HAL open science

A Lifting Decoding Scheme and its Application to Interleaved Linear Codes

Guillaume Quintin

To cite this version:

Guillaume Quintin. A Lifting Decoding Scheme and its Application to Interleaved Linear Codes. 2012. hal-00673938v1

HAL Id: hal-00673938

https://inria.hal.science/hal-00673938v1

Preprint submitted on 24 Feb 2012 (v1), last revised 22 May 2012 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Lifting Decoding Scheme and its Application to Interleaved Linear Codes

Guillaume Quintin
Laboratoire d'informatique de l'X (LIX),
École polytechnique, 91128 Palaiseau, FRANCE,
Email: quintin@lix.polytechnique.fr.

***Abstract*—In this paper we design a decoding algorithm based on a lifting decoding scheme. This leads to a unique decoding algorithm for Reed-Solomon codes over Galois rings with a very low complexity, and a list decoding algorithm. We show that, using erasures in our algorithms, allows to decode more errors than half the minimum distance with a high probability. Finally we apply these techniques to interleaved linear codes over a finite field and obtain a decoding algorithm that can recover more errors than half the minimum distance.**

***Index Terms*—Algorithm design and analysis, Decoding, Error correction, Reed-Solomon codes, Interleaved codes.**

I. Introduction

Reed-Solomon codes (denoted by RS codes in the rest of this paper) form an important and well-studied family of codes. They can be efficiently decoded. See for example [9], [14]. They are widely used in practice [18]. A breakthrough has been made by Madhu Sudan in 1997 about the list decoding of RS codes in [17], further improved by Venkatesan Guruswami and Madhu Sudan in [13]. They showed that RS codes are list decodable up to the generic Johnson bound in polynomial time.

A. Our contributions

Let B be the quotient ring of a discrete valuation ring A. We design a decoding scheme that can be adapted to a wide family of linear codes over B. More precisely let $\mathcal{C}$ be a code over B. If we denote by π a uniformizing parameter of A, given a blackbox decoding algorithm `BlackBoxDec` for $\mathcal{C}/\pi\mathcal{C}$, we can construct a decoding algorithm for $\mathcal{C}$ generalizing [11, algorithm of Section 3]. The constructed decoding algorithm has the property to correct all error patterns that can be corrected by `BlackBoxDec`. We study in details the complexities in the case of Reed-Solomon codes over Galois rings and truncated power series rings.

We improve the construction given in [11, algorithm of Section 3] by integrating an idea used by M. Armand in [3], [4]. This improvement allows to decode more error patterns than `BlackBoxDec` with a high probability at the same low cost. Again we study in detail and give detailed complexities concerning RS codes.

Finally we show that, given any linear code $\mathcal{C}'$ over $\mathbb{F}_q$, we can view interleaved codes with respect to $\mathcal{C}'$ as a code over $\mathbb{F}_q[[t]]/(t^r)$. This allows us to apply the previous techniques to interleaved codes and obtain a decoding algorithm that can decode more errors than half the minimum distance of $\mathcal{C}'$ with a high probability over small alphabets *i.e.* finite fields.

B. Related work

Our approach for a lifting decoding scheme has first been studied in [11], then in [5], [7] RS codes over a commutative finite ring have been studied by M. Armand in [1]–[4]. The decoding of interleaved has been studied in [6], [8].

II. Prerequisites

A. Complexity model

In order to analyse the complexities of our algorithms, we let $\mathsf{I}(n)$ be the time needed to multiply two integers of bit-size at most n in *binary representation*. It is classical that we can take $\mathsf{I}(n) \in O(n \log n 2^{\log^* n})$, where $\log^*$ represents the iterated logarithm of n. Let $\mathsf{M}(n)$ be the cost of multiplying two elements of the finite field $\mathbb{F}_{q^n}$ where q is a power of a prime number in terms of the number of arithmetic operations in $\mathbb{F}_q$. The "soft-Oh" notation $f(n) \in \tilde{O}(n)$ means that $f(n) \in g(n) \log^{O(1)}(3 + g(n))$. It is well known that $\mathsf{M}(n) \in \tilde{O}(n)$. Thus the bit-cost of multiplying two elements of $\mathbb{F}_{p^n}$ is $\tilde{O}(n \log p)$ where p is a prime number.

