C. L. Chen
R. A. Rutledge

# Error Correcting Codes for Satellite Communication Channels

**Abstract:** This paper addresses the problem of efficient forward error correction on differentially encoded, quadriphase-shift-keying (DQPSK) channels. The approach is to design codes to correct the most probable error patterns. First the probability distribution of error patterns is derived. Then a class of convolutional codes that correct any single two-bit error is described. Finally a threshold decodable code that corrects all single, and many double, two-bit errors is presented.

## Introduction

A satellite communication system for digital data transmission must make efficient use of the available bandwidth and power. This can be accomplished by combining forward error correction with an efficient modulation technique. The greater efficiency of phase-shift keying (PSK), as opposed to frequency-shift keying, often leads to the choice of PSK as a modulation technique for satellite channels [1, 2]. The implementation of a PSK modem depends on the constraints of the system. The number of phase states and the type of encoding (direct or differential) can be chosen to effect the desired tradeoff among power, bandwidth, transmission rate, and bit-error probability. In a bandwidth-limited system, quadriphase-shift keying (QPSK) can be used instead of binary phase-shift keying to conserve bandwidth [1-3]. The use of differential encoding to resolve the phase ambiguity at the receiver can save power that would otherwise be required for a residual carrier [4]. For these reasons, differentially encoded QPSK (DQPSK) is a highly efficient modulation technique for satellite communication channels. Fig. 1(a) shows a block diagram of a communication system using DQPSK and forward error correction.

A disadvantage of DQPSK modulation is that each failure in recognizing a phase position results in a two-bit error. A QPSK modem transmits a symbol, or bit pair, by phase-modulating the carrier to some phase $\psi$ chosen from the set {0°, 90°, 180°, 270°}. In the presence of white Gaussian noise, the most likely transmission error is a phase shift of $\pm 90°$ in the estimate of $\psi$. If the symbols are encoded using a Gray code, a 90° phase error in $\psi$ is a single-bit error in the bit pair represented by $\psi$ [5]. When differential encoding is used, information is encoded in the difference between successive phases, $\psi_{i+1} - \psi_i$. If a 90° phase shift, or single-bit error, occurs during the transmission of $\psi_i$, the output of the differential decoder will contain two single-bit errors: one in the estimate of $\psi_i - \psi_{i-1}$ and one in the estimate of $\psi_{i+1} - \psi_i$. Thus, the bit error rate is doubled and bit errors are correlated. The correlation of errors is the more serious problem because it severely degrades the efficiency of a random-error correcting code. If, for example, a single-error correcting convolutional code is used for forward error correction, there is no guarantee that it will correct any double-bit errors.

Forney and Bower [6] encountered this problem in designing a high speed sequential decoder. Their solution was to perform forward error correction before differential decoding [see Fig. 1(b)]. In this scheme, the error correction code (ECC) decoder need not contend with double-bit errors, but it is faced with the same phase ambiguity that the differential encoding was used to resolve. Techniques to resolve this ambiguity include an acquisition search at start-up and whenever the modem undergoes a 90° phase slippage; much of the benefit of differential encoding is lost.

This paper describes a different approach to the problem. Standard $t$-error correcting codes are designed to correct the error patterns most likely to appear in a binary symmetric channel. Instead of using these codes, it seemed advisable to look for new codes that correct the error patterns most likely to appear in a differentially encoded coherent QPSK system. To correct single-channel errors, for example, we looked for a code that corrects the double-bit errors produced by the differential decoder. Because they are highly unlikely, this code need not correct single-bit errors. To find such a code, it is first necessary to characterize the probability distribution

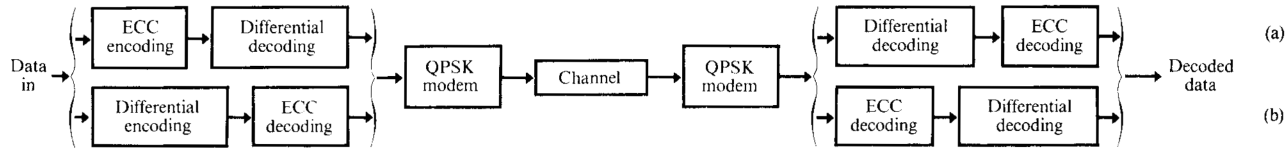



**Figure 1** Differentially encoded QPSK system with forward error correction and differential coding (a) internal to and (b) external to the error control coding.

of the error patterns. This is done in the following section. Next, a class of convolutional codes, which can correct any single-channel error, is presented. Then another convolutional code, which corrects all single-channel errors and 70 percent of double-channel errors, is described.

### Error characterization

A QPSK modem transmits a pair of bits by phase-modulating the carrier to 0°, 90°, 180°, or 270°. The receiving modem estimates the phase and translates it back into a bit pair. Thus, to transmit the bit pair $(I, Q)$, the modem transmits the phase

$$\theta = f(I, Q), \quad (1)$$

where $f(\cdot, \cdot)$ is a 1:1 mapping of $\{(0, 0), (0, 1), (1, 0), (1, 1)\}$ onto $\{0°, 90°, 180°, 270°\}$. The receiving modem computes the maximum likelihood estimate $\hat{\theta}$ of $\theta$ and

$$(\hat{I}, \hat{Q}) = f^{-1}(\hat{\theta}). \quad (2)$$

The channel noise is assumed to be white and Gaussian. If the receiving modem has exact knowledge of the carrier reference phase, it can be shown [4] that the error in $\hat{\theta}$,

$$\phi = \theta - \hat{\theta}, \quad (3)$$

has the probability distribution

$$\Pr(\phi = 0°) = (1-p)^2;$$

$$\Pr(\phi = 90°) = \Pr(\phi = 270°) = p(1-p);$$

$$\Pr(\phi = 180°) = p^2, \quad (4)$$

where $p = 0.5 \operatorname{erfc}(\sqrt{0.5R})$ and $R$ is the signal-to-noise ratio.

