3 BE'ERY, Y., and SNYDERS, J.: 'Optimal soft decision block decoders based on fast Hadamard transformation', *IEEE Trans. Inf. Theory*, 1986, **32**, (3), pp. 355 364

4 VARDY, A., and BE'ERY, Y.: 'Maximum-likelihood soft decision decoding of BCH codes', *IEEE Trans. Inf. Theory*, 1994, **40**, (3), pp. 546 554

5 FORNEY, G.D.: 'Coset codes II: binary lattices and related codes', *IEEE Trans. Inf. Theory*, 1988, **34**, pp. 1152 1187

Construction of simple runlength-limited codes

G. Khachatrian and K.A.S. Immink

The authors describe a new technique for constructing fixed-length (d, k) runlength-limited block codes. The new codes are very close to block-decodable codes, as decoding of the retrieved sequence can be accomplished by observing (part of) the received codeword plus a very small part of the previous codeword. The basic idea of the new construction is to uniquely represent each source word by a (d, k) sequence with specific predefined properties, and to construct a bridge of β, $1 \leq \beta < d$, merging bits between every pair of adjacent words. The new constructions have the virtue that only one look-up table is required for encoding and decoding.

Introduction: Coding methods based on (d, k)-constrained sequences have been predominant in magnetic and optical disks or tapes. A binary sequence is said to be (d, k) constrained if the number of 'zeros' between any pair of consecutive 'ones' is at least d and at most k, $k > d$. Properties and applications of (d, k)-constrained sequences, or runlength-limited (RLL) sequences as they are often called, are surveyed in [1]. In [2 – 5], authors present (d, k) codes compiled from codewords of fixed length that can be decoded without the knowledge of preceding or succeeding codewords. Codes with this property, that is, codes that can be decoded by observing single codewords will be called block(-decodable) codes.

The new technique presented in this Letter is broadly similar to Beenker and Immink's 'Constructions 1 and 2' [3]. An essential, and very useful, property of these constructions, which they have in common with the new construction, is that source words have a one-to-one relationship with finite-length (d) sequences. After translating source words into (d) sequence using a look-up table, the (d) sequences are multiplexed with β bits, called merging bits, which are needed to preserve the predefined runlength constraint.

New construction: The construction is based on the representation of source words by (d) sequences of length n, which are cascaded using β, $d > 1$, merging bits. Choose the parameters d, $d \geq 2$ and n, and let the number of merging bits be $\beta \geq 1$. The basic idea is that one of the merging bits can be set to 'one' if the d constraint would be violated during the cascading of two consecutive words. Let s be the number of trailing 'zeros' of the current (d) sequence, and let t be the number of leading 'zeros' of the upcoming (d) sequence. There is no violation of the d constraint if $t + \beta + s \geq d$. Then the merging bits are set to 'zero' and the n-bit codeword plus the β merging bits are transmitted. If a d constraint violation occur. i.e. $t + \beta + s < d$, we will perform a special operation, which will be described shortly. The number of combinations, where a violation occurs is simply

$$\frac{(d - \beta + 1)(d - \beta)}{2} \qquad d \leq 2, \ \beta \leq 1 \tag{1}$$

A violation of the d constraint can be circumvented by executing an operation involving the merging bits and the two 'ones' that trespass the d constraint. The two trespassing 'ones' are set to 'zero' and a judiciously chosen merging bit is set to 'one'. Let the positions of the merging bits be denoted by $a_1, \ldots, a_\beta$. If, for example, $s = t = 0$, we set the merging bit at position a_1 equal to 'one' and set the 'ones' at the beginning and end of the adjoining (d) sequences to 'zero'. If $(s = 1, t = 0)$, the merging bit at position a_2 is set to 'zero' etc. Clearly, if

$$\frac{(d - \beta + 1)(d - \beta)}{2} \leq \beta \tag{2}$$

it is possible to uniquely denote the trespassing combinations by the β-tuple $0^i10^{\beta-i-1}$, $0 \leq i \leq \beta-1$, where 0^i denotes a string of i consecutive 'zeros'. From eqn. 2 we can easily compute the value of the number of merging bits, β, as a function of d. Table 1 gives the results.

