From results for \((N, K)\) codes with \(N\) non-idle positions, the total number of nonzero symbols is
\[
(q-1) \frac{q^{N'}}{q^{K-1}}
\]
so
\[
W_{\text{avg}} = \frac{(q-1) N_q q^{N_q R - 1} - (q-1) (N_q - N_o) q}{N_q R - (N_q - N_o) R} \cdot \frac{q}{q - q}
\]
C.E.D.

Corollary: \(d_{\min} \leq W_{\text{avg}}\)

B. Deterministic Decoding

1. Feedback Decoding and Definite Decoding

The theorem relating \(d_{\min}\) and decoding with errors holds only for block 0. Two methods for decoding succeeding blocks have been proposed.

a. Feedback Decoding (FD)

It is assumed that all data up to time \(i\) is decoded correctly. Their effects on succeeding blocks \(i+1, \ldots, i+m\) are removed. If blocks \(j-m, \ldots, j\) have been decoded correctly and their effects removed, block \(i+1\) effectively becomes block 0 and the same decoding algorithm can be used to decode block \(i+1\) from the modified blocks \(j+1, \ldots, j+m+1\) with the same error correcting capability. If some previous decoding errors have been made, succeeding decoding will be adversely affected. In fact, noisy error propagation can occur, i.e., if a decoding error is made prior to block \(j+1\), all blocks
from time $j+1$ on can be decoded incorrectly even with no channel errors.

**Example (cont.) Feedback Decoder**

Received symbols

$$y_j^{(1)} = x_j^{(1)} + e_j^{(1)} = u_j^{(1)} + e_j^{(1)}$$  \(1\)

$$y_j^{(2)} = x_j^{(2)} + e_j^{(2)} = u_j^{(1)} + u_j^{(2)} + e_j^{(2)}$$  \(2\)

The modified received symbols are

$$y_j^{(1)'} = y_j^{(1)} = u_j^{(1)} + e_j^{(1)}$$  \(1'\)

$$y_j^{(2)'} = y_j^{(2)} - e_j^{(1)} = u_j^{(1)} + [u_j^{(2)} - u_j^{(1)}]_+ e_j^{(2)}$$  \(2'\)

also

$$y_{j+1}^{(2)'} - y_{j+1}^{(1)'} = u_j^{(1)} + e_j^{(2)} - e_j^{(1)}$$  \(3'\)

Notice that Eq 1', 2', and 3' all contain different error symbols. Assuming $u_j^{(1)} = u_{j-1}^{(1)}$ these equations give three independent estimates of $u_j^{(1)}$. If there are
no channel errors or a single channel error in the constraint length of four received symbols, \( u_j^{(1)} \) can be decoded correctly by a majority vote of \( \xi_j^5, 1', 2', \) and \( 3' \) if \( u_j^{(1)} \) has been decoded correctly. Notice that both \( \xi_j^{(2)} \) and \( \xi_j^{(1)} \) can be nonzero and \( u_j^{(1)} \) can be decoded correctly. However, \( u_j^{(1)} \) will be decoded incorrectly, so the code can correct all single errors in a constraint length. This decoder has no noisy error propagation. Assume that \( u_j^{(1)} \neq u_j^{(1)*} \) and that no channel errors occur in blocks \( j \) and \( j+1 \). Then, Eqs. 1' and 3' give correct estimates of \( u_j^{(1)} \) and the symbol is decoded correctly. Suppose we had chosen the decoding function \( u_j^{(1)*} = \xi_j^{(2)*} - \xi_j^{(1)} + S u_j^{(1)} - u_j^{(1)} + \xi_j^{(2)} \). With no channel errors this decoder correctly when previous decoding is correct. However, once a decoding error is made, all succeeding symbols are decoded incorrectly without any channel errors. This illustrates the important point that noisy error propagation depends on the code and the decoding scheme.

b. Definite Decoding (DD)

Definite decoding does not make use of past decoding estimates when decoding \( \hat{u}_j \). The estimate \( \hat{u}_j \) is calculated from the unmodified received sequence \( \{\xi_{j-m}, \ldots, \xi_j, \xi_{j+1}, \xi_{j+2}, \ldots, \xi_{j+m}\} \).
EXAMPLE: Definite Decoder

\[ E^{(1)} \]
\[ X^{(1)} \]
\[ Y^{(1)} \]
\[ D \]
\[ \oplus \]
\[ E^{(2)} \]
\[ X^{(2)} \]
\[ Y^{(2)} \]
\[ P \]
\[ \oplus \]
\[ \text{MAJORITY} \]
\[ \rightarrow D(\bar{U}^{(1)}(D)) \]

At time \( j+1 \), the decoder can calculate

\[ y^{(1)} = u_j^{(1)} + e_j^{(1)} \quad (1'') \]

\[ y_j^{(2)} - y_j^{(1)} = u_j^{(1)} + e_j^{(2)} - e_j^{(1)} \quad (2'') \]

\[ y_j^{(2)} - y_{j+1}^{(1)} = u_j^{(1)} + e_j^{(2)} - e_{j+1}^{(1)} \quad (3'') \]

Once again these equations give three independent estimates of \( u_j^{(1)} \) and all single errors in a constraint of 6 code symbols can be corrected. There can be no noisy error propagation since the decoding of \( u_j^{(1)} \) is independent of the estimate \( u_j^{(1)} + e_j^{(1)} \). Notice that Equations 1' and 1'' are identical and so are equations 2' and 2''. Equations 3' and 3'' are the same except that the channel error \( e_{j+1}^{(1)} \) appears in Eq 2'' instead of the post decoding error \( u_{j+1}^{(1)} - u_j^{(1)} = e_j^{(1)} \).
If the channel is good enough so that reliable decoding is expected, then intuitively one would expect that $P \in \mathcal{E}^{(1)} \leq 0.3 < P \in \mathcal{E}^{(2)} \leq 0.3$, i.e., that a decoding error is much less probable than a channel error. This indicates that the probability of bit decoding errors should be less with feedback decoding than with definite decoding. This has not been proven analytically. However, simulation indicates that this is true when the best decoding function is used.

2. Syndrome Decoding (for canonical systematic codes)

**Def:** Syndrome sequence.

The syndrome sequence for canonical systematic codes is the difference between the parity check symbols as computed from the received information symbols and the received parity check symbols, i.e.,

$$S^{(j)}(D) = \sum_{k=0}^{\infty} s^{(j)}(k)D = \sum_{k=1}^{K_0} G^{(j)}(k)Y^{(k)}(D) - Y^{(j)}(D)$$

**Note:** For systematic code $X^{(a)}(D) = U^{(a)}(D)$ for $a = 1, \ldots, K_0$

Therefore:

$$S^{(j)}(D) = \sum_{k=1}^{K_0} G^{(j)}(k)U^{(a)}(D) + E^{(a)}(D)$$

$$= \sum_{k=1}^{K_0} G^{(j)}(k)E^{(a)}(D) - E^{(a)}(D)$$

for $j = K_0 + 1, \ldots, N_0$

So $S^{(j)}(D)$ only depends on the error sequence and not on the information vector.

