A vocal response time system for use with sentence verification tasks

CHRIS JAMES

University of Western Ontario, London, Ontario, Canada

A software system for the reliable detection of vocal response onset is described. The system was designed specifically for the measurement of vocal response times to speech stimuli presented aurally in a sound field in the presence of some background noise. The response time extraction method described here is robust to masking noise and extraneous sounds that may be included in the subject's recorded response. In addition, the response words do not have to be limited to a small set because the system is able to differentiate the onset of any speech sounds, including low-energy fricatives. The method described here may be implemented with any computer sound system because it relies only on the sound conversion clock for timing accuracy and uses postprocessing of the signal after acquisition for response extraction. The response time extraction technique as currently implemented does not recognize subject's responses but could be incorporated into an automatic speech recognition system.

The system described here was developed for use with sentence verification tasks (SVTs) where vocal rather than manual responses are elicited. An SVT trial involves the subject making some judgment about or producing a response to an auditorily presented speech stimulus (i.e., not verbatim recall). Authors such as Pisoni, Manous, and Dedina (1987) developed such tasks in attempting to measure effort in speech perception in addition to accuracy. Subjects are generally required to respond manually using a button box or alternative in such tasks. To compare manual and vocal response modes properly, it was necessary to devise a highly accurate vocal response time system. In addition, stimuli were to be presented in noise and in a sound field because some subjects would be using assistive-listening devices. (A "sound field" is where, for example, the stimulus is presented via a loud-speaker in a sound-proofed room, not via headphones.) Thus the system had to be robust to extraneous background noise.

A number of articles have described the design and use of voice-activated switches for measurement of vocal-response times, for example, Algarabel, Sanmartin, and Ahuir (1989) and Hawley and Izatt (1992). The principal problem with these designs is that they use an energy detector (or integrator) as the trigger source. If the responses are to be elicited in the presence of background noise,

such methods become unreliable and may fail completely. Figure 1 (fine dotted line) shows the amplitude envelope for the response five (solid line) recorded in some background noise. The energy envelope evolved rapidly at around 400 msec, whereas the word production started at around 300 msec using the zero-crossing rate as a guide (long dashes). The zero-crossing rate is just a count of the zero crossings in the signal over a time frame. This rate is simple to extract and effectively indicates the frequency characteristics of the signal. An energy detector would not have been triggered at the beginning of this word. In addition, the use of low-energy thresholds that might improve the timing accuracy in this case would likely have been exceeded prematurely by background noise. There are other ways of characterizing the signal, such as autocorrelation coefficients, but these will take relatively longer to compute than the zero-crossing rate.

Automatic speech recognition systems may be used to both time and record responses. Boles (1988) described how to use the IntroVoice I system with Apple-Psych to collect and time vocal responses. According to Boles (personal communication, 1994), the IntroVoice I system uses the output of a bank of filters as a trigger signal. Some spectral weighting is applied to allow lower trigger thresholds and perhaps improved noise rejection over a simple energy detector. The response times produced by this system coincide with the end of a period of energy exceeding a threshold, and thus they are indicative of the end of the response, not the onset. In addition, a pause of at least 160 msec is required between single utterances. Boles's use of a tone complex of fixed duration to measure the accuracy of the system does not reflect the fact that one may want to recognize speech that does not begin (and/or end) with voicing (i.e., just yes and no). In addition, the length of responses will vary, and this will introduce errors into the response latencies relative to the

I would like to thank D. G. Jamieson and T. Schneider at the Hearing Health Care Research Unit, University of Western Ontario, for their advice. I am also grateful to D. Boles and the anonymous reviewers, and to the Ontario Ministry of Health and the Natural Sciences and Engineering Research Council of Canada for financial support. Correspondence should be addressed to C. James, Department of Otolaryngology, University of Melbourne, 32 Gisbourne St., East Melbourne 3002, Victoria, Australia (e-mail: jamesc@mail.medoto.unimelb. edu.au).

