A tutorial on iterative probabilistic decoding and channel estimation: Graph representation, information flow and probabilistic algorithms in coding and communications.*

Joseph J. Boutros

Communications and Electronics Department
ENST, 46 Rue Barrault, 75013 PARIS, France
Email: boutros@enst.fr, Web Page: http://www.comelec.enst.fr/∼boutros

Tutorial version 2.1
19 January 2005

Abstract

Signal detection, channel decoding and parameter estimation are described in a unified manner. First, we introduce the graph representation and the vocabulary used in probabilistic processing of information. Then, we study six important situations in communication and coding theory: Probabilistic detection of quadrature amplitude modulations (QAM) on additive white gaussian noise (AWGN) channel, soft-input soft-output (SISO) decoding of binary convolutional and binary block codes, iterative APP equalization on inter-symbol interference (ISI) channels, iterative APP detection on multiple antenna (MIMO) channels, probabilistic multiuser detection on code division multiple access (CDMA) channels, and APP decoding of binary low-density parity check codes (LDPC), generalized low density (GLD/Tanner) codes and binary parallel Turbo codes. Secondly, we recall the maximum likelihood pilot based channel estimation, and how channel state information (CSI) is essential in probabilistic signal detection and channel decoding. Then, we describe the Expectation-Maximization algorithm as a recursive low complexity approach to ML estimation. We show how the EM can be integrated into the decoding of graph codes and graph systems. Two approaches for processing data and pilots in the EM algorithm are given. Finally, we discuss the relationships between the EM and other algorithms in communications, coding and information theory.

Keywords: A posteriori probability (APP) decoding, a posteriori probability detection, Forward-Backward algorithm, Expectation-Maximization (EM) algorithm, maximum likelihood (ML) estimation, LDPC codes, Turbo codes.

1 On the extrinsic information in iterative detectors/decoders

We show in this document that the strict definition of extrinsic information (Extr) is unique and its expression is always given by a sum-product formula. This is true for all iterative soft-input soft-output (SISO) receivers and decoders, including multiuser detection, multiple antenna detection, equalization of inter-symbol interference channels, decoding of turbo codes, low density parity-check codes and all kind of codes on graphs. On the other hand, the definition of the a posteriori probability (APP) is also unique, but its expression may show a slight difference depending on the existence or the absence of direct channel observation.

*This memorandum can be downloaded in pdf and ps formats from *The Channel Coding Page* at the following URL, http://www.comelec.enst.fr/∼boutros/coding

In the sequel, the symbol $\propto$ means *proportional to* and the symbol $\sim$ means *equivalent to* or *isomorphic to.* We briefly study 6 different situations where we give the a posteriori probability expression and its decomposition into a product of independent quantities in order to extract the extrinsic probability. All these situations can be described by a unified framework: an encoder generating sequences transmitted on a discrete memoryless channel (DMC). These sequences belong to a code C and are called *codewords* in both finite and infinite length cases. The soft-output decoder may be asked to produce soft information on both the encoder binary input and the encoder Euclidean output. The generalization to non-binary inputs and non-Euclidean outputs is straightforward. For simplicity, our DMC is an additive white gaussian noise (AWGN) channel. Soft-input soft-output receivers, including both detectors and decoders, are also called APP receivers or probabilistic receivers.

At time instant i, let x_i denote a M-QAM modulation symbol, where $M = 2^m$. Let b_j denote a binary element belonging to the label of x_i, and let y_i denote the channel output. The a posteriori probability of b_j is defined as $APP(b_j) = P(b_j|\mathbf{y}, C)$, where $\mathbf{y}$ is the vector of all channel outputs, $i = 1 \dots N$. An identical definition is made for $APP(x_i)$. Notice that an equivalent definition is $APP(b_j) = p(b_j, \mathbf{y}|C) \propto P(b_j|\mathbf{y}, C)$. We will restrict our study to the first convention and will use conditional distributions instead of joint distributions. The proof of all given formulas is trivial and needs basic knowledge in probability theory and communication theory. The non-initiated reader should consult basic books such as *Probability, Random Variables and Stochastic Processes* by Athanasios Papoulis, *Digital Communications*, by John Proakis and *Elements of Information Theory* by Cover and Thomas.

2 Introducing a graph representation and some vocabulary

Let us consider a universal variable v represented by a variable node v as illustrated in Fig. 1. The variable may represent a binary element b_j, a QAM symbol x_i, an element of a finite field $GF(Q)$, or any other kind of symbols belonging to a finite set used in Communication and Coding theory. We assume that v is connected to L sub-graphs denoted by *Graph* ℓ, where $\ell = 1 \dots L$. The sub-graphs are not connected together, i.e., the total graph has no cycles.

The sub-graph *Graph* ℓ represents a constraint given by an error-correcting code, the channel memory, or any type of constraints. The graph in Fig. 1 suggests that v must satisfy L independent constraints. A direct observation $obs(v)$ may also be available as shown in the flow from the channel sub-graph to the node v. For example, if $v = x_i$ and the DMC channel is defined by $p(y_i|x_i)$, then the probability $obs(v) = obs(x_i) \propto p(y_i|x_i)$ is a direct channel observation. The exact expression is obtained by normalizing the channel likelihoods, $obs(x_i) = p(y_i|x_i)/\sum_j p(y_j|x_j) \in [0 \dots 1]$.
Given the L constraints and the total channel observation, $\mathbf{y} = (y_1, y_2, \dots, y_N)$ defined in section 1, *Graph* ℓ generates a probabilistic information $Extr_\ell(v)$ illustrated by the flow outgoing the sub-graph box and incoming the variable node. This information is independent from any $Extr_k(v)$, for all $k \neq \ell$. Furthermore, $Extr_\ell(v)$ is independent from any possible direct observation on v. Hence, $Extr_\ell(v)$ is called **extrinsic information** on v generated by *Graph* ℓ. Finally, the a posteriori probability of v is given by the product of all incoming

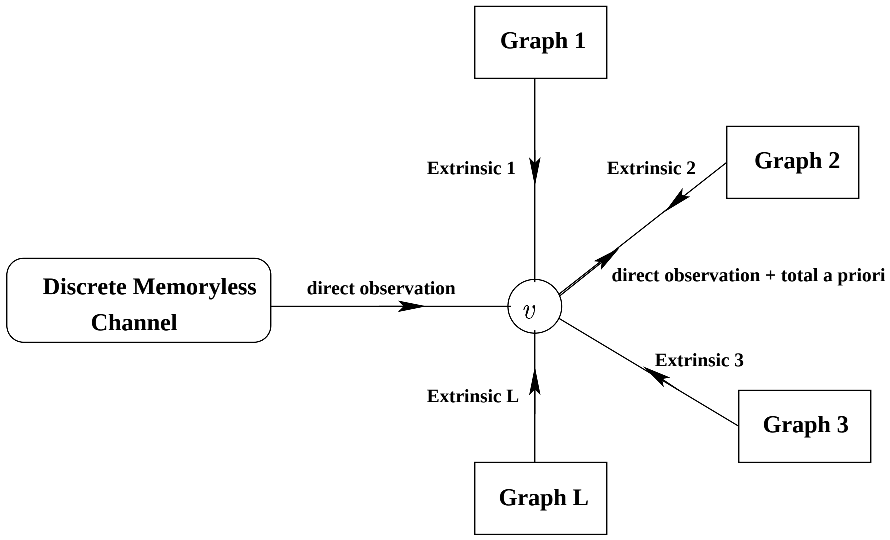


Figure 1: Graph representation of information flow, local neighborhood of v.

probabilistic informations[1]

$$APP(v) \propto obs(v) \times \prod_{\ell=1}^{L} Extr_\ell(v) \tag{1}$$

As usual, the proportionality factor is determined by writing that $\sum_v APP(v) = 1$. Now, let us focus on the edge linking the variable node and any of the L sub-graphs, e.g. *Graph 2*. The information flow from v to *Graph 2* obtained by summing all extrinsics generated by the other sub-graphs is called **a priori information**. Thus, looking to the flow incoming *Graph 2*, we can define the a priori information $\pi_\ell(v) = Extr_\ell(v)$, $\ell \neq 2$. Sub-graphs $1, 3, 4, \ldots L$ act like $L-1$ genies delivering a priori information $\pi(v) \propto \prod_{\ell \neq 2} \pi_\ell(v)$ to *Graph 2*. We can rewrite the a posteriori probability as the product of the two opposite information flows on the edge linking v and *Graph 2*,

$$APP(v) \propto obs(v) \times \pi(v) \times Extr_2(v) \tag{2}$$

The above equation is universal and the reader should recall it literally

APP = Observation x A Priori x Extrinsic

The code C defined in section 1 for the unified framework is given by the constraints imposed by the L subgraphs on all variable nodes. If C_ℓ denotes a code associated to the constraints in *Graph* ℓ, then

$$C = \bigcap_{\ell=1}^{L} C_\ell \tag{3}$$

[1]In practice, for finite length codes, this does not yield the exact a posteriori probability value due to the presence of cycles in the graph of Fig. 1.

The sub-graphs contain other variable nodes[2] than v. These variable nodes, not necessarily of the same kind as v, may receive their own direct observation and are not illustrated in Fig. 1. Therefore, as shown in the examples of section 3, the extrinsic probabilities $\{Extr_\ell(v)\}$ depend on the observation vector $\mathbf{y}$, or a part of this vector, even if $obs(v)$ is not available.

How decoding iterations are defined and why ?

Finally, before terminating this section that describes elementary notions on graph representation for probabilistic receivers, the author would like to elucidate the definition of *iteration*. Therefore, we draw around the variable node v more details about the constraints and other variables in the local neighborhood. This is illustrated in Fig. 2. Edges and nodes corresponding to direct channel observations are not displayed. Variables are drawn as small circles and constraints are drawn as squares. The information flow is assumed to be directed from the graph boundaries to its center. The graph is not necessarily regular. It is equivalent to a tree where the information flow is directed upward, from its leaves toward its root. In Fig 2, the squares are organized in shells around v and deliver their extrinsic messages to variable nodes. Similarly, the variable nodes deliver their a priori and observation messages to the squares located on the inferior shell. When messages propagate from one shell to another, we say that the receiver performs one APP detection/decoding iteration. Obviously, $APP(v)$ that depends on all observations and all constraints cannot be determined after one iteration. If C has infinite length, then the theoretical number of iterations to be applied is infinite. In practice, C has a finite length, the graph is not a tree (cycles are present), and a limited number of iterations is required to reach a quasi steady-state in most conditions.

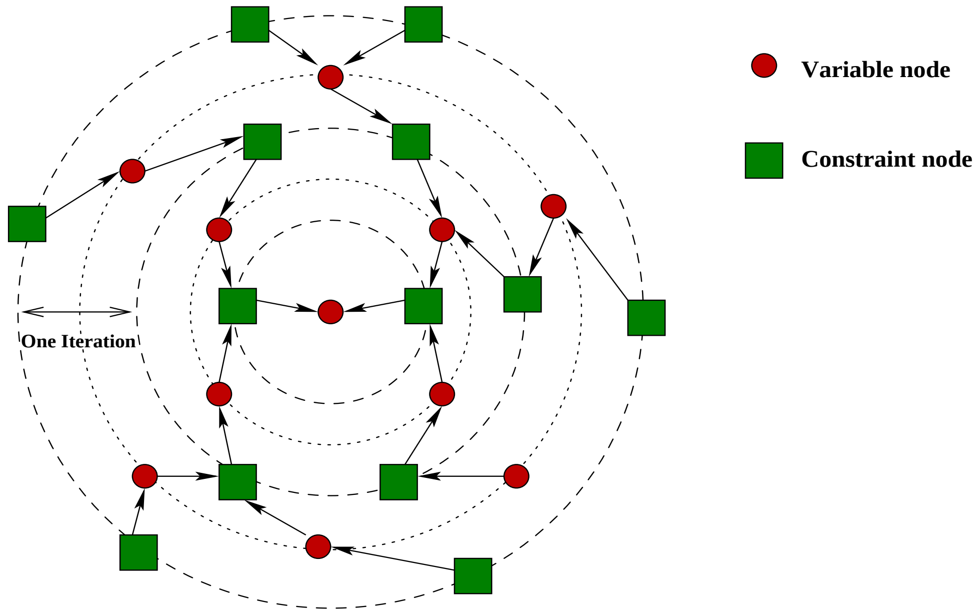


Figure 2: Representation of decoding iterations in the code graph. The information propagation graph is equivalent to a tree where the flow is directed from its leaves to-wards its root. Iterative processing of information is also known as *message passing* or *belief propagation*.

