TSG-RAN Working Group1 meeting #5 **_TSGR1#5(99)561_**
Cheju, KOREA, 1-4, June 1999

Agenda Item: Adhoc 4 & 5
Source: SAMSUNG Electronics Co.
Title: Comments on puncturing algorithm for Turbo codes
Document for: Discussion

---

1. Introduction

In this document, puncturing for Turbo codes is discussed. In 3GPP, the current working assumption is the 8PCCC (8 states parallel concatenated convolutional codes) Turbo codes for data services requiring BER = 10^{-6} > 0. Code puncturing is applied to match physical channel data rate and QoS for a given data service combination. Upon the previous discussion on Turbo code puncturing, systematic puncturing methods, which we define to be a puncturing scheme that does not puncture systematic (information) part of Turbo code, have been preferred and widely used, since it appears to show better performance among all possible methods. In order to achieve the optimum performance with this scheme while applying certain degree of puncturing, some specific rules and puncturing criteria are required and currently have been recommended: not puncturing systematic bits, and puncturing in a balanced way the parity bits in Turbo codes (PCCC).

However, it should further be verified whether such a puncturing really leads to a noticeable performance gain. Thus, in this document, we show that systematic puncturing ensures the maximum coding performance achieved by Turbo codes. For this purpose, a weight spectrum of recursive convolutional codes (RSC) as a component encoder is investigated.

2. Optimal puncturing of RSC for K=4

Among all the studies available for puncturing of Turbo codes, the following rule are applied:

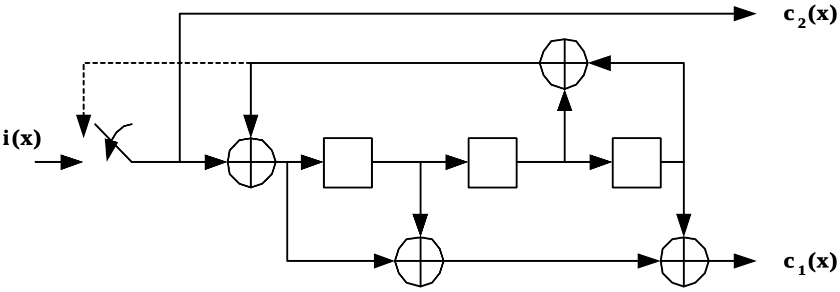
- Systematic code symbols should not be punctured.
- Puncturing of parity symbols should be distributed evenly.

In order to verify these recommendations, we analyzed the weight spectrum of RSC codes of R=1/2 and K=4. In general, iterative decoding algorithm, where a component decoder for R=1/2, K=4 RSC code is applied sequentially in order to decode 8PCCC Turbo codes, is widely used. Therefore, it is reasonable to consider R=1/2, K=4 component encoder to verify puncturing scheme for Turbo codes.

Fig.1 shows an encoder of R=1/2, K=4 RSC code for 8PCCC Turbo code, where the feed back polynomial is $g_2(x)=1+x^2+x^3$ and the parity generator polynomial is $g_1(x)=1+x+x^3$, respectively.

Weight spectrum is given in Table 1, where the d_{free} is the minimum free distance. a_d is the number of incorrect paths of Hamming weight d for $d \geq d_{free}$ that diverge from the correct path and remerge with it sometime later. c_d is the total number of error bits produced by the incorrect paths. In Table 1, weight spectrums of K=4 RSC mother code and some possible punctured RSC codes with puncturing period of 2 are given. Note that the minimum free distance of R=2/3 punctured RSC code is 4. Table 1 shows that systematic puncturing is superior to nonsystematic puncturing in terms of weight spectrum of information symbols. Hamming weight of information symbols of nonsystematic punctured RSC code is larger than that of systematic punctured RSC codes.

According to these results in Table1, it is shown that information symbols of RSC codes should not be punctured in order to obtain low information bit error rate for a given code rate greater than the code rate of the mother code.



K=4 RSC encoder for PCCC

Fig. 1. Block diagram of K=4 recursive systematic convolutional code for Turbo codes.

TABLE 1. WEIGHT SPECTRA OF K=4 RSC CODES WITH CODE RATE OF R-1/2 and 2/3.

	Puncturing matrix P	d_{free}	Weight spectra of information symbols And the number of incorrect paths: a_d, c_d, $d=d_{free}$, $d_{free}+1$, $d_{free}+2$, ……	
Original code R-1/2	None	6	a_d	4,0,20,0,98,0,482,0,2370,0,11654,0,57306,0,281790, …
			c_d	12,0,80,0,490,0,2892,0,16590,0,93232,0,515754,0,2817900,…
Nonsystematic Punctured R-2/3 RSC	10 11	4	a_d	3,9,31,103,335,1122,3707,12298,40775,135144, …
			c_d	10,44,182,704,2687,10151,37539,137736,500245,1803097,…
	01 11	4	a_d	3,9,31,103,335,1122,3707,12298,40775,135144, …
			c_d	10,44,182,704,2687,10151,37539,137736,500245,1803097,…
Systematic Punctured R-2/3 RSC	11 10	4	a_d	3,9,31,103,335,1122,3707,12298,40775,135144, …
			c_d	10,28,130,496,1815,6843,24987,90972,328277,1176261,…
	11 01	4	a_d	3,9,31,103,335,1122,3707,12298,40775,135144, …
		4	c_d	10,28,130,496,1815,6843,24987,90972,328277,1176261,…

- The feed back polynomial is $g_2(x)=1+x^2+x^3$ and the parity generator polynomial is $g_1(x)=1+x+x^3$, respectively.

3. Conclusion

In this document, Samsung verified that the noticeable performance gain of the recommendations in terms of systematic puncturing for 8PCCC Turbo codes is related to the weight spectrum of systematic punctured RSC codes. Therefore, to obtain optimal performance of 8PCCC turbo codes, rate matching algorithm should not puncture information symbols.

4. References

[1] "Puncturing algorithm for Turbo code", LGIC,TSGR1#4(99)338.
[2] "Optimized puncturing scheme for Turbo coding", Fujitsu, TSGR1#4(99)388.
[3] "Additional simulation results of the optimized puncturing scheme for Turbo coding", Fujitsu, TSGR1#4(99)xxx.
[4] "Optimized rate matching after interleaving-rev2", Siemens, TSGR1#3(99)203v2.
[5] "Text proposal for S1.12", HNS, TSGW1#2(99)419.

Contact inform:
Mgkim@metro.telecom.samsung.co.kr