01 Jan 1978

Data Compression in Microprocessor-Based Data Acquisition Systems

Paul D. Stigall
Missouri University of Science and Technology, tigall@mst.edu

Philip Panagos

Recommended Citation

P. D. Stigall and P. Panagos, "Data Compression in Microprocessor-Based Data Acquisition Systems," *IEEE Transactions on Geoscience Electronics*, vol. 16, no. 4, pp. 323 - 332, Institute of Electrical and Electronics Engineers, Jan 1978.
The definitive version is available at https://doi.org/10.1109/TGE.1978.294592

[4] B. F. Howell and P. H. Licastro, "Dielectric behavior of rocks and minerals," *Amer. Mineralog.*, vol. 46, pp. 269-288, Mar.-Apr., 1961.
[5] M. Sucher and J. Fox, *Handbook of Microwave Measurements*, vol. 2. New York: Polytechnic.
[6] C. A. Balanis, W. S. Rice, and N. S. Smith, "Microwave measurements of coal," *Radio Sci.*, vol. 2, no. 4, pp. 413-418, Apr. 1976.
[7] C. A. Balanis, "Dielectric constant and loss tangent measurements at 60 and 90 GHz using the Fabry-Perot interferometer," *Microwave J.*, vol. 14, no. 3, pp. 39-44, Mar. 1971.
[8] D. T. Paris and K. F. Hurd, *Basic Electromagnetic Theory.* New York: McGraw-Hill, 1969, p. 323.

Data Compression in Microprocessor-Based Data Acquisition Systems

PAUL D. STIGALL, MEMBER, IEEE, AND PHILIP PANAGOS, MEMBER, IEEE

Abstract—A first-order predictor routine and an encoding routine were implemented on an Intel-8080-based microprocessor system to determine their ability to compress seismic data. The predictor routine obtained a maximum compression ratio of 2.33. It appears that higher order predictors offer no significant advantage over the first-order predictor in compressing seismic data. The encoder routine obtained a larger compression ratio of 3.94.

The accuracy of reconstructed seismic traces is bounded by a maximum average error per point. For the predictor, the maximum average error per point is shown to be equal to the specified user tolerance. For the encoder, it is shown to be proportional to the maximum quantization error associated with the ranges used in the encoding algorithm. In general, the average error per point is found to be equal to the square root of the mean-squared error regardless of the compression technique used.

A design for a basic data acquisition system utilizing data compression to reduce memory requirements is proposed. Examination of the design indicates that such a system can process no more than three separate data channels at a 1-ms sampling rate and that such a system is cost effective only when small numbers of data channels are to be processed.

I. Introduction

WITH THE advent of microprocessors, small, easily transportable microprocessor-based data acquisition systems with data processing capabilities are now possible for application to on-site field data acquisition. Generally, field data, such as seismic data, can be handled by one of two methods. The first method consists of transmitting the data acquired by the sensors over a telephone line or radio link to a large central computer for processing and storage. This central computer could be a company's main computer facility located hundreds of miles away.

The second method uses equipment at the field site to gather, process, and store the data from the sensors. In the past, this equipment usually consisted of magnetic tape recorders for storage of analog waveforms (no processing of data) or a minicomputer-based system which could also process the data before storage in digital form on magnetic tape or disc.

It is in regard to this second method of data acquisition, which uses on-site data collection, processing, and storage, that the increased use of microprocessors appears most likely. For here, microprocessor-based systems can be smaller, lighter, more rugged, and use less power than minicomputer or analog recording type systems because of large-scale integrated-circuit technology. To maximize these advantages, it is necessary that integrated-circuit random access memory (RAM) chips be used for data storage. However, this reduces the amount of storage available when compared to mass storage devices such as magnetic tape. Therefore, the use of RAM chips as a storage medium requires optimum use of available space.

Data compression is a technique that can be used to store data efficiently in integrated-circuit RAM memories as well as other storage media. Of the various data compression techniques available for use in data storage systems, two of these techniques, a predictor and an encoder, were implemented on the Microkit system, an INTEL-8080-based microprocessor system. Both are analyzed for their capability in compressing seismic data and for inherent errors. A design for a basic microprocessor data acquisition system utilizing data compression is then presented.

II. Analysis of Compression Techniques

To evaluate the applicability of microprocessors to data acquisition systems using data compression and to determine the effectiveness of various compression techniques, a predictor and an encoder routine were implemented on the Microkit microprocessor system. These two techniques

Manuscript received March 13, 1978.
The authors are with the University of Missouri-Rolla, Rolla, MO 65401.

0018-9413/78/1000-0323$00.75

were selected because of their suitability to the manipulative and mathematical capabilities of basic microprocessor systems without requiring additional features such as hardware multiply and divide or other external hardware processing. Each compression method was used to compress a sampled seismic velocity waveform. This compressed data was then stored in the microprocessor system's random access memory until it was punched onto paper tape so that reconstruction of the waveform could be done by an IBM 370 computer.

A. Test Data

The seismic data used was originally recorded at a field test site using geophone assemblies with the analog velocity waveforms being stored on magnetic tape. Upon completion of field tests, the taped seismic data was sampled at the rate of one sample every millisecond by an analog-to-digital (A/D) converter controlled by a Data General Nova minicomputer. The 12-bit sampled values were stored on disk with 1024 points per waveform and with every sampled waveform having its own identifying record number. This sampled seismic data was used as the input to the compression routines implemented on the Microkit system, thereby eliminating the necessity for an A/D converter on the microprocessor system itself.

Once the data has been read without error into the Microkit memory, either compression routine may be used to compress the data and store this compressed data back in the system memory. The following is a discussion of the predictor routine and the results it obtained, after which the encoder routine and its results will be presented. Listings of the compression routines may be found in [1].

B. First-Order Predictor Compression Routine

Predictor algorithms attempt to predict future data values by examining previous data samples. Polynomial predictors accomplish this by passing an nth-order polynomial through $n + 1$ data points [2]. In the case of the first-order predictor, a straight line is placed through two consecutive data points, and all values must fall within a specified tolerance of this prediction line in order to be considered predictable. If a sample value is successfully predicted, it is considered to have the same value as the predicted value.

