EEE8099 – Information Theory & Coding
EEE8104 – Digital Communications

S. Le Goff

School of Engineering @ Newcastle University

Part 7
Practical Error-Correcting Codes
1993 – Today
Turbo Codes – Part 2

Turbo codes

How to explain the astonishing performance of turbo codes ?

Remember the union bound equation for BPSK, AWGN channel (valid under ML decoding assumption):

$$P_{eb} \leq \sum_{d=d_{min}}^{+\infty} e(d) \cdot \mathrm{erfc}\left(\sqrt{dR_c \frac{E_b}{N_0}}\right).$$

Turbo codes do not necessarily have a large minimum Hamming distance between codewords (d_{min}), but their error coefficients $e(d)$ are very small.

Turbo codes

$$
\begin{aligned}
P_{eb} \le e(d_{min})\cdot\mathrm{erfc}\left(\sqrt{d_{min}R_c\frac{E_b}{N_0}}\right) \\
+\ e(d_{min}+1)\cdot\mathrm{erfc}\left(\sqrt{(d_{min}+1)R_c\frac{E_b}{N_0}}\right) \\
+\ e(d_{min}+2)\cdot\mathrm{erfc}\left(\sqrt{(d_{min}+2)R_c\frac{E_b}{N_0}}\right)+\cdots
\end{aligned}
$$

Turbo codes

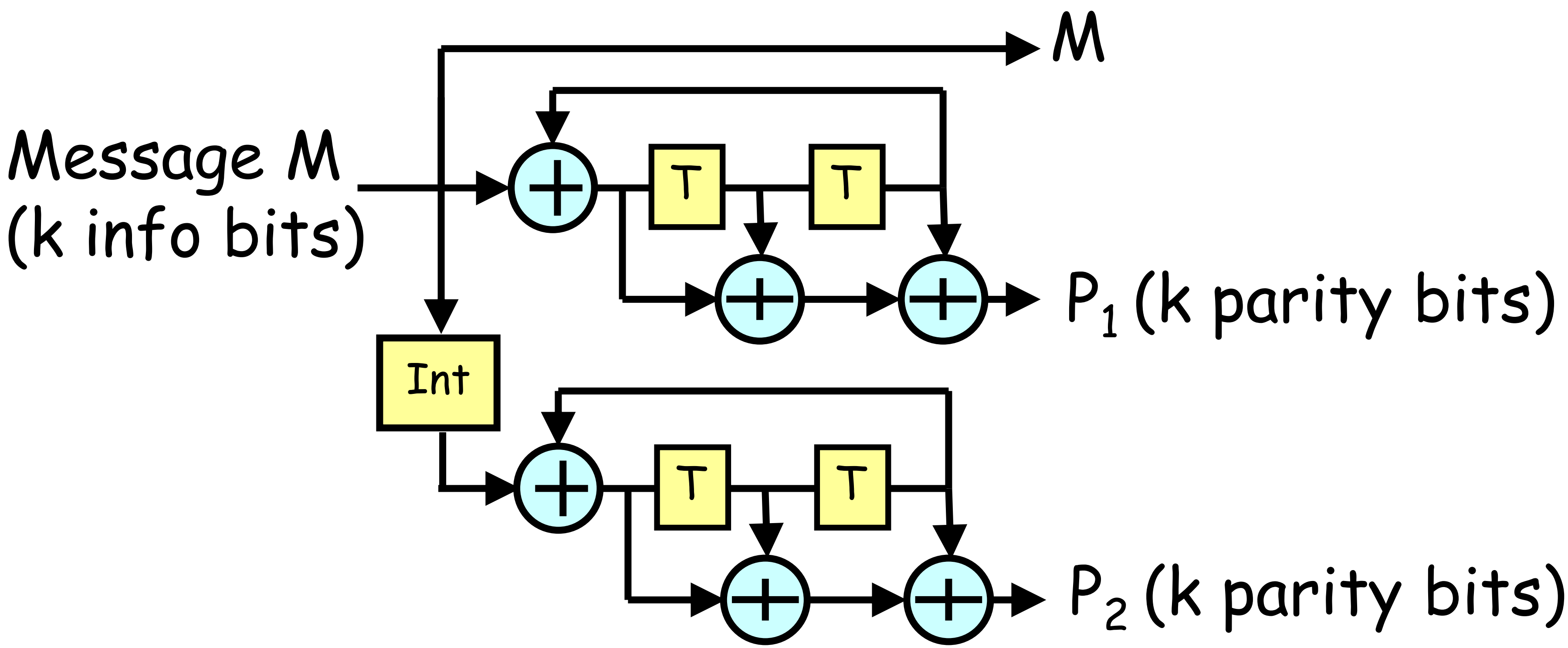
At low SNR, i.e. near the capacity limit, the values of the error coefficients $e(d)$ (especially those associated with the first few terms in the union bound) have a strong influence on the bit error probability P_{eb}.

Therefore, if our goal is to design a near-capacity code, it is probably a good idea to minimise these error coefficients.

Perhaps, minimising these parameters is even more important than maximising the minimum distance d_{min} of the code…

Example: Rate-1/3 (5, 7)//(5, 7) turbo code

Let us study a turbo code designed by parallel concatenation of two rate-1/2, (5, 7) RSC codes.



Example: Rate-1/3 (5, 7)//(5, 7) turbo code

The codeword $C \equiv M + P_1 + P_2$ is composed of k info bits and 2k parity bits.

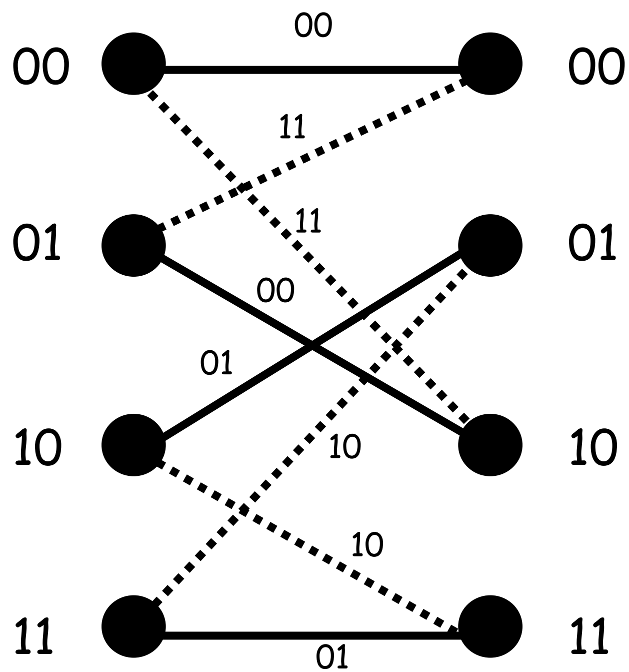
The coding rate is thus $R_c = 1/3$.

Hereafter, we are going to determine the first few terms in the union bound for this turbo code.

Turbo codes are linear codes: we can take the all-zero codeword as a reference and study the Hamming weights of the codewords associated with small-weight messages.

