J. E. Meggitt

Error Correcting Codes for Correcting Bursts of Errors

Abstract: It is observed that the codes of Abramson, Melas and others are essentially described by the characteristic equation that a certain matrix satisfies. Consequently it is found that transformations of these codes are possible provided that the characteristic equation is preserved. These transformations may then be exploited to produce codes that have a simple implementation and, in fact, a general method is indicated by which any code may be implemented when the characteristic equation is known.

In data transmission systems that are subject to noise, it is found that errors do not occur randomly but in bursts. Consequently, much interest has lately centered on the problem of constructing suitable error correcting codes.

In most of the codes described so far, difficulties have occurred with their implementation, because different courses of action have to be followed by their decoders, depending on the nature of the error burst that is detected, and this has made them expensive in equipment. The main purpose of the present paper is to show how some of these codes may be transformed to make their implementation simple.

Fed-back shifting registers, whose logic contains only modulo two adders, are conveniently described in terms of matrices.[1] If the contents of a shifting register are represented by the k by 1 vector $\mathbf{x}$, it is convenient to denote the contents after a shift by

$$\mathbf{Tx}\,, \tag{1}$$

where $\mathbf{T}$ is a k by k matrix, all of whose elements are zero or one.

The behavior of such a shifting register is determined by the characteristic equation of degree k, that $\mathbf{T}$ satisfies

$$F(\mathbf{T})=0\,. \tag{2}$$

By a suitable choice of $F(\mathbf{T})$, it is possible to make the successive contents of the register take 2^k-1 different values. Such a characteristic equation will be denoted in this paper by

$$M(k\mathbf{T})=0\,. \tag{3}$$

For a good discussion of these ideas, the reader should see Reference 1.

It happens that codes for correcting bursts of errors may be constructed in terms of these matrices $\mathbf{T}$.

Application to codes

In the class of codes to be considered, data is transmitted in blocks of n binary digits. In each block there are $n-k$ information digits and k check digits and, as each block is received, error correction is carried out.

If the digits in the block are $a_1 \cdots a_n$, then the class of codes to be considered is defined by the equation

$$a_1\mathbf{x}+a_2\mathbf{Tx}+a_3\mathbf{T}^2\mathbf{x}+\cdots+a_n\mathbf{T}^{n-1}\mathbf{x}=0\,, \tag{4}$$

where $\mathbf{T}$ is a matrix such as has just been described, and n is such that $\mathbf{T}^n=1$. These k linear equations define k of the a's in terms of the other $n-k$, and these are taken to be the check digits.[2]

To correct such a message, the error vector $\mathbf{z}$ is calculated, where

$$\mathbf{z}=a_1'\mathbf{x}+a_2'\mathbf{Tx}+a_3'\mathbf{T}^2\mathbf{x}+\cdots+a_n'\mathbf{T}^{n-1}\mathbf{x}\,, \tag{5}$$

and where the a''s are the received digits. The various mistakes that the code is required to correct are arranged to produce different error vectors $\mathbf{z}$, so that an examination of $\mathbf{z}$ gives sufficient information for the correction of the message.

Transformation of codes

When a code of the form (4) has been designed (which is done by the choice of a suitable matrix $\mathbf{T}$), it may be found that it is a difficult code to implement. It is desirable then to transform the code so that the rules for implementing it are simpler, and yet to retain its fundamental structure. This may be done as follows:

Let $\mathbf{S}$ be a matrix with an inverse. Then from (4),

$$a_1\mathbf{Sx}+a_2(\mathbf{STS}^{-1})\mathbf{Sx}+a_3(\mathbf{STS}^{-1})^2\mathbf{Sx}+\cdots+a_n(\mathbf{STS}^{-1})^{n-1}\mathbf{Sx}=0\,. \tag{6}$$

