Iterative Receivers for Coded MIMO Signaling

Ezio Biglieri  Alessandro Nordio  Giorgio Taricco

Abstract—In this tutorial paper we describe iterative receivers that combine a soft decoder for a general space–time code with a spatial-interference canceler. The performance of these receivers is analyzed by using EXIT charts, a convenient graphical description that yields quite accurate results. By properly combining the EXIT characteristics of the canceler and of the decoder, the convergence behavior of the iterative algorithms can be understood, and design guidelines derived. The idea here is to observe that both the interference canceler and the decoder can be studied by examining the evolution of the extrinsic information passed along to the connected blocks.

I. INTRODUCTION AND MOTIVATION OF THE WORK

Recently, multiple-antenna (MIMO) techniques have been recognized to be capable of greatly increasing the spectral efficiency of wireless systems. For this reason, a considerable research effort is being spent to design space–time codes that approach the impressive values of channel capacity available. Additional work is directed towards the reduction of the complexity of optimum decoding: in fact, maximum-likelihood receivers exhibit a complexity that grows exponentially with the modulation size and the number of antennas, and hence become quickly unpractical as either parameter is large. Thus, in addition to searching for good space–time codes, it is important to seek receivers that achieve a close-to-optimum performance while keeping a moderate complexity: this would remove the practical restriction to small signal constellations or few antennas.

Suboptimal receivers may include linear filters, cancelers of spatial interference, or sphere decoders. In addition, iterative receivers have received a special attention of late in several contexts: see, e.g., [3]–[7], [9], [10], [13]–[16], [19], [20], [22]–[25]. In one of its possible settings, an iterative receiver combines a soft spatial-interference canceler with a soft-input, soft-output (SISO) decoder, as represented schematically in Fig. 1. Before sending its soft decisions to the hard decoder, the SISO decoder iteratively feeds “extrinsic” information (to be properly defined, which we will do in the following) back to the soft canceler. In [2], the combination of a soft canceler with turbo space–time codes was shown to provide a good tradeoff between complexity and performance. There, the received signals are first combined through a linear Minimum Mean Square Error (MMSE) filter, then spatial interference is reduced by feeding back soft decisions provided by the decoder. If turbo codes are used, even the SISO decoder is iterative. This makes the overall receiver doubly iterative, in the sense that preliminary results obtained from a few iterations of the turbo decoding algorithm are used to reduce spatial interference. After this reduction, further turbo-decoding iterations are performed in order to improve on the interference cancellation, and so on.

In this paper, we elaborate in a tutorial fashion on the concept of iterative receivers that combine a soft decoder for a general space–time code with a spatial-interference canceler. The performance of these receivers is analyzed by using EXIT charts [17]. By properly combining the EXIT characteristics of the canceler and of the decoder, convergence of the iterative algorithms can be studied, and design guidelines derived. The idea here is to observe that, for the interference canceler as well as for the decoder, their behavior can be studied by examining how they transform the extrinsic information passed along to the connected blocks. One parameter describing this extrinsic information is, as suggested in [17], mutual information. Thus, by combining in a single chart the input-output characteristics of two blocks, the convergence of a turbo-like algorithm can be given a convenient graphical description, which, although not exact, yields quite accurate results.

This paper is organized as follows. Section II is devoted to the definition of the main concepts and quantities used throughout, from soft-input, soft-output (SISO) processors to extrinsic probabilities and EXIT charts. Section III and IV describe how EXIT charts can be computed for SISO decoders and for other SISO processors, respectively. Section V shows how SISO decoders and interference cancelers can be combined in an iterative receiver, with its performance...
evaluated through EXIT charts. Conclusions are drawn in Section VI.

II. DEFINITIONS

A. Soft and hard decisions

Consider transmission of the $n$-tuple $x = (x_1, \ldots, x_n)$ of symbols chosen from an alphabet $\mathcal{X}$. At the output of the transmission channel a vector $y$ is observed. Following [8, p. 124 ff.], we call a soft decision for $x_i$ the “a posteriori” probability distribution of $x_i$ given $y$, that is, $p(x_i|y)$. Since $p(x_i, y) = p(x_i|y)p(y)$, and $p(y)$ is irrelevant to the decision process, one may also call soft decision the probability distribution $p(x_i, y)$, with $y$ interpreted as a parameter. A hard decision for $x_i$ is a probability distribution such that $p(x_i|y)$ is equal either to 0 or to 1.

B. Receivers and interfaces

A receiver is a system accepting as its input the channel observation $y$, and generating a hard decision on each transmitted $x_i$ based on a suitable decision rule (typically, the minimization of an error probability). An interface accepts $y$ as its input, and outputs a soft decision on each $x_i$. An interface is generally a combination of devices, called soft-input soft-output processors, that generate soft decisions (for a detailed definition, see infra, Section II.E). Eventually a SISO output is passed to the hard decoder. This accepts soft decisions at its input and outputs hard decisions: for example, if $\mathcal{X} = \{\pm1\}$, it chooses $p(x_i = +1|y) = 1$ whenever $p(x_i = +1, y) \geq p(x_i = -1, y)$. The goal of the interface, and hence of the SISO processors forming it, is to process the received data so as to obtain soft decisions as close as possible to correct hard decisions before final decoding.

