(12)

VSC-TR-82-3

SINGLE CHANNEL SEISMIC EVENT DETECTION ON ALASKAN SHORT-PERIOD DATA

Robert R. Blandford
Seismic Data Analysis Center
Teledyne Geotech
314 Montgomery Street
Alexandria, Virginia 22314

11 DEC 81

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

Monitored By:
VELA Seismological Center
312 Montgomery Street
Alexandria, VA 22314

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER VSC-TR-82-3	2. GOVT ACCESSION NO. AD-A116 413	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) SINGLE CHANNEL SEISMIC EVENT DETECTION ON ALASKAN SHORT-PERIOD DATA		5. TYPE OF REPORT & PERIOD COVERED Technical
		6. PERFORMING ORG. REPORT NUMBER SDAC-TR-81-15
7. AUTHOR(s) Robert R. Blandford		8. CONTRACT OR GRANT NUMBER(s) F08606-79-C-0007
9. PERFORMING ORGANIZATION NAME AND ADDRESS Teledyne Geotech 314 Montgomery Street Alexandria, Virginia 22314		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS VT/9709
11. CONTROLLING OFFICE NAME AND ADDRESS VELA Seismological Center 312 Montgomery Street Alexandria, Virginia 22314		12. REPORT DATE 12/11/81
		13. NUMBER OF PAGES 54
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Defence Advanced Research Projects Agency 1400 Wilson Boulevard Arlington, Virginia 22209		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

S ELECT
JUL 2
H

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Seismic Detection	MARS
Signal Detection	Walsh
Detection	Z-log

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Three advanced detectors and a standard detector were tested on 15 of the weakest 40 teleseismic signals associated to events and received at an array in Alaska in the time interval 1-6 November 1980. The signals were originally detected on array beams and the detectors were given the task of detecting the signals on data from a single instrument. The false alarm rate was set at 3 to 4 per hour by comparing single instrument detections to beam detections. The optimum Z-log detector detected all 15; the Walsh detector, 13; and the MARS

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

detector, 11. This relative ranking is in agreement with an earlier study using signals buried in phase-scrambled noise.

The standard detector detected 14 out of the 15, whereas it was worse than all the other three in the early study. This may be because the standard detector takes the absolute value of the filtered data instead of squaring it and thus is more resistant to the effects of spikes and weak precursor signals. More likely the standard detector was at a disadvantage in the earlier study becuase the signal time window was held at 2.5 seconds while the signal windows of the other detectors were much larger as appropriate to the larger signals which were buried in noise.

All existing detectors are probably inadequate, in their present form, to be reliably used to select data segments for transmission because of weaknesses in the rejection of spikes, drop-outs, and noise bursts; weakness in detection in coda; and inability to suppress multiple detections in a strong event. However, all detectors are equally susceptible to improvement in these aspects.

SINGLE CHANNEL SEISMIC EVENT DETECTION ON ALASKAN SHORT-PERIOD DATA

SEISMIC DATA ANALYSIS CENTER REPORT NO.: SDAC-TR-81-15

AFTAC Project Authorization No.:	VELA T/0709/B/PMP
Project Title:	Seismic Data Analysis Center
ARPA Order No.:	2551
Name of Contractor:	TELEDYNE GEOTECH
Contract No.:	F08606-79-C-0007
Date of Contract:	1 October 1980
Amount of Contract:	$1,538,055
Contract Expiration Date:	30 September 1981
Project Manager:	Robert R. Blandford (703) 836-3882

P. O. Box 334, Alexandria, Virginia 22314

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

ABSTRACT

Three advanced detectors and a standard detector were tested on 15 of the weakest 40 teleseismic signals associated to events and received at an array in Alaska in the time interval 1-6 November 1980. The signals were originally detected on array beams and the detectors were given the task of detecting the signals on data from a single instrument. The false alarm rate was set at 3 to 4 per hour by comparing single instrument detections to beam detections. The optimum Z-log detector detected all 15; the Walsh detector, 13; and the MARS detector, 11. This relative ranking is in agreement with an earlier study using signals buried in phase-scrambled noise.

The standard detector detected 14 out of the 15, whereas it was worse than all the other three in the early study. This may be because the standard detector takes the absolute value of the filtered data instead of squaring it and thus is more resistant to the effects of spikes and weak precursor signals. More likely the standard detector was at a disadvantage in the earlier study becuase the signal time window was held at 2.5 seconds while the signal windows of the other detectors were much larger as appropriate to the larger signals which were buried in noise.

All existing detectors are probably inadequate, in their present form, to be reliably used to select data segments for transmission because of weaknesses in the rejection of spikes, drop-outs, and noise bursts; weakness in detection in coda; and inability to suppress multiple detections in a strong event. However, all detectors are equally susceptible to improvement in these aspects.

TABLE OF CONTENTS

LIST OF FIGURES

INTRODUCTION

In a recent study, Blandford, Racine, and Romine (1981) showed that the best detector of signals buried in phase-scrambled noise was one in which first an optimum S/N^2 filter was updated and applied to the data. The filter was followed by computation of a 10-second average of the squared filtered data; the logarithm of this short-term-average (STA) was taken and the result used to recursively update a long-term-average (LTA) with a time constant of 2 minutes. The standard deviation (σ) of the difference between the STA and LTA was also recursively computed and finally, a Z-detection statistic $(STA - LTA)/\sigma$ computed. On Figures 1 and 2, which are taken from the study by Blandford et al., we see that this detector, indicated by the letter Z, does have the highest relative magnitude threshold for a fixed false alarm rate.

The next best detector, on the average, was the Walsh detector, designated in these figures by the letter W. This detector is described in detail by Goforth and Herrin (1981). The data is prefiltered with a recursive frequency filter; the Walsh transform is taken of successive 3.2 second windows with 1.6 second overlap. The absolute values of the Walsh coefficients corresponding to the range of circular frequencies with good S/N are summed, after each is weighted with its own long-term average. This sum is the detection statistic, and the threshold is set by maintaining a histogram of 512 such successive values (a time interval of about 13 minutes) and by comparing each value to a constant times the difference between the 50th and 75th percentile histogram entries. If the threshold is crossed twice consecutively, a detection is declared.

Less than 0.01 m_b units worse than the Walsh detector are the "standard" detectors (run 18 for Pinedale and 24 for NORSAR), a simple upgrade of the online detector in which the short term average is squared instead of the absolute value being taken, and in which a single threshold crossing is sufficient for detection instead of 2/3 or 3/3. These changes are in accord with theories of detection as discussed by Blandford et al. (1981).

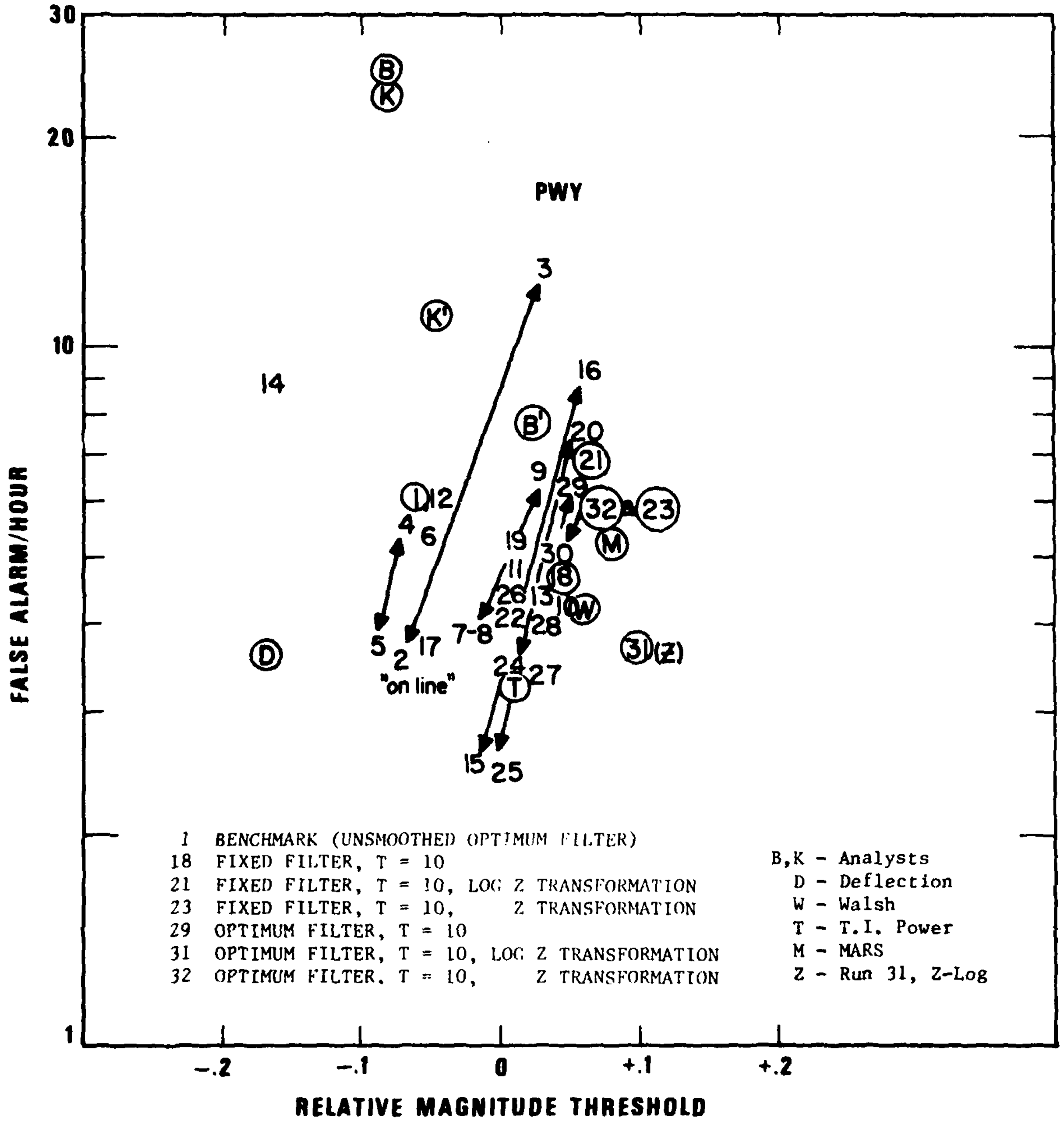

Figure 1. False alarm versus relative magnitude threshold for different detection on Pinedale test tape (from VSC-TR-81-8).

