Chapter 4

"Although the rivers and mountains of the world have not changed, their ancient and modern names are different."

Wen Zhuang, Ming dynasty

Channel Capacity & Turbo Coding

This chapter presents some basic communication principles related to channel capacity and forward error correction (FEC) codes. The Shannon's channel capacity theorem and its practical implications are also discussed, as well as the ways of getting closer to this limit. It is shown that in order to achieve better performance in a communication system, e.g. to get closer to the Shannon's limit, one can use the new state-of-the-art FEC codes. One of the best error correcting codes available today are the Turbo codes. A short introduction to Turbo codes and their characteristics and performance is also provided in this chapter.

The watermarking is seen by the information theory perspective and therefore by applying this theory and by using Turbo coding, the performance of the watermarking system is greatly improved. This conclusion can be easily drawn by analysing the results presented in Chapter 5 and Chapter 6.

4.1 The Channel's Capacity

Broadband providers are naturally interested in increasing transmission distance and data rates, on the one hand to cut down on the amount of physical plant that must be installed, and on the other to increase throughput. To accomplish this, the broadband provider may be tempted to extend the antenna length or increase the transmission power, but these are costly and oftentimes unacceptable alternatives.

An alternative that is becoming more attractive for providing increased performance is the use of powerful FEC. Embedding an FEC codec that implements state-of-the-art coding technology in the transceiver can radically increase the transmitted-data rate or transmission distance, or alternatively decrease the required antenna size and power.

FEC is the addition of redundancy (e.g., parity-check symbols) to a transmitted message, allowing the receiver to decode the received message, check symbols, and correct some limited number of errors in the received-data stream. The ability of FEC to increase the signal-to-noise capability of a communications channel depends on the code used and the channel characteristics.

All channels have a theoretical limit for information rate content at a constant signal-to-noise ratio (SNR) known as the Shannon capacity. The Shannon capacity limit defines the maximum information content for any particular channel. Communications systems that do not use FEC operate far from this limit, often 10 dB or more. Examples of these systems include voice applications and other communications systems where an occasional bit error can be tolerated. To achieve the accuracy and data rates required most of the time (wireless Internet access for example), the system without FEC would require 10 dB greater SNR than a "perfect" system operating at the Shannon capacity.

The use of traditional FEC codes such as Reed-Solomon (RS) coding substantially improves the efficiency of the communications channel allowing operation much closer to the Shannon capacity. For a typical channel, the addition of RS coding allows the system to operate within approximately 4 dB of the Shannon capacity (depending on channel characteristics). The resulting benefit translates into higher data rates, lower bit-error rates (BER), greater transmission distance and greater immunity to interference effects. However, this still leaves considerable room for improvement. After all, to make up for the 4-dB distance from optimum, the system developer must spend valuable resources in terms of transmission power, antenna size and bandwidth.

As an example, a more powerful code that provides a 3-dB coding gain over the RS coding can mean a reduction in antenna diameter by 30 percent, a decrease in transmission power by a factor of two, a transmission distance increased by 40 percent, or increased data throughput by a factor of two. Recent breakthroughs in error-correction coding have led to new FEC codes that can provide this 3-dB performance gain over RS coding. The Turbo codes are one possible example. In some special circumstances, it is possible to approach the Shannon's capacity limit by 0.27 dB using Hamming codes in a turbo decoding scheme [Nickl et al, 1997].

4.1.1 The Noisy Channel Coding Theorem

In 1948, Shannon derived the following formula for the capacity of an additive white Gaussian noise channel (AWGN)

$$C = W \log_2\left(1+\frac{S}{N}\right) \tag{4.1}$$

where the capacity is expressed in bits/sec, W represents the bandwidth of the channel, S is the average signal power and N is the total average noise power of the channel.

Shannon established the noisy channel coding theorem:

1. For information rate $R_{info} < C$ there exists a coding system with arbitrarily low block and bit error rates as we let the code length $n \to \infty$.

2. For information rate $R_{info} > C$ the bit and block error rates are strictly bounded away from zero for any coding system.

The noisy channel coding theorem therefore establishes rigid limits on the maximal supportable transmission rate of an AWGN channel in terms of power and bandwidth.

