Application of bound: In this section we present results of the derived punctured bound for different turbo code rates derived from the original rate 1/3 turbo code. We assume that the channel has additive white Gaussian noise (AWGN). As a representative example, binary phase shift key (BPSK) modulation is used. We apply the bound to a turbo code with two identically constituted encoders with generator functions $(5/7)_8$. We use the algorithm in [3] to calculate the WEF of the constitutent codes.

Fig. 1 shows the bound for a frame length of 500 bits at code rates of 1/3, 2/5, 1/2, and 2/3. As expected, the punctured bounds diverge at signal-to-noise ratios larger than those that occur at a rate of 1/3. The abrupt transition of the bound occurs when the signal-to-noise ratio, E_b/N_0, drops below the threshold determined by the computation cutoff rate R_0, i.e. when $E_b/N_0 < -(1/r)\ln(2^{1-r} - 1)$ for a code rate r [3].

In Fig. 1, the abrupt change occurs at 2.03, 2.2, 2.5 and 3.1 dB, for rates of 1/3, 2/5, 1/2, and 2/3, respectively. Also, as shown in the rate 1/3 code bound calculation, for the punctured bound at higher signal-to-noise ratios, the evaluation of the bound requires only a few terms in the summation. The error floor (the low slope region of the performance curve where the error rate decreases very slowly with increasing signal-to-noise ratio) still exists with the punctured bound.

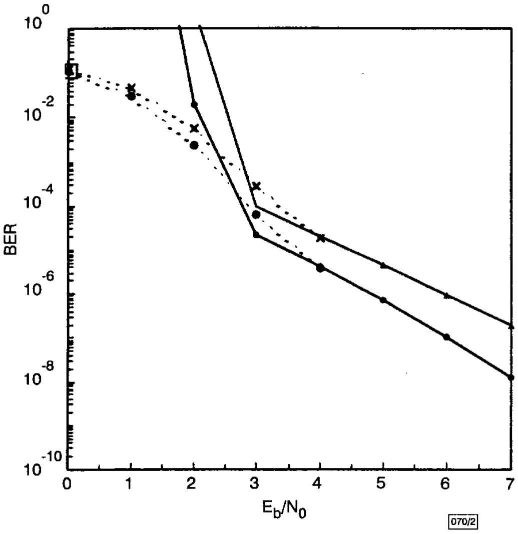

Fig. 2 *Simulated and analytical bound for punctured turbo code*

—●— r = 2/5, bound
– –●– – r = 2/5, simulated
—▲— r = 1/2, bound
– –×– – r = 1/2, simulated

In Fig. 2, a simulated punctured turbo code with a frame length of 192 bits is compared with the analytical punctured bound at rates 2/5 and 1/2. At higher signal-to-noise ratios (greater than R_0), the bound accurately predicts the turbo decoder performance. At signal-to-noise ratios less than R_0, simulation is the only way to predict the performance of turbo codes due to the divergence in the performance of the analytical bound in this region.

Conclusions: The introduction of the hypergeometric puncturing device makes the derivation of the analytical bound of the punctured bound of turbo codes tractable. The hypergeometric puncturing device allows for averaging over all possible puncturing positions. The performance bound has been compared with the simulation results, and the comparison shows that the two bounds, obtained both through analytical and simulation means, are identical at higher signal-to-noise ratios, but the analytical bound still diverges at signal-to-noise ratios lower than the cutoff rate R_0. This bound can also be extended for use with punctured turbo code modulations and for designing punctured turbo codes.

Acknowledgement: The views expressed in this article are those of the author and do not reflect the official policy of the United States Air Force, Department of Defense or the U.S. Government.

