SYNDROME DECODING OF CYCLIC CODES VIA NON-PRESERVING PERMUTATIONS

R. M. Campello de Souza

Deptº. de Eletrônica e Sistemas CT-UFPE, 50.741 Recife-PE

ABSTRACT

A new technique for the decoding of cyclic codes, which is based on the finite field Fourier transform, is introduced. The method can be characterised in general as a transform domain based type of syndrome decoding which makes use of code preserving as well as of non-preserving permutations. Though only binary cyclic codes are considered, the ideas presented can be easily extended to multilevel codes.

I. INTRODUCTION

The analysis, sinthesis and implementation of error control codes for digital communication systems through the guise of the finite field Fourier transform (FFFT) is now an established subject and has put the field of coding theory in a digital signal processing framework [1,2]. In particular, a few frequency domain decoding procedures have been devised which represent interesting options for the design of coded systems.
This paper introduces an improved version of the classical syndrome decoding method, in the sense that shortened syndrome tables are used, thus allowing the decoding of longer codes. The method represents a generalization of a recent work on the subject [3] and introduces the use of non-preserving permutations in the decoding process. It is based upon the partitioning, in the frequency domain, of the set of all correctable error patterns of weight up to t, that are associated with an (n, k, d) binary cyclic code.

An outline of the paper is as follows. In section II we review a few standard facts about the FFFT and some families of finite groups.

In section III we discuss the use of non-preserving permutations in the decoding process and a detailed example is shown to clarify the underlying concepts. The last section contains a summary of the main results presented.

II. PRELIMINARIES

Let $a = (a_0, \ldots, a_{n-1})$ be a vector with components in GF(q). Then its FFFT is the vector $A = (A_0, \ldots A_{n-1})$ with components in $GF(q^m)$, given by

$$A_j = \sum_{i=0}^{n-1} a_i \, \alpha^{ji} \qquad (1)$$

where α is an element of order n of $GF(q^m)$. Without loss of generality, we approach here the case q = 2 and $n = q^m - 1$. The FFFT pair, which is denoted by $(a_i) \leftrightarrow (A_j)$, is entirely analogous to a discrete Fourier transform (DFT) pair and, among the DFT properties which carry over to finite fields, two are particularly important, namely time-shift and scaling. The FFFT components A_j satisfy also the so-called chord properties, e.i., the inverse FFFT (a_i) is GF(q) valued if and only if $A_j^q = A_{jq}$, where indexes are to be considered modulo n [4]. These properties together with some basic facts from the theory of finite groups, which include the Pólya-Burnside technique [5] for counting equivalence classes, are the main elements in the work described here. To obtain a partition of the set of correctable error patterns (or syndromes), we consider the action of finite symmetry groups on that set. In particular, transitivity is a very desirable feature of the groups to be applied, since in this case the set can be partitioned into the minimum number of orbits, which implies a maximum reduction in storage requirements at the decoder. An important group for the purpose of syndrome classification is the group of proper rotations of a regular polygon with n sides [5]. This is a cyclic group of order n, denoted by C_n, generated by a planar rotation of $2\pi/n$ radians. A second group of relevance is the group $P(\ell,s)$ of scalar permutations

$$p_\ell(s): z_n \to z_n$$
$$i \to \ell^s i \pmod n$$

where Z_n denotes the set of integers modulo n. $P(\ell,s)$ is a group of order m, where m is the multiplicative order of ℓ modulo n. In any case the number of orbits induced by the action of the group G on the set S of syndromes, can be evaluated via Burnside's theorem [5] and is given by

$$N = \frac{1}{|G|} \sum_{g \in G} |Fix(g)| \qquad (2)$$

where Fix(g) is the set of elements of S that are fixed under the action of g. Alternatively, N can be obtained through the use of cycle index polynomials [6].

