SOME OBSERVATIONS ON DATA COMPRESSION FOR DIGITAL STRONG-MOTION ACCELEROGRAMS

BY CHIA-YEN PENG AND W. D. IWAN

ABSTRACT

This paper presents the results of a study of data compression for digital strong-motion accelerograms. The general characteristics of a class of data compression algorithms based on time history interpolation are discussed. Six particular compression/decompression algorithms are examined in detail. In four of these algorithms only peaks are recorded with different interpolation schemes being used to recover, or decompress, the original acceleration waveform between peaks. The compression/decompression algorithms based on peaks only are observed to result in generally unacceptable errors that are nonuniform in both the time and frequency domain. The remaining two algorithms are based on selectively sampling the data with a prescribed accuracy control. This approach is termed backward successive approximation (BSA). The algorithms store only those points between peaks which are required to maintain the desired accuracy. One algorithm also incorporates a complementary differential bit reduction (DBR) scheme to further reduce the storage required. An optimum balance between accuracy and degree of compression is achieved when the BSA and DBR schemes are combined.

INTRODUCTION

Digital accelerographs have a number of inherent advantages over their optical analog counterparts, including lower noise, easier data processing, greater time resolution, and the potential for greater signal resolution and dynamic range (Iwan, Moser, and Peng, 1985). One of the obstacles to the greater use of digital accelerographs has been problems related to the storage and subsequent retrieval of the digital data. Some of these problems have been associated with the characteristics of the particular memory device used. For example, magnetic tape recorders have exhibited mechanical problems, and solid state memories have exhibited premature IC failure. However, even when these problems are overcome, there is one overriding limitation which cannot be easily solved: the amount of memory available.

Digital memories, like all memories, are of finite size and can, therefore, contain only a limited amount of data. There is constant pressure from the users of digital recorders to be able to store more bits of data at higher sample rates, for longer durations, and for more events before playback is required. Since there are usually some constraints on how much memory can be incorporated into a given recorder, some major equipment manufacturers have introduced data compression schemes into their recorders.

In its simplest form, data compression may be accomplished by storing only a subset of the actual data and then reconstructing the original data from this subset by means of some mathematical algorithm. The data reconstruction process may be referred to as data decompression. In fact, all digital sampling and recording of analog data can be viewed as a form of this type of data compression. Data compression can also be achieved by manipulation of bit patterns, but this is not considered in the present study.

Data compression has been used successfully in many fields of science and engineering (Makhoul, 1975; Crochiere and Rabiner, 1981). However, its application

to strong-motion earthquake data has never been thoroughly examined. This paper presents the results of a study of various possible compression and decompression schemes as applied to strong-motion data. Based on this study, a relatively simple algorithm is presented which shows promise for application to strong-motion measurement. Using this algorithm, a substantial reduction in required data storage capacity can be achieved. This can be used to obtain improved instrument performance, lower instrument cost, or some combination of these.

GENERAL CONSIDERATIONS

The merit of a particular data compression/decompression algorithm must be judged by two factors: (1) the amount of reduction in required storage that is achieved, and (2) the accuracy of the recovered data. The reduction in storage may be measured by the Data Compression Ratio (DCR), defined as

$$\text{DCR} = \frac{\text{Amount of storage required without data compression}}{\text{Amount of storage required with data compression}}$$

The larger the value of DCR, the more efficient is the compression portion of the algorithm.

The accuracy of a data compression/decompression algorithm may be described in a number of ways. One way would be to present the full time history of the difference between the original and the decompressed data. This is referred to as the compression noise. The time history of the compression noise gives the most complete picture of the accuracy of the algorithm. Alternatively, the maximum or rms values of the compression noise could be given.

However, knowledge of the compression noise or error in the time domain alone can sometimes be misleading. Therefore, it is useful to also examine the nature of the error in the frequency domain. For this purpose, one could either compute the difference in the Fourier amplitude spectra or the Fourier amplitude spectrum of the difference in the time histories of the original and decompressed data. It is felt that the latter measure of the frequency domain error is more meaningful. Hence, this measure is employed in the present study.

