Bulletin of the Seismological Society of America, Vol. 68, No. 5, pp. 1521-1532, October 1978

AUTOMATIC EARTHQUAKE RECOGNITION AND TIMING FROM SINGLE TRACES

By Rex V. Allen

ABSTRACT

A computer program has been developed for the automatic detection and timing of earthquakes on a single seismic trace. The program operates on line and is sufficiently simple that it is expected to work in inexpensive low-power microprocessors in field applications. In tests with analog tapes of earthquakes, the program correctly identified and timed to within 0.05 sec about 70 per cent of the events which would normally be timed in operation of a network. The program evaluates the accuracy of its picks, and its estimates appear to be quite reliable. The algorithm is working at present in a 16-bit minicomputer and appears to be compatible with presently available microprocessors.

Introduction

A new computer program for the automatic processing of seismic data has been developed and tested by the U.S. Geological Survey.

The goal has been to produce a computer program that can duplicate the performance of a human operator in processing P arrivals of single seismometer traces. It must recognize earthquakes, time them to an accuracy of better than 0.05 sec, determine amplitudes, direction of first motion and a few other descriptive parameters, and formulate an estimate of how reliable the pick time is to be considered. To be at all useful, the program must be capable of distinguishing earthquakes from noise sources such as wind or vehicular traffic, and it must be able to operate with small weak earthquakes such as are commonly used in routine operation of a location network such as the USGS Central California Network.

Contraints on the program are simple, but rather stringent. It must be suitable for eventual real-time operation in an inexpensive low-power microprocessor at remote field locations. This constraint means that the program may not "look back" in time, and it may use only a limited amount of memory both for program instructions and for any scratch-pad storage used during operation. By not "looking back", I mean that each sample is processed sequentially, and the program is not allowed to access earlier points unless it has stored them in a separate scratchpad array.

This is an extremely rigorous set of requirements, and in many cases not all the constraints will be required. In some cases for instance, the algorithm will merely be required to recognize an earthquake for recording by an event recorder. In this case precise timing and most measurements will be made later by a human operator. In another possible application a number of processors might be linked together in processing telemetered data at a central collection point for a real-time earthquake locator. In this use it would be possible to cross-reference results between stations and thus simplify the recognition problem.

The algorithm has been developed and tested under the most rigorous constraints and if some can be relaxed its performance should be even better.

Microprocessors have come into use only in the past few years, and are still evolving so rapidly that it is difficult to stay abreast of what is available. Soucek

Any use of trade names or trademarks in this publication is for descriptive purposes only and does not constitute endorsement by the U.S. Geological Survey.

(1976) reviews their capabilities and uses and discusses some of the representative processors currently available. Program development for the system described here was carried out partly on CDC 6600 and 7600 computers and partly on the USGS CDC 1700 16-bit computer in the Menlo Park office.

When operating on the CDC 6600 and CDC 7600 machines, the program processes previously digitized data sets up to 45 sec in length, digitized at 200 samples/sec. In this processing, the restriction of not "looking back" is strictly observed: In this fashion, a rather good simulation of on-line operation can be achieved.

Most of the illustrations in this paper are of these digital data sets, because it is possible to store and later plot detailed time histories of the program operation, including operational parameters. Most of the program development and testing, however, was carried out on the CDC 1700 computer. Since this machine can digitize and operate with analog input signals, it is possible to test a program with long runs of analog tape data sets or to operate in real time by monitoring selected telemetry channels of the USGS Central California Network. This ability to use analog data sets is absolutely necessary in developing an algorithm that is resistant to noise sources. Digital records are unsatisfactory because seismic report libraries contain earthquake records, not noise, and the real test of an automatic picking algorithm is its ability to reject strong noise bursts while still recognizing weak earthquakes.

An illustration of this program's effectiveness is provided in Figure 1, which plots ten arrivals from a magnitude 2.8 earthquake at distances ranging from 50 to 90 km illustrating the accuracy which may be expected.