If $\mathbf{U}$ is written

$$\mathbf{U}=\mathbf{STS}^{-1}, \tag{7}$$

and

$$y=\mathbf{Sx}, \tag{8}$$

then

$$a_1\mathbf{y}+a_2\mathbf{U}\mathbf{y}+a_3\mathbf{U}^2\mathbf{y}+\cdots+a_n\mathbf{U}^{n-1}\mathbf{y}=0. \tag{9}$$

These k equations define a code with different rules for the formation of the check digits from the previous one. However, a mistake with the first code that leads to an error vector $\mathbf{z}$, leads with the new code to an error vector $\mathbf{Sz}$. Since $\mathbf{S}$ is non-singular, the distinct vectors $\mathbf{z}$ correspond to distinct vectors $\mathbf{Sz}$. Thus the new code with $\mathbf{T}$ replaced by $\mathbf{U}$ corrects exactly the same mistakes as the old one.

For any particular code it is possible to search for a convenient matrix $\mathbf{S}$. However, this is a laborious procedure, so a different approach is adopted. It is observed that the matrix $\mathbf{T}$ has a characteristic equation that $\mathbf{T}$ itself must satisfy. Thus

$$F(\mathbf{T})=0, \tag{10}$$

so $$\mathbf{S}F(\mathbf{T})\mathbf{S}^{-1}=0,$$

and since F is a polynomial in $\mathbf{T}$,

$$F(\mathbf{U})=0. \tag{11}$$

Thus the characteristic equation is invariant under the transformation.

It may be shown that the converse is true under the condition that the minimal polynomials for $\mathbf{U}$ and $\mathbf{T}$ are each identical with their characteristic polynomials. This means that if $\mathbf{U}$ and $\mathbf{T}$ are two matrices which have the same characteristic equation, and which satisfy the condition about minimal polynomials, they are related by

$$\mathbf{U}=\mathbf{STS}^{-1}. \tag{12}$$

It follows that if these matrices are used to produce codes, the codes produced must have identical properties. It is then possible to assert that the *properties of a code of the form (4) are determined essentially by the characteristic equation that* $\mathbf{T}$ *satisfies* (provided that the minimal polynomial for $\mathbf{T}$ is the same as the characteristic polynomial, a condition that is usually satisfied by matrices that produce useful codes).

This has two implications. The first is that when searching for new codes systematically it is not necessary to search through the set of all matrices $\mathbf{T}$, but only through the set of possible characteristic equations. The second is that this provides a ready means for simplifying existing codes. The procedure is to find the characteristic equation that an existing matrix $\mathbf{T}$ satisfies and then to choose a matrix $\mathbf{U}$ which is simpler than $\mathbf{T}$, but which satisfies the same equation. It will be explained how this is done. The result is the same as if $\mathbf{T}$ had been transformed explicitly by finding a suitable matrix $\mathbf{S}$.

Examples of known error correcting codes and their associated characteristic equations

• *Example 1*

If $\mathbf{T}$ is chosen to satisfy

$$M(k\mathbf{T})=0, \tag{13}$$

then the codes defined by (4) are such that single-error correction is possible and the block length, $n=2^k-1$. This follows because if say the r^{th} digit is received wrongly,

$$\mathbf{z}=\mathbf{T}^{r-1}\mathbf{x}, \tag{14}$$

and by definition the vectors $\mathbf{T}^{r-1}\mathbf{x}$ are all different for $r=1, 2\cdots n$, so that the value of r may be found and the mistake rectified.

This gives rise to a set of codes first noted by Abramson.[3] The relevant characteristic equation is simply (13).

• *Example 2*

If to the k equations defining the codes 1, is added a total parity check, then it can be seen that double adjacent error correction is also possible, as was also noted by Abramson.