The mapping $f(\cdot, \cdot)$ is chosen to satisfy

$$f^{-1}(\theta) \oplus f^{-1}(\theta + 180°) = (1, 1) \quad (5)$$

for all $\theta$, where $\oplus$ denotes the EXCLUSIVE OR function applied element-by-element. Then it follows from (4) and (5) that the bit errors $\hat{I} \oplus I$ and $\hat{Q} \oplus Q$ are independent random variables with distributions

$$\Pr(\hat{I} \oplus I = 1) = 1 - \Pr(\hat{I} \oplus I = 0) = p, \text{ and}$$

$$\Pr(\hat{Q} \oplus Q = 1) = 1 - \Pr(\hat{Q} \oplus Q = 0) = p. \quad (6)$$

In other words, the channel, including the two QPSK modems, is simply a binary symmetric channel with crossover probability $p$.

In general, however, the receiving modem does not know the carrier reference phase exactly but can estimate it to within $\alpha$ degrees, where $\alpha$ is fixed but unknown. In this case, $\hat{\theta} = \theta + \alpha + \phi$ and, in general, $\hat{\theta} \neq \theta$ even when $\phi = 0$. Differential encoding is used to resolve this problem. The data sequence $\{(I_i, Q_i), i = 1, 2, \cdots\}$ is encoded into $\{\theta_i\}$, where $\theta_i = f(I_i, Q_i)$ as before, but instead of transmitting $\theta_i$, the modem transmits the sequence $\{\psi_i\}$ of partial sums defined by

$$\psi_0 = 0;$$

$$\psi_i = \psi_{i-1} + \theta_i, \; i = 1, 2, \cdots. \quad (7)$$

The receiving modem estimates $\psi_i$ by

$$\hat{\psi}_i = \psi_i + \alpha + \phi_i, \quad (8)$$

where the random errors $\phi_i$ are independent of each other and of the input sequence $\psi_i$, and each $\phi_i$ has the distribution shown in (4). Then $\theta_i$ is estimated by

$$\hat{\theta}_i = \hat{\psi}_i - \hat{\psi}_{i-1}, \quad (9)$$

and $(I_i, Q_i)$ is estimated by

$$(\hat{I}_i, \hat{Q}_i) = f^{-1}(\hat{\theta}_i). \quad (10)$$

If the errors in the decoded sequence are denoted by

$$(E_i, F_i) = (\hat{I}_i, \hat{Q}_i) \oplus (I_i, Q_i), \; i = 1, 2, \cdots, \quad (11)$$

it follows from (8–11) that

$$(E_i, F_i) = f^{-1}(\theta_i) \oplus f^{-1}(\theta_i + \phi_i - \phi_{i-1}). \quad (12)$$

It is convenient to rewrite (12) as

$$(E_i, F_i) = (U_i, V_i) \oplus (X_i, Y_i), \quad (13)$$

where

$$(U_i, V_i) = f^{-1}(\theta_i - \phi_{i-1}) \oplus f^{-1}(\theta_i),$$

$$(X_i, Y_i) = f^{-1}(\theta_i - \phi_{i-1} + \phi_i) \oplus f^{-1}(\theta_i - \phi_{i-1}). \quad (14)$$

Making use of (5), one can show that

$$(U_i, V_i) = \begin{cases} (X_{i-1}, Y_{i-1}) & \text{if } C_{i-1} \oplus X_i \oplus Y_i = 1; \\ (Y_{i-1}, X_{i-1}) & \text{if } C_{i-1} \oplus X_i \oplus Y_i = 0, \end{cases} \quad (15)$$

**Table 1** Possible error patterns when $(X_i, Y_i)$ contains a single-bit error and all other symbols are error free.

| $\hat{C}_i$ | $E_i$ | $F_i$ | $E_{i+1}$ | $F_{i+1}$ |
|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 1 |
| | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| | 0 | 1 | 0 | 1 |

where $C_{i-1} = I_{i-1} \oplus Q_{i-1} \oplus I_i \oplus Q_i$. Therefore, the sequence $\{(E_i, F_i)\}$ is completely determined by $\{(X_i, Y_i)\}$ and $\{(I_i, Q_i)\}$.