Program Operation

In the following discussion of program operation, I shall use the term "event" to mean a section of a seismic record which is noticeably different in "character" from the immediately preceding record. An earthquake will be an event in this sense, as will wind or culturally produced noise bursts. The "character" of a trace is described by a specific function of trace amplitude and first time difference. If the seismic trace is regarded as a time series $f(t)$ with first difference $f'(t)$, then the characteristic function $E(t)$ is defined as

$$E(t) = f(t)^2 + C_2 + f'(t)^2.$$

C_2 is a weighting constant whose purpose is to vary the relative weight assigned the amplitude and first difference depending on digital sample rate and noise characteristics of the individual seismic station. This particular characteristic function was selected because it incorporates quantities descriptive of both amplitude and frequency content of the seismic trace, varies rapidly with changes in either of these parameters, is always positive, and is easy to compute.

The restriction that the program may not "look back" means that an event must be recognized immediately and its arrival time recorded. Anyone who has spent some time looking at seismic records, however, will recognize that most events in the sense defined above will turn out not to be earthquakes. The program must therefore be able to determine very quickly that a spurious event is not an earthquake and return to monitoring the trace. If the event lasts fairly long, for example, a passing vehicle or such, the program must store sufficient information during the event to enable it to decide at the end whether the event was an earthquake and worth recording, or merely noise that should be ignored.

In understanding the logical operations followed by the program, it is easiest to refer to the two flow charts of Figures 2 and 3 and the parameter plot in Figure 4.

Figure 2 is a conventional flow chart and most clearly illustrates the program flow with the various loops and decision points. Figure 3 is a chart of the form suggested by Stewart (1973), and, by including details of transfer functions and some narrative

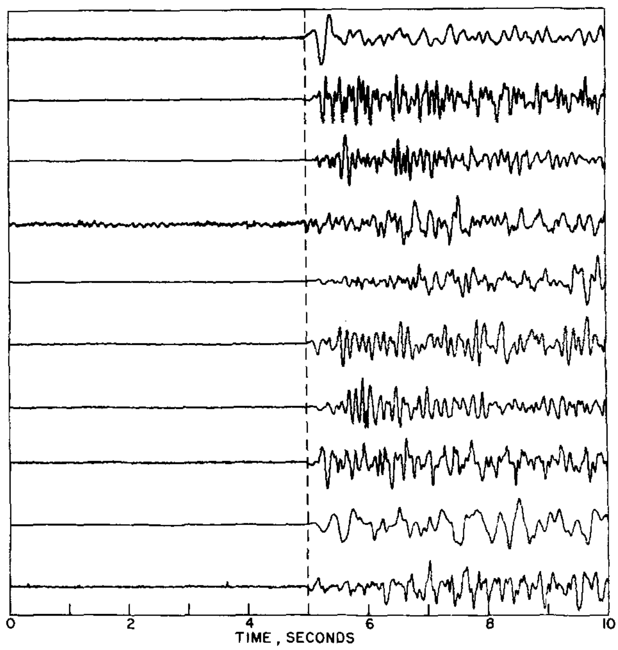

FIG. 1. Program P-arrival picks for magnitude 2.8 earthquake. Dotted vertical line at center of trace is pick point for each trace. Epicentral distances range from 50 to 90 km.

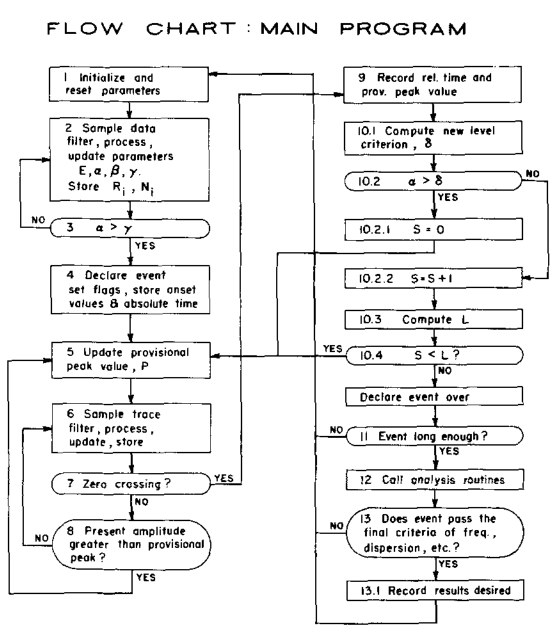

FIG. 2. Program flow chart.

description, it is easier to follow the detailed operation of the main program. The box numbers of Figure 2 correspond to statement numbers of Figure 3 for easy cross reference. Figure 4 is a time history of some of the operating parameters used by the program during a typical event.