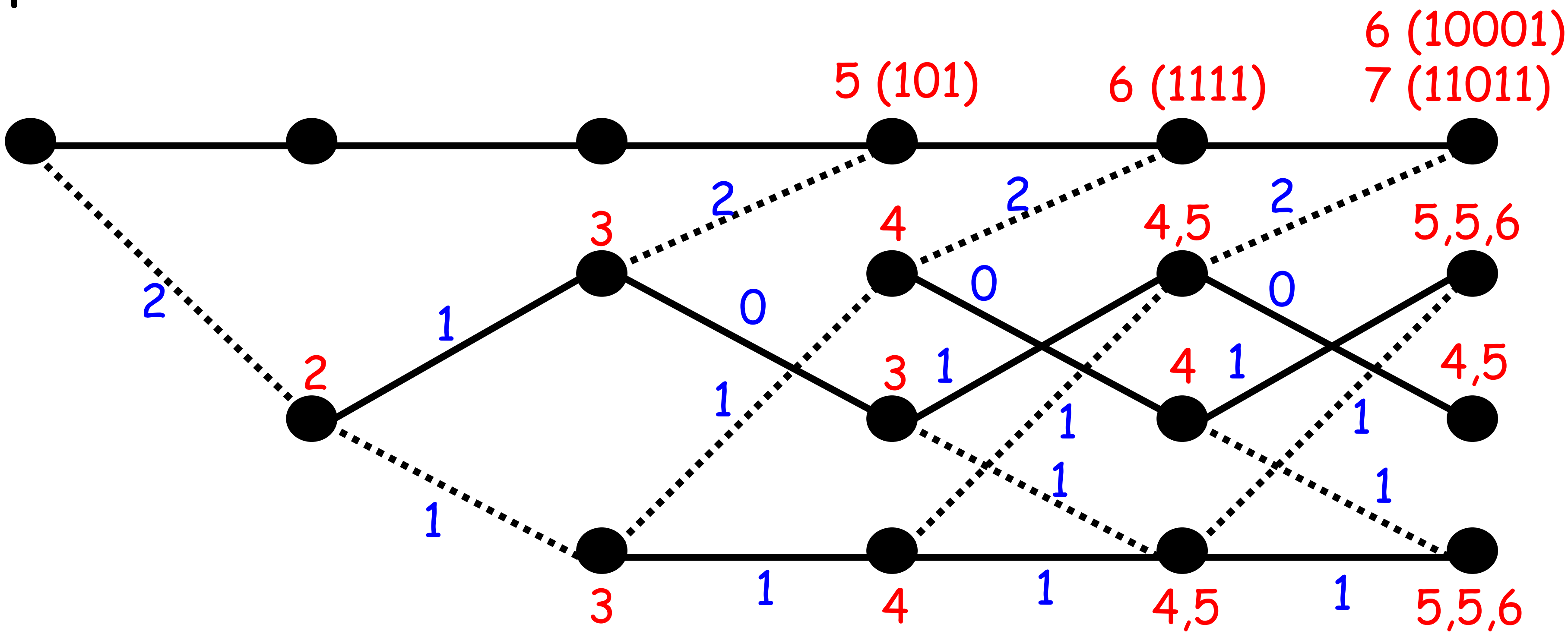
Example: Rate-1/3 (5, 7)//(5, 7) turbo code

The trellis for the first RSC, i.e. the trellis producing the first part (M and P_1) of a codeword, is shown below.

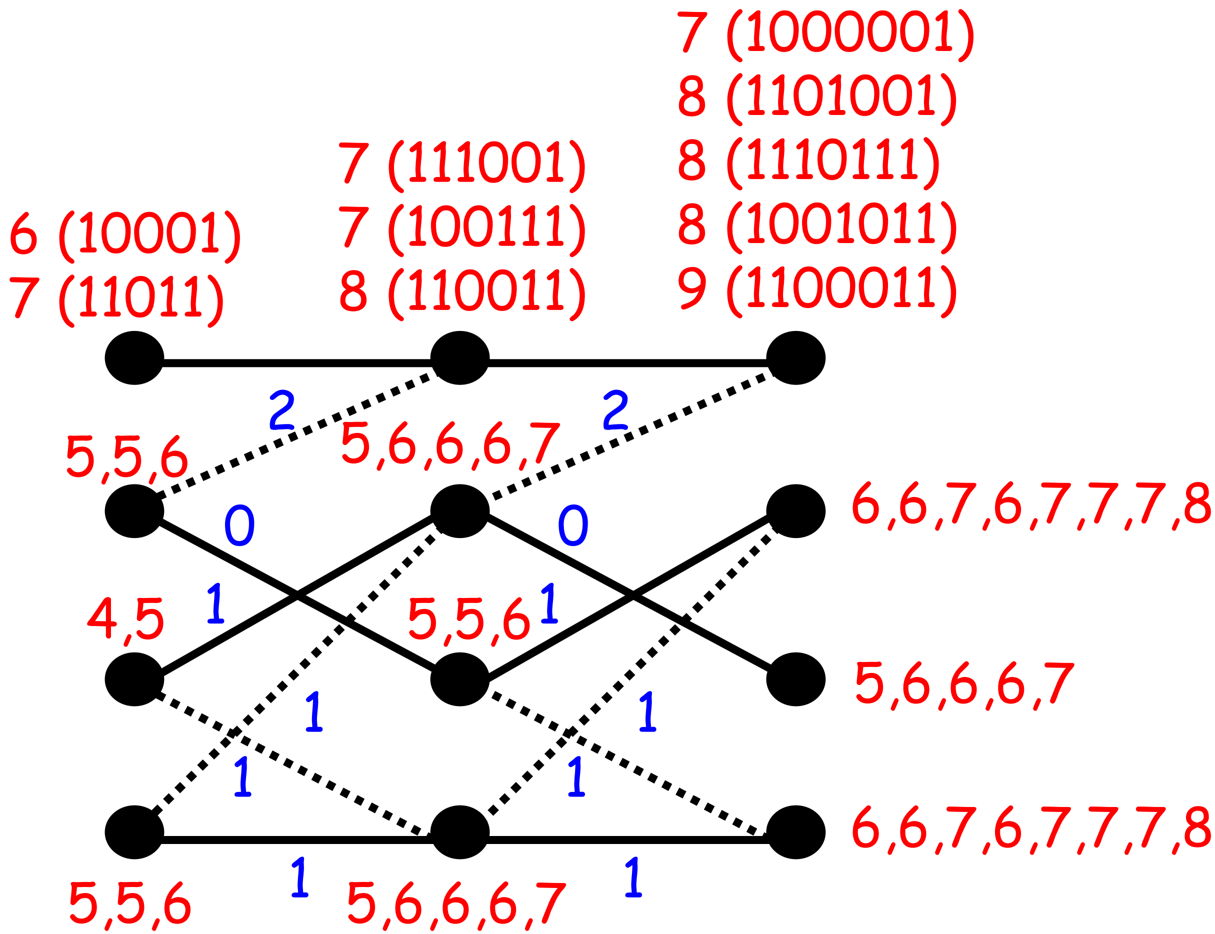


Example: Rate-1/3 (5, 7)//(5, 7) turbo code

Let us, for the first RSC code, perform an exhaustive search for the codewords that are generated when a path leaves the all-zero codeword.



Example: Rate-1/3 (5, 7)//(5, 7) turbo code



Example: Rate-1/3 (5, 7)//(5, 7) turbo code

We have obtained the following results:

- *A* weight-2 message with the (101) pattern leads to a codeword at distance d = 5 from the all-zero codeword.
- *A* weight-2 message (10001) *as* well *as* a weight-4 message (1111) both lead to a weight-6 codeword.
- *A* weight-2 message (1000001) and three weight-4 messages (11011, 111001, 100111) lead to a weight-7 codeword.
- We have not *gone* far enough in the trellis to determine all the patterns that lead to codewords at distances d > 7 from the all-zero codeword.

Example: Rate-1/3 (5, 7)//(5, 7) turbo code

The minimum distance between codewords for the (5, 7) RSC code is $d_{min} = 5$.

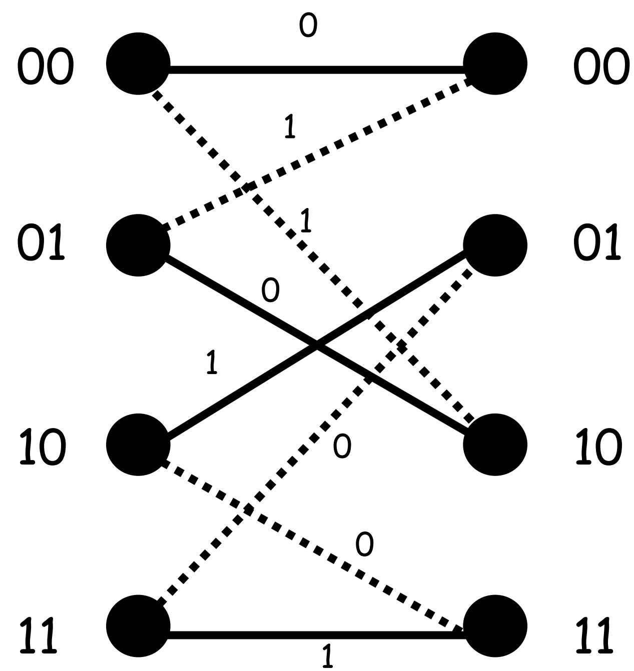
A weight-2 message in which both 1s are separated by L 0s generates a codeword at distance $d = \frac{L+9}{2}$.

We recall that, with RSC encoders, weight-2 messages can generate codewords at large distance from the all-zero codeword. This is due to the “recursivity” of the encoder.

A NRC code would not exhibit this characteristic as, with such an encoder, a small-weight message always generates a small-weight codeword.

