REPEAT-PUNCTURED SUPERORTHOGONAL CONVOLUTIONAL TURBO CODES ON AWGN CHANNELS

N. Pillay, H. Xu and F. Takawira

School of Electrical, Electronic and Computer Engineering, University of Kwa-Zulu Natal, Durban, 4000, South Africa

Abstract: The performance of turbo codes has been shown to be near the theoretical limit in the additive white Gaussian noise channel. By using orthogonal signaling, which allows for bandwidth expansion, the performance of the turbo coding scheme can be improved even further. Since this is a low-rate code, the code is mainly suitable for spread-spectrum modulation applications. In classical turbo codes the frame length is set equal to the interleaver size, however, the codeword distance spectrum of turbo codes improves with an increasing interleaver size. It has been reported that by using repetition and puncturing the performance of turbo codes can be improved. Repeat-Punctured Turbo Codes has shown a significant increase in performance at moderate to high signal-to-noise ratios. In this paper, we study the use of orthogonal signaling and parallel concatenation together with repetition and puncturing to improve the performance of superorthogonal convolutional turbo codes for reliable and effective communications. Simulation results for the additive white Gaussian noise channel are presented together with analytical upper bounds, which have been derived using transfer function bounding techniques.

Key words: orthogonal, repetition, puncturing, turbo codes, parallel concatenation.

1. INTRODUCTION

Multipath fading, interference and noise are all deleterious inputs to signal fidelity at the receiver. In a digital communication system this signal degradation translates to abundance in bit errors. If these bit errors are not controlled then the errors would render the system useless. Fortunately, the information bits transmitted over a digital wireless link can be protected by channel coding. A turbo code can be visualized as a "two-dimensional" or a "multi-dimensional" forward error-correcting convolutional code.

The performance of turbo codes have been reported to be near the Shannon limit [1-5, 7, 12]. Typically for large frame lengths (N=16384) and with iterative decoding, E_b / N_o values of -0.15 dB at a bit-error rate level of 10^{-3} have been reported [11]. At the expense of bandwidth expansion, the performance can be improved even further. For a frame length of N=4000, E_b / N_o values of -0.5 dB has been reported at a bit-error rate (BER) of 10^{-3} with superorthogonal convolutional turbo codes (SCTC), which, is a low-rate code that achieves good distance properties [8, 11]. Due to the resultant low-rate code, these codes are mainly suited for spread-spectrum applications. In the conventional turbo codes the frame length is set equal to the interleaver size [1-6, 12], however, if the information frame length is increased then the resultant performance increases due to a larger interleaver size. This makes it difficult to use turbo codes for, e.g. voice transmission, since the inherent delay would be too great. Repeat-punctured turbo codes (RPTCs) proposed in [13] make use of interleavers of sizes larger than the information frame length and show a significant increase in performance at moderate to high SNRs. For N=1024, simulation results have shown that the RPTC has approximately 1.5 dB coding gain over the conventional turbo codes at a bit-error rate of 10^{-3} [13]. Motivated by Kim *et al* [13], we propose repeat-punctured superorthogonal convolutional turbo codes.

In this paper, we study the use of orthogonal signaling and parallel concatenation together with repetition and puncturing to improve the performance of superorthogonal convolutional turbo codes for reliable and effective communications. Simulation results for the AWGN channel are presented together with analytical upper bounds, which have been derived using transfer function bounding techniques presented/used in [14-16].

The paper is structured such that, in Section 2, we discuss the principals of Repeat-Punctured Superorthogonal Convolutional Turbo Codes (RPSCTCs). Section 3 describes the advantages and disadvantages of the Max-log-MAP algorithm, Section 4 presents the performance analysis and the simulation results on the AWGN channel. Finally, conclusions are drawn in Section 5.

