STATE UNIVERSITY OF NEW YORK
AT STONY BROOK

REPORT NO 71

AFCRL - 66-655 NOTICE NO. 1

THE FRAMING PROBLEM IN BLOCK CODED FEEDBACK COMMUNICATIONS SYSTEMS

by Peter M. Dollard
and Jerome T. Waters

College of Engineering
State University of New York
Stony Brook, New York 11790

Technical Report No. 71

Scientific Report No. 2 on

Contract No. AF 19 (628) -3865

Project No. 5628

Task No. 562801

August 24, 1966

Prepared
for

AIR FORCE CAMBRIDGE RESEARCH LABORATORIES
OFFICE OF AEROSPACE RESEARCH
UNITED STATES AIR FORCE

Bedford, Massachusetts

3176-2-17

AFCRL - 66-655 NOTICE NO. 1

THE FRAMING PROBLEM IN BLOCK CODED FEEDBACK COMMUNICATIONS SYSTEMS

by Peter M. Dollard
and Jerome T. Waters

College of Engineering
State University of New York
Stony Brook, New York 11790

Technical Report No. 71

Scientific Report No. 2 on

Contract No. AF 19 (628) -3865

Project No. 5628

Task No. 562801

August 24, 1966

Prepared
for

AIR FORCE CAMBRIDGE RESEARCH LABORATORIES
OFFICE OF AEROSPACE RESEARCH
UNITED STATES AIR FORCE

Bedford, Massachusetts

Abstract

The literature on word synchronization and framing of block coded binary transmission systems is reviewed in depth, with specific references to an extensive bibliography. Particular attention is paid to the use of comma-free codes derived from linear codes without added redundancy in such a way that the error control properties of the codes are fully retained. The review is directed towards the use of these techniques in a two-way system with feedback, and includes a discussion of the special considerations which arise in this application.

It is shown that all these techniques involve a substantial loss of ultimate information rate, and/or a substantial increase in system complexity over that required for error control alone. A modification of the comma-free codes is suggested in which iterated codes are employed to provide a sync grating intermediate between digit and word sync. This is shown to avoid much of the added complexity of comma-free codes, but at an additional cost in information rate. The report concludes with a brief discussion of still another technique which as yet has received too little attention. With this approach one does not seek absolute sync protection in the absence of channel errors, rather, one seeks highly reliable sync protection with the statistics of both source and channel taken into account.

I. INTRODUCTION AND GENERAL SYSTEMS REMARKS

A mathematical model of a decision-feedback system in which both channels are subject to deep fades or bursts of errors is indicated in Figure 1. The system is protected by the use of a fixed length, binary, linear, error detecting code in conjunction with a word retransmission policy. The influence of the channel disturbances on the appropriate bit synchronization and symbol decoding procedures is reflected in the statistics of the channel model, given in Figure 2. For convenience we assume that the forward and feedback channel statistics are identical and that any interchannel correlation of fading is negligible. The feedback accept/reject reply ($A \leftrightarrow 1, R \leftrightarrow 0$) is encoded as a single information bit in a specified (near future) code word of forward transmission on the alternate channel. Appropriate measures are taken to avoid the $R \rightarrow A$ and to recognize the $A \rightarrow R$ feedback errors, which could otherwise lead to considerable performance deterioration [23,24].

The most obvious additional requirement of the given system is that in order to extract the information bits contained in any regenerated binary sequence, the receivers must be able to distinguish successive code groups. This paper addresses itself to the word synchronization problem. In general what is required is that some single position in a received sequence be identified with respect to the word structure of the code. The remaining digits in that and in successive words may then be identified by simple counting. Due to the nature of the receiver--it attempts to generate binary digits at all times, even when no transmission is taking place--the first transmitted digit cannot be used for this purpose. Some special synchronizing

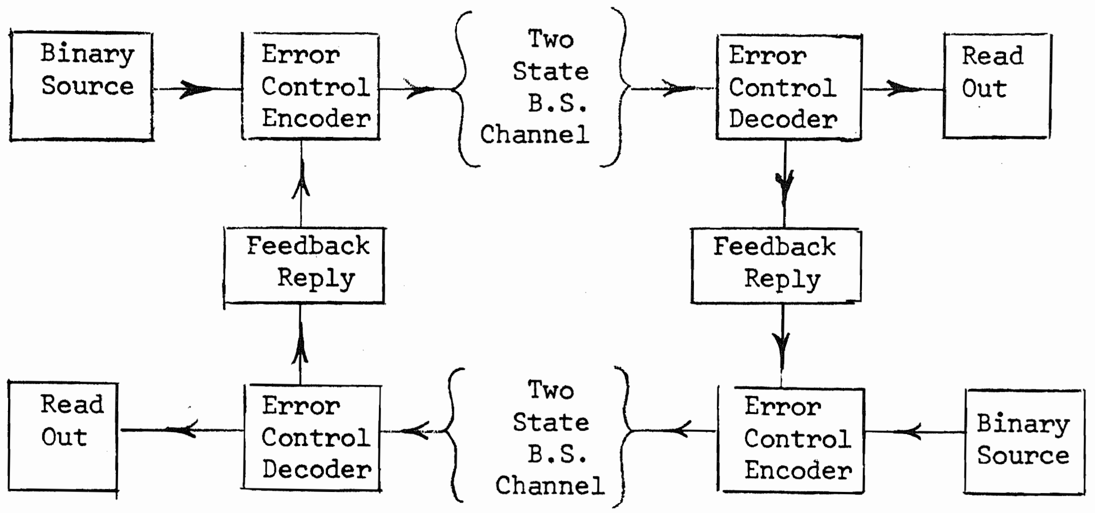
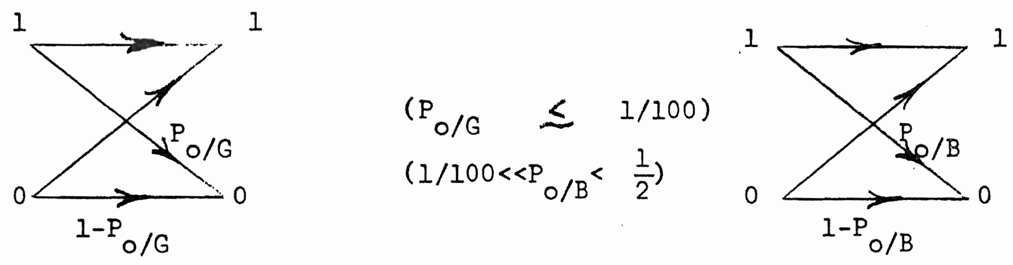


FIGURE 1

GENERAL FEEDBACK SYSTEM


1
1
0
0
$P_{o/G}$
$1-P_{o/G}$
$(P_{o/G} \lesssim 1/100)$
$(1/100 << P_{o/B} < \frac{1}{2})$
1
1
0
0
$P_{o/B}$
$1-P_{o/B}$

a) Good state; occurs with probability P_G

b) Bad State; occurs with probability $P_B = 1 - P_G$

Transition Characteristics:
1) Unknown transition times
2) Rapid state transitions, with channel remaining in each state for a relatively long period.