C. Extrinsic probabilities

“Turbo” processing hinges on the exchange of extrinsic information. To define properly the latter quantity, we use the concept of factor graph (see [12] and references therein). This represents in graphical form the factorization of a function $f(x_1, \ldots, x_n)$ of several variables. The “sum-product” algorithm allows one to compute (exactly and in a finite number of steps) the marginals of the function with respect to each variable. These are defined as the functions

$$f_i(x_i) \triangleq \sum_{x_{\neg i}} f(x_1, \ldots, x_n)$$

obtained by summing $f(x_1, \ldots, x_n)$ over all its arguments consistent with the value of $x_i$. The compact notation $x_{\neg i}$ denotes the vector $(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$ to be summed over. The sum-product algorithm computes, for every edge of the factor graph (which corresponds to one variable), two “messages,” one per each direction, whose product yields the marginal of the function with respect to that variable. It is often convenient to think of these messages as probability distributions, which is obtained by properly normalizing them. For example, with binary variables a message can be thought of as a pair of real numbers summing to 1.

A key feature of the sum-product algorithm is that a message sent along one direction does not depend on the message sent along the other one. Thus, if one of the messages derives directly from the a priori knowledge of the edge variable, or from its measurement at the channel output, the other message depends only on the remaining variables: for this reason the information it carries is called extrinsic. Two examples illustrate this concept.

Example 1: Soft decoding

Consider a code $C$ with $|C|$ words $x = (x_1, \ldots, x_n)$ and assume that the a priori code word probabilities are equal. These are transmitted over a stationary memoryless channel such that the observed $n$-vector $y$ is such that

$$p(y|x) = \prod_{i=1}^{n} p(y_i|x_i)$$

Soft decoding of $C$ consists of computing the probabilities

$$p(x_i, y) = \sum_{x_{\neg i}} p(x, y) = \sum_{x_{\neg i}} p(x)p(y|x) = \sum_{x_{\neg i}} |C|^{-1} \sum_{x \in C} \prod_{j=1}^{n} p(y_j|x_j)$$

where the “Iverson function” $[x \in C]$ takes on value 1 if vector $x$ is a code word, and 0 otherwise. The last equation shows the factorization of the function whose marginals yield the probabilities $p(x_i, y)$. The corresponding factor graph is shown in Fig. 2 (a). Application of the sum-product algorithm yields for each edge the messages shown in Fig. 2 (b).

The upward messages are the probabilities $p(y_i|x_i)$, corresponding to channel observations (these are to be interpreted as functions of $x_i$ with $y_i$ as parameters). The downward messages $e(x_i)$ are the extrinsic probabilities. Since, after proper message normalization, the product $e(x_i)p(y_i|x_i)$ yields $p(x, y)$, we define the extrinsic probabilities as the ratios between $p(x, y)$ and $p(y_i|x_i)$ (suitably normalized):

$$e(x_i) = \frac{p(x_i, y)p(y_i|x_i)}{\sum_{\bar{x}_i \in \mathcal{X}} p(\bar{x}_i, y)p(y_i|\bar{x}_i)}$$

Fig. 2. (a) Factor graph for soft decoding. (b) Messages passed along each edge by the sum-product algorithm.
Notice that, from
\[
p(x_i, y_{-i}) = \sum_{x_i} |C|^{-1} p(y_{-i}|x) = \sum_{x_i} |C|^{-1} p(y_{-i}|x) \prod_{j \neq i} p(y_j|x_j)
\]
we obtain
\[
e(x_i) = p(x_i|y_{-i})
\] (4)

In other words, the extrinsic probability can also be interpreted as the probability of the \(i\)th code word symbol conditioned on all other channel observations, namely, \(y_{-i}\), through the intermediary of the code structure. For a simple example [7], examine the single-parity-check binary code with length 3, whose symbols are \(x_1, x_2\), and \(x_3 = x_1 + x_2\). Information about \(x_1\) can be gathered from the observation of \(y_1\), and also from the separate observation of \(y_2\) and \(y_3\), supplemented by the knowledge of the code structure, i.e., the fact that \(x_3 = x_1 + x_2\). Thus, the latter observation generates the extrinsic information.

\begin{eqnarray*}
\text{--- Example 2: Soft interference cancellation ---}
\end{eqnarray*}

Consider the transmission of \(n\) independent binary symbols \(x = (x_1, \ldots, x_n)\) on a common channel, and the observation of a noisy vector \(y\) whose components are known functions of all symbols (for example, linear combinations with known coefficients). The channel is described by the function \(p(y|x)\).