Define $\mathbf{s} = [s^{(K_0+1)}, \ldots, s^{(N_0)}]$
and \( \vec{S} = [\vec{s}_0, \vec{s}_1, \ldots, \vec{s}_m] \) 

is the \( 1 \times N_a(1-R) \) syndrome vector used to decode block 0 since an error in block 0 can affect only the following \( m \) blocks. As in the case of block codes, it could have been caused by \( q^{R_m} \) different error vectors since the actual error sequence + any one of the \( q^{R_m} \) codewords results in the same syndrome. The optimum syndrome decoder would determine the most likely error pattern consistent with the syndrome.

In a feedback syndrome decoder the errors in block 0, \( \vec{e}_0 \), are estimated from \( \vec{S} \) and then used to remove their effect on \( \vec{S} \). \( \vec{e}_1 \) is then estimated from \( \{\vec{s}_1, \ldots, \vec{s}_{m+1}\} \) using the same algorithm.

Notice that errors in the information symbols of block 0 affect \( \{\vec{s}_0, \ldots, \vec{s}_m\} \) while errors in the check symbols of block 0 only affect \( \vec{s}_0 \). Therefore in decoding \( \vec{e}_1 \), only the estimated information symbol errors are needed to modify \( \{\vec{s}_1, \ldots, \vec{s}_m\} \).

**Example**

\( R = 1/2 \), \( N_o = 2 \)

\( G_{(2)}(D) = D + g_0 + g_1 + g_2 + \ldots + g_m D^m \)

---

**Diagram**

[Diagram of the error correction process]
With the switch in the FD position the decoder is a feedback decoder. In the DP position it is a definite decoder. Notice that the decoder has a memory of $2m+1$.

**Decoder for Simple Example**

\[ E^{(1)} \]

\[ \begin{array}{c}
U^{(1)}(D) \\
\downarrow \\
E^{(2)} \\
\downarrow \\
\left\{ \begin{array}{c}
S_{1}^{(2)} + S_{2}^{(2)} \\
S_{1}^{(2)} + S_{3}^{(2)} \\
S_{1}^{(2)} + S_{4}^{(2)} \\
\end{array} \right. \\
\left\{ \begin{array}{c}
M \\
D \end{array} \right. \\
\end{array} \]

**Encoder | Channel | Decoder**

In DP mode:

\[ S_{j}^{(2)} = e_{j+1}^{(1)} + e_{j}^{(1)} - e_{j+1}^{(2)} \]

In FD mode, the modified contents of the syndrome register is:

\[ S_{j+1}^{(2)} = S_{j}^{(1)} = e_{j+1}^{(1)} + e_{j}^{(1)} - e_{j+1}^{(2)} \]

\[ S_{j}^{(2)}' = S_{j}^{(2)} - e_{j+1}^{(1)*} = e_{j} + [e_{j+1}^{(1)*} - e_{j+1}^{(2)*}] - e_{j} \]

This decoder is equivalent to the decoders on p. 62 and p. 64. Notice that in the feedback mode, the channel error $e_{j+1}^{(1)}$ is replaced by the post-decoding error $e_{j+1}^{(1)*}$. 
3. Threshold decoding [Ref. Massey, B, Collage 3]

The decoding functions for the feedback and definite decoder examples and \( f(\bar{2}) \) for the syndrome decoder example appear to have been "pulled out of the air."

One of the major elements of these decoders was a majority or threshold device. Threshold decoding was developed by Massey as a simple method for implementing the logical function \( f(\bar{2}) \). Threshold decoding is most efficient with binary codes.

Massey [Ref B, p. 91] shows that threshold decoders cannot take advantage of the improved distance structure of codes with symbols from higher order alphabets. Therefore, the discussions here will be limited to binary codes. Threshold decoding is applicable to both block and convolutional codes so the theory will be formulated in a general framework.

Any syndrome digit is a known linear combination of the error symbols. Also any combination of syndrome symbols is once again a known function of error symbols. The syndrome vector calculated at the receiver has \( N_e (1-R) \) components.

Over GF(2) there are \( 2^{N_e (1-R)} \) possible linear combinations of syndrome symbols. Any such known sum will be called a parity check and will be denoted by the symbol \( A_i \). \( A_i \) will be said to check error digit
Def: Orthogonal parity checks

A set \( S_{A,3} \) of parity checks is said to be orthogonal on \( e_m \) if each \( A_i \) checks \( e_m \) but no other error bit is checked by more than one \( A_i \).

Ex: \( A_1 = e_1 + e_2 + e_3 \)
\( A_2 = e_3 + e_4 + e_5 \)
\( A_3 = e_3 + e_6 + e_7 \)

\( S_{A,3} \) orthogonal on \( e_3 \)

For a memoryless channel, a set of \( J \) parity checks orthogonal on \( e_m \) provides \( J \) statistically independent estimates of \( e_m \). This suggests decoding by majority vote. The following theorem characterizes the error correction and detection capabilities of majority decoding.

Theorem: Given a set of \( J = 2t + 1 \) parity checks orthogonal on \( e_m \), any pattern of \( t \) or fewer errors in the bits checked by \( S_{A,3} \) will be decoded correctly and patterns of \( t+1, \ldots, 2t+1 \) errors will cause an error alarm if the following decoding rule is used:

\[
e^*_m = 1 \text{ if more than } \frac{J+t}{2} \text{ of } A_i \text{ are 1}
\]
\[
e^*_m = 0 \text{ if } \frac{J-t}{2} \text{ or fewer have value 1}
\]

Error alarm otherwise.

Proof: Assume \( e_m = 0 \) and no more than \( t \) errors have occurred. In the worst possible case, each error would
occur in a different check so that at most $t = \frac{J - 3}{2}$ checks could be 1 and $e_m$ is correctly decoded as 0. Similarly, no more than $t + \frac{J - 3}{2} + a = \frac{J + a}{2}$ errors will cause at most $\frac{J + a}{2}$ checks to be 1 and $e_m$ is not incorrectly decoded to 1.

Assume that $e_m = 1$ and no more than $t$ errors have occurred. Then at most $t - 1$ of the other error bits checked can be 1. $e_m = 1$ in each check so that $J - (t - 1) = J - (\frac{J - 3}{2} + 1) = \frac{J + 3}{2} + 1$ or more checks must be 1 and $e_m$ is correctly decoded. With no more than $t + a$ errors, at least $J - (t + a - 1) = \frac{J - a}{2} + 1$ checks will be 1 so that $e_m$ is not incorrectly decoded to 0.

Q.E.D.

