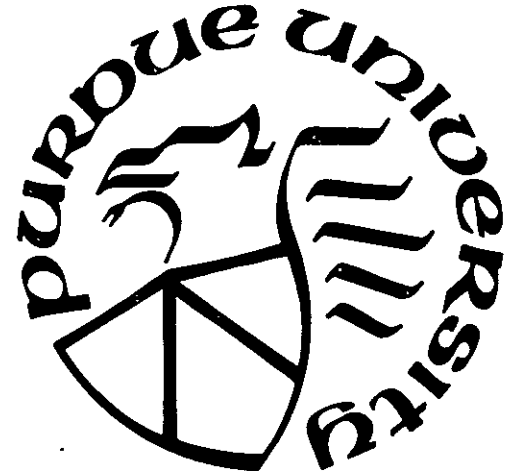
PURDUE UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING



Lafayette, Indiana 47907

CODES FOR PROTECTION FROM SYNCHRONIZATION LOSS AND ADDITIVE ERRORS

G. R. Redinbo

P. A. Wintz

TR-EE 69-43

November, 1969

School of Electrical Engineering
Purdue University
Lafayette, Indiana 47907

This work was partially supported by ARPA and Joint Services Electronics Program N00014-67-A-0226 modification AE, and by NASA grant NGR 15-005-087.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Cyclic codes are practical and efficient codes which protect against the effects of additive errors. However their effectiveness, like that of block codes, requires correct word synchronization at the decoder. Cyclic codes with symbols from a general finite field are modified so that they are also capable of protecting against misframing at the decoder. These codes are modified by altering their distance structure. There are a number of techniques which can be employed. Each method affects different aspects of the code's performance; therefore a complete and comprehensive coverage of all techniques is given.

Results for each modification approach are given for three types of protection from the simultaneous occurrence of additive errors and synchronization errors. The first type is the detection of some kind of error, the second is the detection and the classification of the nature of the error, and the third is the correction of both kinds of errors. Furthermore for each approach results are presented for the cases of symmetrical and unsymmetrical ranges of synchronization errors. The proofs of all results indicate the general strategy for decoding the modified code.

A coset of the original code allocates part of its error-protecting capabilities to synchronization. Results are given for the general class

of cyclic codes. Stronger conclusions are possible when the special case of Reed-Solomon codes is considered. In this case protection from slips of half the code's length in either direction are permitted.

A subset code is derived from a code by removing certain of its vectors so as to produce a code with fewer members which are less sensitive to misframing. Two approaches to subset codes are demonstrated. One is a coset code of an expurgated code while the other is a code with a fixed pattern imbedded in the information digits.

Changing the length of a code when combined with other techniques is another modification approach. The work here improves on the few known results and introduces many new ones so as to complete and consolidate all aspects of this type of approach. Results concerning shortened codes are developed, subset codes are extended to yield another modification approach, and coset codes are lengthened to produc a new scheme.

Two approaches for achieving wide-range slip protection are presented. One uses interleaving while the other combines interleaving with concatenation. With either technique slip protection ranges of half the code's length are possible. The interleaving technique may be coupled with any other approach giving the net effect of greatly expanding the slip protection range of that approach. Combining concatenation and interleaving accomplishes the same result without increasing the complexity of the encoder and decoder to the extent to which they would be if only interleaving were used. It is shown that for wide-range slip protection the error-protecting performance of either approach is superior to any other known approach.

CHAPTER 1

INTRODUCTION

A great deal of research has been devoted to the problem of designing efficient and practical schemes by which information can be coded for reliable transmission through communication channels which corrupt the message with noise. The general class of codes for which the most useful results and consequently the largest body of knowledge has been developed is a class whose members have fixed length, i.e., block codes. These results indicate that the more algebraic structure a class of codes possesses, the easier they are to implement.

Linear codes are a subclass of the block codes. A linear code is equivalent to a subspace of a vector space. The vector space is over a finite field with a prime or the power of a prime number of elements [1]. Linear codes are designed to protect against the types of errors caused by channel noise which are called substitution errors. A substitution error occurs whenever a symbol of the code is changed from its true value. Substitution errors and additive errors are equivalent because of the additive structure of a vector space.

A subclass of the linear codes is the cyclic codes. Cyclic codes have even more algebraic structure because in addition to being equivalent to a vector subspace they have the property that any cyclic permutation of the symbols of any code word is also a code word (closure under a shifting operation). Cyclic codes are practical because they

may be implemented by linear feedback shift registers (Chapter 8 [2]). Because of a cyclic code's easy implementation and structure it will be considered throughout the following work.

The Problem

Cyclic codes are used to combat the effects of additive errors introduced by a communication channel. However all the benefits are predicated upon the assumption that word synchronization is maintained; unfortunately this is not always true. In any communications system there is generally a hierarchy of synchronization levels. Carrier or "chip" synchronization in the modulation and demodulation processes is required in coherent systems. Symbol or bit synchronization is the next higher level. Finally the establishment of word or block synchronization is necessary. A general discussion of all these synchronization levels and their interconnection is contained in a paper by Golomb, et al. [3].

In this work it will be assumed that the lower levels of synchronization have been determined. Therefore the problem is to establish and maintain word synchronization even in the presence of additive errors. Loss of word synchronization at a receiver may result for a number of reasons. Timing inaccuracies or jitter in the clocking circuitry at any level of synchronization could propagate to the word synchronization level. The loss could occur at the start of transmission because the receiver generally must accomplish the synchronization levels in sequence with word synchronization being the last level. The receiver could be in synchronous operation and then lose synchronization because of the insertion or deletion of symbols in the incoming data stream. Two possible causes of this problem are the physical phenomena in the channel

of fading or multipath.

The net and lasting effect of any loss of word synchronization is equivalent to the sequence of words being misframed or slipped at the decoder. Of course this excludes the direct consideration of any word with insertions or deletions. However by investigating the framing of the preceding and succeeding words it is possible to determine the aggregate effect of insertions and deletions in a code word. The study of codes for the correction of insertion or deletion errors has been undertaken by several authors [4-7]. However the direction of the work to be presented here is to modify known error-protecting codes so that they are also capable of protecting against misframing or slip at the decoder. The problem is depicted below. The term synchronization

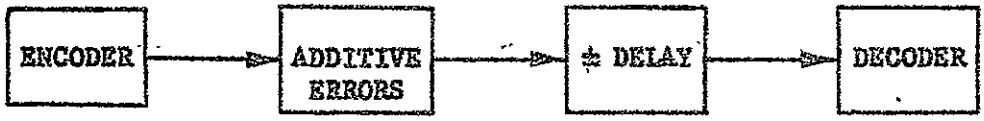


Figure 1.1 Illustration of the Problem

error will be synonymous with misframing.

This problem partially motivated the early work on comma-free codes [8-12]. Comma-free codes are codes which have the property that the misframing of any two adjacent code words cannot produce a code member. But all of this work discounted the effects of noise. It is unrealistic to ignore the effects of additive errors in the synchronization problem of codes which are designed to combat errors. However the work on the noiseless case did serve as a foundation for later work. Reference to other pertinent publications will be given at the

appropriate places in the body of this report. An excellent overview of the history of the work on this problem may be found in a book by Stiffler [13].

The results to be presented in the following chapters will be given in a very general setting because no particular type of channel noise will be assumed. The results will be applicable to any channel which may be modeled as one that introduces substitution errors. The codes which will be exhibited have the capability of protecting against the simultaneous occurrence of additive errors and symbol slippage in a given direction. The results will be given as the maximum number of each which may be protected. The work will deal with the modification of cyclic codes with symbols from a general finite field, GF(q).

There are a number of ways in which a given error-protecting code may be modified so as to give it sync-protecting capabilities also. However each method extracts a price in the form of a degradation in certain aspects of the original code's performance. One way to classify the various methods is according to the technique by which the code is altered. The results will be presented along this type of outline. The advantages of one technique in a set of circumstances may be disadvantages in another situation. Therefore a complete and comprehensive coverage of all methods will be given. The results for each modification approach will be concerned with three types of protection from the conjoint occurrence of additive errors and synchronization errors. The first will be the detection of some type of error, the second will be the detection and the classification of the type of error, and the third will be the correction of both types. Furthermore

results for each modification technique will be given for situations of symmetrical and unsymmetrical sync-protection ranges.

The design and construction of modified codes will be performed upon the basis of the distance structure of the original code. The proofs of all the results will not be simply existence proofs but will indicate the general strategy for decoding the modified codes.

Notation and Preliminaries

Vectors over a finite field, GF(q), (A Galois Field [14]) will be denoted by a letter from the English alphabet with a bar underneath it, e.g., $\underline{a}$. If the vector space has dimension n over GF(q), then every vector may be represented as an n-tuple, e.g., $\underline{a}=(\alpha_0,\alpha_1,\ldots,\alpha_{n-1})$ with $\alpha_i \in GF(q)$. The Hamming weight of a vector is defined as follows (pg. 204-205 [15]):

$$w(\underline{a}) = \sum_{i=0}^{n-1} w_H(\alpha_i)$$

$$w_H(\alpha_i) = \begin{cases} 0 & \text{if } \alpha_i = 0 \\ 1 & \text{if } \alpha_i \neq 0 \end{cases} \qquad (1.1)$$

The Hamming distance between any two vectors $\underline{a}$ and $\underline{b}$, $d(\underline{a},\underline{b})$ is defined in terms of the weight.

$$d(\underline{a},\underline{b}) = w(\underline{a}-\underline{b}) \qquad (1.2)$$

The Hamming distance is a metric on the vector space (pg. 10 [2]). Therefore a vector is the zero vector if and only if the Hamming weight of it is zero, i.e., $\underline{c}=0$ if and only if $w(\underline{c})=0$. This fact and the one

given below will be used in many of the proofs in the following work. If I is any subset of the set of integers $\{0,1,\ldots,n-1\}$, then the following inequality is true.

$$w(\underline{a}) \geq \sum_{i \in I} w_H(a_i) \tag{1.3}$$

It will be presumed that the reader is familiar with the fundamental properties of cyclic codes. There are a number of sources which may be consulted [2,15-17]. Every code vector of a cyclic code with length n has an equivalent representation as a polynomial in the residue class ring of polynomials modulo the polynomial (x^n-1). Thus the code word $\underline{b} = (\beta_0,\beta_1,\ldots,\beta_{n-1})$ may be represented as

$$b(x) = \beta_0 + \beta_1 x + \ldots + \beta_{n-1} x^{n-1} \text{ modulo } (x^n-1) \tag{1.4}$$

The same English letter with the same subscript will be used in each representation, e.g., $\underline{b}_i \Leftrightarrow \underline{b}_i(x)$.

The nature of the problem requires dealing with the misframing of code words. The following descriptive notation will be adopted. Let the $\{\underline{b}_0,\underline{b}_1,\ldots,\underline{b}_{M-1}\}$ be the code words of a block code of length n. For a positive slip s, $\underline{b}_{jk}^{(s)}$ is the vector whose first (n-s) components are the last (n-s) elements of $\underline{b}_j$ and whose last s components are the first s of $\underline{b}_k$. Whereas for a negative slip s, $\underline{b}_{jk}^{(s)}$ has the last s components of $\underline{b}_k$ in its first s places and the first (n-s) components of $\underline{b}_j$ in the remaining places. Figure 1.2 is an illustration of this notation. In many cases it will be necessary to consider the cyclic permutation of a vector $\underline{b}$. $\underline{b}^{(s)}$ will denote a cyclic shift of $\underline{b}$ to the right if s is negative or to the left if s is positive.

The results in the following chapters will be displayed using the following bracket notation. Let y be any real number.

$$[y] = \begin{cases} z & \text{if } y \geq 0 \\ \text{undefined} & \text{if } y < 0 \end{cases} \tag{1.5}$$

z is the smallest positive integer such that $z \leq y$.

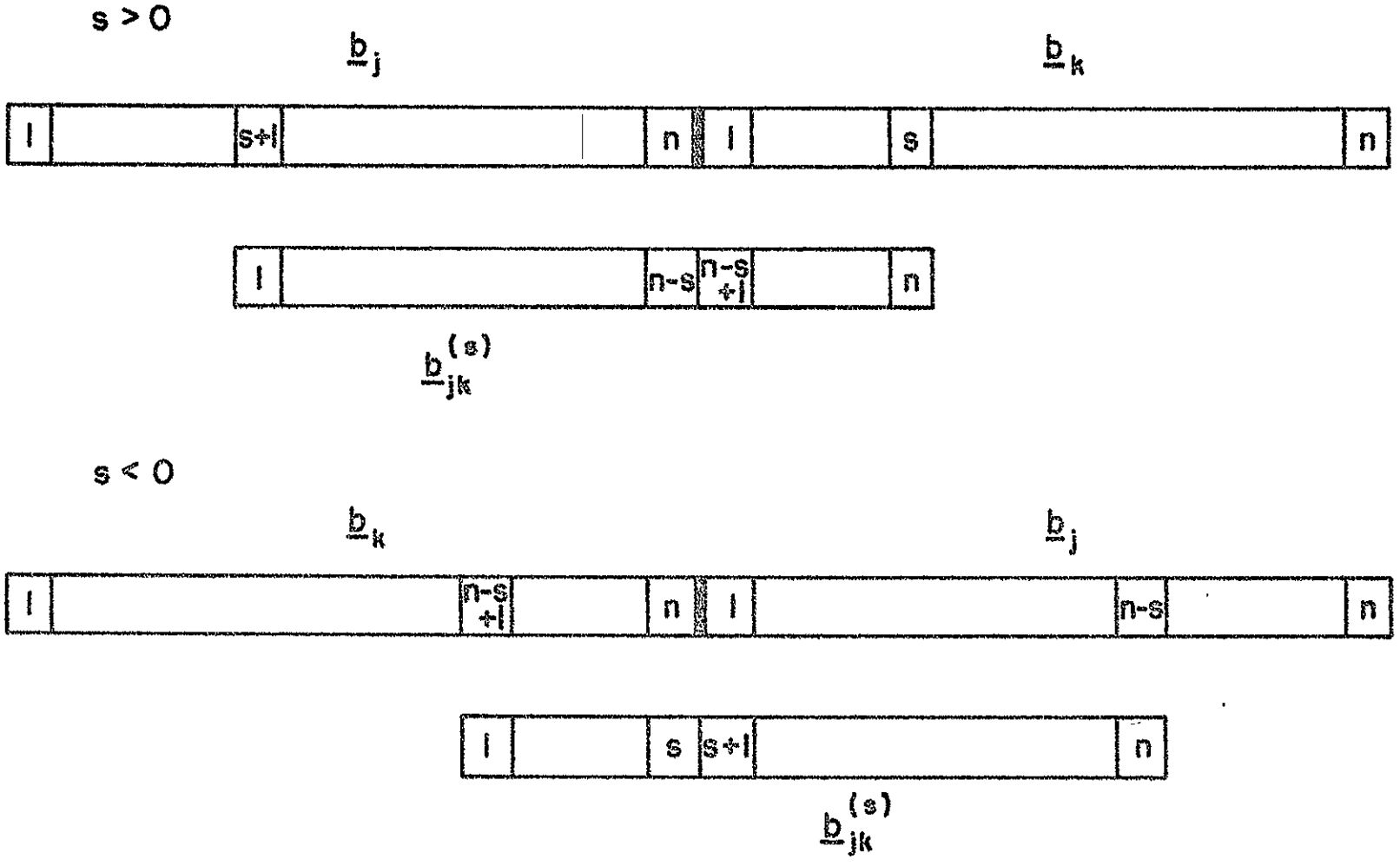


FIGURE I.2 . VISUALIZATION OF THE NOTATION.

CHAPTER 2

COSET CODES

The purpose of this chapter is to demonstrate one type of code design technique used to modify any cyclic code so that it has synchronization error-detecting or error-correcting capabilities in addition to its additive error-detecting or error-correcting abilities. This type is the coset code. A coset code is obtained from a linear code by the addition of a fixed vector to every code word. If $\{\underline{b}_i\}_{i=0}^{M-1}$, $M=q^k$, is an (n,k) linear code and $\underline{c}$ is any fixed vector in the same n-dimensional space, then $\{\underline{b}_i + \underline{c}\}_{i=0}^{M-1}$ is a coset code and $\underline{c}$ is called the coset generator. Obviously if $\underline{c}$ were a code vector, the resulting coset code would be the original code; but this situation will be avoided throughout the chapter.

The first coset code was designed by Stiffler [18]. This result was based upon the tacit assumption that additive errors and synchronization errors do not occur simultaneously. An average over several code words is required to determine if a word timing error has occurred. A different approach was used by Levy [20] in designing self-synchronizing codes when he defined the slip-detecting characteristic $[s,\delta]$ for block codes. A code has $[s,\delta]$ if for all overlap sequences caused by misframing any sequence of code words by s units or less, the Hamming distance from this overlap sequence to any valid code word is at least δ. Thus both types of errors were not allowed to occur

simultaneously. He gave a sufficient condition on the coset generator for altering cyclic codes to obtain the [s,δ] characteristic, but he did not give any explicit form for this vector.

However Tong [19,33] did give such forms for the generator. He also extended the work to provide for correction as well as detection of synchronization errors. But again this work separated the two types of errors. In the special case of Reed-Solomon codes Solomon [21] used the coset approach to achieve a self-synchronizing property, but an averaging operation is prescribed in order to achieve this effect in the presence of additive errors.

Tavares [22] and Tavares and Fukada [23,24] considered all the situations arising from any combination of additive and synchronization errors including the conjoint occurrence of both. Their work deals with the modification of cyclic codes and is basically algebraic in nature and substance. The key point used repeatedly by them is that an (n,k) cyclic code cannot have a vector that has a burst of length less than (n-k+1) (pg. 152 [2]). However the approach to be applied here is based upon the distance properties of the code.

A coset code which has a self-synchronizing capability has an important property. When it is known that the code has been synchronized, it will operate with the full error-correcting power of the code from which it was derived. Even though cyclic codes are extremely sensitive to synchronization errors, coset codes may not be. The very structure which makes them so sensitive is used in the design of the coset code. It is for these two reasons that coset codes derived from cyclic codes have been studied and used [25].

Results for general cyclic codes are presented in the next section and a special class of cyclic codes, the Reed-Solomon codes, are considered in the following section of this chapter. Most results are believed to be new, and some represent a sharpening of previous work.

Results for General Cyclic Codes

There are instances in which the detection of additive errors or synchronization errors is enough. For example, in a two-way communication system with low probabilities of either type of error, the detection of an error and the retransmission of the erroneous part of the message may be sufficient.

The first result is similar to one given by Tavares and Fukada [23,24].

Theorem 2.1

A coset code may be derived from any cyclic (n,k) code with minimum distance d which has the ability of detecting the simultaneous occurrence of e or less additive errors and t or less bits of slip if the following holds.

$$e = \min \left\{ \left[\frac{d-t-4}{2} \right] , \left[\frac{2n-3t-3}{t+1} \right] \right\} \tag{2.1}$$

The coset generator is

$$\underline{c} = \left(0,\ldots,0,\underbrace{0,.,0,1,0,\ldots,0,1,0,\ldots,1,0,\ldots,0,1}_{\left[\frac{e+3}{2}\right] \text{ blocks}}\right) \tag{2.2}$$

The source of the error is not determined by the decoder.

Proof

The coset generator $\underline{c}$ exists if

$$n \geq (t+1)\left(\frac{e+3}{2}\right) \quad \text{or} \quad e \leq \frac{2n-3t-3}{t+1} \tag{2.3}$$

However (2.2) satisfies this inequality.

Let the slipped and corrupted vector $\underline{v}$ be received.

$$\underline{v} = \underline{b}_{jk}^{(s)} + \underline{c}^{(s)} + \underline{r} \tag{2.4}$$

$\underline{r}$ is the additive error vector with $w(\underline{r}) \leq e$ and the slip s is restricted such that $|s| \leq t$.

In order to detect a slip, or an error or a combination of both, it is sufficient to require the following condition:

$$0 < \min_i \; w(\underline{v} - \underline{c} - \underline{b}_i) \tag{2.5}$$

for $s \neq 0$ or $\underline{r} \neq \underline{0}$. This insures that a received vector will not be a code vector. Notice how the code is designed so as to reflect the effects of a slip into a vector which resembles a coset code vector with an error added.

It suffices to consider the following two cases.

a) $\underline{r} \neq 0$ and $s = 0$

$$\min_i \; w(\underline{v} - \underline{c} - \underline{b}_i) = w(\underline{r}) \tag{2.6}$$

Since $\underline{r} \neq \underline{0}$, $w(\underline{r}) > 0$. Thus the inequality of (2.5) is fulfilled for this case.

Define

$$\hat{j} = \left\{ j \; : \; \underline{b}_{\hat{j}} = \underline{b}_{jj}^{(s)} \right\} \tag{2.7}$$

Since the code is cyclic, $\underline{b}_{\hat{j}}$ is also a member. $w'(\underline{y})$ is the minimum of the weights of $\underline{y}$ with either the first s or the last s elements set to zero.

b) $t \geq |s| > 0$

$$w(\underline{v}-\underline{c}-\underline{b}_i) \geq \min\left\{ \left[w'(\underline{c}^{(s)}-\underline{c})-w'(\underline{r}) \right] , \left[d-t-w'(\underline{c}^{(s)}-\underline{c})-w'(\underline{r}) \right] \right\} \tag{2.8}$$

The first term in the minimum expression is from the condition of $i=\hat{j}$ while the second covers the remaining situations. For the $\underline{c}$ of (2.2)

$$e + 3 \geq w' (\underline{c}^{(s)}-\underline{c}) \geq e + 2 \quad \text{for } 0 < |s| \leq t \tag{2.9}$$

Minimizing (2.8) over the index i and employing the appropriate bounds from the equation above yields the following result.

$$\min \; w(\underline{v}-\underline{c}-\underline{b}_i) \geq \min \{ [e+2-e] , [d-t-(e+3)-e] \} \tag{2.10}$$

However from (2.1), $e \leq \frac{d-t-4}{2}$. Thus $d-t-2e-3 \geq 1 > 0$. So inequality (2.5) is satisfied.

Q.E.D.

By requiring a stronger hypothesis the previous theorem will produce a stronger result.

Theorem 2.2

An (n,k) cyclic code has a coset code which is capable of detecting the conjoint occurrence of at most e additive errors and t bits of slip and moreover it has the ability to classify the nature of the error as either additive errors or additive errors and/or slippage. The following relationship is sufficient for the existence of such codes.

$$e = \min \left\{ \left[\frac{d-t-2}{4} \right] , \left[\frac{n-t-1}{t+1} \right] \right\} \tag{2.11}$$

The coset generator is

$$\underline{c} = \Big(0,\ldots,0,\underbrace{\underbrace{0,\ldots,0,1}_{t+1},\underbrace{0,\ldots,0,1}_{t+1},0,\ldots\ldots,1,\underbrace{0,\ldots,0,1}_{t+1}}_{(e+1)\text{ blocks}}\Big) \tag{2.12}$$

Proof

$$n \geq (e+1)(t+1) \quad \text{or} \quad \frac{n-t-1}{t+1} \geq e \tag{2.13}$$

permits this form of $\underline{c}$.

In order to insure the detection of additive errors, slippage or both, require that

$$0 < \min_i w\left(\underline{b}_{jk}^{(s)} + \underline{c}^{(s)} + \underline{r} - \underline{b}_i - \underline{c}\right) \tag{2.14}$$

for $w(\underline{r}) \leq e$ and any j and k and either $0 < |s| \leq t$ or $\underline{r} \neq \underline{0}$ and $|s| \leq t$. This requires all detectable errors to be within a neighborhood of a coset code word. The structure of the coset code is such that slips are transformed into detectable error patterns.

