FUNCTIONAL DIAGNOSABILITY AND RECOVERY FROM MASSIVE FAULTS IN DIGITAL SYSTEMS

A Report Submitted in Fulfillment of the Requirement
for Quarterly Progress Reports for
The Third Quarter May 17, 1970 - August 16, 1970,
The Fourth Quarter August 17, 1970 - November 16, 1970
and for the Final Report

NASA Contract NAS8-25158

Get DRF

by

Gernot Metze
Principal Investigator

Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
November 16, 1970

08

INTRODUCTION

The investigations outlined in the previous progress report have been carried out. Specifically, the JPL-STAR (1), the multiprocessor proposed by Koczela and Wang (2), and the Bell Telephone Systems ESS #2 (3) have been considered with respect to their fault recovery characteristics. This analysis has indicated definite strengths and weaknesses in each design and has suggested possible areas of improvement in the design of reliable spaceborn computers.

The three systems considered were all designed with long term reliability as a primary goal. The two spaceborn computers (1) (2) are intended to have a long (3 - 5 years) design life in the absence of manual repair. The switching system (3) is intended to have an even longer (> 40 years) design life with manual repair. While these two design goals are not identical, it will be seen that many properties relating to the recovery from faults are functions of the general configuration of the system rather than the specific details of construction. A comparison of computers designed under these diverse reliability criteria is thus justified so long as those properties related to configuration are emphasized.

Before discussing the strengths and weaknesses encountered, the general configuration of the three systems will be reviewed.

REVIEW OF JPL-STAR CONFIGURATION

The JPL-STAR (1) is a simple processor with hardware redundancy on a functional level. The computer includes a testing and repair processor (TARP) which continuously monitors conditions in other functional units of the computer and initiates repair when necessary. Three powered copies of the TARP are provided along with a number of unpowered spares. Any output of the TARP to the rest of the system is derived as the majority vote among the three powered copies. In the case of disagreement, that is, when the vote is not unanimous, the disagreeing element is turned off and one of the spares is powered to restore triple modular redundancy to the TARP.

Other functional units of the computer are operated singly with error detection hardware or in duplex with cross checking. In either case, the detection of an error by the TARP causes the program segment to be retried. If the error persists, the faulty unit is turned off and a spare is powered before retrying the program segment again. Read-write memory is so arranged that 2 or more copies of any parameter value may be written; thus critical values may be stored with higher redundancy for added protection. The error checking features include cross checking of duplex units, the use of arithmetic residue codes, or the use of other (m out of n) codes. The choice of checking procedure was determined by the intended use of the functional unit to be checked.

REVIEW OF MULTIPROCESSOR FEATURES

The multiprocessor proposed by Koczela and Wang (2) is a distributed processor configuration in which each processor can work under either local or global control. This leads to a very high degree of variability in the structure of the machine. Basically the computer is composed of a large number of identical cells each of which is a complete processor and memory. The cells are arranged into groups with separate communication buses connecting cells within a group and connecting groups. In each group, a single cell is designated as controller. Other cells within the group may then operate either independently or as slaves to the group controller cell for parallel computations. In a similar manner, one of the groups is chosen as executive. This group controls global communication and resource allocation. The processors can communicate with the bulk store and I-O devices either directly or through the intercell and intergroup bus structure.

The reference mentioned (3) does not give details of the proposed error detection scheme other than to specify that both hardware and software features are to be included. It is therefore assumed for the purposes of comparison that error checking provisions comparable to those of the JPL-STAR are to be implemented in hardware. In this case, since all cells are identical, any cell may be made group controller and any group may be made executive, thus as long as any good cells remain

it is possible to rearrange the system into a working configuration of reduced capability.

REVIEW OF ESS #2 CONFIGURATION

The Bell Telephone System's ESS #2 (3) consists of a full duplex central processor with separate maintenance center. In addition, peripheral equipment is multiplicated so that the loss of one peripheral unit will reduce system capabilities but will not cause a shutdown. The two central processors are normally operated in a parallel mode with one unit on-line and the other duplicating all its operations off-line. The outputs of the two processors are continuously compared by the maintenance center. A mismatch causes the execution of a recovery program which attempts to identify the faulty processor and place it off-line to await repair. Extensive diagnostic software is automatically applied to the faulty processor to aid with manual repair of the faulty processor. If a similar duplex system were operated in an environment where human intervention was not possible, the processors might be segmented with standby units available for each processor segment. These standby units would be switched under program control to restore the second processor. Since control rests entirely with the on-line processor in this system, separate timers are included in each processor. Either processor is capable of interrupting and seizing control of the system if the on-line processor becomes hung up because of a hardware or software error.

COMPARISON OF SYSTEMS CONSIDERED

The three systems considered may be compared with respect to several features relating to recovery from faults. Perhaps most fundamental to the process of fault recovery is the amount of "hard core" in a system, that is, that portion of the system which must be assumed to be operative at all times.

The "hard core" of the JPL-STAR is the TARP which monitors error checking circuits and provides for switching of failed units. The TARP itself is protected by triple modular redundancy on a unit level as well as a supply of unpowered spares which may be switched in to replace a failed active unit.

The ESS #2 in a similar manner has as its "hard core" the maintenance centers which might be protected by redundancy if the system were to operate in the absence of manual intervention. In addition to this maintenance center, at least one of the two control unit timers must be assumed to be operative.

In the case of the distributed multiprocessor considered, the identity of the "hard-core" is not nearly as obvious and perhaps the usual concept of "hard core" does not apply to a distributed multiprocessor. At any rate, a sufficient number of cells must be operative to be capable of deciding whether or not another cell giving an error indication has actually failed. This program has been considered previously in a more general form (4). The lack of an identifiable "hard core" seems to have

a significant effect on the ability of a system to tolerate massive faults. This point will be discussed more fully in the final section of this report.

Faults are probably most readily detected and located in the JPL-STAR and least readily detected and located in the distributed processor. It seems that a fixed configuration system has a definite advantage over a variable configuration system from the standpoint of ease of diagnosability.

