# Permutation Codes for Sources

TOBY BERGER, MEMBER, IEEE, FREDERICK JELINEK, SENIOR MEMBER, IEEE, AND JACK K. WOLF, MEMBER, IEEE

*Abstract*—**Source encoding techniques based on permutation codes are investigated. For a broad class of distortion measures it is shown that optimum encoding of a source permutation code is easy to instrument even for very long block lengths. Also, the nonparametric nature of permutation encoding is well suited to situations involving unknown source statistics.**

**For the squared-error distortion measure a procedure for generating good permutation codes of a given rate and block length is described. The performance of such codes for a memoryless Gaussian source is compared both with the rate-distortion function bound and with the performance of various quantization schemes. The comparison reveals that permutation codes are asymptotically ideal for small rates and perform as well as the best entropy-coded quantizers presently known for intermediate rates. They can be made to compare favorably at high rates, too, provided the coding delay associated with extremely long block lengths is tolerable.**

## I. INTRODUCTION

VARIANT I and Variant II permutation codes were introduced by Slepian [1] for the purpose of reliably transmitting digital data over a certain class of noisy channels. Shortly thereafter, Dunn [2], [3] considered the use of Variant I permutation codes as a means for digitizing vectors generated by a time-discrete memoryless Gaussian source. We have extended the work of Dunn in several ways. In particular, a procedure has been devised that generates, for a broad class of sources, Variant I and Variant II permutation codes of a specified rate and block length that are nearly optimum in the mean-squared-error (MSE) sense. It is shown that the encoding and decoding of source permutation codes is fully instrumentable for any monotonic nondecreasing convex-$\cup$ nonnegative distortion measure. In the Gaussian MSE case, permutation source codes are shown to be asymptotically ideal for low rates and to perform as well as the best entropy-coded quantizers presently known for intermediate rates. The instrumentability of permutation codes and their applicability to situations in which the source statistics are imperfectly known are also discussed.

Calculation of the performance of the Gaussian MSE Variant II codes necessitated the generation of tables of Fraser normal scores for large values of $n$. We have included some of these tables herein, since the only table previously published [4] is restricted to $n \leq 10$.

Manuscript received February 11, 1971. This research, conducted at Cornell University, was supported in part by NSF Grant GK 14449, NASA Contract 2-5643, and AFOSR Contract F44620-71-C-0001.

T. Berger and F. Jelinek are with the School of Electrical Engineering, Cornell University, Ithaca, N.Y. 14850.

J. K. Wolf is with the Department of Electrical Engineering, Polytechnic Institute of Brooklyn, Brooklyn, N.Y. He is now on leave at the Department of Electrical Engineering, University of Hawaii, Honolulu, Hawaii 96822.

## II. ENCODING OF PERMUTATION CODES

Consider a time-discrete information source emitting the sequence of real random variables $\{x_k, k = 1,2,\cdots\}$. The source outputs need not be either statistically independent or identically distributed. We are concerned with block coding (block quantizing) of the $n$-dimensional random vector $\boldsymbol{x} = (x_1,x_2,\cdots,x_n)$. Any set of $M$ $n$-vectors, $B = \{\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_M\}$, constitutes a source block code of rate

$$R = n^{-1} \log_2 M. \tag{1}$$

When the source output vector assumes the value $\boldsymbol{x}$, it should be encoded into whichever $\boldsymbol{y} \in B$ minimizes some prescribed block distortion measure $d(\boldsymbol{x},\boldsymbol{y})$. The resulting per-letter average distortion of the code $B$ is defined as

$$D = n^{-1}E[\min_{\boldsymbol{y}\in B} d(\boldsymbol{x},\boldsymbol{y})], \tag{2}$$

where $E$ denotes expectation with respect to the distribution of $\boldsymbol{x}$.

The optimum encoding procedure for a general block code is very complex. In its worst form, the source output vector must be compared with each of the codewords $\boldsymbol{y}_k$, $k = 1,2,\cdots,M$, and then encoded as the subscript of that codeword that attains the minimum distortion. For very large $M$ this is a horrendous task. The appeal of permutation codes stems principally from the fact, embodied in Theorem 1 below, that they possess a simple optimum encoding procedure for a broad class of interesting distortion measures.

In Slepian's Variant I and Variant II permutation codes, the codewords $\boldsymbol{y}_k$, $k = 1,2,\cdots,M$, are chosen in the following manner.

*Variant I Codes:* The first codeword is an $n$-vector of the form

$$\boldsymbol{y}_1 = (\overset{\leftarrow\, n_1 \,\rightarrow}{\mu_1,\cdots,\mu_1},\overset{\leftarrow\, n_2 \,\rightarrow}{\mu_2,\cdots,\mu_2},\overset{\cdots}{\cdots},\overset{\leftarrow\, n_K \,\rightarrow}{\mu_K,\cdots,\mu_K}), \tag{3}$$

where the $\mu_i$ are $K$ real numbers satisfying $\mu_1 > \mu_2 > \cdots > \mu_K$, and the $n_i$ are positive integers satisfying

$$n_1 + n_2 + \cdots + n_K = n. \tag{4}$$

The other codewords $\boldsymbol{y}_2,\boldsymbol{y}_3,\cdots,\boldsymbol{y}_M$ are all the distinct words that can be obtained by rearranging the components of $\boldsymbol{y}_1$ in all possible ways. There are a total of

$$M = n!/\prod_{i=1}^{K} n_i! \tag{5a}$$

codewords.

*Variant II Codes:* The first codeword $\boldsymbol{y}_1$ is again of the form specified by (3), but with the added proviso that the $\mu_i$ are all nonnegative, $\mu_1 > \mu_2 > \cdots > \mu_K \geq 0$. The other words of the code are formed by assigning algebraic signs to the components of $\boldsymbol{y}_1$ in all possible ways, and then permuting these signed components in all possible ways. The number of codewords in a Variant II code therefore is

$$M = 2^h n!/\prod_{i=1}^{K} n_i!, \tag{5b}$$

where $h = n$ if $\mu_K > 0$ and $n - n_K$ if $\mu_K = 0$.

*Theorem 1:* Consider a block distortion measure of the form

$$d(\boldsymbol{x},\boldsymbol{y}) = g\left(\sum_{t=1}^{n} f(|x_t - y_t|)\right), \tag{6}$$

where $\boldsymbol{x} = (x_1,\cdots,x_n)$, $\boldsymbol{y} = (y_1,\cdots,y_n)$, $g(\cdot)$ is nondecreasing, and $f(\cdot)$ is nonnegative, nondecreasing, and convex $\cup$ for positive arguments. Then optimum encoding of Variant I and Variant II permutation codes with respect to $d(\boldsymbol{x},\boldsymbol{y})$ is accomplished by the simple algorithms described below.

*Variant I Encoding Algorithm:*

1) Replace the $n_1$ largest components of $\boldsymbol{x}$ by $\mu_1$.

2) Replace the next $n_2$ largest components of $\boldsymbol{x}$ by $\mu_2$.

⋮

$K$) Replace the $n_K$ smallest components of $\boldsymbol{x}$ by $\mu_K$.

Use the permutation of $\boldsymbol{y}_1$ that results from these replacements to represent $\boldsymbol{x}$.

*Variant II Encoding Algorithm:*

1) Replace the $n_1$ components of $\boldsymbol{x}$ largest in absolute value by either $+\mu_1$ or $-\mu_1$, the sign chosen to agree with that of the component it replaces.

2) Replace the $n_2$ components of $\boldsymbol{x}$ next largest in absolute value by either $+\mu_2$ or $-\mu_2$, the sign again chosen to agree with that of the component it replaces.

⋮

$K$) Replace the $n_K$ components of $\boldsymbol{x}$ smallest in absolute value by either $+\mu_K$ or by $-\mu_K$, the sign again chosen to agree with the sign of the component it replaces.

Use the codeword that results from these replacements to represent $\boldsymbol{x}$.

*Proof:* See Appendix I.

It should be noted that, if the source letters $x_1,x_2,\cdots,x_n$ are independent and identically distributed, then all codewords of a Variant I code occur equiprobably under optimum encoding. If, in addition, the source letter distribution is symmetric about zero, then the same becomes true for Variant II codes. The requirement of independence imposed on the $x_k$ here can be replaced by the somewhat less stringent requirement of exchangeability as described, for example, by Feller [5].

## III. Minimum-Mean-Squared-Error (MMSE) Permutation Codes

The ever-popular squared-error distortion measure

$$d(\boldsymbol{x},\boldsymbol{y}) = n^{-1}\sum_{t=1}^{n}(x_t - y_t)^2 \tag{7}$$

is a special case of those considered in Theorem 1. In this section, we describe a method for constructing Variant I and Variant II permutation codes of a given rate and block length that are optimum in the squared-error sense.

Define the random variable $\xi_j$, $j = 1,\cdots,n$, to be the $j$th largest component of the source vector $\boldsymbol{x}$, and the random variable $\eta_j$ to be the $j$th largest of the absolute values of the components of $\boldsymbol{x}$. For convenience, let

$$S_i = n_1 + n_2 + \cdots + n_i \tag{8}$$

and define $S_0 = 0$. Then the MSE values of optimally encoded Variant I and Variant II codes are given, respectively, by

$$D_{\mathrm{I}} = n^{-1}E\left[\sum_{i=1}^{K}\sum_{j=S_{i-1}+1}^{S_i}(\xi_j - \mu_i)^2\right] \tag{9a}$$

and

$$D_{\mathrm{II}} = n^{-1}E\left[\sum_{i=1}^{K}\sum_{j=S_{i-1}+1}^{S_i}(\eta_j - \mu_i)^2\right]. \tag{9b}$$

Noting that

$$\sum_{k=1}^{n}\xi_k^2 = \sum_{k=1}^{n}\eta_k^2 = \sum_{k=1}^{n}x_k^2, \tag{10}$$

we can rewrite (9) in the form

$$nD_{\mathrm{I}} = E|\boldsymbol{x}|^2 - 2\sum_{i=1}^{K}\mu_i\sum_{j=S_{i-1}+1}^{S_i}E\xi_j + \sum_{i=1}^{K}n_i\mu_i^2 \tag{11a}$$

$$nD_{\mathrm{II}} = E|\boldsymbol{x}|^2 - 2\sum_{i=1}^{K}\mu_i\sum_{j=S_{i-1}+1}^{S_i}E\eta_j + \sum_{i=1}^{K}n_i\mu_i^2, \tag{11b}$$

where $|\boldsymbol{x}|^2$ is the summation in (10).

