TSG-RAN Working Group 1(Radio) meeting #6 **TSGR1#6(99) 907**
Espoo, Finland 13-16, July 1999

Agenda Item:

Source: **LGIC**

Title: **Simulation Results of Downlink Puncturing Algorithms**

Document for: **Discussion**

---

1. Introduction

In this document, simulation results of downlink puncturing algorithms are presented. An alternative puncturing algorithm(LGIC scheme) is compared with that of Fujitsu and also with the conventional scheme when it is applied to convolutional code or turbo code in an AWGN channel.

2. LGIC downlink puncturing algorithm

The previous LGIC's downlink puncturing algorithm used in the simulation is described as below.

Let's denote:

- N_C = number of bits per matching block
- N_i = number of bits per matched block
- $N = N_C/3$
- $S_0 = \left\{d_1, d_2, \ldots, d_{N_C}\right\}$ = set of N_C data bits

```
$y = N_C - N_i$
$e = 2 * N$
$m = 1$
if($N \% 2 == 1$)
        $flag = 0$
else
        $flag = 1$
endif
if(convolution code)
        $flag = 0$
endif
do while $m <= N$
        $e = e - 2 * y$
        if $e <= -2*N$ then
            puncture bit $3*m, 3*m-1$ from set $S_0$
            $e = e + 4*N$
        end if
        if $e <= 0$ then
            if $flag$ ==0
                puncture bit $3*m$ from set $S_0$
                $flag = 1$
            endif
            if $flag$ ==1
                puncture bit $3*m-1$ from set $S_0$
                $flag = 0$
            endif
            $e = e + 2*N$
        end if
        $m = m + 1$
    end do
```

3. Simulation results

The simulation conditions are summarized as below.

For the convolutional code simulation,

- Information block size is 160
- The code rates of the convolutional code are 1/3, 1/2.
- Constraint length of the constituent code is 9.
- Eight zero bits are used for terminating.
- Floating point Viterbi decoding is used.
- In every simulation point, at least 100 frame errors are counted
- Simulations are carried out in an AWGN channel
- The puncturing number of bits are 25(5%), 50(10%), 76(15%), 101(20%) for 1/3 code rate and 16(5%), 33(10%), 50(15%), 67(20%) for 1/2 code rate.
- Viterbi decoding process using the information of trellis termination is used

Unfortunately, the results for the turbo code when interleaver size in 5119, 5120 can not be provided because of the lack of simulation time.

For the turbo code simulation,

- Interleaver sizes are 320,321
- Internal interleaver is a prime interleaver
- Constraint length of the constituent code is 4.
- Conventional termination method is applied.

- A full MAP with floating point implementation is used for the decoding of the constituent encoders
- Iteration number is 12.
- In every simulation point, at least 100 frame errors in every point are counted
- Simulations are carried out in an AWGN channel
- The puncturing numbers of bits are 97(10%), 194(20%), 324(33.3%), 356(half rate 5%) for 320 interleaver size and 97(10%), 195(20%), 325(33.3%), 357(half rate 5%) for 321.

Figure 1, Figure 2, Figure 3 and Figure 4 show the simulation results in terms of BER and FER for the downlink puncturing algorithms when they are applied to 1/3 convolutional code. LGIC scheme has an almost equivalent performance to the conventional scheme when the values of P are 0.05, 0.1, 0.15, 0.2. It is noticeable that LGIC scheme shows a superiority in performance at BER of 10^{-5} when P is equal or smaller than 0.15.

The simulation results prove that the uniformity in the code symbol space provides an equivalent optimal rule to the conventional scheme[1].

It is noticed that the conventional puncturing algorithm has some degradation in performance when P is 1/6(=0.1667) as shown in Figure 5. In this case, LGIC scheme has a gain of aboult 0.1 dB at BER of 10^{-5} . The performance degradation of the conventional scheme is always observed when P is 1/(multiple of 6) and the conventional algorithm punctures the third bit in the code symbol at this values of P. This means that puncturing every third bit causes performance degradation and the performance of puncturing algorithm is dependent on bit pattern in one code symbol.

Figure 6 ~ 9 show the simulation results of BER and FER performance of downlink puncturing algorithm when code rate is 1/2. LGIC scheme also shows an equivalent performance to the conventional scheme when the values of P are 0.05, 0.1, 0.15, 0.2. The simulation results also prove that when code rate is 1/2, although LGIC scheme does not satisfy perfect unformity over all code symbols, it provides an approximate uniformity of code symbol space.

Fujitsu scheme always shows degradation of performance when it is applied to the convolutional code in comparison to other schemes. This is because Fujitsu scheme makes a consecutive puncturing as described in [9]

Figure 10 ~ 17 show the simulation results of BER and FER performance of downlink puncturing algorithms when they are applied to turbo code.

LGIC scheme shows a slight superiority to Fujitsu scheme in Figure 10 ~ 13 when puncturing is made for 1/3 turbo code. In Figure 13, a noticeable difference occurs and LGIC scheme shows a gain of about 0.1dB at a BER of 10^{-6} . In Figure 15, LGIC scheme shows a very slight superiority to Fujitsu scheme.

Fujitsu scheme only shows a slight superiority in Figure 14 when P=33.3%(half rate) and interleaver size is 320(even number) as shown in [5][6].

In case of turbo puncturing for 1/2 turbo code(P=0.05 for half rate), both turbo puncturing algorithms show almost same performance.