It suffices to consider the same two cases as in the previous theorem. For case a) the proof is identical and for case b), (2.8) is still valid. But for this choice of $\underline{c}$ as in (2.12),

$$w'(\underline{c}^{(s)} - \underline{c}) = 2e + 1 \quad \text{for } 0 < |s| \leq t \tag{2.15}$$

Thus it follows that

$$\min_i w(\underline{b}_{jk}^{(s)} + \underline{c}^{(s)} + \underline{r} - \underline{c} - \underline{b}_i) \geq \min \{[2e+1-e], [d-t-2e-1-e]\} \tag{2.16}$$

Since $e \leq \frac{d-t-2}{4}$ from (2.11),

$$d-t-3e-1 \geq \frac{d-t+2}{4} \geq 1 > 0. \tag{2.17}$$

Since for $w(\underline{r}) \leq e$, any received vector that is perturbed by additive errors only is on or within a distance of e from some coset code vector, it suffices to require that when the received vector contains a slip the following must be true.

$$\min_i w\left(\underline{b}_{jk}^{(s)} + \underline{c}^{(s)} + \underline{r} - \underline{b}_i - \underline{c}\right) \geq e + 1 \text{ for } 0 < |s| \leq t. \tag{2.18}$$

Thus any combination of both types of errors can be distinguished from the occurrence of additive errors alone. Since (2.16) is still true, it only remains to show that $d-t-3e-1 \geq e+1$. But (2.11) implies $4e \leq d-t-2$. Therefore

$$d - t - 2e - 1 \geq 4e + 1 - 3e = e + 1 \tag{2.19}$$

Q.E.D.

The main thrust of the previous theorem is directed at detection and classification of the nature of the errors. If the decoder has provisions for storing 2t additional bits, it is possible to use this theorem to perform slip and error correction by increasing the decoder complexity. The technique is outlined as follows.

1) Determine the distance between the received vector, $\underline{v}$, and the closest coset code word, i.e., compute $\min_i w(\underline{v}-\underline{b}_i-\underline{c}) = J$.

2) If this distance, J, is less than or equal to the code design quantity e, an additive error has occurred. The minimizing code vector is the minimum distance choice for the transmitted one. Note $e < \frac{d}{2}$

from (2.11).

3) However if J is greater than e, the decoder will reframe the received vector (hence the requirement that the decoder have extra storage capacity) and compute the distance from it to the closest coset code vector. If this distance is still greater than e, reframe again. When the correct slip is encountered, this distance will drop to e or less. The requirement (2.18) in the proof of the theorem guarantees that the drop will only occur for the correct value of slip.

Therefore if the decoder strategy described above is used, Theorem 2.2 may be strengthened and extended to provide for correction. The results are stated below in the form of a theorem.

Theorem 2.3

Any (n,k) cyclic code has a coset code which can simultaneously correct e or less additive errors and t or less bits of slip if (2.11) holds.

It is believed that neither this theorem nor the previous one has ever been stated before. These results emphasize the usefulness of the approach taken here--the design of codes from a distance viewpoint. The important property of the coset codes employed in these theorems is that additive errors always occur within a distance of e from a coset code word while slip and additive errors produce vectors with distance greater than e from a word.

A disadvantage of the type of decoder required to implement the above strategy is that the processing time may be prohibitively large. This is a result of the iterative procedure involved. However if the complexity of the decoder is increased, another decoding strategy is

employed in order to obtain the following result. This is similar to one due to Tavares and Fukada [24].

Theorem 2.4

A coset code may be derived from any cyclic (n,k) code with minimum distance d which is capable of simultaneously correcting e or less additive errors and t or less bits of slip provided the following holds:

$$e = \min\left\{\left[\frac{d-2t-3}{4}\right], \left[\frac{n-2t-2}{2t+1}\right]\right\} \tag{2.20}$$

The coset generator is given by

$$\underline{c} = \Big(1,0,\ldots,0,\underbrace{0,\ldots,0,1}_{2t+1},\underbrace{0,\ldots,0,1}_{2t+1},0,\ldots,1,\underbrace{0,\ldots,0,1}_{2t+1}\Big) \tag{2.21}$$

(e+1) blocks

Proof

The existence of $\underline{c}$ is guaranteed by requiring

$$n \geq (e+1)(2t+1) + 1 \quad \text{or} \quad e \leq \frac{n-2t-2}{2t+1} \tag{2.22}$$

Suppose the corrupted and slipped vector presented to the decoder is given by:

$$\underline{v} = \underline{b}_{j_o k_o}^{(s_o)} + \underline{c}^{(s_o)} + \underline{r} \tag{2.23}$$

$\underline{r}$ represents the additive error vector with $w(\underline{r}) \leq e$ and $|s_o| \leq t$. Without loss of generality it is possible to take $s_o \geq 0$. The decoder implements the following strategy.

$$\left\{j,k,s:\ w\left(\underline{v} - \underline{b}_{jk}^{(s)} - \underline{c}^{(s)}\right) \text{ is a minimum with } |s| \leq t\right\} \tag{2.24}$$

Hence it suffices to show

$$w(\underline{r}) = w\left(\underline{v}-\underline{b}_{j_o k_o}^{(s_o)}-\underline{c}^{(s_o)}\right) \triangleq I_{j_o k_o}^{(s_o)} < I_{jk}^{(s)} \triangleq w\left(\underline{v}-\underline{b}_{jk}^{(s)}-\underline{c}^{(s)}\right) \tag{2.25}$$

for any $j \neq j_o$, $k \neq k_o$ and $s \neq s_o$, $|s| \leq t$.

Consider five cases which exhaust all the possibilities.

a) $j \neq j_o$ and any k

$$I_{jk}^{(s_o)} \geq w\left(\underline{b}_{jk}^{(s_o)} - \underline{b}_{jk}^{(s_o)} + \underline{c}^{(s_o)} - \underline{c}^{(s_o)}\right) - w(\underline{r})$$

$$\geq d - s_o - e \geq d - t - e \geq 3e + 2 \tag{2.26}$$

There are at least $(d-s_o)$ nonzero terms in $\underline{b}_{j_o k_o}^{(s_o)} - \underline{b}_{jk}^{(s_o)}$ From (2.20) $d-2t-3 \geq 4e$, and since $d-t \geq d-2t-3$, the last inequality results.

Let

$$\hat{j} = \left\{j : b_{jj}^{(s)} = b_{j_o j_o}^{(s_o)}\right\} \tag{2.27}$$

b) $\hat{j}$ and any k and $0 > s \geq -t$

$$I_{jk}^{(s)} \geq w\left(\underline{b}_{j_o k_o}^{(s_o)} + \underline{c}^{(s_o)} - \underline{b}_{\hat{j}k}^{(s)} - \underline{c}^{(s)}\right) - w(\underline{r})$$

$$\geq w\left(\underline{c}^{(s_o)} - \underline{c}^{(s)}\right) - 2 - W(\underline{r}) \geq 2w(\underline{c}) - 2 - e \geq e+2 \tag{2.28}$$

There are at least $w\left(\underline{c}^{(s_o)} - \underline{c}^{(s)}\right) - 2$ nonzero components in $\left(\underline{b}_{j_o k_o}^{(s_o)} + \underline{c}^{(s_o)} - \underline{b}_{\hat{j}k}^{(s)} - \underline{c}^{(s)}\right)$ from the sth to the $(n-s_o)$th because in this range the definition of $\hat{j}$ guarantees that $\underline{b}_{\hat{j}}$ cancels the elements of $\underline{b}_{j_o}$ and because the form of $\underline{c}$ excludes two nonzero terms of

$\left(\underline{c}^{(s_o)} - \underline{c}^{(s)}\right)$ from this range. Furthermore because of its form

$$w\left(\underline{c}^{(u)} - \underline{c}^{(s)}\right) = 2w(\underline{c}) = 2(e+2) \text{ for } u \neq s \text{ and } |u|, |s| \leq t \tag{2.29}$$

c) $\hat{j}$ and any k and $s \neq s_o$, $t \geq s \geq 0$.

$$\begin{aligned} I_{\hat{j}k}^{(s)} &\geq w\left(\underline{b}_{j_o k_o} + \underline{c}^{(s_o)} - \underline{b}_{\hat{j}k}^{(s)} - \underline{c}^{(s)}\right) - w(\underline{r}) \\ &\geq w\left(\underline{c}^{(s_o)} - \underline{c}^{(s)}\right) - 3 - w(\underline{r}) \geq 2w(\underline{c}) - 3 - e = e+1. \end{aligned} \tag{2.30}$$

In the first $(n-\max(s,s_o))$ components of $\left(\underline{b}_{j_o k_o}^{(s_o)} - \underline{b}_{\hat{j}k}^{(s)} + \underline{c}^{(s_o)} - \underline{c}^{(s)}\right)$ at least the nonzero elements of $\left(\underline{c}^{(s_o)} - \underline{c}^{(s)}\right)$ must appear because of the definition of $\hat{j}$. There are $2w(\underline{c})-3$ of them. Equation (2.29) completes the equality.

d) $j \neq \hat{j}$ and any k and $0 > s > -t$

$$\begin{aligned} I_{jk}^{(s)} &\geq w\left(\underline{b}_{j_o k_o} + \underline{c}^{(s_o)} - \underline{b}_{jk}^{(s)} - \underline{c}^{(s)} - w(\underline{r}) \right. \\ &\geq d - (s_o - s) - \left(w\left(\underline{c}^{(s_o)} - \underline{c}^{(s)}\right) - 2\right) - w(\underline{r}) \\ &\geq d - 2t - 3e - 2 \geq e + 1 \end{aligned} \tag{2.31}$$

There are at least $(d-2(s_o-s))$ nonzero elements of $\left(\underline{b}_{j_o k_o}^{(s_o)} - \underline{b}_{jk}^{(s)}\right)$ from the sth to the $(n-s_o)$th component of which $\left(\underline{c}^{(s_o)} - \underline{c}^{(s)}\right)$ can cancel at most $w\left(\underline{c}^{(s_o)} - \underline{c}^{(s)}\right) - 2$. Equation (2.20) implies that $d-2t \geq 4e+3$ and the last inequality follows.

e) $j \neq \hat{j}$ and any k, and $s = s_o$, $t \geq s \geq 0$.

$$\begin{aligned} I_{jk}^{(s)} &\geq d - \max(s,s_o) - \left(w\left(\underline{c}^{(s_o)} - \underline{c}^{(s)}\right) - 3\right) - w(\underline{r}) \\ &\geq d - t - 3e - 1 \geq e + 2 \end{aligned} \tag{2.32}$$

The minimum number of nonzero terms of $\left(\underline{b}_{j_o k_o}^{(s_o)} - \underline{b}_{jk}^{(s)}\right)$ in its first $(n-\max(s,s_o))$ elements is $(d-\max(s,s_o))$, and $\left(\underline{c}^{(s_o)} - \underline{c}^{(s)}\right)$ can cancel at most $2w(\underline{c})-3$ of them. Since $d-2t-3 \geq 4e$, $d-t-1 \geq 4e+2$. The validity of all five cases has been demonstrated and the proof is complete.

Q.E.D.

The decoding strategy above is to estimate the slip and classify both the transmitted word and either the preceding or succeeding one depending on the direction of the slip, i.e.,

$$\left\{j,k,s:\ w\left(\underline{v} - \underline{b}_{jk}^{(s)} - c^{(s)}\right) \text{ is a minimum, } |s| \leq t\right\} \quad (2.33)$$

$\underline{v}$ is the received vector. It is possible to employ a less complex decoding scheme at the price of reduced performance and also to maintain error and synchronization correcting ability. The decoder estimates the slip and only the code word occupying the largest portion of the received vector. Hence its function is described by:

$$\left\{j,s:\ w\left(\underline{v} - \underline{b}_{jk}^{(s)} - c^{(s)}\right) \text{ is a minimum, } |s| \leq t\right\} \quad (2.34)$$

Corollary 2.1

The conclusion of Theorem 2.4 remains true when a joint decoding strategy is employed if

$$e = \min\left\{\left[\frac{d-4t-4}{4}\right], \left(\left[\frac{n-1}{2t+1}\right] - \left[\frac{t+1}{2}\right] - 1\right)\right\} \quad (2.35)$$

and

$$c = \Big(1,0,\ldots,0,\underbrace{0,\ldots,0,1}_{2t+1},\underbrace{0,\ldots,0,1}_{2t+1},0,\ldots,\underbrace{0,\ldots,0,1}_{2t+1}\Big) \qquad \Big(e+1+\Big\lceil\frac{t+1}{2}\Big\rceil\Big)\text{blocks} \tag{2.36}$$

Proof

Obviously

$$n-1 \geq (2t+1)\Big(e+1+\Big\lceil\frac{t+1}{2}\Big\rceil \text{ or } e \leq \frac{n-1-\Big(1+\Big\lceil\frac{t+1}{2}\Big\rceil\Big)(2t+1)}{(2t+1)} \tag{2.37}$$

The decoder strategy is (2.34). The proof follows the exact outline of the theorem except that the value of $w\big(\underline{c}^{(s_o)} - \underline{c}^{(s)}\big)$ is changed. For this form of $\underline{c}$

$$w\big(\underline{c}^{(s_o)} - \underline{c}^{(s)}\big) = 2w(\underline{c}) = 2\Big(e+2+\Big\lceil\frac{t+1}{2}\Big\rceil\Big) \qquad \text{for } s \neq s_o \text{ and } |s|,\ |s_o| \leq t. \tag{2.38}$$

With this substitution the lower bounds in the five cases considered in the theorem are given as:

a) $I_{jj}^{(s_o)} \geq d-t-e \geq 3e+3t+4$ (2.39)

b) $I_{\hat{j}\hat{j}}^{(s)} \geq e+2+2\Big\lceil\frac{t+1}{2}\Big\rceil \geq e+t+2$ (2.40)

c) $I_{\hat{j}\check{j}}^{(s)} \geq e+1+2\Big\lceil\frac{t+1}{2}\Big\rceil \geq e+t+1$ (2.41)

d) $I_{jj}^{(s)} \geq d-2t-2\Big\lceil\frac{t+1}{2}\Big\rceil -3e-2 \geq d-2t-t-1-3e-2 \geq e+t+1$ (2.42)

e) $I_{jj}^{(s)} \geq d-t-2\Big\lceil\frac{t+1}{2}\Big\rceil -3e-1 \geq d-t-t-1-3e-1 \geq e+2t+2$ (2.43)

Two facts were used in the above inequalities. The first is that

$t \leq 2\left[\frac{t+1}{2}\right] \leq t+1$ and the second is that from (2.35) $e \leq \left[\frac{d-4t-4}{4}\right]$ $\leq \frac{d-4t-4}{4}$. Now the correct values j_o and s_o lead to

$$I_{j_o j_o}^{(s_o)} \leq e+t \tag{2.44}$$

The conclusion easily follows.

Q.E.D.

Several comments are in order concerning the strategies of (2.33) and (2.34). As may be seen from the proof of Theorem 2.4, the set of triples from (2.33) may not be a singleton under the conditions of the theorem. For example, it is possible that $\underline{b}_{j_o k_1}^{(s_o)}$ also produces the same minimum value as does $\underline{b}_{j_o k_o}^{(s_o)}$ because $\underline{b}_{k_o}$ and $\underline{b}_{k_1}$ are identical in the first s_o places. (Recall s_o was assumed to be positive.) Thus the items (j_o, k_o, s_o) and $j_o, k_1, s_o)$ are both in the set. For small values of s_o the number of triples can be large. Nevertheless j_o and s_o always remain fixed. Hence there is a number of answers which are all consistent with the strategy given in (2.33). However no multiplicity of pairs belongs to the set defined by (2.34) under the conditions of Corollary 2.1. The item (j_o, s_o) is the single member.

One method for implementing the strategy of (2.33) is to use a syndrome decoding technique (pg. 36 [2]). Using the equivalent polynomial representation, this technique will be described. The decoder subtracts the coset generator c(x) from the received and framed vector, v(x), and computes the syndrome of this difference, i.e., the remainder polynomial from (v(x) - c(x)) after division by g(x) modulo (x^n-1). A table of syndromes is consulted, and when the identical one is found,

the decoder then has determined the value of the slip (s_o) and an error pattern composed of an additive error ($\underline{r}$) and terms from the adjacent code word of which a piece is framed in v(x). Next the error pattern determined above is subtracted from (v(x) - c(x)). The result is the code word $b_{j_o j_o}^{(s_o)}(x)$. Since s_o has been determined from the table of syndromes, a shift of $b_{j_o j_o}^{(s_o)}(x)$ gives the true code word.

It can be shown that the syndromes from (v(x) - c(x)) are all distinct for distinct values of s_o in the range [-t,t] even for different succeeding or preceding code words (see Theorem 7 [24] or Theorem 3.6 [22]). However this does not mean that the syndromes are unaffected by the error vectors and the parts of the other word. Let {f(x)} denote the remainder term of the division of f(x) by g(x) modulo (x^n-1). Thus the syndrome of (v(x) - c(x)) is expressed as {v(x) - c(x)}. Since $b_{j_o j_o}^{(s_o)}(x)$ (or $x^{s_o} b_{j_o}(x)$) is a member of the code and thus is divisible by g(x), the syndrome becomes:

$$\left\{ b_{j_o k_o}^{(s_o)}(x) - b_{j_o j_o}^{(s_o)}(x) + c(x)\,(1-x^{s_o}) + r(x) \right\} \qquad (2.46)$$

Now the term $\left(b_{j_o k_o}^{(s_o)}(x) - b_{j_o j_o}^{(s_o)}(x) \right)$ is dependent only on the code word $b_{k_o}(x)$. So if $b_{k_o}(x)$ were replaced by $b_{k_1}(x)$ whose first s_o terms were not identical with those of $b_{k_o}(x)$, this term and the new syndrome would be different. In either case, it still would indicate that a slip s_o had occurred. In constructing the table of syndromes and in partitioning it into classes according to the magnitude and sign of the slip, the terms from $b_{k_o}(x)$ and the error terms from r(x) are both used. Thus these two factors are at least implicitly determined whenever a particular syndrome is chosen from the table. In certain cases

there is another error $r'(x)$ which when combined with the effects of the first s_o terms of $b_{k_2}(x)$, will produce the same syndrome. Since the table is normally constructed so as to give the result containing the least number of additive errors, any ambiguity is eliminated. It may be seen from (2.46) that the maximum number of syndromes in this scheme is:

$$(2tq^k + 1)\left(\sum_{i=0}^{e} \binom{n}{i}\right) \tag{2.47}$$

A scheme for performing the joint decoding prescribed by (2.34) under the conditions of Corollary 2.1 is outlined. First the syndrome of $(v(x) - c(x))$ with the first and the last t terms set to zero is computed. From this syndrome the value of the slip s_o and the error pattern from $(v(x) - c(x))$ which has the first and last t terms equal to zero gives the code word $b_{j_o j_o}^{(s_o)}(x)$.

The syndrome of $(v(x) - c(x))$ with the 2t terms set to zero may be represented by:

$$\left\{b_{j_o k_o}^{(s_o)}(x) + x^{s_o} c(x) + r(x) - c(x) - u(x) - U(x)\right\} \tag{2.48}$$

$U(x)$ eliminates the last t terms of $(v(x) - c(x))$ while $u(x)$ eliminates the first t. The number of syndromes in this scheme cannot exceed:

$$(2t+1) \sum_{i=0}^{e} \binom{n}{i} \tag{2.49}$$

It remains to show that the syndromes are all distinct for distinct values of s_o as long as $|s_o| \leq t$. Consider another received and framed vector.

$$v'(x) = b^{(s_1)}_{j_1 k_1}(x) + x^{s_1} c(x) + r'(x) \tag{2.50}$$

Let $w(\underline{r}') \leq e$, and $s_1 \neq s_o$ with $|s_1| \leq t$, and j_1 and k_1 be arbitrary indices. It suffices to show that $\{v(x) - c(x)\} \neq \{v'(x) - c(x)\}$ where the first and last t terms in each expression have been eliminated. Since $b^{(s_o)}_{j_o j_o}(x)$ and $b^{(s_1)}_{j_1 j_1}(x)$ are both code words, the requirement may be written as:

$$\left\{c(x)\left(x^{s_o} - x^{s_1}\right) + r(x) - r'(x) - y(x) - Y(x)\right\} \neq 0 \tag{2.51}$$

$Y(x)$ eliminates the first t terms of $(v(x) - v'(x))$ while $y(x)$ removes the last t. It will be shown that the polynomial in (2.51) is not a representation for a code vector and so it is not divisible by $g(x)$.

$$\begin{aligned} w\left(\underline{c}^{(s_o)} - \underline{c}^{(s_1)} + \underline{r} - \underline{r}' - \underline{y} - \underline{Y}\right) &\leq w\left(\underline{c}^{(s_o)} - \underline{c}^{(s_1)} - \underline{y} - \underline{Y}\right) + w(\underline{r} - \underline{r}') \\ &\leq 2\left(e + 2\left[\frac{t+1}{2}\right]\right) - 4 + 2t + 2e \\ &\leq 4e + 2t + t + 1 \leq d - t - 3. \end{aligned} \tag{2.52}$$

The last inequality follows from (2.35).

$$\begin{aligned} w\left(\underline{c}^{(s_o)} - \underline{c}^{(s_1)} + \underline{r} - \underline{r}' - \underline{y} - \underline{Y}\right) &\geq w\left(\underline{c}^{(s_o)} - \underline{c}^{(s_1)} - \underline{y} - \underline{Y}\right) - w(\underline{r} - \underline{r}') \\ &\geq 2\left(e + 2\left[\frac{t+1}{2}\right]\right) - 4 - 2e = 2\left[\frac{t+1}{2}\right] \geq t \end{aligned} \tag{2.53}$$

Since the polynomial in (2.51) corresponds to a vector whose weight is neither zero nor greater than $(d-1)$, it cannot be a code vector.

The previous theorem and corollary deal with the situation of symmetric slip, i.e., when the range of slip is from -t to +t. This

may not always be the case. In fact the slip may be only unidirectional, e.g., a receiver may lose bits as it attempts to achieve bit synchronization. So these results may be too general for a given problem. However a refinement of these results which will cover all problems is possible.

Let t^- be the number of bits of negative slip and t^+ be the number of bits of positive slip. Further let $t_t = t^- + t^+$ and $t_m = \max(t^+, t^-)$.

Corollary 2.2

a) If the blocks in $\underline{c}$ of (2.21) are (t_t+1) long instead of $(2t+1)$ and if

$$e = \min\left\{\left[\frac{d-t_t-3}{4}\right], \left[\frac{n-2t_t-2}{t_t+1}\right]\right\} \tag{2.54}$$

then there is a coset code which can simultaneously correct e or less additive errors and t^+ bits of slippage in the positive direction or t^- in the negative.

b) Also if the blocks in $\underline{c}$ of (2.36) are (t_t+1) long and there are $e+1+\left[\frac{t_m+1}{2}\right]$ of them, and if

$$e = \min\left\{\left[\frac{d-t_t-2t_m-4}{4}\right], \left(\left[\frac{n-1}{t_t+1}\right] - \left[\frac{t_m+1}{2}\right] - 1\right)\right\} \tag{2.55}$$

then a joint decoding strategy used with a coset code will correct e or less additive errors and simultaneously determine the magnitude and direction of either t^+ bits of positive slip or t^- bits of negative slip.

Each of these results follows easily from their respective proofs.

Reed-Solomon Codes

Since Reed-Solomon codes are cyclic, they can be made self-synchronizing by any of the previous techniques. However for this class of codes there is a more powerful approach. These results will be used in Chapter 5 when concatenated codes are considered.

