1. INTRODUCTION

Bit-Interleaved Coded Modulation (BICM) was first suggested by Zehavi in [1] to improve the Trellis Coded Modulation performance over Rayleigh-fading channels. In BICM, the diversity order is increased by using bit-interleavers instead of symbol interleavers. This improvement is achieved at the expense of a reduced minimum Euclidean distance leading to a degradation over non-fading Gaussian channels [1], [2]. This drawback can be overcome by using iterative decoding (BICM-ID) at the receiver. BICM-ID is known to provide excellent performance for both Gaussian and fading channels.

The iterative decoding scheme used in BICM-ID is very similar to serially concatenated turbo-decoders. Indeed, the serial turbo-decoder makes use of an exchange of information between computationally efficient decoders for each of the component codes. In BICM-ID, the inner decoder is replaced by demapping which is less computationally demanding than a decoding step. Even if this paper focus on iterative decoding for BICM, the results can be applied to the large class of iterative decoders including serial or parallel concatenated turbo decoders as long as low-density parity-check (LDPC) decoders. The turbo-decoder and more generally iterative decoding was not originally introduced as the solution to an optimization problem rendering the analysis of its convergence and stability very difficult. Among the different attempts to provide an analysis of iterative decoding, the EXIT chart analysis and density evolution have permitted to make significant progress [3] but the results developed within this setting apply only in the case of large block length. Another tool of analysis is the connection of iterative decoding to factor graphs [4] and belief propagation [5]. Convergence results for belief propagation exists but are limited to the case where the corresponding graph is a tree which does not include turbo code or LDPC. A link between iterative decoding and classical optimization algorithms has been made recently in [6] where the turbo decoding is interpreted as a nonlinear block Gauss Seidel iteration for solving a constrained optimization problem. In parallel, a geometrical approach has been considered and provides an interesting interpretation in terms of projections. The particular case of BICM-decoding has been studied by Muquet in [7] where the decoding sub-block is interpreted as two successive projections. The interpretation of the demapping sub-block in terms of projection remains unachieved. In [8], the turbo-decoding is interpreted in a geometric setting as a dynamical system leading to new but incomplete results. The failure to obtain complete results is mainly due to the inability to efficiently describe extrinsic information passing in terms of information projection.

Here, we emphasize the connection between iterative decoding and the Dykstra’s algorithm from the convex optimization literature [9]. The extrinsics are exactly the deflected versions of the previous outputs passed through the blocks in the Dykstra’s algorithm.

2. TOOLS

2.1 BICM-ID with soft decision feedback

A conventional BICM system [2] is built from a serial concatenation of a convolutional encoder, a bit interleaver and an M-ary bits-to-symbol mapping (where \( M = 2^m \)) as shown in fig. 1. The sequence of information bits \( \mathbf{b} \) is first encoded by a convolutional encoder to produce the output encoded bit sequence \( \mathbf{c} \) of length \( L_c \) which is then scrambled by a bit interleaver (as opposed to the channel symbols in the symbol-interleaved coded sequence) operating on bit indexes. Let \( \mathbf{d} \) denote the interleaved sequence. Then, \( m \) consecutive bits of \( \mathbf{d} \) are grouped as a channel symbol \( d_k = (d_{km+1}, \ldots, d_{(k+1)m}) \).

The complex transmitted signal \( y_k = s_k + n_k \) \( 1 \leq k \leq L_c/m \) (1)

where \( n_k \) is a complex white Gaussian noise with independent in-phase and quadrature components having two-sided power spectral density \( \sigma_n^2 \).

