$$G(s) = \frac{b_0 + b_1 s + b_2 s^2 + \cdots + b_{n-2}s^{n-2} + b_{n-1}s^{n-1}}{a_0 + a_1 s + a_2 s^2 + \cdots + a_{n-1}s^{n-1} + a_n s^n} \tag{1}$$

and kth-order modified Routh approximant

$$G_k(s) = \frac{d_0 + d_1 s + d_2 s^2 + \cdots + d_{k-2}s^{k-2} + d_{k-1}s^{k-1}}{c_0 + c_1 s + c_2 s^2 + \cdots + c_{k-1}s^{k-1} + c_k s^k} \tag{2}$$

where $k < n$ and $n - k$ is even. Assuming that the parameters (γ_i, δ_i, α_i, β_i), $i = 1, 2, ..., [(k + 1)/2]$ have been computed as in [2], the procedure for determining the denominator and numerator coefficients of eqn. 2 is described below. When n and k are odd, $\alpha_1 = \beta_1 = 0$ are set in the procedure.

(i) Construct the following array:

$$\begin{array}{llllll} e_{11} & e_{12} & e_{13} & \cdots & e_{1,p-2} & e_{1,p-1} \quad e_{1p} \\ e_{21} & e_{22} & e_{23} & \cdots & e_{2,p-2} & e_{2,p-1} \\ e_{31} & e_{32} & e_{33} & \cdots & e_{3,p-2} & \\ \vdots & \vdots & \vdots & \cdots & & \\ e_{p-1,1} & e_{p-1,2} & & & & \\ e_{p1} = 1 & & & & & \end{array} \tag{3}$$

where $p = [(k + 3)/2]$. The first and last elements in each row are computed using (γ, α) parameters as follows:

$$\begin{aligned} e_{i1} &= \gamma_i e_{i+1,1} && i = p-1, p-2, ..., 2, 1 \\ e_{ij} &= \alpha_i e_{i+1,j-1} && (i,j) = (p-1,2), (p-2,3), ..., (2, p-1), (1, p) \end{aligned}$$

The remaining elements are determined by

$$e_{ij} = e_{i+2,j-1} + \alpha_i e_{i+1,j-1} + \gamma_i e_{i+1,j}$$
$$i = p-2, p-3, ..., 2, 1 \qquad j = 2, 3, 4, ..., p-i$$

Then the denominator coefficients of eqn. 2 are given by

$$\begin{aligned} c_{2i-2} &= e_{1i} && i = 1, 2, 3, ..., p \\ c_{2i-1} &= e_{2i} && i = 1, 2, 3, ..., p-1 \end{aligned} \tag{4}$$

(ii) Form the following array:

$$\begin{array}{lllllll} f_{11} & f_{12} & f_{13} & f_{14} & \cdots & f_{1,2q-2} & f_{1,2q-1} \quad f_{1,2q} \\ f_{21} & f_{22} & f_{23} & f_{24} & \cdots & f_{2,2q-2} & \\ \vdots & \vdots & \vdots & \vdots & \cdots & & \\ f_{q-1,1} & f_{q-1,2} & f_{q-1,3} & f_{q-1,4} & & & \\ f_{q1} & f_{q2} & & & & & \end{array} \tag{5}$$

where $q = [(k + 1)/2]$. The first and last elements in each row are computed using (δ, β) parameters and array 3 as follows:

$$\begin{aligned} f_{q1} &= \delta_q \\ f_{q2} &= \beta_q \\ f_{i1} &= \delta_i e_{i+1,1} && i = q-1, q-2, ..., 2, 1 \\ f_{ij} &= \beta_i e_{i+1,[\frac{j}{2}]} && (i,j) = (q-1,4), (q-2,6), ..., (2, 2q-2), (1, 2q) \end{aligned}$$