The possible execution of one of the β distinct merging operations can be uniquely decoded by observing the merging bits, and an action, at the receiver's end, to restore the original (d) sequences is easily performed.

Table 1: Number of required merging bits β against d

d	β
2	1
5	3
9	6
14	10
20	15

Conclusions: We have developed a new construction for systematically designing (d, k) codes. The translation of source words into codewords and vice versa can be accomplished with a single look-up table. The new codes are very 'close' to block-decodable codes, as decoding can be accomplished by observing (part of) the received codeword plus a small part of the previous codeword. As a result, minor error propagation may occur during decoding.

9 November 1998

Electronics Letters Online No: 19990107

G. Khachatrian (*Institute for Problems of Informatics and Automation, Armenian National Academy of Sciences, P. Sevakstr. 1, 375044 Yerevan, Armenia*)

K.A.S. Immink (*Institute for Experimental Mathematics, Ellernstrasse 29, D-45326 Essen, Germany*)

References

1 IMMINK, K.A.S.: 'Coding techniques for digital recorders' (Prentice-Hall, 1991)

2 TANG, D.T., and BAHL, L.R.: 'Block codes for a class of constrained noiseless channels', *Inf. Control*, 1970, **17**, pp. 436–461

3 BEENKER, G.F.M., and IMMINK, K.A.S.: 'A generalized method for encoding and decoding runlength-limited binary sequences', *IEEE Trans. Inf. Theory*, 1983, **IT-29**, (5), pp. 751–754

4 WEBER, J.H., and ABDEL-GHAFFAR, K.A.S.: 'Cascading runlength-limited sequences', *IEEE Trans. Inf. Theory*, 1993, **IT-39**, (6), pp. 1976 1984

5 GU, J., and FUJA, T.: 'A new approach to constructing optimal block codes for runlength-limited channels', *IEEE Trans. Inf. Theory*, 1993, **IT-40**, (3), pp. 774–785

Single trellis decoding of Reed-Solomon and convolutional concatenated code

Sung Ho Ahn and Dong Ku Kim

By using the trellis approaches suggested by Wolf, block codes can be decoded by trellis decoding methods. The authors apply these methods to RS-convolutional concatenated codes and propose a decoding method using a single combined trellis structure comprising a convolutional code and RS code to easily obtain the soft decision decoding gain. The proposed decoding algorithm eliminates the need for a symbol interleaver, thus removing the time delay due to the symbol interleaver. The performance of the soft decision single trellis decoding of RS-convolutional concatenated codes over a fading channel is evaluated by simulation.

Introduction: For future communication systems such as IMT-2000, the RS-convolutional concatenated code has been proposed for high-speed data transmission. The conventional decoding method for this code involves Viterbi decoding for convolutional codes and algebraic hard-decision decoding for RS codes. In the proposed decoding algorithm, the soft-decision trellis decoding method for block codes is applied to RS-convolutional concatenated codes so that the proposed algorithm is based on the whole trellis structure of the convolutional and RS code, instead of the conventional method of using two separate decoders, a Viterbi decoder and algebraic RS hard-decision decoder.

Trellis diagram of cyclic code over GF(q): For an (n, k) cyclic code of generator polynomial $g(x) = g_0 + g_1x + ... + g_rx^r$, where $r = n - k$ and $g_i \in GF(q)$, a method for forming the trellis involving associating the vertices of trellis with the $q^{(n-k)}$ states of the $(n-k)$ stage shift register of encoder. The trellis diagram of this code can be built by tracing the possible states of the shift registers for all possible inputs. Since there are r shift registers and each shift register can contain at most q different elements, there will be at most q^r states in the trellis at any depth, where the depth is the time index of the trellis diagram. The number of states in the trellis at depth j is

$$= \begin{cases} q^j & j = 1, 2, ..., r-1 \\ q^r & j = r, r+1, ..., n-r \\ q^{n-j} & j = n-r+1, ..., n \end{cases}$$

The trellis is repetitive for $j = r + 1, ..., n - r$ [1].