21 October 1999
Electronics Letters Online No: 19991464
DOI: 10.1049/el:19991464

M.M. Salah, R.A. Raines, M.A. Temple and T.G. Bailey (*Department of Electrical and Computer Engineering, Air Force Institute of Technology, USA*)

References

1 BERROU, C., GLAVIEUX, A., and THITIMAJSHIMA, P.: 'Near Shannon limit error-correcting coding and decoding: TURBO-CODES'. Proc. 1993 IEEE Int. Conf. On Communication, Geneva, Switzerland, 1993, pp. 1064–1070

2 BENEDETTO, S., and MONTORSI, G.: 'Unveiling turbo codes: some results on parallel concatenated coding schemes', *IEEE Trans. Inf. Theory*, 1996, **42**, (2), pp. 409–428

3 DIVSALAR, D., DOLINAR, S., POLLARA, F., and McELIECE, R.J.: 'Transfer function bounds on the performance of turbo codes'. TDA Progr. Rep. 42–122, Jet Propulsion Lab., Pasadena, CA, August 15, 1995, pp. 44–55

4 HOGG, R.V., and CRAIG, A.T.: 'Introduction to mathematical statistics' (Englewood Cliffs, NJ: Prentice Hall, 1995), p. 517

Class of multilevel run-length limited trellis codes

Hung-Hua Tang and Mao-Chao Lin

The authors have used a multilevel coding technique to construct run-length limited (RLL) trellis codes with error-correcting capabilities. The constructed codes have better performance than the RLL trellis codes previously proposed, in terms of error-correcting capabilities, coding rates and decoding complexities.

Introduction: In magnetic recording systems, run-length constraints on data are usually required to reduce the effect of intersymbol interference and to support bit synchronisation. These constrained data sequences are called runlength limited (RLL) sequences or (d, k) sequences [1], where d is the minimum run length of zeros between 1s and k is the maximum run length of zeros between 1s. A (d, k) code or RLL code is a set of (d, k) sequences. In [1], $C^n(d, k)$ is used to denote the set of all binary (d, k) sequences of length n. A maximal subset from $C^n(d, k)$ such that any two codewords in this subset can be concatenated without violating the (d, k) constraint is called a maximal concatenatable subset of $C^n(d, k)$ denoted as $C^n_{max}(d, k)$ [2].

In [1], a class of RLL trellis codes (the Lee-Wolf code) was proposed which have a single error-correcting capability. A Lee-Wolf code is constructed by partitioning $C^n_{max}(d, k)$ into two subsets, C_e and C_o, and using a rate $(m-1)/m$ convolutional code with 2^{m-1} states to select RLL n-tuples in C_e and C_o, where C_e and C_o are the sets of even-weight and odd-weight n-tuples in $C^n_{max}(d, k)$, respectively. It is required that $|C_e| \geq 2^{m-1}$ and $|C_o| \geq 2^{m-1}$.

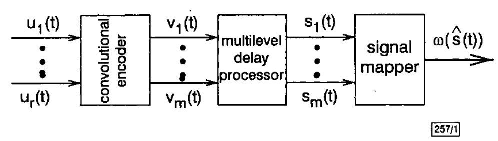

Fig. 1 *Encoding configuration*

In [3], a multilevel coding technique was used to construct trellis coded modulation systems. Consider a signal set Ω that consists of 2^m signal points. Each signal point in Ω is marked by $\omega(\tilde{s})$, where $\tilde{s} = (s_1, s_2, ..., s_m)$, $s_i \in \{0, 1\}$, $i = 1, 2, ..., m$. The encoding of the system in [3] is illustrated in Fig. 1. At time t, an r-bit message $\tilde{u}(t)$ is encoded by a rate r/m convolutional code C resulting in an m-bit code branch $\tilde{v}(t) = (v_1(t), v_2(t), ..., v_m(t))$. The output of the multilevel delay processor is $\tilde{s}(t) = (s_1(t), s_2(t), ..., s_m(t))$. The output of the signal mapper is $\omega(\tilde{s}(t)) \in \Omega$, where $s_l(t) = v_l(t-(m-l)\lambda)$, $l = 1, ..., m$, and λ is a constant. If the mapping between the m-tuple $\tilde{s}$ and the signal point $\omega(\tilde{s})$ is appropriately designed, a large free distance can be achieved for a signal set Ω such as 8PSK or $\{0, 1\}^4$ [3, 4].

In this Letter, we propose to apply the multilevel coding technique of [3] to the design of RLL codes. Since $|C^n_{max}(d, k)|$ is not a power of 2 in general, one of the problems is to choose a subset of $C^n_{max}(d, k)$ that consists of 2^m RLL n-tuples to be the signal set Ω. The design of ω is also a problem. We show that there is simple way to design Ω and ω such that RLL trellis codes are yielded which have a better performance than Lee-Wolf codes. We also show that in some cases, further improvements can be made.

