ITU - Telecommunication Standardization Sector Temporary Document

STUDY GROUP 15 Original: English

Florida, Clearwater, 8-12 January 2001

Question: 4/15

SOURCE[1] : iCODING Technology Inc.

TITLE: G.gen.bis : Turbo Coding for the ADSL DMT Channel

ABSTRACT

This proposal aims to provide a viable turbo coding scheme for the ADSL DMT channel and has the following features:

- Proposed turbo code utilises a revolutionary triple code structure which results in good performance at 3/8 the complexity of a traditional 16-state double code structure.
- Code uses generatable interleavers which have comparable performance to the best S-type interleavers.
- Constellation sizes have been limited to an even power of 2 to simplify the LLR calculations required by the decoder. In additiion, all bit mappings to the constellations use Gray coding which achieves the best performance and allows for the simplified LLR calculations.
- Two approaches to implementing a turbo code for the ADSL DMT channel are presented:
    1. Variable code rate system, which performs puncturing on a per constellation basis, allowing controlled bit mapping to take place,
    2. Constant code rate system, which uses a uniform puncturing system to reduce memory size and simplify the design.
- Both schemes achieve similar performance, with the variable code rate system achieveing slightly greater throughput.
- Variable code rate system achieves throughputs of 2Mbit/s and 5.8 Mbit/s for the 35 dB and 50 db non-flat SNR channels, with corresponding coding gains of 8.5 dB and 8.0 dB.
- Very robust performance is achieved for the impulse noise channels, indicating system may satisfactorily operate without the need of a concatenated Reed-Solomon Code. By utilising an additional interleaver for the data, considerable gain in impulse noise tolerance was achieved.
- The resultant latency of the system is inversely proportional to the channel throughput, requiring the need to vary the block size of the turbo code in order to control the latency.
- Multiple latency paths can be implemented by utilising multiple turbo codecs, each supporting reconfigurable block sizes to reduce the number of codecs required. It should be noted that such a scheme requires a hardware implementation to be effective.

---

[1] Contact: Wade Farrell, Ph.D. T: +61 8 8293 5451
iCODING Technology Inc. F: +61 8 8297 2860
Australian Office : E: wade@icoding.com
14 Dryden Rd
Black Forest, SA 5035
Australia

1 INTRODUCTION

The ADSL discrete multi-tone (DMT) channel can be considered as a series of individual tones, with the SNR of each tone and the number of tones available varying from channel to channel. The constellation assigned to each tone must be flexible in order to take advantage of what SNR is available. This means the overall throughput of the channel is dependent on the SNR profile of the tones. Given this type of channel, we wish to implement a turbo coding scheme which can satistifactorily handle the variable throughput of the channel and provide improved coding gain over the previous standards, whilst at the same time providing acceptable latency, with the option of supporting multiple latency paths.

As turbo codes are block-based coding schemes, given the variable transmission of the DMT channel, it is not possible to synchronise each coded block with the transmission symbol rate. However this is not a problem if we can de-couple the coding from the modulation, removing any dependencies between the two stages. This can be done by double buffering the output of the encoder and the input into the decoder. Double buffering is required as the code block size is independent to the transmission rate, thus code block transitions will occur at any tone number within a symbol period. Further discussion of how double buffering is implementated is provided in Sections 3 and 4.

To be able to buffer the received information before passing it onto the decoder, requires the information to be converted into bit probabilities or bit Log-likelihood ratios (LLRs). The fact that turbo codes can accept LLRs at the decoder is an important property, which means constellation information does not need to be passed to the decoder, achieving the de-coupling required.

Given that we can satisfactorily de-couple the codec from the channel through the use of double buffering, consideration on the type of puncturing and constellation mapping must be made. This proposal presents two such methods; variable code rate and constant code rate which are discussed in Sections 3 and 4. Details of the revolutionary turbo code used is provided in Section 2. Discussion of the corresponding latency and the possibility of multiple latency paths is provided in Section 5. Results are presented for flat SNR AWGN channels, linearly decreasing SNR AWGN channels and impulse noise channels in Section 6.

2 TURBO CODE DETAILS

The turbo code used to generate the results for this proposal has the following properties:

- All bits are encoded
  Many of the previous proposals presented thus far encode the least significant bits of each constellation, leaving the majority of bits of each constellation uncoded. Although this results in comparable performance, it leaves the remaining uncoded bits vulnerable against burst errors. This problem can be solved by the use of concatenated Reed Solomon coding, however it is our intention to design a coding scheme without the need to concatenate a Reed Solomon code.

- Triple Code Structure
  The most notable characteristic of the turbo code used is the fact that it is a triple parallel concatenated convolutional code (PCCC). This is a simple extension to the standard double PCCCs, requiring the addition of another interleaver and constituent code.

- Low Complexity 4-state constituent codes
  Using the triple code structure, it is possible to reduce the code size of each constituent code and still achieve good performance. Using 4-state constituent codes results in 3/8 the complexity per iteration when compared to a 16-state double PCCC implementation, with a performance loss of approximately 0.2 dB given the same number of iterations. However if comparing the two schemes

traditional 16-state double code. In addition, from previous tests it has been shown for AWGN channels that the triple 4-state turbo code can achieve BER=10^{-10} before hitting an error floor.

Figure 1 has been removed from this public document

Figure 1 – Triple Turbo Code Structure

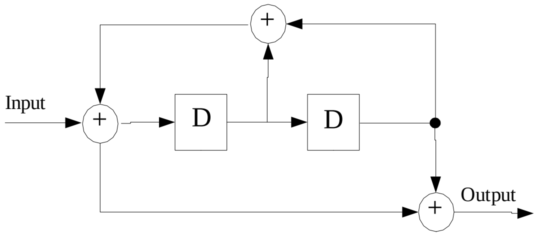

Figure 2 – 4-State Constituent Code (generator polynomials : g0 = 7, g1 = 5)

•Generatable Interleavers

The interleavers used (note two are required for a triple code) are generatable and achieve comparable performance to the S-type interleavers. Utilising generatable interleavers is a very important attribute if either the encoder and/or decoder are to be implemented in hardware, as considerable savings in memory can be made. Full details of the interleavers are not provided here as they are proprietry of iCODING, but results can be reproduced using S-type interleavers.

•Even power of 2 constellations

Due to the simplicity of the LLR calculations, the available constellations have been limited to square constellations. It is possible to implement non-square constellations however more complicated LLR calculations are required, which have not been investigated here.

•Gray mapping of constellations

It has been show in [10] that Gray mapping provides the best performance for turbo codes. In addition such mapping allows for very simple LLR calculations to be made, which are essential if the decoder is to be implemented in hardware.

•Fine Gain Control Facility unused

To simplify simulations, all results in this proposal have been generated without the use of fine gain control. It is expected that using gain control for each tone would result in worthwhile improvement and it is the intention to include this feature in subsequent proposals.

