TABLE II
SET OF VALUES OF r FOR $J = 7$, CHOSEN TO PRODUCE CYCLICALLY DISTINCT SEQUENCES

w	r Values
1	1
2	3, 5, 9
3	7, 11, 13, 19, 21
4	15, 23, 27, 29, 43
5	31, 47, 55
6	63

TABLE III
DESIGN PARAMETER TRADE-OFFS FOR GMW SEQUENCES WITH A CONSTRAINT $J = 7$

		$M = 14$		$M = 28$	
w	$\binom{J}{w}/J$	$N_{GMW}(w)$	L	$N_{GMW}(w)$	L
1	1	756	14	4741632	28
2	3	2268	28	14224896	112
3	5	3780	56	23708160	448
4	5	3780	112	23708160	1792
5	3	2268	224	14224896	7168
6	1	756	448	4741632	28672

Theorem 2 indicates that the GMW sequences counted in (59) do not all have the same linear span. The following example illustrates the breakdown of the N_{GMW} sequences into collections with the same linear span and tabulates balance properties as well.

Example: Consider a design in which the ROM size constrains J to be 7. Since $2^J - 1$ is prime, all binary 7-tuples except 0000000 and 1111111, are base-2 representations of numbers relatively prime to $2^7 - 1$. One set of acceptable choices for r with cyclically inequivalent base-2 representations is shown in Table II, which contains a total of $N_p(7) = 18$ entries, with $\binom{J}{W}/J$ for each value of w.

In this design M must be a multiple of 7. Table III displays the number $N_{GMW}(w)$ of cyclically distinct sequences which can be constructed with an M stage generator, along with other data, for two choices of M, namely, 14 and 28. The GMW sequences in the $M = 14$ design all have periodic correlation peak-to-sidelobe ratio of 16 383 : 1, and are 2-tuple balanced for $w \geq 2$, and 14-tuple balanced for $w = 1$ (the m-sequence subset). Similarly the GMW sequences in the $M = 28$ design all have periodic correlation peak-to-sidelobe ratio of 268 435 455 : 1, and are 4-tuple balanced for $w \geq 2$, and 28-tuple balanced for $w = 1$.

REFERENCES

[1] J. Singer, "A theorem in finite projective geometry and some applications to number theory," *Trans. Amer. Math. Soc.*, vol. 43, pp. 377–385, May 1938.
[2] R. J. Turyn, "On Singer's parametrization and related matters," Appl. Res. Lab., Sylvania Electron. Syst., Waltham, MA, Eng. Note 197, Nov. 10, 1960.
[3] L. D. Baumert, *Cyclic Difference Sets* (Lecture Notes in Math., vol. 182). New York: Springer Verlag, 1971.
[4] B. Gordon, W. H. Mills, and L. R. Welch, "Some new differences sets," *Canadian J. Math.*, vol. 14, pp. 614–625, 1962.
[5] S. W. Golomb, *Shift Register Sequences.* San Francisco: Holden-Day, 1967.
[6] E. L. Key, "An analysis of the structure and complexity of non-linear binary sequence generators," *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 732–736, 1976.

Permutation Codes for the Laplacian Source

STEPHEN A. TOWNES, MEMBER, IEEE, AND J. B. O'NEAL, JR., SENIOR MEMBER, IEEE

Abstract—**Permutation codes for the Laplacian source are developed. The performance of these codes is evaluated and compared with other quantizers and the rate-distortion function. It is shown that there is a bit-rate region in which the permutation codes outperform certain single-sample quantizers.**

I. INTRODUCTION

Permutation coding is a block coding/quantization scheme based upon an ordering relationship between n output samples of a source. The concept was originally introduced by Slepian [1] as a form of channel coding called permutation modulation. The first application to source coding subject to a fidelity criterion was by Dunn [2] in a study of n-dimensional quantizers for Gaussian sources. Berger, Jelinek, and Wolf [3] presented a detailed analysis of the theory of permutation codes as a form of source coding with their application to Gaussian sources as an example. The work that followed included a comparison of permutation codes and optimum quantizers [4], a quasi-permutation coding scheme for Gaussian sources [5], and a comparison of permutation coding and variable bit-rate encoding [6].

The work herein expands upon these earlier treatments by developing permutation codes for a Laplacian source and evaluating their performance theoretically and by simulation. The Laplacian source was chosen because the signals to be encoded and transmitted in speech and television differential pulse-code modulation (PCM) systems have a first-order probability density that is approximately Laplacian. It will be shown that for all four of the block lengths tested there is a bit-rate region in which the permutation codes outperform previously derived single-sample quantizers. It should be noted that since this work was completed, Berger [7] has derived a new family of single-sample quantizers and shown their equivalence to the permutation codes.

II. THEORY

Consider a discrete time source whose output at the ith instant in time is random variable X_i with the continuous probability density function (pdf) $p(x_i)$. (The development for a discrete pdf would be similar.) These source outputs do not have to be identically distributed or independent. The n-vector $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ of outputs from the source is to be encoded by the codeword $\boldsymbol{c} = (c_1, c_2, \cdots, c_n)$ from the set of M n-vectors C, i.e., the infinite set of n-vectors produced by the source is to be mapped into the finite set C. When the vector $\boldsymbol{x}$ is emitted by the source, the codeword $\boldsymbol{c} \in C$, which minimizes some distortion measure $d(\boldsymbol{x}, \boldsymbol{c})$, is chosen to represent $\boldsymbol{x}$. The per-letter (or per-output) average distortion of code C is then

$$D = n^{-1} E\left[\min_{c \in C} d(\boldsymbol{x}, \boldsymbol{c})\right], \tag{1}$$

where the expected value (E) is taken with respect to the distribution of $\boldsymbol{x}$.