The idea is that the total parity check equation specifies whether there has been a single or double error. If there has been a single error, correction is carried out as in Example 1. If on the other hand there has been a double error in digits a_r and a_{r+1}, then the error vector is

$$\mathbf{z}=\mathbf{T}^{r-1}(1+\mathbf{T})\mathbf{x}$$

$$=\mathbf{T}^{r-1}\mathbf{y}, \quad \text{where } \mathbf{y}=(1+\mathbf{T})\mathbf{x}. \tag{15}$$

As before, since the vectors $\mathbf{T}^{r-1}\mathbf{y}$ are all different, the value of r may be found.

In this case the code is based upon a $(k+1)$ *by* $(k+1)$ matrix

$$\mathbf{T}=\begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_1 \end{bmatrix}, \tag{16}$$

where $\mathbf{T}_1$ is a k by k matrix satisfying $M(k\mathbf{T}_1)=0$. The block length is $n=2^k-1$ while there are now $(k+1)$ check digits.

The characteristic equation that $\mathbf{T}$ satisfies is the product of the two characteristic equations that the two diagonal parts of the matrix $\mathbf{T}$ satisfy separately. Thus the characteristic equation is

$$(\mathbf{T}+1)M(k\mathbf{T})=0. \tag{17}$$

• *Example 3*

The codes 2 may in turn be extended as has been done by Melas.[4] In this case the (k_1+k_2) by (k_1+k_2) matrix $\mathbf{T}$ is used, where

$$\mathbf{T}=\begin{bmatrix} \mathbf{T}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_2 \end{bmatrix}, \tag{18}$$

and $\mathbf{T}_1$ satisfies

$$M(k_1\mathbf{T}_1)=0\,,$$

while $\mathbf{T}_2$ satisfies

$$M(k_2\mathbf{T}_2)=0\,,$$

and k_1 and k_2 are chosen so that $2^{k_2}-1$ is a factor of $2^{k_1}-1$. The idea is to use two separate sets of check equations, each of the same form as Example 1. To see exactly how this works, the reader should see Reference 4, but it is clearly not surprising that such a code should be capable of correcting bursts of errors of various kinds. It is seen that there are $k=k_1+k_2$ check digits in the code, while the block length is $n=2^{k_1}-1$.

The characteristic equation that $\mathbf{T}$ satisfies is

$$M(k_1\mathbf{T})M(k_2\mathbf{T})=0\,. \tag{19}$$

• *Example 4*

A further group of codes is based on the (k_1+k_2) by (k_1+k_2) matrix $\mathbf{T}$, where

$$\mathbf{T}=\begin{bmatrix}\mathbf{T}_1 & \mathbf{0}\\ \mathbf{0} & \mathbf{T}_2\end{bmatrix}, \tag{20}$$

and where $\mathbf{T}_2$ satisfies $M(k_2\mathbf{T}_2)=0$, while $\mathbf{T}_1$ is the k_1 by k_1 matrix

$$\mathbf{T}_1=\begin{bmatrix}0&0&0&\cdot&\cdot&\cdot&\cdot&1\\1&0&0&\cdot&\cdot&\cdot&\cdot&0\\0&1&0&\cdot&\cdot&\cdot&\cdot&0\\\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot\\\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot\\0&0&0&\cdot&\cdot&\cdot&1&0\end{bmatrix}. \tag{21}$$

The code then becomes explicitly

$$a_1\begin{bmatrix}1\\0\\0\\\cdot\\\cdot\\\cdot\\\cdot\\\hline\mathbf{x}\end{bmatrix}+a_2\begin{bmatrix}0\\1\\0\\\cdot\\\cdot\\\cdot\\\cdot\\\hline\mathbf{T}_2\mathbf{x}\end{bmatrix}+a_3\begin{bmatrix}0\\0\\1\\\cdot\\\cdot\\\cdot\\\cdot\\\hline\mathbf{T}_2^2\mathbf{x}\end{bmatrix}+\cdots\cdots=0\,. \tag{22}$$

Codes of this form have been described by Fire,[5] and these too are good for correcting bursts of errors of various kinds. It is necessary for k_1 to be chosen to be prime to $2^{k_2}-1$. It is then found that $k=k_1+k_2$ and $n=k_1(2^{k_2}-1)$.