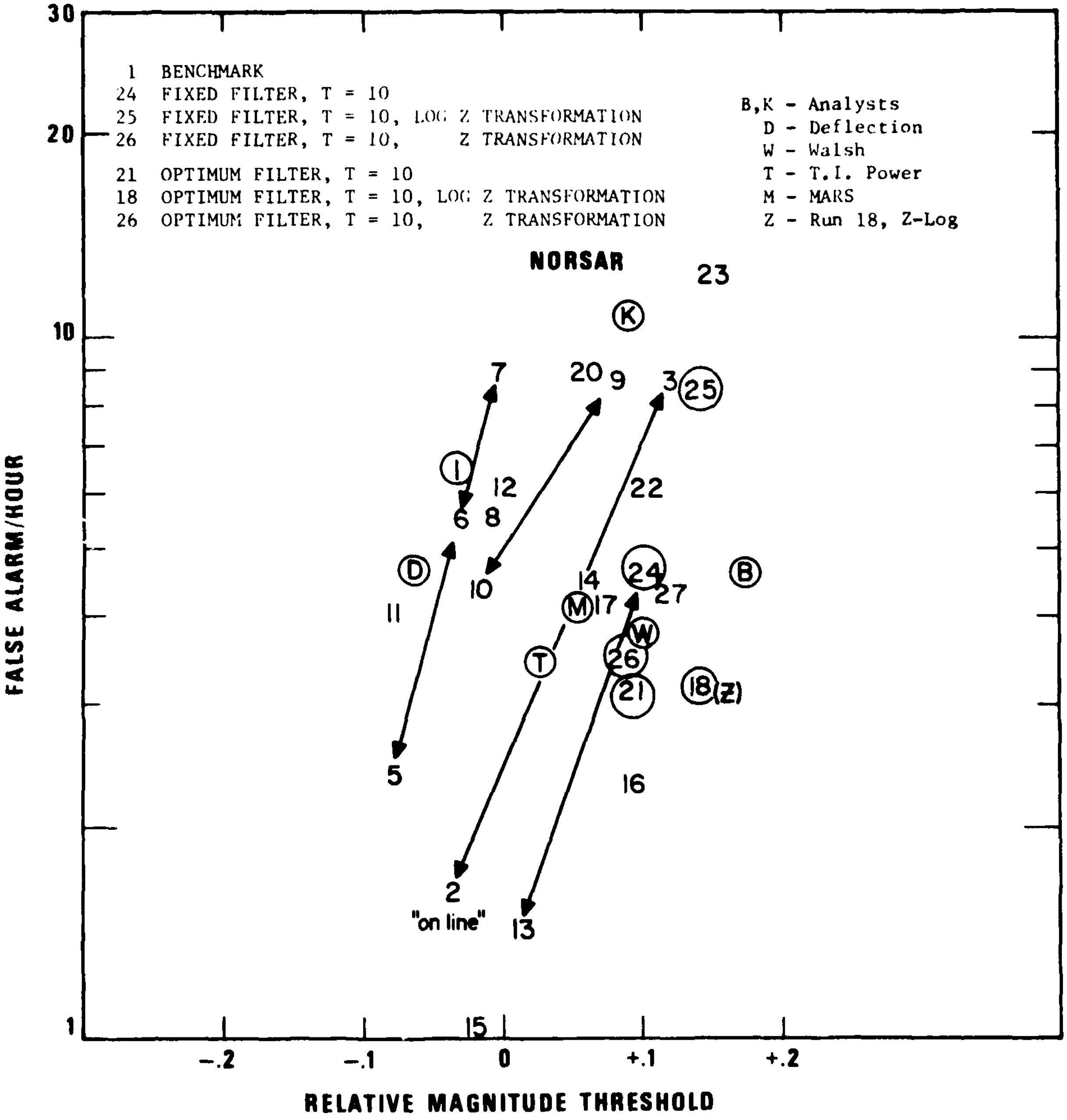

Figure 2. False alarms versus relative magnitude threshold for different detectors on NORSAR test tape (from VSC-TR-81-8).

The MARS detector was also one of the better detectors although, on the average, worse than those mentioned above. This detector is discussed in detail in Farrell et al. (1981). The envelopes of several narrow bandpasses are computed and the mean and standard deviation of the relative maxima of these envelopes are exponentially summed over the previous 100 maxima. If a fixed number of the maxima in a 2.8 second time window exceeds the mean plus a constant times the standard deviation, then a detection is declared. This is a "voting" detector as defined by Wirth et al. (1976), and for this reason, one is not surprised to find that it does not have the best performance.

The "online" detector is the detector most like the detector programmed for LASA by IBM and is in routine use at the SRC at this writing. First, the data is processed through a fixed filter and a 1.5 second short term average (with 1.0 second overlap) of the absolute data values is compared to an LTA exponentially averaged over 60 seconds. Three successive threshold crossings give a detection. After detection, the LTA update time is switched to 6 seconds until the signal shuts off.

A more detailed discussion of the actual implementation of all of the above detectors, together with condensed program listings, may be found in Appendix I.

A possible objection to the study by Blandford et al. (1981) is that the data were not "real", that is, the phase of the noise had been scrambled to ensure that there were no signals in the "noise". By so doing, the authors may have altered the statistics of the noise (for example, by making it more Gaussian) in such a way that a detector which was best on that data would not be best in practice.

To test this hypothesis, we first ran the online detector on BFAK beams for a time span for which experienced analysts had made especially careful analyses of the develocorder beam traces. As discussed in

Appendix I, the result of this was that the detector seemed to be out-performing the analysts, so that there was no way in which the analysts could function as a standard against which the different detectors could be compared.

The remainder of this report discusses the results of using the performance of the detectors on a single instrument as a measure of detector quality, while using the analyst detections of small teleseismic signals on the beam as a standard.

EXPERIMENTAL DESIGN

We examined the A-list for November 1, 2, 4, 5, 6 and found the 40 lowest amplitude signals measured at BFAK which were confirmed as P arrivals from events at distances of greater than 20°. We then searched the bulletins and the raw analyst records and found two 4-hour periods during which no arrivals were detected.

To set the false alarm rate, we first ran the online detector on the direct sum trace and on a single high-gain instrument for the two 4-hour periods. Any detection on the single trace, which is not classified as a spike, local, or drop-out, which is classified as a teleseism, and which is not detected on the beam trace, is defined to be a false alarm. The threshold was adjusted until the false alarm rate (FAR) was in the range 3-4/hour. (Examination of Figures 1 and 2 shows that variation of the FAR over this range corresponds to a variation in magnitude threshold of less than 0.01 m_b.)

We then ran the other detectors on the single channel data. If a detection occurred within 30 seconds of a spike, local, or drop-out as declared by the online detector, then it was not classified as a false alarm, nor was it a false alarm if a detection occurred within 30 seconds of a detection on the beam. Again, the threshold was adjusted for each detector until the false alarm rate was 3-4/hour.

It is of interest to remark that during these quiet 8 hours when simultaneous (often to the tenth of a second) detections would occur on the beam and single instrument, then the beam S/N value would be substantially higher. Also, although all the parameters were set identically, the detection rate was 2-4 times higher on the beam than on the single instrument. These facts certainly suggest that many real signals were being detected by the computer which the analyst did not detect.

The 40 signals mentioned in the first paragraph of this section were analyzed by starting the detectors 15 minutes before the signal start time and seeing if a signal was detected between the times 10 seconds before to 30 seconds after the analyst's start time. (The precursor 10 seconds is to allow for analyst error and for possible window overlap effects in the advanced detectors.)

In the process of recovering these 40 signals from the station log tapes, there were difficulties with 6 tapes, leaving 34 signals to be analyzed. Upon examining the detection results, we found that the Z detector often missed signals if there was another signal detected by it a few minutes before, and that the Walsh detector sometimes failed if it did not have a full 13 minutes of noise to fill the detection histogram before the signal arrival.

The reason for the Z detector's problem is that the LTA decay rate was set to be 2 minutes, and this led to an elevated LTA after events, and thus to a missed signal. A similar problem may exist for the present coding of the MARS detector.

The Z problem can perhaps be solved by adjusting the LTA decay rate to a shorter time for a minute or so after every detection; the Walsh detector weakness should not be a problem in operational practice.

To eliminate these problems, we further discarded 2 events whose log tapes began 5 and 7 minutes before the signal, and also eliminated the 17 events with Z detections in the 5 minutes before the predicted signal when the Z FAR was set to 1/hour. This left a total of only 15 events out of the original 40. While this is a severe reduction in the data base, it seemed to be the only way to evaluate the algorithms fairly for their detection potential alone. Any of the detectors can and must, in practice, be surrounded with substantial "special case" software to handle the complicated situations which arise when multiple weak, moderate, and strong signals arrive, but the detector's success in these cases is little indication of the underlying quality in pure detection. It is, of course, in the handling of these complex situations where the human analyst's capabilities are most impressive.

RESULTS AND DISCUSSION

In the analysis of the 15 single instrument recordings, the Z detector detected all 15, the online detector detected 14, the Walsh detector detected 13, and the MARS detector detected 11. This ranking is in general agreement with the relative performance found by Blandford et al. (1981), except that the performance of the online detector is much better in the present instance. This may be due simply to a statistical fluctuation in such a small sample, or it may be that taking the absolute amplitude is a good thing to do in real data as suggested by Goforth and Herrin (1981). Finally, it is most likely that the standard detector was at a disadvantage in the earlier study because the signal time window was held at 2.5 seconds while the signal windows of the other detector were much larger as appropriate to the larger signals which were buried in noise.

Since this is a new data source, it might be said that none of the detectors have been fairly evaluated since none of them have been optimized across parameter space. I have pretty much set the parameters at values which were close to those used previously, modified by what would seem reasonable for the present application. However, my feeling is that, since these results are roughly consistent with the performance found by the Blandford et al. (1981), the referenced results can be trusted as a guide to practice.

To the extent that this is true, then either the online detector modified to square the data and detect on one threshold crossing, or the Walsh detector would be the "best buy" since they are both very fast, being limited by the recursive filter calculation. My own preference would be for the modified online detector because it follows naturally from classical detection theory. If there is no problem with computer time (the optimum Z detector runs a single channel 16 times faster than real time on the 11/70 under UNIX as a sole user, a dedicated 11/70 analyzing several channels would run 2-3 times faster per channel) then the optimum detector would be the obvious choice. It optimally adapts to changes in the noise spectrum including changes in the microseisms and thus requires less "site tuning" and it is the most sensitive.

It is extremely important to note that before any detector is relied upon to select segments to return to a central site for further analysis, it is important that it be evaluated as to its capability to reject false alarms, to not miss later phases and mixed events,and to not "over-produce" in the coda of an event. To my knowledge, no detector is currently functional which meets all these criteria, although I believe the online system at the SRC is close, being weak mostly in the criteria of later phases and mixed events.

Another interesting result of this study is that we again see how difficult it is to find a standard by which to judge detectors. Had the detectors been only slightly better, or the array only slightly smaller, then all detectors would probably have detected all 15 events and it would have been impossible to rank them. Possibly the only way to get a good sample of real data for comparing enhancements of the optimum detector will be not from analyst picks but by relying on the detector operating on the beam as a flag for candidate signals on the single instruments. This, however, is "flying blind" and careful thought would be required to ensure that errors are not being made.

REFERENCES

Blandford, R. R., D. Racine, and R. Romine (1981). Single channel seismic event detection, SDAC-TR-81-7, (VSC-TR-81-8), Teledyne Geotech, Alexandria, Virginia.

Farrell, W. E., J. Wang, R. C. Goff and C. B. Archambeau (1981). Evaluation of the MARS seismic event detector (Draft), SSS-R-81-4656, Systems, Science and Software, La Jolla, California.

Goforth, T. and E. Herrin (1981). An automatic seismic signal detection algorithm based on the Walsh transform, Bull. Seism. Soc. Am., 71, 1351-1360.

Wirth, M. H., R. R. Blandford, and R. H. Shumway (1976). Automatic seismic array and network detection, Bull. Seism. Soc. Am., 66, 1375-1380.