The fixed configuration systems become inoperative if all copies of any one unit, or a majority of the copies in a majority-vote protected redundant system, become faulty. On the other hand, the variable configuration system becomes inoperative only when every cell fails. Until this time, the system continues to operate at a reduced capacity. This property, as well as that described in the previous paragraph is determined almost entirely by the system configuration rather than by the specific realization.

The process of recovery from failure follows a fixed pattern which is largely determined by hardware in the JPL-STAR and ESS #2 systems. On the other hand, fault recovery procedures would fall almost entirely into the domain of software in the distributed multiprocessor. Either system is likely to be acceptable for the recovery from single solid faults and from single intermittent faults, but they may differ widely in their response to massive faults. Specifically, the multiprocessor configuration seems much more likely to damage its own program

and data storage systems under such conditions. This topic is currently under investigation with respect to specific classes of massive faults.

The foregoing discussion outlines some of the primary aspects in which the three systems considered differ. The investigation has suggested several specific areas to be explored. On the other hand, similarities between the three systems have also led to an area of special interest. Each system uses some form of coding to ensure error-free transmission of data and to detect malfunctions. The provisions are perhaps most extensive in the JPL-STAR where several different codes are used in different segments of the system. The applicability of these coding techniques is discussed further in the next section of the report.

ERROR DETECTING CODES

The use of codes for the detection of errors has been considered from two distinct aspects. The use of arithmetic residue codes within the arithmetic unit of a processor is a common technique. Such codes are easily applied and are relatively efficient for the detection of those errors in coded words which result from single hardware faults. However, they are inefficient in the case of externally caused faults which result in many simultaneous component failures.

Many externally induced faults lead to unidirectional errors, i.e. errors which change some 0's to 1's or some 1's to 0's, but not both.

For example, failure in the power supply to a specific unit may cause all outputs of that unit to go to the 0 level. Special codes such as the "m out of n" code are useful for the detection of such unidirectional errors. A 2 out of 4 code is indeed used for instruction transmission in the JPL-STAR. The motivation in this case, however, does not seem to have been a result of the consideration of unidirectional errors. The properties of codes which detect unidirectional errors are currently being investigated (5).

Research on these topics has led to the discovery of new bounds on the minimum distance of even and odd weighted code vectors of cyclic codes of composite length $n = n_1 n_2$ with n_1 and n_2 relatively prime. Further, the possibility of decoding beyond the BCH bound was shown and an improvement in the decoding of binary cyclic product codes was obtained.

Results of this research are reported in the following three papers:

1. "On the Weight Structure of Cyclic Codes of Composite Length," submitted to the 4th Hawaii International Conference on System Sciences, January 1971.

2. "Decoding Beyond the BCH Bound," submitted to the 4th Hawaii International Conference on System Sciences, January 1971, and, in expanded form, to the IEEE Transactions on Information Theory.

3. "The Decoding of Binary Cyclic Product Codes," submitted to the IEEE Transactions on Communication Technology.

Copies of the abstracts, as well as of the expanded version of 2, are included in an appendix.

CURRENT INVESTIGATION

Since this study is largely exploratory in nature, we have thus far concentrated on the identification of specific areas of high reliability computers which are critical to the recovery from massive faults. It is not clear that massive faults behave in a well-defined statistical manner. For this reason, we have purposely avoided attempts to derive specific equations for the reliability of such systems. Such calculations have been carried out for systems whose failures are assumed to behave in a well-defined statistical manner (6) and may be possible after further study of the specific causes of massive faults in digital systems.

Massive faults in digital systems present a new aspect of fault detection and recovery. A system which is designed to handle single faults may well lose control in the presence of massive faults. This loss of control is likely to lead to "wild" transfers and improper storage references. Specifically, programs and data may be overwritten during the period immediately after the occurrence of a massive fault. The confusion resulting from such an overwrite would be disastrous to a planned mission. Steps need to be taken to prevent such occurrences or at least to detect them and recover from them if a system is to resume operation after a massive but transient fault.

Various solutions to the problem outlined are possible. For example, all program storage might be in read-only memory thus making

destruction of programs impossible. Data, however, could still be lost under this scheme. If the fault occurred during a period when the system was not involved in critical computations, the damaged portions of memory could probably be interrogated and corrected manually, i.e. by the astronaut in a manned mission or by low-speed communication from Earth in an unmanned mission.

The specific action desired during a massive fault is probably a function of when the fault occurs. One can afford to shut down the system and check it out carefully if the fault occurs in midflight but the computation must proceed as best it can in a docking or landing maneuver. This problem is under current investigation with respect to:

a. system configuration

b. time of occurrence of fault

c. possibility of manual interventions.

It has been suggested that a fixed configuration computer is best suited to long unmanned missions because the primary computational need occurs at the end of the mission (7). This may not be appropriate in systems which are expected to tolerate massive faults. Further consideration of the trade-offs available between fixed and variable configuration systems is needed.

As was mentioned previously, the existence of an identifiable "hard core" has a very definite effect on the ability to recover from massive faults. It is proposed to determine under what conditions such a definite "hard core" will exist in a variable configuration computer

and thus under what conditions the reconfigurability and graceful degradation of such a system could be most fully utilized. We suspect that some combination of fixed and variable configuration concepts is most likely to be tolerant of massive faults. Definite progress toward this goal has been made but further effort is needed.

REFERENCES

1. A. Avizienis, "An Experimental Self-Repairing Computer," Preprint Booklet E (Hardware 2), 1968 IFIP Congress, Edinburg, Scotland, pp. E29-E33, August 1968.

2. L. J. Koczela and G. Y. Wang, "The Design of a Highly Parallel Computer Organization," IEEE Trans. on Computers, Vol. C-18, No. 6, June 1969, pp. 520-529.