[2]Fig. 1 shows the local neighborhood of the variable node v, not the detailed graph which includes all variables and all constraints.

3 Six important situations in Communication/Coding theory

From section 2, expressions (1) & (2) and the related comments, any soft-input soft-output detector/decoder used as a building block for iterative processing in a receiver can be drawn as a box with two inputs and two outputs, see Fig. 3. The extrinsic output is determined by means of a sum-product formula as shown in the sequel. The a posteriori output is the product of the two inputs and the extrinsic output as shown previously.

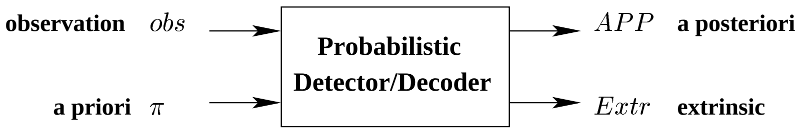


Figure 3: General model for a building block used in iterative decoding.

The general description given in sections 1 & 2 is now clarified via 6 different situations encountered in Communication and Coding theory. Other examples should be added in a future version of this document.

3.1 QAM probabilistic detector on AWGN channel

In this section, the observation length is $N = 1$, $x = x_1$ and $\mathbf{y} = y_1 = y$. Consider a linear quadrature amplitude modulation M-QAM where $M = 2^m$, $m \geq 2$ bits per symbol. The QAM constellation is usually defined by a finite translated subset of $2A\mathbb{Z}^2$, $A \in \mathbb{R}^+$. For square constellations, we have $x = \Re x + \sqrt{-1}\Im x$, and $\Re x, \Im x \in \{\pm A, \pm 3A, \ldots, \pm(\sqrt{M}-1)A\}$. The QAM symbol $x = x(\mathbf{b}) = x(b_1, \ldots b_m)$ is transmitted on a discrete memoryless channel with additive white gaussian noise, $y = x + z$, where $z \sim \mathbb{C}\mathcal{N}(0, 2\sigma^2)$. The system model is depicted in Fig. 4.

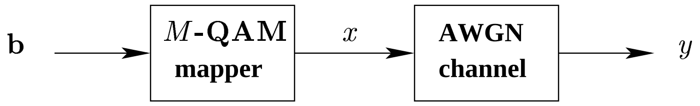


Figure 4: QAM modulation, $b \in GF(2)^m$, $x \in M\text{-QAM} \subset \mathbb{C}$, and $y \in \mathbb{C}$.

Given y, a QAM detector finds $APP(x)$ as follows: $APP(x) = P(x|y) \propto p(y|x) \times \pi(x)$. Then

$$APP(x) \propto obs(x) \times \pi(x) \quad where \quad obs(x) \propto p(y|x) \propto exp\left(-\frac{|y-x|^2}{2\sigma^2}\right) \tag{4}$$

If the a priori information is not available for QAM symbols, $\pi(x) = 1/M$, then

$$APP(x) = obs(x) = \frac{exp\left(-\frac{|y-x|^2}{2\sigma^2}\right)}{\sum_{x' \in QAM} exp\left(-\frac{|y-x'|^2}{2\sigma^2}\right)} \tag{5}$$

From (4), it is obvious that the QAM detector delivers no extrinsic information in the system described by Fig. 4, i.e., $Extr(x) = 1/M$. In all cases, the a posteriori probability of QAM complex symbols is represented by the literal expression

APP = Observation x A Priori

Even though, from a probability theory point of view, the reader should notice that *Observation*, *A Priori* and *Extrinsic* are probabilistic quantities of the same kind. Hence, one may not follow a strict jargon and regard the direct observation of x as an extrinsic or a priori information delivered by the detector. The author does not pretend that there is a unique strict jargon in this field, one may use any vocabulary if the notations and the vocabulary itself are perfectly clear for the readers.

Now, let the QAM detector determine $APP(b_j)$, $j = 1 \dots m$. Starting from the original definition of the a posteriori probability, we can write

$$APP(b_j) = P(b_j|y) = \sum_{\{b_k\},k\neq j} P(\mathbf{b}|y) \propto \sum_{x\in QAM|b_j} p(y|x)\pi(x) \tag{6}$$

In the above expression, the notation $\sum_{\{b_k\},k\neq j}$ means that the sum is made over the $2^{m-1} = M/2$ values of the binary vector $(b_1, \dots, b_{j-1}, b_{j+1}, \dots, b_m)$. Consequently, the notation $\sum_{x\in QAM|b_j}$ means that the sum is made over the $M/2$ QAM symbols associated to a label where the j^th binary element is equal to b_j. Before going further, the reader should notice that there is no direct observation for b_j because the QAM mapper is a non-systematic rate-1 encoder with a binary input in $GF(2^m)$ and a Euclidean output in $\mathbb{C}$. Since $obs(b_j)$ is not available, it is mathematically correct to write $obs(b_j) = 1/2$.

Two simple cases can be distinguished while interpreting (6):

1. Absence of a priori for x. Write $\pi(x) = 1/M$, and suppress $\pi(x)$ from (6). Then,

$$APP(b_j) = Extr(b_j) = \frac{\sum_{x\in QAM|b_j} exp\left(-\frac{|y-x|^2}{2\sigma^2}\right)}{\sum_{x'\in QAM} exp\left(-\frac{|y-x'|^2}{2\sigma^2}\right)} \tag{7}$$

2. A priori of x is present via independent a priori probabilities $\pi(b_k)$ for the m binary elements in the label of x, $k = 1 \dots m$. These probabilities $\pi(b_k)$ may be generated by the decoder of an error-correcting code inserted in the transmitter before the QAM mapper. In this second case, we write $\pi(x) = \prod_k \pi(b_k)$. The APP of a binary element becomes

$$APP(b_j) \propto \sum_{x\in QAM|b_j} p(y|x) \prod_{k=1}^{m} \pi(b_k)$$

   The right-hand side of the above equality is rewritten as a product, this yields

$$APP(b_j) \propto \pi(b_j) \times Extr(b_j) \quad where \quad Extr(b_j) \propto \sum_{x\in QAM|b_j} p(y|x) \prod_{k=1,k\neq j}^{m} \pi(b_k) \tag{8}$$

In all cases, the a posteriori probability of a binary element belonging to the label of a QAM symbol is represented by the literal expression

APP = A Priori x Extrinsic

The interpretation of (6), when a non-binary error-correcting code is inserted before the QAM mapper, is similar to the second case described above and is left to the reader. Finally, the graph representation of $APP(x)$ and $APP(b_j)$, as computed by the QAM detector and given in (4) and (8) respectively, is made in Fig. 5. The product of all flows incoming x and b_i is equal to the variable a posteriori probability. In a communication system, the model of Fig. 5 can be realized by feeding the m binary inputs of the QAM mapper with the output of m distinct error-correcting codes. This model is also realized by a unique error-correcting code (e.g. a convolutional code) separated from the QAM mapper by a pseudo-random interleaver.

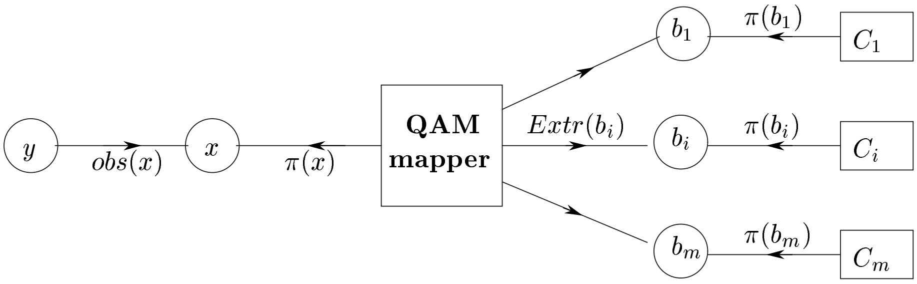


Figure 5: Information flow for bit and symbol nodes in probabilistic QAM detection.

3.2 SISO decoding of binary block and convolutional codes

Consider a linear binary code $C(N,K)_2$ of length N and dimension K. The code C can be constructed algebraically, e.g. a binary BCH code, or by the proper termination of a binary convolutional code. The description below is valid for both classical families of block and convolutional codes. As shown in Fig. 6, the encoder input is $\mathbf{b} = (b_1, b_2, \ldots, b_K)$ where $b_j \in GF(2)$. For simplicity reasons, we assume that the encoder output belongs to a BPSK alphabet[3], $\mathbf{x} = (x_1, x_2, \ldots, x_N)$ where $x_i \in \{\pm A\}$. The simple exercise of combining sections 3.1 and 3.2 is left to the reader.

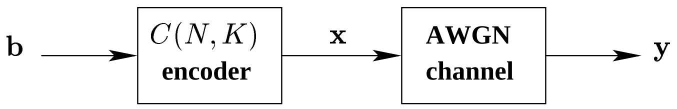


Figure 6: A binary block/convolutional code and a memoryless gaussian channel.

The channel observation is $obs(x_i) \propto p(y_i|x_i) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{(y_i-x_i)^2}{2\sigma^2}\right)$, which leads to

$$obs(x_i) = \frac{1}{1 + exp\left(-\frac{2Ay_i}{\sigma^2}\right)} \tag{9}$$

[3]Binary Phase Shift Keying is equivalent to 2-QAM. A BPSK symbol x_i has values $\pm A$.

The a posteriori probability of a coded symbol is $APP(x_i) = P(x_i|\mathbf{y}, C)$, and if a priori information $\pi(\mathbf{x})$ is available, $\pi(\mathbf{x}) = \prod_{\ell=1}^{N} \pi(x_\ell)$, then we can write

$$APP(x_i) \propto \sum_{\mathbf{x}\in C|x_i} \prod_{\ell=1}^{N} obs(x_\ell)\ \pi(x_\ell)\ \propto\ obs(x_i)\cdot\pi(x_i)\cdot Extr(x_i) \tag{10}$$

$$Extr(x_i) \propto \sum_{\mathbf{x}\in C|x_i} \prod_{\ell\neq i} obs(x_\ell)\ \pi(x_\ell) \tag{11}$$

When the code C is systematic, $APP(b_i) = APP(x_i)$ and $Extr(b_i) = Extr(x_i)$ for $i = 1\ldots K$. If the code C is non-systematic, conditioning in the sum-product formula is made on b_i,

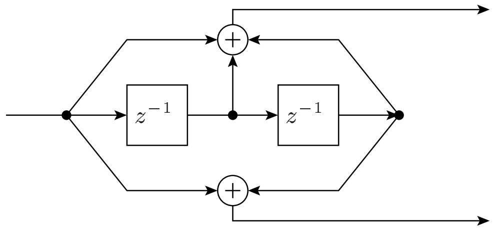
$$APP(b_i) = Extr(b_i) \propto \sum_{\mathbf{x}\in C|b_i} \prod_{\ell=1}^{N} obs(x_\ell)\ \pi(x_\ell) \tag{12}$$

In some situations, the code is non-systematic and a priori information is available on information bits, $\pi(\mathbf{x}) = \prod_{j=1}^{K} \pi(b_j)$, then

$$APP(b_i) \propto \sum_{\mathbf{x}\in C|b_i} \prod_{\ell=1}^{N} obs(x_\ell) \prod_{j=1}^{K} \pi(b_j)\ \propto\ \pi(b_i)\cdot Extr(b_i) \tag{13}$$

$$Extr(b_i) \propto \sum_{\mathbf{x}\in C|b_i} \prod_{\ell=1}^{N} obs(x_\ell) \prod_{j\neq i} \pi(b_j) \tag{14}$$

Version 3 of this tutorial will include material about the trellis structure of block and convolutional codes, and the Forward-backward algorithm. The trellis structure of rate 1/2 non-recursive non-systematic (NRNSC) and recursive systematic (RSC) convolutional codes is given in Fig. 8. The trellis structure of the Hamming code (7,4) defined by its parity-check matrix in (15) is drawn in Fig. 9.