The remaining elements of the array are calculated by

$$\begin{aligned} f_{ij} &= f_{i+1,j-1} + \beta_i e_{i+1,[\frac{j}{2}]} && j = \text{even} \\ f_{ij} &= f_{i+1,j-1} + \delta_i e_{i+1,[\frac{j+1}{2}]} && j = \text{odd} \end{aligned}$$
$$i = q-1, q-2, ..., 2, 1 \qquad j = 2, 3, ..., 2q - 2i + 1$$

The numerator coefficients of eqn. 2 are then given by

$$d_{i-1} = f_{1i} \qquad i = 1, 2, ..., 2q \tag{6}$$

Example: Consider an eighth-order system given by [5]

$$G(s) = \frac{194480+482964s+511812s^2+278376s^3+82402s^4+13285s^5+1086s^6+35s^7}{9600+28880s+37492s^2+27470s^3+11870s^4+3017s^5+437s^6+33s^7+s^8} \tag{7}$$

and we derive a fourth-order modified Routh approximant. The computed values of parameters (γ_i, δ_i, α_i, β_i), $i = 1, 2$ are

$$\gamma_1 = 0.33241 \quad \delta_1 = 6.73407 \quad \alpha_1 = 0.0303 \quad \beta_1 = 1.06061$$
$$\gamma_2 = 1.05073 \quad \delta_2 = 16.45715 \quad \alpha_2 = 0.09862 \quad \beta_2 = 2.58147$$

Recursively applying eqn. 17 of [3], we obtain the following fourth-order approximant:

$$G_4(s) = \frac{7.07569+17.57156s+3.24558s^2+0.1046s^3}{0.34927+1.05073s+1.06462s^2+0.09862s^3+0.002989s^4} \tag{8}$$

On the other hand, arrays 3 and 5 are, respectively, given by

$$\begin{array}{lll} 0.34927 & 1.06462 & 0.002989 \\ 1.05073 & 0.09862 & \\ 1 & & \end{array}$$

$$\begin{array}{llll} 7.07569 & 17.57156 & 3.24558 & 0.1046 \\ 16.45715 & 2.58147 & & \end{array}$$

Then, eqns. 4 and 6 directly yield eqn. 8.

Conclusions: An algorithm has been presented in this Letter by which a modified Routh approximant of desired order can be directly constructed without computing lower-order approximants. The algorithm can also be used to find the improved approximant of [3].

20 July 1999

Electronics Letters Online No: 19991135
DOI: 10.1049/el:19991135

Younseok Choo (*School of Electrical, Electronic and Computer Engineering, Hong-Ik University, San 34-31, Sinan-Ri, Chochiwon-Eup, Yeonki-Kun, Chungnam, 339-701, Korea*)

E-mail: yschoo@wow.hongik.ac.kr

References

1 HUTTON, M.F., and FRIEDLAND, B.F.: 'Routh approximations for reducing order of linear, time-invariant systems', *IEEE Trans.*, 1975, **AC-20**, pp. 329–337

2 HSIEH, C.-S., and HWANG, C.: 'Model reduction of continuous-time systems using a modified Routh approximation method', *IEE Proc. D*, 1989, **4**, pp. 151–167

3 CHOO, Y.: 'Improvement to modified Routh approximation method', *Electron. Lett.*, 1999, **35**, pp. 606–607

4 KAHORADO, E., and GUTIERREZ, J.L.: 'Inversion algorithm to construct Routh approximants', *Electron. Lett.*, 1975, **21**, pp. 424–425

5 KRISHNAMURTHY, V., and SESHADRI, V.: 'Model reduction using the Routh stability criterion', *IEEE Trans.*, 1978, **AC-23**, pp. 729–731

Joint channel equalisation and channel decoding

A. Knickenberg, B.L. Yeap, J. Hámorský, M. Breiling and L. Hanzo

A non-iterative joint equaliser and decoder scheme is proposed which outperforms the iterative turbo equaliser by ~3.4dB at a BER of 10^{-3} over a five-path Gaussian channel.