Main Program Flow

1) Initialize and reset operating parameters and flags.
2) Input digital data, N_i $i = 1, 2, \ldots$..Time index.
   .1) Store present value of Real data $R_{i-1} \leftarrow R_i$
   .2) Filter to remove DC offset, producing updated value of Real data. $R_i = C_1 * R_{i-1} + (N_i - N_{i-1})$
   .3) Calculate weighted derivative $\Delta R_i = C_2 * (N_i - N_{i-1})$
   .4) Store present value of digital data $N_{i-1} \leftarrow N_i$
   .5) Compute characteristic function .... $E_i = R_i^2 + \Delta R_i^2$
   .6) Compute short-term average of E
   $\alpha_i = \alpha_{i-1} + C_3 * (E_i - \alpha_{i-1})$ $.2 < C_3 < .8$
   .7) Compute long-term average of E
   $\beta_i = \beta_{i-1} + C_4\ (E_i - \beta_{j-1})$ $.005 < C_4 < .05$
   .8) Compute reference level $\gamma_i = C_5 * \beta_i$ $C_5 \approx 5.$
3) Decision: Has the short-term average abruptly increased with respect to the long term? $\alpha_i > \gamma_i$?
   .1t) Go to 4
   .2f) Go to 2
4) Declare event and store onset values
   .1) Store absolute time $T_o \leftarrow i$
   .2) Store trace amplitude $A_o \leftarrow R_i$
   .3) Store derivative $D \leftarrow (R_i - R_{i-1})$
   .4) Store noise level, β.
5) Store provisional peak value $P \leftarrow |R_i|$
6) Repeat 2.1 → 2.6.
7) Decision: Check for zero crossing of trace. Zero crossing?
   .1t) Go to 9
   .2f) Go to 8
8) Decision: Is the trace amplitude increasing? $|R_i| > P$?
   .1t) Go to 5
   .2f) Go to 6
9) Store zero-crossing time and previous peak amplitude.
   .1) $T_M\ (i - T_o)$ $M = 1, 2, \ldots\ldots 500.$ Index of peaks in coda.
   .2) $A_M \leftarrow P$
   .3) Increment $M \leftarrow M + 1$
10) Determine whether event should be declared over.
   .1) Compute criterion $\delta = f\ (\beta, M)$
   .2) Decision: Is the value of α above the continuation criterion δ at this zero crossing? $\alpha_i > \delta$?
      .1t) Reset smallcount counter to zero. This counter keeps count of the number of successive zero-crossings which have occurred with a value of α below the criterion level δ.
      $S \leftarrow 0$ Go to 5
      .2f) Increment smallcount counter $S \leftarrow S + 1$
   .3) Compute L, the value of S at which the event is declared over.
   $L = f\ (M)$
   .4) Decision: Should event be declared over? $S < L$?
      .1t) Event not over   Go to 5
      .2f) Event over   Go to 11
11) Decision: Was event long enough to qualify?
   $(i - T_o) > K_1$ and $M > K_2$?
      .1t) Go to 12
      .2f) Go to 1
12) Call analysis routines (Frequency, Dispersion, Weight, Coda Shape, etc.)
13) Decision: Does event pass analysis criteria?
      .1t) Record results, Go to 1
      .2f) Go to 1

FIG. 3. Program flow chart.