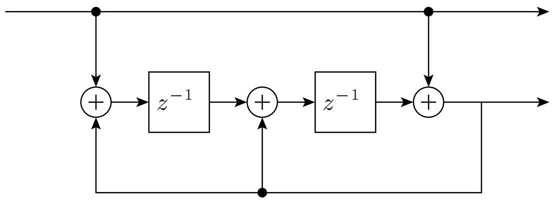


Figure 7: Non-recursive (7,5) convolutional encoder (left) and recursive (1,5/7) convolutional encoder (right). These equivalent codes (same codewords but different input-output mapping) are both 4-state and rate 1/2.

$$H = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \zeta_1 & \zeta_2 & \zeta_3 & \zeta_4 & \zeta_5 & \zeta_6 & \zeta_7 \end{pmatrix} \tag{15}$$

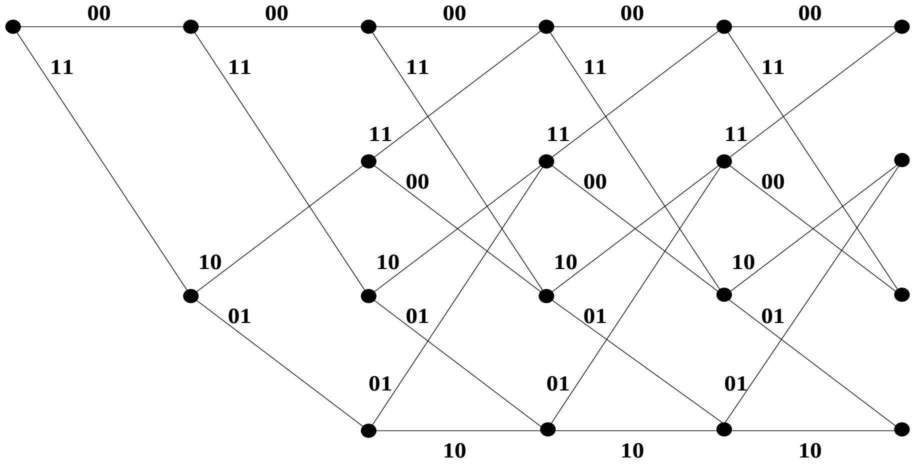


Figure 8: Trellis for equivalent codes NRNSC(7,5) and RSC(1,5/7). Each transition is labeled by two output bits. In the non-systematic case, an upward transition corresponds to a 0-input and a downward transition corresponds to a 1-input. In the systematic case, the input is indicated by the first bit in a transition label.

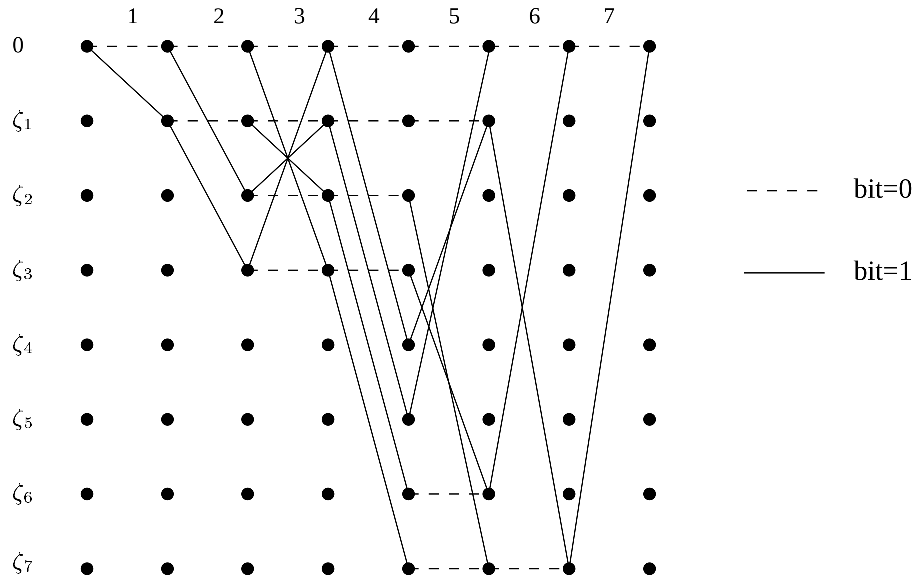


Figure 9: Syndrome trellis of the binary (7,4,3) Hamming code defined by the parity-check matrix in (15). One bit per transition. The 4 trellis sections to the left correspond to the 4 information bits. The trellis has 8 states.

3.3 Iterative APP equalization on ISI channels

Consider an inter-symbol interference channel with memory ν. The ISI channel output before addition of white gaussian noise is $s_i = \sum_{u=0}^{\nu} h_u\ x_{i-u}$. The noisy output is $y_i = s_i + \eta_i$, where η_i is complex gaussian distributed with zero mean and variance σ^2 per real component, $i = 1 \dots N$. The ISI channel is equivalent to a rate-1 convolutional code defined on a complex alphabet. The system model and channel structure are shown in Fig. 10 and 11. The channel state is defined by the content of its tapped delay line. The trellis representation has M transitions per state and a total of M^ν states. Fig. 12 illustrates the ISI trellis for $M = 2$ and $\nu = 2$. Let $\mathcal{T}$ denote the set of ISI codewords, i.e., the set of all possible paths in the ISI trellis. Then, the a posteriori probability for a QAM symbol is evaluated as follows:

$$
\begin{aligned}
APP(x_i) &\propto \sum_{\mathbf{s}\in\mathcal{T}|x_i} p(y|s)\ \pi(s) \ = \sum_{\mathbf{s}\in\mathcal{T}|x_i} \prod_{\ell=1}^{N} p(y_\ell|s_\ell)\ \pi(x_\ell) & (16)\\
&\propto \pi(x_i) \times Extr(x_i) & (17)\\
Extr(x_i) &\propto \sum_{\mathbf{s}\in\mathcal{T}|x_i} \prod_{\ell=1}^{N} p(y_i|s_i) \prod_{u\neq i} \pi(x_u) & (18)
\end{aligned}
$$

Notice that ISI acts like a non-systematic convolutional code, no direct observation is available for transmitted QAM symbols. The channel observation for s_i is given by the classical channel likelihood $p(y_i|s_i) \propto exp\left(-\frac{|y_i - s_i|^2}{2\sigma 2}\right)$. Now, let us write the expression of $APP(b_j)$, where b_j is a binary element belonging to the label of x_i. The a posteriori probability of b_j is obtained by marginalization of symbol probabilities

$$APP(b_j) = \sum_{x_i|b_j} APP(x_i) \ \propto \sum_{x_i|b_j} \pi(x_i)\ Extr(x_i) \tag{19}$$

$$APP(b_j) \propto \pi(b_j) \sum_{x_i|b_j} Extr(x_i) \prod_{\ell\neq j}^{m} \pi(b_\ell) \ \ \propto \ \pi(b_j)\ Extr(b_j) \tag{20}$$

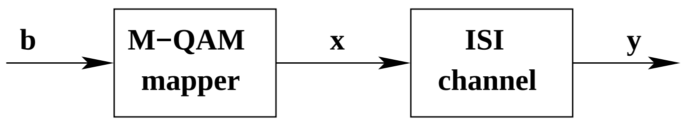


Figure 10: Mapper for M-QAM modulation followed by an inter-symbol interference channel.

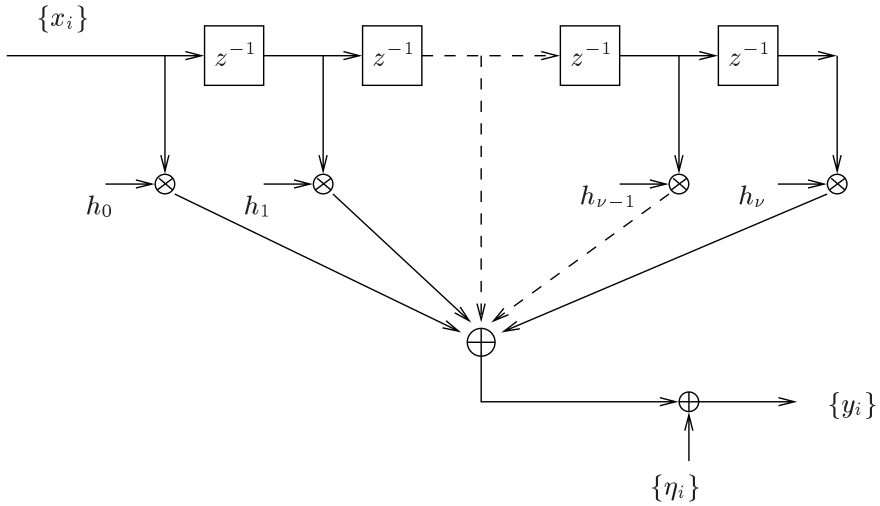


Figure 11: Tapped delay line as a discrete model for an ISI channel of $\nu + 1$ coefficients.

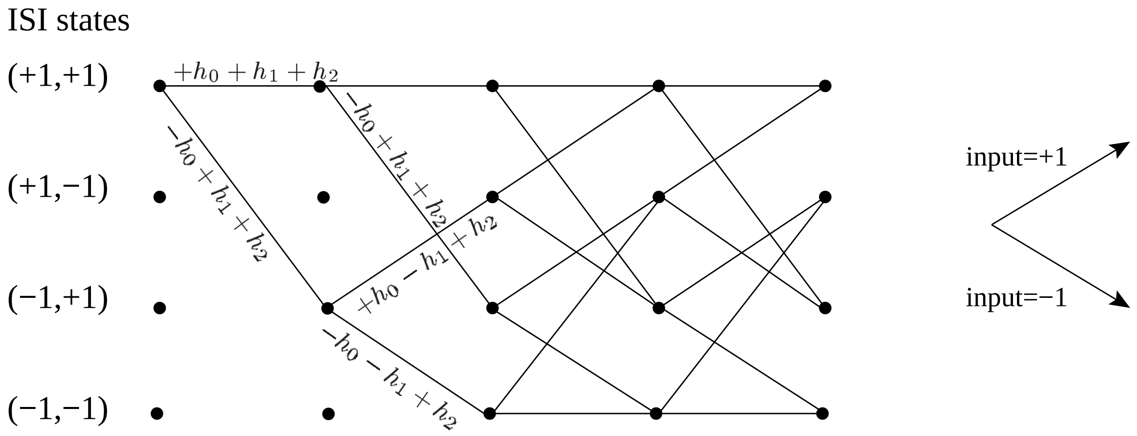


Figure 12: Trellis for a memory $\nu = 2$ inter-symbol interference channel with BPSK input.

