Department of

ELECTRICAL ENGINEERING

UNIVERSITY OF NOTRE DAME, NOTRE DAME, INDIANA

Syndrome-Source-Coding and Its Universal Generalization*

Teofilo C. Ancheta, Jr.
Department of Electrical Engineering
University of Notre Dame
Notre Dame, Indiana 46556

Technical Report No. EE-755

July 17, 1975

Abstract: A method of using error-correcting codes to obtain data compression, call d syndrome-source-coding, is described in which the source sequence is treated as an error pattern whose syndrome forms the compressed data. It is shown that syndrome-source-coding can achieve arbitrarily small distortion with the number of compressed digits per source digit arbitrarily close to the entropy of a binary memoryless source. A "universal" generalization of syndrome-source-coding is formulated which provides robustly-effective, distortionless, coding of source ensembles. Two examples are given comparing the performance of noiseless universal syndrome-source-coding to (1) run-length coding and (2) Lynch-Davisson-Schalkwijk-Cover universal coding for an ensemble of binary memoryless sources.

* This research was supported by the National Aeronautics and Space Administration under NASA Grant NSG 5025 at the University of Notre Dame in liaison with the NASA Goddard Space Flight Center. A portion of this work was presented orally at the IEEE International Symposium on Information Theory, Notre Dame, Indiana, October 27-31, 1974.

I. INTRODUCTION

The conventional method [1-5] for using block error-correcting-codes to perform source encoding (or "data compression") is shown in Fig. 1. The source output is treated as a received codeword, $\underline{r}$, and a channel decoder for the selected code serves as the "source encoder." The channel decoder finds a codeword $\underline{x}$ close (in an appropriate sense) to $\underline{r}$ and delivers as its output the information sequence $\underline{u}$ corresponding to this codeword. The sequence $\underline{u}$ is then the "compressed data." Thus, there are R compressed digits per source letter where R is the code rate i.e., the ratio of the number of information digits to the total number of digits in a codeword. At the user end, a channel encoder is used to convert $\underline{u}$ to the codeword $\underline{x}$, and thus serves as the "source decoder."

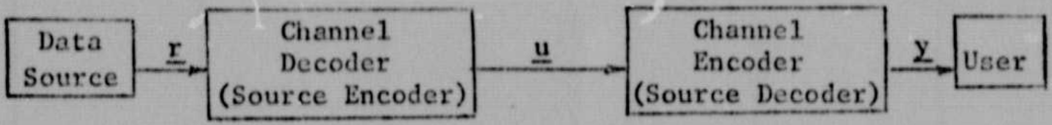
Data Source
$\underline{r}$
Channel Decoder (Source Encoder)
$\underline{u}$
Channel Encoder (Source Decoder)
$\underline{y}$
User

Fig. 1 The Conventional Method of Using Error-Correcting-Codes for Data Compression

Source encoding of the above type using linear error-correcting codes has been shown to be efficient for encoding memoryless, symmetric sources under the Hamming distortion measure in which the average distortion is the average fraction of source digits erroneously reconstructed [5]. For many "real" sources such as the output of measuring devices in space experiments, however, the information source is highly asymmetric with strong memory constraints. In such cases, the scheme of Fig. 1 generally becomes inefficient since the codewords of simply-implemented linear error-correcting-codes do not well-approximate the likely set of source output sequences. Moreover, in any case, the conventional scheme of Fig. 1 has the disadvantage for many applications that the more complex device, viz., the channel decoder, is located at the source side where one wishes to use the least equipment, while the simpler device, viz., the channel encoder, is located at the user side.