The restriction that we will make in designing permutation codes is that each element of C must be a permutation of the other elements. The first codeword in a Variant I permutation code (for a description of Variant II codes see [3]) is an n-vector

Manuscript received January 18, 1983; revised October 24, 1983.
S. A. Townes was with the Department of Electrical Engineering, North Carolina State University, Raleigh, NC 27650; he is now with the Communications Research Section, Jet Propulsion Laboratory, 4800 Oak Grove Drive, Pasadena, CA 91109.
J. B. O'Neal, Jr., is with the Department of Electrical Engineering, North Carolina University, Raleigh, NC 27650.

0018-9448/84/0500-0553$01.00 ©1984 IEEE

of the form

$$c_1 = (\overbrace{u_1, u_1, \cdots, u_1}^{\leftarrow n_1 \rightarrow}, \overbrace{u_2, u_2, \cdots, u_2}^{\leftarrow n_2 \rightarrow}, u_3, \cdots, \overbrace{u_k, \cdots, u_k}^{\leftarrow n_k \rightarrow}), \quad (2)$$

where the u_i and k are real numbers satisfying $u_1 < u_2 < \cdots < u_k$ and the n_i are positive integers such that $n_1 + n_2 + \cdots + n_k = n$. The remaining $M - 1$ codewords in C are all of the distinct words that can be obtained by permuting the components of c_1. There are, therefore, a total of

$$M = \frac{n!}{n_1! n_2! \cdots n_k!} \quad (3)$$

codewords. The rate of this code is then (in bits/sample)

$$R = n^{-1} \log_2 M. \quad (4)$$

The optimum encoding procedure is described in the following theorem from [3].

Theorem: Consider a block distortion measure of the form

$$d(x, c) = g\left(\sum_{t=1}^{n} f(|x_t - c_t|) \right),$$

where $x = (x_1, x_2, \cdots, x_n)$, $c = (c_1, c_2, \cdots, c_n)$, $g(\cdot)$ is nondecreasing, and $f(\cdot)$ is nonnegative, nondecreasing, and convex for positive arguments. Then the optimum encoding of Variant I permutation codes with respect to $d(x, c)$ is accomplished by the algorithm described below.

1) Replace the n_1 smallest components of x by u_1.
2) Replace the next n_2 smallest components of x by u_2.

•
•
•
•

k) Replace the n_k largest components of x by u_k.

Use the permutation of c_1 that results from these replacements to represent x.

The proof of this theorem is also found in [3]. The main effort in permutation coding, then, once the u_1 and n_i are known, is the ordering of x to determine the proper element of C to represent x.

As an illustration, consider the permutation code based on the codeword

$$c_1 = (-1, -1, 0, 0, 0, 1, 1).$$

Here we have $n = 7$, $k = 3$, $n_1 = n_3 = 2$, $n_2 = 3$, $u_1 = -u_3 = -1$, and $u_2 = 0$. There are 210 different codewords and the code rate is 1.1-bits/sample, i.e., each source output can be represented by 1.10 binary digits. Suppose the source emits the vector $x = (2.1, -1.2, 0.8, 0.2, 0.0, 0.2, 0.9)$. The codeword chosen to represent x is then $c_j = (1, -1, 0, 0, -1, 0, 1)$ since the two smallest values of x were mapped into -1's, the three middle values were mapped into 0's and the two largest values were mapped into 1's.

The per-output squared-error distortion measure

$$d(x, c,) = n^{-1} \sum_{t=1}^{n} (x_t - c_t)^2, \quad (5)$$

used so frequently in system performance analysis, is a special case of the class of distortion measures of the theorem. The performance of a Variant I permutation code under this distortion measure can be determined as follows [3]. For the output random vector $X = (X_1, X_2, \cdots, X_n)$, define the random variable Y_j, $j = 1, 2, \cdots, n$ as the jth smallest component of X (i.e., Y_1 is smallest and Y_n is the largest). Also define $s_i = n_1 + n_2 + \cdots + n_i$ and $s_0 = 0$. Then, according to [7, eq. (10a)], the average per-letter distortion will be

$$D = n^{-1} E\left[\sum_{i=1}^{k} \sum_{j=s_{i-1}+1}^{s_i} (Y_j - u_i)^2 \right]. \quad (6)$$

It can then be shown [3] that for fixed k and n_i, the optimum u_i are

$$u_i = n_i^{-1} \sum_{j=s_{i-1}+1}^{s_i} E(Y_j), \quad (7)$$

and the resulting per-letter distortion is

$$D = n^{-1} \left[E\left(\sum_{j=1}^{n} X_j^2 \right) - \sum_{i=1}^{k} n_i u_i^2 \right]. \quad (8)$$

An iterative algorithm for finding the optimum k and n_i that minimize D of (8) subject to a specified rate and block length n is given in [3].

III. Application to the Laplacian Source

A. Introduction

The permutation coding of the Laplacian source is of interest because it has been shown [8] that if one ignores the sample-to-sample correlation in a typical speech waveform and plots the (long-term) distribution of the samples, the Laplacian probability density function provides a reasonable fit to this distribution. The same can also be said of samples of the error signal in a differential system [8]. Other densities may provide a better fit, but the simplicity of the Laplacian pdf makes it easy to work with yet still provide good results.