Preliminaries: In this Section, we describe the properties of the trellis code T shown in Fig. 1 in a way similar to that given in [4]. Consider a signal set Ω which consists of 2^m signal points, where each signal point is a binary n-tuple. The Hamming distance between two signal points $\omega(\tilde{s})$ and $\omega(\tilde{s}')$ will be denoted by $\Delta(\omega(\tilde{s}), \omega(\tilde{s}'))$. Let $\hat{s} = (s_1, s_2, ..., s_m)$ and $\hat{s}' = (s_1', s_2', ..., s_m')$ and define

$$\Delta_j = \begin{cases} \min\limits_{s_j \neq s_j'} \{\Delta(\omega(\hat{s}), \omega(\hat{s}')) : \omega(\hat{s}), \omega(\hat{s}') \in \Omega\} & \text{if } j = 1 \\ \min\limits_{s_j \neq s_j'} \{\Delta(\omega(\hat{s}), \omega(\hat{s}')) : \omega(\hat{s}), \omega(\hat{s}') \in \Omega, \\ \qquad s_i = s_i' \text{ for } 1 \leq i < j\} & \text{if } 1 < j \leq m \end{cases} \tag{1}$$

We then have an m-level distance profile $\Delta_1, ..., \Delta_m\}$. The free distance of T is lower bounded [3, 4] by

$$\Delta_{LB}(\lambda) = \min_{\substack{\bar{v} \in C \\ \hat{v}(0) \neq 0}} \sum_{t=0}^{\lambda-1} \sum_{j=1}^{m} v_j(t) \Delta_j \tag{2}$$

where $\bar{v} = \{..., \hat{v}(0), \hat{v}(1), ...\}$ and $\hat{v}(t) = (v_1(t), ..., v_m(t))$. The bound $\Delta_{LB}(\lambda)$ is an increasing function of λ and will become a constant value Δ_{free} when λ exceeds a threshold number, λ_{min}. Suboptimal decoding for T can be carried out by using the trellis of C [3, 4]. Suboptimal decoding can fully utilise the error-correcting capability guaranteed by Δ_{free}. Hence, the decoding complexity of T is only slightly higher than that of C.

New RLL trellis codes: We now use the coding scheme described in the preceding Section to construct RLL codes with good error-correcting capabilities. It is difficult to find a general rule for choosing Ω and ω that can yield a trellis code T with the largest free distance for a given (d, k) constraint and an arbitrary n. We now choose Ω and ω by following the partition used in [1]. That is, we choose $\Omega = \Omega_{s_1=0} \cup \Omega_{s_1=1}$, where $\Omega_{s_1=0} = \{\omega(s_1 = 0, s_2, ..., s_m) | s_2, ..., s_m \in \{0,1\}\} \subseteq C_e$ and $\Omega_{s_1=1} = \{\omega(s_1=1, s_2, ..., s_m) | s_2, ..., s_m \in \{0, 1\}\} \subseteq C_o$. This gives a distance profile of $\{\Delta_1 = 1, \Delta_2 = 2, \Delta_3 \geq 2, ..., \Delta_m \geq 2\}$.

Using this distance profile, RLL codes are found which are listed in Table 1, where ν is the total number of memory bits in the encoder of C. Compared with Lee-Wolf codes, the codes in Table 1 have a higher error-correcting capability, an identical or higher coding rate and a similar or lower decoding complexity. For example, consider $(d, k) = (1,7)$. For $n = 6$, there is a four-state rate 1/3 Lee-Wolf code with free distance 3, while as can be seen in Table 1, there is a four-state rate 1/3 code with free distance 5. For $n = 12$, there is a 64-state rate 1/2 Lee-Wolf code with free distance 3, while in Table 1, there is an eight-state rate 1/2 code with free distance 5.