CHARACTER OF STRONG-MOTION ACCELEROGRAMS

An examination of the general character of strong-motion accelerograms will help to gain insight into the requirements for compression of such data. As an example of a typical accelerogram, consideration is herein given to an aftershock of the Coalinga earthquake which was recorded on 8 May 1983. The recording was obtained using a Kinometrics PDR-1 digital recorder and FBA-13 accelerometers. The PDR-1 has a 12-bit A to D converter and employs gain-ranging for an overall dynamic range of 102 dB. The FBA-13 has a natural frequency of 50 Hz, and the input (antialiasing) frequency of the PDR-1 was set at 50 Hz. The sample rate for this study was 100 samples per sec. The major characteristics of the three components of the accelerogram are summarized in Table 1. Complete time histories of the acceleration, velocity and displacement for all three components were presented by Iwan *et al.* (Iwan, Moser, and Peng, 1985).

Figure 1a shows a reproduction of the time history of the N35E (L) component of acceleration. In Figures 1b to d, highly magnified segments of the accelerogram are shown for the pre-event, strong shaking, and end-of-event phases of the record. The characteristics displayed are typical of most strong-motion accelerograms. The

TABLE 1
CHARACTERISTICS OF TEST ACCELEROGRAM

Component	Coalinga; 8 May 1983			
	Maximum Acceleration (g)	Maximum Fourier Amplitude Spectrum (g-sec)	Maximum Velocity (cm/sec)	Maximum Displacement (cm)
L-N35E	0.102	0.042	7.37	0.97
T-S55E	0.123	0.046	4.46	0.63
V-Vertical	0.074	0.025	2.17	0.40

figures clearly show how the general amplitude, frequency, and "shape" of the waveform change with time during the recorded event. The vertical component of course has a somewhat different appearance.

During the pre-event phase, from 0.4 to 1.0 sec, in Figure 1b, it is seen that the recorded data is mainly background noise, which has a relatively high frequency and very low amplitude. For much of the time, the input to the recorder is below the resolution of the instrument, leading to a rather random, flat-topped waveform as shown.

During the strong shaking phase, as shown from 7.0 to 7.6 sec in Figure 1c, the acceleration resembles a moderately broad band random process. The maximum frequency content may be fairly high, and the peaks of acceleration do not always occur at regular intervals. However, the time history is much more regular than for the pre-event phase.

Figure 1d shows the end-of-event phase, from 29.4 to 30.0 sec. It is evident that this portion of the accelerograms is dominated by fairly long-period waves with relatively small amplitude. The time history is generally fairly smooth.

The number of peaks in a typical accelerogram will be greatest during the initial pre-event portion of the accelerogram. For the three components of the Coalinga aftershock examined herein, nearly one-quarter of the total number of peaks is accumulated during the pre-event phase, which is only one-tenth of the total duration of the recording. It may therefore be concluded that the amount of pre-event data which is recorded and the degree of information desired from these data will have a strong influence on the amount of storage required when data compression is employed.

In many respects, the compression of digital accelerograms is similar to the problem of digitizing analog accelerograms at unequal-time intervals (Hudson, 1979; Trifunac and Lee, 1973). In both cases, the goal is to pick the minimum number of prominent points on the record which may be used to reproduce an acceptable representation of the complex waveform. A knowledge of the character of strong-motion accelerograms and experience with hand digitization of accelerograms gives some useful insight into possible ways to compress these data.

DATA COMPRESSION BASED ON PEAKS ONLY

Since peaks are a very prominent feature of an accelerogram, it is tempting to consider a simple data compression scheme in which only the peak data are recorded. Intuitively, it would seem that such an approach represents a lower bound on the acceptable number of samples in the compressed record. Before considering more sophisticated schemes, four compression/decompression algorithms based on peak data are examined. In each case, only peaks are recorded, but a different method is used to recover, or to decompress, the acceleration waveform between peaks.