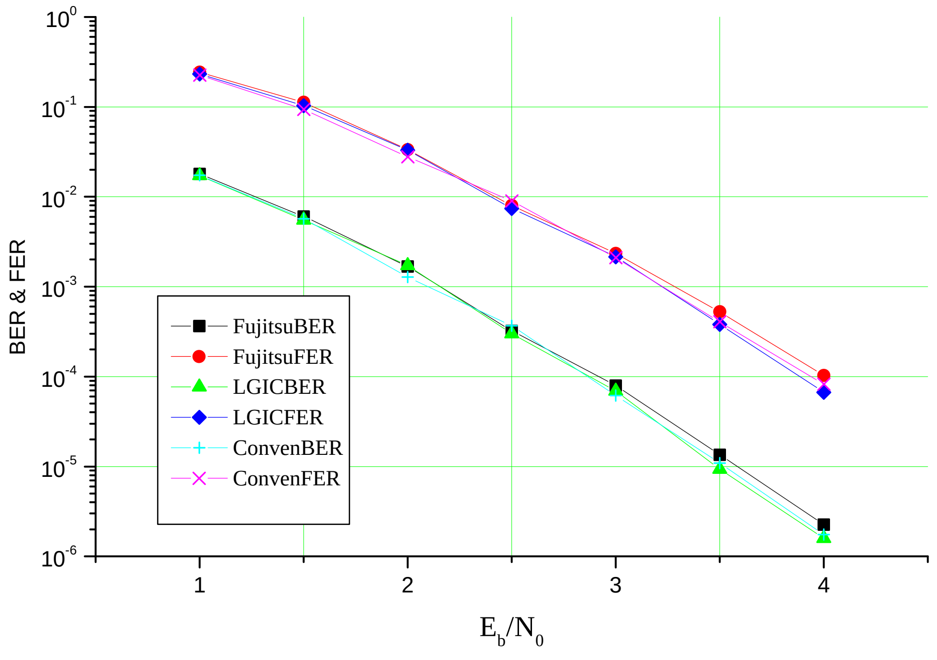

Figure 1. BER and FER of the puncturing schemes for1/3 convolutional code

(P = about 0.05 (25 bits punctured))

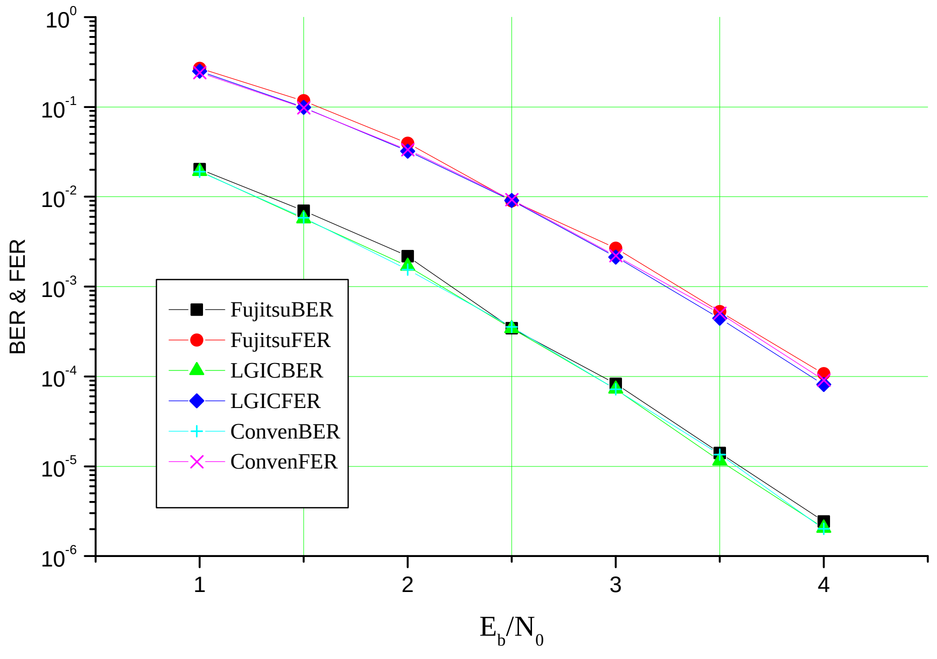

Figure 2. BER and FER of the puncturing schemes for1/3 convolutional code

(P = about 0.1 (50 bits punctured))

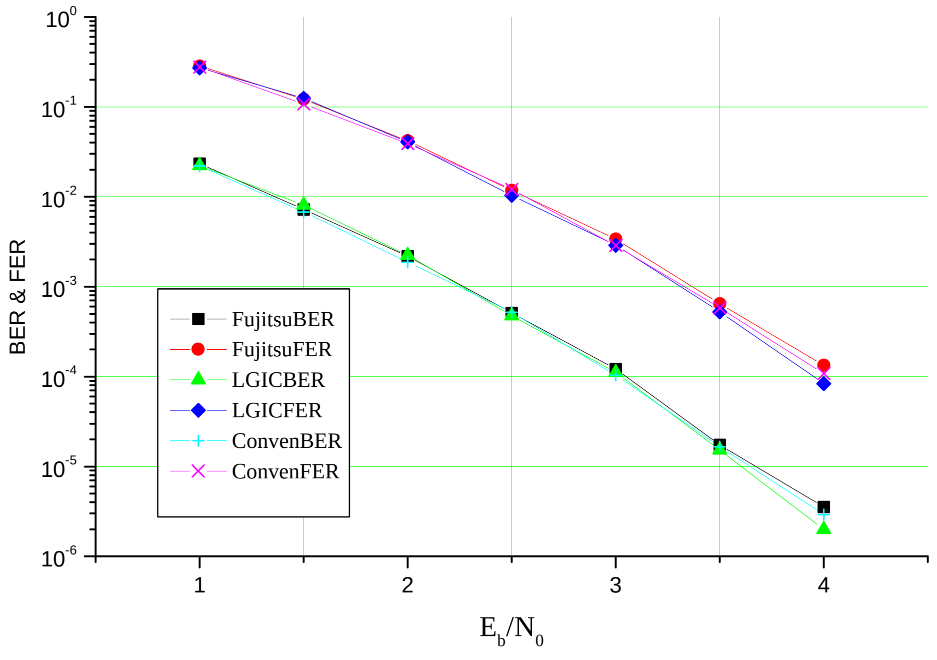

Figure 3. BER and FER of the puncturing schemes for1/3 convolutional code (P = about 0.15 (76 bits punctured))