Table 1: RLL multilevel trellis codes

ν	Δ_{free}	λ_{min}	Generator	n/rate $(d,k) = (1,3)$	n/rate $(d,k) = (2,7)$	n/rate $(d,k) = (1,7)$
2	5	5	$\begin{pmatrix} 1 & 1 & 1 \\ 5 & 7 & 0 \end{pmatrix}$	9/0.222	10/0.200	6/0.333
2	5	6	$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 3 & 0 & 1 & 2 \\ 2 & 1 & 3 & 2 \end{pmatrix}$	10/0.300	11/0.273	8/0.375
3	5	10	$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 3 & 2 & 3 \\ 0 & 2 & 1 & 0 & 2 & 2 & 3 \\ 2 & 1 & 2 & 0 & 3 & 3 & 1 \end{pmatrix}$	15/0.400	16/0.375	12/0.500

In addition to the free distance, the error coefficient (average number of nearest code sequences) N is also a parameter that can be used to evaluate the error-correcting capability of a trellis code. For each code listed in Table 1, its error coefficient N (based on the suboptimal decoding) is not unique and is dependent on the specific choice of Ω and ω. Hence, in Table 1, the error coefficient of each code is not listed. Note that the choice of $\Omega = \Omega_{s_1=0} \cup \Omega_{s_1=1}$ as described at the beginning of this Section is not unique. Consider $(d, k) = (1,7)$ and $n = 6$. The associated code in Table 1 has $N = 45/32$ if Ω and ω are chosen as in Table IV of [1] and has $N = 9/16$ if we map (s_1, s_2, s_3) = (000), (001), (010), (011), (100), (101), (110) and (111), respectively, to $\omega(s_1, s_2, s_3)$ = (100010), (010100), (001010), (100100), (010000), (001000), (000100) and (101010). For comparison, the associated Lee-Wolf code $n = 6$ has $N = 55/64$. For a recording channel that only requires a low error-correcting capability, the free distance is the dominant factor in determining its error performance while the error coefficient N is insignificant unless N is very large. Note that N = 55/64, 45/32 and 9/16 are all small and do not deviate much from 1. Hence, the error-correcting capabilities of these RLL trellis codes are dominated by their free distances.

The codes in Table 1 can be further improved by choosing more appropriate Ω and ω. We use two examples as an illustration.

(i) *Example 1*: For $(d, k) = (1,7)$ and $n = 6$, we map (s_1,s_2,s_3) = (000), (001), (010), (011), (100), (101), (110) and (111), respectively, to $\omega(s_1, s_2, s_3)$ = (101010), (010100), (100100), (001000), (101000), (010000), (000100) and (100010). The distance profile is (1,2,3). Using the generator given in row 1 of Table 1, we have a four-state, rate 1/3 code with free distance of 6 and $N = 9/8$.

(ii) *Example 2*: For $(d,k) = (1,3)$ and $n = 8$, we map (s_1, s_2, s_3) = (000), (001), (010), (011), (100), (101), (110) and (111), respectively, to $\omega(s_1, s_2, s_3)$ = (10100010), (01010100), (10010100), (10001010), (10100100), (01010010), (10010010) and (01001010). The distance profile is (2,2,4). Using the same generator as given in example 1, we have a four-state, rate 1/4 code with free distance of 8 and N = 7/8.

Clearly, these two examples are better than the counterparts in Table 1. As a summary, we conclude that by using the multilevel coding technique proposed in [3] with a proper choice of Ω and ω, we can construct RLL trellis codes with improved error-correcting capabilities.

10 September 1999
Electronics Letters Online No: 19991485
DOI: 10.1049/el:19991485

Hung-Hua Tang and Mao-Chao Lin (*Department of Electrical Engineering, Rm 323, National Taiwan University, Taipei 106, Taiwan, Republic of China*)

E-mail: mclin@cc.ee.ntu.edu.tw

References

1 LEE, P., and WOLF, J.K.: 'A general error-correcting code construction for run-length limited binary channels', *IEEE Trans. Inf. Theory*, 1989, **35**, (6), pp. 1330–1335
2 LEE, P.: 'Combined error-correcting/modulation recording codes'. Ph.D. dissertation, University of California, San Diego, 1988
3 HELLSTERN, G.: 'Coded modulation with feedback decoding trellis codes'. IEEE Conf. Communication, 1993, pp. 1071–1075
4 WANG, J.-Y., and LIN, M.-C.: 'On constructing trellis codes with large free distances and low decoding complexities', *IEEE Trans. Commun.*, 1997, **45**, (9), pp. 1017–1020