To characterise how efficiently a system uses its allotted bandwidth, one can define the *bandwidth efficiency* as

$$\eta = \frac{C}{W} \tag{4.2}$$

The Shannon limit can be calculated as

$$\eta_{\max} = \log_2\left(1+\frac{S}{N}\right) \tag{4.3}$$

Taking into account that

$$S = \frac{kE_b}{T} = RE_b \tag{4.4}$$

where E_b represents the energy per bit, k is the number of bits transmitted per symbol, T is the duration of a symbol and R is the transmission rate (code rate) of the system. Now the Shannon limit can be obtained in terms of the bit energy and noise power spectral density

$$\eta_{\max} = \log_2\left(1+\frac{RE_b}{WN_0}\right) \tag{4.5}$$

where $N = WN_0$ represents the total noise power and N_0 is the one-sided noise power spectral density.

The equation (4.5) can be resolved in order to obtain the minimum bit energy required for reliable transmission, e.g. the *Shannon bound*

$$\frac{E_b}{N_0} \geq \frac{2^{\eta_{\max}} - 1}{\eta_{\max}} \tag{4.6}$$

From equation (4.6) it is possible to establish the fundamental limit for reliable communication. This can be obtained by considering an infinite amount of bandwidth, i.e. $\eta_{\max} \to 0$

$$\frac{E_b}{N_0} \geq \lim_{\eta_{\max} \to 0} \frac{2^{\eta_{\max}} - 1}{\eta_{\max}} = \ln(2) = -1.59dB \tag{4.7}$$

This represents the absolute minimum signal energy to noise power spectral density ratio required to reliably transmit one bit of information, even for unlimited bandwidth or bit rate tending to zero.

The dependence on the arbitrary definition of the bandwidth W is usually not satisfactory. The answer is to normalise these formulas per signal dimension [Wozencraft et al, 1965]. This is useful when the question of waveforms and pulse shaping is not a central issue, since it allows one to eliminate these considerations by treating signal dimensions. In this case Shannon's capacity and the corresponding bound are

$$\begin{aligned} C_d &= \frac{1}{2}\log_2\left(1 + 2\frac{R_d E_b}{N_0}\right) \\ \frac{E_b}{N_0} &\geq \frac{2^{2C_d} - 1}{2C_d} \end{aligned} \tag{4.8}$$

The dependence of Shannon's capacity limit of the code rate is illustrated in **Table 4-1**, for an AWGN channel with QPSK (Quadrature Phase Shift Keying) modulation [Dolinar et al, 1998].

Code rate, R	Capacity limit, $E_b / N_0 \left[dB\right]$
1/2	0
1/3	-0.55
1/4	-0.82
1/6	-1.08
0	-1.59

Table 4-1 Shannon limit for different code rates

4.1.2 Hardware and Software Decoding

A typical communication system can be represented as in [Barbulescu et al, 1996]. Regardless of its source, the information to be transmitted must be translated into a set of signals optimised for the channel over which we want to send it. As **Figure 4-1** shows, the first step is to use a source encoder block for eliminating the redundant part of the signal in order to maximise the information transmission rate. To ensure the secrecy of the transmitted information one could use an encryption scheme. The most important part of the system in the case analysed here, is to protect the signal against the perturbations introduced by the communication channel, which could lead to errors in the transmitted message at the receiving end. This protection is achieved by FEC, using error correction codes that are able to correct the errors at the receiving end. Finally the modulator block generates a signal suitable for the transmission channel.

The importance of using powerful error correction and the economical and practical benefits of such codes was already underlined in the introduction. From coding theory it is known that either by reducing the data rate or increasing the codeword length or the encoding memory, greater protection or coding gain, can be achieved. Unfortunately at the same time the complexity of typical decoding algorithm such as the maximum likelihood decoding algorithms increases exponentially with the encoder memory and the algorithms become difficult to implement. Therefore the increased error correction capability of long codes requires a very high computational effort at the decoder.

In simple systems, the demodulator block from **Figure 4-1** makes a hard decision of the received symbol and passes it to the error control decoder block. In other words, the demodulator decides which of two logical values 0 or 1 was transmitted. No information is

