

FAA-RD-75-62

**Project Report
ATC-49**

DABS Uplink Coding

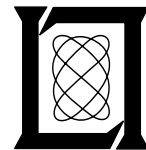
J. T. Barrows

25 July 1975

Lincoln Laboratory

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LEXINGTON, MASSACHUSETTS



Prepared for the Federal Aviation Administration,
Washington, D.C. 20591

This document is available to the public through
the National Technical Information Service,
Springfield, VA 22161

This document is disseminated under the sponsorship of the Department of Transportation in the interest of information exchange. The United States Government assumes no liability for its contents or use thereof.

1. Report No. FAA-RD-75-62	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle DABS Uplink Coding		5. Report Date 25 July 1975	6. Performing Organization Code
		7. Author(s) J. T. Barrows	8. Performing Organization Report No. ATC-49
9. Performing Organization Name and Address Massachusetts Institute of Technology Lincoln Laboratory P.O. Box 73 Lexington, Massachusetts 02173		10. Work Unit No. (TRAIS) 45364 Project No. 034-241-021	11. Contract or Grant No. IAG DOT-FA-72-WAI-261
		13. Type of Report and Period Covered Project Report	
		12. Sponsoring Agency Name and Address Department of Transportation Federal Aviation Administration Systems Research and Development Washington, D.C. 20591	14. Sponsoring Agency Code
15. Supplementary Notes The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology under Air Force Contract F19628-76-C-0002.			
16. Abstract <p>This report details the coding techniques incorporated into the DABS uplink design. Justification is given for the error control method selected in terms of the link characteristics and design constraints. Performance results, including extensive evaluation by simulation and bench test, are presented for the selected code.</p> <p>A binary shortened cyclic code having 24 redundant bits was selected. The overhead due to the code redundancy is minimized by a scheme in which the parity check bits are overlaid on the discrete address field in the encoded message. This code is shown to have the capability of protecting a DABS transponder from accepting an erroneous uplink message with an error probability of less than 10^{-7} in a severe interference environment.</p> <p>This same code will be used in the DABS downlink in a burst erasure correction mode. Results of the analysis of the downlink will appear in a separate report.</p>			
17. Key Words DABS message encoding Cyclic codes Error detection Burst errors ATCRBS interference		18. Distribution Statement Document is available to the public through the National Technical Information Service, Springfield, Virginia 22151.	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 42	

**PAGE PURPOSELY
LEFT BLANK**

TABLE OF CONTENTS

<u>Section</u>	<u>Page</u>
I. INTRODUCTION AND STATEMENT OF THE PROBLEM	1
I-1. Statement of the Coding Problem	1
II. UPLINK CODE PROPERTIES AND IMPLEMENTATION	5
II-1. Code Properties	5
II-2. Encoding Methods; Address/Parity Overlay	9
III. CODE PERFORMANCE IN VARIOUS UPLINK INTERFERENCE ENVIRONMENTS	18
III-1. ATCRBS Uplink Interference Environment	20
III-2. Determination of the Probability of Undetected Message Error by Simulation	23
III-3. Experimental Measurement of Code Performance	29
III-4. Analytic Approximations to Code Performance	33
IV. SUMMARY	35
REFERENCES	37
ACKNOWLEDGMENT	38

LIST OF ILLUSTRATIONS

<u>Fig.</u>	<u>Page</u>
1. Binary division circuit	11
2. Systematic feed-in encoder	14
3. Feed-in decoder: error detection	15
4. Feed-out encoder	17
5. Feed-out decoder: error detection mode	19
6. ATCRBS interrogation modes (uplink)	21
7. Bench test results of code performance in ATCRBS interference	30

LIST OF ILLUSTRATIONS (cont.)

<u>Fig.</u>		<u>Page</u>
8.	Code performance in CW interference	32
9.	Probability of undetected message error (single - delay multipath interference)	34

LIST OF TABLES

<u>Table</u>		
1.	ATCRBS Interference Patterns	22
2.	<u>A Posteriori</u> Probability of Undetected Message Error	27

1. INTRODUCTION AND STATEMENT OF THE PROBLEM

The Discrete Address Beacon System (DABS) will provide an evolutionary upgrading of air traffic surveillance capability, together with an integrated ground/air data link. Both features are required to support the planned automation of air traffic control.

DABS includes a unique code as part of each interrogation which allows the ground sensor to address each aircraft individually so as to control the timing of replies from neighboring aircraft. This eliminates the self-interference caused by overlapping replies (synchronous garble), which is a basic limitation of the present Air Traffic Control Radar Beacon System (ATCRBS). By providing for the inclusion of a message as part of an interrogation or a reply, data link communications can be accommodated on the same channel with a small increase in equipment complexity.

The major factors influencing the design of the DABS signal formats are (1) the need to attain a reliability of service commensurate with the projected surveillance and communications demands of an automated ATC system, (2) electromagnetic compatibility with ATCRBS to allow both systems to operate together in an extended period of evolution from ATCRBS to DABS, and (3) minimization of transponder cost to speed the transition from DABS to ATCRBS. Simply stated, DABS must operate reliably in a severe interference environment without degrading other systems, and it must do this with low cost electronics in general aviation transponders.

In view of these factors, a study was initiated to determine the costs/benefits resulting from the incorporation of error control techniques into both the DABS uplink and downlink messages. This report summarizes the results of this coding investigation for the uplink. The downlink is treated in a separate report.

1-1. Statement of the Coding Problem

The impact of the three major issues of performance, plus the requirement for electromagnetic compatibility and low-cost avionics, are summarized here with respect to the coding problem.

One of the primary responsibilities of DABS will be the delivery of collision avoidance commands and other traffic control commands to aircraft. It is necessary that such command messages be validated before execution. Air Traffic Control commands are validated in the present system by repeating the command back to the ground. This same technique could be used with digital messages in DABS as a low cost method of message validation. However, if a command message could be validated in a single uplink transaction, message delivery would require fewer transmissions and would thus be less strongly affected by link reliability. Coding techniques offer just such means of reliably validating a single uplink transmission that has been correctly received in an aircraft, and such coding techniques need not involve a great amount of circuit complexity. Thus, coding techniques were studied from the outset of the DABS program as a promising means of providing a highly reliable and efficient message validation system with little cost impact on the transponder.

Use of the same uplink operating frequency as ATCRBS (1030 MHz) was attractive from the viewpoint of sharing sensor transmitter and transponder receiving circuitry with ATCRBS. However, this uplink frequency choice necessitated the careful study of the affects of DABS transmissions on ATCRBS transponders. Due to the peculiarities of ATCRBS transponders, it was determined that DABS transmissions would have to initially suppress an ATCRBS transponder and transmit all its data within approximately 30 μ sec (the nominal ATCRBS sidelobe suppression interval). This time constraint, together with the 1030-MHz channel window, implied the need for minimal overhead in the messages for addressing, parity checking and other control, (housekeeping) functions. Moreover, since each DABS transmission would suppress ATCRBS transponders, reliable DABS operations with a minimum number of transmissions (per message delivered) were required to avoid oversuppression of ATCRBS transponders.

The cost considerations of operating the DABS uplink on 1030 MHz and the resulting ATCRBS EMC constraints resulted in a total transmission length of 112 bits sent at a 4-MHz data rate, with 56 bits being devoted to ATC command message information and 24 bits for unique aircraft addressing. In order to eliminate the overhead associated with the redundant parity check bits in coding, an idea was borrowed from the British work on a discrete address system referred to as ADSEL (Address Selective beacon system), in which parity check bits were combined with the aircraft address bits. Instead of having a transponder check two separate message fields to determine if the received message should be displayed, a combined address-parity field allows this operation to be carried out by checking only one field. Whenever the parity check bits resulting from the received message are nonzero, the expected address-parity field is different from the actual received address-parity field, clearly indicating the message should not be displayed. This scheme removes the overhead associated with the use of coding for message validation, which is an important step because of the severe constraints on message length.

Although the foregoing discussion makes coding appear attractive for message validation, the key problem is that of selecting a code that performs adequately in the channel environment. There are numerous error mechanisms affecting the DABS uplink channel. A decision was made early in the coding investigation to restrict attention to errors caused by interference sources. Errors caused by noise only or caused solely by fading of the signal below threshold were not considered here. A rationale for this posture is that errors due to noise alone can be easily dealt with by proper choice of code based on interference considerations. Also, the fading mechanisms that arise from turning aircraft, over-the-horizon transmission, etc., have a duration that is longer than the DABS message and therefore are beyond the control of any coding scheme chosen.

Errors due to interference arise from ATCRBS interrogations, TACAN uplink channels operating near or harmonically related to 1030 MHz,

CW interference and multipath. Of these, ATCRBS interference is the dominant factor and the code search was largely driven by this fact. In particular, there are five ATCRBS interrogation modes (modes 1, 2, 3/A, C and D; each of which is composed of $P_1 - P_3$ pulses with an added P_2 SLS pulse starting $2 \mu\text{sec}$ after the leading edge of P_1 , see Fig. 6). Individual ATCRBS pulses are $0.8 \mu\text{sec}$ in duration as compared with the $0.25\text{-}\mu\text{sec}$ bit duration for DABS. Thus with DABS utilizing DPSK signaling, a single overlaying ATCRBS pulse can interfere with up to 5 DABS bits. Furthermore, all of the ATCRBS modes span $25 \mu\text{sec}$ or less. Thus, in a heavy (uncontrolled) ATCRBS interrogation environment, several pulse sets could overlap a DABS message resulting in a multiple burst error channel. It is this channel that is given the major emphasis in the forthcoming analysis.