It follows from (4), (5), (14), and the independence of the $\phi_i$ that $X_i$ and $Y_i$, $i = 1, 2, \cdots$, are all independent random variables and each is equal to 1 with probability $p$ and is 0 otherwise. Combining all these results and simplifying, we obtain

$$(E_i, F_i) = (X_i, Y_i) \oplus (X_{i-1}, Y_{i-1})\, C_{i-1} \oplus (Y_{i-1}, X_{i-1})\, \overline{C}_{i-1} \oplus (W_{i-1}, W_{i-1}), \quad (16)$$

where

$$W_{i-1} = (X_{i-1} \oplus Y_{i-1})(X_i \oplus Y_i) \quad (17)$$

and $X_i$, $Y_i$, $i = 1, 2, \cdots$, are all independent, identically distributed random variables with distributions

$$\Pr(X_i = 1) = 1 - \Pr(X_i = 0) = p;$$

$$\Pr(Y_i = 1) = 1 - \Pr(Y_i = 0) = p. \quad (18)$$

This completely specifies the probability distribution of the sequence $\{(E_i, F_i)\}$.

For decoding purposes, it is preferable to express $(E_i, F_i)$ in terms of $(\hat{I}_i, \hat{Q}_i)$, since this sequence is observable by the decoder. If $\hat{C}_{i-1}$ is defined by

$$\hat{C}_{i-1} = \hat{I}_{i-1} \oplus \hat{Q}_{i-1} \oplus \hat{I}_i \oplus \hat{Q}_i, \quad (19)$$

then it can be shown that $\hat{C}_{i-1} = C_{i-1}$ whenever $X_{i-1} \neq Y_{i-1}$, so that (16) is equivalent to

$$(E_i, F_i) = (X_i, Y_i) \oplus (X_{i-1}, Y_{i-1})\, \hat{C}_{i-1} \oplus (Y_{i-1}, X_{i-1})\, \overline{\hat{C}}_{i-1} \oplus (W_{i-1}, W_{i-1}). \quad (20)$$

Now consider classes of most likely error sequences $\{(E_i, F_i)\}$. For a fixed-input sequence $\{(I_i, Q_i)\}$, Eq. (16) defines a 1:1 mapping between $\{(E_i, F_i)\}$ and $\{(X_i, Y_i)\}$. Therefore, the probability of any $\{(E_i, F_i)\}$ sequence is equal to the probability of the $\{(X_i, Y_i)\}$ sequence that maps into it, and the most likely $\{(E_i, F_i)\}$ sequences can be characterized in terms of the most likely $\{(X_i, Y_i)\}$ sequences. The $\{(X_i, Y_i)\}$ sequence is distributed like the error sequence of a binary symmetric channel. An appropriate class of $\{(X_i, Y_i)\}$ sequences is, therefore, the set of all sequences containing $n$ or fewer 1's for any choice of $n$. A sequence $\{(E_i, F_i)\}$ may be said to contain $k$ channel errors if it maps into an $\{(X_i, Y_i)\}$ sequence containing exactly $k$ 1's. Then an appropriate class of most likely $\{(E_i, F_i)\}$ sequences is the set of all sequences containing $n$ or fewer channel errors. It is easy to describe these patterns in detail for $n = 1$ and $n = 2$.

- *Single-channel error patterns*

If a single-channel error occurs in $(X_i, Y_i)$ and all other bit pairs $(X_j, Y_j)$ are error free, it is easy to see from (20) that

$$(E_j, F_j) = (0, 0) \quad \text{if } j \neq i \text{ or } i + 1;$$

$$(E_i, F_i) = (X_i, Y_i);$$

$$(E_{i+1}, F_{i+1}) = \begin{cases} (X_i, Y_i) & \text{if } \hat{C}_i = 1; \\ (Y_i, X_i) & \text{if } \hat{C}_i = 0. \end{cases} \quad (21)$$

Then, since $(X_i, Y_i)$ can take the value (0, 1) or (1, 0), there are four possible patterns for $(E_i, F_i, E_{i+1}, F_{i+1})$. These are shown in Table 1.

- *Double-channel error patterns*

Assume that the sequence $(X_i, Y_i)$, $i = 1, 2, \cdots$, contains exactly two 1's. There are three cases, depending on whether the two errors are in the same bit pair $(X_i, Y_i)$, in adjacent bit pairs $(X_i, Y_i)$ and $(X_{i+1}, Y_{i+1})$, or in non-adjacent bit pairs.

*Case 1* If $(X_i, Y_i) = (1, 1)$ and $(X_j, Y_j) = (0, 0)$ for $j \neq i$, it is easy to see from (20) that the only possible error pattern is

$$(E_i, F_i, E_{i+1}, F_{i+1}) = (1, 1, 1, 1),$$

$$(E_j, F_j) = (0, 0) \text{ for } j \neq i \text{ or } i + 1. \quad (22)$$

*Case 2* If $(X_i, Y_i)$ and $(X_{i+1}, Y_{i+1})$ each contain a single error, and $(X_j, Y_j) = (0, 0)$ for $j \neq i$ or $i + 1$, then repeated use of (20) yields

$$(E_j, F_j) = (0, 0) \text{ for } j \neq i \text{ or } i + 1 \text{ or } i + 2,$$

$$(E_i, F_i) = (X_i, Y_i),$$

$$E_{i+1} = F_{i+1} = E_i \oplus F_{i+2} \oplus \hat{C}_i \oplus \hat{C}_{i+1},$$

$$(E_{i+2}, F_{i+2}) = (X_{i+1}, Y_{i+1})\hat{C}_{i+1} \oplus (Y_{i+1}, X_{i+1})\overline{\hat{C}}_{i+1}. \quad (23)$$

From (23), it is clear that four error patterns are possible if $C_i \oplus C_{i+1} = 1$, and four other patterns are possible if $C_i \oplus C_{i+1} = 0$. These patterns are shown in Table 2.