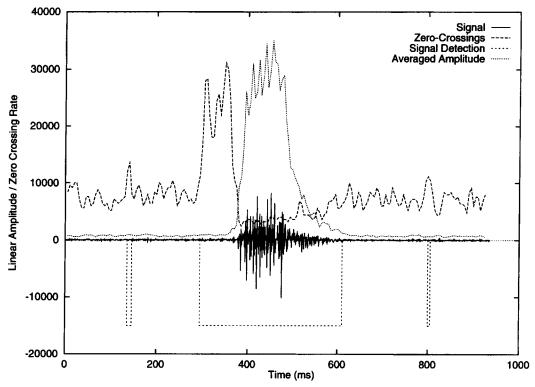


Figure 1. Characterization of a signal in terms of amplitude and zero-crossing rate. The signal in this case is the response word *five*, taken from a recording made in the course of an experiment. The classification of the signal as *signal* or *silence* is also shown.

beginning of production. The system's rejection of background noise and extraneous sounds is also not specified.

The IntroVoice I system is certainly useful for measuring vocal response times when there is no possible interference between the stimulus or conditions and the response. It would be appropriate for use with visual stimuli, and where results are collapsed across different responses so that word lengths may be ignored. In summary, the errors caused by missing weak speech energy at the beginning of the utterance can range into hundreds of milliseconds (as in Figure 1). This is insufficiently accurate for the study of listening effort with SVTs.

REQUIREMENTS

There were various requirements of the system for the present research purposes. It should be (1) automated, reliable, and user friendly; (2) able to cope with a wide range of responses made with free-field stimuli and with interference present (e.g., masking noise): (3) able to integrate with Experiment Control System (ECoS) and Computer Speech Research Environment (CSRE) software (Jamieson, Ramji, Kheirallah, & Nearey, 1992); (4) able to allow response time measurements relative to different parts of the stimulus for interstimulus comparison; and (5) able to produce and record signals with a variety of acquisition platforms.

Here, the focus is on the properties of the response signals and a technique for detecting the onset of the re-

sponse. This is in an effort to satisfy research requirements and overcome the apparent problems with systems that utilize only energy detection to obtain response times. The method outlined here is a modified form of that specified by Rabiner and Sambur (1975; see also Rabiner & Schafer, 1978) to detect the end points of isolated utterances for automatic speech recognition. With this technique, both the energy and zero-crossing rate of the signal are used to define the end points of an utterance. This technique is simpler to use than more advanced techniques, such as those specified by Lamel, Rabiner, Rosenburg, and Wilpon (1981); Wilpon and Rabiner (1987); Savoji (1989); and more recently by Junqua, Mak, and Reaves (1994). Some modifications were made to improve performance under the conditions used in our SVT studies at the Hearing Health Care Research Unit. The resulting algorithm was simple to implement, and the parameter values can be simply related to performance characteristics in terms of precision, speed, and accuracy.

SIGNAL DETECTION

A recording from the microphone can contain several signals: These are the subject's response, the stimulus (at a relatively reduced amplitude), and some background noise, including any mask signals and extraneous sounds. The combination *response* signal is shown in Figure 2 (the stimulus signal is to the left and the subject's re-

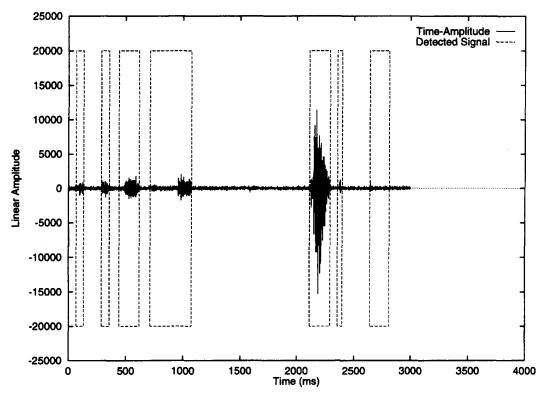


Figure 2. Signal and silence detection using the fragment filter method. The sample includes the stimulus signal (left side) and the subject's response (right side).