Due to the presence of the random bit interleaver, the true maximum likelihood decoding of BICM is too complicated to implement in practice. Figure 2 shows the block diagram...
of the receiver for a BICM-ID system with soft-decision feedback. In the first iteration, the encoded bits are assumed equally likely. The demapping consists in evaluating a posteriori probabilities (APP) for the encoded bits without accounting for the code structure, namely:

\[ p_{\text{APP}}(d_{km+i} = b) = p(d_{km+i} = b | y_k) \sim \sum_{s_k \in \Psi'_b} p(y_k | s_k)p(s_k) \]

where \( \Psi'_b, b \in \{0, 1\} \), denotes the subset of \( \Psi \) that contains all symbols whose labels have the value \( b \) in the \( i^{th} \) position. In the turbo decoding process, the quantities exchanged through the blocks are not a posteriori probabilities (APP) but extrinsic information [10]. The extrinsic information at the output of the demapping \( p(d_{km+i}; O) \) is computed as \( p_{\text{APP}}(d_{km+i}; O) / p(d_{km+i}; I) \) where \( p(d_{km+i}; I) \) is the a priori information for the demapping sub-block. Since the bit interleaver makes the bits independent, the extrinsic information \( p(d_{km+i}; O) \) reads:

\[ p(d_{km+i} = b; O) = K_m \sum_{s_k \in \Psi'_b} p(y_k | s_k) \prod_{j \neq i} p(d_{km+j}; I) \]

and the corresponding APP reads:

\[ p_{\text{APP}}(d_{km+i} = b) = K'_m \sum_{s_k \in \Psi'_b} p(y_k | s_k) \prod_{1 \leq j \leq m} p(d_{km+j}; I) \]

where \( K_m \) and \( K'_m \) are normalization factors and \( O \) and \( I \) refer to the output and the input. Note that \( p(d_{km+i}; O) \) is computed from the a priori probabilities \( p(d_{km+i}; I) \) of the others bits on the same channel symbol. However, we can also write the APP using the whole sequence as:

\[ p_{\text{APP}}(d_{km+i} = b) = p(d_{km+i} = b | y_k) \sim \sum_{s_k, s_j l \in \Psi'_b} p(y | s)p(s) \]

The extrinsic information \( p(d_{km+i}; O) \) is de-interleaved and delivered to the SISO decoder [11] as an a priori information on the encoded bits. Let \( c_i = d_{\sigma^{-1}(km+i)} \) where \( \sigma^{-1} \) is the permutation on the indexes due to the deinterleaver; \( p(c_i; I) \) is the updated input of the Soft Input Soft Output (SISO) decoder. The extrinsic information at the output of the SISO decoder is obtained through:

\[ p(c_i = b; O) = K_c \sum_{c \in \mathcal{C}} I_{\mathcal{C}}(c) \prod_{j \neq i} p(c_j; I) \]

and the corresponding APP is:

\[ p_{\text{APP}}(c_i = b) = K'_c \sum_{c \in \mathcal{C}} I_{\mathcal{C}}(c) \prod_{1 \leq j \leq L_c} p(c_j; I) \]

where \( I_{\mathcal{C}}(c) \) stands for the indicator function of the code, i.e. \( I_{\mathcal{C}}(c) = 1 \) if \( c \) is a codeword and 0 otherwise and \( \mathcal{C} \)

denotes the set of binary words of length \( L_c \) with value \( b \) in the \( i^{th} \) position. \( K_c \) and \( K'_c \) are normalization factors. The extrinsic information \( p(c_i; O) \) is interleaved and delivered to the demapping sub-block as a regenerated a priori information. The process is continued until the APP at the output of the two sub-blocks are the same or until the maximal iteration number is reached.

In the next section, we give an interpretation of the demodulation process via information geometry.

### 2.2 Simple facts from information geometry

Suppose that \( p \) and \( q \) are probability measures defined on subsets of \( \mathcal{H} \) where \( \mathcal{H} \) is the set of the first \( 2^{L_c} \) integers. The I-divergence of \( p \) with respect to \( q \) also called Kullback-Leibler divergence is given by:

\[ D(p \parallel q) = \sum_{k=1}^{2^{L_c}} p(k)\ln(p(k)/q(k)) \]

The minimum of \( D(p \parallel q) \) for \( p \) in a subset \( \mathcal{S} \) of \( \mathcal{H} \) is denoted \( D(p \parallel q) \) or \( D(q \mid \mathcal{S}) \). If a unique minimizer exists it is called the I-projection of \( q \) to \( \mathcal{S} \). Similarly, the minimum of \( D(p \parallel q) \) for \( q \) in a subset \( \mathcal{S} \) of \( \mathcal{H} \) is denoted \( D(p \parallel \mathcal{S}) \). If a unique minimizer exists it is called the reverse I-projection of \( p \) to \( \mathcal{S} \) [12]. These projections can also be termed respectively as backward and forward Bregman projection based on the Bregman distance \( D_f \) with \( f(x) = x\ln(x) - x \).