FIG. 1. Accelerogram for Coalinga aftershock of 8 May 1983. N35E component. Note difference in vertical scales. (a) time history of entire event, (b) time history during pre-event portion, (c) time history during strong shaking portion, (d) time history during end-of-event.

The first data compression/decompression algorithm, DC1, reconstructs the accelerogram from peaks by simple linear interpolation. In order to reduce the errors associated with this simple linear interpolation decompression scheme, cosine interpolation is applied in the second data compression/decompression

algorithm, DC2. Third-order and seventh-order polynomial interpolation de-compression schemes are employed in the compression/decompression algorithms DC3 and DC4, respectively.

An evaluation of the four algorithms is made by examining the data compression ratio and associated errors both in the time and frequency domains using the three components of the Coalinga aftershock. For all four of the peak-only based data compression/decompression algorithms, the data compression ratio for the three components combined will be 2.3, assuming that the values of both the time and acceleration are stored with 12-bit accuracy. The data compression ratio of 2.3 is computed as follows:

$$\text{DCR} = \frac{3 \times N_0 \times B_0}{(P_1 + P_2 + P_3) \times (B_1 + B_2)},$$

where

N_0 = Number of samples in one channel of original data

P_1 = Number of peak samples in longitudinal channel

P_2 = Number of peak samples in vertical channel

P_3 = Number of peak samples in transverse channel

B_0 = Number of bits used to record each sample of original data

B_1 = Number of bits used to record each sample of compressed time data

B_2 = Number of bits used to record each sample of compressed acceleration data.

Hence, for data compression based on peaks only as used herein,

$$\text{DCR} = \frac{3 \times 4096 \times 12}{(824 + 1010 + 788) \times (12 + 12)} = 2.3.$$

This is a substantial reduction but not as much as might be desired. One reason for this is that a large amount of data is retained from the pre-event portion of the record, where the data may not be particularly useful due to low signal-to-noise ratio.

Figure 2 shows the time domain compression noise for algorithms DC1 through DC4. It is seen that the noise is quite substantial, being of nearly the same order of magnitude as the original accelerogram. Furthermore, the error is not uniform in time, reaching a maximum during the strong shaking phase. The values of the maximum and rms acceleration compression noise for all three components of the example accelerogram are summarized in Table 2.

Figure 3 shows the data compression error in the frequency domain for algorithms DC1 through DC4. These errors are also considerable by comparison with the spectrum of the original accelerogram (Figure 3a). Generally speaking, the maximum error occurs at those frequencies associated with the greatest energy content in the original record. The maximum value of the Fourier amplitude spectrum of the compression noise for all three components of the example accelerogram are summarized in Table 2.

FIG. 2. Data compression noise for algorithms DC1–DC4. Coalinga aftershock of 8 May 1983; N35E component. (a) DC1, (b) DC2, (c) DC3 and DC4.

By using the cosine interpolation decompression scheme, DC2, instead of the straight line scheme, DC1, it is seen that the errors in the frequency domain are somewhat reduced in the regions of strong frequency content. However, little improvement is obtained in the time domain over simple linear interpolation.

The other higher-order interpolation schemes, DC3 and DC4, do not work much better than cosine interpolation. The third-order polynomial interpolation employed in DC3 gives results which are only slightly improved over those of DC2. The seventh-order polynomial interpolation employed in DC4 represents an attempt to

