

# Repeat-Puncture Superorthogonal Convolutional Turbo Codes in AWGN and Flat Rayleigh Fading Channels

Narushan Pillay, Hongjun Xu, *Member, IEEE*, Fambirai Takawira, *Member, IEEE*

**Abstract**—Repeat-Puncture Turbo Codes (RPTC), an extension of the conventional turbo coding algorithm, has shown a significant increase in performance at moderate to high signal-to-noise (SNR) ratios. Superorthogonal Convolutional Turbo Codes (SCTC) makes use of superorthogonal signals to improve the performance of conventional Turbo Codes (TC). By combining Superorthogonal Convolutional Turbo Codes and Repeat-Puncture Turbo Codes a coding scheme that exhibits a more superior performance results. In this paper, we study a new low-rate coding scheme Repeat-Puncture Superorthogonal Convolutional Turbo Codes (RPSCTC) that makes use of superorthogonal signaling, together with repetition and puncturing to improve the performance of Turbo Codes for reliable and effective communications. Simulation results in additive white Gaussian noise (AWGN) and flat Rayleigh fading (FRF) channels are presented together with analytical bounds of bit-error probabilities derived from transfer function bounding techniques.

**Index Terms**—orthogonal, repetition, puncturing, turbo codes.

## I. INTRODUCTION

Modern digital cellular systems utilize some form of channel coding to improve bit-error performance. Convolutional coding is the most widely used in these systems. However, operating at high signal-to-noise (SNR) ratios require larger antennas and bandwidths to be utilized resulting in a larger link budget. In the early ninety's Turbo Codes were introduced by Berrou *et.al.* in [1]. The performance was shown to be very close to the theoretical limit. Typically for large frame lengths e.g.  $N=16384$ ,  $E_b / N_o$  values of -0.15dB at a bit-error rate (BER) level of  $10^{-3}$  have been reported. TCs have three enhancements in the coding area. These include the use of recursive systematic codes, random interleaving and the separation of intrinsic and extrinsic information in the decoder for co-operation.

In [2], [5] Superorthogonal Convolutional Turbo Codes are introduced. This makes use of superorthogonal signals to create a low-rate code suitable for spread-spectrum applications. SCTCs exhibit an improved performance over the classical turbo coding algorithm. For small frame lengths e.g.  $N=200$ , and constraint length  $K=4$ ,  $E_b / N_o$  values of approximately 0.7dB at a BER level of

$10^{-3}$  have been reported. Repeat-Puncture Turbo Codes introduced in [7] has shown a significant increase in performance at moderate to high SNRs. In conjunction with superorthogonal convolutional signaling, simulations have shown a much improved performance.

In Section II the concept of Repeat-Puncture Superorthogonal Convolutional Turbo Codes together with their encoding and decoding structures will be introduced. Section III will present the max-log-MAP decoding algorithm used for the iterative decoding of TCs. Section IV will then present the performance evaluation for AWGN and Flat Rayleigh fading channels together with their analytical BER bounds. Finally Section V concludes the paper.

## II. REPEAT-PUNCTURE SUPERORTHOGONAL CONVOLUTIONAL TURBO CODES

### A. Encoder

The structure of the RPSCTC scheme is shown below in figure 1 with a more detailed diagram depicted in figure 2. The first parity sequence  $y_{1,0}, y_{1,1}, \dots, y_{1,2^{m-1}N}$  is produced from the first constituent code with an input data frame of length  $N$ . Since the parallel concatenated code (PCC) is made up of superorthogonal recursive convolutional code (SRCC) component codes the length of the first parity sequence is  $2^{m-1}N$  as seen in [5]. Prior to the second parity sequence  $y_{2,0}, y_{2,1}, \dots, y_{2,2^{m-1}LN}$  being produced, the input data frame is repeated  $L$  times and interleaved by an interleaver  $\pi$  of length  $LN$ . To control the code rate at the output the second parity sequence is then punctured and serially transmitted together with the first parity sequence to the receiver front-end. Each SRCC is shown in detail in figure 2. The Walsh-Hadamard generator uses Walsh functions obtained from Hadamard matrices to generate the orthogonal sequences. For memory,  $m=4$ , Hadamard matrices  $H_8$  and its complementary  $\overline{H_8}$  need to be considered. The Hadamard matrices for  $m=4$  is shown in figure 3.