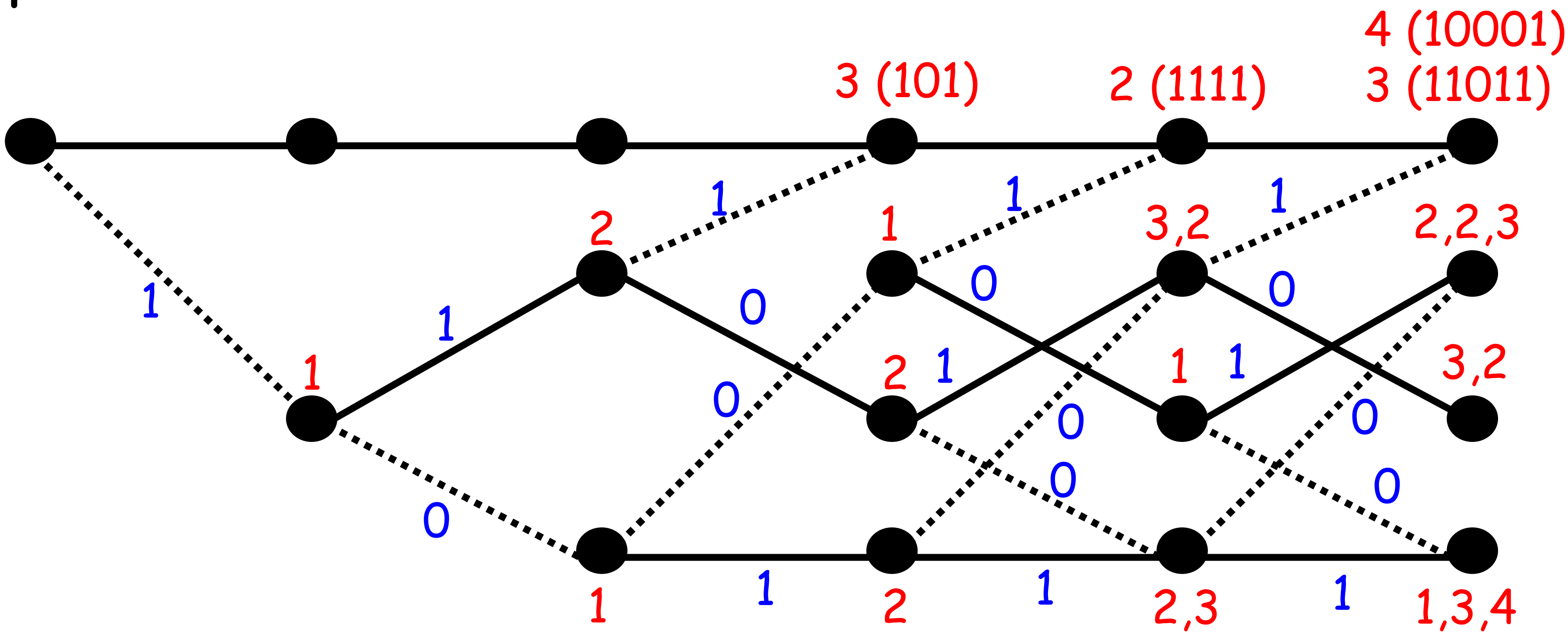
Example: Rate-1/3 (5, 7)//(5, 7) turbo code

Let us now turn our attention to the second trellis, i.e. the trellis that is used to generate the second part (P_2) of the codeword.

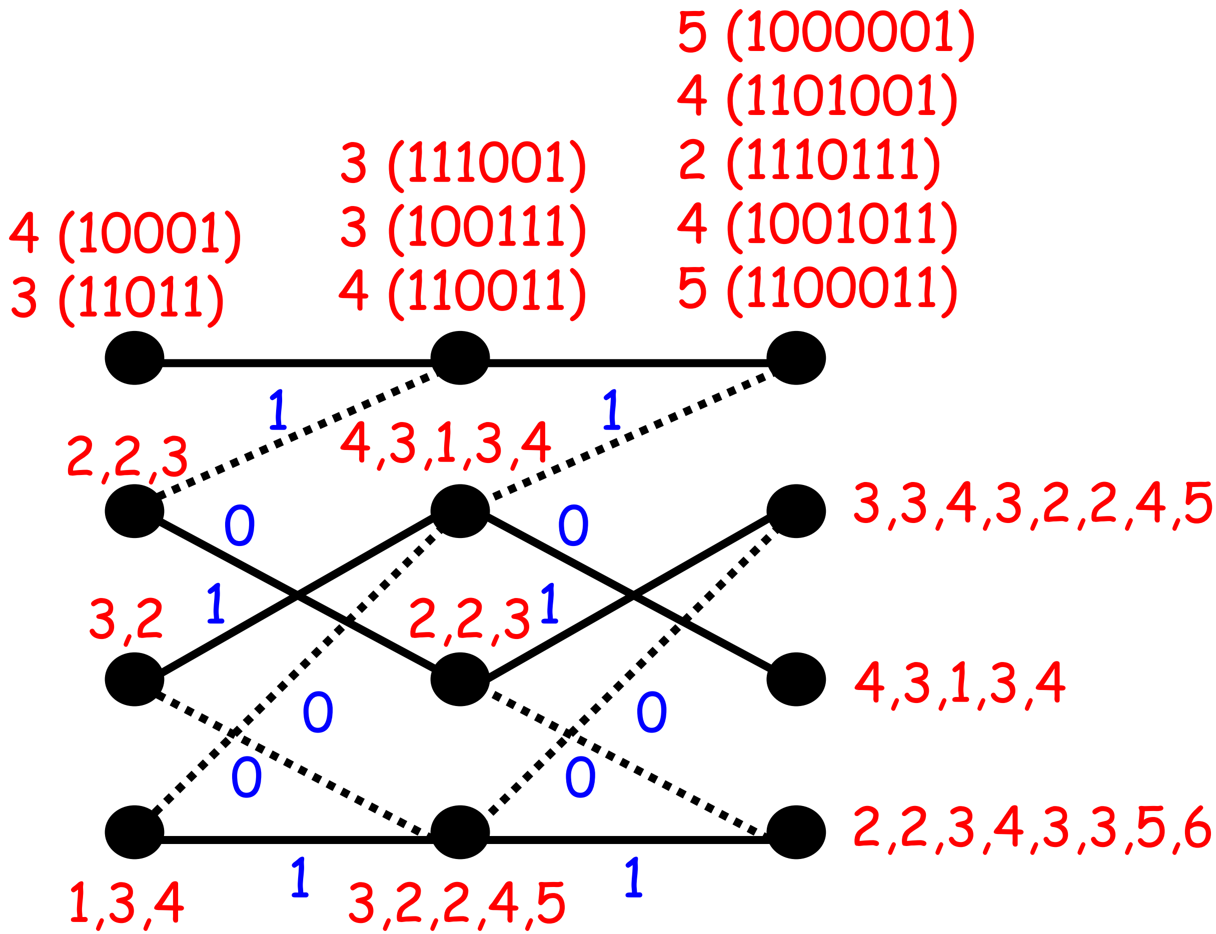


Example: Rate-1/3 (5, 7)//(5, 7) turbo code

Let us, for this second trellis, perform an exhaustive search for the codewords that are generated when a path leaves the all-zero codeword.



Example: Rate-1/3 (5, 7)//(5, 7) turbo code



Example: Rate-1/3 (5, 7)//(5, 7) turbo code

Combining the results of the searches carried out for both trellises, we reach the following conclusions:

Weight-2 messages:
Message …101…: d = 5 for RSC 1 and d = 3 for RSC 2
Message …10001…: d = 6 for RSC 1 and d = 4 for RSC 2
Message …1000001…: d = 7 for RSC 1 and d = 5 for RSC 2

In general, a weight-2 message in which both 1s are separated by L 0s generates a codeword at distance $d = \frac{L+9}{2}$ for RSC 1 and $d = \frac{L+5}{2}$ for RSC 2.

Example: Rate-1/3 (5, 7)//(5, 7) turbo code

Weight-4 messages:
Message …1111…: d = 6 for RSC 1 and d = 2 for RSC 2
Message …11011…: d = 7 for RSC 1 and d = 3 for RSC 2
Message …100111…: d = 7 for RSC 1 and d = 3 for RSC 2
Message …111001…: d = 7 for RSC 1 and d = 3 for RSC 2

All these results appear to be sufficient if we only wish to determine the first few terms of the union bound for the turbo code considered here.

Example: Rate-1/3 (5, 7)//(5, 7) turbo code

Consider now all possibilities:

- M = (…101…) before interleaving becomes M = (…101…) after interleaving: d = 5 + 3 = 8.
- M = (…101…) before interleaving becomes M = (…10001…) after interleaving: d = 5 + 4 = 9.
- M = (…101…) before interleaving becomes M = (…1000001…) after interleaving: d = 5 + 5 = 10.
- M = (…10001…) before interleaving becomes M = (…101…) after interleaving: d = 6 + 3 = 9.
- M = (…10001…) before interleaving becomes M = (…10001…) after interleaving: d = 6 + 4 = 10.
- And so on.

