3GPP TSG RAN WG1 #47 **R1-070544**
Sorrento, Italy, 14-19 January 2007

Agenda Item: **6.4**

Source: **Broadcom**

Title: **Tail-biting encoding for 3GPP LTE turbo code of arbitrary number of information bits and performance comparison**

Document for: **Discussion/Decision**

1 Introduction

Tail-biting termination that gives equal states at beginning and the end of encoding is needed for today's turbo decoding. 3GPP Rel.6 suggests using 6 bits outsize the interleave to terminate the trellis to 0 state which is also the initial state, we call it tail-terminating. It was suggested that turbo coding in 3GPP LTE to use the tail-biting termination without adding extra terminating symbols.

In the first part of this document, we show the performance gain of tail-biting over terminating by several examples using Rel.6 interleave. The analysis on the implementation tail-biting system is also presented.

The second part of this document, we discuss the possibility of tail-biting for turbo codes. Tail-biting will require tail-biting (or circular) encoder, i.e. for any possible information sequence there always exist an initial state for the encoder such that the final state equals to the initial state. However, in R1-062157 [1], it is shown that Rel.6 turbo encoder [2] is not tail-biting encoder for 1/7 of all possible information sequences. On the other hand, the channel coding system of 3GPP LTE has to support arbitrary number of information bits from a consecutive range of integers (e.g. Rel.6 supports the size from 40 to 5114). Therefore, in order to use tail-biting encoder in 3GPP LTE, it is necessary to solve the problem for 1/7 possible length of information sequence.

In the second part of this document, we first show that all turbo codes have the same problem. A detailed analysis on tail-biting encoder of all possible 8 states turbo codes is also given. We than propose a method which gives all possible information sequence a tail-biting termination with maximal one dummy symbol within the interleaved block. With this tail-biting method we can now claim that tail-biting encoding will not limit the size of information sequence but only give the limitation of interleave size.

2 Tail-biting encoding with performance gain

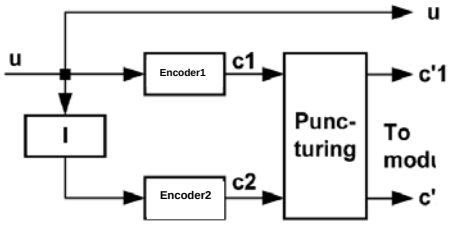

Figure 1, Turbo code.

Definition of tail-biting encoder: Let $u_0,u_1,\ldots,u_{k-1}$ be the information sequence sent to one of the encoders of a turbo code such that $S_0,S_1,\ldots, S_k$ be the corresponded states sequence. The encoding is tail-biting (or circular) for this given information sequence if and only is $S_0=S_k$.

The following 4 figures shows the performance comparisons between tail-biting method and tail terminating method suggested in 3GPP Rel.6 for turbo code. The designed code rate is 1/3. These figures show the obvious performance gain of tail-biting.