It is important to note that the two turbo coding schemes presented below; variable code rate and constant code rate are not code specific, effectively allowing any type of turbo codec to be utilised. However the triple 4-state PCCC presented represents an excellent option of a low complexity solution which still provides adequate coding gains.

3 VARIABLE CODE RATE SCHEME

Figure 3 provides a block diagram of the variable code rate system. It shows separate data and parity output from the encoder. Note that although not indicated in the block diagram, the parity bits output have not been punctured. Therefore the number of parity bits, which is code specific, will consist of 3 and it is the intention to include this feature in subsequent proposals.

It is important to note that the two turbo coding schemes presented below; variable code rate and constant code rate are not code specific, effectively allowing any type of turbo codec to be utilised. However the triple 4-state PCCC presented represents an excellent option of a low complexity solution which still provides adequate coding gains.

3 VARIABLE CODE RATE SCHEME

Figure 3 provides a block diagram of the variable code rate system. It shows separate data and parity output from the encoder. Note that although not indicated in the block diagram, the parity bits output have not been punctured. Therefore the number of parity bits, which is code specific, will consist of 3 bits per data bit for the triple code presented. The output data and parity bits are then double buffered on a frame basis, with the buffer required to store up to 2 x N x (data + parity) bits. Although implementations are likely to require less than this by either pre-puncturing or timing the encoder output so a new frame is only generated just prior to the previous frame being completely transmitted. The output of the double buffer is controlled by the throughput of the channel.

At the receiver, using the chosen constellation and code rate selections, LLRs are calculated for each received data and parity bit and the sequence depunctured before being buffered. When the buffer receives a complete code frame, the buffers are swapped and decoding begins on the received frame. Note that the decoder reads from this buffer for the duration of processing. It is possible to reduce the size of the double buffer by placing it prior to the LLR calculations, thus storing symbols instead of LLRs. This can reduce the memory requirements considerably but additional control is required to handle code frame transitions.

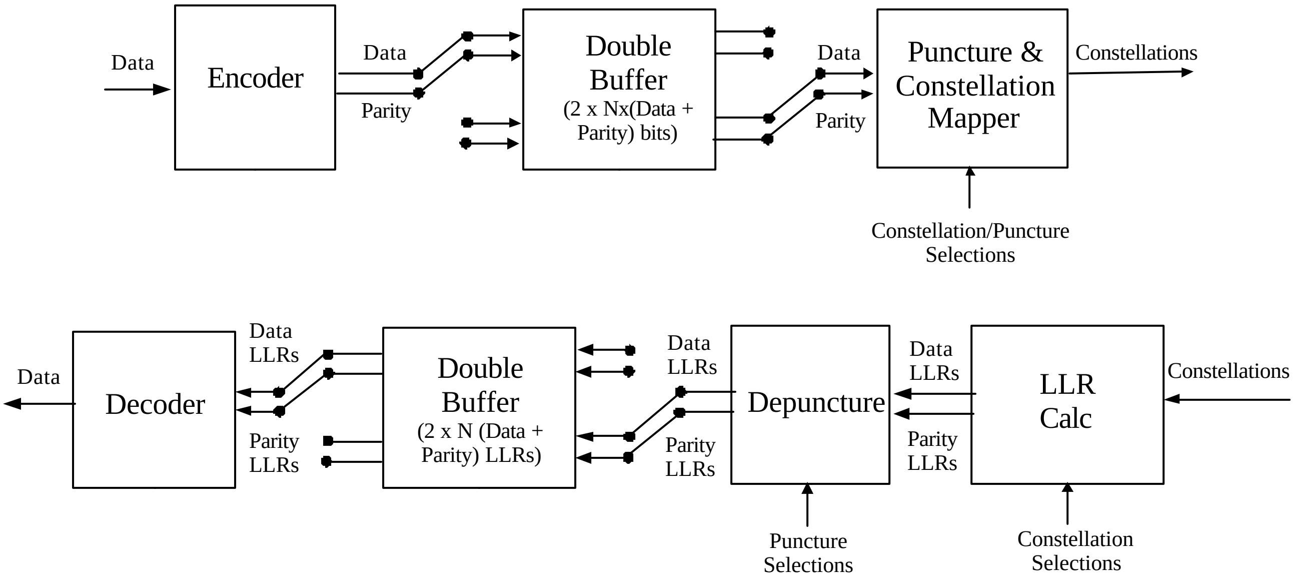

Figure 3 - Variable Code Rate, Transmit and Receive Block Diagram

The main property of the variable code rate scheme is all parity bits are passed to the constellation mapping stage, with puncturing of the parity bits performed on a per constellation basis. Given the operating SNR profile, each tone is assigned a constellation and a code rate. This allows the bit assignment to be controlled for each constellation, aiming to give the greater protection to the data bits and the least protection to the parity bits. For example, for a particular tone, 1024QAM, Rate 7/10 is assigned. This means that 7 data bits and 3 parity bits are assigned to the symbol. Given the triple PCCC which uses rate ½ constituent codes, for the 7 data bits, 21 parity bits are generated of which

18 parity bits are punctured/unused. If this assignment is applied to a total of 200 tones for instance, 1400 data bits are transmitted for each symbol duration.

When a mixture of constellation and code rates are used, the corresponding code rate will be the average of what code rates are assigned, hence the name variable code rate. Selection of constellation/code rates is to be performed upon initialisation of the channel using a simple look-up table which stores the corresponding SNR thresholds for each spectral efficiency supported.

Given the flexibility of varying code rate and constellations, it is important to determine which constellations and code rate combinations achieve the best performance for each spectral efficiency. For example, 16384QAM, Rate 4/7 results in the same spectral efficiency of 8 bps/Hz as 1024QAM Rate 4/5, however performance will differ. In Section 6.1, Table 2 provides a list of the best scheme for each spectral efficiency, showing Eb/No and corresponding SNR at which BER=10^{-7} is achieved. Performance of the variable rate scheme is provided in Section 6.2 using the linearly decreasing SNR channels.

Due to the size of certain constellations and the degree of puncturing that must occur, many of the constellation/code rate bit assignments require more than 1 symbol to cycle through the puncture pattern. This is not a problem if each scheme is implemented using the same cycle period, which means the effect of mixing constellations and code rate selections is reduced. In addition, due to the nature of the ADSL DMT channel, it is likely that the same constellation/code rate selections are grouped together, thus avoiding the problem of mixed puncturing for most of the tones.

4 CONSTANT CODE RATE SCHEME

An alternative to the variable rate scheme is to perform uniform puncturing across the whole coded frame prior to mapping onto constellations. As a result, this scheme achieves a constant code rate which is independent of the channel. Figure 4 details the block diagram of such a scheme. From this diagram, it can be seen that the puncturing occurs prior to buffering, which reduces the buffer size requirements. This scheme has the added benefits of choosing a uniform puncturing rate, compared to the uneven puncturing likely to occur when using the variable code rate scheme. However the output bit sequence after puncturing is mapped directly to the constellations without any consideration to bit protection, resulting in a random sequence of protection levels. It is possible to re-sort the data and parity output from the puncture circuit to provide good bit assignments with each constellation but the results presented in this proposal do not include such a method.

