E. Gorog

Redundant Alphabets with Desirable Frequency Spectrum Properties

Abstract: When alphabets of digital symbols are used to represent information for data processing, storage, and transmission, redundancy in the alphabets is traditionally used for the purpose of error compensation. This paper deals with alphabets of redundant codes, both binary and higher level, where the emphasis is on using redundancy to produce code alphabets with unique properties in their frequency spectra that can be exploited in the design of the system in which they are used.

In particular, techniques are presented for synthesizing alphabets that produce spectral nulls at frequencies $1/kT$, where T is the duration of a word element. Some of the interesting alphabets are a 10-word, 5-bit alphabet with spectrum zero at $1/2T$; a 10-word, 6-bit alphabet with spectrum zero at $1/3T$; a 36-word, 8-bit alphabet with zero at $1/4T$; and a 36-word, 8-bit alphabet with zeros at both 0 and $1/2T$.

Introduction

Traditionally the design of data transmission, multiplexing, and storage systems has been independent of the alphabet selected to represent the information, and redundancy has been used to provide the required protection against certain types of errors that may occur during the transmission or processing of the digital information. In the development of transmission systems, the designer conventionally directs his efforts toward providing equipment that accommodates and concentrates the frequency energy of the information independently of the way it is represented.

In this paper we will describe families of digital alphabets of various sizes that are both redundant (error detecting) and inherently modulating (their signal energy is concentrated into a predetermined range of the frequency spectrum). Since these alphabets can be chosen to exhibit predetermined frequency spectrum envelopes, they can, therefore, match specific requirements in the design of transmission, multiplexing or storage systems. This viewpoint on code construction has implications for the design of systems that process or transmit the coded information. Some of the ways in which the availability of these alphabet families might influence system design are mentioned at the end of the paper.

The alphabet families have been found in a study of the expression for the frequency spectrum of discrete-valued signals. The amplitude-phase form of this expression is

$$S = \frac{1 - e^{-j2\pi fT}}{j4\pi^2 f} \sum_{i=0}^{N-1} a_i e^{-j2\pi fT\,i}, \tag{1}$$

where a_i is an m-ary valued element of an N-element sequence having the form $a_0 a_1 \cdots a_{N-1}$ and T is the duration of a_i. A random sequence of elements will have its first spectrum zero ($S = 0$) at the frequency $1/T$.

Our purpose here will be to show how to construct alphabets of code words that, when transmitted serially, will have spectra with zeros occurring at $f = 1/kT$, where k is an integer or a ratio of relatively prime integers. In most of the cases considered, an expression is derived to calculate the number of n-element words in the alphabet. In presenting the alphabets, we will work from the specific to the general in considering the values of k which define the spectrum zeros. Also, we will begin the discussion by considering only binary alphabets; later we will show the extension to the m-ary case.

First, the condition for obtaining $S = 0$ at $f = 0$ is given. Then the condition for obtaining a zero at $f = 1/2T$ is derived. An example of an alphabet meeting this requirement is shown to consist of ten 5-bit words, where the first lobe of the spectrum envelope occupies 5/8 the bandwidth required for the conventional 4-bit binary decimal code.

Binary alphabets meeting both requirements ($S = 0$ at both $f = 0$ and $f = 1/2T$) are shown next. It is interesting to find how the method of constructing these alphabets differs for the cases in which n, the number of bits per code word, is a multiple of 4, is odd, and is even with $n/2$ odd. A particularly interesting 36-word alphabet of 8-bit words is presented. Called the STEAN code, this alphabet might be useful in the transmission of English alphanumeric data.

The author is located at IBM Corporate Headquarters in Armonk, New York as Technical Advisor to the IBM Chief Scientist.

The first lobe of the spectrum is contained in the frequency range from $f = 0$ to $1/2T$, with the signal energy concentrated around $f = 1/8T, f = 1/4T$ and $f = 3/8T$, as if the data had undergone some kind of modulation.

Next, the conditions are given for constructing alphabets with spectrum zeros at $f = 1/kT$, where k is any integer. Here, the cases where k is a prime number must be treated separately from the others. For $k = 4$ another 36-word alphabet of 8-bit words is shown, the frequency spectrum of which happens to be very close to the signal power spectrum of the public telephone network in the United Kingdom.

Table 1 5-bit decimal code for spectrum zero at $f = 1/2T$.*

0	0 0 0 0 0
1	0 0 0 1 1
2	0 0 1 1 0
3	0 1 0 0 1
4	0 1 1 0 0
5	0 1 1 1 1
6	1 0 0 1 0
7	1 1 0 0 0
8	1 1 0 1 1
9	1 1 1 1 0

* The 0's correspond to the -1's of the analysis.

