder.

NOTES

1 Since the convolutional codes are transparent, differential encoding can be used before the convolutional encoder to help phase ambiguity resolution and, correspondingly, the conversion at the receiving end from NRZ-M to NRZ-L should be performed at the output of the convolutional decoder. Differential encoding after the convolutional encoder is not advised because it introduces considerable loss of performance. It also would require differential detection, which is more complex with soft symbols.

4 REED-SOLOMON CODING

4.1 OVERVIEW

The Reed-Solomon (R-S) codes defined in this section are powerful burst error correcting codes. One of two different error-correcting options may be chosen. For maximum performance (at the expense of accompanying overhead) the E=16 option can correct 16 R-S symbols in error per codeword. For lower overhead (with reduced performance) the E=8 option can correct 8 R-S symbols per codeword. The Reed-Solomon code may be used alone, and as such it provides an excellent forward error correction capability in a burst-noise channel. However, should the Reed-Solomon code alone not provide sufficient coding gain, it may be concatenated with the convolutional code defined in section 3. Used this way, the Reed-Solomon code is the *outer code*, while the convolutional code is the *inner code*.

4.2 GENERAL

4.2.1 DATA RANDOMIZATION

The pseudo-randomizer defined in section 10 shall be used unless the system designer verifies that the concerns identified in the note below are resolved by other means.

NOTE – The recommended Reed-Solomon codes, by themselves, cannot guarantee sufficient bit transitions to keep receiver symbol synchronizers in lock. Because of the quasi-cyclic nature of these codes, undetected decoding errors may result from incorrect codeblock synchronization. The pseudo-randomizer is also used to aid signal acquisition and to mitigate spectral lines in the transmitted signal.

4.2.2 FRAME VALIDATION

The FECF specified in references [1] and [2] is optional. The system designer may choose to use it for additional codeblock validation, particularly with the E=8 code.

NOTE – The Reed-Solomon code with E=16 has an extremely low undetected error rate, and that with E=8 has an undetected error rate low enough for some applications. Therefore the R-S decoder may be used alone to validate the codeblock, and consequently the contained TM Transfer Frame (reference [1]) or AOS Transfer Frame (reference [2]).

4.3 SPECIFICATION

4.3.1 PARAMETERS

The parameters of the selected Reed-Solomon (R-S) code are as follows:

a) J shall be 8 bits per R-S symbol.

b) E shall be 16 or 8 R-S symbols.

~~b)~~c) E=8 shall not be used unless the modulation scheme is 4-dimensional 8PSK trellis-coded modulation (4D-8PSK-TCM).

Comment [GPC1]: ECSS added requirement.

NOTE – E is the Reed-Solomon error correction capability, in symbols, within a R-S codeword.

4.3.2 GENERAL CHARACTERISTICS

The code shall conform to the following general characteristics:

a) J, E, and I (the depth of interleaving) are independent parameters.

b) $n = 2^J - 1 = 255$ symbols per R-S codeword.

c) $2E$ is the number of R-S symbols among n symbols of an R-S codeword representing parity checks.

d) $k = n - 2E$ is the number of R-S symbols among n R-S symbols of an R-S codeword representing information.

4.3.3 FIELD GENERATOR POLYNOMIAL

The field generator polynomial shall be:

$$F(x) = x^8 + x^7 + x^2 + x + 1$$

over GF(2).

4.3.4 CODE GENERATOR POLYNOMIAL

The code generator polynomial shall be:

$$g(x) = \prod_{j=128-E}^{127+E} (x - \alpha^{11j}) \qquad = \sum_{i=0}^{2E} G_i x^i$$

over GF(2^8), where F(α) = 0.

NOTES

1 It should be recognized that α^{11} is a primitive element in GF(2^8) and that F(x) and g(x) characterize a (255,223) Reed-Solomon code when E = 16 and a (255,239) Reed-Solomon code when E = 8.

2 The selected code is a systematic code. This results in a systematic codeblock.

4.3.5 SYMBOL INTERLEAVING

4.3.5.1 The allowable values of interleaving depth ~~are~~ shall be:

a) When E=16, I=1, 2, 3, 4, 5, and 8

~~**4.3.5.1**~~ b) When E=8, I=8.

Formatted: List

NOTE – I=1 is equivalent to the absence of interleaving.

4.3.5.2 The interleaving depth shall ~~normally~~ be fixed on a Physical Channel for a Mission Phase.

NOTE – Discussion of symbol interleaving is contained in 4.4.1.

4.3.6 MAXIMUM CODEBLOCK LENGTH

The maximum codeblock length, in R-S symbols, shall be determined by the following equation:

$$L_{\mathrm{max}} = nI = (2^J - 1)I = 255I$$

4.3.7 SHORTENED CODEBLOCK LENGTH

4.3.7.1 A shortened codeblock length may be used to accommodate frame lengths smaller than the maximum.

NOTE – However, since the Reed-Solomon code is a block code, the decoder must always operate on a full block basis.

4.3.7.2 To achieve a full codeblock, 'virtual fill' shall be added to make up the difference between the shortened block and the maximum codeblock length.