To evaluate the effectiveness of a first-order predictor in compressing data, a first-order predictor compression routine was written for the Microkit system. The routine was then used to compress a seismic trace. The following discussion presents the basic concepts involved and the results obtained with the predictor routine. It concludes with a comparison of the first-order predictor's compression capability and that of higher order predictors.

The first-order predictor routine used the equation

$$y(t) = 2y_{t-1} - y_{t-2}$$

to predict data values [3]. In this equation $y(t)$ is the predicted value, and it is a function of the two previous data samples y_{t-1} and y_{t-2}. Upon execution of the routine, the first two seismic values are stored in memory, and a predicted value for the third sample value is computed per the above equation. Then the difference between the predicted value and the actual sample value is computed. If the difference is less than the user specified tolerance, the sample is considered to be successfully predicted, and the point is not stored. This continues by using the predicted value in place of each sample value predicted until a sample point that is not successfully predicted is reached. This sample value is then stored in memory and is used with the previously predicted sample value to predict the next sample value. This procedure continues until all points have been processed. Thus only those points which were not successfully predicted are stored in memory.

Characteristic of all predictor algorithms, it is not sufficient to just store the unpredictable values since there is no way to determine at what time these points occurred, thereby making reconstruction of the trace impossible. Therefore, some sort of time-tag or run-length indicator is required. In the predictor routine written for the Microkit, every sample point has a bit associated with it such that the bit is set to 1 if a particular sample is successfully predicted and is set to 0 otherwise. Data words made up of these bits are produced which, when combined with the stored unpredictable values, provide sufficient information to allow the compressed waveform to be reconstructed so that each point is within a specified tolerance of the original sample point it replaces.

Reconstruction of the compressed trace was accomplished with a Fortran program written for an IBM 370 computer called the Predictor Reconstructor Program (see [1] for listing of Program). Input data to this program was obtained by having the Microkit system punch the compressed data (the stored unpredicted sample values and the data words consisting of the bits which indicate if a sample was successfully predicted or not) onto paper tape using a teletype interfaced to the system. After compression, the 12-bit (three hexadecimal characters) unpredicted sample values are interleaved in memory so that only three memory locations are required for every two sample values as shown below:

$$XX\ XY\ YY.$$

The X's represent the hexadecimal characters of one sample value and Y's the hexadecimal characters of another. When punched onto tape, each character is represented by its ASCII code. The data on this tape may then be punched onto cards and used as input to the reconstruction program.

Fig. 1 is a plot of the original seismic trace used for the compression analysis of both compression routines. It consists of 1024 sample points at 1-ms intervals. Fig. 2 is a plot of the same trace only after it has been compressed by the first-order predictor routine and then reconstructed. Each predicted value for this plot was required to be within a tolerance range of ±31 of the original sample value. As shown in Fig. 2, this resulted in a jagged-looking reconstructed trace, particularly in the smaller amplitude areas of the trace. However, the information required to reconstruct this trace could be stored in less than half the memory space required for the original trace data. More precisely, using the first-order predictor with a maximum tolerance of 31 to compress the

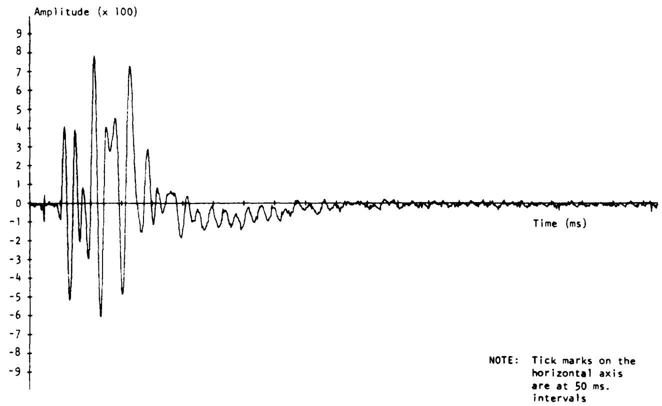

Fig. 1. Original seismic trace.

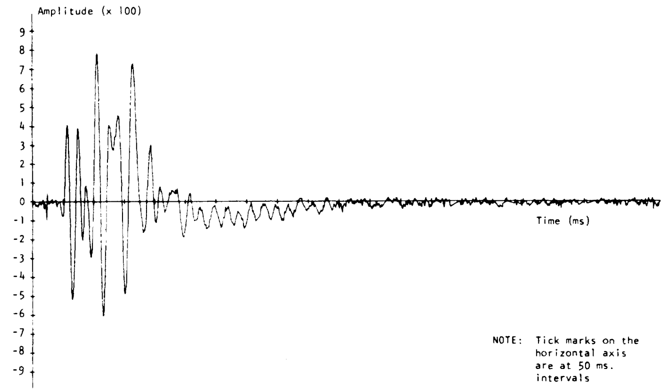

Fig. 3. Reconstructed trace using a first-order predictor, $T = 15$.

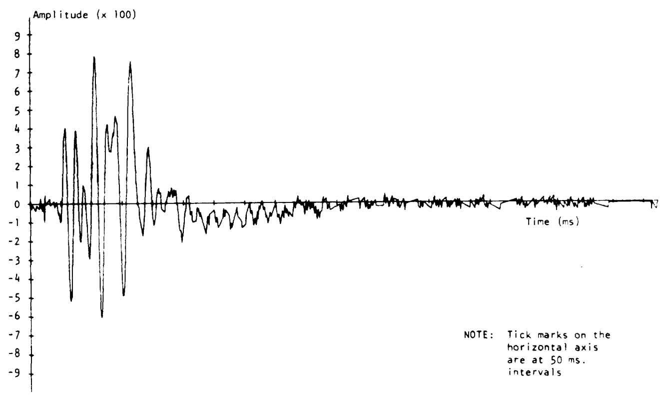

Fig. 2. Reconstructed trace using a first-order predictor, $T = 31$.

TABLE I
PREDICTOR COMPRESSION RESULTS

Predictor Order	Tolerance	MSE_{400}	MSE_{1024}	# of Pts. Predicted	Peak Value	Peak Occurred	Comp. Ratio
1st	31	186.44	213.11	669	784	106	2.33
1st	15	33.89	40.94	523	784	106	1.75
2nd	31	143.18	138.04	419	797	107	1.48
2nd	15	25.48	25.83	336	784	106	1.32
4th	31	65.22	64.01	171	811	106	1.09
4th	15	7.46	7.05	95	784	106	1.01

original data resulted in a compression ratio of 2.33 as listed in row 1 of Table I.

