TSG-RAN Working Group1 meeting #11
San Diego, USA, February 29 – March 3, 2000

TSGR1#11(00)0437

Agenda item:

Source: **NTT DoCoMo, Nortel Networks and Nokia**

Title: **Editorial modifications of channel coding section in 25.212 and 25.222**

Document for: **Decision**

Introduction

This document includes the revised versions of two CRs[1]: CR-060 for TS25.212 and CR029 for TS25.222. The main modification of these revised CRs is one for the description of setp (3) in section 4.2.3.2.3.2, which was modified taking into account the relevant parts of the alternative proposals [2], [3].

Reference

[1] NTT DoCoMo and Nortel Networks, "Editorial modification of channel coding section in 25.212 and 25.222", TSGR1#11(00)0330
[2] Nokia, "CR 25212-030r1: Clarification on Turbo internal interleaver", TSGR1#11(00)0105
[3] Nokia, "CR 25222-016r1: Clarification on Turbo internal interleaver", TSGR1#11(00)0214

3GPP RAN WG1 Meeting #11
San Diego, USA, 29 Feb - 3 Mar 2000

Document R1-00-0437
e.g. for 3GPP use the format TP-99xxx
or for SMG, use the format P-99-xxx

CHANGE REQUEST

Please see embedded help file at the bottom of this page for instructions on how to fill in this form correctly.

25.212	CR	060r1	Current Version:	3.1.1

GSM (AA.BB) or 3G (AA.BBB) specification number ↑ *↑ CR number as allocated by MCC support team*

For submission to:	RAN #7	for approval	X	strategic		(for SMG use only)
list expected approval meeting # here ↑		for information		non-strategic		

Form: CR cover sheet, version 2 for 3GPP and SMG The latest version of this form is available from: ftp://ftp.3gpp.org/Information/CR-Form-v2.doc

Proposed change affects: (U)SIM [ ] ME [X] UTRAN / Radio [X] Core Network [ ]
(at least one should be marked with an X)

Source:	NTT DoCoMo, Nortel Networks and Nokia	**Date:**	3-Mar-2000
Subject:	Editorial changes of channel coding section		
Work item:			

Category:	F	Correction		**Release:**	Phase 2	
	A	Corresponds to a correction in an earlier release			Release 96	
(only one category	B	Addition of feature			Release 97	
shall be marked	C	Functional modification of feature			Release 98	
with an X)	D	Editorial modification	X		Release 99	X
					Release 00	

Reason for change:	To clarify exact functions of channel coding.
Clauses affected:	4.2.3 of TS25.212

Other specs affected:	Other 3G core specifications		→ List of CRs:
	Other GSM core specifications		→ List of CRs:
	MS test specifications		→ List of CRs:
	BSS test specifications		→ List of CRs:
	O&M specifications		→ List of CRs:

Other comments:	This CR is including the content of approved CR 044 of TS25.212.

4.2.3 Channel coding

Code blocks are delivered to the channel coding block. They are denoted by $o_{ir1}, o_{ir2}, o_{ir3}, \ldots, o_{irK_i}$, where i is the TrCH number, r is the code block number, and K_i is the number of bits in each code block. The number of code blocks on TrCH i is denoted by C_i. After encoding the bits are denoted by $y_{ir1}, y_{ir2}, y_{ir3}, \ldots, y_{irY_i}$., where Y_i is the number of encoded bits. The encoded blocks are serially multiplexed so that the block with lowest index r is output first from the channel coding block. The bits output are denoted by $c_{i1}, c_{i2}, c_{i3}, \ldots, c_{iE_i}$, where i is the TrCH number and $E_i = C_iY_i$. The output bits are defined by the following relations:

$$c_{ik} = y_{i1k} \quad k = 1, 2, \ldots, Y_i$$

$$c_{ik} = y_{i,2,(k-Y_i)} \quad k = Y_i + 1, Y_i + 2, \ldots, 2Y_i$$

$$c_{ik} = y_{i,3,(k-2Y_i)} \quad k = 2Y_i + 1, 2Y_i + 2, \ldots, 3Y_i$$

$$\ldots$$

$$c_{ik} = y_{i,C_i,(k-(C_i-1)Y_i)} \quad k = (C_i - 1)Y_i + 1, (C_i - 1)Y_i + 2, \ldots, C_iY_i$$

The relation between o_{irk} and y_{irk} and between K_i and Y_i is dependent on the channel coding scheme.

The following channel coding schemes can be applied to TrCHs:

- Convolutional coding
- Turbo coding
- No channel coding

Usage of coding scheme and coding rate for the different types of TrCH is shown in table 1.

The values of Y_i in connection with each coding scheme:

- Convolutional coding, ½ with rate 1/2: $Y_i = 2*K_i + 16$; rate 1/3 rate: $Y_i = 3*K_i + 24$
- Turbo coding, with rate 1/3 rate: $Y_i = 3*K_i + 12$
- No channel coding,: $Y_i = K_i$

Table 1: Usage of channel coding scheme and coding rate Error Correction Coding Parameters

Type of TrCH	Coding scheme	Coding rate
BCH	Convolutional coding	1/2
PCH		
RACH		
CPCH, DCH, DSCH, FACH		1/3, 1/2
	Turbo coding	1/3
	No coding	

~~Transport channel type~~	~~Coding scheme~~	~~Coding rate~~
~~BCH~~	~~Convolutional code~~	~~1/2~~
~~PCH~~		
~~RACH~~		
~~CPCH, DCH, DSCH, FACH~~		~~1/3, 1/2~~
	~~Turbo Code~~	~~1/3~~
	~~No coding~~	

~~If no code blocks are input to the channel coding ($C_i = 0$), no bits shall be output from the channel coding, i.e. $E_i = 0$.~~

4.2.3.1 Convolutional coding

Convolutional codes with constraint length 9 and coding rates 1/3 and 1/2 are defined.

The configuration of the convolutional coder is presented in figure 3.

Output from the rate 1/3 convolutional coder shall be done in the order output0, output1, output2, output0, output1, output 2, output 0,…,output2. Output from the rate 1/2 convolutional coder shall be done in the order output 0, output 1, output 0, output 1, output 0, …, output 1.

8 tail bits with binary value 0 shall be added to the end of the code block before encoding.

The initial value of the shift register of the coder shall be "all 0" when starting to encode the input bits.