If $K_0$ sets of $J$ parity checks orthogonal on each of the information noise symbols can be found, all patterns of $t = \frac{J}{2}$ or fewer errors in the bits checked can be corrected. Thus, if this is true, $d_{\min} = 2t + 1 = J + 1$. If $J = d_{\min} - 1$ orthogonal checks can be found for each information noise bit, the code is said to be completely orthogonalizable. If a code can be completely orthogonalized, any error pattern guaranteed correctable by $d_{\min}$ can be corrected by majority decoding. In addition, many error patterns of higher weight will generally be corrected also. For example, if $J = 10$, checks orthogonal on $e_m$ are known, then $t = \frac{J - 5}{3} = 5$ and six errors will cause a
decoding error only if each falls in a different parity check. This is unlikely if each \( A_i \) checks several errors. See (Massey 8, pp. 90-91) for an example of threshold decoding for maximal length gives \( P(e = 5 \times 10^{-7}) \), while an algorithm of a more sophisticated decoder makes use of probabilistic information. Given a set of parity checks \( \{ A_i \} \), a minimum probability of error decoder for \( e_m \) is

\[
e_m^* = \begin{cases} 1 & \text{if } P(e_m = 1 \mid \{ A_i \}) > P(e_m = 0 \mid \{ A_i \}) \\ 0 & \text{otherwise} \end{cases}
\]

Using Bayes' rule, the inequality becomes

\[
P(e_m = 1) P(\{ A_i \} \mid e_m = 1) > P(e_m = 0) P(\{ A_i \} \mid e_m = 0)
\]

with a memoryless channel, the \( A_i \) with \( e_m \) fixed are statistically independent for an orthogonal set.

So

\[
P(e_m = 1) \prod_{i=1}^{J} P(A_i \mid e_m = 1) > P(e_m = 0) \prod_{i=1}^{J} P(A_i \mid e_m = 0)
\]

Let \( \rho_e = 1 - \rho_0 = P(e_m = 1) \) and \( \rho_1 = 1 - \rho_1 = P(A_i \neq e_m) \).

\[
P(A_i \neq e_m) = P \text{ if odd number of errors in bits checked by } A_i
\]

exclusive of \( e_m \).

\[
P(A_i = e_m) = P \text{ if odd number of errors in bits checked by } A_i
\]

So

\[
P(A_i \mid e_m = 1) = \rho_i^{1-A_i} \rho_{1-A_i}^{A_i}
\]

with \( A_i \) considered to be a real number. Also

\[
P(A_i \mid e_m = 0) = \rho_i^{1-A_i} \rho_{1-A_i}^{A_i}
\]

Thus the decoding rule is set \( e_m^* = 1 \) if and only if

\[
1 - \prod_{i=1}^{J} \rho_i^{1-A_i} \rho_{1-A_i}^{A_i} > \prod_{i=1}^{J} \frac{1-A_i}{A_i} \rho_i
\]
Taking logarithms and rearranging this becomes

$$\sum_{i=1}^{J} A_i \cdot (z \log \frac{p_i}{p_i}) > \sum_{i=0}^{J} \log \frac{p_i}{p_i}$$

**Theorem:** The minimum probability of error decoder for \( \mathbf{e} \) in \( \mathbf{m} \) is

$$e^* = \begin{cases} 1 & \text{if } \sum_{i=1}^{J} A_i \cdot (z \log \frac{p_i}{p_i}) > \sum_{i=0}^{J} \log \frac{p_i}{p_i} \\ 0 & \text{otherwise} \end{cases}$$

For an additive white Gaussian noise channel calculations show that an effective power gain of about 2 dB over simple majority decoding is achieved with APP decoding.

**Error Probabilities**

**Def:** \( P_i(e) \) = average probability of incorrectly decoding block \( \mathbf{o} \) from the first \( N_A \) code bits.

It can be shown [Money - 2, 6] that with maximum likelihood decoding \( P_i(e) \) approaches 0 exponentially with \( N_A \) for convolutional codes by random coding arguments. However, with threshold decoding it appears that this is not true [1, 7]. For one class of codes, Money has shown that

$$P_i(e) > \frac{1}{2} \left( \frac{p_0}{p_0} \right) ^{\frac{52.8 + i - 17}{2.80}}$$

For a variety of other codes he has calculated \( P_i(e) \) exactly by machine calculation for both APP and majority.
Performance Data for Threshold Decoding of Convolutional Codes

Fig. 16. Performance of $R = \frac{1}{2}$ trial-and-error codes on the Binary Symmetric Channel.

$$n = \text{actual number of noise bits checked by } \{A_d\}$$

$$E = \text{effective constraint length } \leq n_A$$

$$p_0 = \text{transition probability for BSC}$$
decoding. The accompanying figure shows typical results.

\( n_e \) is the number of bits checked so that \( n_e \leq N_0 \). With

APP decoding \( P(e) \) decreases monotonically with \( n_e \) but with

majority decoding \( P(e) \) reaches a minimum and eventually

increases to 1/2. This indicates that threshold decoding is

only useful at moderate constraint lengths.

4. Convolutional Codes for Threshold Decoding

a. Self Orthogonal Codes

A code is said to be self orthogonal if

for each information noise digit the set of syndrome
digits checking that digit form an orthogonal set
on that digit.

Consider the binary code with \( R = 1/2\), \( N_0 = 2 \). Let

\[
G^{(1)}(D) = d_0 + d_1 + \ldots + d_n
\]

with \( 0 < d_0 < d_1 < \ldots < d_n \) and \( \{d_i\} \) integers. Then

\[
S^{(2)}(D) = G^{(2)}(D) E^{(1)}(D) + E^{(2)}(D)
\]

\[
= \left[ d_0 d_1 \ldots d_n \right] E^{(1)}(D) + E^{(2)}(D)
\]

\[
= \left[ d_0 \ldots d_n \right] \left[ e_0 + e_1 D + \ldots \right] + \left[ e_0 + e_1 D + \ldots \right]
\]

The syndrome bits that check the block 0 information

even digit \( e^{(1)}_0 \) are then

\[
S^{(2)}_{d_0} = e^{(1)}_0 + e^{(2)}_{d_0}
\]

\[
S^{(2)}_{d_1} = e^{(1)}_0 + e^{(1)}_{d_1-d_0} + e^{(2)}_{d_1}
\]
Each syndrome bit contains a different check none bit $e^{(i)}$. The syndrome bit $s_j$ contains the $j$ information none bit $e_j$ corresponding to the difference 
$\{ d_i - d_j, i < j \}$ in addition to $e_j^{(1)}$. There are $1 + 2 + \cdots + n = \frac{1}{2} n (n+1)$ differences in these sets.

For $j = 1, \ldots, n$, the set of syndrome digits is orthogonal on $e_j^{(1)}$ if and only if these sets of differences are all distinct. If $d_1 \neq 0$ one would subtract $d_0$ from each $d_i$ to produce a self-orthogonal code with the same set of differences but smaller memory.