2. REPEAT-PUNCTURED SUPERORTHOGONAL CONVOLUTIONAL TURBO CODES

2.1 Encoder

Superorthogonal convolutional turbo codes have been introduced by Komulainen and Pehkonen [8, 11]. These codes utilize the parallel concatenated structure of the conventional turbo codes, but instead make use of orthogonal signals, and are low-rate codes that achieve good distance properties by exploiting bandwidth expansion. If we enlarge an orthogonal signal set to incorporate their complementary sequences we then have a bi-orthogonal signal set [8, 11, 17] with twice the number of members but for the same sequence length. For SCTCs, with BPSK signalling, all encoder output sequences that leave the same state or merge the same state are antipodal. In other words, only one sequence and its complement are needed for every two states of the trellis. This results in a better performance but at the expense of an increase in complexity. There exist various methods for obtaining the orthogonal signal sets. One of the possible options is the Walsh functions obtained from Hadamard matrices [11].

In the conventional turbo codes the information frame length is set equal to the interleaver size [1-6, 7, 12]. The literature shows [1, 2, 4, 5, 10] that the performance of the conventional turbo code improves with an increase in frame length or consequently interleaver size. However, certain applications, e.g. real-time communications would not tolerate very large frame lengths, since larger frame lengths correspond to larger processing delays. Repeat-punctured turbo codes presented by Kim *et al* [13] uses a repeat-puncture mechanism that allows for the use of larger interleavers and constant frame lengths in the same system. Motivated by RPTCs, we investigate repeat-punctured superorthogonal convolutional turbo codes on the AWGN channel.

The structure of the repeat-punctured superorthogonal convolutional turbo code (RPSCTC) encoder is shown in Figure 2 with the superorthogonal convolutional turbo code (SCTC) encoder structure depicted in Figure 1. The form is that of a parallel concatenated code (PCC) where, two constituent encoders, superorthogonal recursive convolutional code (SRCC) 1 and SRCC 2 are linked together by an interleaver π.

Let the number of information bits/frame be N. The encoder for the first constituent code is the same as in superorthogonal convolutional turbo codes studied by others [8, 11]. The structure of any one constituent code for memory, m=4, is presented in Figure 3, employing feedback polynomial $g(D) = 1 + D^3 + D^4$. Studies show [8, 11] that if the memory of the code is m then the length of the output sequence from SRCC 1 is $2^{m-1}N$. The code rate for such a constituent code is $1/2^{m-1}$. SRCC 2 differs a little in RPSCTC from the normal SCTC scheme.

This is because prior to interleaving, to allow for a larger interleaver size, each input information bit is repeated L times thus rendering a larger frame, i.e., LN, to the

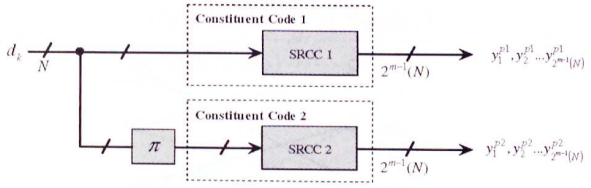

Figure 1: SCTC encoder structure.

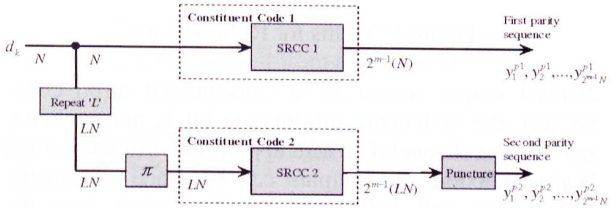

Figure 2: RPSCTC encoder structure.

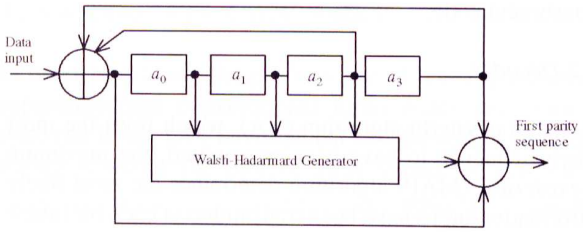

Figure 3: RPSCTC constituent encoder structure, memory, m=4, rate, R=1/8

interleaver and the constituent encoder 2. This results in a sequence of length $2^{m-1}LN$ at the output of SRCC 2. This sequence needs to be punctured to control the overall code rate of the encoder. A simple puncturing pattern was considered, where, the first n bits (second bit) out of every L bits in the second parity sequence, were transmitted. A value of L=2 was used in this paper. L can be increased to lower the rate of the code or to create a series of rate-compatible codes, however, as L increases excessive puncturing would need to be applied and as a result, the BER performance will decrease.