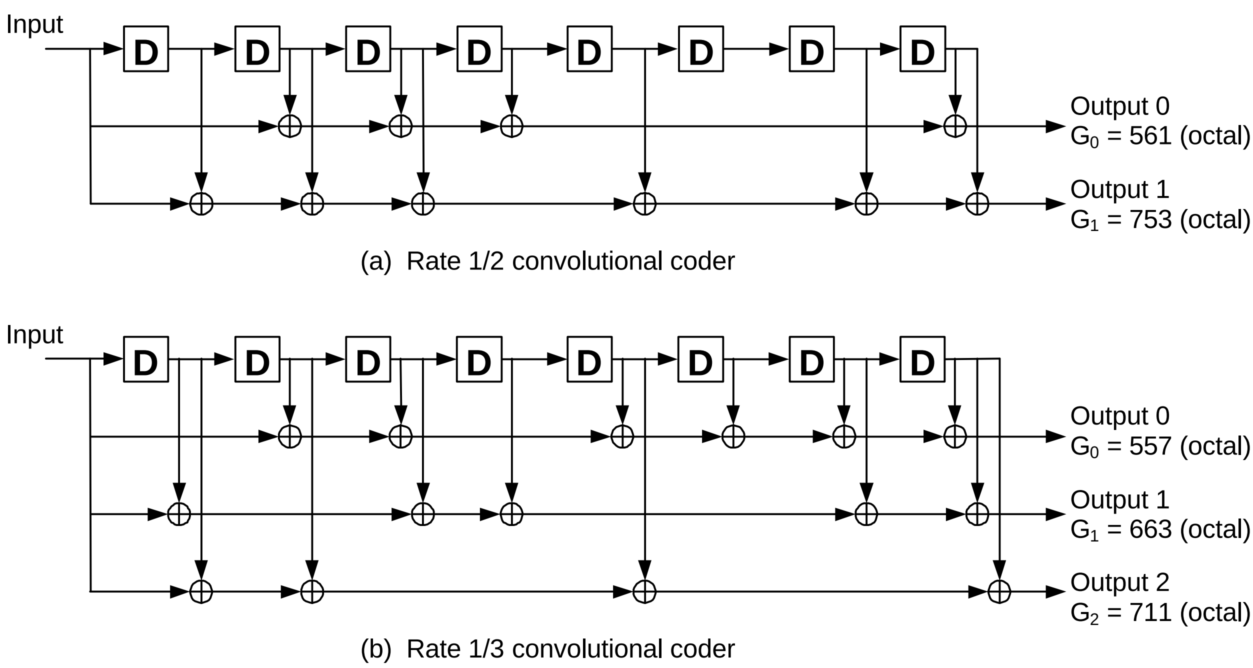


Figure 3: Rate 1/2 and rate 1/3 convolutional coders

4.2.3.2 Turbo coding

4.2.3.2.1 Turbo coder

The ~~turbo coding~~ scheme of Turbo coder is a ~~p~~Parallel ~~c~~Concatenated ~~c~~Convolutional ~~c~~Code (PCCC) with two 8-state constituent encoders and one Turbo code internal interleaver. The coding rate of Turbo coder is 1/3. The structure of Turbo coder is illustrated in figure 4.

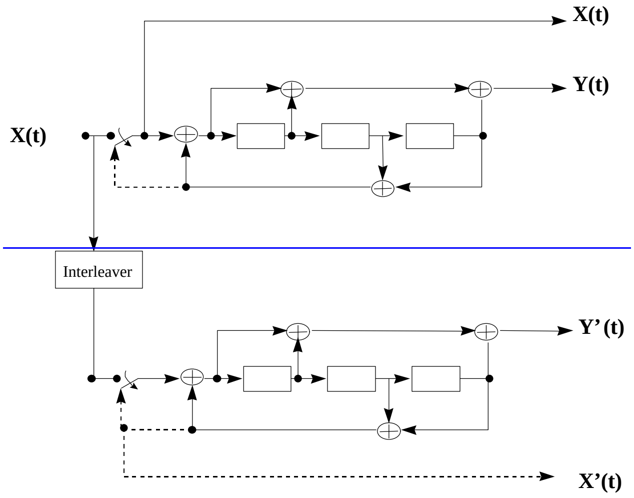
The transfer function of the 8-state constituent code for PCCC is

$$G(D) = \left[1, \frac{n(D)}{d(D)}\right]\left[1, \frac{g_1(D)}{g_0(D)}\right],$$

where~~,~~

$$~~d~~\ g_0(D) = 1 + D^2 + D^3,$$

$$~~n~~\ g_1(D) = 1 + D + D^3.$$



~~Figure 4: Structure of the 8 state PCCC encoder (dotted lines effective for trellis termination only)~~

The initial value of the shift registers of the ~~PCCC~~ 8-state constituent encoders shall be all zeros when starting to encode the input bits.

~~The o~~Output ~~of the PCCC encoder is punctured to produce coded bits corresponding to the desired code rate. For rate 1/3, none of the systematic or parity bits are punctured, and the output sequence~~ from the Turbo coder is ~~X(0), Y(0), Y'(0), X(1), Y(1), Y'(1), etc.~~

$$x_1, z_1, z'_1, x_2, z_2, z'_2, \ldots, x_K, z_K, z'_K,$$

where $x_1, x_2, \ldots, x_K$ are the bits input to the Turbo coder i.e. both first 8-state constituent encoder and Turbo code internal interleaver, and K is the number of bits, and $z_1, z_2, \ldots, z_K$ and $z'_1, z'_2, \ldots, z'_K$ are the bits output from first and second 8-state constituent encoders, respectively.

The bits output from Turbo code internal interleaver are denoted by $x'_1, x'_2, \ldots, x'_K$, and these bits are to be input to the second 8-state constituent encoder.

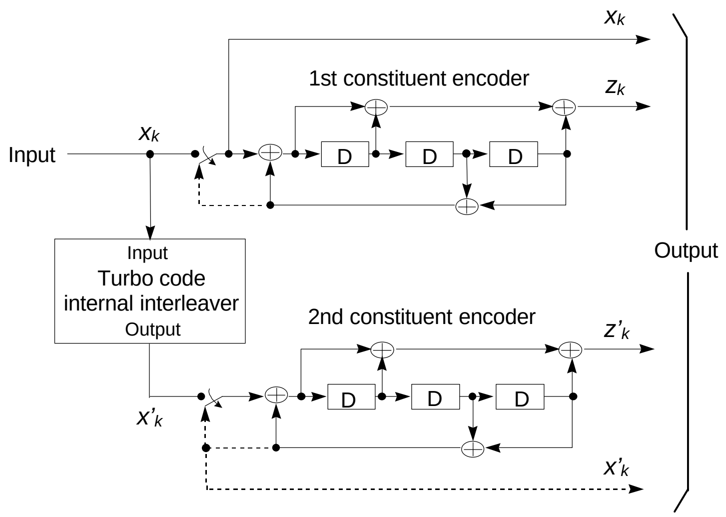


Figure 4: Structure of rate 1/3 Turbo coder (dotted lines apply for trellis termination only)

4.2.3.2.2 Trellis termination for Turbo cod~~ing~~ering

Trellis termination is performed by taking the tail bits from the shift register feedback after all information bits are encoded. Tail bits are padded after the encoding of information bits.

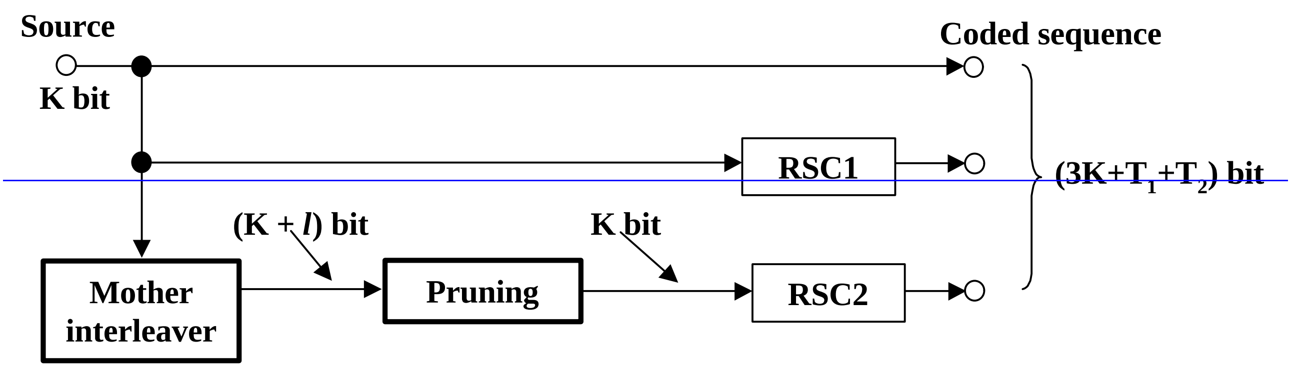
The first three tail bits shall be used to terminate the first constituent encoder (upper switch of figure 4 in lower position) while the second constituent encoder is disabled. The last three tail bits shall be used to terminate the second constituent encoder (lower switch of figure 4 in lower position) while the first constituent encoder is disabled.

