N87-25891

GENERAL PURPOSE ALGORITHMS FOR CHARACTERIZATION OF SLOW AND FAST PHASE NYSTAGMUS

Charles S. Lessard, Ph.D., P.E.
Associate Professor
Department of Industrial Engineering
Texas A&M University
College Station, TX 77840

In the overall aim for a better understanding of the vestibular and optokinetic systems and their roles in space motion sickness, the eye movement responses to various dynamic stimuli are measured. The vestibulo-ocular reflex (VOR) and the optokinetic response, as the eye movement responses are known, consist of slow phase and fast phase nystagmus.

The specific objective of this study is to develop software programs necessary to characterize the vestibulo-ocular and optokinetic responses by distinguishing between the two phases of nystagmus. The overall program is to handle large volumes of highly variable data (nystagmus waveforms) with minimum operator interaction. The programs include digital filters, differentiation, identification of fast phases, and reconstruction of the slow phase with a least squares fit such that sinusoidal or psuedorandom data may be processed with accurate results. The resultant waveform, slow phase velocity eye movements, serves as input data to the spectral analysis programs previously developed for NASA, Johnson Space Center, Neurophysiology Laboratory to analyze nystagmus responses to psuedorandom angular velocity inputs.

NASA Colleague: Millard Reschke, Ph.D., SB X2381

Introduction

Stimulation of the vestibular system by angular acceleration during head movements results in a reflexive eye movement called nystagmus. The resulting response (nystagmus) resembles a sawtooth waveform of which slow rotation of the eyes to maintain gaze on an object are related to the simulus (slow phase) while rapid resets of the eye position are related to some centering mechanism (fast phase). Quantitative properties of nystagmus are important in the characterization of the vestibular and ocular systems. Thus, the specific objective of this study is to develop the necessary software programs to characterize the vestibular and optokinetic responses to sinusoidal stimuli by distinguishing between the two components of nystagmus.

Background

The methods used to analyze nystagmus vary from manual, to real-time automated processing in the time domain. Massoumnia presented models of the slow and fast phase velocities to establish that the slow and fast phase velocity spectra were superimposed. Thereby, he concluded that it was not possible to distinguish between the slow phase velocity and the fast phase velocity by analysis in the frequency domain (7).

In general, there are two time domain methods for identification of a fast phase event. One method is based on the analytical geometric property that points at which the first derivative waveform equals zero correspond to point at which the position waveform is an extremum (maximum or minimum). This method involves digital filtering and differentiating the ocular position signal prior to detecting the fast event with some velocity criterium. The second method is based on identification of the position waveform extrema by a search algorithm. Then a psuedo position waveform is obtained by connecting straight line segments between maxima and minima. The psuedo position waveform is used to calculate slopes (velocities), position changes, and time durations which in turn, are used to identify the fast phase events (8). An assumption of the method is that the information of value which is contained in the nystagmus can be obtained by replacing the detailed time-varying path of the EOG with straight-line approximations between points of maxima and minima. Seven parameter can be extracted by the method as is shown in figure 1. To date, one major vestibular research laboratory in the United States, out of the seven major laboratories which were selected for this study, uses the psuedo waveform method.