3. Various authors, Bell System Technical Journal, October 1969, entire issue.

4. F. Preparata, G. Metze, and R. T. Chien, "On the Connection Assignment Problem of Diagnosible Systems," IEEE Trans. on Electronic Computers, Vol. EC-16, No. 6, pp. 848-854, December 1967.

5. D. Anderson, Doctoral thesis research currently in progress.

6. F. Mathur, "Reliability Modeling and Analysis of a Dynamic TMR System Utilizing Standby Spares," Proceedings Seventh Annual Allerton Conference on Circuit and System Theory, 1969, pp. 243-252.

7. A. Avizienis, "Design of Fault Tolerant Computers," AFIPS Conference Proceedings, 31 (Fall Joint Computer Conference 1967), pp. 773-743.

ON THE WEIGHT STRUCTURE OF CYCLIC CODES OF COMPOSITE LENGTH*

Carlos R. P. Hartmann†
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign

Abstract

The problem of constructing cyclic product codes has been considered by Burton and Weldon and by Abramson. The factoring of cyclic codes by Assmus and Mattson and Goethals. Goethals found new lower bounds on the minimum weight of a subclass of cyclic codes of composite length $n = n_1 n_2$ with $GCD(n_1, n_2)^{**} = 1$. Kasami extended Goethals result. In Goethals and Kasami's work the factorization is applied to the polynomials obtained from the Mattson-Solomon formulation.

In this paper the factorization is applied directly to code vectors and more insight in the weight structure is revealed.

Many cyclic codes of composite length $n = n_1 n_2$ with $GCD(n_1, n_2) = 1$ besides being a subcode of product codes have among the roots of their generator polynomial some n_1^{th} and (or) some n_2^{th} roots of unity. This fact will be shown useful in obtaining new lower bounds for the even and odd weighted code vectors.

Another class of cyclic codes of composite length is introduced. These codes are similar to the cyclic product codes and will be called cyclic quasi-product codes. A better bound for the minimum distance of this class of codes can be obtained.

Some cyclic codes of composite length present the properties of both quasi-product codes and product codes. These properties will be proved useful in obtaining new lower bounds for the even and odd weighted code vectors.

Many of the new bounds for the even and odd weighted code vectors give us the actual weight.

A better bound for the minimum distance of odd weighted code vectors of binary cyclic codes of composite length $n = n_1 n_2$ is obtained. This bound is a generalization of Kasami's theorem.

---

*This work was supported by NAS 8-25158 and by GK-2339.

**GCD(a,b) denotes the greatest common divisor of a and b.

†Now with Systems and Information Science at Syracuse University, Syracuse, New York 13210.

DECODING BEYOND THE BCH BOUND*

Carlos R. P. Hartmann†
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign

Abstract

As is well known, the random error correction capability of a cyclic code is entirely specified by its minimum distance. If a cyclic code has minimum distance, d_{min}, then it is capable of correcting up to $[(d_{min}-1)/2]$** random errors within a code block of length n. Most of the known algebraic decoding methods are unable to correct more than $[(d_o-1)/2]$, where d_o stands for the BCH bound. As is well known many cyclic codes have minimum distance greater than the BCH bound, thus most of the decoding schemes have not fully utilized the error capability of their codes. A more general question concerns our abilities to decode more than $[(d_o-1)/2]$ errors even if $d_{min} = d_o$.

In this work two decoding algorithms to decode beyond the BCH bound are introduced.

For a t-error-correcting BCH code we know the syndromes ($S_j = \sum_i Y_i X_i^j$) S_1, $S_2,\ldots,S_{2t}$. Suppose now that t+s errors occur. Thus we need to know $S_1S_2\ldots S_{2t+2s}$ to be able to decode t+s errors.

In both algorithms equations (usually non-linear) in the unknown syndromes, which are necessary to decode t+s errors, are obtained.

The first algorithm is of general application. It is applicable to any cyclic code even if the code has $d_{min} = d_o$. This algorithm gives all possible solutions for only those that have their locations (X_i) as n^{th} roots of unity and their magnitudes (Y_i) in the ground field.

The second algorithm is of simpler implementation and is applicable to cyclic codes with multiple sets of consecutive roots of generator polynomial and which have $d_{min} \geq d_o+2$. Furthermore, for binary BCH codes up to the decoding of t+2 errors only a set of linear equations must be solved.

A comparison between these decoding algorithms and the existing ones is also given.

---

*This work was supported in part by NSF GK-2339 and in part by NAS 8-25158.

**[x] denotes the integer part of x.

†Now with Systems and Information Science at Syracuse University, Syracuse, New York 13210.

THE DECODING OF BINARY CYCLIC PRODUCT CODES*

David M. Choy, Member, IEEE
Carlos R. P. Hartmann
Coordinated Science Laboratory
University of Illinois
Urbana, Illinois

August, 1970

Abstract

Cyclic product codes have been well-known for its capability of unambiguous correction of both bursts and random errors. Its importance is highly increased as it enjoys the easy implementation of cyclic codes, and at the same time, its algebraic structure being able to be factorized into lower fields suggests a much simpler way of decoding. A cascade decoding technique was introduced by N. Abramson. It retains the great advantage of working in the subfields without going into the original extension field. But it fails to realize up to the actual minimum distance. Recently, majority-logic decoding technique was applied to cyclic product codes. However, at least one of the component codes is required to be L-step majority-logic decodable, and no result on the capacity of burst-error-correction has yet been obtained. In this paper, the cascade decoding technique is extended. Decoding schemes are

*This work was supported in part by NSF GK-2339 and in part by NAS 8-25158.

obtained to realize up to the actual minimum distance $d = d_1 d_2$ (i.e. correct $[(d-1)/2]$ random errors) for binary cyclic product codes with single- or double-error-correcting component codes, if the component codes can realize their own minimum distances d_1 and d_2. At the same time, bursts up to $B_1 = \max\{n_1 + 1, n_2 + 1\}$ and $B_2 = 2B_1$ respectively are corrected. A way to generalize the algorithm for multiple-error-correcting component codes is also proposed.