Soft estimation of \(x_i, i = 1, \ldots, n\), consists of computing
\[
p(x_i, y) = \sum_{x_i} p(y|x) = \sum_{x_i} p(y|x) \prod_{j=1}^n p(x_j)
\] (5)

which is tantamount to marginalizing the function \(p(y|x)p(x_1) \cdots p(x_n)\). The corresponding factor graph is shown in Fig. 3, along with the messages exchanged by the blocks in the application of the sum-product algorithm. Since, after proper message normalization, the product \(e(x_i)p(x_i)\) yields \(p(x_i, y)\), the extrinsic probability is defined as
\[
e(x_i) = \frac{p(y|x_i)}{\sum_{x_i \in C} p(y|x_i)} = \frac{\sum_{x_i \in C} p(y|x)p(x_{-i})}{\sum_{x} p(y|x)p(x_{-i})}.
\] (6)

\begin{itemize}
\item \textbf{D. The turbo algorithm}
\end{itemize}

This consists of coupling SISO processors in such a way that the extrinsic probability output of one processor is fed to the input of another. Consider, in particular, soft interference cancellation of an \(n\)-tuple of coded symbols. The factor graphs of Fig. 2 and 3 can be joined so as to share the edges labeled \(x_1, \ldots, x_n\). The resulting factor graph is shown in Fig. 4.

If it exhibits cycles, then the sum-product algorithm does not generate the a posteriori probabilities, and an iterative (“turbo”) algorithm must be used instead to obtain their approximate values [12]. This algorithm computes repeatedly the two-way messages associated with the edges of the graph, until a termination criterion stops the iterative process. A possible schedule is illustrated in Fig. 4(b): first, \(y\) is observed and the extrinsic probabilities \(\hat{e}(x_i), i = 1, \ldots, n\), are computed and passed along to the code block. This computes \(e(x_i), i = 1, \ldots, n\), by using \(\hat{e}(x_i)\) as if they were the channel observations \(p(y_i|x_i)\) in the algorithm of Fig. 2. Next, the lower block uses \(e(x_i), i = 1, \ldots, n\), as if they were the a priori probabilities \(p(x_i)\) in the algorithm of Fig. 3(b), and so forth.

\begin{itemize}
\item \textbf{E. SISO processors}
\end{itemize}

For proper definition of the turbo algorithm, it is convenient to describe the SISO processors as two-input, two-output devices as shown in Fig. 5. A SISO processor accepts two sets of inputs:

1. \textit{Channel observations}, i.e., the conditional probability distribution \(p(y|x)\), depending on the knowledge of the channel statistics and on the observation of \(y\), and
2. \textit{A priori probabilities}, i.e., the marginal probabilities \(p(x_i)\).

It outputs:

\begin{itemize}
\item \textbf{~}
\end{itemize}
Soft decisions, i.e., the a posteriori probabilities \( p(x_i|y) \), which will eventually be sent to the decoder generating hard decisions.

2. Extrinsic probabilities \( e(x_i) \).

A common, convenient choice is the mutual information

\[
I(\sigma^2) = 0.5 \left[ p(\Lambda|x = -1) + p(\Lambda|x = 1) \right].
\]

If condition (7) is satisfied, then \( \Lambda|x \sim N(x \sigma^2/2, \sigma^2) \), and hence \( I(x; \Lambda) \) depends only on \( \sigma^2 \). We have, explicitly,

\[
I(x; \Lambda) = 1 - \frac{1}{2} \sum_{x \in \{\pm 1\}} \int_{-\infty}^{\infty} p(\Lambda|x) \log_2 \left[ 1 + \frac{p(\Lambda|x) - x}{p(\Lambda|x)} \right] d\Lambda
\]

The behavior of \( J(\sigma^2) \), which can be examined by numerical evaluation of (9), is shown in Fig. 6.

F. EXIT charts

Since the turbo algorithm operates on extrinsic probabilities, its convergence behavior can be studied by examining how these evolve in time. A convenient graphical description of this process is given by EXIT charts [17], which yield quite accurate, albeit not exact, results. An EXIT chart is a graph that illustrates the input-output relation of a SISO processor by showing the transformations induced on a single parameter associated with input and output extrinsic probabilities. Let us focus for simplicity on a binary alphabet \( \mathcal{X} = \{\pm 1\} \). The rationale behind EXIT charts stems from the observation that the logarithmic likelihood ratio (LLR)

\[
\Lambda(x) \triangleq \log \frac{e(x = +1)}{e(x = -1)}
\]

is well approximated by a conditionally normal random variable (we write \( \Lambda|x \sim N(\mu, \sigma^2) \)) whose probability density function (pdf) \( p(\Lambda|x) \) satisfies the “consistency condition”\(^2\),

\[
|\mu| = \frac{\sigma^2}{2} \tag{7}
\]

where \( \mu \) and \( \sigma^2 \) denote conditional mean and variance, respectively. Hence, under this condition, a single parameter (e.g., \( \sigma^2 \)) completely defines \( p(\Lambda|x) \).