Differentiation with respect to $\mu_i$ reveals that the best choice of the parameters $\mu_1,\cdots,\mu_K$ for given $n_1,\cdots,n_K$ is

$$\mu_i = n_i^{-1}\sum_{j=S_{i-1}+1}^{S_i}E\xi_j, \qquad \text{Variant I} \tag{12a}$$

$$\mu_i = n_i^{-1}\sum_{j=S_{i-1}+1}^{S_i}E\eta_j, \qquad \text{Variant II.} \tag{12b}$$

The value of MSE that results when the $\mu_i$ are chosen in accordance with (12) is

$$D = n^{-1}\left(E|\boldsymbol{x}|^2 - \sum_{i=1}^{K}n_i\mu_i^2\right). \tag{13}$$

We stress that the derivation of (12) and (13) did not assume the $x_k$ to be statistically independent or identically distributed. However, when the components of $\boldsymbol{x}$ are highly dependent, permutation codes perform relatively poorly even when the optimum $\mu_i$ are used.

The performance of optimum source permutation codes of a given rate usually improves as the block length $n$ increases. Since it is easy to encode permutation codes optimally even for large $n$, it therefore is desirable to have a method of generating optimum Variant I and Variant II codes of a specified rate and block length $n \gg 1$. However, when $n$ is large, there are many ways in which $K$ and the group sizes $n_1,n_2,\cdots,n_K$ can be chosen so that the code rate $R$ [given by (1) and (5a)] closely approximates a specified value. We now describe an iterative technique that searches for the values of $K$ and $n_1,\cdots,n_K$ that minimize $D$ of (13) for a specified rate and block length.

Define

$$p_i = n_i/n, \qquad i = 1,2,\cdots,K. \tag{14}$$

Then, if each $n_i$ is large, we can use Stirling's formula to approximate the rate $R$ by

$$R \approx \hat{R} \triangleq \begin{cases} -\sum_{i=1}^{K} p_i \log p_i, & \text{Variant I} \\ 1 - \sum_{i=1}^{K} p_i \log p_i, & \text{Variant II } (\mu_K > 0). \end{cases} \tag{15}$$

Furthermore, the MSE for the optimum $\{\mu_i\}$ of (12) is given exactly by

$$D = n^{-1}E|x|^2 - \sum_{i=1}^{K} p_i\mu_i^2. \tag{16}$$

Treating (15) as an equality, we can minimize $D$ with respect to $p_1,p_2,\cdots,p_K$ subject to the rate constraint. The $p_i$ that result are

$$p_i = 2^{-\beta\mu_i^2}/(\textstyle\sum_{j=1}^{K} 2^{-\beta\mu_j^2}), \tag{17}$$

where $\beta$ is chosen so that $\hat{R}$ of (15) equals some specified rate value $R^*$.

Note that, in actuality, we do not have an analytic solution for the best $n_i$ for three reasons. First, although $n_i = np_i$ according to (14), $np_i$ usually is not an integer. Second, each $p_i$ depends on all the $\mu_i$, each of which is, in turn, a complicated function of all the $n_i$ via (12) and (8). Third, the above procedure assumes that $K$ is known, whereas we actually want to find the best $K$. We attack these obstacles as follows. With $K$ temporarily held fixed, we iterate (12), (17), and (14) in that order until the same $\{n_i\}$ set appears on two successive iterations, indicating consistency. The best $K$ then may be found via a simple search routine. An algorithm of this sort that we used to generate good permutation codes is described in Appendix II. Fairly broad conditions under which this algorithm is capable of generating permutation codes that are truly optimum in the MSE sense are derived in Appendixes III and IV. In particular, independent zero-mean Gaussian data satisfy these conditions for both Variant I and Variant II codes.

## IV. Performance for Gaussian Data

The performance of Variant I and Variant II codes generated by the computer algorithm of Appendix II has been calculated for diverse rates and block length $n = 400$ for independent Gaussian data. Smooth curves drawn through the resulting $(R,D)$ points are shown in Fig. 1. Also shown is the lower bound provided by Shannon's [6] rate-distortion function formula

$$R(D) = \tfrac{1}{2}\log_2(\sigma^2/D), \tag{18}$$

where $\sigma^2$ is the variance of the source. For $1 \leq R \leq 3$ the performance achieved by the permutation codes lies nearly on the line

$$R = \tfrac{1}{4} + \tfrac{1}{2}\log_2(\sigma^2/D) \qquad \text{bits/letter.} \tag{19}$$

It is interesting to note that (19) is the same formula that Goblick and Holsinger [7] report to be an accurate fit to the lower envelope of the performance curves of single-sample quantizers with uniform spacing subject to entropy coding. Further comparison of permutation codes and entropy-coded quantizers is given in Section VI.

For rates $R < 1$, the performance of Variant I codes approaches that of the rate-distortion curve. In particular, the lowest rate Variant I code of block length $n$ has $n_1 = 1$ and $n_2 = n - 1$. Its optimum $\mu_i$ are $\mu_1 = E\xi_1$ and $\mu_2 = -\mu_1/(n-1)$. This code is a simplex code, and its performance is given by

$$R = n^{-1}\log n \tag{20}$$

$$D/\sigma^2 = 1 - (n-1)^{-1}\mu_1^2. \tag{21}$$

The asymptotic behavior of $\mu_1$ has been studied extensively by Gumbel [8], who has shown that for independent Gaussian data

$$\mu_1 \sim \sqrt{2\log_e n} + \text{constant}. \tag{22}$$

Combining (20) through (22), we have for large $n$

$$D/\sigma^2 \approx 1 - 2R\log_e 2 + 0(n^{-1}\sqrt{\log n}). \tag{23}$$

When (23) is compared with the asymptotic form of (18) for small $R$, we see that the two agree for large $n$. Thus, the simplex Variant I codes are asymptotically optimum for small $R$ (large $n$). By way of contrast, it is easily shown that the best quantizer with only two representation points for $n$ independent Gaussian source outputs behaves as

$$D/\sigma^2 = 1 - (2/\pi)R\log_e 2, \tag{24}$$

which is not asymptotically optimum.

The computer algorithm described in Appendix II yielded somewhat better Variant I codes than those that Dunn [3] had been able to find by educated guesswork. The following two $n = 400$ codes are easily compared:

Dunn $\{n_i\} = (5,5,35,40,65,100,65,40,35,5,5)$

$R = 2.86367, \quad D = 0.03389$

computer $\{n_i\} = (1,2,7,20,46,77,94,77,46,20,7,2,1)$

$R = 2.79184, \quad D = 0.03362.$

The computer-generated code achieves approximately the same $D$ at a smaller value of $R$.

The performance of maximum-rate Variant I codes ($K = n$, $n_i = 1$, $\mu_i = E\xi_i$) as a function of block length is

$$(R,D/\sigma^2) = (n^{-1}\log n!,\ 1 - n^{-1}S),$$

where

$$S = \sum_{j=1}^{n} (E[\xi_j])^2.$$

Values of $S$ for independent Gaussian data have been tabulated by David *et al.* [9] for $n \leq 400$. Maximum-rate codes are seen to perform quite poorly relative to the rate-distortion bound. The sharp upswing at high rates of the permutation-code performance curves in Fig. 1 may be explained as follows. The $E[\xi_j]$ (or $E[\eta_j]$) values for neighboring values of $j$ do not differ by very much. Unless one

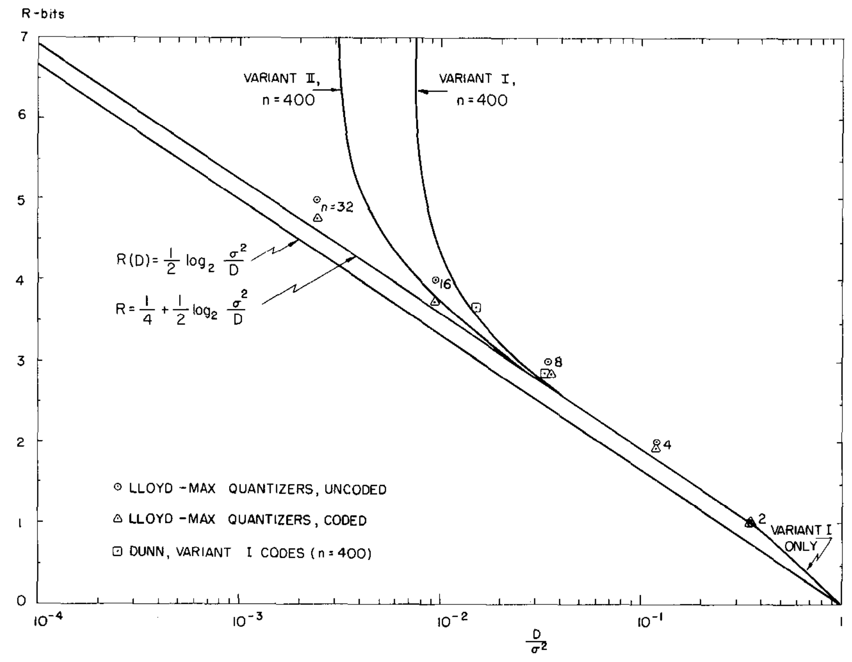

Fig. 1. Comparison of permutation codes, quantizers, and $R(D)$.

collects such neighboring values into groups and represents them by a common $\mu$, the code will contain clusters of codewords with almost identical coordinates in $n$-space. A good code is characterized by widely scattered codewords, so this clustering is highly undesirable and serves to further underscore the importance of the optimum grouping technique developed in Section III and Appendix II.

The only table of Gaussian $E\eta_j$ previously available was that of Klotz [4] for $n \leq 10$. Since much larger values of $n$ had to be investigated, new tables were generated. The Gaussian $E\eta_j$ also form the reference data base in applications of a popular nonparametric statistical test for symmetry of probability densities called the Fraser normal scores test [10]. Accordingly, we have included tables of $E\eta_j$ for $n = 100, 200, 300$, and 400 for reference purposes in Appendix V, together with recursive formulas that can be used to generate tables for the intervening values of $n$.