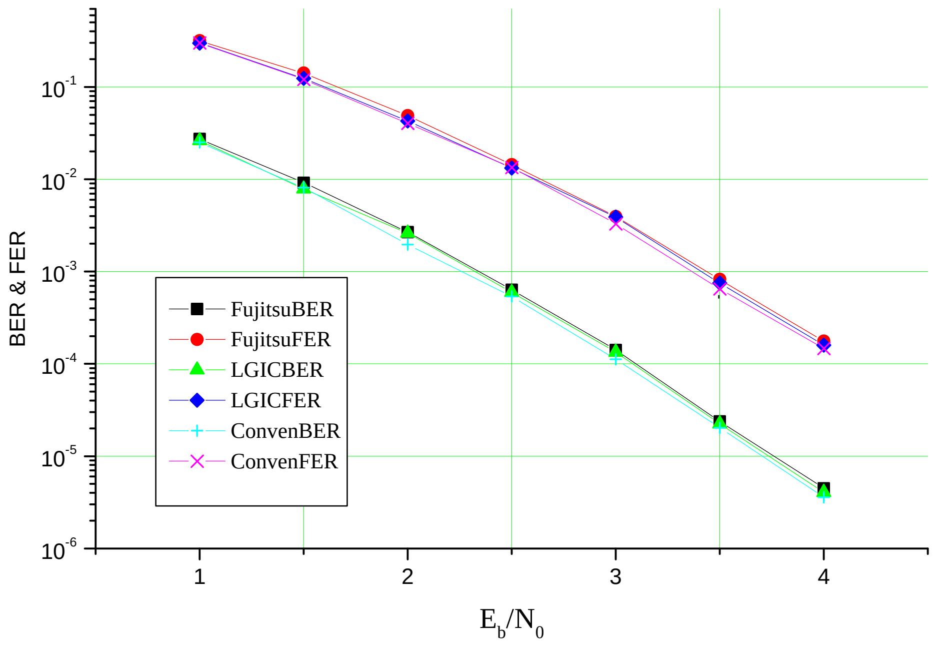

Figure 4. BER and FER of the puncturing schemes for1/3 convolutional code (P = about 0.2 (101 bits punctured))

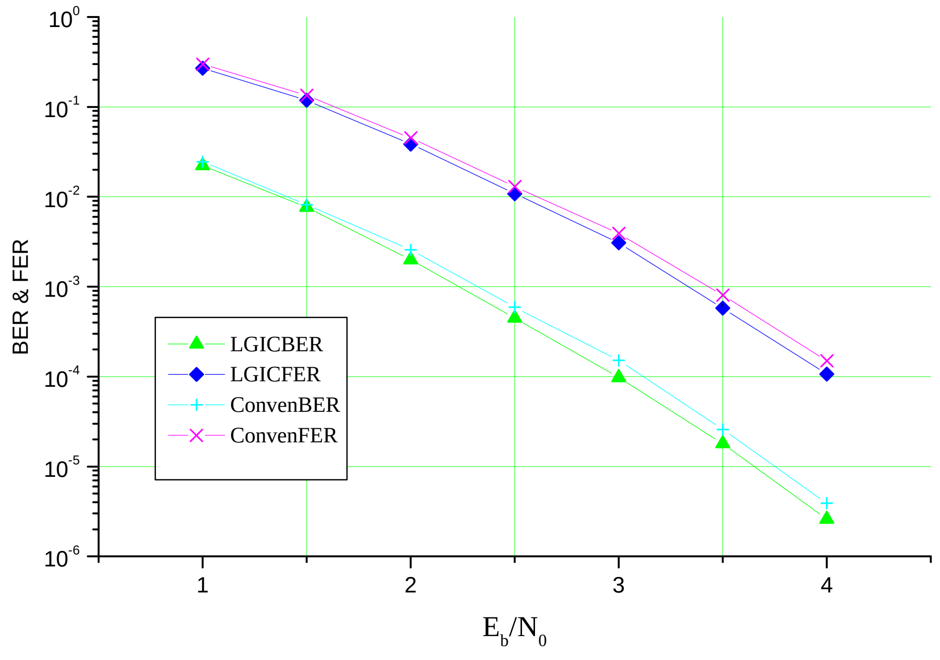

Figure 5. BER and FER of the puncturing schemes for1/3 convolutional code

(P = about 0.1667 (84 bits punctured))