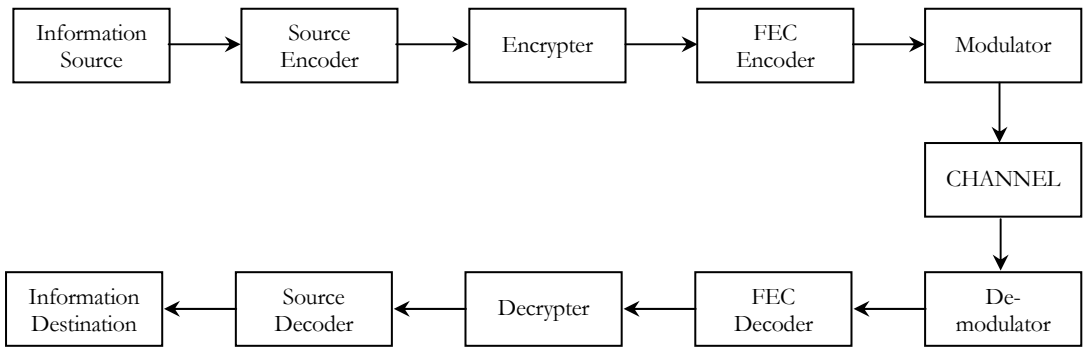

Figure 4-1 The block diagram of a communication system

passed to the FEC decoder about how reliable the hard decision was. Better results can be obtained by using soft input decoding algorithms, e.g. when the quantised analogue received signal is passed directly to the decoder. The same consideration holds for the outputs of the constituent decoders of concatenated codes. By using soft-input-soft-output (SISO) decoders, this information can be passed from one decoder to the next in an iterative fashion. Soft output decision algorithms provide as an output a real number which is a measure of the probability of error in decoding a particular bit. This can be also interpreted as a measure of the reliability of the decoder's hard decision.

It can be shown that the channel capacity of a discrete-input real-output (soft output) memoryless channel (C_{soft}) is greater than that for a discrete-input discrete-output (hard output) memoryless channel (C_{hard}). For a binary symmetric channel with an AWGN distribution and mean value zero this can be proven as follows.

Soft decoding

If the input alphabet is $X=\{x_0, x_1, x_2, ..., x_{q-1}\}$ and the output alphabet is $Y=\{-\infty, \infty\}$ then we can define the channel capacity for the soft decoding case as the mutual information between the channel's input and output maximised over all possible channel input distributions $P(x_j)$

$$\begin{aligned} C &= \max_{P(x_j)} I(X;Y) \\ &= \max_{P(x_j)} \left[H(X) - H(X|Y) \right] \\ &= \max_{P(x_j)} \left[H(Y) - H(Y|X) \right] \\ &= \max_{P(x_j)} \sum_{j=0}^{q-1} \int_{-\infty}^{\infty} P(x_j)\, p(y|x_j) \log \frac{p(y|x_j)}{p(y)} dy \end{aligned} \tag{4.9}$$

Considering the input alphabet restricted to $X=\{-1,+1\}$ and a binary symmetric channel, where $P(-1)=P(+1)=0.5$ then equation (4.9) becomes

$$C_{soft} = \frac{1}{2}\int_{-\infty}^{\infty} p(y|+1) \log_2 \frac{p(y|+1)}{p(y)} dy + \frac{1}{2}\int_{-\infty}^{\infty} p(y|-1) \log_2 \frac{p(y|-1)}{p(y)} dy \tag{4.10}$$

Taking into account that $p(y) = 0.5\,p(y|+1) + 0.5\,p(y|-1)$ and that for symmetry $p(-y|+1) = p(y|-1)$ then equation (4.10) can be further simplified

$$C_{soft} = \int_{-\infty}^{\infty} p(y|+1)\log_2 \frac{p(y|+1)}{p(y)} dy \tag{4.11}$$

For an AWGN channel with zero mean and variance σ^2 the probability density function $p(y|x=m), m=+/-1$ can be defined as follows

$$p(y|x=m) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y-m)^2}{2\sigma^2}\right) \tag{4.12}$$

and the variance σ^2 is

$$\sigma^2 = \left(2R\frac{Eb}{No}\right) \tag{4.13}$$

where R represents the code rate and $\frac{Eb}{No}$ is expressed in dB.

Hard decoding

For a hard decoder the output alphabet is finite too so in this case the input and respectively the output alphabets can be defined as $X = \{x_0, x_1, x_2, ..., x_{q-1}\}$ and $Y = \{y_0, y_1, y_2, ..., y_{r-1}\}$. The channel capacity for this case is

$$\begin{aligned} C &= \max_{P(x_j)} I(X;Y) \\ &= \max_{P(x_j)} \left[H(Y) - H(Y|X)\right] \\ &= \max_{P(x_j)} \sum_{j=0}^{q-1}\sum_{i=0}^{r-1} P(x_j)P(y_i|x_j)\log\frac{P(y_i|x_j)}{P(y_i)} dy \end{aligned} \tag{4.14}$$