FIGURE 2

TWO STATE BINARY SYMMETRIC CHANNEL

When the channel is in either (a) the Good or (b) the Bad State, the transmitted digits are independently subject to the given digit error probabilities.

signal, one that is unambiguously recognizable at the receiver, must be sent initially in order to fix the "counting origin". This initial signal will suffice only if the subsequent counting operation is accurate for the duration of the transmission. For high reliability it is generally necessary to periodically supply the receiver with word synchronization information [1]. Indeed, in many applications, some synchronization information is sent along with each word group that is transmitted [23]. In the system described, a fade of reasonable duration generally has a detrimental effect on the bit synchronizing procedure, the result being that bit gains or losses are often sustained during a blackout. Since these occur at unpredictable times it becomes desirable that synchronization information be made available to the receiver on a more continuous basis. In general, a substantial amount of word synchronization information is included with the data being transmitted; the precise nature and amount being determined as a compromise between the "cost" of transmitting and extracting such information and the desired reliability of operation.

There are several essential differences between feedback and feedforward-only systems that bear on this discussion. The availability of a feedback link represents an improvement over feedforward-only systems in terms of synchronization. When, for example, the receiver has an indication that he is operating asynchronously, he may request explicit synchronization confirmation or re-synchronization information. Thus there is a possibility of decreasing the amount of built-in protection against asynchronous operation that is ordinarily required[1]. However, feedback also presents additional problems. For example, the one-to-one correspondence that exists between an accepted (or rejected) word and its associated feedback reply must be maintained at all times throughout the system. In the system considered, the receiver must reply to each (presumably unacceptable) word transmitted during the bad state in order that it be repeated. The loss or gain of a word due to the loss of correct synchronization will result in the transmitter applying feedback replies incorrectly. Such errors have a pronounced effect on system reliability and, if necessary, protection must be provided. This is effectively a problem of maintaining gross synchronization during a fade--i.e. keeping within the "recovery margin" of the synchronization scheme used--and occurs regardless of the word synchronization procedure invoked[2]. For most of the conceivable word sync procedures, synchronization must always be within one-half a code word in either direction of the correct position in order that, upon regaining sync, no loss or gain of words occurs. If necessary, an artificial procedure must be invoked to provide this gross synchronization during the bad state. The general nature of such a scheme might be as follows: there is some limit to the number of consecutive word retransmissions that can be requested before a code word is "likely" to be lost or gained. This limit must be estimated and when reached, a special sync pattern (designed to resynchronize the system under the appropriate conditions) automatically requested.[3] Finally, notice that in a feedforward

1. More sophisticated utilization of the potential of feedback with respect to word synchronization has generally heretofore received little attention in the literature.
2. Generally the sync procedure is designed on the basis of the good state; sync during the bad state is maintained by extrapolation of some previous sync position using a clock.
3. Procedures have been developed which correct certain bit losses or gains by the periodic addition of special symbols to the data stream [27]. Some of these results, if extended, may be applicable to the above situation.

only system, one measure of the worth of any synchronization scheme (both word and bit) is the rapidity with which correct synchronization can be reestablished if lost, e.g. when emerging from a noisy state. Available information may be lost while synchronization is being established. In a feedback system, however, repeats can be requested until synchronization is re-established with only a slight loss in through-put rate. Thus, in general, minimum time for resynchronization is not as critical. (However, in systems of the type being considered, the resynchronization time must be specified as a small fraction of the expected continuous duration of the good state and may in fact be quite small).

The above remarks illustrate that, with respect to word synchronization, the availability of a feedback link presents both interesting possibilities and new problems. Furthermore, many of the criteria used to evaluate a synchronization procedure undergo a modification or shift in emphasis which is traceable directly to either the addition of a return channel or the change in reception technique from error correction to error detection.

II. REVIEW OF SOME WORD SYNCHRONIZATION PROCEDURES

A variety of word synchronization methods, originally proposed for a feedforward or a feedback channel with fixed statistics, can be considered for adaptation to the given system. A brief summary of some of these methods will be given. For the more obvious and expensive procedures, a few remarks indicating the general techniques involved will be sufficient for the needs of this paper; the reader will find detailed information in the indicated references. A more subtle approach, which at first sight appears considerably less costly and thus more desirable, will be examined in greater detail.

In response to the question, "What can be added to the data being transmitted that will enable the receiver to mark blocks correctly?", a number of framing procedures have evolved. In one method, a special symbol--a signal added to, and distinct from, the original channel signal alphabet--is used to indicate the beginning of blocks. The signal is transmitted immediately before each block, and recognition at the receiver establishes synchronization. In the case at hand, this symbol would be neither a one nor a zero, but a third kind of digit.

As one alternative to the employment of a distinct additional symbol, the waveform representing the first bit of each word could be modified to indicate synchronization. For example, in certain systems amplitude modulation of the first bit in each word has been used [34].

A second alternative is the use of a short sequence of binary digits transmitted periodically or placed as a prefix to each word. Synchronization is then obtained upon recognition of this pattern. The use of such a sequence has been investigated from various viewpoints:

(a) The effect of certain specified system parameters (such as channel error rate, allowable probability of false synchronization and of failure to synchronize to a signal actually sent) on the pattern lengths, composition, and the frequency of pattern transmission required for suitable operation, has been analysed [1]. In a similar fashion, more optimum procedures with respect to frequency of pattern transmission, implementation, and time to obtain synchronization have been established for particular situations [33].

(b) A synchronizing sequence--to be transmitted before each code word--may be chosen first and the other digits which are used in code blocks constrained to keep the synchronizing pattern from appearing within a block or code word. The constraints result in a set of z-tuples called a "prefix synchronized encoding" since each word contains the pattern as prefix. The length and content of the pattern can be chosen so as to make the number of code words satisfying the constraints as large as possible subject to the upper bound fixed by demanding that each word (including prefix) contain a given number of message digits [10]. With appropriate procedures, the encoding can be used to maintain word synchronization. However, such an encoding generally has poor distance properties and is neither linear nor systematic. Thus, both inadequate error control ability (correction/detection) and implementation difficulties are encountered. Some improvement of this situation may be achieved by a judicious choice of additional (stronger) constraints which, in effect, specify a more systematic subset of the original encoding. In this case, a representative word from the encoding will have the form

$$\underline{\longleftarrow A \longrightarrow}\ \underline{B}\ \underline{x}\ \underline{x}\ \underline{x}\ \underline{x}\ \underline{B}\ \underline{x}\ \underline{x}\ \underline{x}\ \underline{x} \cdots \underline{B}\ \underline{x}\ \underline{x}\ \underline{x}$$

where A is the synchronizing pattern, the B's are digits specified by the additional constraints, and the digits labeled x represent unrestricted digits. The unrestricted digits may be taken from the source without further encoding. Although more systematic, this type of encoding still has rather poor error control ability. One possible improvement is to first encode the source for error control (for example, into a linear code), and then use this result to fill the unrestricted digits. However, in this case the synchronizing A-tuple is not protected against channel errors. This can be remedied by reversing the suggested procedure. The source fills the unrestricted digits in the prefix synchronized encoding (subset). The resulting z-tuples are then encoded for error control--transformed into (n,z) linear code words for instance--so that the synchronizing pattern receives the same protection as the source information. However, the output of the error control encoder will then have to be operated on in order to restore the word synchronizing property.[4] This scheme has been investigated and proven useful in terms of a two-way radio link [14].

(c) One can attempt to reverse the role of contraining and constrained elements given above. The block code or class of codes to be used is selected first, and a suitable synchronizing pattern is sought. In the case of binary orthogonal codes [29][5], this approach led to the discovery that, with appropriate changes, but no additions, these codes possess the form of prefix synchronized encodings; hence, they have word synchronizing ability. While the original approach was from the "additive" point of view mentioned previously, the results indicate the comparatively subtle possibility of transforming useful codes (without using the explicit redundancy of a prefix) in order to secure derivative codes with synchronizing ability. This technique will be discussed further on in more detail.