TABLE 2
COMPARISON OF DIFFERENT COMPRESSION/DECOMPRESSION SCHEMES

$\epsilon_1(g)$ $\epsilon_2(g)$	DCR Three Channels	Maximum Acceleration Compression Noise (g)	R.M.S. Acceleration Compression Noise (g)	Maximum Fourier Amplitude Spectrum of Acceleration Noise (g-sec)	Maximum Velocity Error (cm/sec)	Displacement Error. (t = 10 sec) (cm)
DC1	2.3	(L) 0.055	0.002	0.006	1.4	2.7
		(T) 0.027	0.002	0.010	3.4	6.6
		(V) 0.027	0.002	0.005	2.7	7.0
DC2	2.3	0.043	0.002	0.005	1.3	2.6
		0.021	0.002	0.005	3.4	6.7
		0.023	0.002	0.004	2.7	7.0
DC3	2.3	0.044	0.002	0.005	1.3	2.7
		0.022	0.002	0.005	3.4	6.7
		0.023	0.002	0.004	2.7	7.0
DC4	2.3	0.044	0.002	0.005	1.4	2.7
		0.022	0.002	0.005	3.4	6.7
		0.023	0.002	0.004	2.7	7.0
DC5	1.5	0.001	0.0003	0.001	0.32	0.06
		0.001	0.0003	0.001	0.23	0.02
		0.001	0.0003	0.001	0.20	0.02
DC6	5.2	0.001	0.0003	0.001	0.32	0.06
		0.001	0.0003	0.001	0.23	0.02
		0.000	0.001	0.001	0.18	0.05
DC6	4.2	0.001	0.0002	0.001	0.14	0.02
		0.001	0.0002	0.001	0.17	0.02
		0.001	0.001	0.004	0.77	1.85
DC6	8.1	0.008	0.001	0.003	2.90	2.70
		0.008	0.001	0.004	2.20	2.28
		0.008	0.002	0.006	1.23	1.66
DC6	10.7	0.008	0.002	0.007	4.07	5.97
		0.008	0.002	0.005	3.55	2.01

Compression scheme: DC1 = recording only peaks; DC2 = recording only peaks; DC3 = recording only peaks; DC4 = recording only peaks; DC5 = BSA; DC6 = BSA + DBR.
Decompression scheme: DC1 = linear interpolation; DC2 = cosine interpolation; DC3 = 3rd order polynomial interpolation; DC4 = 7th order polynomial interpolation; DC5 = linear interpolation; DC6 = linear interpolation.

help take into account the presence of two closely spaced neighboring peaks, but it gives results which are essentially indistinguishable from those of DC3.

In summary, it is concluded that a data compression/decompression algorithm based on the recording of peak data only results in compression errors that are unacceptably large and that are nonuniform in both the time domain and the frequency domain. The maximum errors generally occur during the strong shaking phase of the accelerograms meaning that significant information may be lost through data compression. More sophisticated interpolation between peaks can reduce the maximum errors in the frequency domain somewhat but appears to have less effect on the time domain errors. A major weakness of the peak-only data compression/decompression algorithms is that the maximum error cannot be specified in either the time domain or the frequency domain.

A BACKWARD SUCCESSIVE APPROXIMATION (BSA) COMPRESSION SCHEME

In view of the clear shortcomings of peak-only based compression schemes and in order to provide a more acceptable approach, consideration is given to alternative

FIG. 3. Fourier Spectrum. Coalinga aftershock of 8 May 1983; N35E component. (a) Original accelerogram, (b) data compression noise for DC1, (c) data compression noise for DC2, (d) data compression noise for DC3 and DC4.

schemes of data compression/decompression. The objective is to develop an algorithm which is simple enough to be easily implemented in a microprocessor and is also capable of retaining the required accuracy in the recorded accelerograms.

The approach proposed is conceptually quite straightforward, being the analog of simple procedures that might be employed in manual digitization of analog accelerograms. First, distinct peaks of the accelerogram are identified. Then, a straight line approximation is visualized between adjacent distinct peaks and the accuracy of this approximation judged against the actual data. Finally, additional data is stored only as necessary to follow any features of the wave form which are not adequately represented by the straight line interpolation between peaks or other important features previously identified.