III. DECODING VIA NON-PRESERVING PERMUTATIONS

Given a partition of the set S of syndromes into orbits we need, from the point of view of decoding, to be able to link every orbit leader with any member of its orbit. Since the received vector r(x) contains all the information about the channel error vector e(x), we need a relationship between r(x) and the orbit to which e(x) belongs. Thus, considering cyclic partitions, if $E_j = \alpha^{rj}$ denotes the syndrome of e(x), then the syndromes of all its shifted versions e'(x) are [7]

$$E'_j = \alpha^{rj + ji_0} \qquad (3)$$

where we assume $j \in C_j$, the cyclotomic coset modulo n over GF(q) whose representative is j, and i_0 defines the shift between e(x) and e'(x). We call the integers

r_j and r'_j the syndrome identifiers and, from (3), they are related by

$$r'_j = r_j + ji_0 \pmod n \qquad (4)$$

which implies the orbit identifying condition

$$R(r'_j, r'_1) = R(r_j, r_1) \qquad (5)$$

where

$$R(r_j, r_k) \triangleq r_j - jk^{-1}r_k \pmod n \qquad (6) .$$

A similar line of thought can be developed for scalar permutations, which leads to the orbit identifying condition

$$r'_j = r_{j/\ell}{}^s \qquad (7)$$

where s is an integer in the interval $\bar{}1, m-1\bar{}$.

Let us examine in more detail the effect of the permutations $p_\ell(1)$ over the syndrome of a vector r(x) ; $p_\ell(1)$ acts on the set {0, 1, ..., n-1} by changing i into $\ell i \pmod n$, which means that in r(x), a polynomial of degree less than n, x^i changes to $x^{\ell i \pmod n}$. In standard permutation decoding one is always interested in finding permutations that will move the errors out of the information section of the codeword , since this makes the syndrome of r(x) equal to a permuted version of e(x) and by simply applying the inverse permutation the error vector can be estimated. The permutations used in such decoding methods must transform codewords into codewords, otherwise the syndrome of $r_{\ell i}(x)$ (the permuted version of r(x)) cannot be calculated.

In the context of cyclic codes over GF(q) the permutation that sends i into i + 1 (mod n) clearly preserves the code; also, if n and q are relatively prime , the permutation $P_q(1)$ sends codewords into codewords. However, what can be said about other values of ℓ besides q? Assuming that ℓ and n are relatively prime we can guarantee that $p_\ell(1)$ leaves the weight of a vector unchanged, thus modifying C into an equivalent code, but which of these values of ℓ preserves the code is a difficult question to be answered beforehand. Nevertheless, the primeness condition is enough to assure that under $p_\ell(1)$ a cyclic code C is mapped into an equivalent cyclic code C' $\bar{}8\bar{}$. This result is very important in the sense that it provides a way out of the problem of calculating the syndrome of $r_{\ell i}(x)$ as is shown by the following lemma:

Lemma 1 Let $$r(x) = \sum_{i=0}^{n-1} r_i x^i$$

be a polynomial over GF(q) and denote by $r_{\ell i}(x)$ the polynomial obtained from r(x) by applying the permutation

$$p_\ell(1): x^i \to x^{\ell i}, \ (\ell, n) = 1.$$

Then α^b is a root of r(x) if and only if $\alpha^{b/\ell}$ is a root of $r_{\ell i}(x)$, where exponents are to be considered modulo n.

Proof: By definition

$$r_{\ell i}(x) = r_0 + r_1 x^\ell + \ldots + r_{n-1} x^{\ell(n-1)}$$

$\therefore$

$$r_{\ell i}(\alpha^{b/\ell}) = r_0 + r_1\alpha^b + \ldots + r_{n-1}(\alpha^b)^{n-1} = r(\alpha^b).$$

QED

We now assume that $p_\ell(1)$ is a non-preserving permutation that is to be applied to r(x) and show how, with the help of lemma 1, we can calculate the syndrome of $r_{\ell i}(x)$. The form of syndrome we are considering employs the zeros of the code, which are the roots of its generator polynomial g(x). Letting α^j denote any such roots, we recall that

$$r(x) = c(x) + e(x)$$

from where, by applying $p_\ell(1)$, we obtain

$$r_{\ell i}(x) = c_{\ell i}(x) + e_{\ell i}(x)$$

But, as α^j is a root of c(x) then, from lemma 1, $\alpha^{j/\ell}$ is a root of $c_{\ell i}(x)$, so that

$$r_{\ell i}(\alpha^{j/\ell}) = c_{\ell i}(\alpha^{j/\ell}) + e_{\ell i}(\alpha^{j/\ell})$$

$\therefore$

$$r_{\ell i}(\alpha^{j/\ell}) = e_{\ell i}(\alpha^{j/\ell}) = E_j.$$

Therefore, all we have to do is to find the roots of the new generator polynomial via the lemma and these are the values to be used for calculating the syndrome of $r_{\ell i}(x)$.