Performance requirements and low-cost considerations for the DABS transponder are the major factors that resulted in the decision to have the error control technique provide an error detection-only capability. In particular, the DABS scheduling subsystem has sufficient capacity to accommodate an occasional retransmission because of a missed message; on the other hand, the acceptance of an erroneous or misdirected message must be strictly guarded against due to the dire consequences for the aircraft involved. Thus, a design goal was to allow a probability of undetected message error of no more than 10^{-7} . Furthermore, the desire for a low-cost transponder operating in real time essentially precluded the implementation of an error correction capability.

The remainder of this report focuses on the search for and the analysis of an optimal code for the DABS uplink using the above consideration as "initial conditions." In Section II the properties of binary cyclic codes (the code class of choice) are reviewed including shift register implementations for encoding and decoding. Examples are given using the specific code ultimately chosen. Section III presents the results of the performance analysis of the chosen code against various interference environments. This

code, which is easily implemented, is seen to possess an error detection capability that ensures a high degree of message reliability. In addition, the technique of overlaying the code parity bits on the address field of the message optimizes the use of overhead bits with small penalty in performance.

II. UPLINK CODE PROPERTIES AND IMPLEMENTATION

The effort to develop an error control capability for the DABS system was based primarily on the need to protect the uplink against the possibility of accepting a message containing errors and/or of accepting a message intended for another aircraft. Thus, the problem was to find an easily implemented code that would minimize the probability of undetected message error. Because of the need for simplicity, low overhead requirements, and the burst nature of the interference, preliminary analysis indicated that the study should concentrate on investigating the class of binary cyclic codes in preference to other techniques such as convolutional codes, finite geometry codes, etc. The relevant properties of these codes are summarized below.

II - 1 Code Properties

The encoding process for any binary block code takes a k -bit binary information vector $\underline{m} = (m_0, m_1, \dots, m_{k-1})$ and maps it into an n -bit code vector $\underline{c} = (c_0, c_1, \dots, c_{n-1})$. This results in an (n, k) code having $r = n - k$ parity bits; the information transmission efficiency is given by the code rate, $R = k/n$. The encoding rule is chosen to maximize the detection and/or correction of transmission errors by introducing a distance between each of the 2^k information sequences.

The Hamming weight of a codeword is the number of its nonzero components. The Hamming distance between two codewords is the number of components in which the two words differ. Since the n bit sequence of all zeros is always a codeword, the minimum distance d , of a binary linear code is the same as the minimum weight (nonzero) word in the code.

The error correction/detection capability is related to a code's minimum distance, and the encoding rule is chosen to maximize d for a given

n and k. In particular, a code having $d \geq 2e + 1$ can correct all occurrences of e or fewer errors in an n bit code word; alternately, the code can detect all occurrences of d - 1 or fewer errors. In order to correct e or fewer errors and simultaneously detect s (s > e) errors, the code must have $d \geq e + s + 1$.

If the demodulator has the capability of outputting erasures (wherein it is questionable whether or not a zero or a one was transmitted), then a distance -d code can correct up to (d - 1) erasures or t errors plus m erasures, where $d \geq 2t + m + 1$. The reason for this increased correction capability is the fact that the locations of the erasures are (presumably) known.

A binary cyclic code is a special case of a block code that can be described in terms of polynomials with binary coefficients such that all codewords, c(x), are multiples of a fixed generator polynomial, g(x), of degree r. Furthermore, this polynomial has a periodicity, $e = e(g(x); r)$ called the natural length of the code, which is given by the smallest integer such that g(x) divides $x^e + 1$. Thus,

$$g(x)h(x) = x^e + 1$$

where h(x) is of degree e-r. As an example, $g(x) = 1 + x^3 + x^6 + x^8$ (1) divides $x^{63} + 1$ and therefore there exists 2^{55} distinct multiples of g(x), modulo $x^{63} + 1$, which constitute the code space. The cyclic nature of these codes is illustrated by the fact that if c(x) is a codeword, then so is $x c(x)$, modulo $x^e + 1$, i. e., for

$$c(x) = m(x) g(x) = c_0 + c_1 x + \dots + c_{e-1} x^{e-1}$$

then

$$x c(x) = c_{e-1} + c_0 x + c_1 x^2 + \dots + c_{e-2} x^{e-1} = m'(x) g(x)$$

In order to broaden the class of codes to be considered, the techniques of "shortening" and "expanding" a cyclic code have been utilized. A cyclic (e, k) code can be shortened to an (N, K) code, $N = e - i$, $K = k - i$ which uses the same generator polynomial. While the shortening procedure destroys the true cyclic property of the base code (i.e., modulo $x^e + 1$), $xc(x)$ is still a codeword if the degree of the codeword, $c(x)$, is less than $e - i - 1$. More importantly, a shortened cyclic code has at least the same error correction/detection capability as the parent code and can be implemented with the same level of complexity.

Similarly, a cyclic (e, k) code with generator polynomial, $g(x)$, can be expanded to an (em, km) code with generator $g(x^m)$. The expansion technique enhances the burst error detection and correction capability of the base code that is summarized below.

In a burst noise environment, the requirements on the code and the decoding technique change substantially. A burst of errors of length b is defined as a sequence whose only nonzero components are among b consecutive bits, the first (and last) of which is nonzero.

In order to detect all single bursts of errors of length b or less, it is necessary and sufficient for the code to have b parity check bits. This condition alternately allows correction of a single burst of erasures of up to b bits.

On the other hand, to correct any single burst of b errors or to correct any two erasure bursts each of length b or less, the code must have at least $2b$ parity check bits. Furthermore, in terms of polynomials, no codeword can have the form

$$x^i b_1(x) + x^j b_2(x)$$

where the degree of $b_\ell(x)$ is less than b , for $\ell = 1, 2$. In other words, every codeword of a single b -burst correcting code must be comprised of at least three distinct burst segments, each of length b bits or less. Thus, it is seen that this is a generalization of a $d = 3$ single independent error correcting

code. In fact, if a cyclic (n, k) code, generated by $g(x)$ with a minimum distance of $d = 3$ is expanded to an (nb, kb) code with generator $g(x^b)$, a code is obtained that can correct any single error burst of length b bits or less.

II - 1.1. Properties of the Code Chosen for DABS

The cyclic code chosen for implementation is the 24-degree generator polynomial

$$\begin{aligned}
 g(x) &= 1 + x^3 + x^{10} + x^{12} + x^{13} + x^{14} + x^{15} + x^{16} + x^{17} + x^{18} \\
 &\quad + x^{19} + x^{20} + x^{21} + x^{22} + x^{23} + x^{24} \\
 &= (1 + x) (1 + x^2 + x^4 + x^5 + x^6) (1 + x + x^3 + x^4 + x^5 + x^6 \\
 &\quad + x^7 + x^8 + x^{10} + x^{13} + x^{15} + x^{16} + x^{17})
 \end{aligned}$$

which has a natural length of about 2.75 million bits ($e = 21(2^{17} - 1)$). This code has been shortened to a length of $n \sim 100$ bits for the DABS message format and to $n \sim 50$ bits for the DABS all-call mode.

This code was first discovered by Kasami (1964) and was determined to be able to correct any single 12-bit error burst or less in a code length of 72 bits (or it can correct any two erasure bursts each of 12 bits or less). Alternately, this code guarantees detection (correction) of any single 24-bit error (erasure) burst over any length.

Although the actual Hamming distance or general burst distance properties cannot be determined analytically, a fairly extensive exercising of this code over a word length of 100 bits indicates that

- . The Hamming distance is 6.
- . The 4-bit burst distance is 4 over a code length of at least 100 bits.
- . The 9-bit burst distance is at least 3 over 89 bits (in other words, any occurrence within 89 bits of two nine-bit error bursts is guaranteed detectable).
- . The 10-bit burst distance is 3 over a code length of 89 bits.

- . The 11-bit burst distance is 3 over a code length of 75 bits.
- . The 12-bit burst distance is 3 over a code length of 72 bits.

As will be seen in a later section, this code possesses a remarkably good multiple burst error detection capability in the ATCRBS interference environment hypothesized for DABS. In fact its performance is as good as any of the numerous codes tested having between 20 and 30 parity bits.

It should be noted that the reciprocal polynomial

$$g^*(x) = 1 + x + x^2 + \dots + x^{12} + x^{14} + x^{21} + x^{24}$$

has the same properties as the code given above. The reciprocal polynomial is defined from the original generator by

$$g^*(x) = x^r g(1/x)$$

and generates an equivalent code of the same natural length as $g(x)$.

II - 2. Encoding Methods; Address/Parity Overlay

In this section various shift register implementations that have been considered for the DABS encoding process for both uplink and downlink are given. In all cases, consideration has been limited to a "systematic" encoding wherein the information sequence is transmitted as it appears from the message source with r parity check bits appended to it.