The idea behind these codes is that the first k_1 equations should be capable of detecting the kind of error burst which is to be corrected, while the other k_2 should define the position of the burst. Again the reader should see Reference 5 for details. $\mathbf{T}_1$ satisfies the characteristic equation

$$\mathbf{T}_1^{k_1}+1=0\,. \tag{23}$$

Thus $\mathbf{T}$ satisfies the characteristic equation

$$(\mathbf{T}^{k_1}+1)M(k_2\mathbf{T})=0\,. \tag{24}$$

It will be seen that the characteristic equations which essentially define a number of important codes may easily be obtained explicitly. It will be shown now how these codes may be simplified by transformations of the type that have been suggested.

Form of transformed codes

There exists a simple method of implementing any code whose matrix $\mathbf{T}$ satisfies a known characteristic equation (providing the minimal polynomial for $\mathbf{T}$ coincides with the characteristic polynomial) and this will now be described. It is valuable because it provides ways of implementing existing codes such as those of Abramson, Melas and Fire, and because it provides a general way of implementing any other code of the form (4) which may be found.

The general characteristic equation has the form

$$\mathbf{T}^k+C_{k-1}\mathbf{T}^{k-1}+C_{k-2}\mathbf{T}^{k-2}+\cdots\cdots+C_1\mathbf{T}+C_0=0\,, \tag{25}$$

where the C's are zero or one. A convenient matrix that satisfies this equation is the associated matrix

$$\mathbf{T}=\begin{bmatrix}C_{k-1}&C_{k-2}&C_{k-3}\cdots C_1&C_0\\1&0&0&0&0\\0&1&0&0&0\\0&0&1&0&0\\\cdot&\cdot&\cdot&\cdot&\cdot\\\cdot&\cdot&\cdot&\cdot&\cdot\\\cdot&\cdot&\cdot&1&0\end{bmatrix}. \tag{26}$$

This matrix has the property that its minimal polynomial is the same as its characteristic polynomial.

With this $\mathbf{T}$, the vector $\mathbf{x}$ of equation (4) may be chosen arbitrarily provided it is such that

$$G(\mathbf{T})\mathbf{x}\neq 0 \tag{27}$$

for any polynomial G of degree less than k. (If this condition is violated, the resulting code is the same as that produced by an r by r matrix $\mathbf{T}^*$ satisfying the characteristic equation of degree r say, $(r<k)$

$$G(\mathbf{T}^*)=0\,,$$

and such a code could be constructed using only r check digits instead of k).

If $\mathbf{x}$ is chosen to be

$$\mathbf{x}=\begin{bmatrix}1\\0\\0\\\cdot\\\cdot\\\cdot\\0\end{bmatrix}, \tag{28}$$

then repeated multiplication by the matrix $\mathbf{T}$ of (26), shows that the vectors $\mathbf{x}$, $\mathbf{Tx}\cdots\mathbf{T}^{k-1}\mathbf{x}$ are linearly independent so that with this $\mathbf{x}$, the condition (27) is bound to be satisfied. Consequently $\mathbf{x}$ is always taken to have this form.

• *Example*

Consider the Abramson code for correcting single and double adjacent errors, that has block length $n=7$.