## V. The Instrumentation Problem

The encoding methods for permutation codes described in Section II assumed that encoding consisted simply of replacing the source word by the closest codeword. In actuality, for transmission over a channel or for storage in a memory, it is necessary to generate a digital representation of the appropriate codeword. This entails performing two operations: sorting and coding.

The most straightforward approach to the sorting problem would be to use one of several available techniques, such as quicksort [11], samplesort [12], or treesort [13], to completely order the source samples (or their absolute values in the case of Variant II codes), and then to group the time indices of the largest $n_1$ samples into one set, the time indices of the next largest $n_2$ samples into a second set, and so on. If $n$ is large and many of the group sizes $n_i$ also are fairly large, it will pay to capitalize on the latter fact to avoid having to perform a complete ordering. In such cases the sorting problem is not trivial. Moreover, its importance is intensified by the fact that the complexity of sorting via complete ordering is known [11] to grow as $n \log n$, which it turns out would dominate the complexity of the rest of the encoder in the limit of large $n$.

Next, the sorted index arrangements must be coded. For small $n$ one might arbitrarily assign a different binary codeword of length

$$\log_2 M = \log_2 n! - \sum_{i=1}^{K} \log_2 n_i!$$

to each of the $M$ different index arrangements and use table look-up. As $n$ grows, this becomes unmanageable rapidly. If $n_i = 1$ for $i = 1,\cdots,K$, then known permutation encoding techniques [14] may be used whose complexity grows linearly with $n$. However, codes with $n_i = 1$ for all $i$ perform poorly for reasons discussed in Section IV. In Appendix VI we describe another technique, based on Jelinek's [15] version of Elias's noiseless coding technique, that applies to all possible values of $\{n_i\}$ and also has been shown [15] to grow only linearly with $n$. It has been called to our attention that a comparable method appears in a patent application of Slepian [16] covering permutation coding schemes for channels.

For Variant II codes it is necessary to encode the sign pattern, also. This poses no problem when all $2^n$ sign patterns are equally likely, as in the case when the source outputs are zero mean and independent. For nonequiprobable sign patterns, a Huffman-type coding technique might be most applicable.

The source decoder is an instrument that translates the "index codewords" into representation words $(y_1,y_2,\cdots,y_n)$, where $y_k = \pm\mu_i$ if $k$ belongs to the $i$th index set; the sign is $+$ for Variant I codes and is determined from the sign portion of the codeword otherwise. A procedure that accomplishes this task with a complexity that also grows only linearly with $n$ is described in Appendix VI.

## VI. Permutation Codes Versus Quantizers

It was noted in Section IV that the MSE performance of permutation codes for an independent Gaussian source is bounded from below by the lower envelope (19) of the performance curves of single-sample quantizers with uniformly spaced levels. These are the best single-sample quantizers presently known in the $(R,D)$ sense [7], [17]. In order for permutation codes to perform close to the curve given by (19) for high $R$, they must have very large block lengths. However, permutation encoding, as described in Sections II and V, is so simple that it nevertheless may be easier to implement such codes than to tackle the buffer overflow problems [18], [19] associated with entropy coding of the highly nonequiprobable outputs of a uniformly spaced multilevel quantizer. In the absence of entropy coding, the best single-sample quantizers are those of Lloyd [20] and Max [21]. For $n = 400$, we see from Fig. 1 that optimum Variant I and Variant II codes outperform uncoded Lloyd–Max quantizers for $R < 3.7$ and $R < 4.5$, respectively.

The relationship between permutation codes and single-sample quantizers perhaps is illuminated further by the following observation. A Variant II code with $K = 1$ (hence $n_1 = n$ and $R = 1$) has representation points $(\pm\sigma\sqrt{2/\pi}, \pm\sigma\sqrt{2/\pi},\cdots,\pm\sigma\sqrt{2/\pi})$, which are identical to those of an optimum 1-bit quantizer. Its performance is given by the point marked in Fig. 1 by the circle and triangle for $n = 2$ and lies almost right on the curve given by (19).

It is worth stressing that the nonparametric nature of permutation encoding makes it well suited to situations in which the source statistics are either unknown or time varying. In such situations one could proceed as follows. Select a code rate $R$ close to the capacity of the channel being used for transmission, and a block length $n$ as long as practical considerations will permit. Partition $n$ into a grouping $\{n_i\}$ that has rate $R$. A good choice for $\{n_i\}$, in the absence of any knowledge to the contrary, is the optimum Gaussian MSE partition for rate $R$ found by the method described in Appendix II. Encode successive $n$-vectors as usual, but also collect sample order statistics. After a duration comparable to the time constant of the phenomenon responsible for the time-varying behavior (if this is known), send the $\{\mu_i\}$ of (12) as calculated from the sample order statistics.[1] Then convert the received permutation rank orders into numerical vectors using this $\{\mu_i\}$ set. Note that the sample $\{\mu_i\}$ yield even better performance than would the "true" $\{\mu_i\}$ were they known. Hence, this may be a desirable procedure to follow even when the source statistics are known.

[1] Rate probably can be conserved by sending only the changes in the $\mu_i$.

It should be appreciated that comparable simplicity cannot be obtained with single-sample quantization techniques in the face of unknown, possibly time-varying source statistics. In order to match the quantization levels to the data, one must store *all* the data first, then calculate and send the levels, and finally quantize the data letter by letter. This amounts to a complicated block coding scheme that requires a much longer coding delay (at least at the transmitter) than does the permutation scheme. In addition, if it is desired to code the quantizer outputs in order to preserve rate, one must send both the sample relative frequencies of the quantization bins and the variable-length code being employed.

## VII. Summary

The principal results of this paper are reiterated below.

1) We have developed techniques for optimizing the parameters of permutation codes for sources.

2) We have shown that in the Gaussian MSE case permutation codes are asymptotically ideal at low rates and perform as well as the best entropy-coded quantizers at intermediate rates.

3) We have argued that permutation codes, in large part because of the simple, essentially nonparametric nature of their optimum encoding algorithms, compare favorably with the other source encoding techniques presently in use, especially in the commonly encountered case in which the source statistics are unknown and/or possibly time varying.

## Acknowledgment

The authors are indebted to Dr. D. Slepian and Dr. A. D. Wyner of the Bell Telephone Laboratories for informative discussions, and to L.-W. Fung of Cornell University for programming their computer calculations.

## Appendix I
## Proof of Theorem 1

We shall establish the theorem for Variant I codes and a distortion measure of the form

$$d(x,y) = \sum_{t=1}^{n} f(|x_t - y_t|) \tag{25}$$

where $f(\cdot)$ is nonnegative, nondecreasing, and convex $\cup$ for positive arguments. The extensions to arbitrary nondecreasing $g(\cdot)$ and to Variant II codes noted in the theorem statement should be obvious.

From the additive nature of (25), it suffices to show that, if $x_1 \geq x_2 \geq \cdots \geq x_n$, then the basic codeword

$$y_1 = (\mu_1,\cdots,\mu_1,\mu_2,\cdots,\mu_2\cdots,\mu_K,\cdots,\mu_K)$$

is the one that minimizes $d(x,y)$. Furthermore, once this has been established for $n = 2$, it is easily established for $n > 2$ by induction. Hence, let $n = 2$ and let $y_1 = (v_1,v_2)$. There are six cases to consider, namely,

| | | | |
|---|---|---|---|
| *Case 1:* | $x_1 \geq v_1 \geq v_2 \geq x_2$ | *Case 4:* | $v_1 \geq x_1 \geq x_2 \geq v_2$ |
| *Case 2:* | $x_1 \geq v_1 \geq x_2 \geq v_2$ | *Case 5:* | $v_1 \geq x_1 \geq v_2 \geq x_2$ |
| *Case 3:* | $x_1 \geq x_2 \geq v_1 \geq v_2$ | *Case 6:* | $v_1 \geq v_2 \geq x_1 \geq x_2.$ |

In each case we must establish that

$$f(|x_1 - v_1|) + f(|x_2 - v_2|) \leq f(|x_1 - v_2|) + f(|x_2 - v_1|). \tag{26}$$

It is clear that Cases 4, 5, and 6 will follow by symmetry once (26) is established for Cases 1, 2, and 3, respectively.

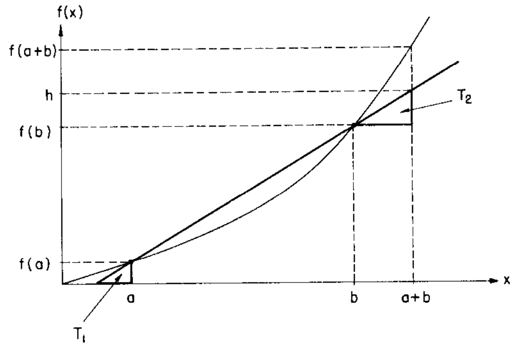

Fig. 2. Sketch of $f(x)$ and the similar triangles $T_1$ and $T_2$.

*Case 1:* We have $v_1 - x_2 \geq v_2 - x_2 \geq 0$ and $x_1 - v_2 \geq x_1 - v_1 \geq 0$. Hence, (26) follows from the monotonicity of $f(\cdot)$.

Before treating Cases 2 and 3, we note that if we can establish (26) for $\tilde{f}(|x - v|) = f(|x - v|) - f(0)$, then it clearly will hold for $f(\cdot)$ as well. Hence we lose no generality by assuming $f(0) = 0$.

*Lemma:* If $a \geq 0$ and $b \geq 0$, then $f(a) + f(b) \leq f(a + b)$.