Example: Rate-1/3 (5, 7)//(5, 7) turbo code

Consider a weight-2 message in which both 1s are separated by L 0s before interleaving becomes a message in which both 1s are separated by L′ 0s after interleaving.

Such message being denoted as M(L, L′).

With this message, the turbo encoder generates a codeword at distance

$$d(L, L') = \frac{L+9}{2} + \frac{L'+5}{2} = \frac{L+L'+14}{2}$$

from the all-zero codeword.

Example: Rate-1/3 (5, 7)//(5, 7) turbo code

Consider now the weight-4 messages.

- M = (…1111…) before interleaving becomes M = (…1111…) after interleaving: d = 6 + 2 = 8.
- M = (…1111…) before interleaving becomes M = (…11011…) after interleaving: d = 6 + 3 = 9.
- M = (…1111…) before interleaving becomes M = (…100111…) after interleaving: d = 6 + 3 = 9.
- M = (…1111…) before interleaving becomes M = (…111001…) after interleaving: d = 6 + 3 = 9.
- M = (…11011…) before interleaving becomes M = (…1111…) after interleaving: d = 7 + 2 = 9.
- And so on.

Example: Rate-1/3 (5, 7)//(5, 7) turbo code

At this stage, we have clearly shown that no codeword has a Hamming weight lower than 8.

As a result, the minimum distance between codewords of our turbo code is $d_{min} = 8$.

This is not such a high value given the rather low coding rate of this turbo code.

Let us now determine the expression of the error coefficient $e(d = 8)$.

Example: Rate-1/3 (5, 7)//(5, 7) turbo code

To determine the expression of $e(d = 8)$, we must take into account two configurations:

(1) A pattern '101' before interleaving becomes '101' after interleaving;

(2) A pattern '1111' before interleaving becomes '1111' after interleaving.

Let us consider the first possibility.

How many messages M satisfy condition (1) ?

Example: Rate-1/3 (5, 7)//(5, 7) turbo code

Focusing first on the first RSC encoder, the number of weight-2 messages with the pattern '101' in them is given by the number of possible permutations of one element (the pattern '101') in a word of (k-2) elements.

This number is thus equal to the binomial coefficient

$$\binom{k-2}{1} = k - 2.$$

For each of these (k-2) messages, what is the probability that, after random interleaving, we still see the same pattern '101' at the input of the second RSC encoder?

Example: Rate-1/3 (5, 7)//(5, 7) turbo code

The number of possible permutations of 2 bits in a word of k bits is given by the binomial coefficient

$$\binom{k}{2} = \frac{k(k-1)}{2} \approx \frac{k^2}{2}.$$

Among these $\binom{k}{2}$ possible permutations, how many of them have the '101' pattern?

The answer is $\binom{k-2}{1} = k - 2$, as shown earlier.

Example: Rate-1/3 (5, 7)//(5, 7) turbo code

Thus, the probability to have a message with the pattern '101' after interleaving is given by

$$\Pr(101) = \frac{\binom{k-2}{1}}{\binom{k}{2}} \approx \frac{2(k-2)}{k^2}.$$

Finally, the average number of messages with a pattern '101' before interleaving that remains a pattern '101' after interleaving is

$$\Pr(101) \cdot (k-2) \approx \frac{2(k-2)^2}{k^2} \approx 2.$$

Example: Rate-1/3 (5, 7)//(5, 7) turbo code

This corresponds to a value of the parameter $w_{d_{min}}$ given by $w_{d_{min}} \approx 2 \times 2 = 4$.

Now, we must add to this number the contribution of the weight-4 messages with a pattern '1111' before interleaving that remains a pattern '1111' after interleaving.

There are $\binom{k-3}{1} = (k-3)$ different messages with a pattern '1111'.

Example: Rate-1/3 (5, 7)//(5, 7) turbo code

The number of possible permutations of 4 bits in a word of k bits is given by the binomial coefficient

$$\binom{k}{4} = \frac{k(k-1)(k-2)(k-3)}{2\times3\times4} \approx \frac{k^4}{24}.$$

Among these $\binom{k}{4}$ possible permutations, how many of them have the '1111' pattern?

The answer is $\binom{k-3}{1} = k - 3$, as shown earlier.

Example: Rate-1/3 (5, 7)//(5, 7) turbo code

Thus, the probability to have a weight-4 message with the pattern '1111' after interleaving is given by

$$\Pr(1111) = \frac{\binom{k-3}{1}}{\binom{k}{4}} \approx \frac{24(k-3)}{k^4}.$$

Finally, the average number of messages with a pattern '1111' before interleaving that remains a pattern '1111' after interleaving is

$$\Pr(1111)\cdot(k-3) \approx \frac{24(k-3)^2}{k^4} \approx \frac{24}{k^2}.$$

Example: Rate-1/3 (5, 7)//(5, 7) turbo code

This corresponds to a value of the parameter $w_{d_{min}}$ given by $w_{d_{min}} \approx 4 \times \frac{24}{k^2} = \frac{96}{k^2}$.

The error coefficient $e(d = 8)$ is finally given by $e(d = 8) \approx \frac{4+\frac{96}{k^2}}{2k} \approx \frac{2}{k}$.

This result shows us that:

(1) The error coefficient is very small (something we have never seen so far with any other family of codes);

Example: Rate-1/3 (5, 7)//(5, 7) turbo code

(2) The error coefficient is inversely proportional to the message length k;

(3) One can only focus on weight-2 messages and actually ignore the impact of the weigth-4 (and beyond) messages on the union bound. This simplifies the analysis of our code.

This last point is due to the fact that N 1s ($N \geq 2$) initially grouped together at a certain location in a message M are very likely to be scattered by random interleaving.

Example: Rate-1/3 (5, 7)//(5, 7) turbo code

The probability of these N 1s to be scattered strongly depends on the value of N.

For instance, the probability that N 1s will be successive bits in a k-bit word is given by

$$\Pr(N) = \frac{\binom{k-N+1}{1}}{\binom{k}{N}} = \frac{N!(k-N+1)!}{k!} = \frac{N!}{\prod_{i=0}^{N-2}(k-i)}.$$

Assuming that $N \ll k$, we can write $\Pr(N) \approx \frac{N!}{k^{N-1}}$.

With $N = 2$, we obtain $\Pr(N = 2) \approx \frac{2}{k}$.

Example: Rate-1/3 (5, 7)//(5, 7) turbo code

With $N = 4$, we obtain $\Pr(N = 4) \approx \frac{24}{k^3}$.

With $N = 6$, we obtain $\Pr(N = 6) \approx \frac{720}{k^5}$.

The codewords at small distances from the all-zero codeword can only be generated by messages in which the 1s are grouped together both <u>before and after interleaving</u>.

The few computations shown above clearly indicate that this is by far more likely to happen with weight-2 messages than with any other types of messages.

Example: Rate-1/3 (5, 7)//(5, 7) turbo code

But, even with these weight-2 messages, this is still unlikely to happen.

At the turbo encoder output, there are codewords with small Hamming weights, thus resulting in a rather small distance d_{min}, but there are (fortunately) very few of them, thus resulting in low values for the error coefficients $e(d_{min})$, $e(d_{min} + 1)$, $e(d_{min} + 2)$, etc.

This is the secret behind the near-capacity error performance of turbo codes.

Example: Rate-1/3 (5, 7)//(5, 7) turbo code

Now, how to find the expressions of the error coefficients $e(d = 9)$, $e(d = 10)$, $e(d = 11)$, … for our turbo code ?

We have just discovered that we only need to consider weight-2 messages from now on (as long as we stick to the first few error coefficients, which should be more than enough).

We can use an interesting expression derived earlier.

Example: Rate-1/3 (5, 7)//(5, 7) turbo code

The expression shown below is applicable to any weight-2 message, denoted as $M(L, L')$, in which both 1s are separated by L 0s before interleaving and separated by L′ 0s after interleaving.

With this message, the turbo encoder generates a codeword at distance

$$d(L, L') = \frac{L+9}{2} + \frac{L'+5}{2} = \frac{L+L'+14}{2}$$

from the all-zero codeword.

Example: Rate-1/3 (5, 7)//(5, 7) turbo code

Messages for which L = L′ = 1 produce the codewords at distance $d_{min} = 8$ from the all-zero codeword.

Messages for which (L = 1, L′ = 3) or (L = 3, L′ = 1) produce the codewords at distance $d_{min} = 9$ from the all-zero codeword.