A code was defined to be self-orthogonal if the syndrome digits checking the block of information none digits formed orthogonal sets. However, for self-orthogonal codes the set of syndrome digits which check any information error bit form an orthogonal set. Consider the $k = \frac{1}{2}$, $n = 2$ code. The syndrome digits checking $e_j^{(1)}$ are

$$s_j^{(1)} = e_j^{(1)} + \sum_{k=1}^{n} e_j^{(1)} d_i - d_k \sum_{k=1}^{n} e_j^{(1)} d_i - d_k$$

$$s_j^{(2)} = \sum_{k=0}^{n} e_j^{(1)} d_i - d_k$$

$$s_j^{(3)} = \sum_{k=0}^{n} e_j^{(1)} d_i - d_k$$

$$s_j^{(4)} = \sum_{k=0}^{n} e_j^{(1)} d_i - d_k$$
where \( e_{ij} = 0 \) if \( i < j \). If this set is not orthogonal, then for some \( \alpha \neq \beta \) \( (\alpha > \beta) \)

\[
\bar{d}_i - d_i = d_j - d_j
\]

\[
\bar{d}_i - d_i = d_j - d_j
\]

If \( \alpha > \beta \) this is impossible since all positive differences distinct. If \( \alpha < \beta \)

\[
\bar{d}_i - d_i = d_j - d_j > 0 \text{ since } \alpha > \beta
\]

and also get contradiction (for \( \bar{d}_i - d_i < 0 \))

This implies that a majority decoder for a self-orthogonal code can be used in either the feedback or definite mode with the same error correction radius.

A method for finding self-orthogonal codes, described by Robinson, is based on the concept of difference sets.

**Difference Set**

A difference set of order \( n+1 \) and modulus \( N \) is a set \( \{d_0, \ldots, d_{n+1}\} \) of \( n+1 \) integers with \( 0 \leq d_0 < d_1 < \cdots < d_{n+1} < N \) such that \( (d_j - d_i) \mod N \) are all non-zero and distinct for \( i \neq j \). There are \( (n+1)^n \) possible differences and the residues \( (d_j - d_i) \mod N \) are all between 0 and \( N \) for \( i \neq j \).

It is easy to show that \( x \mod N = y \mod N \) if and only if \( x = y \).

If each residue is distinct then \( N-1 \geq (n+1)^n \)

or \( n^2 / n+1 \). A difference set of modulus \( N = n^2 / n+1 \)

is said to be perfect. It can be shown that perfect difference sets exist for every \( n = p^r \) where \( p \) is prime and \( r \) is an integer and it is known that no other perfect
Difference sets exist for $n \leq 1600$.

**Ex:** The following two sets are perfect difference sets:

- $\{0, 1, 4, 6\}$ - $n = 3$, $N = 3^2 - 3 + 1 = 13$
- $\{0, 3, 4, 9, 11\}$ - $n = 4$, $N = 21$

Thus every difference set describes an $R = \frac{1}{2}$, $N_0 = 2$ binary code with $G^{(2)}_{(i)}(D) = \sum_{k=0}^{n} D^k$ that is self-orthogonal. Perfect difference sets give self-orthogonal codes with $J = n + 1 = p^r$ with the shortest possible memory.

Thus: For self-orthogonal codes with $R = \frac{1}{2}$, $N_0 = 2$,

$\text{d}_{\text{min}} = J + 1$

**Proof:** Always $\text{d}_{\text{min}} \geq J + 1$. But $G^{(2)}_{(i)}(d)$ has only $J = n + 1$ non-zero terms so that $G^{(2)}_{(i)}(0) = 1$ gives a code word with $J + 1$ ones. Q.E.D.

Thus majority decoding is an efficient decoding scheme for self-orthogonal codes.

**Ex:** $R = \frac{1}{2}$, $N_0 = 2$ / Perfect difference set $\{0, 1, 4, 6\}$

$$G^{(2)}_{(i)}(D) = 1 + D + D^4 + D^6$$

---

[Diagram of a circuit or system]
Difference sets exist for $n > 600$.

**Ex:** The following two sets are perfect difference sets

- $\{0, 1, 4, 6, 3\}$, $n = 3$, $N = 3^2 + 3 + 1 = 13$
- $\{0, 3, 4, 9, 11\}$, $n = 4$, $N = 21$

Thus, every difference set describes an $R = \frac{1}{2}$, $N_0 = 2$

binary code with $G_{(1)}^{(2)}(b) = \sum_{k=0}^{N_0} b^k$ that is self-orthogonal. Perfect difference sets give self-orthogonal codes with $J = n + 1 = p^r$ with the shortest possible memory.

**Thm:** For self-orthogonal codes with $R = \frac{1}{2}$, $N_0 = 2$

$d_{\text{min}} = J + 1$

**Proof:** Always $d_{\text{min}} \geq J + 1$. But $G_{(1)}^{(2)}(b)$ has only $J = n + 1$ non-zero terms so that $G_{(1)}^{(2)}(0) = 1$ gives a code word with $J + 1$ ones. Q.E.D.

Thus, majority decoding is an efficient decoding scheme for self-orthogonal codes.

**Ex:** $R = \frac{1}{2}$, $N_0 = 2$, perfect difference set $\{0, 1, 4, 6\}$

$$G_{(1)}^{(2)}(b) = 1 + b + b^4 + b^6$$
(N_0, N_0-1) self orthogonal Codes

Let
\[ G(0) = 1 + d_1 z^{-1} + \ldots + d_r z^{-r} \]

\[ G_{(0)}(c_j) = 1 + d_1 c_j^{-1} + \ldots + d_r c_j^{-r} \]

\[ G_{(N_0-1)}(0) = 1 + d_1 c^{-1} + \ldots + d_r c^{-r} \]

and \( d_0 = d_0^2 = \ldots = d_{(N_0-1)} = 0 \)

The syndrome digits that check \( e_0^n \) are

\[ s_0 = e_0 + e_1 z^{-1} + e_2 z^{-2} + \ldots + e_{(N_0-1)} z^{-N_0+1} \]

\[ s_{d_1} = e_0 + e_{d_1} z^{-1} + \sum_{i=0}^{d_1-2} e_i z^{-i} + \ldots + e_{d_1-d_2-1} z^{-d_1-d_2+2} \]

\[ s_{d_21} = e_0 + e_{d_21} z^{-1} + e_{d_21-d_1} z^{-d_21-d_2+1} + \ldots + e_{(N_0-1)} z^{-d_21-d_2+2} \]

\[ \ldots \]

\[ s_{d_{r1}} = e_0 + e_{d_{r1}} z^{-1} + \sum_{i=0}^{d_{r1}-d_{(N_0-1)}} e_i z^{-i} \]

As in the \((2,1)\) case, the \( r (r+1)/2 \) difference \( d_{ij} - d_{ik} \) for \( 0 \leq i < j \) and \( r = 1, \ldots, r \)

must be distinct. In addition, positive difference of the form \( d_{i1} - d_{r1} \) for \( k \neq 1 \) must

be distinct. Therefore, from j-th row

\[ d_{j1} - d_{j1} \neq d_{j1} - d_{k1} \text{ for } j \neq k, k \neq 1 \]
or \( d_{31} - d_{3} \neq d_{14} - d_{1} \)

Therefore, the differences in powers of \( D \) in the polynomials other than \( G^{(N_0)} \) cannot equal the \( r(r+1)/2 \) differences from \( \{d_{01}, \ldots, d_{r3}\} \).