Let $\Gamma = \{\underline{B}_0,\ldots,\underline{B}_M\}$, $M = q^K-1$, be an (N,K) Reed-Solomon code generated by the polynomial: $G(z) = \prod_{i=1}^{D-1} (z-\lambda^i)$ over GF(q). λ is a primitive Nth root of unity. For Reed-Solomon codes recall that N=q-1 and that the minimum distance D=N-K+1, i.e., a maximum-distance separable code.

Theorem 2.5

There is a coset code derivable from an (N,K) R-S code which can simultaneously correct E or less additive errors and T or less bits of slippage where

$$E = \left\lfloor \frac{N-K-2T-1}{2} \right\rfloor \tag{2.56}$$

as long as either $K \nmid N$ or if it does, then require $N > 2KT$. The coset generator is given by

$$\underline{C} = \left(1,\lambda^K,\lambda^{2K},\ldots\ldots\ldots\ldots,\lambda^{(N-1)K}\right) \tag{2.57}$$

Proof

First it will be shown that if $2K < N$, $\underline{C}$ is in a (N,K+1) Reed-Solomon code, Γ', which contains Γ as a proper subcode. Thus the minimum distance of Γ' is $D'=N-K$. Let the generator of Γ' be $G'(z) = \prod_{i=1}^{D'-1} (z-\lambda^i)$ over $GF(q^K)$. For any j, $0 < j \leq N-K-1$

$$C(\lambda^j) = 1 + \lambda^{K+j} + \lambda^{2(K+j)} + \ldots + \lambda^{(N-1)(K+j)}$$
$$= \frac{1-\lambda^{(K+j)N}}{1-\lambda^{K+j}} = 0 \tag{2.58}$$

since λ is primitive Nth root of unity and since $\lambda^{K+j} \neq 1$ because $j < N-K$. Therefore $\underline{C} \in \Gamma'$, but $\underline{C} \notin \Gamma$ because $C(\lambda^{N-K}) = \sum_{}^{N} 1 \neq 0$.

Suppose, just for the sake of definiteness, that the corrupted and slipped vector presented to the decoder is given by

$$\underline{V} = \underline{B}_{j_o k_o}^{(s_o)} + \underline{C}^{(s_o)} + \underline{r} \tag{2.59}$$

$\underline{r}$ represents the error vector and $T \geq s_o \geq 0$. The decoder strategy is

$$\left\{j,k,s: \; w\left(\underline{V} - \underline{B}_{jk}^{(s)} - \underline{C}^{(s)}\right) \text{ is a minimum with } |s| \leq T\right\} \tag{2.60}$$

Thus it suffices to show that

$$I_{j_o k_o}^{(s_o)} \triangleq w\left(\underline{V} - \underline{B}_{j_o k_o}^{(s_o)} - C^{(s_o)}\right) < I_{jk}^{(s)} \triangleq w\left(\underline{V} - \underline{B}_{jk}^{(s)} - \underline{C}^{(s)}\right) \tag{2.61}$$

for $j \neq j_o$, $k \neq k_o$ and $s \neq s_o$, $|s| \leq T$, whenever $w(\underline{r}) \leq E$.

Clearly $I_{j_o k_o}^{(s_o)} \leq E$. Furthermore

$$\underline{C} - \underline{C} = (1 - \lambda^{sK})\, \underline{C} \tag{2.62}$$

So

$$w\left(C - C^{(s)}\right) = \begin{cases} N & \text{if } s \not\equiv 0 \bmod N \\ 0 & \text{if } s \equiv 0 \bmod N \end{cases} \tag{2.63}$$

Now consider the following cases which exhaust all the possible combinations of j,k, and s.

a) $j \neq j_o$ and any k

$$I_{jk}^{(s_o)} \geq w\left(\underline{B}_{j_o k_o}^{(s_o)} - \underline{B}_{jk}^{(s_o)}\right) - w(\underline{r})$$

Since the code is cyclic, the first $(N-s_o)$ components of $\left(\underline{B}_{j_o k_o}^{(s_o)} - \underline{B}_{jk}^{(s_o)}\right)$ are equal to the first $(N-s_o)$ elements of a code word of Γ, and because $j \neq j_o$ it is not the zero word. In addition it follows from the maximum-distance separable property [26] there can be $(K-1)$ zeros at most in these positions.

$$I_{jk}^{(s_o)} \geq N - K - T - E + 1 \tag{2.64}$$

From (2.56), $N-K-2T-1 \geq 2E$, and so $I_{jk}^{(s_o)} \geq E+2+T$. Let

$$\hat{j} = \left\{ j : \underline{B}_{jj}^{(s)} = B_{j_o j_o}^{(s_o)} \right\} \tag{2.65}$$

b) $\hat{j}$ and any k and $0 > s \geq T$

$$I_{\hat{j}k}^{(s)} \geq w\left(\underline{C}^{(s_o)} - \underline{C}^{(s)}\right) - 2T - w(\underline{r}) \geq N-2T-E \tag{2.66}$$

The definition of $\hat{j}$ implies that at most only the first s and last s_o components of $\left(\underline{B}_{j_o k_o}^{(s_o)} - \underline{B}_{\hat{j}k}^{(s)}\right)$ can cancel elements of $\left(\underline{C}^{(s_o)} - \underline{C}^{(s)}\right)$. However from (2.56) $N-2T-K-1 \geq 2E$; this $I_{\hat{j}k}^{(s)} \geq E+K+1$.

$$I_{\hat{j}k}^{(s)} \geq N-T-E \geq E+K+1+T \tag{2.67}$$

This results from an argument analogous to the one in b) above.

d) Any $j \neq \hat{j}$ and any k and any $s \neq s_o$, $T \geq s \geq 0$.

$$I_{jk}^{(s)} \geq w\left(\underline{B}_{j_o k_o}^{(s_o)} - \underline{B}_{jk}^{(s)} + \underline{C}^{(s_o)} - \underline{C}^{(s)}\right) - w(\underline{r})$$

The first $(N-\max(s,s_o))$ components of the vector $\left(\underline{B}_{j_o k_o}^{(s_o)} - \underline{B}_{jk}^{(s)} + \underline{C}^{(s_o)} - \underline{C}^{(s)}\right)$ are the first $(N-\max(s,s_o))$ elements of a code word in Γ' because each vector in the sum is a member of that cyclic code. In addition they are not from the zero vector since $j \neq \hat{j}$ and $s \neq s_o$. Because $D' = N-K$, there can be at most K zeros in these positions. So

$$I_{jk}^{(s)} \geq N - K - \max(s, s_o) - E \geq N - K - T - E \qquad (2.68)$$

Since $N-2T-K-1 \geq 2E$, $I_{jk}^{(s)} \geq E+T+1$.

e) Any $j \neq \hat{j}$ and any k and any $s < 0$, $|s| \leq T$

The $(s+1)$th to $(N-s_o)$th elements of $\left(\underline{B}_{j_o k_o}^{(s_o)} - \underline{B}_{jk}^{(s)} + \underline{C}^{(s_o)} - \underline{C}^{(s)}\right)$ and the corresponding portion of a nonzero code vector in Γ'. By similar reasoning as in d) above,

$$I_{jk}^{(s)} \geq N - K - (s_o - s) - E \geq N - K - 2T - E \qquad (2.69)$$

Again since $N-2T-K-1 \geq 2E$, $I_{jk}^{(s)} \geq E+1$.

Thus (2.61) is verified for all the cases and the theorem is proved.

Q.E.D.

Again the decoder in this theorem performs triple classification (2.33). If however its complexity is reduced by programming it for double classification (2.34), a self-synchronizing capability is still obtained.

Corollary 2.3

If a joint decoding strategy is used, the conclusion of Theorem 2.5 is still valid if

$$E = \left\lfloor \frac{N-K-3T-1}{2} \right\rfloor \tag{2.70}$$

The coset generator remains the same as (2.57)

Proof

The decoder strategy is given by (2.34). The lower bounds on $I_{jj}^{(s)}$ for all $j \neq j_o$ and $s \neq s_o$, $|s| \leq T$ is the same as in each of the five cases in the theorem. But

$$I_{j_o j_o}^{(s_o)} \leq E + T \tag{2.71}$$

Note that $I_{jj}^{(s)} > E + T \geq I_{j_o j_o}^{(s_o)}$.

Q.E.D.

This theorem immediately yields a corollary concerning the bound on the distance from any misframed vector to any code word in this coset code.

Corollary 2.4*

The coset code derived from a Reed-Solomon code as in Theorem 2.5 has the property that

$$w\left(\underline{B}_{j_o k_o}^{(s_o)} + \underline{C}^{(s_o)} - \underline{B}_j - \underline{C}\right) \geq N - K \min\,(K, |s_o|) \tag{2.72}$$

for any $s_o \not\equiv 0$ modulo N as long as $K \nmid N$ or excluding those $s_o \equiv 0$ modulo T if $N = KT$.

---

*This result was first presented by Solomon [21], but he omitted the necessary condition that $K \nmid N$. There are numerous counterexamples. A(63, 9) R-S code over GF(64) with slip of ±7 and adjacent $\underline{0}$ vectors gives a zero weight.

Proof

Without loss of generality assume $s_o > 0$. The first $(N-s_o)$ elements of $\left(\underline{B}_{j_o k_o}^{(s_o)} + \underline{C}^{(s_o)} - \underline{B}_j - \underline{C}\right)$ are the relative elements of a code vector in the Γ'code. Since $s_o \not\equiv 0 \bmod N$ and $K \nmid N$, this vector is nonzero in Γ'. Thus there can be at most K zeros among these positions. (See case d) of the theorem.) Also since the code is cyclic the last s_o elements are nonzero and can have at most min $(K, |s_o|)$ zeros among them. Therefore

$$w\left(\underline{B}_{j_o k_o}^{(s_o)} + \underline{C}^{(s_o)} - \underline{B}_j - \underline{C}\right) \geq N - K - \min (K, |s_o|)$$

Q.E.D.

Stronger results are also possible when detection or when detection and classification are desired of Reed-Solomon codes.

Theorem 2.6

For any (N,K) Reed-Solomon code there is a coset code which can detect the concomitant occurrence of E or less additive errors and either any amount of slippage as long as $K \nmid N$ or T or less bits if $N = K(T+1)$.

$$E = N = 2K - 1 \tag{2.73}$$

The coset generator is given by (2.57).

Proof

Let

$$\underline{V} = \underline{B}_{jk}^{(s)} + \underline{C}^{(s)} + \underline{r} \tag{2.74}$$

with $w(\underline{r}) \leq E$. $\underline{r}$ represents the additive error vector. In order to detect either or both types of errors it is sufficient that

$$0 < \min_i \quad w(\underline{V} - \underline{B}_i - \underline{C}) \tag{2.75}$$

for $r \neq 0$ or $s \not\equiv 0 \bmod N (|s| \leq T)$

a) If $\underline{r} \neq 0$ and $s \equiv 0 \bmod N$

$$\min_i \; w(\underline{V} - \underline{B}_i - \underline{C}) = w(\underline{r}) > 0 \tag{2.76}$$

because $\underline{r} \neq \underline{0}$.

b) If $s \not\equiv 0 \bmod N$ $(|s| \leq T)$

$$w(\underline{V} - \underline{B}_i - \underline{C}) \geq w\left(\underline{B}_{jk}^{(s)} - \underline{B}_i + \underline{C}^{(s)} - \underline{C}\right) - w(\underline{r}) \tag{2.77}$$

However Corollary 2.4 applies and so using (2.73),

$$\min w(\underline{V} - \underline{B}_i - \underline{C}) \geq N-K-\min(K, |s|)-E \geq N-2K-E = +1 \tag{2.78}$$

Q.E.D.

Theorem 2.7

An (N,K) Reed-Solomon code has a coset code which is capable of concurrently detecting E or less additive errors and either $\lceil \frac{N}{2} \rceil$ bits of slippage if $K \nmid N$ or at most T bits if $N = K(T+1)$. Moreover it can classify the nature of the error.

$$E = \left\lceil \frac{N - 2K - 1}{2} \right\rceil \tag{2.79}$$

The coset generator is (2.57).

Proof

The proof of the detection claim follows the proof of Theorem 2.6 since the value of E here is less than or equal to the value given by (2.73).

Since $w(\underline{r}) \leq E$, any received vector containing only additive errors is a Hamming distance of at most E from some coset code word.

Thus it is sufficient to require

$$\min_i \; w(\underline{V} - \underline{B}_i - \underline{C}) \geq E + 1 \quad \text{for } s \not\equiv 0 \bmod N \tag{2.80}$$

Hence the occurrence of additive errors alone can be distinguished from slip errors with or without additive errors.

Again using Corollary 2.4 and employing (2.79),

$$\min_i w(\underline{V} - \underline{B}_i - C) \geq (N-2K) - E \geq 2E+1-E \geq E+1 \tag{2.81}$$

Q.E.D

As in the case of general cyclic codes it is possible to use this theorem to simultaneously perform additive and slip error correction. The decoder must have an additional storage of 2T or N code bits depending on whether $K|N$ or not. An outline of the decoding steps is given below.

1) Compute the distance, J, between the received vector $\underline{V}$ and the closest coset code word, i.e., $J = \min_i w(\underline{V} - \underline{B}_i - \underline{C})$.

2) If $J \leq E$, an additive error has occurred and the minimum distance decoder choice is given. Note $E \leq (\frac{D-1}{2})$ for the R-S codes by observing (2.79).

3) However if $J > E$, the decoder will reframe the received vector (thus the extra storage requirement) and compute the distance J_1, between it and its nearest coset code neighbor. If $J_i > E$, reframe and compute again. When the correct slip is found, $J_i \leq E$. The last part of the proof insures the uniqueness of the slip value as found by this procedure.

Therefore if the decoding strategy outlined above is implemented, the results of Theorem 2.7 can be used for slip and additive error

correction.

Theorem 2.8

For any (N,K) Reed-Solomon code there is a coset code which can simultaneously correct E or less additive errors and either $\lceil \frac{N}{2} \rceil$ or less bits of slip when K∤N or T or less bits of slip if N = K(T+1)

$$E = \left\lceil \frac{N-2K-1}{2} \right\rceil \tag{2.82}$$

The coset generator is given by (2.57).

Just as Corollary 2.2 provides results for the general cyclic code when the slip is not symmetrical, the following corollary treats the same circumstances when Reed-Solomon codes are involved. Let T^+ be the number of bits of slip in the positive direction while T^- denotes the number in the negative direction. Further define $T_m = \max(T^+, T^-)$ and $T_t = T^+ + T^-$.

Corollary 2.5

There is a coset code derivable from an (N,K) R-S code which can simultaneously correct E or less additive errors and (a) T^+ or less bits of positive slip and T^- or less bits of negative slip where

$$E = \min\left\{ \left\lceil \frac{N-2K+1}{2} \right\rceil , \left\lceil \frac{N-K-T_t-1}{2} \right\rceil \right\} \tag{2.83}$$

as long as either K∤N or $\frac{N}{K} > T_m$. (b) either $\lceil \frac{N}{2} \rceil$ or less bits of slip in either direction if K∤N or T^+ or less of positive slippage and T^- of negative if $N = K(T_m+1)$.

$$E = \left\lceil \frac{N-2K-1}{2} \right\rceil \tag{2.84}$$

Part (a) is a refinement of Theorem 2.5 and likewise (b) is one

of Theorem 2.8. The proof of this corollary follows easily from the respective theorems.

Examples

Several examples will be presented to demonstrate the results of this chapter. They are given in Table 2.1 and Table 2.2. All the results deal with the simultaneous correction of both additive errors and slip. In order to demonstrate the approach for general cyclic codes, binary BCH codes (pg. 164 [2] or pg. 176 [15]) are used. The codes have length n, information content k and a lower bound d on the minimum distance. Since the bound in some instances is not the true minimum distance [30], the additive error performance as indicated in Table 2.1 may be a lower bound on the true performance. Table 2.2 gives the results using Reed-Solomon codes over the field $GF(2^k)$. Since K does not divide N in any of these examples, the slip range of Theorem 2.8 is $[\frac{N}{2}]$ independent of the value of T. These examples will be combined to give some examples of another approach in Chapter 5. The tables are intended to show the versitility of the techniques of this chapter, but they by no means begin to exhaust the possibilities.

Table 2.1. Performance Capabilities of the Coset Codes of Several Binary Cyclic Codes

Code Parameters		Slip Correction Range	Maximum Number of Correctable Additive Errors, e, Using the Technique of		
(n,k)	d	t	Theorem 2.3	Theorem 2.4	Corollary 2.1
(31,6)	15	1	3	2	1
		3	2	1	*
		9	1	*	*
(63,36)	11	1	2	1	0
		2	1	1	*
		5	1	*	*
(63,30)	13	1	2	2	1
		3	2	1	*
		7	1	*	*
(63,24)	15	1	3	2	1
		4	2	1	*
		9	1	*	*
(63,18)	21	1	4	4	4
		3	4	3	1
		7	3	1	*
		11	2	*	*
		15	1	*	*
(127,99)	9	1	1	1	0
		3	1	0	*
(127,78)	15	1	3	2	1
		2	2	2	0
		3	2	1	*
		9	1	*	*
(127,15)	55	1	13	12	11
		2	12	12	10
		3	12	11	9
		5	12	10	7
		6	11	8	5
		8	11	6	2
		10	10	5	0
		14	7	3	*
		20	5	2	*
		24	4	1	*
		30	3	*	*

Table 2.1. (Continued)

(n,k)	d	t	Theorem 2.3	Theorem 2.4	Corollary 2.1
		41	2	*	*
		49	1	*	*
(31,16)	7	1	1	0	*
(127,92)	11	1	2	1	0
		5	1	*	*
(127,64)	21	1	4	4	3
		2	4	3	2
		3	4	3	1
		5	3	2	*
		7	3	1	*
		11	2	*	*
		15	1	*	*
(127,36)	31	1	7	6	5
		3	6	5	3
		5	6	4	1
		8	5	3	*
		10	4	2	*
		17	3	*	*
		21	2	*	*
		25	1	*	*
(15,5)	7	1	1	0	*
(45,5)	21	1	4	4	3
		2	4	3	2
		3	4	3	1
		5	3	2	*
		7	3	1	*
		11	2	*	*
		15	1	*	*
(63,45)	7	1	1	0	*
(63,30)	13	1	2	2	1
		3	2	1	*
		7	1	*	*
(63,10)	27	1	6	5	4
		3	5	4	2
		8	4	2	*
		13	3	*	*
		17	2	*	*
		21	1	*	*

Table 2.1. (Continued)

(n,k)	d	t	Theorem 2.3	Theorem 2.4	Corollary 2.1
(127,85)	13	1	2	2	1
		3	2	1	*
		7	1	*	*
(127,50)	27	1	6	5	4
		3	5	4	2
		6	4	3	*
		8	4	2	*
		10	3	1	*
		17	2	*	*
		21	1	*	*
(63,7)	31	1	7	6	5
		3	6	5	3
		5	6	4	1
		9	5	2	*
		11	4	1	*
		20	2	*	*
		25	1	*	*
(127,8)	63	1	15	14	13
		5	14	10	7
		8	13	6	2
		10	10	5	0
		13	8	3	*
		17	6	2	*
		20	5	2	*
		24	4	1	*
		28	3	1	*
		41	2	*	*
		57	1	*	*

Table 2.2. Performance Capabilities of the Coset Codes of Several Reed-Solomon Codes over GF(2^k)

Code Parameters			Slip Correction Range	Maximum Number of Correctable Additive Errors, E, Using the Technique of		
k	(N,K)	D	T	Theorem 2.8	Theorem 2.5	Corollary 2.3
3	(7,2)	6	1	1	1	0
			3	1	*	*
3	(7,1)	7	1	2	1	1
			3	2	*	*
4	(15,7)	9	2	0	1	0
4	(15,4)	12	1	3	4	3
			2	3	3	2
			4	3	1	*
			7	3	*	*
4	(15,2)	14	1	5	5	4
			2	5	4	3
			3	5	3	1
			5	5	1	*
			7	5	*	*
5	(31,15)	17	6	0	1	*
5	(31,10)	22	2	5	6	5
			4	5	6	4
			5	5	5	2
			6	5	4	1
			7	5	3	*
			15	5	*	*
5	(31,7)	25	2	8	9	8
			4	8	7	5
			6	8	5	2
			8	8	3	*
			15	8	*	*
5	(31,3)	29	1	12	12	12
			3	12	10	9
			5	12	8	6
			7	12	6	3
			10	12	3	*
			15	12	*	*

Table 2.2. (Continued)

k	(N,K)	D	T	Theorem 2.8	Theorem 2.5	Corollary 2.3
6	(63,31)	33	14	0	1	*
6	(63,24)	40	2	7	8	7
			4	7	8	7
			6	7	8	5
			8	7	8	4
			10	7	8	3
			12	7	7	1
			14	7	5	*
			16	7	3	*
			18	7	1	*
			31	7	*	*
6	(63,16)	48	2	15	16	15
			4	15	16	14
			8	15	15	11
			10	15	13	8
			12	15	11	5
			16	15	7	*
			20	15	3	*
			31	15	*	*
6	(63,8)	56	2	23	24	23
			5	23	22	19
			10	23	17	12
			15	23	12	4
			20	23	7	*
			25	23	2	*
			31	23	*	*
7	(127,63)	65	2	0	1	0
			30	0	1	*
7	(127,45)	83	2	18	19	18
			5	18	19	16
			10	18	19	14
			15	18	19	11
			20	18	19	9
			25	18	15	3
			30	18	10	*
			35	18	5	*
			63	18	*	*
7	(127,16)	112	2	47	48	47
			10	47	45	40
			20	47	35	25

Table 2.2. (Continued)

k	(N,K)	D	T	Theorem 2.8	Theorem 2.5	Corollary 2.3
			30	47	25	10
			40	47	15	*
			50	47	5	*
			63	47	*	*
8	(255,127)	129	2	0	1	0
			62	0	1	*
8	(255,95)	161	2	32	33	32
			10	32	33	28
			20	32	33	23
			30	32	33	18
			40	32	33	13
			50	32	29	4
			60	32	19	*
			70	32	9	*
			127	32	*	*
8	(255,63)	193	2	64	65	64
			10	64	65	60
			30	64	65	50
			50	64	45	20
			60	64	35	5
			70	64	25	*
			80	64	15	*
			90	64	5	*
			127	64	*	*
8	(255,35)	221	2	92	93	92
			10	92	93	88
			30	92	79	64
			50	92	59	34
			60	92	49	19
			70	92	39	4
			80	92	29	*
			90	92	19	*
			100	92	9	*
			127	92	*	*

CHAPTER 3

SUBSET CODES

The coset codes of the previous chapter provide synchronization detection or correction capabilities by suitably choosing the coset generator. Hence each original code vector is translated. However the price of obtaining the additional capabilities in this manner is that the additive error detecting or correcting efficiency of such codes is reduced whenever additive errors and bit slippage occur together. The codes of this chapter are derived from cyclic codes by removing certain vectors from the code before any other alteration is applied. The intent is to delete some of those vectors which are cyclic shifts of a subset of the original code. The effect of this is to obtain a subcode which is less sensitive to bit slippage. Nevertheless even after modification the rate of the resulting code is reduced. However this decrease in the rate performance is reflected either in the total or partial lack of a decrease in the additive error detecting or correcting efficiency whenever both types of errors occur simultaneously. Hence there is a trade-off between these two performance standards.

Only a small amount of work on detection or correction of synchronization errors by these subset codes has been done, and all of that is quite recent [22,27,28]. It is believed that most of the results in this chapter are original. Furthermore they are presented

in a logical sequence beginning with those which pertain to the detection of any type of error and culminating in the presentation of those which deal with the correction of both additive and slip errors.