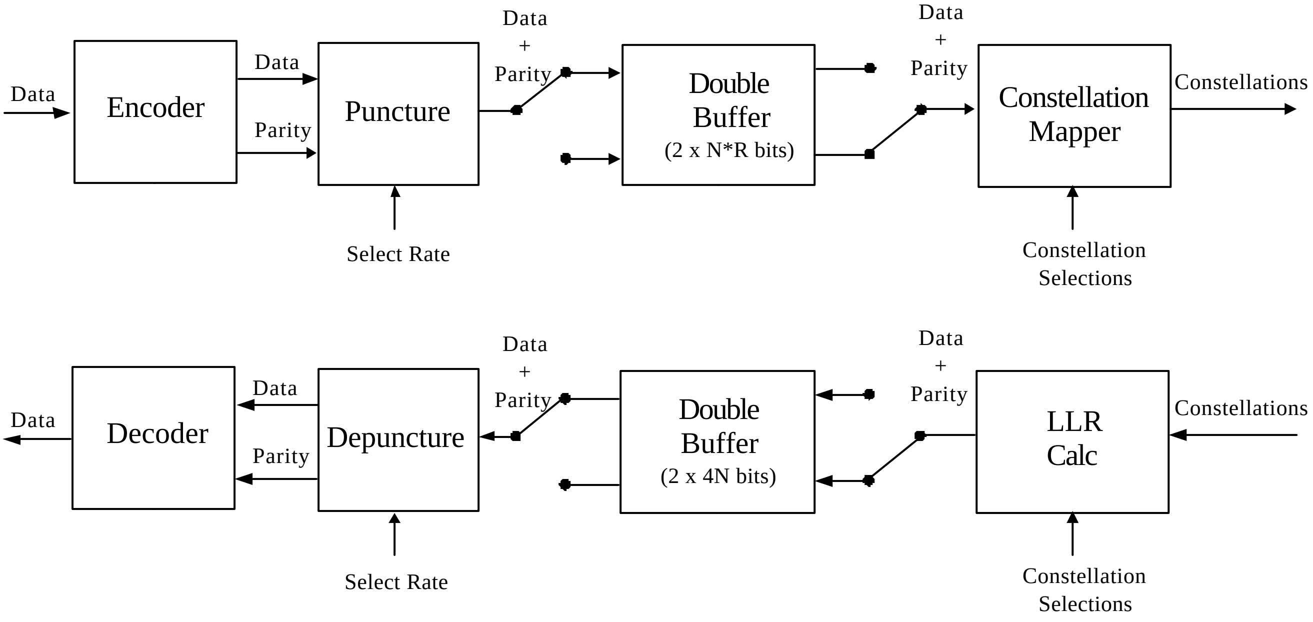

Figure 4 – Constant Code Rate Block Diagram

Another problem of the scheme is that the maximum throughput is limited by the code rate selected. If a low code rate is selected to perform well in low SNR channels, such as rate 2/3, then given a maximum constellation size of 16384QAM, the maximum spectral efficiency is only 9.33 bps/Hz. To alleviate this problem and provide a good range of throughputs requires the implementation of a number of code rates, with the code rate selected upon initialisation of the channel. Implementing various code rates does not represent a difficult problem as effectively only the puncture/depuncture circuits and corresponding buffer sizes need to be flexible. However it does add a further complexity to the initialisation process, which would need to consider all of the possible code rates when assigning constellations.

5 LATENCY ISSUES

As turbo coding is block based, it is important to calculate the resultant latency of each scheme. Figure 5 shows the basic components that make up the transmission latency. For comparison purposes, processing components B) and D) are not considered as they are implementation dependent. Component C) constitutes the main latency and can be calculated as the time to transmit 1N (where N = turbo code block size), which is simply the block size divided by the channel throughput. Component A) is also not considered as it is assumed that data is presented to the transmitter instantaneously. However if the data is presented at the same rate as the channel throughput, then an additional 1N must be added to the total latency. Table 1 provides a list of example latencies for various throughputs and block sizes.

Given that the latency is inversely proportional to the throughput of the channel, it is likely a system will need to support a set of block sizes in order to satisfy latency requirements. Table 1 shows that if throughputs ranging from 384 kbit/s to 6 Mbps, the resultant latency ranges from 2.6 ms to 40.3 ms for N = 15488. However the maximum latency would be reduced to 5.2 ms if a block size of 2048 is utilised.

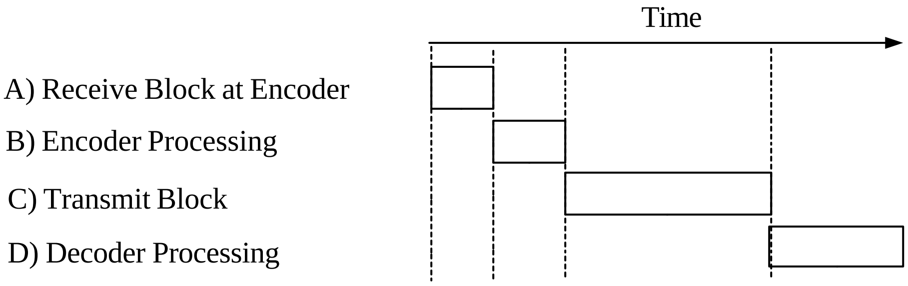

Figure 5 – Latency Components

Note that Figure 5 does not indicate the effect of double buffering. At the decoder, no additional latency results due to double buffering as once a complete block has been received, the decoder is activated and processes the block whilst the receiver stores the received data for the next block in the alternate buffer. However at the encoder, additional latency may result depending on how much overlap is implemented. An efficient implementation would encode the next block at the last required moment and thus would not result in additional latency.

Table 1 - Example Latencies

Throughput	**Block Size N**	**1N Latency**
384 kbps	2048	5.2 ms
384 kbps	8192	21.3 ms
384 kbps	15488	40.3 ms
2 Mbps	2048	1.0 ms
2 Mbps	8192	4.1 ms
2 Mbps	15488	7.7 ms
6 Mbps	2048	0.3 ms
6 Mbps	8192	1.4 ms
6 Mbps	15488	2.6 ms

5.1 Multiple Latency Paths

Given the block based nature of turbo coding, it is not possible to support multiple latency paths within the one turbo code. However this does not preclude the use of multiple latency paths. As detailed in Figure 6-6 of [4], it is possible to have completely separate latency paths each having their own encoder, with each assigned a portion of the available tones. Using this model combined with the schemes presented above, it is possible to support various latencies by implementing separate turbo encoder/decoder pairs, each utilising a different block size. Such a scheme is presented in a simplified block diagram of Figure 6.

In addition, it is possible to vary the block size of each turbo code at run-time. This flexibility allows a large range latencies to be supported for only a small number of code paths implemented if block sizes are made adaptive to the latency requirements. Varying block sizes can be performed on a per block basis if the corresponding block size is transmitted with the block, allowing the system to respond to latency requirements quickly.