The transmitted bits for trellis termination shall then be

~~X(t) Y(t) X(t+1) Y(t+1) X(t+2) Y(t+2) X'(t) Y'(t) X'(t+1) Y'(t+1) X'(t+2) Y'(t+2)~~ $x_{K+1}, z_{K+1}, x_{K+2}, z_{K+2}, x_{K+3}, z_{K+3}, x'_{K+1}, z'_{K+1}, x'_{K+2}, z'_{K+2}, x'_{K+3}, z'_{K+3}$.

4.2.3.2.3 Turbo code internal interleaver

~~Figure 5 depicts the overall 8 state PCCC Turbo coding scheme including Turbo code internal interleaver.~~ The Turbo code internal interleaver consists of bits-input to a rectangular matrix, intra-row and inter-row permutations of the rectangular matrix, and bits-output from the rectangular matrix with pruning. The bits input to the Turbo code internal interleaver are denoted by $x_1, x_2, x_3, \ldots, x_K$, where K is the integer number of the bits and takes one value of 40 $\le K \le$ 5114. The relation between the bits input to the Turbo code internal interleaver and the bits input to the channel coding is defined by $x_k = o_{irk}$ and $K = K_i$. ~~of mother interleaver generation and pruning. For arbitrary given block length K, one mother interleaver is selected from the 134 mother interleavers set. The generation scheme of mother interleaver is described in section 4.2.3.2.3.1. After the mother interleaver generation, *l* bits are pruned in order to adjust the mother interleaver to the block length K. Tail bits T_1 and T_2 are added for constituent encoders RSC1 and RSC2, respectively. The definition of *l* is shown in section 4.2.3.2.3.2.~~



~~Figure 5: Overall 8 State PCCC Turbo Coding~~

The following section specific symbols are used in sections 4.2.3.2.3.1 – 4.2.3.4.3.3:

K	Number of bits input to Turbo code internal interleaver
R	Number of rows of rectangular matrix
C	Number of columns of rectangular matrix
p	Prime number
v	Primitive root
$s(i)$	Base sequence for intra-row permutation
q_j	Minimum prime integers
r_j	Permuted prime integers
$T(j)$	Inter-row permutation pattern
$U_j(i)$	Intra-row permutation pattern
i	Index of matrix

j Index of matrix
k Index of bit sequence

4.2.3.2.3.1 Bits-input to rectangular matrixMother interleaver generation

The bit sequence input to the Turbo code internal interleaver x_k The interleaving consists of three stages. In first stage, the input sequence is written into the rectangular matrix as follows: row by row. The second stage is intra-row permutation. The third stage is inter-row permutation. The three-stage permutations are described as follows, the input block length is assumed to be K (320 to 5114 bits).

First Stage:

(1) Determine the number of rows R of the rectangular matrix such that

$$R = \begin{cases} 5, \text{if } (40 \le K \le 159) \\ 10, \text{if } ((160 \le K \le 200) \text{ or } (481 \le K \le 530)) \\ 20, \text{if } (K = \text{any other value}) \end{cases}$$

R = 10 (K = 481 to 530 bits; Case 1)
R = 20 (K = any other block length except 481 to 530 bits; Case 2)

where the rows of rectangular matrix are numbered 0, 1, 2, …, R - 1 from top to bottom.

(2) Determine the number of columns C of rectangular matrix such that

if (481 ≤ K ≤ 530) then
  p = 53 and Case-1; C = p = 53.
else Case-2;
  (i) fFind minimum prime p such that,
    0 =< (p + 1) - K/R ≥ 0,
  and determine C such that,
  (ii) if (0 =< p - K/R ≥ 0) then go to (iii),
    if (p - 1 - K/R ≥ 0) then
      C = p - 1.
    else
      C = p.
    end if
  else
    C = p + 1.
  end if
end if

where the columns of rectangular matrix are numbered 0, 1, 2, …, C - 1 from left to right.

(iii) if (0 =< p-1-K/R) then C = p-1,

else C = p.

(3) Write Tthe input bit sequence x_k of the interleaver is written into the $R \times C$ rectangular matrix row by row starting with bit x_1 from in column 0 of row 0.:

$$\begin{bmatrix} x_1 & x_2 & x_3 & \cdots x_C \\ x_{(C+1)} & x_{(C+2)} & x_{(C+3)} & \cdots x_{2C} \\ \vdots & \vdots & \vdots & \cdots \vdots \\ x_{((R-1)C+1)} & x_{((R-1)C+2)} & x_{((R-1)C+3)} & \cdots x_{RC} \end{bmatrix}.$$

Second Stage:

A. If C = p

4.2.3.2.3.2 Intra-row and inter-row permutations

After the bits-input to the $R \times C$ rectangular matrix, the intra-row and inter-row permutations are performed by using the following algorithm:

(1) ~~(A-1)~~ Select a primitive root ~~g_0~~ v from table 2.

(2) ~~(A-2)~~ Construct the base sequence ~~c~~ $s(i)$ for intra-row permutation as~~:~~

~~c~~ $s(i) = [$ ~~g_0~~ $v \times$ ~~c~~ $s(i-1)] \bmod p$, $i = 1, 2, \ldots, (p-2)$., and ~~c~~ $s(0) = 1$.

(3) ~~(A-3)~~ Let $q_0 = 1$ be the first prime integer in $\{q_j\}$, and ~~S~~select the consecutive minimum prime integers ~~set~~ $\{q_j\}$ (j = 1, 2, …, R - 1) such that

g.c.d$\{q_j, p - 1\} = 1$,

$q_j > 6$, and

$q_j > q_{(j-1)}$,

where g.c.d. is greatest common ~~divider~~divisor. ~~And $q_0 = 1$.~~

(4) ~~(A-4)~~ Permute ~~The set~~ $\{q_j\}$ ~~is permuted~~ to make ~~a new set~~ $\{$~~p~~$r_j\}$ such that

~~$p_{P(j)}$~~ $r_{T(j)} = q_j$, $j = 0, 1, \ldots, R-1$,

where ~~P~~$T(j)$ indicates the original row position of the j-th permuted row, and $T(j)$ is the inter-row permutation pattern defined as the one of the following four kind of patterns: Pat_1, Pat_2, Pat_3 and Pat_4 depending on the number of input bits K. ~~in the third stage.~~

$$T(j) = \begin{cases} Pat_4 & \text{if } (40 \le K \le 159) \\ Pat_3 & \text{if } (160 \le K \le 200) \\ Pat_1 & \text{if } (201 \le K \le 480) \\ Pat_3 & \text{if } (481 \le K \le 530) \\ Pat_1 & \text{if } (531 \le K \le 2280) \\ Pat_2 & \text{if } (2281 \le K \le 2480) \\ Pat_1 & \text{if } (2481 \le K \le 3160) \\ Pat_2 & \text{if } (3161 \le K \le 3210) \\ Pat_1 & \text{if } (3211 \le K \le 5114) \end{cases},$$

where Pat_1, Pat_2, Pat_3 and Pat_4 have the following patterns respectively.

Pat_1: {19, 9, 14, 4, 0, 2, 5, 7, 12, 18, 10, 8, 13, 17, 3, 1, 16, 6, 15, 11}
Pat_2: {19, 9, 14, 4, 0, 2, 5, 7, 12, 18, 16, 13, 17, 15, 3, 1, 6, 11, 8, 10}
Pat_3: {9, 8, 7, 6, 5, 4, 3, 2, 1, 0}
Pat_4: {4, 3, 2, 1, 0}

(5) ~~(A-5)~~ Perform the j-th (j = 0, 1, 2, …, R - 1) intra-row permutation as~~:~~

if ($C = p$) then

~~c~~ $U_j(i) =$ ~~c~~ $s([i \times$ ~~p~~ $r_j] \bmod (p-1))$, $i = 0, 1, 2, \ldots, (p-2)$., and ~~$c$~~ $U_j(p-1) = 0$,

where ~~c~~ $U_j(i)$ is the input bit position of i-th output after the permutation of j-th row.

end if

~~B. I~~if ($C = p + 1$) then

~~(B-1) Same as case A-1.~~
~~(B-2) Same as case A-2.~~
~~(B-3) Same as case A-3.~~
~~(B-4) Same as case A-4.~~

(B-5) Perform the j-th (j = 0,1, 2, …, R-1) intra-row permutation as:

$cU_j(i) = cs([i \times pr_j] \bmod (p-1))$, $i = 0, 1, 2, \ldots, (p-2)$., $cU_j(p-1) = 0$, and $cU_j(p) = p$,

(B-6) If (K = C × R) then exchange $c_{R-1}(p)$ with $c_{R-1}(0)$.

where $cU_j(i)$ is the input bit position of i-th output after the permutation of j-th row., and

if ($K = C \times R$) then

Exchange $U_{R-1}(p)$ with $U_{R-1}(0)$.

end if

end if

C. Iif ($C = p - 1$) then

(C-1) Same as case A-1.