Coset of Expurgated Codes

Definition 3.1

Let Δ be an (n,k) cyclic code with minimum distance d generated by g(x). Let Δ' denote the cyclic code generated by the composite (f(x)g(x)). The deg f(x) = a and f(0) $\neq$ 0, and furthermore it has exponent u, i.e., $f(x) \mid (x^u - 1)$ but $f(x) \nmid (x^t - 1)$ for any $t < u$ [1,29].

The cyclic code Δ' is formed by expugating the code Δ (pg. 335 [15]). Thus Δ' is a (n,k-a) subcode of Δ. The code to be transmitted will be a coset code derived from Δ' by using $\underline{g} \leftrightarrow g(x)$ as the coset generator. So the modified subset code to be considered in this section is given by:

$$\{\underline{b}_i + \underline{g} : \underline{b}_i \in \Delta'\} \tag{3.1}$$

It will be convenient to define a subset I of the index integers for the vectors of Δ.

$$I = \{\text{integers } i : \underline{b}_i \in \Delta'\} \tag{3.2}$$

Employing the code of (3.1) it is possible to give a result concerning the concomitant detection of both types of errors.

Theorem 3.1

Given an (n,k) cyclic code there is a coset code of an (n,k-a) cyclic code which can detect the concurrent occurrence of $|s|$ bits of slippage in either direction and e(s) or less additive errors if

$$d - |s| - 1 = e(s) \tag{3.3}$$

and

$$|s| \leq u - \tag{3.4}$$

Moreover $u \leq q^a - 1$ and equality is possible if and only if $f(x)$ is a primitive polynomial.

Proof

Let the corrupted and slipped vector which is received be designated by:

$$\underline{v} = \underline{b}_{jk}^{(s)} + \underline{g}^{(s)} + \underline{r} \tag{3.5}$$

$\underline{r}$ represents the additive errors with $w(\underline{r}) \leq e(s)$ and j and k are both in the set I defined by (3.2). In order to be able to detect either an additive error or a synchronization error or both, it suffices to require that

$$\min_{I} w(\underline{v} - \underline{b}_i - \underline{g}) > 0 \tag{3.6}$$

for $\underline{r} \neq \underline{0}$ or $0 < |s| \leq u - 1$. Thus no subcode coset vector can be obtained by misframing the corrupted incoming data stream.

a) For $\underline{r} \neq \underline{0}$ and $s = 0$

$$\min_{I} (\underline{v} - \underline{b} - \underline{g}) = w(\underline{r}) > 0 \tag{3.7}$$

b) For any $\underline{r}$ and any s such that $0 < |s| \leq u - 1$, consider the following inequality which holds for any $i \in I$.

$$w(\underline{v}-\underline{b}_i-\underline{g}) \geq w\left(\underline{b}_j^{(s)}-\underline{b}_j+\underline{g}^{(s)}-\underline{g}\right) - |s| - w(\underline{r}) \tag{3.8}$$

Since $j \in I$, $\underline{b}_j^{(s)} \in \Delta'$ and so $\left(\underline{b}_j^{(s)} - \underline{b}_i\right) \in \Delta'$. But $(\underline{g}^{(s)} - \underline{g}) \notin \Delta'$ if $|s| \leq u-1$. A proof of this fact follows. Suppose this is not the case. $(\underline{g}^{(s)} - \underline{g}) \in \Delta'$ if and only if $(\underline{g}^{(s)} - \underline{g}) \leftrightarrow g(x)\,(x^s-1)$ modulo (x^n-1) is divisible by $(f(x)g(x))$ modulo (x^n-1). However this is possible if and only if $f(x) \mid (x^s-1)$ modulo (x^n-1). But $f(x) \nmid (x^t-1)$ mod (x^n-1) for any $|t| < u$. Note if $t < 0$, $(x^t-1) \equiv -x^{n+t}\,(x^{-t}-1)$ modulo (x^n-1). This contradiction establishes the fact.

Therefore $\left(\underline{b}_j^{(s)} - \underline{b}_i + \underline{g}^{(s)} - \underline{g}\right) \neq 0$ for any $i \in I$. But it is a code vector of Δ. So it follows that:

$$\min_I w(\underline{v} - \underline{b}_i - \underline{g}) \geq d - |s| - e(s) \qquad (3.9)$$

The right hand side is strictly positive by using (3.3).

The "moreover" statement easily follows from the definition of the exponent of a polynomial and also from the definition of a primitive polynomial (Thm. 13, pg. 130 [1] or section 29 [29]). The existence of primitive polynomials over any finite field is well known.

Q.E.D.

It must be noted that in this theorem the additive error detection capabilities, $e(s)$, of this block code are a function of the magnitude of the slip that has actually occurred. If there is no slip, the usual bound on error detection is the result.

Theorem 3.2

Every (n,k) cyclic code can be modified into an (n,k-a) block code which is capable of detecting the simultaneous occurrence of at most e additive errors and t or less bits of slippage (independent of direction) if

$$e = \left\lfloor \frac{d-t-1}{2} \right\rfloor \tag{3.10}$$

where

$$t - u - 1 \leq q^a - 2 \tag{3.11}$$

Furthermore this block code can distinguish between additive errors and a combination of both types of errors. Equality can be achieved in (3.11) by using a primitive polynomial to generate the expurgated code.

Proof

The block code is the coset code given in (3.1). Let the generic form of the received vector, $\underline{v}$, be given as in (3.5). The detection part of this theorem as well as the existence of the equality in (3.11) is proved in the same manner as in the previous theorem.

To be able to distinguish between additive errors alone ($s=0$) and any combination of both ($s\neq 0$) it suffices to require for any s, $0 < |s| \leq u-1$, and any $\underline{r}$, $w(\underline{r}) \leq e$ that:

$$\min_{I} \; w(\underline{v} - \underline{b}_i - \underline{g}) \geq e + 1 \tag{3.12}$$

This is evident from the fact that if $s=0$, all received vectors are within a distance of $e+1$ of a coset code member.

Again as in the proof of Theorem 3.1, (3.9) is valid.

$$\min \; w(\underline{v} - \underline{b}_i - \underline{g}) \geq d - |s| - e \geq e + 1 \tag{3.13}$$

The right inequality results from (3.10) which implies $d-|s|-1 \geq 2e$.

Q.E.D.

Just as it was possible in Chapter 2 to use Theorem 2.2 as a basis for

a correction scheme, the results of the previous theorem dealing with detection and classification of errors will be extended so as to permit the simultaneous correction of both additive errors and slippage. This extension necessitates increasing the complexity of the decoder to take advantage of the code's structure.

Theorem 3.3

For any (n,k) cyclic code there is an (n,k-a) block code which can correct the conjoint occurrence of e or less additive errors and t or less bits of slip - independent of the direction - if

$$e = \left\lceil \frac{d-t-1}{2} \right\rceil \tag{3.14}$$

and

$$t = u-1 \leq q^2-2 \tag{3.15}$$

Equality will hold if the only if f(x) is a primitive polynomial.

Proof

The validity of all the conclusions of the theorem is demonstrated once the decoding strategy is outlined. The steps of this strategy are given below as well as being depicted in figure 3.1.

1) Compute the Hamming distance between the framed vector $\underline{v}_o$ and the closest member of the block code. That is determine the quantity J_o.

$$J_k = \min_I w(\underline{v}_k - \underline{b}_i - \underline{g}) \tag{3.16}$$

2) If the received vector is within a distance of e+1 from a possible block code vector, i.e., $J_o \leq e$, only an additive error has occurred. Then the block code vector which gives J_o in (3.16) is the

minimum distance choice as the transmitted one.

3) However if J_o is greater than e, a combination of errors has occurred. So the decoder must reframe, obtaining $\underline{v}_1$, and determine the distance J_1 to the closest neighbor. Continue reframing and computing the distance J_k until the distance is less than e+1. Then the slip is corrected and moreover any additive errors are also corrected by choosing the minimizing block code member. The uniqueness of the solution is guaranteed by the stipulation (3.12) in the previous theorem's proof.

Q.E.D.

The important feature of the code's design is that additive errors always result in a received vector that is within a sphere about the true coset code vector whereas for any slip in the designated range the received one is within a concentric shell about some coset code vector. The decoding scheme is an iterative one. The choice of the sequence of values of slip by which it searches is generally guided by any statistical knowledge about the slip.

If a less complex decoding strategy is used, correction of conjointly occurring errors is possible but at a degradation in both additive error and slip correction performance. This result is equivalent to one due to Tavares [22].

Theorem 3.4

Any (n,k) cyclic code may be modified into an (n,k-a) block code as defined in (3.1) which has the capability of simultaneously correcting e or less additive errors and t or less bits of slip where:

$$e = \left[\frac{d-1}{2}\right] - t \qquad (3.17)$$

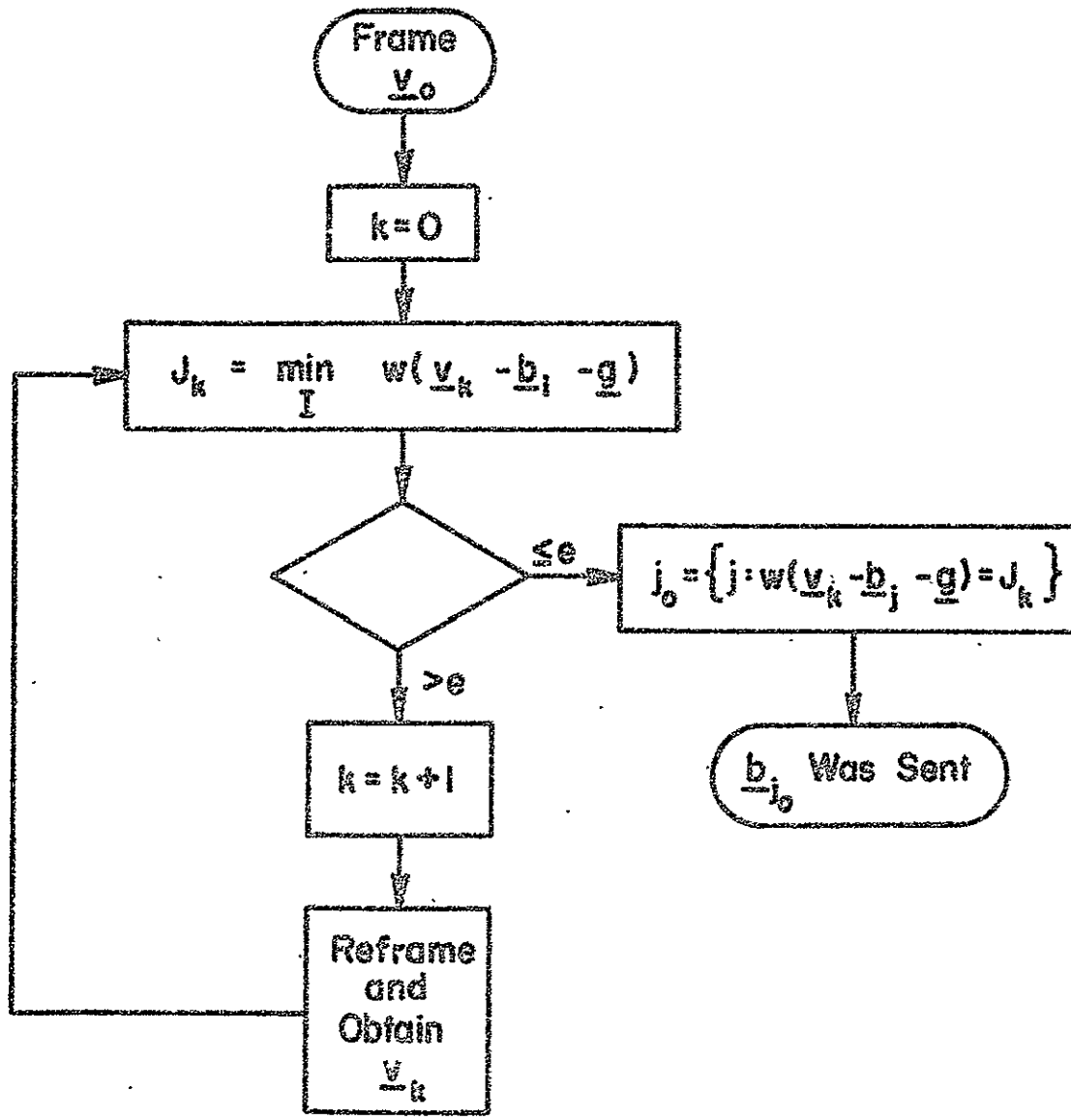


FIGURE 3.1. DECODER STRATEGY FOR THEOREM 3.3 .

and

$$t \leq \frac{u-1}{2} \leq \frac{q^a-2}{2} \tag{3.18}$$

Furthermore $t = \left[\frac{q^a-2}{2}\right]$ if f(x) of Definition 3.1 is primitive.

Proof

Suppose that the framed vector is given by:

$$\underline{v} = \underline{b}_{jk}^{(s)} + \underline{g}^{(s)} + \underline{r} \tag{3.19}$$

$\underline{r}$ represents the additive errors with $w(\underline{r}) \leq e$ and $|s| \leq t$.

The decoding procedure is outlined.

1) Determine $\underline{b}_i \in \Delta$ such that $w(\underline{v} - \underline{b}_i)$ is a minimum.

2) Determine the remainder term of $\frac{b_i(x)}{f(x)g(x)}$ modulo (x^n-1). This term corresponds to a value (both magnitude and sign) of the slip s.

3) Then the transmitted vector is $\underline{b}_i^{(-s)}$.

It suffices to show that if $\underline{v}$ in (3.19) is decoded as above, the results are $\underline{b}_j$ and s.

a) For any $\underline{b}_i \in \Delta$

$$w(\underline{v}-\underline{b}_i) \leq w\left(\underline{b}_{jj}^{(s)}-\underline{b}_i+\underline{g}^{(s)}\right) + |s|+w(\underline{r}) \leq w\left(\underline{b}_{j_1}-\underline{b}_i\right) + t+e \tag{3.20}$$

Now $\underline{b}_{j_1} = \underline{b}_{jj}^{(s)} + \underline{g}^{(s)}$ is in Δ. Therefore

$$\min_i w(\underline{v} - \underline{b}_i) \leq t + e \leq \frac{d-1}{2} \tag{3.21}$$

So the unique choice of a vector in Δ which satisfies this inequality is $\underline{b}_{j_1}$ because of (3.17).

b) The remainder term of $\frac{b_{j_1}(x)}{f(x)g(x)}$ must be considered.

$$b_{j_1}(x) = x^s\, b_j(x) + x^s\, g(x) \bmod (x^n-1) \qquad (3.22)$$

Now $x^s\, b_j(x)$ is in Δ' and so is divisible by $f(x)g(x)$ modulo (x^n-1). However $f(x)g(x)$ divides $x^s g(x)$ if and only if $f(x)$ divides x^s with all divisions modulo (x^n-1). Thus the remainder term required is exactly the remainder of $\frac{x^s}{f(x)}$. This term will be denoted as $\{\frac{x^s}{f(x)}\}$. To be able to establish a unique correspondence between values of slip and the remainder terms, it must be shown that if $m \neq s$ and $|m|$ and $|s|$ are both less than $\frac{u}{2}$, then $\{\frac{x^s}{f(x)}\} \neq \{\frac{x^m}{f(x)}\} \bmod (x^n-1)$. Or equivalently show $\{\frac{x^m(x^{s-m}-1)}{f(x)}\} \not\equiv 0 \bmod (x^n-1)$. But $f(x) \nmid x^y$ for any integer y since $f(0) \neq 0$. So finally $\{\frac{(x^{s-m}-1)}{f(x)}\} \not\equiv 0 \bmod (x^n-1)$ is sufficient for the uniqueness. However $|s-m| < u$ and $f(x) \nmid (x^y-1)$ for $y < u$ from the definition of exponent. Thus the remainder terms in this range are distinct.

c) The unique choice for the transmitted vector is

$$\underline{b}_{j_1}^{(-s)} = \underline{b}_j + \underline{g}$$

As mentioned in the proof of Theorem 3.1, the exponent $u \leq q^a-1$.

Q.E.D.

The decoder in this scheme performs decoding as if the original code were being used. This removes the additive errors. It then takes advantage of the fact that some of the cyclic shifts of the original code vectors have been removed. Of the $(q^a-1)\, q^{k-a}$ vectors which have been removed, u-1 are made to correspond with a synchronization error. The vector g corresponds to s=0. The computation of the remainder term is equivalent to determining a syndrome (pg. 36 [2]) in the Δ' code. Thus s=0 corresponds to 1. The decoder must have a memory faculty in

order to obtain the value of slip from the syndrome.

As was pointed out in chapter 2, results for a symmetric slip range may be of limited use. So the previous theorem will be refined to include the case when the slip correction range is unsymmetrical.

Corollary 3.1

If

$$e = \left[\frac{d-1}{2}\right] - t_m \tag{3.23}$$

and

$$t_t \leq u - 1 \leq q^a - 2 \tag{3.24}$$

then there is a (n,k-a) block code derivable from any cyclic (n,k) code which can conjointly correct e or less additive errors and at most t^+ bits of positive slip and t^- bits of negative slip. In addition

$$t_t = t^+ + t^- \quad \text{and} \quad t_m = \max(t^+, t^-) \tag{3.25}$$

Proof

The value of s in (3.19) is restricted by $-t^- \leq s \leq t^+$. Equation (3.20) is true if t is replaced by t_m. With this change part a) of the proof is the same. Also demonstrating that $\left\{\frac{(x^{s-m}-1)}{f(x)}\right\} \neq 0 \bmod (x^n-1)$ where $s \neq m$ and $-t^- \leq s, m \leq t^+$ is sufficient to complete the proof. But $|s-m| \leq t_t$ and $t_t < u$. Thus $f(x) \nmid (x^{s-m}-1)$ modulo (x^n-1) and so the remainder is nonzero.

Q.E.D.

Even though the decoder strategy remains the same, it must be pointed out that in step 2 the correspondence between a particular remainder term and the value of a slip may change when the results of the

corollary are applied. Of course, the remainder of 1 still corresponds to s=0.

A Subset Code Containing a Fixed Pattern

Since word synchronization is valuable information, at least directly to the receiver, it can be sent as part of the message content of each code word. As it will be demonstrated, the error correcting ability of this type of code will be equal to the parent code. However the rate will be altered in a manner directly proportional to the slip correcting range of these codes.

The codes to be constructed below are cyclic codes in which the information bits are located in special blocks with respect to each word. In addition certain of these will always contain a fixed pattern. This pattern enables the decoder to detect if a synchronization loss has occurred and moreover to determine its magnitude and direction. There is nothing esoteric about the pattern to be employed here. It has been used by Mandelbaum [27] in a technique very similar to the one to be presented below. It was introduced by Sellers [5] in a different context for correcting bit-loss and bit-gain errors with burst error correcting codes.

Suppose there is an (n,k) cyclic code. Recall the definition of t^+, t^-, t_t and t_m from Corollary 3.1. Any cyclic code is combinatorially equivalent to a systematic cyclic code which has at least t^- information slots in the beginning components and at least $t^+ + 1$ information bits in the last elements of every code vector. Hence a necessary assumption is $k \leq t_t + 1$. A subset code with q^{k-t_t-1} members is constructed by choosing all of the vectors from the systematic code

which have zeros in the first t^- places and also ones in the last $(t^+ + 1)$ positions. That is $\underline{b}_j$, a member of this subset code, is given by:

$$\underline{b}_j = \left(\underbrace{0,\ldots,0}_{t^-}, b_{j,t^-+1},\ldots,b_{j,n-t^+-2}, \underbrace{1,\ldots,1}_{t^++1}\right) \qquad (3.26)$$

Thus the pattern is t^- zeros first and $(t^+ + 1)$ ones last.

This subset code will be transmitted and the decoder will perform the operation prescribed for the (n,k) systematic cyclic code. If $\underline{b}_j$ and $\underline{b}_k$ are in the subset code and if $t^+ \geq s > 0$, then $\underline{b}_{jk}^{(s)}$ has the following form.

$$b_{jk}^{(s)} = \left(\underbrace{0,\ldots,0}_{t^- - s}, b_{j,t^-+1},\ldots,b_{j,n-t^+-2}, \underbrace{1,\ldots,1}_{t^++1}, \underbrace{0,\ldots,0}_{s}\right) \qquad (3.27)$$

But this is nothing more than a cyclic shift of a code word $\underline{b}_j$ which was in the systematic code. The same is true of the form of $\underline{b}_{jk}^{(s)}$ if $(-t^-) \leq s < 0$. Note that the magnitude and direction of the slip s is easily determined. Let the received vector $\underline{v}$ be given by

$$\underline{v} = \underline{b}_{jk}^{(s)} + \underline{r} \qquad (3.28)$$

Then the additives error vector, $\underline{r}$, is correctable by the decoder since $\underline{b}_{jk}^{(s)}$ is a code vector. This result is summarized by the following theorem.

Theorem 3.5

Given any (n,k) cyclic code there is an $(n, k-t^- -t^+ -1)$ block code which can simultaneously correct e or less additive errors and t^+ bits of positive slip or t^- bits of negative slip.

$$e = \left[\frac{d-1}{2}\right] \tag{3.29}$$

The technique presented above is equivalent to another method which is a coset code of an expurgated code. A subcode of the systematic cyclic code is formed by selecting those members which have t^- zeros in the first components and $(t^+ + 1)$ zeros in the last positions. The coset generator is given by:

$$\underline{c} = \left(0,\ldots\ldots,0,\ \overbrace{1,\ 1,\ldots,1}^{t^+ + 1}\right) \tag{3.30}$$

Therefore a generic term of this coset code is depicted in (3.26). If the additive error-correcting decoding algorithm normally used for the systematic cyclic code is employed on the coset code, all additive errors within $\frac{d}{2}$ will be corrected. The effects of any slip upon the coset generator $\underline{c}$ is easily detected, and the results of Theorem 3.5 are obtained. Another choice of a coset generator is given by:

$$\underline{c} = \left(0,\ldots\ldots,\overbrace{0,\ldots\ldots,0}^{t^+},\ 1\right) \tag{3.31}$$

If this generator is used (or equivalently this pattern), the codes of Shiva and Sequin [28] which they call the "Modified Version" are combinatorially equivalent to this coset code. However the results here are much stronger.

Although this equivalence exists between subset codes with a fixed pattern and coset codes derived from subcodes, the fixed pattern viewpoint is preferable. It is the choice of the pattern which is embedded in the information bits of the code that is important. This pattern must be chosen such that slips are detectable.

Comparison of Results

The best choice of self-synchronizing subset codes depends upon the criteria which the codes must meet. The various results of this chapter were achieved by compromises between error correction capabilities, slip correction capabilities, and code rate. In addition the complexity of the decoding strategy may be modified, and this effects the other performance factors.