B. Error correcting codes

In this section we let A be any commutative ring with identity and n be a positive integer. Let C be a subset of A^n. We call C an *error correcting code over* A or simply a *code over* A. If C is a submodule of A^n we say that C is a *linear* code over A. The integer n is called the *blocklength* of C. If C is a linear code and that C is free of a rank k, we say that C has *parameters* $[n, k]_A$.

Definition 1. Let $u = (u_1, \ldots, u_n) \in A^n$. We call the integer

$$|\{i \in \{1, \ldots, n\} : u_i \neq 0\}|$$

the *Hamming weight* or simply the *weight* of u and denote it by $w(u)$. Let v be another vector of A^n. The integer $w(u-v)$ is called the *Hamming distance* or simply *distance* between u and v and denoted by $d(u, v)$.

The integer $d = \min_{u,v \in C \text{ and } u \neq v} d(u, v)$ is called the *minimum distance of* C. Note that when C is a linear code we have $d = \min_{u \in \mathcal{C} \setminus \{0\}} w(u)$, we then say that C has parameters $[n, k, d]_A$ if C is free of rank k.

Definition 2. Suppose that C is free of rank k. A matrix whose rows form a basis of C is called a *generator* matrix of C.

The generator matrix is used to encode a message. The elements $m \in A^k$ will be called *messages* inthe rest of this paper. The definition of the generator matrix that we have choosen induces a one-to-one correspondance between messages and codewords, the map $m \mapsto mG$ is a A-linear embbeding $A^k \to A^n$. Under this map, we will identify messages and codewords.

Let $\mathfrak{m}$ be a maximal ideal of A. The vector space $C/\mathfrak{m}C$, if not reduced to zero, is a linear code of parameters $[n, \leq k, \leq d]_{A/\mathfrak{m}}$ of generator matrix G'. The matrices G and G' have the same number of columns but can have a different number of rows. However G' can be deduced from G, first compute $G'' = G \mod \mathfrak{m}$, then remove from G'' rows to obtain a basis of $C/\mathfrak{m}C$.

Definition 3. Borrowing the terminology of [11, Section 3], if G and G'' have the same number of rows and colmuns and that $G \mod \mathfrak{m} = G'$ then $\mathcal{C}$ is called a *splitting* code.

C. Reed-Solomon codes over rings

Reed-Solomon codes over rings are defined in a slightly different way as their field counterpart. We denote by $A[X]_{<k}$ the submodule of $A[X]$ consisting of all the polynomials of degree at most $k-1$ of $A[X]$.

Definition 4. Let $x_1, \ldots, x_n$ be points of A such that for $i \neq j$, $x_i - x_j \in A^\times$ (where $A^\times$ is the group of units of A). The submodule of A^n generated by the vectors $(f(x_1), \ldots, f(x_n)) \in A^n$ is called a *Reed-Solomon* code over A. We will denote a Reed-Solomon code by a RS codes in the rest of this article. The n-tuple $(x_1, \ldots, x_n)$ is called the *support* of the RS code.

Proposition 5. *Let $\mathcal{C}$ be a RS code over A. Then $\mathcal{C}$ has parameters $[n, k, d = n-k+1]_A$.*

Proposition 6. *Let $\mathcal{C}$ be a RS code of parameters $[n, k, d = n-k+1]_A$ over a discrete valuation ring A of uniformizing parameter π. Then $\mathcal{C}/\pi^r\mathcal{C}$ is a RS code of parameters $[n, k, d]_{A/(\pi^r)}$ over $A/(\pi^r)$. Moreover of $(x_1, \ldots, x_n)$ is the support of $\mathcal{C}$ then $(x_1 \mod \pi^r, \ldots, x_n \mod \pi^r)$ is the support of $\mathcal{C}/\pi^r\mathcal{C}$.*

III. Improved π-adic lifting.

In this section we let A be a discrete valuation ring (also denoted by DVR). We denote by π a uniformizing parameter of A and by $\kappa = A/(\pi)$ the residual field of A. We also let $\mathcal{C}$ be a free splitting linear code over A of parameters $[n, k, d]_A$ and of generator matrix G. We denote by $\mathcal{C}'$ the linear code $\mathcal{C}/\pi\mathcal{C}$ and G' a generator matrix of $\mathcal{C}'$ deduced from G, *i.e.* $G' = G \mod \pi$.