Besides using a subjective visual comparison of traces to determine how well a reconstructed trace reproduces the original, it is possible to compute the mean-squared error (MSE) for each reconstructed trace which provides a numerical basis for comparison. It will be seen that large MSE's are not unacceptable when the reconstructed trace is used for visual analysis. As listed in row 1 of Table I for the reconstructed trace of Fig. 2, the MSE was found to be 186.44 over the first 400 points while it was 213.11 over all 1024 points. Since it is four and one-half times smaller, this latter value compares very favorably to a maximum possible MSE of 961 over all 1024 points when using a tolerance of 31.

Also listed in Table I for the first-order predictor with tolerance of 31 is the peak value obtained by the trace and the number of the sample which attained the peak value. The original trace reached a peak of 784 at the 106th sample which are the same values obtained by the reconstructed trace. Thus the basic characteristics of the original trace were accurately preserved, and the MSE is relatively small indicating that visually the reconstructed trace is very similar to the original. As can be seen, this last statement is particularly true over the part of the trace having the large amplitude excursions.

The same first-order predictor routine was used to compress the seismic trace in Fig. 1, only this time a maximum tolerance of 15 was specified. The resulting reconstructed trace is shown in Fig. 3, and the associated information is tabulated in row 2 of Table I. Decreasing the maximum tolerance resulted in a drop in the compression ratio of 25 percent from 2.33 to 1.75. However, a sizable reduction of 81 percent in the MSE over all 1024 points was obtained going from 213.11 to 40.94. The decreased MSE is evident in Fig. 3 where the jaggedness of Fig. 2 has been smoothed out and the smaller amplitudes near the end of the trace are no longer represented by such long line segments. Also, the peak value and sample at which it occurs are the same as for the original trace. Therefore, the smaller maximum tolerance results in greater accuracy of reproduction, but this is attained with a substantial drop in the compression ratio.

For comparison purposes, second- and fourth-order predictor compression programs were written in Fortran for an IBM 370 computer. Both programs were run with the same maximum tolerance values of 15 and 31 as used in the microprocessor-implemented first-order predictor routine. The results of both these programs are tabulated in Table I, and the resulting reconstructed traces are shown in Figs. 4–7.

The combined results of Table I indicate that as the order of the predictor increases the compression ratio decreases. In fact, there is virtually no compression of data when the fourth-order predictor compression algorithm is used, which is why the traces obtained in Figs. 6 and 7 are nearly identical to the original trace. That the traces are nearly identical is understandable since only 95 points were successfully predicted when the tolerance equalled fifteen. Thus 929 points of the total 1024 were identical to the original points.

Of the two predictors, the results of the second-order predictor compare most closely to those of the first-order

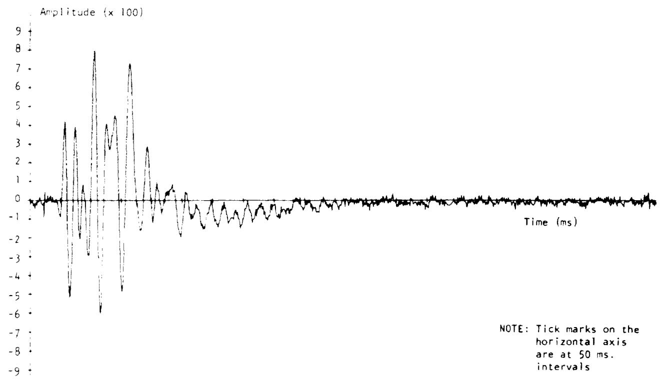

Fig. 4. Reconstructed trace using a second-order predictor, $T = 31$.

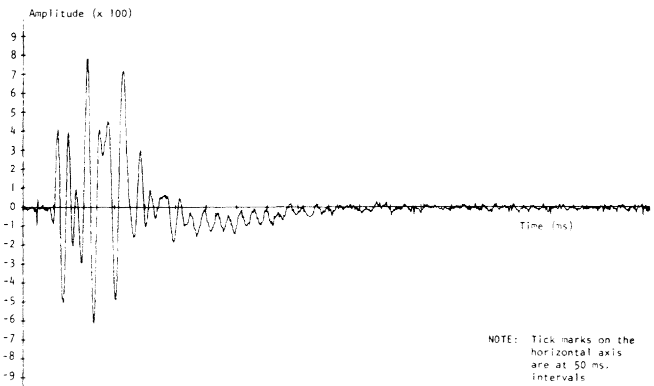

Fig. 5. Reconstructed trace using a second-order predictor, $T = 15$.

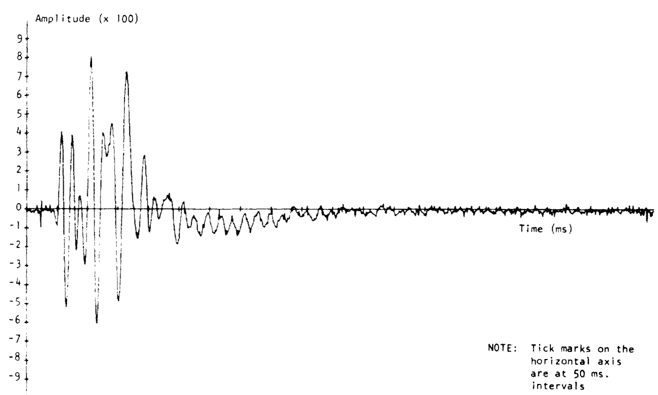

Fig. 6. Reconstructed trace using a fourth-order predictor, $T = 31$.