Corresponding probability distributions: these are in fact estimated from random observations.

EXIT charts describe the evolution of \( p(\Lambda|x) \) by showing the evolution of one parameter derived from it. There are several possible choices for this parameter (a thorough discussion and a comparison can be found in [21]). A common, convenient choice is the mutual information \( I(x; \Lambda) \) between \( x \) and \( \Lambda \), defined as\(^3\)

\[
I(x; \Lambda) = \frac{1}{2} \sum_{x \in \{\pm 1\}} \int_{-\infty}^{\infty} p(\Lambda|x) \log_2 \frac{p(\Lambda|x)}{p(\Lambda)} \, d\Lambda \tag{8}
\]

with \( p(\Lambda) = 0.5[p(\Lambda|x = -1) + p(\Lambda|x = +1)] \).

\(^1\)Here we drop the subscript \( i \) to simplify notation.

\(^2\)This condition has been first derived in [17] and is a straightforward consequence of the fact that the noise is Gaussian distributed.

\(^3\)The notation here is not the most felicitous one, as it does not distinguish between the random variable \( x \) and the values it takes on. We put up with it, as it is commonly used in the literature.

---

Fig. 5. Basic block diagram of a SISO processor.

Fig. 6. Plot of the function \( J(\sigma^2) \) defined in eq. (9).
relations can be factored into the product \( \prod \)

It follows that we can define the a priori, channel observation, decision, and extrinsic mutual informations as \( I^a \triangleq I(x; \mu^a) \), \( I^o \triangleq I(x; \mu^o) \), \( I^d \triangleq I(x; \mu^d) \), and \( I^e \triangleq I(x; \mu^e) \), respectively.

We are now ready to describe a SISO processor by giving its extrinsic information transfer (EXIT) function

\[
I^e = T(I^a, I^o)
\]

Several examples of EXIT functions can be found in the literature (see, e.g., [9], [17], [19]), all obtained by Monte Carlo simulation. The general algorithm used to derive the values of \( I^e \) from those of \( I^a, I^o \), and hence the EXIT function \( T \), can be outlined as follows (in the next two Sections it will be specialized to SISO decoders and other processors):

1. Generate a sample input vector \( x \) with random entries in \( \{\pm 1\} \).
2. Generate the SISO-processor input message \( \mu^a(x) \) satisfying the constraint

\[
I(x; \mu^a(x)) = I^a
\]

3. Generate the SISO-processor input message \( \mu^o(x) \) satisfying the constraint

\[
I(x; \mu^o(x)) = I^o
\]

(In this step, the output sample vector \( y \) is generated according to the pdf \( p(y|x) \).)
4. Operate the SISO processor to obtain the extrinsic probabilities \( e(x_i) \) at its output.
5. Estimate \( I^e \) by using the approximation (10).

Notice that the EXIT-chart analysis is approximate, as it is based on the assumption of independent extrinsic probabilities, which holds for an infinite-length interleaver. Thus, some inaccuracies must be expected [11], [17], [19]. Nevertheless, the practical usefulness of EXIT charts for convergence predictions is unquestioned.

III. EXIT CHARTS OF SISO DECODERS

In this Section we specialize to SISO decoders the algorithm for the derivation of EXIT functions. Under the assumption of a stationary memoryless channel with perfect channel state information (CSI) at the receiver, the conditional pdf \( p(y|x) \) can be factored into the product \( \prod_i p(y_i|x_i) \), and we have the relations

\[
I^o = J(\sigma_o^2) \quad I^a = J(\sigma_a^2)
\]

deriving from (9). Here, \( \sigma_o^2 \) is the variance of the additive noise.

The block diagram of the system used to compute the mutual information transfer function is depicted in Fig. 7.

A random vector \( b \in \{\pm 1\}^k \) of uncoded symbols, \( k \leq n \), is generated, and passed to the encoder. This outputs the code word \( x \in \{\pm 1\}^n \). Vector \( x \) is then passed to a random generator (labelled \( \sigma_b^2 \)) which outputs an LLR vector whose entries \( \Lambda_b^i \) satisfy

\[
\Lambda_b^i|b_i \sim \mathcal{N}(b_i, \frac{\sigma_b^2}{2}), \sigma_b^2
\]

for \( i = 1, \ldots, n \). Similarly, the source vector \( b \) is passed to a random generator (labelled \( \sigma_s^2 \)) which outputs an LLR vector whose entries \( \Lambda_s^i \) satisfy

\[
\Lambda_s^i|x_i \sim \mathcal{N}(x_i, \frac{\sigma_s^2}{2}), \sigma_s^2
\]

\( i = 1, \ldots, k \). The variances \( \sigma_b^2 \) and \( \sigma_s^2 \) are equal to \( J^{-1}(I^a) \) and \( J^{-1}(I^o) \), respectively. After operating the SISO decoder on inputs \( \Lambda_s^i \) and \( \Lambda_b^i \), the mutual information at the output of the SISO decoder is computed by applying (10) to \( \Lambda^e \).