Introduction: In the context of iterative turbo equalisation [1] both the convolutional encoder and the channel seen in Fig. 1 can be described with the aid of a trellis or a finite state machine. Soft information is exchanged between the equaliser and decoder, which employ soft-in/soft-out (SISO) algorithms, such as Bahi's maximum a posteriori (MAP) algorithm [3], iteratively, until a termination criterion is met [5]. It is anticipated, however, that an amalgamated non-iterative scheme would achieve a better performance than that of two separate algorithms.

Non-iterative joint equalisation and decoding scheme: In [4] the optimum non-iterative turbo-decoding algorithm was introduced for the decoding of parallel concatenated codes, which are also referred to as turbo codes. By contrast, in this Letter we propose a non-iterative joint channel decoding and equalisation scheme for the serially concatenated convolutional encoder plus dispersive channel arrangement of Fig. 1, which outperforms iterative turbo equalisers. In contrast to iterative turbo equalisation algorithms [5], where we need a certain number of iterations in order to achieve a good performance, according to our proposed solution we carry out equalisation and decoding in a joint, non-iterative step. We note here that conceptually a multipath channel can be seen as a specific convolutional encoder, having a coding rate of $R = 1$ and a constraint length K, which is equal to the maximum delay of the channel expressed in terms of the time units k' characteristic of the output of the decoder. We then have a system of two serial encoders, namely the convolutional encoder and the channel, separated by an interleaver, namely the channel interleaver.

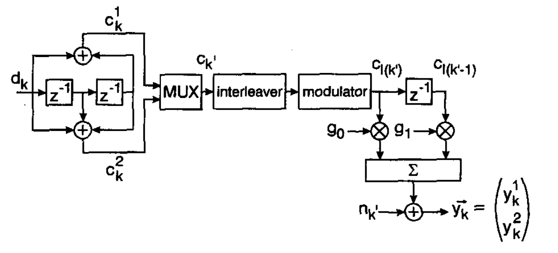

Fig. 1 *Convolutional encoder, interleaver and channel complex*

For the proposed joint equalisation and decoding scheme we have to construct a finite-state machine, which models the whole system. The input of this finite-state machine is constituted by the original data bits and its output is the channel output. In a finite-state machine the output $\vec{y}_k$ and the next state S_{k+1} are fully determined by the current state S_k and the current input data bit d_k. However, the interleaver renders the receiver's task complex.

We consider the example of Fig. 1, where the multipath channel has $L = 2$ paths, corresponding to an inner encoder memory length of $\nu_I = L - 1 = 1$. The convolutional outer encoder of Fig. 1 has a constraint length of $K = 3$ and a memory length of $\nu_o = K - 1 = 2$. Each encoded bit is determined by the current incoming data bit d_k and the ν_o previous ones. These encoded bits pass through the interleaver and enter the multipath channel in the interleaved order. We denote the interleaver function by $l(k')$, which assigns the input time instant k' to the output time instant $l(k')$. Hence the output of the channel at time instant k' depends on $c_{l(k')}, c_{l(k'-1)}, \ldots, c_{l(k'-\nu_I)}$. Each convolutionally encoded bit in turn depends on the last ν_o data bits. Finally, we obtain the output of the channel as a function of the data bits as follows:

$$\begin{aligned} y_{k'} &= g_0[d_{l(k)} \oplus d_{l(k)-1} \oplus \cdots \oplus d_{l(k)-\nu_o}] \\ &\quad + g_1[d_{l(k-1)} \oplus d_{l(k-1)-1} \oplus \cdots \oplus d_{l(k-1)-\nu_o}] \\ &\quad \vdots \\ &\quad + g_{\nu_I}[d_{l(k-\nu_I)} \oplus d_{l(k-\nu_I)-1} \oplus \cdots \oplus d_{l(k-\nu_I)-\nu_o}] \end{aligned}$$

where g_i, $i = 0, \ldots, \nu_I$ represents the symbol-spaced dispersive channel impulse response taps. In conventional trellises, which are used to describe convolutional codes, the encoder state is determined by the bits in the encoder's shift-register. In our serially concatenated convolutional encoder plus channel we define a super-state S_k^*, which is determined by all the bits that are used to calculate the channel output, except the actual input bit, which is often referred to as the state transition bit.