Corresponding to an input information stream, $d = d_1, d_2, d_3, ..., d_N$ the output from SRCC 1 is $y_1 = (y_1^{p1}, y_2^{p1}, ..., y_{2^{m-1}N}^{p1})$ and the output from the second constituent encoder prior to puncturing is $y_2 = (y_1^{p2}, y_2^{p2}, ..., y_{2^{m-1}LN}^{p2})$. The punctured sequence is then given by $y_2 = (y_1^{p2}, y_2^{p2}, ..., y_{2^{m-1}N}^{p2})$. The trellis for memory, m=2 is depicted in Figure 4, in which, the recursive nature of the code is clearly evident. In addition it can be seen that all encoder output sequences that either leave the same state or merge the same state are antipodal.

In the case of RPSCTC and SCTC there is no common

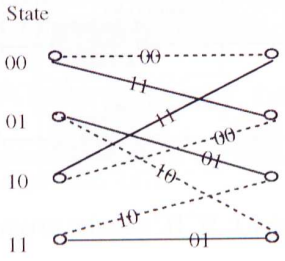

Figure 4: Trellis for RPSCTC, m=2.

channel output stream as in conventional turbo codes because the systematic information bit is not explicitly sent via the channel, but with appropriate mapping of the Walsh functions the outputs can be made systematic. Komulainen and Pehkonen [8, 11] explain how the recursiveness of a code can lead to it being systematic. Walsh functions obtained from Hadarmard matrices were used to generate the set of orthogonal sequences for each constituent code.

2.2 Decoder

Unlike the Viterbi algorithm (VA), which finds the most likely sequence to have been transmitted, the maximum *a-posteriori* (MAP) algorithm determines the most likely information bit to have been transmitted at each bit time k [9]. Using *a-posteriori* probabilities (APP), for small probability of bit-error (P_b) the performance difference between MAP and soft-output Viterbi algorithm (SOVA) is very small. However, at low signal energy to noise ratios (E_b / N_o) and high P_b values MAP outperforms SOVA [12]. The MAP algorithm proceeds somewhat like the VA, but in two directions, forward and backward, over a block of code bits. Assuming an AWGN channel we start with the log-likelihood ratio (LLR) $L(\hat{d}_k)$,

$$L(\hat{d}_k) = \log\left[\sum_v \lambda_k^{1,v}\right] - \log\left[\sum_v \lambda_k^{0,v}\right] \qquad (1)$$

where,

$$\left[\sum_v \lambda_k^{1,v}\right] = \sum_v \alpha_k^v \delta_k^{1,v} \beta_{k+1}^{f(1,v)}, \qquad (2)$$

and

$$\left[\sum_v \lambda_k^{0,v}\right] = \sum_v \alpha_k^v \delta_k^{0,v} \beta_{k+1}^{f(0,v)}. \qquad (3)$$

(2) represents the conditional probability $P(d=1|x)$ and (3) represents the conditional probability $P(d=0|x)$, where x is the bit received at time k. α_k^v, $\delta_k^{1,v}$ and $\beta_{k+1}^{f(1,v)}$ are the forward-state metric, reverse-state metric and branch metric respectively for $P(d=1|x)$, where v is the encoder state at bit time k and $f(1,v)$ is the next state also for $d=1$. The decoding structure as shown in Figure 5 makes use of an iterative technique. An outer decoder and an inner decoder both in co-operation produce soft decisions which are mutually exchanged, thus increasing the reliability of the decisions in the next iteration. At the output of Decoder 2 LN soft decisions are computed but only N decisions are needed. To make this of good use, each pair of decisions is averaged to produce N soft decisions.