Notice that the SISO decoder can output extrinsic information on both uncoded and coded bits, which in terms of log-likelihood ratios can be denoted by \( \Lambda^c_u \) and \( \Lambda^c_c \). Hence, the mutual informations \( I^c_u \) and \( I^c_e \) can be evaluated.

The choice of dealing with \( I^c_u \) or \( I^c_e \) depends on the application considered. Reference [17], analyzing the transfer of information between the constituent decoders of parallel concatenated codes, uses the mutual informations \( I^c_u \), because the constituent decoders share the extrinsic probabilities of uncoded bits. In contrast, refs. [9], [19], investigating turbo equalization and MIMO iterative receivers, use the mutual informations \( I^c_e \), since the SISO equalizer and the SISO decoder share the extrinsic probabilities of coded bits. As we deal with coded MIMO systems, here we use the mutual information on coded bits, \( I^c_e \), hereafter denoted only by \( I^e \). In the following, to enhance the simulation efficiency, we implement the SISO decoder by using the log-MAP BCJR algorithm [1].

Figs. 8–10 show some examples of EXIT charts relevant to some recursive systematic convolutional (RSC) codes and turbo-codes.

Fig. 8 refers to a rate-1/2 RSC code with octal generators (5, 7). The curves plot the mutual information \( I^o \) against \( I^a \) using \( I^a \) as parameter.

Fig. 9 refers to several rate-1/3, rate-1/2, and rate-2/3 RSC codes with different generators and number of states. Rate-2/3 codes are obtained by puncturing corresponding rate-1/2 codes. The curves plot the mutual information \( I^c \) against \( I^o \) assuming \( I^a = 0 \).

Fig. 10 refers to a rate-1/2 parallel turbo-code whose constituent RSC encoders have generators (5, 7). The curves plot the mutual information \( I^c \) against \( I^o \) assuming \( I^a = 0 \).

\footnote{We omit again the subscript \( i \) here, for simplicity’s sake.}
Ia unit step function whose level transition occurs at a value which in turn yields \( \Lambda \) where
\[
\sigma \approx 0.2 \text{ and } I_o > \rho,
\]

A notable common feature of these EXIT charts is that, for \( I^a = 0 \), they can be regarded as smoother versions of a unit step function whose level transition occurs at a value of \( I^o \) equal to the code rate \( \rho \). This can be interpreted by observing that, when \( I^a = 0 \), \( I^o \) is equivalent to the mutual information exchanged between the transmitted symbol \( x \) and the received signal \( y \), and hence equals the capacity. A capacity-achieving code can attain reliable communication if and only if \( I^o > \rho \), and hence its EXIT curve would exhibit a sharp transition of the extrinsic mutual information from 0 (unreliable communication) to 1 (reliable communication) in correspondence of \( I^o = \rho \). Finite-complexity codes exhibit the smoother behavior exhibited by the EXIT curves.

Notice also how the transition near \( \rho \), which is symmetric for convolutional decoders, becomes asymmetric for turbo decoders. These also show a migration from the “unreliable communication” condition slower than convolutional codes of similar rate, but, as the number of iteration increases, a faster acquisition of the “reliable communication” condition.

A. Error probabilities on EXIT charts

Estimates of the error probability of a coded system can be superimposed to EXIT charts to yield insight on the receiver performance. By assuming the random conditional LLR \( \Lambda^d|x \) to be Gaussian distributed with mean \( \sigma^2_a/2 \) and variance \( \sigma^2_o \), the bit error probability (BER) can be approximated by
\[
P_b(e) \approx Q\left(\frac{\mu_a}{\sigma_a}\right) = Q\left(\frac{\sigma_d}{2}\right)
\]

where \( Q(\cdot) \) is the Gaussian tail function. Since \( \Lambda^d = \Lambda^o + \Lambda^a + \Lambda^e \), the assumption of independent LLR’s leads to: [17]
\[
\sigma^2_d = \sigma^2_e + \sigma^2_a + \sigma^2_o
\]

which in turn yields
\[
P_b \approx Q\left(\frac{\sqrt{J^{-1}(I^o) + J^{-1}(I^a) + J^{-1}(I^o)}}{2}\right)
\]

