

Intersected low-density parity-check and convolutional codes

Edward A. Rutzer
Cavendish Laboratory, University of Cambridge
Madingley Road, Cambridge CB3 0HE
ear23@mrao.cam.ac.uk

Abstract

We look at the intersection of convolutional codes with high rate low-density parity-check codes. Threshold values are obtained with EXIT charts. Simulations show a gain of 1.5dB over a standard deep-space code.

1 Introduction

In current communication systems it is common to use a serial concatenation of a Reed-Solomon code with a convolutional code (SCRSCC) [4, 6]. Commonly the convolutional code is decoded to a maximum likelihood codeword and this is then used as an input to a Reed-Solomon decoder. Information is not used effectively in this process since the convolutional code is returning a hard decision. A Reed-Solomon decoder is in general not capable of using soft information.

Large block codes can approach the Shannon Limit, for example low-density parity-check codes [3]. In general, these suffer from worse than linear time encoding algorithms and the need for a complete block to be received before decoding can be attempted (this is a problem for low-speed telemetry links). To keep encoders close to linear time and allow early decoding, it is useful to keep the structure of a convolutional code combined with a high rate block code.

Convolutional codes with a soft-input soft-output decoder [12] have been shown to perform well in concatenated coding schemes [2]. This paper looks at replacing the Reed-Solomon component code of the SCRSCC with a low-density parity-check code [13].

Iterative decoding [7] will be used. We will use extrinsic information transfer (EXIT) charts [16] to analyse a code's performance in terms of properties of the messages being passed during decoding.

This paper builds on an earlier short paper [14].

2 Code Construction

To maintain a simple graph structure we look at the intersection of a low-density parity-check code with a convolutional code. We will call this code an ICG code (intersected convolutional Gallager code). An example factor graph showing the constraints satis-

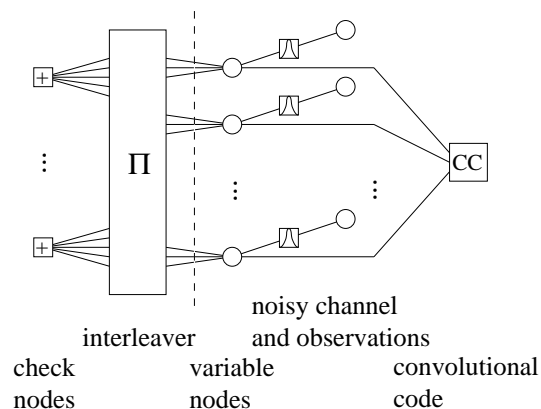


Figure 1: The factor graph of an ICG code, shown with a Gaussian noise channel. The point of message analysis, section 3, is shown with a dotted line.

fied is shown in figure 1. Decoding is carried out by belief propagation on the factor graph [7]. The iterative belief propagation is terminated when a tentative decoding is found that is a codeword of both the low-density parity-check code and the convolutional code (or a maximum number of iterations is reached).

With the LDPC component kept to be high rate, the majority of the encoding can happen in linear time with the convolutional code. The final few bits will be encoded by matrix multiplication.

3 EXIT chart analysis

The behaviour of the iterative decoding can be shown using an EXIT chart. An EXIT chart plots the transfer of information back and forth between two constituent parts of the decoding graph during decoding. For each part we seek to obtain the function relating the information input and information output in an iteration. For this function to exist, it is necessary that the graph section be loop-free. In figure 1 the graph has been shown split into two loop-free sections; the left section consists of the check nodes and the right section consists of the variable nodes and the convolutional code. The transfer function of the right section depends on the channel noise level; the left section does not. An advantage of ICG codes over turbo codes [2] is that the form of the transfer functions can

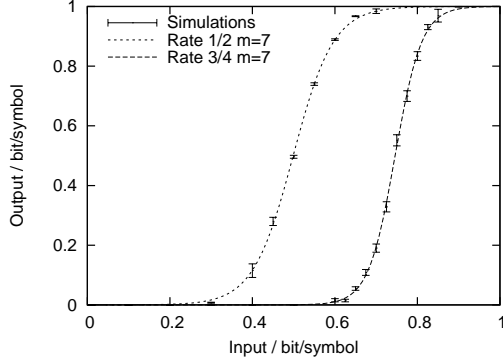


Figure 2: EXIT charts of $m=7$, $R=3/4$ and $m=7$, $R=1/2$ convolutional codes showing simulation results and tanh fits.

