On Efficient Decoding of Alternant Codes over a Commutative Ring [⋆]

Graham H. Norton and Ana Sălăgean

Algebraic Coding Research Group, Centre for Communications Research
University of Bristol, U.K.

1 Introduction

Let R be a commutative ring e.g. the domain of p-adic integers or a Galois ring. We define alternant codes over R, which includes BCH and Reed-Solomon codes. We also define a corresponding key equation and concentrate on decoding alternant codes when R is a domain or a local ring. Our approach is based on minimal realization (MR) of a finite sequence [4, 5], which is related to rational approximation and shortest linear recurrences. The resulting algorithms have quadratic complexity.

When R is a domain, the error-locator polynomial is the unique monic minimal polynomial of the finite syndrome sequence (Theorem 2), and can be easily obtained using Algorithm MR of [4] (which is division-free). The error locations and magnitudes can then be computed as over a field. In this way we can efficiently decode any alternant code over a domain.

Recall that a Hensel ring is a local ring which admits Hensel lifting. (It is well-known that a finite local ring, such as a Galois ring, is a Hensel ring.) We characterize the *set* of monic minimal polynomials of a finite syndrome sequence over a Hensel ring (Theorem 3). It turns out that the monic minimal polynomials coincide modulo the maximal ideal M of R (Theorem 4) when R is a local ring. This yields an efficient new decoding algorithm (Algorithm 1) for alternant codes over a local ring R, once a monic minimal polynomial of the syndrome sequence is known. For determining the error locations, it is enough to find the roots of the image of any such monic minimal polynomial in the residue field R/M. After determining the error locations, the error magnitudes can be easily computed.

When R is a finite chain ring (e.g. a Galois ring) we invoke Algorithm MP of [5] to find a monic minimal polynomial.

We note that a modification of the Berlekamp-Massey algorithm for $\mathbb{Z}_m$ was given in [8], where it was claimed [*loc. cit.*, Introduction] (without proof) to decode BCH codes defined over the integers modulo m. An algorithm to decode

[⋆] Research supported in part by the U.K. Engineering and Physical Sciences Research Council under Grant L07680. Copyright Springer-Verlag Berlin Heidelberg 1999. Current addresses: G. H. Norton, Department of Mathematics, University of Queensland, Brisbane 4072, Australia, ghn@maths.uq.edu.au; A. Salagean, Department of Computer Science, Loughborough University, Leicestershire LE11 3TU, UK, a.m.salagean@lboro.ac.uk.

BCH and Reed-Solomon codes over a Galois ring has also been given in [3]. However this algorithm may require some searching see [*loc. cit.*, Conclusions, p. 1019] and their decoding algorithm requires root-finding in R itself, which is also less efficient.

For more details and proofs, we refer the reader to [7].

2 Alternant Codes over a Commutative Ring

Let R be a commutative ring with $1 \neq 0$ and let $N(R)$ denote the subset of R consisting of all elements which are *not* zero-divisors.

The following definition of alternant codes over R generalises the definition over fields.

Definition 1 (Alternant codes). *Let T be a subring of R and $d \geq 2$. Suppose that $\alpha = (\alpha_1 \ldots, \alpha_n)$ and $y = (y_1, \ldots, y_n)$ are such that $\alpha_i, y_i \in N(R)$ and $\alpha_i - \alpha_j \in N(R)$ for $1 \leq i < j \leq n$. If*

$$H = \begin{bmatrix} y_1 & y_2 & \cdots y_n \\ y_1\alpha_1 & y_2\alpha_2 & \cdots y_n\alpha_n \\ y_1\alpha_1^2 & y_2\alpha_2^2 & \cdots y_n\alpha_n^2 \\ \vdots & \vdots & \vdots \\ y_1\alpha_1^{d-2} & y_2\alpha_2^{d-2} & \cdots y_n\alpha_n^{d-2} \end{bmatrix} \tag{1}$$

then the alternant code of length n and alphabet T defined by H is the T-module

$$\mathcal{A}(\alpha, y, d) = \{c \in T^n : \ Hc^{\text{tr}} = 0\}.$$

As usual, H is called the parity check matrix of $\mathcal{A}(\alpha, y, d)$.

As in the case of fields, we have:

Theorem 1. *The minimum Hamming distance of $\mathcal{A}(\alpha, y, d)$ is at least d.*

3 A key equation

For decoding alternant codes over a ring we follow the main steps for their algebraic decoding over a finite field, except that we rely on minimal realization of a finite sequence which was introduced in [4]. For some advantages of the minimal realization approach, see [5, Introduction]. See also the expository account in [6], especially *loc. cit.* Section 8, which discusses the application to a finite sequence of syndromes over a finite field.

Suppose that a codeword $c \in \mathcal{A}(\alpha, y, d)$ is received as $r = c + e$. We have to find the error vector e given the syndrome vector $Hr^{\text{tr}} = He^{\text{tr}}$.