Binary alphabets having envelope spectrum zero at f = 0

To find the subset of all N-bit sequences which have no dc component, i.e., $S = 0$ at $f = 0$, the condition

$$\sum_{i=0}^{N-1} a_i = 0 \tag{2}$$

must be satisfied.* If $N = \lambda n$, the sequence can be decomposed into λ words of n bits each. Then, a sufficient condition for Eq. (1) to be satisfied is that

$$\sum_{i=(j-1)n}^{jn-1} a_i = 0\,, \qquad j = 1, 2, \cdots, \lambda\,.$$

In its simplest interpretation this expression means that the i-th word of the λ-word sequence must contain the same number of $+1$'s as -1's. Such an interpretation requires, of course, that n, the number of bits per pattern, be even. The sets of patterns that satisfy the above condition for n even are called "$n/2$ out of n" codes; each of these codes contains $C^n_{n/2}$ different patterns, where $C^n_{n/2}$ is the number of combinations of n bits taken $n/2$ at a time. The "4 out of 8" code is one member of this family; it has been used, for example, by IBM in the control of data transmission.

For the case of n odd and λ even the expression can be interpreted to mean that the j-th word must have one less (more) $+1$ than -1 if j is even (odd), and one more (less) $+1$ than -1 if j is odd (even). The sets of words that satisfy this interpretation can be called "$(n - 1)/2$ out of n" codes for j even, and "$(n + 1)/2$ out of n" codes for j odd.

Binary alphabets having envelope spectrum zero at f = 1/2T

Another subset of the N-bit sequences, in which $S = 0$ at $f = 1/2T$, can be found by imposing the condition that

$$\sum_{i=0}^{N-1} a_i(-1)^i = 0\,. \tag{3}$$

* The situation in which $S = 0$ for all f means that the amplitude is zero over the entire spectrum. This can happen if, and only if, there is no signal present. That is, when $a_i = 0$ for all i. Therefore to avoid ambiguity in the present analysis we have chosen to represent the binary values a_i by $+1$ and -1 rather than by 1 and 0.

For words of n bits each, a sufficient condition for Eq. (3) is

$$\sum_{i=(j-1)n}^{jn-1} a_i(-1)^i = 0 \qquad j = 1, 2, \cdots, \lambda\,.$$

The j-th word of a λ-word sequence will satisfy this condition if, for n even, a reversal of the signs of its odd-numbered bits produces a new word having the same number of $+1$'s as -1's. There are $C^n_{n/2}$ different patterns that meet the condition for any even n. A set of patterns that satisfy the condition for the special case of $n = 8$ has been found previously and informally referred to by the author as the "transmission-adapted code."[1]

For n odd and λ even, the condition will be met if the sequence of words is chosen such that any pair of adjacent words has the desired properties. That is, for the j-th word, when there is one more (less) -1 in the odd-(even-) numbered bits than in the even-(odd-)numbered bits, the pair of words, j and $j + 1$, will then have the same number of -1's (and $+1$'s) in their n odd-numbered and n even-numbered bits.

The subset of patterns which meets the condition of Eq. (3) for n odd contains $C^n_{(n-1)/2}$ different words. As an example, $n = 5$ produces 10 words. This suggests that one might use a 5-bit code rather than the conventional 4-bit code to represent the decimal numbers. With such a code the frequency of the first zero of the spectrum is reduced by a factor of 2 as compared to that of all thirty-two 5-bit words, and by a factor of 1.6 as compared to that of the ten 4-bit BCD words (see Fig. 1). Table 1 shows the 5-bit decimal code. The following fortuitous relationship occurs, which should simplify the circuitry for generating the code:

$$d \rightarrow (3d), \qquad (d = 0, 1, 2, 3, 4, 5, 6)$$
$$d \rightarrow [3\,(d + 1)], \qquad (d = 7, 8, 9),$$

where the decimal number d is transformed to the base-2 representation of the number at the right-hand-side of the arrow.

It should be noted that the minimum distance of the code is 2, so that if "bit detection" is used, a single error within a character will always be detected.

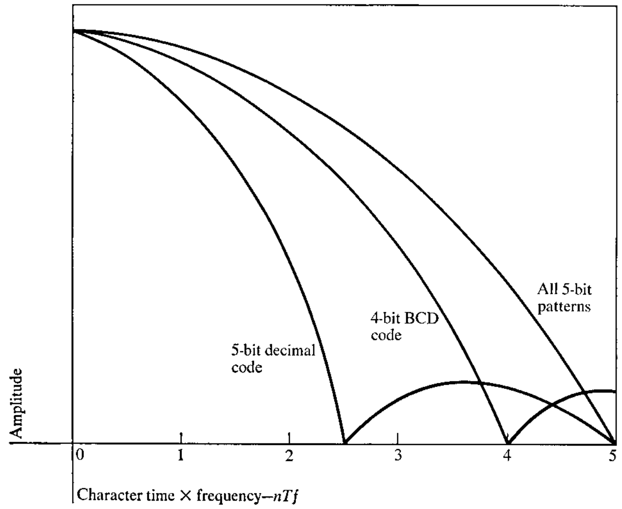


Figure 1 Locations of first zeros of spectra for decimal alphabets.