The Laplacian source generates a sequence of outputs, each of which is a random variable with the pdf

$$p(x) = \frac{1}{2} e^{-|x|}, \qquad -\infty < x < \infty.$$

B. Theoretical Performance

The theoretical performance of a permutation code for a block of n samples, number of "quantization levels" k, and numbers of samples $n_1, n_2, \cdots, n_k$ assigned to each level can be determined from (4) and (8). The goal, then, is to find the set $\{k, n_1, \cdots, n_k\}$ such that, for a specified rate of the code, the distortion is minimized. It should be noted that, because of the finiteness of the number of possible codes of a specified block length, it may not be possible to obtain a code of exactly the rate needed, but for a sufficiently long block length one can get very close. One can find the best code for a specific rate by an exhaustive search, but as the block length grows, the practicality of this decreases. The method used for generating the codes presented here was to first use the algorithm presented in [3] to obtain first cut parameter values, and then manually adjust the parameters until very little change in distortion was noted at the specified rate. These codes, then, are good codes, but not necessarily the best. The expected values of the order statistics for the Laplacian pdf were generated as detailed in Appendix.

The theoretical performance of permutation codes for block lengths of 10, 20, 100, and 400 samples are shown in Fig. 1 for bit rates up to 3 bits/sample. The rate R versus normalized distortion (D/σ^2) curves for the block lengths of 100 and 400 samples were formed by computing the performance at 0.5, 1.0, 1.5, 2.0, 2.5, and 3.0 bits/sample and then drawing a smooth curve between these points. For block lengths of 10 and 20 samples, the performance was plotted for all codes for which $n_1 = n_k \leq n_2 = n_{k-1} \leq \cdots$, and then a lower bound was drawn through these points as shown in Fig. 2 for $n = 10$. The parameters for codes for $n = 100$ and 400 are presented in Tables I and II. The performance of these codes has been verified by simulation.

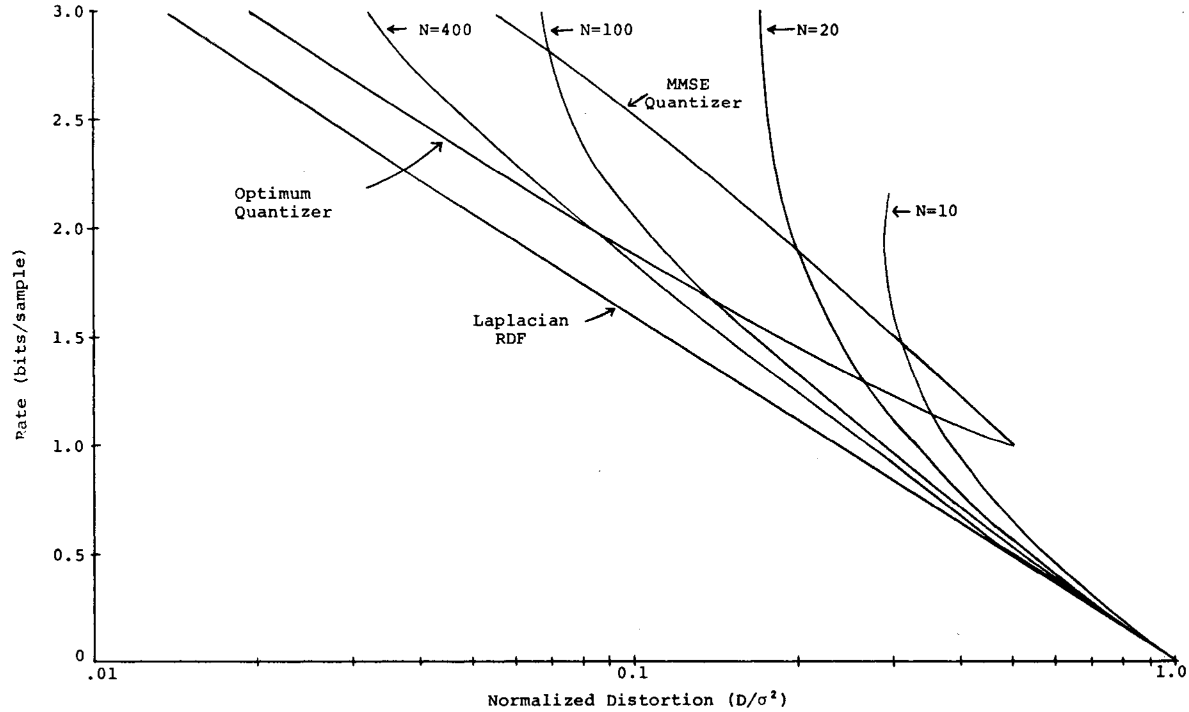

Fig. 1. Performance of (Variant I) permutation codes for a Laplacian source and blocks of 10, 20, 100, and 400 samples.