When the number of parity bits is less than the number of information bits to be encoded ($r < k$), encoding is most easily performed by an r -stage linear feedback shift register. These shift registers are composed of storage devices (unit delay flip flops), modulo-2 adders, constant multipliers, and gates.

Starting with a k -bit information sequence $\underline{m} = (m_0, m_1, \dots, m_{k-1})$ with a polynomial representation of $m(x) = m_0 + m_1 x + \dots + m_{k-1} x^{k-1}$, it is desired to encode so that the resulting polynomial, $c(x)$, is a multiple of $g(x)$.

In both implementations to be considered, this operation is performed by use of a division circuit. To see this, $m(x)$ is first premultiplied by x^r to obtain

$$m_1(x) = x^r m(x) = (0, 0, \dots, 0, m_0, m_1, \dots, m_{k-1}) \quad (2)$$

a polynomial whose first r coefficients (low order) are zero. Upon dividing $x^r m(x)$ by $g(x)$, a remainder is obtained that can be represented as a polynomial, $R(x)$, of degree $< r$, i. e.,

$$x^r m(x) = Q(x) g(x) + R(x) \quad (3)$$

with $R(x) = p_0 + p_1 x + \dots + p_{r-1} x^{r-1} = (p_0, p_1, \dots, p_{r-1})$.

Thus

$$\begin{aligned} c(x) &= x^r m(x) - R(x) = Q(x) g(x) \\ &= (p_0, p_1, \dots, p_{r-1}, m_0, m_1, \dots, m_{k-1}) \end{aligned} \quad (4)$$

This polynomial is a multiple of $g(x)$, and furthermore, the information sequence is transmitted unaltered from its original form (since $p_i = 0$ or 1 we have $p_i \equiv -p_i$ in binary arithmetic).

II - 2.1 Feed-in Encoder/Decoder

Figure 1 shows a general feed-in division circuit for the polynomial, $g(x) = 1 + g_1 x + g_2 x^2 + \dots + g_{r-1} x^{r-1} + x^r$. The nomenclature, "feed-in," is used to describe the fact that the feedback connections are made by breaking into the chain of storage registers and inserting modulo-2 adders. Since $g_i = 0$ or 1, the multipliers simply indicate whether or not a connection exists.

The input polynomial, $m(x)$, to be divided enters high-order bits first into the low-order end of the register. The quotient is dynamically generated at the output and at any bit time the contents of the shift register

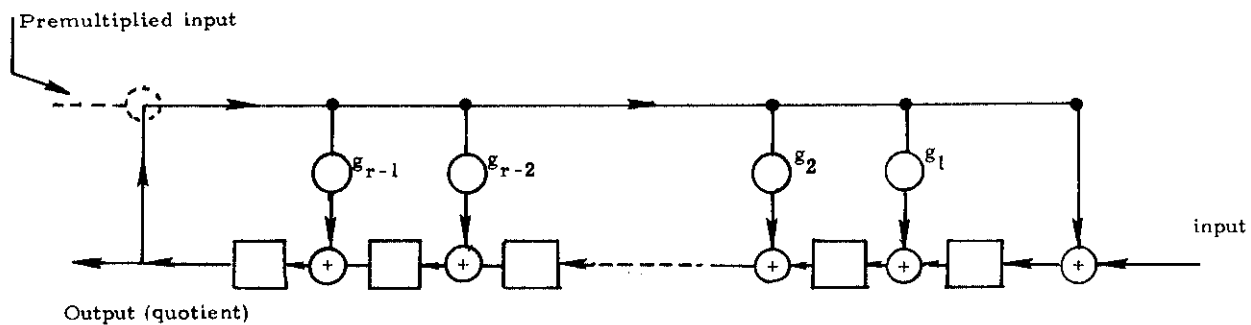


Fig. 1. Binary division circuit.

is a partial remainder. At the end of $n = k + r$ clock units, the register contains the sought-for remainder, $R(x)$. Note that there is an r -bit delay before an output starts to appear. By changing the input location to the high order side of the register as shown in Fig. 1 by dotted lines, the input is automatically premultiplied by x^r . This now means that after k input shifts, the contents of the register is the sought-for remainder which can be shifted out on line (without feedback) to satisfy Equation 4.

After demodulation at the transponder, the received message, $v(x)$, will be composed of the transmitted codeword, $c(x)$, embedded in an additive error sequence, $e(x)$; that is $v(x) = c(x) + e(x)$. The first step in any decoding procedure is to calculate the syndrome (remainder), $S(x) = (s_0, s_1, \dots, s_{r-1})$, by dividing $v(x)$ by the **generator** polynomial $g(x)$, i.e.,

$$S(x) \equiv v(x) \text{ modulo } g(x).$$

But since

$$v(x) = c(x) + e(x) = Q(x)g(x) + e(x)$$

Then

$$S(x) \equiv e(x) \text{ modulo } g(x). \tag{5}$$

Thus, the syndrome is a function only of the errors made in transmission. If the only capability required is the detection of errors, the process is complete since a nonzero syndrome indicates that one or more errors have occurred in transmission. Furthermore, a received message containing errors will go undetected if, and only if, the error sequence, $e(x)$, is a valid codeword; for then $v(x)$ will be a valid (but different) codeword yielding an all-zero remainder. If a correction capability is desired, additional circuitry is needed to manipulate the syndrome to correct independent or burst errors and/or erasures up to the capability of the code.

II - 2.1.1 Encoding

In Fig. 2, the above results are used to construct the encoding circuit for the 24-degree DABS polynomial including the implementation of the address/parity overlay. The message to be encoded is given as

$$M(x) = a(x) + x^r m(x) = (a_0, a_1, \dots, a_{r-1} \vdots m_0, m_1, \dots, m_{k-1}) \quad (6)$$

where $a(x)$ is the address of the aircraft to be interrogated, and the number of parity bits has been chosen to coincide with the dimension of the address field. With the contents of the register initially set to zero, the message is input high order bits first. During the first k shifts, gates G-1 and G-2 are open (enabled) while gate G-3 is closed (disabled). This allows the information to be input for division while simultaneously being outputted to the modulator. After the k information bits have been input, gates G-1 and G-2 are closed and G-3 is open which allows the remainder to be added without feedback to the address, bit-by-bit (modulo-2), and forwarded to the modulator.

II-2.1.2 Decoding for Error Detection

The decoding circuitry for an error detection-only capability is almost an exact replica of the encoding device shown in Fig. 2 with a slight variation to handle the address verification. Fig. 3 illustrates the process. The received sequence is fed into the shift register (initialized to zeros) and simultaneously into a storage buffer. During the first k shifts, gate G-1 is enabled and G-2 is disabled. After the last information bit is input (after the k th shift), gate G-1 is disabled and G-2 is enabled allowing the ADDRESS-CHECK buffer to be filled. If no errors have occurred, the transmitted address will appear there after the n th shift. If this address checks with that of the receiving aircraft, the stored information sequence is forwarded to the display unit.

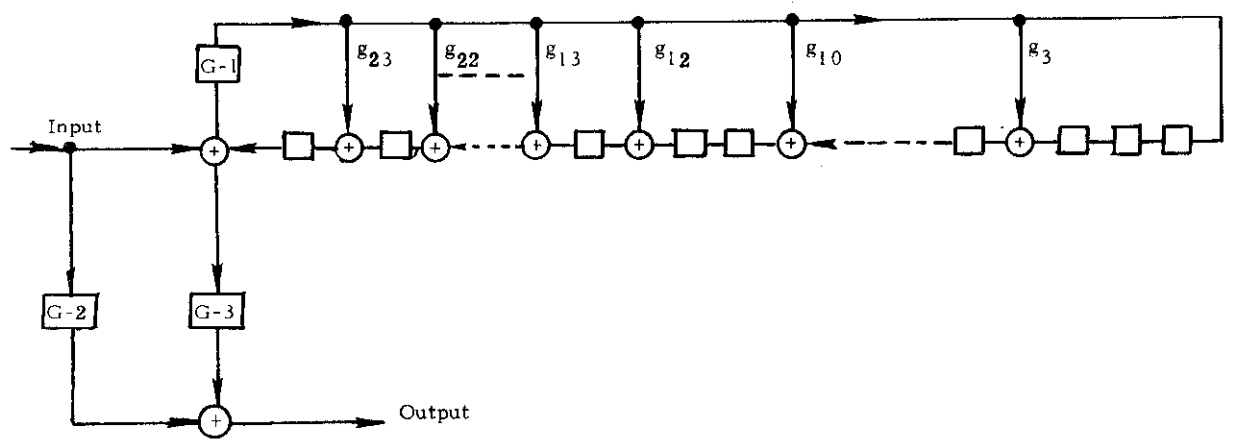


Fig. 2. Systematic feed-in encoder.