Now, we examine the projections onto linear and exponential families. They are of particular interest in our context and they give rise to a Pythagorean theorem.

**Definition 1 ([13])** For any given function \( f_1, f_2, \ldots, f_r \) on \( \mathcal{H} \) and numbers \( \alpha_1, \alpha_2, \ldots, \alpha_r \), the set

\[ \mathcal{L} = \{ p : \sum_{x} p(x)f_i(x) = \alpha_i, 1 \leq i \leq k \} \]

if non empty, will be called a linear family of probability distributions. Moreover the set \( \mathcal{E} \) of all \( p \) such that

\[ p(x) = cq(x)\exp(\sum_{i=1}^{r} \theta_i f_i(x)), \text{ for some } \theta_1, \ldots, \theta_r \]

will be called an exponential family of probability distributions; here \( q \) is a given distribution and \( c \) is a normalization factor.

The linear family \( \mathcal{L} \) is completely defined by the functions \( f_1, f_2, \ldots, f_r \) and the scalars \( \alpha_1, \alpha_2, \ldots, \alpha_r \). The exponential family \( \mathcal{E} \) is completely defined by the distribution \( q \) (which belongs to \( \mathcal{E} \)) and the functions \( f_1, f_2, \ldots, f_r \). Let \( \mathcal{E} \cdot \mathcal{E} \) denote the product manifold ie the set of all \( L_c \)-variate distribution with independent components. This family will reveal to be of great importance for our study. It is clear that \( \mathcal{E} \cdot \mathcal{E} \) is an exponential family [7]. The Pythagorean theorems are stated below:

**Theorem 1 ([13])** The I-projection \( p^* \) of \( q \) onto a linear family \( \mathcal{L} \) is unique and satisfies the Pythagorean identity

\[ D(p \parallel q) = D(p \parallel p^*) + D(p^* \parallel q) \quad \forall p \in \mathcal{L} \]
Similarly, it exists a Pythagorean identity for reverse I-projections on exponential families.

**Theorem 2 ([12])** The reverse I-projection $q^*$ of $p$ onto an exponential family $\mathcal{E}$ is unique and satisfies the Pythagorean identity

$$D(p \parallel q) = D(p \parallel q^*) + D(q^* \parallel q) \quad \forall q \in \mathcal{E}$$

For I-projections onto $\mathcal{L}_\mathcal{E}$, analytical results exist. The I-projection $p^*$ of $q$ onto a linear family (using notations in definition 1) reads [14]:

$$p^*(x) = q(x) \exp\left(-\sum_i \mu_i (f_i(x) - a_i)\right) \quad (9)$$

where the $\{\mu_i\}$ are Lagrange multipliers determined from the constraints. In the particular case where the functions $f_i$ in the definition of the linear family are the parity-check equations of a code and if $\{\alpha_i\} = 0$, then the I-projection in (9) reads [14]:

$$p^*(x) = q(x) \exp(-\mu_0) I_1(x) I_2(x) \ldots I_r(x) \quad (10)$$

where $I_i(x)$ is the indicator function for all vectors $x$ which satisfy constraint $f_i$ and $\exp(-\mu_0)$ is a normalization constant. The indicator function for a codeword is $I_\mathcal{E}(x) = I_1(x) I_2(x) \ldots I_r(x)$ since a codeword must satisfy all of the parity constraint simultaneously. Thus, the I-projection $p^*$ of $q$ onto the linear family $\mathcal{L}_\mathcal{E}$ formed with the parity-check equations of the code $\mathcal{E}$ reads:

$$p^*(x) = \frac{q(x) I_\mathcal{E}(x)}{\sum x q(x) I_\mathcal{E}(x)} \quad (11)$$