It has been shown in equation (17) that the appropriate characteristic equation is

$$(\mathbf{T}+1)M(3\mathbf{T})=0\,. \tag{29}$$

A possible form for $M(3\mathbf{T})$ is

$$M(3\mathbf{T})=\mathbf{T}^3+\mathbf{T}+1\,. \tag{30}$$

Thus the characteristic equation for $\mathbf{T}$ is

$$\mathbf{T}^4+\mathbf{T}^3+\mathbf{T}^2+1=0\,. \tag{31}$$

Thus from (26) $\mathbf{T}$ is taken to have the form

$$\mathbf{T}=\begin{bmatrix}1&1&0&1\\1&0&0&0\\0&1&0&0\\0&0&1&0\end{bmatrix}. \tag{32}$$

Using (28) the coding equations then become

$$a_1\begin{bmatrix}1\\0\\0\\0\end{bmatrix}+a_2\begin{bmatrix}1\\1\\0\\0\end{bmatrix}+a_3\begin{bmatrix}0\\1\\1\\0\end{bmatrix}+a_4\begin{bmatrix}1\\0\\1\\1\end{bmatrix}+a_5\begin{bmatrix}0\\1\\0\\1\end{bmatrix}+a_6\begin{bmatrix}0\\0\\1\\0\end{bmatrix}+a_7\begin{bmatrix}0\\0\\0\\1\end{bmatrix}=0\,. \tag{33}$$

This is just a transformation of Abramson's form for this code which is of course

$$a_1\begin{bmatrix}1\\1\\0\\0\end{bmatrix}+a_2\begin{bmatrix}1\\0\\1\\0\end{bmatrix}+a_3\begin{bmatrix}1\\1\\0\\1\end{bmatrix}+a_4\begin{bmatrix}1\\1\\1\\0\end{bmatrix}+a_5\begin{bmatrix}1\\1\\1\\1\end{bmatrix}+a_6\begin{bmatrix}1\\0\\1\\1\end{bmatrix}+a_7\begin{bmatrix}1\\0\\0\\1\end{bmatrix}=0\,. \tag{34}$$

It will be seen that there is much more symmetry in the new code (33) and this is exploited in its implementation which is considered next. The power of the transformation is best appreciated when the method is applied to a more complicated code. For brevity this is left to the reader.

Implementation

For brevity too, the encoding and decoding apparatus that is in general required will be described only for the particular case of the code (33). However, the method is absolutely general and the extensions will be clear.

When the message is transmitted, digits will be sent serially in order, starting with a_1. The last four digits (in general the last k) are chosen to be the check digits and from (33) it is seen that they are defined by

$$\begin{aligned}a_4&=a_1+a_2 & a_5&=a_2+a_3\\ a_6&=a_3+a_4 & a_7&=a_4+a_5\,.\end{aligned} \tag{35}$$

It is observed that each equation has the form of the previous one, with the digit labels increased by one, and that each check digit is defined in terms of only the earlier digits. This is a general property of codes defined by (26) and (28) as may be shown as follows.

Let the first of the k check equations (4) be

$$\sum_{i=1}^{n} a_i b_i=0\,. \tag{36}$$

Then from the form of $\mathbf{T}$ (26), it can be seen that (4) may be written out in full as

$$a_1\begin{bmatrix}b_1\\b_n\\b_{n-1}\\\cdot\\\cdot\\\cdot\\b_{n-k+2}\end{bmatrix}+a_2\begin{bmatrix}b_2\\b_1\\b_n\\\cdot\\\cdot\\\cdot\\b_{n-k+3}\end{bmatrix}+a_3\begin{bmatrix}b_3\\b_2\\b_1\\\cdot\\\cdot\\\cdot\\b_{n-k+4}\end{bmatrix}+\cdots\cdots+a_n\begin{bmatrix}b_n\\b_{n-1}\\b_{n-2}\\\cdot\\\cdot\\\cdot\\b_{n-k+1}\end{bmatrix}=0\,. \tag{37}$$

By construction

$$\mathbf{x}=\begin{bmatrix}b_1\\b_n\\b_{n-1}\\\cdot\\\cdot\\\cdot\\b_{n-k+2}\end{bmatrix}=\begin{bmatrix}1\\0\\0\\\cdot\\\cdot\\\cdot\\0\end{bmatrix}. \tag{38}$$

Thus $b_{n-k+2}=b_{n-k+3}=\cdots\cdots b_n=0$, while since the last vector in (37) cannot be 0, $b_{n-k+1}=1$.