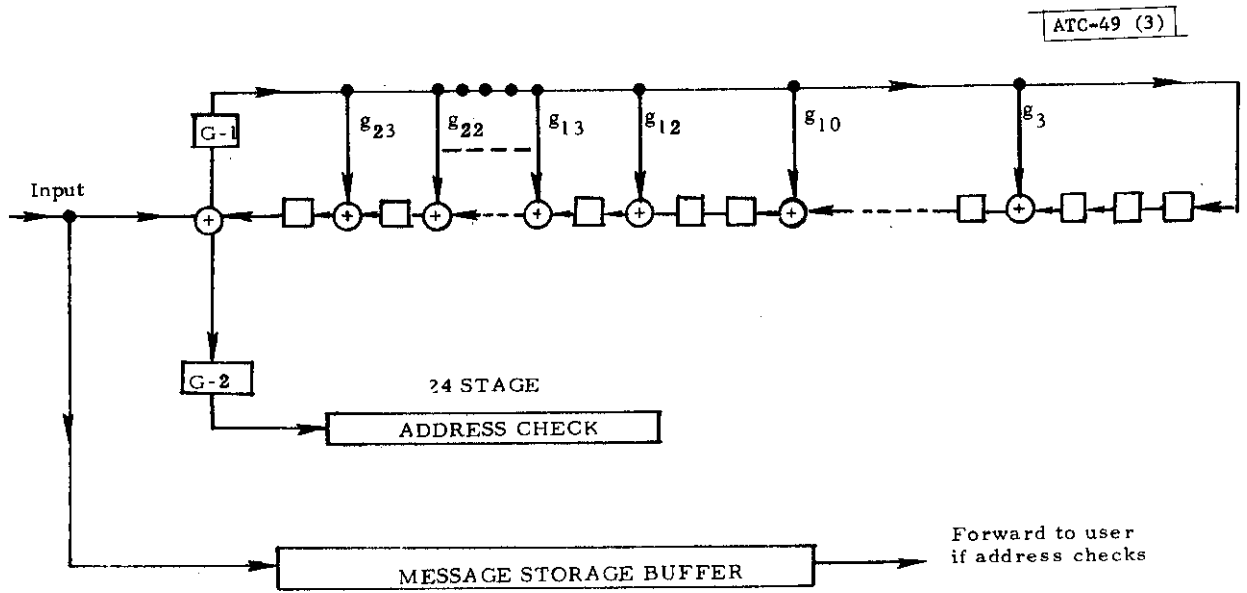


Fig. 3. Feed-in decoder: error detection.

II-2.2 Feed-out Encoding and Decoding

As seen above, the circuitry needed to effect an error detection capability is extremely simple both in concept and in hardware. The one possible drawback to the circuits detailed above is the fact that modulo-2 adders must be inserted between shift register elements. This may increase the manufacturing costs since there presently exist off-the-shelf chips that contain series connections of untapped storage devices (flip-flops). To circumvent this potential problem, an alternate implementation is given for the proposed encoding and decoding polynomial. Recall that the circuits given above can be described as division devices that dynamically generate a quotient (which is not used directly) and store a partial remainder within the register.

II-2.2.1 Encoder Implementation

Figure 4 illustrates the encoding process for the 24-degree polynomial proposed for the DABS system. In this case the message input to the device is given as

$$M'(x) = x^{n-r} a'(x) + m'(x) = (a'_{r-1}, \dots, a'_1, a'_0, m'_{k-1}, \dots, m'_1, m'_0) \quad (7)$$

where now the bits are input low order (subscript) first. (This should cause no confusion if we let $m'_i = m_{k-1-i}$, etc., where m_j are the coefficients used in the previous description given by Equation 6.) With gates G-1 and G-4 enabled and gates G-2 and G-3 disabled, the first k bits are input to the shift register and simultaneously output to the modulator for channel encoding. After the k th bit has been input, the contents of the register is a partial quotient which, if shifted r more times without feedback, will dynamically generate the appropriate parity bits (remainder). Therefore at this point gates G-1 and G-4 are disabled.

Up to this point the encoding has proceeded as in the previous implementation. Now, however, an alternate method of dealing with the address/parity overlay is given. In particular, after the k th information bit is shifted in, the shift register is switched from a division mode to a multiplication mode. This is accomplished by enabling gates G-2 and G-3

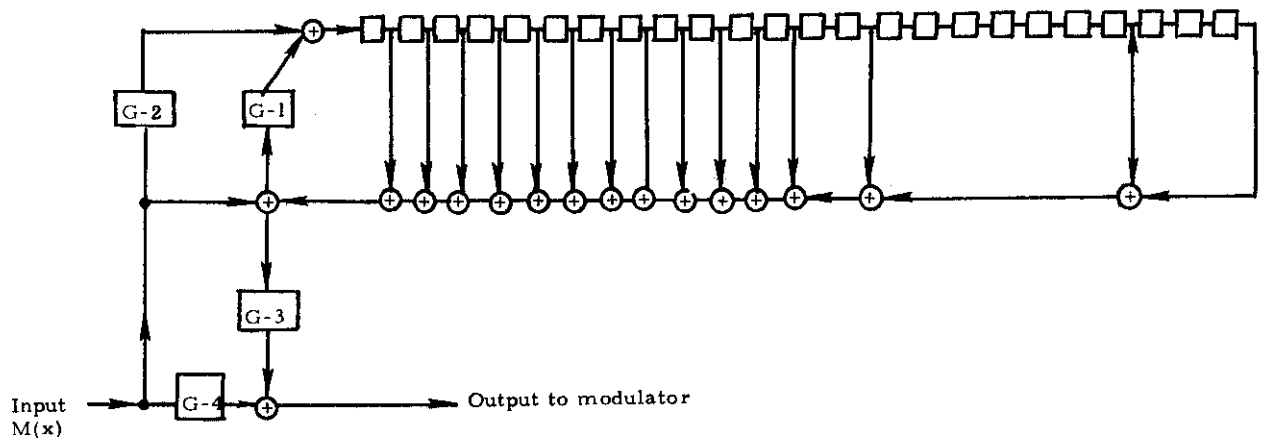


Fig. 4. Feed-out encoder.

(G-1 and G-4 are disabled). During the next r shifts, the circuit is effecting a partial multiplication of $a'(x)$ by $g(x)$ and modulo-2 adding the results to the remainder being generated from the contents of the register. Thus the transmitted codeword is given as

$$c'(x) = m'(x) + x^{n-r}R'(x) + x^{n-r} \{g(x) a'(x)\}_r \quad (8)$$

where $\{g(x) a'(x)\}_r$ indicates that the $(2r-1)$ degree polynomial, $g(x)a'(x)$, has been truncated so as to retain only the low-order r bits.

II-2.2.2 Decoding for Error Detection Only

Figure 5 shows the decoding circuit for the encoded sequence discussed above. The entire message is shifted into the division circuit as well as into a storage buffer. After all n bits have been input, the contents of the register will be the correct address if no errors have occurred in transmission.

Thus it is seen that by introducing at the transmitter the slight complication of switching from a division mode to a multiplication mode, the need for two gates and an extra ADDRESS CHECK register at the receiver has been eliminated. It should be noted that this same technique can be utilized with the feed-in implementations given in Figs. 2 and 3.

In summary, the implementations described above are quite simple, require no exotic hardware, and introduce essentially no time delay into the system. Although the techniques have been illustrated for a specific code, the concepts used are general and apply whether or not the cyclic code has been shortened.

III. CODE PERFORMANCE IN VARIOUS UPLINK INTERFERENCE ENVIRONMENTS

The error control technique considered for the DABS uplink was confined to an error detection capability only, largely because of the requirement for simple hardware operating in real time. The search for a code was driven by the requirement to achieve an undetected message error

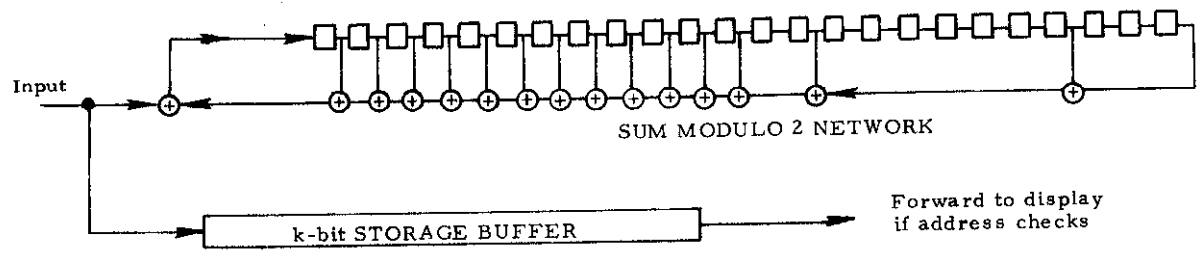


Fig. 5. Feed-out decoder: error detection mode.

probability of 10^{-7} or less in a heavy interference environment. The principle source of interference is due to the ATCRBS system which utilizes an uplink frequency of 1030 MHz and is presently in wide use. To a much smaller extent TACAN/DME beacons operating near or harmonically related to 1030 MHz could also contribute to the uplink interference problem (the nearest [existing] TACAN/DME uplink channel utilizes 1024 MHz). Other sources of interference are delay multipath and CW interference.