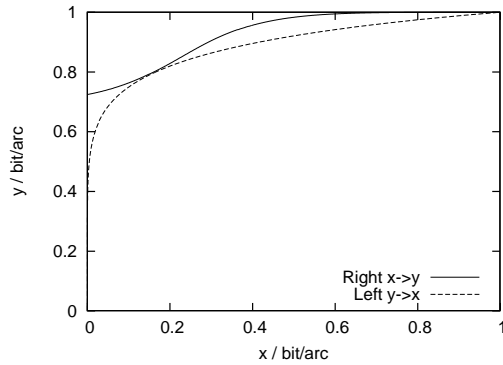


Figure 3: EXIT chart of $j = 1$ ICG code at threshold

be altered by adjusting the degree sequences of the nodes.

Approximate forms for the transfer function of the variable node and check nodes were used as detailed in appendix A. The transfer function of the convolutional code was simulated and then a tanh function fitted to it. The fit was tried for a range of different convolutional codes and good fits were generally found. Some examples are shown in figure 2.

An example EXIT chart is shown in figure 3. In the limit of large block size a code will decode if a swath exists between the two curves. The channel noise level was altered by binary division to find the threshold where the curves just intersect.

4 Threshold Optimization

Using a global optimization package (DIRect [8]) with “fractional phantom distributions” [15] optimization of the degree sequence of the low-density parity-check nodes was conducted. For high rate component codes, this suggested that the largest gain in performance for high rate low-density parity-check component codes could be achieved by a simple mixture of weight 0 and weight 1 columns – equivalent

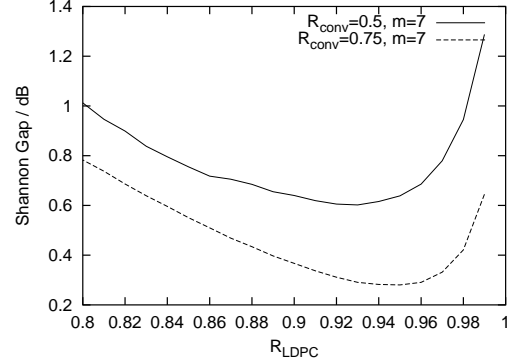


Figure 4: The gap from the Shannon Limit for ICG codes with two different convolutional component codes and the low-density parity-check codes made from a mixture of weight 1 and weight 0 columns.

to only having some of the nodes of the convolutional code involved with the intersection with the low-density parity-check code. Intuitively this corresponds to the low-density parity-check code having a lower rate and being able to give better help to the bits to which it is connected. When the rates of the convolutional and low-density parity-check component codes become similar then a more complex irregular sequence with higher weight columns is preferred.

The gap between the threshold and the Shannon Limit for codes with a mixture of weight 1 and weight 0 columns is shown in figure 4. It can be seen that there is a minimum in these graphs.

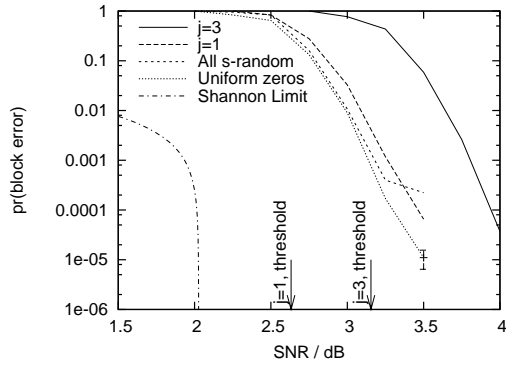
With weight 1 columns there is local structure in the parity-check matrix. If the parity checks are uniformly interleaved then this spreads out the structure. However a burst of errors from the convolutional code can easily knock out a large number of parity checks. Instead we applied an S-random interleaver [5] to the parity-checks.

With no weight 0 columns this strategy works well. For a code with weight 0 columns a poor error floor is found. Many sequential symbols of the convolutional code can be disconnected from the low-density parity-check code. This can be at the same position as a burst of errors from the convolutional code which leads to a bad error floor. Better results are found if the weight 0 columns are regularly spaced through the code and then an S-random interleaver applied to the other columns.