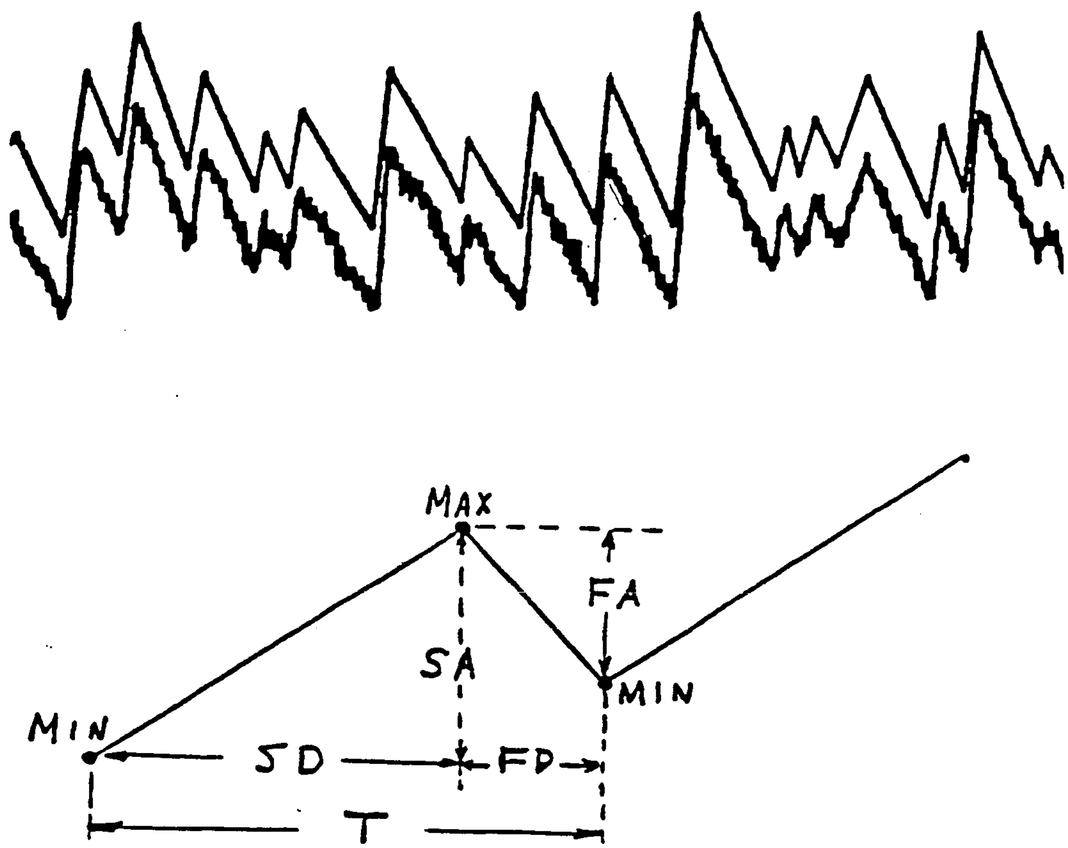

Fig. 1. Upper trace shows the psuedo position EOG waveform superimposed on the raw data. Lower drawing is an expanded version of the straight line approximation between points of extrema (MAX and MIN). SA denotes slow phase amplitude, FA is fast phase amplitude, SV is slow phase velocity, FV is fast phase velocity, SD is the slow phase duration, FD is the fast phase duration and T is the period; where frequency is 1/T.

Characteristic Parameters

In characterizing the slow phase velocity component vestibular researchers have used the engineer's approach to the analysis of a system, i.e., observe the response to a predetermined excitation in order to determine the system's transfer function. Knowledge of the system's transfer function permits prediction of the system response to other deterministic excitations. The most commonly used excitation signals in signal analysis are single frequency sinusoid, step, and impulse functions. The most popular analytical method is the use of a single frequency sinusoidal stimulus which applies only to linear systems analysis. The only parameters that can vary are the magnitude and phase of the resulting sinusoid. The frequency of the input and output are unchanged unless the system is nonlinear. Thus, gain (the ratio of output magnitude to input magnitude) and phase (the delay of signal transmission through the system) at a constant frequency are the only two parameters necessary to characterize the slow phase velocity systems.

Characterization of the fast phase velocity components of nystagmus (saccades) is not obtained by the frequency response method but rather by identifying true saccades, then measuring the maximum velocity of the fast phase segment (saccade), the total change of amplitude from the start to the end of the fast phase segment, and the duration of the saccade. A least square exponential curve fit to the saccade maximum velocity versus total change of the saccade amplitude data results in two best fit coefficients. In addition saccade velocity, accuracy, and reaction time were used test the oculomotor system (2,3,4 and 5).

Current Processing

A general review of the analog and digitizing processes used by the laboratory reveals no agreement on filtering or digitizing as seen in Table I. Most laboratories used DEC LSI-11 computers with 12 bit analog-to-digital converters. The sampling rate at which the EOG data is digitized depends upon the objective of the analysis. If the principle goal is to identify and quantify the characteristics of the fast phase velocity components of nystagmus (saccades) as a means of diagnosing vestibulo-ocular disorders (3), then the data is sampled at 200 samples per second. Whereas, if the primary interest is solely on removal of the fast phases as a means of obtaining a more accurate estimate of the slow phase velocity parameters, then lower sampling rates are used.