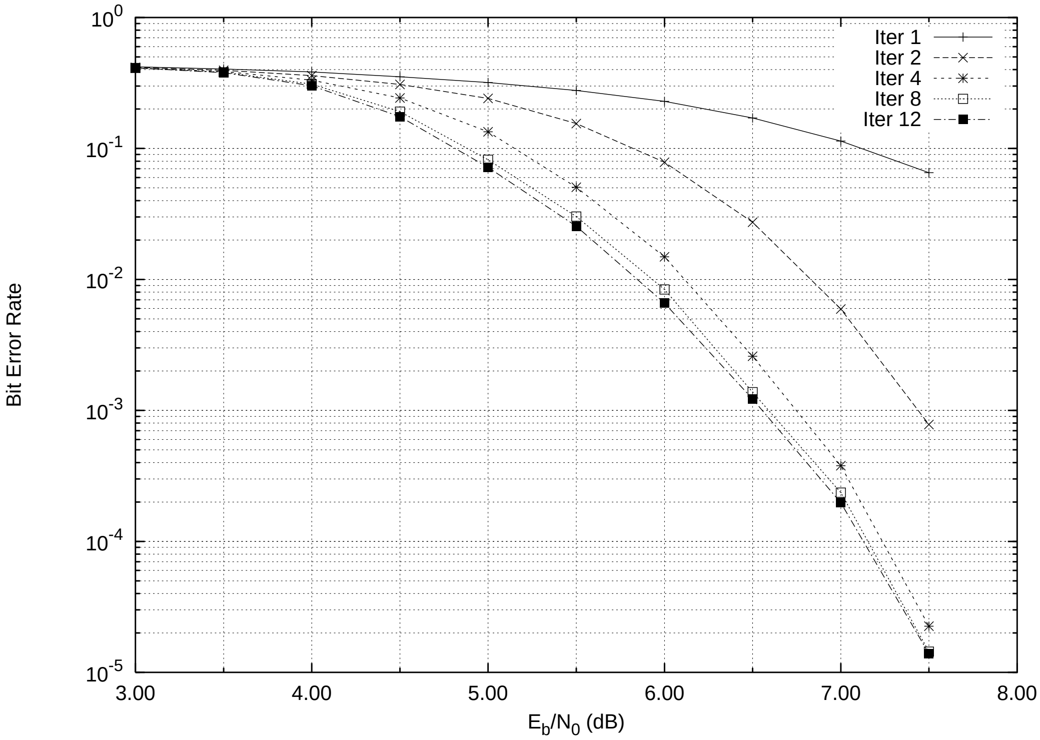


Figure 13: Iterative APP equalization performance for a memory $\nu = 4$ ISI channel and BPSK input encoded with a 64-state rate 1/2 convolutional code NRNSC(133,171).

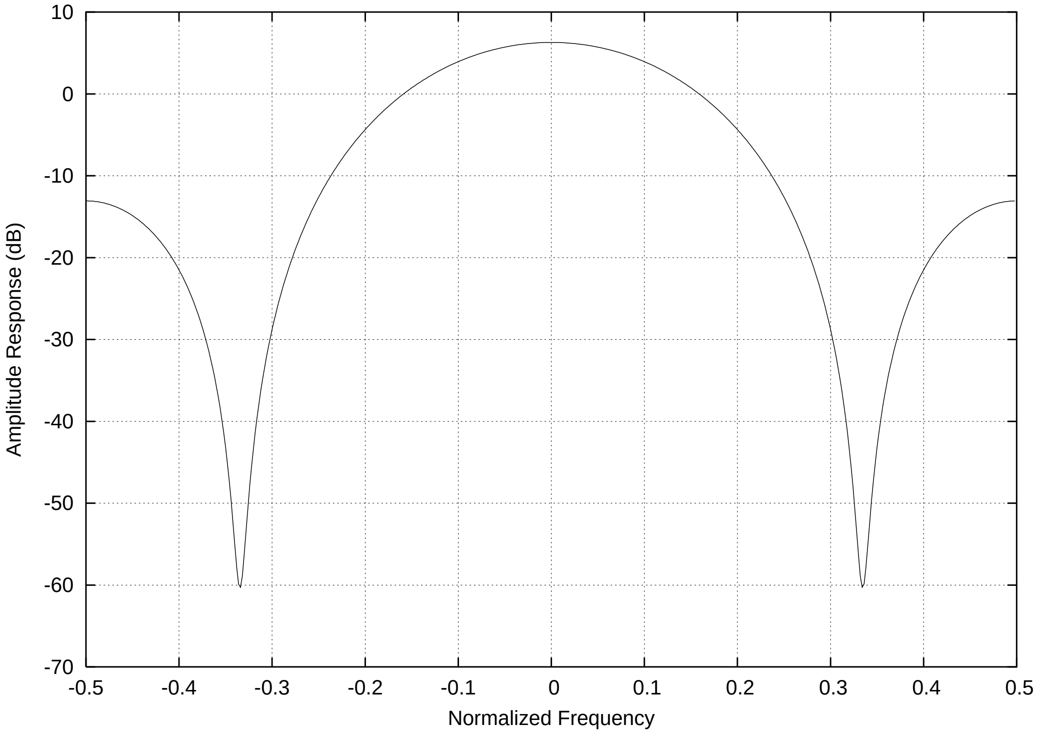


Figure 14: Amplitude frequency response for the 5-coefficient ISI channel used above in iterative equalization and decoding.

3.4 Iterative APP detection on MIMO channels

Notations and definitions are similar to previous sections. The description herein will be very compact. Assume a frequency non-selective multiple-input multiple-output (MIMO) channel with n_t transmit antennas and n_r receive antennas. The channel coefficients are given by the entries of a $n_t \times n_r$ matrix $H = [h_{ij}]$, where h_{ij} is the complex fading of the channel path linking transmit antenna i to receive antenna j.
The channel output is $\mathbf{y} = \mathbf{x}H + \eta$, where $\mathbf{x} \in (M-QAM)^{n_t} \subset \mathbb{C}^{n_t}$, $\mathbf{y} \in \mathbb{C}^{n_r}$, and η is an additive white complex gaussian noise vector perturbing the receive antennas. The system model (not including the error-correcting code) is illustrated in Fig. 15. Expressions for a posteriori probability and extrinsic information are similar to those found in section 3.1, the modulation alphabet of size M is replaced by a multidimensional alphabet $\Omega = (M-QAM)^{n_t}$ of size $M^{n_t} = 2^{mn_t}$.

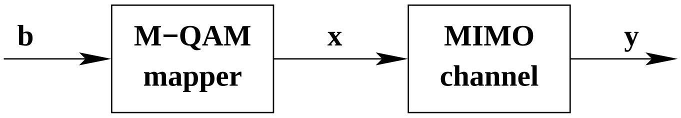


Figure 15: Mapper for M-QAM modulation followed by a multiple antenna channel.

The observation of a multidimensional symbol is

$$obs(\mathbf{x}) \propto p(\mathbf{y}|\mathbf{x}) \propto exp\left(-\frac{\|\mathbf{y}-\mathbf{x}H\|^2}{2\sigma^2}\right) \tag{21}$$

Independent a priori information is supposed to be available, $\pi(\mathbf{x}) = \prod_{j=1}^{mn_t} \pi(b_j)$. The multidimensional a posteriori is $APP(\mathbf{x}) \propto obs(\mathbf{x})\pi(\mathbf{x})$. No extrinsic $Extr(\mathbf{x})$ is generated by the detector because the MIMO channel is DMC. Recalling that $\mathbf{x} = (x_1, x_2, \ldots, x_{n_t})$, where $x_i \in M-QAM$, the a posteriori probability for a complex QAM symbol is

$$APP(x_i) = \sum_{\mathbf{x}\in\Omega|x_i} APP(\mathbf{x}) \;\propto\; \sum_{\mathbf{x}\in\Omega|x_i} obs(\mathbf{x}) \prod_{\ell=1}^{n_t} \pi(x_\ell) \;\propto\; \pi(x_i)\; Extr(x_i) \tag{22}$$

In the above APP, we assumed that $\pi(x_\ell) = \prod_{j=1}^{m} \pi(b_{(\ell-1)m+j})$. Now, the a posteriori information for binary elements is

$$APP(b_j) \propto \pi(b_j) \underbrace{\sum_{\mathbf{x}\in\Omega|b_j} obs(\mathbf{x}) \prod_{\ell\neq j} \pi(b_\ell)}_{Extr(b_j)} \tag{23, 24}$$

3.5 Probabilistic multiuser detection on CDMA channels

Notations and definitions are similar to previous sections. The description herein will be very compact. The channel model is drawn in Fig. 16. The received sample at the joint detector input is $\mathbf{y} = y = \sum_{i=1}^{K} \omega_i x_i + \eta$, where ω_i is the fading coefficient for user i and K is the number of users. The CDMA channel is non-systematic. Hence, the a posteriori probability for a symbol x_i will be proportional to the product of a priori and extrinsic informations without any direct observation, i.e., $obs(x_i) = 1/M$. The Cartesian product of the K modulation alphabets is denoted $\Omega = (M - QAM)^K$. The a posteriori probability for a transmitted M-QAM symbol by user i is

$$APP(x_i) = P(x_i|y, code\ constraints) \ \propto \ \pi(x_i)\ Extr(x_i) \tag{25}$$

$$Extr(x_i) \ \propto \sum_{x_i \in \Omega | x_i} p(y|\mathbf{x}) \prod_{\ell \neq i} \pi(x_\ell) \tag{26}$$

where the channel likelihood is given by

$$p(y|\mathbf{x}) \ \propto \ exp\left(-\frac{|y - \sum_{\ell=1}^{K} \omega_\ell x_\ell|^2}{2\sigma^2}\right) \tag{27}$$

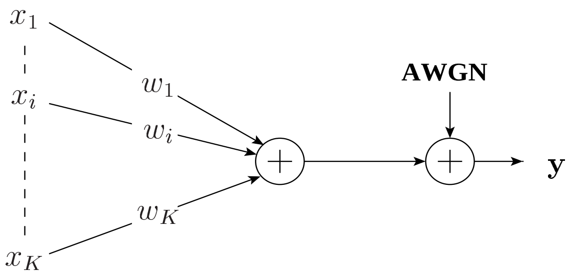


Figure 16: Chip synchronous code division multiple access channel for K users.

The performance of a CDMA system with $K = 4$ users is illustrated in Fig. 17. The symbols belong to a BPSK modulation, $x_i \pm A$. Each user encodes its information with a rate $R_c = 1/4$ 16-state non-recursive non-systematic convolutional code with octal generators (25,27,33,37). No pseudo-noise (PN) sequence spreading is applied. Hence, the spreading factor is $S_f = 1/R_c = 4$. The CDMA system load is $K/S_f = 1 = 100\%$. Also, the considered users have equal amplitude $w_i = 1$ for all i, i.e., the K users have identical signal-to-noise ratios per bit.