APPENDIX I

April 7, 1981 Geotech Memorandum from R. R. Blandford,
"Evaluation of SDAC On-Line Detector on Nov. 1, 2, 1980 -
On-Line ALK Data - Revision of 30 March 1981 Memo

TO: R. Van Nostrand
FROM: R. R. Blandford, R. Baumstark
DATE: April 7, 1981
SUBJECT: Evaluation of SDAC On-Line Detector on Nov. 1, 2, 1980
On-Line ALK Data - Revision of 30 March 1981 memo.

cc: M. Shore, P. Kovacs, A. Hill, Capt. P. Terry, Maj. G. Ullrich,
D. von Seggern, Z. Der, R. Romine

Summary

We have run the on-line detector on the Nov. 1 and 2, 1980 data base which was picked by experienced analysts. For speed of analysis, we used the DP tapes as a data source. In terms of initial teleseisms, the detector works very well, detecting 50/57 = 88% of all signals and 42/44 = 95% of those not designated as "possible background", "probable background" or "possible dram. In addition, the on-line detector detected 14 signals in the first 4 hours of Nov. 1 which were not detected by analysts but which upon close examination with regional beams and filters we felt were good teleseismic picks. The 14 signals are labeled A,B,C. Both analysts felt that the "A" signals speak for themselves without beams or filters; that the B signals were good and the C signals questionable. One of the two analysts felt that beams and filters did not actually held him to decide if any signals were real but did offer more information about signals assumed to be real. It is important to state that in our opinion these are judgement calls and there is no way we know to be certain that one plot is signal and the other is noise.

At the 14 in four hour rate, there would be 143 missed by the experienced analyst in the 41 hours of DPS tape data available. There is no particular reason to think that these are not "real signals." This is not to say that the Geotech analysts are "better" than experienced analysts, only that they could concentrate more because they did not have to do the routine scanning.

In Table I we see a calculation to show that in one day we would expect there to be 46 signals on a monitored channel (46 x 5 = 230 over all of ALK) that would be "unconfirmable" at ALK in the sense that the signal would not be automatically detected at 3 or more stations in total. (The calculation assumes that there are 5 m_b = 4.0 events per day on the average, and that the exponential law for seismicity holds with b = -1.0. The S/N ratio is assumed to vary across the network with σ = 0.4, characteristic of LASA.) The number of unconfirmable detections would be 79 in 41 hours. Note that with higher or lower false alarm rates, detection numbers could change by a factor of two and there would be no way to "confirm" that a weak detection was a true signal or false alarm. Note that most of the "unconfirmed" detections come from magnitude 3.0 events. This calculation is of course only for plausibility. Thresholds might be lowered when looking for confirming events, but in this case, even more "one-station" detections will be made, further complicating the issues. However, using the machine-Geotech analyst as a standard then the probability of detection of the experienced analyst is only ≈ 50/(50 + 143) ≈ 25%. Clearly this makes it very difficult to use the experienced analyst as a standard. Although the experienced analysis is only "preliminary" even a two-fold increase in detections will leave the situation extremely ambiguous and impossible to interpret satisfactorily.

If there is no standard, then it seems impossible to quantitatively compare one detector to another using on-line data. (Except that if a detector was markedly inferior to the present on-line system it could be found to be so. For example a careful study of all the data by experienced analysts may make it possible to set thresholds such that the automatic detectors detect all of the first pass detections of the experienced analysts plus other good picks without any questionable picks. This would show that the DP is superior to the analyst in this narrow field. But the results of TR-81-7 would suggest that it would be impossible to assert that weak detections by the best automatic detectors are or are not signals since most analyst "false alarms" in that paper looked like signals and many true signals detected by the automatic detector showed little evidence of signal. The same phenomena are noted in our on-line system where the detector triggers on arrivals which would not be selected by the analyst unless he had knowledge from other stations as to the signal's shape and arrival time.)

For these reasons, we suggest that the aims (comparing teleseismic detectors) of this detection study be rethought. If it is truly desired to compare different detectors then perhaps more "scrambled phase" data of different types should be created. Another line of work would be to run two detectors simultaneously (higher frequency, longer window for the second detector) to improve detections of locals which were only detected with ~ 50% reliability, and to test possible improvements in detection of later phases by adjusting the LTA update time constant and threshold in the first 30 seconds to 1 minute after an initial detection.

Teleseismic Detection Results

The on-line detector at the SDAC is exactly as discussed in detail in an Appendix to the recent report by Blandford, Racine and Romine (1981, SDAC-TR-81-7). Briefly, a 1.2-3 Hz prefilter is followed by an STA/LTA detector with an STA of 1.5 seconds and a recursively updated LTA of ~ 30 seconds. The STA is shifted 0.5 seconds at a time and 3 threshold crossings in a row are required for detection. In running the on-line detector on DPS tapes the time period 1324 to 2045 Nov. 2 had to be skipped because of data gaps.

A 12 March 1981 VSC memo from R. Kimmel transmitted experienced analysis results for 01 and 02 Nov. 1980 of an infinite velocity beam from BFAK. The analysis broke detections down into event groups with initial and later phase, and the events characterized as dram (locals), and pint or quart (multiple or impulsive teleseism). Remarks such as possible background, possible dram (of a pint), possible noise, probable noise, possible regional motion (of a pint), possible dram motion (of a quart), and possible surface motion (of a pint) indicate analyst uncertainty over some picks. There are a total of 13 such picks and 44 with no uncertainty indicated for a total of 57. Of the 57, 50 were detected by the machine for a probability of detection of 88%. If we consider only the 44 "more certain" picks, then 42 of them were detected for 95% probability of detection. These statistics are tabulated in Table II.

In Figure 1, we see plots of the two events missed by the machine and about which the analyst indicated no uncertainty. The upper signal is rather high frequency---perhaps a regional event although not called that by the analyst. Perhaps also it is simply the temporary interference of the microseisms revealing the on-going high frequency noise--that is--a false alarm. The lower event is certainly weak, although we agree that it should be picked. In Figure 2 we see plots of the 5 signals not detected by the machine out of the population of 13 teleseisms of which the analysts were uncertain. The second and third signals from the top do show indications of being "rolling" background.

In Figure 3, we see 14 signals detected by machine in the first 4 hours of Nov. 2 which were not picked by the experienced analyst but which seem to us to be every bit as "real" as the analyst picks. In almost every case, there is a definite change in waveform character around the indicated detection time which persists well beyond the 2.5 second window needed for detection triggering. As discussed in the summary, this leads to an estimate of analyst probability of detection of only 25% if the machine-analyst were adopted as a standard. This can not be thought to be impossible when one considers the recent results of the analyst-detector comparison on signals buried in phase-scrambled noise in SDAC-TR-81-7.

In Figure 4 we see two signals which were discarded by the analyst on the basis that they were not thought to be signals. Again, one analyst used beams and filtering to make this judgement, the other did not find these aids to be a help in the narrow question of pure detection, although he felt that they are a great help in putting out a bulletin. Beam plots for all signals are available for inspection. It seems worth remarking that Figure 4 may show valid signals. Good detections were made in TR-81-7 that looked as poor, and the on-line detector has triggered on signals as poor which have been shown by analysis of the complete event to be real.

Detection Timing

As discussed in the Appendix of SDAC-TR-81-7, the on-line detector has a detection-time refinement feature in which the STA/LTA is "backed up" using a lower threshold than that used to determine detection. It has been our experience that this detector is generally better at picking weak starts than is the human analyst as judged by its better agreement with start times determined by overlay techniques using stronger signals from nearby stations. This may be seen to some degree in Table III where the median machine-analyst time is -0.3 sec (machin late) for signals of which the analyst was certain, but about which the analyst indicated some uncertainty over start time or signal type. The early detection for weak as opposed to strong signals suggests that the automatic detector is more accurate. A definitive test would have to involve signals buried in noise or perhaps a formalization of the analyst comments.

Later Phases

One of the most spectacular failings of the present detector is its failure to detect later phases. The present on-line system very rapidly builds up the LTA after a detection and then returns to the very slow LTA rate (30 seconds) existing before the detection. In Table I we see that only 29/74 later phases were detected. The reasons for the present LTA post-detection update parameters are strictly historical and we are free to change them. Perhaps a change for 1 minute after an initial detection to a 10-sec LTA integration time coupled with a lowered threshold will result in better later phase detection.

Local Signals

The on-line system has been designed to detect small, impulsive, short-period teleseismic signals. The filter design assumes $t^* = 0.3$. For regional phases 0.1 would be better and a signal window of 5-20 seconds would seem appropriate. We would run both detectors in parallel and see if a more useful assortment of detections is made and whether AA will then turn out a better Alaskan regional bulletin.

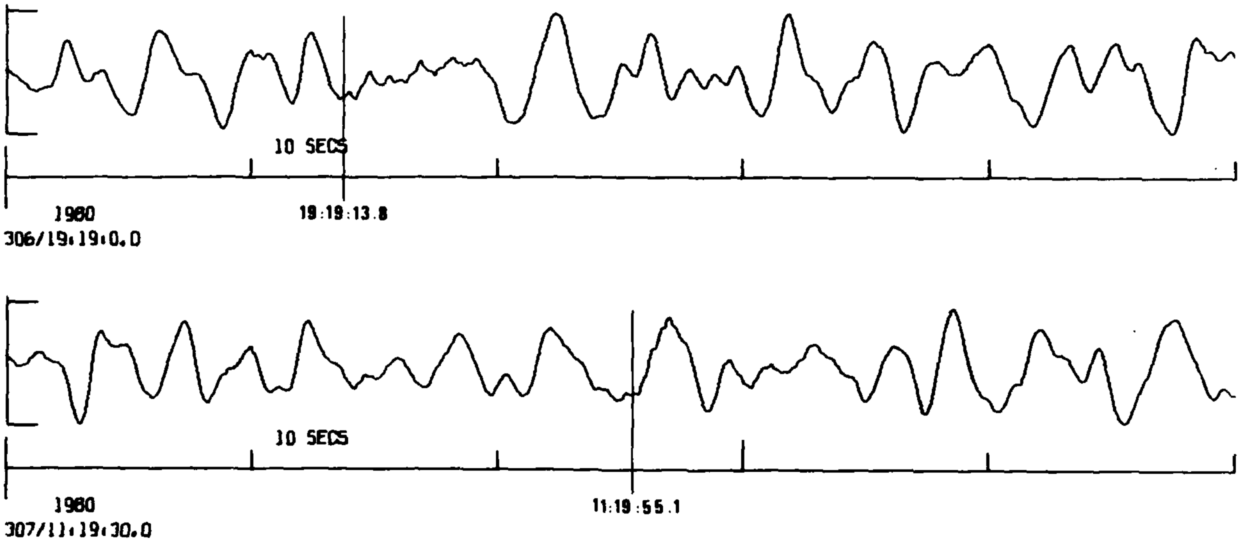

Figure 1. Initial "teleseisms" missed by machine.