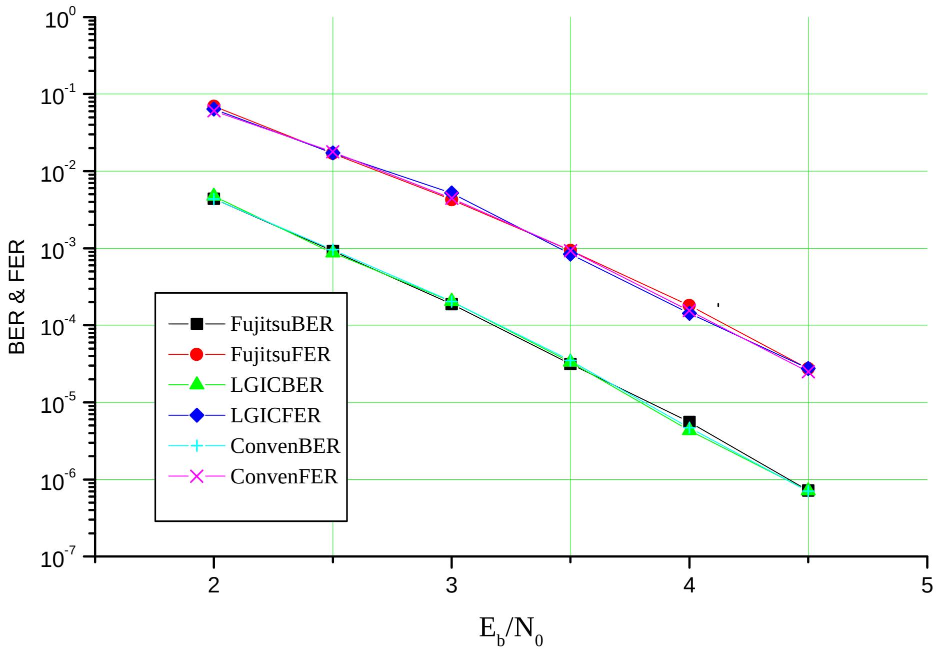

Figure 6. BER and FER of the puncturing schemes for1/2 convolutional code

(P = about 0.05 (16 bits punctured))

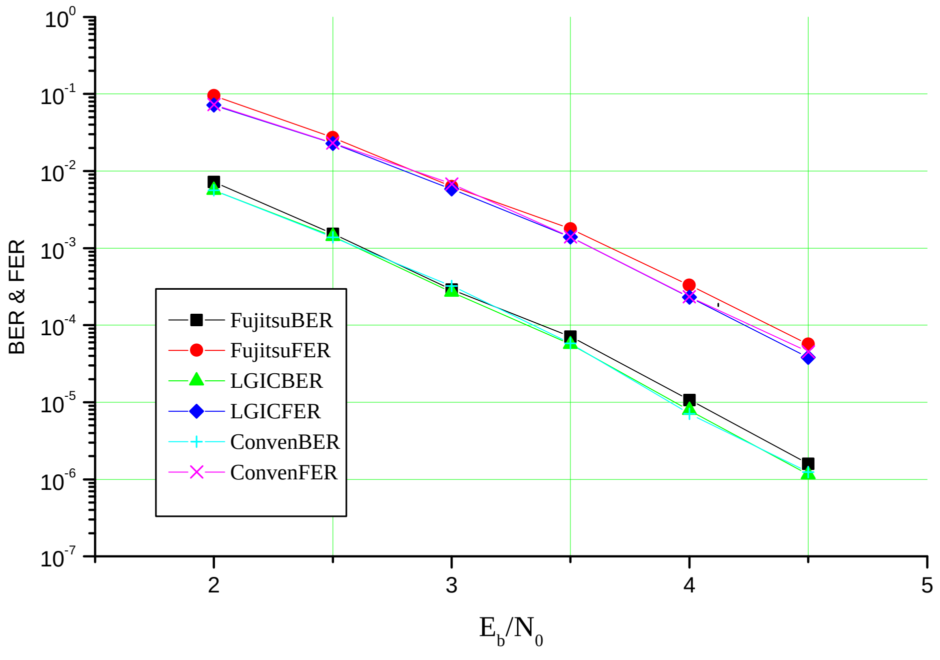

Figure 7. BER and FER of the puncturing schemes for1/2 convolutional code (P = about 0.1 (33 bits punctured))

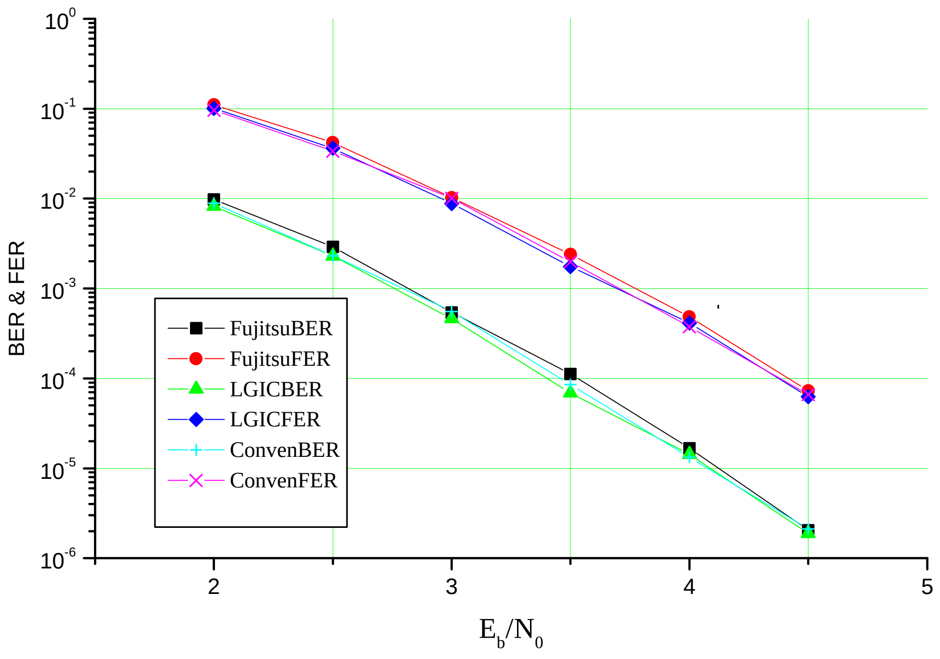

Figure 8. BER and FER of the puncturing schemes for1/2 convolutional code (P = about 0.15 (50 bits punctured))