The algorithm may be described mathematically as follows: Let a_i denote the numerical value of the acceleration which is measured during the i th sampling of the data, and let $\hat{a}_i$ denote any local extremum of the data as determined from the condition

$$a_i = \hat{a}_i \quad \text{if } (a_i - a_{i-1})(a_{i+1} - a_i) < 0 \quad (1)$$

Let a *distinct peak* be defined as a local extremum of the data which differs from the immediately adjacent extrema by more than some prescribed amount, ϵ_1 . Then, if $\hat{a}_m$ is a distinct peak, the next distinct peak, $\hat{a}_n$, will be the first extremum such that

$$|\hat{a}_n - \hat{a}_m| > \epsilon_1; \quad n = \text{min. integer} > m \quad (2)$$

The first distinct peak may be taken as the initial data point of the record or the first extremum encountered.

When a distinct peak, $\hat{a}_m$, is detected, the time and acceleration data associated with that sample is permanently recorded. Then, the subsequent data is stored in a temporary buffer memory until the next distinct peak, $\hat{a}_n$, is encountered. When $\hat{a}_n$ is detected, the $n - m$ samples of acceleration data in the temporary memory are examined to determine just which subset of the data must be permanently recorded in order to achieve a prescribed accuracy of the compressed data between the two distinct peaks. The parameter ϵ_1 allows the algorithm to effectively ignore very small fluctuations in the accelerogram, such as those which occur from background noise during the pre-event phase. This parameter also influences the size of the buffer memory required and the compression processing time.

Given any two data points r and s , a linear approximation, $\tilde{a}_i(r, s)$, to some data point, a_i , which lies between these two points may be constructed as

$$\tilde{a}_i(r, s) = a_r[1 - (i - r)/(s - r)] + a_s(i - r)/(s - r); \quad r < i < s \quad (3)$$

Then, the difference between the actual acceleration and the linear approximation will be

$$\delta_i(r, s) = |a_i - \tilde{a}_i(r, s)|; \quad r < i < s \quad (4)$$

where, by definition $\delta_r(r, s) = \delta_s(r, s) = 0$. Let the data point k be defined such that

$$\delta_k(r, s) = \text{Max}_{i=r,s} \delta_i(r, s) \quad (5)$$

Then, the point k corresponds to the point of maximum difference between the actual and approximate acceleration between the points r and s .

The following sequence of steps may be performed to determine a minimum set of data points which must be retained between adjacent distinct peaks so as to have a difference between the actual data and linear approximation which is nowhere greater than a prescribed value ϵ_2 .

1. Set $r = m$ and $s = n$.
2. Compute $\delta_k(r, s)$ from equation (5).
3. If $\delta_k(r, s) > \epsilon_2$, then set $s = k$ and go to step 2.
4. If $\delta_k(r, s) \leq \epsilon_2$, then store data for point s .
 - a. If $s \neq n$, then set $r = s$ and $s = n$ and go to step 2.
 - b. If $s = n$; done, go to next set of distinct peaks.

Several variations of the above scheme can be envisioned, such as storing the data associated with each a_k as determined, then minimizing the difference on the interval between these points, and finally reordering the stored data. However, the scheme outlined above gives the desired error control and is quite efficient computationally.

The compression algorithm is illustrated for a typical set of acceleration data in Figure 4. In this example, there are adjacent distinct peaks corresponding to samples $i = m = 80$ and $i = n = 89$. The straight lines, $\tilde{a}(r, s)$, are the continuous

FIG. 4. Implementation of backward-successive-approximation compression scheme.

representations of the linear approximations $\bar{a}_i(r, s)$. In Figure 4a, the initial sequence of backward approximation is shown arriving at the first permanently stored data point, $i = 82$. Figure 4b shows the results of the completed approximation between the adjacent peaks. A total of two intermediate data points ($i = 82, 86$) must be stored in this case to meet the error criterion established. It is seen that these two points are just the points required to describe the major features of the waveform between the distinct peaks.