In this paper we describe an alternative method for using linear error-correcting-codes to achieve data compression. This scheme, which we shall refer to as syndrome-source-coding, is diagrammed in Fig. 2. The fundamental principle of syndrome-source-coding is that the source output is treated as a channel error pattern, $\underline{e}$, rather than as the received channel codeword. A syndrome-former, which is a simple linear device of the same complexity as an encoder for the code, serves as the "source encoder." The syndrome $\underline{s}$, i.e., the pattern of parity check failures, is then taken as the compressed data. There are 1-R compressed digits per source letter since the number of syndrome digits equals the number of redundant digits in a codeword. At the user end, an "error-pattern-estimator" for the code, i.e., a device which produces a likely (in an appropriate sense) error pattern $\hat{\underline{e}}$ consistent with $\underline{s}$, is used as the "source decoder." This device is the heart of the syndrome decoder for the code, and dominates the complexity of the decoder in most cases.

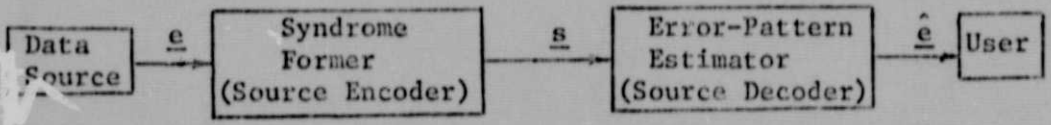
Data Source
$\underline{e}$
Syndrome Former (Source Encoder)
$\underline{s}$
Error-Pattern Estimator (Source Decoder)
$\hat{\underline{e}}$
User

Fig. 2 The Syndrome Source Coding Method of Using Error-Correcting-Codes for Data Compression

Since the error patterns which are correctable by the decoding schemes traditionally considered in channel coding studies tend to be asymmetric (e.g., for binary codes, the set of correctable error patterns usually includes all sequences with a sufficiently small number of 1's) and tend to match memory effects (e.g., the correctable errors often include all "bursts" in which all the 1's in the error pattern are confined to some small span of consecutive positions), the syndrome-source-coding method appears well-suited to the compression of many real sources. In other words, the set of error-patterns correctable by simply-implemented decoders for known block codes seem to approximate well the "likely" set of source output sequences for many real sources. Moreover, syndrome-source-coding has the advantage for many applications that the simpler device, viz., the syndrome-former, is located at the source side while the more complex device, viz., the error-pattern-estimator, is located at the user side.

For simplicity, we restrict ourselves hereafter to the binary case so that the source output digits and the code digits are in the finite field GF(2). In Section II we give a simple analysis of syndrome-source-coding and show that, for a memoryless binary source and for arbitrarily small distortion, the required number of transmitted digits per source letter can be made arbitrarily close to the source entropy. In Section III, we describe a "universal" form of syndrome-source-coding which is the major practical contribution of this note. In Section IV, we compare the performance of universal-syndrome-source-coding for memoryless binary sources to the performance of a well-known universal coding scheme and to run-length coding. Finally, in Section V, we trace the development of syndrome-source-coding and show its relation to certain other source-coding methods.

II. SOME THEORETICAL CONSIDERATIONS

With any binary source, we associate the *additive channel* in which the source output forms the error pattern, i.e., in which the received word is $\underline{r} = \underline{x} + \underline{e}$ where $\underline{x}$ is the transmitted word, where $\underline{e}$ is the source output which is assumed to be statistically independent of $\underline{x}$, and where the addition is component-by-component in GF(2). A given syndrome decoder for a given linear code to be used on this channel would comprise two devices, viz., the *syndrome-former* that computes $\underline{s} = \underline{r}\, H^T$ where H is the parity-check matrix of the code so that $\underline{s} = \underline{r}\, H^T = (\underline{x} + \underline{e})H^T = \underline{e}\, H^T$, and the *error-pattern-estimator* whose input is $\underline{s}$ and whose output is the estimate $\hat{\underline{e}}$ of $\underline{e}$. The corresponding estimate $\hat{\underline{x}}$ of $\underline{x}$ is, of course, $\hat{\underline{x}} = \underline{r} - \hat{\underline{e}}$. Suppose that P_e is the average fraction of digits in $\underline{x}$ (or, equivalently, in $\underline{e}$) which are incorrectly decoded. We have then as an immediate consequence of the syndrome-source-coding configuration shown in Figure 2:

Theorem: The average Hamming distortion for syndrome-source-coding of a given binary source coincides with the per-digit error probability P_e when the corresponding syndrome decoder is used with the given linear code on the additive channel associated with the source.