*Proof:* See Fig. 2. A straight line is drawn through the points $(a, f(a))$ and $(b, f(b))$. Since $f(\cdot)$ is convex $\cup$ and $f(0) = 0$, the line intersects the abscissa at a nonnegative value. Triangles $T_1$ and $T_2$ are similar. The base of $T_2$ is larger than the base of $T_1$, so the altitude of $T_2$ is larger than $f(a)$, the altitude of $T_1$. Thus the straight line intersects the point $(a + b, h)$ where

$$f(a) + f(b) \leq h \leq f(a + b). \qquad \text{Q.E.D.}$$

*Case 2:* We have

$$\begin{aligned} f(|x_1 - v_1|) + f(|x_2 - v_2|) &\leq f(|x_1 - x_2|) + f(|x_2 - v_2|) \\ &\leq f(|x_1 - v_2|) \\ &\leq f(|x_1 - v_2|) + f(|x_2 - v_1|) \end{aligned}$$

where the first inequality follows from $x_2 \leq v_1$ and monotonicity, the second from the lemma, and the third from nonnegativity.

*Case 3:* We have

$$f(|x_2 - v_2|) \leq f(|x_1 - v_2|),$$

since in this case $x_2 - v_2 \leq x_1 - v_2$ and $f$ is nondecreasing. Let

$$\tilde{f}(|\alpha|) = f(|\alpha + x_2 - v_1|) - f(|x_2 - v_1|).$$

Applying the lemma to $\tilde{f}(\cdot)$ yields

$$\begin{aligned} &f(|x_1 - v_1|) + f(|x_2 - v_2|) \\ &\quad = \tilde{f}(|x_1 - x_2|) + \tilde{f}(|v_1 - v_2|) + 2f(|x_2 - v_1|) \\ &\quad \leq \tilde{f}(|x_1 - x_2 + v_1 - v_2|) + 2f(|x_2 - v_1|) \\ &\quad = f(|x_1 - v_2|) + f(|x_2 - v_1|). \qquad \text{Q.E.D.} \end{aligned}$$

## Appendix II
## An Algorithm for Generating Good Permutation Codes

The following computational algorithm, based on the theory of Section II, generates permutation codes of a specified rate $R^*$ and block length $n$ that are good in the MSE sense. Fig. 3 provides a flowchart description of the algorithm. Fairly broad conditions under which the algorithm is capable of generating codes that are truly optimum are derived in Appendixes III and IV.

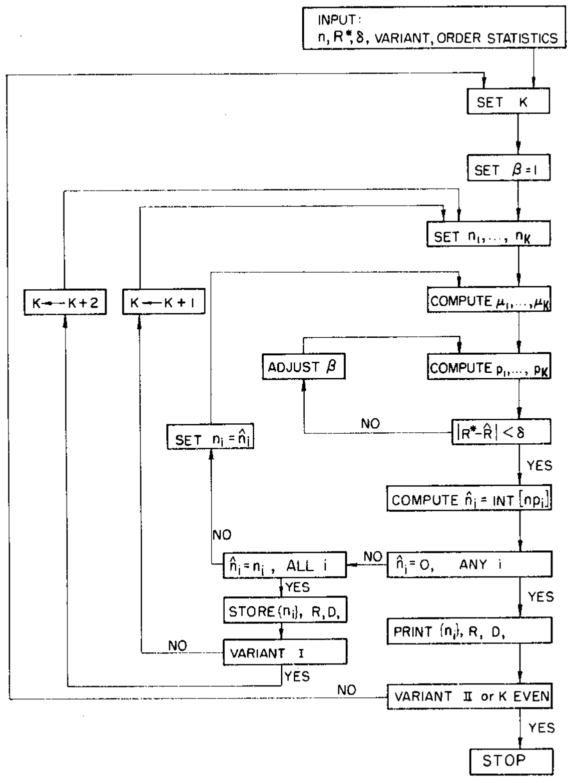

Fig. 3. Flow chart for permutation code algorithm.

1) Input $n$, $R^*$, Variant I or II, and $E\xi_j$ or $E\eta_j$ for $j = 1, \cdots, n$.

2) If Variant I, set $K$ equal to the smallest odd integer whose base 2 log exceeds $R^*$. If Variant II, set $K$ equal to the smallest integer whose base 2 log exceeds $R^*$.

3) Set[2] $\beta = 1$ and make the integers $n_i$, $i = 1, \cdots, K$, approximately equal; e.g., if $K$ divides $n$, set $n_i = n/K$ for all $i$.

4) Compute $\mu_1, \mu_2, \cdots, \mu_K$ from (13).

5) Evaluate the $p_i$ by (17). Adjust $\beta$ until (15) is satisfied for $\hat{R} = R^*$.[3]

6) Compute new $n_i$ as the closest integers to $np_i$ such that $\sum_{i=1}^{K} n_i = n$.

7) If $n_i = 0$ for any $i$, proceed to step 11).

8) If the new and old $n_i$ agree for all $i$, proceed to step 9). Otherwise, return to step 4).

9) Store $n_1, \cdots, n_K, D$, and the exact value of $R$ calculated from (2) and (5).

10) If Variant I, replace $K$ by $K + 2$, reduce the largest $n_i$ by 2, relabel $n_i$ as $n_{i+1}$ for $i = 1, \cdots, K - 2$, and put $n_1 = n_K = 1$. If Variant II, replace $K$ by $K + 1$, reduce the largest $n_i$ by 1, relabel $n_i$ as $n_{i+1}$ for $i = 1, \cdots, K - 1$, and put $n_1 = 1$. Return to step 4).

11) Print $\{n_i\}$, $R$, and $D$ stored in step 9). Go to step 13) unless $K$ is odd and code is Variant I.

12) Set $K$ equal to the smallest even integer whose base 2 log exceeds $R^*$, and return to step 3).

13) Stop.

[2] The desired value of $\beta$ usually is positive. The reason is that (17) implies that $p_i$, and hence $n_i$, decreases with increasing $\eta_i^2$ for $\beta > 0$, a property that is shown in Appendix III to be desirable in a broad class of interesting problems.

[3] It follows from (17) that

$$\frac{d\hat{R}}{d\beta} = -\beta \log_2 e \left[ \sum_{i=1}^{K} p_i \mu_i^4 - \left( \sum_{i=1}^{K} p_i \mu_i^2 \right) \right] = -\beta \log_2 e \operatorname{var}(\mu^2),$$

so $\hat{R}$ is a monotonic decreasing function of $\beta$. Hence, the value of $\beta$ that satisfies (15) is unique and can be determined rapidly by a modified Newton-Raphson method.

Table I illustrates the convergence of a typical sequence of $\{n_i\}$ sets obtained with this algorithm in a practical example.

It is shown in Appendix III that, if $E\eta_j$ is a convex-$\cup$ function of $j$, then the Variant II algorithm can generate permutation codes that are truly optimum in the MSE sense. Moreover, it is shown in Appendix IV that, in the important case of statistically independent source outputs whose absolute values are identically distributed according to a probability density function $f_{|X|}(\cdot)$ that is nonincreasing on $[0,\infty)$, $E\eta_j$ is indeed a convex-$\cup$ function. We conjecture, but unfortunately are unable to prove that, when this convexity prevails, the Variant II algorithm always produces a code whose $K$ and $\{n_i\}$ differ from their optimum counterparts by at most 1; the unit inaccuracies are caused by the diophantine nature of the problem.

The corresponding situation for Variant I codes is as follows. The Variant I algorithm can generate MMSE codes whenever $E\xi_j$, is convex $\cup$ for $j \in \{1,2,\cdots,[n/2]\}$ and is convex $\cap$ for $j \in \{[n/2]+1,\cdots an\}$. When this convexity prevails, we conjecture that the algorithm always will generate a code that is optimum in the diophantine sense described previously. In the case of independent identically distributed source outputs, the desired convexity will be in effect whenever $f_X(\cdot)$ is unimodal, symmetric, and zero mean.

TABLE I
SEQUENCE OF $\{n_i\}$ SETS FOR INDEPENDENT STANDARDIZED GAUSSIAN DATA (VARIANT I, $n = 400$, $R^* = 1.5$, $K$ ODD)

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $K = 3$ | | | | 133 | 134 | 133 | | | | |
| | | | | 100 | 200 | 100 | | | | |
| | | | | 99 | 202 | 99 | | | | same |
| | | | | 99 | 202 | 99 | | | | |
| $K \to K+2$ | | | 1 | 99 | 200 | 99 | 1 | | | |
| | | | 2 | 90 | 216 | 90 | 2 | | | |
| | | | 2 | 85 | 226 | 85 | 2 | | | |
| | | | 2 | 84 | 228 | 84 | 2 | | | |
| | | | 2 | 83 | 230 | 83 | 2 | | | same |
| | | | 2 | 83 | 230 | 83 | 2 | | | |
| $K \to K+2$ | | 1 | 2 | 83 | 228 | 83 | 2 | 1 | | |
| | | 1 | 3 | 77 | 238 | 77 | 3 | 1 | | |
| | | 1 | 4 | 75 | 240 | 75 | 4 | 1 | | |
| | | 1 | 4 | 74 | 242 | 74 | 4 | 1 | | same |
| | | 1 | 4 | 74 | 242 | 74 | 4 | 1 | | |
| $K \to K+2$ | 1 | 1 | 4 | 74 | 240 | 74 | 4 | 1 | 1 | |
| | 0 | 1 | 4 | 69 | 252 | 69 | 4 | 1 | 0 | |
| Print | $n_1 = n_7 = 1$, $n_2 = n_6 = 4$, $n_3 = n_5 = 74$, $n_4 = 242$ $R = 1.47514$, $D = 0.18595$ | | | | | | | | | |

## APPENDIX III

## A NECESSARY CONDITION FOR OPTIMALITY OF THE ALGORITHM OF APPENDIX II

Observe from (14) and (17) that the $p_i$, and hence the $n_i$, always are monotonic functions of the $\mu_i^2$. This is a consequence of the fact that in step 5) the $\mu_i$ are not permitted to vary in accordance with (12) while the $p_i$ (hence the $n_i$) are being optimized. As a result, the algorithm of Appendix II can converge only to codes that are characterized by groupings $\{n_i\}$ that generate $\mu_i$ via (12), which are such that $n_i$ actually varies monotonically with $\mu_i^2$. It now will be shown that, for a broad and interesting class of examples, the optimum code indeed is characterized by an inverse relationship between $n_i$ and $\mu_i^2$. In such cases, therefore, it is possible for the said optimum code to be found by the algorithm of Appendix II with $\beta > 0$.