If a set of \( (N_0-1) r \) numbers can be found such that this is true, then

\[ \{ s_{01}^{(N_0)}, \ldots, s_{r3}^{(N_0)} \} \]

are a set of \( J = r+1 \) checks orthogonal on \( e_0^{(i)} \).

Using the same reasoning, it follows that if the sets of differences for

\[ \{ d_{01}, d_{11}, \ldots, d_{13} \} \]

\[ \{ d_{22}, \ldots, d_{23} \} \]

\[ \{ d_{33(N_0-1)}, \ldots, d_{r3(N_0-1)} \} \]

are all distinct and each set of differences contains different numbers, then

\[ \{ s_{01}^{(N_0)}, s_{11}^{(N_0)}, \ldots, s_{r3}^{(N_0)} \} \text{ for } 1 \leq i \leq N_0-1 \]

form a set of \( J = r+1 \) checks orthogonal on \( e_0^{(i)} \).

Sets of integers with these properties are given by Weldon in:


pp 407-408

for \( D = \{1/2, 2/3, 3/4, 4/5\} \)
Using threshold decoding $t = J/2$ so that $d_{\text{min}} \geq J+1 = r+2$. Let $U^{(1)}(D) = 1,$

\[ U^{(2)}(D) = \cdots = U^{(N_0-1)}(D) = 0. \]

Then the corresponding code word has $x_0^{(1)} = 1,$ $x_0^{(N_0)} = 1,$ $x_0^{(N_0)} = 1,$ \ldots \ $x_0^{(r+1)} = 1$, and all other $x_i = 0$. Therefore $d_{\text{min}} = J+1 = r+2$ and these codes are completely orthogonalizable.

Each set of differences has $r(\text{r+1})$ numbers so that the largest number in the total set must lie at least $(N_0-1) r(\text{r+1})$ since there are $(N_0-1)$ sets of differences. Thus

\[ m \geq (N_0-1) r(\text{r+1})/2 \text{ and } N_A \geq N_0 (m+1) \geq N_0 \left[ \frac{(N_0-1) r(\text{r+1}) + 1}{2} \right]. \]

Error Propagation in FB mode

A threshold element will output a 1 if $t > J/2$ ones appear at its input. With no future channel errors, future syndromes are 0. The feedback connections complement the T ones and adds $r+1 - T$ additional ones. Thus the weight of the syndrome changes by $T - (r+1 - T) = 2T - (r+1)$ \leq $2T - 3 > 0$ for each 1 output. Therefore, the syndrome register will eventually clear to 0 with no future channel errors.
Consider the code with generator polynomials

\[ g^{(1)}_1(D) = 1 \]

\[ g^{(2)}_1(D) = g^{(N_0)}_1(D) \]

\[ \vdots \]

\[ g^{(N_0)}_1(D) = g^{(N_0)}_{N_0-1}(D) \]

where \( g^{(i)}_j \), \( i = 1, \ldots, N_0-1 \) are the generator polynomials for the \( (N_0, N_0-1) \) codes discussed previously. The resulting code has \( J = (N_0-1)(r+1) \) checks orthogonal on \( c_i^{(1)} \). Thus \( d_{\text{min}} \geq J+1 \). Let \( C(D) = 1 \). Then first constraint length of code word has \( w(8) = 14(N_0-1)(r+1) \) so \( d_{\text{min}} = J+1 \) and code completely orthogonalizable.

**Ex:**

(a) \((3, 2)\) code, \( d=5 \) \( \Rightarrow J = d-1 = 4 \)

consider sets \( \{0, 9, 12, 13\}, \{0, 6, 4, 13\} \)

differences \( r/(r+1)/2 = 3, 4/2 = 6 \)

\[
\begin{array}{cccc}
0 & 8 & 9 & 12 \\
\hline
1 & 8 & & \\
2 & 9 & 1 & \\
3 & 12 & 4 & 3 \\
\end{array}
\]

\[
\begin{array}{cccc}
0 & 6 & 1 & 13 \\
\hline
1 & 6 & & \\
2 & 11 & 5 & \\
3 & 13 & 7 & 2 \\
\end{array}
\]
Encoder

Decoder

(5) (3, 1) code, $J = 2 \times 4 = 8
In the FD mode whenever the threshold element emits a 1, the number of 1's stored in the syndrome register is reduced by at least two and this is true in general for self orthogonal codes. This ensures that this decoder has no error propagation since if there are no channel errors of the time i, then at time i + m only 0's enter the syndrome register from the left. No matter what the contents of the register at time i + m, it will clear to all 0's soon since this occurs when the threshold element has no 1 outputs for m consecutive decisions. Therefore, there can be only a finite number of 1 outputs because the number of 1's is reduced by at least 2 for each 1 output.

6. Uniform Codes

On page 60 it was shown that for convolutional codes, \( W_{\text{avg}} = \left( \frac{q-1}{q} \right) \left[ N_0 + \frac{N_0}{q^{m-1}} \right] \).

A class of convolutional codes will be presented here with \( K_0 = 1 \), \( N_0 = q^m \) and \( R = \frac{1}{q^m} \) and \( d_{\text{min}} = W_{\text{avg}} = \left( \frac{q-1}{q} \right) \left( \frac{m+1}{q} \right) q^m = \left[ \frac{(q-1)(m+1)+1}{q} \right] q^m \).

Let \( g_{(i)}^{(1)}(D) = 1 \) and \( g_{(i)}^{(1)}(D) = 1 + g_{(i)}^{(1)} D + g_{(i)}^{(2)} D^2 + \cdots + g_{(i)}^{(m)} D^m \)

for \( i = 1, \ldots, N_0 = q^m \) such that \( \left[ g_{(1)}^{(i)}, \ldots, g_{(m)}^{(i)} \right] \) for \( i = 1, \ldots, N_0 \) are all the distinct \( q^m \)-tuples.

Since \( g_{(i)}^{(1)}(0) = 1 \), the codes are systematic. With this convention, the \( K_0 \times N_0 \) matrices \( G_i \) on p. 33.
are \( (N_0 = q^n) \) new vectors

\[
G_i = \left[ g^{(i)}_1, g^{(i)}_2, \ldots, g^{(i)}_{N_i} \right], \quad i = 0, \ldots, m.
\]

Since \( G_i^{(1)} \) has constant term 1 for all \( i \),

\[
G_0 = \left[ 1, 1, \ldots, 1 \right]
\]

and \( g^{(1)}_i = 0 \) for \( n^2 i \geq 1 \).