3.6 APP decoding of binary Turbo/LDPC/GLD codes

Notations and definitions are similar to previous sections. The description herein will be very compact. Consider a binary compound error-correcting code and an ideal AWGN channel with BPSK modulation. The system model is drawn in Fig. 18 and the compound code structure is found in Fig. 19. The compound code (also called *graph code* or *concatenated code*) is defined by bit nodes and constraint nodes. Let us consider the 3 most important type of graph codes:

- **Low-density parity-check (LDPC) codes**. A codeword has length N bits, the latter are represented by the N left nodes in the bipartite graph. For example, make $d_b = 3$ and $d_c = 6$. Take $C_0 = SPC(d_c, d_c - 1, 2)_2$ a single parity-check code. The graph has $L = \frac{d_b N}{d_c}$ check nodes. The LDPC coding rate is $R_c \geq 1 - \frac{d_b}{d_c} = 1/2$. Equality occurs with high probability for large N. This description corresponds to a regular LDPC code. Irregular LDPC codes are obtained by defining irregular degree distributions for

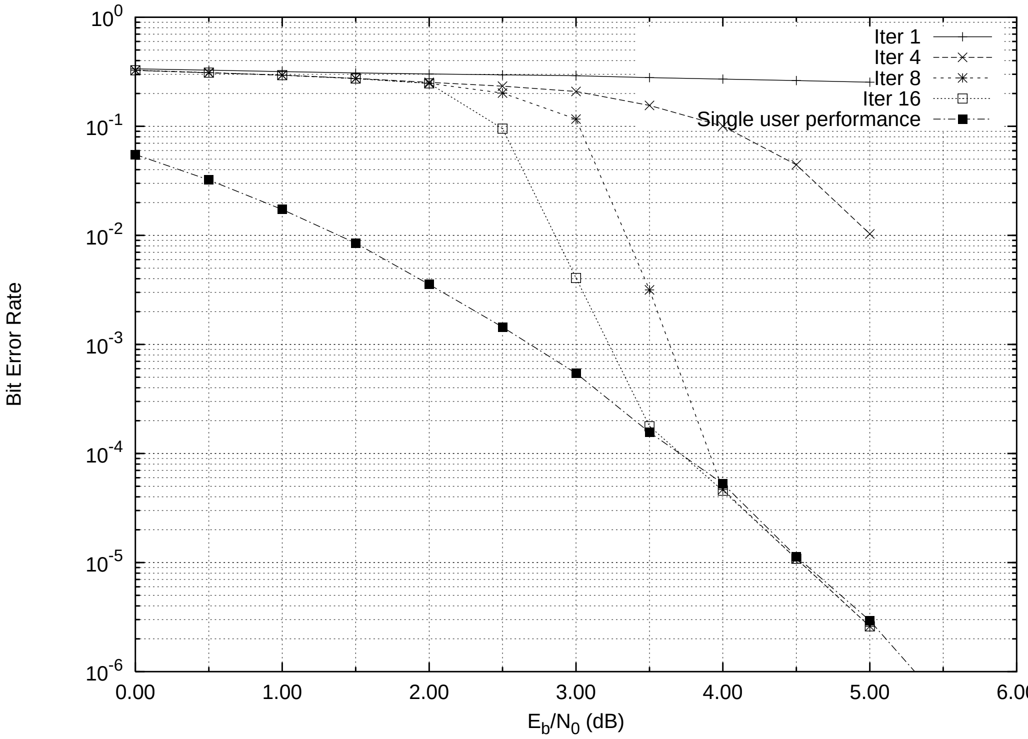


Figure 17: Iterative APP joint detection in CDMA with $K = 4$ users on a gaussian channel. The NRNSC code is a rate 1/4 16-states (25,27,33,37) for all users. Same SNR per bit for all users. Each user pseudo-randomly interleaves its $N = 8192$ bits before transmitting on the multiple access channel. No PN spreading. System load is 100%.

bit nodes and check nodes. The decoding of an LDPC code is based on decoding of its SPC constituent. In this case, we can apply to C_0 what has been described in section 3.2. However, C_0 is an $SPC(d_c, d_c - 1, 2)$ which leads to simpler expressions for the extrinsic information. Let b_j, $j = 1 \dots d_c$, denote the d_c bits of an SPC code, then

$$Extr(b_j = 1) \ = \ \frac{1}{2} \times \left(1 - \prod_{\ell=1, \ell \neq j}^{d_c} (1 \ - \ 2p_\ell) \right) \tag{28}$$

where $p_\ell \propto \pi(b_\ell = 1)\, obs(b_\ell = 1)$. The a posteriori probability is obtained by multiplying the 3 informations, $APP(b_j) \propto obs(b_j)\pi(b_j)Extr(b_j)$. The input a priori of b_j is obtained by multiplying the incoming extrinsics to the check node from all other bit nodes, as described in section 2.

- **Generalized low-density (GLD/Tanner) codes**. A codeword has length N bits, the latter are represented by the N left nodes in the bipartite graph. For example, make $d_b = 2$ and $d_c = 20$. Take $C_0 = BCH(20, 15)_2$, the binary BCH code constructed by shortening the $(31, 26)_2$. In general, C_0 may be any binary $BCH(n, k)_2$ where $n = d_c$. The graph has $L = \frac{d_b N}{d_c}$ subcode nodes. The GLD coding rate is $R_c \geq 2\frac{k}{n} - 1$. Equality occurs with high probability for large N. This description corresponds to a regular GLD code. Irregular GLD codes are obtained by defining irregular degree distributions for bit nodes and check nodes, and different C_0 constituents. The decoding of a GLD code is based on decoding of its C_0 constituent, as described in section 3.2.

- **Parallel turbo (Turbo) codes**. Make $L = 2$ subcode nodes. Take $C_0 = RSC(1, g_2/g_1)$, a recursive systematic convolutional code of rate 1/2 and generator polynomials $g_1(x)$ and $g_2(x)$. The N bit nodes in the left side of the graph are considered as information bits. Make the left degree $d_b = 2$. Make the right degree $d_c = N$, and add N parity bits of degree 1 to each subcode node. Then, the graph code is a rate 1/3 parallel turbo code. If one considers $L = 1$ subcode node with a total degree $d_c = 2N$, the new graph code is an equivalent version of a turbo code with a unique constituent.

Notice that in the 3 cases above, the matching between left nodes and right nodes in the code graph is random. Some algebraic deterministic graphs have been also studied in the literature.

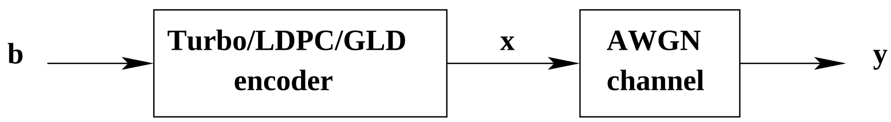


Figure 18: Binary Turbo/LDPC/GLD encoder on AWGN channel, BPSK modulation.

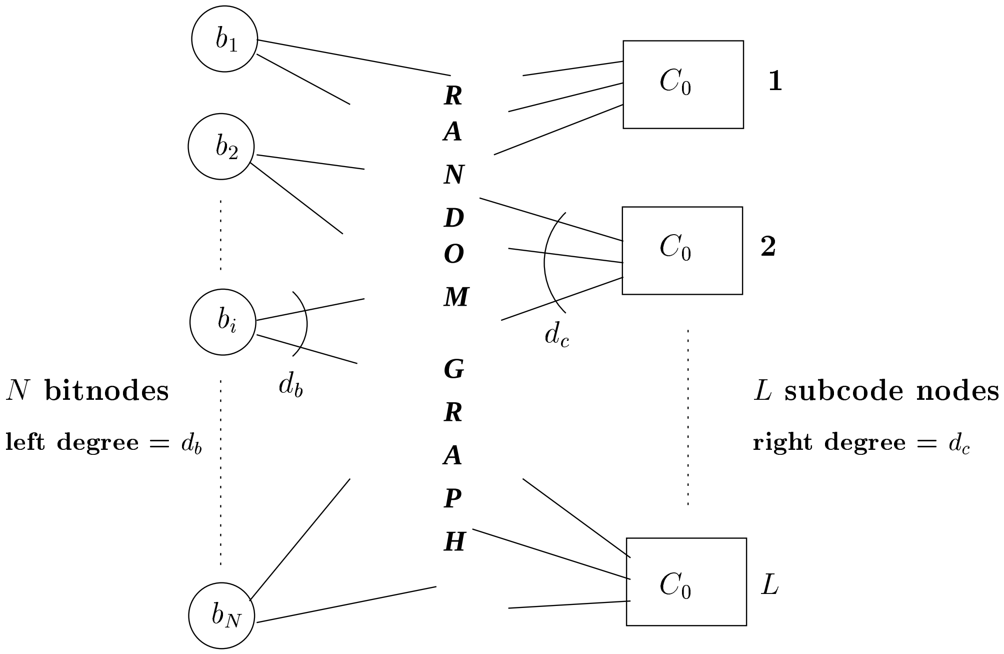


Figure 19: Bipartite graph for a regular binary Turbo/LDPC/GLD code.

4 Parameter estimation via expectation-maximization (EM)

In previous sections, we explained (at least, we tried to explain) in a simplified and a universal manner, how to write probabilistic expressions useful for decoding information in communication systems. Take the following simplistic example. The probabilistic observation of a binary symbol b_i on an ideal coherent real channel with additive white gaussian noise $\mathcal{N}(0, \sigma^2)$ and binary phase shift-keying modulation is computed by 9

$$obs(x_i = +A) \;=\; \frac{p(y_i|x_i = -A)}{p(y_i|x_i = +A) + p(y_i|x_i = +A)} \;=\; \frac{1}{1 + exp\left(-\frac{2Ay_i}{\sigma^2}\right)} \tag{29}$$

The symbol amplitude A and the noise variance σ^2 are required in the evaluation of $obs(x_i)$. In general, channel state information (CSI) is mandatory for detection and decoding. The source distribution, denoted by SSI (source state information), is mandatory for non-universal source decoding. SSI may also improve the quality of service if it is available for the channel decoder. Thus, the parameter θ to be estimated at the receiver side includes both SSI and CSI as indicated in Fig. 20. In the simplistic example above, θ is the vector of parameters $\theta = (A, \sigma^2)$, the binary source is assumed to be i.i.d. and uniformly distributed, $P(-A) = P(+A) = 1/2$. On a MIMO channel, when the source is uniform, parameters are $\theta = CSI = (A, H, \sigma^2)$. On a multiple access channel, CSI is given by the K channel parameters of the K users.

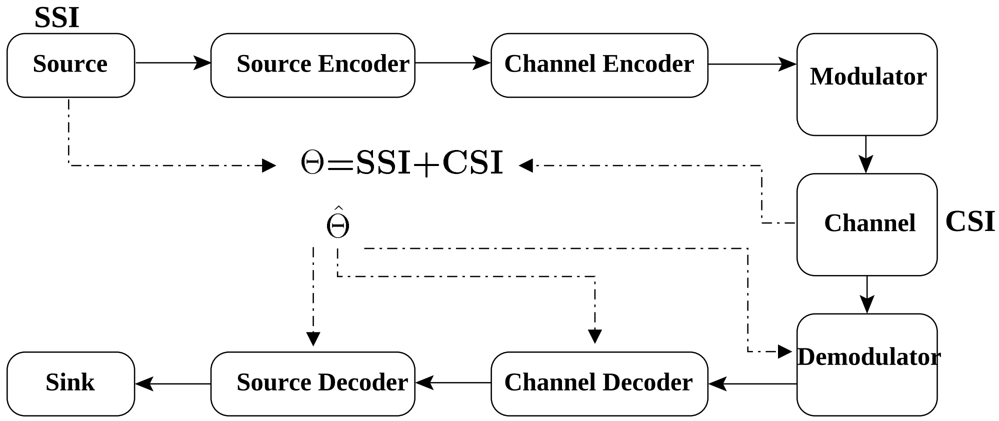


Figure 20: General model for a communication system. Parameters to be estimated are the source distribution and the channel state information. Quality of service is improved by feeding the receiver with the estimated value of system parameters.