The reverse I-projection $q^*$ of $p$ onto $\mathcal{E}_\mathcal{D}$ also admits a closed-form expression:

$$q^*(x) = q^*(x_1, x_2, \ldots, x_i) = \prod_i p_i(x_i) \quad (12)$$

where $p_i(x_i)$ is the marginal distribution on $x_i$ of the probability measure $p$. In the next section, the link between I-projections, reverse I-projections and iterative decoding is emphasized.

### 2.3 Interpretation of iterative decoding

Let $p_{\text{APP}}(d)$ respectively $p_{\text{APP}}(c)$ denote the probability measure belonging to the product space $\mathcal{E}_\mathcal{D}$ with marginal distributions $p_{\text{APP}}(d_{km+1})_{i=1,\ldots,m; k=1,\ldots,L_c/m}$ respectively $p_{\text{APP}}(c_{i1})_{1 \leq i \leq L_c}$. From (5) and (12), we can conclude that $p_{\text{APP}}(d)$ is the reverse I-projection of $p(y | s)p(d; I)$ onto $\mathcal{E}_\mathcal{D}$. With an AWGN channel, $p(y | s)$ is a Gaussian distribution, namely

$$p(y | s) = K \exp\left(-\frac{\|y - s\|^2}{2\sigma^2}\right)$$

where $K$ is a normalization factor. Thus, from (9), $p(y | s)p(d; I)$ is the I-projection of $p(d; I)$ onto the linear family $\mathcal{L}_\mathcal{D} = \{p : \sum x p(x) = 1 \quad \sum x p(x) \| y - x \|^2 = \alpha (\sigma^2)\}$. Let $\Pi_\mathcal{D}$ respectively $\Pi_\mathcal{E}$ denote the I-projection onto the linear family $\mathcal{L}_\mathcal{D}$ respectively the reverse I-projection onto the exponential family $\mathcal{E}$. Thus $p_{\text{APP}}(d) = D_{\mathcal{D}}(p(d; I))$ where:

$$D_{\mathcal{D}} : \mathcal{E}_\mathcal{D} \rightarrow \mathcal{E}_\mathcal{D} : q \mapsto \Pi_\mathcal{E}[\Pi_\mathcal{D}(q)]$$

Note that the linear family involved in the demapping is changing from one iteration to another. However, at each iteration, it exists a linear family such that $p(y | s)p(d; I)$ is the I-projection of $p(d; I)$ onto this linear family. From (7), (11) and (12), we can also conclude that $p_{\text{APP}}(c) = D_{\mathcal{E}}(p(c; I))$ where:

$$D_{\mathcal{E}} : \mathcal{E}_\mathcal{D} \rightarrow \mathcal{E}_\mathcal{D} : q \mapsto \Pi_\mathcal{D}[\Pi_\mathcal{E}(q)]$$

The extrinsic information is obtained by the point-wise division of the APP at the output of each block with the input of the same block. The geometric interpretation of the iterative decoding is summarized in fig. 3. Without extrinsic propagation (ie with APP propagation), the iterative decoding would be the alternate projection algorithm of Csiszár [15] or more generally the method of Bregman retractions. The convergence of these algorithms is well established for projection onto closed convex sets. The linear families are closed convex sets. The exponential families are log-convex and are not closed [13]. Thus, the classical results on the convergence of alternate minimization can not be applied in the setting of iterative decoding so there is no guarantee that the APP provided by the demapping and decoding block will converge toward the same limit. Dykstra’s algorithm is a well-known algorithm with stronger properties than the alternate projection procedure. In the following section, we prove that the propagation of extrinsic rather than APP is Dykstra’s algorithm with I-projections and reverse I-projections.