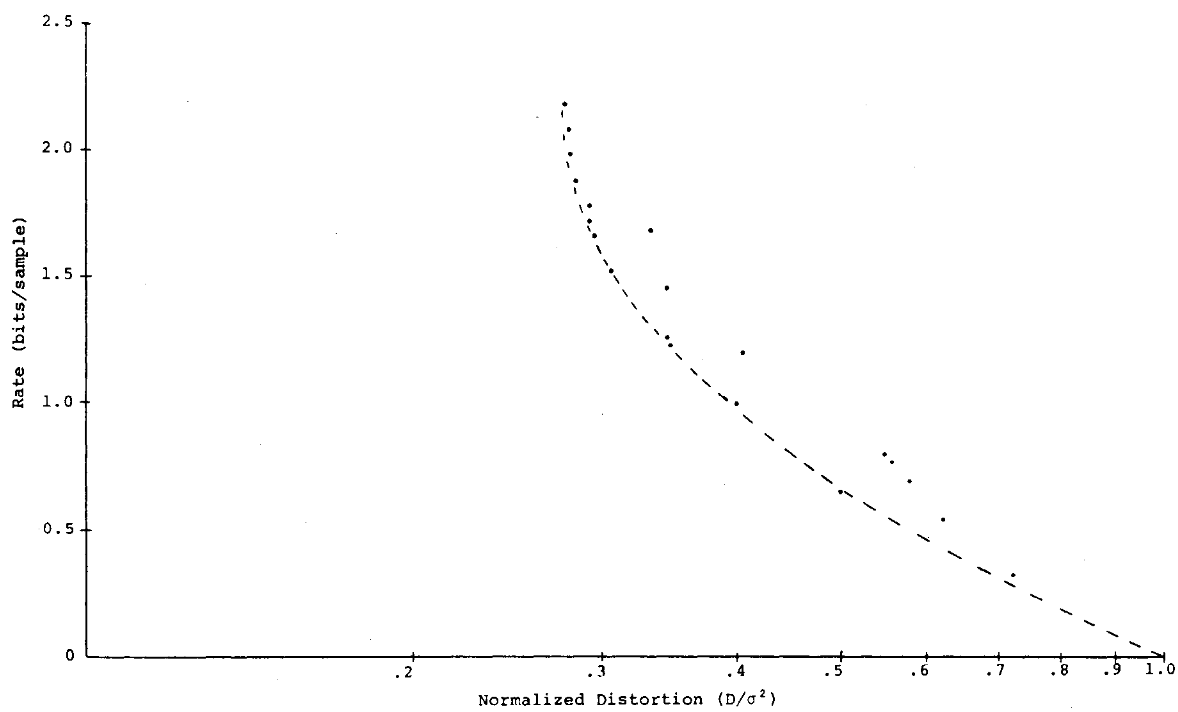

Fig. 2. Performance of permutation codes of block length 10 samples and $n_1 = n_k \leq n_2 = n_{k-1} \leq \cdots$.

The curves show that, as might be expected, the code with the longer block length performs better than one with a shorter block length. At very low bit rates (< 0.5 bits/sample) the codes are fairly close together, but as the bit rate increases, the difference becomes much more noticeable. As the block length increases, however, the encoding delay also increases, and for $n = 400$ and an 8-KHz sampling rate (traditionally used for speech), one would have to wait 50 ms at least. For $n = 100$, the delay is 12.5 ms. In speech coding it is known that any delay exacerbates the effect of echoes. Even if there are no echoes, a delay of 250 ms or more becomes objectionable. Because 250 ms corresponds to a delay of 2000 samples at 8 KHz, it would be possible to increase the block length substantially over the 400 samples used here and reap the benefits of improved performance for speech signals if echoes were not a factor.

C. Comparisons with Quantizer Performance

Fig. 1 also shows the rate-distortion function (RDF) [9], the performance of the optimum single sample quantizer [9], and the

TABLE I
(VARIANT I) PERMUTATION CODE PARAMETERS FOR BIT RATES FROM 0.5 TO 3.0 BITS / SAMPLE AND A BLOCK LENGTH OF 100 SAMPLES

R	D/σ^2	k	i	n_i	u_i
0.5073640	0.5160202	7	1	1	-4.4942303
			2	1	-3.4942303
			3	2	-2.8275637
			4	92	0.0
1.0333972	0.2800800	7	1	1	-4.4942303
			2	2	-3.2442303
			3	7	-2.0957043
			4	80	0.0
1.5024658	0.1649978	7	1	1	-4.4942303
			2	2	-3.2442303
			3	16	-1.6250395
			4	62	0.0
1.9951378	0.1062397	11	1	1	-4.4942303
			2	1	-3.4942303
			3	2	-2.8275637
			4	4	-2.8275637
			5	16	-1.1892216
			6	52	0.0
2.4960481	0.0754522	13	1	1	-4.4942303
			2	1	-3.4942303
			3	1	-2.9942303
			4	4	-2.3317303
			5	9	-1.5007121
			6	15	-0.777368
			7	38	0.0
2.9924334	0.0677083	12	1	1	-4.4942303
			2	3	-3.0497859
			3	8	-1.8810810
			4	10	-1.0982940
			5	12	-.5924596
			6	16	-.1845875

performance of the minimum mean-squared error (MMSE) single sample quantizer for the Laplacian source [9]. The efficacy of the permutation codes can be judged by comparing them with these curves.

Table III presents the comparison between the RDF and $n = 100$ and 400 permutation codes in terms of the signal-to-noise ratio (SNR $= \sigma^2/D$). Note that for $n = 400$, the performance is within 1 dB of the RDF for bit rates of less than 1.5 bits/sample, and for $n = 100$ the performance is within 1 dB of the RDF for bit rates less than 1.0 bits/sample. As the bit-rate increases, though, the code performance diverges from the RDF. Observation of the tendency of the permutation-code performance curves to get closer to the RDF with no apparent bound as n increases, leads to the conjecture that for a large enough n, it may be possible to get arbitrarily close to the RDF. No proof of this statement exists at the present.

The optimum single sample quantizer is one that minimizes the mean-squared error subject to the constraint of a fixed-output entropy rate. This quanizer requires a complex buffering system for the entropy coding for which the difficulties and modifications have been discussed in [6], [10], [11]. The modifications introduced to avoid buffer underflow and overflow tend to make the performance slightly less than optimum. The bit rates below which the permutation codes of $n = 10$, 20, 100, and 400 outperform the optimum quantizer are 1.15, 1.30, 1.70, and 2.00 bits/sample, respectively. In fact, at 1 bit/sample these codes are 1.14, 2.20, 2.52, and 2.70 dB better, respectively. The trade-off in performance between these two schemes must be considered in light of the complexity of the ordering of the source outputs for the permutation codes. The permutation codes make synchronization easier since they are block codes.