For simplicity, the ensuing discussion will be restricted to Variant II codes only, after which the implications for Variant I codes will be discussed. Since the optimum $\mu_i^2$ for Variant II codes necessarily decrease with increasing $i$, it will suffice to show that the optimum $n_i$ are monotonic increasing with $i$. From (12b) the values of the $\mu_i$ that minimize the MSE for given $\{n_i\}$ are

$$\mu_i = \frac{1}{n_i} \sum_{j=n_1+\cdots+n_{i-1}+1}^{n_1+\cdots+n_i} \bar{\eta}_j, \qquad i = 1,\cdots,K \tag{27}$$

where $\bar{\eta}_j \triangleq E\eta_j$. From (13) the resulting minimum MSE is

$$D = n^{-1}\left[E|X|^2 - \sum_{i=1}^{K} n_i\mu_i^2\right]. \tag{28}$$

*Theorem 2:* Suppose $\bar{\eta}_j$ is a convex-$\cup$ function of $j$, i.e., suppose

$$\Delta^2\bar{\eta}_j = \bar{\eta}_{j+2} - 2\bar{\eta}_{j+1} + \bar{\eta}_j \geq 0, \qquad 1 \leq j \leq n-2. \tag{29}$$

Then the optimum $n_i$ increase monotonically with $i$.

*Proof:* We shall show that, if $n_{l-1} > n_l$ for some $l$, then $D$ can be decreased by reversing the roles of $n_{l-1}$ and $n_l$. That is, if a new grouping $\{n_i'\}$ is defined by

$$n_i' = \begin{cases} n_i, & i \neq l-1 \text{ or } l \\ n_l, & i = l-1 \\ n_{l-1}, & i = l \end{cases} \tag{30}$$

then

$$D' < D, \tag{31}$$

where $D'$ is defined by the right-hand side of (28) with the $n_i$ replaced by the $n_i'$ and with the $\mu_i$ recalculated from (27) using the $n_i'$ in place of the $n_i$.

We establish (31) as follows. Let $L = n_1 + n_2 + \cdots + n_{l-2}$, and let

$$b = n_{l-1} = n_l' > n_l = n_{l-1}' = a. \tag{32}$$

Then

$$\begin{aligned} n(D - D') = {} & \frac{1}{a}(\bar{\eta}_{L+1} + \cdots + \bar{\eta}_{L+a})^2 + \frac{1}{b}(\bar{\eta}_{L+a+1} + \cdots + \bar{\eta}_{L+a+b})^2 \\ & - \frac{1}{b}(\bar{\eta}_{L+1} + \cdots + \bar{\eta}_{L+b})^2 \\ & - \frac{1}{a}(\bar{\eta}_{L+b+1} + \cdots + \bar{\eta}_{L+a+b})^2. \end{aligned}$$

It follows after some algebraic manipulation that

$$\begin{aligned} nab(D - D') &= a[(y+z)^2 - (x+y)^2] + b(x^2 - z^2) \\ &= (x - z)[(b-a)(x+z) - 2ay] \end{aligned} \tag{33}$$

where

$$x \triangleq \sum_{j=1}^{a} \bar{\eta}_{L+j}, \qquad y \triangleq \sum_{j=a+1}^{b} \bar{\eta}_{L+j}, \qquad z \triangleq \sum_{j=b+1}^{a+b} \bar{\eta}_{L+j}. \tag{34}$$

But $x > z$ because of (32) and the fact that $\bar{\eta}_1 > \bar{\eta}_2,\cdots,\bar{\eta}_n > 0$, so $D \geq D'$ iff $(b-a)(x+z) - 2ay \geq 0$, or equivalently iff

$$\frac{x+z}{a} - \frac{2y}{b-a} \geq 0. \tag{35}$$

Upon substituting (34) into (35), we see that proving the theorem has been reduced to establishing the inequality

$$\frac{1}{a}\sum_{j=1}^{a} \bar{\eta}_{L+j} - \frac{2}{b-a}\sum_{j=a+1}^{b} \bar{\eta}_{L+j} + \frac{1}{a}\sum_{j=b+1}^{a+b} \bar{\eta}_{L+j} \geq 0. \tag{36}$$

That (36) is indeed a valid inequality is a consequence of the convexity hypothesis (29). To see this, plot the points $\bar{\eta}_j$ versus $j$ and then connect them by straight lines. This results in a piecewise linear convex-$\cup$ function of a continuous variable, call it $\bar{\eta}(x)$, $1 \leq x \leq n$. Define the function $\bar{\eta}^*(x)$, $1 \leq x \leq n$, to equal $\bar{\eta}(x)$ except in the interval $L + a + 1 < x < L + b$, wherein it consists of a straight line segment joining $\bar{\eta}(L + a + 1)$ to $\bar{\eta}(L + b)$. Then $\bar{\eta}^*(x)$ also is convex $\cup$. Moreover the three terms on the left side of (36) each are upper bounds to the corresponding terms in the inequality

$$\bar{\eta}^*\left(L + \frac{a+1}{2}\right) - 2\bar{\eta}^*\left(L + \frac{a+1}{2} + \frac{b}{2}\right) + \bar{\eta}^*\left(L + \frac{a+1}{2} + b\right) \geq 0 \tag{37}$$

which is valid because $\bar{\eta}^*$ is convex $\cup$. Hence, (36) also is valid and the theorem is proved.

The corresponding result for Variant I codes is stated below. The proof is a reasonably straightforward extension of that of Theorem 2 and therefore is omitted.

*Theorem 3:* If $E\xi_j$ is convex $\cup$ for $j \in \{1,2,\cdots,[n/2]\}$ and convex $\cap$ for $j \in \{[n/2]+1,\cdots,n\}$, then the optimum $n_i$ vary inversely with the optimum $\mu_i^2$.

## Appendix IV

## Convexity of Order Statistics for Independent Data

Theorem 2 leads us to search for conditions under which the $\eta_j$ will satisfy the convexity hypothesis (29). The following theorem provides an important class of examples in which (29) is satisfied.

*Theorem 4:* If the source outputs are statistically independent and identically distributed in such a way that the probability density function $f_{|x|}(\cdot)$ of the absolute value of a source output is a nonincreasing function of positive argument, then the $\eta_j$ satisfy the convexity hypothesis (29).

*Proof:* The starting point for the proof is Pearson's [22] formula for the average difference between the $j$th and the $(j+1)$th largest of $n$ independent identically distributed random variables, namely,

$$\binom{n}{j}\int_{-\infty}^{\infty} \Phi^{n-j}(1-\Phi)^j\,dx,$$

where $\Phi(\cdot)$ is the cumulative distribution function common to each of the random variables. In the present instance we are interested in the ordered absolute source outputs in which case Pearson's formula reads

$$-\Delta\bar{\eta}_j = \binom{n}{j}\int_0^{\infty} \Phi^{n-j}(1-\Phi)^j\,dx \tag{38}$$

where

$$\Phi(x) = \int_0^x f_{|x|}(t)\,dt. \tag{39}$$

It follows that

$$\Delta^2\bar{\eta}_j = \Delta\bar{\eta}_{j+1} - \Delta_j\bar{\eta} = \int_0^{\infty}\left[\binom{n}{j}\Phi^{n-j}(1-\Phi)^j - \binom{n}{j+1}\Phi^{n-j-1}(1-\Phi)^{j+1}\right]dx. \tag{40}$$

Upon changing the integration variable from $x$ to $z = \Phi(x)$, we obtain

$$\Delta^2\bar{\eta}_j = \int_0^1 h(z)\left[\binom{n}{j} z^{n-j}(1-z)^j - \binom{n}{j+1} z^{n-j-1}(1-z)^{j+1}\right]dz \tag{41}$$

where

$$h(z) = \frac{1}{f_{|x|}(x)} = \frac{1}{f_{|x|}(\Phi^{-1}(z))}. \tag{42}$$

Observe that, since $z$ is an increasing function of $x$ and vice versa, $h(z)$ is a nondecreasing function of $z$ because $f_{|x|}(x)$ has been assumed to be a nonincreasing function of $x$.

Using the beta density

$$\beta_{\mu,\upsilon}(x) = \frac{\Gamma(\mu+\upsilon)}{\Gamma(\mu)\Gamma(\upsilon)}(1-x)^{\mu-1}x^{\upsilon-1}, \qquad 0 \le x \le 1, \tag{43}$$

we can rewrite (41) in the form

$$(n+1)\Delta^2\bar{\eta}_j = \int_0^1 h(z)[\beta_{j+1,n-j+1}(z) - \beta_{j+2,n-j}(z)]\,dz. \tag{44}$$

Assume temporarily that $h(z)$ is differentiable, and define

$$b(z) = \int_0^z [\beta_{y+1,n-j+1}(t) - \beta_{j+2,n-j}(t)]\,dt. \tag{45}$$

Integrate (44) by parts, noting that $b(0) = b(1) = 0$, to obtain

$$(n+1)\Delta^2\bar{\eta}_j = -\int_0^1 h'(z)b(z)\,dz. \tag{46}$$

Next, use the fact that $\beta_{l,m}(\cdot)$ is the probability density that governs the $l$th largest of $l+m-1$ independent random variables distributed uniformly on [0,1]. Accordingly,

$$b(z) = P[U^{j+1} \le z] - P[U^{j+2} \le z], \tag{47}$$

where $U^k$ denotes the $k$th largest of $n+1$ independent uniformly distributed random variables, and $P[\cdot]$ denotes the probability of the event within the brackets. But $(U^{j+1} \le z) \Rightarrow (U^{j+2} \le z)$, so

$$b(z) \le 0, \qquad 0 \le z \le 1. \tag{48}$$

When (48) is coupled with the fact that $h'(z) \ge 0$ because $h(z)$ is nondecreasing, it follows from (46) that $\Delta^2\bar{\eta}_j \ge 0$ as required for the application of Theorem 2. The extension to piecewise differentiable $h(\cdot)$ is straightforward and therefore is omitted.

The result corresponding to Theorem 4 for ordinary rather than absolute order statistics is as follows.

*Theorem 5:* If the source outputs are statistically independent and identically distributed according to a unimodal symmetric zero-mean probability density, then the $E\xi_j$ satisfy the convexity relations necessary for the application of Theorem 3.