Except for the results on orthogonal codes, all of the preceding methods require the sacrifice of information transmitting ability in order to maintain word synchronization; available power and channel time (or

4. Since the parity checks may resemble the synchronizing prefix.
5. These codes are not necessarily linear. See [26].

bandwidth) that might otherwise be used to transmit information must be used to transmit synchronizing symbols or patterns, thus lowering the information rate, generally without any improvement in error control ability, and at the same time often demanding somewhat more complex decoding procedures. A much more desirable situation would be one in which some method was available that allowed for a continuous stream of digits to be received, and words separated, with no synchronization signal necessary. In this case, although some implicit trade-off with respect to other system parameters might be expected, the more exorbitant "price" of the previous methods would be avoided. The particular method to be considered here is that of using a recently discovered subclass of the so called comma-free codes to maintain word synchronization.

If a sequence of block code words, each of length n,[6] is transmitted in the absence of noise, the receiver is confronted with a sequence of the form

$$\ldots d_n a_1 a_2 a_3 \ldots a_n b_1 b_2 \ldots . b_n c_1 c_2 \ldots . c_n$$

It is evident that if the possible overlap words are not code words, then code words may be delineated by merely comparing all possible consecutive sequences of n digits against the code dictionary. In this case, knowing only the code in use, the receiver can place word commas himself with no additional information required. A code possessing the above property is referred to as comma-free. More precisely, if

$$a_1 a_2 a_3 \ldots\ldots\ldots\ldots a_n \text{ and } b_1 b_2 b_3 \ldots\ldots\ldots\ldots b_n$$

are any two (not necessarily distinct) code words from a code dictionary, the code is said to be invulnerable to synchronization at position r if the

6. Variable length encodings have also been examined for word synchronization ability. See [11].

sequence

$$a_{n+1}a_{n+2}a_{n+3}\ldots\ldots a_n b_1 b_2 b_3 \ldots\ldots b_r$$

is not a code word. If a code is invulnerable at all positions r=1,2,... .....,n-1, then it is comma-free.

In order for comma-free codes to prove useful in maintaining word synchronization in a system which encompasses a noisy channel, at least two additional properties are required:

1) Because of additive errors, the comma positions are no longer uniquely specified. Some reliable statistical procedure for the estimation of commas must be available, with a relatively simple implementation.

2) Since the set of comma-free elements or words effectively replaces a code for the system, both error control ability and ease of code implementation are also required.

Work on comma-free codes began from a constructive point of view [5,12,13], and while sets of comma-free words were evolved, they generally had little error control ability and did not possess simple algebraic properties. Both analysis and implementation proved difficult, if not impossible. While many qualitative features of comma-free codes--such as the maximum dictionary size for given n were investigated, no reasonably general classes of useful codes were forthcoming. When approached from this point of view, the specifications of comma-freedom without added redundancy[7], simple algebraic structure, reasonably general formulation, and ability to be used in such a fashion that a reasonable amount of error control is exhibited, appear too demanding.

7. The prefix synchronized encodings previously mentioned are generally comma-free. The prefix does not represent either information bits or parity bits.

The alternative, of starting with known codes which already possess structure and error control ability and attempting to derive comma-free sets, was eventually more fruitful. Since linear codes possess both error control ability and algebraic structure leading to simple mechanization, the comma-free property in any of these codes would prove very useful. However, one immediately notices that any dictionary that forms a group is not invulnerable to synchronization at any position since it contains the zero vector. In particular, no linear code is comma-free. Thus, a whole class of known and useful codes do not have the desired property. However, it has recently been pointed out that in many of these cases, although the code dictionary itself is not comma-free, the code may have one or more cosets[8] that are comma-free or at least invulnerable in some positions[9]. Thus, rather than use the code words as a basis for transmission, a comma-free coset of the code (if it exists) can be used with no loss in code error-combating ability (since a coset has the same distance structure) and with a word synchronization gain. Moreover, the precise relationship between code and coset is known, so that transitions from a string of code words to a string of coset words and vice-versa is known and easily accomplished. Finally, an apparently simple statistical estimation scheme for placing commas in the presence of noise is available. This word synchronization scheme, if applicable to a reasonably large class of useful linear codes, has all of the obvious requirements of a more ideal solution to the problem of providing word synchronization. In what follows, the results of a more detailed examination [32] of these comma-free codes is summarized. Particular attention is given to the "cost"

8. The structure of linear codes (and the definition of coset) is given in [26].
9. Analagous results have been obtained for the general class of orthogonal codes. See [30].

of using such a scheme in the system previously outlined.

III COMMA-FREE ENCODINGS DERIVED FROM LINEAR CODES

The necessary and sufficient conditions for the existence of comma-free cosets of linear, binary codes have been given by Stiffler [28]. While his results apply in general, the extensive calculations which are encountered in investigating each code individually have been avoided only in the case of cyclic codes, by the establishment of the following:

Theorem: Any (n,k) binary cyclic code has a coset, specified by coset leader $c=(10000\ldots0)$, which is invulnerable to synchronization at all positions r which satisfy either of the inequalities

$$r \leq n-k-1 \quad \text{or}$$

$$n-r \leq n-k-1.$$

Moreover, if $k \leq (n-1)/2$, this coset[10] is comma-free. If $k > (n-1)/2$, no coset of the code is comma-free.

Figure 3 illustrates some results based on Stiffler's theorems for cyclic codes. As can be seen (and extrapolated) from Figure 3, this formulation (1) generates a large class of comma-free or invulnerable codes, where each code satisfies the additional conditions that (2) it is systematic and (3) has significant error correction/detection ability, with (4) each word bit representing either an actual (source) information bit or a parity check bit. In a general sense, these properties ensure the availability of comma-free codes for application in a variety of systems, their practical utility in terms of implementation and error control ability, and their low cost[11] with

10. This is not the only coset. For example, the theorem applies with $c=(0000\ldots.001)$. Other invulnerable or comma-free cosets also exist.
11. Relative to the conventional "additive" approaches which inject special sync symbols of some type and thus decrease the rate k/n without any contribution to the error control ability.