sponse is to the right). The first aim in identifying the onset of the vocal response is to identify what constitutes no signal or silence. The recording is segmented into equal-sized, overlapping frames of 10 msec. The mean amplitude and zero-crossing rate of these frames is calculated. The background silence of the recording is characterized by looking for a region, for example, 10 to 20 frames long, of lowest mean amplitude. The mean and standard deviation of amplitude and mean zero-crossing rate of this region are recorded. These values are used to generate a criterion of what is a signal frame and what is a background frame. The signal detection results for the response five in noise are shown in Figure 1 (lower part). One can see how the criterion has picked the response signal starting at an increase in zero crossings. In addition, there are two extraneous regions of signal on either side of the response. Reducing the severity of the criterion will increase sensitivity but also increase the number of these spurious regions. However, these spurious points can be removed by a fragment-filter algorithm, as outlined below.

The automatic detection of the signal in background noise can be likened to a human signal detection task in that we can define a two-dimensional state table where we have two null hypotheses: A portion of the recording was classified (1) as *signal* when it was signal and (2) as noise when it was noise. The alternative hypotheses are that noise was mistaken for signal and signal was mis-

taken for noise. We can increase or decrease the overall number of *signal* classifications by adjusting the criterion for classification of the analysis frames. To remove the error rate in false classifications, we can initially specify a minimum duration for an *on-off signal* period, say, 50 msec. This is similar to an integrator, except that we maintain the actual onset point (not some point where the summed energy is enough to throw the switch). However, if a frame of signal is mistaken for a frame of noise, we may lose the very beginning of the onset of the signal. This occurs for weak fricatives and plosives, especially in background noise.

Here we apply a method of associating short fragments of signal with a portion of signal of sufficient length (fragment filter). We first search for a portion of signal of sufficient length, and then search backward and forward to see whether there are any fragments of signal close enough to be considered part of a signal (say, within 40 msec). We repeat this process until there are no fragments within reach. Thus, short, isolated fragments of signal are ignored without throwing away fragments at the onset of a signal. Some balancing of the silence rejection time (maximum gap between signal fragments) and signal rejection time (minimum duration of a signal fragment, say, 15 msec) is required to optimize the error rate. The response time is taken from the beginning of the detected response. Also imposed is the further criterion that the response signal is the signal with a region of highest mean amplitude; this rejects the stimulus signal if presented in a sound field. The results of this procedure are shown in Figure 2 for a sample containing the stimulus and response recorded free field. We can apply this defragmentation procedure more than once, so that initially, small pieces of signal with small gaps between can be collected together. These composite fragments of signal and silence may in turn be combined to form larger gaps and larger pieces.

IMPLEMENTATION

An outline of the response time extraction procedure is given in the Appendix. This is expressed in pseudo C programming language. Those processes specific to the method presented here are given in detail. For more information on the calculation of zero-crossing rates and energy for speech signals, see Rabiner and Schafer (1978). Each frame of response data is assumed to have been prerecorded and either read from an audio data file or collected in real-time. For each frame of data, the zerocrossing rate and average amplitude are calculated. The lowest mean amplitude across n contiguous frames is obtained via a search. The size of n should be tuned so that a period of background noise n times the frame length should be available in a majority of recordings. The larger the value of n, the better the characterization of the background noise. The average zero-crossing rate is also calculated for these low-energy frames. These parameters are used to characterize background noise or silence. The frame with the largest amplitude is also recorded. The basic criterion for classifying a frame as containing silence or signal is generated on the basis of the noise characterization parameters. The actual thresholds of the criterion may be varied to obtain the best results for a particular implementation. I chose to use a criterion similar to that used by Rabiner and Sambur (1975) after some experimentation. As each frame is classified, contiguous

portions of signal and silence may be marked by start points and end points, as shown in Figure 1 (lower dashed frames). These points are then passed to the fragmentfilter algorithm. The minimum silence and minimum signal durations are specified. New start and end points are produced so that these minima are incorporated. These new points can then be passed again to the fragmentfilter algorithm with new, larger, minimum durations. This process can be repeated several times and serves to concatenate short pieces of signal that would otherwise be classified as silence by the fragment filter. I found that values of 50 and 25 msec for minimum signal duration and minimum silence duration, respectively, gave good results with single-word responses using SVTs in noise. These periods could be increased or decreased depending on the application. For example, if sentences are to be isolated, the minima would be increased. If words in a continuous discourse are to be isolated, the minima would be reduced. There is a tradeoff between speed and accuracy. The longer the analysis frames, the better the characterization. The greater the overlap of frames, the greater the accuracy. Longer frames and greater overlap will increase the time required to obtain the response time. The start of the section of signal containing the greatest mean amplitude is used as the total response time (i.e., from the beginning of the presentation of the stimulus).

