Interleavers for Unpunctured Symmetric Turbo Codes

Johann A. Briffa
Dept. Comms. & Computer Eng.,
University of Malta
Msida MSD 06, Malta
E-mail: jabbrif@eng.um.edu.mt

Victor Buttigieg
Dept. Comms. & Computer Eng.,
University of Malta
Msida MSD 06, Malta
E-mail: vjbutt@eng.um.edu.mt

Abstract — The Turbo code interleaver design problem is considered for relatively large block sizes, where the effect of trellis termination is less marked. An optimised interleaver design technique based on simulated annealing is proposed — performance is significantly better than the Berrou-Glavieux interleaver without an increase in delay.

I. INTRODUCTION

The classical use of interleavers is to randomise the location of errors, enabling the use of random-error-correcting codes on channels with burst error patterns. Turbo coding also introduces a further dimension to interleaver requirements, due to the effects of the iterative algorithm. Most optimised interleaver design techniques in the literature are based on the JPL’s $S$-random interleaver algorithm [1]. While $S$-random interleavers perform well, the technique was not intended as a basis for advanced interleaver design. Its main shortcomings are that it is not guaranteed to produce the required interleaver and that it only aims at achieving a spread $S$.

II. OPTIMISED INTERLEAVER DESIGN

Simulated annealing [2] can be used to design optimised interleavers by defining an energy function based on a predefined set of requirements. We use a random interleaver as an initial state, and define the perturbation function as a swap of two random interleaver entries, ensuring that the interleaver is always valid. The energy function used is:

$$E = \sum_{i,j} \frac{5\nu}{\sqrt{(i-j)^2 + (\lambda(i)-\lambda(j))^2}}$$  \hspace{1cm} (1)

where $i,j \in [0,\tau-1]$, $\tau$ is the block size, $\nu$ is the encoder memory, and $\lambda()$ is the interleaving function. This energy function attempts to ‘push’ bit-pairs away from the origin in the Input-Output Distance Spectrum (IODS) [1], increasing the spread of the interleaver. In contrast with the JPL technique it does not guarantee a particular spread; however, it pushes points away from the origin even beyond the spread boundary.

III. RESULTS

We restrict ourselves to unpunctured rate $\frac{1}{2}$ symmetric Turbo codes with $\nu = 2$ and generator $[1, 5/7]$. In order to avoid the effects of trellis termination, we also choose $\tau = 1024$. As a reference for performance, we implement a uniform interleaver by using a different random interleaver for each block simulated [3]. We compare our interleaver design with this uniform interleaver, a rectangular interleaver, the design used by Berrou and Glavieux [4], and an $S$-random interleaver in Fig. 1. Our design achieves a BER of $10^{-5}$ at $E_b/N_0 = 1.35$ dB.

$^1$A two-dimensional histogram, with the axes being the distance between bit-pairs at the input and output of the interleaver [3].

$^2$Polynomials are denoted as $g_o$ or $g_o/g_r$, where $g_o$ is the feedback polynomial, in octal.

![Fig. 1: Turbo code BER simulation (10 iterations)](image)

IV. CONCLUSIONS

Our new interleaver design performs at least as well as the $S$-random interleaver. However, using our technique it is easier to include design restrictions, for example to make the interleaver correctly-terminating or odd-even. Also, more sophisticated energy functions matched to the component codes may be considered, particularly for use with punctured codes.

Utilising some performance enhancement techniques, the complexity of the energy function grows only as $O(\tau)$, making it suitable for use with large block sizes.

REFERENCES