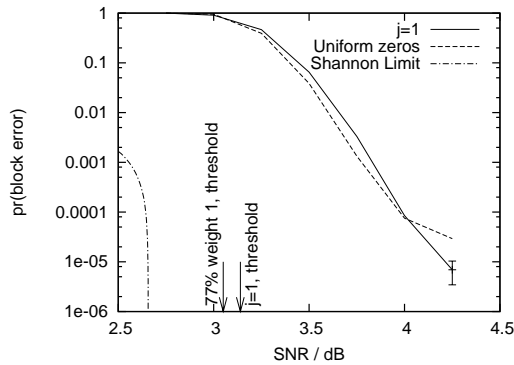
5 Code construction results

Ideally we would like the low-density parity-check component code rate to be close 1 to keep the encoding complexity close to linear. For a slow telemetry application this means we do not need to suspend data transmission to transmit a large block of parity bits.

Simulations are shown with two different rates of



(a) Two regular and two irregular $R=0.9$ low-density parity-check codes to give $R \approx 0.65$ ICG codes



(b) One regular and one irregular $R=0.95$ low-density parity-check code to give $R \approx 0.7$ ICG codes

Figure 5: Convolutional code ($N = 3000$, $R = 3/4$, $m = 7$) intersected with low-density parity-check codes

low-density parity-check component codes in figure 5.

In figure 5(a) an $R = 0.9$ low-density parity-check component code is used. Two regular codes with $j = 1$ and $j = 3$ are shown with two irregular codes (both with 80% weight 1 columns and 20% weight 0 columns). Firstly one can see the improvement of the $j = 1$ over the $j = 3$ regular low-density parity-check component code. When used alone $j = 3$ low-density parity-check codes are good codes, however strong codes are often not good constituent codes in an iterative decoding scheme. When the waterfall region is steep (as with a low-density parity-check code) the constituent decoder gives an “all-or-nothing” answer; this does not help iterative decoding. Also shown is the effect on the error floor between having uniformly distributed weight 0 columns and having them as part of the S-random permutation. A significant reduction in the error floor can be seen.

The use of an S-random interleaver imposes a maximum rate on R_{LDPC} . $R_{LDPC} = 0.95$ was approxi-

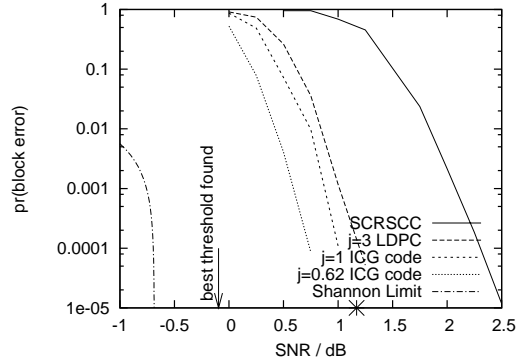


Figure 6: A comparison of ICG with SCRSCC and LDPC codes. All codes have $N = 4092$ and $R = 0.4360$. The constituent convolutional codes are identical. One point of the *bit* error rate of a larger block size ($N = 130572$) iteratively decoded SCRSCC is shown with an asterisk [10].

mately the highest rate that could be simulated with $N = 3000$. With a higher rate the S-random interleaver could not spread the parity bits widely enough. The parameter S of the interleaver generally satisfies $S < \sqrt{N/2}$ [5]. However to widely distribute the parity checks we need $S > \text{row weight}$. The simulation results are shown in figure 5(b). The irregular code had 23% weight 0 columns and 77% weight 1 columns. It can be seen that there is an error floor with weight 0 columns despite having the weight 0 columns uniformly distributed.

6 Comparison with a deep-space standard

A standard for deep-space communications is a rate 1/2 convolutional code serially concatenated with (255,223) Reed-Solomon codes [4]. This was compared to two ICG codes (one regular with $j = 1$ and the other with 38% of the columns being weight 0 and the rest weight 1). Simulation results are shown in figure 6. For comparison a regular $j = 3$ low-density parity-check block code is shown. It can be seen that a gain of 1.5dB over the deep-space standard is achieved with the ICG code and that the ICG code compares favourably with a non-optimized sparse graph block code.

Recently some iterative techniques for decoding SCRSCC codes using state pinning have been introduced [10]. With multiple interleaved Reed-Solomon codes a gain of about 1dB can be found by conducting iterative decoding. This is smaller than the gain found at $N = 3000$ (equivalent to no interleaving) with ICG codes. As N is increased the ICG code performance should further approach its threshold. A value for the *bit* error rate of an iteratively decoded SCRSCC with interleaver depth $I = 32$ ($N = 130572$) from [10] is shown in addition in figure 6.

7 Discussion

The error floor present with weight 0 columns still needs further work to try and reduce it further. It may be possible to ensure that the weight zero columns do not match very low weight error patterns of the convolutional code.