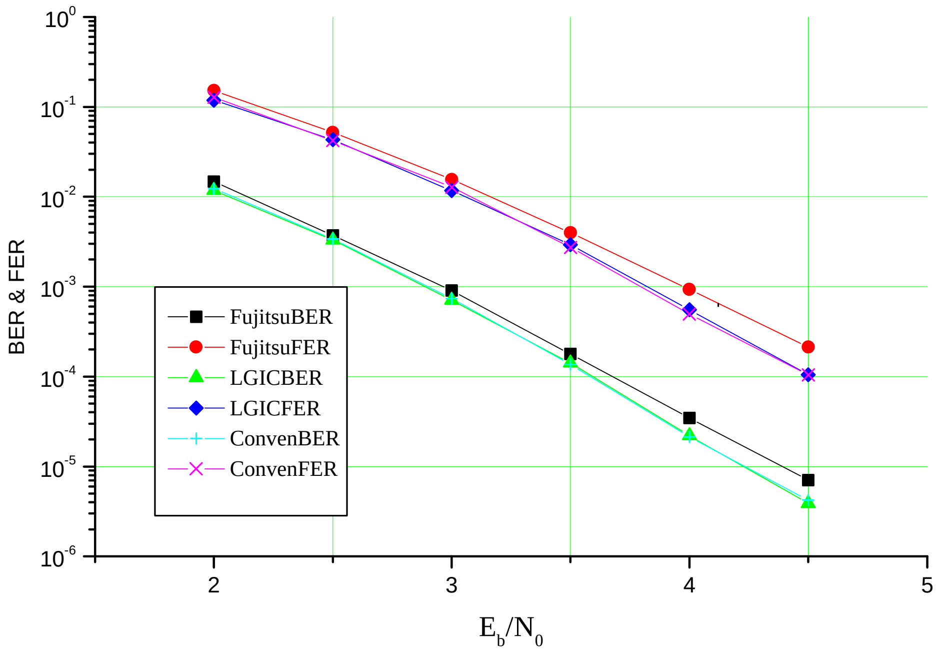

Figure 9. BER and FER of the puncturing schemes for1/2 convolutional code (P = about 0.2 (67 bits punctured))

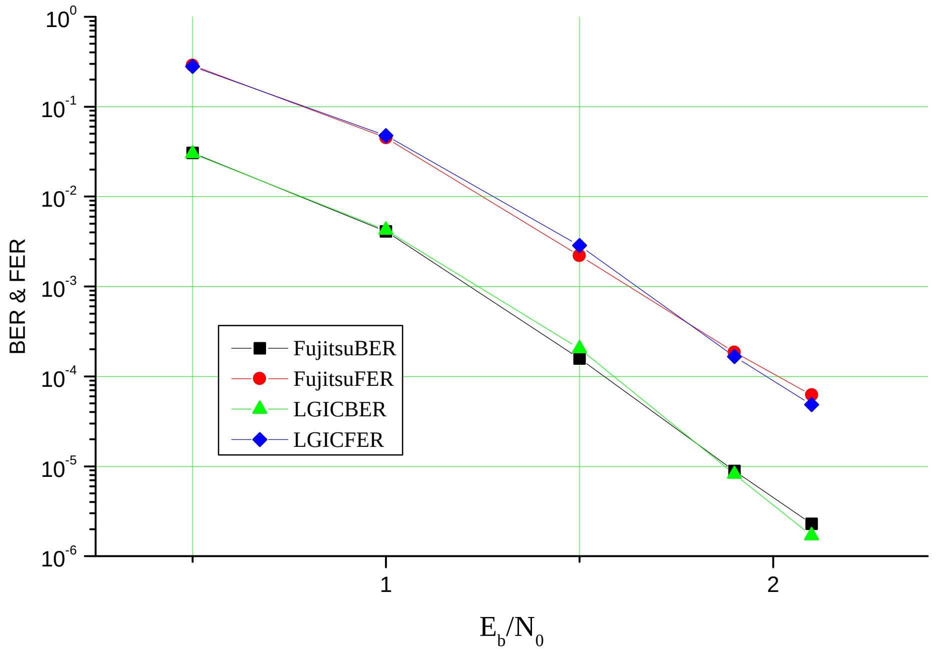

Figure 10. BER and FER of the puncturing schemes for1/3 turbo code (interleaver size = 320, P = about 0.1 (97 bits punctured))