Before revealing parameter estimation with the means of EM algorithm, the author would like to propose an appetizer. Consider the following communication system:

- A non-uniform binary i.i.d. source with distribution $\mu = P(1) = 0.1$ and $1 - \mu = P(0)$.
- The source output is encoded by a linear systematic binary LDPC(N,K) encoder. No source code in this communication system.
- LDPC code symbols are BPSK modulated and transmitted on an ideal channel, as in the above simplistic example.

- The CSI (A, σ^2) is perfectly known by the LDPC probabilistic decoder. The receiver has no source decoder.

Then, two situations can be encountered. First, the LDPC decoder has no access to SSI, it assumes that $\hat{\mu} = 1/2$. In the second situation, the LDPC decoder has perfect knowledge $\hat{\mu} = \mu = 0.1$. In Fig. 21, the word error rate in both situations is plotted versus the signal-to-noise ratio. The reader is invited to meditate on the illustrated performance.

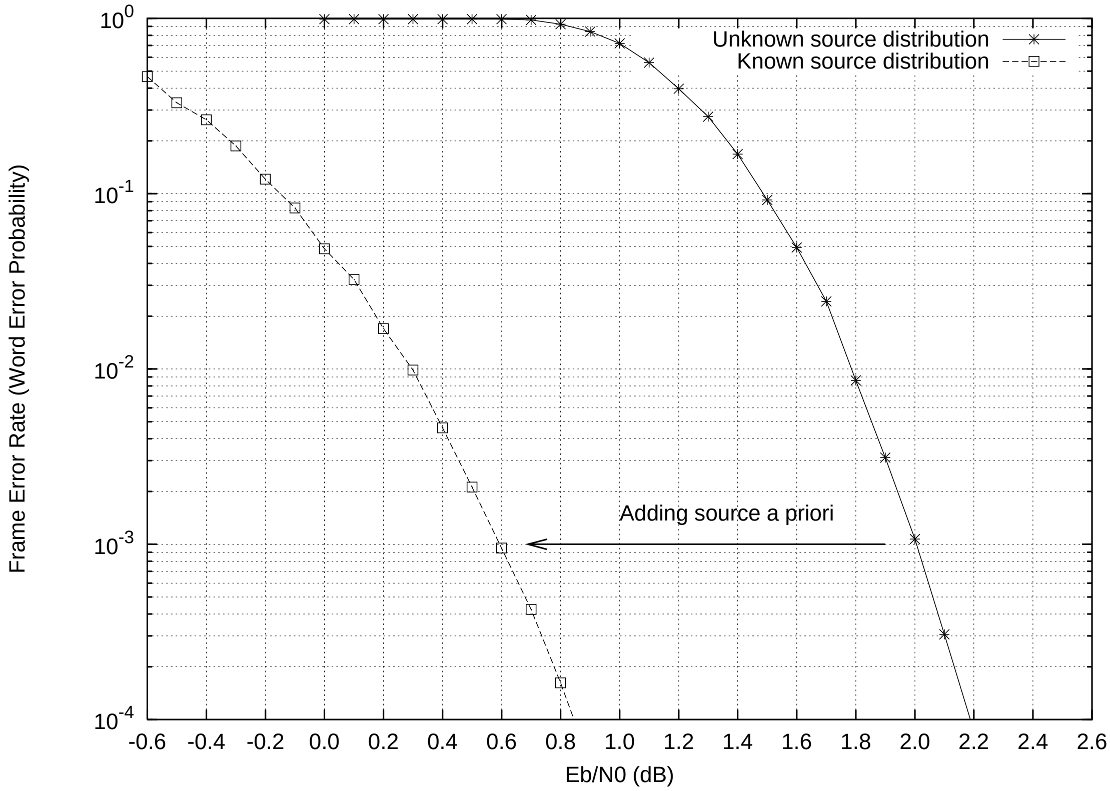


Figure 21: Word error rate versus signal-to-noise ratio per bit for a systematic regular LDPC code, $d_b = 3$, $d_c = 6$, rate 1/2. The code length is $N = 2000$ bits and dimension is $K = 1000$. Channel is gaussian with BPSK input. The non-uniform binary source distribution is $P(0) = 0.9$ and $P(1) = 0.1$. Number of probabilistic decoding iterations is 100. Measure of at least 500 bit errors and 100 word errors for each plotted point.

4.1 General statement of the EM algorithm

Let $\mathbf{x}$ denote the symbol vector at the channel input. Let $\mathbf{y}$ denote the observation vector at the channel output. In communication and coding systems, components of $\mathbf{x}$ (e.g. the QAM symbols x_i defined in previous sections) belong to a finite discrete alphabet. The channel output $\mathbf{y}$ can be discrete (e.g. BSC) or continuous (e.g. AWGN channel). As depicted in Fig. 22, for simplicity reasons, the general model of Fig 20 is reduced to a representation where $\mathbf{x}$ is the direct source output. It is straightforward to place an encoder (systematic or non-systematic) between the source and the channel without a major change in EM equations.

As indicated in Fig. 22, $\mathbf{x}$ is called the missing data, $\mathbf{y}$ is called the incomplete data, and $\kappa = (\mathbf{x}, \mathbf{y})$ is the complete data. Indeed, if the complete data κ is available, then the source distribution can be easily estimated from $\mathbf{x}$, and the channel state information can be easily estimated from κ. When κ is available, maximum-likelihood (ML) estimation of θ is obtained by maximizing the likelihood or equivalently the log-likelihood

$$\hat{\theta}_{ML} = \arg \max_{\theta} \log p(\kappa|\theta) = \log p(\mathbf{x}, \mathbf{y}|\theta) \tag{30}$$

In a communication system, the incomplete data is the only available observation to the decoder. The transmitted vector $\mathbf{x}$ is missing, and ML estimation as in (30) cannot be performed. In such situations, ML estimation becomes

$$\hat{\theta}_{ML} = \arg \max_{\theta} \log p(\mathbf{y}|\theta) \tag{31}$$

Unfortunately, in many cases, analytical or numerical ML estimation by maximizing $\log p(\mathbf{y}|\theta)$ is not possible because an explicit expression for the log-likelihood does not exist, or because maximization over θ is an extremely difficult task. The EM algorithm provides a recursive solution to (31). Given a current parameter value θ^i at iteration i, the EM algorithm computes an update θ^{i+1}. The final EM estimate depends on the initial value θ^0. The EM algorithm is numerically stable, and its convergence is relatively fast in most communication and coding systems. In difficult scenarios, θ^0 should be determined by the use of pilot symbols.

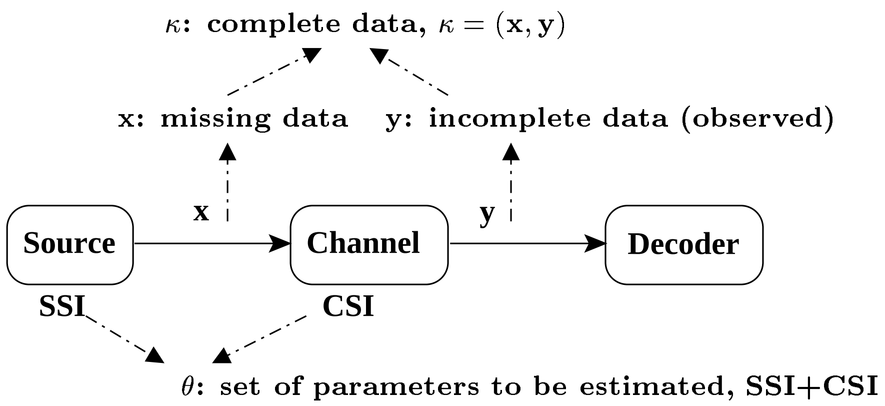


Figure 22: Simplified communication system model for introducing EM vocabulary.

Expectation-maximization was employed in various algorithms in the past. Dempster, Laird and Rubin gave an exact formulation for it in 1977, proved its convergence and called it EM algorithm. For example, the so-called Baum-Welch algorithm (known at least since 1970) includes an EM parameter estimation and a forward-backward recursion (similar to the BCJR published in 1974).

When the symbol vector $\mathbf{x}$ is missing, the key idea is to replace the log-likelihood function by its mathematical expectation over $\mathbf{x}$, given the observed data and the current parameter value. The estimation algorithm proceeds in two steps at each iteration:

- E-step (with source a priori, SSI): Compute the auxiliary function

$$\begin{aligned} Q(\theta|\theta^i) &= E[\log p(\mathbf{x},\mathbf{y}|\theta) \mid \mathbf{y},\theta^i] && (32)\\ &= \sum_{\mathbf{x}} \log p(\mathbf{x},\mathbf{y}|\theta)\ P(\mathbf{x}|\mathbf{y},\theta^i) && (33)\\ &= \sum_{\mathbf{x}} \log p(\mathbf{x},\mathbf{y}|\theta)\ APP_i(\mathbf{x}) && (34) \end{aligned}$$

- M-step: Update the parameter

$$\theta^{i+1} = \arg\ \max_{\theta}\ Q(\theta|\theta^i) \quad (35)$$

When the missing data does not depend on the parameter, i.e., no a priori information on the source distribution, then writing $p(\mathbf{x},\mathbf{y}|\theta) = p(\mathbf{y}|\mathbf{x},\theta)p(\mathbf{x})$ yields a new auxiliary function:

- E-step (without source a priori, no SSI): Compute the auxiliary function

$$\begin{aligned} Q(\theta|\theta^i) &= E[\log p(\mathbf{y}|\mathbf{x},\theta) \mid \mathbf{y},\theta^i] && (36)\\ &= \sum_{\mathbf{x}} \log p(\mathbf{y}|\mathbf{x},\theta)\ P(\mathbf{x}|\mathbf{y},\theta^i) && (37)\\ &= \sum_{\mathbf{x}} \log p(\mathbf{y}|\mathbf{x},\theta)\ APP_i(\mathbf{x}) && (38) \end{aligned}$$

- M-step: Update the parameter

$$\theta^{i+1} = \arg\ \max_{\theta}\ Q(\theta|\theta^i) \quad (39)$$

Now, let us examine the convergence of EM recursion. What necessary condition is satisfied if $\theta = \theta^i = \theta^{i+1}$? From (33), we can write

$$Q(\theta|\theta^i) \propto \sum_{\mathbf{x}} \log p(\mathbf{x},\mathbf{y}|\theta)\ p(\mathbf{x},\mathbf{y}|\theta^i) \quad (40)$$

Derive with respect to θ, and force the derivative to zero at $\theta = \theta^i = \theta^{i+1}$,

$$\frac{\partial Q}{\partial \theta} \propto \sum_{\mathbf{x}} \frac{\partial p(\mathbf{x},\mathbf{y}|\theta)}{\partial \theta}\ \frac{p(\mathbf{x},\mathbf{y}|\theta^i)}{p(\mathbf{x},\mathbf{y}|\theta)} = \frac{\partial p(\mathbf{y}|\theta)}{\partial \theta} = 0 \quad (41)$$

The above equality, obtained at EM convergence, is the necessary condition satisfied by the ML estimate of θ. This tells us that EM stops in a local optimum which may not be a global optimum. Definitely, the initial value θ^0 is very critical in many cases.
The above analysis is made for the steady-state of convergence. So, does the likelihood of the observed data increase at every EM iteration ? The answer is yes:
Since $p(\mathbf{x},\mathbf{y}|\theta) = p(\mathbf{y}|\theta)\ p(\mathbf{x}|\mathbf{y},\theta)$, (32) becomes $Q(\theta|\theta^i) = E[\log p(\mathbf{x}|\mathbf{y},\theta)|\mathbf{y},\theta^i] + \log p(\mathbf{y}|\theta)$. Then, we write the difference of log-likelihoods $i+1$ and i,

$$\begin{aligned} \log p(\mathbf{y}|\theta^{i+1}) - \log p(\mathbf{y}|\theta^i) &= Q(\theta^{i+1}|\theta^i) - Q(\theta^i|\theta^i) && (42)\\ &+ E[\log p(\mathbf{x}|\mathbf{y},\theta^i)|\mathbf{y},\theta^i] - E[\log p(\mathbf{x}|\mathbf{y},\theta^{i+1})|\mathbf{y},\theta^i] && (43)\\ &= Q(\theta^{i+1}|\theta^i) - Q(\theta^i|\theta^i) + D(p(\mathbf{x})|\mathbf{y},\theta^i||p(\mathbf{x}|\mathbf{y},\theta^{i+1})) && (44)\\ &\geq 0 && (45) \end{aligned}$$

In (44), we used the inequality $Q(\theta^{i+1}|\theta^i) \geq Q(\theta^i|\theta^i)$ and the fact that Kullback-Leibler distance $D(p||q)$ is non-negative.