Single trellis diagram of serial concatenated code: An RS-convolutional concatenated code is composed of a convolutional inner code and Reed-Solomon outer code. In the conventional decoding method, the RS code corrects burst-decoding errors at the output of a Viterbi decoder. Thus, the Viterbi decoder can obtain the soft decision decoding (SDD) gain easily, but an RS decoder cannot. In the proposed algorithm, the trellis structure of the RS code is combined with the Viterbi decoder to obtain the SDD gain of the RS code as well as the convolutional code. Fig. 1 shows the block diagram of the proposed single trellis decoding algorithm.

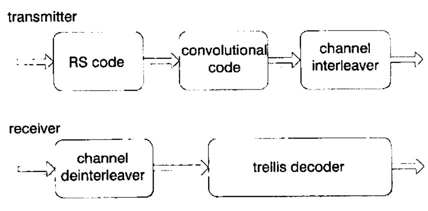

Fig. 1 *Block diagram of single trellis decoding algorithm with channel interleaver*

Since RS codes correct the decoding errors at the output of a Viterbi decoder in the conventional decoding method, a symbol interleaver between the RS code and convolutional code is required to randomise decoding errors of the Viterbi decoder. Meanwhile, the proposed algorithm makes the outer decoder and inner decoder work as a single decoder and allows the RS code to correct channel errors directly. Thus a symbol interleaver is not required. Fig. 2 shows the encoder structure of a concatenated code.

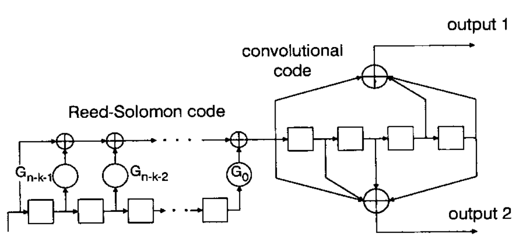

Fig. 2 *Single structure of concatenated encoder*

In Fig. 2, an (N, K) RS encoder, where $N = 2^m-1$, and an (n, k) convolutional encoder are considered as a single encoder. The states of an RS encoder and convolutional encoder are shifted simultaneously for input symbols. Once an input symbol is shifted to this encoder, it requires a 1-time transition of states of the RS encoder, and an RS encoder output symbol is transformed to m-bits simultaneously. These bits then go into the convolutional encoder. Thus m-time shifts in the states of the convolutional code occur for each input symbol. The trellis diagram of this code is built by tracing all the possible states of this encoder for all possible input symbols.

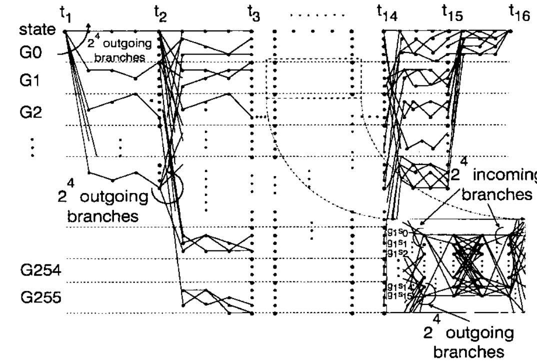

Fig. 3 *Single trellis diagram of concatenated code ((15, 13) RS-(2, 1, 5) convolutional code)*

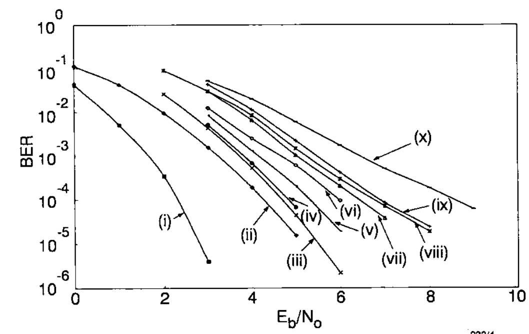

Fig. 4 *Performance of single trellis decoding and conventional decoding on Rayleigh fading channels*

(i) single trellis (AWGN)
(ii) conventional (AWGN)
(iii) single trellis (f_dT_c = 1/3)
(iv) single trellis no channel interleaver (f_dT_c = 1/3)
(v) single trellis (f_dT_c = 1/7)
(vi) single trellis no channel interleaver (f_dT_c = 1/7)
(vii) conventional (f_dT_c = 1/3)
(viii) conventional no channel interleaver (f_dT_c = 1/3)
(ix) conventional (f_dT_c = 1/7)
(x) conventional no channel interleaver (f_dT_c = 1/7)