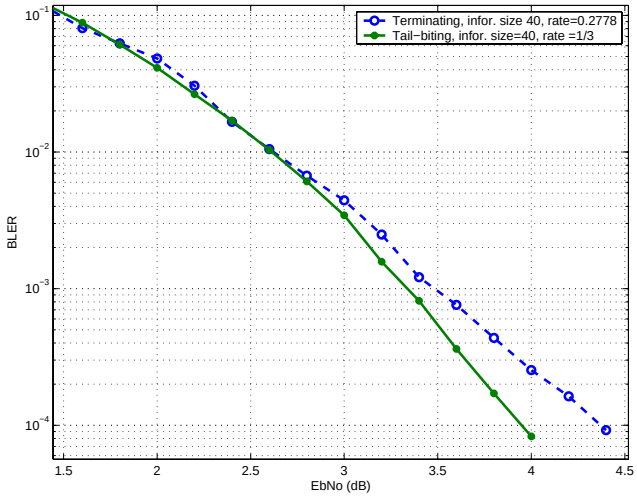

Figure 1 Rate 1/3 turbo code with information size 40 and using Rel.6 interleave

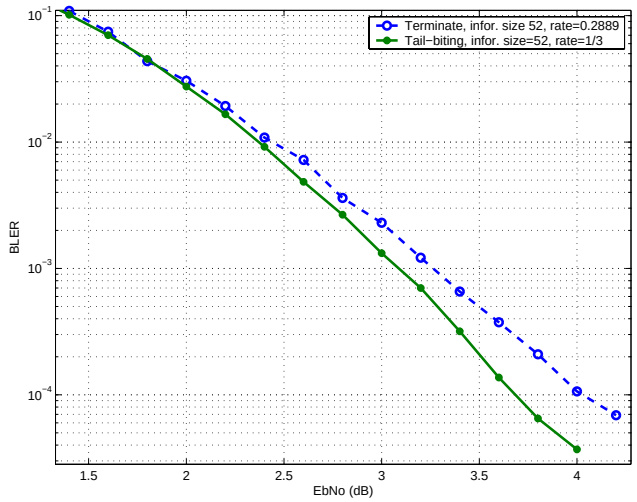

Figure 2 Rate 1/3 turbo code with information size 52 and using Rel.6 interleave

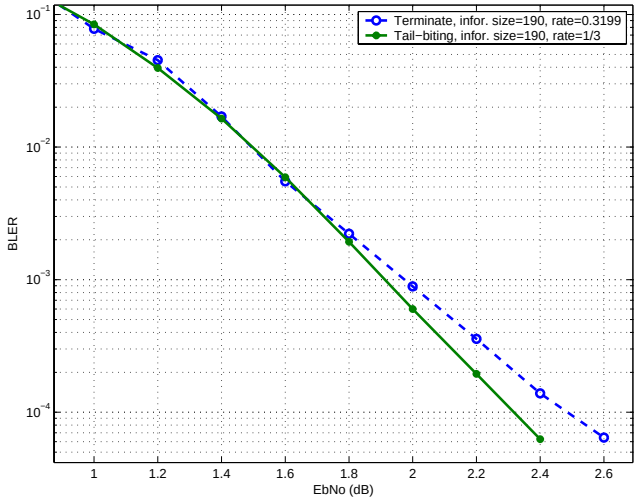

Figure 3 Rate 1/3 turbo code with information size 190 and using Rel.6 interleave

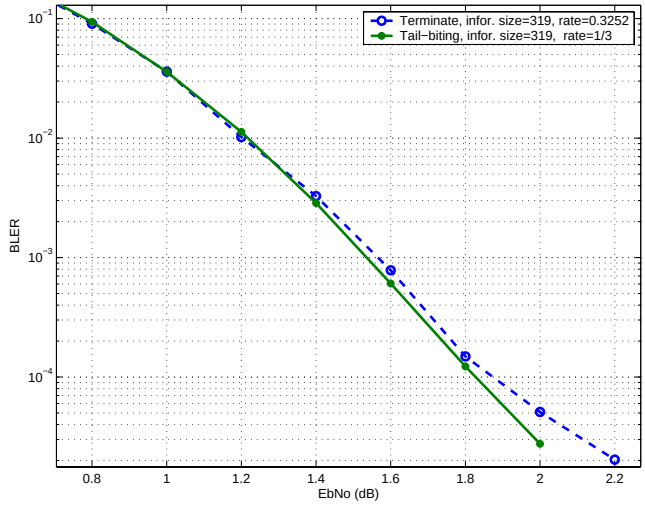

Figure 4 Rate 1/3 turbo code with information size 319 and using Rel.6 interleave

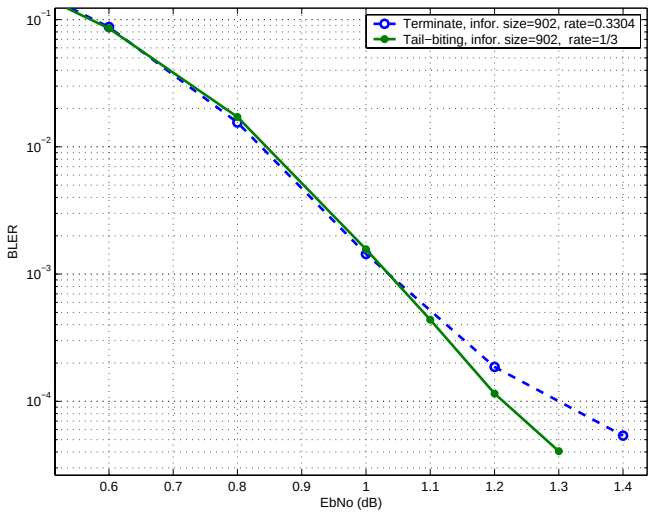

Figure 5 Rate 1/3 turbo code with information size 902 and using Rel.6 interleave

At BLER=1e-4, one can see >0.4 dB gain for information size 40, >0.35dB gain for size 52, 0.17dB gain for size 190, 0.05dB (0.15dB at 2e-5) gain for size 319 and 0.08dB (0.12dB at 4.e-5) gain for size 902. The gain is not negligible, especially for short block sizes.

3 The implementation issues

3.1 Encoding

Different to tail-terminating turbo coding, tail-biting need encoding information bits twice. This will cause extra one frame latency. However, since the encoder is a shift register the following example show that adding extra latency is acceptable or we can modify the encoder to make it up.

Example Suppose the frame size = 5,000 bits, 10 Mbps (0.5 msec. Frame). Let us take the clock = 150 MHZ . The encoder can process 1bit/cycle. Then we have

time/bit=1/(150e+6)=6.7e-9

Tail-terminating encoding: *time/frame* = 5000*6.7e-9=33e-6 seconds
Tail-biting encoding: *time/frame* = 67e-6 seconds

If one wants the tail-biting encoder has the same throughput as terminating encoder, we can do the following:

A) Let encoding process 2 bit/cycle for tail-biting. Then *time/frame* = 33e-6 seconds. The increase of hardware is very small compared to the encoder with 1 bit/cycle

B) Faster clock to 300 MHZ since the encoding has bitwise operation only.