We will henceforth assume that $d = 2t + 1 \geq 3$ and that the number of errors is $w = wt_H(e) \leq t$. Let $i_1, i_2, \ldots, i_w$ be the positions of the errors. As usual, $\alpha_{i_1}, \ldots, \alpha_{i_w}$ are called the error locations and $e_{i_1}, \ldots, e_{i_w}$ the error magnitudes. We write m for $1 - 2t$; note that $m \leq -1$.

Definition 2 (Syndrome sequence). *The syndrome sequence of the error e is the finite sequence $s_0, s_{-1}, \dots, s_m$ over R, denoted $s|m$ and defined by:*

$$s_i = \sum_{k=1}^{n} e_k y_k \alpha_k^{-i} = \sum_{j=1}^{w} e_{i_j} y_{i_j} \alpha_{i_j}^{-i}.$$

for $i = 0, -1, \dots, m$.

Definition 3 (Error polynomials). *We define the error-locator and error-evaluator polynomials by*

$$\sigma_e = \prod_{j=1}^{w} (X - \alpha_{i_j}) \text{ and } \omega_e = \sum_{j=1}^{w} e_{i_j} y_{i_j} \prod_{\substack{k=1,\dots,w \\ k \neq j}} (X - \alpha_{i_k}).$$

Note that in the classical literature σ_e^* and $X^{\deg(\sigma_e)-1-\deg(\omega_e)}\omega_e^*$ are called the error-locator and the error-evaluator polynomial respectively, where f^* denotes the reciprocal of $f \in R[X]$.

Definition 4 (Key equation). *Let $\Gamma = \sum_{i=m}^{0} s_i X^i \in R[X^{-1}]$. We say that $(f, h) \in R[X] \times XR[X]$ is a solution of the key equation if f is monic,* $\deg(h) \leq \deg(f) \leq -m$ *and*

$$\Gamma \equiv h/f \bmod X^{m-1}. \tag{2}$$

A solution (f, h) is called minimal *if* $\deg(f)$ *is minimal.*

As in the classical case we easily obtain:

Proposition 1. *If $w \leq t$ then $(\sigma_e, X\omega_e)$ is a solution of the key equation.*

The minimality of the solution $(\sigma_e, X\omega_e)$ is not obvious, but will follow from Theorem 2 when R is a domain and from Theorem 4 when R is local.

We now recall some definitions from [5]. For $f \in R[X]$ and $G \in R[X^{-1}]$, $f \cdot G$ denotes their product in $R[X^{-1}, X]$ and $(f \cdot G)_j$ is the coefficient of X^j in $f \cdot G$. We write $\mathrm{lc}(f)$ for the leading coefficient of $f \in R[X] \setminus \{0\}$.

Definition 5. *([5]) Let $r \in R \setminus \{0\}$. The r-annihilator set of $s|m$ is*

$$\mathrm{Ann}(s|m, r) = \{f \ : \ \mathrm{lc}(f) = r,\ (f \cdot \Gamma)_j = 0 \textit{ for } m + \deg(f) \leq j \leq 0\}.$$

A polynomial f is said to be an *annihilating polynomial of the sequence $s|m$* if $f \in \mathrm{Ann}(s|m, r)$ for some r.

A non-zero polynomial in $\mathrm{Ann}(s|m, r)$ of minimal degree is called *a minimal polynomial of the sequence $s|m$*, and we write $\mathrm{Min}(s|m, r)$ for those minimal polynomials of $s|m$ with leading coefficient r. (For the equivalence between minimal polynomials and shortest linear recurrences of a finite sequence, see [6, Corollary 2.3], which is valid for any R.)

Recall from [4] that for $f \in R[X]$, $\beta(f, s|m) \in XR[X]$ is defined by

$$\beta(f, s|m) = \sum_{j=1}^{\deg(f)} (f \cdot \Gamma)_j X^j.$$

The connection between the key equation and minimal polynomials of $s|m$ becomes clear from the following lemma:

Lemma 1. *The pair $(f, h) \in R[X] \times XR[X]$ is a minimal solution of the key equation (2) if and only if $\deg(f) \leq -m$, $f \in \mathrm{Min}(s|m, 1)$ and $h = \beta(f, s|m)$.*

4 Decoding over a Domain

Theorem 2. *If R is a domain then for all $r \in R \setminus \{0\}$, $\mathrm{Min}(s|m, r) = \{r\sigma_e\}$.*

We can now develop a decoding algorithm for alternant codes over a domain. Algorithm MR of [4] computes a minimal polynomial f and the corresponding $\beta(f, s|m)$ for *any sequence $s|m$ over a domain.* But from Theorem 2, we know that for a syndrome sequence, such a polynomial f must be the error locator polynomial multiplied by some non-zero constant. Hence, after applying Algorithm MR to the sequence of syndromes, we simply divide the output polynomials f and $\beta(f, s|m)$ by the leading coefficient of f, thus obtaining σ_e and $X\omega_e$. The algorithm has quadratic complexity. We then proceed as in the classical (field) case: we compute the error locations as the roots of σ_e (which are of the form $\alpha_{i_1}, \ldots, \alpha_{i_w}$) and the error magnitudes as $e_{i_j} = \omega_e(\alpha_{i_j})/(\sigma_e'(\alpha_{i_j}) y_{i_j})$.