Fig. 3 shows the single trellis diagram of a concatenated code for the (15, 13) RS code and (2, 1, 5) convolutional code. We select the (15, 13) RS code and (2, 1, 5) convolutional code as an example of moderate complexity. Each path is labelled with codewords generated by an RS code and convolutional code for input symbols, where the path is a connection of states from depth 1 to N. The states of the trellis diagram are denoted by g_is_j in which g_i is the state of the RS code and s_j is the state of the convolutional code. All states are divided into subgroups G_0, G_1, ..., G_{254} , G_{255}, where $255 = (2^4)^2$ and $G_i = \{g_is_j, j = 0, ..., 15\}$. Therefore, every state of the single trellis structure can be uniquely identified. Each subgroup shifts to another subgroup for one input symbol, and the RS output symbol is mapped to the corresponding four bits. During one state transition of the RS code, the state transition of a convolutional code occurs four times within the same subgroup. The decoding process determines the surviving paths by comparing the metrics of all eight output bits, where $8 = 4 \times (2/1)$, for an input symbol. The single trellis decoding method requires a high complexity, however. Thus practical approaches to reduce complexity are currently under investigation.

Simulation results: In Fig. 4, the single trellis decoding performance of a (15, 13) RS code and (2, 1, 5) convolutional code, obtained through simulation over an AWGN channel and Rayleigh fading channel, is shown. For the proposed system, 15-output symbols of an RS code are converted to 60 bits and four tail bits are added, and 1/2 convolutional encoder generates 128 coded bits. The size of the channel interleaver at the output outer code is set to 128 bits, which is the length of a codeword. In a conventional system, the encoding process is the same as the proposed one except that there is a symbol interleaver between the RS and convolutional code. The symbol interleaver stores 15 consecutive RS codewords. Two conditions for the fading rate over a Rayleigh fading channel are considered; $f_dT_c = 1/3$, and $f_dT_c = 1/7$, where f_d is the Doppler shift and T_c is the coded bit duration. The effect of the presence or absence of a channel interleaver is also considered. Curves (i) and (ii) show that single trellis decoding achieves a gain of ~2dB at a BER of 10^{-4} over the conventional decoding for an AWGN channel.

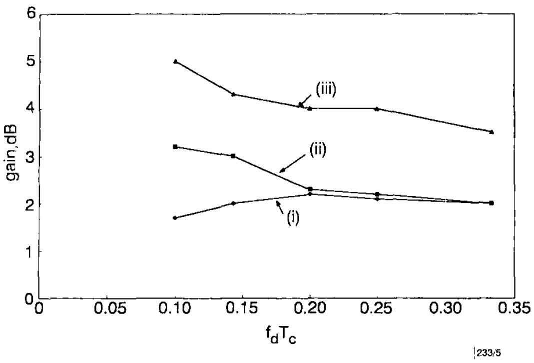

Fig. 5 *Gain of single trellis decoding over conventional decoding method on Rayleigh fading channels*

(i) proposed and conventional system with channel interleavers
(ii) both systems without channel interleavers
(iii) both systems without channel interleavers and symbol interleaver

Comparisons between (iii) and (vii), and between (v) and (ix) show that single trellis decoding gives a 2dB gain in a Rayleigh fading channel, when both decoding methods have a channel interleaver. When channel interleavers are absent for both codes, the proposed decoding method ((iv) and (vi)) obtains a gain of 2.5 and 3dB over the conventional decoding method ((viii) and (x)) at $f_dT_c = 1/3$, and $f_dT_c = 1/7$, respectively. Fig. 5 represents the gains of the proposed method over the conventional method against f_dT_c for the three different cases at 10^{-4} BER. Curve (i) shows that the gain remains constant regardless of the value of f_dT_c. Curve (ii) represents the case where both the proposed and conventional systems are without channel interleavers, and (iii) represents the case where both systems are without channel interleavers and the conventional system is without a symbol interleaver. Curves (ii) and (iii) show that gain increase as f_dT_c decreases. This shows that the proposed scheme is more robust in burst error channels than the conventional method.