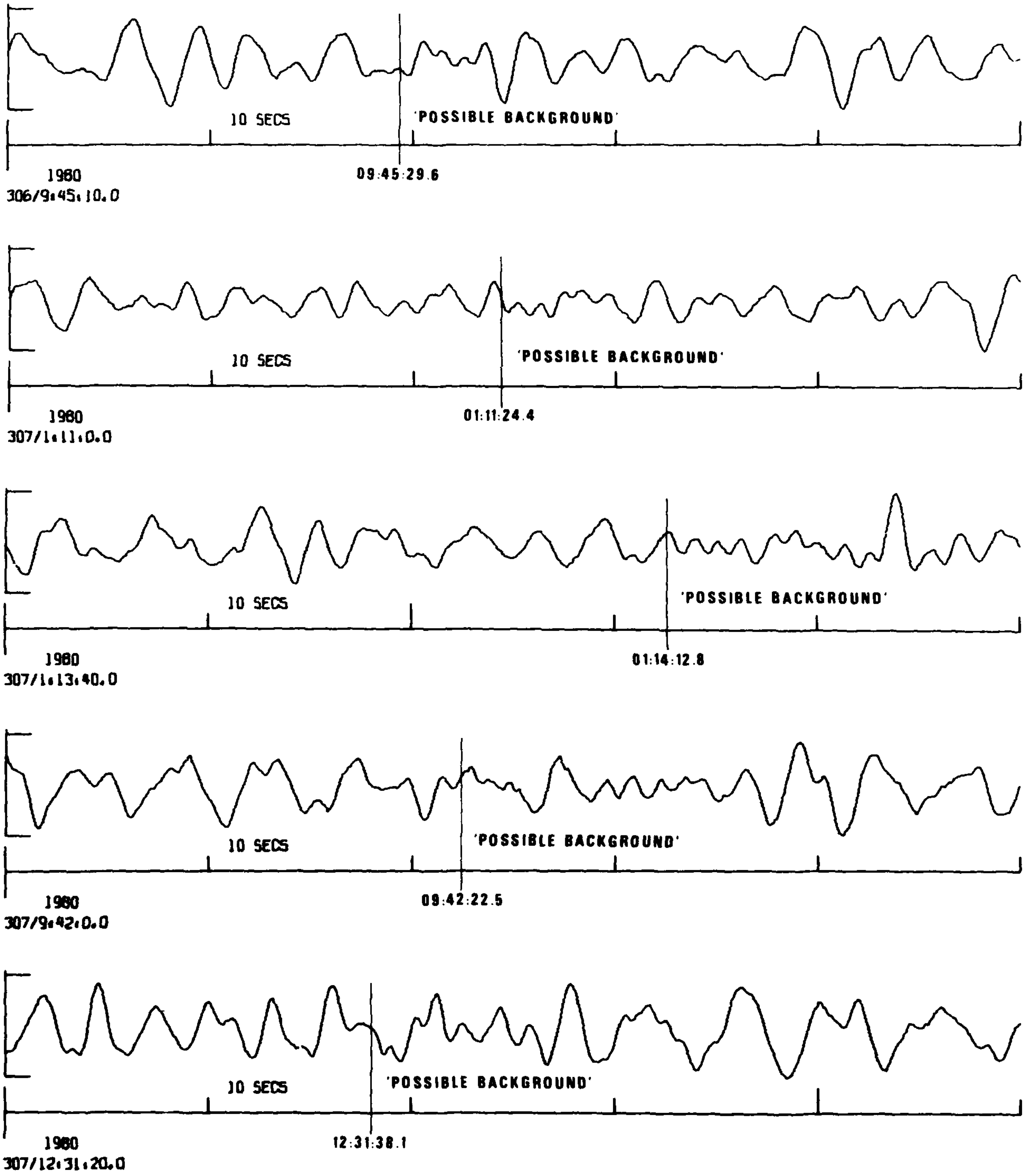

Figure 2. Initial "teleseism" misses called "possible background" by experienced analyst and missed by machine.

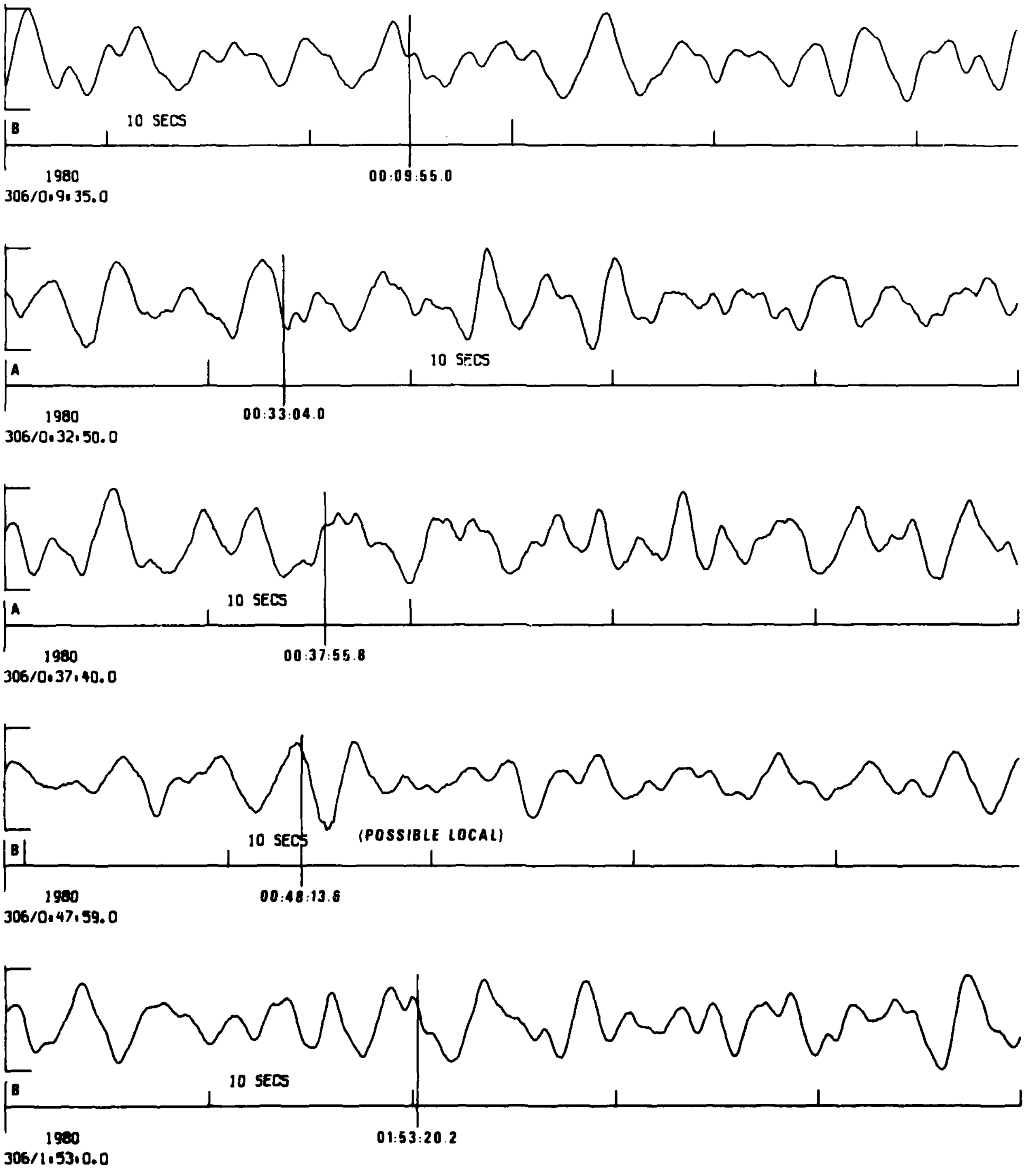

Figure 3-1. Machine detections missed by experienced analysts and rated good picks by Geotech analysts. First 4 hours of Nov. 1.

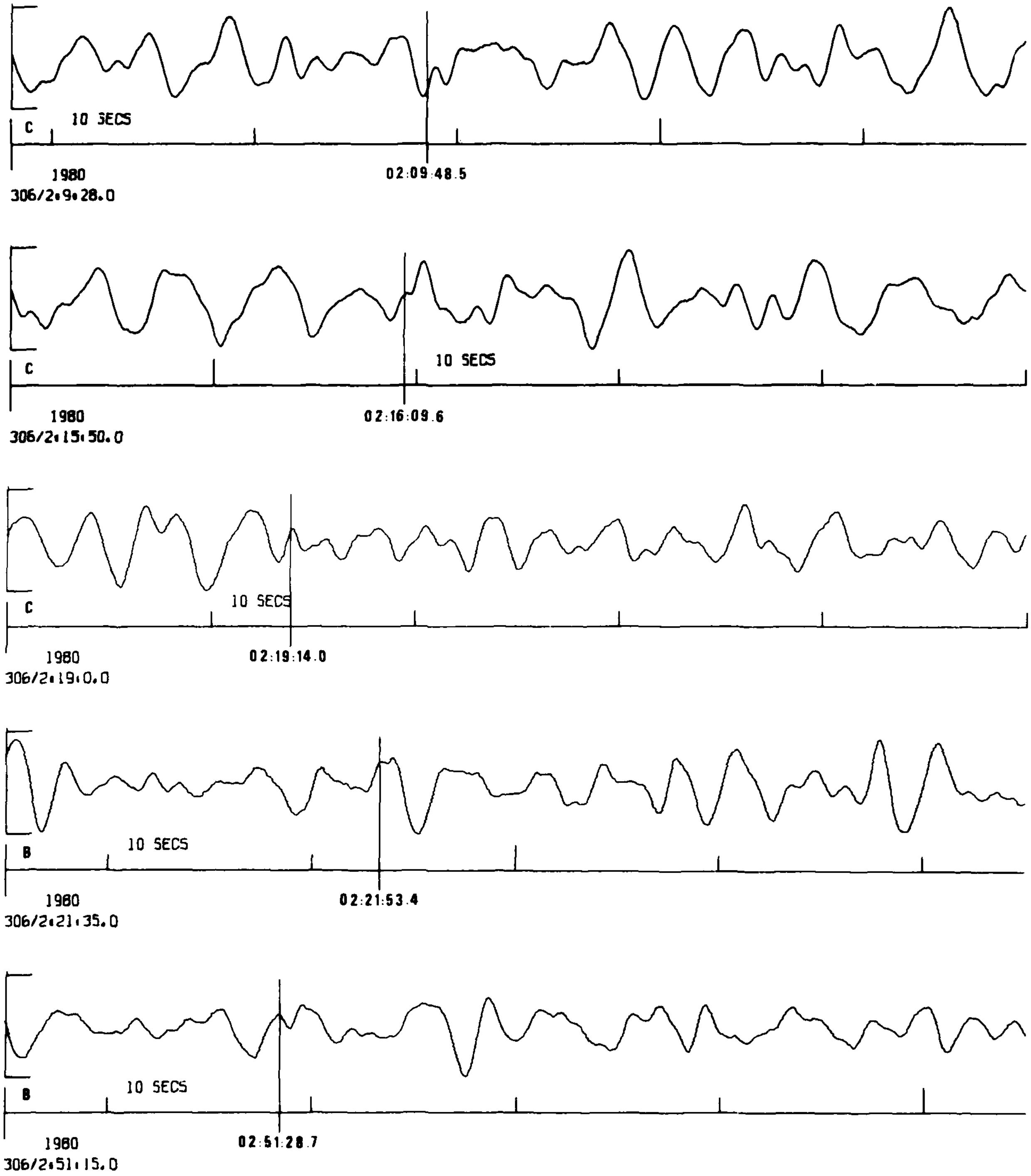

Figure 3-2. Machine detections missed by experienced analysts and rated good picks by Geotech analysts. First 4 hours of Nov. 1.