The vocal-response time measurement method as outlined above was incorporated into a system for obtaining responses to sentences presented in various levels of noise. An IBM-AT-compatible computer was used to control presentation of stimuli and recording of responses via a Tucker Davis Technologies (Gainesville, FL) AP2 interface board and DD1 signal acquisition module. Recording of the response signal was started synchronously with presentation of the stimulus and continued for up to 6 sec. At this stage, the responses are marked by the experimenter. After each subject has com-

Table 1
Error in Milliseconds for Response Times (RT) Extracted From Problematic Cases Including the Presence of 60-Hz Pickup (at About -25 dB RMS Relative to Signal) and Very Low Energy Frication

Word	Speaker	Problem	RT (msec)	ΔT_{FF} msec	ΔT_{RS} msec
Two	Male 1	Hum	2,022	+3	+3
Four	Male 1	Hum	2,234	+1	-155
No	Male 1	Hum	1,981	+4	-100
Eight	Male 1	Hum	2,104	+4	-144
No	Male 1	Hum	1,910	-6	-136
Four	Female 1	Low frication	1,879	+116	-66
Four	Female 1	Low frication	2,123	+87	+87
Four	Female 2	Low frication	2,030	+20	+106
Four	Female 1	Low frication	2,037	+48	+74
No	Female 1	Clicks	1,325	-20	+20
Yes	Female 1	Clicks	1,518	-38	-3
Mean error/msec				32(36)	81(51)

Note—The response times indicated include the stimulus. The difference between the actual response times and those extracted by the fragment-filter method described herein (ΔT_{FF}) and by Rabiner and Sambur's (1975) method (ΔT_{RS}) are listed.

pleted the task, the recorded responses are processed to produce a label file indicating all regions of signal and the most probable response signal. These response label files are compared with stimulus label files to obtain latencies relative to various parts of the stimuli so that responses to different stimuli can be compared.

EVALUATION

Dummy runs of experiments were used to acquire real data so that the accuracy of the extraction process could be optimized and made robust. Some of these trials were run in quiet and others with a degree of background noise present. The response labels produced by the system for several hundred response signals, including digits and yes and no, seem accurate to the degree of frame overlap (±5 msec in this case). There were a few cases produced in quiet where some 60-Hz pickup was evident, and some responses were produced in fairly low noise (> 30 dB S/N) with very weak frication, which proved to be the most problematic for silence detection. Rabiner and Sambur (1975) recommended high-pass filtering of signals to eliminate the former problem, but I chose to study the performance of the system without filtering as a worst case. Table 1 compares the performance of the system described here (fragment filter, or FF) with that specified by Rabiner and Sambur (1975) (RS) for the problematic cases outlined above. Table 2 does the same for cases recorded with signal-to-noise ratio down to 20 dB. The Rabiner and Sambur method is compared because it might be considered a standard technique for word isolation in automatic speech recognition. It is certainly superior to any technique based only on simple energy detection.

The actual verbal response times (RT) in Tables 1 and 2 were measured using the CSRE editor. One can play any portion of the recorded response at the beginning of the response. The best measurement of the onset may be made by drawing the end of the playback portion back until no phonation can be heard. The time of this position is the total RT and is indicated by the program. Unfortunately, the human auditory system is still the best detector of speech sounds in noise.