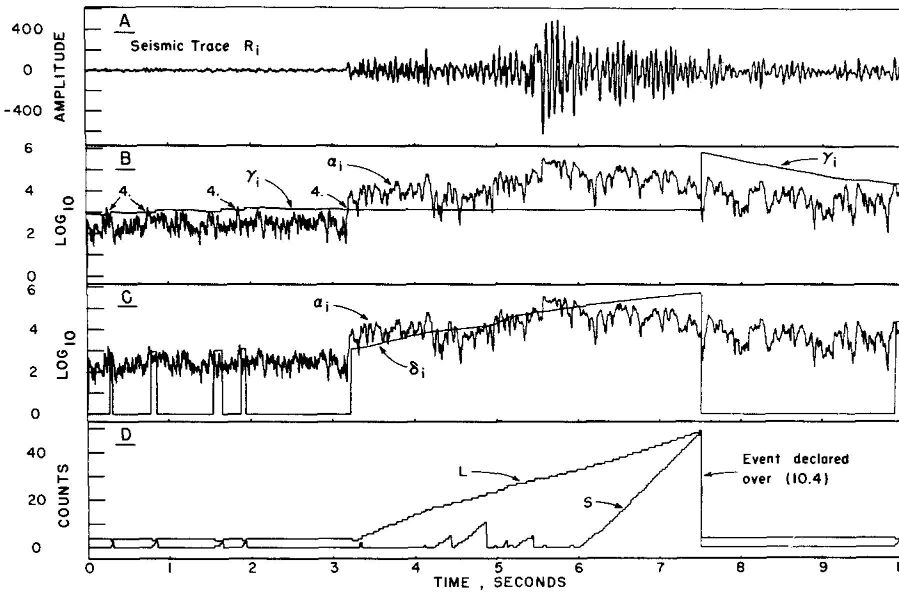

FIG. 4. Operating parameters during an event. (a) Seismic trace. (b) Short-term average α_i, of characteristic function and reference level, γ_i. Points indicated as "4" refer to statement 4 in program flow charts. Note false onsets before arrival of real event. Amplitude scale is logarithmic. (c) Short-term average α_i, and the criteria level δ, carried during observation of an event. Logarithmic amplitude scale. (d) Quantities L and S, which determine point at which event is declared over.

The following description will reference principally Figure 3, and I will attempt to discuss the program in sufficient detail that an interested reader can code the algorithm for any available computer. Paragraph numbers refer to the corresponding operation in Figure 3. Parameters mentioned in the discussion are defined in the Figure 3 flow chart. Table 1 is an alphabetized parameter list which will aid in locating the definitions in Figure 3.

PROGRAM DESCRIPTION

1. This step is principally concerned with resetting M, the peak index number, the event relative time count, and, in some versions of the program, the quantity α, discussed below.
2. In this sequence the seismic trace is sampled, digitized to produce the 12-bit integer value N_i at time i, and the integer value is processed to remove any D.C. bias in the digital data. The resulting value, R_i, is a floating-point number in the present program version, with the constant Cl controlling the time constant of the high-pass filter function.

   2.3 The weighted difference ΔR_i is computed.

   2.4 The present value of N_i is stored for use with the next sample.

   2.5 The characteristic function E is computed as discussed earlier.

The program now computes a short-term (~0.01 sec) average α_i and a long-term (~2 sec) average β_i of the quantity E_i and multiplies β_i by threshold constant C5 to produce the reference level γ_i.

3. If $\alpha_i > \gamma_i$, an event is declared (Figure 4).

The procedure of comparing short- and long-term averages is similar to the picking algorithms described by Ambuter and Soloman (1974), and Stevenson (1976), with the rather important exception that the averages are carried out on the characteristic function rather that on seismic trace amplitudes. This procedure is

TABLE 1

PARAMETER LIST AND DEFINITION DIRECTORY

Parameter	Statement*	Parameter	Statement
A	4.2, 9.2	P	5.
C1	2.2	R	2.2
C2	2.6	ΔR	2.3
C3	2.7	S	10.2.1
D	4.3	T	4.1, 9.1
E	2.5	α	2.6
i	2.	β	2.7
L	10.3	γ	2.8
M	9.1	δ	10.1
N	2.		

* Statement numbers refer to Figure 3.

inherently more powerful, since the quantity E is sensitive to changes in both amplitude and frequency and varies as the sum of the squares of these two parameters. Stewart (1977) makes use of a characteristic function that is a modified first difference term. The function described here is more powerful but could not be used in an application such as Stewart's in which 100 or more traces are to be monitored by one processor.

