

TITLE:

A Proposal for Turbo Code Interleaving

SOURCE:

Motorola

ABSTRACT:

A generic method for generating pseudo-random interleaver addresses is presented for the turbo code interleaver. The characteristics of this turbo interleaver are: simple implementation, performance comparable to the best known interleaver, no reliance on look-up tables, small and deterministic latency in generating the pseudo-random addresses, and easy extension to a wide range of information rates. The performance of this interleaver is evaluated through simulation and compared with that of other proposals.

1.0 Introduction

In this contribution, a turbo interleaver method is proposed having the following desirable characteristics:

1. Simple implementation with no reliance on large look-up tables
2. Good performance
3. No consecutive punctures
4. Flexibility in switching between arbitrary information bit rates

2.0 The Proposed Turbo Code Interleaver - Mathematical Model

For a given information block length N , the turbo interleaver first needs to find the minimum m and n such that $N \leq 2^n \times (2^m + 1)$. For simplicity, n can also be a predetermined fixed value. In the proposed interleaver $n=5$ for all interleaver sizes.

Assuming $N = 2^n \times (2^m + 1)$, since 2^n and 2^m+1 are *relatively prime*, there exists $N!$ *one-to-one* mappings between

$$\{x(i): x(i) = i, i = 0, 1, \dots, N - 1\} \quad (1)$$

and

$$\{y(p, q) = p \times (2^m + 1) + q: p = 0, 1, \dots, 2^n - 1; q = 0, 1, \dots, 2^m\} \quad (2)$$

We choose the following mapping

$$x(i) \Leftrightarrow y(i) = p(i) \times (2^m + 1) + q(i) = 2^m p(i) + p(i) + q(i) \quad (3)$$

where

$$\begin{aligned} p(i) &= br_n(i) \\ q(i) &= pn_m(i) \end{aligned} \quad (4)$$

for $i = 0, 1, \dots, N - 1$, where pn_m is the m -bit PN generator with two additional states, 0 and 2^m , added prior to the initial state. We call this an *augmented PN generator* with period of 2^m+1 and br_n is the n -bit bit reverse sequence generator with period of 2^n , *i.e.*,

$$br_n(i) = \sum_{j=0}^{n-1} b_{n-j-1} 2^j \quad (5)$$

where b_j is defined as

$$\sum_{j=0}^{n-1} b_j 2^j = \text{mod}_{2^n}(i) \quad (6)$$

For $N < 2^n \times (2^m + 1)$, puncturing is needed.

The proposed interleaver diagram is shown in Figure 1. The maximum length sequence generator polynomials were chosen to minimize the occurrence of input error events of weight two and separation less than 28 being mapped to events of weight two and separations of 7 and 14. That is

$$D^{k_1}(1 + D^j) \rightarrow D^{k_2}(1 + D^{7l}) \quad (7)$$

where $7 \leq j \leq 28$, $1 \leq l \leq 2$, and k_1 and k_2 are integers. Because there can be no consecutive punctures, the occurrence of low output weight error events is minimized for arbitrary data rates, i.e., the interleaver is *puncture friendly*.