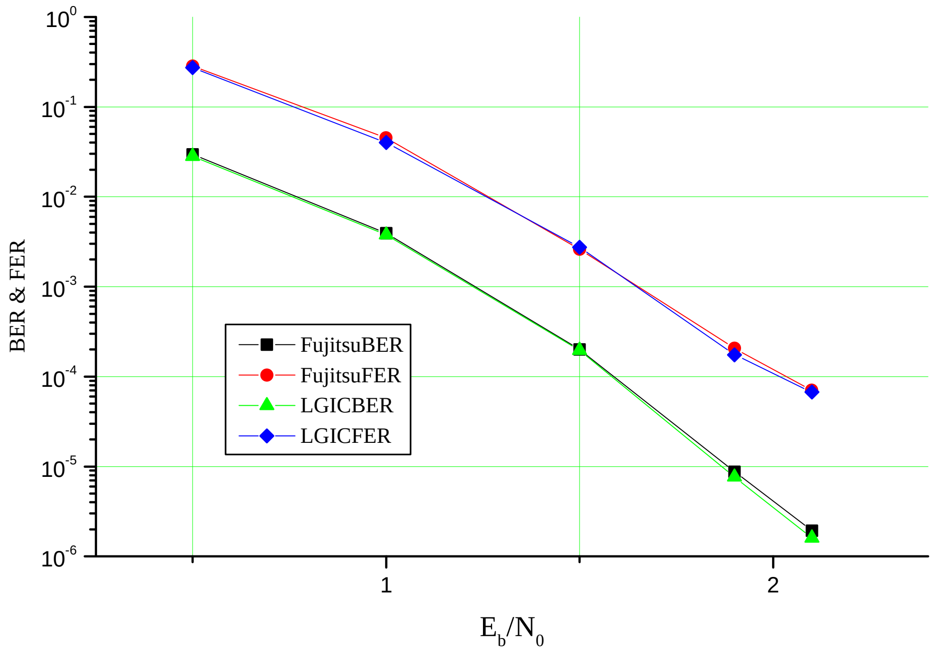

Figure 11. BER and FER of the puncturing schemes for1/3 turbo code

(interleaver size = 321, P = about 0.1 (97 bits punctured))

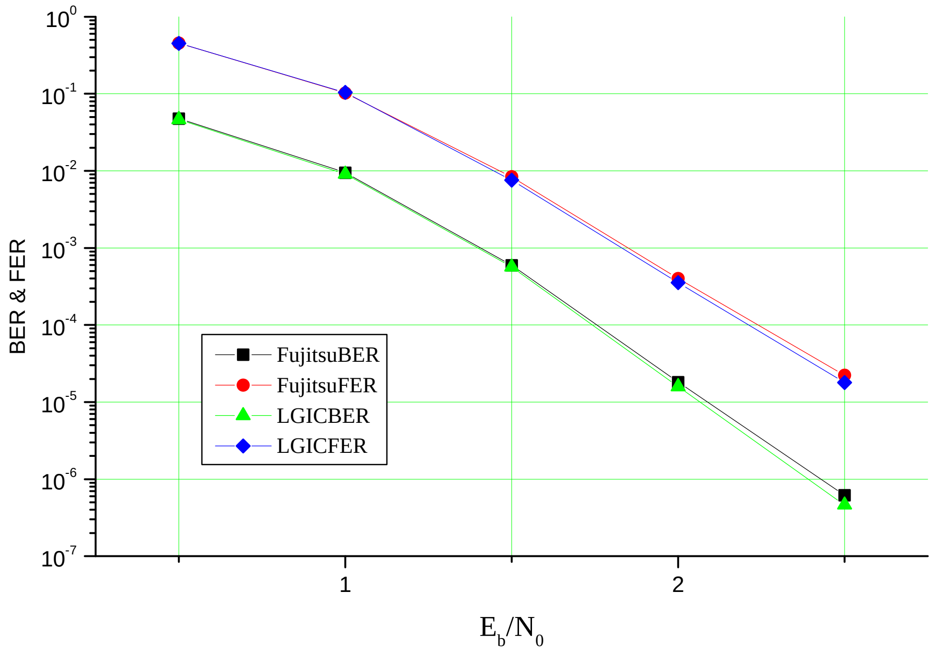

Figure 12. BER and FER of the puncturing schemes for1/3 turbo code

(interleaver size = 320, P = about 0.2 (194 bits punctured))

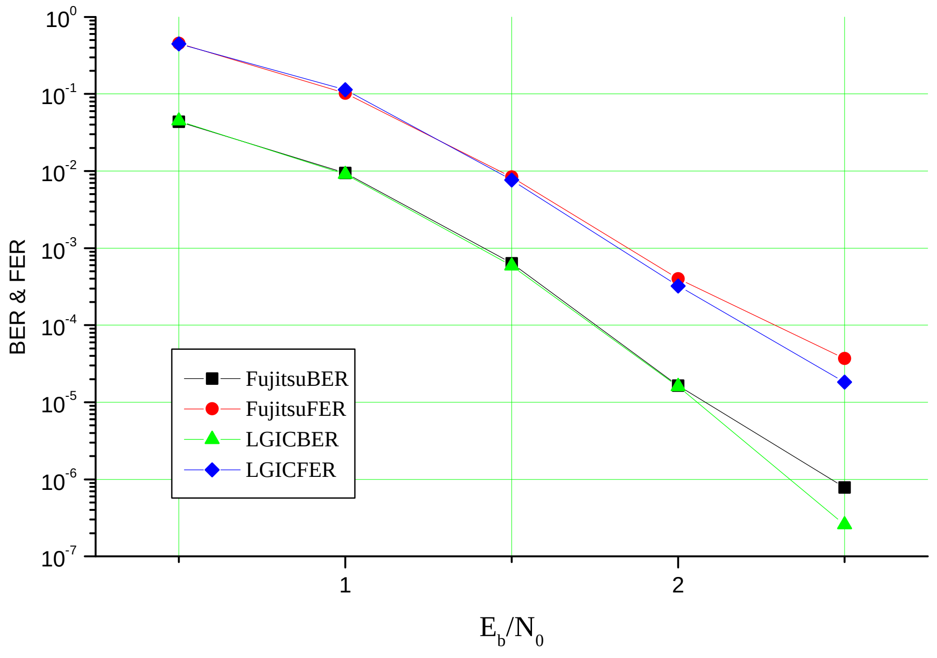

Figure 13. BER and FER of the puncturing schemes for1/3 turbo code

(interleaver size = 321, P = about 0.2 (195 bits punctured))