In the following paragraphs each of these interference sources is discussed and the performance of the selected code is determined by simulation and/or bench test. Because of its dominant effect, the emphasis is on code performance in heavy ATCRBS interference, and code selection was based on these results.

III - 1. ATCRBS Uplink Interference Environment

The potential interference patterns arising from ATCRBS interrogation pulse sets are shown in Fig. 6. Also considered are interference patterns consisting of a single P_2 pulse and also a $P_1 - P_2$ pulse pair. It has been seen from other studies that the P_2 suppression pulse has one of the largest a priori probabilities of occurrence, along with the $P_1 - P_2$ pair that arises from improved SLS.

The number of DABS bits affected by ATCRBS interrogations depends on the DABS signaling technique, bit rate and word length, as well as the amount of overlap. The individual ATCRBS pulse sets under consideration (due to their higher likelihood of occurrence), along with the DABS bits possibly affected for the case of 4 megabit/sec DPSK signaling, are listed in Table 1. The DABS word occupies bit positions (1, 2, ..., N), where $N \leq 120$ and each of the ATCRBS pulse sets are shown in their zero-shift positions. For example, the zeroth shift of $P_1 P_2$ is assumed to overlay bits (1 - 5, 9 - 13) of the DABS message and the i th shift occupies bit positions ($i + 1, i + 2, \dots, i + 5, i + 9, \dots, i + 12$) for i ranging from $-m+1 = -12$ to $N + m - 1 = N + 12$. The number of bit positions spanned by the ATCRBS pulse pattern is denoted by m . The interference patterns for pulse amplitude modulation (PAM) at a bit rate of 4 Mbps are shown for comparison.

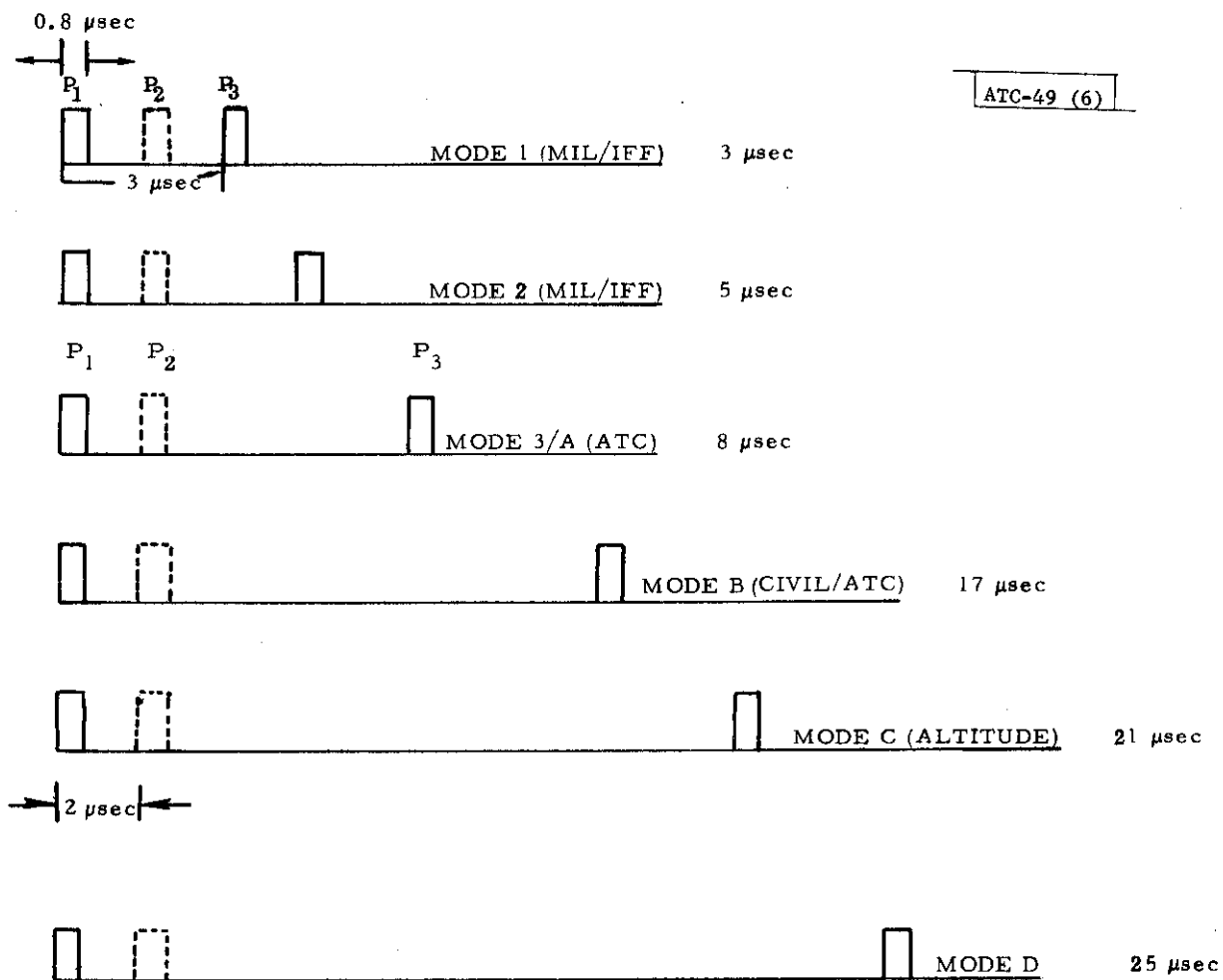
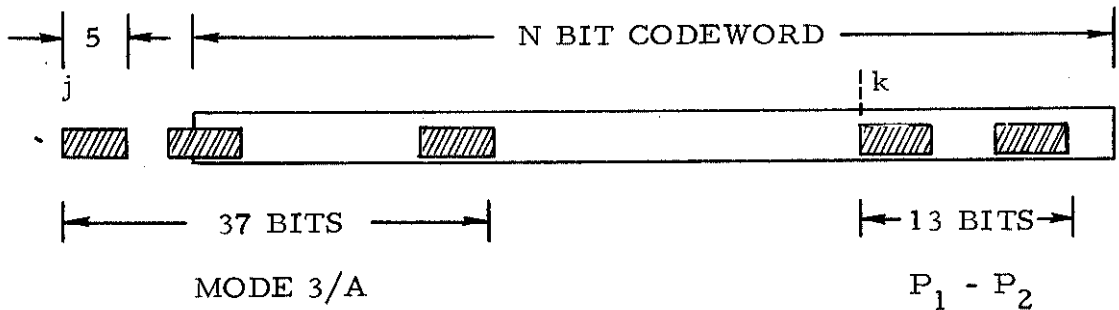


Fig. 6. ATCRBS interrogation modes (uplink)

TABLE 1
ATCRBS INTERFERENCE PATTERNS

MODE M	DPSK 4 Mbps	PAM 4 Mbps
P_2	1-5	1-4
$P_1 P_2$	1-5, 9-13	1-4, 9-12
$P_1 P_3$ Mode 2	1-5, 21-25	1-4, 21-24
3/A	1-5, 33-37	1-4, 33-36
C	1-5, 85-89	1-4, 85-88
$P_1 P_2 P_3$ Mode 2	1-5, 9-13, 21-25	1-4, 9-12, 21-24
3/A	1-5, 9-13, 33-37	1-4, 9-12, 33-36
C	1-5, 9-13, 85-89	1-4, 9-12, 85-88

DABS block occupies bit positions $\{1, 2, \dots, N\}$. Each left-justified interference pattern above can overlap a DABS word in any one of $N + m - 1$ bit positions, where m is the span of bits covered by the pattern (burst).



Each 0.8- μ sec ATCRBS P-pulse overlapping a DABS word is assumed to be able to adversely affect up to 5 consecutive bits with DPSK while only 4 bits can be affected using PAM. The reason for including an extra bit in the interference pulses for DPSK is that bit decisions are based on phase transitions between signaling intervals resulting in a one-bit "memory." However since the fifth bit will in practice be adversely affected only "some of the time" error statistics will be determined for both types of interference listed in Table 1.

The ATCRBS modes 1, B, and D and the military mode 4 have been determined to have a sufficiently low a priori probability of occurrence to be excluded from the test set.

In order to reduce the search for a code giving acceptable performance in this multiple burst noise environment, attention was restricted to the class of (shortened) cyclic codes having between 20 and 30 parity check bits. Moreover, the prospective codes were further restricted by requiring a minimum burst error detection capability that could be determined analytically. In particular, each of the following pulse sets overlapping the DABS message in any configuration had to be guaranteed detectable in order to admit a code for further consideration.

- (a). P_2 and P_2' pulses (spanning 5 bits each)
- (b). $P_1P_2P_3$, any 3-pulse ATCRBS mode
- (c). P_1P_2 and P_2' occurrences (when possible)

It should be noted that if 5-bit error pulses are considered, the P_1 - P_2 suppression pair could span a total of 13 DABS bits; in this case, to guarantee detection of any error set of type (c) requires a minimum of 26 parity bits. Also to satisfy case (b), a code having a 5-bit burst distance of at least 4 is needed, i. e., the nonzero bits of any codeword cannot be contained within three or fewer intervals each of 5 bits or less.