Restricting the input and output alphabets to $X = \{-1, +1\}$ and respectively $Y = \{-1, +1\}$ and taking into account that $P(-1) = P(+1) = 0.5$, for a binary symmetric channel we can define

$$\begin{aligned} P(-1|+1) &= \int_{-\infty}^{0} p(y|+1)\,dy = \hat{P} \\ P(+1|-1) &= \int_{0}^{\infty} p(y|-1)\,dy = \hat{P} \end{aligned} \tag{4.15}$$

Then we can obtain the channel's capacity for the hard decoding case as

$$C_{hard} = 1 + \hat{P}\log_2 \hat{P} + (1-\hat{P})\log_2(1-\hat{P}) \tag{4.16}$$

For an AWGN channel, $\hat{P}$ can be defined as

$$\hat{P} = Q\left(\sqrt{\frac{2Eb}{No}}\right) \tag{4.17}$$

where the Q function is defined as

$$Q(x) = \int_{x}^{\infty} \frac{1}{\sqrt{2\pi}}\left(-\frac{t^2}{2}\right)dt \tag{4.18}$$

Conclusion

The comparison between the soft and hard decoding is illustrated for the binary symmetric channel case in **Figure 4-2**. It can be seen that at low signal to noise ratios C_{soft} is greater than C_{hard} by approximately 2dB. This shows very well the advantage of using soft decision decoding rather then the classical hard decision approach.

4.2 Turbo Codes

Starting with early 1990's concepts like iterative decoding, soft output decision algorithms, special encoding techniques and information transfer techniques were combined in order to create more powerful error correction codes. The combination of these concepts led to appearance of a new class of powerful error correcting codes: the Turbo codes, which made possible communications very close to Shannon's limit. For example, the first Turbo code

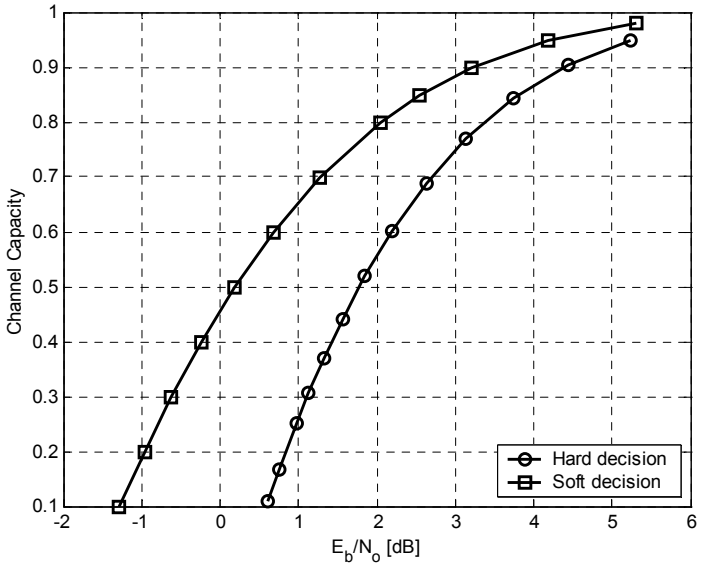

Figure 4-2 Channel capacity for soft and hard decision decoding for a binary symmetric channel

proposed in the literature achieved a bit error rate lower than 10^{-5} within 0.7dB of Shannon's limit.

The Turbo codes were introduced in [Berrou et al, 1993]. They represent a particular class of parallel concatenation of two recursive systematic convolutional codes. In other words, strictly speaking a Turbo code is a 2PCCC (Parallel Concatenated Convolutional Code). Today the term has a more general connotation.

4.2.1 The Structure of a Turbo Code (2PCCC)

The encoder

The block scheme of the encoder is presented in **Figure 4-3**. Since we are dealing with a convolutional code the input sequence is organised in blocks of length N. The first block of data is encoded by the ENC_1 block which is a half rate recursive systematic encoder. The same block of data is interleaved by the interleaver block INT and then encoded by the second encoder ENC_2. Like the first encoder, ENC_2 is a half rate recursive systematic encoder.

The role of the interleaver is to rearrange the order of the information bits from the input. In this way the interleaver increases the minimum distance of the Turbo code and therefore its error correction capability. The design of the interleaver is a key factor which determines the good performance of a Turbo code.