We note that the above theorem applies to all linear codes, whether block or convolutional, since nothing prevents $\underline{x}$ and $\underline{e}$ from being semi-infinite vectors.

To illustrate the use of this theorem, consider the binary memoryless source for which the probability of emitting a 1 is p. The entropy of this source is $H = - p \log_2 p - (1 - p) \log_2 (1 - p)$. The associated additive channel is just the binary symmetric channel with crossover probability p. The capacity of this channel is $C = 1 - H$. It is well-known that, for any $\varepsilon > 0$ and any $\delta > 0$, there exists a block length n and a linear block code of rate $R > C - \delta$ such that $P_e < \varepsilon$ for the syndrome decoder in which $\underline{\hat{e}}$ is the minimum weight solution of $\underline{s} = \underline{\hat{e}} H^T$ (maximum weight if $p > 1/2$), see, e.g., [6, p.206]. But, since $1 - R < 1 - C + \delta = H + \delta$ is the number of compressed digits per source letter when this code and decoder are used in syndrome-source-coding of the same source, we have:

Corollary: For a memoryless binary source with entropy H, given any $\varepsilon > 0$ and any $\delta > 0$, there is a syndrome-source-coding scheme based on a linear block code that achieves average Hamming distortion less than ε and utilizes less than $H + \delta$ compressed digits per source letter.

It should be evident that the above corollary remains true if "block" is changed to "convolutional." The above corollary furnished some corroboration of our claim that the set of correctable error patterns for linear codes well-approximates the set of typical sequences for asymmetric sources.

III. DISTORTIONLESS UNIVERSAL SYNDROME-SOURCE-CODING

Guided by the concepts of "universal noiseless coding" [7], we now introduce a generalization of syndrome-source-coding that permits the same source coding scheme to compress many different sources effectively.

Let V denote the vector space of all 2^n binary n-tuples. Let $V_1, V_2, \ldots V_M$, $M = 2^m$, be a set of linear block codes of length n (i.e. subspaces of V) with rates $R_1 \geq R_2 \geq \ldots \geq R_M$. Let $E_1, E_2, \ldots E_M$ be a partition of V such that the n-tuples in E_i all have distinct syndromes relative to the code V_i, i.e. all fall into distinct cosets of V_i. We shall ordinarily take $V_M = \{\underline{0}\}$ so that such a partition of V is certainly possible. By <u>noiseless universal syndrome-source-coding</u> (NUSSC) we mean the coding scheme for a binary source in which the source output n-tuple $\underline{e}$ is encoded as an m bit prefix identifying the index i such that $\underline{e} \in E_i$, followed by the $n(1 - R_i)$ bit syndrome $\underline{s} = \underline{e}\, H_i^T$ where H_i is a parity-check matrix for V_i. At the user side, the source is reconstructed without distortion by using the prefix to identify which code's error-pattern-estimator should be applied to the syndrome $\underline{s}$ to yield $\underline{e}$.

Normally, one would choose the partition $E_1, E_2, \ldots E_M$ in such a way that the most likely source sequences lie in the leading blocks of this partition because the average syndrome length is then minimized. However, some compromise with such a strict assignment rule may be dictated by the desire to simplify the requisite error-pattern-estimators.