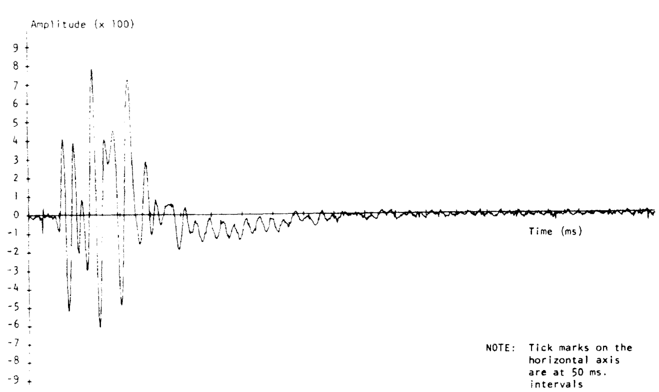

Fig. 7. Reconstructed trace using a fourth-order predictor, $T = 15$.

TABLE II
ENCODER COMPRESSION ROUTINE RANGES AND CODEWORDS

Range	From - To	Codeword	Probability
1	0 - 8	0	.525
2	9 - 19	10	.259
3	20 - 39	110	.107
4	40 - 59	1110	.031
5	60 - 79	11110	.013
6	80 - 100	111110	.015
7	101 - 122	1111110	.014
8	123 - 153	11111110	.015
9	154 - 174	111111110	.009
10	174 - 211	111111111	.005

predictor. When the tolerance equals 31, both have about the same MSE, especially over the first 400 points. However, the second-order predictor has a much smaller compression ratio. The above statements also apply when a tolerance of 15 is specified. In most cases, therefore, it would appear that the first-order predictor offers the best trade-off among compression ratio, MSE, and preservation of trace characteristics, thus eliminating the need for higher order predictor compression algorithms when compressing seismic data.

C. Encoder Compression Routine

Encoding data to reduce memory storage requirements calls for statistical knowledge of the data. The basic information required is the distribution of the sample value amplitudes. Once the probabilities of occurrence for a particular range of values is determined, a short codeword may be assigned to the ranges which contain values that occur more frequently and longer codewords to those whose points occur less frequently. Other methods of encoding are discussed in [4].

To reduce the dynamic range covered by the sample amplitude values of the original trace, the difference between consecutive values was computed by the encoder routine and then encoded. This is a technique similar to that used in communications systems for Delta modulation.

Table II lists the codewords and ranges of difference values used by the encoder routine. If a difference value falls within a particular range, it is represented by the corresponding codeword, and this codeword then is stored in memory in place of the difference. Upon decoding, the codeword is given the middle value of the range it represents.

The encoder compression routine was implemented on the Microkit system and used to compress the same seismic trace as did the first-order predictor. Upon execution of the encoder routine, the first sample point of the original data is stored in memory to serve as a reference point for the decoder program. The encoder routine then computes the difference between consecutive samples and stores the codeword for each difference value in memory. It also sets the bit associated with the difference to 1 if the difference is negative. Thus the codeword associated with the difference for every two consecutive points is stored in memory and, therefore, no time tag is required as it was in the predictor routine.

Once all points have been processed, the stored encoded data described above may then be used for input to the

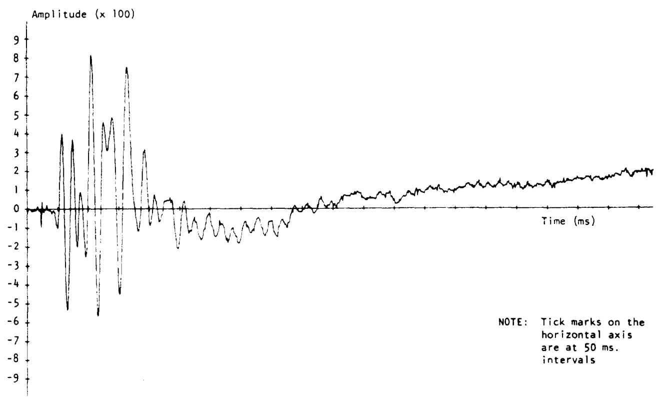

Fig. 8. Decoded seismic trace.

TABLE III
ENCODER ROUTINE RESULTS

MSE_{400}	834.6
MSE_{1024}	9256.4
Peak Value	815.5
Peak Occurred	106
Comp. Ratio	3.94

decoder program. The decoder program written in Fortran for the IBM 370 reads the encoded data and decodes it. The difference represented by each codeword is assumed by the program to have the value of the middle point of the range in which it falls. These decoded difference values are added in such a way as to reconstruct the points of the trace. For example, the difference between the first and second samples is decoded and added to the stored value of the first sample. Remember that the first sample point is always stored by the encoder routine. This sum is equal to the value of the second sample point as shown below for the general case.

Let

$$D_i = S_{i+1} - S_i$$

where D_i is the ith difference computed by the encoder and S_i is the ith sample value.

The value of D_i is then encoded by the encoder routine. The value of S_{i+1} is then computed by the decoder program as follows:

$$\hat{S}_{i+1} = \hat{S}_i + E[D_i]$$

where $E[D_i]$ is the encoded value of the ith difference. Note that if the encoded value of the difference is equal to the value computed before encoding, then $\hat{S}_{i+1}$, the decoded value, is exactly equal to S_{i+1}, the original sample value, if $\hat{S}_i$ equals S_i. Thus the error between original and decoded sample values is determined by the accuracy with which the difference is encoded. In other words, the quantization error of the difference values must be minimized. As will be discussed shortly, this error can cause a propagation of error through all points decoded.

The seismic trace shown in Fig. 8 is the result of encoding and decoding the original seismic trace shown in Fig. 1 using the ranges and codewords of Table II. It should be noted that the probabilities for ranges 5-8 do not always decrease with increasing codeword length. For example, the probability of occurrence for points in range 5 is less than that of range 6. This, of course, is opposite the desired order of decreasing probability with increasing codeword length. However, to encode the difference values with the ranges not in consecutive order results in greater complexity of the encoding routine. It was decided that the increased complexity of the routine would not provide a sufficient increase in compression to warrant assigning codewords to the ranges in a nonconsecutive order. In fact, very little change in compression ratio would result since the probabilities for ranges 5-8 are almost equal.

The encoding routine obtained a compression ratio of 3.94. This means that the trace of Fig. 8 could be stored in only slightly more than one fourth the memory space required to store the original data. However, as can be seen in Fig. 8, the trace looks like the original only over about the first half of the trace. The remaining points of the trace are removed from the horizontal axis by a steadily increasing offset. The offset is due to the accumulation of error resulting from the quantization of the difference values as mentioned above. Since each difference is added to a computed sample value to produce the next sample value, the error associated with the difference to be added plus all previous differences required to produce the computed sample are combined. Of course, these summed errors may tend to cancel, but this is not entirely the case with the seismic data used for this analysis.