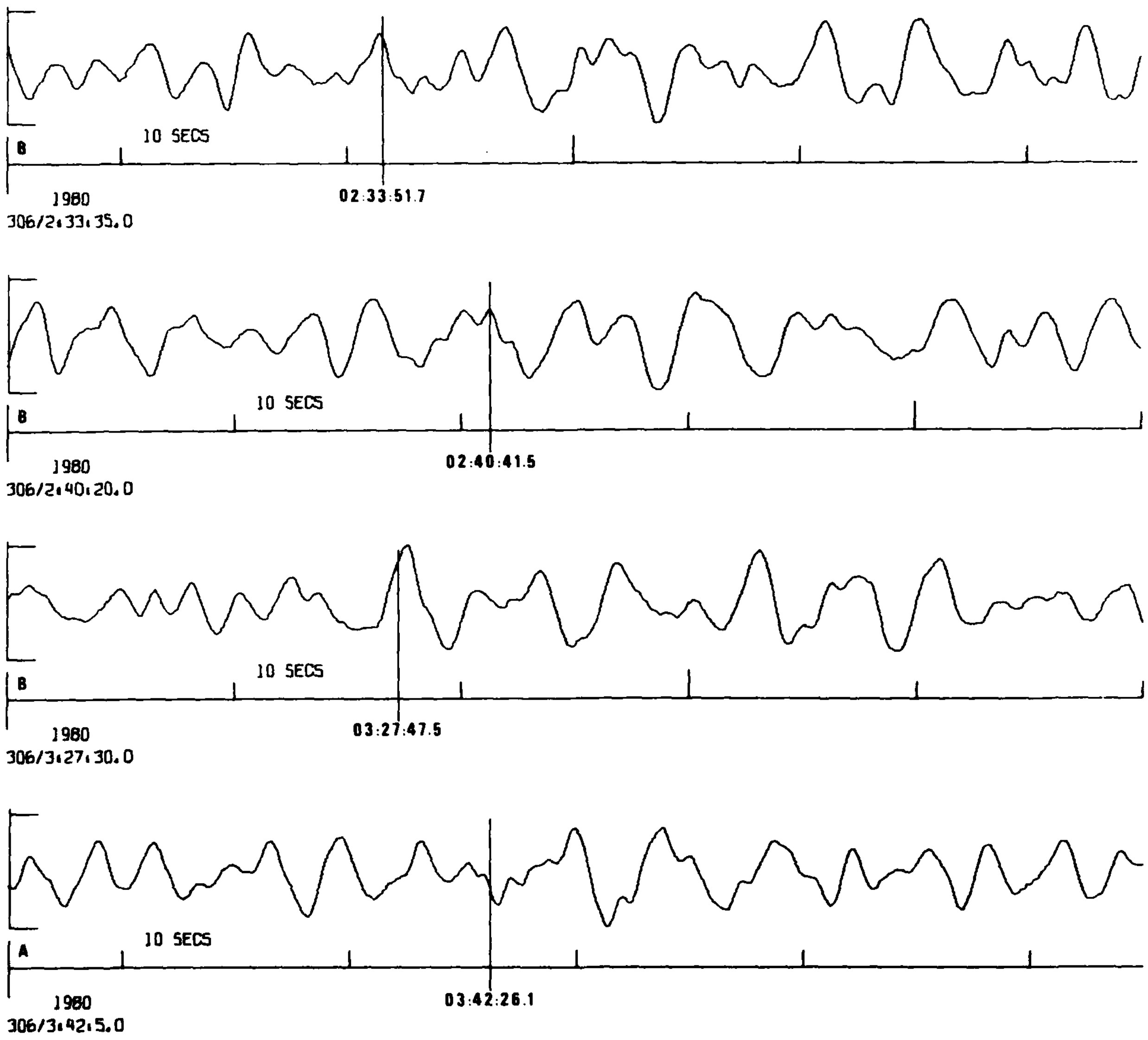

Figure 3-3. Machine detections missed by experienced analysts and rated good picks by Geotech analysts. First 4 hours of Nov. 1.

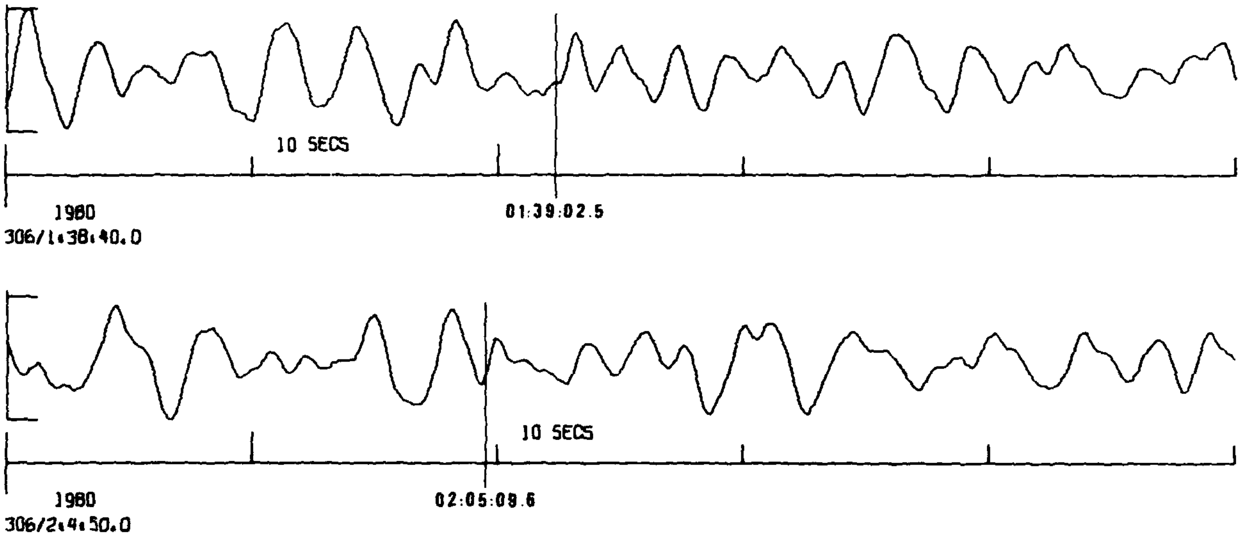

Figure 4. Machine picks rejected by Geotech analysts as not showing good signal.

TABLE I

m_b	$N(m_b)$	$P_d(m_b)$	P(D = 1)	P(D = 2)	P(D = 3)	P(D = 4)	P(D = 5)	N(1)	N(2)	N(3)	N(4)	N(5)
4.0	5.0	.9	.0004	.008	.07	.33	.59	.0	.0	.4	1.6	3.0
3.9	6.2	.84	.0027	.028	.14	.39	.44	.0	.2	.9	2.4	2.8
3.8	7.9	.77	.0096	.072	.26	.39	.23	.1	.6	2.0	3.1	1.8
3.7	10.0	.69	.031	.14	.30	.36	.17	.3	1.4	3.0	3.6	1.7
3.6	12.5	.59	.083	.24	.34	.26	.08	1.0	3.0	4.2	3.2	1.0
3.5	15.8	.50	.156	.31	.31	.16	.031	2.5	4.9	4.9	2.5	.5
3.4	20.0	.41	.25	.34	.23	.076	.010	5.0	6.7	4.6	1.5	.2
3.3	25.0	.31	.35	.31	.13	.028	.0024	8.8	7.7	3.2	.2	.1
3.2	31.5	.23	.40	.24	.07	.010	.0006	12.6	7.6	2.2	.3	.0
3.1	39.9	.16	.40	.15	.024	.002	.0001	15.8	6.0	1.0	.0	
3.0	50.0	.11	.34	.085	.008	.0004	.0	17.0	4.2	.4		
2.9	62.5	.077	.28	.046	0.000	0.0000		17.5	2.8	.1		
2.8	79.0	.041	.17	.014				13.4	1.1	0.0		
2.7	100	.023	.105	.004				10.4	.4			
2.6	125	.012	.055	0.000				6.9	.1			
2.5	158	.0072	.035					5.5	0.0			
2.4	200	.0031	.015					3.0				
2.3	250	.0013	.007					1.8				
2.2	315	.0006	.003					.9				
2.1	397	.00024	.0013					.5				
2.0	500	.0001	.0004					.2				
	2400							123	54	27	19	11

Initial P Detections/Channels/Day for $m_b \leq 4.0$

$$123/5 + 54 \times 2/5 + 27 \times 3/5 + 19 \times 4/5 + 11 \times 5/5 \quad \text{or}$$

$$\underbrace{24.6 + 21.6}_{46.2\ \text{"unconfirmed"}} + \underbrace{16.2 + 15.2 + 11}_{42.2\ \text{"confirmed"}} = 88.6$$

$$Pd(m_b) = P\{(m_b - 3.5)/.4\} \quad \text{where } P(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-r^2/2}\, dr$$

Theoretical calculation of confirmed and unconfirmed detections at network of 5 equal stations. P(D = i), probability of exactly i out of 5 detecting. N(i) number of i-station events/day.

TABLE II

Probabilities of Machine Detection Assuming Experienced Analyst As A Standard, Nov.1 and 2 1980 Omitting 1324-20:45 on Nov. 2

	Detected			Not Detected			
	Initial Teleseisms	Local	Later Phase	Initial Teleseisms	Local	Later Phase	"Spike" Problem
	50	19	29	2	15	43	10
(Possible Background)				5	3	2	
Sum				7	18	45	

Total Teleseisms 50 + 7 = 57

Prob. of Detection of Teleseisms 50/57 = 0.88

Total Number of Teleseisms "Possible Background" + "Possible Dram" - 13

Total Not Detected - 5

Prob. of Detection of "Possible Background" + "Possible Dram" 8/13 = 0.62

Prob. of Detection of "Teleseisms" (50 - 8)/(57 - 13) = 42/44 = 0.95

TABLE III

Time Residuals of Teleseismic Analyst Detections Also Detected by Machine

Analyst-Machine

Normal					Poss. Noise, Dram, Earlier Start, Mixed with Dram	
5.6	0.1	0.7	-0.1	0.3	0.3	-0.2
-0.3	0.2	-0.4	-0.4	1.3	4.9	
-0.5	-0.3	-2.4	-0.2		-0.3	
-1.3	0.2	-1.0	-0.5		1.0	
0.2	-0.5	-0.3	-0.3		-0.3	
0.0	-0.8	-0.3	-0.2		-1.8	
0.0	0.0	-0.3	0.3		1.2	
-4.4	-0.4	-0.7	1.6		-0.4	
-0.3	-0.1	-0.4	-0.4		0.5	
Med -0.3 (machine late)					Med +0.3 (machine early)	

APPENDIX II

Program Descriptions and Condensed Listings

Z-Detector

The Z-detector is basically the same as the standard detector except that an optimum S/N**2 filter is used instead of a fixed filter, the STA is computed as the log of the sum of the squared filtered data instead as of the sum of the absolute values, and finally detection is on the Z variable equal to (STA - LTA)/σ. where σ is the standard deviation of the STA about the LTA.