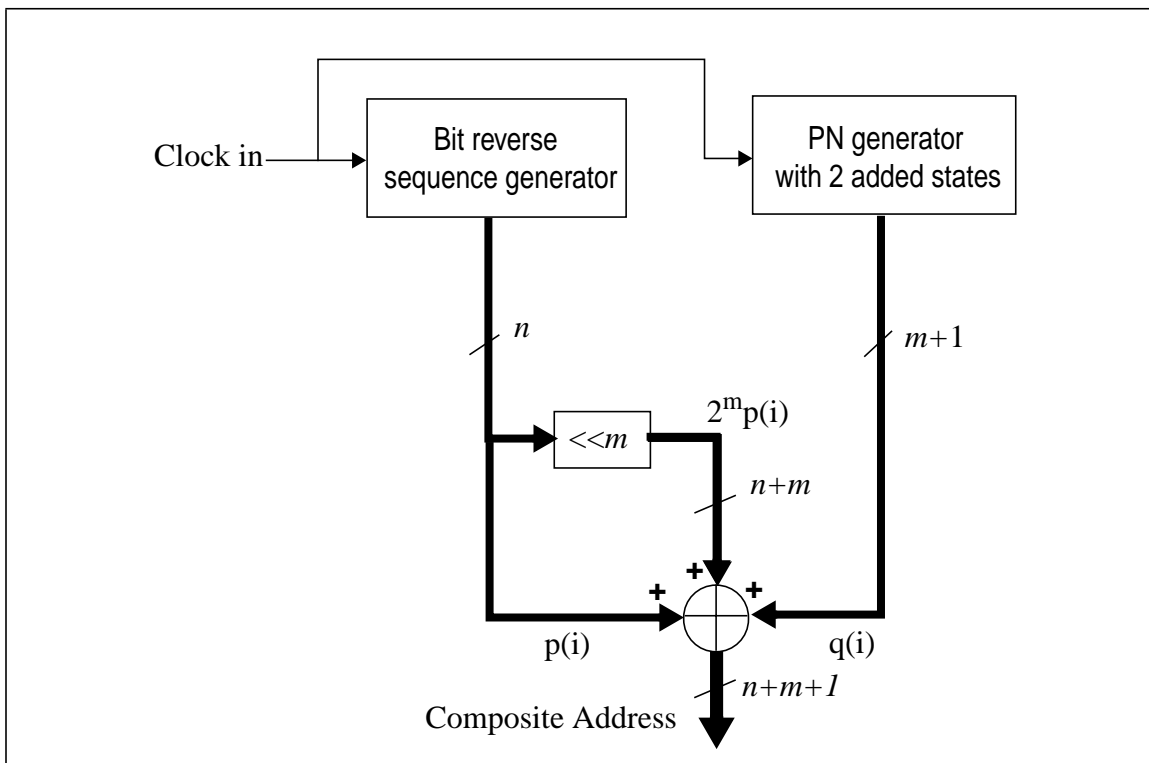


Figure 1: Block diagram of Motorola's pn-generator interleaver.

3.0 Description

The interleaver may also be described by a $2^n \times 2^m + 1$ rectangular array. The array is filled row by row. The rows are then permuted according to the bit reversal rule. The array is then read out by taking elements from each row in succession, clocking the augmented PN generator once each read, until the 2^n th row is reached at which time the reading continues at the first row. The order in which elements of a row are read is determined by the current state of the augmented PN generator. Mathematically this may be described as follows: Let $a(i)$ be the sequence of information bits to be interleaved. The array, $r(k, l)$, is filled according to

$$r(k, l) = a(2^n k + l). \quad (8)$$

The interleaved sequence $b(i)$ is then

$$b(i) = r[br_n(i), pn_m(i)] \quad (9)$$

where again $br_n(i)$ is the bit reversed version of $mod_{2^n}(i)$ and $pn_m(i)$ is the state of the augmented PN generator after i clocks.

The following summarizes the operation of the proposed interleaver:

1. Find the minimum m for the required block length and the corresponding predetermined generator polynomial.
2. Initialize the augmented PN generator to 1 and the bit reverse sequence generator to 0;
3. The n bit output from the bit reverse sequence generator is left shifted toward the MSB by m bits and added to the sum of the bit reverse and the augmented PN generator outputs. If the composite address is not a legal address, discard it.
4. Clock both generators once.
5. Loop back to step 3 until all information bits are interleaved.

A table of n and m values is shown below for block sizes 320, 640, and 5120.

Block size	n	m
320	5	4
640	5	5
5120	5	8

Table 1: Values for n and m

4.0 Performance

Figures 2 through 7 show the UMTS phase II¹ performance of the Motorola interleaver for block lengths of 320, 640, and 5120. All data points were generated with the Motorola phase II turbo code simulator. Results for the GF, MIL, and CDI interleavers are virtually identical to previous results presented in [4].

5.0 References

- [1] Interleaver Design for Turbo Codes Based on the Performance of Iterative Decoding, Johan Hokfelt, Ove Edfors, and Torleiv Manseng, Lund University, Department of Applied Electronics. August 12, 1998.
- [2] A General Turbo Interleaver Design Technique with Near Optimal Performance, Hughes Network Systems, Tdoc SMG2 UMTS-L1 337/98.
- [3] Algebraic Interleavers for Turbo Codes, CANON CRF, Tdoc SMG2 UMTS-L1 571/98.
- [4] Preliminary Simulation Results for Phase II Simulations on Turbo Codes, Hughes Network Systems, Tdoc SMG2 UMTS-L1 599/98.
- [5] Further Simulation Results on Turbo Codes, Hughes Network Systems, Tdoc SMG2 UMTS-L1 701/98.