DECODING BEYOND THE BCH BOUND*

Carlos R. P. Hartmann†
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
Urbana, Illinois

Abstract

In this work two decoding algorithms to decode beyond the BCH bound are introduced. The first is of general application. It decodes cyclic codes beyond the BCH bound independent of their minimum distance. The second is of simpler implementation and is applicable to cyclic codes with multiple sets of consecutive roots of the generator polynomial and which have minimum distance at least two greater than the BCH bound. A comparison between these decoding algorithms and the existing ones is also given.

*This work was supported in part by the National Science Foundation under Grant No. GK-2339 and NAS 8-25158.

†Now with Systems and Information Science at Syracuse University, Syracuse, New York 13210.

DECODING BEYOND THE BCH BOUND

1. Introduction

As is well known, the random error correction capability of a cyclic code is entirely specified by its minimum distance. If a cyclic code has minimum distance, d_{min}, then it is capable of correcting up to $[\frac{d_{min}-1}{2}]$* random errors within a code block of length n. Most of the known algebraic decoding methods are unable to correct more than $[\frac{d_o-1}{2}]$, where d_o stands for the BCH bound. As is well known many cyclic codes have minimum distance greater than the BCH bound, thus most of the decoding schemes have not fully utilized the error capability of their codes. A more general question concerns our abilities to decode more than $[\frac{d_o-1}{2}]$ errors even if the minimum distance of the cyclic code is d_o. Among the known algorithms to decode beyond the BCH bound are those of Berlekamp [1] and Tzeng. [2]

The problem of decoding BCH codes can be considered as one of finding the error locations X_i and error values Y_i from the syndromes equations $S_j = \sum_i Y_i X_i^{m_o+j-1}$ $(j = 1,2,\ldots,2t)$. X_i is an n^{th} root of unity, where n is the length of the code and $Y_i \in GF(q)$. Berlekamp has shown that this is equivalent to the solution of the key equation [3]

$$(1+S(z))\sigma(z) \equiv \omega(z) \text{ modulo } z^{2t+1}$$

where

$$S(z) = \sum_{k=1}^{\infty} S_k z^k \qquad S_k \in GF(q^m)$$

$$\sigma(z) = \prod_i (1-X_i z) = 1 + \sigma_1 z + \ldots + \sigma_d z^d$$

---

*[x] denotes the integer part of x.

and

$$\omega(z) = \sigma(z) + \sum_i z\, X_i\, Y_i \prod_{j \neq i} (1 - X_j z)$$

An iterative algorithm was found by Berlekamp [3] to solve the key equation. An intermediate step of the solution can be written as the solution of the key equation

$$(1+S(z))\sigma^{(k)} \equiv \omega^{(k)} \text{ modulo } z^{k+1}$$

Given 2t successive syndrome terms $S_1, S_2, \ldots, S_{2t}$, the algorithm will terminate with $\sigma(z) = \sigma^{(2t)}$ and $\omega(z) = \omega^{(2t)}$. If there are no more than t errors in the received vector then $d \leq t$. All the d roots of $\sigma^{(2t)}$ will be n^{th} roots of unity. All error magnitudes given by $\omega^{(2t)}$ are in GF(q). Thus we have a successful decoding.

If there are more than t errors in the received vector then, in most cases, at least one of the following failures will occur:

(1) Not all the roots of $\sigma^{(2t)}$ will be n^{th} roots of unity.

(2) Not all the error magnitudes will be in GF(q).

Suppose t+s errors occur. The main plot of this paper is to find a set of equations (usually non-linear) in the unknown syndromes which are necessary to decode t+s errors. The solution of this set of equations will permit the decoding of t+s errors.

First let us introduce some notations and definitions. Let

$$\sigma^{(2t)} = 1 + \sigma_1^{(2t)} z + \ldots + \sigma_{d-1}^{(2t)} z^{d-1} + \sigma_d^{(2t)} z^d \quad (\sigma_d^{(2t)} \neq 0) \qquad d \leq t$$

be the polynomial obtained by the iterative decoding algorithm [3]. Thus

$$S_j + \sigma_1^{(2t)} S_{j-1} + \ldots + \sigma_{d-1}^{(2t)} S_{j-d+1} + \sigma_d^{(2t)} S_{j-d} = 0 \qquad (1)$$

$$d + 1 \leq j \leq 2t$$

Let us define

$$S_k^{(2t)} = -(\sigma_1^{(2t)} S_{k-1}^{(2t)} + \ldots + \sigma_{d-1}^{(2t)} S_{k+1-d}^{(2t)} + \sigma_d^{(2t)} S_{k-d}^{(2t)}) \qquad (2)$$

and

$$S_{-k}^{(2t)} = \frac{-1}{\sigma_d}(S_{d-k}^{(2t)} + \sigma_1^{(2t)} S_{d-k-1}^{(2t)} + \ldots + \sigma_{d-1}^{(2t)} S_{-k+1}^{(2t)}) \qquad (3)$$

where

$$S_k^{(2t)} = S_k \qquad 1 \leq k \leq 2t$$

2. General Decoding Scheme

By this algorithm all sets of values $(S_{2t+1}, S_{2t+2}, \ldots, S_{2t+2s})$ that are solutions of the set of non-linear equations given by the next theorem, will give us $\sigma^{(2t+2s)}$ and $\omega^{(2t+2s)}$ such that all the roots of $\sigma^{(2t+2s)}$ will be n^{th} roots of unity and all error magnitudes given by $\omega^{(2t+2s)}$ belong to GF(q).

The next theorem is a generalization of theorems 10.53 and 10.54 in Berlekamp [1].