We can also compare the MMSE quantizer and the permutation codes. For $n = 10$, 20, and 100, the bit rates below which the permutation codes outperform the MMSE quantizer are 1.45, 1.80, and 2.80 bits/sample. At 3 bits/sample, the $n = 400$ permutation code is still about 1.87 dB better than the MMSE quantizer. Notice that the permutation codes for $n = 10$, 20, 100, and 400 show the same improvement over the MMSE quantizer at 1 bit/sample as they did over the optimum quantizer. The MMSE quantizer is very simple to implement and does not add any delay to the system, but the permutation codes can provide significant improvement at the expense of added delay and complexity.

IV. CONCLUSION

Permutation codes for the Laplacian source have been derived and analyzed for low bit rates. These codes were compared with MMSE and entropy-coded quantizers and found to perform at least as well over certain ranges of bit rate.

APPENDIX
GENERATION OF EXPECTED VALUES OF ORDER STATISTICS FROM A LAPLACIAN DISTRIBUTION

A. Introduction

Govindarajulu [12] has shown that the expected value of the random variable $Y_{i:n}$, the ith-order statistic from n independent samples from a symmetric pdf $p(x)$, can be related to the

TABLE II
(VARIANT I) PERMUTATION CODE PARAMETERS FOR BIT RATES FROM 0.5 TO 3.0 BITS / SAMPLE AND A BLOCK LENGTH OF 400 SAMPLES

R	D/σ²	k	i	n_i	u_i
.5095529	.4853925	3	1	18	-3.3816744
			2	1	0.0
1.0040699	.2686778	7	1	1	-5.8767825
			2	6	-4.0184492
			3	31	-2.2796768
			4	324	0.0
1.5078656	.1468080	7	1	1	-5.8767825
			2	11	-3.5823709
			3	59	-1.6752082
			4	258	0.0
2.0039856	.0822450	11	1	1	-5.8767825
			2	2	-4.6267825
			3	6	-3.5499967
			4	16	-2.5056425
			5	74	-1.2394660
			6	202	0.0
2.5050459	.0489056	13	1	1	-5.8767825
			2	3	-4.4323381
			3	7	-3.3217372
			4	12	-2.4876113
			5	28	-1.7135408
			6	68	-.8849562
			7	162	0.0
3.0021015	.0321717	17	1	1	-5.8767825
			2	2	-4.6267825
			3	3	-3.8101158
			4	7	-3.0708198
			5	10	-2.4220859
			6	20	-1.8187271
			7	32	-1.2345409
			8	74	-.5998784
			9	102	0.0

TABLE III
SNR COMPARISON OF RDF AND PERMUTATION CODES OF BLOCK LENGTHS OF 100 AND 400 SAMPLES

Rate bits/sample	SNR (dB)		
	RDF	n=400	n=100
0.5	3.15	3.14	2.87
1.0	6.62	5.71	5.53
1.5	9.43	8.33	7.70
2.0	12.66	10.85	9.74
2.5	15.69	13.11	11.22
3.0	18.68	14.93	11.69

expected values of the random variable $V_{j:k}$, the jth-order statistic from k independent samples from the related one-sided pdf $p(z)$, by the expansion

$$E(Y_{i:n}) = 2^{-n}\left[\sum_{k=0}^{i-1}\binom{n}{k}E(V_{i-k:n-k}) - \sum_{k=i}^{n}\binom{n}{k}E(V_{k-i+1:k})\right]. \tag{A1}$$

In the case of the zero-mean Laplacian pdf with a variance of 2 and the exponential pdf with mean and variance of 1, it is known [13, ex. 3.1.1] that

$$E(V_{i:n}) = \sum_{j=n-i+1}^{n} j^{-1}, \tag{A2}$$

and thus

$$E(Y_{i:n}) = 2^{-n}\left[\sum_{k=0}^{i-1}\binom{n}{k}\left(\sum_{j=n-i+1}^{n-k} j^{-1}\right) - \sum_{k=i}^{n}\binom{n}{k}\left(\sum_{j=i}^{k} j^{-1}\right)\right]. \tag{A3}$$

This can be simplified by realizing that

$$\binom{n}{k} = \binom{n}{n-k},\ k \le \operatorname{int}\left[\frac{n}{2}\right], \tag{A4}$$

where n and k are integers as usual. Eq. (A3) then becomes

$$E(Y_{i:n}) = -\left(\sum_{j=i}^{n-i} j^{-1}\right)\left(\sum_{k=0}^{i-1}\binom{n}{k}2^{-n}\right) - \sum_{k=i}^{n-i}\binom{n}{k}2^{-n}\left(\sum_{j=i}^{k} j^{-1}\right). \tag{A5}$$

This is the form used in the following computations.

B. Computation

The computation required to find $E(Y_{i:n})$ can be substantially reduced by the use of the symmetry of the pdf of X and the symmetry of the binomial coefficients as stated in (A4). From

TABLE IV
EXPECTED VALUES OF ORDER STATISTICS FROM THE LAPLACIAN DISTRIBUTION $n = 100$

I	E[Yi:n]	E[Yi+1:n]	E[Yi+2:n]	E[Yi+3:n]	E[Yi+4:n]
1	-4.494230	-3.494230	-2.994230	-2.660897	-2.410897
6	-2.210897	-2.044230	-1.901373	-1.776373	-1.665262
11	-1.565262	-1.474353	-1.391020	-1.314097	-1.242668
16	-1.176001	-1.113501	-1.054678	-0.999122	-0.946491
21	-0.896491	-0.848872	-0.803417	-0.759939	-0.718272
26	-0.678272	-0.639811	-0.602774	-0.567059	-0.532577
31	-0.499243	-0.466985	-0.435735	-0.405432	-0.376019
36	-0.347446	-0.319665	-0.292633	-0.266306	-0.240645
41	-0.215611	-0.191164	-0.167261	-0.143858	-0.120906
46	-0.098350	-0.076131	-0.054181	-0.032428	-0.010796