In the first step of the decoding the corrupted first parity sequence from the channel is sent to Decoder 1 to produce N LLR (soft) estimates of the sequence using no *a-priori* information (This is because the probability of receiving a '1' or a '0' is equal at this stage). Using Equation 4 the extrinsic values are computed then repeated L times and permuted by the same interleaver mapping at the encoder and then sent to Decoder 2 as *a-priori* information. This sharing of information to improve the decisions is somewhat like co-operation. Decoder 2 then produces LN extrinsic decisions as described by Equation 5.

$$L_{e1}(d_k) = L_{decoder1}(d_k) - L_c(x_k) - L_a(d_k) \qquad (4)$$

$$L_{e2}(d_k) = L_{decoder2}(d_k) - L_c(x_k) - L_{e1}(d_k) \qquad (5)$$

These LN extrinsic decisions are de-interleaved and averaged and N soft extrinsic decisions are sent to Decoder 1 as *a-priori* information. This process is iterated several times after which LN LLRs are taken from Decoder 2, de-interleaved, then averaged (Equation 6) to produce N LLRs and compared to a zero-threshold to yield hard decisions. The sign of the LLR represents the hard decision and the amplitude of the LLR represents the degree of reliability. Hard decisions are never used as input to a decoder since this will degrade performance drastically [12].

$$L_{e,i_{average}} = \frac{L_{e,2i} + L_{e,2i+1}}{L}, \quad i=0, 1\ldots N-1 \qquad (6)$$

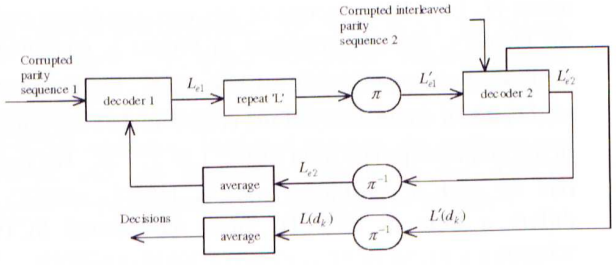

Figure 5: RPSCTC decoder structure.

3. MAX-LOG MAP ALGORITHM

Although the MAP algorithm yields the best overall performance in the decoding of turbo codes [7, 12], the implementation of RPSCTC in hardware would lead many problems because of the increase in complexity. When we use the maximization approximation instead, for the forward and backward recursions, i.e., α_k and β_k, this leads to an approximation error in the computation of these variables [9]. For high SNRs this error is comparable to noise, thus degrading performance, and for low SNRs it is much less than the noise power. Thus by using the Max-Log-MAP algorithm we are settling for a degraded performance but a significantly lower complexity for a hardware implementation.

4. PERFORMANCE ANALYSIS

3.1 Analytical BER Bounds

Transfer function bounding techniques studied by Divsalar *et al* [14] were used to derive the BER bound for RPSCTC. Starting with the state transition matrix shown in Equation 7, [14, 16]

$$A(L,I,D)=\begin{pmatrix} L^l I^i D^{d_{0,0}} & \cdots & L^l I^i D^{d_{0,j}} \\ \vdots & \ddots & \vdots \\ L^l I^i D^{d_{z,0}} & \cdots & L^l I^i D^{d_{z,j}} \end{pmatrix} \quad (7)$$

where, in a monomial $L^l I^i D^d$, l is always equal to 1 for a single input bit, and i and d are either zero or one depending on the input and output weights respectively for the z^{th} to the j^{th} state of the state diagram. For an encoder with 2^m states with the last m edges as termination edges, the generating function is given by Equation 8.

$$T(L,I,D)=\sum_{l\geq 0}\sum_{i\geq 0}\sum_{d\geq 0} L^l I^i D^d t(l,i,d) \quad (8)$$

$$\text{Since } I+A+A^2+A^3+\ldots=(I-A)^{-1} \quad (9)$$

The transfer function for each constituent encoder can be expressed in the form of Equation 10.

$$T(L,I,D)=[(I-A(L,I,D))^{-1}]_{0^m,0^m} \quad (10)$$

Denote the probability of producing a codeword fragment of weight d given a randomly selected input sequence of weight i by Equation 11 for constituent encoder 1 and Equation 12 for constituent encoder 2.