*Case 3* If $(X_i, Y_i)$ and $(X_j, Y_j)$ each contain a single error, all other bit pairs are error free, and $|i - j| > 1$, then the only 1's in the sequence $\{(E_i, F_i)\}$ are in the nonoverlapping sequences $(E_i, F_i, E_{i+1}, F_{i+1})$ and $(E_j, F_j, E_{j+1}, F_{j+1})$. The possible error patterns can be determined independently for each sequence, as in the case of a single error.

## A class of single-channel error correcting convolutional codes

Given any positive integer $m$, it is possible to construct a single-error correcting convolutional code with decoding constraint length $mn$, where $n = 2^{m-1}$, and rate $(n-1)/n$. A class of codes satisfying these conditions was introduced, and shown to be optimal, by Wyner and Ash [7]. The codes are based on the truncated parity check matrix

$$H = [D, TD, T^2D, \cdots, T^{m-1}D], \tag{24}$$

where the columns of $D$ are the $n$ distinct binary $m$-tuples having first elements equal to one, and $T$ is the $m \times m$ shift matrix defined by

$$T_{i,j} = \begin{cases} 1 \text{ if } i-j=1, \\ 0 \text{ otherwise.} \end{cases} \tag{25}$$

The ordering of the columns of $D$ is arbitrary. For reasons that will become clear later, the $j$th column of $D$ is set equal to the binary representation of $2^m - j$. Thus, for $m = 4$,

$$D = \begin{bmatrix} 1&1&1&1&1&1&1&1 \\ 1&1&1&1&0&0&0&0 \\ 1&1&0&0&1&1&0&0 \\ 1&0&1&0&1&0&1&0 \end{bmatrix}. \tag{26}$$

This code corrects a single-bit error in any $n$-bit block if the $m-1$ preceding and following blocks are error free.

On a differentially encoded QPSK channel, this code is unsuitable. It miscorrects all single-channel errors (because they appear as double-bit errors) and almost all multiple-channel errors. It is easy to restore the single-channel error correcting capability of $H$ by degree-two interleaving. This technique, however, doubles the constraint length, requiring more hardware and increasing the probability that two or more channel errors occur within the constraint length. Interleaving is unnecessary because there exists a class of convolutional codes, with constraint length $mn$ and rate $(n-1)/n$, which corrects single-channel errors in a differentially encoded, coherent, QPSK system.

The truncated parity check matrix for a code in this class is

$$H' = [DV, TDV, T^2DV, \cdots, T^{m-1}DV], \tag{27}$$

where $D$ and $T$ are defined above and $V$ is the $n \times n$ matrix

**Table 2** Possible error patterns when $(X_i, Y_i)$ and $(X_{i+1}, Y_{i+1})$ each contains a single-bit error and all other symbols are error free.

| $\hat{C}_i \oplus \hat{C}_{i+1}$ | $E_i$ | $F_i$ | $E_{i+1}$ | $F_{i+1}$ | $E_{i+2}$ | $F_{i+2}$ |
|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| | 1 | 0 | 0 | 0 | 0 | 1 |
| | 0 | 1 | 0 | 0 | 1 | 0 |
| | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| | 1 | 0 | 1 | 1 | 0 | 1 |
| | 0 | 1 | 1 | 1 | 1 | 0 |
| | 0 | 1 | 0 | 0 | 0 | 1 |

$$V = \begin{bmatrix} I & O & O & \cdots & O \\ I & I & O & \cdots & O \\ I & I & I & \cdots & O \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ I & I & I & \cdots & I \end{bmatrix}, \tag{28}$$

with

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \text{ and } O = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}. \tag{29}$$

If $m = 4$, then

$$DV = \begin{bmatrix} 0&0&1&1&0&0&1&1 \\ 0&0&1&1&0&0&0&0 \\ 0&0&1&1&1&1&0&0 \\ 0&0&1&0&0&0&1&0 \end{bmatrix}. \tag{30}$$

The matrix $H'$ is used to encode the data in $n$-bit blocks before differential encoding and to decode the data after differential decoding [see Fig. 1(a)]. If the $k$th ECC encoded block is denoted by

$$w_k = (I_{kb+1}, Q_{kb+1}, \cdots, I_{kb+b}, Q_{kb+b}), \tag{31}$$

where $b = n/2$, then $Q_{kb+b}$ is a parity check bit chosen to satisfy

$$(w_{k-m+1}, w_{k-m+2}, \cdots, w_k) \times h' = 0, \tag{32}$$

where $h'$ is the last row of $H'$, and $\times$ denotes inner product modulo 2. The $k$th block received by the ECC decoder,

$$\hat{w}_k = (\hat{I}_{kb+1}, \hat{Q}_{kb+1}, \cdots, \hat{I}_{kb+b}, \hat{Q}_{kb+b}), \tag{33}$$

and the previous $m-1$ received blocks are used to determine the $k$th syndrome bit:

$$s_k = (\hat{w}_{k-m+1}, \hat{w}_{k-m+2}, \cdots, \hat{w}_k) \times h'. \tag{34}$$