3.2 Decoding

It is obvious that if one does not take advantage of the tail-biting property on decoding, the tail-biting turbo coding performance will be worse then that of tail-

terminating turbo coding. In the following we will explain how we can apply tail-biting property to the decoding without extra hardware and latency cost.

3.2.1 Serial decoding with slide window technique

Iorder to save the area, it is well known in the industry that slide window technique [3] should be used. In the slide-window decoding, the turbo block is divided to several windows, see Figure 6.

To decode tail-terminating turbo code, the start values for computing the forward state metric α are given by previous windows, except the first window. Since the initial state is known, the start values (which equivalent to the probability of states) of the first window can be consider given. The computation starts from window 1 to the following windows. Therefore, the start values for computing the backward state metric β are never known. To solve this problem, a warm up (or cold start) procedure (blue color in Fig.6) has to be added so that the start value can be estimated. This estimate use the bit metric of the next window, except the one for the last window (light blue in Fig. 6). For the last window, the warm up is start from the first window since the final state is equal to the initial state. This procedure is based on "*the fact that the VA can start cold in any state at any time; initially, the state metrics generated are nearly worthless, but after a few constraint lengths, the set of state metrics are as reliable as if the process had been started at the initial (or final) node* [3]." In fact this will not cost any additional hardware since the cold start hardware is the same hardware used in the window and the values (expect the last one which is the start value of the current window) obtained in the warm up time need not be store.

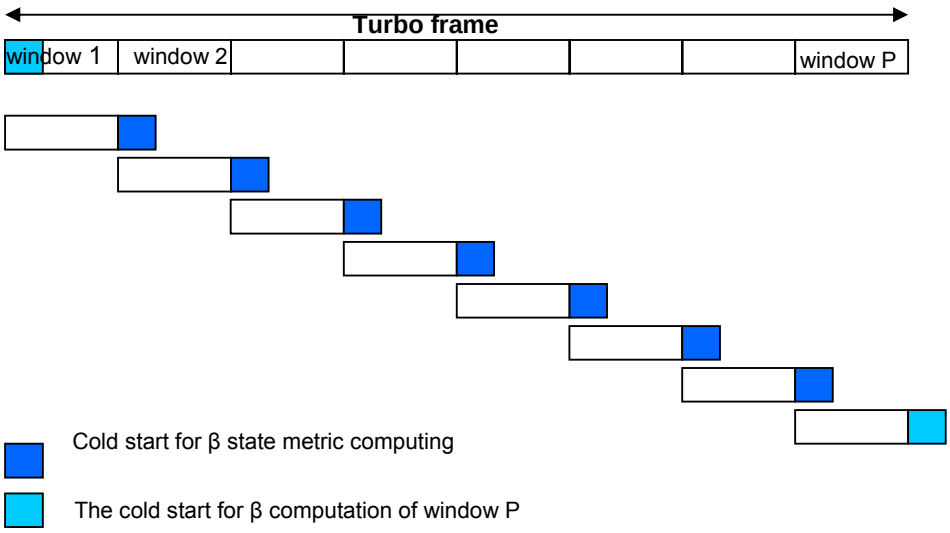

Figure 6 Tail-Terminating decoding with slide windows

To decoding tail-biting turbo code (see Fig.7), the only difference is on the first window which need cold start from the last window since the last state is equal to the first state given by tail-biting. This one additional procedure uses the same hardware for α computation, and it uses the same time as cold state of β. Therefore, every thing

is given. Our simulation results in the previous figures are obtained by using this technique.

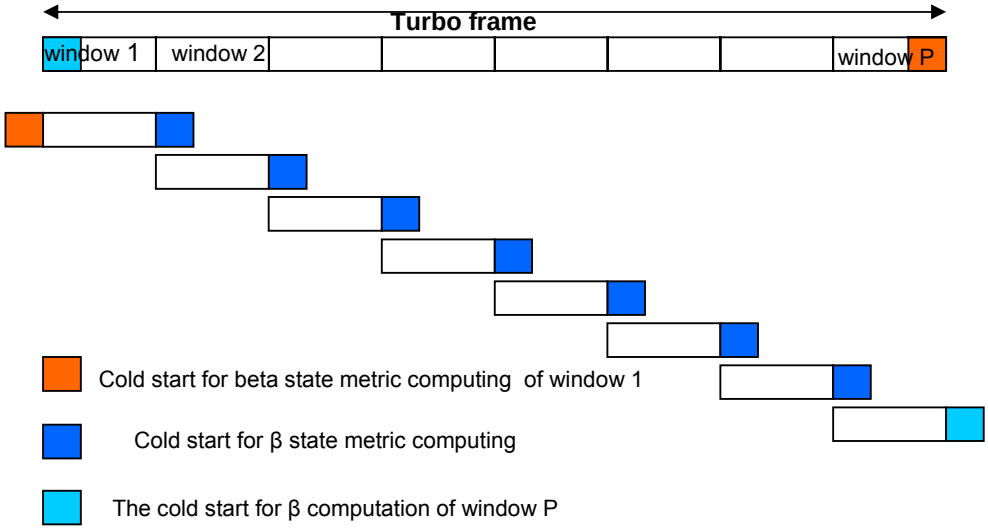

Figure 7 Tail-biting decoding with slide windows

3.2.2 Parallel decoding

The difference of parallel decoding to the slide-window serial decoding is that all α computation and β computation need warm up (see Fig.8 and Fig.9) since all P processors starts at the same time.