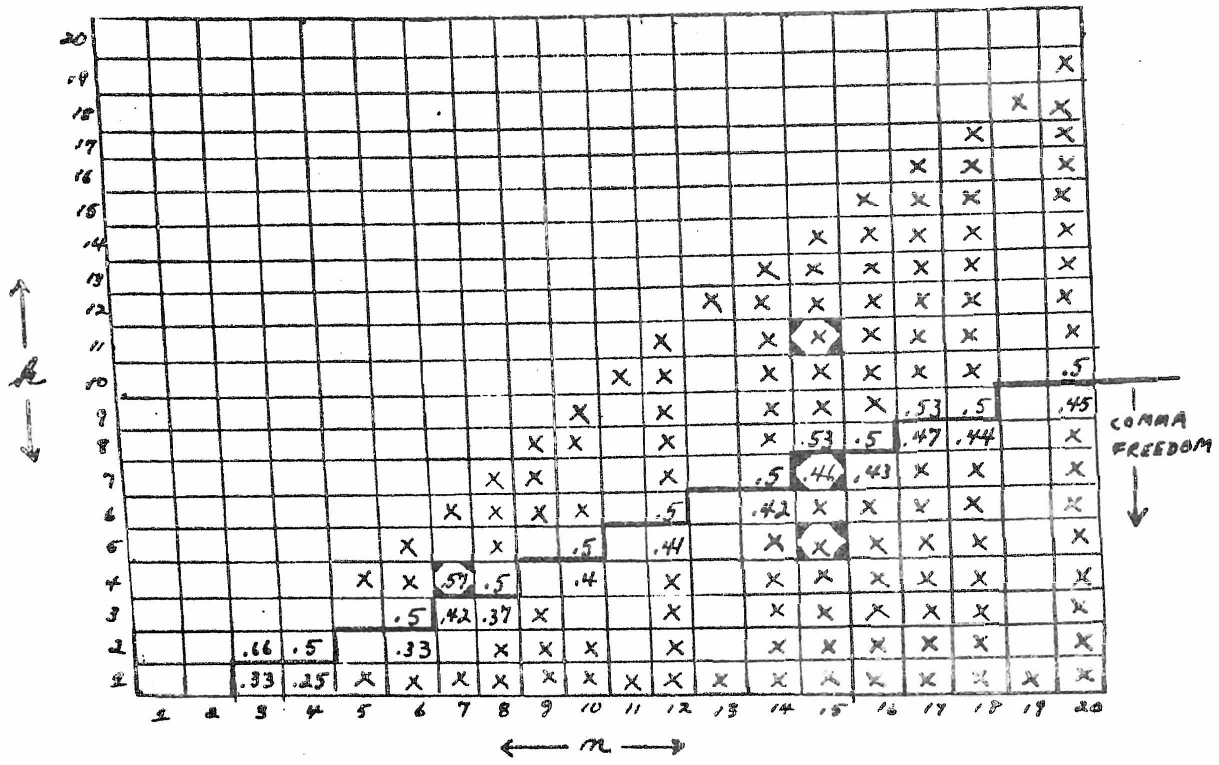


FIGURE 3

SHORT CYCLIC CODES AND COMMA FREEDOM

X	;	Code exists.
#	:	Code exists; number given is k/n for that code.
⬢	:	Code describable as Bose-Chaudhuri code.

Empty box indicates that no cyclic code exists for given values of k and n.

All codes below the dark line have comma-free cosets. All codes above the dark line have various degrees of invulnerability (see theorem).

Graph constructed by reducing X^n+1 to irreducible factors. For example,

$$X^5+1=(X+1)(X^4+X^3+X^2+X+1)$$

over GF(2) where both factors are irreducible. Hence, there are only two generating functions for n=5. Note that g(X)=X+1 generates a (5,4) cyclic code.

respect to the system's capability for reliable information transmission. Thus, as a class, cyclic codes[12] possess (implicitly) a great deal of synchronization ability. Approximately $\frac{1}{2}$ of the available codes possess comma-free cosets, and the majority of these codes have from reasonable to good correction/detection ability [26]. The remaining half of the cyclic codes possess invulnerability properties. The simple relationship between code and coset allows one to simultaneously take full advantage of the systematic error control properties of the linear code, and the synchronization properties of the coset.

However, the use of cyclic coset codes with a comma-free property limits the attainable transmission rate to less than one-half, i.e. as $n \to \infty$, $k/n \to \frac{1}{2}$. This limitation may or may not be severe, depending on the nature of the system in which the code is to be used. For example, to adequately protect a system (small P_e) it may be necessary to devote at least one-half of the bits in each word to parity bits. In this case, comma-freedom may be secured without further restricting the rate, since comma-free codes are available with a rate that is reasonably close to $\frac{1}{2}$ (even when a short length constraint is imposed as in Figure 3). If however, the required redundancy does not limit the rate to $\leq \frac{1}{2}$, then comma-freedom (if a cyclic code is used) may impose a somewhat severe rate constraint. If only invulnerability in a certain number of places is required, the rate constraint is considerably weaker; so much so that the rate constraints dictated by the required protection will usually dominate. In the case where comma-freedom is required and does impose a severe rate constraint if cyclic codes

12. The structure of cyclic codes is given in [26]. As can be seen, this subclass contains many codes of virtually all lengths, has a wide range of error control ability relative to both independent random errors and burst type errors, and represents codes which are among the easiest to mechanize.

are used, one can consider the possibility of employing a comma-free coset (if one exists) of a non-cyclic linear code of the desired size or consider the constraint as part of the price of obtaining the necessary synchronization. Once the total "cost" is ascertained, it may then be compared to that of the conventional "additive" approaches to determine which method is more desirable.

As indicated in Appendix I, reliable operation (negligible P_e) of the given system does not necessarily demand that one-half the bits in each word be devoted to check bits. However, in cases where equipment and calculation loading is to be minimized by a trade-off with transmission rate, a rate appreciably greater than $\frac{1}{2}$ cannot be expected. Hence, if necessary one can attempt to use a comma-free coset of a cyclic code to provide word synchronization. Virtually any amount of invulnerability may also be secured at a negligible (initial) cost.

Thus, the price of securing comma-free or invulnerability properties for transmission is negligible in many cases. In the case of cyclic codes, the central restriction is a rate limitation which may or may not be "costly". In the case of non-cyclic linear codes, this restriction does not necessarily apply: the complications involved in any investigation in this area have precluded any general results. Indeed, the practical problem of determining linear, systematic, non-cyclic codes with comma-free cosets for codes of any appreciable length, is still unsolved.

The above, of course, does not indicate the "cost" of using such a procedure to maintain word synchronization.

IV THE USE OF COMMA-FREE CODES

By virtue of the comma-free property, when a sequence of words is received without error, the code words can be identified by comparing possible n-tuples against the code dictionary. Any sequence of 2n-1 digits considered will contain only a single error free word (n-tuple), namely the one for which synchronism is correct. However, under the assumption that additive errors occur on the channel, the comma-free property is not preserved. The transmitted elements of the comma-free coset are not necessarily received as elements of this comma-free coset; therefore, the comma positions may not be apparent. A statistical establishment of commas must be made according to some procedure such as the following:

The receiver chooses a synchronization position and observes several words. Each word is transformed from the coset back to the code by the addition of c (the coset leader) and decoded into some code word according to the maximum likelihood criterion. The average number of errors per word for this synchronization position is calculated and recorded. The receiver then goes back to the original received sequence, shifts the assumed sync position (in a consistent search pattern), and repeats the process. After n assumed synchronization positions (for an (n,k) code) have been considered, the position with the lowest corresponding average error/word figure is taken as representing the true word synchronization. Decoding according to the desired criterion may then be undertaken. When invulnerability in several positions is in use, the same procedure applies except that the invalid (vulnerable) positions are excluded from consideration.

The cost involved in using a comma-free code with the above procedure can easily be seen to include the decoding--generally including correction--of every possible received n-tuple. At the receiver the synchronization information is in terms of the number of errors actually present in the n-tuple determined by any synchronization position SP=1,2,..,n. This information can be extracted only "on the average", requiring that for any fixed position the N previously received "words" be considered. The placing of commas necessitates the recognition and counting of these errors to a degree of accuracy commensurate with the desired reliability. In general, the reliability of the statistical estimation scheme considered depends on at least

(a) the number of errors contained in any overlap word due to the comma-free property. This in turn is dependent on the actual linear code used and the coset element which specifies the comma-free code.

(b) the number of errors induced in each overlap word by the channel. This is directly dependent on P_o, the average per digit error rate.

(c) the recognition ability of the code in use and how this ability is used. This is a function of the distance structure of the code (minimum distance), the code parameters n and k, and the designer's correlation between the decoding mode used and the actual counting procedure. For example, when any single errors are correctable and an unknown number and type of higher weight error patterns are detectable, some decision must be made on the contribution of each detection to the count.