Before giving two applications of EM on BSC and gaussian channel respectively, we would like to compare EM approach with other known algorithms in communication and coding theory. As a first comparison, consider non-coherent detection of signals on a gaussian channel. The continuous-time complex channel model is $y(t) = s(t)\ e^{j\phi} + \eta(t)$, where ϕ is a random phase, $s(t)$ is the transmitted signal with unitary energy, $\eta(t)$ is the additive complex white gaussian noise with power spectral density $2\sigma^2$, and $y(t)$ is the observed signal. The probabilistic observation $obs(s(t)) \propto p(y(t)|s(t))$ cannot be computed directly since $y(t)$ depends on ϕ. The classical solution is found by assuming that ϕ is uniformly distributed in $[0 \ldots 2\pi]$ (no phase a priori) and then $p(y(t)|s(t))$ is obtained by a mathematical expectation on ϕ,

$$obs(s(t)) \propto p(y(t)|s(t)) \ = \ \int_0^{2\pi} p(y(t)|s(t),\phi)\ p(\phi)d\phi \ \propto \ I_0\left(\frac{|< y(t), s(t) >|}{\sigma^2}\right) \tag{46}$$

where the scalar product is $< y(t), s(t) >= \int y(t)s^*(t)dt$. The philosophy of non-coherent detection is similar to EM, just do expectation on missing data.
In the second example, we compare EM to Blahut-Arimoto algorithm for calculation of channel capacity and rate-distortion function. The algorithm of Blahut and Arimoto is a special case of Csiszár-Tusnády alternating optimization. For calculation of channel capacity, Blahut-Arimoto algorithm is a Csiszár-Tusnády alternating maximization where channel capacity is written as

$$C \ = \ \max_{q(x|y)} \max_{r(x)} \sum_x \sum_y r(x)p(y|x) \log \frac{q(x|y)}{r(x)} \tag{47}$$

Given a current input distribution $r(x)$, the best conditional distribution is

$$q(x|y) \ \propto \ r(x)\ p(y|x) \tag{48}$$

Then, by solving a constrained maximization, the input distribution is updated with

$$r(x) \ \propto \ \prod_y \ (q(x|y))^{p(y|x)} \tag{49}$$

The capacity expression in (47) plays the role of an auxiliary function and the recursive update given by (48) and (49) is equivalent to an EM parameter update.

4.2 An example of EM estimation on BSC channel

The channel is binary symmetric with transition probability $\lambda \in [0..1/2]$. The source output at time j is added (sum in GF(2)) to an error e_j, $y_j = x_j + e_j$, $j = 1 \ldots N$. The source distribution is $\mu = P(x_j = 1) \in [0..1]$.
By observing the incomplete data $\mathbf{y}$, we wish to estimate $\theta = (\mu, \lambda)$. The EM auxiliary function is $Q(\theta|\theta^i) = E_{\mathbf{x}}[\log P(\kappa|\theta)|\mathbf{y}, \theta^i]$. Let us write the joint distribution as a function of parameters λ and μ,

$$P(\kappa|\theta) = P(\mathbf{x},\mathbf{y}|\theta) = P(\mathbf{y}|\mathbf{x},\theta)P(\mathbf{x}|\theta) = \mu^{w_H(\mathbf{x})}(1-\mu)^{N-w_H(\mathbf{x})}\lambda^{w_H(\mathbf{e})}(1-\lambda)^{N-w_H(\mathbf{e})} \tag{50}$$

where $w_H(\mathbf{x}) = \sum_{j=1}^N x_j$ and $w_H(\mathbf{e}) = \sum_{j=1}^N e_j$ are the Hamming weights (sum in $\mathbb{Z}$) of the source output and the channel error vector respectively. Keeping the same notation $E_{\mathbf{x}}[]$

for mathematical expectation over $\mathbf{x}$ with the joint distribution $P(\mathbf{x}|\mathbf{y}, \theta^i)$, i.e., the APP at iteration i for the codeword x, then

$$E_{\mathbf{x}}[w_H(\mathbf{x})] = \sum_{j=1}^{N} \tilde{x}_j \qquad E_{\mathbf{x}}[w_H(\mathbf{e})] = \sum_{j=1}^{N} \tilde{e}_j \tag{51}$$

where $\tilde{x}_j = E_{\mathbf{x}}[x_j]$ is a soft symbol and $\tilde{e}_j = E_{\mathbf{x}}[e_j]$ is a soft error component.

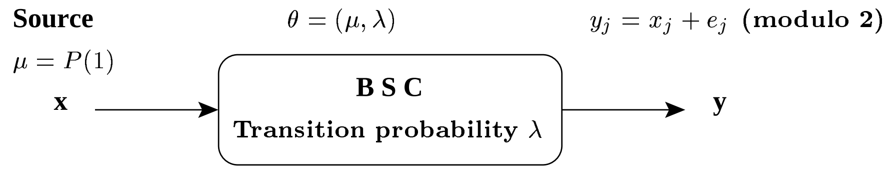


Figure 23: EM estimation with a binary source on a BSC channel. The parameter to be estimated includes the source distribution and the channel transition probability.

The evaluation of $\tilde{x}_j = \sum_{\mathbf{x}} x_j P(\mathbf{x}|\mathbf{y}, \theta^i)$ is intractable in most cases, e.g., $|C(N,K)_2| = 2^K$ is not a small number. Hence, we use the following approximation which has a negligible degradation on EM performance and a very low evaluation complexity

$$\sum_{\mathbf{x}} x_j P(\mathbf{x}|\mathbf{y}, \theta^i, \textit{code constraints}) = \sum_{x_j} x_j P(x_j|\mathbf{y}, \theta^i, \textit{code constraints}) \tag{52}$$

$$\tilde{x}_j = \sum_{\mathbf{x}} x_j APP_i(\mathbf{x}) = \sum_{x_j} x_j APP_i(x_j) \tag{53}$$

Important Notice: Soft symbols are computed by their mathematical expectation with marginal a posteriori $APP(x_j)$ instead of joint a posteriori $APP(\mathbf{x})$. This linear time evaluation is used to compute soft symbols (also called *expected symbols*) in all kind of channels. In special circumstances, second or higher order moments are required for EM estimation, e.g., taking $j \neq \ell$, the expected product of symbols is

$$\widetilde{x_j x_\ell} = \sum_{x_j, x_\ell} x_j x_\ell APP_i(\mathbf{x}) \approx \sum_{x_j} x_j APP_i(x_j) \sum_{x_\ell} x_\ell APP_i(x_\ell) = \tilde{x}_j \tilde{x}_\ell \tag{54}$$

Second or higher order moments are evaluated by approximating joint a posteriori probability by the product of its marginals. In practice, for finite length constraints, soft output decoders generate an acceptable approximation for the marginal distribution $APP_i(x_j)$.
Using the above notice, soft symbols are obtained by

$$\tilde{x}_j = APP_i(x_j = 1) \qquad \tilde{e}_j = APP_i(e_j = 1) = [APP_i(x_j = 1)]^{\bar{y}_j}[1 - APP_i(x_j = 1)]^{y_j} \tag{55}$$

Finally, the auxiliary function on BSC channel with parameter $\theta = (\mu, \lambda)$ is

$$Q(\theta|\theta^i) = \sum_{j=1}^{N} \tilde{x}_j \log\left(\frac{\mu}{1-\mu}\right) + N\log(1-\mu) + \sum_{j=1}^{N} \tilde{e}_j \log\left(\frac{\lambda}{1-\lambda}\right) + N\log(1-\lambda) \tag{56}$$

By writing $\partial Q/\partial\mu = 0$ and $\partial Q/\partial\lambda = 0$, the EM parameter updates for the source distribution and the channel transition probability are

$$\mu^{i+1} = \frac{\sum_{j=1}^{N} \tilde{x}_j}{N} \qquad \lambda^{i+1} = \frac{\sum_{j=1}^{N} \tilde{e}_j}{N} \tag{57}$$

Fig. 24 illustrates the performance for a computer simulation of EM on BSC with binary BCH codes. In two iterations, the error rate for EM estimation is superimposed with the error rate when parameters are perfectly known!

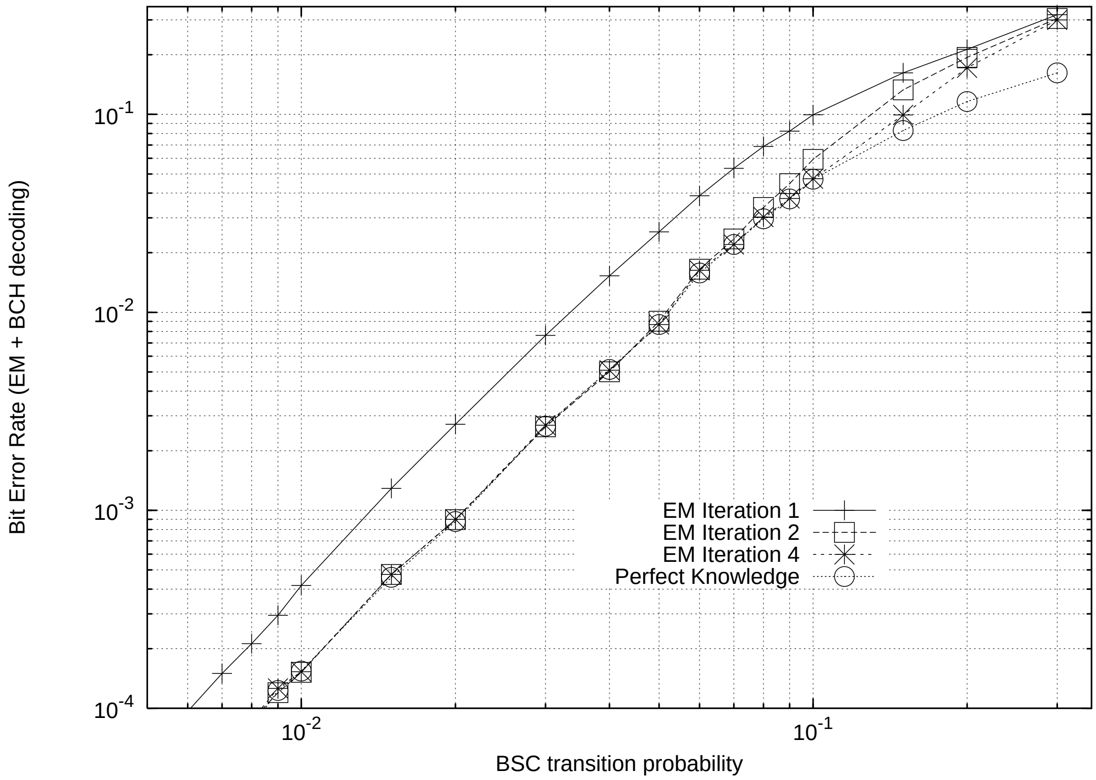


Figure 24: Source and channel parameters estimation via the EM algorithm. The plot above shows the bit error rate for a posteriori probability decoding of a binary BCH code on BSC. The source distribution is $\mu = P(1) = 0.1$. The initial values for EM recursion were $\mu_0 = 0.5$ and $\lambda_0(BSC) = 0.1$. The BCH code is a shortened $(n = 30, k = 20, t = 2)_2$. Estimation is based on frames of length 3000 bits (100 BCH codewords).