Given the latencies presented in Table 1, it would be possible to support a low latency path for speech for certain throughputs. However if close to zero latency is required for speech, uncoded constellations can used and transmitted in parallel to the coded transmission. Implementing multiple turbo decoders in the receiver is naturally suited to a hardware implementation due to the parallel processing required. However if a simple code such as the 4-state triple code is utilised, multiple decoders can cost effectively be implemented in the one chip.

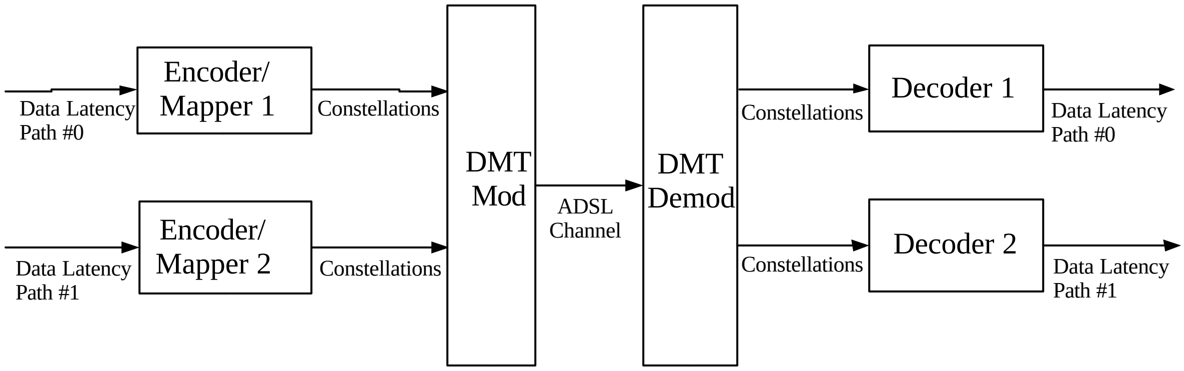

Figure 6 – Dual Latency Path Implementation

6 SIMULATION RESULTS

The aim of these simulations are to:

- Determine the AWGN flat SNR performance of the 4-state triple code for spectral efficiencies of 4 and 12 bps/Hz for comparitive purposes.
- Compare the performance of the constant and variable code rate schemes for both flat and non-flat SNR channels.
- Determine the maximum throughputs achieveable and calculate the associated coding gains for the non-flat SNR channels.
- Determine the impulse noise immunity with and without an additional interleaver.

All simulations have been performed using

- a simplied ADSL channel, treating each tone as a separate AWGN channel, orthogonal from the other channels.
- a floating point version of the turbo decoder.
- no fine gain control.

All simulation results, unless otherwise mentioned, have been achieved using at least 100 error events.

6.1 FLAT SNR RESULTS

Using flat SNR AWGN channels, performance was tested for both constant code rate and variable code rate systems. Table 2 presents the best constellation and code rate combinations for a large range of spectral efficiencies for the variable code rate scheme. This table includes the Eb/No and corresponding SNR, within +/- 0.05 dB, required to achieve BER = 10^{-7} for 4 iterations of the decoder. Tables 3 and 4 provide similar results but for the constant code rate scheme using code rates 2/3 and 4/5. Note due to the number of results required, these measurements do not satisfy the coding guidelines of 100 error events and thus can only be considered as estimates. To compare the two schemes, results from all these tables have been plotted in Figure 7. It can be seen there is little difference in performance between the two schemes, although constant rate 2/3 diverges slightly at high spectral efficiences. However it can also be seen from both the tables and the graph that the variable rate scheme provides a larger range of spectral efficiences and thus can make better use of tones available.

In addition to the table of results, BER performance curves were also obtained for the 4 bps/Hz and 12 bps/Hz to allow comparisons with previous proposals to be made. Figure 8 shows the 4 bps/Hz performance using a Rate 2/3 64QAM scheme, using the 4-state triple code, with block size N=15488, across 8 iterations. Figure 5 shows the performance using a Rate 6/7 16384 QAM scheme, N-15488, to achieve 12 bps/Hz. Both sets of curves use the variable code rate scheme to make use of the controlled protection level assignments for each bit. These results show comparable performance when compared to previous proposals, although it is difficult to make direct comparisons due to differences in block sizes and complexity.

Table 2 - Constellation and Code Rate Performance for Variable Code Rate Scheme, N=15488

Spectral Eff (bps/Hz)	**Constellation**	**Code Rate**	**~Eb/No@10^{-7} (dB)**	**SNR (dB)**
0.5	QPSK	1/ 4	0.9	-2.1
1	QPSK	1/ 2	1.6	1.6
2	16QAM	1/ 2	3.7	6.7
3	16QAM	3/ 4	5.9	10.7
4	64QAM	2/ 3	8.2	14.2
5	256QAM	5/ 8	11.1	18.1
6	256QAM	3/ 4	13.3	21
7	1024QAM	7/10	16	24.5
8	1024QAM	4/ 5	18.2	27.2
9	4096QAM	3/4	21.1	30.6
10	16384QAM	5/ 7	24.2	34.2
11	16384QAM	11/14	26.5	36.9
12	16384QAM	6/ 7	28.8	39.6
13	16384QAM	13/14	30.9	42

Table 3 - Constant Rate 2/3 Performance across Constellations, N=15488

Spectral Eff (bps/Hz)	**Constellation**	**~Eb/No@10^{-7} (dB)**	**SNR (dB)**
1.3	QPSK	2.4	3.6
2.7	16QAM	5.2	9.5
4.0	64QAM	8.7	14.7
5.3	256QAM	12.1	19.4
6.7	1024QAM	16.1	24.3
8.0	4096QAM	20.5	29.5
9.3	16384QAM	24.1	33.8

Table 4 - Constant Rate 4/5 Performance across Constellations, N=15488

Spectral Eff (bps/Hz)	**Constellation**	**~Eb/No@10^{-7} (dB)**	**SNR (dB)**
1.6	QPSK	3.2	5.1
3.2	16QAM	6.5	11.6
4.8	64QAM	10.2	17.0
6.4	256QAM	14.2	22.3
8.0	1024QAM	18.5	27.5
9.6	4096QAM	22.8	32.6