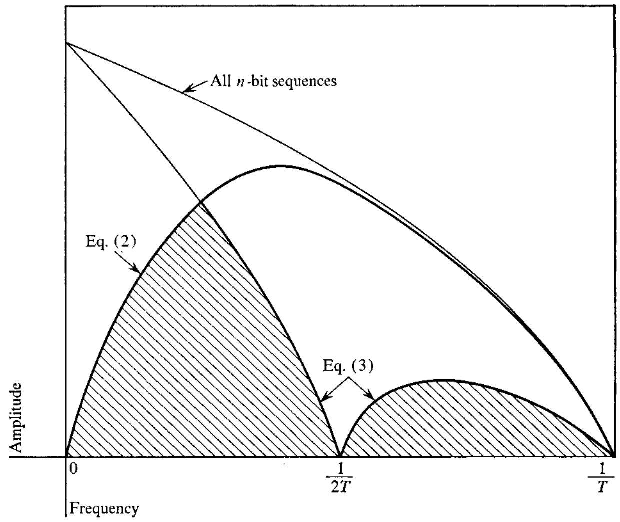


Figure 2 Frequency spectra envelopes for binary sequences satisfying various conditions. Cross-hatched area is bounded by envelope corresponding to new family of patterns.

Binary alphabets having envelope spectrum zeros at both f = 0 and f = 1/2T

• *n is a multiple of 4*

The subset of all N-bit sequences in which $S = 0$ at both $f = 0$ and $f = 1/2T$ is the intersection of two subsets which satisfy the Eqs. (2) and (3), respectively. The conditions for this new subset can also be stated as

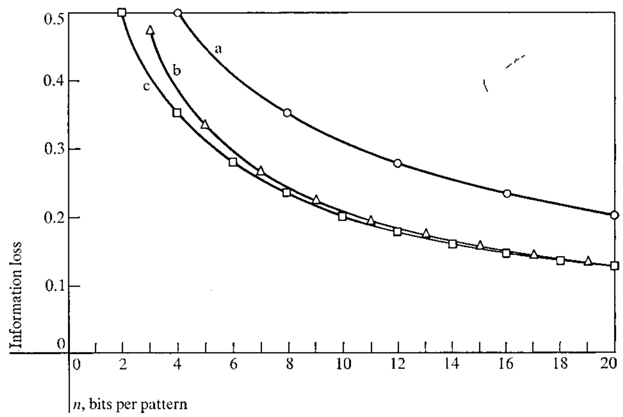
$$\sum_{i=0}^{(N/2)-1} a_{2i} = 0 \tag{4}$$

$$\sum_{i=0}^{(N/2)-1} a_{2i+1} = 0\,. \tag{5}$$

For $N = \lambda n$, where there are λ words of n bits each, conditions (4) and (5) each independently produce $C^{n/2}_{n/4}$ different words so that the subset in which both conditions are met contains $(C^{n/2}_{n/4})^2$ different words. Figure 2 shows the envelopes of the spectra obtained by satisfying Eqs. (1), (2) and (3) independently; the cross-hatched area of the figure is bounded by the envelope of the last new subset.

This new family of binary alphabets is particularly suited to data transmission. The theoretical minimum frequency required for binary data transmission without bit interference is known[2] to be $1/2T$. In the case of the new family of alphabets, the entire first lobe is within the stated limit so that transmission at the rate of $1/nT$ words per second should be achieved practically through the use of non-ideal filters of bandwidth $1/2T$ Hz. The binary elements of the words are transmitted at a rate of $1/T$ bits per second and the "information" transmission rate is $(2/n \log C^{n/2}_{n/4})\ /T$ bit per second. Curves comparing the information loss for the new family with that of other reduced-band families are shown in Fig. 3.

Figure 3 Information loss, $1 - 1/n \log_2 C(n)$, for various families of binary sequences. Curve a is the solution for $C(n) = (C^{n/2}_{n/4})^2$ which gives the number of different patterns in the family that is exemplified by the STEAN code for $n = 8$; curve b is the solution for $C(n) = C^{n}_{(n-1)/2}$ which corresponds to the family that contains the 5-bit decimal code; curve c is the solution for $C(n) = C^{n}_{n/2}$, the number of patterns in the families exemplified by the "4 out of 8" code and the transmission adapted code for $n = 8$.