1. The current simulation results use the initial UMTS phase II channel simulator. Results are currently being compiled with the improved channel estimator #2 [5].

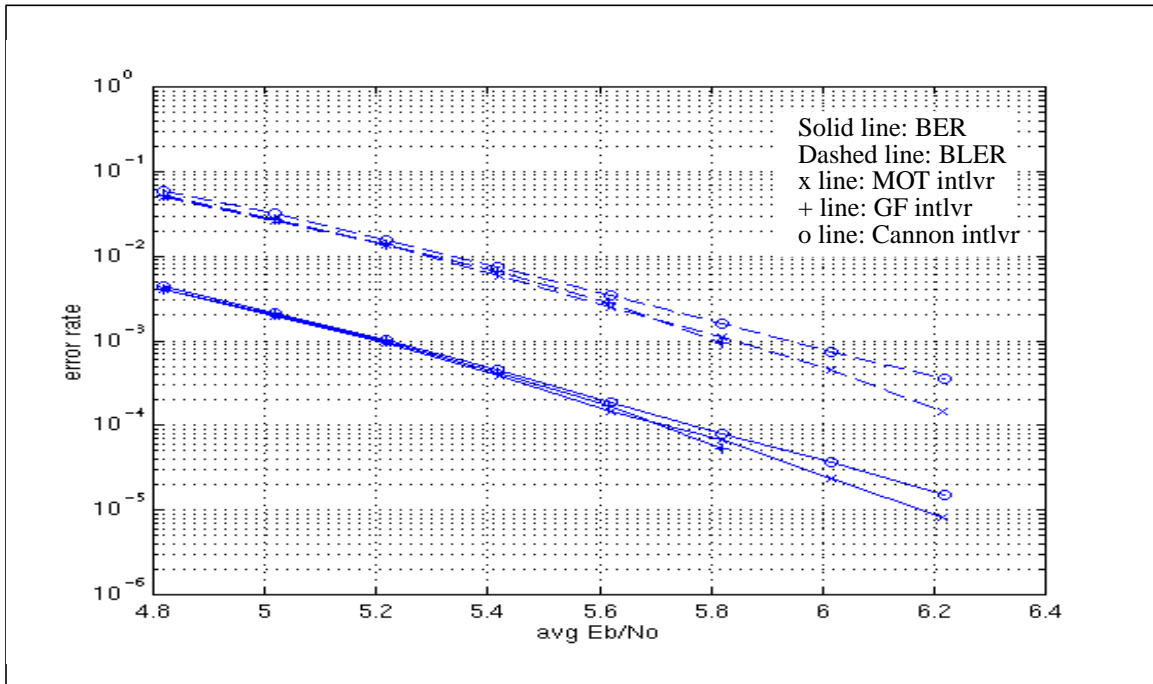


Figure 2: Turbo interleaver comparison using 8 states 4 iterations turbo code, data rate 32 kbps, 10 ms interleaving interval, mobile speed 30 kmph