(C-2) Same as case A-2.

(C-3) Same as case A-3.

(C-4) Same as case A-4.

(C-5) Perform the j-th (j = 0,1, 2, …, R-1) intra-row permutation as:

$cU_j(i) = cs([i \times pr_j] \bmod (p-1)) - 1$, $i = 0, 1, 2, \ldots, (p-2)$,

where $cU_j(i)$ is the input bit position of i-th output after the permutation of j-th row.

end if

Third Stage:

(1) Perform the inter-row permutation based on the following P(j) (j = 0, 1, ..., R - 1) patterns, where P(j) is the original row position of the j-th permuted row.

P_A: {19, 9, 14, 4, 0, 2, 5, 7, 12, 18, 10, 8, 13, 17, 3, 1, 16, 6, 15, 11} for R = 20

P_B: {19, 9, 14, 4, 0, 2, 5, 7, 12, 18, 16, 13, 17, 15, 3, 1, 6, 11, 8, 10} for R = 20

P_C: {9, 8, 7, 6, 5, 4, 3, 2, 1, 0} for R = 10

The usage of these patterns is as follows:

Block length K: P(j)

320 to 480-bit: P_A

481 to 530-bit: P_C

531 to 2280-bit: P_A

2281 to 2480-bit: P_B

2481 to 3160-bit: P_A

3161 to 3210-bit: P_B

3211 to 5114-bit: P_A

(2) The output of the mother interleaver is the sequence read out column by column from the permuted R × C matrix starting from column 0.

Table 2: Table of prime *p* and associated primitive root *v*

p	v	p	v	p	v	p	v	p	v
7	3	47	5	101	2	157	5	223	3
11	2	53	2	103	5	163	2	227	2
13	2	59	2	107	2	167	5	229	6
17	3	61	2	109	6	173	2	233	3
19	2	67	2	113	3	179	2	239	7
23	5	71	7	127	3	181	2	241	7
29	2	73	5	131	2	191	19	251	6
31	3	79	3	137	3	193	5	257	3
37	2	83	2	139	2	197	2		
41	6	89	3	149	2	199	3		
43	3	97	5	151	6	211	2		

~~p~~	~~g_0~~	~~p~~	~~g_0~~	~~p~~	~~g_0~~	~~p~~	~~g_0~~	~~p~~	~~g_0~~
~~17~~	~~3~~	~~59~~	~~2~~	~~103~~	~~5~~	~~157~~	~~5~~	~~211~~	~~2~~
~~19~~	~~2~~	~~61~~	~~2~~	~~107~~	~~2~~	~~163~~	~~2~~	~~223~~	~~3~~
~~23~~	~~5~~	~~67~~	~~2~~	~~109~~	~~6~~	~~167~~	~~5~~	~~227~~	~~2~~
~~29~~	~~2~~	~~71~~	~~7~~	~~113~~	~~3~~	~~173~~	~~2~~	~~229~~	~~6~~
~~31~~	~~3~~	~~73~~	~~5~~	~~127~~	~~3~~	~~179~~	~~2~~	~~233~~	~~3~~
~~37~~	~~2~~	~~79~~	~~3~~	~~131~~	~~2~~	~~181~~	~~2~~	~~239~~	~~7~~
~~41~~	~~6~~	~~83~~	~~2~~	~~137~~	~~3~~	~~191~~	~~19~~	~~241~~	~~7~~
~~43~~	~~3~~	~~89~~	~~3~~	~~139~~	~~2~~	~~193~~	~~5~~	~~251~~	~~6~~
~~47~~	~~5~~	~~97~~	~~5~~	~~149~~	~~2~~	~~197~~	~~2~~	~~257~~	~~3~~
~~53~~	~~2~~	~~101~~	~~2~~	~~151~~	~~6~~	~~199~~	~~3~~		

4.2.3.2.3.3~~2~~ Bits-output from rectangular matrix with ~~Definition of number of~~ pruning ~~bits~~

After intra-row and inter-row permutations, the bits of the permuted rectangular matrix are denoted by y'_k:

$$\begin{bmatrix} y'_1 & y'_{(R+1)} & y'_{(2R+1)} & \cdots y'_{((C-1)R+1)} \\ y'_2 & y'_{(R+2)} & y'_{(2R+2)} & \cdots y'_{((C-1)R+2)} \\ \vdots & \vdots & \vdots & \cdots \quad \vdots \\ y'_R & y'_{2R} & y'_{3R} & \cdots \quad y'_{CR} \end{bmatrix}.$$

The output of the Turbo code internal interleaver is the bit sequence read out column by column from the intra-row and inter-row permuted $R \times C$ matrix starting with bit y'_1 in row 0 of column 0 and ending with bit y'_{CR} in row R - 1 of column C - 1. The output is pruned by deleting bits that were not present in the input bit sequence, i.e. bits y'_k that corresponds to bits x_k with $k > K$ are removed from the output. The bits output from Turbo code internal interleaver are denoted by $x'_1, x'_2, \ldots, x'_K$, where x'_1 corresponds to the bit y'_k with smallest index k after pruning, x'_2 to the bit y'_k with second smallest index k after pruning, and so on. ~~The output of the mother interleaver is pruned by deleting the *l*-bits in order to adjust the mother interleaver to the block length K, where the deleted bits are non-existent bits in the input sequence.~~ The number of bits output from Turbo code internal interleaver is K and ~~T~~the total number of pruneding bits ~~number *l*~~ is ~~defined as:~~

~~$l =$~~ $R \times C - K$.~~;~~

~~where R is the row number and C is the column number defined in section 4.2.3.2.3.1~~

4.2.3.3 Concatenation of encoded blocks

After the channel coding for each code block, if C_i is greater than 1, the encoded blocks are serially concatenated so that the block with lowest index r is output first from the channel coding block, otherwise the encoded block is output from channel coding block as it is. The bits output are denoted by $c_{i1}, c_{i2}, c_{i3}, \ldots, c_{iE_i}$, where i is the TrCH number and $E_i = C_i Y_i$. The output bits are defined by the following relations:

$c_{ik} = y_{i1k}$ $k = 1, 2, \ldots, Y_i$

$c_{ik} = y_{i,2,(k-Y_i)}$ $k = Y_i + 1, Y_i + 2, \ldots, 2Y_i$

$c_{ik} = y_{i,3,(k-2Y_i)}$ $k = 2Y_i + 1, 2Y_i + 2, \ldots, 3Y_i$

$\ldots$

$c_{ik} = y_{i,C_i,(k-(C_i-1)Y_i)}$ $k = (C_i - 1)Y_i + 1, (C_i - 1)Y_i + 2, \ldots, C_i Y_I$

If no code blocks are input to the channel coding ($C_i = 0$), no bits shall be output from the channel coding, i.e. $E_i = 0$.