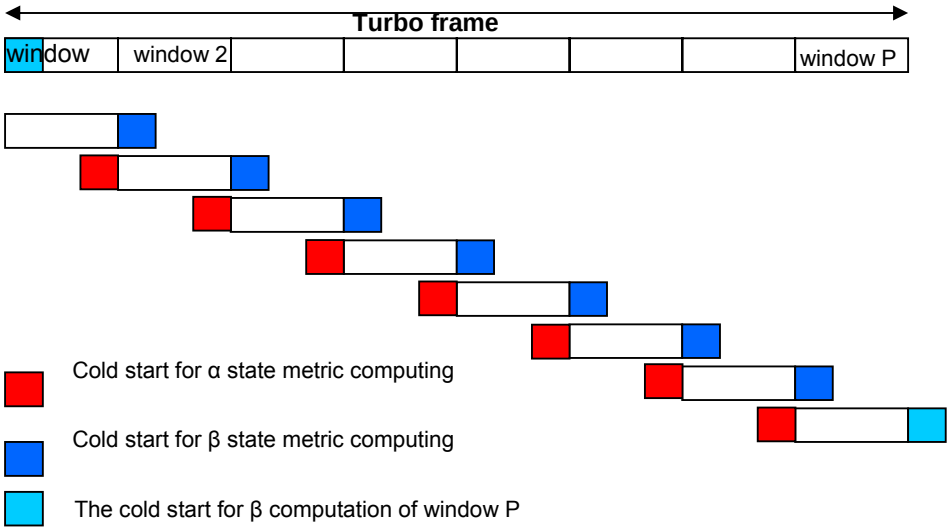

Figure 8 Tail-terminating parallel decoding

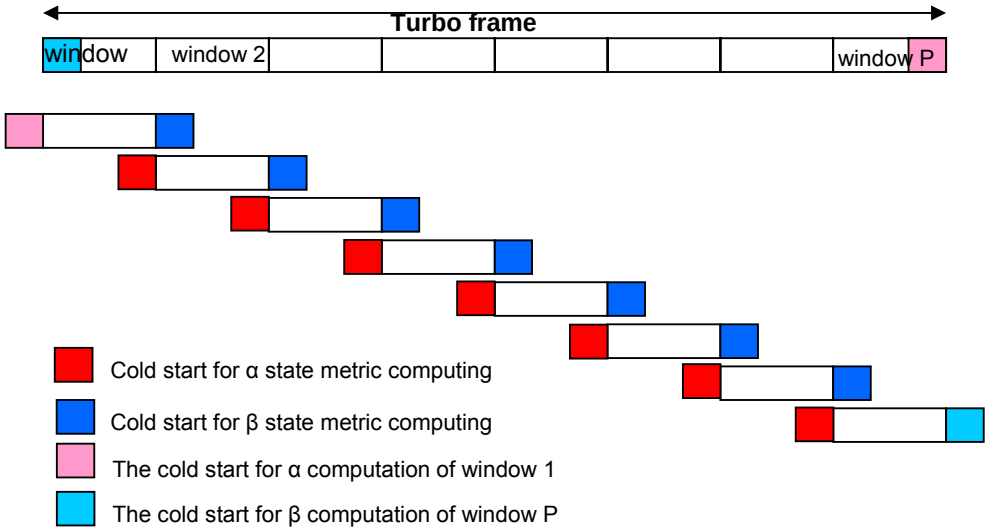

Figure 9 Tail-biting parallel decoding

The only difference between tail-biting and tail-terminating parallel decoding is the cold start for α of the first window, which can be done from the last window (pink color in Fig.9) thanks to the tail-biting. Obviously, there is no need to add extra hardware and latency.

4 State-space realization of convolutional encoder and the necessary and sufficient condition of tail-biting encoder

In this section we recall the definition of state-space realization of convolutional encoder from [4-7]. Using this realization we will give the necessary and sufficient condition of tail-biting encoding.