The important aspect of the result for the FF method employed here is the relative frequency of errors of large magnitude. The RS algorithm for our purposes is too sensitive to outlying portions of signal that are classified as signal. The RS algorithm searches backward (and forward if searching for an end) from a portion of sufficient amplitude for frames with zero-crossing rates meeting the criterion. If three such frames are found within a certain time, then the start of the earliest of these frames is marked as the start point. In fact, the RS algorithm is not suitable for measuring response latencies between the end of stimulus and onset of response of less than 250 msec, since it searches this far back from the onset of significant energy in the response word. The FF method as implemented here does this up to 40 msec. Both methods could be modified to make use of the characteristics

Table 2
Error in Milliseconds for Response Times (RT)
Extracted From Cases Recorded With One Male Speaker
in Signal-to-Noise Ratio (Speech-Shaped Noise)
Circa 20 dB RMS

Circa 20 db iciris							
Word	Speaker	RT (msec)	ΔT_{FF} (msec)	$\Delta T_{RS}(\text{msec})$			
False	Male 2	1,898	+22	+7			
False	Male 2	3,103	+17	+17			
False	Male 2	1,996	+4	+4			
False	Male 2	1,690	+40	-170			
True	Male 2	1,964	+1	-249			
True	Male 2	2,039	+1	-169			
True	Male 2	1,977	+3	-247			
True	Male 2	2,560	+0	-150			
Three	Male 2	1,619	+1	+1			
Yes	Male 2	2,118	+2	+2			
Six	Male 2	2,505	+15	+15			
Four	Male 2	1,974	+56	-149			
Six	Male 2	1,546	+4	-59			
Yes	Male 2	1,551	-1	-1			
No	Male 2	1,265	+5	+5			
Yes	Male 2	1,019	+1	-175			
Mean er	ror/msec		11(16)	88(91)			

Note—These cases were taken randomly from a set of about 700 responses. The response time indicated is the total response time including stimulus. The errors ΔT_{FF} and ΔT_{RS} from two extraction methods, fragment-filter (FF) and Rabiner and Sambur (1975) (RS), are compared.

of likely response words. For example, yes and no do not start with a long portion of frication whereas four and five do.

CONCLUSION

A system has been described to measure vocal-response times to speech stimuli. Much attention has been given to the structure of the response signals so that the onset of the response is very accurately detected even in the presence of some background or spurious noise. This system could easily be adapted for use with visual or haptic stimuli where vocal responses are to be elicited. The method is currently implemented on an IBM compatible PC and integrated with ECoS and CSRE running on Tucker Davis Technologies hardware. A range of computer audio systems could be used. In addition, speech recognition could be added using proprietary or custom software (since the responses are already isolated). The method of detection used here can also be applied to accurately identify portions of silence in a signal before performing other forms of processing. The system appears reliable and accurate in practice and is currently being used to assess SVT materials for use in measures of listener-effort with hearing aids.

REFERENCES

ALGARABEL, S., SANMARTIN, J., & AHUIR, F. (1989). A voice-activated key for the Apple Macintosh computer. *Behavior Research Methods, Instruments, & Computers*, 21, 67-72.

Boles, D. B. (1988). Voice recognition with the Apple-Psych system.
 Behavior Research Methods, Instruments, & Computers, 20, 158-163.
 HAWLEY, K. J., & IZATT, E. J. (1992). An inexpensive sound activated

switch for experiments measuring vocal response times. Behavior Research Methods, Instruments, & Computers, 24, 523-527.

JAMIESON, D. G., RAMJI, K., KHEIRALLAH, I., & NEAREY, T. M. (1992).
 CSRE: A research environment. In J. J. Ohala, T. M. Nearey, B. L.
 Derwing, M. M. Hodge, & G. E. Wiebe (Eds.), Proceedings of the Second International Conference on Spoken Language Processing (pp. 1127-1130). Edmonton: University of Alberta.

JUNQUA, J.-C., MAK, B. K.-W., & REAVES, B. (1994). A robust algorithm for word boundary detection in the presence of noise. *IEEE Transactions on Speech & Audio Processing*, 2, 406-412.

Lamel, L. F., Rabiner, L. R., Rosenburg, A. E., & Wilpon, J. G. (1981). An improved endpoint detector for isolated word recognition. *IEEE Transactions on Acoustics, Speech & Signal Processing*, 29, 777-785.