As listed in Table III, the accumulated error causes the MSE's to be large compared to those obtained by the first-order predictor. However, the MSE for the first 400 points yields an average error per point of about 29, which is less than the maximum average error per point of 31 when using a predictor with a tolerance of 31. The average error per point actually obtained by the predictor with a tolerance of 31 was not that large, however, only equaling approximately 14.0. Table III also indicates that the peak value obtained by the encoded trace is very close to the original trace value of 784, and its time of occurrence is exactly the same as that of the original. Visual inspection of Fig. 8 shows that the large amplitude part of the trace is very nearly the same as the original over the same part.

Although it was not attempted, it is believed that the offset of the points in the second half of the trace could be eliminated by digital filtering of the data before plotting. A digital implementation of a high-pass filter should remove the slowly increasing offset without having ill effect on the desired information since it represents a much lower frequency compared to that of the data. However, even with the offset, the general shape of the entire trace is very similar to that of the original if it is examined while ignoring the shift in position of the low amplitude points. Ignoring the offset, the encoder routine maintains the general shape and characteristics of the waveform while providing large compression of the input data.

D. Error Analysis of the Predictor and Encoder Routines

The following error analysis of the two compression routines is straightforward. In both cases, the quantization error due to the sampling of the original analog data by the A/D is omitted from the calculations since it is insignificant compared to the errors introduced by the compression routines themselves.

It can be shown that the average error per point is equal to the square root of the MSE. For the predictor, the largest error for any one point is equal to the specified tolerance. If the tolerance T is equal to some specified value, the maximum MSE for all 1024 points is computed as follows:

$$\mathrm{MSE}_{\max} = \frac{\sum_{1=1}^{1024} T^2}{1024} = \frac{1024\ T^2}{1024} = T^2.$$

Since the average error per point equals the square root of the MSE, the maximum average error per point for the predictor is equal to T. Intuitively, this is the expected result. Thus maximum MSE in the predictor compression routine may be controlled by proper selection of the tolerance, but decreased MSE will usually be achieved at the expense of the compression ratio as previously demonstrated.

In the predictor routine, allowing the predicted value to be within a tolerance of the original value introduces the only error inherent to the routine. In the encoder routine, the only error introduced by the routine itself is the quantization error of the difference values since each value is represented by the middle value of the range in which it falls. The worst case error for a difference value would then be equal to one half the largest range of values used. For example, in Table II range 10 is the largest with a width of 37. Thus, if a difference of value 174 were computed, it would be considered to have the value of 192.5, which is a quantization error of 18.5 or one half the range's width. If the maximum quantization error possible is represented by q_m, then the maximum MSE is computed as shown below, where $\hat{S}_i$ is the ith decoded sample value and D_i is the ith difference, which is the value of the ith computed difference before encoding. Thus $D_i + q_m$ represents the decoded difference value.

Since S_1 is the value of the first point of the original data then

$$\hat{S}_2 = S_1 + D_1 + q_m$$

$$\hat{S}_3 = \hat{S}_2 + D_2 + q_m = S_1 + D_1 + D_2 + 2q_m$$

$$\hat{S}_4 = \hat{S}_3 + D_3 + q_m = S_1 + D_1 + D_2 + D_3 + 3q_m$$

$$\vdots$$

$$\hat{S}_i = S_1 + \sum_{j=1}^{i-1} D_j + (i-1)q_m.$$

Therefore,

$$\mathrm{MSE}_{\max} = \frac{\sum_{i=1}^{K} [(i-1)q_m]^2}{K} \tag{1}$$

$$= \frac{\sum_{i-1}^{K} (i-1)^2\, q_m^2}{K}$$

$$= \frac{\left[\frac{K(K+1)(2K+1)}{6}\right] - K^2 \quad q_m^2}{K}$$

$$= \left(\frac{1}{3}K^2 - \frac{1}{2}K + \frac{1}{6}\right) q_m^2. \tag{2}$$

Thus the maximum MSE over K points is proportional to the square of the largest quantization error possible. Note that in the computation of the decoded sample values the errors (q_m's) do tend to accumulate and are propagated throughout the remaining points as indicated by Fig. 8.

In Table III, the MSE for the first 400 points is shown to be 834.6 for the decoded trace of Fig. 8. This compares to a maximum possible MSE computed using (2) of approximately 18×10^6. The five orders of magnitude difference between the two values is due in part to the fact that using q_m equal to 18.5 is not a good approximation of the average maximum quantization error. The average maximum quantization error for the 10 ranges is computed as follows:

$$q_{\text{avg}} = \sum_{i=1}^{10} P_i \times q_{m_i} \cong 5.6$$

where P_i is the probability of occurrence for points in the ith range and q_{m_i} is the maximum quantization error for the ith range. Using this value for q_m in (2) yields a maximum MSE of 1.6×10^6. This value is still much larger than the MSE actually obtained. Thus it must indicate that the errors do tend to cancel, although not completely, since there is an obvious offset of the small amplitude sample values over the second half of the trace.

Depending on the data to be encoded, the quantization error can have a significant effect on the MSE and, therefore, the error per point. In fact, the error inherent in the encoding routine is extremely sensitive to the quantization error of the difference values. Theoretically, to obtain a maximum MSE equal to that of the predictor compression routine with a tolerance of 31 requires a maximum average quantization error of 1.35×10^{-1}. This reduced MSE, of course, would be obtained with a substantial drop in compression, and, possibly, a compression ratio of less than one would result. This value of the average q_m does not take into account any tendency for cancellation of error in the decoded trace. Cancellation is found to be very significant in reducing the MSE of the trace in Fig. 8 and should not be discounted when encoding seismic data.

III. A Proposed Acquisition System

This section briefly presents a possible design for a data acquisition system that uses data compression to reduce memory storage requirement. It is assumed that the data to be compressed is similar to the seismic data previously discussed and, therefore, that either the predictor or encoder compression routine may be used by the system to reduce memory requirements. The following discussion explains why various components of the system were chosen and how they are used by the system to compress and store the acquired data. A more general presentation of design procedures for seismic data acquisition systems is available in [5].