3GPP RAN WG1 Meeting #11
San Diego, USA, 29 Feb - 3 Mar 2000

***Document* R1-00-0437**
e.g. for 3GPP use the format TP-99xxx
or for SMG, use the format P-99-xxx

CHANGE REQUEST

Please see embedded help file at the bottom of this page for instructions on how to fill in this form correctly.

25.222 CR 029r1 Current Version: 3.1.1

GSM (AA.BB) or 3G (AA.BBB) specification number ↑ *↑ CR number as allocated by MCC support team*

For submission to: RAN #7
list expected approval meeting # here ↑

for approval	X	strategic		*(for SMG use only)*
for information		non-strategic		

Form: CR cover sheet, version 2 for 3GPP and SMG The latest version of this form is available from: ftp://ftp.3gpp.org/Information/CR-Form-v2.doc

Proposed change affects: (U)SIM [ ] ME [X] UTRAN / Radio [X] Core Network [ ]
(at least one should be marked with an X)

Source: NTT DoCoMo, Nortel Networks and Nokia **Date:** 3-Mar-2000

Subject: Editorial changes of channel coding section

Work item:

Category:
(only one category shall be marked with an X)

F	Correction		
A	Corresponds to a correction in an earlier release		
B	Addition of feature		
C	Functional modification of feature		
D	Editorial modification	X	

Release:

Phase 2	
Release 96	
Release 97	
Release 98	
Release 99	X
Release 00	

Reason for change: To clarify exact functions of channel coding.

Clauses affected: 4.2.3 of TS25.222

Other specs affected:

Other 3G core specifications		→ List of CRs:
Other GSM core specifications		→ List of CRs:
MS test specifications		→ List of CRs:
BSS test specifications		→ List of CRs:
O&M specifications		→ List of CRs:

Other comments: This CR is including content of approved CR25.222-021 and changes for Table 4.2.3.1 as stated in CR25.222-017.

$$o_{iC_ik} = x_{i(k+(C_i-1)K_i)} \quad k = 1, 2, \ldots, K_i - Y_i$$

$$o_{iC_ik} = 0 \quad k = (K_i - Y_i) + 1, (K_i - Y_i) + 2, \ldots, K_i$$

4.2.3 Channel coding

Code blocks are delivered to the channel coding block. They are denoted by $o_{ir1}, o_{ir2}, o_{ir3}, \ldots, o_{irK_i}$, where i is the TrCH number, r is the code block number, and K_i is the number of bits in each code block. The number of code blocks on TrCH i is denoted by C_i. After encoding the bits are denoted by $y_{ir1}, y_{ir2}, y_{ir3}, \ldots, y_{irY_i}$ ~~.~~, where Y_i is the number of encoded bits. ~~The encoded blocks are serially multiplexed so that the block with lowest index r is output first from the channel coding block. The bits output are denoted by $c_{i1}, c_{i2}, c_{i3}, \ldots, c_{iE_i}$, where i is the TrCH number and $E_i = C_iY_i$. The output bits are defined by the following relations:~~

~~$c_{ik} = y_{i1k} \quad k = 1, 2, \ldots, Y_i$~~

~~$c_{ik} = y_{i,2,(k-Y_i)} \quad k = Y_i + 1, Y_i + 2, \ldots, 2Y_i$~~

~~$c_{ik} = y_{i,3,(k-2Y_i)} \quad k = 2Y_i + 1, 2Y_i + 2, \ldots, 3Y_i$~~

~~…~~

~~$c_{ik} = y_{i,C_i,(k-(C_i-1)Y_i)} \quad k = (C_i - 1)Y_i + 1, (C_i - 1)Y_i + 2, \ldots, C_iY_i$~~

The relation between o_{irk} and ~~Y~~y_{irk} and between K_i and Y_i is dependent on the channel coding scheme.

The following channel coding schemes can be applied to transport channels:

- Convolutional coding
- Turbo coding
- No ~~channel~~ coding

Usage of coding scheme and coding rate for the different types of TrCH is shown in table 4.2.3-1. The values of Y_i in connection with each coding scheme:

- Convolutional coding~~, ½~~ with rate 1/2: $Y_i = 2*K_i + 16$; rate 1/3 ~~rate~~: $Y_i = 3*K_i + 24$
- Turbo coding~~,~~ with rate 1/3 ~~rate~~: $Y_i = 3*K_i + 12$
- No ~~channel~~ coding~~,~~: $Y_i = K_i$

Table 4.2.3-1: Usage of channel coding scheme and coding rate ~~Error Correction Coding Parameters~~

Type of TrCH	Coding scheme	Coding rate
BCH	Convolutional coding	1/2
PCH		
RACH		
DCH, DSCH, FACH, USCH		1/3, 1/2
	Turbo coding	1/3
	No coding	

Transport channel type	Coding scheme	Coding rate
BCH	Convolutional code	1/2
PCH		
FACH		
RACH		
DCH, DSCH, USCH		1/3, 1/2
	Turbo code	1/3
	No coding	

4.2.3.1 Convolutional Ccoding

- Convolutional codes with Cconstraint length K=9. and Ccoding rates 1/3 and 1/2 are definedand 1/3.

- The configuration of the convolutional coder is presented in figure 4-2.

- The oOutput from the rate 1/3 convolutional coder shall be done in the order output0, output1, output2, output0, output1, output2, output0, …, output2. (When coding Output from the rate is 1/2 convolutional coder shall be done in the order, output0, output1, output0, output1, output0, …, is done up to output 1).

8 tail bits with binary value 0 shall be added to the end of the code block before encoding.

- The initial value of the shift register of the coder shall be "all 0" when starting to encode the input bits.

- K-1 tail bits (value 0) shall be added to the end of the code block before encoding.