After this introduction, we see the condensed listing of the routine which replaces STA/LTA for the standard detector. Condensation is indicated by three dots in a vertical line. The omitted code can generally be found by reference to the complete listing of the STA/LTA routine for the on-line detector which can also be found in this Appendix. A few lines of code such as type declarations unique to the Z detector are missing from the listing; for a full understanding of the code, the original listings should be consulted. A full discussion of the standard detector can be found in Blandford et al. (1981).

In the calling sequence, 'cigs' is the standard deviation which is passed back to the calling routine, 'dpsb', as are the LTA, QR, etc. The variables 'spec', 'rsp', and 'tsa' are respectively: the noise spectrum updated to use in the optimum filter, the amplitude spectrum of the instrument response, and the spectral shape due to t*. The "np" variables have to do with the length of the FFT window used to compute the optimum filter. In our case np=256, and np2=129, the number of raw data points to be filtered and the dimension of the spectral arrays brought along.

First, the tapers are created, then the initial values are created, using either the standard recursive filter or the optimum filter. We shall assume throughout the following that the optimum filter is used.

Subroutine 'cfil', which may be found following 'STA/LTA', actually applies the filter to the data and is quite straightforward. The filter is applied to the first 256 points of the RAW data. The result is placed in the 'filter' array, the 76th point of which is equivalenced to the first point of the filter array. Since the 'STA' calculation loop (Do 1000) runs over NPTS=150 points beginning at filter (1) it runs only to point 75+150=225 of the 256 filtered points and so there is no problem about running over the end of the filtered data.

Subroutine 'Rfine' has been modified to work with the Z variable instead of the STA/LTA ratio; and thus it is necessary to pass to it the 'cig' value.

The 'LTA' and 'cig' values are computed recursively, just as the 'LTA' value is updated in the standard dector, except that we do not allow the 'cig' to be updated when a large signal arrives, since the transition from signal to niose is not characteristic of the variance of either signal or noise by itself.

After the Do 1000 loop, we have the update of the noise spectrum using the parameters set at the beginning of the routine. The equation 'tsud=1200' sets the exponential updating of the spectrum to 1200 seconds, or 20 minutes. 'dtsud' is set to 15, in accordance with the fact that the spectra are updated only when a new set of 15 seconds worth of data has been read in.

```
        Subroutine Stalta (CHANID, RAW, ICHAN, CHAN, XX,STA,
     &   statim,cigs,LTA, QR, DETN,DETECT,STATUS,
     &  spec,rsp,tsa,irf,np,np2,np2d)
        Parameter (ISR = 10)
        Parameter (S = 20, NS = 3, QP = 3)
c...    STALTA computes the short & long term averages of the
c...    filtered input.  z=(Lta-sta)/cig is compared to
c...    a threshold value to define a detection.
c...    filter values squared and Log taken
c
c...    Initial Setup Section
        .
        .
        .
                   do 1 i=1,np
    1      rawb(i)=1.0
           call taper(rawb,np,.2)
           npt=np/10
           do 2 i=1,npt
    2      tap(i)=rawb(i)
        .
        .
        .
c
c          spectral update time;should eventually be put in control
c          parameters
           tsud=1200.
           dtsud=15.
           e2sud=1.0-exp(-dtsud/tsud)
           sigsud=1.0-e2sud
c
c...    Compute initial LTA and thresholds
c
           if(irf.eq.1)Then
           Call Rfil (NPTS,BCOEF(0,ICHAN(12)),RAW(151),filter)
           else
           Do 17 i=1,np
   17      rawb(i)=RAW(i)
           Call Detrnd(rawb,np,0)
           Call Taper(rawb,np,.1)
           Do 18 i=1,np
           rawc(i)=Cmplx(rawb(i),0.0)
   18      continue
           Call Nlogn(8,rawc,-1.0)
           do 19 i=1,np2
   19      spec(i)=2.*(Real(rawc(i))**2+Aimag(rawc(i))**2)
           call cfil(RAW,np,spec,specr,np2,filteq,rsp,tsa,npt,tap)
           endif
           LTA = 0
           cig=0.
           cigs=0.
           Do 20 i = 1,S*NS
           if(irf.ne.1)then
           LTA=LTA+(filter(i+90))**2
           else
           LTA=LTA+(filter(i))**2
```

```
           endif
   20      continue
           LTA=Log(LTA/(Float(S*NS)))
           If (first) Then
              first = .false.
              Call Header
           Endif
           saqout = .false.
           statim = timei+dtsta
           do 21 i=1,S*NS
           if(irf.ne.1)then
           cigs=cigs+(LTA-Log((filter(i+90))**2))**2
           else
           cigs=cigs+(LTA-Log((filter(i))**2))**2
           endif
   21      continue
           cigs=cigs/Float(S*NS)
           DETECT(4) = 0.0
           DETECT(5) = 0.0
           Call Report (CHANID,statim,'Initial    ',0.0,LTA,
     &          statim,saqout,DETECT(4),curzr)
        Endif
c
c...    Get channel calibration and delay
c
        .
        .
        .
                isud=1
c
c       isud=1 if spectrum to be updated
c
        if(irf.eq.1)then
c
c...    Load up previous filtered data to avoid start-up ring
c...    Filter the data to be used in the STA/LTA calculation
c
        Do 40 i = 1,75
   40      filteq(i) = XX(i)
        Call Rfil (NPTS,BCOEF(0,ICHAN(12)),RAW(76),filter)
        Do 42 i = 1,75
   42      XX(i) = filteq(i+150)
        else
        call cfil(RAW,np,spec,specr,np2,filteq,rsp,tsa,npt,tap)
        endif
c
c...    Set up display times and files
        .
        .
        .
                nsta = 1
        ena=0.
        cig=sqrt(cigs)
        Do 1000 filptr = 1,NPTS,S
        cursta = 0.0
        il = filptr-(S*(NS-1))
```

```
        i2 = filptr+(S-1)
        Do 100 i = il,i2
  100      cursta = cursta+(filter(i))**2
        cursta=Log(cursta/(Float(S*NS)))
        if(cig.eq.0.)cig=0.1
        curz=(cursta-LTA)/cig
        nsta = nsta+1
        .
        .
        .
                Do 110 i = 1,QP-1
           STA(i) = STA(i+1)
  110      QR(i) = QR(i+1)
        STA(QP) = cursta
        big = .false.
        .
        .
        .
        Call Refine (filteq,rawptr, STA(QP-1), LTA, detptr,cig)
        .
        .
        .
                   curzr=log((DETECT(2)-DETECT(3))/cig)
           Call Report (CHANID,statim,STATUS,
     &      DETECT(2),DETECT(3),DETECT(1),saqout,DETECT(4),curzr)
           isud=0
c
c...          if detect don't update spectrum
c
        Endif
c
c...    Update LTA every 'R' STA's;
c
  150      Continue
           If (nsta.eq.R) Then
              If (DETN) Then
                 If (valid) Then
                    LTA = e2sigd*cursta + sigdf*LTA
c                   no cig update if signal start where STA/LTA
c                   very large
                 Else
                    valid = .true.
                 Endif
              Else
                 LTA = e2sig *cursta + sigf *LTA
                 cigs=  e2sig *(LTA-cursta)**2+sigf*cigs
                 cig=sqrt(cigs)
              Endif
              tsrt = TRNON
              tend = TRNOFF
           nsta = 0
           ena=ena+1.
           Endif
c...    Update start and end detection thresholds.
c
           th = tsrt
```

```
          If (DETN) th = tend
1000   Continue
       if(isud.eq.1)then
       ncnt=ncnt+1
       do 999 i=1,np2
       spec(i)=e2sud*specr(i)+sigsud*spec(i)
 999   continue
       endif
       End
```

```
      subroutine cfil(RAW,np,spec,specr,np2,filteq,rsp,tsa,npt,tap)
      complex filtex(256),cf
c     129 is np2
      real RAW(*),spec(*),specr(*),filteq(*),rsp(*),tsa(*),enp,tap(*)
      integer*4 i,np,np2,npt
      do 10 i=1,np
   10 filteq(i)=RAW(i)
      call detrnd(filteq,np,0)
c     call taper(filteq,np,.1)
      do 11 i=1,npt
      filteq(i)=filteq(i)*tap(i)
   11 filteq(np-i+1)=filteq(np-i+1)*tap(i)
      do 20 i=1,np
   20 filtex(i)=cmplx(filteq(i),0.0)
      call Nlogn(8,filtex,-1.0)
      do 25 i=1,np2
   25 specr(i)=Real(filtex(i))**2+Aimag(filtex(i))**2
      filtex(1)=cmplx(0.,0.)
      do 30 i=2,np2
      if(spec(i).eq.0.)spec(i)=spec(128)
      cf=rsp(i)*tsa(i)/spec(i)
      filtex(i)=filtex(i)*cf
      if(i.ne.np2)filtex(np-i+2)=filtex(np-i+2)*cf
   30 continue
      call Nlogn(8,filtex,1.0)
      enp=np
      do 40 i=1,np
   40 filteq(i)=enp*Real(filtex(i))
      return
      end
```

Walsh Detector

The basic idea of the Walsh detector is that, first, the data are frequency filtered as in the standard detector to remove the microseisms, then selected Walsh coefficients are calculated for successive 32 point windows of 10 sps data. The successive windows overlap by 16 points; the Walsh coefficients are prewhitened with a recursively updated noise spectrum, which is updated only when there are no threshold crossings. The sum of the absolute values of the whitened coefficients is calculated and compared to a threshold which, if crossed for two successive windows, results in a detection.

One of the most interesting features of the Walsh detector is the method of setting the threshold. The successive 'STA' values are used to maintain a histogram and the threshold is calculated as the sum of the histogram median plus 'xk' times the difference between the 75% point and the median. 'xk' is typically in the range 3.0-3.7. This procedure for setting the threshold is non-parametric and there is evidence that it is an excellent procedure which should be considered for any detection. it need not be restricted in application to the Walsh detector.

Moving to a consideration of the code, thee parameters S=16 and NS=2 provide for a 32 point window overlapping by 16 points, QP=2 provides for the requirement of 2 successive detection (Q/QP) where Q is set equal to 2 in the "data cards". 'indlo' and 'indhi' are the "indif" bounding indices of the Walsh 'sequences' whose amplitudes are to be calculated for detection. 'bakgln' is the number of overlapping 32 point spectral windows used to compute the detection histograms. In the actual calculations reported here this value is set to 512, amounting to an 'averaging' time of 13.6 minutes. 'xk' is the threshold mentioned above.

The subroutine 'cwalrt' computes the 'indif' Walsh functions 'iy' which are of length 32. The routine 'walspc' computes the coefficients of 'a' of these Walsh functions weighted by the coefficients 'swt'. After the initialization the 'swt' values are updated recursively, near the end of the subroutine.