4. When an event is declared, the absolute time is recorded along with the present values of trace amplitude, derivitive, and β. β, the long-term average of E, will be a measure of the background noise level at the time of detection and will be used later in assigning a reliability weight to the pick.
5. The program now enters a pair of nested loops in which it searches for a peak amplitude and the subsequent zero crossing. When the zero crossing is detected, the program breaks out of the loop, records the zero-crossing time relative to event onset and the signed amplitude of the preceding peak. The time and amplitude information is stored in a pair of scratchpad arrays for later use by analysis routines.

10.0 and 10.2 At each zero crossing of R_i, the present value of α is compared with a continuation criterion level δ. If $\alpha < \delta$, the smallcount register, S, is incremented by 1. If $\alpha > \delta$, S is reset to 0. Therefore, S always contains the number of consecutive zero crossings that have occurred at which α is less than the continuation criterion level.

The criterion, δ, must start at a level comparable to the reference level γ at the beginning of an event, and it must increase during an event in some fashion to ensure that the event is terminated even if the noise level should abruptly increase and remain indefinitely at the new high level, as during a very sudden windstorm, for example.

Furthermore, experience with earthquake swarms and aftershock sequences indicates that the observation loop during an event should be terminated as rapidly as possible (well before the end of the coda) in order to enable the program to resume searching for subsequent earthquakes partially buried in the coda of preceding events.

I tried several methods in arriving at the present algorithm for computing δ, including linear and parabolic increase with time and with M, the number of observed peaks in the event. The most successful version is a two-segment parabolic increase with M, in which the increase is made much steeper after $M = 60$. This

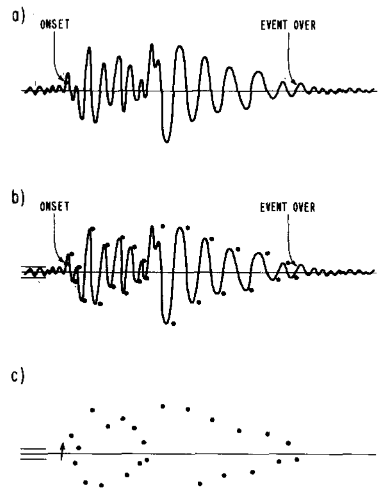

FIG. 5. Schematic earthquake with data stored during events. (a) Seismic event with onset and end points. (b) Dots represent zero-crossing times and previous peak amplitudes. Amplitude bars indicate background noise preceding onset, and arrow gives first difference at onset. (c) Data stored for use by analysis routines after event terminates.

scheme seems successful in terminating observations as soon as possible for small, very close earthquakes without losing them entirely, and also in enabling the program to continue observation and recording of more distant events with typically longer period waves. In applications where measurement of coda length is very important, this method would obviously be unsatisfactory.

10.3 An event will be declared over when some number of zero crossings with $\alpha > \delta$ have occurred. This termination number, L, will be small, typically 3, at the start of an event to enable the program quickly to reject noise spikes or other very short-term interference. When the algorithm is well into a larger event, however, L must be considerably larger to ensure that an earthquake observation is not terminated too early during a quiet period between phase arrivals. The present version of the program uses the relation

$$L = 3 + M/3$$

where M is the current number of observed peaks in an event.

10.4 This decision of whether the event is over is simply a comparsion of S and L, with branches to continue the program in the observation loop or to terminate observation of the event as required.

11. This is an elementary test to throw out very short events such as noise spikes. Typically, the criteria for further consideration are that the event must have been longer than 1.5 sec and have recorded more than 40 peaks. During normal operation, the progam will be producing false onsets about once per minute, and this test returns it quickly to the search loop without wasting time on analysis or using up recording space.

12. The analysis routines are a set of subroutines that examine the data recorded during the event and produce quantitative descriptors of the event. They are discussed later.

13. After the program calls the analysis routines, their results may be examined to determine whether the event was probably an earthquake and worth

noting, or merely a long noise event which can be ignored. If it was cataloged as an earthquake, the required descriptors are recorded. The program then returns to (1) and reenters the search loop.