4.3 An example of EM estimation on Gaussian channel

The channel input-output relation is given by $y_j = Ae^{\jmath\phi}x_j + \eta_j$, where $\jmath = \sqrt{-1}$, the amplitude A is real positive, ϕ is uniformly distributed between 0 and 2π, and η_j is a zero mean complex gaussian noise with variance σ^2 per component. The symbol x_j belongs to a QAM constellation and time index is denoted by j, $j = 1 \ldots N$. The source in this example is uniform (SSI not available).

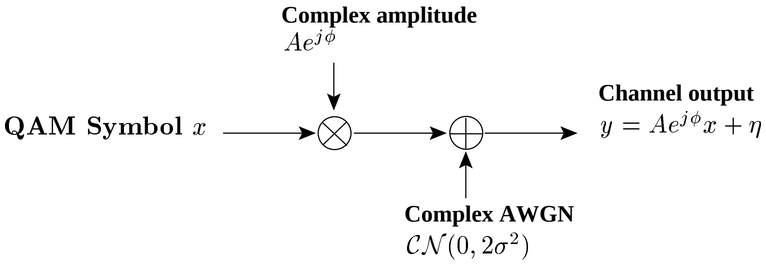


Figure 25: EM estimation on a gaussian channel. The source is assumed to be uniform. The parameter to be estimated includes real amplitude, signal phase, and noise variance.

By observing the incomplete data $\mathbf{y}$, we wish to estimate $\theta = (A, \phi, \sigma^2)$. The EM auxiliary function is $Q(\theta|\theta^i) = E_{\mathbf{x}}[\log p(\mathbf{y}|\mathbf{x}, \theta)|\mathbf{y}, \theta^i]$, where the joint distribution is

$$p(\mathbf{y}|\mathbf{x}, \theta) = \frac{1}{(2\pi\sigma^2)^N} exp\left(-\frac{\sum_{j=1}^N |y_j - Ae^{\jmath\phi}x_j|^2}{2\sigma^2}\right) \tag{58}$$

The next step is straightforward. Use the important notice displayed in the BSC section above for computing soft symbols, i.e. expectation with marginals instead of joint distribution, and write $\partial Q/\partial\theta = 0$ to get the following EM parameter recursions,
Phase update:

$$\phi^{i+1} = -Arg\sum_{j=1}^N \widetilde{x}_j y_j^* \tag{59}$$

Amplitude update:

$$A^{i+1} = \frac{\sum_{j=1}^N \Re\{\widetilde{x_j y_j^*}\}}{\sum_{j=1}^N \sum_{x_j} APP_i(x_j)\ |x_j|^2} = \frac{\sum_{j=1}^N \Re\{\tilde{x}_j y_j^*\}}{\sum_{j=1}^N \widetilde{|x_j|^2}} \tag{60}$$

Noise variance update:

$$\sigma^{2\ (i+1)} = \frac{1}{2N}\left(\sum_{j=1}^N \sum_{x_j} APP_i(x_j)\ |y_j - A^i e^{\jmath\phi^i} x_j|^2\right) = \frac{1}{2N}\sum_{j=1}^N \widetilde{|y_j - A^i e^{\jmath\phi^i} x_j|^2} \tag{61}$$

Computer simulation results for EM with a parallel turbo code are shown in Fig. 26 (Bit error rate versus signal-to-noise ratio) and Fig. 27 (Word error rate versus signal-to-noise ratio).

4.4 Different methods for combining pilots and data in EM

To be written later in version 3 of this tutorial.

4.5 On the mean squared error of estimators: The Cramér-Rao bound

To be written in version 3 of this tutorial.

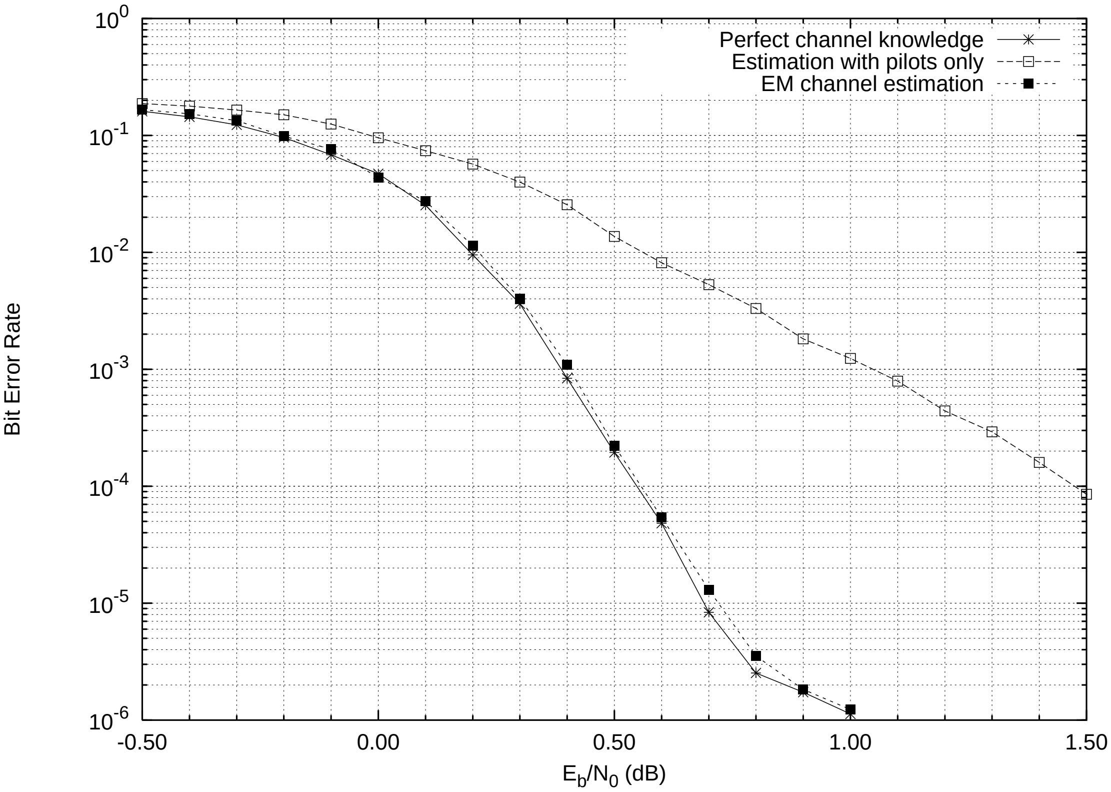


Figure 26: EM estimation on a gaussian channel. Bit error for a rate 1/3 parallel turbo code, interleaver size is 1600, RSC octal generators are (37,21), optimal bi-dimensional 40×40 quasi-cyclic interleaver. Measure of 500 bit errors and 100 block errors. Number of decoding iterations is up to 40. Channel amplitude uniformly distributed between 0.1 and 10. Channel phase uniformly distributed between 0 and 2π. Number of pilots is 50 bits (1% pilots).

5 Conclusions

To be written in version 3 of this tutorial.

6 Bibliography

Add references per topic.
Topics are: EM algorithm, Forward-Backward, Turbo and LDPC codes, CDMA, MIMO, etc.
Cite also new books like Lin & Costello 2004 and David MacKay 2003.

7 Acknowledgments

The author wishes to thank Professor Ezio Biglieri (Politecnico di Torino, Italy) and Professor Marc Moeneclaey (Ghent University, Belgium) who initiated the motivation for writing the tutorial on iterative probabilistic decoding and EM channel estimation respectively. Also, the help of Doctor Nicolas Gresset (ENST, France) was precious while writing this document.

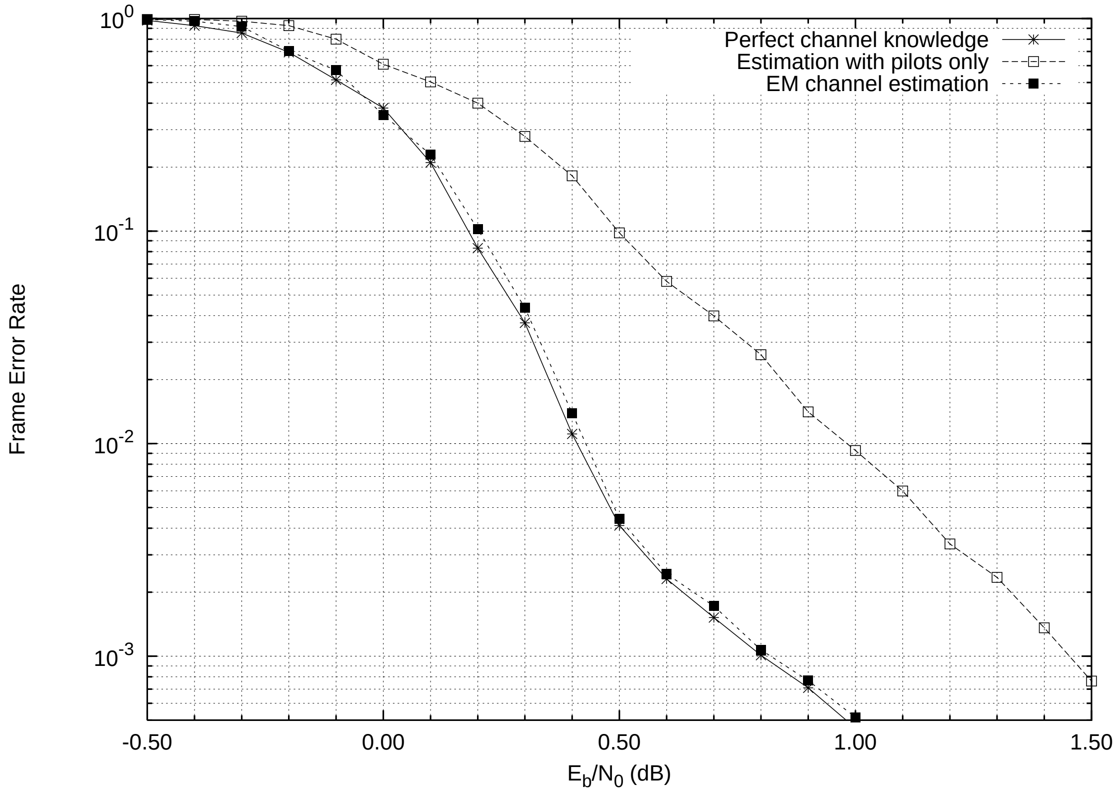


Figure 27: EM estimation on a gaussian channel. Frame error for a rate 1/3 parallel turbo code, interleaver size is 1600, RSC octal generators are (37,21), optimal bi-dimensional 40×40 quasi-cyclic interleaver. Measure of 500 bit errors and 100 block errors. Number of decoding iterations is up to 40. Channel amplitude uniformly distributed between 0.1 and 10. Channel phase uniformly distributed between 0 and 2π. Number of pilots is 50 bits (1% pilots).