TABLE V
EXPECTED VALUES OF ORDER STATISTICS FROM THE LAPLACIAN DISTRIBUTION $n = 200$

I	E[Yi:n]	E[Yi+1:n]	E[Yi+2:n]	E[Yi+3:n]	E[Yi+4:n]
1	-5.184884	-4.184884	-3.684884	-3.351550	-3.101550
6	-2.901550	-2.734884	-2.592027	-2.467027	-2.355916
11	-2.255916	-2.165006	-2.081673	-2.004750	-1.933321
16	-1.866655	-1.804155	-1.745331	-1.689776	-1.637144
21	-1.587144	-1.539525	-1.494071	-1.450592	-1.408926
26	-1.368926	-1.330464	-1.293427	-1.257713	-1.223230
31	-1.189897	-1.157639	-1.126389	-1.096086	-1.066674
36	-1.038102	-1.010325	-0.983298	-0.956982	-0.931341
41	-0.906341	-0.881950	-0.858141	-0.834885	-0.812158
46	-0.789936	-0.768197	-0.746920	-0.726087	-0.705678
51	-0.685678	-0.666071	-0.646840	-0.627972	-0.609453
56	-0.591272	-0.573414	-0.555871	-0.538629	-0.521680
61	-0.505013	-0.488620	-0.472491	-0.456618	-0.440993
66	-0.425608	-0.410457	-0.395531	-0.380825	-0.366333
71	-0.352047	-0.337962	-0.324074	-0.310375	-0.296861
76	-0.283528	-0.270370	-0.257383	-0.244562	-0.231903
81	-0.219402	-0.207055	-0.194857	-0.182805	-0.170895
86	-0.159121	-0.147481	-0.135968	-0.124578	-0.113306
91	-0.102145	-0.091087	-0.080126	-0.069254	-0.058460
96	-0.047734	-0.037065	-0.026442	-0.015852	-0.005282

[13, eq. 3.1.8], the symmetry of $p(x)$ gives

$$E(Y_{i:n}) = -E(Y_{n-i+1:n}). \tag{A6}$$

Thus only int$[n/2]$ of the expected values need be calculated.

The problem encountered in calculating the binomial probabilities $\binom{n}{k}2^{-n}$ is the magnitude of the numbers involved. For large n, $\binom{n}{k}$ can be very large and 2^{-n} can be very small, both out of the range of most computers. The procedure used was to calculate $P(k)$, the kth binomial probability, by setting the largest one $P(L)$ to some arbitrary large number and using the recursion relation

$$P(k) = P(k-1) \cdot (n-k+1)/k, \qquad L+1 \le k \le n. \tag{A7}$$

Symmetry was used to fill out the lower half. Since

$$\sum_{k=0}^{n} \binom{n}{k} 2^{-n} = 1, \tag{A8}$$

the scale factor needed to bring the $P(k)$ down to their true values can easily be found.

The computations of $E(Y_{i:n})$ were performed using the unscaled $P(k)$ values in place of $\binom{n}{k}2^{-n}$ in (A5) and then these numbers were scaled by the computed scale factor. This technique avoided underflow and overflow problems.

C. Results

The $E(Y_{i:n})$ for $n = 100$, 200, 300, and 400 are shown in Tables IV–VII.

TABLE VI
EXPECTED VALUES OF ORDER STATISTICS FROM THE LAPLACIAN DISTRIBUTION $n = 300$

I	E[Yi:n]	E[Yi+1:n]	E[Yi+2:n]	E[Yi+3:n]	E[Yi+4:n]
1	-5.589517	-4.589517	-4.089517	-3.756183	-3.506183
6	-3.306183	-3.139517	-2.996660	-2.871660	-2.760548
11	-2.660548	-2.569639	-2.486306	-2.409383	-2.337954
16	-2.271288	-2.208788	-2.149964	-2.094409	-2.041777
21	-1.991777	-1.944158	-1.898703	-1.855225	-1.813559
26	-1.773559	-1.735097	-1.698060	-1.662346	-1.627863
31	-1.594530	-1.562272	-1.531022	-1.500718	-1.471307
36	-1.442735	-1.414958	-1.387930	-1.361615	-1.335974
41	-1.310974	-1.286583	-1.262774	-1.239518	-1.216791
46	-1.194569	-1.172829	-1.151553	-1.130720	-1.110311
51	-1.090311	-1.070704	-1.051473	-1.032605	-1.014086
56	-0.995904	-0.978047	-0.960503	-0.943262	-0.926313
61	-0.909646	-0.893253	-0.877124	-0.861251	-0.845626
66	-0.830241	-0.815090	-0.800164	-0.785458	-0.770966
71	-0.756680	-0.742595	-0.728707	-0.715008	-0.701494
76	-0.688161	-0.675003	-0.662016	-0.649196	-0.636537
81	-0.624037	-0.611692	-0.599497	-0.587448	-0.575544
86	-0.563779	-0.552151	-0.540657	-0.529293	-0.518057
91	-0.506946	-0.495957	-0.485088	-0.474335	-0.463697
96	-0.453170	-0.442754	-0.432444	-0.422240	-0.412139
101	-0.402139	-0.392238	-0.382434	-0.372726	-0.363110
106	-0.353586	-0.344152	-0.334807	-0.325547	-0.316373
111	-0.307282	-0.298273	-0.289345	-0.280495	-0.271723
116	-0.263027	-0.254407	-0.245860	-0.237385	-0.228982
121	-0.220648	-0.212384	-0.204187	-0.196057	-0.187992
126	-0.179991	-0.172054	-0.164179	-0.156365	-0.148612
131	-0.140916	-0.133279	-0.125698	-0.118172	-0.110700
136	-0.103279	-0.095909	-0.088587	-0.081312	-0.074080
141	-0.066890	-0.059738	-0.052622	-0.045538	-0.038482
146	-0.031451	-0.024440	-0.017446	-0.010463	-0.003487