The algorithms for processing the eye movements vary in structure primarily because of the ultimate aim of the analysis. Laboratories interested in retaining fast phase

TABLE I

Comparison of Analog Antialiasing Filtering and Digitizing in Vestibular Laboratories

	LAB A	B	C	D	E	F	G
ANALOG							
PREAMPS	IH	IH	IH	IH	IH	IH	IH
FILTER TYPE	Bu	6-pt Bessel	2-pt Bu	none	Bu	Bu	
CUTOFF (Hz)	35	41.6	80	--	35	35	
GAIN	--	--	--	--	--	--	--
DIGITAL							
COMPUTER	PDP 11/34	LSI 11	LSI 11	LSI 11/73	LSI 11/73	LSI 11/73	LSI 11
A/D (Bits)	12	12	12	12	12	12	12
SAMPLING RATE (S/S)	122.8	200	200	200	200	100	120

IH Fabricated in-house

Bu Butterworth filter

S/S Samples per second

information use higher bandwidth digital filters in order to maintain waveform and timing accuracy.

Two laboratories use optimal band limited derivatives (BLD) in lieu of some smoothing routine followed by a two-point central difference equation (CDE). The two-point central difference is popular as a first order differentiator because of its speed, simplicity, accuracy and low pass filtering (1). The laboratories using optimal filtering techniques convolve a finite impulse response (FIR) filter with an finite impulse differentiator to obtain velocity or acceleration filters. TABLE II compares the digital processes for eye movement analysis used by the seven vestibular laboratories.

It is interesting to note that three laboratories identify three classes of fast phase events but only one laboratory uses the saccade velocity information. In the evaluation of the slow phase velocity, most laboratories have algorithms which remove the fast phase events and fill-in the removed points with a linear extrapolation across either the position or velocity waveform. Two laboratories use a least squares sinusoidal fit to the velocity curve without filling the gap between slow velocity segments. Three laboratories have equally spaces intervals after the waveform reconstruction and only two of those laboratories

TABLE II

Comparison of Digital Processes

For Eye Movement Analysis

	Laboratory						
	A	B	C	D	E	F	G
FILTER TYPE	BLD	None	7-pt LP	LP ReF	BLD	None	15-pt LP FIR
CUTOFF (Hz)	--	--	20	35	--	--	25
DERIVATIVES							
1st		2-pt CDE	2-pt CDE	3-pt CDE		2-pt FWD	4-pt CDE
2nd	--	--	--	--	3-pt CDE	--	--
VEL. FILTER CUTOFF	31-pt 25	--	--	--	9-pt FIR	--	--
ACCEL. FILTER CUTOFF	61-pt 30 Hz	-- --	-- --	-- --	-- --	-- --	-- --
EXTRAPOLATION	Lin	Lin	None	None	Lin	Lin	Lin
SLOW PHASE WAVE RECONSTRUCTION	Vel	Vel	None	None	Vel	Pos	Pos
FAST PHASE CLASSIFICATION		3	--	3	3	--	--
LEAST SQUARES TRANSFORM	--	--	Sin	Avg SPV	--	Sin	--
DIRECT FOURIER TRANSFORM	--	yes	--	--	--	--	--
FFT	yes	--	--	--	--	--	yes

CDE Central difference equation

FWD Forward difference equation

FFT Fast Fourier Transform

Vel Velocity

Sin Sinusoidal

Pos Position

Lin Linear

ReF Recursive Filter

use a fast fourier transform to obtain the frequency response parameters.

NASA FPID Program

Development of the fast phase identification (FPID) program began prior to evaluation of the seven U.S. laboratories. The engineering approach dictated the analytical geometric method. Horizontal eye movements (VOR) response to a sinusoidal stimulation of the vestibular system is digitized at 120 samples per second (Fig 2) and filtered with a digital 15-point, low-pass, finite-impulse-response (FIR) filter. The FIR filter cutoff is set at 25 Hz. The signal at 36 Hz is -40.1 db. Several FIR filters and smoothing routines were evaluated before selection of the final 15-point, FIR filter. The filtered EOG (position) signal is shown in figure 3.