### 3. MAIN RESULT

#### 3.1 Dykstra’s algorithm with I-projections

Dykstra’s algorithm respectively the method of cyclic projections are often employed to solve best approximation respectively convex feasibility problems. The Dykstra’s algorithm is an iterative procedure which (asymptotically) finds the nearest point of any given point onto the intersection of a family of closed convex sets (best approximation problem). In the method of cyclic projections, the output of the previous projection is delivered to the next projection whereas, in the Dykstra’s algorithm, only a reflected version of the previous output is given as an input to the next projection. The algorithm was first proposed and analysed by Dykstra in 1983 for orthogonal projection onto closed convex sets. This work was extended to I-projections in 1985 [9]. Here, we focus on the Dykstra’s algorithm with I-projections [9].

Let $\Pi_{\mathcal{L}_i}$ stand for the I-projection onto $\mathcal{L}_i$. Here $i \in \{1,2,3,4\}$ (as in the iterative decoding) however the procedure is valid for $t$ projections with $t$ finite. In the following, the interleaver/deinterleaver is omitted. Actually, these operators realize permutations of the bit indexes. As far as Kullback-Leibler minimizations are concerned, these
permutations have no insights on the result of the projection. All the products and divisions are point-wise operators: \( u = (pq)/r \) stands for \( u(k) = (p(k)q(k))/r(k)\forall k. 

**Dykstra’s algorithm**

- **Initialization**
  Let \( s_{1,1} = r \) and let \( p_{1,1} = \Pi_{\mathcal{E}}(s_{1,1}). \)
  Let \( s_{1,2} = p_{1,1} = r(p_{1,1}/s_{1,1}). \) We note that if \( s_{1,1}(k) = 0, \) then so is \( p_{1,1}(k). \) We take 0/0 to be 1.
  Set \( p_{1,2} = \Pi_{\mathcal{E}}(s_{1,2}). \)
  Let \( s_{1,3} = p_{1,2} \) and set \( p_{1,3} = \Pi_{\mathcal{E}}(s_{1,3}). \)
  Let \( s_{1,4} = p_{1,3} \) and set \( p_{1,4} = \Pi_{\mathcal{E}}(s_{1,4}). \)

- **Iteration n**
  Let \( s_{n,1} = p_{n-1,4}/(p_{n-1,1}/s_{n-1,1}) \) and let \( p_{n,1} = \Pi_{\mathcal{E}}(s_{n,1}). \)
  For \( i = 2,3,4 \) let \( s_{n,i} = p_{n-i,4}/(p_{n-i,1}/s_{n-1,i}) \) and let \( p_{p,n,i} = \Pi_{\mathcal{E}}(s_{n,i}). \)

For closed convex set, this procedure converges towards \( u \in \mathcal{E} = \cap_k \mathcal{E}_k \) and \( D(u \| r) = \arg\min_{p \in \mathcal{P}} D(p \| r) \) be the procedure converges towards the closest point to \( r \) in \( \mathcal{E}. \) Note that, for closed convex sets, the classical alternating minimization procedure will also converge to a point in \( \cap_k \mathcal{E}_k, \) not necessarily the closest point to \( r \) in \( \cap_k \mathcal{E}_k. \) For closed convex sets, Dykstra’s algorithm solves the best approximation problem whereas the alternating minimization procedure solves convex feasibility problems. In that sense, Dykstra’s algorithm exhibits stronger properties than the alternating minimization procedure.

### 3.2 Linking Dykstra’s algorithm and iterative decoding

For comparison, we provide below the iterative procedure commonly used in iterative decoding.