TABLE VII
EXPECTED VALUES OF ORDER STATISTICS FROM THE LAPLACIAN DISTRIBUTION $n = 400$

I	E[Yi:n]	E[Yi+1:n]	E[Yi+2:n]	E[Yi+3:n]	E[Yi+4:n]
1	-5.876783	-4.876783	-4.376783	-4.043449	-3.793449
6	-3.593449	-3.426783	-3.283925	-3.158925	-3.047814
11	-2.947814	-2.856905	-2.773572	-2.696649	-2.625220
16	-2.558554	-2.496054	-2.437230	-2.381674	-2.329043
21	-2.279043	-2.231424	-2.185969	-2.142491	-2.100824
26	-2.060824	-2.022363	-1.985326	-1.949611	-1.915129
31	-1.881795	-1.849537	-1.818287	-1.787984	-1.758573
36	-1.730001	-1.702223	-1.675196	-1.648880	-1.623239
41	-1.598239	-1.573849	-1.550040	-1.526784	-1.504057
46	-1.481834	-1.460095	-1.438819	-1.417985	-1.397577
51	-1.377577	-1.357969	-1.338739	-1.319871	-1.301352
56	-1.283170	-1.265313	-1.247769	-1.230528	-1.213579
61	-1.196912	-1.180519	-1.164390	-1.148517	-1.132892
66	-1.117507	-1.102355	-1.087430	-1.072724	-1.058231
71	-1.043946	-1.029861	-1.015972	-1.002274	-0.988760
76	-0.975427	-0.962269	-0.949282	-0.936461	-0.923803
81	-0.911303	-0.898958	-0.886762	-0.874714	-0.862809
86	-0.851045	-0.839417	-0.827923	-0.816559	-0.805323
91	-0.794212	-0.783223	-0.772353	-0.761601	-0.750962
96	-0.740436	-0.730019	-0.719710	-0.709506	-0.699405
101	-0.689405	-0.679504	-0.669700	-0.659991	-0.650376
106	-0.640852	-0.631418	-0.622072	-0.612813	-0.603639
111	-0.594548	-0.585539	-0.576610	-0.567761	-0.558989
116	-0.550293	-0.541673	-0.533125	-0.524651	-0.516248
121	-0.507914	-0.499650	-0.491453	-0.483323	-0.475258
126	-0.467258	-0.459322	-0.451448	-0.443635	-0.435883
131	-0.428191	-0.420558	-0.412982	-0.405463	-0.398000
136	-0.390593	-0.383240	-0.375941	-0.368694	-0.361500
141	-0.354357	-0.347265	-0.340223	-0.333230	-0.326285
146	-0.319389	-0.312539	-0.305737	-0.298980	-0.292269
151	-0.285602	-0.278979	-0.272400	-0.265865	-0.259371
156	-0.252919	-0.246509	-0.240140	-0.233811	-0.227521
161	-0.221271	-0.215060	-0.208887	-0.202752	-0.196655
166	-0.190594	-0.184570	-0.178582	-0.172629	-0.166712
171	-0.160830	-0.154981	-0.149167	-0.143386	-0.137638
176	-0.131923	-0.126240	-0.120588	-0.114968	-0.109378
181	-0.103818	-0.098288	-0.092786	-0.087312	-0.081866
186	-0.076446	-0.071051	-0.065681	-0.060334	-0.055008
191	-0.049704	-0.044418	-0.039149	-0.033897	-0.028658
196	-0.023431	-0.018213	-0.013004	-0.007800	-0.002600

Only one half of the $E(Y_{i:n})$ are presented since the other half can be found from relation (A6).

References

[1] D. Slepian, "Permutation modulation," *Proc. IEEE*, vol. 53, no. 3, pp. 228–236, Mar. 1965.
[2] J. G. Dunn, "The performance of a class of n-dimensional quantizers for a Gaussian source," presented at *Proc. Symp. on Signal Transmission and Processing*, Columbia Univ., New York, NY, May 13–14.
[3] T. Berger, F. Jelinek, and J. K. Wolf, "Permutation codes for sources," *IEEE Trans. Inform. Theory*, vol. IT-18, no. 1, pp. 160–169, Jan. 1972.
[4] T. Berger, "Optimum quantizers and permutation codes," *IEEE Trans. Inform. Theory*, vol. IT-18, no. 6, pp. 759–765, Nov. 1972.
[5] M. U. Chang, "Quantization-permutation encoding," Master's thesis, Cornell Univ., June 1975.
[6] S. Y. Tung, "Buffer instrumented variable-length encoding," Master's thesis, Cornell Univ., Ithaca, NY, June 1975.
[7] T. Berger, "Minimum entropy quantizers and permutation codes," *IEEE Trans. Inform. Theory*, vol. IT-28, no. 2, pp. 149–157, Mar. 1982.
[8] M. D. Paez, and T. H. Glisson, "Minimum mean-squared-error quantization in speech PCM and DPCM systems," *IEEE Trans. Commun.*, vol. COM-20, no. 4, pp. 225–230, Apr. 1972.
[9] P. Noll, and R. Zelinski, "Bounds on quantizer performance in the low bit-rate region," *IEEE Trans. Commun.*, vol. COM-26, no. 2, pp. 300–305, Feb. 1978.
[10] F. Jelinek and K. S. Schneider, "On variable-length-to-block coding," *IEEE Trans. Inform. Theory*, vol. IT-18, no. 6, pp. 765–774, Nov. 1972.
[11] F. Jelinek, "Buffer overflow in variable length coding of fixed rate sources," *IEEE Trans. Inform. Theory*, vol. IT-14, no. 3, pp. 490–501, May 1968.
[12] Z. Govindarajulu, "Best linear estimates under symmetric censoring of the parameters of a double exponential population," *Amer. Statist. Assoc. J.*, pp. 248–258, Mar. 1966.
[13] H. A. David, *Order Statistics*. New York: Wiley, 1979.