Until enough time has passed so that the required number of values have been computed to fill the histogram (nh.gt.nhp), the detection is determined using the recursively updated LTA calculated from the STA values going into the histogram. After this time the criteria is (cursta.gt.hld), and 'hld', the threshold, is calculated as discussed above.

```
        Subroutine Stalta (CHANID, RAW, ICHAN, CHAN, XX,STA,
     &   statim,LTA, QR, DETN,DETECT,STATUS)
        Parameter (ISR = 10)
        Parameter (S =16, NS = 2, QP = 2,indlo=8,indhi=21,
     &  bakgln=512,indif=14)
c
c       Walsh Transform version
c        version for single channel, for more channels arrays
c        must be defined in dpsw to save val,time,hld,swt,and
c        other variables, and must be passed through calling
c        sequence.
c
        .
        .
        .
                real swt(indif),hld,a(indif),val(bakgln)
        real time(bakgln)
        integer*2 iy(S*NS,indif),i50,i75,nsta
        .
        .
        .
        c       Initial set up
        .
        .
        .
                do 7 i=1,bakgln
        val(i)=i
    7   time(i)=i
        nh=0
c      setting histogram variables to some initial state
c
        .
        .
        .
                   nhp=bakgln
           i50=(bakgln+1)/2
           i75=(i50+bakgln+1)/2
           i=0
           do 8 k=indlo,indhi
           jj=k-1
           i=i+1
           mm=5
           call cwalrt(mm,jj,iy(1,i))
    8      continue
c
c...    Compute initial LTA and thresholds
c
           Call Rfil (NPTS,BCOEF(0,ICHAN(12)),RAW(151),filter)
           nn=S*NS
           nnn=nn
           nnn2=nn
           do 21 i=1,indif
   21      swt(i)=1.0
           call walspc(filter,nnn,indif,iy,a,swt,cursta)
           do 20 i=1,indif
   20      swt(i)=abs(a(i))
```

```
           cursta=indif
           LTA=indif
        .
        .
        .
                   Call Report (CHANID,statim,'Initial   ',0.0,LTA,
     &          statim,saqout,DETECT(4),val(i50))
        xk=TRNON
        th=2.2
        .
        .
        .
                Do 40 i = 1,75
   40      filteq(i) = XX(i)
        Call Rfil (NPTS,BCOEF(0,ICHAN(12)),RAW(76),filter)
        Do 42 i = 1,75
   42      XX(i) = filteq(i+150)
        .
        .
        .
                nsta = 1
        Do 1000 filptr = 1,NPTS,S
        cursta = 0.0
        il = filptr-(S*(NS-1))
        i2 = filptr+(S-1)
        if(i2.gt.NPTS)il=NPTS-nnn2
        svnf=75
        nnn=nn
        call walspc(filter(il),nnn,indif,iy,a,swt,cursta)
        nsta = nsta+1
        statim = statim + dtsta
        .
        .
        .
                Do 110 i = 1,QP-1
        If ((cursta.ge.th*LTA.and.(nh.le.nhp)).or.((nh.gt.
     & nhp).and.cursta.gt.hld)) Then
           big = .true.
        .
        .
        .
                   Call Report (CHANID,statim,STATUS,
     &      DETECT(2).DETECT(3),DETECT(1),saqout,DETECT(4),i50)
        .
        .
        .
  150     Continue
c
c   update histogram with cursta, and weights swt also
c
              if(big.eq..false.)then
              nhhp=nhp
              call hist(val,time,nhhp,cursta)
              hld=val(i50)+xk*(val(i75)-val(i50))
              nh=nh+1
              do 200 i=1,indif
```

```
 200         swt(i)=e2sig*abs(a(i))+sigf*swt(i)
             endif
          If (nsta.eq.R) Then
       .
       .
       .
             Endif
             tsrt = TRNON
             tend = TRNOFF
          nsta = 0

       End
```

MARS Detector

The general flow in this program is similar to that in the Z-detector program in that blocks of 256 data points are filtered and then analyzed from point 76 to point 225 filteq. First the envelopes, 'e', are calculated fr the 256 point window, using filters with center frequencies 'fc', and Q values qe=6*fc. We use 9 windows which span the frequency range from .875 to 3.125 Hz using the same spacing, 0.25 Hz, as used at NORSAR by Farrell et al (1981). The Q values, qe=6*fc are different from those of 12*fc used by Farrell et al in order to give the same overlap of the bands in frequency as those authors for the successful performance of the MARS detector on the PWY data where the Q's were also 12*fc but the filters were spaced at only 0.125 Hz. qe=6*fc also gives filter response times of about 6 seconds, the probable signal length for this data set which contains signals from smaller events than those on the VSC detection tapes.

After the envelopes are computed using subroutine 'nbe', the envelope maxima in a running window of 2.7 seconds are calculated and it is determined if they lie above a threshold. (This window length is the "dispersion parameter" in the notation of Farrell et al; their value for this parameter was 2.8 seconds. In the present study it is 2.7 secondss because of the method of creating this window. The threshold is parameterized as the amplitude parameter γ --in the program 'gam'--and is set equal to 1.8, the same value as in Farrell et al. The actual threshold for each band is equal to the mean of the maxima for that band plus gam* (standard deviation about the mean of the maxima). The mean and standard deviation ('mue' and 'cige' in the program) are calculated via recursive exponentiation such that the e-folding time is 100x0.9=90 seconds.

The "bandwidth" parameter of Farrell et al was set equal to 1 Hz (k=4) for NORSAR and 0.62 Hz (k=5) for PWY. In this study we select 0.75 Hz (k=3). The k parameter is signified in the present program by 'kd'.

In the program the parameter 'nfc2' is the index of the filter whose envelope values are printed out in the event of a detection and has no other function in the calculations.

```
        Subroutine Stalta (CHANID, RAW, ICHAN, CHAN, XX,STA,
     &   statim,LTA, QR, DETN,DETECT,STATUS)
        Parameter (ISR = 10)
        Parameter (S =9, NS = 3, QP = 3,np=256,nfc=9,kd=3,nfc2=6)
c
c       version for MARS, one channel only
c
        .
        .
        .
                Data fc/3.,2.75,2.5,2.25,2.,1.75,1.5,1.25,1./
c
c...    initial setup section
c
        gam=TRNON
        do 11 i=1,nfc
   11   qe(i)= 6.*fc(i)
        dff=float(ISR/2)/float(np/2)
        al=log(2.)/2.
        alp=sqrt(al/3.14159)
        do 10 i=1,nfc
        do 10 j=1,np/2+1
        f=float(j)*dff
        ex=al*((qe(i)/fc(i))**2)*(f-fc(i))**2
        if(ex.gt.20.)ex=20.
        fil(i,j)=alp*exp(-ex)
   10   continue
           nps=np
           nfcs=nfc
           call nbe(RAW,nps,e,nfcs,fil,nps/2+1)
           do 9 i=1,nfc
           mue(i)=0.
           do 8 j=2,np-1
           if(e(i,j).gt.e(i,j-1).and.e(i,j).gt.e(i,j+1)
     & .and.e(i,j).gt.mue(i))mue(i)=e(i,j)
    8      continue
           mue(i)=mue(i)/1.8
           cige(i)=0.52*mue(i)
    9      continue
           LTA=mue(nfc2)
        .
        .
        .
                   Call Report (CHANID,statim,'Initial   ',0.0,LTA,
     &          statim,saqout,DETECT(4),cige(nfc2))
        .
        .
        .
        c       envelope calculation
        call nbe(RAW,nps,e,nfcs,fil,nps/2+1)
        .
        .
        .
                npkt=100.
        eps=1./npkt
        lmeps=1.-eps
```

```
          Do 1000 filptr = 1,NPTS,S
          cursta = 0.0
          il = filptr-(S*(NS-1))
          i2 = filptr+(S-1)
          nk=0
          Do 100 i = 1,nfc
          ngam(i)=0
          do 100 j=75+il+1,75+i2-1
          if(e(i,j).gt.e(i,j-1).and.e(i,j).gt.
     &    e(i,j+1).and.ngam(i).eq.0)then
           if(i.eq.nfc2)cursta=e(i,j)
           if(e(i,j).gt.(mue(i)+gam*cige(i)))then
           nk=nk+1
           ngam(i)=1
           endif
          mue(i)=eps*e(i,j)+1meps*mue(i)
           if(i.eq.nfc2)LTA=mue(i)
          cige(i)=sqrt(eps*(mue(i)-e(i,j))**2+1meps*cige(i)**2)
          endif
100       continue
          .
          .
          .
                  If (nk.ge.kd) Then
             nks=nk
             big = .true.
          .
          .
          .
                     Call Report (CHANID,statim,STATUS,
     &       DETECT(2),DETECT(3),DETECT(1),saqout,DETECT(4),float(nks))
          .
          .
          .
                  End
```

```
      subroutine nbe(x,np,e,ne,fil,mp)
      integer*4 ne,i,np,mp,ii,ifd
      real x(*),e(ne,np),fil(ne,np/2+1),y(256),s(129),
     &z(256)
      do 110 i=1,np
  110 y(i)=x(i)
      call taper(y,np,.2)
      do 100 ii=1,ne
      do 90 i=1,mp
   90 s(i)=fil(ii,i)
      do 80 i=1,np
   80 z(i)=y(i)
      ifd=2
      call fmphl(z,np,s,mp,ifd)
c
c     have phaseless filter trace
c
      do 70 i=1,np
   70 e(ii,i)=z(i)
      ifd=3
      call fmphl(z,np,s,mp,ifd)
c
c     z is now the hilbert transform
c
      do 60 i=1,np
   60 e(ii,i)=sqrt(e(ii,i)**2+z(i)**2)
  100 continue
      return
      end
```