When an event has been declared over, the program has stored a description of the event with which the analysis routines can work. The information available in this data set is

1. B, a measure of the noise level at event detection.
2. A_0, the trace amplitude at detection.
3. D, the trace first difference at detection.
4. The array A containing the peak amplitude of each peak in the observed wave train.
5. The array T containing the time of each zero crossing relative to event onset.

Figure 5 illustrates these quantities for a schematic earthquake.

Analysis Subroutines

I have tried a larger number of analysis techniques to examine this data set and complete description of even those I currently use would be too lengthy to include. Instead, I will briefly describe a few of the key routines to illustrate what can be done with available stored data.

A key subroutine IPPUT (Peaks Per Unit Time) divides the observed coda into half-second intervals and counts the number of peaks in each interval, storing the number in a short scratchpad array. It is then easy to compute the mean frequency of the seismic signal and measure the variation in frequency during the event by computing the mean and standard deviation of the array. Cultural noise such as passing vehicles will typically be quite monochromatic compared to earthquakes.

Another routine quantifies the shape of the observed coda in terms of relative steepness of onsets and decays in the envelope shape. The routine selects the largest peak in each half-second interval and compares it with the largest in the following interval. The difference in amplitude is squared, and if the envelope is increasing in amplitude, the square is added to the variable "UP"; if the envelope is decreasing, it is added to "DOWN". When the entire coda has been processed, the number SLOPE is computed as

$$\text{SLOPE} = \text{SQRT (UP/DOWN)}.$$

Earthquakes with characteristic steep onsets and slow decays tend to produce values of SLOPE of 1 or greater, whereas noise bursts tend to have values of 1 or less. The SLOPE value in itself is not a sure discriminator of earthquakes, and especially tends to break down for longer events.

Subroutine WEIGHT is used to determine the weight, or measure of reliability to be placed on the P-arrival pick. It makes use of β, a measure of background noise level at event onset, D, the first difference at observed onset, and peak amplitudes of the first 3 peaks. The exact form of the algorithm is still being changed, but, as an example, to be weighted "0", the highest quality, an event must meet the following criteria

(1) $D > \sqrt{B}$
(2) $A_1 > 450$
(3) $A_1/\sqrt{B} > 4$
(4) $A_2 > 6\sqrt{B}$ or $A_3 > 6$.

These criteria are successively relaxed for weights of 1, 2, and 3.

Magnitude determination is an unsolved problem. In principle, the magnitude of a local earthquake can be estimated from a single seismogram by using the coda length (Lee *et al.,* 1972). This determination cannot be made by this program in its present version because of the decision discussed earlier to terminate observation of an event as soon as possible. As a result, most of the coda is lost, especially in larger events, and the reported duration is artificially shortened. Several alternative methods have been tried including variations on the scheme described by DeNoyer (1959). None has been entirely satisfactory, and this problem will require further investigation.

Performance Evaluation

Processing of digital data sets is useful in developing algorithms and for detailed examination of program behavior, but for evaluation of program performance, there is no substitute for the processing of analog signals. Only in this way is it possible to operate a system in real time or to simulate real time operation by the use of analog tapes.

This program was tested on the USGS CDC 1700 computer using analog tapes as input. The 1700 system incorporates an analog-digital converter (ADC) for input and a Siemens multi-channel strip chart recorder for monitoring the input signal and program operation. A line printer is available for output of program results.

In a typical test run, the computer monitors an analog signal from a tape recorded during an aftershock sequence or an earthquake swarm. The tape will carry records from many telemetered seismic stations, plus two serial timing channels: WWVB and locally generated IRIG C. During test runs, the computer generates an event flag that is set high at the onset of an event and then reset low when the event is declared over. This flag output is displayed on the strip chart recorder next to the seismic trace selected for monitoring and provides an easily read indication of program status. The false-trigger rate can easily be observed and the program's accuracy in picking *P* arrivals can be checked.

If an event is observed which passes the length criteria (item 11 of Figure 3), the program prints out on the line printer the time of the arrival to the nearest second, as read from a time code translator, first motion, and results of the analysis subroutines.

The strip chart record is run at 25 mm/sec so it is possible to judge timing quite accurately by observing the program status flag. Detected first motion, weight, and other relevant parameters required for program evaluation are available in the line-printer output. Figure 6 shows a section of strip chart record for a typical small earthquake with the line printer output superimposed.