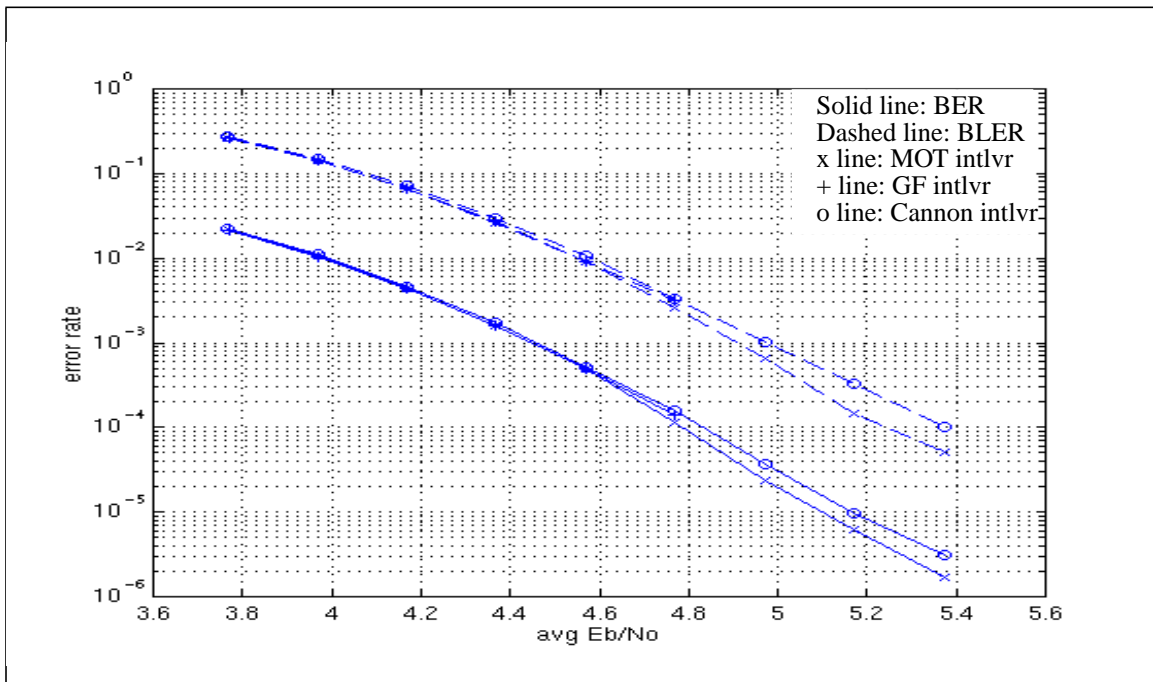


Figure 3: Turbo interleaver comparison using 8 states 4 iterations turbo code, data rate 32 kbps, 10 ms interleaving interval, mobile speed 3 kmph

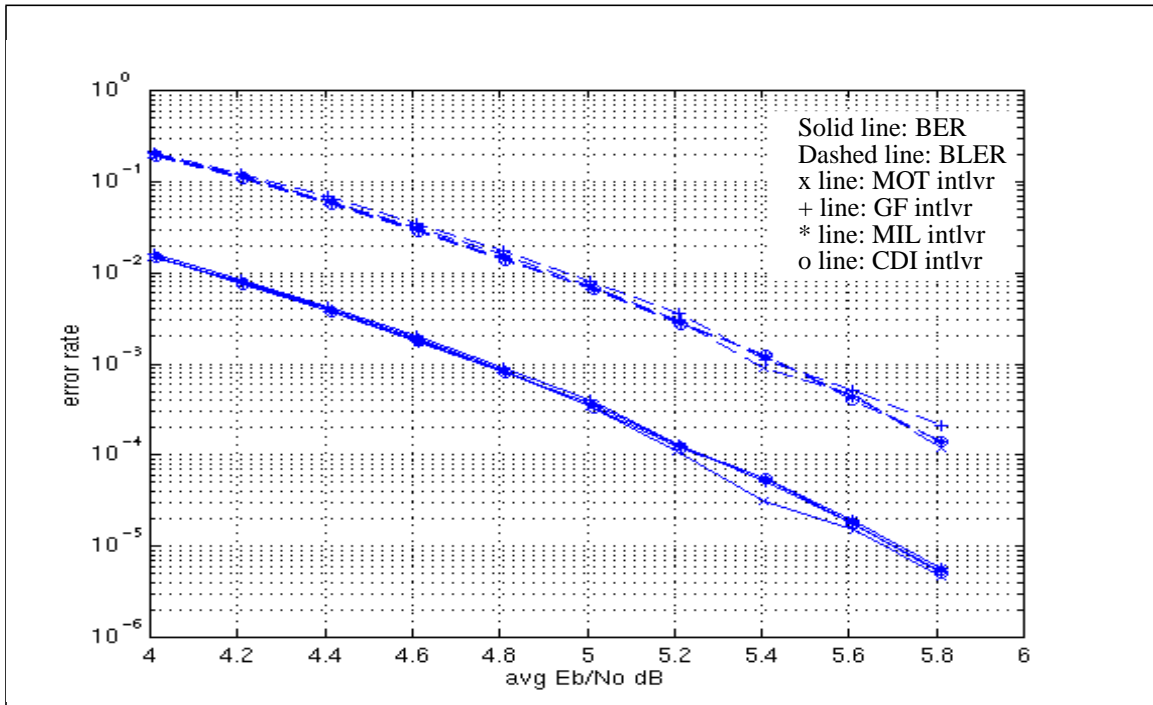


Figure 4: Turbo interleaver comparison using 8 states 4 iterations turbo code, data rate 64 kbps, 10 ms interleaving interval, mobile speed 30 kmph