Messages for which (L = 1, L′ = 5) or (L = 5, L′ = 1) or (L = L′ = 3) produce the codewords at distance $d_{min} = 10$ from the all-zero codeword.

And so on.

Example: Rate-1/3 (5, 7)//(5, 7) turbo code

We must determine the number of messages M(L, L').

First, how many messages M in which both 1s are separated by L 0s are there ?

This number is *given* by the number of possible permutations of one element (the pattern 1 followed by L 0s followed by 1) in a word of (k-L-1) elements.

This number is thus equal to the binomial coefficient

$$\binom{k-L-1}{1} = k - L - 1.$$

Example: Rate-1/3 (5, 7)//(5, 7) turbo code

For each of these (k-L-1) messages, what is the probability that, after random interleaving, both 1s are separated by L′ 0s?

The number of possible permutations of 2 bits in a word of k bits is given by the binomial coefficient

$$\binom{k}{2} = \frac{k(k-1)}{2} \approx \frac{k^2}{2}.$$

The number of configurations for which two 1s are separated by L′ 0s is (k-L′-1).

Example: Rate-1/3 (5, 7)//(5, 7) turbo code

Thus, the probability to have a sequence in which two 1s are separated by L′ 0s after interleaving is given by

$$\Pr(L') = \frac{k - L' - 1}{\binom{k}{2}} \approx \frac{2(k - L' - 1)}{k^2}.$$

Finally, the average number of messages M(L, L′) is

$$\Pr(L')\cdot(k - L - 1) \approx \frac{2(k - L' - 1)\cdot(k - L - 1)}{k^2}.$$

We are here concerned with the first few terms in the union bound, i.e. the small-weight codewords.

Example: Rate-1/3 (5, 7)//(5, 7) turbo code

Since small-weight codewords are always generated by encoding low-weight messages with small values of L and L', we can now assume that $k \gg L + 1$ and $k \gg L' + 1$.

Under this assumption, the number of messages $M(L, L')$ becomes equal to

$$\frac{2(k - L' - 1)\cdot(k - L - 1)}{k^2} \approx \frac{2k^2}{k^2} = 2.$$

This number does not depend on L and L'.

Example: Rate-1/3 (5, 7)//(5, 7) turbo code

A weight-8 codeword is obtained when M is such that L = L′ = 1. The contribution of this message to the union bound is thus represented by the error coefficient $e(d=8) \approx \frac{2\times 2}{2k} = \frac{2}{k}$.

A weight-9 codeword is obtained when M is such that L + L′ = 4, i.e. (L = 1, L′ = 3) or (L = 3, L′ = 1). The contribution of both possibilities to the union bound is thus represented by the error coefficient $e(d=9) \approx \frac{2}{k} + \frac{2}{k} = \frac{4}{k}$.

Example: Rate-1/3 (5, 7)//(5, 7) turbo code

A weight-10 codeword is obtained when the message is such that L + L′ = 6, which corresponds to the 3 following possibilities: (L = 1, L′ = 5) or (L = 5, L′ = 1) or (L = 3, L′ = 3) .

The contribution of these three weight-2 messages to the union bound is thus represented by the error coefficient $e(d = 10) \approx \frac{2}{k} + \frac{2}{k} + \frac{2}{k} = \frac{6}{k}$.

Example: Rate-1/3 (5, 7)//(5, 7) turbo code

A weight-11 codeword is obtained when the message is such that L + L′ = 8, which corresponds to the 4 following possibilities: (L = 1, L′ = 7) or (L = 7, L′ = 1) or (L = 3, L′ = 5) or (L = 5 and L′ = 3).

The contribution of these four weight-2 messages to the union bound is thus represented by the error coefficient $e(d = 11) \approx \frac{2}{k} + \frac{2}{k} + \frac{2}{k} + \frac{2}{k} = \frac{8}{k}$.

And so on.

Example: Rate-1/3 (5, 7)//(5, 7) turbo code

These results show that:

(1) The minimal distance of this rate-1/3 turbo code is only equal to 8, and thus not much larger than that of a rate-1/2 (5, 7) RSC code alone for which $d_{min} = 5$.

(2) The error coefficients are much smaller than those of a traditional code (Hamming, convolutional, etc). This phenomenon is due to the use of a random interleaver inside the encoder and known as called "spectral thinning".

Example: Rate-1/3 (5, 7)//(5, 7) turbo code

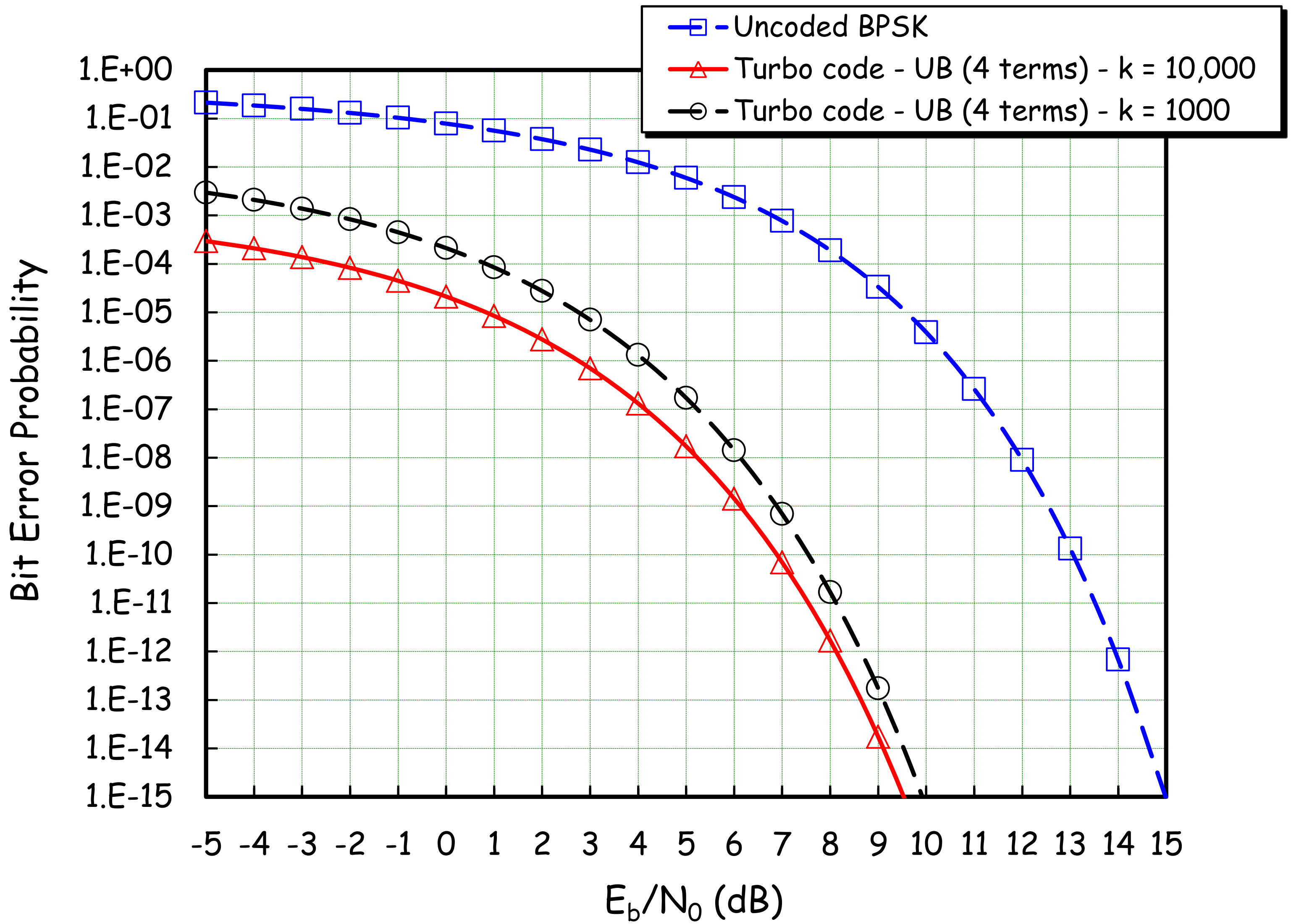
(3) As k is increased, the error coefficient values become even smaller. This is often referred to as the "interleaver gain".