A test run normally consists of half an hour to an hour of an analog tape. After the run is completed, the strip chart and line printer output are given to one of the people who normally scan records of the USGS seismic net for scoring.

To be scored a "Hit", an event must be picked with proper first motion and with a timing error of less than 0.05 sec. A "miss" is scored for an event that normally would be picked in a net operation and that the program missed entirely. In addition, various forms of errors are tabulated, such as "1M", indicating a wrong first motion but tolerable timing; "T", indicating a timing error greater than 0.05 sec; or "T?", indicating a questionable but not definitely wrong time in the case of emergent arrivals. The score of the run is taken to be the number of hits divided by the total number of earthquakes and generally runs about 60 to 80 per cent. It should be

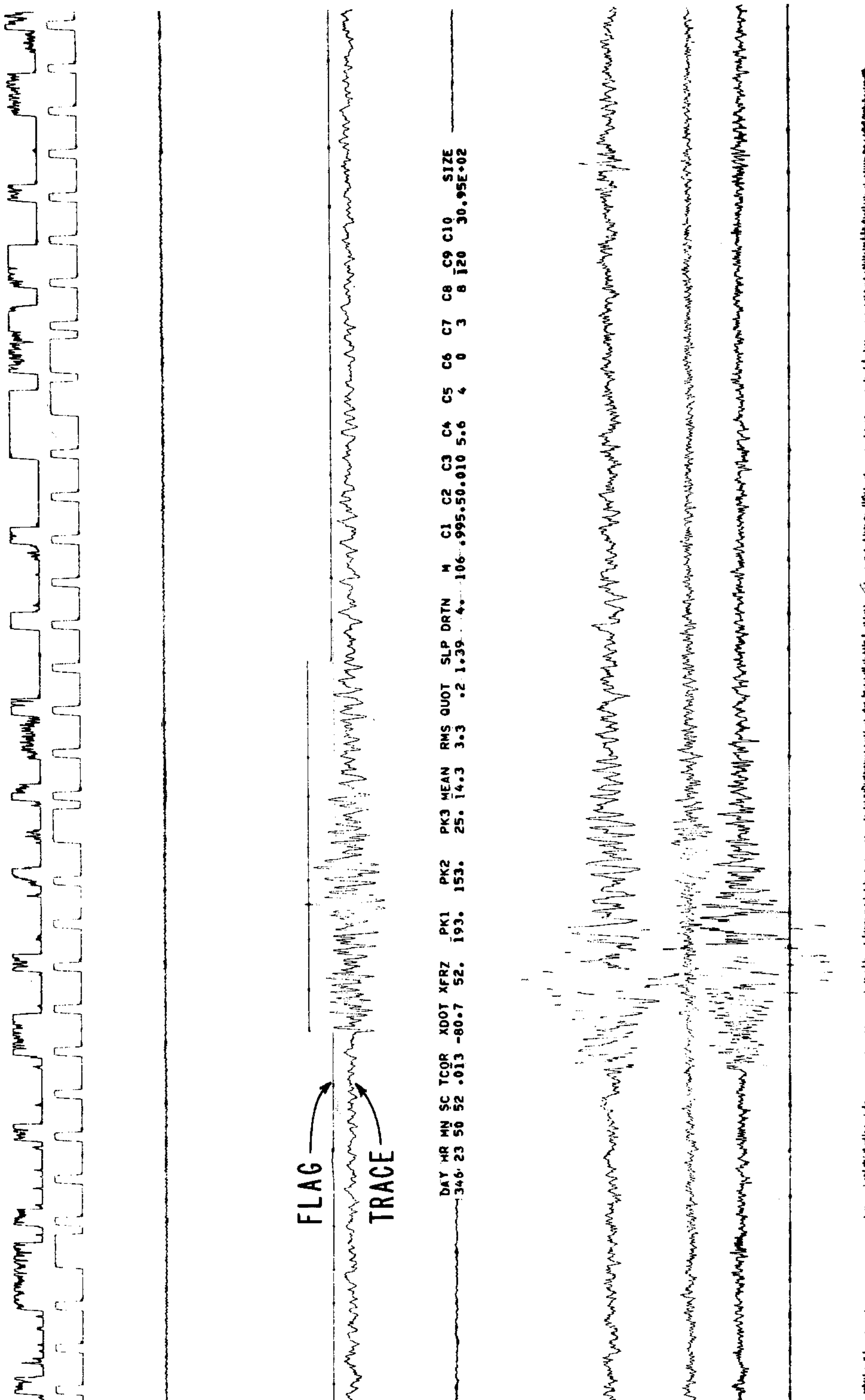

FIG. 6. Analog record of computer pick for a small earthquake during a test run. The trace being monitored by the program is labeled "Trace". "Flag" indicates program status; when it is set high, a pick has been made. The timing traces are 1 sec and ½ sec. The superimposed line of print is the output of the line printer describing the event and listing some program constants.

realized that the figure is representative of the number of really reliable picks made by the program. If marginal picks are included its performance figure is correspondingly improved. In addition, the program's criteria were so stringent as to exclude almost all false picks on noise bursts. If some false picks are allowed performance figures again can be improved. It is not possible to assign a single performance index which covers all cases since each application will have different requirements and different evaluation criteria, but the 60 to 80 per cent figure is probably conservative for most applications.

In these test runs of an hour or less, the program usually makes no picks at all on noise sources, so a different test scheme was used to evaluate this facet of its performance. A particularly noisy station in the Central California Network, PKH, was monitored in real time during several extended periods. In the longest test run of 44 hours, the program reported 36 events, all of which were noise bursts. (There was no earthquake recorded on this station during any of the test runs.) Of these 36 events, 33 were weighted 3 by the program and 3 were weighted 2. No events were reported with weights of 1 or 0, corresponding to reliable picks.