Note that `BlackBoxDec` can return one or more codewords hence it can be a list decoding algorithm but we do not impose that it returns *all* codewords within distance τ from y.

Algorithm 1 `BlackBoxDec`

Input: A positive integer $\tau \leq n$ and a received vector y of κ^n (with zero or more erasures).

Output: A nonempty set $U \subseteq \kappa^k \times \kappa^n$ satisfying

$$(m, e) \in U \Rightarrow y = mG' + e \text{ and } w(e) \leq \tau \qquad (1)$$

or $\emptyset$ (meaning FAILURE).

Algorithm 2 Decoding from valuation i up to valuation r.

Input: A positive integer $\tau \leq n$, two nonnegative integers $i \leq r$, a received vector y of A^n (with zero or more erasures) and a blackbox decoding algorithm `BlackBoxDec` for $\mathcal{C}(\pi)$.

Output: A nonempty set $U \subseteq \kappa^k \times \kappa^n$ satisfying

$$(m, e) \in U \Rightarrow y = mG + e \mod \pi^{r-i} \text{ and } w(e) \leq \tau \qquad (2)$$

or $\emptyset$ (meaning FAILURE).

if $i = r$ **then**
  return $\{0\}$.
end if
Call to `BlackBoxDec` with input $(y \mod \pi)$ and τ to obtain the set S.
if `BlackBoxDec` returns $\emptyset$ (FAILURE) **then**
  return $\emptyset$ (FAILURE).
end if
$U \leftarrow \emptyset$.
for each $(m_0, e_0) \in S$ **do**
  $y_1 \leftarrow \pi^{-1}(y - m_0G - e_0)$.
  Put erasures in y_1 at the locations indicated by $\mathrm{Supp}(e_0)$.
  Call recursively Algorithm 2 with input τ, $i+1$, r, y_1 and `BlackBoxDec` to obtain the set T.
  for each $(m_1, e_1) \in T$ **do**
    if $|\mathrm{Supp}(e_0) \cup \mathrm{Supp}(e_1)| \leq \tau$ **then**
      $U \leftarrow U \cup \{(m_0 + \pi m_1, e_0 + \pi e_1)\}$.
    end if
  end for
end for
return U.

Algorithm 3 Decoding up to precision r.

Input: A positive integer $\tau \leq n$, a positive integer r, a received vector y of A^n (with zero or more erasures) and a blackbox decoding algorithm `BlackBoxDec` for $\mathcal{C}(\pi)$.

Output: A nonempty set $U \subseteq \kappa^k \times \kappa^n$ satisfying

$$(m, e) \in U \Rightarrow y = mG' + e \mod \pi^r \text{ and } w(e) \leq \tau \qquad (3)$$

or $\emptyset$ (meaning FAILURE).

return the set returned by the call to Algorithm 2 with input τ, 0, r, y and `BlackBoxDec`.

Proposition 7. *Suppose that* `BlackBoxDec` *returns* all *the codewords from $\mathcal{C}'$ within distance τ from $y \in \kappa^n$. Then Algorithm 2 can decode up to τ errors up to precision r.*

Proof: The proof is done by descending induction on i. For $i = r$ and $i = r - 1$ the proposition holds.

Now let $i < r-1$, $(m, e) \in \kappa^k \times \kappa^n$ and $c = mG$ such that $w(e \mod \pi^{r-i}) \leq \tau$ and $y = c + e$. There exists $(m_0, e_0) \in S$ such that $c_0 = m_0 G = c \mod \pi$, $e = e_0 \mod \pi$ and $\mathrm{Supp}(e_0) \subseteq \mathrm{Supp}(e)$. If we count erasures as errors, we have $w(e) \leq \tau$ and therefore $w(\pi^{-1}(e_0 - e)) \leq \tau$. On the other hand we have $mG = m_0 G \mod \pi$ *i.e.* $mG' = m_0 G'$ in $\mathcal{C}'$ whence $m = m_0 \mod \pi$. Therefore $\pi^{-1}(mG - m_0 G) = (\pi^{-1}(m - m_0))G \in \mathcal{C}$.