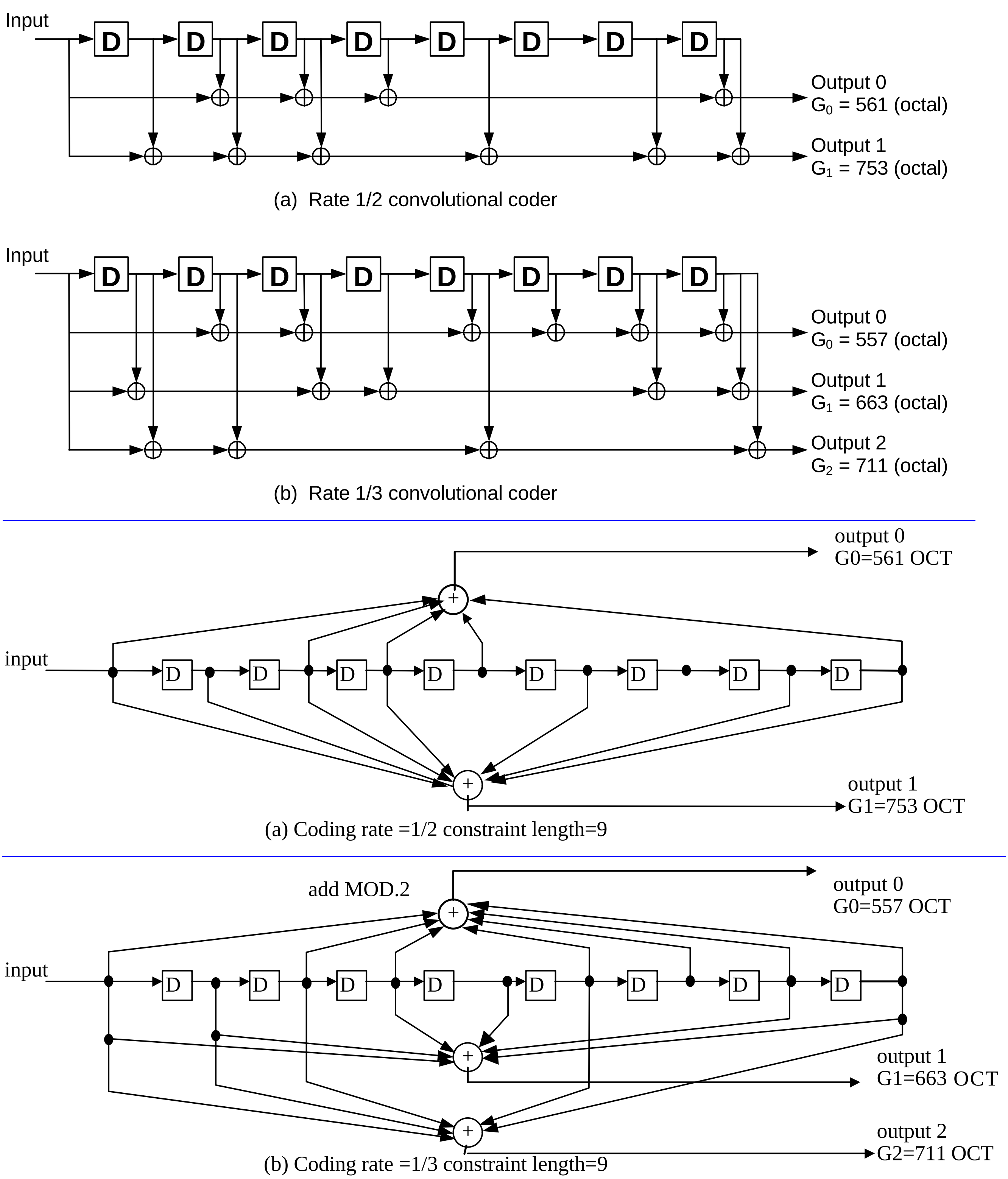


Figure 4-2: Rate 1/2 and rate 1/3 ~~C~~convolutional ~~C~~coders

4.2.3.2 Turbo coding

4.2.3.2.1 Turbo coder

The scheme of Turbo coder is a ~~For data services requiring quality of service between 10^{-3} and 10^{-6} BER inclusive, p~~Parallel ~~c~~Concatenated ~~c~~Convolutional ~~c~~Code (PCCC) with two 8-state constituent encoders and one Turbo code internal interleaveris used. The coding rate of Turbo coder is 1/3. The structure of Turbo coder is illustrated in figure 4-3.

The transfer function of the 8-state constituent code for PCCC is

$$G(D) = \left[1, \frac{n(D)}{d(D)}\right]\left[1, \frac{g_1(D)}{g_0(D)}\right],$$

where,

$d\ g_0(D) = 1 + D^2 + D^3,$

$n\ g_1(D) = 1 + D + D^3.$

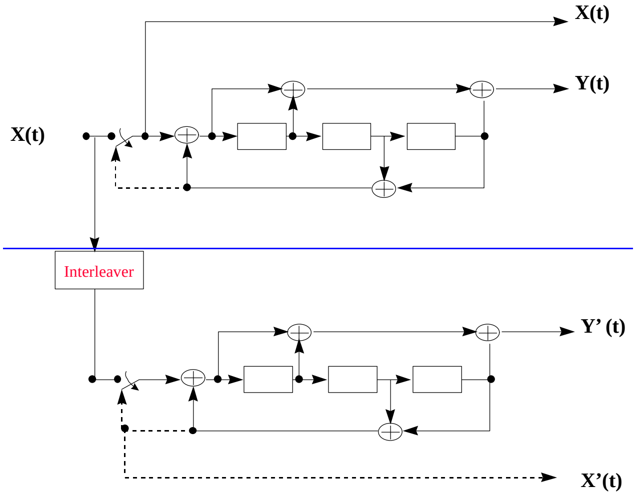


Figure 4-3: Structure of the 8-state PCCC encoder (dotted lines effective for trellis termination only)

The initial value of the shift registers of the PCCC 8-state constituent encoders shall be all zeros when starting to encode the input bits.

The oOutput of the PCCC encoder is punctured to produce coded bits corresponding to the desired code rate. For rate 1/3, none of the systematic or parity bits are punctured, and the output sequence from the Turbo coder is X(0), Y(0), Y'(0), X(1), Y(1), Y'(1), etc.

$x_1, z_1, z'_1, x_2, z_2, z'_2, \ldots, x_K, z_K, z'_K,$

where $x_1, x_2, \ldots, x_K$ are the bits input to the Turbo coder i.e. both first 8-state constituent encoder and Turbo code internal interleaver, and K is the number of bits, and $z_1, z_2, \ldots, z_K$ and $z'_1, z'_2, \ldots, z'_K$ are the bits output from first and second 8-state constituent encoders, respectively.

The bits output from Turbo code internal interleaver are denoted by $x'_1, x'_2, \ldots, x'_K$, and these bits are to be input to the second 8-state constituent encoder.

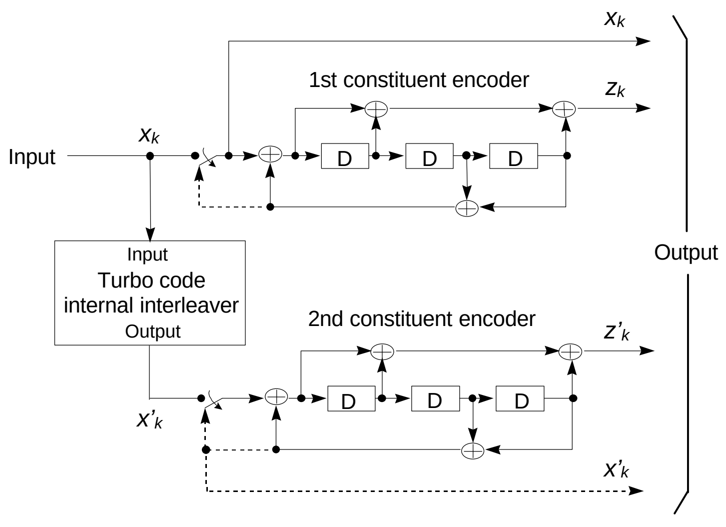


Figure 4-3: Structure of rate 1/3 Turbo coder (dotted lines apply for trellis termination only)

4.2.3.2.2 Trellis termination ~~in~~for ~~t~~Turbo coder

Trellis termination is performed by taking the tail bits from the shift register feedback after all information bits are encoded. Tail bits are padded after the encoding of information bits.

The first three tail bits shall be used to terminate the first constituent encoder (upper switch of figure 4-3 in lower position) while the second constituent encoder is disabled. The last three tail bits shall be used to terminate the second constituent encoder (lower switch of figure 4-3 in lower position) while the first constituent encoder is disabled.

The transmitted bits for trellis termination shall then be

~~X(t) Y(t) X(t+1) Y(t+1) X(t+2) Y(t+2) X'(t) Y'(t) X'(t+1) Y'(t+1) X'(t+2) Y'(t+2)~~ $x_{K+1}, z_{K+1}, x_{K+2}, z_{K+2}, x_{K+3}, z_{K+3}, x'_{K+1}, z'_{K+1}, x'_{K+2}, z'_{K+2}, x'_{K+3}, z'_{K+3}$.