Bear in mind that the program was instructed to report all events longer than the minimum threshold. Although it called the analysis routines and reported their results, it did not make use of these results in deciding what should be reported. Use of these results can reduce considerably the number of false reports.

In the preceding discussion, it has been assumed that the program will always be required to operate independently, that is, that it will report only results, and that the seismic record will never be available for study by human operators. There are indeed applications in which this mode of operation may be required. Operation of a micro-earthquake network in deep ocean areas might be one example. In such a case, the bottom package might be required to operate independently for up to several months and to store information on several hundred earthquakes for later retrieval. If the descriptors are restricted to a few hundred bits of information, this becomes a practical possibility.

At another extreme, the program might record the entire seismic trace, either analog or digital, for each event of sufficient length to qualify as a possible earthquake of interest. In this mode, the resulting output would be an edited tape containing only events. The advantage of this mode is that all trace detail is preserved; the disadvantage, of course, is that a much larger body of data must be recorded, the hardware problems and costs are thereby multiplied.

In principle, if the recognition algorithm works properly, equipment can be built to operate at either of these extremes or in some intermediate mode, if desired.

Further tests and development planned for the immediate future should demonstrate more clearly the strengths, weaknesses, and potential uses of this algorithm.

References

Ambuter, B. P. and S. C. Solomon (1974). An event-recording system for monitoring small earthquakes, *Bull. Seism. Soc. Am.* **64,** 1181–1188.

DeNoyer, J. (1959). Determination of the energy in body and surface waves, Pt. II, *Bull. Seism. Soc. Am.* **49,** 1–10.

Lee, W. H. K., R. E. Bennett, and K. L. Meagher (1972). A method of estimating magnitude of local earthquake from signal duration, *U.S. Geol. Surv. Open File Report.*

Stewart, G. W. (1973). *Introduction to Matrix Computations*, Academic Press, New York.

Stewart, S. W. (1977). Real-time detection and location of local seismic events in central California, *Bull. Seism. Soc. Am.* **67,** 433–452.

Stevenson, R. (1976). Microearthquakes at Flathead Lake, Montana: A study using automatic earthquake processing, *Bull. Seism. Soc. Am.* **66,** 61–79.

Soucek, B. (1976). *Microprocessors and Microcomputers,* John Wiley and Sons, New York.

U.S. Geological Survey
345 Middlefield Road
Menlo Park, California 94025

Manuscript received December 8, 1977