IV. EXIT CHARTS OF OTHER SISO PROCESSORS

Let us consider a multiple-input multiple-output (MIMO) system equipped with \( t \) transmit and \( r \) receive antennas. The received signal can be modelled by the following equation
\[
Y = HS + Z
\]

where \( S \) is a \( t \times L \) matrix (space–time code word) of symbols belonging to the constellation \( S \) with size \( |S| = 2^m \), \( H \) is a \( r \times t \) channel matrix, and \( Z \) is a matrix of iid complex Gaussian noise samples with zero mean and variance \( \sigma^2_z \). Denoting by \( \ell \) the time index \( (\ell = 1, \ldots, L) \) and by \( Y_\ell, S_\ell \), and \( Z_\ell \) the \( \ell \)-th columns of \( Y, S, \) and \( Z \), respectively, we can rewrite eq. (14) as
\[
y_\ell = Hs_\ell + z_\ell = \sum_{i=1}^{t} h_i s_{\ell,i} + z_\ell
\]

where \( h_i \) is the \( i \)-th column of \( H \). In order to simplify notation, we shall drop the time index \( \ell \) in the following. Moreover,
we define the input binary $mt$-vector as $\mathbf{x} \triangleq (x_1^T, \ldots, x_t^T)$, where $x_i \triangleq (x_{i1}, \ldots, x_{im})$ and $x_{ij} \in \{\pm1\}$. The binary vector $\mathbf{x}$ is mapped to symbol vector $\mathbf{s}$. The EXIT chart of the SISO receiver is evaluated according to the block diagram of Fig. 12. Here, vector $\mathbf{x}$ is first generated, then passed through the modulator to yield vector

$$\mathbf{s} = \varphi_m(\mathbf{x}) = (\varphi_m(x_1), \ldots, \varphi_m(x_1)),$$

which is passed through the channel to obtain the received vector $\mathbf{y}$ providing, in turn, the message $\mu^o$ consisting of

$$p(y|x) = (\pi\sigma_x^2)^{-t} \exp(-\|y - H\varphi_m(x)\|^2/\sigma_x^2)$$

(16)
sampled at all possible values of $x \in \{\pm1\}^m$. The other input messages are obtained, as LLRs, according to

$$\Lambda^o_{ij} | x_{ij} \sim \mathcal{N} \left( x_{ij} \frac{\sigma_x^2}{2} \frac{\sigma^2}{\sigma^2} \right)$$

with $\sigma_x^2 = J^{-1}(I^o)$. The evaluation of the extrinsic probability distribution (message $\mu^o$) depends on the type of SISO processor considered. In the following we describe three types of SISO processors.

### A. MAP equalizer

When a maximum a posteriori (MAP) equalizer is employed, and for a fixed given matrix $H$, following the considerations of Example 2 we obtain the SISO extrinsic output as

$$e(x_{ij}) = \frac{p(x_{ij})^{-1} \sum_{x_{-ij}} p(y|x)p(x)}{\sum_{x_{ij}} p(y|x)p(x_{-ij})}$$

(17)

where $p(y|x)$ is as in (16). Notice that, in this case, $p(y|x) \neq \prod p(y_i|x_i)$, so that direct evaluation of $I^o$ is difficult. Thus, we express the mutual information $I^o$ as $T(I^a, \sigma_x^2)$ instead of $T(I^a, I^o)$, and apply again the approximation (10) to the samples $\Lambda^o_{ij}$ derived from (17).

The computational complexity of evaluating (17) (exponential in the product $mt$) has led researchers to devise suboptimal, low-complexity SISO processors based on soft interference cancellation. These processors are based on the combination of a linear filter and an interference canceler (IC).

### B. Interference cancelers with linear filtering

Interference cancellation is based on the generation of soft estimates $\hat{s}$ of the transmitted symbol vector $s$ that are used to eliminate, in an iterative fashion, the spatial interference. For each transmit antenna, $i = 1, \ldots, t$, the soft estimates are computed as follows:

$$\hat{s}_i = \sum_{s_i \in S} s_i p(s_i)$$

(18)

where, assuming that the bits contributing to the transmission of $s$ are independent, $p(s_i) = p(x_i) = \prod_{j=1}^m p(x_{ij})$ if $s_i = \varphi_m(x_i)$.

### Example: Binary PAM

As a special case of interest, let us consider binary PAM with $S = \{\pm1\}$ and the identity map $s = x$. Since $\Lambda = \log[p(x = +1)/p(x = -1)]$, we have

$$\hat{s} = (-1)^i \frac{1}{1 + e^\Lambda} + (1)^i \frac{e^\Lambda}{1 + e^\Lambda} = \tanh \frac{\Lambda}{2}$$

Assuming $\Lambda \sim \mathcal{N}(\sigma^2/2, \sigma^2)$, the following pdf of $\hat{s}$ is obtained:

$$p(\hat{s}) = \frac{2}{1 - \tilde{s}^2} \frac{1}{2\pi\sigma} \exp \left( -\frac{(2\arctanh \tilde{s} - \sigma^2/2)^2}{2\sigma^2} \right)$$

(19)

Then, the IC block outputs, for each antenna $i$, the following soft values

$$\hat{y}_i = y - H\hat{s} + h_i \hat{s}_i$$

$$= h_i s_i + \sum_{j \neq i} h_j (s_j - \hat{s}_i) + z$$

which are subsequently processed by the antenna-specific linear filters as described in the following.

$^5$ $x_{-ij}$ denotes the vector $x$ without the entry $x_{ij}$. 

Fig. 11. BER chart of an iterative receiver plotted as a function of $I^o$ and $I^a$ and considering $I^o = 0$. 