III-2 Determination of the Probability of Undetected Message Error by Simulation

Assuming a code is found with the above capability over a message length of N bits, the next most frequently occurring interference patterns to

test the code against are combinations of patterns taken two at a time from the set given in Table 1, i. e., for a given code, the probability of undetected message error (UDME) arising from modes M and M' overlapping a message in any (all) position is to be determined. This is denoted by

$$P_u(M, M') = \Pr \left[\begin{array}{l} \text{Modes M and M' overlap a DABS message} \\ \text{and cause an undetected error.} \end{array} \right] \quad (9)$$

Since this probability must be analyzed by computer, it is more appropriate to determine the quantity

$$P(\text{UDME}/M, M') = P_u(M, M')/P(M, M'),$$

where $P(M, M') = P(M) P(M')$

is the a priori probability of occurrence of two independent ATCRBS interference patterns. The sought-for probability is determined by first finding the probability of an undetected word error when M and M' occur at fixed positions j and k, respectively, within the DABS message, where $-m + 1 \leq j \leq N + m - 1$ and $m' + 1 \leq k \leq N + m' - 1$; and m, m' are the lengths of the patterns in bits. Thus

$$P_u(M @ j, M' @ k) = \Pr[M @ j, M' @ k] \Pr[\text{UDME} | M @ j, M' @ k] \quad (10)$$

But

$$\Pr[M @ j] = P(M) \Pr[M @ j | M] = \frac{1}{N + m - 1} P(M) \quad (11)$$

with

$$P(M) = \sum_{j=-m+1}^{n-1} \Pr[M \text{ occurs at position } j] \quad (12)$$

In other words, it is assumed that M can overlap the DABS message in any of $N + m - 1$ positions with equal probability, given that it occurs at all.

Combining the above

$$P_u(M, M') = \frac{P(M) P(M')}{(N + m - 1)(N + m' - 1)} \sum_j \sum_k \Pr[\text{UDME} \mid M @ j, M' @ k] \quad (13)$$

with the sought-for quantity given as

$$P\{\text{UDME} \mid M, M'\} = \frac{1}{(N + m - 1)(N + m' - 1)} \sum_j \sum_k \Pr[\text{UDME} \mid M @ j, M' @ k] \quad (14)$$

It remains to determine the quantities $\Pr[\text{UDME} \mid M @ j, M' @ k]$ and sum them appropriately. For each pair of locations j, k of M and M' , the simulation determines the number of codewords that have nonzero bits confined totally within the bit positions overlapped by M and M' .^{*} Once this is done it is necessary to impose a probability distribution on the occurrence of bit errors within each sub-burst of the overlapping interference patterns. In lieu of an analytical model, which is extremely difficult to pin down, the rather harsh assumption is made that individual bit errors occur within an ATCRBS sub-burst (a 5-bit P pulse) with probability 0.5. This results in

$$\Pr[\text{UDME} \mid M @ j, M' @ k] = (2^{\nu_{jk}} - 1)/2^{\delta_{jk}} \quad (15)$$

where $2^{\nu_{jk}} - 1$ is the number of codewords (excluding all-zero word) contained within the interference patterns and $2^{\delta_{jk}}$ is the total number of sequences possible. (In the vernacular of vector spaces, ν_{jk} is the nullity and δ_{jk} is the dimension of the subspace defined by the overlapped bits.)

* The computer forms a submatrix from the columns of the codes parity check matrix corresponding to the overlapped positions and determines the nullity of this submatrix.

Each of the codes tested had a codeword length on the order of 100 bits, which required approximately 10000 calculations of the form in Eq. 15 for each pair of interference modes taken from Table 1. Each run required about 2 minutes of computer time.

III - 2.1 Code Performance in Simulated ATCRBS Interference

Results of the determination of the a posteriori probability, $P\{\text{UDME} \mid M, M'\}$, are given in Table 2 for the (96, 72)* shortened cyclic code having generator polynomial

$$g(x) = 1 + x^3 + x^{10} + x^{12} + x^{13} + \dots + x^{23} + x^{24}.$$

All 36 sets of ATCRBS interference patterns from Table 1, taken two at a time with repetition, were tested for two cases: (1) each P-pulse within a pattern spans 5 DABS bits, and (2) each P-pulse within a pattern spans 4 DABS bits. As mentioned previously, these cases apply to 4 Mbps DPSK and 4 Mbps PAM signaling, respectively. However, since the fifth bit of a burst is less likely to be in error than the other 4 bits in the DPSK format, the actual a posteriori error probability can be taken as some suitably chosen weighted average of the results in each case.

As seen in Table 2, all but two of the entries are less than 10^{-6} , and after multiplying by the appropriate a priori probability, $P(M, M')$, each result is reduced further by at least two orders of magnitude. Thus in all cases the probability of undetected message error is less than 10^{-7} .

* The message length of 112 bits (and 56 bits) had not been decided on until after the simulation results had been obtained. The error in the results presented below in going from $n = 96$ to $n = 112$ bits is negligible.

TABLE 2

A POSTERIORI PROBABILITY OF UNDETECTED MESSAGE ERROR

Mode Pairs M, M'		Pr(UDME M, M') 5-bit burst	Pr(UDME M, M') 4-bit burst
P ₂	- P ₂	0	0
	P ₁ P ₂	1.17 x 10 ⁻⁶	0
	P ₁ P ₃ Mode 2	0	0
	P ₁ P ₃ Mode 3/A	0	0
	P ₁ P ₃ Mode C	0	0
	P ₁ P ₂ P ₃ Mode 2	1.85 x 10 ⁻⁷	0
	P ₁ P ₂ P ₃ Mode 3/A	4.49 x 10 ⁻⁷	0
	P ₁ P ₂ P ₃ Mode C	6.78 x 10 ⁻⁷	0
	P ₁ P ₂	P ₁ P ₂	5.91 x 10 ⁻⁷
P ₁ P ₃ Mode 2		5.48 x 10 ⁻⁷	0
P ₁ P ₃ Mode 3/A		5.74 x 10 ⁻⁷	0
P ₁ P ₃ Mode C		1.06 x 10 ⁻⁶	0
P ₁ P ₂ P ₃ Mode 2		3.08 x 10 ⁻⁷	3.65 x 10 ⁻⁹
P ₁ P ₂ P ₃ Mode 3/A		6.49 x 10 ⁻⁷	4.25 x 10 ⁻⁷
P ₁ P ₂ P ₃ Mode C		8.11 x 10 ⁻⁷	4.72 x 10 ⁻¹⁰
P ₁ P ₃ Mode 2	P ₁ P ₃ Mode 2	5.34 x 10 ⁻⁸	0
	P ₁ P ₃ Mode 3/A	1.67 x 10 ⁻⁸	0
	P ₁ P ₃ Mode C	2.33 x 10 ⁻⁹	0
	P ₁ P ₂ P ₃ Mode 2	8.53 x 10 ⁻⁸	1.59 x 10 ⁻⁸
	P ₁ P ₂ P ₃ Mode 3/A	1.82 x 10 ⁻⁷	8.81 x 10 ⁻⁹
	P ₁ P ₂ P ₃ Mode C	3.27 x 10 ⁻⁷	0
P ₁ P ₃ Mode 3/A	P ₁ P ₃ Mode 3/A	1.57 x 10 ⁻⁸	0
	P ₁ P ₃ Mode C	8.56 x 10 ⁻⁹	1.4 x 10 ⁻⁸
	P ₁ P ₂ P ₃ Mode 2	1.19 x 10 ⁻⁷	1.96 x 10 ⁻⁹
	P ₁ P ₂ P ₃ Mode 3/A	2.08 x 10 ⁻⁷	2.32 x 10 ⁻⁹
	P ₁ P ₂ P ₃ Mode C	3.21 x 10 ⁻⁷	1.06 x 10 ⁻⁸
P ₁ P ₃ Mode C	P ₁ P ₃ Mode C	3.78 x 10 ⁻⁹	0
	P ₁ P ₂ P ₃ Mode 2	1.98 x 10 ⁻⁷	7.89 x 10 ⁻¹⁰
	P ₁ P ₂ P ₃ Mode 3/A	4.36 x 10 ⁻⁷	2.11 x 10 ⁻⁹
	P ₁ P ₂ P ₃ Mode C	6.51 x 10 ⁻⁷	1.03 x 10 ⁻⁸
P ₁ P ₂ P ₃ Mode 2	P ₁ P ₂ P ₃ Mode 2	6.08 x 10 ⁻⁸	2.0 x 10 ⁻⁸
	P ₁ P ₂ P ₃ Mode 3/A	7.64 x 10 ⁻⁸	5.79 x 10 ⁻⁸
	P ₁ P ₂ P ₃ Mode C	2.80 x 10 ⁻⁷	6.06 x 10 ⁻⁹
P ₁ P ₂ P ₃ Mode 3/A	P ₁ P ₂ P ₃ Mode 3/A	3.0 x 10 ⁻⁷	1.20 x 10 ⁻⁷
	P ₁ P ₂ P ₃ Mode C	5.64 x 10 ⁻⁷	2.34 x 10 ⁻⁷
	P ₁ P ₂ P ₃ Mode C	7.22 x 10 ⁻⁷	3.87 x 10 ⁻⁹

The above code was one of a number of codes tested against these interference patterns and except for an $r = 26$ parity bit code, $(g(x) = 1 + x^2 + x^3 + x^5 + x^7 + x^9 + x^{11} + x^{13} + x^{16} + x^{17} + x^{20} + x^{22} + x^{23} + x^{24} + x^{26})$, it yielded the best overall performance.