Based on the compression scheme described above, the accuracy and data compression ratio are determined primarily by the parameter ϵ_2 . However, ϵ_1 also influences the compression ratio and accuracy, especially for the pre-event portion of the record. For example, if a large enough value of ϵ_1 is specified, then most of the background noise in the pre-event data can be ignored by the compression scheme. More pre-event data can be preserved if a smaller ϵ_1 is specified. In practice, some combination of ϵ_1 and ϵ_2 must be chosen, depending on the compression ratio and accuracy desired in the particular application. Note that the algorithm is only

effective in eliminating pre-event noise if $\epsilon_2 \geq \epsilon_1$. If $\epsilon_2 < \epsilon_1$, once the second distinct peak is encountered, the algorithm will demand that many points between the first and second peak be stored.

In order to examine the accuracy of the backward successive approximation (BSA) data compression algorithm, this scheme is applied to the same data as the previously examined schemes. The results (DC5 and DC6) are shown in Figure 5. The values of ϵ_1 and ϵ_2 for this case have both been taken to be 0.001 g. It is clear from the figure that the compression noise nearly vanishes during the strong shaking phase and that the maximum value is indeed everywhere less than 0.001 g. Furthermore, the error in the frequency domain is also quite small. The data compression noise for all three components of the example accelerogram is summarized in Table 2. It is noted that it has been necessary to sacrifice some of the gain achieved in data compression ratio (DCR) by peak-only compression for the improvement obtained in accuracy. For the error bound specified, the resulting DCR is 1.5, which corresponds to a saving of one-third in storage required over the uncompressed case. This is a significant reduction, but a much greater reduction can be achieved if other steps are taken along with BSA.

DIFFERENTIAL BIT REDUCTION (DBR)

Improvement of the BSA scheme may be achieved with a simple bit reduction technique, differential bit reduction (DBR). It is found that fewer bits of memory are required if only the difference in values between the current and previously recorded data are stored instead of full values.

FIG. 5. Data compression noise for algorithms DC5 with $\epsilon_1 = \epsilon_2 = 0.001$ g. Coalinga aftershock of 8 May 1983; N35E component. (a) Time history, (b) Fourier Amplitude Spectrum. Note differences in scales from Figures 2 and 3.

In this case, the DCR will be given by

$$\text{DCR} = \frac{3 \times N_0 \times B_0}{(C_1 + C_2 + C_3) \times (B_3 + B_4)},$$

where

C_1 = Number of compressed data samples in longitudinal channel

C_2 = Number of compressed data samples in vertical channel

C_3 = Number of compressed data samples in transverse channel

B_3 = Number of bits taken to record time difference

B_4 = Number of bits taken to record acceleration difference.

It should be noted that the word lengths specified by B_3 and B_4 are set initially by the user. This approach saves the overhead needed in designating word boundaries for variable length words. Once the preset word boundaries are reached, the BSA algorithm automatically stores additional data points as necessary to satisfy the prescribed word size. The higher the frequency content of the acceleration, the greater will be the possible change in acceleration from one time step to the next. However, this difference can be reduced by increasing the sampling rate. Thus, there is a relationship between the word sizes needed for B_3 and B_4 . Additional studies of typical strong-motion accelerograms would be required to select the optimum values of these parameters.

For the case of DC5 using DBR with $\epsilon_1 = \epsilon_2 = 0.001 g$, B_3 and B_4 may be taken to be 3 and 4, respectively. Then

$$\text{DCR} = \frac{3 \times 4096 \times 12}{(1391 + 1289 + 1336) \times (3 + 4)} = 5.2.$$

The combination of BSA and DBR is referred to as DC6, and full results using this scheme are given in Table 2. Also shown are results for various combinations of ϵ_1 and ϵ_2 . The parameter ϵ_2 clearly has the greatest influence on both the compression noise and compression ratio. As may be expected, the larger the acceptable error, the higher will be the data compression ratio obtained. Hence, one must in general strike a compromise between accuracy and storage requirements. No attempt has been made in this study to determine the optimal values of ϵ_1 and ϵ_2 .

For the last two entries in the table, B_3 and B_4 were taken as 4 and 7, respectively. If these values would have been used in the first two examples of DC6, the DCR would change from 5.2 to 8.1 and from 4.2 to 10.7.