Thresholds could be further investigated. Firstly a theoretical model for the threshold of an ICG code could be worked upon. Secondly a study of different block sizes could be carried out to find out whether the approximate threshold values are approached as the block size increases.

8 Conclusion

Intersected convolutional and low-density parity-check codes have been presented and shown to compare favourably with other coding schemes whilst showing computational advantages. Experimentally approximate threshold values have been found, some less than 0.3dB from the Shannon Limit. Simulations of medium size blocks have shown code performance within 1.5dB of the Shannon Limit at a block error probability of 10^{-5} .

A Approximate EXIT chart form

Following [17] an approximate form for the EXIT charts will be used. A detailed derivation is not given in [17] so one is given below.

We will assume the messages are Gaussian distributed. First we need to look at the form of log-likelihood ratio (LLR) messages from sources with Gaussian noise. Following [11] let L be the LLR ensemble from the Gaussian distributed real space ensemble Y :

$$\Pr(y|\mu = +\mu_y)dy = \frac{1}{\sqrt{2\pi}\sigma_y} \exp\left(-\frac{(y-\mu_y)^2}{2\sigma_y^2}\right) dy \quad (1)$$

By definition the LLR for a BPSK signal is defined as follows:

$$\begin{aligned} l &\equiv \log(\Pr(y|\mu = +\mu_y)) - \log(\Pr(y|\mu = -\mu_y)) \quad (2) \\ &= \frac{2\mu_y}{\sigma_y^2} y \quad (3) \end{aligned}$$

We can then use this substitution to find the conditional distribution of L :

$$\begin{aligned} \Pr(l|\mu = +\mu_y)dl &= \frac{dy}{dl} \frac{1}{\sqrt{2\pi}\sigma_y} \exp\left(-\frac{((\sigma_y^2/2\mu_y)l - \mu_y)^2}{2\sigma_y^2}\right) dl \quad (4) \end{aligned}$$

$$= \frac{1}{\sqrt{2\pi} \frac{2\mu_y}{\sigma_y}} \exp\left(-\frac{\left(l - \frac{2\mu_y^2}{\sigma_y}\right)^2}{2\left(\frac{2\mu_y}{\sigma_y}\right)^2}\right) dl \quad (5)$$

which leads to the result that

$$L \sim N(\mu_l, \sigma_l) \quad (6)$$

$$\text{where } \sigma_l^2 = 2\mu_l \quad (7)$$

We can now look at the information between a LLR message with standard deviation σ_l and a bipolar distribution in real space, X . To follow [17] we will call this function $J(\sigma_l)$.

$$\begin{aligned} J(\sigma_l) &= I(X;L) = H(X) - H(X|L) \quad (8) \end{aligned}$$

$$= 1 - \sum_{x=\pm 1} \int_{-\infty}^{+\infty} \Pr(x,l) \log_2 \frac{1}{\Pr(x|l)} dl \quad (9)$$

$$= 1 - \sum_{x=\pm 1} \int_{-\infty}^{+\infty} \frac{1}{2} \Pr(l|x) \log_2 \frac{1}{\Pr(x|l)} dl \quad (10)$$

which by symmetry:

$$J(\sigma_l) = 1 - \int_{-\infty}^{+\infty} \Pr(l|x = +1) \log_2 \frac{1}{\Pr(x = +1|l)} dl \quad (11)$$

For our bipolar X we can easily calculate

$$\begin{aligned} \Pr(x = +1|l) &= \frac{\Pr(l|x = +1)}{\Pr(l|x = +1) + \Pr(l|x = -1)} \quad (12) \end{aligned}$$

$$= \frac{1}{1 + \exp(-l)} \quad (13)$$

Substituting equation 13 and equation 5 into equation 11 we get

$$\begin{aligned} J(\sigma_l) &= 1 - \int_{-\infty}^{+\infty} \frac{\exp\left(-\frac{(l-\sigma_l^2/2)^2}{2\sigma_l^2}\right)}{\sqrt{2\pi}\sigma_l} \log_2\left(1 + e^{-l}\right) dl \quad (14) \end{aligned}$$

This can be evaluated numerically. The approximate piecewise closed forms for J and J^{-1} from [17] were used in simulations in this paper.

We know that variable nodes add log-likelihood messages [9]. Hence the extrinsic information transfer function of a variable node with an incoming extrinsic information I and degree d_v is

$$f_v(I) \approx J\left(\sqrt{(d_v - 1)J^{-1}(I)^2 + \sigma_{ch}^2}\right) \quad (15)$$

where σ_{ch} is the log-likelihood standard deviation of the message from the channel.