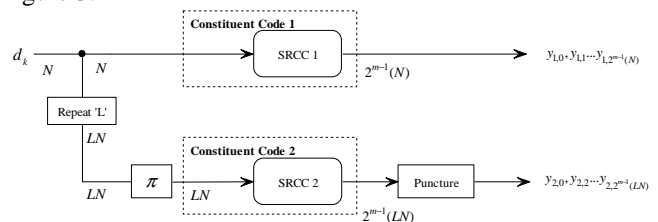


Fig.1. Structure of RPSCTC Encoder

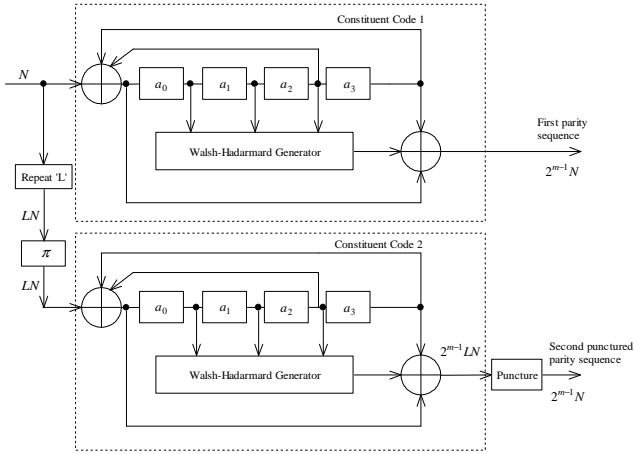


Fig.2. Detailed structure of the RPSCTC encoder

$$H_8 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \quad \overline{H}_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

Fig.3. Hadamard matrices for  $m=4$

### B. Decoder

The decoding for Repeat-Puncture Superorthogonal Convolutional Turbo Codes is an iterative process. Figure 4 shows the procedure of decoding for RPSCTC. The first corrupted parity sequence is sent to the first decoder. The decoder then produces log-likelihood ratios,  $L(\hat{d}_k)$  according to equation (1),

$$L(\hat{d}_k) = \log \left[ \sum_m \lambda_k^{1,m} \right] - \log \left[ \sum_m \lambda_k^{0,m} \right] \quad (1)$$

where,

$$\sum_m \lambda_k^{1,m} = \sum_m \alpha_k^m \delta_k^{1,m} \beta_{k+1}^{f(1,m)} \quad (2)$$

and

$$\sum_m \lambda_k^{0,m} = \sum_m \alpha_k^m \delta_k^{0,m} \beta_{k+1}^{f(0,m)}. \quad (3)$$

assuming zero *a-priori* information since, the probability of the first bit being either zero or one is equal. In equations (2) and (3)  $\alpha_k^m$  is the forward state metric,  $\beta_{k+1}^{f(i,m)}$  ( $i = 0,1$ ) is the reverse state metric and  $\delta_k^{i,m}$  ( $i = 0,1$ ) is the branch metric all computed over the trellis of depth  $k$ . Extrinsic information is then produced using equation (4),

$$L_e(\hat{d}_k) = L(\hat{d}_k) - L_c(x_k) - L_{apriori}(d_k) \quad (4)$$

where  $L_c(x_k)$  is the channel measurement and  $L_{apriori}(d_k)$  is achieved from the last iteration. The  $N$  extrinsic decisions are then sent to a repeater which yields  $LN$  extrinsic decisions which are then interleaved. The second component decoder uses the  $LN$  extrinsic information as *a-*

*priori* information gleaned from the decoding process. Again decoder 2 produces extrinsic information according to (4). These decisions are deinterleaved and averaged according to (5) to produce  $N$  decisions. The process of decoding is iterated several times after which the averaged decisions are compared to a zero threshold to yield the estimated input data sequence.

$$L_{e,i,average} = \frac{L_{e,2i} + L_{e,2i+1}}{2}, \quad i=0, 1 \dots N-1 \quad (5)$$

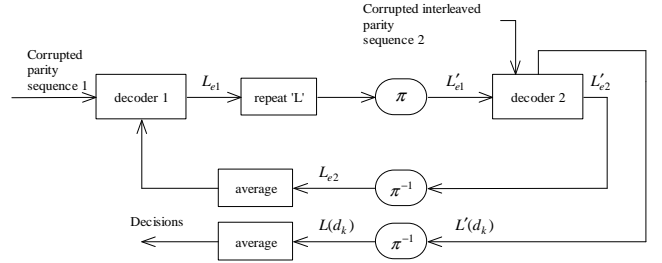


Fig.4. Structure of RPSCTC Decoder

### III. MAX-LOG MAP DECODING ALGORITHM

Although the maximum *a-posteriori* (MAP) algorithm yields the best overall performance in the decoding of turbo codes [5], [6] a hardware implementation of RPSCTC would lead to many problems due to the increase in complexity. When we use the maximization approximation instead, for the forward and backward state metrics, i.e.,  $\alpha_k$  and  $\beta_k$ , this leads to an approximation error in the computation of these variables. For high SNRs this error is comparable to noise, thus degrading performance, while for low SNRs it is much less than the noise power. The general objective of the max-log-MAP algorithm is to find the best path of the trellis for hard decisions, and for the decision on each input bit to evaluate the reliability of that decision. Thus by using the max-Log-MAP algorithm we are settling for a degraded performance but a significantly lower complexity for a hardware implementation.