There are three main results in this chapter which deal with the simultaneous correction of both additive errors and bit slippage. They are given by Theorems 3.3, 3.4, and 3.5. A comparison will be made between the additive error performances and also between the rates with the slip correction range t as the independent variable. In Theorems 3.3 and 3.4 t will be allowed to assume its maximum value, i.e., $t = q^a-2$ in Theorem 3.3 and $2t = q^a-2$ in Theorem 3.4. Let e_i denote the maximum number of correctable errors as given by Theorem 3.i and let R_i be the corresponding rate. The following quantities are displayed in figure 3.2 for a typical (n,k) cyclic code. Even though t is an integer valued variable, it will be allowed to be real valued here for the sake of graphic clarity.

$$e_3 = \left\lfloor \frac{d-t-1}{2} \right\rfloor \quad ; \quad R_3 = \left(\frac{k-\log_q(t+2)}{n} \right)$$

$$e_4 = \left\lfloor \frac{d-1}{2} \right\rfloor - t \quad ; \quad R_4 = \left(\frac{k-\log_q(2t+2)}{n} \right)$$

$$e_5 = \left\lfloor \frac{d-1}{2} \right\rfloor \quad ; \quad R_5 = \left(\frac{k-2t-1}{n} \right)$$

The error performance of Theorem 3.5, e_5, is always superior to the others but its rate is always inferior. Also its correction range is larger than either of the others. The performances of Theorem 3.3 is slightly better than those of Theorem 3.4. However the former requires an iterative decoding procedure.

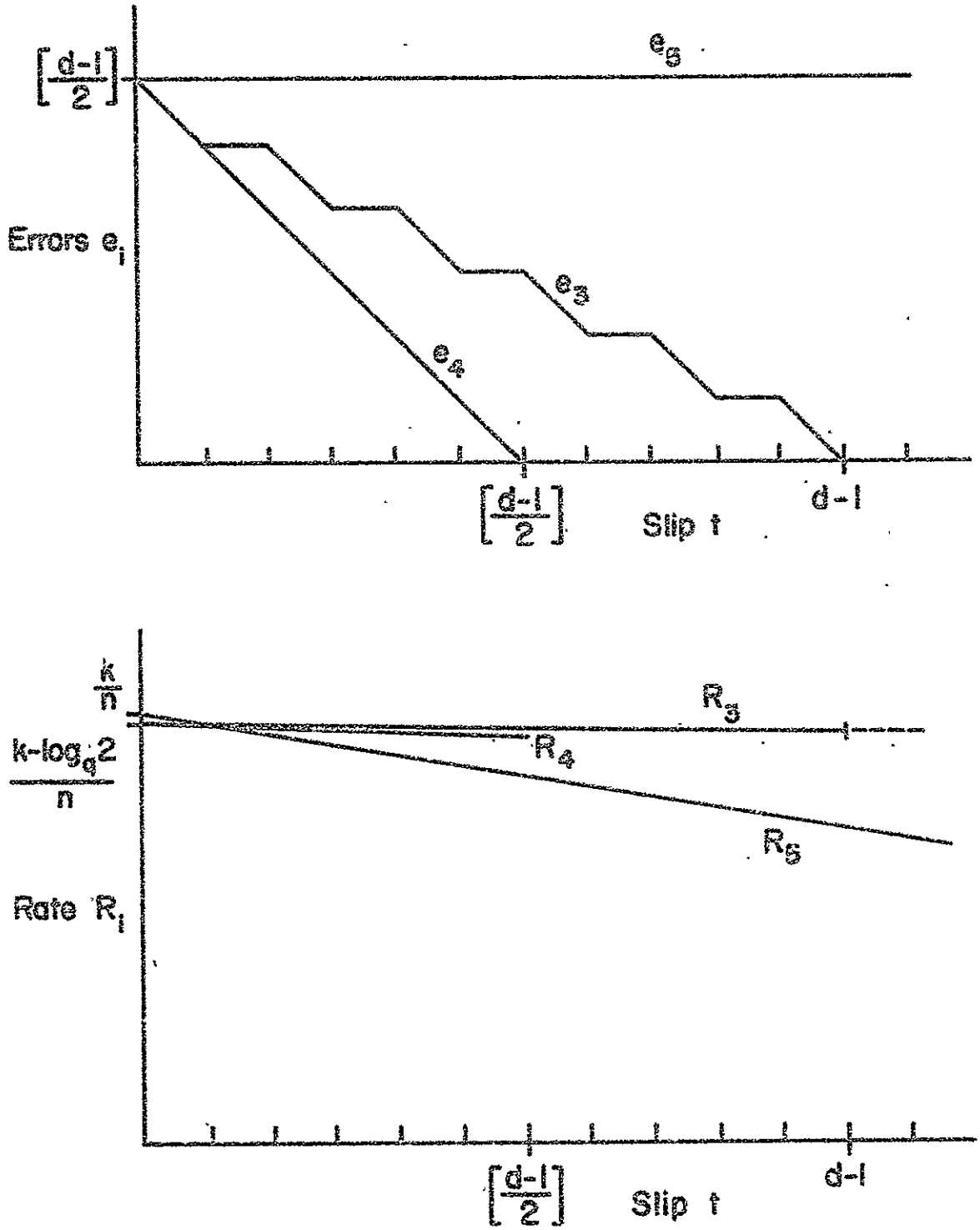


FIGURE 3.2. TYPICAL RATE AND ERROR PERFORMANCE OF SUBSET CODES.

CHAPTER 4

LENGTH ALTERED CODES

Codes which have immunity to synchronization loss as well as to additive error corruption have been constructed in the previous chapters from a known code by modifying some aspects of its structure. However none of these techniques changes the length of the original code. The concept of lengthening a sequence of information digits by appending check digits so as to protect against additive disturbances has a counterpart in dealing with synchronization errors. Two sections of this chapter deal with construction techniques which extend a known code by afixing additional digits to each member so as to check for synchronization loss. The resultant code retains the error-correcting ability of the original code. This is in contrast to the insertion between words of special synchronization sequences, e.g., Barker sequences. These sequences are very often sensitive to additive errors.

There is no analogous concept in coding theory to indicate that shortening a known code would diminish its vulnerability to synchronization errors. However it will be demonstrated that by removing certain portions from every code word the synchronization sensitivity of the code is reduced even in the presence of additive errors. Shiva and Seguin [28] were the first to present any results concerning the shortening of codes for the correction of synchronization errors whereas Caldwell introduced the concept of extending codes for the

same purpose in his work with BCH codes [31,32].

None of the methods employing length alteration use that technique solely. It is always used in conjunction with some other approaches. The additive error performance of these codes is uniformly better than the coset codes or the subset codes of the preceding chapters. But their rates are reduced, and furthermore when no synchronization errors are present, the efficiency of these codes is lower than the parent codes from which they were derived. The decoding strategies recommended for the codes in this chapter have two general procedures in common. First before any other processing the receiver always returns the length of the received vector to that of the original code by either adding or deleting digits depending on the nature of the length alteration. The remaining steps in the decoding are based upon the structure of the original code which generally is better known than that of the altered code.

The results in each section start with the problem of detecting either or both types of errors and conclude with those pertaining to the simultaneous correction of both types. In order not to obscure the salient properties of these codes, in most cases the results concerning symmetrical slip are presented before the unsymmetric case is considered. This dichotomy does not overwhelm the reader with details which can be presented easily as an elaboration of the simpler case with little or no further proof.

Shortened Codes

The basic procedure for shortening a code to be used in this section may be succi outlined in two steps. First select a set of

vectors which have predetermined values either in their leading or trailing positions or in both places. Second remove those positions from every vector transmitted.

Let Δ be an (n,k) cyclic code which is generated by the polynomial g(x). First choose a subset defined as:

$$\left\{\underline{b}_i \leftrightarrow b_i(x) = \left(x^t\gamma(x)g(x) + \alpha_i x^{k-1}g(x)\right) \bmod (x^n-1)\right.$$

$$\left. : \deg \gamma(x) < k-t-1,\ \gamma(0) \neq 0 \text{ and } \alpha_i \neq 0\right\} \qquad (4.1)$$

This is a subset code consisting of those vectors which have t zeros in the first places, a nonzero term in the (t+1)st component since $g(0) \neq 0$ and $\gamma(0) \neq 0$ and finally a nonzero term in the last place since $\alpha_i \neq 0$. These are $(q-1)\,q^{k-t-2}$ choices of information bits represented by $\gamma(x)$ and $(q-1)$ choices of $\alpha_i \neq 0$. Thus the subset has $q^{k-t-2}(q-1)^2$ members. Secondly shorten all the vectors in the subset (4.1) by removing the first t bits. The result is a block code of length $n' = n-t$.

Definition 4.1

Let Σ denote the block code of length $n' = n-t$ and with $q^{k-t-2}(q-1)^2$ members as constructed in the preceding paragraph.

Note that there is a one - one mapping from Σ into the original code Δ. It will be convenient to designate the following subset of the indices of Δ.

$$I = \{\text{index } i : \underline{b}_i \in \Delta \text{ and the shortened version of } \underline{b}_i \text{ is in } \Sigma\} \qquad (4.2)$$

So it is possible to enumerate the members of Σ as $\{\underline{a}_i\}_{i\in I}$ where $\underline{a}_i$ is an n'-tuple and each $\underline{a}_i$ corresponds to exactly one $\underline{b}_i \in \Delta$. Vectors from

Σ will be transmitted. However the decoder will add t zeros to the beginning of each received vector so as to transform every n'-tuple into an n-tuple. Suppose that the received and framed n'-tuple is given by

$$\underline{v}' = \underline{a}_{jk}^{(s)} + \underline{r}' \tag{4.3}$$

$\underline{r}'$ is the additive error n'-tuple and $|s| \leq t$. Note that j and k are both in I of (4.2). After $\underline{v}'$ has been prefixed by t zeros the result is an n-tuple denoted as $\underline{v}$. Now $\underline{v}$ may be written as:

$$\underline{v} = \underline{b}_j^{(s)} + \underline{r} - \underline{z} \tag{4.4}$$

where $\underline{b}_j$ corresponds to $\underline{a}_j$ and $\underline{r}$ is the n-tuple resulting from prefixing the n'-tuple $\underline{r}'$ with t zeros.

$$\underline{z} = \begin{cases} (\underbrace{0,\ldots,0}_{t-s},\ \alpha,\ \underbrace{\text{other terms},\ 0,\ldots,0}_{s}\ \beta,\ \underbrace{\text{other terms}}_{s}); & \text{if } s \geq 0 \\ (\underbrace{\text{other terms},\ \zeta}_{-s}\ \underbrace{0,\ldots,0,}_{t+s}\ \underbrace{\text{other terms},\ \eta,\ 0,\ldots,0}_{-s}); & \text{if } s \leq 0 \end{cases} \tag{4.5}$$

α is the nonzero term in the (t+1)st position of $\underline{b}_j$, and β is the nonzero term in the (t+1)st place of $\underline{b}_k$ which corresponds to $\underline{a}_k$. Whereas ζ is the nonzero term of $\underline{b}_j$ in the nth place and η is the nth term of $\underline{b}_k$ (nonzero of course).

Theorem 4.1

Suppose there is an (n,k) cyclic code which can detect a burst of length at most t in the first t components and also detect another burst of length at most t positions, and in addition detect at most e

additive errors in the last (n-t) positions. Then there is a block code of length $n' = n-t$ and with $(q-1)^2\, q^{k-t-2}$ members which can detect e or less additive errors and t or less bits of slippage in either direction even if both occur simultaneously.

Proof

The block code to be used is Σ in Definition 4.1. Assume that the received vector $\underline{v}'$ is as given by (4.3) with $w(\underline{r}') \leq e$. The decoder will operate on $\underline{v}$ given in (4.4). It suffices to show that an additive error is detected if $0 < |s| \leq t$ or $\underline{r} \neq \underline{0}$ with $w(\underline{r}) \leq e$.

a) $\underline{r} \neq \underline{0}$ and $s = 0$

Since $\underline{r} \neq \underline{0}$ and $w(\underline{r}) \leq e$, $\underline{v}$ has an additive error in the last (n-t) places and no burst errors, i.e., $\underline{z} = \underline{0}$. So by hypothesis the error is detectable.

b) $0 < |s| \leq t$ and any $\underline{r}$ with $w(\underline{r}) \leq e$

Since $s \neq 0$, it follows from (4.5) that $\underline{z}$ has one burst of length s in its first t positions and another one of length s in either the next t places or the last t places. So $\underline{z}$ is a detectable pattern as well as $\underline{r}$ since $w(\underline{r}) \leq e$ and since $\underline{r}$ begins with t zeros.

Q.E.D.

Corollary 4.1

If there is an (n,k) cyclic code, with minimum distance d, and if

$$e = d - 2t - 1 \tag{4.6}$$

then there exists a block code of length $n' = n-t$ with $(q-1)^2\, q^{k-t-2}$ vectors which can detect the concurrent occurrence of e or less additive errors and t or less bits of slippage in either direction.

Proof

The block code is Σ constructed from Δ as above Definition 4.1. The proof follows easily from the theorem because any error-correcting code of minimum distance d will detect the occurrence of two bursts of length t and e additive errors if $e + 2t \leq d - 1$.

Q.E.D.

Theorem 4.2

Let Δ be an (n,k) cyclic code which can correct a burst of length at most t in the first t positions and correct a second burst of length at most t either in the second t places or in the last t places. Furthermore Δ can correct at most e additive errors in the last (n-t) positions. Then it is possible to derive a block code from Δ which has the capability of simultaneously correcting t or less bits of slippage in either direction and e or less additive errors. This modified code has length $n' = n-t$ and contains $(q-1)^2 q^{k-t-2}$ members.

Proof

Consider the Σ code as derived from the Δ code in a manner as described above Definition 4.1. Assume that the slipped and corrupted n'-tuple received at the decoder is $\underline{v}'$ given by (4.3). The decoder strategy is outlined below.

1) Extend $\underline{v}'$ to $\underline{v}$ given in (4.4) by adding t zeros to the beginning of $\underline{v}'$.

2) Correct the additive and burst errors by using the Δ code as a basis for this correction.

3) If the corrected vector from above has a nonzero term anywhere in the first t positions, reframe the received n'-tuple and start at

step 1) again. On the other hand if this corrected vector has t zero in the first places, the additive and slip errors have both been corrected. After step 1) the decoder investigates the n-tuple $\underline{v}$ as given by (4.4). $\underline{z}$ of (4.5) has two bursts which are correctable by the hypothesis. If $w(\underline{r}) \leq e$, then $\underline{r}$ represents a correctable additive error pattern. So the decoded vector in step 2) is $\underline{b}_j^{(s)}$ where $\underline{b}_j$ corresponds to $\underline{a}_j$. But if $s \neq 0$ and $|s| < t$, $\underline{b}_j^{(s)}$ is a cyclic shift of a member of Δ which has a nonzero term somewhere in the first t positions by the very construction of Σ. However if $s = 0$, $\underline{b}_j$ has t zeros in the first places. Since $\underline{b}_j$ corresponds to $\underline{a}_j$, the correct vector has been determined.

Q.E.D.

Corollary 4.2

If there is an (n,k) cyclic code, Δ, with minimum distance d, and if

$$e = \left\lfloor \frac{d-4t-1}{2} \right\rfloor \tag{4.7}$$

then there is a block code (length n' and $(q-1)^2 q^{k-t-2}$ members) which can conjointly correct e or less additive errors and t or less bits of slippage (independent of direction).

Proof

Σ is the block code. If $e + 2t \leq \frac{d-1}{2}$, the cyclic code has all of the properties of the one required in the theorem.

Q.E.D.

Corollary 4.3

There is a block code which can concurrently correct e or less additive errors and t^+ or less bits of slip in the positive direction

and t^- or less bits of negative slip. A sufficient condition for this is the existence of an (n,k) cyclic code which is capable of correcting a burst of length at most $t_m = \max(t^+,t^-)$ in the first t_m positions and a second burst either in the next t^- places of length at most t^- or in the last t^+ positions of length at most t^+, and also e or less additive errors in the last $(n-t_m)$ components. This code has length $(n-t_m)$ and contains $(q-1)^2\, q^{k-t_m-2}$ members.

Proof

The block code is one derived from the cyclic code by the method above Definition 4.1 except with t replaced by t_m. The decoder strategy is the same as in the theorem except again with t replaced by t_m. The proof is obvious by noting the location of the bursts in $\underline{z}$.

Q.E.D.

The synchronization correction techniques inherent in the previous results are achieved by an iterative procedure. It may be desirable to determine both the magnitude and direction of the slip directly at the decoder without any sort of search. In order to accomplish this an (n,k) cyclic code, Δ, generated by the polynomial g(x), must be modified in a slightly different fashion from the way the code, Σ, of Definition 4.1 was derived. The subset to be shortened is given by

$$\left\{\underline{b}_i \leftrightarrow b_i(x) = x^t\gamma(x)g(x) \bmod (x^n-1) \right.$$
$$\left. : \deg \gamma(x) < k-2t-1 \text{ and } \gamma(0) \neq 0\right\} \qquad (4.8)$$

The vectors of this subset code are those vectors of Δ which begin

and end with t zeros and have a nonzero term in the (t+i)st position. These are $(q-1)\,q^{k-2t-1}$ vectors in this subset. Now shorten the subset code by removing the first and last t positions of each vector. Hence a block of length $n'' = n-2t$ has been constructed.

Definition 4.2

Let Σ' denote the block code constructed above. So Σ' is the set of n''-tuples, $\{\underline{a}_i\}_{i\in I'}$ where each $\underline{a}_i$ corresponds to exactly one $\underline{b}_i \in \Delta$ and where

$$I' = \{\text{index } i : \underline{b}_i \in \Delta \text{ and its shortened version is in } \Sigma'\} \tag{4.9}$$

This block code can be used for additive error and slip error correction even if both types of errors occur in the same vector.

Theorem 4.3

Suppose there exists an (n,k) cyclic code which has the capabilities of correcting e or less additive errors occurring in the middle (n-2t) positions of any vector and two bursts each at most t bits long with one occurring somewhere in the first 2t places and the other somewhere in the last 2t places. Then there is a block code of $(q-1)\,q^{k-2t-1}$ members each of length $n'' = n-2t$ which can simultaneously correct e or less additive errors and t or less bits of slip regardless of direction.

Proof

Suppose the slipped and corrupted n''-tuple received at the decoder is given as:

$$\underline{v}'' = \underline{a}_{jk}^{(s)} + \underline{\varepsilon}'' \tag{4.10}$$

$\underline{r}''$ represents the additive errors encountered during transmission. Assume that $w(\underline{r}'') \leq e$ and that $|s| \leq t$. Recall that j and k are both in the set I' of (4.9).

The decoder performs the following strategy. It adds a prefix and suffix to each received n''-tuple of t zeros. The resulting vector $\underline{v}$ is decoded with respect to the minimum distance procedure relative to the cyclic code A. The position index of the first nonzero term of the decoded vector is subtracted from the value (t+1), and it gives the magnitude and the direction of the slip.

It is possible to write the extended version of $\underline{v}''$ as:

$$\underline{v} = \underline{b}_j^{(s)} + \underline{r} - \underline{z} \tag{4.11}$$

$\underline{b}_j$ corresponds to $\underline{a}_j \in \Sigma'$ and $\underline{r}$ is the n''-tuple $\underline{r}''$ extended by adding t zeros to the beginning and to the end.

$$\underline{z} = \begin{cases} \left(\underbrace{0,\ldots,0}_{t-s},\underbrace{\alpha,\text{other terms}}_{s},0,\ldots,0,\underbrace{\beta,\text{other terms}}_{s},\underbrace{0,\ldots,0}_{t}\right); \\ \qquad\qquad \text{if } s \geq 0 \\ \left(\underbrace{0,\ldots,0}_{t},\underbrace{\text{other terms}}_{-s},0,\ldots,0,\underbrace{\text{other terms}}_{-s},\underbrace{0,\ldots,0}_{t+s}\right); \\ \qquad\qquad \text{if } s \leq 0 \end{cases} \tag{4.12}$$

α is the first nonzero term of $\underline{b}_j$ while β is that of $\underline{b}_k$.

$(\underline{r}-\underline{z})$ is a vector consisting of a combination of additive and burst errors which is correctable by the hypothesis. So the decoded vector is $\underline{b}_j^{(s)}$. But $\underline{b}_j$ begins and ends with t zeros and has a nonzero term in the (t+1)st position. Therefore $\underline{b}_j^{(s)}$ begins with (t-s) zeros and always has a nonzero term in the (t-s+1) position. Subtracting it from the quantity (t+1) gives s.

Q.E.D.

Since any code of minimum distance d can correct the two bursts and the additive errors required by the hypothesis of the theorem if $\frac{d-1}{2} \geq e + 2t$, the proof of the following corollary parallels that of the theorem.

Corollary 4.4

Let Δ be an (n,k) cyclic code with minimum distance d. If

$$e = \left[\frac{d-4t-1}{2}\right] \tag{4.13}$$

then there is a block of length $n'' = n-2t$ which is derivable from Δ and which is capable of simultaneously correcting at most e additive errors and at most t bits of slip (independent of the direction). This code is composed of $(q-1)\, q^{k-2t-1}$ vectors and a decoder can determine both the magnitude and the direction of the slip without any search procedure.

An alteration of the code construction technique used for the symmetric case produces similar results when the expected slip is in an unsymmetrical range.

Corollary 4.5

There is a block code which has the correction capabilities of at most e additive errors and either at most t^+ bits of positive slip or at most t^- bits of negative slip. A sufficient condition for this is the existence of an (n,k) cyclic code which can correct e or less additive errors occurring in those places from the (t^++1)st to the $(n-t^-)$st inclusively and also can correct either a burst in the first t^+ positions and a second one between the $(n-t^+-t^-)$th and the $(n-t^-+1)$ places or a burst between the t^+th place and the (t^++t^-+1)th place and

a second burst in the last t^- positions. Let $t_t = t^+ + t^-$. The length of this block code is $n'' = n+t_t$ and it has $(q-1)\,q^{k-t_t-1}$ members.

Proof

The subset code to be shortened is the collection of those vectors which begin with t^+ zeros and end with t^- zeros and have a nonzero term in the (t^++1)st position. The subset is shortened by removing the first t^+ positions and the last t^- positions.

The decoder adds zeros in these places and after additive error correction the position index of the first nonzero term in the decoded vector is subtracted from t^++1 to obtain the magnitude and direction of the slip. The proof is obvious once the form of $\underline{z}$ in an equation similar to (4.11) for the received vector after the zeros have been added is given for these circumstances.

$$\underline{z} = \begin{cases} \left(\underbrace{0,\ldots,0}_{t^+-s},\underbrace{\alpha,\text{other terms}}_{s},0,\ldots,0,\underbrace{\beta,\text{other terms}}_{s},\underbrace{0,\ldots,0}_{t^-}\right); & \text{if } s \geq 0 \\ -\left(\underbrace{0,\ldots,0}_{t^+},\underbrace{\text{other terms}}_{-s},0,\ldots,0,\underbrace{\text{other terms}}_{-s},\underbrace{0,\ldots,0}_{t^-+s}\right); & \text{if } s \leq 0 \end{cases} \tag{4.14}$$

Q.E.D.

Extended Subset Codes

Each code word is lengthened by buffering it with sequences which are suitably chosen parts of the word itself. This reduces the effects of synchronization errors, but it does not add enough redundancy so that their effects may be confused with or cancelled by those due to

additive errors. However selecting a subset of the original code allows the separation of the effects of both types of errors.