• *The STEAN code, an alphanumeric frequency concept code alphabet*

A specific application of the new family is given for the case $n = 8$. There are 256 different words of 8-bit length. Of these words 70 satisfy the condition of Eq. (2) ("4 out of 8" codes), and 70 that of Eq. (3) (transmission-adapted codes). The intersection of the two groups of 70 words gives 36 words which satisfy simultaneously the conditions of Eqs. (4) and (5). It is evident that a one-to-one correspondence between the 36 words and the 26 letters and 10 numbers of the English alphanumeric alphabet can be established. This has been done in defining the STEAN code (Simple Transmission of English Alpha-Numeric

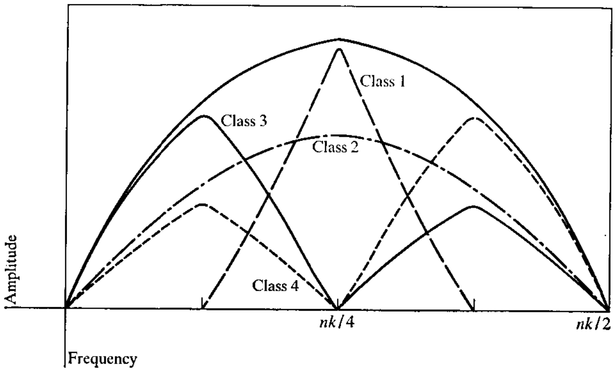


Figure 4 Spectrum classes for new family of binary sequences.

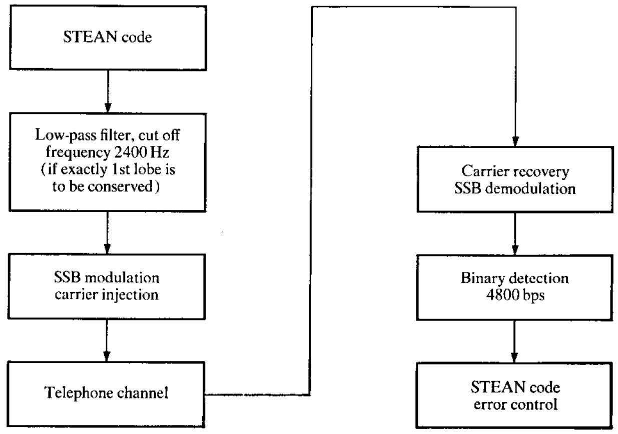


Figure 5 Transmission scheme for STEAN-coded data.

data). Furthermore, a detailed frequency-spectrum analysis of the 36 words reveals that they can be subdivided into the four classes shown in Fig. 4.

Using statistics on letter usage obtained by Seibel,[3] words were assigned to each of the letters and numbers. The words in spectrum classes 1 and 2 were assigned to the most-used letters and to the numbers because the shape of the spectrum is better for these classes than for classes 3 and 4. Table 2 gives the usage statistics and word assignments for the suggested STEAN code; it is expected here that procedure control characters are either eliminated in the system design or (if still needed) consist of very specific combinations of code words.

The error-control capability of this code can also be exploited if "bit-detection" is used. Every error which transforms a word of the STEAN code into one of the 220 words that do not belong to it will be detected. Since Eqs. (4) and (5) are satisfied independently, any burst of 2 errors in a word will be detected.

A simple transmission scheme like that of Fig. 5 could be used to transmit STEAN-coded information. Using the 2.4 kHz bandwidth provided by a telephone line, single-sideband modulation with carrier injection is sufficient to place the data spectrum directly within the telephone channel in such a way that a straightforward 2-level detection scheme, after demodulation by the recovered carrier, permits transmission at a rate of 600 characters per second.

This specific alpha-numeric code and transmission scheme have not yet been tested; however, a series of modems based on the configuration given in Fig. 5 has been successfully implemented.[4] These modems correspond to the case $n = 4$, where the four words,

$$(C_1^{\,2})^2 = 4,$$

0011, 0110, 1001 and 1100,

are precisely the digital binary signal elements corresponding to a four-phase modulation scheme.

Table 2 The STEAN code.

Character	*Binary pattern**	*Relative usage*	*Spectrum class*
0	1 1 0 0 1 1 0 0		1
1	0 1 1 0 0 1 1 0		1
2	0 0 1 1 0 0 1 1		1
3	1 0 0 1 1 0 0 1		1
4	1 1 1 0 0 1 0 0		2
5	0 1 1 1 0 0 1 0		2
6	0 0 1 1 1 0 0 1		2
7	1 0 0 1 1 1 0 0		2
8	0 1 0 0 1 1 1 0		2
9	0 0 1 0 0 1 1 1		2
A	1 0 0 1 0 0 1 1	.088	2
B	1 0 1 0 0 1 0 1	.014	3
C	1 1 0 1 0 0 1 0	.032	3
D	1 1 0 0 1 0 0 1	.041	2
E	0 0 0 1 1 0 1 1	.122	2
F	0 1 1 0 1 0 0 1	.020	3
G	1 0 1 1 0 1 0 0	.020	3
H	1 0 0 0 1 1 0 1	.046	2
I	1 1 0 0 0 1 1 0	.050	2
J	0 1 0 1 1 0 1 0	.002	3
K	0 0 1 0 1 1 0 1	.008	3
L	1 0 0 1 0 1 1 0	.040	3
M	0 1 0 0 1 0 1 1	.022	3
N	0 1 1 0 0 0 1 1	.076	2
O	1 0 1 1 0 0 0 1	.076	2
P	1 1 1 0 0 0 0 1	.020	4
Q	1 1 0 0 0 0 1 1	.001	4
R	1 1 0 1 1 0 0 0	.064	2
S	0 1 1 0 1 1 0 0	.066	2
T	0 0 1 1 0 1 1 0	.088	2
U	1 0 0 0 0 1 1 1	.023	4
V	0 0 0 0 1 1 1 1	.010	4
W	0 0 0 1 1 1 1 0	.018	4
X	0 0 1 1 1 1 0 0	.002	4
Y	0 1 1 1 1 0 0 0	.017	4
Z	1 1 1 1 0 0 0 0	.001	4

* The 0's correspond to −1's of the analysis.