Consider code with \( m \) then

\[
\tilde{x}_m = \left[ u_0, u_1, \ldots, u_m \right] \left[ \begin{array}{c} G_m \\ G_{m-1} \\ \vdots \\ G_0 \end{array} \right]
\]

and with the specified constraints on the \( G_i \)’s

\[
\tilde{x}_m = \left[ u_0, \ldots, u_m \right] \left[ \begin{array}{c} 0 \\ \vdots \\ A_m \\ \vdots \\ 0 \end{array} \right]
\]

where the columns of \( A_m \) are all the \( q^{m-1} \) distinct non-zero \( m \)-tuples over \( \text{GF}(q) \). If \( u_0 \neq 0 \) and \( u_m = 0 \), then \( \left[ x^{(i)}_m, \ldots, x^{(q^n)}_m \right] \) is a non-zero code word in a maximal length code and \( x^{(i)}_m = 0 \). Therefore \( W(\tilde{x}_m) = (q-1)q^{m-1} \).

If \( u_0 \neq 0 \) and \( u_m \neq 0 \) then \( [u_m, \ldots, u_m] \) is added to the maximal length code word. This changes to 0 only
those \( q^{m-1} \) positions that contain \(-u_0\) but makes the \( q^{m-1} \) positions with zeros, non-zeros so that
\[
W(\vec{x}_m) = (q-1)q^{m-1} \text{ whenever } u_0 \neq 0. \text{ Notice that }
W(\vec{x}_o) = W(u_o, \ldots, u_o) = q^m \text{ for } u_0 \neq 0.
\]
For \( 1 \leq j < m \)

\[
\vec{x}_j = [u_o, \ldots, u_j] \begin{bmatrix}
0 & A_j(j) \\
1 & \cdots & 1
\end{bmatrix}
\]

where \( A_j(j) \) is the last \( j \) rows of \( A_m \).

For \( u_0 \neq 0 \), this is equivalent to the case for
\( \vec{x}_m \) where \( u_0 = u_1 = \ldots = u_{m-j-1} = 0 \), \( u_{m-j} \neq 0 \), and
\( u_{m-j+1}, \ldots, u_m \) are arbitrary. Thus the analysis for
\( \vec{x}_m \) is directly applicable so that \( W(\vec{x}_j) = (q-1)q^{m-1} \)
for \( u_0 \neq 0 \). Combining these results

\[
W(\vec{x}_0) = q^m + m(q-1)q^{m-1} = q^m \left[ (q-1)(m+1) + 1 \right]
\]

\( u_0 \neq 0 \)

**Decoding Binary Uniform Codes**

For \( q = 2 \), \( W(\vec{x}) = 2^{m-1}(n+2) = d_{\min} \)

The decoding procedure will be illustrated by the following example. Consider the \( R = 1/4 \)

binary uniform code with

\[
g^{(1)}(d) = 1
\]

\[
g^{(2)}(d) = 1 + d
\]

\[
g^{(3)}(d) = 1 + d^2
\]

\[
g^{(4)}(d) = 1 + d + d^2
\]
For this code \( m = 2 \), \( N_4 = (m+1)N_0 = 12 \), \( d_{min} = 8 \)

The syndrome transforms are
\[
S^{(i)}(D) = (\gamma^{(i)}(D)E^{(i)}(D) + E^{(i)}(D))_{j = 2, 3, 4}
\]

and at time \( k \)
\[
S^{(i)} = \frac{1}{2} \sum_{i=0}^{k-2} S^{(i)} e^{(i)} + e^{(i)}
\]

For feedback decoding, the equations for the modified syndrome register used for estimating \( e^{(i)} \) at times \( t, t+1, t+2, \ldots, t+m \) can be written as

| \( S^{(2)} \) | \( S^{(2)} \) | \( 0 \) | \( 1 \) | \( 0 \) | \( 0 \) | \( 1 \) | \( 0 \) | \( 0 \) | \( 1 \) | \( 0 \) | \( 0 \) | \( 1 \) | \( 0 \) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| \( S^{(3)} \) | \( S^{(3)} \) | \( S^{(3)} \) | \( S^{(3)} \) | \( S^{(3)} \) | \( S^{(3)} \) | \( S^{(3)} \) | \( S^{(3)} \) | \( S^{(3)} \) | \( S^{(3)} \) | \( S^{(3)} \) | \( S^{(3)} \) | \( S^{(3)} \) |
| \( S^{(4)} \) | \( S^{(4)} \) | \( S^{(4)} \) | \( S^{(4)} \) | \( S^{(4)} \) | \( S^{(4)} \) | \( S^{(4)} \) | \( S^{(4)} \) | \( S^{(4)} \) | \( S^{(4)} \) | \( S^{(4)} \) | \( S^{(4)} \) | \( S^{(4)} \) |

\[
e^{(n)} = \frac{1}{2} e^{(n)} - e^{(n-2)}
\]

The three \( (m+1) \times (2m+1) \) sub-matrices in the first matrix after the equality sign on the right are known as "parity parallelograms." Assuming that all past decoding decisions have been correct, i.e., \( e^{(n)} = 0 \) for \( i < k \), a set of checks orthogonal on
\( e^{(1)} \) can be formed by the following rules:

1. Use first row in each "parity parallelogram"
2. Use any other row which checks only one digit in \( E^{(1)} \) other than \( e^{(1)} \)
3. Add to any other row that checks \( e^{(1)} \) and at least two other digits in \( E^{(1)} \) that unique row which checks the same other digits in \( E^{(1)} \) but not \( e^{(1)} \).

For the example above, this results in the following set of parity checks:

\[
\begin{align*}
S^{(1)} &= e^{(1)} A + e^{(1)} B + e^{(2)} C \\
S^{(2)} &= e^{(1)} A + e^{(2)} B + e^{(1)} C \\
S^{(3)} &= e^{(1)} A + e^{(2)} B + e^{(2)} C \\
S^{(4)} &= e^{(1)} A + e^{(1)} B + e^{(1)} C \\
S^{(5)} &= e^{(1)} A + e^{(1)} B + e^{(2)} C \\
S^{(6)} &= e^{(1)} A + e^{(1)} B + e^{(2)} C \\
S^{(7)} &= e^{(1)} A + e^{(1)} B + e^{(2)} C \\
S^{(8)} &= e^{(1)} A + e^{(1)} B + e^{(2)} C \\
S^{(9)} &= e^{(1)} A + e^{(1)} B + e^{(2)} C \\
\end{align*}
\]

This gives \( 2^7 = 2^{\min - 1} \) checks orthogonal on \( e^{(1)} \) so that the code is completely orthogonalizable.

For any uniform code, the procedure outlined above always results in \( 2^{\min - 1} \) checks orthogonal on \( e^{(1)} \).