**Iterative decoding**

- **Initialization**
  Let \( v_{1,1} = (1/2^4 \ldots 1/2^4) \) and set \( p_{1,1} = \Pi_{\mathcal{L}}(v_{1,1}). \)
  Let \( v_{1,2} = p_{1,1} \) and set \( p_{1,2} = \Pi_{\mathcal{E}}(p_{1,1}). \)
  Let \( v_{1,3} = p_{1,2}/v_{1,1} \) and set \( p_{1,3} = \Pi_{\mathcal{L}}(v_{1,3}). \)
  Let \( v_{1,4} = p_{1,3} \) and set \( p_{1,4} = \Pi_{\mathcal{E}}(p_{1,3}). \)

- **Iteration n**
  Let \( v_{n,1} = p_{n-1,4}/v_{n-1,3} \) and set \( p_{n,1} = \Pi_{\mathcal{L}}(v_{n,1}). \)
  Let \( v_{n,2} = p_{n,1} \) and set \( p_{n,2} = \Pi_{\mathcal{E}}(p_{n,1}). \)
  Let \( v_{n,3} = p_{n,2}/v_{n,1} \) and let \( p_{n,3} = \Pi_{\mathcal{L}}(v_{n,3}). \)
  Let \( v_{n,4} = p_{n,3} \) and set \( p_{n,4} = \Pi_{\mathcal{E}}(p_{n,3}). \)

Even if, at first glance, the two procedure seems slightly different, they produce exactly the same sequence of projected distributions onto \( \mathcal{E} \) \( i.e., \) they produce the same sequences \( \{p_{n,2}\} \) and \( \{p_{n,4}\} \) as stated in theorem 3. Note that, in the iterative decoding, \( \{p_{n,2}\} \) and \( \{p_{n,4}\} \) are the APPs at the output of each sub-block which are intended to converge towards the same solution \( p^* \). The hard decisions rely on \( p^*. \) Thus \( \{p_{n,2}\} \) and \( \{p_{n,4}\} \) are of particular importance in our setting.

**Theorem 3** Iterative decoding and the Dykstra’s algorithm with \( r = (1/2^4 \ldots 1/2^4) \). \( \Pi_{\mathcal{E}} = \Pi_{\mathcal{L}}, \) \( \Pi_{\mathcal{E}} = \Pi_{\mathcal{E}} \) and \( \Pi_{\mathcal{E}} = \Pi_{\mathcal{E}} \) lead to the same sequence of projected distributions \( \{p_{n,2}\} \) and \( \{p_{n,4}\}. \)

**Proof:**

- **Initialization.** By definition, \( p_{1,1}, p_{1,2} \) and \( p_{1,4} \) are the same in the two procedures. In the iterative decoding, \( p_{1,3} = \Pi_{\mathcal{E}}(v_{1,3}) = \Pi_{\mathcal{E}}(p_{1,2}/v_{1,1})\). Since \( v_{1,1} = (1/2^4 \ldots 1/2^4) \) then \( p_{1,3} = \Pi_{\mathcal{E}}(p_{1,2}) \) which is the definition of \( p_{1,3} \) in the Dykstra’s algorithm.

- **Iteration n.** We prove here that \( s_{n,2} \) and \( v_{n,2} \) are proportional to:

\[
\frac{p_{n-1,4}p_{n-2,4} \ldots p_{1,4}}{p_{n-1,2}p_{n-2,2} \ldots p_{1,2}} \exp\left(-\frac{\|y-s(d)\|^2}{2\sigma^2}\right) \tag{13}
\]

and that \( s_{n,4} \) and \( v_{n,4} \) are proportional to:

\[
\frac{p_{n,2}p_{n-1,2} \ldots p_{1,2}}{p_{n-1,4}p_{n-2,4} \ldots p_{1,4}} I_{\mathcal{E}}(c) \tag{14}
\]

For \( n = 2 \) case,

\[
s_{2,2} = \frac{p_{2,1}p_{1,1}}{p_{2,1}} = \frac{r_{p_{1,1}} p_{1,1} p_{1,2} p_{1,3} p_{1,4}}{p_{2,1} p_{1,2} p_{1,3} p_{1,4} I_{\mathcal{E}}(c) = \frac{p_{2,1} p_{2,2}}{p_{1,4}} I_{\mathcal{E}}(c) \tag{14}
\]

So \( s_{n,2} \approx \frac{p_{n-1,2} p_{n-1,4} \ldots p_{1,2}}{p_{n-1,4} p_{n-2,4} \ldots p_{1,4}} I_{\mathcal{E}}(c) \) which can be simplified as \( s_{n,2} \approx \frac{p_{n-1,2} p_{n-1,4} \ldots p_{1,2}}{p_{n-1,4} p_{n-2,4} \ldots p_{1,4}} I_{\mathcal{E}}(c) \) since \( p_{n-1,1} \) is the projection of \( s_{n-1} \) onto \( \mathcal{L} \). This proves that \( s_{n,2} \) is proportional to the expression in (13).