4.2.3.2.3 Turbo code internal interleaver

~~Figure 4-4 depicts the overall 8-State PCCC Turbo coding scheme including Turbo code internal interleaver.~~ The Turbo code internal interleaver consists of bits-input to a rectangular matrix, intra-row and inter-row permutations of the rectangular matrix, and bits-output from the rectangular matrix with pruning.~~mother interleaver generation and pruning.~~ The bits input to the Turbo code internal interleaver are denoted by $x_1, x_2, x_3, \ldots, x_K$, where K is the integer number of the bits and takes one value of 40 $\leq K \leq$ 5114. The relation between the bits input to the Turbo code internal interleaver and the bits input to the channel coding is defined by $x_k = o_{irk}$ and $K = K_i$.~~For arbitrary given block length K, one mother interleaver is selected from the 134 mother interleavers set. The generation scheme of mother interleaver is described in section 4.2.3.2.3.1. After the mother interleaver generation, *l*-bits are pruned in order to adjust the mother interleaver to the block length K. Tail bits T_1 and T_2 are added for constituent encoders RSC1 and RSC2, respectively. The definition of *l* is shown in section 4.2.3.2.3.2..~~

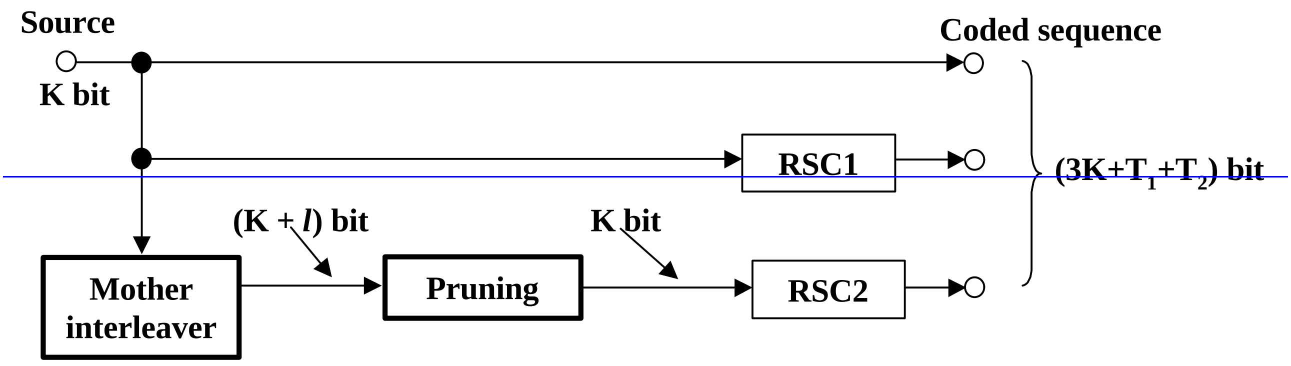


Figure 4-4: Overall 8 State PCCC Turbo Coding

The following section specific symbols are used in sections 4.2.3.2.2.3.1 – 4.2.3.4.3.3:

K	Number of bits input to Turbo code internal interleaver
R	Number of rows of rectangular matrix
C	Number of columns of rectangular matrix
p	Prime number
v	Primitive root
$s(i)$	Base sequence for intra-row permutation
q_j	Minimum prime integers
r_j	Permuted prime integers
$T(j)$	Inter-row permutation pattern
$U_j(i)$	Intra-row permutation pattern
i	Index of matrix
j	Index of matrix
k	Index of bit sequence

4.2.3.2.2.3.1 Bits-input to rectangular matrixMother interleaver generation

The bit sequence input to the Turbo code internal interleaver x_k The interleaving consists of three stages. In first stage, the input sequence is written into the rectangular matrix as follows: row by row. The second stage is intra-row permutation. The third stage is inter-row permutation. The three-stage permutations are described as follows, the input block length is assumed to be K (320 to 5114 bits).

First Stage:

(1) Determine the number of rows R of the rectangular matrix such that

$$R = \begin{cases} 5, \text{if } (40 \le K \le 159) \\ 10, \text{if } ((160 \le K \le 200) \text{ or } (481 \le K \le 530)) \\ 20, \text{if } (K = \text{any other value}) \end{cases}$$ R=10 (K = 481 to 530 bits; Case-1)

R=20 (K = any other block length except 481 to 530 bits; Case-2)

where the rows of rectangular matrix are numbered 0, 1, 2, …, R - 1 from top to bottom.

(2) Determine the number of columns C of rectangular matrix such that

```
if (481 ≤ K ≤ 530) then
    p = 53 and C = p.
else
Case-1; C = p = 53

Case-2;

    (i) fFind minimum prime p such that,
        0 =< (p + 1) - K/R ≥ 0,
    and determine C such that
    (ii) if (0 =< p - K/R ≥ 0) then go to (iii)
        if (p - 1 - K/R ≥ 0) then
            C = p - 1.
        else
            C = p.
        end if
    else
        C = p + 1.
    end if
end if
```

where the columns of rectangular matrix are numbered 0, 1, 2, …, C - 1 from left to right.

(iii) if (0 =< p-1-K/R) then C=p-1.

Else C = p.

(3) Write Tthe input bit sequence x_k of the interleaver is written into the $R \times xC$ rectangular matrix row by row starting with bit x_1 from in column 0 of row 0.:

$$\begin{bmatrix} x_1 & x_2 & x_3 & \dots x_C \\ x_{(C+1)} & x_{(C+2)} & x_{(C+3)} & \dots x_{2C} \\ \vdots & \vdots & \vdots & \dots \vdots \\ x_{((R-1)C+1)} & x_{((R-1)C+2)} & x_{((R-1)C+3)} & \dots x_{RC} \end{bmatrix}.$$

Second Stage:

A. If C = p

4.2.3.2.3.2 Intra-row and inter-row permutations

After the bits-input to the $R \times C$ rectangular matrix, the intra-row and inter-row permutations are performed by using the following algorithm:

(1) (A-1) Select a primitive root g_0 v from table 4.2.23-2.

(2) (A-2) Construct the base sequence cs(i) for intra-row permutation as:

$c(i) = [g_0 \times c(i-1)] \bmod p$ $s(i) = [v \times s(i-1)] \bmod p$, $i = 1, 2, \dots, (p-2)$., and cs(0) = 1.

(3) (A-3) Let $q_0 = 1$ be the first prime integer in $\{q_j\}$, and Sselect the consecutive minimum prime integers set $\{q_j\}$ (j = 1, 2, …, R - 1) such that

g.c.d$\{q_j, p - 1\} = 1$,

$q_j > 6$, and

$q_j > q_{(j-1)}$,

where g.c.d. is greatest common dividerdivisor. And $q_0 = 1$.

(4) (A-4) Permute The set $\{q_j\}$ is permuted to make a new set $\{pr_j\}$ such that

$p_{P(j)} r_{T(j)} = q_j$, $j = 0, 1, \dots, R - 1$,

where PT(j) indicates the original row position of the j-th permuted row, and $T(j)$ is the inter-row permutation pattern defined as the one of the following four kind of patterns: Pat_1, Pat_2, Pat_3 and Pat_4 depending on the number of input bits K. in the third stage.

$$T(j) = \begin{cases} Pat_4 & \text{if } (40 \le K \le 159) \\ Pat_3 & \text{if } (160 \le K \le 200) \\ Pat_1 & \text{if } (201 \le K \le 480) \\ Pat_3 & \text{if } (481 \le K \le 530) \\ Pat_1 & \text{if } (531 \le K \le 2280) \\ Pat_2 & \text{if } (2281 \le K \le 2480) \\ Pat_1 & \text{if } (2481 \le K \le 3160) \\ Pat_2 & \text{if } (3161 \le K \le 3210) \\ Pat_1 & \text{if } (3211 \le K \le 5114) \end{cases},$$

where Pat_1, Pat_2, Pat_3 and Pat_4 have the following patterns respectively.

Pat_1: {19, 9, 14, 4, 0, 2, 5, 7, 12, 18, 10, 8, 13, 17, 3, 1, 16, 6, 15, 11}
Pat_2: {19, 9, 14, 4, 0, 2, 5, 7, 12, 18, 16, 13, 17, 15, 3, 1, 6, 11, 8, 10}
Pat_3: {9, 8, 7, 6, 5, 4, 3, 2, 1, 0}
Pat_4: {4, 3, 2, 1, 0}

(5) (A-5) Perform the j-th (j = 0, 1, 2, …, C R - 1) intra-row permutation as:

if ($C = p$) then

$c_j(i) = c([i \times p_j] \bmod (p-1))$ $U_j(i) = s([i \times r_j] \bmod (p - 1))$, i = 0, 1, 2, …, (p - 2)., and $cU_j(p - 1) = 0$,

where $cU_j(i)$ is the input bit position of i-th output after the permutation of j-th row.

end if

Iif (C = p + 1) then

(B-1) Same as case A-1.