A. System Hardware

1) General Components: The proposed data acquisition system is shown in Fig. 9. The microprocessor selected for this system is the INTEL 8080A-1. The 8080A-1 has the fastest instruction cycle time of the various 8080 CPU's available and can execute a single pass of the predictor compression routine in a maximum time of about 0.32 ms. This would allow three different data channels to be sampled at a 1-ms rate and compressed and stored in memory using the routine as it was written for the Microkit system. The encoder routine, however, requires about 0.42 ms maximum for each pass. This execution time could be reduced by replacing each call to a subroutine by the instructions of the particular subroutine called. Such substitution was not done in the encoder routine since it involved tedious repetition of instructions and since it did not affect analysis of the encoder's compression capabilities. Also, general optimization of the existing routine could be done to reduce execution time. Thus either of the routines would allow three data channels to be processed with a 1-ms sampling rate in a properly designed system based on an 8080A-1.

The 8080A-1 is supported by a family of chips which when used can help reduce chip count and eliminate external logic circuitry. For this reason and because of availability, internal configuration, and speed, the 8080A-1 was chosen for this system. One of the chips in the family essential to the operation of the 8080A-1 is the 8224 clock chip. This chip utilizes a clock crystal to generate the two nonoverlapping clock signals ϕ_1 and ϕ_2 necessary for CPU operation.

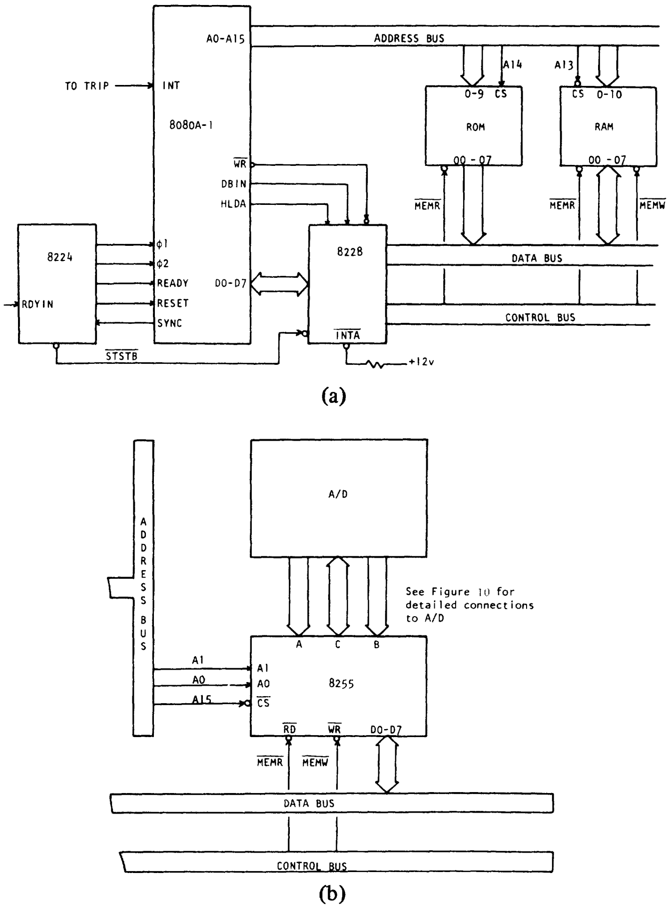

Fig. 9. Data acquisition system hardware design.

The 8224 also contains logic that produces the status strobe ($\overline{\text{STSTB}}$) signal and allows a wait state to be requested on the RYDIN line. In general, these last two functions are required for synchronizing the rest of the system to the CPU. If the 8224 chip were not used, significant logic circuitry would be required to replace it, thereby increasing chip count and the possibility of malfunction due to the increased number of electrical connections required.

For much the same reasons, another member of the family, the 8228 system controller, was used. This chip takes the status information available on the data bus during the first state of every machine cycle and generates all signals required to interface the CPU to the RAM, ROM, and I/O components. It also serves as a bidirectional bus driver while providing the necessary drive capability required for the 8080A-1 data bus. This chip could be replaced by two 8216's (4-bit bidirectional bus driver), an 8212 (8-bit I/O-port) and some logic gates and inverters to generate the control signals, but this would result in an unnecessary increase in complexity.

2) Memory Requirements: In order for three different data channels to be processed, the CPU must operate at the maximum possible clock rate. However, if the memory does not have a sufficiently small access time, the CPU will be kept waiting for data or instructions to be transferred to or from memory, thereby increasing the overall execution time for each pass of the compression routine. It was determined that if the memory access time were less than 365 ns, then the CPU would never be kept waiting for these transfers to be completed. Thus the CPU could operate at its maximum rate.

In keeping with the desire to minimize chip count and still allow maximum processor speed, a static MOS RAM was selected for this system. Using static memory eliminates the need for either software or hardware refreshing. The software refresh would increase the execution time of the compression routine, thereby possibly reducing the maximum sample rate attainable, while hardware refresh would increase chip count and the physical complexity of the system. In this proposed system, only three quarters of a kilobyte of RAM is required for storage of the compressed data if a compression ratio of four is assumed. This figure is based on 1024 samples for each of three data channels with each channel sampled at a maximum 1-ms rate. The 3 kilobytes of memory required to store this uncompressed data would be reduced by a factor of four (the compression ratio) through the use of real time data compression. The resulting memory requirements are thereby reduced to three quarters of a kilobyte of memory storage. Because of the small amount of memory needed, the disadvantages of static compared to dynamic memory, those being higher cost, higher power consumption, and lower density, are offset by the decrease in system complexity.

One memory chip which meets the system requirements is the INTEL 2102AL-2. It is a 1024 X 1 static memory chip having an access time of 250 ns and very low standby power consumption. Eight of these chips would provide the memory space for the compressed data while leaving 256 bytes for scratch pad use. The extra space could also be used to store more compressed data from each channel.