Theorem 2.1: Let $S_j = \sum_{i=1}^{t} Y_i X_i^{\,j}$ with $Y_i \neq 0$, $X_i \neq 0$, $X_i \in GF(q^m)$ and $X_i \neq X_j$ for $i \neq j$ then

(a) $Y_i \in GF(q)$ iff $(S_j)^q = S_{jq}$ for $j = m_o' + kq^\theta \qquad (0 \leq k < t)$

(b) If $Y_i \in GF(q)$ then $X_i^{\,n} = 1$ iff $(S_j)^{q^\delta} = S_{jq^\delta + q^\gamma n}$ or $(S_j)^{q^\delta} = S_{jq^\delta - q^\gamma n}$ for $j = m_o + kq^\varphi \qquad (0 \leq k < t)$

where m_o', m_o, θ, γ, δ and φ are integers.

Proof:

(a) If $(S_j)^q = S_{jq}$ for $j = m_o' + kq^\theta \qquad (0 \leq k < t)$

then

$$\sum_{i=1}^{t} Y_i X_i^{\,jq}(1 - Y_i^{\,q-1}) = 0 \qquad j = m_o' + kq^\theta \qquad (0 \leq k < t)$$

Since $Y_i \neq 0$, $X_i \neq 0$, $X_i \neq X_j (i \neq j)$ and $GCD(q, q^m - 1)^* = 1$ the solution of the van der Monde determinant of this system of equations gives

$$Y_i^{\,q-1} = 1 \qquad 1 \leq i \leq t$$

If $Y_i \in GF(q)$ then $(S_j)^q = S_{jq}$ for any integer j.

(b) The proof is analogous to the proof of part (a).

Theorem 2.1 can also be useful in checking if all the roots of $\sigma^{(2t)}$ are n^{th} roots of unity and all error magnitudes given by $\omega^{(2t)}$ are in $GF(q)$. So for the binary systematic cyclic codes we check if $\sigma^{(2t)}$ divides $z^n + 1$ in

---

*GCD(a,b) stands for the greatest common divisor of a and b.

parallel with the Chien search [4]. If $\sigma^{(2t)}$ divides z^n+1 we apply the Chien search only to the information digits. If $\sigma^{(2t)}$ does not divide z^n+1 we stop or apply the decoding of more than t errors.

The conditions given by this theorem require, in general, much less computation than the conditions given by theorem 10.53 of Berlekamp [1]. Theorem 10.53 requires that all roots of $\sigma^{(2t)}$ will be n^{th} roots of unity iff $S_{n+k}^{(2t)} = S_k^{(2t)}$ for $k = 1,2,\ldots,D(2t)$, where $D(2t)$ is the degree of $\sigma^{(2t)}$.

Let V_n be a cyclic code of length n generated by $g(x) = \text{LCM}\{\prod_{i=1}^{2t} m_{m_o+i-1}^{(x)}\}^*$ $\in GF(q)[x]$. So we know $S_1,S_2,\ldots,S_{2t}$. Suppose the weight of the coset leader specified by this set of syndromes is t+s. By Berlekamp's iterative algorithm [3] we can find $\sigma^{(2t+2s)}$ such that $\sigma^{(2t+2s)}$ has no repeated roots, $\sigma^{(2t+2s)}$ satisfy Newton's identities up to S_{2t+2s} and each $\sigma_i^{(2t+2s)}$ $(1 \le i \le t+s)$ is a function of the unknown syndromes necessary to decode up to t+s errors. So if we call these unknown syndromes x_i $1 \le i \le M$ where $M \le 2s$, then by (2) and (3) we can find $S_{2t+2s+j}^{(2t+2s)}$ $(j \ge 1)$ as functions of x_i $(1 \le i \le M)$. Let

$$f_k(x_1,x_2,\ldots,x_M) = S_{jq^{\delta}\pm q^{\gamma}n} - (S_j)^{q^{\delta}} = 0 \qquad j = m_o+kq^{\varphi}$$

$$\tilde{f}_k(x_1,x_2,\ldots,x_M) = S_{iq} - (S_i)^q = 0 \qquad i = m_o'+kq^{\theta}$$

$$0 \le k \le t+s-1$$

---

*LCM{a(x),b(x)} stands for the least common multiple. $m_i(x)$ is the minimum function of β^i, where β is a primitive n^{th} root of unity.

By theorem 2.1 the common solution of this set of 2t+2s equations will give us $\sigma^{(2t+2s)}$ which divides z^n-1 and Y_i that belongs to GF(q). If there is no common solution more than t+s errors are present.

In general the solution of non-linear equations is difficult to obtain. But if the code is a binary BCH code generated by $g(x) = \text{LCM}\{\prod_{i=1}^{2t} m_i(x)\}$ and s = 1 (decoding of t+1 errors) the solutions of the non-linear system of equations are the roots of $f(x_1)$ where

$$f(x_1) = \text{GCD}(f_o(x_1), f_1(x_1), \ldots, f_t(x_1))$$

Example 2.2: Consider the 2-error-correcting BCH code of block length 31. Suppose $S_1 = \beta^{27}$ and $S_3 = \beta^{17}$ where β satisfies $\beta^5 + \beta^2 + 1 = 0$. By the Chien-Tzeng version of Berlekamp's algorithm [5] we obtain

$$\sigma^{(4)} = 1 + \beta^{27}z + \beta^{26}z^2$$

By applying (2) we obtain $(S_{10}^{(4)})^4 = \beta^{15}$ and $S_9^{(4)} = \beta^3$. Since $(S_{10}^{(4)})^4 \neq S_9^{(4)}$ more than 2 errors occur. Assume 3 errors and consider $S_5 = x_1$. By applying the iterative algorithm [5] 1 step further we obtain

$$\sigma^{(6)} = 1 + \beta^{27}z + (\beta^9 x_1 + \beta^{18})z^2 + (\beta^5 x_1 + \beta^3)z^3.$$

By theorem 2.1 we need to impose

(a) $(S_8^{(6)})^4 = S_{32-31}^{(6)} = S_1^{(6)}$ which is already satisfied.