The result is a rate 1/3 turbo code, with the output given by the triplet (v1i, v2i, v3i). Since the code is systematic ui = v1i is the input data at time i and v2i and v3i are the two parity bits at time i. Sometimes the parity bits can be "punctured" using a multiplexing switch in order to obtain higher coding rates.

The decoder

By encoding the same information twice but in different order, the Turbo codes have the advantage of exchanging information between the two constituent decoders. The more "scrambled" the information sequence is for the second encoder the more "uncorrelated" (independent) the information exchange is. This is in fact one of the keys that allows continuous improvement in correction capability when the decoding process is iterated.

As already stated, the Turbo codes use soft output decision algorithms. There are two important categories of soft output decision algorithms. The first category includes the maximum likelihood decoding algorithms which minimise the probability of bit error, such as the maximum a posteriori (MAP) algorithm. The second category includes the maximum

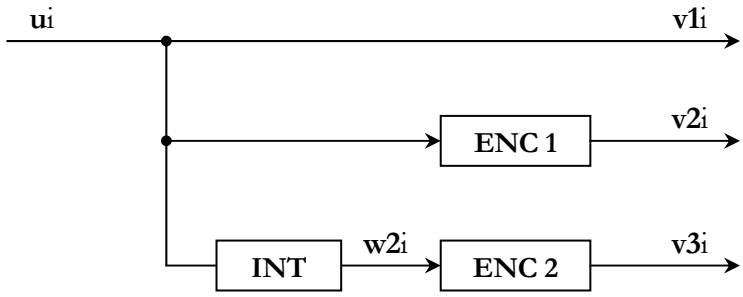

Figure 4-3 The Turbo encoder

likelihood decoding algorithms which minimise the probability of word or sequence error, such as the Soft Output Viterbi Algorithm (SOVA). Although the SOVA algorithm has a soft output, it is sub optimal. The decoder described in this section uses the MAP algorithm and it is presented in **Figure 4-4**.

The decoding principle is briefly described below. First, the MAP2 decoder which corresponds to the encoder ENC2 decodes the information present at its input and initialises the probability Pw20i(r2) with the value 0.5. The decoder MAP2 also calculates the new extrinsic probability Pow20i(r3) using the Pw20i(r2) probability and the Gaussian probabilities of r3i and respectively the interleaved version of r1i, Px20i(r1).

The extrinsic information computed at this step Pow20i(r3), gives a more precise information about the bit w2i. Since w2i is the common information between the two encoders ENC1 and ENC2, the extrinsic information refers to this particular bit. The extrinsic information is then deinterleaved and passed to the first decoder MAP1.

This decoder will start decoding the information from its input, taking into account the extrinsic information supplied by the MAP2 decoder. Therefore when MAP1 decoder

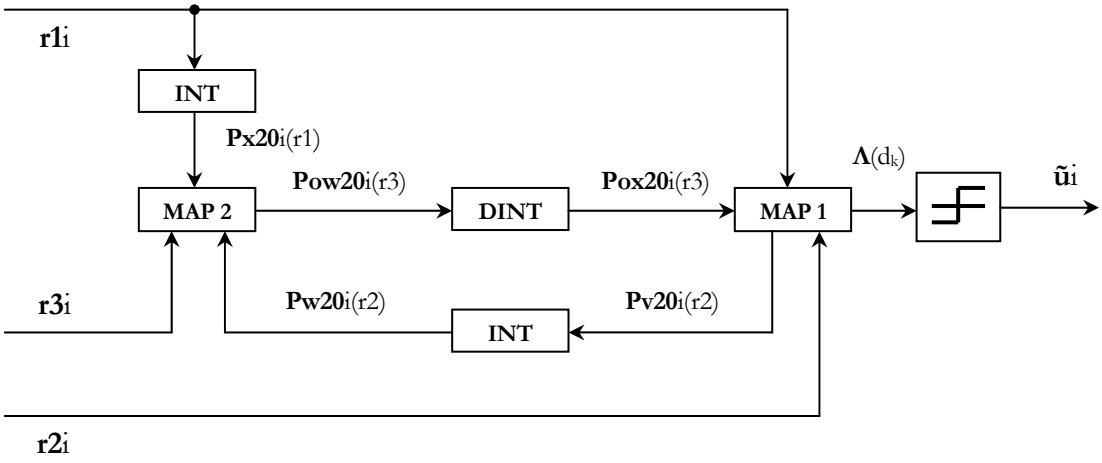

Figure 4-4 The iterative Turbo decoder

begins its operation it already has a better knowledge about the w2i bit since it uses the extrinsic information supplied by MAP2 rather than the start value equal with 0.5. The MAP1 decoder corresponds to the encoder ENC1.