It will be observed that the use of DBR in conjunction with the BSA scheme provides a greater increase in data compression ratio than does the use of BSA alone. Thus, DBR is a very important component in the overall data compression methodology. It should be noted, however, that DBR by itself would not be capable of achieving such high compression ratios as those reported. The two schemes must be used together to achieve the DCR levels indicated. DBA provides a rational means of selecting the data to be stored within a fixed word length format so that the inherent capabilities of DBR can be fully exploited.

INTEGRATED COMPRESSION NOISE

In many applications, the time histories of velocity and displacement are also important. These can be derived by direct integration of the acceleration time history. However, the integration process is very sensitive to any long period noise present in the recorded acceleration data. The long period errors in the acceleration

FIG. 6. Integrated data compression noise for algorithm DC1. Coalunga aftershock of 8 May 1983; N35E component. (a) Original velocity time history, (b) velocity noise, (c) original displacement time history, (d) displacement noise.

FIG. 7. Integrated data compression noise for algorithm DC6 with $\epsilon_1 = \epsilon_2 = 0.001$ g. Coalinga aftershock of 8 May 1983; N35E component. (a) Velocity noise, (b) displacement noise.

time history resulting from data compression can therefore be expected to affect the accuracy of the velocity and displacement obtained by integration. In order to examine the general nature of the errors associated with integration, the compression noise time history may be integrated directly to obtain the error time histories of velocity and displacement resulting from data compression. These error time histories are referred to as integrated compression noise.

The integrated compression noise associated with the algorithm DC1 is shown in Figure 6 along with the actual velocity and displacement time histories. Clearly, the integration errors due to this algorithm are unacceptably large. This is typical for all four data compression/decompression algorithms based on peaks only. The results are summarized in Table 2. Figure 7 shows the integrated compression noise associated with the BSA data compression/decompression algorithm. Compared with the results of DC1, the integrated compression noise is greatly reduced and is considered acceptable during the strong shaking phase of the motion of this relatively short duration event. If the peak displacement is achieved during the strong-motion portion of the accelerogram, it is believed that the proposed data compression scheme will not introduce any overly large errors. However, more study is required to verify this assertion.

Filtering is often employed to reduce the effect of long period errors in analog accelerograms, and this approach could also be employed here to minimize corresponding errors in the compressed digital data if desired. However, care must be exercised in the interpretation of such filtered data.

CONCLUSIONS

Based on the results presented herein, it is concluded that extreme caution must be exercised in attempting compression of strong-motion data. However, the BSA data compression algorithm presented, or some variation thereof, appears to be capable of providing significant data compression ratios while maintaining control of the compression noise. The availability of such data compression algorithms should enable future digital strong-motion recorders to provide greater flexibility between accuracy and capacity.

ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation and in part by Kinemetric Systems, Inc., whose assistance is greatly appreciated by the authors. The opinions expressed are those of the authors and do not necessarily reflect those of the supporting agencies or organizations.

REFERENCES

- Crochiere, R. E. and L. R. Rabiner (1981). Interpolation and decimation of digital signals: A tutorial review, *Proc. IEEE* **69**, 330–331.
- Hudson, D. E. (1979). *Reading and Interpreting Strong Motion Accelerograms*, Earthquake Engineering Research Institute, Berkeley, California.
- Iwan, W. D., M. A. Moser, and C. Y. Peng (1985). Some observations on strong-motion earthquake measurement using a digital accelerograph, *Bull. Seism. Soc. Am.* **75**, 1225–1246.
- Makhoul, J. (1975). Linear prediction: A tutorial review, *Proc. IEEE* **63**, 561–580.
- Trifunac, M. D. and V. Lee (1973). Routine Computer Processing of Strong-Motion Accelerograms, Earthquake Engineering Research Laboratory Report No. 73-03, California Institute of Technology, Pasadena, California.

CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA 91125

Manuscript received 24 April 1987