This algorithm can decode, in particular, BCH and Reed-Solomon codes over the p-adic integers of [1].

5 Decoding over a Local Ring

We now assume that R is a local ring with maximal ideal M and residue field $K = R/M$. We extend the canonical projection $R \to K$ to a projection $R[X] \to K[X]$ and denote the image of $f \in R[X]$ under this projection by $\overline{f}$.

When R is a Hensel ring we can characterize the monic minimal polynomials of the syndrome sequence:

Theorem 3. *If R is a Hensel ring and $\overline{\alpha}_1, \ldots, \overline{\alpha}_n$ are distinct then*

$$\mathrm{Min}(s|m, 1) = \left\{ \prod_{j=1}^{w} (X - \alpha_{i_j} - z_j) \ : \ z_j e_{i_j} y_{i_j} = 0 \textit{ for some } z_j \in R, j = 1, \ldots, w \right\}.$$

Our decoding algorithm is based on the following result:

Theorem 4. *If R is a local ring and $\overline{\alpha}_1, \ldots, \overline{\alpha}_n$ are distinct then $\sigma_e \in \mathrm{Min}(s|m, 1)$ and for any $\mu \in \mathrm{Min}(s|m, 1)$ we have*

$$\overline{\mu} = \overline{\sigma}_e = \prod_{j=1}^{w} (X - \overline{\alpha}_{i_j}).$$

We can now develop a decoding algorithm for alternant codes over a local ring, provided we have an algorithm that computes a monic minimal polynomial for $s|m$. The latter can be achieved for sequences of syndromes of BCH and Reed-Solomon codes over $\mathbb{Z}_{p^a}$ (see [3], [8]), over finite local commutative rings (see [2]) and for any sequence over a finite chain ring (see [5]). A method of computing the error once we have a monic minimal polynomial f is discussed in [2, 3]: (i) the roots of f in R are found and (ii) the ones that differ from some α_i by a zero-divisor are selected. Our method searches for the roots of $\overline{f} \in K[X]$ among $\overline{\alpha}_1, \ldots, \overline{\alpha}_n$ and is therefore more efficient.

Algorithm 1 (Decoding $\mathcal{A}(\alpha, y, d)$ over a local ring)

```
Input: $r = (r_1, \ldots, r_n)$ containing at most $t = (d-1)/2$ errors, where $t \geq 1$.
Output: $c = (c_1, \ldots, c_n)$, the nearest codeword.

0. Let $m = 1 - 2t$.
1. Compute the syndrome sequence $s|m$ as $(s_0\ s_{-1}\ \ldots\ s_m)^{\mathrm{tr}} = Hr^{\mathrm{tr}}$. If $s|m = (0, \ldots, 0)$, return $r$.
2. Compute a monic minimal polynomial $\mu$ for the sequence $s|m$.
3. Compute the roots $\overline{\alpha}_{i_1}, \ldots, \overline{\alpha}_{i_w}$ of $\overline{\mu}$ in $K$. Then the errors occurred at positions $i_1, \ldots, i_w$.
4. Compute $\sigma_e = \prod_{j=1}^{w} (X - \alpha_{i_j})$.
5. Compute $\sigma'_e$ and $\omega_e = \beta(\sigma_e, s|m)/X$.
6. Set $e = (0, \ldots, 0)$ and for $j = 1, \ldots, w$, put $e_{i_j} = \omega_e(\alpha_{i_j})/(\sigma'_e(\alpha_{i_j}) y_{i_j})$. Return $r - e$.
```

Algorithm 1 can decode, in particular, BCH and Reed-Solomon codes over Galois rings.

Acknowledgement. The authors gratefully acknowledge financial support from the U.K. Engineering and Physical Sciences Research Council (EPSRC). The second author was supported by EPSRC Grant L07680.

References

1. A. R. Calderbank and N. J. A. Sloane. Modular and p-adic codes. *Designs, Codes and Cryptography*, 6:21–35, 1995.
2. A. A. de Andrade and R. Palazzo, Jr. Construction and decoding of BCH codes over finite commutative rings. *Linear Algebra and its Applications*, 286:69–85, 1999.
3. J. C. Interlando, R. Palazzo, and M. Elia. On the decoding of Reed-Solomon and BCH codes over integer residue rings. *IEEE Trans. Inform. Theory*, 43(3):1013–1021, 1997.

4. G. H. Norton. On the minimal realizations of a finite sequence. *J. Symbolic Computation*, 20:93–115, 1995.
5. G. H. Norton. On minimal realization over a finite chain ring. *Designs, Codes and Cryptography*, 16:161–178, 1999.
6. G. H. Norton. On shortest linear recurrences. *J. Symbolic Computation*, 27:323–347, 1999.
7. G. H. Norton and A. Sălăgean. On the key equation over a commutative ring. *Designs, Codes and Cryptography*, 1999. To appear.
8. J. A. Reeds and N. J. A. Sloane. Shift-register synthesis (modulo m). *SIAM J. Computing*, 14:505–513, 1985.