PISONI, D. B., MANOUS, L. M., & DEDINA, M. J. (1987). Comprehension of natural and synthetic speech: The effects of predictability on the verification of sentences controlled for intelligibility. *Computer Speech & Language*, 2, 303-320.

RABINER, L. R., & SAMBUR, M. R. (1975). An algorithm for determining the endpoints of isolated utterances. *The Bell System Technical Journal*, **54**, 297-315.

RABINER, L. R., & SCHAFER, R. W. (1978). Digital processing of speech signals. Englewood Cliffs, NJ: Prentice-Hall.

SAVOJI, M. H. (1989). A robust algorithm for accurate endpointing of speech signals. Speech Communication, 8, 45-60.

WILPON, J. G., & RABINER, L. R. (1987). Application of hidden Markov models to automatic speech endpoint detection. Computer Speech & Language, 2, 321-341.

APPENDIX Vocal Response Time Measurement

```
C Code Fragment for Silence/Signal detection by Chris J. James
This code assumes that the digitally recorded response signal is stored in a data file in a 16-bit 2's
complement PCM format, sampling rate fSampleRate.
Variable naming scheme:
x a data buffer defined as short int * x.
dX etc double precision floating point (double).
fX etc single precision floating point (float).
1X etc long integers (unsigned long or signed long).
Where X are hopefully self-explanatory names.
/* Prototypes and enums etc */
// returns average number of crossings per sample in a frame
double FrameMeanZeros (short * audiodata, long lFrameSize);
// returns average absolute amplitude per sample in a frame
double FrameMeanAmp (short * audiodata, long lFrameSize);
// returns arithmetic mean of n double precision numbers
double dMean (double * data, long n);
// returns standard deviation of n double precision numbers
double dStanDev (double * data, long n);
// are we looking for the beginning of silence (FindOff) or signal (FindOn)
enum Find {FindOff, FindOn};
// beginning of fragment
/* OBTAINING THE ZERO-CROSSINGS AND MEAN AMPLITUDE FOR EACH FRAME */
  // loop for lFrames—this is the number of frames
  // in your entire response file (remember to take into account
  // frame overlap.
  for (1=0;1<1Frames;++1)
    // read in a frame of data to x
    read (hFile, (char *)x, (lFrameSize)*sizeof(short));
    // calculate zero-crossings and mean amplitude
    dZeros[1] = FrameMeanZeros (x, lFrameSize)/fFrameSize;
    dAmp[1] = FrameMeanAmp(x, 1FrameSize);
    // what is the current position of the middle of the frame
    // in samples
    1CurPos = (l*(lFrameSize-lOverlap) + lFrameSize/2);
```

```
// when we have sufficient frames look at average
    // across lAvgOver frames.
    if (1>1AvgOver){ //1AvgOver is mentioned as n in article text
       // calculate mean mean amplitude
       dAvgAmp = dMean ((dAmp+l-lAvgOver), lAvgOver);
       // is the new mean lower than the old lowest
       if (dAvgAmp < dLowAmp){
         dLowAmp = dAvgAmp; // store low mean
         // get mean zero-crossings across lAvgOver frames
         dAvgZer = dMean ((dZeros+l-lAvgOver), lAvgOver);
         // and standard deviation
         dSdZer = dStanDev ((dZeros+l-lAvgOver), lAvgOver);
    // is the new mean bigger than the old biggest
    if (dAvgAmp > dHiAmp){
       // store the new biggest
       dHiAmp = dAvgAmp;
       // store the position
       lHiAmpPos = lCurPos;
    // rewind the file by the overlap in samples
    lseek (hFile, (-lOverlap)*sizeof(short), SET_CUR);
  // re-store the lowest mean amplitude for the sample
  // this is the model for the background noise
  dAvgAmp = dLowAmp;
/* OBTAINING THE NOISE REJECTION CRITERION */
  1/ Now choose the threshold parameters as specified by Rabiner
  if (dAvgAmp<0.001) // stop divide by zero next line! arbitrary min
    dAvgAmp=0.001;