Let Δ and Δ' be codes as defined in Definition 3.1. Furthermore define the set of integers J as:

$$J = \{\text{index } j : \underline{b}_j \in \Delta'\} \tag{4.15}$$

As in several cases before, let t^+ be the maximum number of bits or positive slip to be corrected while t^- denotes the negative slip. Furthermore define two other symbols.

$$t_t = t^+ + t^- \quad \text{and} \quad t_m = \max(t^+, t^-) \tag{4.16}$$

Now consider the coset of the subcode Δ' as given in (3.1), i.e., $\{\underline{b}_i + \underline{g}\}_{i \in J}$. This coset will be cyclically extended to yield a block code of length $n' = n + t_t$ by affixing a prefix of t^- bits and a suffix of t^+ bits to each code word. The prefix is the last t^- elements of the word in the same relative position if $t^- \leq n$ or is $\lceil \frac{t^-}{n} \rceil$ repetitions of the word preceded by the last $(t^- - n\lceil \frac{t^-}{n} \rceil)$ places of the word in the same order if $t^- > n$. Similarly the suffix is the first t^+ positions if $t^+ \leq n$ or is $\lceil \frac{t^+}{n} \rceil$ repetitions of the word followed by the first $(t^+ - n\lceil \frac{t^+}{n} \rceil)$ places of the word if $t^+ > n$. This cyclic extension technique is made more graphic in the following explanation. If $\underline{b}$ is a member of the coset code (3.1), its extended version $\underline{c}$ is a member of a $(n+t_t, k-a)$ block code.

$$\underline{b} = (b_0, b_1, \ldots, b_{n-1}) \tag{4.17}$$

$$\underline{c} = \Big(b_{n-t^- + n\lceil \frac{t^-}{n} \rceil - 1}, \ldots, b_{n-1}, \underbrace{\underline{b}, \ldots, \underline{b}}_{\lceil \frac{t^-}{n} \rceil \text{ vectors}}, \underbrace{\underline{b}, \underline{b}, \ldots, \underline{b}}_{\lceil \frac{t^+}{n} \rceil \text{ vectors}}, b_0, \ldots, b_{t^+ - n\lceil \frac{t^+}{n} \rceil - 1}\Big)$$

The above construction procedure is implicitly contained in the following definition.

Definition 4.3

Let $\mathcal{C}$ denote this cyclically extended code with members $\underline{c}_i$ corresponding to $(\underline{b}_i + \underline{g})$ for each $i \in J$.

The $\mathcal{C}$ code will be transmitted. Throughout this section the first step in the decoding strategy will be to treat the received word as if no slip had occurred and then to remove the appended parts of the n'-tuple, i.e., the decoder frames the n bits from the $(t^- + 1)$st position to the $(n + t^-)$st place inclusively of the received vector. Suppose that the received n'-tuple is given by $\underline{y}'$.

$$\underline{y}' = \underline{c}_{jk}^{(s_o)} + \underline{r}' \tag{4.18}$$

$\underline{r}'$ represents additive errors, and it will be assumed that it has at most e nonzero components in any burst of length n or less. Furthermore assume that $t^+ \geq s \geq -t^-$. Notice that j and k are both in the set J, (4.15). The first step of the decoding strategy will yield an n-tuple, $\underline{y}$. Because of the construction of members of $\mathcal{C}$, $\underline{y}$ may be written as follows.

$$\underline{y} = \underline{b}_j^{(s)} + \underline{g}^{(s)} + \underline{r} \tag{4.19}$$

$\underline{r}$ has the $(t^- + 1)$st to the $(n + t^-)$st components of $\underline{r}'$ in its n positions while $\underline{b}_j + \underline{g}$ corresponds to the cyclically extended vector $\underline{c}_j$. Also s is given by:

$$s = s_o \left(1 - \frac{n}{|s_o|}\left[\frac{|s_o|}{n}\right]\right) \tag{4.20}$$

Theorem 4.4

Suppose there is an (n,k) cyclic code with minimum distance d. It is possible to construct an $(n+t_e, k-a)$ block which can detect the simultaneous occurrence of at most e additive errors and either t^+ bits of slip in the positive direction or t^- bits in the negative direction as long as the following are satisfied:

$$t^+ - n\left[\frac{t^+}{n}\right] \leq u - 1 \leq q^a - 2$$
$$t^- - n\left[\frac{t^-}{n}\right] \leq u - 1 \leq q^a - 2 \tag{4.21}$$

and

$$e = d - 1 \tag{4.22}$$

Furthermore equality is possible in (4.21) if and only if f(x) is a primitive polynomial.

Proof

The first step of the decoding process gives $\underline{y}$ of (4.19). The next step is to perform additive error detection on $\underline{y}$ as if the code were the coset code of A' as given in (3.1). It suffices to consider two cases which are mutually exclusive and exhaust all the possibilities.

a) $\underline{r} \neq \underline{0}$ and $s = 0$

Since $e = d-1$, normal error detecting procedures indicate an error since $(\underline{b}_j + \underline{g}) \in A'$.

b) $s \neq 0$ with $-(t^- - n[\frac{t^-}{n}]) \leq s \leq (t^+ - n[\frac{t^+}{n}])$ (or $-t^- \leq s_o \leq t^+$ from (4.20)).

If under these conditions the following inequality holds, the decoder will have detected an error.

$$\min_{i \in J} w(\underline{v} - \underline{b}_i - \underline{g}) > 0 \tag{4.23}$$

However

$$w(\underline{v} - \underline{b}_i - \underline{g}) \geq w\left(\underline{b}_j^{(s)} + \underline{g}^{(s)} - \underline{b}_i - \underline{g}\right) - w(\underline{e}) \tag{4.24}$$

But $\left(\underline{b}_j^{(s)} + \underline{g}^{(s)}\right)$ cannot be a member of the coset of the subcode Δ' by an argument identical with part b) of the proof of Theorem 3.1. So $w\left(\underline{b}_j^{(s)} + \underline{g}^{(s)} - \underline{b}_i - \underline{g}\right) \geq d$ for any $i \in J$. Thus (4.23) becomes:

$$\min \; w\left(\underline{b}_j^{(s)} + \underline{g}^{(s)} - \underline{b}_i - \underline{g}\right) \geq d - e \tag{4.25}$$

Now $d-e = 1$ from (4.22). The "furthermore" statement follows as it did in Theorem 3.1.

Q.E.D.

This theorem suggests an approach for the correction of both types of errors.

Theorem 4.5

The conclusion of Theorem 4.4 is valid for the conjoint correction of both types of errors under the same hypothesis except that the expression for e in (4.22) is replaced by:

$$e = \left[\frac{d-1}{2}\right] \tag{4.26}$$

Proof

The first step performed by the decoder is as before to frame an n-tuple, $\underline{v}$, from the received n′-typle, $\underline{v}'$. Next it performs additive error correction by treating the framed vector as a corrupted vector form the coset code of (3.1). If the number of additive errors which have occurred is less than or equal to e, the decoder has determined

the transmitted vector. However if the number of errors exceeds e, then the strategy will have the decoder reframe and perform the additive error-correcting step again.

Consider the two cases below:

a) $s = 0$

Then

$$\min_{i \in J} w(\underline{v} - \underline{b}_i - \underline{g}) \leq w(\underline{r}) \leq e \qquad (4.27)$$

b) $s \neq 0$ with $(t^+ - n[\frac{t^+}{n}]) \geq s \geq -(t^- - n[\frac{t^-}{n}])$

Equation (4.25) from part b) of the previous proof is still valid and is pertinent here. But from (4.26), $d-e \geq d - \frac{(d-1)}{2} = \frac{d}{2} + \frac{1}{2} \geq e+1$. Hence using the iterative decoding strategy outlined above gives the proper vector.

Q.E.D.

Altering the decoding strategy employed in the previous theorem leads to a different result. The decoding procedure used here gives the magnitude and direction of any slip as well as the coset word. However this extra feature requires an increase in the decoder's complexity and allows a smaller slip-correction range. This result was first presented by Weldon [34], who generalized the work of Caldwell [31] and Bose and Chadwell [32].

Theorem 4.6

From any (n,k) cyclic code with minimum distance d, it is possible to construct an $(n+t_t, k-a)$ block which has the capability of simultaneously correcting e or less additive errors and either t^+ bits of positive slip or t^- bits of negative as long as following conditions

are fulfilled:

$$t_t - n([\frac{t}{n}^+] + [\frac{t}{n}^-]) \leq u - 1 \leq q^a - 2 \qquad (4.28)$$

and

$$e = \left[\frac{d-1}{2}\right] \qquad (4.29)$$

Equality is achieved in the first equation if and only if f(x) is a primitive polynomial.

Proof

The first step in decoding is to frame the n-tuple. Next additive error correction with respect to the larger code Δ is performed. Finally the syndrome with respect to (f(x)g(x)) modulo (x^n-1) is computed. The value of this syndrome gives the magnitude and direction of the slip.

According to this strategy if $w(\underline{r}) \leq e$ in $\underline{v}$, the decoder decides that $\underline{b}_j^{(s)} + \underline{g}^{(s)}$ was transmitted. Just as in the proof of Corollary 3.1, the syndromes (the remainder term of $\frac{(b_j(x) + g(x))x^s}{f(x)g(x)}$) are distinct if the total range of s is less than or equal to u-1. Thus the total range is $(t_t - n([\frac{t}{n}^+] + [\frac{t}{n}^-]))$.

Q.E.D.

It is apparent that the codes of this section can be used even when the range of the slip is quite large and even when it is multiples of the original length. Of course the rate is directly and adversely effected. The problem of dealing with wide-range slips is treated in the following chapter; so any discussion about these codes from that viewpoint will be transposed to there.

Extended Coset Codes

In this section the coset of a code will be cyclically extended. This modification technique yields codes which have a higher rate than those of the previous section at larger slip values which are still less than $\frac{n}{2}$. This approach is a compromise between the reduction of rate which is found in the extended subset codes and the reduction of additive error correction capability resulting from the use of the coset codes of Chapter 2. This technique tends to moderate the loss in each of the performance criteria.

Consider a coset of an (n,k) cyclic code, Δ, which has minimum distance d. Let $\underline{c}$ be the generic coset generator. So $\{\underline{b}_i + \underline{c}\}_{i=0}^{M-1}$, with $M = q^k$, is a coset code which will be cyclically extended by prefixing each n-tuple by its last t elements, maintaining their respective order, and suffixing each one by its first t components in their order.

Definition 4.4

Let Ω be the (n',k) block code constructed from the coset code $\{\underline{b}_i + \underline{c}\}_{i=0}^{M-1}$ by cyclically extending it at each end by t position. $n' = n+2t$. Furthermore let $\underline{f}_i \in \Omega$ correspond to $(\underline{b}_i + \underline{c})$ of the coset code.

The block code Ω will be used for transmission. Hence a typical received n'-tuple is $\underline{v}'$.

$$\underline{v}' = \underline{f}_{jk}^{(s)} + \underline{r}' \tag{4.30}$$

$\underline{r}'$ represents the additive errors. $\underline{f}_j$ and $\underline{f}_k$ are both in Ω. It will be assumed throughout the remaining parts of this section that $|s| \leq t$

and that every burst of length n or less of $\underline{r}'$ has weight of e or less. The first step in every decoding strategy to be discussed here will be to disregard the first t and the last t components of $\underline{y}'$ in order to obtain an n-tuple $\underline{y}$. It is obvious from the construction of Ω that $\underline{y}$ has the following form.

$$\underline{y} = \underline{b}_j^{(s)} + \underline{r} + \underline{c}^{(s)} \tag{4.31}$$

Because of the assumptions concerning $\underline{y}'$, $w(\underline{r}) \leq e$ and $|s| \leq t$. The remaining steps of the decoding strategy will always process the n-tuple, $\underline{y}$, using the structure of the coset code. The form of the coset generator $\underline{c}$ and the exact decoder operations are independent variables at this point.

Theorem 4.7

The existence of an (n,k) cyclic code with minimum distance d and the requirement that

$$e = \left\{\left[\frac{d-3}{2}\right], \left[\frac{2(n-t-1)}{t+1}\right]\right\} \tag{4.32}$$

are sufficient to imply the existence of an (n+2t, k) block code which can detect the conjoint occurrence of at most e additive errors and or less bits of slip (in either direction).

Proof

The form of a coset generator is given by:

$$\underline{e} = \Big(0,\ldots,0,\underbrace{0,\ldots,0,1}_{t+1},\underbrace{0,\ldots,0,1}_{t+1},0,\ldots\ldots\ldots,0,1,\underbrace{0,\ldots,0,1}_{t+1}\Big) \tag{4.33}$$

$\left[\frac{e+2}{2}\right]$ blocks

This form exists if

$$n \geq (t+1)\frac{(e+2)}{2} \quad \text{or} \quad \frac{2(n-t-1)}{t+1} \geq e \tag{4.34}$$

It is sufficient to require that the received vector $\underline{v}$ of (4.31) is not a member of the coset code if $\underline{r} \neq \underline{0}$ or $0 < |s| \leq t$. Under these conditions this is equivalent to the statement below.

$$\min_i \; w(\underline{v} - \underline{b}_i - \underline{c}) > 0 \tag{4.35}$$

All situations are covered by two cases.

a) $\underline{r} \neq \underline{0}$ and $s = 0$

$$\min_i \; w(\underline{v} - \underline{b}_i - \underline{c}) = w(\underline{r}) > 0 \tag{4.36}$$

b) $0 < |s| \leq t$ and any $\underline{r}$ such that $w(\underline{r}) \leq e$

$$w(\underline{v}-\underline{b}_i-\underline{c}) \geq \min\left\{w\left(\underline{c}^{(s)}-\underline{c}+\underline{r}\right),\ \left(w(\underline{b}_j^{(s)}-\underline{b}_i)-w(\underline{c}^{(s)}-\underline{c})-w(\underline{r})\right)\right\}$$

$$\geq \min\left\{\left(w(\underline{c}^{(s)}-\underline{c})-e\right),\ \left(d-w(\underline{c}^{(s)}-\underline{c})-e\right)\right\} \tag{4.37}$$

From (4.33), $w(\underline{c}^{(s)}-\underline{c}) = 2\left[\frac{e+2}{2}\right]$ as long as $s \neq 0$, $|s| \leq t$. Furthermore $e+1 \leq 2\left[\frac{e+2}{2}\right] \leq e+2$. Combining these statements gives:

$$\min_i \; w(\underline{v} - \underline{b}_i - \underline{c}) \geq \min\{(e+1-e),\ (d-(e+2)-e)\} \tag{4.38}$$

However (4.32) implies that $e \leq \frac{d-3}{2}$; so $d-2e-2 \geq 1$.

Q.E.D.

Theorem 4.8

Again assuming the existence of Δ, there is an $(n+2t,k)$ block code which has the ability of not only detecting the simultaneous presence of at most e additive errors and at most t bits of slip

(independent of direction) but also classifying the nature of the errors provided that:

$$e = \min \left\{ \left[\frac{d-3}{4} \right], \left[\frac{n-t-1}{t+1} \right] \right\} \tag{4.39}$$

Proof

The coset generator is specified below.

$$\underline{c} = \left(0,\ldots,0,\underbrace{0,\ldots,0,1,0,\ldots,0,1,0,\ldots\ldots,0,1,0,\ldots,0,1}_{(e+1)\text{ blocks}}\right) \tag{4.40}$$

Thus a sufficient condition for this form of $\underline{c}$ is:

$$n \geq (e+1)\ (t+1) \quad \text{or} \quad \left(\frac{n-t-1}{t+1}\right) \geq e \tag{4.41}$$

The proof of the detection part parallels that of the previous one. Thus its proof will be condensed. The proof of case a) is identical and (4.37) of case b) is still valid. However $w(\underline{c}^{(s)}-\underline{c}) = 2(e+1)$ if $0 < |s| \leq t$. So

$$\min_i \ (\underline{v} - \underline{b}_i - \underline{c}) \geq \min \{(2e+2-e), (d-2(e+1)-e)\} \tag{4.42}$$

$$\geq \{(e+2), (e+1)\}$$

The last inequality comes from the use of (4.39) which implies $4e \leq d-3$ or equivalently $d-3e-2 \geq e+1$.

If the received vector's only corrupting influence has been additive errors, it will be within a Hamming distance of at most e from a coset vector. The equality in (4.36) demonstrates this. However if any slip has occurred, (4.42) above shows that the received vector

must be at least a distance of (e+1) from any coset vector. Therefore it is possible to distinguish between the occurrence of additive errors alone and any combination of nonzero slip and additive errors.

Q.E.D.

If the decoder strategy outlined for Theorem 2.3 is employed with the vector $\underline{v}$, the previous theorem is the basis for a correction procedure. Equation (4.42) in the above proof guarantees the uniqueness of correct slip. This result will be stated as a theorem but the proof will be omitted since it is obvious from the steps of the strategy and the steps of the previous proof.

Theorem 4.9

Suppose there is an (n,k) cyclic code with minimum distance d. Then it is possible to cyclically extend it to form an (n+2t,k) block code which can concurrently correct at most e additive errors and at most t bits of slip regardless of its direction whenever (4.39) is true.

The decoding technique employed for correction may be unappealing in certain situations. So another result which requires a different decoding strategy is presented. It implements a decoding procedure which determines the pairs of integers which comprise the following set.

$$\left\{j,\tau : w\left(\underline{v}-\underline{b}_j^{(\tau)}-\underline{c}^{(\tau)}\right) \triangleq L_{j\tau} \text{ is a minimum for } 0 \le j \le M-1 \text{ and } |\tau| \le t\right\} \tag{4.43}$$

Under suitable conditions it will be shown that this set is a singleton for each received vector $\underline{v}$, (4.31).

Theorem 4.10

It is possible to construct an (n+2t,k) block code from any (n,k) cyclic code having minimum distance d. This block code has the ability of correcting e or less additive errors and t or less bits of slip - both in direction and magnitude - if

$$e = \min\left\{\left[\frac{d-3}{4}\right], \left[\frac{n-2t-1}{2t+1}\right]\right\} \tag{4.44}$$

Proof

The block code is of course the Ω code of Definition 4.4 using the form of $\underline{c}$ given here.

$$\underline{c} = \Big(0,\ldots,0,\underbrace{\underbrace{0,\ldots,0,1}_{2t+1},\underbrace{0,\ldots,0,1}_{2t+1},0,\ldots,0,1,0,\ldots\ldots\ldots,0,1,\underbrace{0,\ldots,0,1}_{2t+1}}_{(e+1)\text{ blocks}}\Big) \tag{4.45}$$

This and similar forms of $\underline{c}$ will exist because from (4.44),

$$e \leq \frac{(n-2t-1)}{2t+1} \quad \text{or} \quad (e+1)\ (2t+1) \leq n.$$

It suffices to consider three cases which exhaust all situations and show that $L_{i\tau} > L_{js}$ for any combination of i and τ such that $i \neq j$ or $\tau \neq s$, $|\tau| \leq t$.

a) $i \neq j$ and $\tau = s$

$$L_{is} \geq w\left(\underline{b}_j^{(s)} - \underline{b}_i^{(s)}\right) - w(r)$$

$$\geq d - e \geq 4e + 3 - e = 3(e+1) \tag{4.46}$$

The use of (4.44) gives the last inequality. Let

$$\hat{i} = \left\{ i : \underline{b}_{jj}^{(s)} - \underline{b}_{ii}^{(\tau)} \right\}$$

b) $i = \hat{i}$ and $\tau \neq s$

$$L_{\hat{i}\tau} \geq w\left(\underline{c}^{(s)} - \underline{c}^{(\tau)}\right) - w(\underline{r}) \tag{4.47}$$

However as long as $|s|$, $|\tau| \leq t$, and $s \neq \tau$,

$$w\left(\underline{c}^{(s)} - \underline{c}^{(\tau)}\right) = 2(e+1) \tag{4.48}$$

$$L_{\hat{i}\tau} \geq 2(e+1)-e = e+2 \tag{4.49}$$

c) $i \neq \hat{i}$ and $\tau \neq s$, and $|s|$, $|\tau| \leq t$

$$L_{i\tau} \geq w\left(\underline{b}_{j}^{(s)} - \underline{b}_{i}^{(\tau)}\right) - w\left(\underline{c}^{(s)} - \underline{c}^{(\tau)}\right) - w(\underline{r})$$

$$\geq d - 2(e+1) - e \geq e + 1 \tag{4.50}$$

Again the last inequality results from (4.44).

Now $L_{js} \leq e$ and so the required condition for all of these cases true.

Q.E.D.

The extension of this result to the situation in which unsymmetric slip is allowed is straightforward and so its proof is omitted.

Corollary 4.6

The theorem is valid for t^+ bits of positive and t^- bits of negative slip if the variable 2t is replaced by $t_t = t^+ + t^-$. ($\underline{c}$ is of the same form as (4.45) except that 2t is exchanged for t_t).

Comparison of Results

In order to present a comparison of the three techniques introduced in this chapter, a result from each of the three sections will be considered. All will deal with the simultaneous correction of symmetric slip and additive errors, and each represents the most powerful result for its type of protection. The results are given by Corollary 4.2 (Shortened Codes), Theorem 4.5 (Extended Subset Codes), and Theorem 4.9 (Extended Coset Codes). A comparison among the additive error performances of these three and among the rates will be made using the slip correction range t as the independent variable.

The subscripting of certain variables will be accomplished by using the last digit of the number of the theorem or corollary to which it pertains. The error performance, e_2, and the rate, R_2, of Corollary 4.2 are given by:

$$e_2 = \left\lfloor \frac{d-4t-1}{2} \right\rfloor$$

$$R_2 = \frac{k-t-Q}{n-t} \tag{4.51}$$

$$Q = 2(1 - \log_q(q-1)) \tag{4.52}$$

The maximum number of correctable errors using the technique of Theorem 4.5 is e_5

$$e_5 = \left\lfloor \frac{d-1}{2} \right\rfloor \tag{4.53}$$

Since the situation of symmetric slip is being considered, the slip correction range is constrained by the following inequality.

$$t - n\left[\frac{t}{n}\right] \leq q^a - 2 \tag{4.54}$$

It will be assumed that $t < n$ and that t always is maximum with respect to these conditions, i.e., $t=q^a-2$. Then the rate is given by:

$$R_5 = \frac{k-\log_q(t+2)}{n+2t} \tag{4.55}$$

The results for extended coset codes from Theorem 4.9 are given below.

$$e_9 = \min\left\{\left[\frac{d-3}{4}\right], \left[\frac{n-t-1}{t+1}\right]\right\}$$

$$R_9 = \frac{k}{n+2t} \tag{4.56}$$

These quantities are compared in Figure 4.1 for a typical (n,k) cyclic code. The independent variable t is allowed to be real valued instead of integer valued so that curves and not series of dots appear in the figure. Employing the bound on the minimum distance, $d \leq n-k+1$ [26], it is possible to show that $\left[\frac{d-1}{4}\right] \leq \left[\frac{n}{2}\right]$ and that $\frac{d-3}{4} \leq \frac{n-2}{4}$. So the position of $\left[\frac{d-1}{4}\right]$ to the left of $\left[\frac{n}{2}\right]$ will always be true and the $\left[\frac{d-3}{4}\right]$ term in the expression for e_9, (4.56), will always be dominant for $t \leq 1$.

Shortened codes have a better error correction performance, e_2, at small values of slip range than that of the extended coset codes e_9. But the rate of the former, R_2, is poorer than that of the latter, R_9. The shortened codes always are inferior to the extended subset codes, and they also have a limited useful slip correction range. Nevertheless in certain instances shortened codes may require less complexity to implement than either of the others. The rate R_9 of extended coset

codes is always superior to that of the extended subset codes. However the superiority of the error correction performances is reversed.