Finally, by putting all these results together, we obtain the first few terms in the union bound:

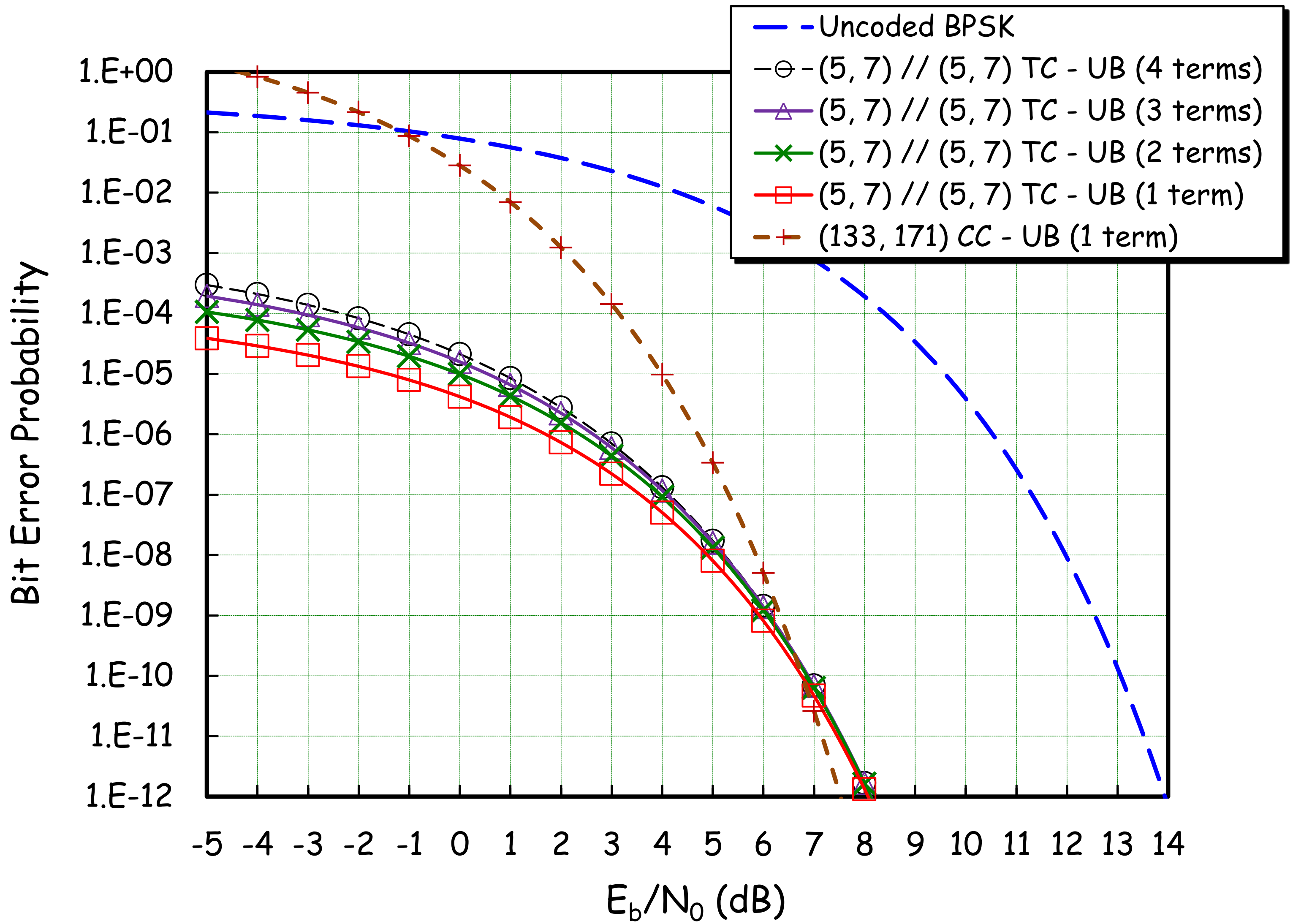
$$P_{eb} \approx \frac{2}{k}\cdot\mathrm{erfc}\left(\sqrt{\frac{8}{3}\frac{E_b}{N_0}}\right) + \frac{4}{k}\cdot\mathrm{erfc}\left(\sqrt{\frac{9}{3}\frac{E_b}{N_0}}\right)$$

$$+\frac{6}{k}\cdot\mathrm{erfc}\left(\sqrt{\frac{10}{3}\frac{E_b}{N_0}}\right) + \frac{8}{k}\cdot\mathrm{erfc}\left(\sqrt{\frac{11}{3}\frac{E_b}{N_0}}\right) + \cdots$$

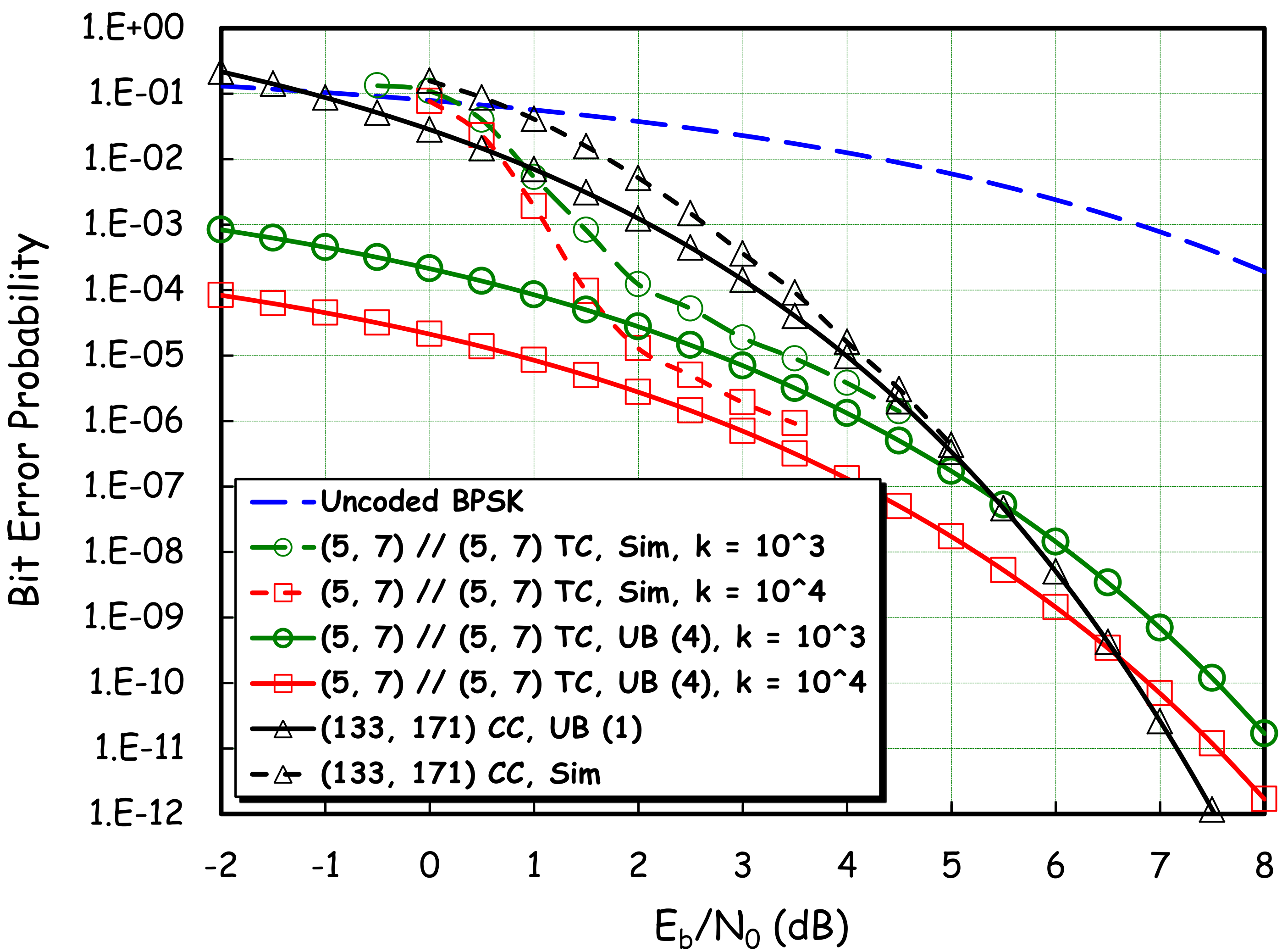
Rate-1/3 (5, 7)//(5, 7) turbo code.
The larger the random interleaver, the better the code.