A substantial simplification can be made at the source-encoder side when $V_1 \supset V_2 \supset \ldots \supset V_{M-1} \supset V_M = \{\underline{0}\}$. In this case, the syndrome for V_M is just $\underline{s} = \underline{e}$ and hence is trivially formed. Letting $k_1 \geq k_2 \geq \ldots \geq k_{M-1}$ be the code dimensions, we note that the nesting of the codes implies that the parity-check matrix H_{M-1} for code V_{M-1} can be selected so that its first $n - k_i$ rows form the parity-check matrix H_i for code V_i, $1 \leq i < M$. Thus, only the syndrome-former for code V_{M-1} need be implemented as its first $n - k_i$ digits will be the desired syndrome $\underline{s}$ when $\underline{e} \in E_i$ for $1 \leq i < M$.

It would now be easy to prove theorems on the effectiveness of NUSSC, say for the ensemble of memoryless binary sources. But we believe the key concepts as well as the practical potential of NUSSC will be made more apparent from some examples in which NUSSC is compared to some well-known data-compression schemes.

IV. EXAMPLES AND COMPARISON

As a first example of NUSSC, take $n = 15$ and $M = 4$, and let (1) $V_1 = V$ be the trivial (15,15) code such that $\underline{s}$ is the empty string, (2) V_2 be the (15,11) Hamming code, (3) V_3 be the (15,7) double-error-correcting BCH code, and (4) $V_4 = \{\underline{0}\}$ be the trivial code such that $\underline{s} = \underline{e}$. For the partition of V to be used, let (1) E_1 be $\{\underline{0}\}$, (2) E_2 be the 15 n-tuples of weight 1, (3) E_3 be the 105 n-tuples of weight 2, and (4) E_4 be the n-tuples of weight 3 through 15 inclusive. Because $V_1 \supset V_2 \supset V_3 \supset V_4 = \{\underline{0}\}$, we note that, by our previous discussion, only the syndrome-former for code V_3 need be implemented. We also note that the error-pattern-estimators for V_1 and V_4 are trivial, that for V_3 is a simple threshold decoder [8, p.95], and that for V_2 is a simple combinatorial device.

In general, for NUSSC, the average number of compressed digits per source letter, ν, is given by

$$\nu = \frac{m}{n} + (1 - R_1)\, P_1 + \ldots + (1 - R_M)\, P_M \qquad (1)$$

where P_i is the probability that the source sequence $\underline{e}$ will lie in E_i. For our example and for the memoryless binary source considered above, $P_1 = (1 - p)^{15}$, $P_2 = 15p(1 - p)^{14}$, $P_3 = 105p^2(1 - p)^{13}$ and $P_4 = 1 - P_1 - P_2 - P_3$. Using these values in (1) and defining the efficiency, η, of the source coding scheme by

$$\eta = \frac{H}{\nu}$$

(which is the ratio of the smallest average number of compressed digits per source letter that suffice for distortionless source reconstruction to the average number in the given distortionless coding scheme), we can calculate the efficiency of the NUSSC of our example. The results of this calculation are shown in Figure 3 for the ensemble of memoryless binary sources with $.01 \leq p \leq .5$. We see that this particular NUSSC scheme is quite robust, maintaining an efficiency of about 50% or more over the entire source ensemble and an efficiency of 80% or more over the interesting range $.045 \leq p \leq .5$.

For comparison to NUSSC, we also give in Figure 3 the efficiency of run-length (RL) coding and the efficiency of Lynch-Davisson-Schalkwijck-Cover (LDSC) coding.

In RL coding, the coder transmits an m-bit number giving the radix-2 form of the number of consecutive 0's (possibly none) between each 1 emitted by the source; except that, when this run-length is $2^m - 1$ or greater, the coder then transmits the radix-2 form of $2^m - 1$ followed by the code for the remaining number (possibly zero) of 0's in the run.

LDSC coding which was the earliest "universal" noiseless coding scheme and remains one of the most useful, operates as follows. The code for n source digits, $n = 2^m - 1$, is an m-bit prefix giving the radix-2 form of the Hamming weight, w, of this n-tuple followed by $\lceil \log_2 [\binom{n}{w} - 1] \rceil$ bits giving the rank of the particular weight w sequence in the lexicographical ranking of the $\binom{n}{w}$ such n-tuples. (Here, $\lceil \ \rceil$ denotes the smallest integer not less than the enclosed number.) This scheme was first described by Lynch [9] who gave a ranking algorithm. Davisson [10] noted its "universal" character and gave an inverse for the ranking algorithm. Later, Schalkwijk [10] simplified the ranking algorithm, while Cover [11] simplified the inverse algorithm and also generalized the algorithms for use with other sets of sequences.