Proof: Consider the last row of each parity parallelogram. Each row ends in a 1. Since \( 2^{(1)} = 1 + 2^{(1)} + 2^{(2)} + \ldots + 2^{(\min)} \)
for $j = 2, \ldots, 2^n$ correspond to all $2^n$ possible polynomials except $\mathbf{e}_{i+1}^{(j)}(0) = 1$. Of the rows have a 1 in the position corresponding to $\mathbf{e}_i$. Except for the row $[0, \ldots, 0, 1, 0, \ldots, 0, 1]$, there is a row identical to one with a 1 in the position corresponding to $\mathbf{e}_i$ except for a 0 in the position corresponding to $\mathbf{e}_i$. If these rows are added pairwise, they form a set of $2^{m-1}$ orthogonal checks on $\mathbf{e}_{1/2}$. Each checks only $\mathbf{e}_{1/2}$ and two corresponding check noise bits. The row $[0, \ldots, 0, 1, 0, \ldots, 0, 1]$ must be taken alone since $[0, \ldots, 0, 1]$ is not an allowable row. This results in a check on $\mathbf{e}_{1/2}$, $\mathbf{e}_{1/2}$ and the corresponding check noise bit. All these together form a set of $2^{m-1}$ checks orthogonal on $\mathbf{e}_{1/2}$. The same analysis can be applied to the set of $m-1$ at rows. Only $[0, \ldots, 0, 1, 0, \ldots, 0, 1, 0]$ is taken alone and the checks $\mathbf{e}_{1/2}$, $\mathbf{e}_{1/2}$ and a check noise bit. The others only check $\mathbf{e}_{1/2}$ and a check noise bit. This set together with the set for the $m$ rows form a set of $2 \times 2^{m-1}$ checks orthogonal on $\mathbf{e}_{1/2}$. The same reasoning holds for all but the 1st rows. Each first row only checks $\mathbf{e}_i$ and a check noise bit. Therefore the total number of resulting checks is

$$J = m (2^{m-1}) + (2^m - 1) = 2^{m-1} (m + 2) - 1 = d_{\text{min}} - 1$$

$$\sum_{n=1}^{N_0} 1^{st \text{ \text{rows}}} \quad \text{Q.E.D.}$$
Notice that by this construction every information and check noise bit is checked by the resulting set.

Def: Effective constraint length \( N_e \)

The effective constraint length, \( N_e \), is the total number of distinct bits checked by the orthogonal \( J \) parity checks.

Therefore, for uniform codes \( N_e = N_A \).

**Error Propagation**

Previous decoding errors only affect the \( N_e - 1 \) checks corresponding to the first row of the parity parallelograms due to the method of constructing the set of \( J \) checks. For the example we have been considering

\[
\begin{bmatrix}
  z(1) \\
  z(2) \\
  z(4)
\end{bmatrix} = \begin{bmatrix}
  e(1) \\
  e(2) \\
  e(4)
\end{bmatrix} \begin{bmatrix}
  0 & 1 & 1 \\
  1 & 0 & 1
\end{bmatrix} + \text{other terms}
\]

Notice that \( A_2 \) generates a maximal length block code with the weight of all non-zero words equal to 2. Therefore a previous decoding error causes exactly two of the \( J \) orthogonal checks to be complemented. This is equivalent to receiving two channel errors that affect different checks. Since \( d_{\text{min}} = 8 \), one real channel error in \( \left[ \tilde{z}_k, \ldots, \tilde{z}_m \right] \) can be corrected. Thus in addition to no noisy errors
propagation, some channel errors can still be
corrected when previous decoding errors have
been made. In general past decoding errors
among $E^{(a)}, \ldots, E^{(b)}$ act as exactly $2^{m-1}$
channel errors, each in a different check.

Since $d_{\min} = (m+2)2^{m-1}$, $t = \left\lceil \frac{d_{\min}-1}{2} \right\rceil = n2^{m-2} + 2^{m-1}-1$
errors can be corrected with no previous decoding
errors. With previous decoding errors only
$2^{m-1}$ channel errors or less are guaranteed
to be corrected.

References for Deterministic Decoders

2. J.P. Robinson, "Error Propagation and Definite
   Decoding of Recurrent Codes," Digital Systems Lab-
   Th., 17-12, pp. 132-134, April 1966.
5. P.D. Sullivan, "Control of Error Propagation in
   Convolutional Codes," Ph.D. Thesis, U. of Notre Dame,
   Dept of E.E., June 1966.
6. Massey, "Advances in Threshold Decoding," in
   Advances in Communication Systems (Editor
Table 12.2. Self-orthogonal Codes.

<table>
<thead>
<tr>
<th>( R = 1/2 )</th>
<th>( R = 2/3 )</th>
<th>( R = 1/4 )</th>
<th>( R = 3/4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n )</td>
<td>( d )</td>
<td>( n )</td>
<td>( d )</td>
</tr>
<tr>
<td>( n_0 = 2 )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>(0, 1)</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>14</td>
<td>(0, 2, 5, 6)</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>36</td>
<td>(0, 2, 7, 13, 16, 17)</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>72</td>
<td>(0, 7, 10, 16, 30, 31, 35)</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>112</td>
<td>(0, 2, 14, 21, 29, 32, 45, 49, 54, 55)</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>172</td>
<td>(0, 2, 6, 24, 29, 40, 43, 55, 68, 75, 76, 85)</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>256</td>
<td>(0, 5, 28, 38, 41, 49, 50, 66, 75, 92, 107, 121, 123, 127)</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>360</td>
<td>(0, 6, 19, 40, 58, 67, 78, 83, 109, 132, 133, 162, 105, 169, 177, 179)</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>434</td>
<td>(0, 2, 10, 22, 53, 56, 82, 83, 89, 98, 130, 148, 153, 167, 188, 192, 205, 216)</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>568</td>
<td>(0, 24, 30, 43, 55, 71, 75, 89, 104, 125, 127, 162, 167, 189, 206, 215, 272, 275, 262, 263)</td>
<td></td>
</tr>
<tr>
<td>( n_0 = 3 )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>9</td>
<td>(0, 1)</td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td>42</td>
<td>(0, 8, 9, 12)</td>
</tr>
<tr>
<td>13</td>
<td>7</td>
<td>123</td>
<td>(0, 2, 6, 24, 29)</td>
</tr>
<tr>
<td>17</td>
<td>9</td>
<td>271</td>
<td>(0, 3, 15, 28, 36, 38)</td>
</tr>
<tr>
<td>21</td>
<td>11</td>
<td>355</td>
<td>(0, 1, 6, 25, 32, 72, 100, 108, 120, 130)</td>
</tr>
<tr>
<td>( n_0 = 4 )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>16</td>
<td>(0, 1)</td>
</tr>
<tr>
<td>13</td>
<td>5</td>
<td>80</td>
<td>(0, 2, 3, 15, 19)</td>
</tr>
<tr>
<td>19</td>
<td>7</td>
<td>272</td>
<td>(0, 5, 15, 34, 35, 42)</td>
</tr>
<tr>
<td>( n_0 = 5 )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>16</td>
<td>(0, 1)</td>
</tr>
<tr>
<td>13</td>
<td>5</td>
<td>80</td>
<td>(0, 2, 3, 15, 19)</td>
</tr>
<tr>
<td>19</td>
<td>7</td>
<td>272</td>
<td>(0, 5, 15, 34, 35, 42)</td>
</tr>
</tbody>
</table>

The problem of correcting errors in codes is not as well understood as the basic properties of codes; (2) decoding is simpler in classes of codes having equipment required is; which are capable of correction. As with block codes, codes are less well understood; optimal burst-correcting codes, block codes in that. Because is, it is much more difficult to analyze for the probability of a burst decoding error tends to become as well. That is, for most of the practical cases and consequently informal codes were mentioned; a useful and powerful technique key and low erorr rate must gh not as well developed and do, are competitive with the important parameters of convolutional codes. The fol-

sets of integers which specify the acting ability of the code.
Table 12.2 Self-orthogonal Codes.  