(d) the sample size N which must be used to validate the contention that "on the average" the correct synchronization position contains fewer errors than any asynchronous position. This must be large enough so that on the average for any asynchronous position the errors mentioned in (a) and (b) above add rather than cancel--and so distinguish the correct position (provided they are recognized and counted).

The precise nature of the scheme's dependence on these quantities is quite complex in general. However, in the extreme case of small error rate, a sample size of several words, and reasonable error recognition ability of the code, the correct comma positions are easily determined with a high degree of reliability. For high error rates, the synchronization information transmitted in terms of the comma-free code is not available at the receiver and the procedure breaks down, generally giving an incorrect indication. Although commas must be estimated continuously, various testing policies may be devised which guard against the use of results obtained under unfavorable circumstances (fade). During this time, synchronization is determined solely by a clock which was set by some previous reliable estimate and which will be corrected after emergence from the fade.

The implementation of this scheme requires that the several (N) words used to compute each average be stored at the receiver. Moreover for each new bit received, a new average is computed. Thus the procedure requires the continuous processing and storage of several hundred (nN) previously received bits. In addition, regardless of the synchronous decoding mode, care must be taken to ensure that the code used to protect the system has an amount of recognition (generally correction) ability sufficient to maintain a reliable comma estimating procedure. In general, the synchronous decoding mode may be independent of the decoding done for comma estimation, so that much of the equipment and calculation required is in addition to that required for synchronous decoding. Finally, if synchronization is lost during a fade, the comma estimating procedure--which has a memory of N previously received words--in conjunction with the synchronization-correction testing procedure introduces a resynchronization delay of perhaps several hundred bits when emergence from the bad state occurs.

Notice that although the errors in word synchronization are confined to within $\frac{1}{2}$ a code word length in either direction of the correct position, there are still n possible synchronization positions, so that comma-freedom is required to determine the correct position. Invulnerability can only be used if errors in synchronism are confined to some smaller fraction e.g. $\frac{1}{4}$ n) of a code word. Since n will usually be large, it is not unreasonable to assume that in the given system the synchronization errors are so confined, and that invulnerability can be used to maintain synchronization. However, it can be shown that although the calculations at the vulnerable positions are avoided, the equipment required for comma estimation in this case is the same. Thus, the comments given for comma-freedom largely apply, and the "price" of using this lesser amount of synchronization ability is substantially the same.

The "cost" involved in the above can vary widely from system to system. The code ability required and the decoding involved in placing commas may or may not be completely in excess of the system's synchronous code and decoding requirements. For example feedback system with simple detection-retransmission will require correction decoding for the comma estimations which would not be required for synchronous operation. However, even when the decoding mode of comma placement "matches" the system's synchronous decoding mode--as in a simple feedforward-only system--only the decoding for SP=n is non-additional, since this represents synchronous decoding. In general, the comma estimating process and the fact that the information must be extracted "on the average", introduce somewhat expensive system additions.

A common ground for the comparison of these comma-free codes and the more conventional "additive" methods is difficult to find. Compared to the special sync pattern procedures, the comma-free approach demands considerably more work in order to determine the correct synchronism, since the sync-acquisition strategy is more involved. On the other hand, depending on the code size and redundancy required to protect the system, and the amount of synchronization ability desired, this approach may not require the sacrifice of information transmitting capability in order to maintain synchronization. Where this latter characteristic is a predominant system "desirable", the available comma-free codes should certainly be considered. For example, the requirements of feedforward-only systems (significant redundancy, full correction decoding, et cetera) are compatible with the procedures necessary for the use of comma-free or invulnerable codes, so that the price paid for their synchronization ability is somewhat competitive. The structure of feedback systems, however, is much less compatible, and the use of these codes--while applicable in terms of providing synchronization--involves rate constraints, implementation procedures, et cetera, which severely limit the area of profitable application. The conventional "additive" procedures have the distinct advantage of being applicable under virtually all circumstances, generally with comparable or less overall cost. In addition, the particular possible advantage of the comma-free approach--avoiding an additional rate loss due to synchronization signals--is of somewhat less significance in the feedback context. In the above sense, the comma-free approach examined here certainly does not represent "the" solution to the word synchronization problem, especially with respect to feedback systems.

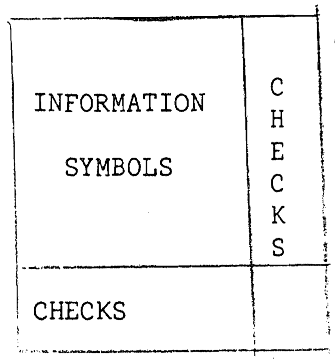
V ALTERNATIVE APPROACHES

With the above investigation as background, there are several closely related possibilities that should be investigated, the main endeavor being directed toward developing, if possible, a psuedo comma-free procedure for word synchronization (in particular, one that is better suited to the feedback system under consideration). The following represents a summary of an investigation of one alternative scheme [32]. Several other possible schemes are suggested and reference made to potential areas that have been largely neglected.

Two phenomena involved in the above use of comma-free codes lead to the difficulties mentioned. First,comma-freedom assures only that any overlap word to be decoded is not a code word. The distance of this overlap word from a true code word can not be specified in general, and in many overlaps this distance is minimal (one). The few errors in any single overlap could possibly be cancelled by channel induced errors so that the assumed position is likely to be mistaken for the correct one. This situation can only be avoided by considering that on the average over N code words it becomes unlikely. Second, for all possible synchronization positions, many of the words examined in any N word average could be expected to contain at least one error. Hence to distinguish the correct position more errors must be recognized and entered in the count, leading to correction mode decoding requirements. The avoidance of these two points would represent a significant improvement of this method.

One possible scheme is to develope comma-free codes in which any overlap to be decoded differs from any true code word in at least T positions, thus providing more synchronization information per bit or per word. For a small error rate, when any word determined by an incorrect synchronization position is decoded (after complementation) it will differ from a true code word by something close to T errors. Hence, without the averaging procedure, its synchronization position is not likely to be mistaken for the correct synchronization position. The consequent storage, calculations and delay might be avoided or significantly reduced. However, in this case the count requires the recognition of almost all the errors contained in any n-tuple and hence the use of additional correction ability. The correction decoding might be avoided, while maintaining a good estimate of the actual number of errors present, by considering each word as a sequence of many smaller words, applying simple error detection to these smaller words, and adding the number of errors detected (each detection counting as a single error). Such a smaller synchronization mesh could be obtained by considering iterated codes.