## Appendix V

## Tables of Fraser Normal Scores

Let $\boldsymbol{x} = (x_1,\cdots,x_n)$ be a vector of independent identically distributed standardized normal (Gaussian) random variables. Define the random variable $\eta_j$ to be the $j$th largest of the absolute values of the components of $\boldsymbol{x}$, $j = 1,2,\cdots,n$. The Fraser normal scores $E\eta_j$ are used in a statistical test for the symmetry of probability distributions [4], [10] and in the construction of optimum Variant II source permutation codes as described in this paper. The one table of Fraser normal scores previously published [4] covers only the range $n \le 10$.

In Table II we give the Fraser normal scores $E\eta_j$, $j = 1,\cdots,n$, to an accuracy of nine decimal places for $n = 100$, 200, 300, and 400. For intermediate values of $n$, one can use the general recursive formulas [23]

$$E[\alpha_{n,j+1}] = \frac{1}{j}\{nE[\alpha_{n-1,j}] - (n-j)E[\alpha_{n,j}]\} \tag{49}$$

$$E[\alpha_{n-1,j}] = \frac{1}{n}\{jE[\alpha_{n,j+1}] + (n-j)E[\alpha_{n,j}]\} \tag{50}$$

where $\alpha_{n,j}$ is the $j$th largest random variable from an independent identically distributed sample of size $n$. The rate of accumulation of computational error is smaller when one goes from $n$ to $n-1$ by means of (50) than when one goes from $n$ to $n+1$ by means of (49) [23].

The claim of nine-place accuracy in Table II is based on the following test. Tables for $n = 100$, 200, 300, and 400 originally were computed via a double precision routine to 12 decimal places. Then (50) was used to iterate from 400 down to 300, whereupon all entries in the table for 300 so obtained were observed to agree with those of the original table for 300 for at least nine digits after the decimal point. Similar consistency checks were run from 300 down to 200, 200 down o 100, and 100 dow n to 10.

## Appendix VI

## Encoding and Decoding of Permutations

The optimum permutation encoding procedures described in Theorem 1 result in a codeword of the form