The results given in the previous subparagraph made no mention of the effects of overlaying the parity bits on the 24-bit address field. Because of superposition, the vast majority of errors generated in the uplink transmission will cause the address to change so as to be unrecognized by the target aircraft. The undetected errors, as determined from the above analysis, will leave the address unchanged and therefore cause the target aircraft to accept an erroneous message.

There is another undetected error probability which is associated with a misdirected message. Given two aircraft in the main beam of an interrogator, it is possible that a message sequence intended for one of the aircraft will be corrupted sufficiently by noise so that the second aircraft decodes the message and recognizes its own address. Because of the 2^{24} possible number of addresses, this a posteriori event will occur approximately 1 time in $\sim 6 \times 10^8 = 2^{24}$ tries.

III-2.2 Code Performance in a TACAN/DME Interference Environment

A single TACAN/DME reply pulse, R , has a duration of $3.5 \pm 0.5 \mu\text{sec}$ and therefore spans between 16 and 19 bits in the DABS uplink message format. Furthermore, in the X mode, TACAN pulses occur in pairs with a separation of $\sim 12 \mu\text{sec}$ (leading edge-to-leading edge). Thus, in order to obtain a complete assessment of the capability of the code, the undetected message error is determined given that an X mode TACAN pulse pair overlaps the DABS data block with sufficient power to cause bit errors.

The TACAN $R_1 R_2$ pulse pair is assumed to span 1 - 17 and 49 - 65 in a zero-shift position overlaid on a 96-bit DABS data block. The overall undetected message error probability is obtained by summing the contribution for each of 160 overlapping shifts of the pulse pair. For completeness, similar results were obtained using a 15-bit and also a 19-bit overlap per TACAN pulse. All three conditional probabilities are

summarized below.

$$\begin{aligned} P \{ \text{UDME} \mid R_1 R_2; 15\text{-bit overlap per pulse} \} &= 1.82 \times 10^{-8} \\ P \{ \text{UDME} \mid R_1 R_2; 17\text{-bit overlap per pulse} \} &= 1.89 \times 10^{-8} \\ P \{ \text{UDME} \mid R_1 R_2; 19\text{-bit overlap per pulse} \} &= 2.14 \times 10^{-8} \end{aligned} \quad (16)$$

The overall probability of undetected message error is obtained by multiplying the above results by the a priori probability of an overlap, $P\{R_1 R_2\}$, which is a function of the TACAN interference environment. For a nominal TACAN environment of 3600 pp/sec, $P\{R_1 R_2\} < 0.15$.

An analysis of the code against all possible overlaps of two TACAN pulse pairs was also simulated. This event could occur in a heavy TACAN interference environment. Using a 17-bit overlap per TACAN pulse, a conditional error probability is obtained as

$$P\{\text{UDME} \mid R_1 R_2, R'_1 R'_2\} = 6.37 \times 10^{-8}$$

III-3. Experimental Measurement of Code Performance

Once the code was chosen based on the simulation results of the previous paragraph, this same code was tested experimentally to further corroborate its performance in heavy ATCRBS interference. In addition the code was bench tested in the presence of one or two CW interference sources and also in a delay multipath environment.

III.3.1 Verification of Performance in ATCRBS Interference

For the purposes of verifying performance in ATCRBS interference, a test bench was assembled that provided a means of generating coded DABS messages garbled by two independent ATCRBS "modes." The 24-bit code was used and the resulting binary message modulated according to the 4 Mbit/sec DPSK uplink format. Results of these tests are shown in Fig. 7 for the ATCRBS mode pairs indicated in each of the three graphs. Each point in a graph represents the results of 10^8 trials; the overlapped DABS message positions due to each of the two ATCRBS modes were

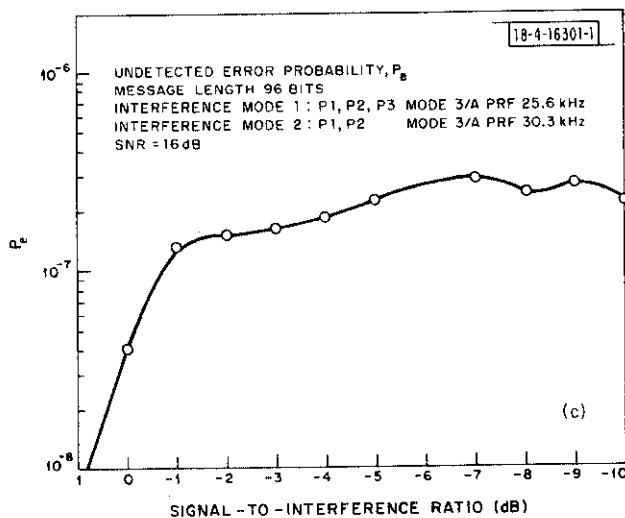
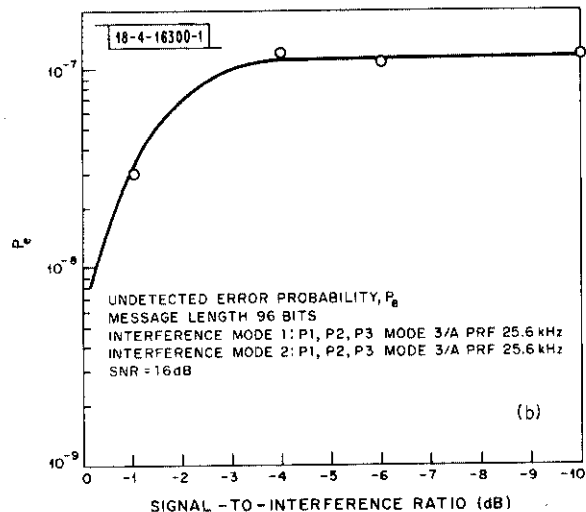
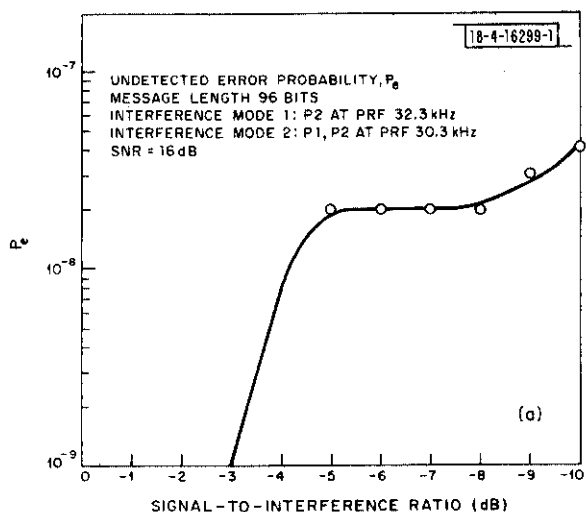


Fig. 7. Bench test results of code performance in ATRBS interference.

determined pseudorandomly at each trial. In each case presented, the DABS message length was 96 bits. Similar results were obtained for a 48 bit encoded DABS message.

In comparing these a posteriori results with the appropriate simulation results from Table 2, it is seen that the simulation predictions are higher (more conservative). The discrepancy, in large part, is due to the conservative assumption that each overlapped bit has a probability of 0.5 of being in error, whereas in practice (i. e., experimentally), the average bit error for DPSK should be much less even for poor signal-to-interference ratios. Another possible reason for the differences between the simulated and experimental results is the limited number of trials (10^8) used to obtain the data points. Using 10^8 trials to obtain error probabilities in the range of 10^{-8} to 10^{-7} could introduce a substantial variance in the results.

III-3.2 Performance in the Presence of CW Interference

The encoded and modulated DABS message was also tested in the presence of one and two CW sources of interference. The interference generated operated at a frequency close to 1030 MHz and overlapped on the entire message with a random relative phase. The results for one and two CW sources are shown in Figs. 8 (a) and (b), respectively.

III-3.3 Performance in a Delay Multipath Environment

Bench tests were also conducted to measure the susceptibility of the DABS uplink coding to multipath interference. Multipath interference was simulated by generating a delayed replica of an encoded 96-bit message (at logic level) and appropriately phase modulating two independent, but phase-stable, carriers with the original and delayed messages. The two modulated carriers were then combined prior to demodulation by a DPSK receiver. The demodulated bit sequence was then fed into an uplink decoder (error detector). Control of the relative power level of the carriers was maintained to within a fraction of a dB. The amount of delay between the message replica and the original could be controlled to within a fraction of the 0.25 μ -sec chip width.