On-Line Detector

```
Apr  8 11:00 1981  rit.h Page 1


c
c...    Detection Criterion Common
c
c...    NPTS    - No. of DATA points to process
c...    S       - No. of DATA points to average for intermediate sta (Y)
c...    NS      - No. of Y's to average for STA
c...    R       - Interval in STA units between LTA update
c...    Q       - q  in q/q' criterion
c...    QP      - q' in q/q' criterion
c...    TRNON   - Turn-on  Threshold
c...    TRNOFF  - Turn-off Threshold
c...    TSUBC   - Normal LTA Decay Time constant
c...    TSUBCD  - Decay Time constant during detection
c
c...    IDX     - Coherency distance for Spkchk routine
c...    BRATIO  - 'Big Spike' ratio   "    "     "
c...    SRATIO  - 'Small Spike' ratio "    "     "
c
c...    IFLO    - Low FFT_channel for Spectral Ratio Calculation
c...    IFMID   - Middle   "     "     "     "     "
c...    IFHI    - High     "     "     "     "     "
c...    IFCUT   - Cutoff FFT_channel for inverse filtering
c...    SPECT   - Local/Teleseism Discrimination Ratio
c...    BCOEF   - B-coefficients for recursive filters
c
        Real TRNON, TRNOFF, TSUBC, TSUBCD
        Real BRATIO, SRATIO, SPECT, BCOEF(0:6,3)
        Integer*2 IDX, NPTS, R, Q
        Integer*2 IFLO,IFMID,IFHI, IFCUT
        Common /rit/ TRNON, TRNOFF, TSUBC, TSUBCD,
     &   BRATIO, SRATIO, SPECT, BCOEF,
     &   NPTS, R, Q, IDX, IFLO,IFMID,IFHI, IFCUT
c
c   End of rit.h
```

```
c
c...    Time of Day variables
c
c...    TIMEI   - Start time of current 'nsec' buffer
c...    TIMERS  - Start time of run
c...    TIMERF  - Finish "   "  "
c...    TIMETS  - Start time of tape
c...    TIMETF  - Finish "   "  "
c
        Real TIMEI
        Real TIMERS, TIMERF
        Real TIMETS, TIMETF
        Integer IYR,IDOY,IHR,IMN,ISC,ICS
        Integer SYR,SDOY
        Integer FYR,FDOY
c
c...    I/O File Definition variables
c
        Integer TAPFD, LCNT
        Character DRIVNO
        Character DBNAME*7
        Integer*2 RUNSRT(5)
        Integer SAQLUN, EPXLUN
        Character*15 SAQFIL, EPXFIL
        Character*80 COMMNT
        Character*7 SMLSTR
c
        Common /run/ TIMEI,TIMERS,TIMERF,TIMETS,TIMETF,
     &   IYR,IDOY,IHR,IMN,ISC,ICS,
     &   SYR,SDOY, FYR,FDOY,
     &   SAQLUN, EPXLUN,
     &   LCNT, RUNSRT,
     &   TAPFD, DBNAME, COMMNT, SMLSTR,
     &   DRIVNO, SAQFIL,EPXFIL
c
```

```
      Subroutine Stalta (CHANID, RAW, ICHAN, CHAN, XX,STA,
     &  LTA, QR, DETN,DETECT,STATUS)
      Parameter (ISR = 10)
      Parameter (S = 5, NS = 3, QP = 3)
c
c...  STALTA computes the short & long term averages of the
c...  filtered input.  The ratio of STA/LTA is compared to
c...  a threshold value to define a detection.
c
      Include 'rit.h'
      Include 'run.h'
c
      Character*14 CHANID
      Integer*4 ICHAN(*)
      Real*4 CHAN(*)
      Real RAW(*), XX(*), STA(*), LTA
      Integer*2 QR(*)
      Logical DETN
      Real DETECT(6)
      Character STATUS*10
c
      Real calib, delay
      Complex xc1(64), xc2(64)
      Real dt, df, dtlta, dtsta, fltlta
      Real qsum, sumlo, sumhi, fmax
      Real e2sig, sigf, e2sigd, sigdf
      Real statim, rawtim,filtim
      Real xr1(64), xr2(64)
      Real filteq(300), filter(150)
      Equivalence (filter(1),filteq(76))
      Real tsrt,tend,th, cursta
      Integer*2 i,j, i1,i2
      Integer*2 iknt, nsta
      Integer*2 detptr,filptr,rawptr,fftptr
      Logical big, qqp, first
      Logical detsrt, detend, saqout
      Logical drop, BSPK, SSPK,valid
c
      Data first/.true./
      Data filteq /300*0.0/
c
c...  Initial Setup Section
c...  Done only if LTA = -999.9
c
      If (LTA.eq.-999.9) Then
c
c...  Define scaling factors
c
         dt = 1.0/Float(ISR)
         df = 1.0/(64*dt)
         dtsta = S*dt
         dtlta = (R*S)*dt
         e2sig = 1.0 - Exp(-dtlta/tsubc)
         sigf = 1.0 - e2sig
         e2sigd = 1.0 - Exp(-dtlta/tsubcd)
         sigdf = 1.0 - e2sigd
```

```
c
c...    Compute initial LTA and thresholds
c
           Call Rfil (NPTS,BCOEF(0,ICHAN(12)),RAW(151),filter)
           LTA = 0
           Do 20 i = 1,S*NS
   20         LTA = LTA+Abs(filter(i))
           If (first) Then
              first = .false.
              Call Header
           Endif
           saqout = .false.
           statim = timei+dtsta
           fltlta = Float(LTA)/(S*NS)
           DETECT(4) = 0.0
           DETECT(5) = 0.0
           Call Report (CHANID,statim,'Initial   ',0.0,fltlta,
     &           statim,saqout,DETECT(4))
        Endif
c
c...    Get channel calibration and delay
c
        calib = chan(5)
        delay = chan(6)
c
c...    Load up previous filtered data to avoid start-up ring
c...    Filter the data to be used in the STA/LTA calculation
c
        Do 40 i = 1,75
   40      filteq(i) = XX(i)
        Call Rfil (NPTS,BCOEF(0,ICHAN(12)),RAW(76),filter)
        Do 42 i = 1,75
   42      XX(i) = filteq(i+150)
c
c...    Set up display times and files
c...    Filter time delay is 0.3 sec for Rfil
c
        rawtim = timei-15.0+delay
        filtim = timei-15.0+delay-0.3
        statim = timei- 7.5+delay-0.3-dtsta
c
c...    Define thresholds
c
        tsrt = TRNON*LTA
        tend = TRNOFF*LTA
        th = tsrt
        If (DETN) th = tend
c
c...    Set up STA & LTA Storage
c...    Begin Loop over data
c
        nsta = 1
        Do 1000 filptr = 1,NPTS,S
        cursta = 0.0
        i1 = filptr-(S*(NS-1))
        i2 = filptr+(S-1)
```

```
        Do 100 i = il,i2
  100      cursta = cursta+Abs(filter(i))
        nsta = nsta+1
        statim = statim + dtsta
c
c...    Shift detection flag array;
c...    Check for STA/LTA > threshold;
c
        Do 110 i = 1,QP-1
           STA(i) = STA(i+1)
  110      QR(i) = QR(i+1)
        STA(QP) = cursta
        big = .false.
        QR(QP) = 0
        If (cursta.ge.th) Then
           big = .true.
           QR(QP) = 1
        Endif
        qsum = 0
        Do 112 i = 1,QP
  112      qsum = qsum+QR(i)
        qqp = .false.
        If (qsum.ge.Q) qqp = .true.
c
c...    Check for detection:
c...    big = true if last STA > threshold
c...    qqp = true if q/q' STA's > threshold
c...    DETN = true if a detection is in progress
c
        detsrt = .false.
        detend = .false.
c
c...    Detection Start?
c
        If (big .and. .not.DETN .and. qqp) Then
           detsrt = .true.
           valid = .true.
           DETN = .true.
           th = tend
           rawptr = filptr+75
           Call Refine (filteq,rawptr, STA(QP-1), LTA, detptr)
           DETECT(1) =  filtim + (detptr-1)*dt
           DETECT(2) = 0.0
           Do 120 i = 1,QP
                If (STA(i).gt.DETECT(2)) DETECT(2) = STA(i)
  120      Continue
           DETECT(2) = DETECT(2)/Float(S*NS)
           DETECT(3) =  LTA/Float(R*S)
c
c...    Check for detection on data drop-out.
c
           iknt = 0
           drop = .false.
           Do 125 i = (detptr-(ISR/4)),detptr+ISR
                If (RAW(i).ne.0) Then
                   iknt = 0
```

```
                Else
                   iknt = iknt+1
                Endif
                If (iknt.ge.ISR/2) drop = .true.
  125       Continue
            If (drop) Then
                STATUS = 'Drop-out'
                DETECT(4) = -999.99
                valid = .false.
                Goto 150
            Endif
c
c...     Load up FFT buffer
c
            fftptr = detptr-(64/2)
            Do 130 i = 1,64
                j = fftptr+(i-1)
                xrl(i) = RAW(j)
  130       Continue
c
c...     Check for Big spikes
c
            Call Spkchk (64, xrl, IDX, BRATIO, BSPK)
            If (BSPK) Then
                STATUS = 'Spike_1'
                ichan(10) = ichan(10)+1
                valid = .false.
            Endif
c
c...     Detrend & taper the data
c
            Call Detrnd (xrl,64,0)
            Call Taper  (xrl,64,0.1)
c
c...     Load up complex array & do FFT
c
            Do 132 i = 1,64
  132           xcl(i) = Cmplx(xrl(i),0.0)
            Call Nlogn (6,xcl,-1.0)
c
c...     Load up inverse transform buffer
c...     Remove low frequency components from FFT data;
c...     Do inverse transform
c
            Do 134 i = 1, 64
  134           xc2(i) = xcl(i)
c
            xc2(1) = (0.0,0.0)
            Do 140 i = 2,IFCUT
                xc2(i) = (0.0,0.0)
                xc2(64-(i-2)) = (0.0,0.0)
  140       Continue
            Call Nlogn (6,xc2,+1.0)
c
c...     Pull out inverse xformed data
c
```

```
          Do 142 i = 1,64
 142           xr2(i) = Real(xc2(i))
c
c...    Check for Small spike
c
          If (.not.BSPK) Then
               Call Spkchk (64,xr2,1024,SRATIO, SSPK)
               If (SSPK) Then
                  STATUS = 'Spike_2'
                  ichan(11) = ichan(11)+1
                  valid = .false.
               Endif
          Endif
c
c...    Compute the amplitude spectra.
c
          DETECT(5) = 0.0
          fmax = 0.0
          Do 144 i = 1,33
               xr1(i) = Sqrt(Real(xc1(i))**2 + Aimag(xc1(i))**2)
               If (xr1(i).ge.fmax) Then
                  fmax = xr1(i)
                  DETECT(5) = Float(i-1)*df
               Endif
 144      Continue
c
c...    Compute the spectral ratio
c
          sumlo = 0.0
          Do 146 i = IFLO,IFMID
 146           sumlo = sumlo+xr1(i)
          sumhi = 0.0
          Do 148 i = IFMID+1,IFHI
 148           sumhi = sumhi+xr1(i)
c
c...    If valid detection, use Spectral Ratio for Local/Teleseismic
c...    discrimination.  ALL detections with logSR < -0.15 are called
c...    teleseisms.
c
          DETECT(4) = -999.99
          If (sumlo.gt.0.0 .and. sumhi.gt.0.0)
     &     DETECT(4) = Log10(sumhi/sumlo)
          If (DETECT(4).lt.-0.15) valid = .true.
          If (valid) Then
               If (DETECT(4).gt.SPECT) Then
                  STATUS = 'Local'
                  ichan(8) = ichan(8)+1
               Else
                  STATUS = 'Teleseism'
                  ichan(9) = ichan(9)+1
               Endif
          Endif
        Endif
c
c...    Continue Detection?
c
```

```
        If (DETN .and. qqp) th = tend
c
c...    Detection End?
c
        If (.not.big .and. DETN .and. .not.qqp) Then
           detend = .true.
           DETN = .false.
           th = tsrt
           saqout = .false.
           If (STATUS.eq.´Local´ .or. STATUS.eq.´Teleseism´) Then
                saqout = .true.
           Endif
           Call Report (CHANID,statim,STATUS,
     &      DETECT(2),DETECT(3),DETECT(1),saqout,DETECT(4))
        Endif
c
c...    Update LTA every ´R´ STA´s;
c
  150      Continue
           If (nsta.eq.R) Then
              If (DETN) Then
                 If (valid) Then
                    LTA = e2sigd*cursta + sigdf*LTA
                 Else
                    valid = .true.
                 Endif
              Else
                 LTA = e2sig *cursta + sigf *LTA
              Endif
              tsrt = TRNON*LTA
              tend = TRNOFF*LTA
           nsta = 0
           Endif
c
c...    Update start and end detection thresholds.
c
           th = tsrt
           If (DETN) th = tend
 1000   Continue
c
        Return
        End
```