For check nodes we can use a duality property which holds for a binary erasure channel [1]:

$$f_c(I) = 1 - f_v(1 - I) \quad (16)$$

and write

$$f_c(I) \approx 1 - J\left(\sqrt{d_c - 1} J^{-1}(1 - I)\right) \quad (17)$$

where d_c is the check node degree. Simulations were carried out with Gaussian messages to find out how accurate this approximation is. In figure 7 it can be seen that the approximation is reasonable.

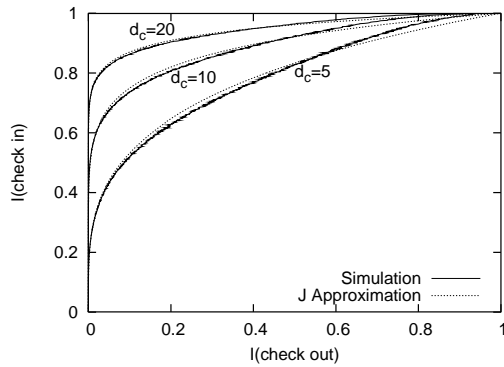


Figure 7: EXIT charts of check nodes of different degrees

References

- [1] Alexei Ashikhmin, Gerhard Kramer, and Stephan ten Brink. Extrinsic information transfer functions: A model and two properties. In *36th Annual Conference on Information Sciences and Systems*, March 2002.
- [2] Claude Berrou and Alain Glavieux. Near optimum error correcting coding and decoding: Turbo-codes. *IEEE Transactions on Communications*, 44(10):1261–1271, October 1996.
- [3] Sae-Young Chung. On the design of low-density parity-check codes within 0.0045dB of the Shannon limit. *IEEE Communications Letters*, 5(2):58–60, February 2001.
- [4] Consultative Committee for Space Data Systems. *Telemetry Channel Coding*, October 2002. CCSDS 101.0-B-6.
- [5] Dariush Divsalar and Fabrizio Pollara. Multiple turbo codes for deep-space communications. Technical report, Jet Propulsion Laboratory, May 1995.
- [6] Jeff Foerster and John Liebetreu. FEC performance of concatenated Reed-Solomon and convolutional coding with interleaving. Technical Report 802.16.1pc-00/33, IEEE 802.16, June 2000.
- [7] Brendan J. Frey. *Graphical Models for Machine Learning and Digital Communication*. The MIT Press, 1998.
- [8] Joerg M. Gablonsky and C. Tim Kelley. A locally-biased form of the DIRECT algorithm. *Journal of Global Optimization*, 21:27–37, 2001.
- [9] Robert G. Gallager. *Low-Density Parity-Check Codes*. MIT Press, 1963.
- [10] Joachim Hagenauer, Elke Offer, and Lutz Papke. Matching Viterbi decoders and Reed-Solomon decoders in a concatenated system. In Stephen B. Wicker and Vijay K. Bhargava, editors, *Reed-Solomon Codes and their Applications*. IEEE, 1994.
- [11] Peter Hoeher, Ingmar Land, and Ulrich Sorger. Log-likelihood values and Monte Carlo simulation – some fundamental results. In *2nd International Symposium on Turbo Codes and Related Topics*, September 2000.
- [12] Rolf Johannesson and Kamil Sh. Zigangirov. *Fundamentals of Convolutional Coding*. IEEE, 1998.
- [13] David J.C. MacKay. Good error correcting codes based on very sparse matrices. *IEEE Transactions on Information Theory*, 45(2):399–431, 1999.
- [14] Edward A. Ratzert. Intersected low-density parity-check and convolutional codes. In *Postgraduate Research Conference in Electronics, Photonics, Communications and Software*, April 2002.
- [15] Thomas J. Richardson, M. Amin Shokrollahi, and Rüdiger L. Urbanke. Design of capacity-approaching irregular low-density parity-check codes. *IEEE Transactions on Information Theory*, 47(2):619–637, February 2001.
- [16] Stephan ten Brink. Convergence of iterative decoding. *Electronics Letters*, 35(10):806–808, May 1999.
- [17] Stephan ten Brink, Gerhard Kramer, and Alexei Ashikhmin. Design of low-density parity-check codes for multi-antenna modulation and detection. Submitted to *IEEE Transactions on Communications*, June 2002.