### IV. PERFORMANCE ANALYSIS

#### A. Simulation Results in AWGN Channel

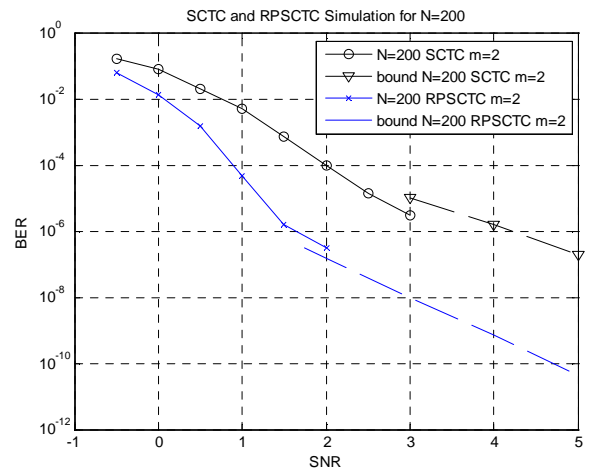


Fig.5. SCTC and RPSCTC Simulation in AWGN for  $m=2$

Repeat-Puncture Superorthogonal Convolutional Turbo Codes was simulated in the additive white Gaussian noise

channel. The results are shown in figure 5 and figure 6 which show the graph of  $E_b / N_o$  versus bit-error rate. The simulation and its bound for memories,  $m=2$  and  $m=4$ , are shown in figure 5 and figure 6 respectively.

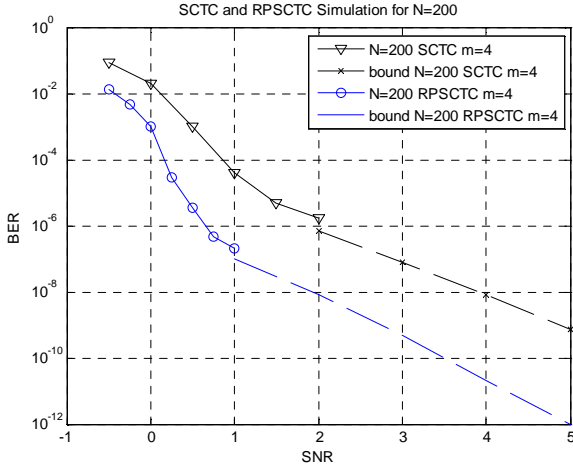


Fig.6. SCTC and RPSCTC Simulation in AWGN for  $m=4$

An input frame length of  $N=200$  was used and the stopping criterion was 50 frame errors. A uniform interleaver was chosen and a straightforward puncturing pattern was used, although other extravagant patterns could be investigated to yield a better performance. 18 iterations were chosen for the decoder, although this could be decreased at higher SNRs to shorten simulation time without a compromise on accuracy.

### B. Simulation Results in Flat Rayleigh Fading Channel

Rayleigh fading is a statistical model for the effect of a propagation environment, (or the heavy build up of urban environments), on a radio signal and is a reasonable model for tropospheric and ionospheric signal propagation.

For non-line-of-sight (NLOS) between the transmitter and receiver, the scheme was also simulated in a Rayleigh fading channel. The results are presented in figures 7 and 8. Again the measure of quality of service (QoS) chosen is the bit-error rate versus SNR plot. 150 frame errors were used as a stopping criterion for the simulations for both SCTC and RPSCTC.