The first check equation (36) therefore reduces to

$$\sum_{i=1}^{n-k} a_i b_i = a_{n-k+1} .$$

The second becomes

$$\sum_{i=1}^{n-k} a_{i+1} b_i = a_{n-k+2}$$

$$\cdots\cdots\cdots$$

$$\cdots\cdots\cdots \qquad (39)$$

The k^{th} becomes

$$\sum_{i=1}^{n-k} a_{i+k-1} b_i = a_n ,$$

and this proves the assertion.

Encoder

An encoder that imposes the conditions (35) is shown in Fig. 1. The squares indicate a shifting register. Information flows into the register at times 1, 2, 3 as shown by the timing pulses applied to the input gate, while data is fed back at times 4, 5, 6, 7, when the check digits are formed. The configuration is shown at time 4. At time 5 the first digit a_1 is transmitted, a_2 and a_3 move one place right, while the check digit $a_4=a_1+a_2$ is placed in the position occupied by a_3 at time 4. This process continues as all the check digits are formed.

An alternative design uses a shifting register that takes values $\mathbf{T}^r\mathbf{x}(r=0, 1, 2\cdots)$ in turn, and this is used to steer information into four check digit stores in a fairly obvious way.

The general design of the encoder of Fig. 1 imposes the conditions (39). This is done by using a shifting register of length $n-k$, the feed back connections to which are determined by the b's of Eq. 37.

Decoder

A decoder is shown in Fig. 2. The operation of the decoder is first to calculate the error vector.

$$\begin{aligned} \mathbf{z}_1 &= a_1+a_2+a_4 \\ \mathbf{z}_2 &= a_2+a_3+a_5 \\ \mathbf{z}_3 &= a_3+a_4+a_6 \\ \mathbf{z}_4 &= a_4+a_5+a_7 \end{aligned} \qquad (40)$$

The values of the $\mathbf{z}$'s indicate the nature and position of errors. The received message is fed into a shifting register and the adder A forms the sums $\mathbf{z}$ automatically at times 4, 5, 6, 7, and they are fed to a second shifting register B. The decoder is shown at time 8 (which is the same as time 1, times being measured from the arrival of the first digit) when all the transmitted digits occupy the main shifting register, while the four $\mathbf{z}$'s occupy the shifting register B. Henceforth until time 4, the register B is fed back so that its contents are continually changing, while simultaneously information leaves the main register. As this happens the detection circuit C, looks for coincidences and when coincidences are found, the digit currently being output from the end of the main register is inverted and corrected. Simultaneously register B is altered to indicate that the burst pattern it is now required to correct is a simpler one, since the first digit of it has already been corrected.

Theory of decoder

The role of the fed-back shifting register B is to operate repeatedly on its content $\mathbf{z}$ with $\mathbf{T}^{-1}$. The connections to it are in fact determined by the C's in equation (26). The detection circuit is arranged to detect the vectors $\mathbf{x}$ and $(1+\mathbf{T})\mathbf{x}$ which have the form shown in Fig. 2.

Thus if the r^{th} digit alone is wrong,

$$\mathbf{z}=\mathbf{T}^{r-1}\mathbf{x} , \qquad (41)$$

and coincidence will be detected after $(r-1)$ shifts. However, at this time the r^{th} digit is currently being output and so it is inverted as is required. The detection circuit also adds $\mathbf{x}$ to the present contents ($\mathbf{x}$ in this case) of the register B, so it now contains zero and no further correction will take place.

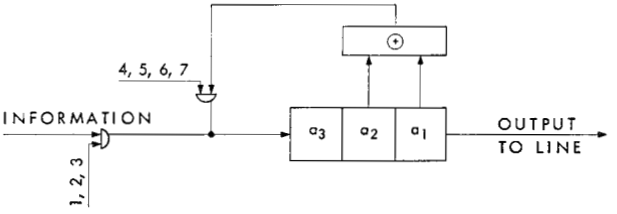

Figure 1 **Configuration required for an encoder.**

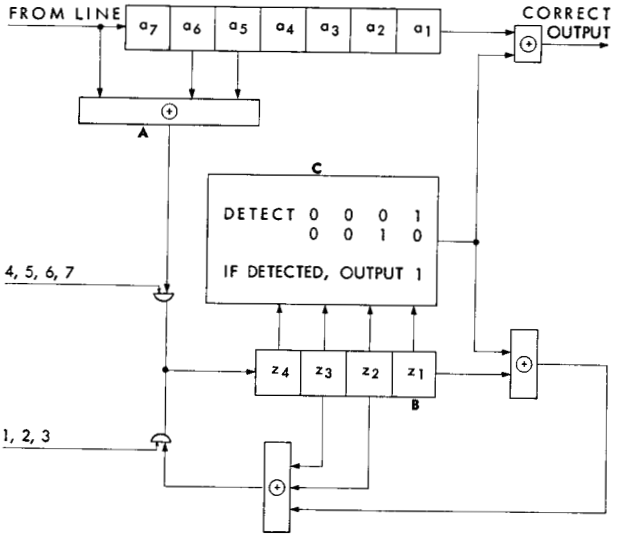

Figure 2 **Configuration required for a decoder.**

For a double adjacent error, when the r^{th} and $(r+1)^{\text{th}}$ digits are both wrong

$$\mathbf{z}=\mathbf{T}^{r-1}(1+\mathbf{T})\mathbf{x}\,, \tag{42}$$

and again coincidence will be detected after $(r-1)$ shifts and the r^{th} digit corrected. The detection circuit adds $\mathbf{x}$ to the present contents (now $(1+\mathbf{T})\mathbf{x}$) of register B, so that it now contains $\mathbf{Tx}$. This leads to the detection of $\mathbf{x}$ at the next digit time, and the subsequent correction of the $(r+1)^{th}$ digit, as is required.

When the apparatus is designed for a general code, which corrects amongst other things, bursts of the form $q_1q_2\cdot\cdot\cdot\cdot\cdot q_p$, then it is necessary to detect in register B,

$$(q_1+q_2\mathbf{T}+q_3\mathbf{T}^2+\cdot\cdot\cdot\cdot+q_p\mathbf{T}^{p-1})\mathbf{x} \tag{43}$$

and to make provision for the correction of all shorter bursts.

The operation then is merely an extension of the operation just described. The only other change it is necessary to make for a general code, is to alter the lengths of registers and to arrange the correct feed back connections. Simple formulae can be given for these connections in terms of the C's of equation (26).

Conclusion

A simple implementation has been given for any code that is described by a characteristic equation, and it has been observed that many codes can be characterized in this way. Hence this is a powerful approach to the coding problem.

References

1. B. Elspas, "The Theory of Autonomous Linear Sequential Networks," *IRE Trans. on Circuit Theory,* **CT-6,** 45, March 1959.
2. G. E. Sacks, "Multiple Error Correction by Means of Parity Checks," *IRE Trans. on Information Theory,* **IT-4,** 145, December 1958.
3. N. M. Abramson, "A Class of Systematic Codes for Non-Independent Errors," Technical Report No. 51, December 30th, 1958, Stanford Electronics Laboratory, California.
4. C. M. Melas, "A New Group of Codes for Correction of Dependent Errors in Data Transmission," *IBM Journal of Research and Development,* **4,** 58-65, January 1960.
5. P. Fire, "A Class of Multiple-Error-Correcting Binary Codes for Non-Independent Errors," Sylvania Electronic Systems Report, presented at A.I.E.E. Meeting, Chicago, October 1959.

Revised manuscript received January 20, 1960