The proposed decoder operates on the basis of constructing the encoder's super-trellis constituted by the previously defined super-states S_k^* and invoking the log-MAP algorithm for joint channel equalisation and decoding. In contrast to iterative turbo equalisers, where a soft-decision value is required for each bit, in this particular application the Viterbi algorithm [2] could have also been utilised, since there is no need for producing the soft outputs provided by the log-MAP algorithm, only the most likely transmitted sequence has to be found. More explicitly, the proposed non-iterative scheme refrains from passing soft-decision information iteratively between the separate trellises of the equaliser and channel decoder and operates on the basis of the super-trellis, rather than on the two independent trellises of the equaliser and channel decoder. We will demonstrate that invoking the supertrellis in a non-iterative step improves the performance of the amalgamated scheme in comparison to that of an iterative turbo equaliser.

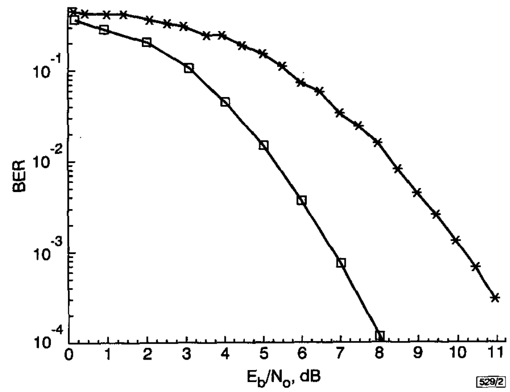

Fig. 2 *Comparison of iterative against non-iterative turbo equalisation over five-path Gaussian channel using convolutional code*

Convolutional code has following parameters: $R = \frac{1}{2}$, $K = 5$, $G[0] = 37$, $G[1] = 21$, $g_0 = g_4 = 0.227$, $g_1 = g_3 = 0.46$, $g_2 = 0.688$, 2×2052 interleaver
□ five-path non-iterative
✳ five-path iterative, 8th iteration

Results and conclusion: In Fig. 2 we have compared the BER performance of the iterative turbo equalisation technique performing eight iterations and the non-iterative joint equalisation and decoding scheme, over a dispersive five-path Gaussian channel. We used a rate $R = \frac{1}{2}$ constraint length $K = 5$ convolutional code with octal generator polynomials $G[0] = 37$ and $G[1] = 21$. The other simulation parameters were: channel coefficients of $g_0 = g_4 = 0.227$, $g_1 = g_3 = 0.46$, $g_2 = 0.688$; interleaver size of $2 \times 2052 = 4104$ bits. Here, the non-iterative joint equalisation and decoding scheme achieved an SNR gain of ~3.4dB at a BER of 10^{-3}. We also note that the performances of the Viterbi algorithm, max-log-MAP and the log-MAP algorithms were also compared and was found to be virtually identical. These performance improvements were achieved in the context of the $M \times N$ interleaver, where we opted for $M = 2$ at the cost of a reasonable complexity. For example, for $K = 5$ and a five-path channel we have $\nu_o + \nu_I = 4 + 4$ state bits, hence there are a moderate number of $2^8 = 256$ states. Although constrained by the specific interleaver, the underlying super-trellis structure and the performance of the non-iterative joint equalisation and decoding scheme are interesting also from a pure information theoretical point of view.