- *n is even but n/2 is odd*

When n is not a multiple of 4, n-bit alphabets producing N bit sequences which will satisfy both (4) and (5) still exist, but their construction requires that all the bits of some words be systematically inverted.

We will briefly show how to construct such alphabets, first in the case where n is even and $n/2$ is odd.

Let us define the j-th word of a λ-word sequence by the following relationships:

$$\sum_{i=0}^{(n/2)-1} a_{2i} = -1 , \tag{6}$$

$$\sum_{i=0}^{(n/2)-1} a_{2i+1} = -1 . \tag{7}$$

Here, we are saying that there is one more -1 than $+1$ in both the odd-numbered bits and the even-numbered bits. Now, if we invert this situation in the $(j+1)$-st word, the following relationships occur:

$$\sum_{i=n/2}^{n-1} a_{2i} = 1 , \tag{8}$$

$$\sum_{i=n/2}^{n-1} a_{2i+1} = 1 , \tag{9}$$

such that for the 2 adjacent words we will get:

$$\sum_{i=0}^{n-1} a_{2i} = 0 , \tag{10}$$

$$\sum_{i=0}^{n-1} a_{2i+1} = 0 . \tag{11}$$

This means that conditions (4) and (5), which assure spectrum zeros at both $f = 0$ and $f = 1/2T$, are satisfied for any message constituted of an even number of words defined by (6) and (7), provided the bit values of every second word are inverted.

It can be noticed that there are $C^{n/2}_{(n-2)/4}$ combinations of bits that will satisfy independently conditions (6) and (7), so that there will be

$$C(n) = [C^{n/2}_{(n-2)/4}]^2$$

different words within this alphabet.

- *n is odd*

In the case where n is odd, let us define the j-th word in a sequence by the following relationships:

$$\sum_{i=0}^{[(n+1)/2]-1} a_{2i} = -1$$

$$\sum_{i=0}^{[(n-1)/2]-1} a_{2i+1} = 0 . \tag{13}$$

This says that there are an equal number of $+1$'s and -1's in the odd-numbered bits, and there is one more -1 than $+1$ in the even-numbered bits. If the same conditions hold for the $(j+1)$-st word, these two consecutive words will be a sequence of $2n$ bits having the relationships:

$$\sum_{i=0}^{n-1} a_{2i} = -1 \tag{14}$$

$$\sum_{i=0}^{n-1} a_{2i+1} = -1 . \tag{15}$$

Relationships (14) and (15) are the same as (6) and (7), except that two n-bit characters are involved instead of one. This means that the basic conditions (4) and (5) are satisfied for any message constituted of a multiple of 4 characters as defined by Eqs. (12) and (13), provided all the bits of every third and fourth character are inverted.

If the number of bits per word is expressed as $n = 4h + r$, ($r = 1, 2, 3, 4$ and $h = 1, 2, 3, \cdots$), the number of words in the alphabets for odd n is given by

$$C(n) = [C^{(n-1)/2}_{(n-1)/4}]\,[C^{(n+1)/2}_{(n-1)/4}] \quad \text{for} \quad r = 1 ,$$

$$C(n) = [C^{(n-1)/2}_{(n+1)/4}]\,[C^{(n+1)/2}_{(n+4)/4}] \quad \text{for} \quad r = 3 .$$

Compare these formulas with the previously derived expressions for even n:

$$C(n) = [C^{n/2}_{n/4}]^2 \quad \text{for} \quad r = 0 ,$$

$$C(n) = [C^{n/2}_{(n-2)/4}]^2 \quad \text{for} \quad r = 2 .$$

Table 3 gives the values of $C(n)$ for word lengths up to 20 bits. The following recurrent formulas for the number of words per alphabet can easily be deduced from the table:

$$C(n+1) = 2C(n) \qquad \text{for} \quad r = 2, 3 .$$

$$C(n+1) = \frac{2h+1}{h+1}\, C(n) \quad \text{for} \quad r = 0, 1 .$$

Table 3 Number, $C(n)$, of characters in an n-bit binary alphabet having spectrum zeros at both $f = 0$ and $f = 1/2T$.

n	$C(n)$
4	4
5	6
6	9
7	18
8	36
9	60
10	100
11	200
12	400
13	700
14	1225
15	2450
16	4900
17	8820
18	15876
19	31752
20	62504

This completes the analysis of binary alphabets that produce spectrum zeros at both $f = 0$ and $f = 1/2T$. The next section will still be concerned with synthesis of binary alphabets, but the procedure will be extended to include alphabets that produce spectrum zeros at almost any desired frequency.