If the error pattern in $\hat{w}_k$ is denoted by

$$e_k = \hat{w}_k \oplus w_k = (E_{kb+1}, F_{kb+1}, \cdots, E_{kb+b}, F_{kb+b}), \quad (35)$$

then it is clear from (32) and (34) that

$$s_k = (e_{k-m+1}, e_{k-m+2}, \cdots, e_k) \times h'. \quad (36)$$

To explain the decoding algorithm, and to prove that the code corrects all single-channel errors, we first examine the syndromes corresponding to the single-channel error patterns shown in Table 1. We assume that the $k$th block received by the differential decoder, $(X_{kb+1}, Y_{kb+1}, \cdots, X_{kb+b}, Y_{kb+b})$, contains a single-bit error, and the $m-1$ preceding and following blocks are error free. It is easy to see from Table 1 that the possible values of $e_k$ are the rows of the $n \times n$ matrices $M_0$ and $M_1$ defined by

$$M_0 = \begin{bmatrix} J & I & O & \cdots & O \\ O & I & J & \cdots & O \\ O & O & J & \cdots & O \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ O & O & O & \cdots & I \end{bmatrix}, \quad M_1 = \begin{bmatrix} I & I & O & \cdots & O \\ O & I & I & \cdots & O \\ O & O & I & \cdots & O \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ O & O & O & \cdots & I \end{bmatrix}, \quad (37)$$

where $I$ and $O$ are given by (29) and

$$J = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \quad (38)$$

When $m = 4$, (37) becomes

$$M_0 = \begin{bmatrix} 0&1&1&0&0&0&0&0 \\ 1&0&0&1&0&0&0&0 \\ 0&0&1&0&0&1&0&0 \\ 0&0&0&1&1&0&0&0 \\ 0&0&0&0&0&1&1&0 \\ 0&0&0&0&1&0&0&1 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1 \end{bmatrix},$$

$$M_1 = \begin{bmatrix} 1&0&1&0&0&0&0&0 \\ 0&1&0&1&0&0&0&0 \\ 0&0&1&0&1&0&0&0 \\ 0&0&0&1&0&1&0&0 \\ 0&0&0&0&1&0&1&0 \\ 0&0&0&0&0&1&0&1 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1 \end{bmatrix}. \quad (39)$$

The matrix $M_0$ contains all error patterns for which $\hat{C}_i$ is equal to zero, and $M_1$ contains all patterns for which $\hat{C}_i$ is equal to one. Error patterns $e_{k-m+1}$ and $e_{k+1}$ may have non-zero elements due to overlapping of an error in the previous block, but only in the first two positions. It is easy to see that

$$e_{k-m+2} = e_{k-m+3} = \cdots = e_{k-1} = e_{k+2} = \cdots = e_{k+m-1} = (0, 0, \cdots, 0). \quad (40)$$



Decoding of the $k$th block is based on syndrome bits $s_k, s_{k+1}, \cdots, s_{k+m-1}$, which can be represented as a column vector:

$$z = (s_k, s_{k+1}, \cdots, s_{k+m-1})^t. \quad (41)$$

When we compute $z$ using (36), all block error patterns other than $e_k$ drop out. This includes $e_{k-m+1}$ and $e_{k+1}$ because the first two columns of $DV$ are equal to $(0, 0, \cdots, 0)^t$. Therefore,

$$z = DVe_k^t, \quad (42)$$

and the possible values of $z$ are therefore the columns of $S_0$ and $S_1$, where $S_0$ and $S_1$ are $m \times n$ matrices defined by

$$S_0 = DVM_0^t \text{ and } S_1 = DVM_1^t. \quad (43)$$

The ordering of the columns of $D$ was chosen in such a way that the sum, modulo 2, of columns $j+1$, $j+2$, $j+3$, and $j+4$ is equal to $(0, 0, \cdots, 0)^t$, for $j = 0, 4, 8, \cdots, n-4$. The matrix $V$ was chosen as

$$V = [M_1^t]^{-1}. \quad (44)$$

Using these facts, we can reduce (43) to

$$S_0 = S_1 = D. \quad (45)$$

The first step in decoding is to find the unique integer $r$ such that the $r$th column of $D$ is equal to $z$. Then, from (43) and (45), $e_k$ is equal to the $r$th row of either $M_0$ or $M_1$. In either case, the channel error must have occurred in symbol $(X_i, Y_i)$, where $i$ is given by

$$i = kb + [(1 + r)/2] \quad (46)$$

and $[x]$ denotes the largest integer less than or equal to $x$. Since $i$ is known, $\hat{C}_i$ can be computed using (19). Then

$$e_k = r\text{th row of } \begin{cases} M_0 \text{ if } \hat{C}_i = 0; \\ M_1 \text{ if } \hat{C}_i = 1. \end{cases} \quad (47)$$

Finally, it is easy to see that

$$e_{k+1} = \begin{cases} (0, 0, 0, \cdots, 0) \text{ if } r < n-1; \\ (1, 0, 0, \cdots, 0) \text{ if } (r, C_i) = (n-1, 1) \text{ or } (n, 0); \\ (0, 1, 0, \cdots, 0) \text{ if } (r, C_i) = (n-1, 0) \text{ or } (n, 1). \end{cases} \quad (48)$$

Obviously, this procedure leads to correct decoding of any block containing a single-channel error, as long as the $m-1$ preceding and following blocks are error free. Then the probability that an uncorrectable error occurs in a block, given that the $m-1$ preceding blocks are error free, can be approximated by [8, 9]

$$P_B = \left[ \binom{n}{2} + n^2 (m-1) \right] p^2, \quad (49)$$

where

$$p = 0.5\,\mathrm{erfc}(\sqrt{0.5R}) \quad (50)$$

is the channel error probability. The probability that an information bit is decoded incorrectly, given that the previous $mn$ bits were decoded correctly, can be approximated by

$$P_b = P_B/(n-1). \quad (51)$$

The bit error probability $P_b$, for the case $m = 4$, is plotted as a function of $R$ in Fig. 2.