Let us now examine how a non-preserving permutation $p_\ell(1)$ may be used to decode a cyclic code. The process involves essentially the same ideas as with preserving permutations, the only difference being every time $p_\ell(1)$ is applied to r(x) we need to make use of lemma 1 to calculate the syndrome of $r_{\ell i}(x)$. In general, this means that the set of orbit identifier numbers r_j may change, which implies that a different set of R functions is to be used in relation to $r_{\ell i}(x)$. In the worst case it will be necessary to use m distinct sets of R functions, one for each time that $p_\ell(1)$ is applied, where m is the multiplicative order of ℓ modulo n. However, this worst case condition very seldom arises and the successive sets of identifiers generated at every step usually have a very large intersection . With respect to the computational burden involved in the syndrome recalculations, they can be easily avoided if we remember that all the necessary information to find e(x) is contained in the syndrome of r(x) and the new syndromes will not add any new information to the problem. What they tell us, and this is of fundamental importance in the scheme, is to where in the orbit leaders we should look now (after $p_\ell(1)$ has been used) to make a comparison; that is, a possibly different R function should be used in the orbit identification and its identity is given by the new syndrome. To calculate it we observe that when we change i to $i\ell \pmod n$, the coset C_j is mapped into the one that contains $(j/\ell) \pmod n$; if we use directly the value $\alpha^{j/\ell}$ to calculate the syndrome of $r_{\ell i}(x)$ then, by lemma 1, we should get the same answer when evaluating r(x) at $x = \alpha^j$. On the other and, if we make the exponent of α be the representative of the coset that contains $(j/\ell) \pmod n$, then the syndrome of $r_{\ell i}(x)$ is going to be a power of the syndrome of r(x). Both approaches may be adopted and, since they basically trade circuit hardware against computational complexity, the choice between

them should take into consideration the decoder design priorities. In either case, once the orbit of e(x) is found, i_0 is computed through the r_j values and after a cyclic shift of i_0 places is applied to the orbit leader, an inverse permutation completes the decoding.

The main steps for decoding a cyclic code with the non-preserving permutation $p_\ell(1)$ can be summarised as follows.

(i) We first calculate the syndrome of r(x)

$$E'_j = r(\alpha^j) = \alpha^{r'_j}$$

and set s = 0

(ii) We then compare, for all orbit leaders, the R functions $R(r'_b, r'_a)$ and $R(r_b, r_a)$, where b and a designate the representatives of the cyclotomic cosets associated with the roots of $r_{\ell^s i}(x)$. If an equality is found then we have made an orbit identification. This means that we need to cyclically shift its leader by i_0 positions, where

$$i_0 \equiv a^{-1}(r'_a - r_a) \pmod n$$

and the error locations are

$$i'_j \equiv (i_j + i_0)\ell^{m-s} \pmod n$$

for $1 \leq j \leq w$, with $w \leq t$. However, if no equality is found, we increase the value of s by one and return to the beginning of (ii), etc.

The second step of the above algorithm can be executed in an alternative way as follows. Instead of using a set of identifiers given by the coset representatives, we consider the set $r_{j/\ell}$. This is equivalent to computing the syndrome of $r_{\ell^s i}(x)$ via the roots α^{j/ℓ^s} which, by lemma 1, is equal to the syndrome of r(x). Thus we may store all the m values of the R functions associated with the set $r_{j/\ell}s$, $0 \leq s \leq m-1$, for every orbit leader, comparing them with the value assumed by the set of R functions associated with r(x), since now only its syndrome is being calculated. One of these comparisons will produce a positive answer, and after that the procedure to identify i_0 and the error positions is the same as before.

To illustrate the above explanations, let us examine a simple example.