In Figure 3, we show the efficiency of RL coding with $m = 4$ (in which runs up to length 15 have a single 4 bit code) for the binary memoryless sources with $.01 \leq p \leq .5$. We show also the efficiency of $n = 15$ LDSC coding for the same source ensemble. We note that $n = 15$ NUSSC significantly outperforms both other coding methods over most of the given range of p. The complexity of implementing the NUSSC scheme of Figure 3 would appear to be intermediate

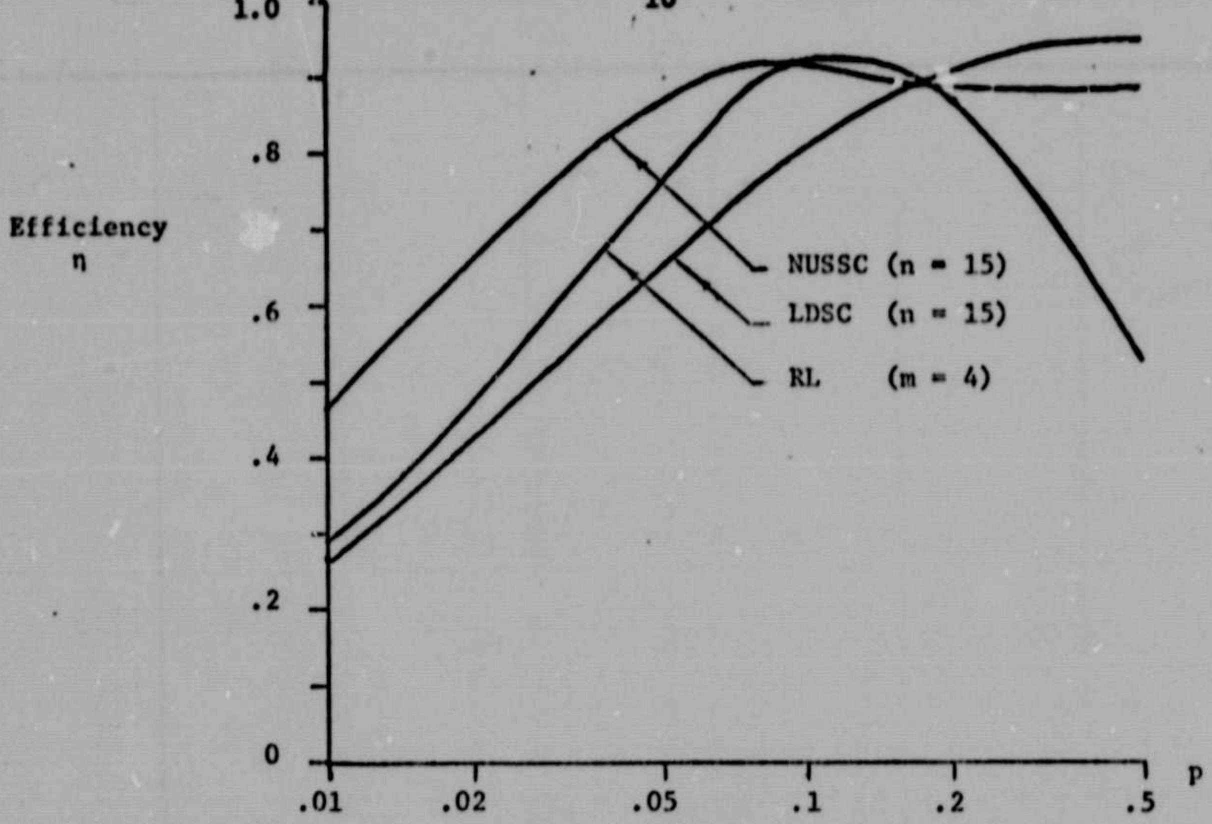

Fig. 3. Efficiencies of NUSSC, LDSC, and RL coding of length 15 sequences from memoryless binary sources with probability p of emitting a "1".