We deduce from the above that

$$y_1 = \pi^{-1}(y - (c_0 + e_0)) = \pi^{-1}(c - c_0) + \pi^{-1}(e_0 - e).$$

and that, with the inductive hypothesis, there exists $(m_1, e_1) \in T$ such that $\pi^{-1}(c - c_0) = m_1 G \mod \pi^{r-(i+1)}$ and $\pi^{-1}(e - e_0) = e_1 \mod \pi^{r-(i+1)}$. ∎

We now have the straightforward proposition which gives the complexity of Algorithm 3 in terms of bit-operations.

Proposition 8. *Suppose that the number of codewords returned by* `BlackBoxDec` *is at most $L > 1$. Denote by $\mathsf{Lift}(\mathcal{C})$ the complexity of lifting a codeword of $\mathcal{C}'$ into a codeword of $\mathcal{C}$ up to precision r in terms of the number of bit-operations. Denote by $\mathsf{Dec}(\mathcal{C})$ the complexity of algorithm* `BlackBoxDec` *in terms of the number of bit-operations. Then Algorithm 3 performs at most*

$$\frac{L^r - 1}{L - 1}\left(\mathsf{Lift}(\mathcal{C}) + \mathsf{Dec}(\mathcal{C})\right) = O(L^{r-1})\left(\mathsf{Lift}(\mathcal{C}) + \mathsf{Dec}(\mathcal{C})\right)$$

If $L \leq 1$ then Algorithm 3 performs at most $r\left(\mathsf{Lift}(\mathcal{C}) + \mathsf{Dec}(\mathcal{C})\right)$ bit-operations.

We are now able to give a decoding algorithm for $\mathcal{C}/\pi^r\mathcal{C}$

Algorithm 4 Decoding algorithm for $\mathcal{C}/\pi^r\mathcal{C}$.

Input: A positive integer $\tau \leq n$, a received vector y of $(A/(p^r))^n$ (with zero or more erasures) and a blackbox decoding algorithm `BlackBoxDec` for $\mathcal{C}(\pi)$.

Output: A nonempty set $U \subseteq \kappa^k \times \kappa^n$ satisfying

$$(m, e) \in U \Rightarrow y = mG' + e \text{ and } w(e) \leq \tau \qquad (4)$$

or $\emptyset$ (meaning FAILURE).

1: Lift $y \in (A/(p^r))^n$ into $y' \in A^n$.
2: $S \leftarrow$ the set returned by the call to Algorithm 2 with input τ, 0, r, y' and `BlackBoxDec`.
3: **return** $\{c \mod \pi^r : c \in S\}$.

The interesting part of Algorithm 2 (and hence of all other algorithms) resides in the `BlackBoxDec` argument. In the previous section we have shown that if `BlackBoxDec` is a classical decoding algorithm then Algorithm 3 becomes a decoding algorithm with the same decoding radius as `BlackBoxDec`.

From now we suppose that $\kappa = A/(\pi)$ is a finite field. Every element of $B = A/(\pi^r)$ can be written as $u\pi^s$ where $u \in B^\times$ and s is unique such that $0 \leq s \leq r - 1$.

We apply Algorithm 4 to RS codes over B when B is a Galois ring or the ring of truncated power series over a finite field. These codes are splitting codes which is a consequence of Proposition 6.

Proposition 9. *We suppose that $\mathcal{C}$ is a RS code over B. If $B = \mathrm{GR}(p^r, s)$ then*

- *if* `BlackBoxDec` *is a unique decoding algorithm (*i.e. *that can decode up to $\tau = \left\lfloor \frac{d-1}{2} \right\rfloor$) then Algorithm 4 can decode up to τ errors with at most $\tilde{O}(r(n \log^2 n + nk)s \log s \log\log s \log p)$ bit-operations,*
- *if* `BlackBoxDec` *is the Guruswami-Sudan list decoding algorithm that can decode up to $J = \left\lceil n - \sqrt{(k-1)n} \right\rceil - 1$ errors then Algorthim 4 can (list) decode up to J errors with at most $\tilde{O}\left([n(|\kappa| - 1)]^{r-1} n^7 k^5 s \log s \log\log s \log p\right)$ bit-operations.*