11.2	16384QAM	27.3	37.8

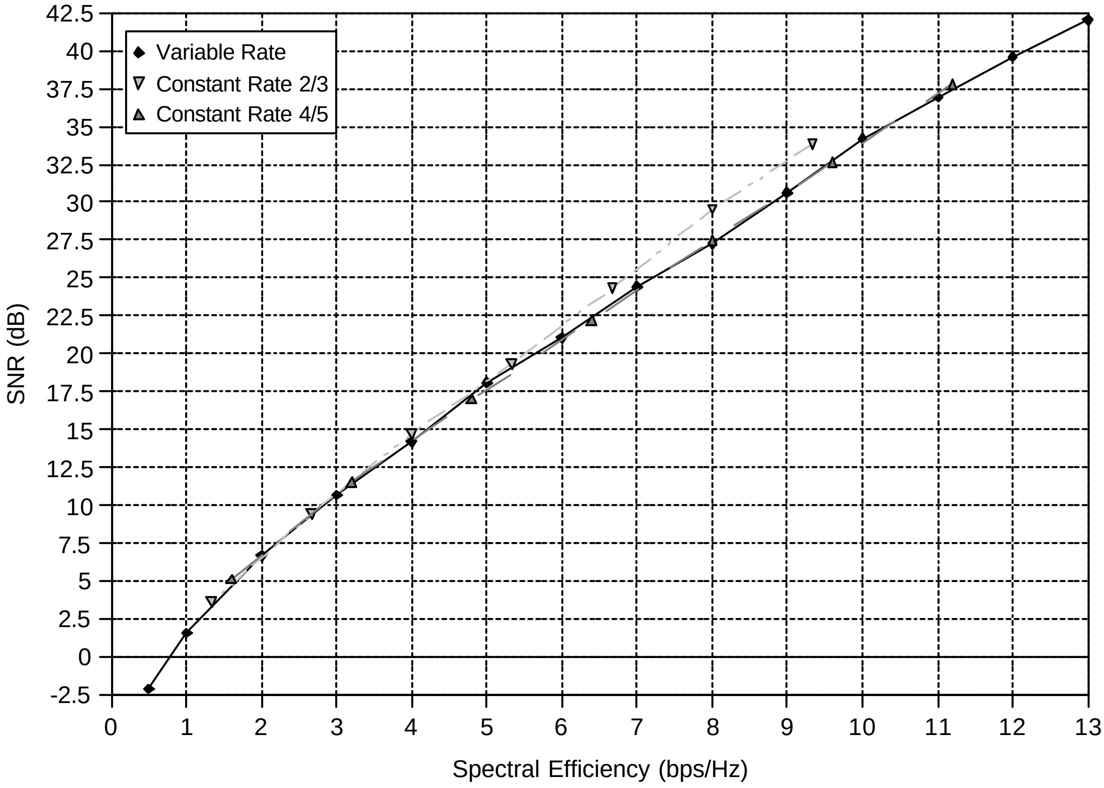

Figure 7 – Comparison of the SNR required to achieve BER=10^{-7}, 4 iterations against Spectral Efficiency