Consider a rate k_0/n_0 convolutional encoder of degree m, let the input sequence be $\mathbf{u} = (u_0, \cdots, u_{N-1})$ where $u_i = (u_{i,k_0-1}, \cdots, u_{i,0})$ and the output sequence be $\mathbf{x} = (x_0, \cdots, x_{N-1})$ where $x_i = (x_{i,n_0-1}, \cdots, x_{\mathbf{i,0}})$. Moreover, let $\mathbf{S}_t = (s_{m-1}^{(t)}, \cdots, s_0^{(t)})$ be the encoding state at time t. Then there exists an m by m matrix A, m by k_0 matrix B, n_0 by m matrix C and k_0 by n_0 matrix D, which is called *state-space realization* of the encoder, such that

$$\mathbf{S}_t^T = (s_{m-1}^{(t)}, \cdots, s_0^{(t)})^T = A(s_{m-1}^{(t-1)}, \cdots, s_0^{(t-1)})^T + Bu_t^T = A\mathbf{S}_{t-1}^T + Bu_t^{T-1} \qquad \text{(EQ-1)}$$

and $x_t^T = C\mathbf{S}_{t-1}^T + Du_t^T$. The generate matrix of this convolutional encoder is

$$C(A,B,C,D) = G(x) = D + C(x^{-1}I_m - A)^{-1}B \qquad \text{(EQ-2)}$$

Example 1 Consider convolutional encoder in Rel.6 turbo code depictured in Fig.3. The encoder has degree 3 and $s_0^t = s_2^{t-1} + s_1^{t-1} + u_t, s_1^t = s_0^{t-1}, s_2^t = s_1^{t-1}$. Thus,

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, D = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

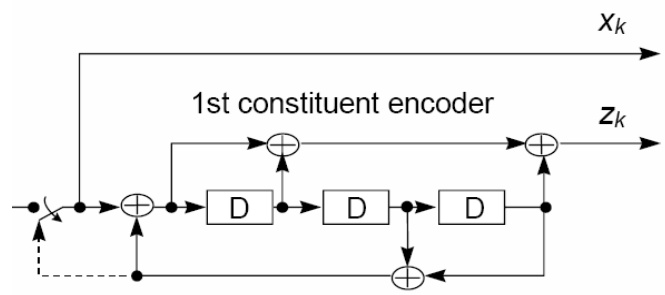

Figure 3. 1st constituent convolutional encoder of Rel.6

In [8], a sufficient condition is given for an encoder being tail-biting for any information sequence with a given block size. In the following, we prove this condition is necessary for an encoder with minimal degree (i.e. the number of states cannot be reduced).

Theorem 1 Let the matrices (A,B,C,D) be the state-space realization of a convolutional encoder with minimal degree m. This encoder is tail-biting for any information sequence of block size N≥m if and only if A^N+I_m is invertible.

Proof Let $\mathbf{u} = (u_0, \cdots, u_{N-1})$ be any information sequence of size N. Let $(s_{m-1}^{(N)}, \cdots, s_0^{(N)})^T$ be the final state of the encoding with the given information sequence, than by (EQ-1) we have

$$(s_{m-1}^{(N)}, \cdots, s_0^{(N)})^T = A^N (s_{m-1}^{(0)}, \cdots, s_0^{(0)})^T + \sum_{r=0}^{N-1} A^{N-1-r} B u_r^T$$

Thus, the encoding is tail-biting for the given sequence if and only if

$$(s_{m-1}^{(N)}, \cdots, s_0^{(N)}) = (s_{m-1}^{(0)}, \cdots, s_0^{(0)}).$$

This implies that there is a solution to the system of linear equations

$$(A^N + I_m)(s_{m-1}^{(0)}, \cdots, s_0^{(0)})^T = \sum_{r=0}^{N-1} A^{N-1-r} B u_r^T, \qquad \text{(EQ-3)}$$

where I_m is m by m binary identity matrix. On the other hand, by [6], degree m is minimal if and only if the m matrices B,AB,…A^{m-1}B are linear independent. This

implies that $\sum_{r=0}^{N-1} A^{N-1-r} Bu_r^T$ can run over entire space $\{0,1\}^m$ when N≥m with all possible input sequences of size N. This implies (EQ-3) has a solution for all possible information sequences if and only if A^N+I_m is invertible. This proves the theorem.

■

5 Tail-biting encoding is not perfect

It was stated in [1] that turbo encoder in Rel.6 [2] may not offer tail-biting for information sequence of size multiple of 7. In this section, we show that any encoder has this defect, but may with different number. In the following we will only consider recursive encoder since turbo code uses recursive convolutional encoder as its constituent encoder (see [10]).

Theorem 2 For any recursive convolution encoder of minimal degree m there exists a positive integer P such that this encoder gives no tail-biting termination for some information sequences of size tP (t>0).

Proof. Let (A,B,C,D) be the state-space realization this convolutional encoders. Since there are finite number of m x m binary matrices, there exist two positive integers, u and v, such that $A^u=A^v$. Suppose u<v. then we have $A^u(A^{(v-u)}+I_m)=0$. Let P=v-u, we have

a) Both A^u and A^P+I_m is none invertible

b) A^u invertible but $A^P+I_m=0$ (i.e. none invertible)

c) A^P+I_m invertible but $A^u=0$. This implies $G(x)=D+B(x^{-1}I_m-A)^{-1}C$ is a polynomial matrix [8]. Thus the encoder is none recursive. This gives a contradiction.