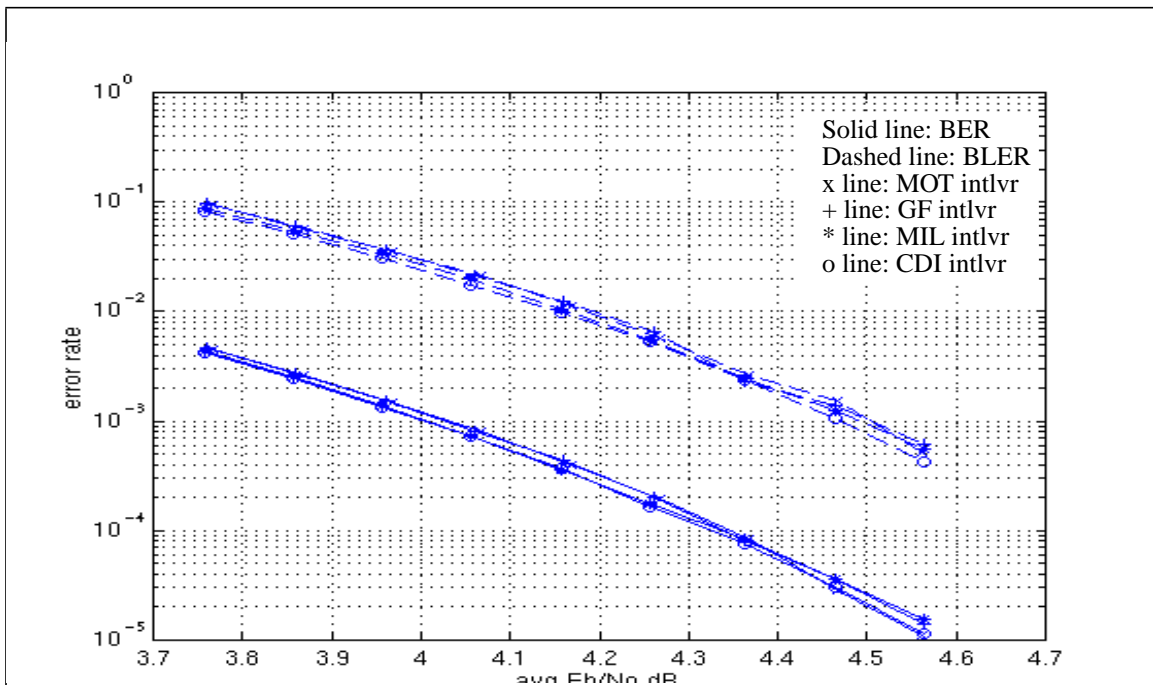


Figure 5: Turbo interleaver comparison using 8 states 4 iterations turbo code, data rate 64 kbps, 10 ms interleaving interval, mobile speed 3 kmph