Source Coding Bounds Using Quantizer Reproduction Levels

WILLIAM A. PEARLMAN, MEMBER, IEEE AND ALI CHEKIMA

***Abstract*—Constraining the reproduction alphabet to be of small size in encoding continuous-amplitude memoryless sources has been shown to give very small degradation from the ideal performance of the rate-distortion bound. The optimum fixed-size reproduction alphabet and its individual letter probabilities are required in order to encode the source with performance approaching that of theory. These can be found through a somewhat lengthy, but convergent, algorithm. Given reasonably chosen fixed sets of reproduction letters and / or their probabilities, we define new rate-distortion functions which are coding bounds under these alphabet constraints. We calculate these functions for the Gaussian and Laplacian sources and the squared-error distortion measure and find that performance near the rate-distortion bound is achievable using a reproduction alphabet consisting of a small number of optimum quantizer levels.**

I. Introduction

The encoding of a discrete-time continuous-amplitude memoryless source is often accomplished by single-sample quantization followed by coding to produce a per-symbol bit rate approaching the minimum of the entropy of the quantized sequence. This so-called "entropy coding" is accomplished either by variable-length buffer-instrumented Huffman codes [1] or permutation codes [2], [3] operating on long sequences of quantized values. A quantizer is *optimum* when it gives the smallest entropy for a given average distortion, as defined by a single-letter distortion measure (usually squared-error). An optimum quantizer may in theory require a large number of quantization or reproduction levels, especially as the rate grows larger. In practice, however, the optimum performance is closely approximated with a finite number M of reproduction levels, and M grows smaller as the rate decreases or distortion increases.

Except for the trivial zero-rate case, the optimum quantizer is not an optimum encoder, because its rate-versus-distortion characteristic is above that of the theoretical minimum given by the rate-distortion function. For the Gaussian source and squared-error distortion measure, Goblick and Holsinger [4] have shown numerically that the optimum uniform quantizer attains a rate 0.25 bits above the rate-distortion bound for rates above 0.75 bits. Gish and Pierce [5] derived for general sources and distortion measures that, as the number of reproduction levels grows large, the minimal rate is achieved by a uniform quantizer. This rate is only 0.255 above the Shannon lower bound, which, in turn, coincides with the rate-distortion function in the region of low distortion for many sources. Current evidence is that the absolute width of the gap between the optimum quantizer rate-versus-distortion characteristic and the rate-distortion function successively narrows at low rates until it vanishes at zero rate. Berger [3] has recently obtained this result numerically for Gaussian and Laplacian sources with squared-error distortion. Recently, Farvardin and Modestino [6] have verified this low rate behavior for several classes of source distributions with squared-error distortion. Moreover, they find only negligible differences in performance between optimum nonuniform and optimum uniform quantizers. For a one bit quantizer of a unit variance Laplacian source, Netravali and Saigal [7] found an average distortion between 0.26 and 0.27, consistent with the aforementioned researchers. Our own calculations, using specialized forms of algorithms to be described, independently corroborate [3] and [7] for Gaussian and Laplacian sources. We remark that Noll and Zelinski [8] have previously reported much larger gaps in performance between the optimum quantizer and the rate-distortion bound for several non-Gaussian sources at one bit. These results, however, were obtained with the constraints of symmetric quantizers with an even number of levels, the minimum entropy of which is one bit.

Finamore and Pearlman [9] have recently exhibited block codes that realize coding performance better than that of the optimum quantizer and close to the rate-distortion bound and use only a small number of reproduction values. The construction of such codes and the search techniques for finding the codeword for a given source sequence are explained in detail by Pearlman [10]. In this and the previous paper [9] codes are found with performance close to that promised by the theory. The selection of letters in the codewords depends upon finding a set of optimum reproduction values and an associated set of optimal probabilities. The computation and storage requirements for implementing these codes are moderate enough to make them a viable alternative to entropy-coded quantization for a combination of rate and source density offering sufficient performance gain.

The vehicle for the theory of Finamore and Pearlman [9] is a rate-distortion function with a constrained-size reproduction alphabet (set of reproduction values). The rate values of this function for a given average distortion are theoretically obtainable by coding with a given size reproduction alphabet. These functional values provide bounds on rates for encoding with a reproduction alphabet constrained only in size. These rates cannot exceed the rates of the corresponding optimum quantizer with the same number of reproduction values. Moreover, calculations of this rate-distortion function for the Gaussian and Lapla-

Manuscript received July 16, 1982; revised July 1, 1983. This research was sponsored by the Air Force Office of Scientific Research, Air Force Systems Command, USAF, under Grant AFOSR-81-0188. The United States Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation hereon. Portions of this paper were presented at the IEEE International Conference on Communications, Philadelphia, PA, June 1982.

The authors are with Electrical, Computer, and Systems Engineering Department, Rensselaer Polytechnic Institute, Troy, NY 12181.

0018-9448/84/0500-0559$01.00 ©1984 IEEE