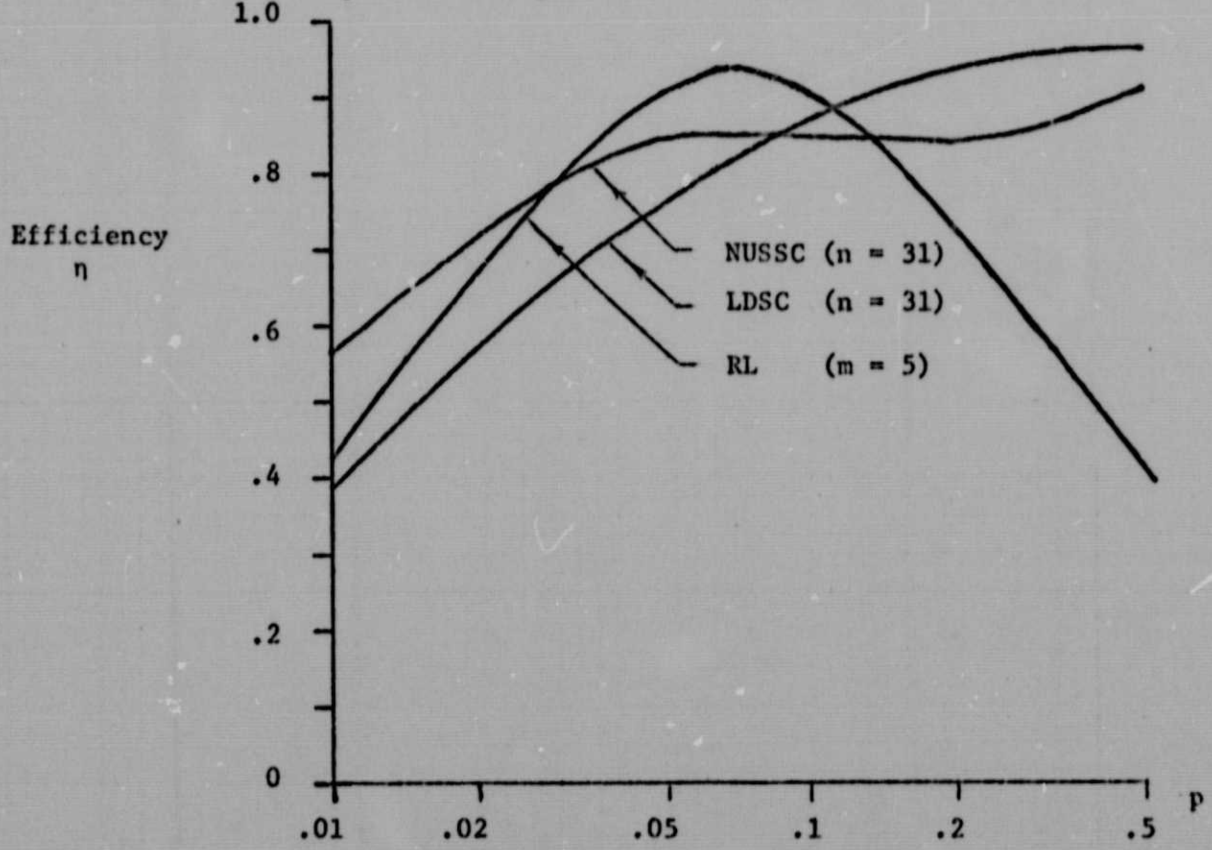

Fig. 4. Efficiencies of NUSSC, LDSC, and RL coding of length 31 sequences from memoryless binary sources with probability p of emitting a "1".

between that of the simple RL coding scheme and that of the LDSC coding scheme.

