Punctured Turbo Code based ARQ schemes in Rayleigh Fading

M. Kousa, S. A. Al-Semari, A. B. Adinoyi and A. H. Mugaibel
Electrical Engineering Department
King Fahd University of Petroleum & Minerals
Dhahran 31261, Saudi Arabia.

Extended Abstract

Turbo codes based automatic repeat request (ARQ)can be a powerful technique to enhance the reliability of transmission and throughput efficiency of data networks giving the astonishing performance of turbo codes[1]. For time-varying channels of wireless system, the throughput can be maximized by adaptively matching the error correction capability of the code to the prevailing channel conditions. This forms the motivation for type-II ARQ schemes. The problem of constructing adaptive-rate error correction codes has been addressed in this work. The throughput efficiency of hybrid ARQ based on punctured turbo code is investigated. This work like others, is inspired by Hagenauer [2] proposal of rate-compatible punctured convolutional codes (RCPC)
The RCPT encoder encodes the information packet and stores all the coded bits in buffers for any possible retransmissions, see Figure 1. A set of puncturing rules is defined in the puncturing pattern to ensure the code adaptation. An example is shown in Table 1 for period of 4. Note that rate compatibility is required for this Hybrid ARQ schemes, therefore, the high rate codes must be embedded in the lower rate codes. At each step the transmitter needs only to transmit supplemental code symbols.

Considering puncturing patterns of period 4, one can produce the rates, $\{1, \frac{4}{5}, \frac{2}{3}, \frac{4}{7}, \frac{1}{2}, \frac{4}{9}, \frac{2}{5}, \frac{4}{11}, \frac{1}{3}\}$. For a less pathological channel the system may try to use all codes in this set. Hence, high throughput may be obtained. A pessimistic assumption of bad channels could consider an exponential jump in the code rate selection. The general steps involved in the RCPT-ARQ protocol are given as follows.

1) Encode the k^{th} packet with the rate 1/3 turbo code and buffer all the coded bits. Choose a set of puncturing patterns

2) First transmit the bits specified in the first pattern of the set (usually the information bits only).

3) The receiver assembles the received bits, inserting erasures for the bits that have not been transmitted according to the puncturing, and decoding is attempted. We have assumed that all error patterns can be

detected. If the frame received by the sink is error free, an ACK is sent to the transmitter and the protocol is reset and returned to step 1. Otherwise, a NACK is transmitted back.

4) Upon reception of NACK the transmitter sends additional bits according to the next pattern of the set. Go back to step 3.

The first m transmissions will all be distinct. They exhaust all the code rates in the set, reaching the full turbo code of rate 1/3. We say this scheme has a degree of freedom of adaptation of m. If an error-free packet cannot be passed yet, different design options can be considered. The simplest is to accept a given frame error rate and pass the frame as is. Another option is to hold the decoder in full turbo but the transmitter is reset to step 2. In this case code combining is done.

The bound on throughput, ρ, has been derived. Throughput is the ratio of the length of the information frame to the average total number of bits, N_a, required to be transmitted before the information frame is passed to the sink. For the type-II HARQ system, N_a, has been bounded for a three stages of adaptation, the FER at stages 1, 2 and 3 are F_1,F_2, F_3 respectively. For puncturing pattern having a period, $p = 2$ (can be extended to $p > 2$), the N_a is obtained,

$$\begin{aligned} N_a &= 1(1-F_1) + 2F_1(1-F_2) + 3F_1F_2(1-F_3) + 4F_1F_2F_3(1-F_3) + 5F_1F_2F_3^2(1-F_3) + \cdots \\ &= 1(1-F_1) + 2F_1(1-F_2) + F_1F_2(1-F_3)\sum_{n=0}^{\infty}(n+3)F_3^n, \end{aligned} \quad (1)$$

After some manipulations of (1), we arrive at,

$$N_a = 1 + F_1 + F_1F_2/(1-F_3). \quad (2)$$

Finally the throughput efficiency, ρ, can be found,

$$\rho = 1/N_a, \quad (3)$$

For real-time applications, a certain maximum number of requests (n transmissions) is allowed, thus limiting the delay to an acceptable level. The N_a is modified and the sys has to cope with residual frame error.

We have proposed a way of evaluating the bound on FER for punctured turbo code base on specific puncturing patterns used to obtain a RCPT-ARQ scheme. Hence the bound on throughput can be evaluated. Some simulation

results are presented for turbo-hybrid ARQ scheme in Figure 2. It shows the throughput of patterns of periods 2, 4 and 8 yielding the best performance. We observed the improvement in the throughput by using a larger period. However, the improvement is of diminishing nature. The improvement is attributed to the freedom offered by larger period in distributing the puncturing bits.