Binary alphabets with S = 0 at f = 1/kT

In order for a sequence of n bits, $a_0 a_1 a_2 \cdots a_{n-1}$, to have a spectrum zero at the frequency $1/kT$, it is sufficient to satisfy the following condition:

$$\sum_{i=0}^{n-1} a_i e^{-j2\pi i/k} = 0 . \tag{16}$$

A graphical interpretation of this equation states that the sum of n vectors in the complex plane must be zero. However, the identity

$$e^{-j2\pi i/k} = e^{-j2\pi(i+k)/k} \tag{17}$$

indicates that one may construct a desired sequence by balancing k rather than n vectors. Since the phase angles are determined completely by the value of k, the vectors can only be balanced by choosing their amplitudes correctly. If, for the time being, we restrict the word length, n, to an integral multiple of k, i.e., $n = ks$, then we can denote the vector amplitudes by the summation

$$A_i = \sum_{r=0}^{s-1} a_{i+rk} , \qquad i = 0, 1, 2, \cdots, k-1 \tag{18}$$

and Eq. (16) becomes

$$\sum_{i=0}^{k-1} A_i e^{-j2\pi i/k} = 0 . \tag{19}$$

- *k is a prime number*

If k is a prime number, the system can be in equilibrium only if all the vectors are equal in amplitude:

$$A_0 = A_1 = \cdots = A_{k-1} . \tag{20}$$

As an example, if $k = 3$ and $n = 6$, the following relationship must hold:

$$A_0 = A_1 = A_2 ,$$

$$a_0 + a_3 = a_1 + a_4 = a_2 + a_5 . \tag{21}$$

Since the amplitudes are each the sum of two binary elements, they can take only three possible values in this example: -2, 0, $+2$. There is just one word such that the three amplitudes equal $+2$: 1 1 1 1 1 1; and similarly, one word such that the amplitudes equal -2: -1 -1 -1 -1 -1 -1. Eight different words exist such that the three amplitudes equal zero. These ten six-bit words are presented in Table 4 as a decimal code.

Table 4 6-bit decimal code with spectrum zero at $f = 1/3T$.* The binary patterns in this alphabet are arranged so that the character corresponding to the decimal number, d, is the base-2 representation of the decimal number $7d$.

0	0 0 0 0 0 0
1	0 0 0 1 1 1
2	0 0 1 1 1 0
3	0 1 0 1 0 1
4	0 1 1 1 0 0
5	1 0 0 0 1 1
6	1 0 1 0 1 0
7	1 1 0 0 0 1
8	1 1 1 0 0 0
9	1 1 1 1 1 1

* The 0's correspond to the -1's of the analysis.

In general, for k a prime number and n an integral multiple of k, the size of the binary alphabets is given by

$$C\left[k, \frac{n}{k}\right] = \sum_{i=0}^{n/k} (C_i^{n/k})^k .$$

- *k is not a prime number*

Suppose that $k = c \times d$, where c and d are integer factors of k. Then, solutions of Eq. (19) are given by

$$A_u = A_{u+vc} , \tag{22}$$

where $u = 0, 1, 2, \cdots, c-1$, and $v = 1, 2, 3, \cdots, d-1$. The complete alphabet for a given k is the union of the solutions to Eq. (22) for each possible pair of factors; e.g., if $k = 12$, Eq. (22) should be solved under the conditions 6×2, 4×3, 3×4, and 2×6 to find all the words in the alphabet.

As an example, for $k = 4$ and $n = 8$, the following relationships will hold:

$$A_0 = A_2, \qquad a_0 + a_4 = a_2 + a_6$$
$$A_1 = A_3, \qquad a_1 + a_5 = a_3 + a_7.$$

These expressions are valid here since it is not necessary that all four vectors be equal in order to balance the system. As long as diametrically opposed pairs of vectors are equal, the system will be in equilibrium.

In the present example each vector can have the values $+2$, 0, or -2. There is one combination of four bits such that $A_0 = A_2 = +2$, and one combination such that they equal -2; four combinations yield zero. Similarly, there are six combinations of four bits that produce $A_1 = A_3 = +2$, 0, or -2. Since each of the six combinations for $A_0 = A_2$ can be combined with each of the six for $A_1 = A_3$, the alphabet of 8-bit words producing a spectrum zero at $1/4T$ contains 36 words.