Example: We consider the (15, 7, 5) binary BCH code generated by

$$g(x) = 1 + x + x^2 + x^4 + x^8 .$$

We choose the value 7 for ℓ since $p_7(1)$ is a permutation that does not preserve this code; in fact it changes the zeros of the code (α^3 and α^7) into α and α^3. The semi-partition of S results in four equivalence classes; using the first approach described in (ii) the decoder needs to have the following information

ORBIT LEADERS	IDENTIFIERS	R FUNCTION	R FUNCTION
(i_1, i_2)	(r_1, r_3, r_7)	$R(r_3, r_1)$	$R(r_3, r_7)$
0	0, 0, 0	0	0
0, 1	4, 14, 9	2	8
0, 3	14, 7, 13	10	10
0, 5	10, $-\infty$, 10	$-\infty$	$-\infty$

By lemma 1 we can determine the effect of $p_7(1)$ on the zeros of the code and the corresponding cosets. The resulting changes are:

Roots, C_s	Roots, C_s	Roots, C_s	Roots, C_s
α^3, C_3	α^9, C_3	α^{12}, C_3	α^6, C_3
α^7, C_7	α^1, C_1	α^{13}, C_7	α^4, C_1

(each pair of cosets is connected by $p_7(1)$)

and we see that only two distinct pairs of cosets are involved; hence, just two R functions are needed (see table). The decoding of

$$r(x) = 1 + x^2 + x^4$$

is completed for s = 3 and (syndrome computation)

$$E'_1 = r_{7^3 i}(\alpha) = \alpha^2$$

$$E'_3 = r_{7^3 i}(\alpha^3) = \alpha^8$$

$$R(r'_3, r'_1) \equiv 2 \pmod{15}$$

Therefore, we find

$$i_0 = r'_1 - r_1 = 2 - 4 \equiv 13 \pmod{15};$$

the error positions are

$$i'_j = (i_j + i_0)\ell^{m-s} \pmod n$$

$$i'_1 = (0 + 13)7^{4-3} \pmod{15} = 1$$

and

$$i'_2 = (1 + 13)7^{4-3} \pmod{15} = 8$$

The estimated error vector is then

$$\hat{e}(x) = x + x^8$$

and r(x) is decoded as

$$\hat{c}(x) = r(x) - \hat{e}(x) = 1 + x + x^2 + x^4 + x^8.$$

IV. CONCLUSIONS

Decoding methods for error control codes which use syndrome look-up tables can be applied to any linear (n, k, d) code, resulting in minimum error probability. However, they become impractical to implement for large values of (n-k) due to memory constraints in the decoder.

By applying FFFT based techniques, a partitioning

of the set of all syndromes into equivalence classes is obtained, which implies a reduction in the required storage, thus allowing the decoding of longer codes . A modified syndrome look-up table decoding algorithm for cyclic codes is introduced, which is based upon the use of permutations, but has the interesting aspect that permutations which are not code preserving are also allowed. The properties of the FFFT are used to relate the syndrome of the permuted orbit leaders to the syndrome of the received, possibly erroneous code word. These ideas establish a family of syndrome look-up table type decoders whose complexity varies as a function of the type and number of permutations ap - plied.

V. REFERENCES

1. R. E. Blahut, Theory and Practice of Error Control Codes, Addison Wesley, 1983.
2. R. M. Campello de Souza, Algebraic Decoding and the Sampling Theorem, International Symposium on Information Theory, junho 19-24, Kobe, Japan, 1988.
3. R. M. Campello de Souza, Finite Transforms and the Decoding of Reed-Solomon Codes, International Symposium on Information and Coding Theory, Campinas, SP, 1987.
4. R. E. Blahut, Transform Techniques for Error - Control-Codes, IBM Journal of Research and Development, Vol. 23, Nº 3, pp. 299-315, May/1979.
5. W. J. Gilbert, Modern Algebra with Applications , John Wiley, 1976.
6. H. S. Stone, Discrete Mathematical Structures and Their Applications, SRA, 1972.
7. R. M. Campello de Souza, Transform Techiques for Channel Coding, PhD thesis, The Electrical Engineering Laboratories, University of Manchester, England, 1983.
8. V. Pless, Introduction to the Theory of Error-Correcting Codes, John Wiley, 1982.

Este trabalho recebeu apoio parcial do Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq.