The Lenkurt® Demodulator

NEWS FROM LENKURT ELECTRIC

VOL. 10 NO. 6 JUNE, 1961

SPECIAL CODE TECHNIQUES
for Improved Data Transmission

Although transmission methods and equipment improve steadily, it is still necessary to fight the battle for better communications on more than one front. The need for error-free data transmission is growing, and the practical limitations of existing communications plant indicate that transmission improvements alone will not be adequate to satisfy the need. One way of improving the quality and rate of information transfer is to organize what is sent, as well as the means by which it is transmitted. This article reviews one of the more rewarding "new frontiers" of communications, the study of improved coding techniques.

Communication requires that information be transported from one place to another and, for this purpose, must be converted into a form suitable for handling. Languages, words, even oral speech are actually "codes" which permit ideas and concepts to be transmitted between individuals.

Electrical communication requires additional conversions in order to fit the words or other symbols for transmission. Sound waves are converted to a variable voltage; electrical pulses, like drum beats or smoke signals, provide the means for transmitting letters and numerical data.

Regardless of the exact means of transmission, some form of symbolic language or code must always be used for carrying information from its source to its destination, and most of these codes and languages are inherently wasteful. In language, some words are used more than others, and letters occur in predictable patterns. This predictability and pattern in sounds, letters, and words make it possible to receive the meaning of a spoken or written message, even when some part of it is altered or deleted in transmission. A reader's familiarity with the words and syntax of a language allow him to supply missing or incorrect letters and words in the text. The prolonged sounds of speech, and their inflection and pattern preserve the intelligibility of speech ex-

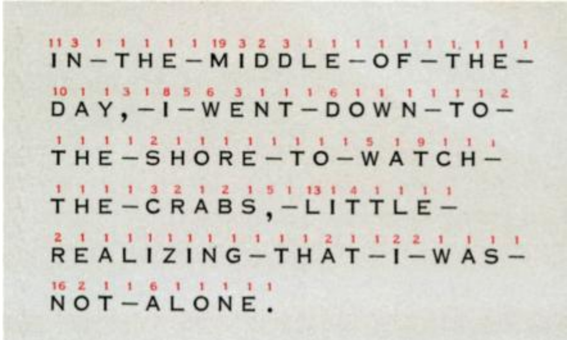

Figure 1. Predictability of language is indicated by large number of characters correctly guessed on first try. Numerals indicate number of guesses required to identify each character. Only a few characters required many guesses; the rest were obvious.

cept in the presence of extreme interference.

A simple experiment will confirm how predictable language actually is. A short passage of written prose is selected, and the subject is asked to guess the characters (including spaces and punctuation) one at a time. The subject continues guessing until he names each character correctly. As each character is guessed, it is written down as an aid in predicting the next character. The results of such an experiment are shown in Figure 1. The numerals show the number of guesses required for each character. Of the 109 symbols in the text, the subject guessed correctly on his first try 79 times, and was able to identify all 109 characters in 235 attempts. This is an average of only about two guesses (or information "bits") per character. Further work along this line has indicated that long passages of English text have an actual information content of only about *one* bit per letter. This means that, theoretically, it should be possible to transmit text by pulses no more numerous than the letters themselves, thus enabling us to throw away 24 of the 26 letters, without loss of communication! Although this ideal

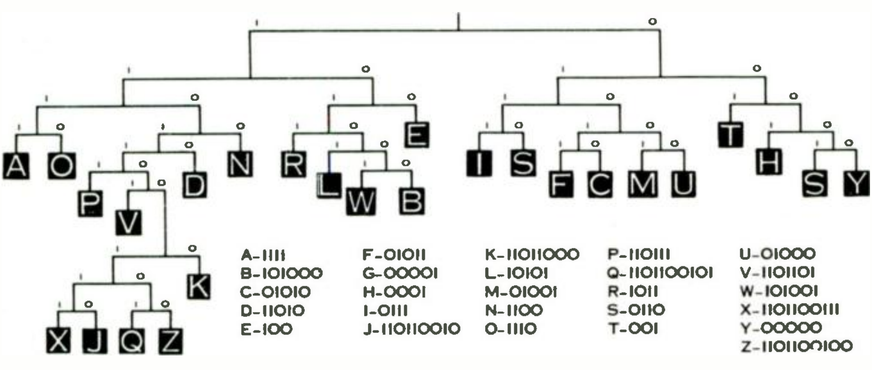

Figure 2. Efficient binary code for English requires average of only 4 bits per character by taking advantage of language statistics.

will not be achieved, it provides a goal to be approached as closely as possible.

Statistical Codes

Practical codes can be made more efficient by designing them to fit the statistics of the language. Thus, letters which occur most frequently — *E, T*, and *A*, for instance — are represented by the shortest code symbols, while the least probable characters have longer symbols. Figure 2 shows such a code which has an average information content of about 4 bits per character. By contrast, the standard teletypewriter code employs 5 bits per character, not counting synchronizing pulses.

Although the additional, redundant symbols and pattern in language may help overcome errors, *unsystematic* redundancy is wasteful, and merely lowers the rate of communication. It follows logically that the more redundancy removed, the more efficient the communications channel, but, the greater the likelihood of error due to interference. Since interference is always present to some degree, a very efficient communications system would use a code in which all message redundancy was eliminated in order to obtain maximum information rate; then, just enough redundancy would be re-inserted to overcome the interference present in the transmission path.

Error Detection

One of the fastest growing fields in communication is the transmission of numerical data. Unlike spoken and written languages which are largely redundant, data messages have no inherent redundancy. The machine-generated characters occur without pattern, and errors cannot be detected by inspection, as in the case of text. To complicate matters, data errors cannot be tolerated to the same extent as errors in text, because control operations or machine calculations may be completely ruined by a single error. Yet the high speed with which data are generated and transmitted makes the occurrence of errors more likely.

One way of overcoming errors in handling and transmitting high-speed data is to design codes which, by their very construction and organization, are able to detect or even correct errors automatically. Unfortunately, such codes cannot be created without adding redundancy. The problem then becomes one of finding a coding method that provides maximum error-free transmission with the least possible redundancy.

LETTER	SYMBOL	LETTER	SYMBOL
A	0000	E	1111
B	0011	F	1100
C	0101	G	1010
D	0110	H	1001

Shown above is one type of error-detecting code. Note that although there are only eight characters, four binary digits are required for each, instead of three. A single error in any symbol will transform it into a combination not used for any other character in the code. For instance, the group **1111** *(E)* could be converted to **0111**, **1011**, **1101**, or **1110** by a single error. *Double* errors, however, convert the symbol into another valid character, and *neither* error is detectable. Although this particular code can reveal single errors, it cannot correct them, since certain error combinations could result from errors in any one of several code groups. For instance, the group 1101 could be produced by a single error in the symbols for *C, E, F,* or *H.*

Parity Checks

A widely used error detection scheme is the so-called parity check. An extra digit is added to the regular binary code so that there will always be an even (or odd) number of 1's in each code group. A single error will cause an odd number of 1's to appear at the receiver, indicating an error. A single parity check will detect all odd numbers of errors, but will not detect double errors or other even-count errors, since the count of 1's will still provide the required even number. By adding an additional parity check for *every other* digit, all odd numbers of errors and about half the even number of errors can be detected. A third parity check added for the remaining digits will further reduce the undetectable errors.

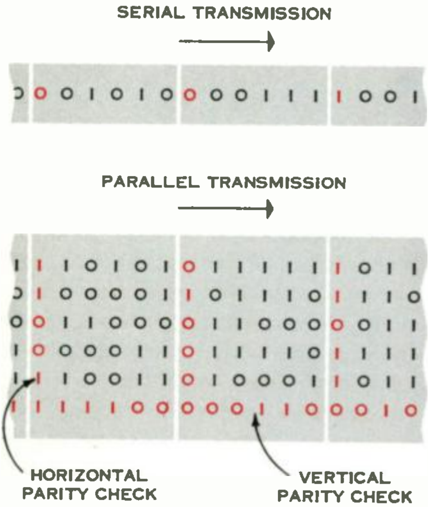

Figure 3. Typical parity checks for serial and parallel transmission. Check digit is chosen so that sum of marks in each block of data is always even (odd, in some codes). Error is revealed if wrong count is received. Compensating double errors may go undetected.

Parity checks provide some protection against errors, but, like all redundancy, they slow down the transmission of the message. If a single parity check is used with each five-digit code group, as shown in Figure 3, the message will contain about 16% redundancy. This can be reduced by increasing the number of information digits for each check digit, but this increases the probability of undetectable errors occurring.

Many types of parity check systems exist. Where parallel transmission is used (tape-to-tape computer data, for instance), parity checks may be used in both the horizontal and vertical directions, in order to reduce the chance of data errors going undetected.

A related approach to error detection uses a fixed ratio of marks and spaces for all code characters. When designed to reduce the likelihood of compensating errors, this code can be very effective in detecting most errors.

Error Correction

It is not enough merely to identify the existence of errors. Some form of restoration or correction of the message is required in order to complete the transmission or control function. One basic form of error correction is to transmit the message several times in the hope that errors will not destroy identical portions of each message. A similar approach would be to transmit each digit several times and count the bits received. A majority count would presumably reveal the correct digit. Obviously, this method fails if more than half the digits are in error.

A more economical way of correcting errors requires a two-way channel and some form of error detection. The message is divided into blocks or groups for transmission. When an error is detected, the receiver signals this fact to the transmitter over the return channel, and the block of information containing the error is transmitted again.

Retransmission may be a very efficient method of error control in channels where interference occurs only infrequently, but then destroys large numbers of symbols at a time. If the transmitted data are generated in "real time," however, (as in the case of radar warning or telemetering and control applications), the retransmission may impair the timeliness of the information, and may require too much data storage equipment. Similarly, in cases where a single transmitter sends to several receivers over different paths, error correction by retransmission becomes cumbersome and inefficient, because an error in any link requires that the message be retransmitted to all stations.

"Minimum Distance" Codes

Error-correcting codes which do not require retransmission have been devised, using principles similar to those used in the code of Figure 1. Error correction is obtained by adding additional redundant digits so that an erroneous code group still most nearly resembles the intended group, despite changes occurring in one (or more) binary digits. Obviously, the redundancy is greatly increased.

Mathematicians specializing in information theory and advanced coding techniques find it useful to describe codes in terms of geometry, so that each character in the code is located at a "corner" or vertex of a geometrical figure. Thus, a code having 2 digits could be described by a square with all four combinations located at the four corners. A code with three digits would require a three-dimensional figure for the eight possible combinations, and a four-digit code requires a solid having four dimensions to adequately describe its properties. Although it is difficult or impossible to diagram multi-dimensional figures accurately on paper, they are relatively easy to handle mathematically.

Since each code combination, whether an error or a correct symbol, lies at a

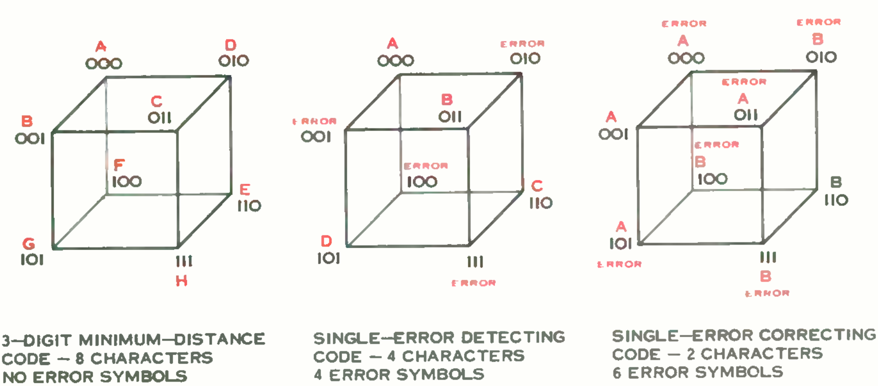

Figure 4. Solid geometry allows mathematicians easy way to analyze efficiency and other characteristics of complicated codes. Simple 3-digit example shown requires three-dimensional figure. Each vertex represents a code combination assignable to characters or errors.

vertex of the solid figure, a change in one digit represents the difference between one vertex and an adjacent one. Two changes move it two places, and so forth. The ideal code, then, will use the least possible number of code combinations, but separates all *valid* (non-error) code groups by as many locations as possible. The less the "distance" between correct symbols, the lower the redundancy. If additional "distance" is placed between valid characters, the code can either detect multiple errors or correct single errors, depending on how the code is set up. Figure 4 diagrams how a geometric figure can be used to express "distance" between symbols, and shows how efficiency or information capacity can be traded for error correction or detection capability.

Burst-Correcting Codes

Although many error-detecting and error-correcting codes have been devised, actual transmission links may render them rather useless. Many observations have shown that errors do not necessarily occur at random, but often come in "bursts." A code that has the ability of correcting two or three errors is useless against interference that may destroy five or ten consecutive digits.

A number of special "burst-correcting codes" have been devised which can correct many consecutive errors. Some of these codes may require such elaborate equipment as to make them impractical. Others are simpler but require large amounts of redundancy.

Figures 5 and 6 diagram a typical approach to burst-correction. The message is fed continuously into the first stage of a so-called "shift register," a chain of identical stages, each of which stores one binary digit. When a "shift" signal is applied to all stages of the register, each stage transfers its stored digit to the following stage, and accepts the digit from the preceding stage. In a seven-stage shift register, therefore, seven shifts are required for each digit to pass through the register.

As shown in Figure 5, the digits in stages 1 and 4 of the encoder are sampled by a "check digit generator." The check digit generator produces a **0** at its output if the two digits sampled from the shift register are alike, or a **1** if they differ. The resulting redundant check digits are transmitted alternately with the message digits from the last stage of the register.

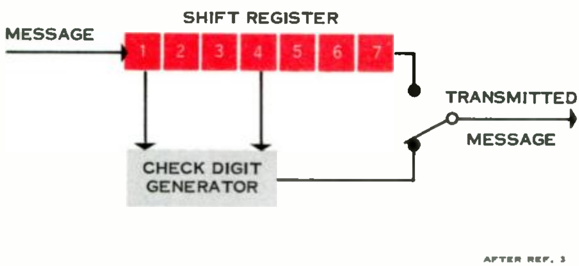

Figure 5. Encoder for error-burst correction. Check digits are transmitted alternately with each message digit.

At the decoder, a reverse process is used. The alternate message and check digits are separated and fed into individual shift registers, as shown in Figure 7. Check digits are again calculated from the message digits and compared with the check digits actually received. Since each message digit was used twice

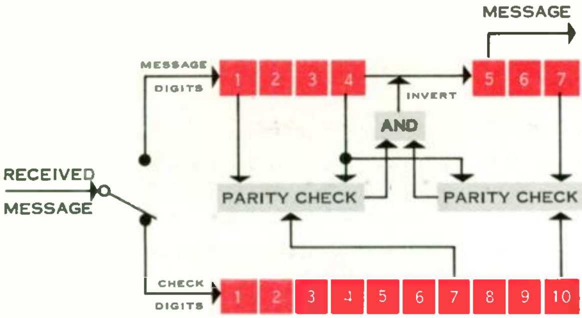

Figure 6. Decoder for burst-correcting code. Twenty error-free digits are required between bursts to "recharge" check digit register, thus imposing severe limitation on this type of code.