It should be noted here that if the transmission speed is 3000 bits per second, the frequency spectrum envelope of this 36-word alphabet is very close to that of the available frequency bandwidth of the public telephone network

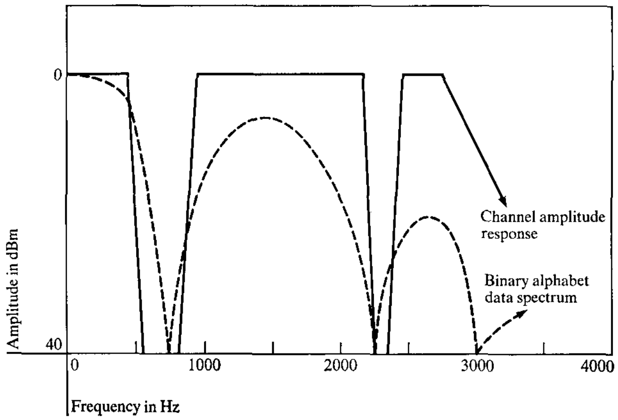


Figure 6 Channel amplitude response of the public telephone network of the United Kingdom. (After Cook.[4])

in the United Kingdom[4] as shown in Fig. 6. It can be further observed that the code satisfies the relationships,

$$a_0 - a_2 + a_4 - a_6 = 0 \quad (23)$$
$$a_1 - a_3 + a_5 - a_7 = 0,$$

while in the STEAN code described earlier, the relationships happen to be

$$a_0 + a_2 + a_4 + a_6 = 0$$
$$a_1 + a_3 + a_5 + a_7 = 0. \quad (24)$$

A very simple transcoder from one code to the other consists of inverting every third, fourth, seventh, and eighth bit of either code. The maximum frequency contribution of one alphabet corresponds to the minimum frequency contribution of the other.

• *n is not a multiple of k*

When n is not a multiple of k, alphabets with spectrum zeros at $1/kT$ exist, but the techniques for synthesizing them involve grouping the words into sequences long enough to satisfy the basic conditions. For instance, if $k = 3$ and $n = 7$, messages constituted of multiples of 3 words will satisfy the conditions if the words are properly synthesized. We will not discuss the problem of constructing these alphabets in this paper.

• *k is a rational fraction*

This study was done for k being any integer, but it can easily be extended to any rational fraction. The extension relationship is simply given by the fact that if an alphabet has a zero frequency spectrum at $f = 1/kT$ it will necessarily have also a zero frequency contribution at $f = x/kT$ if x and k are relatively prime numbers.

As an example, it is easy to check that for $k = 4$ and $x = 3$ the relationships $A_0 = A_2$ and $A_1 = A_3$ are precisely the same as for $x = 1$ where A_i represents the same sum of elements in both cases.

More general alphabets can be constructed if an alphabet is required with preselected zero frequency contribution at $f = 1/k_1T$, $f = 1/k_2T$, $f = 1/k_3T$. It will simply be the intersection of the alphabets presenting no frequency contribution at $f = 1/k_1T$, $f = 1/k_2T$, and $f = 1/k_3T$. In Fig. 7 frequency spectrum envelopes corresponding to various alphabets are shown.

Binary alphabets having envelope spectrum zero at both f = 0 and f = 1/kT

Alphabets with zero spectrum amplitude at $f = 0$ are determined by the following equation:

$$\sum_{i=0}^{k-1} A_i = 0\,. \quad (25)$$

Alphabets such that the frequency spectrum is zero at both $f = 0$ and $f = 1/kT$, where k is a prime number, will be such that each word satisfies simultaneously Eq. (19) and Eq. (25). That is,

$$A_i = 0\,, \qquad i = 0, 1, 2, \cdots, k-1\,. \quad (26)$$

In a special case where $n = 2k$ the alphabets consist of 2^k characters or k information bits. The particular systems can be characterized by a signal element of the following nature:

$$\underbrace{100\cdots0}_{k} - \underbrace{100\cdots0}_{k}\,.$$

In general, if $A_i = 0$ has C solutions, the alphabet will consist of C^k elements. Reference 3 corresponds to $C = 2$, $k = 2$.

Figure 7 Spectrum envelopes of binary alphabets chosen to satisfy various spectrum characteristics. The alphabet chosen to produce a zero at $f = 1/4T$ (and consequently, at $f = 3/4T$) contains 36 characters. The alphabet that produces zeros at both $f = 0$ and $f = 1/4T$ contains 18 characters, as does the alphabet with zeros at both $f = 1/4T$ and $f = 1/2T$. The alphabet giving zeros at $f = 0, f = 1/5T$, and $f - 1/2T$ contains 16 characters.