If $B = \kappa[[t]]/(t^r)$ then

- *if* `BlackBoxDec` *is a unique decoding algorithm (*i.e. *that can decode up to $\tau = \left\lfloor \frac{d-1}{2} \right\rfloor$) then Algorithm 4 can decode up to τ errors with at most $\tilde{O}(r(n \log^2 n + nk))$ arithmetic operations over κ.*
- *if* `BlackBoxDec` *is the Guruswami-Sudan list decoding algorithm that can decode up to $J = \left\lceil n - \sqrt{(k-1)n} \right\rceil - 1$ errors then Algorthim 4 can (list) decode up to J errors with at most $\tilde{O}\left([n(|\kappa| - 1)]^{r-1} n^7 k^5\right)$ arithmetic operations over κ.*

Proof: We first treat the cases where `BlackBoxDec` decode up to τ. The cost of `BlackBoxDec` is $O(n \log^2 n)$ by [14] the cost of lifting a codeword from $\mathcal{C}'$ into $\mathcal{C}$ up to precision r is the cost of computing $m_0 G$ in step 10. But $m_0 \in \kappa^k$, therefore the vector-matrix product $m_0 G$ can be computed with at most nk multiplications over κ and nk additions over A up to precision r.

For the cases where `BlackBoxDec` is a list decoding algorithm that can decode up to J errors, we have $L \leq n(|\kappa| - 1)$ by [12, Corollary 3.3, page 36]. The cost of `BlackBoxDec` is $O(n^7 k^5)$ by [12, Lemma 6.13, page 111] ∎

We show that if we choose a decoding algorithm able to handle errors and erasures for `BlackBoxDec` then we can decode, with a non negligible probability, further than half the minimum distance and further than the generic Johson bound.

Definition 10. Following the terminology of [15, Paragraph 2.1, page 404] we say that an element of B has *filtration* s if it is written $u\pi^s$ where $u \in B^\times$.

We let q be the cardinality of κ. Then the cardinality of B is q^r while the cardinality of $\frac{A/(\pi^s)}{A/(\pi^{s+1})}$ is q.

Proposition 11. *Let $\mathcal{C}$ be a splitting code over B of parameters $[n, k]_B$. Suppose that* `BlackBoxErasuresDec` *is*

Algorithm 5 `BlackBoxErasuresDec`

Input: A received vector y of κ^n with ϵ erasures and at most $\tau(\epsilon)$ errors.

Output: All the codewords within distance $\tau(\epsilon)+\epsilon$ from y or $\emptyset$ (FAILURE).

provided as the `BlackBoxDec` *argument to Algorithm 4. The number of error vectors of weight w, provided that ϵ erasures occured, which can be corrected by Algorithm 4 is at least*

$$N(\epsilon,B,w)=\binom{n}{\epsilon}q^{r\epsilon}\sum_{(v_0,\ldots,v_{r-1})\in V_w}\left[\binom{n-\epsilon}{w}\times\prod_{i=0}^{r-1}\binom{w-v_0-\cdots-v_{i-1}}{v_i}(q-1)^{v_i}q^{v_0+\cdots+v_{i-1}}\right] \quad (5)$$

where

$$V_w=\{(v_0,\ldots,v_{r-1})\in\mathbb{N}^r : v_0+\cdots+v_{r-1}=w \text{ and } 0\le v_0\le\tau(0) \text{ and } 0\le v_1\le\tau(v_0) \text{ and } \ldots 0\le v_{r-1}\le\tau(v_0+\cdots+v_{r-2})\},$$

hence the fraction of correctable error patterns is at least

$$P(\epsilon,B,w)=\frac{\sum_{i=0}^{w}N(\epsilon,B,w)}{\sum_{i=0}^{w}\binom{n}{i}(q^r-1)^i} \quad (6)$$

Proof: Let $e\in B^n$ be an error vector. We denote by $v_i(e)$ for $i=1,\ldots,r-1$ the number of coordinates of e of filtration i. The number of error vectors $e\in B^n$ such that $(v_0(e),\ldots,v_{r-1}(e))\in V_w$ is given by formula (5). Let c be a codeword of $\mathcal{C}$ and $y=c+e$ with $v_i=v_i(e)$ for $i=0,\ldots,r-1$ and $(v_0,\ldots,v_{r-1})\in V_w$. The rest of the proof is similar to the proof of Proposition 7. ∎

Proposition 12. *Let $\mathcal{C}$ be a splitting code over B of parameters $[n,k,d]_B$. Then there exists a decoding algorithm such that $\tau(\epsilon)=\lfloor\frac{d-\epsilon-1}{2}\rfloor$.*

Proof: This is a consequence of [16, Theorem 1.7, page 16]. ∎

Proposition 13. *Let $\mathcal{C}$ be a Reed-Solomon code over B of parameters $[n,k,d=n-k+1]_B$ then there exists*

- *a unique decoding algorithm which can correct errors and erasures with $\tau(\epsilon)=\lfloor\frac{n-\epsilon-k}{2}\rfloor$ and*
- *a list decoding algorithm which can correct errors and erasures with $\tau(\epsilon)=\left\lceil(n-\epsilon)-\sqrt{(k-1)(n-\epsilon)}\right\rceil-1$.*
- *a unique decoding algorithm which can correct up to w errors and ϵ erasures with $w\le n-\epsilon-k$ and which does succeed for a fraction of at least $P(\epsilon,B,w)$ error patterns.*

In addition the costs of Algorithm 4 are the same as the ones given in Proposition 9.

Proof: For the first item, see for exemple [9, Section 4, page 7 and 8] while for the second item see [13, Theorem 16, page 1762]. The third item is a consequence of the first item and Proposition 11. ∎

IV. Application to interleaved linear codes.

In this section we let A be the power series ring over the finite field $\mathbb{F}_q$ namely $A=\mathbb{F}_q[[t]]$, $\pi=t$ and $B=\mathbb{F}_q[[t]]/(t^r)$. We recall the construction of interleaved codes and show that all interleaved codes over $\mathbb{F}_q$ are exactly codes over B. We let $\mathcal{C}'$ be a linear code over $\mathbb{F}_q$ of parameters $[n,k,d]_{\mathbb{F}_q}$ and of generator matrix G'.

Let r messages $m_0,\ldots,m_{r-1}\in\mathbb{F}_q^k$ and their encodings $c_0=m_0G',\ldots,m_{r-1}G'$. For $i=0,\ldots,r-1$ and $j=1,\ldots,n$ define c_{ij} to be the j-th coordinate of c_i and $s_j=(c_{0,j},\ldots,c_{r-1,j})$.

$$\begin{array}{|c|c|c|c|cc}
\hline
c_{0,1} & c_{0,2} & \cdots & c_{0,n} & \rightarrow & c_0\\
\hline
c_{1,1} & c_{1,2} & \cdots & c_{1,n} & \rightarrow & c_1\\
\hline
\vdots & \vdots & \vdots & \vdots & & \\
\hline
c_{r-1,1} & c_{r-1,2} & \cdots & c_{r-1,n} & \rightarrow & c_{r-1}\\
\hline
\downarrow & \downarrow & & \downarrow & & \\
s_1 & s_2 & & s_n & &
\end{array}$$

The transmitted vectors over the channel are not $c_1,\ldots,c_{r-1}\in\mathbb{F}_q^n$ but $s_1,\ldots,s_n\in\mathbb{F}_q^r$. We will make an abuse of notation and call such an encoding scheme a *interleaved code with respect to $\mathcal{C}'$ and of degree r*. Usually the vector s_j (for $j=1,\ldots,n$) is seen as an element of $\mathbb{F}_{q^r}$, but we can associate the element $\sum_{i=0}^{r-1}c_{i,j}t^i\in B$ to s_j. In this context, if $y=(y_1,\ldots,y_n)\in(\mathbb{F}_q^r)^n$, the weight of y is the nonnegative number $|\{i\in\{1,\ldots,n\} : y_i\neq 0\}|$ and if y corresponds to the received word then the weight of the error is $|\{i\in\{1,\ldots,n\} : y_i\neq s_i\}|$. We assume throughout this section that all errors with weight at most τ occur with the same probability regardless of the weight of the transmitted codeword.