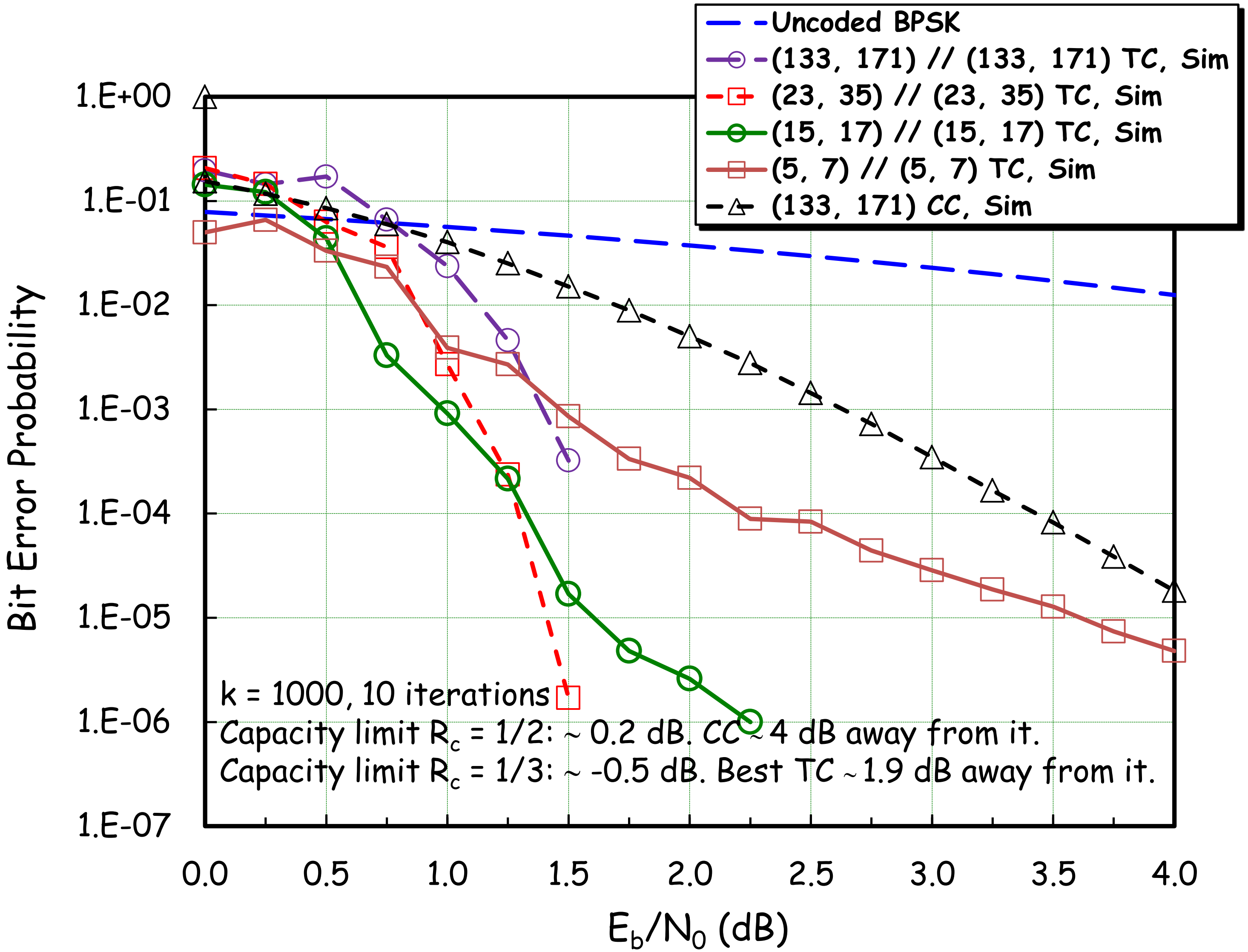
Union bound of the rate-1/3 (5, 7)//(5, 7) turbo code with k = 10,000 - (Unfair) comparison with a "traditional" code (rate-1/2, K = 7, (133, 171) CC)



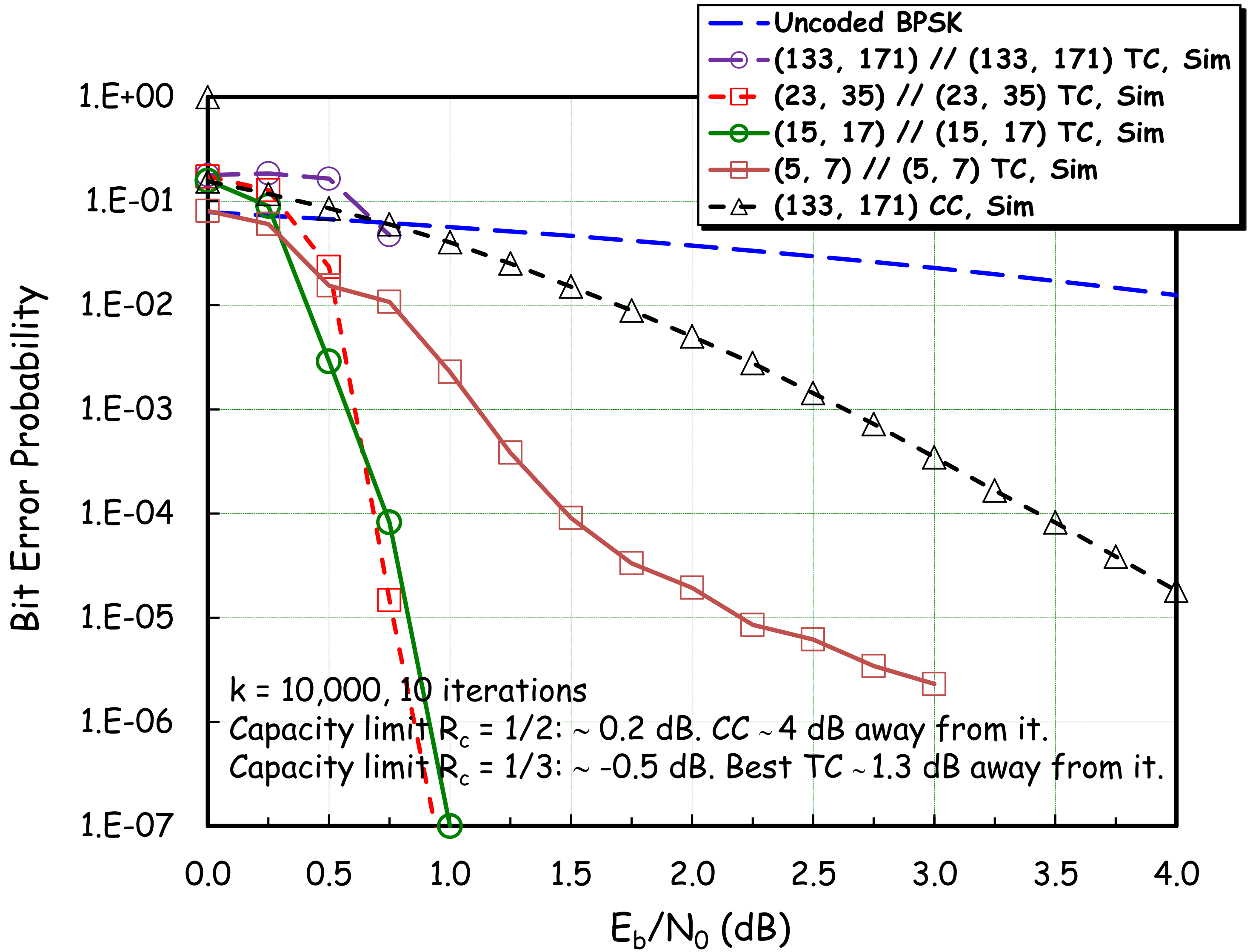
Comparison between union bound and simulation results → the union bound of a turbo code is not quite as tight as it is for a CC. Due to the non-ML decoding of turbo codes?