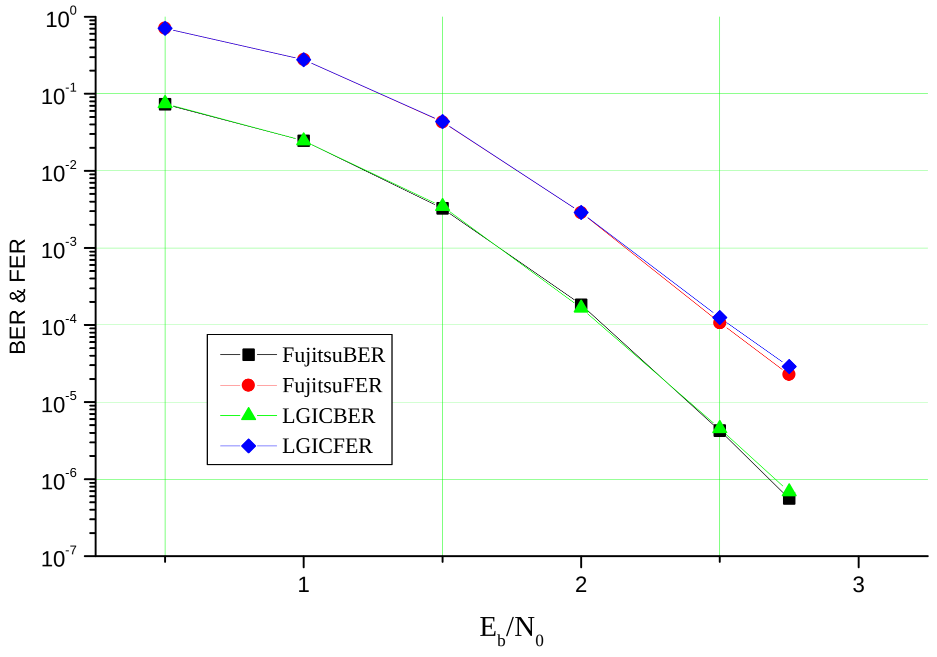

Figure 14. BER and FER of the puncturing schemes for1/3 turbo code

(interleaver size = 320, P = about 0.333 (324 bits punctured))

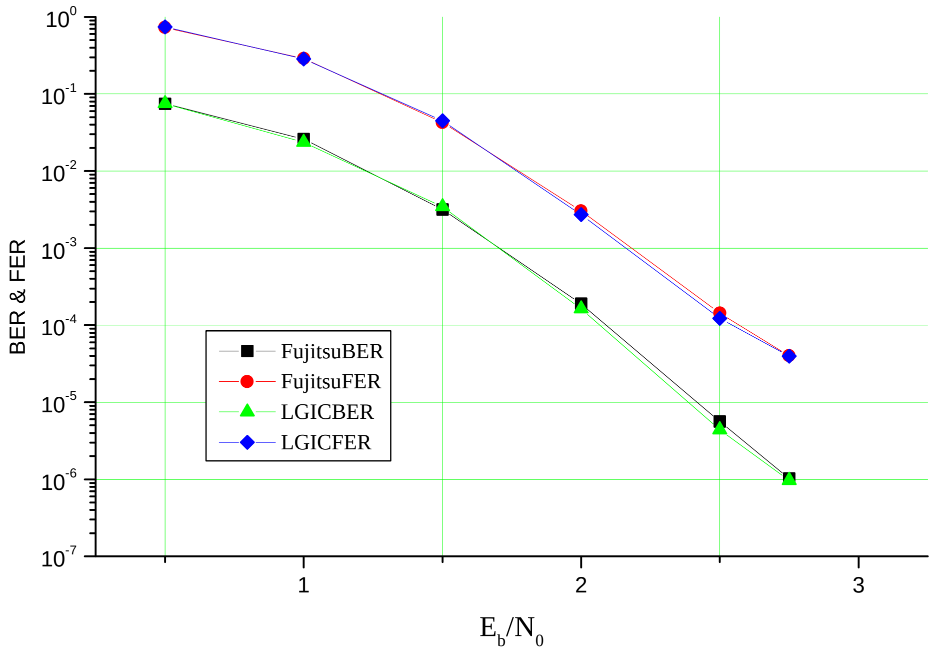

Figure 15. BER and FER of the puncturing schemes for1/3 turbo code (interleaver size = 321, P = about 0.333 (195 bits punctured))