The basis for the approach is found in the consideration of an iterated code word in matrix from (as at the left), where each row word is



encoded as an (n_1,k_1) code word, and each column word is encoded as an (n_2,k_2) code word. The (n_1,k_1) and (n_2,k_2) codes are then chosen to be invulnerable and/or comma-free. Under certain conditions an iterated code coset may be formed which, as a code, has a "stronger" comma-free type property. One set of conditions on the composition codes is the following (assuming cyclic composition codes):

1) Each code must contain the vector of all ones, and

2) Each code must have a coset which is either invulnerable in r positions or comma-free, and which is specified by c=(1000...0).[13]

For example, it is easily shown that if (1) is satisfied and both the (n_1,k_1) and the $n_2,k_2)$ codes have comma-free cosets, then the resulting (n_1n_2,k_1k_2) iterated code has a coset specified by

$$\hat{\xi} = \begin{matrix} 1 & 0 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 1 & 1 & \ldots & 1 \\ \vdots & \vdots & & & & \vdots \\ 0 & 1 & 1 & 1 & \ldots & 1 \end{matrix}$$

(i.e.$\hat{\xi}$ is a member of the coset) which is comma-free in the "stronger" sense that any overlap of any two coset words differs from a true coset word by at least $T=\min(n_1,n_2)$ positions. Thus the (225,49) iterated code formed with the (15,7) code as both the row and column codes, is comma-free of order T=15: since the (15,7) contains the vector of all ones and has a comma-free coset ($k \leq (n-1)/2$) specified by c=(100000000000000). Thus, with respect to the iterated code coset specified by

$$\hat{\xi} = \begin{matrix} 1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 0&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ & \vdots & & & & & & & & & & & & & \cdot \\ 0&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \end{matrix} ,$$

every possible overlap "word" differs from any true coset word in at least

13. Condition (2) can be given in much more general form: i.e. cosets specified by c_1 and c_2, both arbitrary. As long as (1) is satisfied, the appropriate $\hat{\xi}$ can be formed. See [32].

15 places.

Similarly, row code comma-freedom and column code invulnerability can be combined to yield an iterated code which is invulnerable in the stronger $T=\min(n_1,n_2)$ sense: the number of invulnerable positions (rn,) depending on the invulnerability of the column code (r). For example, the (15,11) cyclic code can be shown to contain the vector of all ones. The code has a coset specified by c=(100000000000000) which is invulnerable to synchronization at all positions

$$|r| \leq n-k-1$$

i.e. at positions 1,2,3,12,13, and 14. The (225,77) iterated code formed with the (15,7)code as the row code and the (15,11) code as the column code has a coset specified by

$$\hat{\xi} = \begin{matrix} 1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ \cdot \\ \cdot \\ 0&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \end{matrix}$$

which possesses the "stronger" property that the iterated code coset is

(a) not only invulnerable to synchronization at all positions $|r| \leq 45$ i.e. synchronization positions SP=1,2,3,....,44, 45,180,181,....,224,

but

(b) at any of these positions the overlap word considered differs from a coset word by at least 15 positions.

The requirement (1) stated previously is in fact notunduly severe. For cyclic codes, Figure 4 duplicates Figure 3 and indicates which of these codes satisfy condition (1). Every code designated in Figure 4 is eligible

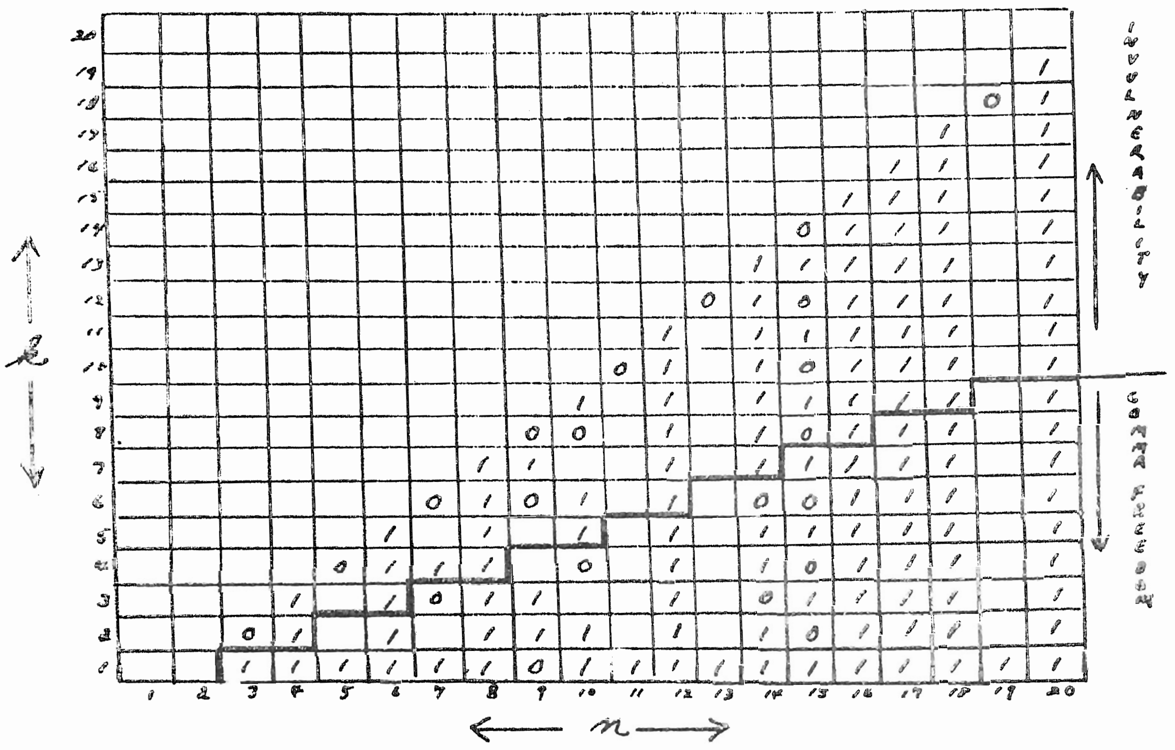


FIGURE 4

SHORT CYCLIC CODES CONTAINING THE VECTOR

OF ALL ONES

$\boxed{0}$: Code exists but does not contain the vector of all ones.

$\boxed{1}$: Code exists and contains the vector of all ones.

as a composition code (with itself or other codes). Thus, the formulation yields a reasonable number of linear comma-free codes known to be of high order, and a large number of linear codes with the "stronger" invulnerability property.

The implementation of the above type codes in the given feedback system is easily accomplished. The "recognition and counting" of errors for the various sync-positions is performed on the sub-synchronization mesh defined by the iterated code--for example, simply by counting the number of rows and columns detected to be in error in each received iterated "word". Thus, correction mode decoding is avoided by detections based on shorter code words. The dependence of the comma estimating procedure on N previously received (iterated) words has been removed, with a consequent reduction in the storage and calculation requirements. Also, there is a substantial improvement with respect to resynchronization delay: in general at most one iterated word transmitted and received completely during the good state is required before resynchronization is expected to occur. Finally, since more synchronization information is available at the receiver, the procedure is considerably more reliable than the previous one at higher rates.

Unfortunately, the developement is inadequate in other respects. In general, the class of such codes does not include all codes specified by arbitrary values of n and k--since not all linear codes are admissible as composition codes. In terms of cyclic composition codes, the majority of which are admissible since they contain the vector of all ones and possess comma-free or invulnerability properties, the rate restrictions given for these "simple"[14] cyclic comma-free codes are amplified. Thus the derived codes exist only for certain values of n and k and for small k/n: the derived iterated codes which are comma-free of order $\min(n_1,n_2)$ necessarily have $k_1k_2/n_1n_2 < \frac{1}{4}$, while the invulnerable iterated codes necessarily have $k_1k_2/n_1n_2 < \frac{1}{2}$. Finally, little is known in general about the class of non-cyclic linear codes which possess comma-free cosets. Thus a moderately large class of codes which are readily seen to possess this "stronger" comma-free property are very "costly" to use, while the investigation of another possible class must be pursued on the basis of a trial and error examination of the particular composition codes.