Fig. 12. Block diagram for the derivation of the EXIT chart of a SISO canceler.
processors as it achieves a better value of $\Lambda$.

\[ \text{transmit antenna per symbol and per information bit, respectively.} \]

1) **MMSE filter:** The MMSE filter operates so as to minimize the mean square error (MSE) $\mathbb{E}[|f_i^\dagger \tilde{y}_i - x_i|^2]$. As a result, the filter vector $f_i$ is obtained as

\[ f_i = \left[ \sigma_i^2 I_T + \mathbf{H} \Sigma_i^2 \mathbf{H}^\dagger \right]^{-1} \mathbf{h}_i \]  

where $\Sigma_i^2 = \text{diag}(\sigma_i^2, \ldots, \sigma_{i-1}^2, 1, \sigma_{i+1}^2, \ldots, \sigma_T^2)$ and the variances $\sigma_i^2$ are given by

\[ \sigma_i^2 = \mathbb{E}[|s_i - \tilde{s}_i|^2] = \sum_{s_i \in S} |s_i|^2 p(s_i) - |\tilde{s}_i|^2 \]  

Recalling eq. (19), the output of the $i$th filter is given by

\[ \tilde{y}_i \triangleq f_i^\dagger \tilde{y}_i = \mu_i c_i + \beta_i \]  

where $\mu_i = f_i^\dagger \mathbf{h}_i$ and where $\beta_i$ is a complex Gaussian random variable with zero mean and variance $\sigma_{\beta_i}^2 = \mu_i - \mu_i^2$.

Extrinsic probabilities are finally computed as follows:

\[ e(x_{ij}) = \sum_{x_{i\neq j}} p(\tilde{y}_i|x_i) \prod_{j' \neq j} p(x_{ij'}) \]  

The computational complexity involved is linear in $t$ and exponential in $m$, the number of bits per symbol.

2) **Maximum Ratio Combining filter:** The Maximum Ratio Combining (MRC) filter is based on the filter vector $f_i = \mathbf{h}_i$. Again, the filter output can be written as in (22) where $\mu_i = \mathbf{h}_i^\dagger \mathbf{h}_i$ and

\[ \sigma_{\beta_i}^2 = \sum_{j \neq i} |\mathbf{h}_i^\dagger \mathbf{h}_j|^2 \beta_j^2 + \sigma_z^2 \]  

Figure 14 shows the EXIT chart of the MAP, IC+MMSE, and IC+MRC SISO processors considered here. In this case we assume $r = t = 4$ (four transmit and receive antennas), a complex channel matrix $\mathbf{H}$ as in [9], a QPSK signal set, and $1/\sigma_z^2 = E_s/N_0 = 1$ dB (solid lines) or $-2$ dB (dashed lines). The curves show that the MAP equalizer outperforms all other processors as it achieves a better value of $I^e$ at any given $I^d$.

\[ (\text{Here and in the following, } E_s \text{ and } E_x \text{ denote the average energy per transmit antenna per symbol and per information bit, respectively.}) \]

V. APPLICATIONS: ITERATIVE MIMO RECEIVERS

A. Deterministic channel

In a turbo device two SISO processors are connected together so that the extrinsic output of each one is connected to the other’s a priori input. Usually, interleavers are inserted in order to reduce the message correlation.

The SISO processors may be both MAP decoders, which results into a turbo decoder. In this case, the convergence of the turbo decoder has been extensively studied by using EXIT charts [17].

Nevertheless, EXIT charts apply with any pair of SISO processors and portray the behavior of the turbo device by showing the transfer functions of the mutual informations involved. Let us focus on the interface illustrated in Fig. 15. The abscissa of the EXIT chart is the input mutual information $I_{\text{can}}^a$ of the SISO canceler coinciding with the extrinsic output mutual information $I_{\text{can}}^e$ of the SISO decoder ($I_{\text{can}}^a = I_{\text{can}}^e$). The ordinate is the extrinsic output mutual information $I_{\text{dec}}^e$ of the SISO canceler coinciding with the channel observation input mutual information $I_{\text{dec}}^a$ of the SISO decoder ($I_{\text{can}}^a = I_{\text{can}}^e$). Thus, an EXIT chart contains the transfer functions $I_{\text{can}}^a = T_{\text{can}}^a(I_{\text{can}}^a, I_{\text{can}}^e)$ plotted with $I_{\text{can}}^a$ on the absissa and $I_{\text{can}}^e$ on the ordinate, and $I_{\text{dec}}^e = T_{\text{dec}}^e(I_{\text{dec}}^a, I_{\text{dec}}^e)$ plotted with $I_{\text{dec}}^a$ on the ordinate and $I_{\text{dec}}^e$ on the abscissa. The iterations successfully converge if the equilibrium point $I_{\text{can}}^a = I_{\text{can}}^e = 1$ is reached.