The filtered signal is then differentiated with a central difference equation. The program permits the operator to select the polynomial order of the differentiator up to a sixth order central difference equation with error of order (h^6), where h is the sampling interval of 0.00833 seconds (8.33 msec.). Higher order equations result in better accuracy, but increase computation time. Hence, the fourth order (4-point) central difference equation with error of order (h^4) is used to obtain the first derivative of the EOG since it gives

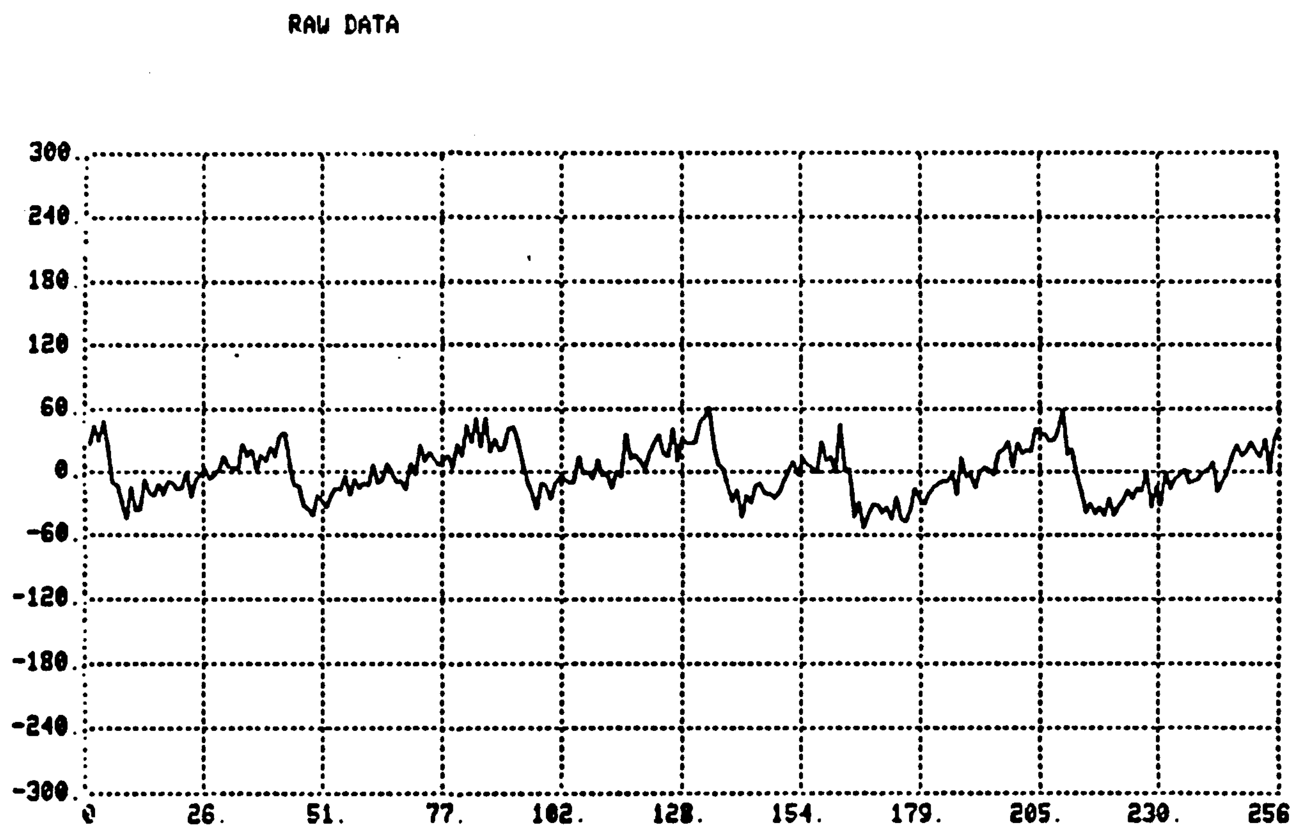

Fig. 2. Raw horizontal eye movements are shown after removal of the D.C. offset.