  // add 2 SDs of the Zero-crossing rate for the noise, this is
  // the zero-crossing rate criterion.
  dAvgZer += 2*dSdZer;
  dAvgTwe += 2*dSdTwe;
  // multiply the lowest mean amplitude by 4.0
  dLowAmp = 4.0 * dAvgAmp;
  // 3 per cent of the difference between the largest mean amplitude
  // and the lowest plus the lowest
  dAvgAmp = 0.03* (dHiAmp - dAvgAmp) + dAvgAmp;
  // Rabiner's condition on choice of lowest amplitude
  // whichever is the lower is the amplitude criterion
  if(dAvgAmp > dLowAmp)
    dAvgAmp = dLowAmp;
  FindFlag = FindOff; // we look for silence to begin with
  1Cnt=0; // count the number of silence/signal blocks
  for (l=0; l< lFrames; ++1){
    // the actual time in samples
    // (divide by fSamplingRate to obtain time)
```

```
lCurPos = (l*(lFrameSize-lOverlap) + lFrameSize/2);
      // is the frame silence?
      if (dAmp[1] \le (dAvgAmp) && (dZeros[1] \le dAvgZer))
        // are we looking for the start of silence?
        if (FindFlag==FindOff){
          // store the current position
          lOffPos[lCnt] = lCurPos;
          // now we want to find signal
          FindFlag=FindOn;
        }
     // no its signal, are we looking for signal?
     else if (FindFlag==FindOn){
        // store the current position
        IOnPos[ICnt] = ICurPos;
        // now we want to find silence
        FindFlag=FindOff;
        // increment the block counter
        ++lCnt;
     }
/* FRAGMENT FILTERING */
Now we have the beginning (IOnPos) and ends (IOffPos) of blocks of signal stored in arrays. We now pass
these arrays to the fragment filter function called "Gobble". We also provide storage for the "filtered"
On-Off positions, the minimum Signal (lMinOn) and silence (lMinOff) durations and the number of points
we put in. The function returns the number of whole on-off blocks found.
  lNewCnt = Gobble(lOffPos, lOnPos, lMinOff, lMinOn, lCnt, lNewOff, lNewOn);
  // We may wish to re-filter with larger minimum durations
  for (i=0;i<iMaxRepeat;++i){
     for (int j=0;j<iNewCnt;++j){
       IOffPos[j]= INewOff[j];
       lOnPos[j] = lNewOn[j];
     lNewCnt = Gobble(lOffPos, lOnPos, lNewMinOff[i], lNewMinOn[i], lNewCnt, lNewOff, lNewOn);
/* INTERPRETATION OF OUTPUT */
The arrays lNewOn and lNewOff should be stored. The position lHiAmpPos should lie between a pair of
points INewOn[n] and INewOff[n+1]. These points denote the beginning and end of the response signal.
The positions may be converted to times by dividing by the sampling rate of the digitized response signal.
/* The FRAGMENT FILTER FUNCTION */
The working of this function is not immediately obvious! This is the briefest implementation of the
algorithm I could think of.
```

```
long Gobble (long *1Off, long *1On, long lMinOffGap, long lMinOnGap,
    long lPoints, long *lDigOff, long *lDigOn)
{
  long i=1, j=1, k=0, l=0;
  while (i \le (lPoints)) {
    // store the current "off"!
    lDigOff[1] = lOff[k];
    // find the next on period larger than the minimum:
    // a period of signal starts at IOn[x] and ends IOff[x+1]!
    // i.e. ignore small bits of signal for the time being
    while ((lOff[i] - lOn[i-1]) < lMinOnGap && i < lPoints)
       ++i;
    j = i-1;
    // search back for a significant off
    // i.e. if there is less than the minimum off time
    // between bits of signal "join" them up
    while ((lOn[j] - lOff[j]) < lMinOffGap && j > 0)
       -j; //i.e. minus-minus
    k = i;
    // ditto search forward
    while ((lOn[k] - lOff[k]) < lMinOffGap && k < lPoints)
       ++k;
    // store next "on"
    lDigOn[l] = lOn[j];
    ++l;
    i=k+1;
  return 1; // the number of blocks (pairs of points)
```

(Manuscript received July 14, 1994; revision accepted for publication November 28, 1994.)