By way of analogy to LDSC coding, one can view NUSSC as a similar indexing of n-tuples where the index of an n-tuple $\underline{e}$ in the set E_i is taken as its syndrome $\underline{s}$ under the parity-check matrix H_i, rather than as its rank in some ordering of the sequences in E_i. The linear mapping from $\underline{e}$ to $\underline{s}$ would seem generally simpler to implement than a lexicographical ranking algorithm for a comparably-sized set of sequences, although in general the former mapping does not use the minimum number of encoded bits while the latter always does. The greater efficiency of n = 15 NUSSC over n = 15 LDSC coding as shown in Figure 3 might thus seem paradoxical, but is explained by the fact that reduction of the prefix length from 4 to 2 for the NUSSC compared to LDSC, because the former partitions V into 4 rather than 16 subsets, more than compensates for the slightly less efficient indexing of the sets. Indeed, the practical potential of NUSSC would seem to reside in the wealth of possible partitions of V whose component sets can be simply indexed by syndromes of linear codes.

As another example of NUSSC, we take n = 31 and M = 8 and choose: $V_1 = V$ and $E_1 = \{\underline{0}\}$; V_2, V_3 and V_4 as the (31,26), (31,21) and (31,16) BCH codes with E_2, E_3 and E_4 as the n-tuples of weights 1, 2 and 3 respectively; V_5 as the (31,11) BCH code and E_5 as the n-tuples of weights 4 and 5; V_6 as the (31,6) BCH code and E_6 as the n-tuples of weights 6 and 7; V_7 as the (31,1) BCH code and E_7 as the n-tuples of weights 9 through 15 inclusive; and $V_8 = \{\underline{0}\}$ and E_8 as the n-tuples of weight 16 and greater.

In Figure 4, we show the efficiency for this n = 31 NUSSC scheme, as well as the efficiencies of m = 5 RL coding and n = 31 LDSC coding, for the same ensemble of sources as used in Figure 3. Our discussion of Figure 3 applies almost verbatim to Figure 4. We do note, however, that, as can be seen by comparing Figures 4 and 5, the n = 31 NUSSC tends to give more nearly uniform performance than n = 15 NUSSC although with a smaller "peak efficiency"; in a sense, the n = 31 NUSCC scheme is the more "universal."

V. HISTORICAL BACKGROUND AND REMARKS

The first explicit use of "syndrome-source-coding" appears to be by Ohnsorge [13] who considered using the syndrome of a length-n, t-error-correcting, cyclic code to code n-tuples of weight t or less. Fung, Tavares and Stein [14] used a very similar procedure, except that, rather than always coding source sequences of length n, the" coded n' source digits with n - n' dummy 0's appended when there were t 1's in the first n' positions and n' $\leq$ n. This pre-conditioning of the source results in distortionless encoding. There seems to have been no prior use of "universal" syndrome-source-coding, however.

Less explicit, but earlier, use of what can be interpreted as syndrome-source-coding was made by Blizard [15] who considered "convolutional coding" of a binary source with code rate greater than unity (so that compression is achieved) coupled with a sequential decoder whose "metric" is determined by the source statistics. Forney [16] also considered, at about the same time, a similar scheme but discarded it as impractical because of the heavy computational load on the sequential decoder; Forney did, however, interpret the encoded sequence as a syndrome sequence for a code of rate less than unity. More recently, there has been work on joint source-channel encoding which is similarly related to syndrome-source-coding. Koshelev [17] has studied the encoding of sources by a convolutional encoder (without the usual pre-encoding to remove source redundancy before channel coding) in which the encoded output is directly transmitted through the channel and the source sequence is recovered by sequential decoding. He proved that it is possible to obtain arbitrarily small average Hamming distortion whenever the rate R of the code is less than R_{comp}/H_{comp}, where R_{comp} is the usual computational cutoff of the channel and H_{comp} is a quantity depending only on the source (H_{comp} > H.) Hellman [18] pursued the same approach as Koshelev, but noted that, in the noiseless case when