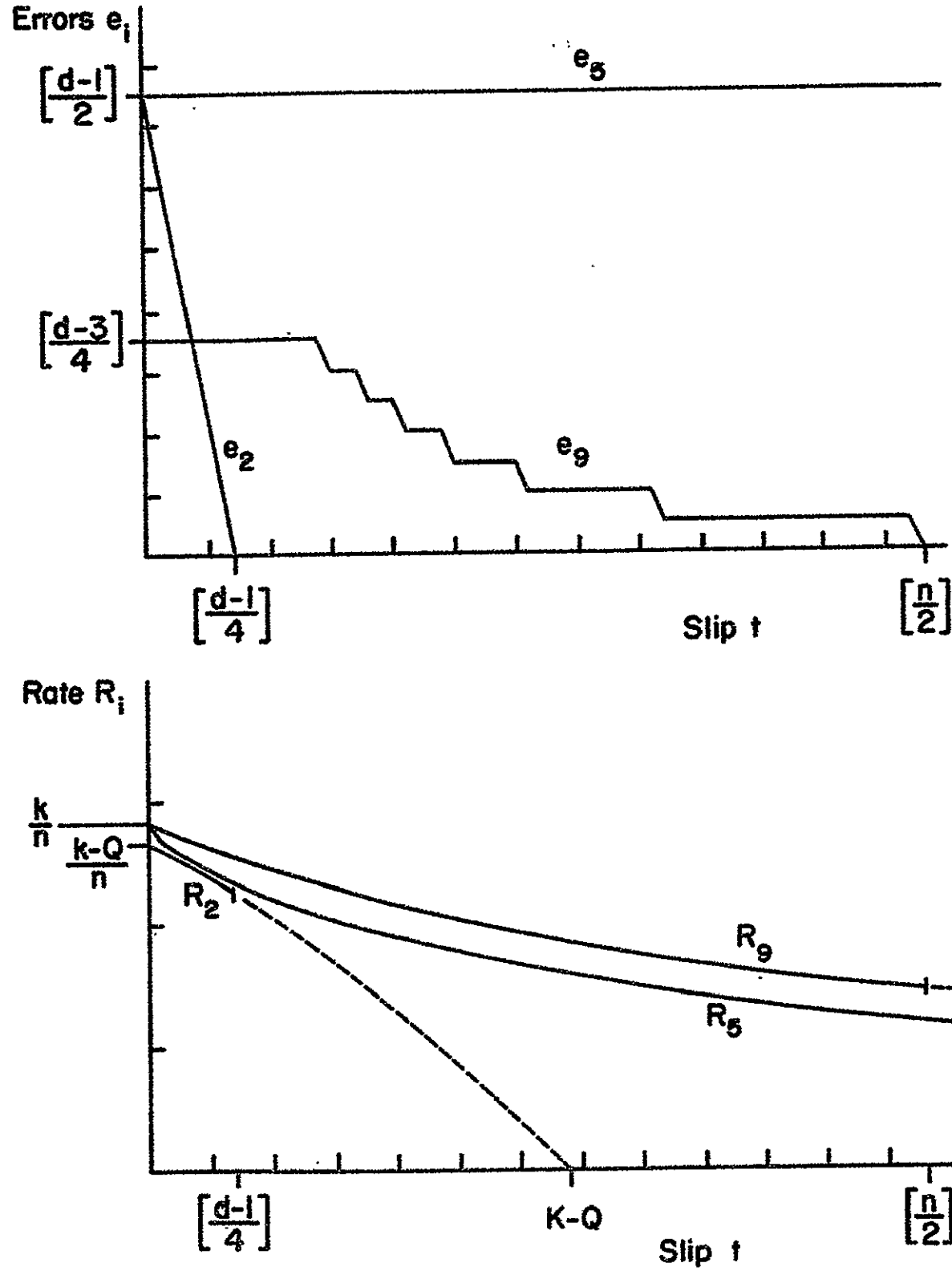


FIGURE 4.1 . TYPICAL RATE AND ERROR PERFORMANCE OF LENGTH ALTERED CODE.

CHAPTER 5

INTERLEAVED AND CONCATENATED CODES

The codes to be constructed in this chapter are designed to protect against slips which lie in a range of at least half of the code's length. For example if a block code has length ℓ, then there are codes which will be able to protect against the conjoint occurrence of additive errors and slips in the range from $-[\frac{\ell}{2}]$ to $[\frac{\ell}{2}]$ bits. The terms "protection" or "security" are used throughout this chapter in a general sense to mean either detection, detection and classification, or correction capabilities when dealing with some type of error.

Two different approaches for achieving a wide range of slip protection will be presented. One uses an interleaving technique while the other combines concatenation with interleaving. The interleaving of codes has been used in conjunction with burst error portection. Concatenating two codes was introduced so that the advantages of long codes for additive error protection could be gained by a more economical implementation. The extended subset codes constructed in Chapter 4 can also protect against large values of slip. So they will be compared with the two approaches to be developed here. However it will be shown that these approaches are superior in performance.

The results to be presented are of a very general nature. They may be coupled with any of the other codes contained in previous chapters. Therefore the following definition is necessary.

Definition 5.1

Let Λ denote a block code of length ℓ which can protect against the simultaneous occurrence of e or less additive errors and t or less bits of slip in either direction. Denote the M members of Λ by $\{\underline{f}_i\}_{i=0}^{M-1}$.

Interleaved Codes

In order to protect against burst errors, one approach is to interleave or interlace the components of several code words into a new order before transmitting them. Then the receiver reorders the components to reconstruct the original code words before any decoding is performed. The intent of such a scheme is to reduce the effects of a burst of errors on each code word by spreading the errors over several words. Similar logic can be applied to the case of synchronization errors. Smaller values of slip may be allotted to each of several code words by interleaving them. If a code is used which offers synchronization and additive error protection, then the overall performance of this code when it is interleaved always is increased.

One representation of the output of the encoder of the code is a stream of ℓ-dimensional vectors

$$\cdots,\underline{f}_{j_{-2m-1}},\cdots,\underline{f}_{j_{-m-1}},\underline{f}_{j_{-m}},\cdots$$

$$\cdots,\underline{f}_{j_{-1}},\underline{f}_{j_0},\underline{f}_{j_1},\cdots,\underline{f}_{j_{m-1}},\underline{f}_{j_m},\cdots,\underline{f}_{j_{2m-1}},\cdots \qquad (5.1)$$

Instead of sending this stream, it will be interleaved to order m. This process is described by depicting the interleaving of the m

ℓ-tuples, $\underline{f}_{j_o}, \ldots, \underline{f}_{j_{m-1}}$. First form the $(m \times \ell)$ array X which has these vectors as rows.

$$\begin{bmatrix} \underline{f}_{j_o} \\ \underline{f}_{j_1} \\ \vdots \\ \underline{f}_{j_{m-1}} \end{bmatrix} = X = (x_{i,j}) \qquad \begin{matrix} 0 \le j \le m-1 \\ 0 \le j \le \ell-1 \end{matrix} \tag{5.2}$$

Now perform the interleaving by sending each column of X in succession instead of each row. Hence the stream of elements is:

$$\ldots, \underbrace{x_{o,o}, x_{1,o}, \ldots, x_{m-1,o}}_{\text{first column}}, \underbrace{x_{o,1}, \ldots, x_{m-1,1}}_{\text{second column}}, x_{o,2}, \ldots$$

$$\ldots, \underbrace{x_{o,\ell-1}, \ldots, x_{m-1,\ell-1}}_{\text{last column}}, \ldots\ldots\ldots \tag{5.3}$$

If there were no errors or slip, the receiver would reconstruct the array X and then the decoder would extract the information bits from each row (a code word). However suppose that additive errors are introduced and that there is a positive slip s_o. It is possible to write s_o,

$$s_o = (s-1)\, m+u, \quad s \ge 1, \quad 0 \le u < m \tag{5.4}$$

The array Y formed at the receiver becomes:

$$\begin{bmatrix} \underline{f}^{(s-1)}_{j_u, j_{m+u}} & + & \underline{r}_u \\ \vdots & \vdots & \vdots \\ \underline{f}^{(s-1)}_{j_{m-1}, j_{2m-1}} & + & \underline{r}_{m-1} \\ \underline{f}^{(s)}_{j_o, j_m} & + & \underline{r}_o \\ \vdots & \vdots & \vdots \\ \underline{f}^{(s)}_{j_{u-1}, j_{m+u-1}} & + & \underline{r}_{u-1} \end{bmatrix} = Y \qquad (5.5)$$

The ℓ-tuples $\underline{r}_i$ account for the additive errors. If the slip s_o is negative, it may be written,

$$s_o = s\,m + u \qquad s \leq 0, \quad 0 \leq u < m \qquad (5.6)$$

Then the array Y is:

$$\begin{bmatrix} \underline{f}^{(s)}_{j_{m-u}, j_{-u}} & + & \underline{r}_{m-u} \\ \vdots & \vdots & \vdots \\ \underline{f}^{(s)}_{j_{m-1}, j_{-1}} & + & \underline{r}_{m-1} \\ \underline{f}^{(s+1)}_{j_o, j_{-m}} & + & \underline{r}_o \\ \vdots & \vdots & \vdots \\ \underline{f}^{(s+1)}_{j_{m-u-1}, j_{-u-1}} & + & \underline{r}_{m-u} \end{bmatrix} = Y \qquad (5.7)$$

The decoder operates on the rows of the array Y in either case. If there is no more than e additive errors occurring in every burst of length ℓ or less, then $w(\underline{r}_i) \leq e$ for $0 \leq i \leq m-1$ in (5.5) or (5.7). Moreover if the value of s in (5.4) or (5.6) is such that $|s| \leq t$, then each row of Y is protected against either type of error because of the capabilities of the code, Λ, in Definition 5.1. Therefore the interleaved code has total protection ability of at most S bits in either direction.

$$S = m(t+1)-1 \tag{5.8}$$

This general result may be summarized in a theorem.

Theorem 5.1

Suppose there is a block code of length ℓ which provides security from e or less additive errors and t or less bits of slip regardless of direction. Then if this code is interleaved to order m, the resulting performance is protection against at most e additive errors and at most S bits of slip independent of direction. S is given in (5.8).

It is possible to couple this general result employing interleaving with any of the codes which have been constructed in the three preceding chapters. First consider all of the results concerned with a symmetric slip range. The conclusion in every theorem and corollary would state that there is a particular type of code which can protect against (detect, detect and classify, or correct) the simultaneous occurrence of at most e errors and S bits of slip. However in the hypothesis of each theorem or corollary t is replaced by $\left(\left\lceil \frac{S+1}{m} \right\rceil -1\right)$. Note that increasing the interleaving order m decreases the value of the symmetric slip range required by the hypothesis while the value of S

in the conclusion remains unchanged. The net effect is to increase the slip range without degrading the error performance. Two examples of coupling the interleaving technique with other approaches will be given as corollaries to the theorem. Their proofs are obvious from Theorem 5.1 once the other result from a previous chapter is identified. The first one is based upon Theorem 2.3.

Corollary 5.1

Any (n,k) cyclic code has a coset code which, when interleaved to order m, can simultaneously correct e or less additive errors and S or less bits of slip if

$$e = \min \left\{ \left[\frac{d - \left[\frac{S+1}{m}\right] - 1}{4} \right], \left[\frac{n}{\left[\frac{S+1}{m}\right]} - 1 \right] \right\} \tag{5.9}$$

Increasing the interleaving order increases the error performance. Since these are derived from coset codes, they will perform as normal (n,k) cyclic codes whenever it can be determined that there is no slip.

Another example is provided by considering Theorem 3.3 which deals with subset codes. The symbols u and f(x) are given in Definition 2.1.

Corollary 5.3

For any (n,k) cyclic code it is possible to interleave to order m an (n,k-a) block code which in the aggregate has the capability of conjointly correcting e or less additive errors and S or less bits of slip if

$$e = \left[\frac{d - \left[\frac{S+1}{m}\right]}{2} \right] \tag{5.10}$$

and

$$\left[\frac{S+1}{m}\right] = u \leq q^a - 1 \tag{5.11}$$

The inequality becomes an equality if and only if f(x) is a primitive polynomial.

The results which deal with the situation of an unsymmetric slip range can also be extended by using interleaving. In any theorem or corollary the conclusion would have t^+ the positive slip range replaced by S^+ and t^- the negative slip range replaced by S^-. But in the hypothesis $t^{\pm}$ is replaced by $\left(\left[\frac{S^{\pm}+1}{m}\right]-1\right)$, respectively. For example a result from Chapter 4 on shortened codes, Corollary 4.5, can be coupled with the interleaving approach.

Corollary 5.3

There is a block code, interleaved to order m, which has the correction capabilities of at most e additive errors and either at most S^+ bits of positive slip or at most S^- bits of negative slip. A sufficient condition for this is the existence of an (n,k) cyclic code which can correct e or less additive errors occurring in those places from the $\left[\frac{S^+ +1}{m}\right]$st to the $\left(n-\left(\left[\frac{S^- +1}{m}\right]-1\right)\right)$st inclusively and also can correct either a burst in the first $\left(\left[\frac{S^+ +1}{m}\right]-1\right)$ positions and a second one between the $\left(n-\left[\frac{S^+ +1}{m}\right]-\left[\frac{S^- +1}{m}\right]-2\right)$nd and the $\left(n-\left[\frac{S^- +1}{m}\right]\right)$th places or a burst between the $\left(\left[\frac{S^+ +1}{m}\right]-1\right)$st place and the $\left(\left[\frac{S^+ +1}{m}\right]+\left[\frac{S^- +1}{m}\right]-1\right)$st place and a second burst in the last $\left(\left[\frac{S^- +1}{m}\right]-1\right)$ positions. Let $t_t = \left(\left[\frac{S^+ +1}{m}\right]+\left[\frac{S^- +1}{m}\right]-2\right)$. Then the length of the block code before the interleaving is $n'' = n-t_t$, and it has $(q-1)\,q^{k-t_t-1}$ members.

Concatenated Codes

The concatenating of codes for error correction was introduced by Forney [35]. One of the advantages of this approach is that the complexity of the encoder and the decoder is reduced. Concatenation and

interleaving will be combined to provide an increase in slip and additive error protection ranges while the complexity of the decoder and encoder is not increased to the extent that it would be if interleaving alone were employed. This combination can produce slip protection in excess of $Q[\frac{\ell}{2}]$ bits where ℓ is the length of a block code and Q is an integer.

The basic idea of concatenation is simple. Information digits from $GF(q^k)$ are encoded and then each element of the code vector is treated as a set of information digits from GF(q) and encoded again. The net result is a long code word with components from GF(q). The decoding is performed in two steps just as the encoding was done except, of course, it is performed in reverse order. The code over $GF(q^k)$ is known as the outer code while that over GF(q) is the inner code. The inner code will be interleaved as developed in the previous section. The outer code will be a coset code of a Reed-Solomon code. The simpler case of symmetric slip will be treated first. The results for unsymmetric slip will be presented at the conclusion of this section.

The general principle of concatenated codes as will be used here is depicted in Figure 5.1. Additive errors and slip are introduced by the inner channel. The outer channel is a convenient dichotomy for describing the concatenation concept. Let $\Gamma = \{\underline{B}_i\}_{i=0}^{M-1}$, $M = q^{Kk}$, be an (N,K) Reed-Solomon code over $GF(q^k)$. Thus $N = q^k-1$ and the minimum distance $D = N-K+1$. Further let $\lambda \in GF(q^k)$ be a primitive Nth root of unity. Now any element $\beta \in GF(q^k)$ may be written as: ([14] or section 18 [29])

$$\beta = x_o + x_1\lambda + x_2\lambda^2 + \ldots + x_{k-1}\lambda^{k-1} \qquad (5.12)$$

Each $x_i \in GF(q)$. So β may be equivalently represented over GF(q) as a k-tuple:

$$\beta = (x_0, x_1, x_2, \ldots, x_{k-1}) \tag{5.13}$$

Let $\underline{C}$ be the coset generator given by (2.66). From Chapter 2 it is known that the coset code, $\{\underline{B}_i + \underline{C}\}_{i=0}^{M-1}$ is capable of protecting against (detecting, detecting and classifying, and correcting) the simultaneous occurrence of at most E additive errors and at most T bits of slippage. Each N-tuple of this coset code is comprised of elements from $GF(q^k)$ which may be represented as a k-tuple over GF(q).

From Theorem 5.1 it is possible to obtain a special (n,ak) code by interleaving which has the capability of simultaneously protecting against e or less additive errors and S or less bits of slip where $S \geq [\frac{n}{2}]$. Now encode a of the components of an N-tuple into an n-tuple over GF(q).* The total code length through the inner channel is nN. At the special decoder the symbols in the N-tuples are secure if e or less additive errors have occurred in each n-tuple through the inner channel. They could be misframed though because of the ambiguity associated with slips which are integer multiples of $[\frac{n}{2}]$. But the outer code can protect against T or less bits of slip if the inner decoder has not made more than $[\frac{E}{a}]$ mistakes with the inner code. Therefore the overall system is secure from U or less bits of slip in the inner channel if the additive errors are such that more than e occur in a framed n-tuple in the inner channel at most $[\frac{E}{a}]$ times. The

*The integer a is the interlacing order for the concatenated codes. When a = 1, this is conventional concatenated coding.

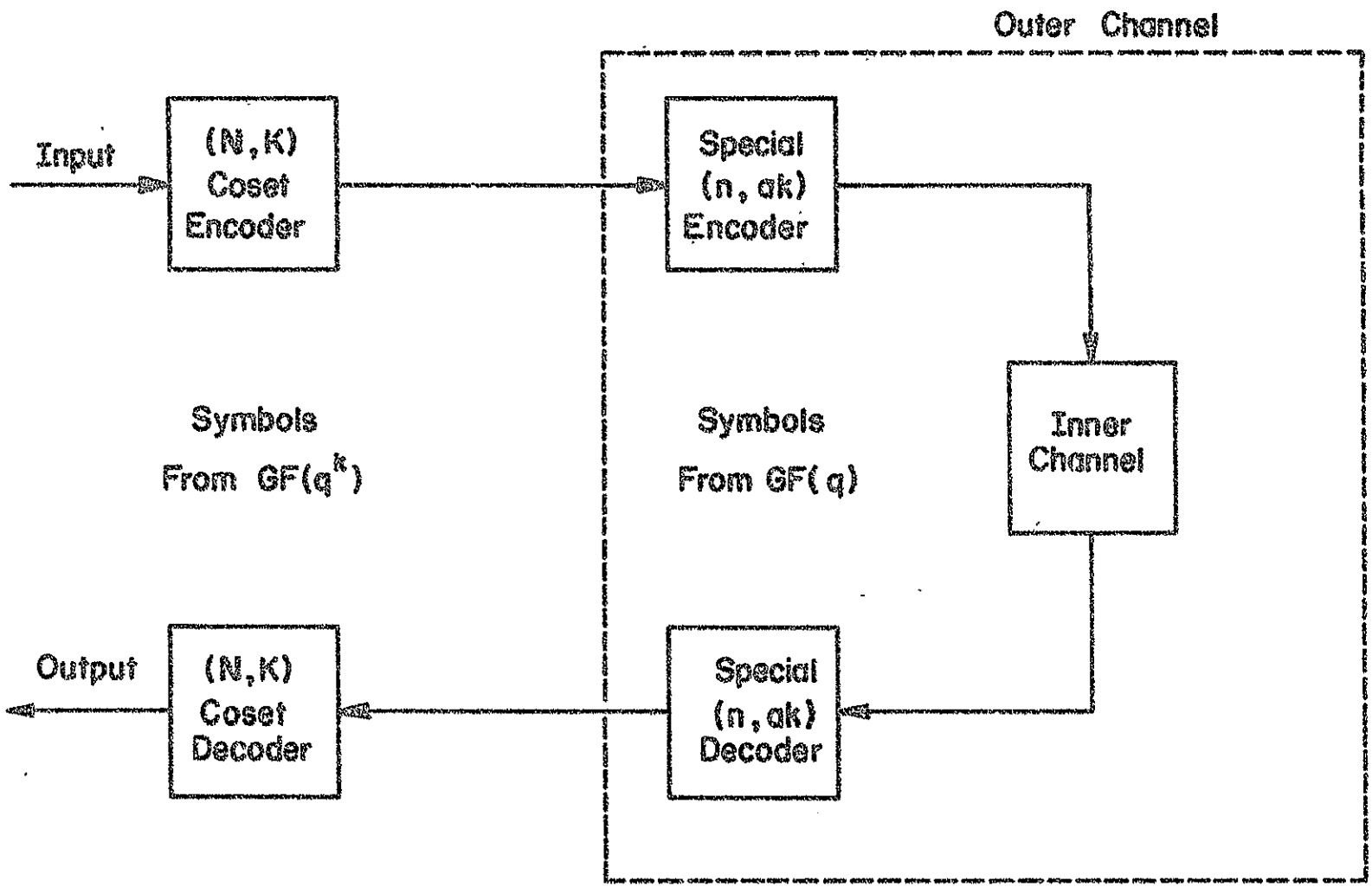


FIGURE 5.1. ILLUSTRATION OF CONCATENATION.

quantity U is given by:

$$U = S + n[\frac{T}{a}] \tag{5.14}$$

There are numerous combinations of types of protection which are possible by using concatenated codes. The code for the inner channel is a result of Theorem 5.1, and any code in the previous chapters can be used with that theorem. The outer code is a coset code of a Reed-Solomon code to which a section of Chapter 2 is devoted. The results in that section concern detection, detection and classification, and correction.

The extension of the results of this section to the case of an unsymmetrical slip range is presented. The overall positive slip protection range is denoted as U^+ while U^- represents the negative one. Recall that S^+ and S^- are the respective slip ranges of the inner code as discussed in the previous section and that T^+ and T^- are the ones for the outer code as given above Corollary 2.5. Thus the following relationships are true.

$$U^+ = \begin{cases} S^+ & \text{if } S^+ < [\frac{n}{2}] \\ S^+ + n[\frac{T^+}{a}] & \text{if } S^+ \geq [\frac{n}{2}] \end{cases}$$

$$U^- = \begin{cases} S^- & \text{if } S^- < [\frac{n}{2}] \\ S^- + n[\frac{T^-}{a}] & \text{if } S^- \geq [\frac{n}{2}] \end{cases} \tag{5.15}$$

Comparisons and Examples

One combination of the possible types of protection available from concatenated codes will be chosen as a basis for comparison with the

other approaches. The outer and the inner code will both be coset codes capable of simultaneous correction of both additive errors and slip in a symmetric range. So if synchronous operation is quaranteed, the overall code may operate with its full error-correcting capabilities. Furthermore the decision as to the mode of the code's operation is made at the decoder. It is for these reasons that coset codes will be considered in this section. Specifically the inner code is an interleaved code derived according to Theorem 2.4, and the outer code conforms to the construction given in Theorem 2.5.

Since concatenated codes may be viewed as long block codes over the inner channel, the question arises as to whether it might be possible to achieve better error and slip performance by considering longer codes in the first place. There are three other approaches with which one could construct these longer self-synchronizing codes. Each represents the most powerful known technique of its type. They are coset codes (Theorem 2.3), subset codes (Theorem 3.3) and extended subset codes (Theorem 4.5). It will be shown that concatenated codes are superior to each of these when considered as wide-range self-synchronizing codes. Consequently it will be seen that interleaved codes are also superior.

In order to provide a basis for comparison of these approaches it will be assumed that the lengths and rates of each, considered as a code over GF(q), are equal and that the slip correction ranges, again over GF(q), are also equal. The performance criterion for comparison will be the error-correcting capabilities as the slip correction range, U, increases. For the concatenated approach the maximum number of

correctable additive errors over GF(q) (in the inner channel) is underbound by max $\{e,[\frac{E}{a}]\}$. It follows from the two theorems (Theorems 2.4 and 2.5) on which these codes were based that this quantity is:

$$\max\{e,[\tfrac{E}{a}]\} = \max\left\{\min\left(\left[\tfrac{d-2t-3}{4}\right],\left[\tfrac{n-2t-2}{2t+1}\right]\right),\left[\frac{\min\left(\left[\frac{N-2K+1}{2}\right],\left[\frac{N-K-2T-1}{2}\right]\right)}{a}\right]\right\} \tag{5.16}$$

The dependence between U, t and T is obtained from (5.8) and (5.14) and is given by:

$$U = m(t+1) - 1 + n[\tfrac{T}{a}] \tag{5.17}$$

Note that there is an extra degree of freedom in the choice of U in the form of the interleaving order, m. Thus it is possible to increase U without altering the lower bound given in (5.16).