Conclusion: The decoding method of an RS-convolutional concatenated code based on a single trellis is proposed. The symbol interleaver required between a convolutional code and an RS code in the conventional method is removed. The simulation of a (15, 13) RS - (2, 1, 5) convolutional concatenated code shows that the proposed decoding method gives a 2–5dB gain over the conventional decoding method on AWGN and Rayleigh fading channels.

Acknowledgments: This research is financially supported by Institute of Information Technology Assessment (IITA), project C1-98-0274-00.

28 October 1998
Electronics Letters Online No: 19990009

Sung Ho Ahn and Dong Ku Kim (*Radio Communications Engineering Department, Yonsei University, 134 Shinchon-Dong, Seodaemun-Gu, Seoul 120-749, Korea*)

References

1 WOLF, J.K.: 'Efficient maximum likelihood decoding of linear block codes using a trellis', *IEEE Inf. Theory*, 1978, **IT-24**, pp. 76–80

Soft-input soft-output algorithms for recursive convolutional codes

Xiao Ma, Baoming Bai and Ximmei Wang

A new method of implementing soft-input soft-output algorithms for recursive systematic convolutional codes is proposed. The new implementation is based on the equivalent non-systematic codes and has lower complexity.

Introduction: Since turbo codes [1] were proposed by Berrou *et al.* in 1993, parallel concatenated codes (PCCs) and serial concatenated codes (SCCs) have given rise to great interest in the coding community. Based on the concept of a uniform interleaver and using the analytical technique of union bound, Benedetto [2, 3] proved that recursive systematic convolutional (RSC) codes work properly as constituent codes (CCs) of PCCs and as the inner codes of SCCs. In both cases, the soft-input soft-output (SISO) decoder is necessary, which accepts as inputs the probability distributions of the information and code symbols labelling the edges of the code trellis, and forms as outputs an update of these probability distributions based on the code constraints. Some SISO algorithms have been investigated (see the references listed here and the references therein). In this Letter, we propose a new method of implementing these algorithms, based on the equivalent non-systematic (NSC) codes. We also show that the computation complexity can be reduced by means of this new implementation.

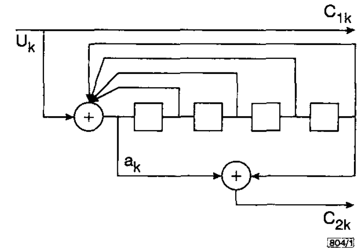

Fig. 1 *RSC encoder*

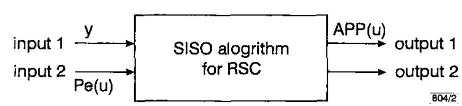

Fig. 2 *SISO decoder for RSC*

SISO decoder: For simplicity, we consider the binary encoder shown in Fig. 1. We denote the source output sequence by $u = (u_1, u_2, ..., u_N)$, where $u_k \in \{0, 1\}$. Without loss of generality, suppose that the initial state of the encoder is $S_0 = (0, 0, 0, 0)$. At each time k, the encoder accepts one symbol u_k and outputs two symbols $c_k = (c_{1k}, c_{2k})$, where $c_{1k} = u_k$. The code symbols c_k enter the modulator, which performs a one-to-one mapping of the symbols with modulator signals $x_k = (x_{1k}, x_{2k})$. The signals x_k are transmitted over a stationary memoryless channel with output symbols $y_k = (y_{1k}, y_{2k})$. The channel is characterised by the transition probability distribution $P(y|x)$. The decoder performs the task of estimating the source output sequence from the channel output sequence $y = (y_1, y_2, ..., y_N)$. To minimise the symbol error rate, the *a posteriori* probabilities (APPs) ($P(u_k = 0|y)$, $P(u_k = 1|y)$) will be calculated for all k. Note that $P(u_k = 0|y) + P(u_k = 1|y) = 1$; we can describe the SISO decoder as follows (see Fig. 2):

$$\text{Input1:} \quad y = (y_1, y_2, ..., y_N)$$

and