For the iterative decoding, \( v_{n,2} \approx \frac{p_{n-1,2} p_{n-1,4} \ldots p_{1,2}}{p_{n-1,4} p_{n-2,4} \ldots p_{1,4}} I_{\mathcal{E}}(c) \) which proves that \( s_{n,2} \) is proportional to the expression in (13). So we have, \( \Pi_{\mathcal{E}}(v_{n,2}) = \Pi_{\mathcal{E}}(p_{n,2}) \).

In the same way, we have \( s_{n,4} = \frac{p_{n-1,4} p_{n-1,2} \ldots p_{1,4}}{p_{n-1,4} p_{n-2,2} \ldots p_{1,4}} I_{\mathcal{E}}(c) \) which is equivalent to \( s_{n,4} = \frac{p_{n-1,4} p_{n-1,2} \ldots p_{1,4}}{p_{n-1,4} p_{n-2,2} \ldots p_{1,4}} I_{\mathcal{E}}(c) \). So, \( s_{n,4} \) is proportional to the expression in (14).
For the iterative decoding, \( v_{n,4} \propto v_{n,3}I_{\mathcal{E}}(c) \). The definitions of \( v_{n,3} \) and \( v_{n,1} \) leads to \( v_{n,4} \propto \frac{p_{n-1}2^{n-1,3}}{p_{n-1,1}}I_{\mathcal{E}}(c) \). Since \( v_{n-1,4} = p_{n-1,3} \) is the projection of \( v_{n-1,3} \) onto \( \mathcal{L}_{\mathcal{E}} \), we finally obtain \( v_{n,4} \propto \frac{p_{n-1}2^{n-1,4}}{p_{n-1,4}} \) which proves that \( v_{n,4} \) is proportional to the expression in (14). So the projections of \( v_{n,4} \) and \( s_{n,4} \) onto \( \delta_{\mathcal{P}} \) are the same.

\( \square \)

In the original version of the Dykstra’s algorithm, all the projections are I-projections onto (non-varying) closed convex sets. In the iterative decoding, I-projections onto closed convex sets are involved as long as reverse I projections onto log-convex sets. Thus, the convergence results can not be extended straightforwardly to iterative decoding. In particular, there is no guarantee that the iterative decoding converges towards the closest point to \( r = (1/2^{L_c}...1/2^{L_n}) \) in the set \( \mathcal{L}_{\mathcal{E}} \cap \mathcal{L}_{\mathcal{M}} \cap \delta_{\mathcal{P}} \). However, based on the duality between projections onto linear and exponential families, the reverse I-projection onto the set of separable densities is equivalent to an I-projection onto a particular (varying) linear family (see [16]). Thus, iterative decoding can also be written with I-projections onto closed convex sets. The difference with the classical Dykstra’s algorithm is limited to the “varying” nature of the linear family involved in the minimization process. A recent work gives some elements to tackle this problem. Indeed, Niesen and al. proposed in [17] a generalization of the alternating minimization procedure of Csiszar to the case of projections onto time-varying sets. This proof combined with the convergence results in [9] seems a promising direction of investigation for the derivation of new convergence results for iterative decoding.

4. CONCLUSION

In this paper, we have presented some tools and concepts of information geometry that apply for the description of iterative receivers. The extrinsic propagation is very similar to the deflected output propagation used in Dykstra’s algorithm. This similarity suggests that convergence results might be developed via this analogy.

REFERENCES