NOTES

1 The characteristics and limitations of virtual fill are covered in 4.3.8.2.

2 Since the virtual fill is not transmitted, both encoder and decoder need to be set to insert it with the proper length for the encoding and decoding processes to be carried out properly.

4.3.7.3 When an encoder (initially cleared at the start of a block) receives $kI–Q$ symbols representing information (where Q, representing fill, is a multiple of I, and is less than kI), $2EI$ check symbols shall be computed over kI symbols, of which the leading Q symbols shall be treated as all-zero symbols.

NOTE – A $(nI–Q, kI–Q)$ shortened codeblock results.

5 CONCATENATED CODING

5.1 Concatenated codes shall consist of a combination of a Reed-Solomon code with E=16 defined in section 4 with one of the convolutional codes defined in section 3.

~~**5.1**~~ NOTE - Reed-Solomon code with E=8 is not concatenated with one of the convolutional codes.

Formatted: Note level 1

5.2 The Reed-Solomon code shall be the outer code, and the convolutional code shall be the inner code.

6 TURBO CODING

6.1 OVERVIEW

Turbo codes are binary block codes with large codewords (hundreds or thousands of bits). Turbo codes may be used to obtain even greater coding gains than those provided by concatenated coding systems. They are systematic and inherently non-transparent.

6.2 GENERAL

6.2.1 DATA RANDOMIZATION

The pseudo-randomizer defined in section 10 shall be used unless the system designer verifies that the concerns identified in the note below are resolved by other means.

NOTE – The recommended Turbo codes, by themselves, cannot guarantee sufficient bit transitions to keep receiver symbol synchronizers in lock. The pseudo-randomizer is also used to aid signal acquisition and to mitigate spectral lines in the transmitted signal.

6.2.2 FRAME VALIDATION

When Turbo codes are used with TM or AOS Transfer Frames, the FECF specified in references [1] or [2], respectively, shall be used to validate the Transfer Frame.

NOTE – While providing outstanding coding gain, Turbo codes may still leave some undetected errors in the decoded output.

6.3 SPECIFICATION

NOTE – A Turbo encoder is a combination of two simple encoders. The input is a frame of k information bits. The two component encoders generate parity symbols from two simple recursive convolutional codes, each with a small number of states. The information bits are also sent uncoded. A key feature of Turbo codes is an interleaver, which permutes bit-wise the original k information bits before input to the second encoder.

The recommended Turbo code is a systematic code that shall conform to the following specifications:

a) Code type shall be systematic parallel concatenated Turbo code.

b) Number of component codes shall be two (plus an uncoded component to make the code systematic).

c) Type of component codes shall be recursive convolutional codes.

d) Number of states of each convolutional component code shall be 16.

e) Nominal code rates shall be r = 1/2~~, 1/3,~~ or 1/4~~, or 1/6~~ (selectable).

NOTE – Because of 'trellis termination' symbols (see 6.3j)), the true code rates (defined as the ratios of the information block lengths to the codeblock lengths in table 6-2) are slightly smaller than the nominal code rates. In this Recommended Standard, the term 'code rate' always refers to the nominal code rates, r = 1/2, 1/3, 1/4, or 1/6.

f) The specified information block lengths k shall be those specified in table 6-1. The corresponding codeblock lengths in bits, $n=(k+4)/r$, for the specified code rates shall be those specified in table 6-2.

NOTE – Information block lengths are chosen for compatibility with the corresponding Reed-Solomon interleaving depths, also shown in table 6-1.

12.6 MANAGED PARAMETERS FOR TURBO CODE

The managed parameters for Turbo code shall be those specified in table 12-4.

Table 12-4: Managed Parameters for Turbo Code

Managed Parameter	Allowed Values
Nominal Code Rate (*r*)	1/2, ~~1/3,~~ 1/4~~, 1/6~~
Information Block Length (*k,* bits)	1784, 3568, 7136, 8920

12.7 MANAGED PARAMETERS FOR LOW-DENSITY PARITY-CHECK CODING OF A TRANSFER FRAME

The managed parameters for LDPC coding for a Transfer Frame shall be those specified in table 12-5.

Table 12-5: Managed Parameters for Low-Density Parity-Check Code

Managed Parameter	Allowed Values
Code Rate (*r*)	1/2, 2/3, 4/5, 7/8
Information Block Length (*k*, bits)	1024, 4096, 16384 (if *r*=1/2, 2/3, or 4/5), 7136 (if *r*=7/8)

12.8 MANAGED PARAMETERS FOR LOW-DENSITY PARITY-CHECK CODING OF A STREAM OF SMTFS

The managed parameters for LDPC coding of a stream of SMTFs shall be those specified in table 12-6.

Table 12-6: Managed Parameters for Low-Density Parity-Check Code

Managed Parameter	Allowed Values
Slice Length (i.e., Information Block Length [*k*, bits])	1024, 4096, 16384 (if *r*=1/2, 2/3, or 4/5), 7136 (if *r*=7/8)
LDPC Codeblock Size (Number of Codewords)	m = 1, 2, 3, 4, 5, 6, 7, 8