(b) $(S_{10}^{(6)})^4 = S_{40-31}^{(6)} = S_9^{(6)}$ which leads to

$$f_1(x_1) = x_1^8 + \beta^{18}x_1^3 + \beta^{29}x_1^2 + \beta^9 x_1 + \beta^{27} = 0$$

(c) $(S_{12}^{(6)})^4 = S_{48-31}^{(6)} = S_{17}^{(6)}$ which leads to

$$f_2(x_1) = x_1^7 + \beta^{11}x_1^6 + \beta^{22}x_1^5 + \beta^{26}x_1^4 + \beta^{13}x_1^3 + \beta^{24}x_1^2 + \beta^{21}x_1 + \beta^{28} = 0$$

Thus

$$f(x_1) = GCD(f_1(x_1), f_2(x_1)) = (x_1 + \beta^3)(x_1 + \beta^{16})(x_1 + \beta^{22})(x_1 + \beta^{26})(x_1 + \beta^{28})$$

So we have 5 possible error vectors.

(1) $S_5 = \beta^3$ gives us $e_1(x) = x^{16} + x^{19} + x^{22}$

(2) $S_5 = \beta^{16}$ gives us $e_2(x) = 1 + x^{28} + x^{30}$

(3) $S_5 = \beta^{22}$ gives us $e_3(x) = x^5 + x^{10} + x^{29}$

(4) $S_5 = \beta^{26}$ gives us $e_4(x) = x^2 + x^8 + x^{23}$

(5) $S_5 = \beta^{28}$ gives us $e_5(x) = x^6 + x^{12} + x^{24}$

Example 2.3: (Berlekamp [1] p. 239) Consider the 5-error-correcting binary BCH code of block length 31. Suppose $S_1 = \beta^9$, $S_3 = \beta^3$, $S_5 = \beta^{15}$, $S_7 = \beta^{28}$, $S_9 = \beta^{27}$. By the iterative decoding algorithm we arrive at

$$\sigma^{(10)} = 1 + \beta^9 z + \beta^{17}z^2 + \beta^7 z^3 + \beta^7 z^4 + \beta^{20}z^5$$

By applying (2) we obtain $S_{13}^{(10)} = \alpha^{14}$ and $(S_{11}^{(10)})^4 = \alpha^{20}$. Since $S_{13}^{(10)} \neq (S_{11}^{(10)})^4$ more than 5 errors occur. Assume 6 errors and consider $S_{11} = x_1$. By applying the iterative algorithm [5] 1 step further we obtain

$$\sigma^{(12)} = 1 + \beta^9 z + (\beta^{14} + \beta^7 x_1)z^2 + (\beta^{20} + \beta^{16} x_1)z^3 + (\beta^5 + \beta^5 x_1)z^4 +$$

$$+ (\beta^{21} + \beta^2 x_1)z^5 + (\beta^{16} + \beta^{11} x_1)z^6$$

By theorem 2.1 we need to impose

(a) $(S_7^{(12)})^4 = S_{28-31}^{(12)} = S_{-3}^{(12)}$ which leads to

$$f_1(x_1) = x_1^3 + \beta^{18} x_1^2 + \beta^{15} x_1 + \beta^{16} = 0$$

(b) $(S_8^{(12)})^4 = S_{32-31}^{(12)} = S_1^{(12)}$ which is already satisfied.

(c) $(S_9^{(12)})^4 = S_{36-31}^{(12)} = S_5^{(12)}$ which is already satisfied.

(d) $(S_{10}^{(12)})^4 = S_{40-31}^{(12)} = S_9^{(12)}$ which is already satisfied.

(e) $(S_{11}^{(12)})^4 = S_{44-31}^{(12)} = S_{13}^{(12)}$ which leads to

$$f_2(x_1) = x_1^4 + \beta^7 x_1^2 + \beta^{18} x_1 + 1 = 0$$

(f) $(S_{12}^{(12)})^4 = S_{48-31}^{(12)} = S_{17}^{(12)}$ which leads to

$$f_3(x_1) = x_1^4 + \beta^{11} x_1^3 + \beta^{28} x_1^2 + \beta x_1 + \beta^{28} = 0$$

Thus

$$f(x) = \mathrm{GCD}(f_1(x_1), f_2(x_1), f_3(x_1)) = x_1 + \beta^{22}$$

So for $S_{11} = \beta^{22}$ we obtain

$$e(x) = 1 + x^{26} + x^{27} + x^{28} + x^{29} + x^{30}$$

Berlekamp's decoding algorithm for more than t errors [1] gives for this example a quartic equation which has 2 solutions over $GF(2^5)$ and only one is acceptable.

Since in example 2.3 (b), (c), (d) are always satisfied by $\sigma^{(10)}$ if deg $\sigma^{(10)} \leq 3$* then $\sigma^{(10)}$ divides $z^{31} + 1$. This suggests the next theorem.

Theorem 2.4: Let V_n be a binary cyclic code of length $n = 2^m-1$ generated by $g(x) = \text{LCM}\{\prod_{i=1}^{2t} m_i(x)\}$. Let δ, K be integers such that $1 \leq 1 + 2^\delta k \leq 2t$ for $0 \leq k < K$ and for each k $(0 \leq k < K)$ exist integers θ and φ such that $2^{m-\delta} + k = \theta 2^\varphi$ with $1 \leq \theta \leq 2t$. Thus if deg $\sigma^{(2t)} \leq K$ then $\sigma^{(2t)}$ divides $z^n + 1$.

Proof

For the binary case if deg $\sigma^{(2t)} \leq t$ then $(S_j^{(2t)})^{2^\varphi} = S_{j2^\varphi}^{(2t)}$ for any j. Thus if $1 \leq 1 + 2^\delta k \leq 2t$ and $2^{m-\delta} + k = \theta 2^\varphi$ (with $1 \leq \theta \leq 2t$) for $0 \leq k < K$ we will have

$$(S_{2^{m-\delta}+k}^{(2t)})^{2^\delta} = S_{2^m+2^\delta k-(2^m-1)} = S_{1+2^\delta k}^{(2t)} \qquad 0 \leq k < K$$

So if deg $\sigma^{(2t)} \leq K$ by theorem 2.1 $\sigma^{(2t)}$ divides z^n+1.