$R_{comp} = 1$, R would normally be greater than 1 and the scheme would be performing data compression. Using random coding arguments, Hellman showed that there exist convolutional codes for memoryless sources that obtain arbitrarily small average Hamming distortion with the number of compressed bits per source letter arbitrarily close to the source entropy. The syndrome-source-coding interpretation, as given in Section II, provides a perhaps simpler path to the same result.

ACKNOWLEDGMENT

I am grateful to one of the referees for calling my attention to the work of Blizard and Hellman. I am also grateful to Professor J. L. Massey for his assistance with the preparation of this paper.

REFERENCES

[1] T. Berger, Rate Distortion Theory: A Mathematical Basis for Data Compression, Englewood Cliffs, N.J., Prentice Hall, 1971, (see pp. 200-207).

[2] T. Berger "Data Compression Theory and Practice," Conference Record, IEEE National Telecomm. Conf., Atlanta, Ga., Vol. II, pp. 28A1-28A4, Nov. 26-28, 1973.

[3] F. Jelinek, "Tree Encoding of Memoryless Time Discrete Source with Fidelity Criterion," IEEE Trans. on Info. Theory, Vol. IT-15, pp. 584-590, Sept. 1969.

[4] A. J. Viterbi and J. K. Omura, "Tree Encoding of the Memoryless Discrete Time Sources with the Fidelity Criterion," IEEE Trans. on Info. Theory, Vol. IT-20, pp. 325-331, May 1974.

[5] T. J. Goblick Jr., "Coding for Discrete Information Source with Distortion Measure," Ph.D. Dissertation, Dept. of Elect. Engr., M.I.T. Cambridge, Mass., 1962.

[6] R. G. Gallager, Information Theory and Reliable Communication, Wiley and Sons, Inc. New York, 1968.

[7] L. D. Davisson, "Universal Noiseless Coding," IEEE Trans. on Info. Theory, Vol. IT-19, pp. 783-795, November 1973.

[8] J. L. Massey, Threshold Decoding, The M.I.T. Press, Cambridge, Mass., 1963.

[9] T. J. Lynch, "Sequence Time Coding for Data Compression," Proc. of the IEEE, Vol. 54, pp. 1490-1491, October 1966.

[10] L. D. Davisson, "Comments on Sequence Time Coding," Proc. of the IEEE, Vol. 54, p. 2010, December 1966.

[11] J. P. M. Schalkwijk, "An Algorithm for Source Coding," IEEE Trans. on Info. Theory, Vol. IT-18, pp. 395-399, May 1972.

[12] T. M. Cover, "Enumerative Source Encoding," IEEE Trans. on Info. Theory, Vol, IT-19, pp. 73-76, January 1973.

[13] H. Ohnsorge, "Data Compression System for the Transmission of Digitalized Signals," Conference Record, IEEE International Conference on Communications, Seattle, Washington, Vol. II, pp. 485-488, June 1973.

[14] K. C. Fung, S. Tavares, and J. M. Stein, "A Comparison of Data Compression Schemes Using Block Codes," Conference Record, IEEE International Electrical and Electronics Conference, Toronto, Canada, pp. 60-61, October 1973.

[15] R. B. Blizard, "Convolutional Coding for Data Compression," Report R69-17, Martin Marietta Corporation, Denver Division, 1969.

[16] G. D. Forney, Private Communication, June 3, 1975.

[17] V. N. Koshelev, "Direct Sequential Encoding and Decoding for Discrete Sources," IEEE Trans. on Info. Theory, Vol. IT-19, pp. 340-341, May 1973.

[18] M. E. Hellman, "Convolutional Source Encoding," To appear in IEEE Trans. on Info. Theory, November 1975.