AFTER REF. 3

at the coder to calculate check digits (position 1, then position 4), two separate parity checks are required to eliminate ambiguity in identifying errors. If parity fails in both check circuits at the same time, the digit in stage 4 of the message register is known to be wrong, and is corrected as it shifts into stage 5. Thus, all digits in stages 5, 6, and 7 must be correct unless the burst-length capacity of the system has been exceeded.

This type of decoder can correct bursts having a length one less than the number of stages in the message shift register. Thus, the system diagrammed can correct bursts up to six digits in length. Longer bursts could be corrected by using larger shift registers. An important disadvantage of this type of error-correcting system is that a 20-digit error-free transmission is required following each corrected burst in order to clear the erroneous check digits from the decoder check digit register. This sharply limits the usefulness of this system in dial-up circuits or others subject to frequent error bursts.

Conclusions

The field of error detection and correction by coding is rather new, and much remains to be learned about achieving the desired performance. Some systems, such as those used to read and transmit punched cards, are already using ingenious error-correcting systems but this approach has not yet been adapted to circuits which transmit more conventional data serially over one-way paths. There is great promise of simple ways of overcoming all kinds of interference in existing transmission systems, thus leading to much higher information rates than are now possible over practical circuits. •

Bibliography

1. R. W. Hamming, "Error Detecting and Error Correcting Codes," *Bell System Technical Journal;* April, 1950.
2. D. A. Bell, *Information Theory and Its Engineering Applications (2nd Edition);* Pitman Publishing Corp., New York, 1956.
3. D. W. Hagelbarger, "Recurrent Codes: Easily Mechanized, Burst-Correcting Binary Codes," *Bell System Technical Journal;* July, 1959.
4. A. A. Alexander, R. M. Gryb, and B. W. Nast, "Capabilities of the Telephone Network Data Transmission," *Bell System Technical Journal;* May, 1960.
5. C. M. Melas, "Reliable Data Transmission through Noisy Media — A Systems Approach," *AIEE Paper CP61-377;* February, 1961.

Lenkurt Electric Co., Inc.
San Carlos, California

Bulk Rate
U.S. Postage
Paid
San Carlos, Calif.
Permit No. 37

MR. J. J. O'BRIEN
6606 FIFTH ST.
RIO LINDA, CALIF.
159-1159

R-1771

Form 3547
Requested

Lenkurt Establishes New District Offices in U. S.

SAN CARLOS
1000 Howard Avenue
San Carlos, California
Phone: DDD 415
LYtell 1-8461
TWX - S. CarBel 100

NEW YORK
730 Third Avenue
New York 17, New York
Phone: DDD 212
MUrray Hill 7-2810
TWX - NY 1-249

DALLAS
1425 Southland Center
Dallas 1, Texas
Phone: DDD 214
RIverside 8-6841

MINNEAPOLIS
7600 Highway No. 7
Minneapolis 26, Minnesota
Phone: DDD 612
WEst 5-3329
TWX - MP-1084

ATLANTA
361 E. Paces Ferry Road, N.E.
Atlanta 5, Georgia
Phone: DDD 404
237-6489
TWX - AT-21

CHICAGO
300 West Washington St.
Chicago 6, Illinois
Phone: DDD 312
ANdover 3-1321

WASHINGTON
900 Bender Building
1120 Connecticut Avenue
Washington 6, D.C.
Phone: DDD 202
STerling 3-2576

LENKURT ELECTRIC SAN CARLOS, CALIFORNIA, U.S.A. GENERAL SYSTEM

District Offices
New York • Minneapolis
Atlanta • Dallas
Chicago • San Carlos
Washington, D.C.

The Lenkurt Demodulator is a monthly publication circulated free to individuals interested in multi-channel carrier, microwave radio systems, and allied electronic products. Permission to reproduce material from the *Demodulator* will be granted upon request. Please address all correspondence to the Editor.