The ROM must meet the same requirements as were met by the RAM memory. One chip which does so is the INTEL 3624, a 512 X 8 bipolar PROM chip. This chip allows sufficient room for storage of either compression program. Note that a linear select scheme is used to select the desired I/O device. This is possible in this small system because no more than 1024 locations are addressed by the CPU thereby leaving six address lines free for use as select lines. Linear selecting eliminates the necessity of decoding address lines as is sometimes required in systems of larger size. This helps to reduce system complexity by reducing chip count.

3) External Equipment Interfacing: The last part of the system to be discussed concerns the interfacing of the A/D and possibly other external hardware to the acquisition system. The following discussion assumes a fixed amplification of the input signals. A discussion of variable gain amplifiers and their interfacing is found in [5].

An INTEL 8255 programmable peripheral interface chip is used to interface all external hardware to the system bus. It can be configured by the proper control word into mode 0 so that ports A, B, and half of C are used for input and the other half of port C for output. Address lines $A0$ and $A1$ select the desired port while $A15$ is the line which selects the 8255. The direction of data flow is controlled by the lines connected to the $\overline{\text{RD}}$ and $\overline{\text{WR}}$ pins.

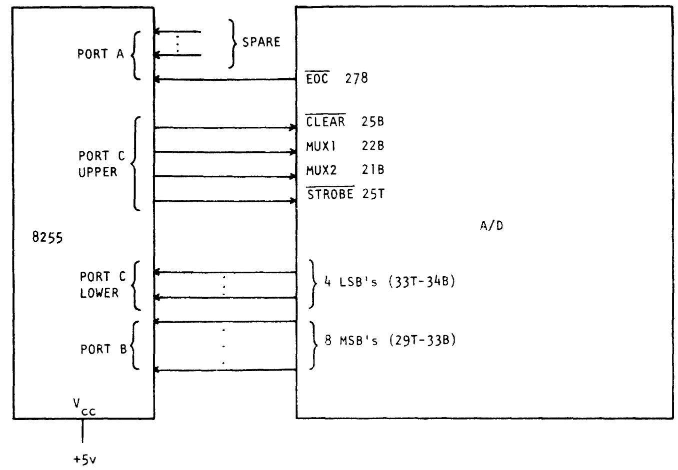

Fig. 10. Configuration of the 8255 and A/D.

All eight lines of port B and the four input lines of C are used to input the 12-bit data words from the A/D. The connections related to this discussion are shown in Fig. 10. The connections to the A/D shown in this figure are based on the Analogic MP6812 and are generally those required for any A/D. The remainder of the port C lines are used to start the sample and conversion process ($\overline{\text{STROBE}}$), select the input channel to the multiplexer (MUX1, MUX2), and to reset the MUX address lines to channel zero ($\overline{\text{CLEAR}}$). Only one line of port A is used with A/D. This line is used to input the signal ($\overline{\text{EOC}}$) which indicates the end of conversion for the sample taken. When this line goes low, the data is ready to be read into the system for processing. The seven remaining lines of port A can be used for inputs from the operator. Latches can be used to indicate the position of switches that could be set by the operator to convey various information to the stored program. Thus some flexibility could be afforded without reprogramming the PROM chip.

The A/D conversion time in this system is not overly critical because of the relatively slow sample speeds. Excellent hybrid 12-bit A/D's capable of 8-μs conversion times are available which are more than satisfactory and cost under \$100 [6]. The main problem with interfacing the A/D to the system with an 8255 is that the 8255 has a long delay when transferring data to or from the system bus. Thus, every time the 8255 is selected, an extra wait state must be inserted to give the data time to propagate through it.

Data acquisition can be synchronized with the data source when necessary. For example, when the signal to detonate the charge used in seismic exploration is given, a signal can also be sent to the data acquisition system. If the CPU is in the halt state awaiting the signal, applying the signal to the interrupt request line (INT) will cause the processor to leave the halt state. By tying the $\overline{\text{INTA}}$ line of the 8228 to a +12-V supply, a RST 7 is forced onto the bus when the interrupt occurs causing the CPU to begin execution at location 56_{10}. A jump to the compression routine could be stored starting at this location.

After the data has been compressed and stored in RAM, it is ready to be output to a permanent storage device for another system which will decompress, analyze, and plot the data. Semiconductor memory requires power at all times or else the information stored therein is lost. This leaves two main ways to output the compressed data. The first of these is to have an external connector which allows another device or system access to the acquisition system data and address busses. This system could then access the memory and obtain the desired data. This method could only be used, of course, after the appropriate safeguards had been taken to assure electrical compatibility between the systems. The second method would be to have the RAM memory mounted on its own removable card with a small battery to supply the power necessary to maintain the stored data. This card could then be inserted in the system used to decompress the data. Although other interfacing methods could be provided such as for cassette recorders, they would require greater software and hardware complexity on the acquisition system's part.

B. System Cost

The approximate cost for the semiconductor chips and A/D for the proposed system using price per quantity of 100 is \$200. This is an approximate savings of \$70 over other systems which sample three-data channels at 1-ms rates but do not use data compression to reduce the amount of memory required to store the sampled data. All of the savings are due to the fact that data compression reduces the number of 1024×1 RAM chips required to store data from twenty-four to eight. Also, reducing the number of 1024×1 RAM chips required from twenty-four to eight yields an added benefit of reducing system complexity.

The proposed system is, of course, a general design, and many improvements could be made to make the system suit the more specific needs of a particular user. For example, only three channels can be processed at 1-ms sampling rates using the proposed system as shown. The number of channels could be increased by using a bipolar microprocessor and bipolar RAM memory. The processor speed would be much greater and this would allow more channels to be processed. However, as in all designs, the various options or trade offs must be considered in light of the requirements the system must meet and the resources available to the designer.

IV. Conclusions

Of the many compression techniques, predictors, and encoders appear most applicable to basic microprocessor data acquisition systems. For predictor compression techniques, it was determined that higher order predictors do not provide a significant benefit over the first-order predictor to warrant their use. In fact, a general trend was noted indicating that as the order of the predictor increases, the compression ratio drops substantially while little improvement in the MSE is obtained.

In a more theoretical vein, it was shown that the maximum average error per point for the predictor is equal to the specified tolerance T. Thus the maximum average error per point can be controlled by proper selection of T.