Therefore, A^P+I_m must be none invertible. Moreover, for any integer t>0, suppose

$$A^{tP}+I_m=(A^P+I_m)(A^{(t-1)P}+A^{(t-2)P}+\cdots+I_m)\cdot$$

is invertible. Then there exists and m by m matrix V such that

$$(A^P+I_m)(A^{(t-1)P}+A^{(t-2)P}+\cdots+I_m)V=I_m$$

That is to say A^P+I_m is invertible, which contradicts the previous conclusion. Therefore, $A^{tP}+I_m$ is also none invertible. Thus by Theorem 1, this recrsive encoder does not give tail-biting termination for some information sequence of size tP.

■

Based on Theorem 2 there is no need to choose other turbo code for tail-biting purpose.

6 Tail-biting on 8 states (m=3) turbo codes

In this section we investigate all possible degree 3 convolutional encoders for turbo code and try to find which one is best for tail-biting. Let us recall the definition of similarity of two m by m matrices. Two m by m matrices A_1 and A_2 are said similar if

there exists a invertible matrix S such that $A_1=SA_2S^{-1}$. It is easy to prove that code with state-space realization (A,B,C,D) and (SAS^{-1},SB, CS^{-1}) have the same encoder matrix G(x) (also see [11]). The set of all 3x3 binary matrices can be partitioned into several classes such that every class contains all similar matrices. Those classes also can be divided to 3 big categories, namely I) classes with invertible matrices, II) classes with nilpotent matrix and IV) classes with none invertible and none nilpotent. In fact, there are 14 such classes. Therefore, we only need consider 14 matrices that representative of each class.

1) Nilpotent (3 representatives)

$$A_1=\begin{bmatrix}0&0&0\\0&0&0\\0&0&0\end{bmatrix},\quad A_2=\begin{bmatrix}0&1&0\\0&0&0\\0&0&0\end{bmatrix},\quad A_3=\begin{bmatrix}0&1&0\\0&0&1\\0&0&0\end{bmatrix}$$

Obviously, these matrices are none recursive and will not be considered as a constituent encoder of turbo code

2) Non-invertible and non-nilpotent (5 representatives)

$$A_4=\begin{bmatrix}1&0&0\\0&0&0\\0&0&0\end{bmatrix},\quad A_5=\begin{bmatrix}0&1&0\\1&0&0\\0&0&0\end{bmatrix},\quad A_6=\begin{bmatrix}1&1&0\\1&0&0\\0&0&0\end{bmatrix},\quad A_7=\begin{bmatrix}1&0&0\\0&1&0\\0&0&0\end{bmatrix}$$

Encoder with these 4 state matrices will give a disconnected memory. The encoder with disconnected memory will not give the best d_2 needed by turbo codes (see [11]).

$$A_8=\begin{bmatrix}1&1&0\\0&0&1\\0&0&0\end{bmatrix}\rightarrow A_8^t=\begin{bmatrix}1&1&1\\0&0&0\\0&0&0\end{bmatrix},\ t>1\rightarrow A_8^t+I_3=\begin{bmatrix}0&1&0\\0&1&1\\0&0&1\end{bmatrix}$$

Since $A_8^3+I_3$ is non-invertible, according to Theorem 1 the encoder (A,B,C,D) either is not minimal degree or it is not tail-biting for some information sequences. For examples, a) take B=$[1\ 1\ 1]^T$, we have BA=$[0\ 1\ 0]^T$ and BA^2=$[1\ 0\ 1]^T$. Then (EQ-3) has no solution for many information sequences of size >2; b) take B=$[1\ 1\ 0]^T$, then the encoder can be reduced to a degree 2 encoder *(A',B',C',D)* with

$$A'=\begin{bmatrix}1&1\\0&1\end{bmatrix}\ \text{and}\ B'=\begin{bmatrix}1\\1\end{bmatrix}.$$

3) Invertible (6 representatives)

$A_9=I_3$ which gives a disconnected memory encoder.

$$A_{10}=\begin{bmatrix}0&0&1\\0&1&0\\1&0&0\end{bmatrix}\rightarrow A_{10}^2=I_3\ A_{11}=\begin{bmatrix}1&0&1\\0&1&0\\1&0&0\end{bmatrix}\rightarrow A_{11}^3=I_3$$

The encoders with these matrices have disconnected memory.

$$A_{12}=\begin{bmatrix}0&1&1\\1&0&1\\1&0&0\end{bmatrix}\rightarrow A_{12}^4=I_3,\ A_{12}^2+I=\begin{bmatrix}0&0&1\\0&0&1\\0&0&0\end{bmatrix},\ A_{12}^3+I=\begin{bmatrix}1&1&1\\1&1&0\\0&0&0\end{bmatrix}$$

$$A_{13}=\begin{bmatrix}0&1&0\\0&0&1\\1&1&0\end{bmatrix}\rightarrow A_{13}^{7}=I,\ A_{13}^{1}+I_3=\begin{bmatrix}1&1&0\\0&1&1\\1&1&1\end{bmatrix}$$