In conclusion, puncturing has been employed to generate sets of rate-compatible punctured turbo codes (RCPT) that are suitable for type-II ARQ schemes. The scheme performance was measured in terms of its throughput. On the overall, we can summarize our findings in the following points:

- The patterns of puncturing have a significant effect on the overall throughput. When a code set is designed along the guidelines of good puncturing this work presents, a substantial boost in throughput in obtained. Our simulation results show that a throughput difference of 15 - 20 % between the best pattern set and the worst pattern set is possible.
- When the period of puncturing is prolonged, even higher throughput values are possible provided that the puncturing patterns are properly selected.
- Increasing the degree of freedom of adaptation can improve the system throughput provided that the puncturing period is long enough to allow the design of good puncturing pattern sets. An improvement in throughput of up to 10 % was obtained for the higher periods 4 and 8 when m is increased from 3 to 5.

We believe that the approach and findings of this work are enlightening for the design of puncturing turbo codes for ARQ schemes.

References

[1] Claude Berrou and Alain Glavieux "Near Optimum Error Correcting Coding And Decoding: Turbo-Codes". *IEEE Trans on Information Theory*, VOL-44, pp.1261-1271, October, 1996.

[2] J. Hagenauer, "Rate compatible punctured convolutional codes (RCPC-codes) and their application" , *IEEE Transactions on Communications*, Apr. 1988, vol. 36, pp. 389-400.

Table 1: Rate set puncturing pattern for period 4

Name	Pattern
R_{41}	$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$
R_{42}	$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$
R_{43}	$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$

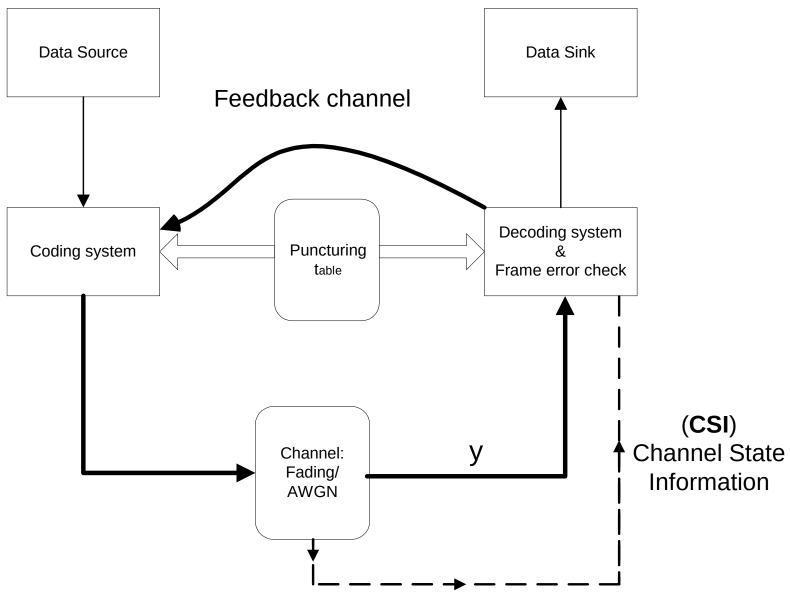

Figure 1: Block diagram of a turbo-based FEC/ARQ system over fading channel.

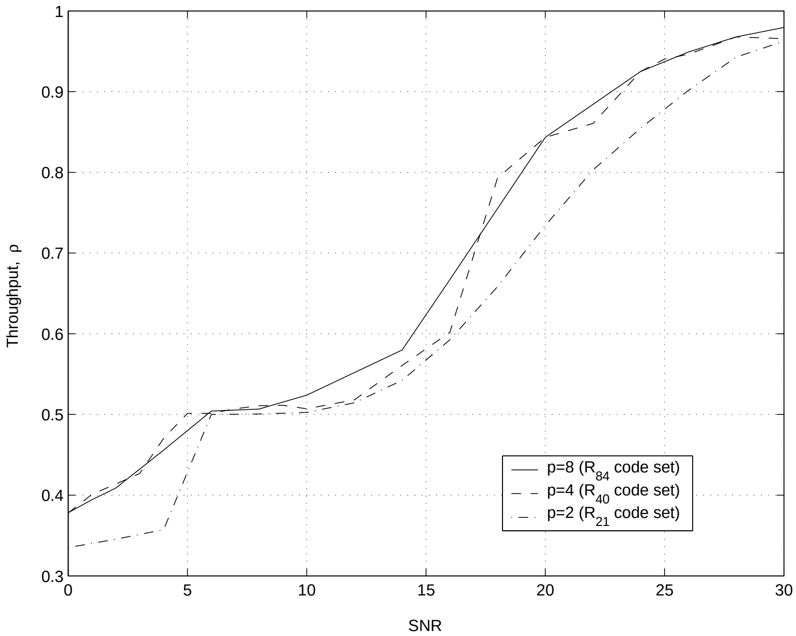

Figure 2: Throughput comparison of different periods with $m = 5$.