The MAP1 decoder computes the new extrinsic information Pv20i(r2) which is then interleaved and passed back to the MAP2 decoder for a new iteration. The MAP1 decoder also calculates the *a posteriori* probability $\Lambda(d_k)$ by using the extrinsic information Pox20i(r3) and the Gaussian probabilities corresponding to r2i and respectively r3i. When the desired number of iteration was completed, this information is passed to the hardware decision block who gives at its output the error corrected sequence ũi corresponding to the originally encoded data sequence ui.

4.2.2 Several Particularities of the Turbo Codes

It is well known that the performance of the convolutional codes improves with increasing constraint length. This is not the case for Turbo codes. In fact, the best constituent codes of a Turbo code have a very small constraint length.

The performance of convolutional codes does not improve significantly with the decreasing of the code rate; in fact the difference between rate 1/3 and rate 1/128 is of the order of a few tenths of a dB in the case of convolutional codes [Barbulescu et al, 1996]. The Turbo codes instead achieve a very significant coding gain for lower coding rates. For practical code rates between 1/2 and 1/6, **Figure 4-5** illustrates the Eb/No required to achieve a BER of 10^{-6} as a function of coding rate, for both convolutional codes and Turbo codes [Barbulescu et al, 1996]. Analysing the difference in Eb/No between the 1/2 and 1/6 cases for both

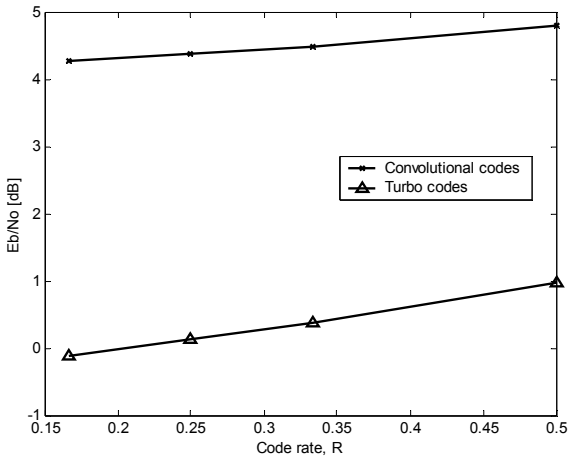

Figure 4-5 Eb/No required to achieve a BER=10^{-6} for convolutional codes and Turbo codes

convolutional codes and Turbo codes it can be seen that this difference is twice as much for the Turbo codes. So it can be concluded that lower rate Turbo codes provide significantly more coding gain (twice as much) than lower rate convolutional codes.

The PCCC are the best choice when the $BER \geq 10^{-6}$, but however for much lower bit rate requirements other codes could be better (SCCC for example). This is because for the PCCC usually a change in the slope of the BER curve appears for $BER < 10^{-7}$ depending of the interleaver size and design. At low Eb/No the PCCC performs better than SCCC but increasing the Eb/No the SCCC schemes outperform PCCC schemes. The crossover point depends again of the interleaver size and design.

Referring strictly to the PCCC it was proven [Divsalar et al, 1995] that the interleaver gain term depends on the number of codes in the concatenated system, and the probability of error is

$$BER \sim \frac{1}{N^{m-1}} \tag{4.19}$$

where N is the interleaver length and m is the number of component codes.

4.3 Turbo Codes in Watermarking

If we regard the watermark channel as a communications system with input X (the watermark data) and output Y, the channel capacity is formally defined as the maximum mutual information (section 4.1.2),

$$C_{chan} = \max_{p(x)} \; I(X;Y) = \max_{p(x)} \left[H(X) - H(X \mid Y) \right] = \max_{p(x)} \left[H(Y) - H(Y \mid X) \right] \tag{4.20}$$

where the maximum is taken over all possible distributions $p(x)$. Term $H(X \mid Y)$ represents information loss due to channel noise, which will be a combination of the host video and signal processing (compression or other forms of attack). If the loss is modelled as the addition of an independent Gaussian noise source, $Z \sim N(0, \sigma_z^2)$, i.e. $Y_i = X_i + Z_i$, where Z is a continuous random variable, then equation (4.20) reduces to

$$C_{chan} = \frac{1}{2} \; \log_2 \left(1 + \frac{\sigma_x^2}{\sigma_z^2} \right) \left[\text{bits/symbol} \right] \tag{4.21}$$

providing $X \sim N(0, \sigma_x^2)$.