29 July 1999
Electronics Letters Online No: 19991126
DOI: 10.1049/el:19991126

A. Knickenberg, B.L. Yeap, J. Hámorský, M. Breiling and L. Hanzo (*Department of ECS, University of Southampton, SO17 1BJ, United Kingdom*)

E-mail: lh@ecs.soton.ac.uk

References

1 DOUILLARD, C., PICART, A., JÉZÉQUEL, M., DIDIER, P., BERROU, C., and GLAVIEUX, A.: 'Iterative correction of intersymbol interference: Turbo-equalization', *European Trans. Commun.*, 1995, **6**, pp. 507–511

2 VITERBI, A.J.: 'Error bounds for convolutional codes and an asymptotically optimum decoding algorithm', *IEEE Trans.*, 1967, **IT-13**, pp. 260–269

3 BAHL, L.R., COCKE, J., JELINEK, F., and RAVIV, J.: 'Optimal decoding of linear codes for minimising symbol error rate', *IEEE Trans. Inf. Theory*, 1974, pp. 284–287

4 BREILING, M., and HANZO, L.: 'Non-iterative optimum super-trellis decoding of turbo codes', *Electron. Lett.*, 1997, **33**, (10), pp. 848–849

5 BAUCH, G., KHORRAM, H., and HAGENAUER, J.: 'Iterative equalization and decoding in mobile communications systems'. European Personal Mobile Communications Conf., 1997, pp. 301–312

Optimal interleaver design for concatenated coding in DSL systems

M. Cao, K.R. Subramanian and V.K. Dubey

The Reed-Solomon code and multidimensional TCM (MTCM) can be concatenated to improve the performance of discrete multitone (DMT) modulated digital subscriber line (DSL) systems. The criteria for designing an effective multidimensional TCM (MTCM) scheme with efficient interleaver are described. The results of simulations for evaluating the performance of the proposed coding scheme are presented.

Selection of components in concatenated system: A concatenated coding scheme which combines multi-dimensional trellis coding modulation (MTCM) with the Reed-Solomon code (RS) can be used to improve the performance of a digital subscriber line (DSL) channel [1, 2]. According to our research findings [2], it can improve the system performance if the structure of the convolutional code is chosen properly. For example, the symbol error probability of a 2/3 16-state convolutional encoder with quadrature amplitude modulation (QAM) over an AWGN (additive white Gaussian noise) channel, according to the analysis in [2], can be described as follows:

$$P_{symb} \leq \frac{1}{2} Q\left(\sqrt{\left(\frac{d_{free}^2}{4\sigma^2}\right)}\right) \cdot e^{\left(\frac{d_{free}^2}{8\sigma^2}\right)} \frac{\partial}{\partial I} T(D,I)\Bigg|_{I=1, D=e^{-\frac{1}{4\sigma^2}}}$$
$$= Q\left(\sqrt{3\tau k \cdot SNR/snr_i}\right) \cdot e^{(1.5k\tau \cdot SNR/snr_i)}$$
$$\times \left(20e^{-2 \cdot SNR/E} + 1152e^{-2.5SNR/E} + 14592e^{-3 \cdot SNR/E} + \ldots\right) \quad (1)$$

The description of the parameters can be found in [2]. Its BER performance can be demonstrated to be superior to that of the original QAM design. To estimate its performance in realistic way, the statistical equivalent free distance (EFD) is proposed in [2] which takes into account an average distance. We observe from the simulation results that the EFD is much more effective for estimating the coding gain than is the direct distance. Furthermore, increasing the dimensions of TCM can improve the BER performance much more than is possible by changing the code rate; however, increasing the number of states improves the performance marginally. A well-designed interleaver is also crucial for this integrated coding system, because it can optimise the error propagation pattern to match the decoder's capability fully. Existing standards or published designs are not fully compatible with MTCM, thus leading to unexpected redundancy and serious transmission latency. Therefore an interleaver specifically designed for MTCM is the main issue addressed in this Letter.

Interleaving for various TCM codes: In the asymmetrical DSL (ADSL) standard, a convolutional interleaver is used with a specified interleaving buffer size for several classes of transmission [3]. However, it is not a specialised design for TCM schemes. Maximum-likelihood convolutional decoders normally generate error bursts in consecutive bits [4]. Therefore the interleaver should be designed to optimally make the errors from the TCM decoder sparse. A larger interleaver depth can enable the burst errors to be randomised by the trellis decoder more efficiently; however, it leads to increased buffer size and latency.