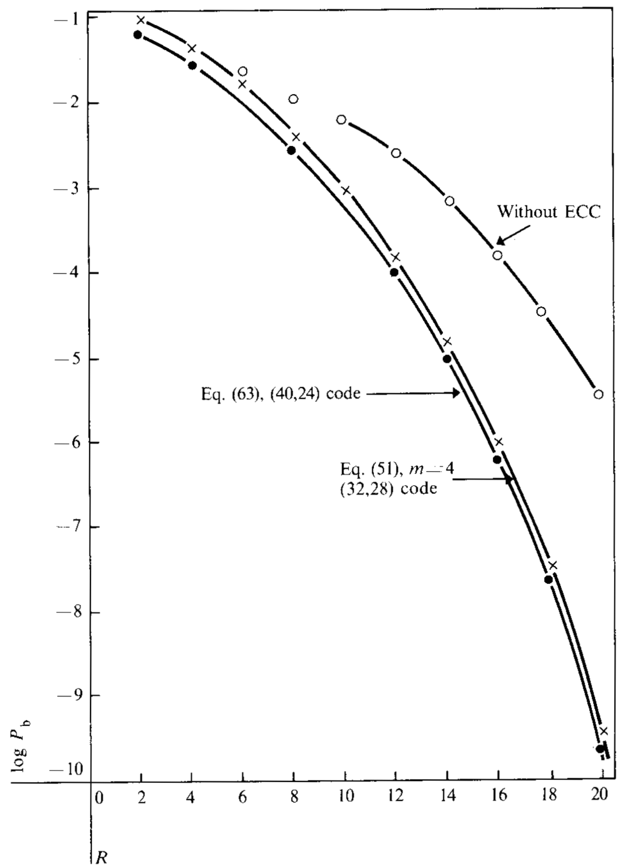



**Figure 2** Bit error probability as a function of the signal-to-noise ratio.

### A rate 3/5 threshold decodable code

The data conversion from a switched connection service of 19.2 kilobits per second (kbs) transmission rate to the network channel rate of 32 kbs allows 40 percent redundancy, which can be used to design error correcting codes for data protection. A rate 3/5 (40, 24) threshold decodable convolutional code for data rate conversion and error protection is described in this section. This code was obtained by trial and error because the methods of the previous section could not be successfully applied. The (40, 24) code designed for DQPSK is capable of correcting any single-symbol error ($\phi_i$ in the *Error characterization* section) in a sequence of 40 bits. It also corrects about 70 percent of the double-channel errors in 40 bits.

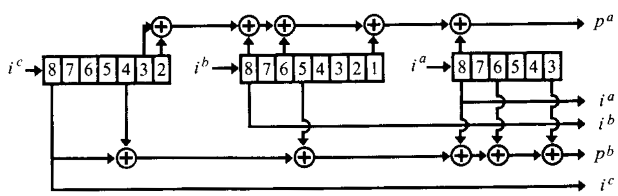
- *Encoder*

Figure 3 shows the encoder. It consists mainly of shift registers and EXCLUSIVE OR gates. Two parity check bits are generated for every three input information bits according to the following equations:

$$p_8^a = i_8^a \oplus i_8^b \oplus i_6^b \oplus i_1^b \oplus i_3^c \oplus i_2^c,$$
$$p_8^b = i_8^a \oplus i_6^a \oplus i_3^a \oplus i_5^b \oplus i_8^c \oplus i_4^c, \quad (52)$$

where the additions are EXCLUSIVE OR operations; $i_k^a$, $i_k^b$, $i_k^c$ are the three information bits at time $k$; and $p_k^a$, $p_k^b$ are the check bits generated at time $k$.

**Figure 3** Encoder for the (40, 24) code showing encoded bits in the order $\cdots i_2^c p_2^b i_2^b i_2^a p_2^a i_1^c p_1^b i_1^b i_1^a p_1^a$; $\oplus$ represents the EXCLUSIVE OR operation.



- *Decoder*

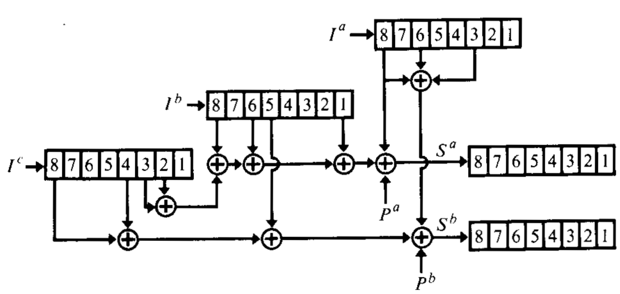
Decoding is done in two steps. In the first step, two sets of syndrome bits are calculated and stored. In the second step, error bits are estimated and the actual error correction is made.