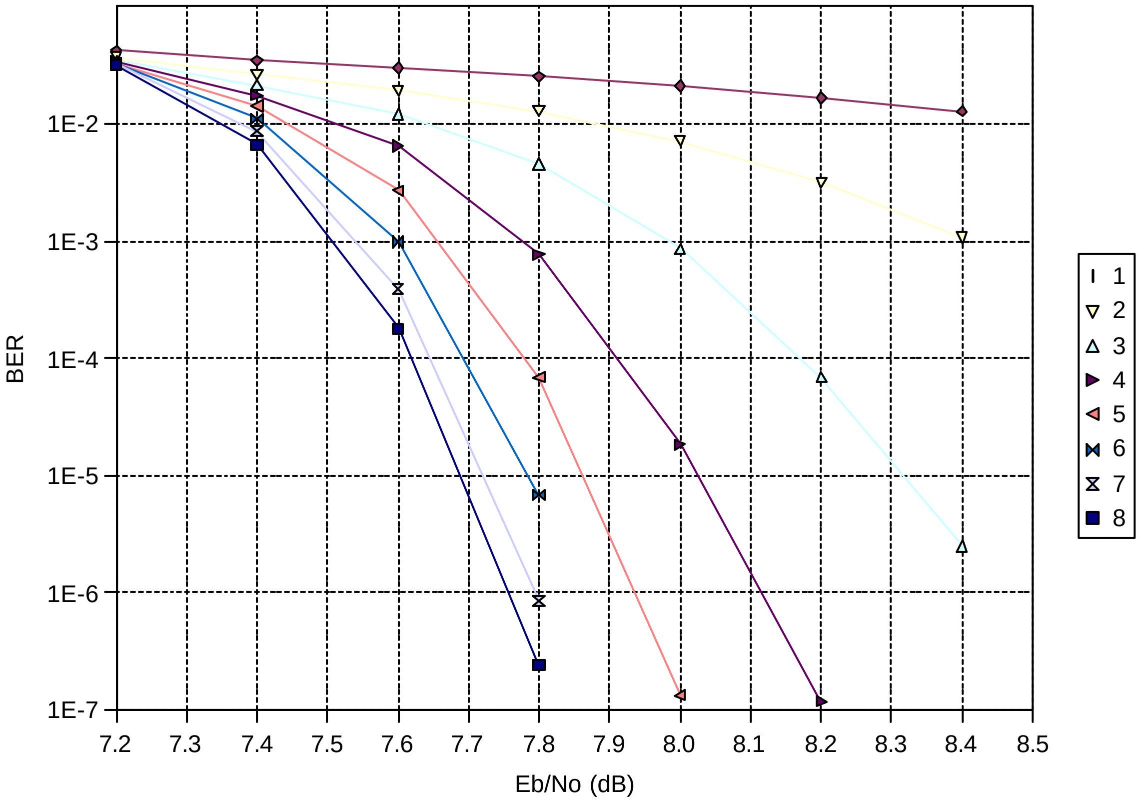

Figure 8 – BER Performance for 64QAM Rate 2/3 (4 bps/Hz), N=15488

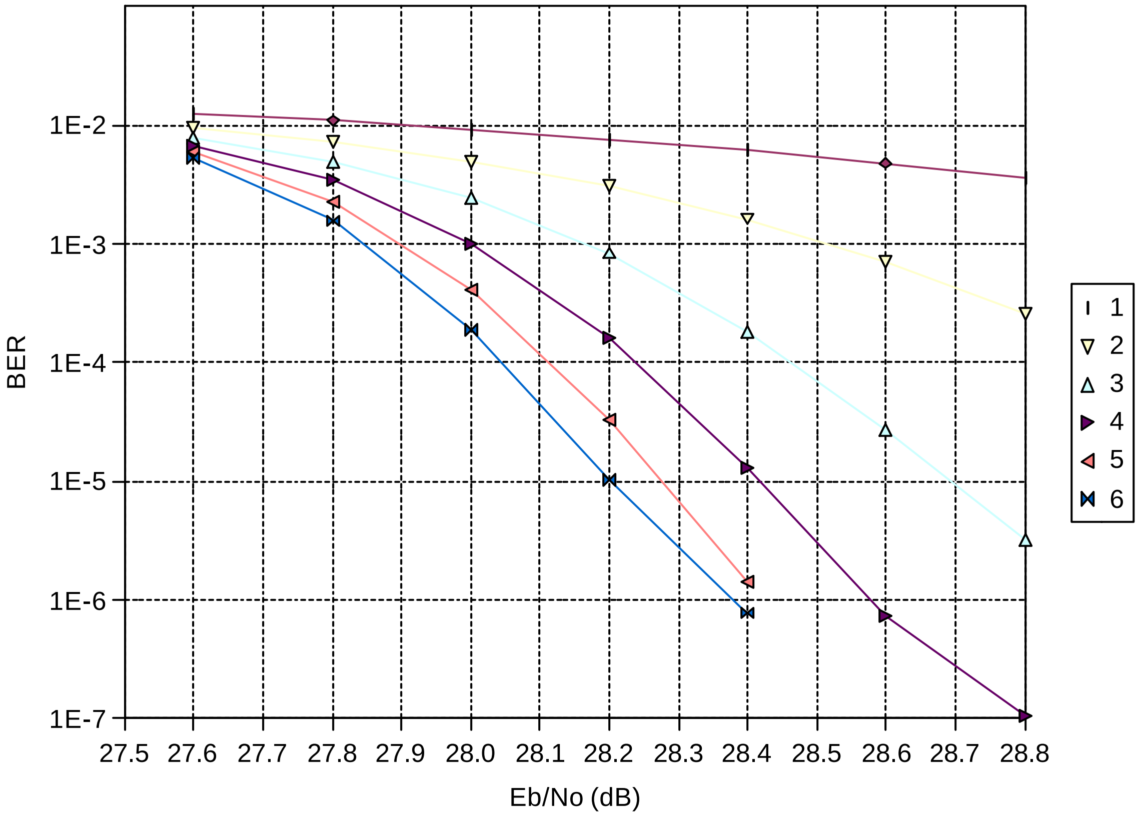

Figure 9 – BER Performance for 16384QAM Rate 6/7 (12 bps/Hz), N=15488

6.2 NON-FLAT SNR RESULTS

Using the results from Tables 2 to 4 as a basis, constellation/coding schemes were developed specifically for the 35 dB and 50 dB linearly decreasing channels, specified in [3]. The original thresholds from these tables were shifted down in SNR, increasing the throughput, until the desired BER = 10^{-7} performance was achieved at 4 iterations. The resultant SNR ranges are listed in Table 5 for constant rate schemes and Table 6 for the variable rate schemes. Note the constant rate schemes use different rates, thus the thresholds cannot be compared directly. However the variable rate schemes should use approximately the same threshold for each spectral efficiency, regardless of the channel. In comparing the 100 and 200 tones channel schemes, which were developed independently, thresholds are within 1dB of each other, meaning a universal scheme could be implemented with little or no loss.

It is important to note that one advantage of trellis based coding schemes, is the introduced correlation between data bits in the sequence. This correlation has the added benefit that when certain constellations receive greater SNR, which is the case when using a linearly decreasing SNR channel, that the surrounding bits in the trellis sequence benefit with improved error correcting ability. As a result if the code points derived from Tables 2 to 4 were shifted considerably to achieve BER = 10^{-7}, however further gains would be expected if fine gain control was utilised.

Table 7 provides the resultant throughputs achieved for each scheme presented in Table 5 and 6. From this table if can be seen that the variable code rate scheme achieves greater performance than the constant code rate scheme for both 35 and 50 dB channels, although the increase in throughput is only 4.7% and 3.4% respectively.

Coding gain is also provided for each scheme, calculated as the difference in SNR of an uncoded scheme at the same throughput using the same SNR slope, as detailed in [3]. Each uncoded scheme was developed in a similar manner to the coding schemes. The SNR at BER = 10^{-7} was calculated for each constellation and used as the initial SNR thresholds. These thresholds were then decreased until the desired throughput was achieved. It is important to note that these uncoded schemes do not take advantage of non-square constellations and the fine gain adjustment which are normally available. Therefore it would be expected that the coding gains would decrease slightly if these facilities were added, however this maybe offset by improved coding performance if such facilities were also added to the coding schemes.

Table 5 – SNR Thresholds for Constant Rate Schemes

Constellation	**SNR Range (dB) (Rate 2/3, 100 tone Channel)**	**SNR Range (dB) (Rate 4/5, 200 tone Channel)**
16384QAM	32.55 → 35.0	34.5 → 50.0
4096QAM	27.3 → 32.2	29.25 → 34.25
1024QAM	22.05 → 26.95	24.25 → 29.0
256QAM	17.5 → 21.7	18.75 → 24.0
64QAM	12.6 → 17.15	13.75 → 18.5
16QAM	6.3 → 12.25	8.25 → 13.5
QPSK	2.1 → 5.95	2.75 → 8.0

Table 6 – SNR Thresholds Used for Variable Rate Schemes

Constellation	**SNR Range (dB) (100 tone Channel)**	**SNR Range (dB) (200 tone Channel)**
16384QAM, Rate 13/14		42.75 → 50.0
16384QAM, Rate 6/7		38.25 → 42.5
16384 QAM, Rate 11/14		35.25 → 38.0
16384QAM, Rate 5/7	33.25 → 35.0	32.35 → 35.0
4096QAM, Rate 3/4	29.75 → 32.9	29.25 → 32.0
1024QAM, Rate 4/5	26.25 → 29.4	25.75 → 29.0
1024QAM, Rate 7/10	24.15 → 25.9	23.5 → 25.5
256QAM, Rate 3/4	19.95 → 23.8	19.5 → 23.25
256QAM, Rate 5/8	16.8 → 19.6	16.25 → 19.25
64QAM, Rate 2/3	12.6 → 16.45	12.75 → 16.0
16QAM, Rate 3/4	9.8 → 12.25	8.75 → 12.5
16QAM, Rate 1/2	4.2 → 9.45	5.25 → 8.5
QPSK, Rate 1/2	1.4 → 3.95	1.25 → 5.0
QPSK, Rate 1/4	-2.45 → 1.05	-2.25 → 1.0

Table 7 – Measured Throughput for 35 dB & 50 dB Channels, @ BER = 10^{-7}, for 4 Iterations

Scheme	35 dB, 100 tone Channel			50 dB 200 tone Channel		
	Throughput (kbps)	Code Rate	Coding Gain (dB)	Throughput (kbps)	Code Rate	Coding Gain (dB)
Variable Rate	2024	0.62	8.5	5828	0.7	8
Constant Rate	1936	0.67	7.8	5636	0.8	7.1

To show how the performance varies against the changes in throughput for a given channel, Figure 10 plots the BER against throughput for the constant code rate 4/5 scheme, using the 50 dB non flat SNR channel. This graph plots performance curves for iterations 1 to 6. Note all measured points below 10^{-7} have been measure with less than 100 error events. It can be seen from this graph that increasing the computation by 50 % from 4 iterations to 6 iterations results in a throughput increase from 5635 kbps to 5710 kbps (at BER = 10^{-7}), an increase of only 1.3 %. This does not represent a good return for the extra computation. However overall it can be seen that good performance is achieved for iterations 3 to 6, allowing some flexibility in choosing the number of iterations for a particular implementation.