The turbo code of Rel.6 uses convolutional encoder with A_{13} as a state matrix. Further more we have

$$A_{13}^{2}=\begin{bmatrix}0&0&1\\1&1&0\\0&1&1\end{bmatrix},A_{13}^{2}+I_3=\begin{bmatrix}1&0&1\\1&0&0\\0&1&0\end{bmatrix},\ A_{13}^{3}=\begin{bmatrix}1&1&0\\0&1&1\\1&1&1\end{bmatrix},A_{13}^{3}+I_3=\begin{bmatrix}0&1&0\\0&0&1\\1&1&0\end{bmatrix},$$

$$A_{13}^{4}=\begin{bmatrix}0&1&1\\1&1&1\\1&0&1\end{bmatrix},A_{13}^{4}+I_3=\begin{bmatrix}1&1&1\\1&0&1\\1&0&0\end{bmatrix},\ A_{13}^{5}=\begin{bmatrix}1&1&1\\1&0&1\\1&0&0\end{bmatrix},A_{13}^{5}+I_3=\begin{bmatrix}0&1&1\\1&1&1\\1&0&1\end{bmatrix},$$

$$A_{13}^{6}=\begin{bmatrix}1&0&1\\1&0&0\\0&1&0\end{bmatrix},A_{13}^{6}+I_3=\begin{bmatrix}0&0&1\\1&1&0\\0&1&1\end{bmatrix}$$

$$A_{14}=\begin{bmatrix}0&1&0\\0&0&1\\1&0&1\end{bmatrix}\rightarrow A_{14}^{7}=I_3,\ A_{14}^{1}+I_3=\begin{bmatrix}1&1&0\\0&1&1\\1&0&0\end{bmatrix},$$

Duo-binary turbo code [12] uses A_{14} as its constituent encoder Fig.4.

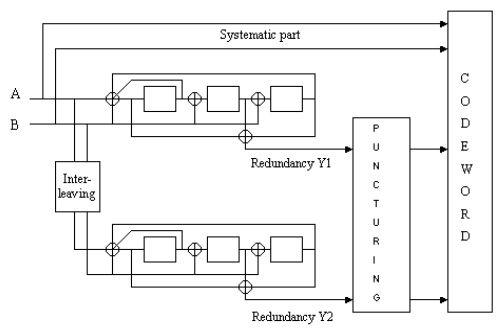

Figure 4. Duo-binary turbo code

Moreover, we have

$$A_{14}^{2}=\begin{bmatrix}0&0&1\\1&0&1\\1&1&1\end{bmatrix},A_{14}^{2}+I_3=\begin{bmatrix}1&0&1\\1&1&1\\1&1&0\end{bmatrix},\ A_{14}^{3}=\begin{bmatrix}1&0&1\\1&1&1\\1&1&0\end{bmatrix},A_{14}^{3}+I_3=\begin{bmatrix}0&0&1\\1&0&1\\1&1&1\end{bmatrix},$$

$$A_{14}^{4}=\begin{bmatrix}1&1&1\\1&1&0\\0&1&1\end{bmatrix},A_{14}^{4}+I_3=\begin{bmatrix}0&1&1\\1&0&0\\0&1&0\end{bmatrix},\ A_{14}^{5}=\begin{bmatrix}1&1&0\\0&1&1\\1&0&0\end{bmatrix},A_{14}^{5}+I_3=\begin{bmatrix}0&1&0\\0&0&1\\1&0&1\end{bmatrix},$$

$$A_{14}^{6}=\begin{bmatrix}0&1&1\\1&0&0\\0&1&0\end{bmatrix},A_{14}^{6}+I_3=\begin{bmatrix}1&1&1\\1&1&0\\0&1&1\end{bmatrix}$$

Furthermore, we can have the following proposition.

Proposition 1 Let $A=A_{13}$ or A_{14}. Then for any positive integer $n=7q+i$, $0\le i\le 6$,

$$A^n+I_m=A^i+I_m=\begin{cases}0 & i=0\\ invertible & i\neq 0\end{cases}$$

We may extend Proposition 1 to the following.

Proposition 2 Let m=2,3,4,5,6. There exists an m by m binary matrix A such that $A^{2^m-1} = I_m$. Moreover, for any such matrix A and any positive integer n=pq+i, 0≤i≤p-1, where p=2^m-1,

$$A^n + I_m = A^i + I_m = \begin{cases} 0 & i = 0 \\ invertible & i \neq 0 \end{cases}$$

7 New tail-biting termination method for arbitrary information length

Proposed tail-biting termination method for arbitrary number of information symbols

In the following we only give the method for m=3 since the most likely 3GPP LTE will adapt 8 states turbo code. For the case m=2 and m>3 the method is similar.

Let (A,B,C,D) be state space realization of the 8 states convolutional encoder with A being similar to either A_{13} or A_{14} listed in the last section.