$$y = (\mu_{i_1},\mu_{i_2},\cdots,\mu_{i_n}), \qquad 1 \le i_l \le K, \tag{51}$$

TABLE II
FRASER NORMAL SCORES FOR $n = 100, 200, 300,$ AND 400

| | N=100 | N=200 | | N=300 | | | N=400 | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| I | 001-100 | 001-100 | 101-200 | 001-100 | 101-200 | 201-300 | 001-100 | 101-200 | 201-300 | 301-400 |
| 1 | 2.746957688 | 2.968601388 | 0.671610465 | 3.091973030 | 0.964417163 | 0.429504477 | 3.176988992 | 1.147392353 | 0.673047448 | 0.317946729 |
| 2 | 2.414684194 | 2.658082090 | 0.663800636 | 2.792392288 | 0.957797257 | 0.424937531 | 2.884516377 | 1.141367801 | 0.669128262 | 0.314658612 |
| 3 | 2.231102282 | 2.488557454 | 0.656031193 | 2.629776289 | 0.951219100 | 0.420379467 | 2.726335277 | 1.135384415 | 0.665219340 | 0.311373910 |
| 4 | 2.101079543 | 2.369488849 | 0.648301256 | 2.516005081 | 0.944681886 | 0.415830154 | 2.615936123 | 1.129441426 | 0.661320570 | 0.308092581 |
| 5 | 1.999049864 | 2.276688471 | 0.640609971 | 2.427609676 | 0.938184835 | 0.411289468 | 2.530325827 | 1.123538084 | 0.657431840 | 0.304814583 |
| 6 | 1.914356194 | 2.200110671 | 0.632956502 | 2.354861495 | 0.931727184 | 0.406757281 | 2.459985017 | 1.117673661 | 0.653553040 | 0.301539874 |
| 7 | 1.841510255 | 2.134596155 | 0.625340038 | 2.292771160 | 0.925308192 | 0.402233470 | 2.400036156 | 1.111847448 | 0.649684061 | 0.298268413 |
| 8 | 1.777302014 | 2.077133977 | 0.617759785 | 2.238430223 | 0.918927138 | 0.397717912 | 2.347638131 | 1.106058755 | 0.645824797 | 0.295000159 |
| 9 | 1.719686002 | 2.025809124 | 0.610214970 | 2.189990651 | 0.912583321 | 0.393210486 | 2.300986850 | 1.100306910 | 0.641975140 | 0.291735070 |
| 10 | 1.667275698 | 1.979325631 | 0.602704840 | 2.146202870 | 0.906276054 | 0.388711070 | 2.258863120 | 1.094591260 | 0.638134988 | 0.288473105 |
| 11 | 1.619086221 | 1.936764293 | 0.595228657 | 2.106181242 | 0.900004673 | 0.384219547 | 2.220403270 | 1.088911168 | 0.634304237 | 0.285214225 |
| 12 | 1.574391837 | 1.897448674 | 0.587785705 | 2.069274480 | 0.893768527 | 0.379735799 | 2.184972447 | 1.083266012 | 0.630482784 | 0.281958388 |
| 13 | 1.532641733 | 1.860866008 | 0.580375280 | 2.034989185 | 0.887566982 | 0.375259709 | 2.152089902 | 1.077655188 | 0.626670531 | 0.278705555 |
| 14 | 1.493407553 | 1.826617985 | 0.572996698 | 2.002942317 | 0.881399421 | 0.370791161 | 2.121382552 | 1.072078106 | 0.622867377 | 0.275455687 |
| 15 | 1.456349315 | 1.794388813 | 0.565649289 | 1.972830347 | 0.875265242 | 0.366330043 | 2.092554867 | 1.066534192 | 0.619073224 | 0.272208743 |
| 16 | 1.421192460 | 1.763923729 | 0.558332398 | 1.944408536 | 0.869163858 | 0.361876242 | 2.065368631 | 1.061022884 | 0.615287975 | 0.268964684 |
| 17 | 1.387711932 | 1.735014111 | 0.551045386 | 1.917476557 | 0.863094694 | 0.357429644 | 2.039628921 | 1.055543637 | 0.611511536 | 0.265723471 |
| 18 | 1.355720838 | 1.707486879 | 0.543787627 | 1.891868292 | 0.857057191 | 0.352990141 | 2.015174144 | 1.050095917 | 0.607743811 | 0.262485066 |
| 19 | 1.325062201 | 1.681196793 | 0.536558510 | 1.867444402 | 0.851050803 | 0.348557622 | 1.991868796 | 1.044679203 | 0.603984707 | 0.259249429 |
| 20 | 1.295602835 | 1.656020736 | 0.529357434 | 1.844086818 | 0.845074997 | 0.344131979 | 1.969598088 | 1.039292987 | 0.600234132 | 0.256016523 |
| 21 | 1.267228716 | 1.631853402 | 0.522183815 | 1.821694591 | 0.839129252 | 0.339713105 | 1.948263900 | 1.033936773 | 0.596491995 | 0.252786308 |
| 22 | 1.239841433 | 1.608603983 | 0.515037079 | 1.800180708 | 0.833213058 | 0.335300893 | 1.927781676 | 1.028610077 | 0.592758206 | 0.249558748 |
| 23 | 1.213355433 | 1.586193608 | 0.507916663 | 1.779469616 | 0.827325919 | 0.330895238 | 1.908078015 | 1.023312426 | 0.589032676 | 0.246333803 |
| 24 | 1.187695844 | 1.564553313 | 0.500822017 | 1.759495278 | 0.821467349 | 0.326496035 | 1.889088777 | 1.018043358 | 0.585315318 | 0.243111438 |
| 25 | 1.162796750 | 1.543622439 | 0.493752603 | 1.740199629 | 0.815636872 | 0.322103182 | 1.870757575 | 1.012802421 | 0.581606043 | 0.239891613 |
| 26 | 1.138599793 | 1.523347333 | 0.486707892 | 1.721531327 | 0.809834025 | 0.317716575 | 1.853034561 | 1.007589174 | 0.577904768 | 0.236674293 |
| 27 | 1.115053049 | 1.503680299 | 0.479687366 | 1.703444746 | 0.804058354 | 0.313336112 | 1.835875447 | 1.002403187 | 0.574211406 | 0.233459439 |
| 28 | 1.092110100 | 1.484578742 | 0.472690517 | 1.685899152 | 0.798309414 | 0.308961694 | 1.819240700 | 0.997244036 | 0.570525873 | 0.230247015 |
| 29 | 1.069729271 | 1.466004454 | 0.465716847 | 1.668858018 | 0.792586772 | 0.304593221 | 1.803094879 | 0.992111311 | 0.566848088 | 0.227036985 |
| 30 | 1.047872993 | 1.447923031 | 0.458765867 | 1.652288464 | 0.786890001 | 0.300230593 | 1.787406083 | 0.987004607 | 0.563177968 | 0.223829312 |
| 31 | 1.026507273 | 1.430303372 | 0.451837099 | 1.636160775 | 0.781218687 | 0.295873713 | 1.772145492 | 0.981923531 | 0.559515431 | 0.220623960 |
| 32 | 1.005601246 | 1.413117271 | 0.444930070 | 1.620448011 | 0.775572421 | 0.291522483 | 1.757286979 | 0.976867697 | 0.555860399 | 0.217420893 |
| 33 | 0.985126796 | 1.396339062 | 0.438044320 | 1.605125662 | 0.769950806 | 0.287176807 | 1.742806780 | 0.971836728 | 0.552212791 | 0.214220074 |
| 34 | 0.965058230 | 1.379945318 | 0.431179394 | 1.590171369 | 0.764353452 | 0.282836588 | 1.728683217 | 0.966830253 | 0.548572529 | 0.211021467 |
| 35 | 0.945372004 | 1.363914600 | 0.424334846 | 1.575564669 | 0.758779976 | 0.278501733 | 1.714896457 | 0.961847912 | 0.544939536 | 0.207825039 |
| 36 | 0.926046481 | 1.348227229 | 0.417510239 | 1.561286791 | 0.753230003 | 0.274172147 | 1.701428307 | 0.956889351 | 0.541313735 | 0.204630752 |
| 37 | 0.907061730 | 1.332865102 | 0.410705142 | 1.547320466 | 0.747703167 | 0.269847737 | 1.688262037 | 0.951954221 | 0.537695050 | 0.201438571 |
| 38 | 0.888399343 | 1.317811523 | 0.403919132 | 1.533649777 | 0.742199110 | 0.265528410 | 1.675382222 | 0.947042186 | 0.534083406 | 0.198248462 |
| 39 | 0.870042280 | 1.303051061 | 0.397151793 | 1.520260013 | 0.736717477 | 0.261214074 | 1.662774614 | 0.942152911 | 0.530478730 | 0.195060390 |
| 40 | 0.851974732 | 1.288569421 | 0.390402715 | 1.507137554 | 0.731257925 | 0.256904637 | 1.650426022 | 0.937286071 | 0.526880947 | 0.191874319 |
| 41 | 0.834182005 | 1.274353338 | 0.383671497 | 1.494269761 | 0.725820115 | 0.252600010 | 1.638324206 | 0.932441347 | 0.523289985 | 0.188690215 |
| 42 | 0.816650409 | 1.260390474 | 0.376957741 | 1.481644885 | 0.720403715 | 0.248300102 | 1.626457791 | 0.927618426 | 0.519705772 | 0.185508044 |
| 43 | 0.799367169 | 1.246669338 | 0.370261059 | 1.469251984 | 0.715008400 | 0.244004824 | 1.614816183 | 0.922817002 | 0.516128237 | 0.182327771 |
| 44 | 0.782320339 | 1.233179203 | 0.363581065 | 1.457080850 | 0.709633851 | 0.239714088 | 1.603389503 | 0.918036774 | 0.512557310 | 0.179149361 |
| 45 | 0.765498731 | 1.219910047 | 0.356917383 | 1.445121943 | 0.704279753 | 0.235427804 | 1.592168516 | 0.913277448 | 0.508992920 | 0.175972781 |
| 46 | 0.748891847 | 1.206852482 | 0.350269639 | 1.433366336 | 0.698945802 | 0.231145887 | 1.581144585 | 0.908538735 | 0.505435000 | 0.172797997 |
| 47 | 0.732489824 | 1.193997708 | 0.343637468 | 1.421805659 | 0.693631694 | 0.226868249 | 1.570309613 | 0.903820351 | 0.501883481 | 0.169624975 |
| 48 | 0.716283376 | 1.181337463 | 0.337020506 | 1.410432057 | 0.688337134 | 0.222594803 | 1.559655999 | 0.899122019 | 0.498338294 | 0.166453681 |
| 49 | 0.700263752 | 1.168863976 | 0.330418398 | 1.399238146 | 0.683061831 | 0.218325464 | 1.549176602 | 0.894443467 | 0.494799375 | 0.163284081 |
| 50 | 0.684422691 | 1.156569933 | 0.323830793 | 1.388216975 | 0.677805501 | 0.214060148 | 1.538864701 | 0.889784427 | 0.491266655 | 0.160116142 |
| 51 | 0.668752382 | 1.144448438 | 0.317257344 | 1.377361996 | 0.672567863 | 0.209798768 | 1.528713961 | 0.885144636 | 0.487740070 | 0.156949832 |
| 52 | 0.653245433 | 1.132492981 | 0.310697709 | 1.366667029 | 0.667348643 | 0.205541242 | 1.518718409 | 0.880523838 | 0.484219555 | 0.153785115 |
| 53 | 0.637894836 | 1.120697413 | 0.304151550 | 1.356126234 | 0.662147571 | 0.201287486 | 1.508872400 | 0.875921779 | 0.480705045 | 0.150621960 |
| 54 | 0.622693940 | 1.109055914 | 0.297618536 | 1.345734093 | 0.656964380 | 0.197037416 | 1.499170599 | 0.871338212 | 0.477196477 | 0.147460333 |
| 55 | 0.607636425 | 1.097562975 | 0.291098337 | 1.335485378 | 0.651798811 | 0.192790950 | 1.489607955 | 0.866772893 | 0.473693788 | 0.144300202 |
| 56 | 0.592716277 | 1.086213371 | 0.284590628 | 1.325375136 | 0.646650608 | 0.188548006 | 1.480179682 | 0.862225583 | 0.470196916 | 0.141141534 |
| 57 | 0.577927768 | 1.075002146 | 0.278095089 | 1.315398669 | 0.641519519 | 0.184308502 | 1.470881239 | 0.857696047 | 0.466705798 | 0.137984295 |
| 58 | 0.563265435 | 1.063924590 | 0.271611404 | 1.305551514 | 0.636405295 | 0.180072357 | 1.461708318 | 0.853184055 | 0.463220373 | 0.134828454 |
| 59 | 0.548724061 | 1.052976226 | 0.265139260 | 1.295829432 | 0.631307695 | 0.175839490 | 1.452656822 | 0.848689380 | 0.459740581 | 0.131673978 |
| 60 | 0.534298661 | 1.042152793 | 0.258678346 | 1.286228389 | 0.626226480 | 0.171609821 | 1.443722856 | 0.844211800 | 0.456266362 | 0.128520834 |
| 61 | 0.519984461 | 1.031450233 | 0.252228358 | 1.276744544 | 0.621161413 | 0.167383270 | 1.434902712 | 0.839751096 | 0.452797656 | 0.125368991 |
| 62 | 0.505776893 | 1.020864678 | 0.245788993 | 1.267374238 | 0.616112264 | 0.163159758 | 1.426192856 | 0.835307054 | 0.449334404 | 0.122218415 |
| 63 | 0.491671572 | 1.010392437 | 0.239359951 | 1.258113979 | 0.611078805 | 0.158939204 | 1.417589919 | 0.830879462 | 0.445876547 | 0.119069076 |
| 64 | 0.477664289 | 1.000029985 | 0.232940937 | 1.248960439 | 0.606060813 | 0.154721531 | 1.409090686 | 0.826468113 | 0.442424029 | 0.115920940 |
| 65 | 0.463751001 | 0.989773956 | 0.226531659 | 1.239910434 | 0.601058067 | 0.150506661 | 1.400692088 | 0.822072802 | 0.438976791 | 0.112773976 |
| 66 | 0.449927818 | 0.979621128 | 0.220131825 | 1.230960924 | 0.596070351 | 0.146294514 | 1.392391189 | 0.817693331 | 0.435534776 | 0.109628152 |
| 67 | 0.436190992 | 0.969568420 | 0.213741148 | 1.222109000 | 0.591097452 | 0.142085014 | 1.384185185 | 0.813329501 | 0.432097928 | 0.106483438 |
| 68 | 0.422536912 | 0.959612882 | 0.207359345 | 1.213351879 | 0.586139160 | 0.137878082 | 1.376071388 | 0.808981118 | 0.428666192 | 0.103339800 |
| 69 | 0.408962094 | 0.949751685 | 0.200986134 | 1.204686893 | 0.581195268 | 0.133673643 | 1.368047229 | 0.804647993 | 0.425239511 | 0.100197207 |
| 70 | 0.395463171 | 0.939982118 | 0.194621235 | 1.196111486 | 0.576265573 | 0.129471619 | 1.360110241 | 0.800329937 | 0.421817831 | 0.097055628 |
| 71 | 0.382036890 | 0.930301579 | 0.188264372 | 1.187623208 | 0.571349875 | 0.125271934 | 1.352258061 | 0.796026767 | 0.418401098 | 0.093915030 |
| 72 | 0.368680102 | 0.920707570 | 0.181915270 | 1.179219706 | 0.566447977 | 0.121074512 | 1.344488423 | 0.791738300 | 0.414989257 | 0.090775383 |
| 73 | 0.355389755 | 0.911197690 | 0.175573657 | 1.170898722 | 0.561559685 | 0.116879277 | 1.336799150 | 0.787464359 | 0.411582256 | 0.087636654 |
| 74 | 0.342162991 | 0.901769633 | 0.169239264 | 1.162658085 | 0.556684807 | 0.112686154 | 1.329188150 | 0.783204768 | 0.408180040 | 0.084498813 |
| 75 | 0.328996640 | 0.892421179 | 0.162911823 | 1.154495711 | 0.551823154 | 0.108495067 | 1.321653413 | 0.778959355 | 0.404782558 | 0.081361829 |
| 76 | 0.315888210 | 0.883150193 | 0.156591067 | 1.146409593 | 0.546974543 | 0.104305941 | 1.314193008 | 0.774727948 | 0.401389757 | 0.078225671 |
| 77 | 0.302834890 | 0.873954618 | 0.150276734 | 1.138397799 | 0.542138788 | 0.100118702 | 1.306805074 | 0.770510382 | 0.398001585 | 0.075090310 |
| 78 | 0.289834038 | 0.864832474 | 0.143968561 | 1.130458472 | 0.537315711 | 0.095933275 | 1.299487821 | 0.766306491 | 0.394617991 | 0.071955713 |
| 79 | 0.276883082 | 0.855781850 | 0.137666288 | 1.122589820 | 0.532505133 | 0.091749585 | 1.292239526 | 0.762116112 | 0.391238925 | 0.068821850 |
| 80 | 0.263979511 | 0.846800904 | 0.131369657 | 1.114790118 | 0.527706880 | 0.087567558 | 1.285058526 | 0.757939088 | 0.387864336 | 0.065688687 |
| 81 | 0.251120875 | 0.837887859 | 0.125078410 | 1.107057702 | 0.522920778 | 0.083387122 | 1.277943220 | 0.753775259 | 0.384494173 | 0.062556194 |
| 82 | 0.238304780 | 0.829041000 | 0.118792293 | 1.099390966 | 0.518146658 | 0.079208202 | 1.270892064 | 0.749624472 | 0.381128388 | 0.059424338 |
| 83 | 0.225528883 | 0.820258670 | 0.112511051 | 1.091788360 | 0.513384352 | 0.075030726 | 1.263903565 | 0.745486573 | 0.377766930 | 0.056293090 |
| 84 | 0.212790891 | 0.811539266 | 0.106234433 | 1.084248389 | 0.508633693 | 0.070854619 | 1.256976286 | 0.741361414 | 0.374409750 | 0.053162424 |
| 85 | 0.200088557 | 0.802881242 | 0.099962187 | 1.076769607 | 0.503894520 | 0.066679808 | 1.250108836 | 0.737248845 | 0.371056802 | 0.050032307 |
| 86 | 0.187419674 | 0.794283099 | 0.093694063 | 1.069350618 | 0.499166670 | 0.062506220 | 1.243299872 | 0.733148722 | 0.367708035 | 0.046902705 |
| 87 | 0.174792078 | 0.785743389 | 0.087429813 | 1.061990071 | 0.494449985 | 0.058333782 | 1.236548096 | 0.729060900 | 0.364363403 | 0.043773579 |
| 88 | 0.162173639 | 0.777260710 | 0.081169189 | 1.054686660 | 0.489744308 | 0.054162420 | 1.229852252 | 0.724985239 | 0.361022857 | 0.040644898 |
| 89 | 0.149592264 | 0.768833704 | 0.074911944 | 1.047439122 | 0.485049484 | 0.049992061 | 1.223211126 | 0.720921598 | 0.357686352 | 0.037516652 |
| 90 | 0.137035887 | 0.760461054 | 0.068657833 | 1.040246233 | 0.480365361 | 0.045822632 | 1.216623543 | 0.716869841 | 0.354353840 | 0.034388854 |
| 91 | 0.124502476 | 0.752141486 | 0.062406611 | 1.033106808 | 0.475691788 | 0.041654072 | 1.210088365 | 0.712829832 | 0.351025274 | 0.031261501 |
| 92 | 0.111990022 | 0.743873764 | 0.056158033 | 1.026019701 | 0.471028616 | 0.037486333 | 1.203604490 | 0.708801438 | 0.347700610 | 0.028134491 |
| 93 | 0.099496542 | 0.735656686 | 0.049911862 | 1.018983799 | 0.466375699 | 0.033319335 | 1.197170850 | 0.704784527 | 0.344379800 | 0.025007542 |
| 94 | 0.087020075 | 0.727489091 | 0.043667860 | 1.011998025 | 0.461732890 | 0.029152890 | 1.190786410 | 0.700778969 | 0.341062800 | 0.021880376 |
| 95 | 0.074558678 | 0.719369848 | 0.037425759 | 1.005061333 | 0.457100047 | 0.024986801 | 1.184450168 | 0.696784638 | 0.337749565 | 0.018753611 |
| 96 | 0.062110428 | 0.711297859 | 0.031185289 | 0.998172709 | 0.452477029 | 0.020821487 | 1.178161151 | 0.692801405 | 0.334440050 | 0.015629081 |
| 97 | 0.049672415 | 0.703272060 | 0.024946419 | 0.991331169 | 0.447863696 | 0.016657520 | 1.171918415 | 0.688829148 | 0.331134210 | 0.012504911 |
| 98 | 0.037245745 | 0.695291414 | 0.018708530 | 0.984535759 | 0.443259910 | 0.012491511 | 1.165721043 | 0.684867744 | 0.327832002 | 0.009373127 |
| 99 | 0.024825534 | 0.687354914 | 0.012471756 | 0.977785551 | 0.438665534 | 0.008328245 | 1.159568147 | 0.680917072 | 0.324533381 | 0.006252911 |
| 100 | 0.012410908 | 0.679461582 | 0.006235631 | 0.971079644 | 0.434080434 | 0.004163860 | 1.153458863 | 0.676977012 | 0.321238305 | 0.003125163 |