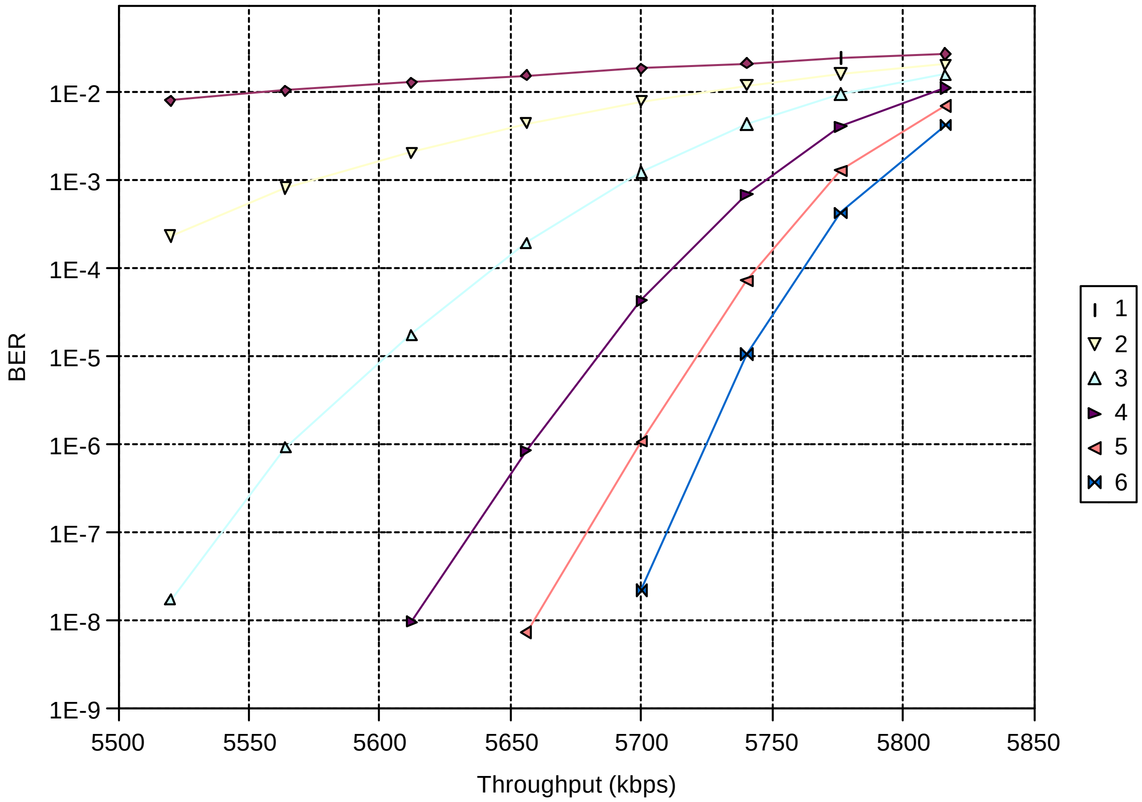

Figure 10 – BER Performance against throughput for various iterations of the constant rate 4/5 scheme (50 dB non flat SNR channel)

6.3 IMPULSE NOISE RESULTS

Using the guidelines set in [1], burst error performance was tested for two spectral efficiencies; 4 bps/Hz and 12 bps/Hz given flat SNR channels, with SNR = 21.5 dB and 45.5 dB respectively. The chosen schemes for testing were

- 64QAM Rate 2/3 using the variable code rate bit assignments, N=15488, for 4 bps/Hz,
- 16384QAM Rate 6/7, using the variable code rate bit assignments, N=15488, for 12 bps/Hz.

With no guidelines specifying the number of impulse noise events per code frame, all tests have been performed using one impulse event per code frame, with the location of the impulse noise uniformly distributed across the frame, except for the extents of the frame when the impulse event would have stretched over two code frames.

Since both variable rate and constant rate schemes retain the bit ordering for data and parity, they can be susceptible to burst errors. If a symbol or two experiences an impulse of noise, a large section of the trellis becomes in error. Given the 2nd or 3rd constituent code parity sequences use an interleaved version of the data sequence, they will assist in correcting these errors, however it is possible to improve the situation by interleaving the all important data bits prior to constellation mapping. Performance results shown in Table 8 and Figures 11 to 12 show considerable gain when the data bits are interleaved. Since the parity bits are also in sequence it is likely that additional gains are possible if all the bits are interleaved, however this has been left open to further work. In comparison to previous proposals, the results here clearly outperform the TCM+RS scheme of [7] and the combined uncoded/TTCM(PCCC) scheme of [8]. Comparible performance is achieved to the turbo code + Reed Solomon results, presented in [9], without the need of a Reed Solomon outer code.

Further tests are required to determine the impulse noise performance for smaller and possibly larger block sizes.