The search--among codes that are well suited to synchronous operation, e.g. cyclic codes--for codes that have desirable synchronization or framing properties[15], has not been completed. Although one of the most promising classes of such codes proposed to date has been shown to be somewhat inadequate[16], there are many remaining possibilities. For example, it is clear that full comma-freedom is not necessary. In general, a code having the property that most of the overlap words are not code words has valuable framing ability. However, little work has been done in this area of "almost" comma-free codes. The majority of investigations of comma-freedom to date have been conducted on the deterministic basis of satisfying the definition--while largely neglecting the statistical framework in which the property is to be used. Linear codes per se have been by-passed with respect to self synchronizing properties since the presence of the zero vector violates the comma-free definition. However, in many cases, the probability of having to transmit two consecutive zero vectors is quite small. In

14. Refers to fact that the order is either minimal or unknown.
15. These include not having special sync symbols.
16. For example, we have seen that no cyclic code has a comma-free coset unless $k/n < \frac{1}{2}$.

addition, even when such a condition arises, the effect on the statistical comma-estimating procedure is generally not so great as to completely invalidate any results. The possibility of isolating and either removing or replacing "troublesome" vectors in "almost" comma-free codes could also be considered. The potential of "almost" comma-free codes with respect to synchronization, and that of linear codes with respect to "almost" comma-freedom has never really been evaluated. In the above sense, it is clear that all linear codes have some degree of synchronization ability. By definition, such a code has a logical structure that must be satisfied by each code word. In certain cases, one would expect that, on the average, this structure would be verified for correct synchronism and denied to some degree for asynchronous positions. In this sense, random linear codes have much more synchronization ability than, say, cyclic codes, since with high probability, overlap words in the latter case will be code words (a cyclic permutation of a code word is another code word). For random codes, the most desirable situation would seem to be one where the number of code vectors is small in relation to the total number of n-tuples appearing in the standard array, and the great majority of coset leaders (minimum weight in the coset) are high weight. Overlaps will then most often be non-code n-tuples which differ from code words by significant distances. However, since small weight error patterns are not in predominance, such a code may not be well suited for synchronous operation, and some compromise between error control and framing ability may be necessary. In terms of cyclic codes, the place to look is in the area of "reversible" transformations which yield a range space sufficiently removed from the cyclic structure;

so that significant sync ability (in the above sense) is possible. Some encouraging results in this area are already available [9]. Since a great number of potential transformations exist, more results should be forthcoming. In any case, it seems clear that the initial approach of starting with codes well suited to synch acquisition, and working toward codes effective in the synchronous mode, has been properly reversed.

APPENDIX I

ESTIMATE OF THE CODE REDUNDANCY REQUIRED TO PROTECT THE SYSTEM UNDER CONSIDERATION

Let P_e = the probability of decoding error, i.e. the probability that a code word is accepted at the receiver and incorrectly decoded. We require that P_e be very small, $\sim 2^{-30}$ or 10^{-10}. Let S=the number of candidate synchronization positions available to the receiver. Clearly max(S)=n, and for synchronous decoding S=1. If synchronization errors are confined to less than $\frac{1}{2}$ a code word in either direction of the correct position, S=n.

$$P_e = P_G \cdot P(e/G) + P_B \cdot P(e/B)$$

where P_G is the probability that the channel is in the good state and P(e/G) is the probability of error given the good state.

We assume that the code to be used is chosen so that P(e/G) is sufficiently small. In general this is a constraint on the code's error correcting or detecting ability in relation to the channel error statistics of the good state.

Assuming that P_B is reasonably large, the code must be chosen so that

$$P(e/B) = P(\text{accept}/B) \cdot P(e/\text{accept},B) < 2^{-30}$$

To approximate this expression we assume that when the system is in the bad state ($P_o \sim \frac{1}{2}$), the receiver, after assuming some synchronization position, is essentially looking at a random sequence of words. Once he accepts a word it is equally likely to be any code word. Hence

$$P(e/\text{accept},B) = 1 - \frac{1}{2^n} \sim 1 \quad \text{for } k>4 \quad ;$$

also

$$\underset{\text{synchronous}}{P(\text{accept}/B)} = \frac{\#\text{ of acceptable or correctable words}}{\text{total }\#\text{ of possible words}}$$

Therefore if the receiver can correct e or fewer errors (e=0 for simple error detection), it is required that

$$P(\text{accept}/B) = S \sum_{i=o}^{e} \binom{n}{i} 2^k \sim S2^{-(n-k)} \sum_{i=o}^{e} \binom{n}{i} \leq 2^{-30}$$

since the number of acceptable words is increased by virtue of the various possible synchronization positions.

(a) If complete comma-freedom is required in a detection retransmission system, S=n and

$$P_e \sim n2^{-(n-k)} \leq 2^{-30}$$

or $$(n-k) \geq 30 + \log_2(n)$$

For n in the 100⟷200 range, $\log_2(n)$ is 7 or 8. Therefore, at most we require (n-k) > 40, i.e. 40 check bits per word. Theoretically, a large k/n is available by taking n (or k) sufficiently large. However, the storage and calculation (binary comparisons per word) requirements increase in proportion to k(n-k) [24]. For n in the range 100⟷200, the possible k/n variation is .4⟷.7. Thus, if storage and calculation loading is to be minimized a k/n limitation of $\frac{1}{2}$ (imposed by using a comma-free code) is not unduly severe. In general however, the system can operate reliably--with synchronization by any conventional "additive" method--at transmission rates considerably above this. Notice that the effect of uncertainty in synchronization contributes little (additive $\log_2$ term) in the determination of the required redundancy. The results are only slightly altered (less redundancy) if several position invulnerability is required, or if a sync correction testing procedure narrows the candidate sync positions down to the single "extrapolated" one specified by the clock.

(b) If comma-freedom is required in a system that includes single error correction with the detection retransmission, then

$$P_e \sim n(1+n)2^{-(n-k)} \leq 2^{-30}$$

or $$(n-k) \geq 30 + \log_2(n) + \log_2(n+1) \sim 30 + 2\log_2(n)$$

For the same range of n, $2\log_2(n) \sim 15$ so that we require

$$(n-k) > 15$$

The associated possible variation of k/n is .35 .67. Again in certain cases, the system requirements (other than synchronization) may require that the code length be short, implying that the rate k/n is constrained to $< \frac{1}{2}$.

Bibliography

[1] Barker, R. H., "Group Synchronizing of Binary Digital Systems," in Communication Theory, W. Jackson, Ed., Butterworths Scientific Publications, London, Eng. pp. 273-287, 1953.

[2] Bernice, R. J., "An Analysis of Retransmission Systems," and "Comparisons of Error Control Techniques," IEEE Trans. on Communication Technology, vol. IT-12, no. 4, pp. 135-144.

[3] Burton, H. O., E. J. Weldon, Jr., "Cyclic Product Codes, "IEEE Trans. on Information Theory, vol. IT-11, no. 3, pp. 433-439.

[4] Costas, J. P., "Synchronous Communications, "Proc. IRE, vol. 44, pp. 1713-1718, Dec. 1956.

[5] Crick, F. H. C., J. S. Griffith, and L. E. Orgel, "Codes without Commas," Proc. Nat'l Acad. Sci. U.S., vol. 43, pp. 416-421, May 1957.

[6] Davenport, W. B., and W. L. Root, Random Signals and Noise, McGraw-Hill Book Co., Inc., New York, N.Y., Chapter 14, 1958.

[7] Eastman, W. L., and J. Evan, "On Synchronizable and PSK Synchronizable Block Codes," IEEE Trans. on Information Theory, vol. IT-10, pp. 351-356, Oct. 1964.

[8] Eastman, W. L., "On the Construction of Comma-Free Code," IEEE Trans. on Information Theory, vol. IT-11, no. 2, pp. 263-267, April, 1965.