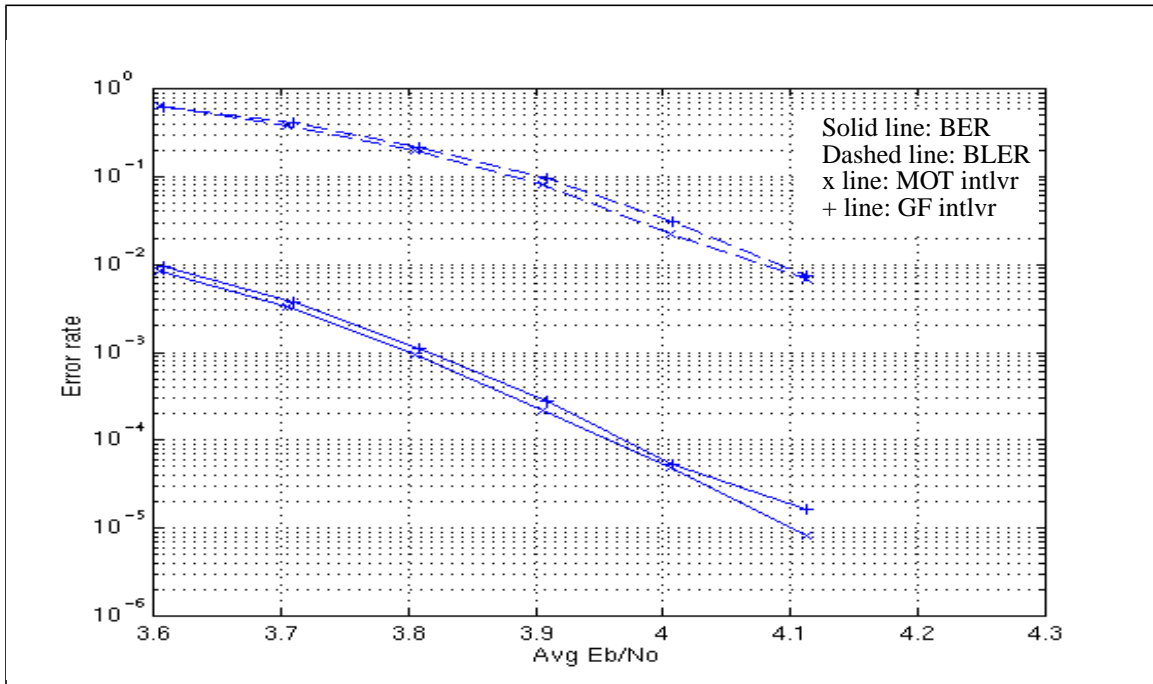


Figure 6: Turbo interleaver comparison using 8 states 4 iterations turbo code, data rate 64 kbps, 80 ms interleaving interval, mobile speed 30 kmph

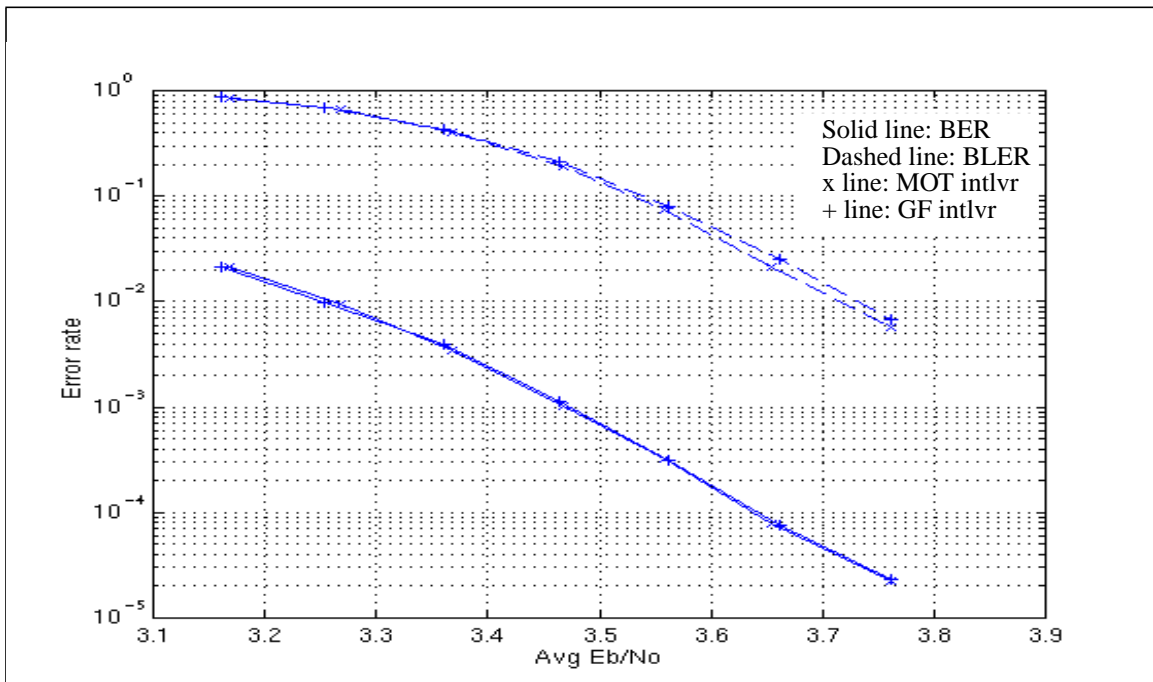


Figure 7: Turbo interleaver comparison using 8 states 4 iterations turbo code, data rate 64 kbps, 80 ms interleaving interval, mobile speed 3 kmph