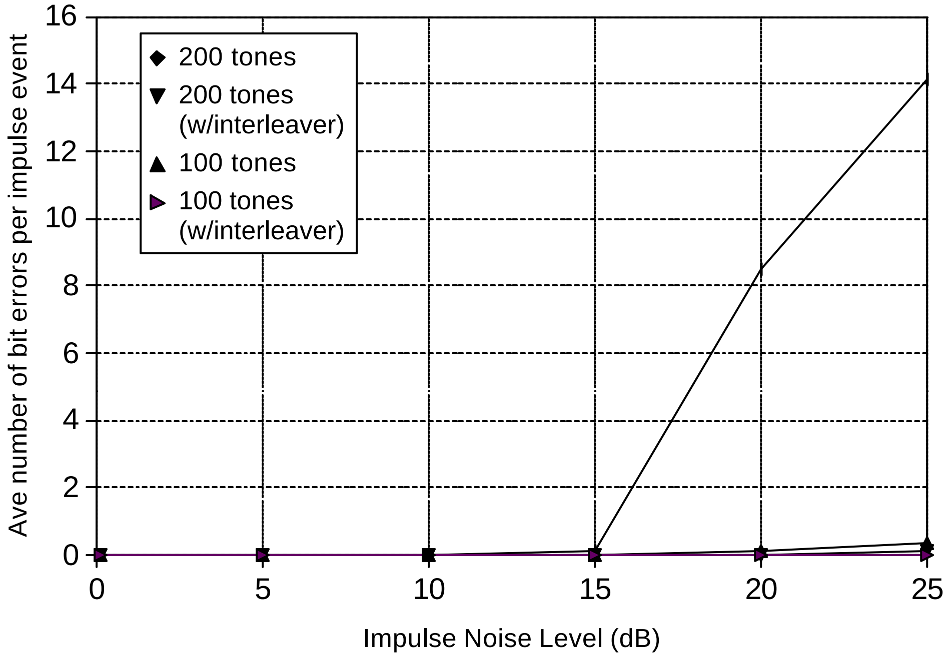

Figure 11 – Average Number of Bit Errors per Burst for 4 bps/Hz

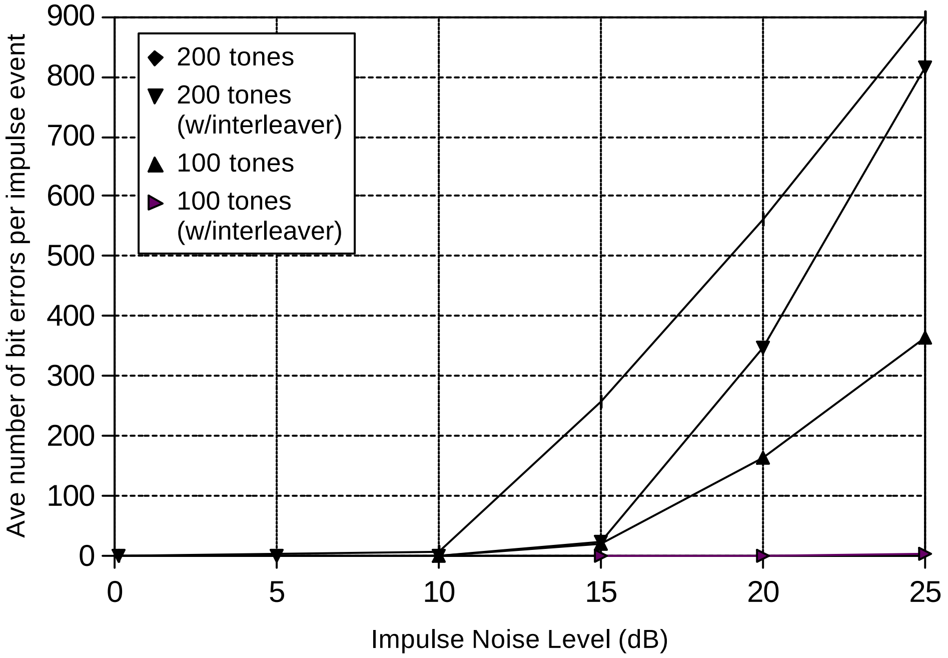

Figure 12 – Average Number of Bit Errors per Burst for 12 bps/Hz

Table 8 – Average Number of Bit Errors per Burst for 4 and 12 bps/Hz, N=15488

Scheme	**Tones**	**Spectral Eff (bps/Hz)**	**Impulse Noise Level**					
			0dB	**5dB**	**10dB**	**15dB**	**20dB**	**25dB**
Var Rate	100	4	0	0	0	0	0	0
Var Rate (w/int)	100	4	0	0	0	0	0.1	0.3
Var Rate	200	4	0	0	0	0.13	8.5	14.1
Var Rate (w/int)	200	4	0	0	0	0	0	0.1
Var Rate	100	12	0	0	0	21.6	165	365
Var Rate (w/int)	100	12	0	0	0	0.1	1.4	2.8
Var Rate	200	12	0	0	6	258	563	899
Var Rate (w/int)	200	12	0	0	0	25	350	815

7 CONCLUSIONS

This proposal has presented two viable options for implementing a turbo coding scheme for the ADSL DMT channel. Although the variable code rate scheme has shown to achieve a greater throughput of 3-5 % for the channels tested, it requires a slightly more complex implementation than the constant code rate scheme. However further tests are required to determine how each scheme fares using the smaller block sizes and with a larger range of non-flat SNR channels.

Alternative turbo coding schemes, such as the one presented in [5], encode the least protected bits of the constellations, with the remaining bits transmitted uncoded. This method achieves similar performance as the schemes presented in this proposal and with lower complexity due to smaller block sizes. Note it is still possible to implement such a scheme for a DMT channel, using a similar technique as the variable rate scheme presented. However we believe that encoding all of the bits potentially provides far greater protection against burst noise, without the need of an RS outer code and also achieves superior BER performance below 10^{-7}.

Each scheme has been tested using the triple 4-state PCCC turbo code with reliable results obtained. Although this code results in a slight loss in performance over traditional turbo codes given the same number of iterations, if comparing coding schemes with the same level of computation, the 4-state triple code easily exceeds the performance of 16-state double codes. In addition, due to the steepness of the performance curves, such differences in code performance only results in minimal differences in throughput. In addition, using the simpler decoder structure provides greater flexibility in implementation. This includes the possibility of implementing in software, if only a single coded latency path is required or alternatively providing greater flexibility in implementing multiple decoders in the one chip. It was shown that good performance can achieved with iterations ranging from 3 to 6, allowing a low complexity design to use less iterations and only suffer a small reduction in throughput.

8 FUTURE WORK

There is considerable scope to perform future work with the schemes presented in this proposal. In particula, future work will include:

- Implementation of the turbo coding schemes on a hardware testbed to improve testing speed and to calculate implementation loss, when using the variety of constellations.

- Testing of various block sizes to determine performance for both non-flat SNR and impulse noise channels.

- Utilisation of the fine gain control feature to improve the overall performance.

9 REFERENCES

[1] "G.gen: Coding Ad Hoc Report," ITU standard contribution, BA-108R1, Antwerp, Belgium, 19-23 June, 2000.
[2] "Report of the Ad Hoc on Improved Coding Gain," ITU standard contribution, BI-110, Goa, India, 23-27 October, 2000.
[3] "G.gen: G.dmt.bis: G.lite.bis: Proposal for Evaluation of Net Coding gain on a channel with non-flat SNR," ITU standard contribution, BI-116, Goa, India, 23-27 October, 2000.
[4] "G.lite.bis: Framer Text for Draft Recommendation," ITU standard contribution, BI-037, Goa, India, 23-27 October, 2000.
[5] "G.gen.bis: A Turbo TCM Scheme with Low Decoding Complexity," ITU standard contribution, BI-090, Goa, India, 23-27 October, 2000.
[6] L. Zhang, A. Yongacogl, "Turbo Coding in ADSL DMT Systems," 2nd International Symposium on Turbo Codes & Related Topics, Brest, Frame, 2000.
[7] "G.gen: G.dmt.bis: G.lite.bis: Performance of G.992.1 Trellis Coding with/without Reed-Solomon Coding in BA-108R1 format," ITU standard contribution, BI-092, Goa, India, 23-27 October, 2000.
[8] "G.gen: Proposal of Decision making process for Turbo Coding & report of performance evaluation of proposed TTCM(PCCC) with R-S code and without R-S code," ITU standard contribution, BI-093, Goa, India, 23-27 October, 2000.

[9] "G.gen: Results of the requirements requested in the Coding Ad hoc report (BA-08R1) for the proposed Turbo Codes for ADSL modems by VoCAL Technologies Ltd in BA-020R1," ITU standard contribution, HC-073, Huntsville, Canada, 31 July – 4 August, 2000.

[10] S. Benedetto, G. Montorsi, "Versatile Bandwidth-efficient Parallel and Serial Turbo ?????? Coded Modulation," 2nd International Symposium on Turbo Codes & Related Topics, Brest, Frame, 2000.