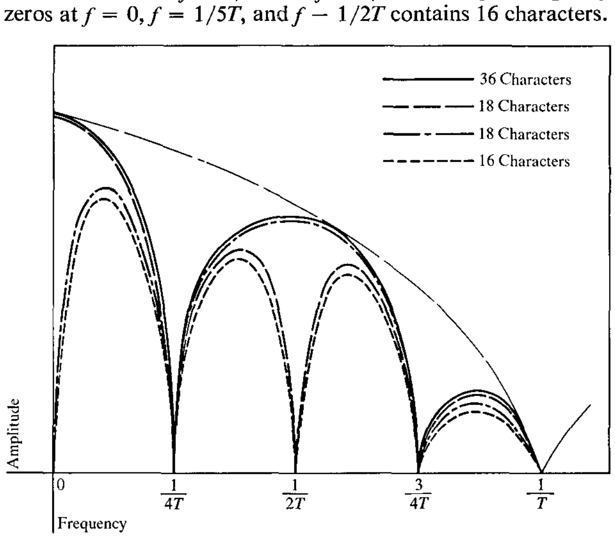


Table 5 Number, $C(n)$, of characters in an n-element, m-level alphabet having spectrum zeros at both $f = 0$ and $f = 1/2T$.

n	$C(n)$		
	$m = 2$	$m = 3$	$m = 4$
4	4	9	16
5	6	21	48
6	9	49	144
7	18	133	528
8	36	361	1936
9	60	963	6820
10	100	2601	24025

If bit detection is used, any burst of k errors in a word will be detected in these alphabets.

When k is not a prime number, ($k = cd$), solutions will be given by satisfying simultaneously Eq. (22) and the following equation:

$$\sum_{u=0}^{c-1} A_u = 0 . \tag{27}$$

Non-binary alphabets

Clearly, the basic theory holds for any digital signal whether it is binary or not, and the same techniques will apply to the construction of an m-level alphabet. The only difference being that the value "permitted" for the a's will be denoted as:

$-1, 0, +1$ for $m = 3$.
$-3, -1, +1, +3$ for $m = 4$
$-2, -1, 0, +1, +2$ for $m = 5$, etc.

Table 5 gives a comparison of the numbers of characters which exist within binary, ternary, and quarternary alphabets for n less than and equal to 10. This table concerns again the case where the spectrum frequency is zero at both $f = 0$ and $f = 1/2T$.

The same technique such as systematic inversion of every second character could and should be used in some instances. $m = 4$ and $n = 6$ is an example. Naturally, the size of the alphabet, which corresponds to the number of code combinations satisfying the basic relationship, increases very rapidly with m.

If transition (zero crossing) within the data signals is required for clock recovery, the sequence $00 \cdots 0$ when m is odd should be eliminated. For example, 132 distant voice signal amplitudes can be encoded at an 8 kHz sampling rate into a 7 element self-clocking 3 level digital code which will have no frequency contribution at $f = 0$ and $f = 28$ kHz.

Three level digit coding such as proposed in references 4, 6 and 7 generate sequences belonging to the same family (no frequency contribution at both $f = 0$ and $f = 1/2T$) and, therefore, satisfy Eqs. (2) and (3). In these cases $N - 2$ bits are encoded into a N-bit message where the two first and the two last elements are binary digits while the remaining $N - 4$ elements are ternary digits. The interest here is in the simplicity of these coding schemes and in their efficiency for long messages. In order to compare them formally with the fixed-length character alphabets, one would let $m = 3 - (4/N)$; the size of the alphabets is 2^{N-2} characters.

Conclusions

We have described in this paper a number of alphabet and sub-alphabet families which can be stored in a computer, have unique characteristics in their frequency spectra, and may fit particular requirements. These requirements may come from the user as, for example, the assignment of characters with frequencies in the middle of the band to represent information requiring maximum security; or they may come from the transmission system as, for example, avoiding data contributions at specified frequencies.

The application of conventional "time domain equalization" or "character detection" schemes is not straightforward. Conventional time domain equalization is based on signal element responses. Here, an individual impulse response is not a meaningful way to characterize the transmission system. Special algorithms for the decisions that provide best recognition will have to be developed since the procedures will be based on whole characters and not on random combinations of single bits. Another interesting aspect is that if extra error protection is required the error control characters must belong to the same alphabet.

More work is still needed to get a better understanding of these alphabets and where they will be used most effectively. The purpose of this paper is simply to show the existence of these families of natural frequency concept code alphabets.

References

1. E. Gorog and J. Pierret, "Coding for Data Transmission," *IBM Technical Disclosure Bulletin* **10**, No. 1, 1295 (1968).
2. H. Nyquist, "Certain Topics in Telegraph Transmission Theory," *AIEE Transactions* **47**, 617 (1928).
3. R. Seibel, "Letter Sequences (N-Grams): 1. Frequencies in Samples of Text from the (London) Times," IBM Research Report RC-972, Yorktown Heights, New York.
4. C. M. Melas and E. Gorog, "A Digital Data Transmission System with Optimum Bandwidth Utilization," *Proceedings of the National Communications Conference*, Utica, October, 1964.
5. F. W. Cook, "Medium Speed Digital Data Transmission," *Systems Technology*, No. 1, 43 (June, 1967).
6. A. Lender, "Correlative Level Coding for Binary Data Transmission," *IEEE Spectrum* **3**, No. 2, 104 (1966).
7. S. K. Becker, E. R. Kretzmer, and S. H. Sheehan, "A New Signal Format for Efficient Data Transmission," *BSTJ* **45**, No. 5, 755 (1966).

Received October 22, 1966.
Revised January 22, 1968.