![Fig. 15. Block diagram of a turbo interference canceler.](image-url)
receive antennas and the extrinsic probabilities output by the decoder, while its outputs are the extrinsic probabilities. These are sent to the input of the SISO decoder corresponding to the channel observations. No \textit{a priori} information is available to the SISO decoder input, so that \( P_{a\,\text{dec}} = 0 \).

As an example, we consider the combination of a SISO decoder (based on a rate-1/2 convolutional code with generators (5,7)) and a SISO canceler (based on MMSE interference cancellation) on a MIMO system with 4 transmit and 4 receive antennas. Additionally, QPSK modulation is assumed, and the channel matrix \( \mathbf{H} \) is chosen as in [9], with \( E_b/N_0 = -2 \) dB. Fig. 16 illustrates the first few iterations of the turbo device operation. The figure shows the EXIT functions of the SISO decoder (dashed line) and of the SISO canceler (solid line). They are taken from Figs. 9 (after abscissa-ordinate inversion) and 14, respectively. The dotted lines plot the constant-BER curves computed by using (13). The arrows indicate the first few iterations of the turbo algorithm: vertical arrows correspond to interference cancellation, while horizontal arrows correspond to decoding. The points labeled \( k = 0, 1, 2 \) correspond to the extrinsic mutual information at the output of the SISO decoder after \( k \) iterations. Finally, BER values are reported in the figure (bottom left) obtained by Monte Carlo simulation for comparisons with the values computed by using (13) (dotted curves). Fig. 17 shows the BER for the same system obtained by simulation (solid lines) and by EXIT chart analysis (points), for \( k = 0, 1, 2, 8 \) iterations.

**B. Quasi-static channel**

In quasi-static conditions, the channel matrix \( \mathbf{H} \) is random, and changes independently from codeword to codeword. This implies that the SISO-canceler EXIT function changes with \( \mathbf{H} \), and should be evaluated for a large number of samples in order to estimate the error performance of the system. On the contrary, the SISO-decoder EXIT function remains constant at a fixed value of \( \sigma^2_a \) (or \( E_b/N_0 \)). The computational burden necessary for convergence analysis might be heavy, but can be substantially alleviated by observing (through computational experience) that the SISO-canceler EXIT function exhibits in most cases an almost linear behavior and, as a consequence, only two points are needed to plot it as a straight line. Numerical results showed that the MAP canceler EXIT function is more linear than the MMSE and MRC ones. Also, the level of \( E_b/N_0 \) considered seems to have little influence on the linearity of the canceler EXIT function.

The approximation is illustrated by Fig. 18, which considers the same system as that of Fig. 16 and plots in addition the straight-line approximation of the EXIT function of the SISO canceler. The convergence points (obtained by the intersection of the SISO canceler and decoder EXIT functions) are denoted by \( C \) and \( C' \) for the proper and approximate SISO-canceler EXIT functions, respectively. Obviously, these points lie on the decoder EXIT function and represent the asymptotic performance attainable with an infinite number of iterations. It must be noted that the straight-line approximation leads to nonconservative convergence estimates, due to the upward convexity of the exact EXIT function of the SISO canceler. Nevertheless, numerical results show that the approximation...
the number of iterations from 1 to 8.

Finally, Fig. 20 compares the BER obtained by simulation and by “linearized” EXIT chart analysis to show their distribution for the same system parameters. In the case considered, an error of up to about 0.5 dB.

is fairly accurate.

A sample set of convergence points is plotted in Fig. 19 to show their distribution for the same system parameters. The points have been obtained by using different, randomly generated matrices $\mathbf{H}$ with iid circularly symmetric complex Gaussian random entries with zero mean and unit variance (independent MIMO Rayleigh fading channel). It is seen that the distribution of the points is quite concentrated, thus validating the assumption that their variance is close enough to zero.

Finally, Fig. 20 compares the BER obtained by simulation (solid lines) and by the “linearized” EXIT chart analysis (dots) and for $k = 0, 1, 2, 8$ iterations. The figure shows that the EXIT chart analysis provides slightly nonconservative results and, in the case considered, an error of up to about 0.5 dB. It can also be noticed from the figure that the error increases slightly with increasing $E_b/N_0$ and decreases by increasing the number of iterations from 1 to 8.

VI. CONCLUSIONS

After a general introduction on iterative MIMO receivers, we have analyzed the performance of combinations of a SISO decoder and a spatial-interference canceller over the MIMO channel. We showed that EXIT charts can be used in conjunction with “linearization” of the interference-cancellation characteristics so as to extend their applicability from the case of constant channel to the case of quasi-static fading channel. The resulting approximation yields results that are very close to simulation but are obtained in a considerably shorter time.

ACKNOWLEDGMENTS

The authors are grateful to Tor Aulin, Joseph Boutros, and Joachim Hagenauer for useful discussions concerning some of the topics covered in this paper. They also thank the Guest Editor Murat Uysal and Erik Larsson, Arthur Hashizume for their comments and support in the preparation of the final manuscript.

REFERENCES