(B-2) Same as case A-2.

(B-3) Same as case A-3.

(B-4) Same as case A-4.

(B-5) Perform the j-th (j = 0,1, 2, …, R-1) intra-row permutation as:

$c_j(i) = c([i \times p_j] \bmod (p-1))$ $U_j(i) = s([i \times r_j] \bmod (p - 1))$, i = 0, 1, 2, …, (p - 2)., $cU_j(p - 1) = 0$, and $cU_j(p) = p$,

where $cU_j(i)$ is the input bit position of i-th output after the permutation of j-th row., and

(B-6) Iif ($K = C \times R$) then

eExhange $cU_{R-1}(p)$ with $cU_{R-1}(0)$.

end if

end if

Iif ($C = p$ - 1) then

(C-1) Same as case A-1.

(C-2) Same as case A-2.

(C-3) Same as case A-3.

(C-4) Same as case A-4.

(C-5) Perform the j-th (j = 0,1, 2, …, R-1) intra-row permutation as:

$c_j(i) = c([i \times p_j] \bmod (p-1))$ $U_j(i) = s([i \times r_j] \bmod (p - 1)) - 1$, i = 0, 1, 2, …, (p - 2).,

where $cU_j(i)$ is the input bit position of i-th output after the permutation of j-th row.

end if

Third Stage:

Perform the inter-row permutation based on the following P(j) (j=0,1, …, R-1) patterns, where P(j) is the original row position of the j-th permuted row.

P_A: {19, 9, 14, 4, 0, 2, 5, 7, 12, 18, 10, 8, 13, 17, 3, 1, 16, 6, 15, 11} for R=20

P_B: {19, 9, 14, 4, 0, 2, 5, 7, 12, 18, 16, 13, 17, 15, 3, 1, 6, 11, 8, 10} for R=20

P_C: {9, 8, 7, 6, 5, 4, 3, 2, 1, 0} for R=10

The usage of these patterns is as follows:

Block length K: P(j)

320 to 480-bit: P_A

481 to 530-bit: P_C

531 to 2280-bit: P_A

~~2281 to 2480-bit: P_B~~

~~2481 to 3160-bit: P_A~~

~~3161 to 3210-bit: P_B~~

~~3211 to 5114-bit: P_A~~

~~(2) The output of the mother interleaver is the sequence read out column by column from the permuted R × C matrix starting from column 0.~~

Table 4.2.3-2: Table of prime *p* and associated primitive root *v*

p	*v*	*p*	*v*	*p*	*v*	*p*	*v*	*p*	*v*
7	3	47	5	101	2	157	5	223	3
11	2	53	2	103	5	163	2	227	2
13	2	59	2	107	2	167	5	229	6
17	3	61	2	109	6	173	2	233	3
19	2	67	2	113	3	179	2	239	7
23	5	71	7	127	3	181	2	241	7
29	2	73	5	131	2	191	19	251	6
31	3	79	3	137	3	193	5	257	3
37	2	83	2	139	2	197	2		
41	6	89	3	149	2	199	3		
43	3	97	5	151	6	211	2		

~~p~~	~~g_0~~	~~p~~	~~g_0~~	~~p~~	~~g_0~~	~~p~~	~~g_0~~	~~p~~	~~g_0~~
~~17~~	~~3~~	~~59~~	~~2~~	~~103~~	~~5~~	~~157~~	~~5~~	~~211~~	~~2~~
~~19~~	~~2~~	~~61~~	~~2~~	~~107~~	~~2~~	~~163~~	~~2~~	~~223~~	~~3~~
~~23~~	~~5~~	~~67~~	~~2~~	~~109~~	~~6~~	~~167~~	~~5~~	~~227~~	~~2~~
~~29~~	~~2~~	~~71~~	~~7~~	~~113~~	~~3~~	~~173~~	~~2~~	~~229~~	~~6~~
~~31~~	~~3~~	~~73~~	~~5~~	~~127~~	~~3~~	~~179~~	~~2~~	~~233~~	~~3~~
~~37~~	~~2~~	~~79~~	~~3~~	~~131~~	~~2~~	~~181~~	~~2~~	~~239~~	~~7~~
~~41~~	~~6~~	~~83~~	~~2~~	~~137~~	~~3~~	~~191~~	~~19~~	~~241~~	~~7~~
~~43~~	~~3~~	~~89~~	~~3~~	~~139~~	~~2~~	~~193~~	~~5~~	~~251~~	~~6~~
~~47~~	~~5~~	~~97~~	~~5~~	~~149~~	~~2~~	~~197~~	~~2~~	~~257~~	~~3~~
~~53~~	~~2~~	~~101~~	~~2~~	~~151~~	~~6~~	~~199~~	~~3~~		

4.2.3.2.3.3~~2~~ Bits-output from rectangular matrix with ~~Definition of the number of~~ pruning ~~bits~~

After intra-row and inter-row permutations, the bits of the permuted rectangular matrix are denoted by y'_k:

$$\begin{bmatrix} y'_1 & y'_{(R+1)} & y'_{(2R+1)} & \cdots y'_{((C-1)R+1)} \\ y'_2 & y'_{(R+2)} & y'_{(2R+2)} & \cdots y'_{((C-1)R+2)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ y'_R & y'_{2R} & y'_{3R} & \cdots & y'_{CR} \end{bmatrix}.$$

The output of the Turbo code internal interleaver is the bit sequence read out column by column from the intra-row and inter-row permuted $R \times C$ matrix starting with bit y'_1 in row 0 of column 0 and ending with bit y'_{CR} in row R - 1 of column C - 1. The output is pruned by deleting bits that were not present in the input bit sequence, i.e. bits y'_k that corresponds to bits x_k with $k > K$ are removed from the output. The bits output from Turbo code internal interleaver are denoted by $x'_1, x'_2, \ldots, x'_K$, where x'_1 corresponds to the bit y'_k with smallest index k after pruning, x'_2 to the bit y'_k with second smallest index k after pruning, and so on. ~~The output of the mother interleaver is pruned by deleting the *l*-bits in order to adjust the mother interleaver to the block length K, where the deleted bits are non-existent bits in the input sequence.~~ The number of bits output from Turbo code internal interleaver is K and ~~T~~the total number of prun~~ing~~ed bits ~~number *l*~~ is ~~defined as:~~

~~l =~~ $R \times C - K$.~~,~~

where R is the row number and C is the column number defined in section 4.2.3.2.3.1.

4.2.3.3 Concatenation of encoded blocks

After the channel coding for each code block, if C_i is greater than 1, the encoded blocks are serially concatenated so that the block with lowest index *r* is output first from the channel coding block, otherwise the encoded block is output from channel coding block as it is. The bits output are denoted by $c_{i1}, c_{i2}, c_{i3}, \ldots, c_{iE_i}$, where *i* is the TrCH number and $E_i = C_iY_i$. The output bits are defined by the following relations:

$$c_{ik} = y_{i1k} \quad k = 1, 2, \ldots, Y_i$$

$$c_{ik} = y_{i,2,(k-Y_i)} \quad k = Y_i + 1, Y_i + 2, \ldots, 2Y_i$$

$$c_{ik} = y_{i,3,(k-2Y_i)} \quad k = 2Y_i + 1, 2Y_i + 2, \ldots, 3Y_i$$

$$\ldots$$

$$c_{ik} = y_{i,C_i,(k-(C_i-1)Y_i)} \quad k = (C_i - 1)Y_i + 1, (C_i - 1)Y_i + 2, \ldots, C_iY_I$$

If no code blocks are input to the channel coding ($C_i = 0$), no bits shall be output from the channel coding, i.e. $E_i = 0$.