I) Pre-compute the followings states for i=1,2,3,4,5,6

$$S_{i,1} = (A^i + I_3)^{-1}\begin{bmatrix}0\\0\\1\end{bmatrix}, S_{i,2} = (A^i + I_3)^{-1}\begin{bmatrix}0\\1\\0\end{bmatrix}, S_{i,3} = (A^i + I_3)^{-1}\begin{bmatrix}0\\1\\1\end{bmatrix},$$

$$S_{i,4} = (A^i + I_3)^{-1}\begin{bmatrix}1\\0\\0\end{bmatrix}, S_{i,5} = (A^i + I_3)^{-1}\begin{bmatrix}1\\0\\1\end{bmatrix}, S_{i,6} = (A^i + I_3)^{-1}\begin{bmatrix}1\\1\\0\end{bmatrix}, S_{i,7} = (A^i + I_3)^{-1}\begin{bmatrix}1\\1\\1\end{bmatrix}$$

II: Pre-store the above 42 index-state pairs as a look-up-table $L(i,b_{(2)}) = S_{i,b}$, where b=1,2,3,4,5,6,7 and $b_{(2)}$ is the 3 bits binary representation of b. Moreover, let L(i,0)=0 state.

III) Tail-biting encoding method for information block size = k. Let $u_0,u_1,\ldots,u_{k-1}$ be the information symbols.

1. Let m=(k mod 7). If m=0, pad one more symbol u_k=0 and let N=k+1 and M=1, otherwise let N=k and M=m.
2. With 0 state encoding information symbols $u_0, u_1,\ldots,u_{N-1}$ to find the final state S_{final} (do not store the encoded symbols). Then use Look-up table to find the initial state S_0=L(M, S_{final}).
3. Use S_0 as initial state to encode $u_0,u_1,\ldots,u_{N-1}$

This tail-biting termination can be proved by Proposition 1.

■

The new tail-biting termination method offers

1) **Flexibilily:** Tail-biting works for any size of information sequence

2) **Less overhead:** at most one dummy symbol overhead. Better than adding 2m overhead termination bits for an 2^m state convolutional code used in 3GPP Rel.6 turbo codes and other turbo codes

3) **Better performance:** all the bits are interleaved while the 6 termination bits are not interleaved.

8 Conclusion

After showing no turbo encoder can has tail-biting constituent encoder for any information sequences of arbitrary size, we proposed a tail-biting termination method for possible 3GPP LTE turbo coding. This method adds one dummy symbol to 1/7 of all possible information symbol sizes. There is no extra symbol needs to be added to the other 6/7 information sizes. This efficient method will allow tail-biting termination for arbitrary information block sizes.

9 References

[1] HighDimension Ltd, "System impact of Rel'6 turbo coding tail-bits removal," R1-062157, Tallinn, Estonia, Aug 28th – Sep 1st, 2006

[2] 3GPP TS 25.212 V6.8.0 (2006-06)

[3] A. J. Viterbi, "An Intuitive Justification and a Simplified Implementation of the MAP Decoder for Convolutional Codes," *IEEE JSAC, VOL. 16, NO. 2,* pp. 260-264 FEBRUARY 1998

[4] J. B. Anderson and S. M. Hladik, "Tailbiting MAP Decoders," *IEEE JSAC, VOL. 16, NO. 2,* FEBRUARY 1998

[5] J. L. Massey and M. K. Sain, "Codes, automata, and continuous systems: Explicit interconnections," *IEEE Trans. Automat. Contr.*, vol. AC-12, no. 6, pp. 644–650, 1967.

[6] R.J. McEliece. The Algebraic Theory of Convolutional Codes. In *Handbook of Coding Theory*, R. Brualdi, W.C. Human and V. Pless (eds.). Elsevier Science Publishers, Amsterdam, The Netherlands,1998

[7] J. Rosenthal, and E. V. York, "BCH Convolutional Codes," *IEEE Trans. on Inform. Theory, Vol.45, No.6, PP.1833-1842, Sept. 1999*

[8] C. Weiß, C. Bettstetter and S. Riedel, "Code Construction and Decoding of

Parallel Concatenated Tail-Biting Codes**,"** *IEEE Trans on Information Theory,Vol.47, No.1, pp.366-386, Jan. 2001*

[9] H. Gluesing-Luerssen and G. Schneider,"State space realizations and monomial equivalence for convolutional codes", *arXiv:cs.IT/0603049, Mar. 2006*

[10] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: turbo codes", *Proc. of IEEE ICC '93, Geneva, pp. 1064-1070,May 1993*.

[11] C. Fragouli and R. Wesel, "Turbo-Encoder Design for Symbol-Interleaved Parallel Concatenated Trellis-Coded Modulation," IEEE Trans. on Communications, Vol. 49, No.3, PP. 425-435, March 2001

[12] Duo-Binary Turbo Codes for Evolved UTRA, *3GPP TSG RAN WG1#46,R1-061973*

Contactor: BZ Shen

e-mail: bzshen@broadocm.com