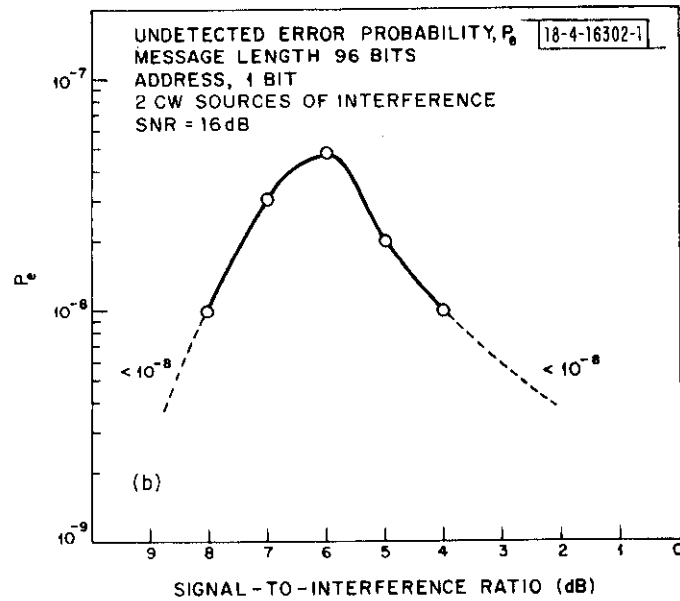
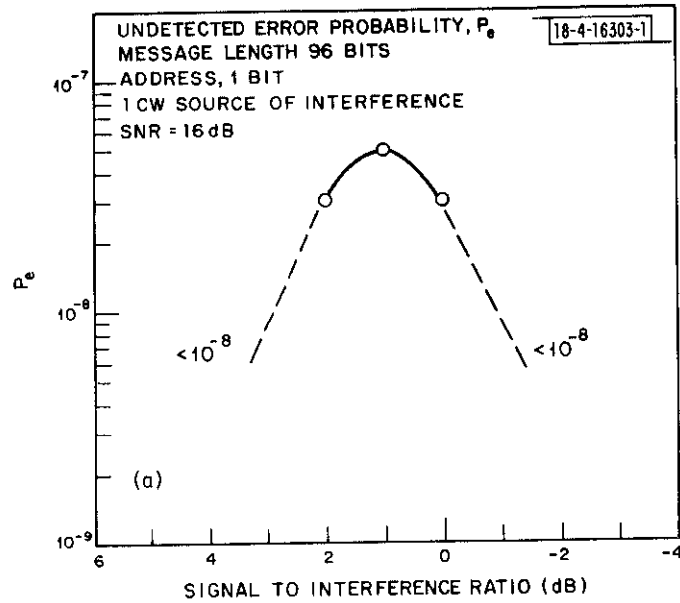


Fig. 8. Code performance in CW interference.

The demodulated message was also compared directly to the original message to determine if any bit errors occurred. The occurrence of an undetected message was automatically recorded whenever the DABS uplink decoder failed to detect that the message had been demodulated erroneously. By repeating the procedure a large number of times, the probability of an undetected message error was determined.

Results of the experiments, shown in Fig. 9, indicate that the undetected message error probability is less than 10^{-7} for any particular multipath environment. The interference-to-signal ratio is defined as the power ratio of the two carriers, with the understanding that the interference is the carrier of the delayed message. The signal-to-interference ratio was varied from -4 to +6 dB, and multipath delays up to 30 nsec were simulated. In all cases the signal-to-noise ratio was 16 dB, and the encoded message consisted of 96 bits.

III-4 Analytic Approximations to Code Performance

For completeness the performance of the chosen code in more severe interference environments than were used in the foregoing discussions must be considered. However, a computer analysis for interference composed of three or more ATCRBS patterns taken from Table 1 would take at least 100 times as long as the results per run given in Table 2. Moreover, the a priori probability of the multiple occurrences of these patterns becomes quite small. Thus, in lieu of a practical alternative, two rough approximations to the a posteriori undetected message error probability are now given.

Let $B(x)$ of degree b describe a "super burst" which encompasses the whole span of DABS bits possibly affected by a multiple set of ATCRBS modes. An undetected message error will occur if

$$B(x) = Q(x) g(x)$$

where $Q(x)$ is of exact degree $b - r$ and $q_0 = q_{b-1-r} = 1$. The $b - r - 2$ interior coefficients of $Q(x)$ can take on all values 0 or 1 which results in a total of 2^{b-r-2} undetectable b -bit bursts starting in a given bit position. Since there are a total of 2^{b-2} b -bit bursts, the fraction of them that are codewords

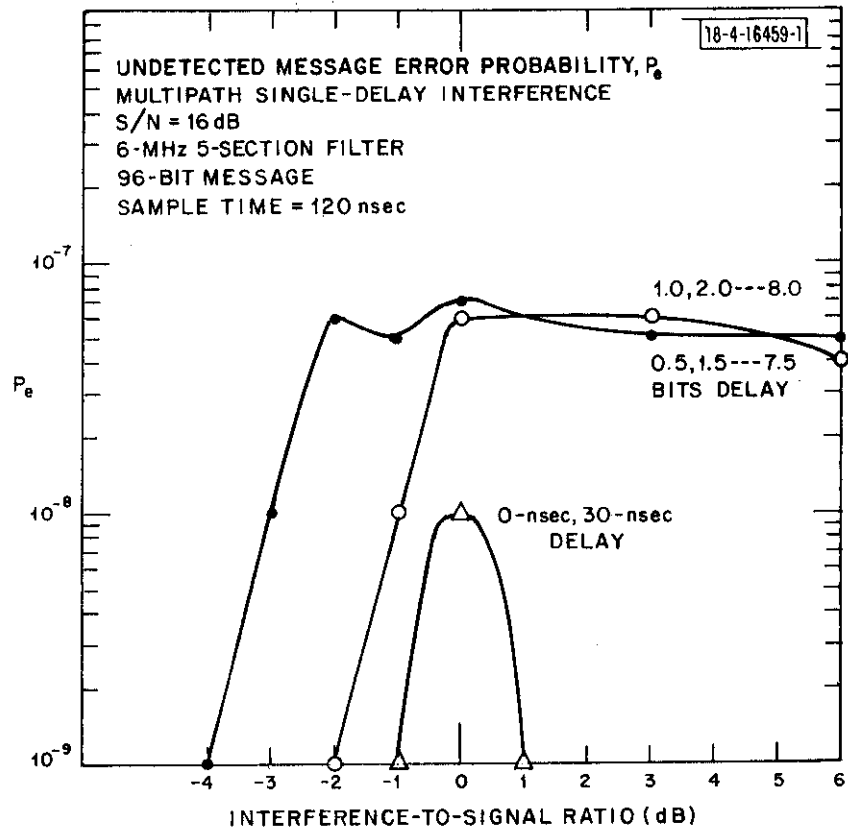


Fig. 9. Probability of undetected message error (single - delay multipath interference).

is 2^{-r} . This can be considered an overbound since it does not account for the distribution of errors within the sets of ATCRBS patterns making up this superburst.

Another approximation can be obtained by considering the independent error detection capability of the code using the Binary Symmetric Channel. The undetected word error probability is given as

$$P_u = \sum_{i=d}^n A_i p^i (1-p)^{n-i}$$

where d = the minimum distance of the code, p is the bit error probability (independent and constant from bit to bit), and A_i is the number of codewords of weight i in the code, with $\sum_{i=d}^n A_i = 2^k - 1$ = the number of nonzero codewords. Setting $p = 0.5$, the result is

$$P_u = 2^{-n} \sum_{i=d}^n A_i = 2^{-n} (2^k - 1) \doteq 2^{-r}$$

where $r = n - k$. Because this approximation assumes independent errors (with a high probability of occurrence) when in fact the channel errors are of a burst nature, this again can be construed as an overbound to the probability of undetected message error.

IV. SUMMARY

The 24-bit shortened cyclic code chosen for the DABS uplink message is seen to perform quite satisfactorily in a severe interference environment. In fact, of all the candidate codes originally tested, the chosen code exhibited the best overall performance. In every case analyzed the unconditional probability of undetected message error is better than 10^{-7} and in most cases is less than 10^{-8} . Moreover, the proposed address/parity overlay technique affords maximal utilization of the limited number of bits in the DABS message format with only a small sacrifice in misdirected message probability.

In addition, this same code is proposed for use in the DABS downlink, thus allowing considerable simplification in transponder hardware. In another report it will be shown that this code has properties that are well suited to the downlink interference environment. Its use in a burst erasure correction technique has the potential for substantially improving the downlink reliability and throughput.

REFERENCES

- [1] T. Kasami, "Optimum Shortened Cyclic Codes for Burst-Error Correction," *IEEE Transactions on Information Theory*, Vol. IT-9, pp. 105-109 (April 1963).
- [2] T. Kasami and S. Matoba, "Some Efficient Shortened Cyclic Codes for Burst-Error Correction," *IEEE Transactions on Information Theory*, Vol. IT-10, pp. 252-253 (July 1964).
- [3] W. W. Peterson, "Error Correcting Codes," the MIT Press, Cambridge, Massachusetts (1961).

ACKNOWLEDGMENT

The author gratefully acknowledges the contributions made by A. Smith pertaining to the bench test evaluation of the error detection capability. Also the numerous discussions with T. J. Goblick and J. R. Johnson are greatly appreciated. Special thanks go to L. Balboni for outstanding programming support.