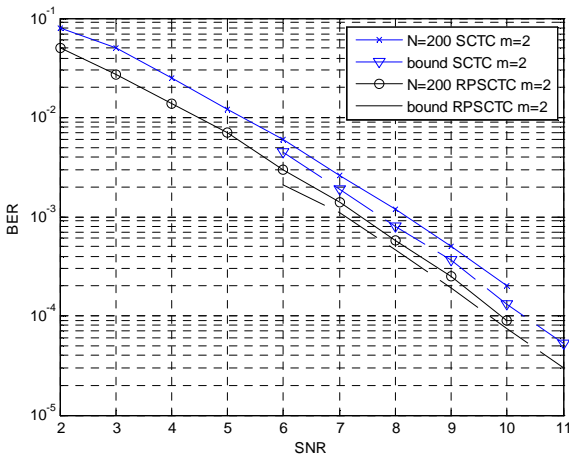


Fig.7. SCTC and RPSCTC Simulation in Rayleigh fading channel for  $m=2$

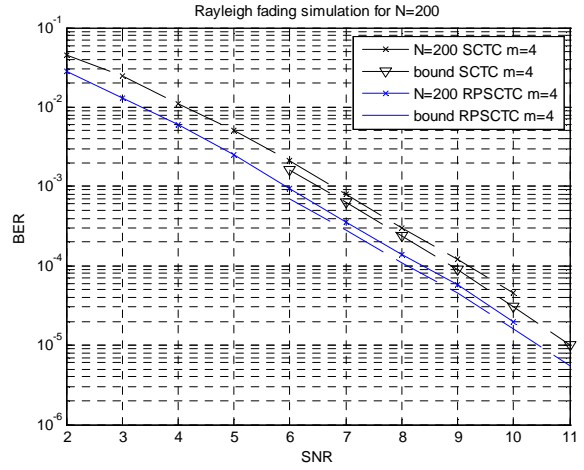


Fig.8. SCTC and RPSCTC Simulation in Rayleigh fading channel for  $m=4$

The velocity used for the moving receiver was  $v_c = 50\text{km/h}$ , and a  $f_c = 2\text{GHz}$  carrier frequency was used with Doppler frequency  $f_d$  given by (6). The speed of light  $c$  is  $3 \cdot 10^8$ .

$$f_d = \frac{v_c f_c}{c} \quad (6)$$

### C. Analytical BER Bounds

Transfer function bounding techniques studied in [8] was used to derive the union bound for Repeat-Puncture Superorthogonal Convolutional Turbo Codes. For an AWGN channel we start with the state transition matrix shown in equation (7)

$$A(L, I, D) = \begin{pmatrix} L^1 I^i D^d_{0,0} & \cdots & L^1 I^i D^d_{0,j} \\ \vdots & \ddots & \vdots \\ L^1 I^i D^d_{z,0} & \cdots & L^1 I^i D^d_{z,j} \end{pmatrix} \quad (7)$$

where, in a monomial  $L^l I^i D^d$ ,  $l$  is always equal to 1, and  $i$  and  $d$  are either zero or one, depending on the input and output weights respectively for the  $z^{\text{th}}$  to the  $j^{\text{th}}$  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 (8).

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

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

Then the transfer function for each constituent encoder can be expressed in the form of (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 (11) for constituent encoder 1

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

and (12) for constituent encoder 2.

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

The bound can be obtained from (13)

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

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

For application of the bound to Rayleigh fading channels, two-codeword probability is needed. For channels with side-information (SI) and channels with no-side-information (NSI) the two-codeword probability is given either by (15) or (16) respectively.

$$p_2^{SI}(d) \leq \frac{1}{2} \left( \frac{1}{1 + E_s / N_o} \right)^d \quad (15)$$

$$p_2^{NSI}(d) \leq \frac{1}{2} \left( e^{\beta/\alpha} \cdot \frac{\sqrt{1 + 2/\beta} - 1}{\sqrt{1 + 2/\beta} + 1} \right)^d \quad (16)$$

where  $\beta = \sqrt{\alpha^2 - 1}$  and  $\alpha = E_b R / N_o$

## V. CONCLUSION

In this paper we have presented the results for a new scheme, Repeat-Puncture Superorthogonal Convolutional Turbo Codes. The scheme was simulated in both the AWGN channel and a flat Rayleigh fading channel, and presented together with its union bound, derived with transfer function bounding techniques. RPSCTC offers a more superior performance than Superorthogonal Convolutional Turbo Codes in additive white Gaussian noise as well as NLOS channels.

## 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] 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.
- [3] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," IEEE Trans. Inform. Theory, Mar. 1974, vol. IT-20, pp. 284-287.
- [4] 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.
- [5] 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., February 1998, vol. 16, no.2, pp. 196-205.
- [6] B. Sklar, Digital Communications. Fundamentals and Applications. Beijing 2001
- [7] 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.
- [8] 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.
- [9] 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.
- [10] 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.
- [11] Eric K. Hall, Stephen G. Wilson, "Design and Performance of Turbo Codes on Rayleigh Fading Channels," Department of Electrical Engineering, University of Virginia.
- [12] Andrew J. Viterbi, "An Intuitive Justification and a Simplified Implementation of the MAP Decoder for Convolutional Codes," IEEE JSAC, Feb. 1998, vol. 16, pp. 255-264.