Figure 4 shows the syndrome bit calculation for the decoder. Let $p'^a_k$, $i'^a_k$, $i'^b_k$, $p'^b_k$, and $i'^c_k$ be a block of received bits at time $k$. The syndrome bits are calculated according to (52) as follows:

$$S_8^a = p'^a_8 \oplus i'^a_8 \oplus i'^b_8 \oplus i'^b_6 \oplus i'^b_1 \oplus i'^c_3 \oplus i'^c_2,$$
$$S_8^b = p'^b_8 \oplus i'^a_8 \oplus i'^a_6 \oplus i'^a_3 \oplus i'^b_5 \oplus i'^c_8 \oplus i'^c_4. \quad (53)$$

Let $e_k^a$, $e_k^b$, $e_k^c$ be the error bits corresponding to information bits $a$, $b$, $c$, respectively, at time $k$. Then,

**Figure 4** Decoder for the (40, 24) code, showing the syndrome bit calculation.



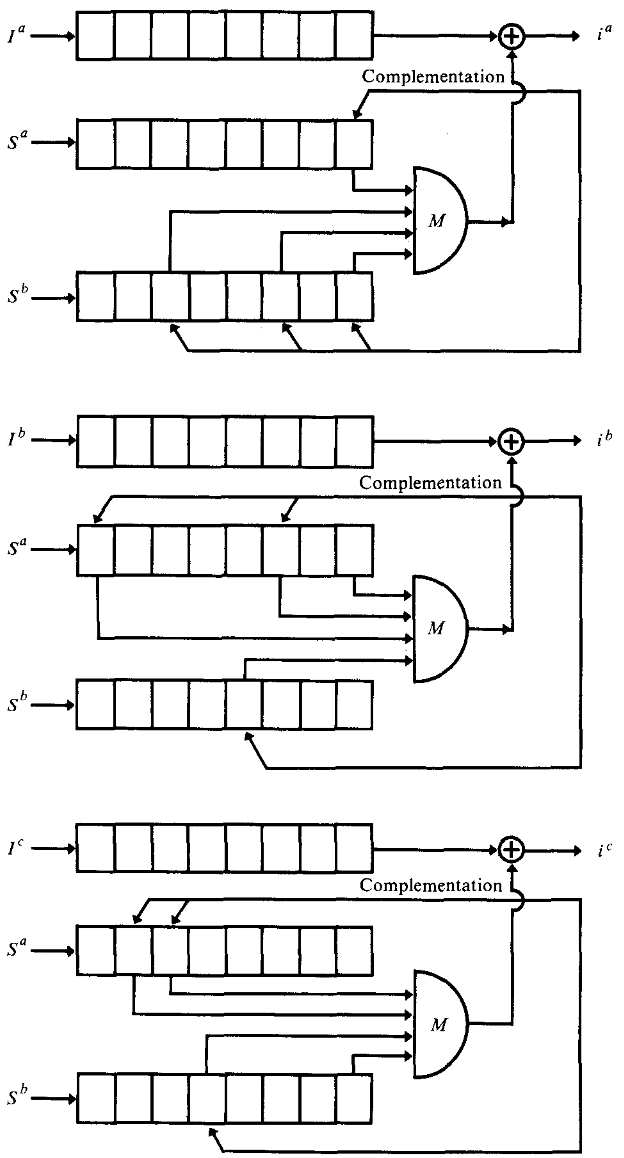



**Figure 5** Complementation of syndrome bits in error correction with the (40, 24) code.

$$i'^a_k = i^a_k \oplus e^a_k,$$
$$i'^b_k = i^b_k \oplus e^b_k,$$
$$i'^c_k = i^c_k \oplus e^c_k. \qquad (54)$$

The value of error bit $e^a_1$ is estimated as

$$e^a_1 = 1 \text{ if 3 or 4 of } \{S^a_1, S^b_1, S^b_3, S^b_6\} \text{ are 1;}$$
$$e^a_1 = 0 \text{ if 2, 3, or 4 of } \{S^a_1, S^b_1, S^b_3, S^b_6\} \text{ are 0.} \qquad (55)$$

If $e^a_1$ is estimated to be 1, then $i'^a_1$ is EXCLUSIVE ORed with $e^a_1$ to yield $i^a_1$. In this case, the syndrome bits $S^a_1$, $S^b_1$, $S^b_3$, and $S^b_6$ are complemented before the next bit is decoded. The complementation removes the effect of $e^a_1$ from the syndrome bits. Similarly, the estimates of $e^b_1$ and $e^c_1$ are determined by the following rules:

$$e^b_1 = \begin{cases} 1 \text{ if 3 or 4 of } \{S^a_1, S^a_3, S^a_8, S^b_4\} \text{ are 1,} \\ 0 \text{ otherwise;} \end{cases} \qquad (56)$$

$$e^c_1 = \begin{cases} 1 \text{ if 3 or 4 of } \{S^a_6, S^a_7, S^b_1, S^b_5\} \text{ are 1,} \\ 0 \text{ otherwise.} \end{cases} \qquad (57)$$

If the value of $e^b_1$ or $e^c_1$ is estimated as 1, certain syndrome bits are complemented as shown in Fig. 5.