<table>
<thead>
<tr>
<th>$n_0 = 4$ (continued)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>$R = 1/4$</th>
<th>$R = 3/4$</th>
<th>$n$</th>
<th>Sets of integers that specify code</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d$</td>
<td>$d$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>9</td>
<td>520</td>
<td>(0, 9, 33, 37, 33, 97, 122, 129)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0, 11, 13, 23, 62, 76, 79, 123)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0, 19, 35, 50, 71, 77, 117, 125)</td>
</tr>
</tbody>
</table>

$n_0 = 5$

<table>
<thead>
<tr>
<th>$R = 1/5$</th>
<th>$R = 4/5$</th>
<th>$n$</th>
<th>Sets of integers that specify code</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d$</td>
<td>$d$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>25</td>
<td>(0, 4, 1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0, 2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0, 3)</td>
</tr>
<tr>
<td>17</td>
<td>5</td>
<td>145</td>
<td>(0, 8, 17, 22)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0, 7, 19, 23)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0, 3, 28)</td>
</tr>
<tr>
<td>25</td>
<td>7</td>
<td>490</td>
<td>(0, 9, 33, 37, 38, 97)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0, 14, 17, 61, 87, 95)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0, 39, 39, 51, 62, 65)</td>
</tr>
</tbody>
</table>

This table is taken from the more complete version given in Robinson and Bernstein. For a discussion of the codes tabulated here, see Sec. 12.2.2.

Table 12.3 lists orthogonalizable codes of efficiency 1/2 and 1/3 for various values of minimum distance. It is taken from Table II of Ref. [2] where codes of efficiency 1/5 and 1/10 are also listed. For a discussion of these codes see Sec. 12.2.3.

Along with the usual parameters of $d$ and $n$, the basic parity-check matrix $h$ of each code is given. To save space it is given in the following form. Each row of the $(n_0 - 1)$-row matrix is given on a set of integers corresponding to the blocks in which a nonzero term appears in that row. The $n_0 - 1$ sets are distinguished by an exponent corresponding to the row number. Thus the matrix

$$h = \begin{bmatrix}
000 & 000 & 000 & 100 & 110 \\
100 & 100 & 100 & 000 & 101
\end{bmatrix}$$

is represented as $h$ and is

The inform.
codes, that is, check sums (t table. Each $s$
parity-check r
of the orthog.
notation rules for

* By permission of J. L. Massey.
## TABLE I

<table>
<thead>
<tr>
<th>r</th>
<th>J</th>
<th>Number of Elements</th>
<th>Maximum Element</th>
<th>First Rows of Triangles</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>5</td>
<td>30</td>
<td>36</td>
<td>(3, 10, 11, 12)(15, 14, 2, 4)(7, 18, 1, 8)</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>40</td>
<td>50</td>
<td>(14, 20, 13, 3)(30, 5, 6, 4)(18, 7, 19, 2)(8, 9, 22, 1)</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>(1)(2)(3)(4)(5)</td>
</tr>
<tr>
<td>3</td>
<td>15</td>
<td>15</td>
<td>32</td>
<td>(1, 7, 6)(3, 10, 4)(9)(2, 13)</td>
</tr>
<tr>
<td>4</td>
<td>30</td>
<td>32</td>
<td>63</td>
<td>(1, 14, 6)(4, 8, 13)(7, 2, 18)(5, 6, 17)(10, 19, 3)</td>
</tr>
<tr>
<td>5</td>
<td>50</td>
<td>63</td>
<td>104</td>
<td>(1, 21, 6, 3)(3, 9, 10)(31)(4, 31, 17)(2, 26, 29, 12)(8, 26, 5, 13)</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>(1)(2)(3)(4)(5)(6)</td>
</tr>
<tr>
<td>3</td>
<td>18</td>
<td>19</td>
<td>40</td>
<td>(3, 16, 12)(20, 10, 8)(22, 4, 13)(1, 14, 21)(29, 5, 6)(23, 2, 7)</td>
</tr>
<tr>
<td>4</td>
<td>36</td>
<td>40</td>
<td>76</td>
<td>(27, 29, 5, 6)(24, 22, 4, 13)(20, 10, 8, 33)(1, 14, 21, 37)(13, 23, 2, 7)(43, 3, 16, 12)</td>
</tr>
<tr>
<td>5</td>
<td>60</td>
<td>76</td>
<td>125</td>
<td>(42, 43, 6, 16)(49, 43, 25, 2, 7)(1, 14, 21, 37, 47)(20, 10, 8, 33, 53)(53, 27, 29, 5, 6)</td>
</tr>
<tr>
<td>6</td>
<td>90</td>
<td>125</td>
<td>146</td>
<td>(29, 24, 22, 4, 13)</td>
</tr>
</tbody>
</table>

* Only the first row of each triangle is given.

\[ \begin{align*}
\alpha_1 &= \beta_1 - \alpha_2 \\
\beta_2 &= \alpha_1 \\
\gamma_1 &= \beta_1 - \gamma_2
\end{align*} \]

The first column of each triangle is the largest number in the triangle.)

**Further Results on the Synchronization of Binary Cyclic Codes**

Abstract—It is shown that certain cost codes derived from binary cyclic codes can determine the magnitude of a synchronization error, as well as its direction by examining only the syndrome of the received n-tuple. For such cost codes, therefore, the need for a search procedure to recover synchronization is eliminated. In addition, the range of slip that can be detected and corrected for noisy channels is extended.

**Introduction**

The problem of loss of synchronization, or slip, for binary cyclic codes has recently attracted interest [1]-[3]. One of the techniques for obtaining codes capable of detecting and correcting synchronization error is to form a cost code [1]-[4] from a given (n, k) binary cyclic code by adding a fixed polynomial to each code word before transmission. In a previous paper [4], the ability of such cost codes to detect and correct synchronization error was examined by using the vector-matrix representation of cyclic codes. This correspondence derives some new results on the same problem by using the polynomial representation of cyclic codes as used in [3]. In particular, it is shown that there exist (n, k) cost codes that can determine both the magnitude and direction of the slip by examining only the syndrome of the received n-tuple. Hence, unlike previous procedures for cost codes [3], [4], the present method does not require any search procedure. In passing, it may be mentioned that some other techniques also possess this feature [5], [6].

Except for some changes in notation, the formulation of the polynomial approach to slip will follow Tung [6]. Fig. 1 illustrates a left slip of 4 digits, where \( \Lambda(x) \), \( B(x) \), and \( C(x) \) are any three consecutive cost code words. If there were no slip, the receiver...