Q.E.D.

*deg $\sigma^{(2t)}$ stands for the degree of $\sigma^{(2t)}$

3. Decoding Scheme for Cyclic Codes with Multiple Sets of Consecutive Roots*

Suppose now that the generator $g(x)$ of the cyclic code V_n has among its roots besides $\beta^1, \beta^2, \ldots, \beta^{2t}$ the set of consecutive roots $\beta^{m_i}, \beta^{m_i+1}, \ldots, \beta^{m_i+t+s}$ $1 \le i \le M$. As in section 2, we can find $\sigma^{(2t+2s)}$. Then each Newton identity will give us one equation

$$\sum_{j=0}^{t+s} S_{m_i+t+s-j}\, \sigma_j^{(2t+2s)} = 0 \qquad \sigma_o^{(2t+2s)} = 1 \qquad 1 \le i \le M$$

that the unknown syndromes have to satisfy.

If the code is a binary BCH code and $s = 1$ or 2, then a linear system in the unknown syndromes is obtained. To be able to obtain an independent set of equations we should require $d_{min} \ge 2t+3$ for $s=1$ and $d_{min} \ge 2t+5$ for $s=2$.

3.1 Decoding of t+1 errors (s=1)

Let V_n be a binary cyclic code of length n generated by $g(x)$, and among the roots of $g(x)$ are $\beta^1, \beta^2, \ldots, \beta^{2t}, \beta^{m_1}, \beta^{m_1+1}, \ldots, \beta^{m_1+t+1}$. Suppose $d_{min} \ge d_o+2$.

Since we know $S_1, S_2, \ldots, S_{2t}$, by the iterative decoding algorithm [5] we can obtain

$$\sigma^{(2t)} = 1 + \sigma_1^{(2t)} z + \sigma_2^{(2t)} z^2 + \ldots + \sigma_t^{(2t)} z^t$$

Suppose t+1 errors occur, thus Chien search [4] will not succeed. Consider now S_{2t+1} as an unknown. By the decoding algorithm [5] we have

---

*Cyclic codes with multiple sets of consecutive roots of the generator polynomial were introduced at first in [6].

$$\sigma^{(2t+2)} = \sigma^{(2t+1)} = \sigma^{(2t)} + \Delta(2t)\Delta(n')^{-1} z^{2t-n'} \sigma^{(n')} \tag{2}$$

where $n' = 2j \quad (0 \le j \le t-1)$

$$\Delta(2t) = S_{2t+1} + \sigma_1^{(2t)} S_{2t} + \ldots + \sigma_t^{(2t)} S_{t+1} \tag{3}$$

$$\sigma^{(n')} = 1 + \sigma_1^{(n')} z + \sigma_2^{(n')} z^2 + \ldots + \sigma_k^{(n')} z^k \tag{4}$$

Let

$$\Delta = \Delta(2t)\Delta(n')^{-1} \tag{5}$$

By (2), (3), (4), (5) and Newton's identities we arrive at

$$\Delta A = B$$

where

$$A = \sum_{i=0}^{\deg \sigma^{(n')}} \sigma_i^{(n')} S_{m_1+n'+1-t-i} \qquad \sigma_o^{(n')} = 1$$

$$B = \sum_{i=0}^{\deg \sigma^{(2t)}} \sigma_i^{(2t)} S_{m_1+t+1-i} \qquad \sigma_o^{(2t)} = 1$$

Thus if we can find Δ we can find $\sigma^{(2t+2)}$.

3.2 Decoding of t+2 errors (s=2)

Let V_n be a binary cyclic code of length n generated by g(x), and among the roots of g(x) we have $\beta^1, \beta^2, \ldots, \beta^{2t}, \beta^{m_1}, \beta^{m_1+1}, \ldots, \beta^{m_1+t+2}, \beta^{m_2}, \beta^{m_2+1}, \ldots, \beta^{m_2+t+2}$. Suppose $d_{min} \geq d_o+4$. Thus we can obtain

$$\sigma^{(2t)} = 1 + \sigma_1^{(2t)} z + \ldots + \sigma_t^{(2t)} z^t$$

Suppose t+2 errors occur. In this case we need to inspect two cases:

(a) $\Delta(2t) = 0$

If $\Delta(2t) = 0$ then

$$\sigma^{(2t+2)} = \sigma^{(2t+1)} = \sigma^{(2t)}$$

and by a procedure analogous to that used in 3.1, we obtain

$$\sigma^{(2t+4)} = \sigma^{(2t+3)} = \sigma^{(2t)} + \Delta z^{2t+2-n'} \sigma^{(n')}$$

where

$$\Delta A_j = B_j \qquad j = 1,2$$

with

$$A_j = \sum_{i=0}^{\deg \sigma^{(n')}} \sigma_i^{(n')} S_{m_j+n'-t-i} \qquad \sigma_o^{(n')} = 1$$

$$B_j = \sum_{i=0}^{\deg \sigma^{(2t)}} \sigma_i^{(2t)} S_{m_j+t+2-i} \qquad \sigma_o^{(2t)} = 1$$

Now if $\Delta(2t) = 0$ then

$$\frac{B_1}{A_1} = \frac{B_2}{A_2} \tag{6}$$

and

$$\deg \sigma^{(2t+4)} = t+2 \tag{7}$$

So first we check (6) and (7), if both are satisfied then $\Delta(2t)$ can be zero, otherwise $\Delta(2t) \neq 0$.