For the encoder compression technique, it was shown that the quantization error associated with encoded difference values tend to accumulate. This is the only error inherent to the compression routine. An equation was derived which can be used to calculate the largest MSE possible with this technique. This equation indicates that the maximum MSE is proportional to the square of the largest quantization error possible. However, a more accurate figure is obtained if the average quantization error is used in place of the maximum value.

Generally, both the first-order predictor and the encoder accurately reproduced the original seismic trace characteristics of peak value and its time of occurrence. The predictor obtained a compression ratio of 2.33 when T equalled 31, while the encoder compression ratio was even larger at 3.94. As mentioned above, the encoder tends to accumulate the quantization error associated with the encoded difference values. This caused a slowly increasing offset in the decoded trace, particularly over the last 500 points. However, this offset could be removed by passing the data through a digitally implemented high-pass filter before plotting the decoded data. This would also reduce the MSE to a value much closer to that of the predictor over all 1024 points. Since both techniques preserve the original trace characteristics and visually are very similar to the original (after filtering for the encoded trace), either one could be used successfully to compress seismic data to reduce memory requirements. The encoder, because of its higher compression ratio, is particularly appealing even though decoding with filtering is a more complex process than that required to reconstruct the predictor trace.

The design of the proposed data acquisition system demonstrates that a basic microprocessor-based system utilizing one of the compression techniques can process three separate data channels at a 1-ms sampling rate. It was found that the limiting factors to processing more data channels are the slow processor speed and the length of the compression routine. The compression routine length is basically fixed, therefore, increasing processor speed is the only applicable remedy. By going to bipolar components in a single processor system or staying with MOS components in a multiprocessor system, processor speed can be greatly increased. This would allow more data channels to be processed.

The cost of the proposed system components is about \$200. This, of course, assumes that the system uses a compression routine capable of a compression ratio of four and components purchased in quantities of 100. The savings over a comparable system that does not use data compression to reduce memory requirements is about \$70. It must be noted, however, that the system that does not use data compression can sample about ten times the number of data channels the proposed system can. Such a system would cost about \$1200 for components [5]. For the same amount of money, only 18 data channels could be processed by the proposed system. Thus it would appear that basic microprocessor systems utilizing data compression are better suited for use where small numbers of data channels are to be processed and where a small portable system is a definite requirement.

References

[1] P. Panagos, "Microprocessor application to data acquisition systems utilizing data compression," Masters thesis, University of Missouri-Rolla, Rolla, MO, 1977.

[2] C. M. Kortman, "Redundancy reduction—A practical method of data compression," *Proc. IEEE*, vol. 55, pp. 253–263, Mar. 1967.

[3] C. A. Andrews, J. M. Davis, and G. R. Schwarz, "Adaptive data compression," *Proc. IEEE*, vol. 55, pp. 267–277, Mar. 1967.

[4] Y. Dishon, "Data compaction in computer systems," *Computer Design*, pp. 85–90, Apr. 1977.

[5] E. B. Adams, "Microprocessor-controlled data acquisition systems," Masters thesis, University of Missouri-Rolla, Rolla, MO, 1977.

[6] E. L. Zuch, "Where and when to use which data converter," *IEEE Spectrum*, pp. 39–42, June 1977.

Effect of Sonde Eccentricity on Responses of Conventional Induction-Logging Tools

STANLEY C. GIANZERO, MEMBER, IEEE

Abstract—**This paper develops the mathematical analysis for calculating the responses of a typical induction-logging tool in usual conductive drillings muds, when it is positioned at a certain standoff distance from the nearest side of the borehole wall. A complete investigation, including skin effect, reveals that the contribution of the skin effect is adequately compensated for by boosting the induction-log signal by the amount of skin effect that would be expected if the sonde were surrounded by a homogeneous medium. Thus a geometrical-factor-type theory is sufficiently general to account for the effect of eccentricity of the sonde position in the borehole. The theoretical equations developed are applied to a focused induction-logging tool with a medium depth of investigation (ILm) and a focused induction-logging tool with deep investigation (ILd). The results are found to be in fair agreement with experimental results. Some disparity in the two sets of results can be attributed to the effect of the finite size of the mandrel, which is not taken into account in the theoretical development.**

I. Introduction

Several papers have been published dealing with the responses of an induction-logging tool located within a wellbore [1]-[5].

In most of these works [2]-[5], a primary consideration was the influence of skin effect (sometimes referred to as propagation effect) upon the voltage induced in a receiver coil by the conduction currents induced in the media surrounding the sonde. The purpose of these investigations was to provide a more exact treatment of the problem than already provided by the so-called "geometrical-factor theory" of the earlier work by Doll [1]. In all of these studies the transmitting and receiving coils of the induction-logging tool were considered to be coincident with the axis of the wellbore.

Manuscript received April 4, 1977; revised April 28, 1978.
The author is with the Schlumberger-Doll Research Center, Ridgefield, CT.

In actuality, unless certain centralizing devices are employed, the induction sonde is displaced from (or eccentered relative to) the axis of the wellbore. Indeed, it is the usual practice to purposely operate with the tool positioned at a certain "standoff" distance from the nearest side of the borehole wall. Thus it is of considerable interest to the well-log analyst to be able to know the effect of the eccentering on the tool response.

This paper has as a purpose to provide the analysis necessary to calculate the influence of eccentering upon the response characteristics of any induction-logging tool. It should be noted that the mathematics is similar to that used in closely related antenna problems [6]-[11].

We shall consider the problem first in complete generality, i.e., skin effect will not be neglected. We shall then follow with the development of the approximate theory without skin effect. The mathematical relations developed will be applied to two focused induction-logging tools commonly used in practice, namely, the ILm and the ILd. The results found with the two theoretical approaches will be compared with each other and with results from experimental measurements.

II. Exact Theory—Including Skin Effect

As shown in Fig. 1, let a transmitter coil of an induction-logging sonde be represented by the magnetic dipole produced by a small current loop which is located in a cylindrical borehole of radius a and conductivity σ_1. A cylindrical system of coordinates (ρ, ϕ, z) is chosen such that the axis of the borehole is coincident with the z axis and also parallel to the orientation of the magnetic dipole. The magnetic dipole is located at the arbitrary position (ρ_0, ϕ_0, z_0) in the borehole which is surrounded by a homogeneous formation of conductivity σ_2.

0018-9413/78/1000-0332$00.75