$$p(d_1 \mid i)=\frac{t(N,i,d_1)}{\sum_{d_1} t(N,i,d_1)} \approx \frac{t(N,i,d_1)}{\binom{N}{i}}. \quad (11)$$

$$p(d_2 \mid i)=\frac{t(LN,Li,d_2)}{\binom{LN}{Li}}. \quad (12)$$

The bound can be obtained from Equation 13:

$$p_{bit} \leq \sum_{d=d_{\min}}^{n} \sum_{i} \sum_{d1} \sum_{d2} \frac{i}{N}\binom{N}{i} p(d_1 \mid i)p(d_2 \mid i)p_2(d) \quad (13)$$

$$\text{where } p_2(d) \leq Q\left(\sqrt{2dRE_b / N_o}\right) \quad (14)$$

3.2 Simulation on the AWGN channel

In order to evaluate the performance of repeat-punctured superorthogonal convolutional turbo codes simulations were run in the C++ environment and the results plotted. Figure 6 shows the plot of BER versus E_b / N_o for memory, $m=4$, for SCTC and RPSCTC, which corresponds to the code rate $R=\frac{1}{15}$. This code rate is due to the first bit of the Walsh sequences being punctured in the second parity sequence from the encoder. The union bound is also shown in Figure 6. A similar plot for SCTC versus RPSCTC with memory depth $m=2$ with corresponding rate $R=1/3$ is depicted in Figure 7.

A uniform interleaver was chosen and a straightforward puncturing pattern was used although other extravagant patterns could be investigated to yield better performances. A frame length of N=200 was chosen and the stopping criterion was set at 30 frame errors. 18 iterations were chosen for the decoders although this could be decreased at higher SNRs without a compromise in the simulation accuracy. The RPSCTC scheme exhibited a higher computational complexity than SCTC in the simulation; this is mostly due to the second decoder handling an input sequence twice the size than that of the input sequences to the decoders in superorthogonal convolutional turbo codes. It can be seen from Figure 6 and Figure 7 that approximately 0.5dB of coding gain is obtained for RPSCTC compared to SCTC at a BER level of 10^{-3} and an approximately 1.5dB coding gain is achieved at a BER level of 10^{-6}. It can also be seen that the simulation results converge to the analytical bounds.

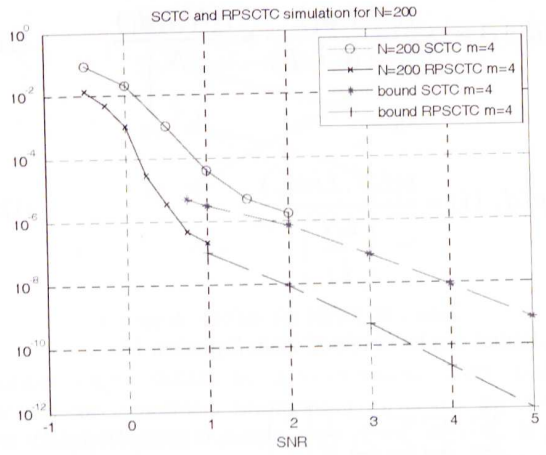

Figure 6: Simulation for SCTC and RPSCTC in the AWGN channel, N=200, m=4, R=1/15, including the analytical bounds.

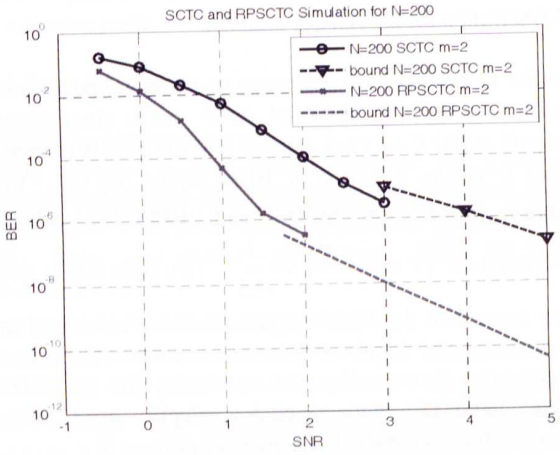

Figure 7: Simulation for SCTC and RPSCTC in the AWGN channel, N=200, m=2, R=1/3, including the analytical bounds

5. CONCLUSION

In this paper we have shown how repetition and puncturing can be used to improve the performance of superorthogonal convolutional turbo codes in an AWGN channel. The quality of service (QoS) was presented in terms of the BER and the increase in performance between the scheme presented and SCTC was evident. The concreteness of the transfer function bounding technique was also proved in the derivation of the bound for the RPSCTC scheme.