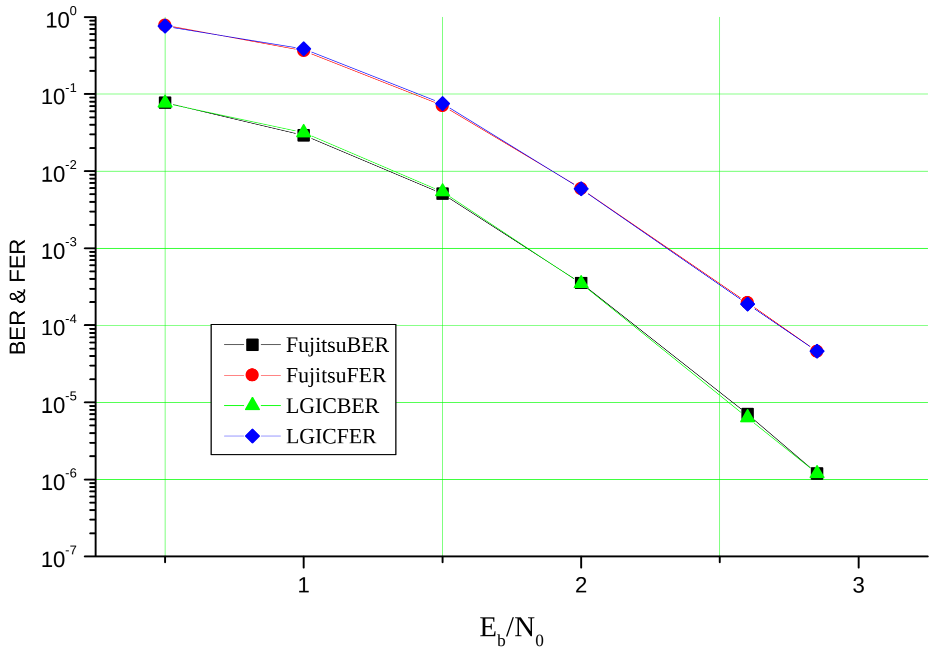

Figure 16. BER and FER of the puncturing schemes for1/2 turbo code (interleaver size = 320, P = about 0.05 (356 bits punctured))

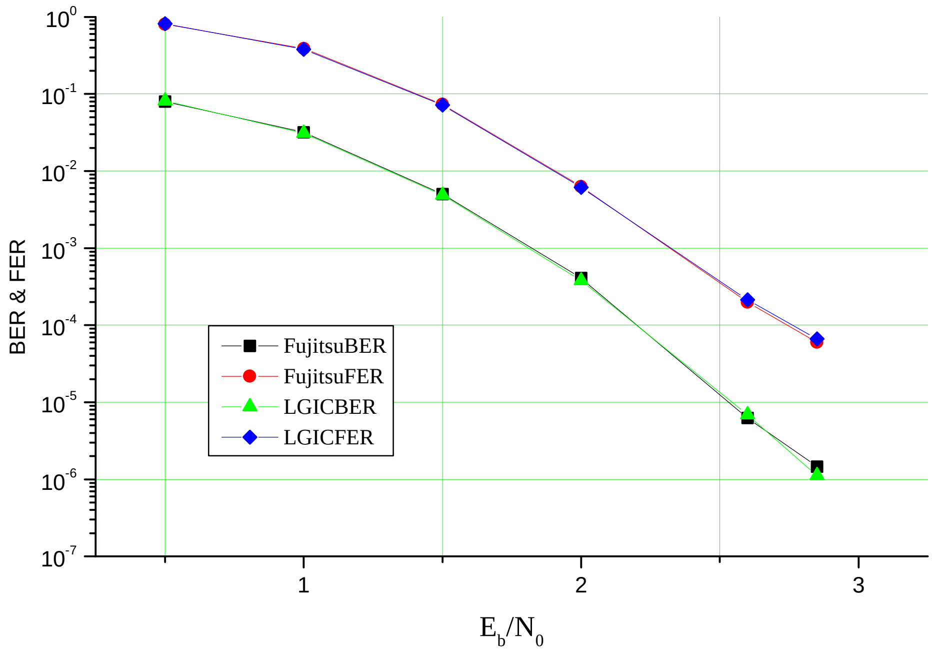

Figure 17. BER and FER of the puncturing schemes for1/2 turbo code

(interleaver size = 321, P = about 0.05 (357 bits punctured))

4. Conclusion

In this document, simulation results of downlink puncturing algorithms are presented for the purspose of comparison. The simulation results show that the application of LGIC scheme to convolutional code does not suffer performance degradation in BER range below 10^{-5} compared to the conventional puncturing algorithm. LGIC scheme has a superiority to the conventional scheme in BER range of 10^{-5} when puncturing ratio is below 0.15. The simulation results also inform that LGIC scheme shows a superiority of performance to Fujitsu scheme in more cases when it is applied to turbo code for the interleaver size of 320, 321.

Thus, it can be concluded that LGIC puncturing algorithm is almost optimal in both channel coding method and is a good candidate for common downlink puncturing algorithm.

Also LGIC downlink puncturing scheme can be easily unified with the code symbol based uplink puncturing algorithm as presented in [2].

5. References

[1] 3GPP TSG RAN WG1 R1-654 Comparison of downlink puncturing algorithms,LGIC

[2] 3GPP TSG RAN WG1 R1-908 Code Symbol Base Uplink Punturing Algorithm,LGIC

[3] 3GPP TSG RAN WG1 Multiplexing and channel coding (FDD) TS 25.212 v2.0.0 (1999-06)

[4] 3GPP TSG RAN WG1 R1-338 Puncturing algorithm for Turbo code, LGIC

[5] 3GPP TSG RAN WG1 R1-698 Further results of rate matching algorithm for downlink, Fujitsu

[6] 3GPP TSG RAN WG1 R1-665 Tail bits and puncturing of rate 1/2 turbo coding, Fujitsu

[7] 3GPP TSG RAN WG1 R1-388 Optimised puncturing scheme for Turbo coding, Fujitsu

[8] 3GPP TSG RAN WG1 R1-456 Additional simulation results of the optimized puncturing scheme for Turbo coding, Fujitsu

[9] ETSI SMG2 UMTS L1 550/98 An improved algorithm for puncturing and repetition - additional results, Siemens