with the appropriate algebraic signs attached in the Variant II case. We shall present below a scheme that maps the $M = n!/\prod_{i=1}^{K} n_i!$ such permutations $y$ into $M$ points $\pi(y)$ spaced uniformly in the unit interval, and then represents each $\pi(y)$ by the first $Q$ digits in its binary fraction expansion, where $Q$ is the smallest integer greater than $\log_2 M$. The procedure for generating the binary codeword

$$\Phi(y) = s_1 s_2 \cdots s_Q \tag{52}$$

corresponding to $y$ is most easily described by the following Algol-type program. For Variant II codes, $n$ more binary digits are appended to $\Phi(y)$ to supply the sequence of algebraic signs.

*Encoding Algorithm*

```
 1) $\pi \leftarrow \frac{n_1!\,n_2!\cdots n_K!}{n!}$.
    $P \leftarrow \frac{1}{n}$.
    $I(i) \leftarrow n_i, \quad i = 1,2,\cdots,K$.
    $I(0) \leftarrow 0$.
    $l \leftarrow 0$.
 2) $l \leftarrow l + 1$.
 3) $\pi \leftarrow \pi + P \sum_{i=0}^{i_l - 1} I(i)$.
 4) If $l = n - 1$, go to 8). Otherwise continue.
 5) $P \leftarrow P \frac{I(i_l)}{n - l}$.
 6) $I(i_l) \leftarrow I(i_l) - 1$.
 7) Go to 2).
 8) $j \leftarrow 0$.
 9) $j \leftarrow j + 1$.
10) If $\pi < 2^{-j}$, $s_j \leftarrow 0$. Otherwise ($s_j \leftarrow 1$ and $\pi \leftarrow \pi - 2^{-j}$).
11) If $j < Q$, go to [9]. Otherwise stop.
```

The corresponding Algol-type program for recovering the $i_l$, and hence $y$, from $\Phi(y)$ is as follows.

*Decoding Algorithm*

```
 1) $P \leftarrow n \sum_{j=1}^{Q} s_j 2^{-j}$.
    $I(i) = n_i, \quad i = 1,2,\cdots,K$.
    $l \leftarrow 0$.
 2) $l \leftarrow l + 1$.
 3) $R \leftarrow 0$.
    $i \leftarrow 0$.
 4) $i \leftarrow i + 1$.
 5) $R \leftarrow R + I(i)$.
 6) If $R < P$, go to 4). Otherwise continue.
 7) $i_l \leftarrow i$.
 8) If $l < n - 1$, continue. Otherwise go to 12).
 9) $P \leftarrow (P - R + I(i_l))(n - l)/I(i_l)$.
10) $I(i_l) \leftarrow I(i_l) - 1$.
11) Go to 2).
12) $I(i_l) \leftarrow I(i_l) - 1$.
13) $i \leftarrow 0$.
14) $i \leftarrow i + 1$.
15) If $I(i) = 0$, go to 14). Otherwise continue.
16) $i_n \leftarrow i$.
17) Stop.
```

It can be shown that the encoding and the decoding algorithms described above require neither memory nor computational time to grow more than linearly with $n$ [15].

## REFERENCES

[1] D. Slepian, "Permutation modulation," *Proc. IEEE*, vol. 53, Mar. 1965, pp. 228–236.
[2] J. G. Dunn, "Coding for continuous sources and channels," Ph.D. dissertation, Dep. Elec. Eng., Columbia Univ., New York, N.Y., 1965.
[3] ——, "The performance of a class of $n$-dimensional quantizers for a Gaussian source," in *Proc. Symp. Signal Transmission and Processing*, Columbia Univ., New York, N.Y., May 13–14, 1965, pp. 76–81.
[4] J. Klotz, "Small sample power and efficiency for the one sample Wilcoxon and normal scores tests," *Ann. Math. Statist.*, vol. 34, 1963, pp. 624–632.
[5] W. Feller, *An Introduction to Probability Theory and Its Applications*, vol. 2. New York: Wiley, 1966, p. 225.
[6] C. E. Shannon, "Coding theorems for a discrete source with a fidelity criterion," *IRE Nat. Conv. Rec.*, pt. 4, Mar. 1959, pp. 142–163.
[7] T. J. Goblick, Jr., and J. L. Holsinger, "Analog source digitization: A comparison of theory and practice," *IEEE Trans. Inform. Theory* (Corresp.), vol. IT-13, Apr. 1967, pp. 323–326.
[8] E. J. Gumbel, *Statistics of Extremes.* New York: Columbia Univ. Press, 1958.
[9] F. N. David *et al.*, *Normal Centroids, Medians and Scores for Ordinal Data.* Cambridge: Cambridge Univ. Press, 1968.
[10] J. Hájek and Z. Sidak, *Theory of Rank Tests.* New York: Academic Press, 1967.
[11] C. A. R. Hoare, "Quicksort," *Comput. J.*, vol. 5, 1962, pp. 10–15.
[12] W. D. Frazer and A. C. McKellar, "Samplesort: A sampling approach to minimal storage tree sorting," *J. Ass. Comput. Mach.*, vol. 17, July 1970.
[13] R. W. Floyd, "Treesort 3," *Commun. Ass. Comput. Mach.*, vol. 7, 1964, p. 701.
[14] D. H. Lehmer, "Teaching combinatorial tricks to a computer," in *Combinatorial Analysis*, R. Bellman and M. Hall, Jr., Eds. Providence, R.I.: Amer. Math. Soc., 1960.
[15] F. Jelinek, *Probabilistic Information Theory.* New York: McGraw-Hill, 1968, pp. 479–489.
[16] D. Slepian, U.S. Patent 3 396 351.
[17] H. Gish and J. N. Pierce, "Asymptotically efficient quantizing," *IEEE Trans. Inform. Theory*, vol. IT-14, Sept. 1968, pp. 676–683.
[18] F. Jelinek, "Buffer overflow in variable length coding of fixed rate sources," *IEEE Trans. Inform. Theory*, vol. IT-14, May 1968, pp. 490–501.
[19] K. Schneider and F. Jelinek, "Variable length-to-block coding of fixed rate sources for transmission through fixed rate noiseless channels," submitted to *IEEE Trans. Inform. Theory.*
[20] S. Lloyd, "Least square quantization in PCM," Bell Telephone Lab., Internal Memo., 1959.
[21] J. Max, "Quantizing for minimum distortion," *IEEE Trans. Inform. Theory*, vol. IT-6, Mar. 1960, pp. 7–12.
[22] K. Pearson, "Note on Francis Galton's problem," *Biometrika*, vol. 1, 1902, pp. 390–399.
[23] H. L. Harter, "Expected values of normal order statistics," *Biometrika*, vol. 48, 1961, pp. 151–165.