Proposition 14. *The trasmitted words over the channel using interleaved linear codes are exactly the same as the transmitted words using linear codes over truncated power series.*

Proof: Let $G=G'$ be the generator of the linear $\mathcal{C}$ code over B of parameters $[n,k,\le d]_B$. Then $\mathcal{C}/t\mathcal{C}=\mathcal{C}'$ and $\mathcal{C}$. We have $c_i=m_iG'$ for $i=0,\ldots,r-1$. As a consequence we have

$$\left(\sum_{i=0}^{r-1}m_it^i\right)G=\sum_{i=0}^{r-1}(m_iG)\,t^i=\sum_{i=0}^{r-1}c_it^i=\left(\sum_{i=0}^{r-1}c_{i,1}t^i;\sum_{i=0}^{r-1}c_{i,2}t^i;\ldots;\sum_{i=0}^{r-1}c_{i,n}t^i\right)\cong(s_1,s_2,\ldots,s_n)$$

This shows that the transmitted words using interleaved linear codes corresponds exactly to a codeword of $\mathcal{C}$. Moreover the weight of $(s_1,\ldots,s_n)$ as defined above is the same as the Hamming weight of $(\sum_{i=0}^{r-1}c_{i,1}t^i;\sum_{i=0}^{r-1}c_{i,2}t^i;\ldots;\sum_{i=0}^{r-1}c_{i,n}t^i)$. ∎

	2	3	4	5	6
7	1.0	1.0	1.0	1.0	1.0
8	0.96	0.98	0.99	0.99	0.99
9	0.81	0.94	0.96	0.97	0.98
10	0.49	0.80	0.88	0.91	0.91
11	0.0073	0.53	0.70	0.75	0.78
12	0.00012	0.14	0.38	0.48	0.53

Fig. 1. Probability of decoding more errors for a Goppa code of parameters $[256, 200, 15]_{\mathbb{F}_2}$.

	3	4	5	6
22	1.00000	1.00000	1.00000	1.00000
23	0.999997	0.999999	0.999999	0.999999
24	0.999972	0.999994	0.999997	0.999998
25	0.999844	0.999963	0.999981	0.999987
26	0.999384	0.999840	0.999917	0.999940
27	0.998099	0.999469	0.999715	0.999789
28	0.995114	0.998531	0.999185	0.999391
29	0.989079	0.996477	0.997984	0.998470
30	0.978112	0.992458	0.995554	0.996581

Fig. 2. Probability of decoding more errors for a Extended BCH code of parameters $[256, 100, 46]_{\mathbb{F}_2}$.

Theorem 15. *Given a linear code $\mathcal{C}'$ over $\mathbb{F}_q$ of parameters $[n, k, d]_{\mathbb{F}_q}$ and a unique decoding algorithm* `BlackBoxErasuresDec` *from errors and erasures that can correct ϵ erasures and $\tau(\epsilon)$ errors with at most $\mathsf{Dec}(\mathcal{C}')$ arithmetic operations over $\mathbb{F}_q$, there exists a unique decoding algorithm for interleaved codes with respect to $\mathcal{C}'$ and of degree r from errors and erasures that can correct ϵ erasures and $\tau(\epsilon)$ errors with at most $r\mathsf{Dec}(\mathcal{C}')$ arithmetic operations over $\mathbb{F}_q$. Moreover it can correct errors of weight at most $w > \tau(\epsilon)$ with probability at least $P(\epsilon, B, w)$ where P is defined by equation* (6) *with also at most $r\mathsf{Dec}(\mathcal{C}')$ arithmetic operations over $\mathbb{F}_q$.*

Proof: As $G = G'$ there is no need to lift a codeword from $\mathcal{C}'$ into $\mathcal{C}$ and the complexities is a consequence of Proposition 8. The existence of both algorithm is ensured by Proposition 14 and Proposition 11. ■

In Tables 1 and 2, the first row gives the degrees of interleaving and the first column shows the number of errors up to which we want to decode. The second row corresponds to half the minimum distance and, as expected, all the probabilities are 1.0. We can see that the probability increases when de degree of interleaving is increasing and that codes with a high minimal distance are good candidates for interleaving.

V. Conclusion

In this paper we designed a decoding algorithm based on a lifting decoding scheme. It allowed us to obtaine a unique decoding algorithm for RS codes over Galois rings with a very low complexity. We also applied this scheme to get a list decoding algorithm for RS code over Galois rings. We then show that using erasures at appropriate positions in the proposed algorithms allows to decode more errors than half the minimum distance. Finally we applied these techniques to decode interleaved linear codes over a finite field and get a decoding algorithm that can decode more errors than half the minimum distance.

Acknowledgment

The author would like to thank Daniel Augot for his precious advices and readings of this article and Grégoire Lecerf for his careful readings of this article.

References

[1] M. A. Armand, "Improved list decoding of generalized Reed-Solomon and alternant codes over rings," in *IEEE International Symposium on Information Theory 2004 (ISIT 2004)*, 2004, p. 384.

[2] ——, "List decoding of generalized Reed-Solomon codes over commutative rings," *IEEE Trans. Inform. Theory*, vol. 51, no. 1, pp. 411–419, 2005.

[3] M. Armand, "Improved list decoding of generalized reed-solomon and alternant codes over galois rings," *IEEE Trans. Inform. Theory*, vol. 51, no. 2, pp. 728–733, feb 2005.

[4] M. Armand and O. de Taisne, "Multistage list decoding of generalized reed-solomon codes over galois rings," *Communications Letters, IEEE*, vol. 9, no. 7, pp. 625–627, jul 2005.

[5] N. Babu and K.-H. Zimmermann, "Decoding of linear codes over galois rings," *Information Theory, IEEE Transactions on*, vol. 47, no. 4, pp. 1599–1603, may 2001.

[6] D. Bleichenbacher, A. Kiayias, and M. Yung, "Decoding of interleaved reed solomon codes over noisy data," in *Automata, Languages and Programming*, ser. Lecture Notes in Computer Science, J. Baeten, J. Lenstra, J. Parrow, and G. Woeginger, Eds. Springer Berlin / Heidelberg, 2003, vol. 2719, pp. 188–188.

[7] E. Byrne, "Lifting decoding schemes over a galois ring," in *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, ser. Lecture Notes in Computer Science, S. Boztas and I. Shparlinski, Eds. Springer Berlin / Heidelberg, 2001, vol. 2227, pp. 323–332.

[8] D. Coppersmith and M. Sudan, "Reconstructing curves in three (and higher) dimensional space from noisy data," in *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, ser. STOC '03. New York, NY, USA: ACM, 2003, pp. 136–142.

[9] S. Gao, "A new algorithm for decoding Reed-Solomon codes," in *Communications, Information and Network Security, V. Bhargava, H.V. Poor, V. Tarokh, and S. Yoon*. Kluwer, 2002, pp. 55–68.

[10] J. v. z. Gathen and J. Gerhard, *Modern computer algebra*, 2nd ed. Cambridge University Press, 2003.

[11] M. Greferath and U. Vellbinger, "Efficient decoding of $\mathbb{Z}_{p^k}$-linear codes," *IEEE Trans. Inform. Theory*, vol. 44, no. 3, pp. 1288–1291, may 1998.

[12] V. Guruswami, *List decoding of error-correcting codes: winning thesis of the 2002 ACM doctoral dissertation competition*, ser. Lecture Notes in Computer Science. Springer, 2004.

[13] V. Guruswami and M. Sudan, "Improved decoding of Reed-Solomon and algebraic-geometric codes," *IEEE Trans. Inform. Theory*, vol. 45, pp. 1757–1767, 1998.

[14] J. Justesen, "On the complexity of decoding Reed-Solomon codes (corresp.)," *IEEE Trans. Inform. Theory*, vol. 22, no. 2, pp. 237–238, Mar. 1976.

[15] M. Lazard, "Graduations, filtrations, valuations," *Publications Mathématiques de L'IHÉS*, vol. 26, pp. 15–43, 1965.

[16] R. Roth, *Introduction to coding theory*. Cambridge University Press, 2006.

[17] M. Sudan, "Decoding Reed-Solomon codes beyond the error-correction diameter," in *the 35th Annual Allerton Conference on Communication, Control and Computing*, 1997, pp. 215–224.

[18] S. Wicker and V. Bhargava, *Reed-Solomon Codes and Their Applications*. John Wiley & Sons, 1999.