To use the coset approach on a code of equal length and rate requires the existence of a cyclic (nN,akK) code. Just for the sake of argument, the required existence will be assumed. The minimum distance of this cyclic code is at most (nN-akK+1) [26]. So from Theorem 2.3 its error-correcting capability is at most the following expression:

$$\min\left\{\left[\frac{(nN-ak+1)-U-2}{4}\right] , \left[\frac{nN-U-1}{U+1}\right]\right\} \tag{5.18}$$

This is strictly monotone decreasing in U, and thus the concatenated approach has superior error performance as U increases and all other factors remain identical.

For the subset codes of Theorem 3.3 to have the same rate and length as the concatenated codes, there must be a cyclic code over GF(q) of length nN and information content of at least $kK+\log_q(U+2)$. Assume such a cyclic code exists. Using a bound due to Singleton [26], the minimum distance of this code is at most $\{nN-(kK+\log_q(U+2))+1\}$. So from (3.14) the number of correctable additive errors is at most:

$$\frac{nN-(kK+\log_q(U+2))+1-U-1}{2} \tag{5.19}$$

Since this overbound of the true error performance of subset codes is strictly decreasing in the correctable slip U, the superiority of concatenated codes is established.

The cyclic extension of subset codes is a technique which has unlimited slip correction capabilities. Again assume that the proper cyclic code exists. Referring to Theorem 4.5 it is seen that its length must be (nN-2U) while the information content must be at least $kK+\log_q(U-nN[\frac{U}{nN}]+2)$. Employing the same bound on the minimum distance as above, the error performance given by (4.26) is at most:

$$\frac{(nN-2U-(kK+\log_q(U-nN[\frac{U}{nN}]+2))+1-1}{2} \tag{5.20}$$

Again as U increases (5.20) always decreases and is exceeded by (5.16); so the concatenated approach to wide-range self-synchronizing codes has better performance than its most powerful competitors.

The versitility and capabilities of this approach will be demonstrated by several examples which are presented in Table 5.1. They result from concatenating coset codes derived from Reed-Solomon codes

over $GF(2^k)$ with interleaved coset codes derived from binary cyclic codes. In all cases both the inner and the outer code is designed for the simultaneous correction of both types of errors. Some of the examples used in Table 2.1 and Table 2.2 are concatenated to produce the examples given in Table 5.1. Hence there are three possible choices of inner code additive error performance and three for the outer code for the same value of overall slip correction range. Since the information content, K, of the outer code does not divide its length, N, in these examples, then the outer channel's error performance using the technique of Theorem 2.8 is fixed for any slip range in the outer channel of less than $[\frac{N}{2}]$.

Table 5.1. Performance Capabilities of Several Concatenated Codes

Parameters of Outer Code over $GF(2^k)$		Parameters of Inner Code over GF(2)	Interleaving Order for Inner Code	Maximum Number of Additive Errors Correctable by the Inner Decoder Using the Technique of			Maximum Number of Additive Errors Correctable by the Outer Decoder Using the Technique of			Overall Slip Correction Range
k	(N,K)	(n,ak)	m	Thm. 2.3	Thm. 2.4	Cor. 2.1	Thm. 2.8	Thm. 2.5	Cor. 2.3	U
3	(7,2)	(31,6)	8	3	2	1	1	*	*	46
			2	1	*	*	1	*	*	50
		(63,18)	16	4	4	4	1	*	*	31
			4	3	1	*	1	*	*	31
		(127,15)	32	13	12	11	1	*	*	63
			8	11	6	2	1	*	*	71
			4	5	2	*	1	*	*	83
			2	1	*	*	1	*	*	99
4	(15,4)	(63,36)	16	2	1	0	3	4	3	31
			6	1	*	*	3	4	3	35
		(63,24)	16	3	2	1	3	4	3	31
			7	2	1	*	3	*	*	97
		(127,8)	8	13	6	2	3	3	2	198
			8	13	6	2	3	1	*	325
			8	13	6	2	3	*	*	452
4	(15,2)	(63,36)	16	2	1	0	5	5	4	31
			6	1	*	*	5	5	4	35
		(127,8)	11	14	10	7	5	5	4	65
			11	14	10	7	5	4	3	192
			5	8	3	*	5	*	*	450
5	(31,10)	(15,5)	8	1	0	*	5	6	5	45
			8	1	0	*	5	4	1	105
			8	1	0	*	5	*	*	240
		(45,5)	12	4	4	3	5	6	5	113
			3	3	1	*	5	4	1	293
			3	3	1	*	5	*	*	698
			2	1	*	*	5	*	*	706
5	(31,10)	(63,10)	16	6	5	4	5	6	5	94
			4	4	2	*	5	4	1	224
			4	4	2	*	5	*	*	476
			2	1	*	*	5	*	*	484
		(127,50)	32	6	5	4	5	5	2	63
			16	5	4	2	5	*	*	190
		(127,15)	32	13	12	11	5	4	1	317
			11	12	10	7	5	4	1	319

Table 5.1. (Continued)

k (N,K)	(n,ak)	m	Thm. 2.3	Thm. 2.4	Cor. 2.1	Thm. 2.8	Thm. 2.5	Cor. 2.3	U
		11	12	10	7	5	*	*	700
		4	5	2	*	5	*	*	718
5 (31,7)	(15,5)	8	1	0	*	8	9	8	45
		8	1	0	*	8	*	*	240
	(45,5)	8	4	3	2	5	5	2	246
		3	3	1	*	5	5	2	246
		8	4	3	2	5	*	*	698
		3	3	1	*	5	*	*	698
	(63,30)	16	2	2	1	5	4	1	94
		8	2	1	*	5	4	1	94
		8	2	1	*	5	*	*	157
	(63,10)	8	5	4	2	5	6	4	157
		8	5	4	2	5	4	1	220
		4	4	2	*	5	4	1	224
		2	1	*	*	5	*	*	484
	(127,50)	16	5	4	2	8	9	8	63
		16	5	4	2	8	*	*	190
		6	3	1	*	8	*	*	192
	(127,15)	11	12	10	7	8	5	2	319
		11	12	10	7	8	*	*	700
		8	11	6	2	8	5	2	325
		4	5	2	*	8	*	*	718
5 (31,3)	(63,30)	16	2	2	1	12	6	3	94
		8	2	1	*	12	*	*	157
	(63,10)	16	6	5	4	12	6	3	220
		16	6	5	4	12	3	*	346
		4	4	2	*	12	3	*	350
		4	4	2	*	12	*	*	476
		2	1	*	*	12	*	*	484
	(127,50)	6	3	1	*	12	3	*	192
		3	1	*	*	12	*	*	192
	(127,15)	11	12	10	7	12	10	9	192
		5	7	3	*	12	10	9	201
		5	7	3	*	12	3	*	455
		4	5	2	*	12	3	*	464
		4	5	2	*	12	*	*	718
6 (63,31)	(31,6)	8	3	2	2	0	1	*	449
		4	2	1	1	0	1	*	449
		2	1	*	*	0	1	*	453
	(63,36)	11	1	1	*	0	1	*	158
		6	1	*	*	0	1	*	161
	(127,36)	11	6	4	1	0	1	*	319
		4	3	*	*	0	1	*	325

Table 5.1. (Continued)

k	(N,K)	(n,ak)	m	Thm. 2.3	Thm. 2.4	Cor. 2.1	Thm. 2.8	Thm. 2.5	Cor. 2.3	U
6	(63,16)	(31,6)	8	3	2	1	15	15	11	263
			4	2	1	*	15	11	5	387
			2	1	*	*	12	*	*	484
		(63,24)	16	3	2	1	15	11	5	220
			7	2	1	*	15	7	*	286
			4	1	*	*	15	*	*	480
		(127,36)	11	6	4	1	15	11	5	319
			4	3	*	*	15	*	*	706
6	(63,8)	(31,6)	8	3	2	1	23	17	12	325
			4	2	1	*	23	7	*	635
			2	1	*	*	23	*	*	980
		(63,18)	16	4	4	4	23	12	4	346
			4	3	1	*	23	12	4	346
			2	1	*	*	23	*	*	661
		(127,78)	32	3	2	1	23	12	4	190
			16	2	1	*	23	*	*	317
		(127,36)	11	6	4	1	23	12	4	319
			4	3	*	*	23	*	*	706
7	(127,31)	(63,7)	16	7	6	5	32	33	32	157
			6	6	4	1	32	27	17	1295
			3	4	1	*	32	7	*	2555
			2	2	*	*	32	*	*	4010
8	(255,35)	(31,16)	8	1	0	*	92	93	92	46
			8	1	0	*	92	9	*	1565
		(63,24)	16	3	2	1	92	79	64	661
			7	2	1	*	92	49	19	1294
			4	1	*	*	92	*	*	2685
		(127,64)	16	4	3	1	92	93	88	190
			11	3	2	*	92	59	34	827
			6	2	*	*	92	29	*	1341
			4	1	*	*	92	9	*	1587
		(127,8)	11	14	10	7	92	93	88	1335
			8	13	6	2	92	79	64	3881
			5	8	3	*	92	49	19	7689
			3	4	1	*	92	29	*	10234
			2	1	*	*	92	*	*	16244

CHAPTER 6

SUMMARY

The results given in this work have been presented in a very general context because no particular system's model has been assumed. The codes which have been developed have the capability of protecting against the simultaneous occurrence of additive errors and of the loss of positions from true synchronization in a given direction (bit slippage). The results are given as the maximum number of each which may be protected. This work has dealt exclusively with the modification of cyclic codes with characters from a general finite field, GF(q), This type of code has been used because of its added algebraic structure and easy implementation.

There are a number of ways in which a given error-protecting code may be modified so as to endow it with sync-protecting capabilities. However each method extracts a price in the form of a degradation in certain aspects of the original code's performance. The various methods are classified according to the parameters of the code that are altered, and the results here are presented along this type of outline. The advantages of one method in one set of circumstances may be disadvantages in another set. Therefore a complete and comprehensive coverage of all methods is given. Results concerning the detection of additive errors and slippage, the detection and the classification of the nature of the error, and the correction of both types of errors are exhibited for each

modification approach. The situations of symmetrical and unsymmetrical slip ranges are also considered.

The design and construction of these modified codes is performed from the viewpoint of minimum distance decoding. Therefore the proofs of all the results are not simply existence proofs but offer the general strategy for decoding such codes.

One strategy which is used to obtain new and superior correction results is an iterative one. If a received vector is within a prescribed sphere around any modified code member, that member is the optimum choice for the received one in the sense of a minimum distance criterion. However if the received vector is within a concentric shell about the prescribed sphere, then the decoder must reframe and check to determine if this yields a vector within some other sphere. This strategy is analogous to the correlation of synchronization sequences except in the case above a decrease in the distance is sought instead of an increase in the correlation value.

Joint and triple estimation schemes are also employed to obtain new results. The joint estimator is a less complex version of the triple one, and so the results in the joint case are not as powerful.

The technique of employing a coset of the original code allocates part of the error-protecting power of a code to synchronization protection. The construction of coset codes involves the proper choice of a coset generator - the fixed vector which is added to all code members. The length and rate of the original code are not changed. This approach has a very important advantage. Whenever synchronous operation is maintained, the code may operate with its full error-protecting capabilities.

The decision as to the mode of the code's operation is made at the decoder which is an appealing prospect for one-way communication systems. New results for all types of protection are given for the coset codes of general cyclic codes. Even though Reed-Solomon codes fit this category, stronger results than those which could be obtained above are presented for this special case. There are three theorems dealing with Reed-Solomon codes which permit protection from slips of half the code's length in either direction. One theorem deals with detection of both types of errors, one with detection and classification, and one with correction.

Subset codes are derived from cyclic codes by removing certain vectors before any other modification is applied. The purpose of these deletions is to eliminate some of the vectors which are cyclic shifts of a subset of the original code. Since synchronization loss appears as a shift or slip, the effect of this modification is to produce a subset code which is less sensitive to slippage. The rate of the subset codes is less than that of the original code. However the protection ranges for both additive errors and slip are much better than that which is possible by using coset codes.

Two approaches to subset codes are demonstrated. The first combines expurgating or removing members with the use of a coset generator. The second imbeds a fixed pattern in the information digits of the code. In either case the initial step in the decoding strategy is the same. It treats the received vectors as if only additive errors have perturbed them. The remaining steps in the strategy separate the type of error if more than detection of some kind of error is being considered. The work in this chapter represents the most comprehensive treatment of subset

codes known.

The concept of lengthening a sequence of information digits by inserting check digits is the basis of additive error-protecting codes. It has a counterpart when protection from loss of synchronization is desired. Extending the length of a cyclic code always allows the decoder to frame a portion of only one word. Another approach is to shorten the code at the encoder. Since the added length is appended at the decoder, a portion of an adjacent word is located in the body of the code vector and not at either end.

None of the methods employing length alterations use it exclusively. It is always used in conjunction with some other modification, e.g., lengthening a coset code. The additive error and slip protection performance of length altered codes is better than that of coset codes, but in general neither these codes nor subcodes exhibits a universal superiority over the other. Length altering a code diminishes its rate. When these codes are operating synchronously, the additive error performance is lower than that of the parent codes from which they were derived. The decoding strategies for length altered codes have a common feature. The original length of the code is recovered at the decoder by adding or deleting digits depending upon the nature of the length alteration. The remaining steps are based upon the structure of the parent code.

There have been very few results previously presented on length altered codes. The work here shows improvements on these scattered results and introduces new results so as to complete and consolidate all aspects of this area. Results concerning shortened codes are developed,

subset codes are lengthened to give another approach, and coset codes are extended to produce a new modification scheme. This last approach offers a compromise between the reduction of rate which is inherent in extended subset codes and the reduction in additive error and slip protection capabilities in coset codes. There is a moderation in the loss of each of these performance criteria.

The extended subset codes have capabilities of wide-range slip protection. Two other approaches for achieving this are presented. One uses an interleaving technique while the other combines concatenation with interleaving. With either construction, slip protection ranges of up to half of the code's length are possible. The interleaving approach as introduced here is a method which may be coupled with any other technique contained in this work for protection from additive errors and slippage. The net effect is to greatly expand the slip protection range capabilities of the other technique. Interleaving allows smaller values of slippage to be spread over several code words rather than the total amount effecting each and every word.

Concatenation and interleaving are combined to provide an increase in the slip protection range. This is accomplished without increasing the complexity of the encoder and decoder to the extent to which they would be if interleaving alone were used. It is shown that for wide range slip protection the error performance of either construction is superior to any other know approach.

REFERENCES

[1] A.A. Albert, Fundamental Concepts of Higher Algebra, Chicago: University of Chicago Press, 1956.

[2] W.W. Peterson, Error-Correcting Codes, Cambridge, Mass.: MIT Press, 1961.

[3] S.W. Golomb, J.R. Davey, I.S. Reed, H.L. VanTrees, and J.J. Stiffler "Synchronization," IEEE Transactions on Communications Systems, Vol. CS-11, pp. 481-491, 1963.

[4] L. Calabi and W.E. Hartnett, "A Family of Codes for the Correction of Substitution and Synchronization Errors," IEEE Transactions on Information Theory, Vol. IT-15, pp. 102-106, 1969.

[5] F.F. Sellers, Jr., "Bit Loss and Gain Correction Code," IRE Transactions on Information Theory, Vol. IT-8, pp. 35-38, 1962.

[6] J.D. Ullman, "Near-Optimal, Single-Synchronization-Error-Correcting Code," IEEE Transactions on Information Theory, Vol. IT-12, pp. 418-424, 1966.

[7] J.D. Ullman, "On the Capabilities of Codes to Correct Synchronization Errors," IEEE Transactions on Information Theory, Vol. IT-13, pp. 95-105, 1967.

[8] W.L. Eastman, "On the Construction of Comma-Free Codes," IEEE Transactions on Information Theory, Vol. IT-11, pp. 263-267, 1965.

[9] S.W. Golomb and B. Gordon, "Codes with Bounded Synchronization Delay," Information and Control, Vol. 8, pp. 355-372, 1965.

[10] W.L. Eastman and S. Even, "On Synchronizable and PSK-Synchronizable Block Codes," IEEE Transactions on Information Theory, Vol. IT-10, pp. 351-356, 1964.

[11] B.H. Jiggs, "Recent Results in Comma-Free Codes," Canadian Journal of Mathematics, Vol. 15, pp. 178-187, 1963.

[12] S.W. Golomb, B. Gordon, and L.R. Welch, "Comma-Free Codes," Canadian Journal of Mathematics, Fol. 10, pp. 202-209, 1958.

[13] J.J. Stiffler, Synchronization in Communications Systems, Englewood Cliffs, New Jersey: Prentice-Hall, Inc., Chapter 14, 1969.

[14] B.L. van der Waerden, Modern Algebra, New York: Fredrick Ungar Publishing Col, Vol. 1, Section 37, 1931.

[15] E.R. Berlekamp, Algebraic Coding Theory, New York: McGraw-Hill Book Company, Inc., 1968.

[16] R.B. Ash, Information Theory, New York: Interscience Publishers, pp. 87-169, 1965.

[17] R.W. Lucky, J. Salz, and E.J. Weldon, Jr., Principles of Data Communications, New York: McGraw-Hill Book Company, Inc., pp. 277-374, 1968.

[18] J.J. Stiffler, "Comma-Free Error-Correcting Codes," IEEE Transactions on Information Theory, Vol. IT-11, pp. 107-112, 1965.

[19] S.Y. Tong, "Synchronization Recovery Techniques for Binary Cyclic Codes," The Bell System Technical Journal, Vol. 45, pp. 561-595, 1966.

[20] J.E. Levy, "Self-Synchronizing Codes Derived form Binary Cyclic Codes," IEEE Transactions on Information Theory, Vol. IT-12, pp. 286-290, 1966.

[21] G. Solomon, "Self-Synchronizing Reed-Solomon Codes," IEEE Transactions on Information Theory, Vol. IT-14, pp. 608-609, 1968.

[22] S.E. Tavares, "A Study of Synchronization Techniques for Binary Cyclic Codes," Doctoral Dissertation, Department of Electrical Engineering, McGill University, Montreal, Canada, 1968.

[23] S.E. Tavares and M. Fukada, "Matrix Approach to Synchronization Recovery for Binary Cyclic Codes," IEEE Transactions on Informa- Theory, Vol. IT-15, pp. 93-102, 1969.

[24] S.E. Tavares and M. Fukada, "Further Results on the Synchronization of Binary Cyclic Codes," Department of Electrical Engineering Report, McGill University, Montreal, Canada, 1968.

[25] W. Miller, R. Muller, T. Taylor, and T. Yagelowich, "Self-Synchronizing Bi-Orthogonal Coded PCM Telemetry System," National Aeronautics and Space Administration, Technical Report R-292, 1968.

[26] R.C. Singleton, "Maximum Distance Q-Nary Codes," IEEE Transactions on Information Theory, Vol. IT-10, pp. 116-118, 1964.

[27] D. Mandelbaum, "A Note on Synchronization Error-Correcting Codes," Information and Control, Vol. 13, pp. 429-432, 1968.

[28] S.G.S. Shiva and G. Sequin, "On the Correction of Synchronization and Additive Errors," Department of Electrical Engineering Report, University of Ottawa, Ottawa, Canada, 1968.

[29] L.E. Dickson, Linear Groups with an Exposition of the Galois Field Theory, New York: Dover Publications, Inc., 1958.

[30] T. Kasami and N. Tokura, "Some Remarks on BCH Bounds and Minimum Weight of Binary Primitive BCH Codes," IEEE Transactions on Information Theory, Vol. IT-15, pp. 408-414, 1969.

[31] J.G. Caldwell, "Synchronizable Error-Correcting Codes," Doctoral Dissertation, Department of Statistics, University of North Carolina, Chapel Hill, North Carolina, 1966.

[32] R.C. Bose and J.G. Gladwell, "Synchronizable Error-Correcting Codes," Information and Control, Vol. 10, pp. 616-630, 1967.

[33] S.Y. Tong, "Correction of Synchronization Errors with Burst Error-Correcting Cyclic Codes," IEEE Transactions on Information Theory, Vol. IT-15, pp. 106-109, 1969.

[34] E.J. Weldon, Jr., "A Note on Synchronization Recovery with Extended Cyclic Codes," Information and Control, Vol. 13, pp. 354-356, 1968.

[35] G.D. Forney, Jr., "Concatenated Codes," M.I.T. Research Laboratory of Electronics Report, TR 440, 1965.

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)

School of Electrical Engineering, Purdue University
Lafayette, Indiana 47907

2a. REPORT SECURITY CLASSIFICATION

Unclassified

2b. GROUP

3. REPORT TITLE

CODES FOR PROTECTION FROM SYNCHRONIZATION LOSS AND ADDITIVE ERRORS

4. DESCRIPTIVE NOTES (Type of report and, inclusive dates)

Scientific

5. AUTHOR(S) (First name, middle initial, last name)

G. R. Redinbo and P. A. Wintz

6. REPORT DATE

November, 1969

7a. TOTAL NO. OF PAGES

114 + 6 + 2

7b. NO. OF REFS

35

8a. CONTRACT OR GRANT NO.

No. N00014-67-A-0226 modification AE

8b. PROJECT NO.

9a. ORIGINATOR'S REPORT NUMBER(S)

TR-EE 69-43

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

Unlimited

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY

ARPA and Joint Services Electronics Program

13. ABSTRACT

Cyclic codes are practical and efficient codes which protect against the effects of additive errors. However their effectiveness, like that of block codes, requires correct word synchronization at the decoder. Cyclic codes with symbols from a general finite field are modified so that they are also capable of protecting against misframing at the decoder. These codes are modified by altering their distance structure. There are a number of techniques which can be employed. Each method affects different aspects of the code's performance; therefore a complete and comprehensive coverage of all techniques is given.

Results for each modification approach are given for three types of protection from the simultaneous occurrence of additive errors and synchronization errors. The first type is the detection of some kind of error, the second is the detection and classification of the nature of the error, and the third is the correction of both kinds of errors. Furthermore for each approach results are presented for the cases of symmetrical and unsymmetrical ranges of synchronization errors. The proofs of all results indicate the general strategy for decoding the modified code.

A coset of the original code allocates part of its error-protecting capabilities to synchronization. Results are given for the general class of cyclic codes. Stronger conclusions are possible when the special case of Reed-Solomon codes is considered. In this case protection from slips of half the code's length in either direction are permitted.

A subset code is derived from a code by removing certain of its vectors so as to

produce a code with fewer members which are less sensitive to misframing. Two approaches to subset codes are demonstrated. One is a coset code of an expurgated code while the other is a code with a fixed pattern imbedded in the information digits.

Changing the length of a code when combined with other techniques is anothe modification approach. The work here improves on the few known results and i introduces many new ones so as to complete and consolidate all aspects of this type of approach. Results concerning shortened codes are developed, subset codes are extended to yield another modification approach, and coset codes are lengthened to produce a new scheme.

Two approaches for achieving wide-range slip protection are presented. One uses interleaving while the other combines interleaving with concatenation. With either technique slip protection ranges of half the code's length are possible. The interleaving technique may be coupled with any other approach giving the net effect of greatly expanding the slip protection range of that approach. Combining concatenation and interleaving accomplishes the same result without increasing the complexity of the encoder and decoder to the extent to which they would be if only interleaving were used. It is shown that for wide-range slip protection the error-protecting performance of either approach is superior to any other known approach.