[9] Frey, A. H., Jr., "Message Framing and Error Control," IEEE Trans. on Military Electronics, vol. 9, no. 2, pp. 143-147, April, 1965.

[10] Gilbert, E. N., "Synchronization of Binary Messages," IRE Trans. on Information Theory, vol. IT-6, no. 4, pp. 470-477, Sept., 1960.

[11] Gilbert, E. N., and E. F. Moore, "Variable Length Binary Encodings," Bell Syst. Tech. J., vol. 38, pp. 933-967, July, 1959.

[12] Golomb, S. W., B. Gordon, and G. R. Welch, "Comma-Free Codes," Canadian J. Math., vol. 10, no. 2, pp. 202-209, 1958.

[13] Golomb, S. W., L. R. Welch, and M. Delbruck, "Construction and Properties of Comma-Free Codes," Biol. Med. Danske Vid. Selsk., vol. 23, no. 9, pp. 1-34, 1958.

[14] Hansen, J. C., "An Integrated 'Error Free' Communication System," IEEE Trans. on Space Electronics and Telementry, vol-set-9, no. 3, pp. 92-98, Sept., 1963.

[15] Harris, B., and K. C. Morgan, "Binary Symmetric Decision Feedback Systems," AIEE Trans. on Communications and Electronics, vol. 38, Sept., 1958.

[16] Harris, B., A. Hauptschein, K. C. Morgan, L. S. Schwartz, "Binary Communication Feedback Systems," AIEE Trans. on Communication and Electronics, vol. 40, pp. 960-969, January, 1959.

[17] Jacobs, I., "Optimum Error Detection Codes for Noiseless Decision Feedback," IRE Trans. on Information Theory, vol. IT-8, no. 6, pp. 359-371, Oct., 1962.

[18] Jiggs, B. H., "Recent Results in Comma-Free Codes," Canadian J. Math., vol. 15, pp. 178-187, 1963.

[19] Kantz, W. H., "Fibonacci Codes for Synchronization Control," IEEE Trans. on Information Theory, vol. IT-11, no. 2, pp. 284-292, April, 1965.

[20] Kendall, W. B., and I. S. Reed, "Path Invariant Comma-Free Codes," IRE Trans. on Information Theory, vol. IT-8, pp. 350-355, Oct., 1962.

[21] Lawton, J. C., "Comparison of Binary Data Transmission Systems," Proc. Nat'l. Conv. on Military Electronics, Wash., D.C., pp. 54-61, June, 1958.

[22] Magnum, J. P. "Digital Synchronization of PCM Telemeters," Proc. Nat'l. Telemetering Conference, paper no. 5-3, May, 1962

[23] Metzner, J. J., and K. C. Morgan, "Coded Binary Decision Feedback Communication Systems," Transactions on Communication Systems, Institute of Radio Engineers, vol. CS-8, no. 2, June 1960.

[24] Metzner, J. J., and K. C. Morgan, "Reliable Fail-Safe Binary Communication," Wescon Convention Record, Institute of Radio Engineers, vol. 4, pt. 5, pp. 192-206, August, 1960.

[25] Payne, R. C., and P. Painter, Jr., "Bit Synchronization and Data Regeneration Techniques," Proc. Nat'l. Telemetering Conf., vol. 2, paper no. 5-2.

[26] Peterson, W. W., Error Correcting Codes, M.I.T. Press Cambridge, Mass., and John Wiley and Sons, Inc., New York, N.Y., 1961.

[27] Sellers, F. F., Jr., "Bit Loss and Gain Correction Codes," IRE Trans. on Information Theory.

[28] Stiffler, J. J., "Comma-Free Error Correcting Codes," IEEE Trans. on Information Theory, vol. IT-11, no. 5, pp. 107-112, Jan. 1965.

[29] Stiffler, J. J., "Synchronization Methods for Block Codes, "IRE Trans. on Information Theory, vol. IT-8, no. 5, pp. S 25-34, Sept. 1962.

[30] Stiffler, J. J., "Synchronization of Telemetry Codes," IRE Trans. on Space Electronics and Telemetry, vol. SET-8, no. 2, pp. 112-117, June, 1962

[31] Van Horn, J. H., "A Theoretical Synchronization System for Use With Noisy Digital Signals," IEEE Trans. on Communication Technology, vol. 12, no. 3, pp. 82-90, Sept., 1964.

[32] Waters, J. T., "Word Synchronization and Comma-Freedom in Decision Feedback Systems, "Master's Thesis, Dept. of Applied Analysis, State University of New York at Stony Brook, Stony Brook, N.Y., June 1966.

[33] Willard, M.W., "Optimum Code Patterns for PCM Synchronization," Proc. Nat'l Telemetering Conf., paper no. 5-5, May, 1962.

[34] Willard, M. W., "PCM Telemetry Synchronization," Proc. Nat'l. Telemetering Conf., pp. 11-51-11-86, May, 1961.

[35] Wintz, P. A. and J. C. Hancock, "An Adaptive Receiver Approach to the Time Synchronization Problem," IEEE Trans. on Communication Technology, pp. 90-96, March 1965.

[36] Wolf, J. K., "On Codes Derivable From the Tensor Product of Check Matrices," IEEE Trans. on Information Theory, vol. IT-11, no. 2, pp. 281-284, April, 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)* College of Engineering, State University of New York at Stony Brook, Stony Brook, Long Island, New York	2a. REPORT SECURITY CLASSIFICATION Unclassified
	2b. GROUP
3. REPORT TITLE The Framing Problem in Block Coded Feedback Communications Systems	
4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)* Scientific Report Interim	
5. AUTHOR(S) *(Last name, first name, initial)* Dollard, Peter M.; Waters, Jerome T.	
6. REPORT DATE August, 1966	7a. TOTAL NO. OF PAGES 39 / 7b. NO. OF REFS 36
8a. CONTRACT OR GRANT NO. AF 19(628)-3865	9a. ORIGINATOR'S REPORT NUMBER(S) Technical Report No. 71; Scientific Report No. 2
b. PROJECT AND TASK NO. 5628-01	
c. DOD ELEMENT 61445014	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)* AFCRL-66-655
d. DOD SUBELEMENT 681305	
10. AVAILABILITY/LIMITATION NOTICES NOTICE NO. 1	
11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Hq. AFCRL, OAR (CRB), United States Air Force, L.G. Hanscom Field, Bedford, Mass.

13. ABSTRACT

The literature on word synchronization and framing of block coded binary transmission systems is reviewed in depth, with specific references to an extensive bibliography. Particular attention is paid to the use of comma-free codes derives from linear codes without added redundancy in such a way that the error control properties of the codes are fully retained. The review is directed towards the use of these techniques in a two way system with feedback, and includes a discussion of the special considerations which arise in this application. It is shown that all these techniques involve a substantial loss of ultimate information rate, and/or a substantial increase in system complexity over that required for error control alone. A modification of the comma-free codes is suggested in which iterated codes are employed to provide a sync grating intermediate between digit and word sync. This is shown to avoid much of the added complexity of comma-free codes, but at an additional cost in information rate. The report concludes with a brief discussion of still another technique which as yet has received too little attenion. With this approach one does not seek absolute sync protection in the absence of channel errors, rather, one seeks highly reliable sync protection with the statistics of both source and channel taken into account.

DD FORM 1 JAN 64 1473

Unclassified
Security Classification

Unclassified

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Feedback communications						
Synchronization						
Framing, word						
Comma-free codes						
Prefix codes						
Coset codes						

Unclassified

Security Classification