- *Error correcting capability*

Let $E^a_k$ and $E^b_k$ be the error bits corresponding to the two parity checks at time $k$. Then

$$p'^a_k = E^a_k \oplus p^a_k,$$
$$p'^b_k = E^b_k \oplus p^b_k. \qquad (58)$$

Equations (53), (54), and (58) imply that

$$S^a_8 = E^a_8 \oplus e^a_8 \oplus e^b_8 \oplus e^b_6 \oplus e^b_1 \oplus e^c_3 \oplus e^c_2,$$
$$S^b_8 = E^b_8 \oplus e^a_8 \oplus e^a_6 \oplus e^a_3 \oplus e^b_5 \oplus e^c_8 \oplus e^c_4. \qquad (59)$$

Let

$$\overline{S} = [S^a_1\, S^a_2 \cdots S^a_8\, S^b_1\, S^b_2 \cdots S^b_8],$$
$$\bar{e} = [E^a_1\, e^a_1\, e^b_1\, E^b_1\, e^c_1\, E^a_2\, e^a_2\, e^b_2\, E^b_2\, e^c_2\, E^a_3 \cdots E^a_8\, e^a_8\, e^b_8\, E^b_8\, e^c_8].$$

Assume that no decoding error occurred before time 1, i.e., $E^a_k$, $E^b_k$, $e^a_k$, $e^b_k$, $e^c_k$ are all zero for $k < 1$. Then (59) can be expressed as

$$\overline{S} = \bar{e} \cdot H^t, \qquad (60)$$

where $H =$

$$\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1
\end{bmatrix}.$$

Now consider the set $\{S^a_1, S^b_1, S^b_3, S^b_6\}$ which is equal to

$$\bar{e} \cdot \begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1
\end{bmatrix}^t. \qquad (61)$$

According to the rules for the estimate of $e^a_1$, the value of $e^a_1$ is correctly determined if the number of errors is two or less; i.e., $e^a_1$ is correctable for single-bit channel errors.

A symbol channel error can be either a one-bit or a two-bit error in the two-bit symbol. A two-bit error means that the corresponding signal has been shifted 180°. This two-bit error at the output of the differential decoder becomes a four-bit error of the form 1111 as discussed in the *Error characterization* section. From (61) and the decoding rules, the value of $e_1^a$ can be determined correctly if a four-bit error of the form 1111 is in a sequence of 40 bits. The same conclusion can be made on the estimates of $e_1^b$ and $e_1^c$. Therefore, the code can correct any symbol channel error in 40 bits.

The code is incapable of correcting all possible double-channel errors. An experimental computer program has been used to test the correcting capability of double-channel errors by enumerating all possible error patterns. It showed that 70 percent of double-channel errors are correctable. Therefore, the probability that an uncorrectable error pattern begins in a block, given that the previous seven blocks are error free, can be approximated by

$$P_B = 0.3\left[\binom{5}{2} + 5 \times 35\right] p^2, \qquad (62)$$

and the probability that an information bit is decoded incorrectly, given that the previous seven blocks are error free, can be approximated by

$$P_b = P_B/3 = 18.5\, p^2. \qquad (63)$$

This bit error probability $P_b$ is plotted as a function of $R$ in Fig. 2.

## Acknowledgments

We thank M. Y. Hsiao and N. A. Vogel for their help and encouragement in pursuing the work reported in this paper. We are also grateful to J. W. Kennedy and his colleagues and D. C. Bossen for useful discussions.

## References

1. C. J. Wolejsza, Jr., A. M. Walker, and A. M. Werth, "PSK Modems for Satellite Communications," *Proceedings of the 1969 INTELSAT/IEE International Conference on Digital Satellite Communications*, London, England, p. 127.
2. J. A. Heller and I. M. Jacobs, "Viterbi Decoding for Satellite and Space Communication," *IEEE Trans. Communications Technology* **COM-19,** 835 (1971).
3. J. G. Puente, W. G. Schmidt, and A. M. Werth, "Multiple-Access Techniques for Commercial Satellites," *Proc. IEEE* **59,** 218 (1971).
4. W. C. Lindsey and M. K. Simon, *Telecommunication Systems Engineering*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1973.
5. W. R. Bennett and J. R. Davey, *Data Transmission*, McGraw-Hill Book Co., Inc., New York, 1965.
6. G. D. Forney and E. K. Bower, "A High-Speed Sequential Decoder: Prototype Design and Test," *IEEE Trans. Communications Technology* **COM-19,** 821 (1971).
7. A. D. Wyner and R. B. Ash, "Analysis of Recurrent Codes," *IEEE Trans. Information Theory* **IT-9,** 143 (1963).
8. C. V. Freiman and J. P. Robinson, "A Comparison of Block and Recurrent Codes for the Correction of Independent Errors," *IEEE Trans. Information Theory* **IT-11,** 445 (1965).
9. C. L. Chen and R. A. Rutledge, "Bounds on the Probability of Error for Convolutional Codes," to appear in *IEEE Trans. Information Theory* **IT-22,** 1976.

*Received January 17, 1975; revised March 27, 1975*

*The authors are located at the IBM System Products Division laboratory, P.O. Box 390, Poughkeepsie, New York 12602.*