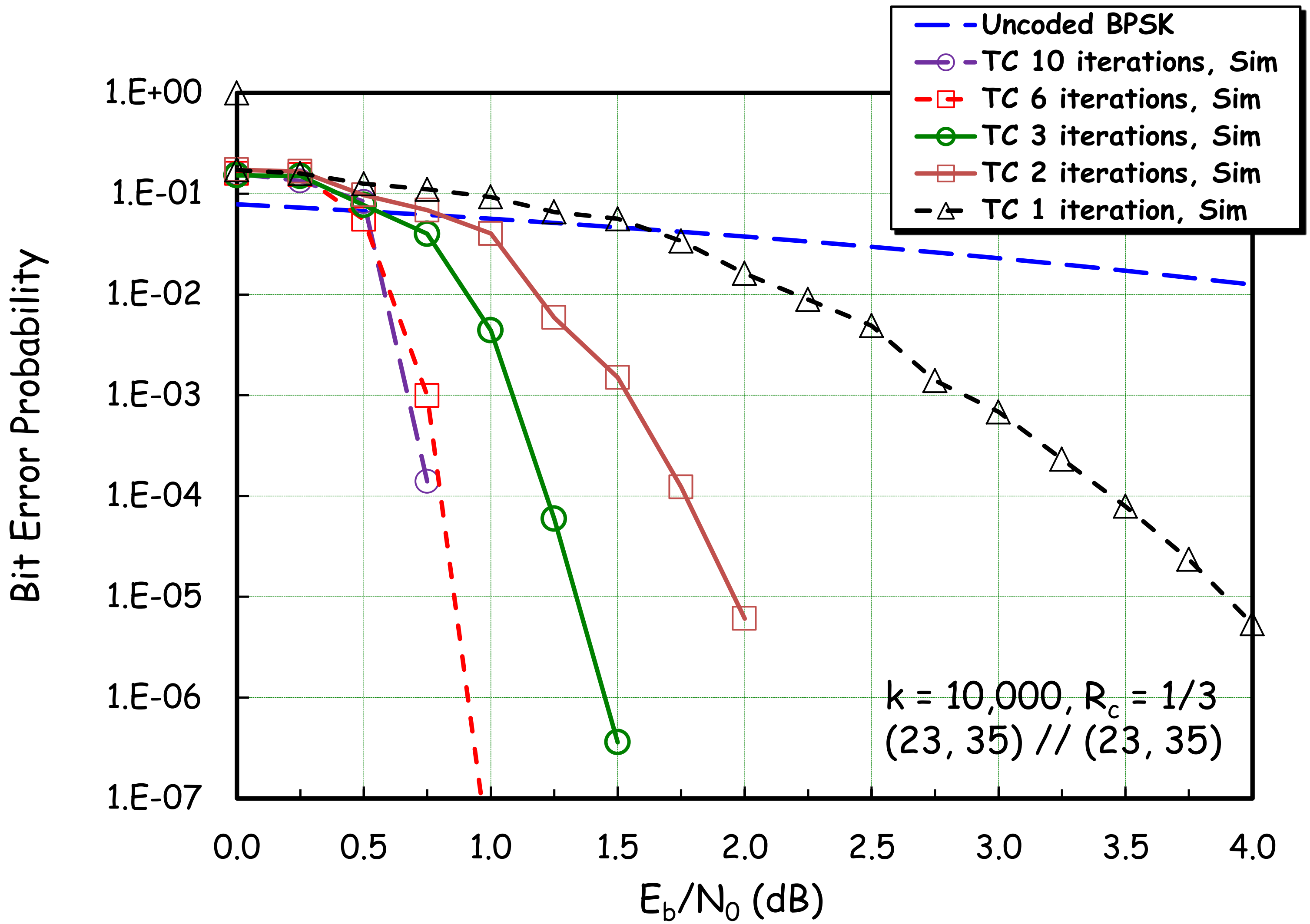
Comparison between several rate-1/3 turbo codes built using different RSC codes, for k = 1000. Best results obtained with 8- and 16-state RSC codes.



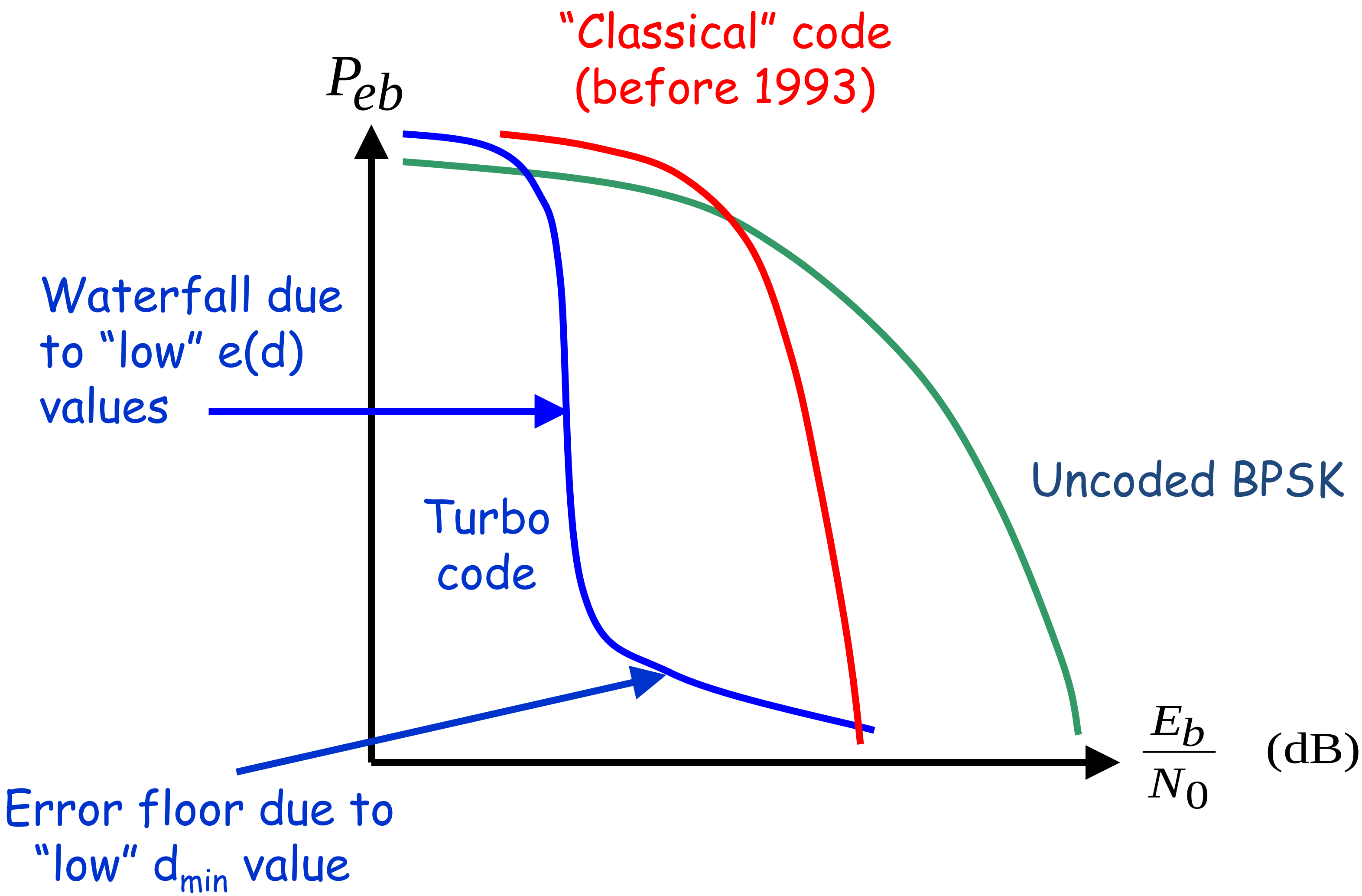
Comparison between several rate-1/3 turbo codes built using different RSC codes, for k = 10,000. Best results obtained with 8- and 16-state RSC codes.



Error performance is improved by increasing the number of decoding iterations, but there is no real need to use more than 5 – 10 iterations in practice.



Turbo codes


"Classical" code
(before 1993)
P_{eb}
Waterfall due
to "low" e(d)
values
Turbo
code
Uncoded BPSK
$\frac{E_b}{N_0}$ (dB)
Error floor due to
"low" d_{min} value

Turbo codes

It is even possible to design the interleaver so as to obtain a code with a very high d_{min}.

To do it, we have to make sure that the parameter (L + L') is <u>always</u> a high number. In other words, if L is small, then L' should be large and, if L is large, then L' can be small → We must use a "structured" interleaver rather than a random one.

Carefully designing the interleaver can lead to very high values of d_{min}, and thus guarantee excellent error performance at high SNR (almost no error floor).

Examples of turbo codes

Ex 1: CCSDS (Consultative Committee for Space Data Systems) 16-state turbo code with k = 3,568.

- $R_c = \frac{1}{2}$: $d_{min} = 20$, $w_{d_{min}} = 1$ → $G \approx 10.00$ dB
- $R_c = \frac{1}{3}$: $d_{min} = 40$, $w_{d_{min}} = 9$ → $G \approx 11.25$ dB
- $R_c = \frac{1}{4}$: $d_{min} = 56$, $w_{d_{min}} = 6$ → $G \approx 11.46$ dB
- $R_c = \frac{1}{6}$: $d_{min} = 93$, $w_{d_{min}} = 6$ → $G \approx 11.90$ dB

Ex 2: UMTS rate-1/3, 8-state turbo code with k ranging from 40 to 5,114. For k = 320, $d_{min} = 24$, $w_{d_{min}} = 4$ → $G \approx 9.03$ dB.