In this thesis the HVS models are employed in order to maximize the signal power. The watermarking channel is illustrated in **Figure 4-6** and can be modelled as a gain factor cascaded with a Gaussian noise source $Z \sim N(0, \sigma_z^2)$ (the gain and variance depending upon the host media, MPEG compression, geometric attack, etc.).

For example, one could estimate a basic operational capacity as follows. Suppose that all N_p pixels ($N_p = 720 \times 576$) in the frame are transformed via the DCT or DWT, and that the channel noise is simply that of the host video. Assuming C_i is i.i.d. for simplicity, the noise power per video frame is $N_p \overline{C^2}$, where $\overline{C^2}$ is the mean coefficient power for a particular video sequence. If only one data bit is embedded per video frame (corresponding to a spread spectrum chip rate $c_r = N_p$), and there is no FEC, the *SNR* is

$$SNR = N_p \frac{\left(\alpha \quad \overline{HVS}\right)^2}{\overline{C^2}} = N_p \overline{SNR} \tag{4.22}$$

where $\overline{SNR}$ is a measured mean *SNR* for the video sequence and $\overline{HVS}$ is the mean embedding strength given by the HVS model for the sequence.

If N_b data bits are embedded into a frame (using all the coefficients) the signal to noise ratio per uncoded data bit reduces to $SNR_u = SNR / N_b$, and the data rate or capacity for an uncoded system of frame rate F_r is

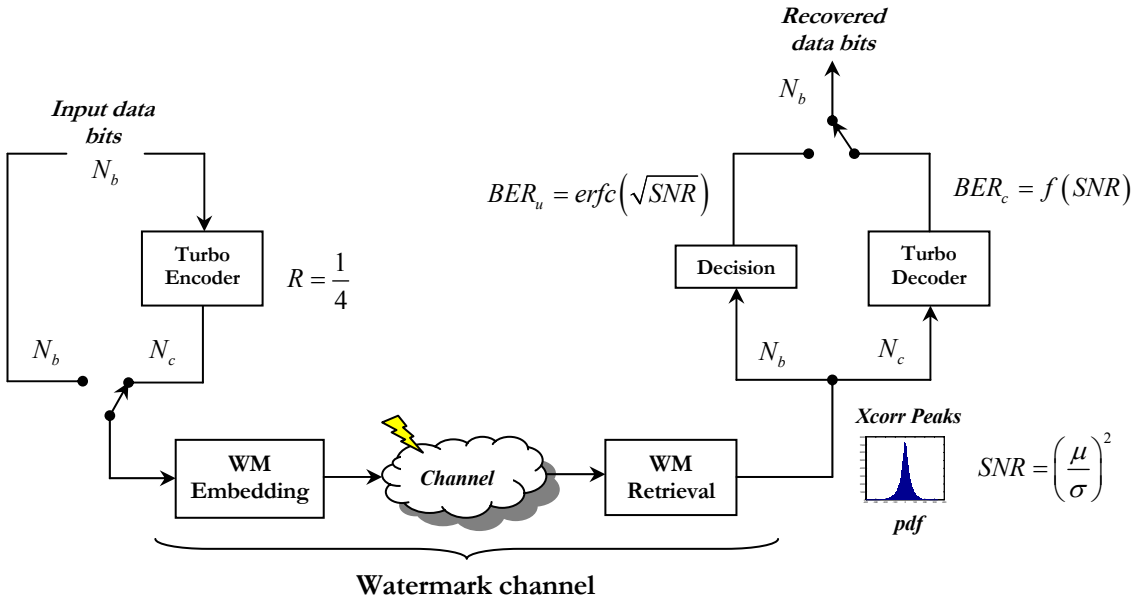

Figure 4-6 The watermarking channel

$$D_r = N_b F_r = \frac{N_p \quad \overline{SNR} \quad F_r}{SNR_u} \tag{4.23}$$

For a coded system of rate *R*, with $N_c = N_b / R$ bits embedded over N_f video frames, we have

$$D_r = \left(\frac{N_c \cdot R}{N_f} \right) \quad F_r \tag{4.24}$$

The operational capacity can be defined as the maximum value of D_r for which the *BER* does not exceed a tolerable level (typically 10^{-8}).