(b) $\underline{\Delta(2t) \neq 0}$

If $\Delta(2t) \neq 0$ then

$$\sigma^{(2t+2)} = \sigma^{(2t+1)} = \sigma^{(2t)} + \Delta_1 z^{2t-n'} \sigma^{(n')} \tag{8}$$

$$\sigma^{(2t+4)} = \sigma^{(2t+3)} = \sigma^{(2t+2)} + \Delta_2 z^{2t+2-n''} \sigma^{(n'')} \tag{9}$$

where

$$n'' = n' \text{ if } n' - \deg \sigma^{(n')} \geq 2t - \deg \sigma^{(2t)}$$

$$n'' = 2t \text{ if } n' - \deg \sigma^{(n')} < 2t - \deg \sigma^{(2t)}$$

By (8), (9) and Newton's identities we obtain

$$\sum_{i=0}^{\deg \sigma^{(2t)}} \sigma_i^{(2t)} S_{m_j+t+2-i} + \Delta_1 \sum_{i=0}^{\deg \sigma^{(n')}} \sigma_i^{(n')} S_{m_j+n'+2-t-i} + \Delta_2 \sum_{i=0}^{\deg \sigma^{(n'')}} \sigma_i^{(n'')} S_{m_j+n''-t-i} = 0$$

So if for j=1 and 2 we can find two linearly independent equations then we can solve for Δ_1 and Δ_2 and obtain $\sigma^{(2t+4)}$.

Example 3.3:

Consider the binary reversible cyclic (31,11) code generated by $g(x) = m_1(x)m_3(x)m_7(x)m_{15}(x)$. By [6] $d_{min} \geq 9$. Let $S_1 = \beta^{12}$, $S_2 = \beta^{24}$, $S_3 = \beta^{20}$, $S_4 = \beta^{17}$, $S_{14} = \beta^{28}$, $S_{15} = \beta^{15}$, $S_{16} = \beta^{6}$, $S_{17} = \beta^{10}$, $S_{27} = \beta^{27}$, $S_{28} = \beta^{25}$, $S_{29} = \beta^{29}$ and $S_{30} = \beta^{30}$.

By using S_1, S_2, S_3 and S_4 we arrive at

$$\sigma^{(2)} = 1 + \beta^{12}z$$

and

$$\sigma^{(4)} = 1 + \beta^{12}z + \beta^{17}z^2$$

Since $S_{17} + \sigma_1^{(4)}S_{16} + \sigma_2^{(4)}S_{15} \neq 0$, then more than 2 errors occur. First we assume 3 errors and apply 3.1. So

$$\Delta = \frac{S_{17} + \sigma_1^{(4)}S_{16} + \sigma_2^{(4)}S_{15}}{S_{15} + \sigma_1^{(2)}S_{14}} = \beta^{30}$$

and

$$\sigma^{(6)} = 1 + \beta^{12}z + z^2 + \beta^{11}z^3$$

but since

$$S_{30} + \sigma_1^{(6)}S_{29} + \sigma_2^{(6)}S_{28} + \sigma_3^{(6)}S_{27} \neq 0$$

more than 3 errors are present. So we assume 4 errors and apply 3.2. If $\Delta(4) = 0$ then $\deg \sigma^{(8)} = 5$ which is impossible. Thus $\Delta(4) \neq 0$. For $\Delta(4) \neq 0$ we obtain $n'' = 4$ and the following system of equations:

$$\sum_{i=0}^{2} \sigma_i^{(4)} S_{31-i} + \Delta_1 \sum_{i=0}^{1} \sigma_i^{(2)} S_{29-i} + \Delta_2 \sum_{i=0}^{2} \sigma_i^{(4)} S_{29-i} = 0$$

$$\sum_{i=0}^{2} \sigma_i^{(4)} S_{32-i} + \Delta_1 \sum_{i=0}^{1} \sigma_i^{(2)} S_{30-i} + \Delta_2 \sum_{i=0}^{2} \sigma_i^{(4)} S_{30-i} = 0$$

From this system we obtain $\Delta_1 = \beta^2$ and $\Delta_2 = \beta^{23}$ by assuming $S_{31} = S_o = 0$. Thus

$$\sigma^{(8)} = 1 + \beta^{12} z + \beta^{21} z^2 + \beta^8 z^3 + \beta^9 z^4 = (z+1)(z+\beta^{30})(z+\beta^{28})(z+\beta^{26}).$$

and the error vector is

$$e(x) = 1 + x + x^3 + x^5$$

4. Conclusions

The possibility of decoding beyond the BCH bound has been shown. The general decoding algorithm in contrast with Berlekamp's method [1], gives all possible solutions for only those that have their locations as n^{th} roots of unity and their magnitudes in the ground field GF(q). The decoding of cyclic codes with multiple sets of consecutive roots and with $d_{min} \geq d_o + 2$ is

more general and requires less computation than Tzeng's method [2]. Furthermore, for binary BCH codes up to the decoding of t+2 errors only a set of linear equations must be solved.

ACKNOWLEDGMENT

The author would like to thank Professor R. T. Chien of the University of Illinois for his guidance and encouragement during the course of this research.

REFERENCES

[1] Berlekamp, E. R., Algebraic Coding Theory, McGraw-Hill, New York, 1968.

[2] Tzeng, K.K.M., "On Iterative Decoding of BCH Codes and Decoding Beyond the BCH Bound," Technical Report R-404, Coordinated Science Laboratory, University of Illinois, January, 1969.

[3] Berlekamp, E. R., "Non binary BCH Decoding," Institute of Statistics Mimeo Series No. 502, Department of Statistics, University of North Carolina, December, 1966.

[4] Chien, R. T., "Cyclic Decoding Procedures for the Bose-Chaudhuri-Hocquenghem Codes," IEEE Trans. on Information Theory, Vol. IT-10, pp. 357-363, October, 1964.

[5] Chien, R. T. and K.K.M. Tzeng, "On Iterative Decoding of BCH Codes," Proceedings of the Eleventh Midwest Symposium on Circuit Theory, pp. 599-600, May, 1968.

[6] Hartmann, C.R.P., "On the Minimum Distance Structure of Cyclic Codes and Decoding Beyond the BCH Bound," Technical Report R-458, Coordinated Science Laboratory, University of Illinois, February, 1970.