Simulation results and analysis: A series of interleaving depths are compared for various RS-TCM concatenated codes, to evaluate their performance and determine an optimum scheme, e.g. Fig. 1 for RS-4D TCM. We find that when the interleaving depth increases, its interleaving gain becomes saturated. Suppose that a range of δ = 0.5dB for this gain limit is acceptable; then the least-interleaving depth for this δ is 8 and we denote it as an efficient interleaver depth (EID). The EID is useful for the design of an optimal interleaver depth from the point of view of both avoiding large system latency and achieving sufficient interleaving. Interleaving gain against interleaver depth is plotted in Fig. 2*a*, in which the results of 2D, 4D and 8D are also included, where the number of TCM states is fixed at 16. It can easily be observed that the EID is strongly related to the order of dimension of these TCM codes. A concatenated system with higher dimensional TCM

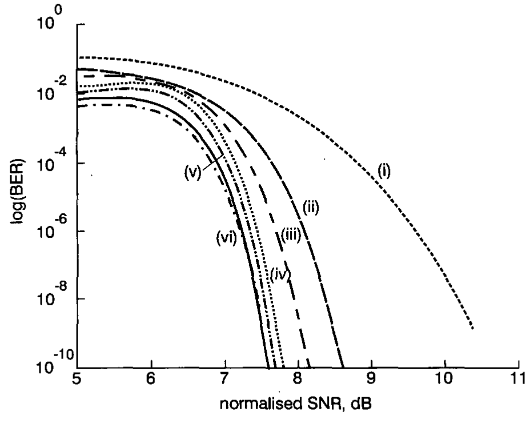

Fig. 1 *BER performance comparison of various interleaver depths (4D)*

(i) no interleaving
(ii) *RS*(212, 194), *IND* = 2
(iii) *RS*(212, 194), *IND* = 4
(iv) *RS*(212, 194), *IND* = 8
(v) *RS*(212, 194), *IND* = 16
(vi) *RS*(212, 194), *IND* = 32
–·–·–·– *RS*(212, 194), *IND* → ∞

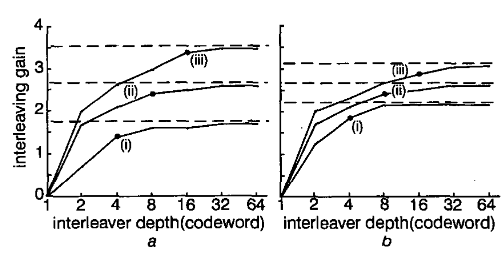

Fig. 2 *Interleaving gains against interleaving depth based on different TCM*

a Fixed number of trllis states
(i) *RS*(212, 194) + 2D 16-state
(ii) *RS*(212, 194) + 4D 16-state
(iii) *RS*(212, 194) + 8D 16-state
b Number of trellis states varied
(i) *RS*(212, 194) + 2D 8-state
(ii) *RS*(212, 194) + 2D 16-state
(iii) *RS*(212, 194) + 2D 64-state

requires a larger EID. The main reason is that the error bursts brought about by the high dimension will convert to multiple 2D symbol errors at the same noise level. For example, if a 4D dimension TCM inner code is used, the information bits of every two subchannels formed a 4D symbol. As a result, any error bursts caused by the convolutional decoder will generate double the number of subchannel errors; thus the number of RS symbol errors will also double. As a side effect, the correct symbols between two error bursts are also doubled on average when the total symbol error (proportional to BER) is the same. Therefore the 4D concatenated system requires a larger interleaver depth for distributing the symbol errors. For the same reason, 8D TCM requires an even higher EID. The number of TCM trellis states is another factor in determining the EID. As shown in Fig. 2*b*, a larger state number reflects a higher EID. When the state number increases, the error symbols converge more strongly to form bursts