6. REFERENCES

[1] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in Proc. IEEE Int. conf. Commun., ICC'93, Geneva, Switzerland, May 1993, vol. 2, pp. 1064-1070.

[2] D. Divsalar and F. Pollara, "Multiple Turbo Codes for Deep-Space Communications," The Telecommunications and Data Acquisition Progress Report 42-121, January-March 1995, Jet Propulsion Laboratory, Pasadena, California, May 15, 1995, pp. 66-77.

[3] D. Divsalar, F. Pollara, "Serial and Hybrid Concatenated Codes with Applications," Jet Propulsion Laboratory, California institute of Technology.

[4] A. H. Mugaibel, M. A. Kousa, "Understanding Turbo Codes," King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia.

[5] Benedetto, Sergio and Montorsi, Guido, "Unveiling turbo-codes: some results on parallel concatenated coding schemes," IEEE Transactions on Information Theory, vol. 42, No. 2, March 1996, pp. 409-428.

[6] H. R. Sadjadpour, N. J. A. Sloane, M. Salehi, G. Nebe, "Interleaver Design for Turbo Codes," *IEEE Journal on Selected Areas in Communications*, Vol. 19, No. 2, May 2001, pp. 831-837.

[7] H. Taub, D. L. Schilling, Principles of Communication Systems, The City College of New York, 1986.

[8] P. Komulainen and K. Pehkonen, "A low-complexity superorthogonal turbo-code for CDMA applications," in Proc. IEEE Int. Symp. Personal, Indoor, mobile Radio Commun., PIMRC '96, Taipei, Taiwan, R.O.C.,Oct. 1996, vol. 2, pp. 369-373.

[9] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate" IEEE Trans. Inform. Theory, vol. IT-20, pp. 284-287, Mar. 1974.

[10] S. Benedetto and G. Montorsi , "Performance evaluation of parallel concatenated codes," in Proc. IEEE Int. Conf. Commun., ICC'95, Seattle,WA, June 1995, vol. 2, pp. 663-667.

[11] P. Komulainen and K. Pehkonen, "Performance evaluation of Superorthogonal Turbo Codes in AWGN and flat Rayleigh fading channels," in IEEE Journ. On Sel. Areas in Commun.,vol. 16, no.2, February 1998,pp. 196-205.

[12] B.Sklar, Digital Communications. Fundamentals and Applications. Beijing 2001

[13] Y. Kim, J. Cho, W. Oh and K. Cheun, "Improving the performance of turbo codes by repetition and puncturing," Division of Electrical and Computer Engineering, Pohang University of Science and Technology.

[14] D. Divsalar, S. Dolinar, and F. Pollara ,"Transfer Function Bounds on the Performance of Turbo Codes," TDA Progress Report 42-122, August 15, 1995, Communications Systems and Research Section, R. J. McEliece California Institute of Technology, pp. 44-55.

[15] D. Divsalar and F. Pollara, "Multiple Turbo Codes for Deep-Space Communications," The Telecommunications and Data Acquisition Progress Report 42-121, January-March 1995, Jet Propulsion

Laboratory, Pasadena, California, pp. 66-77, May 15, 1995.

[16] H. Xu, F. Takawira, "Performance bounds of Turbo Codes with Redundant Input," School of Electrical and Computer Engineering, Inha University, Korea, and Guilin Institute of Electronic Technology, P.R. China, School of Electrical and Electronic Engineering, University of Natal, South Africa.

[17] Carl Fredrik Leanderson, "Low-rate Turbo Codes," Department of Applied Electronics, Lund University, SE-221 00 LUND, Sweden, 1998.