The *SNR* can be defined as in **Figure 4-6**. Since the cross-correlator performs a sequence of correlation sums, it follows from the Central Limit theorem that the cross-correlation peaks have a normal distribution [Ambroze et al, 2001]. This is very convenient for the iterative Turbo decoder, which generally assumes a Gaussian input. Thus, for any particular system, the distribution mean μ, and variance σ^2 define a *SNR* of the channel

$$SNR = \left(\frac{\mu}{\sigma} \right)^2 \tag{4.25}$$

and the corresponding *BER* for an uncoded system is simply

$$BER_u = Q\left[\mu / \sigma \right] = Q\left[\sqrt{SNR_u} \right] \tag{4.26}$$

For a coded system, μ and σ define a signal to noise ratio SNR_c at the decoder input,

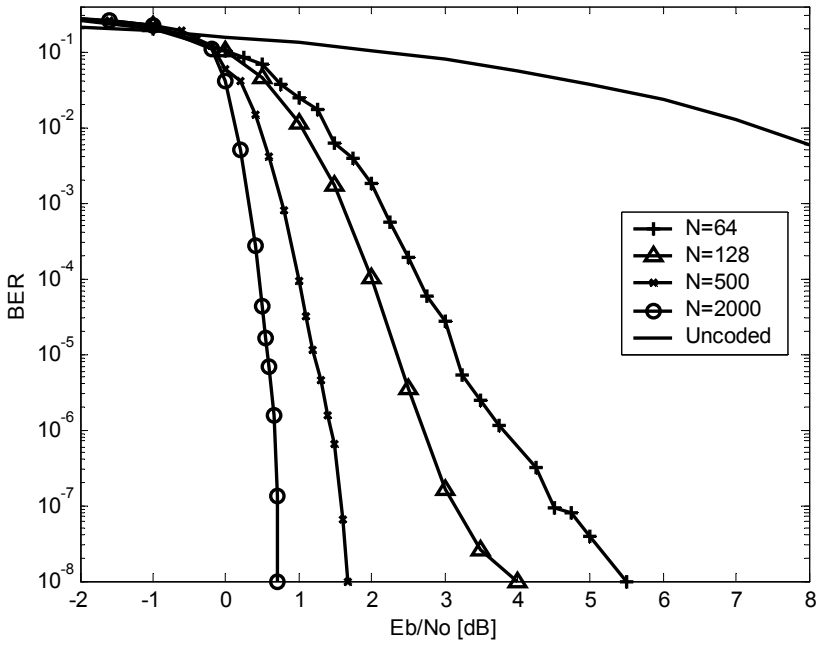

Figure 4-7 The performance of the 3PCCC Turbo code for different block lengths

and the decoded bit error rate is $BER_c = f(SNR_c)$ where *f* is a known function for a particular iterative decoder.

The FEC code used in this thesis is a rate 1/4 multiple parallel concatenated convolutional code (3PCCC) [Ambroze, 2000] rather than the basic turbo code (2PCCC) in order to improve performance [Ambroze et al, 2001]. The performance of this code is presented in **Figure 4-7**. The structure of the 3PCCC Turbo encoder and decoder is presented in **Appendix 1**.

The use of FEC reduces the chip rate by a factor *R* due to the fact that now we have to embed $N_c = N_b/R$ coded bits instead of N_b. This increases the variance of the channel distribution, resulting in increased *BER*, and the FEC decoder must more than compensate for this increase in order to provide coding gain. As it will be shown in the next chapters, the Turbo code improves the performance of the watermarking system in a significant manner.

4.4 Conclusions

The watermarking channel is a very difficult channel characterised by high levels of noise (the video sequence itself represents the noise) and low power of the watermark signal (due to the visibility constraints). The situation is even worse when taking into account various attacks which increase even further the noise from the system. This translates to a relatively low SNR at the input of the FEC decoder.

This fully justifies the use of soft decision based FEC codes, particularly the use of Turbo codes which are known for their very good performance under difficult conditions (very low SNR). Using other error correction codes like the BCH codes (used by some authors in their watermarking systems) which have a hard decision algorithm, automatically leads to a 2dB drop in performance when compared with Turbo codes, or other soft decision based algorithms. This is a very significant loss in a watermarking system, which cannot be afforded.

Digital watermarking seems to be yet another successful application area for Turbo codes, joining many other already consecrated application areas like deep space applications, satellite communications, speech and image transmission and many others.