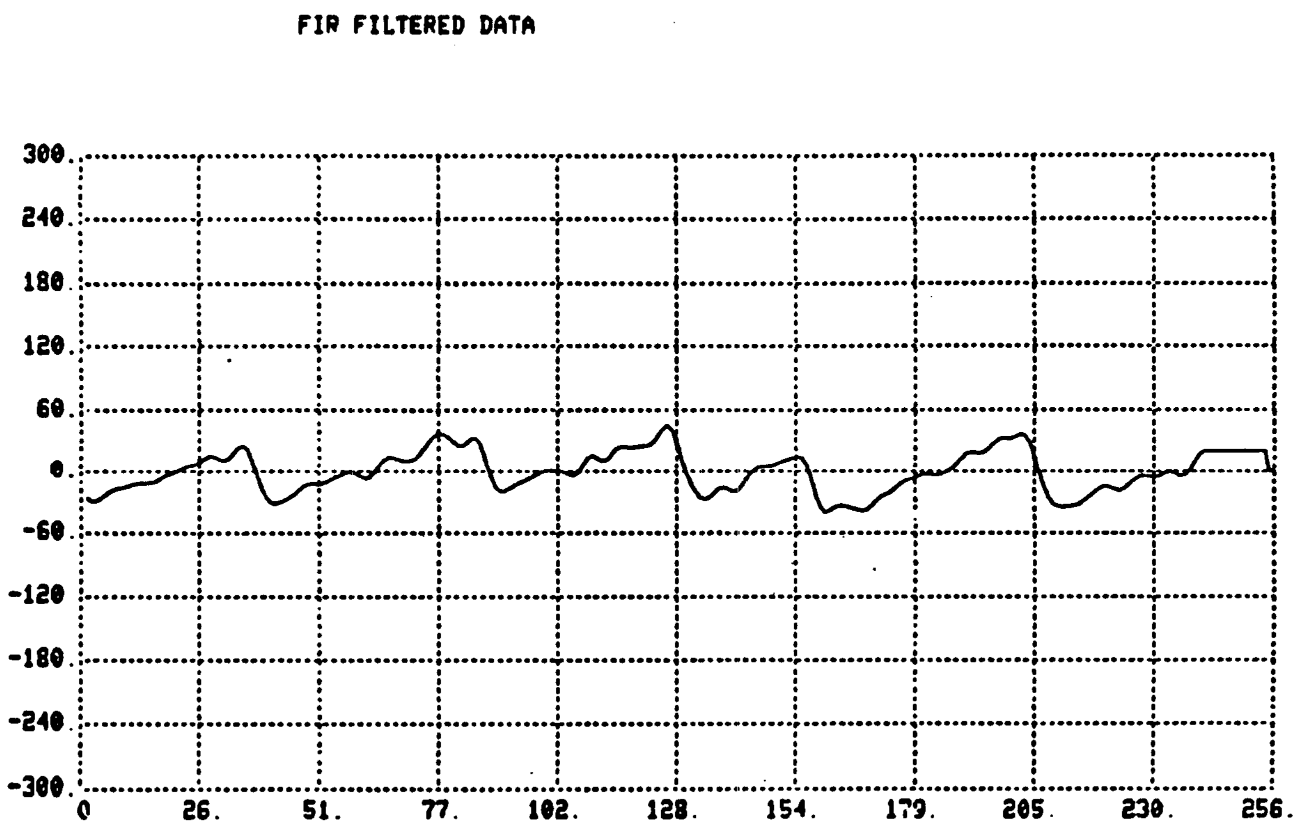

Fig. 3. Horizontal eye movements after FIR filtering.

accurate results with only two points on either side of center (fig. 4).

The second derivative is obtained by differentiating the filtered EOG position signal with a 5-point, second derivative central difference equation of error order (h^4). The second derivative is shown in figure 5. The third derivative of the EOG position signal was also obtained with a 7-point central difference equation (6). The third derivative waveform was noisier than the lower derivative filters so it was removed from the program.

Following differentiation, the program computes the root-mean-square (RMS) value of the various derivatives waveforms in order to set a threshold value above the noise level so as to reduce false detection errors. From RMS values and approximate signal-to-noise ratios of each derivative waveform, detection of a fast velocity event is based on exceeding the first derivative threshold value rather than the second derivative as used by Massoumnia (7). The first derivative offersd the least amount of noise and the best signal-to-noise ratio for threshold detection.

Once the threshold of the EOG velocity signal is exceeded the search point is move backward to find where the derivative zero crossing occurrs. This point is flagged as the beginning of the fast phase event and the search is

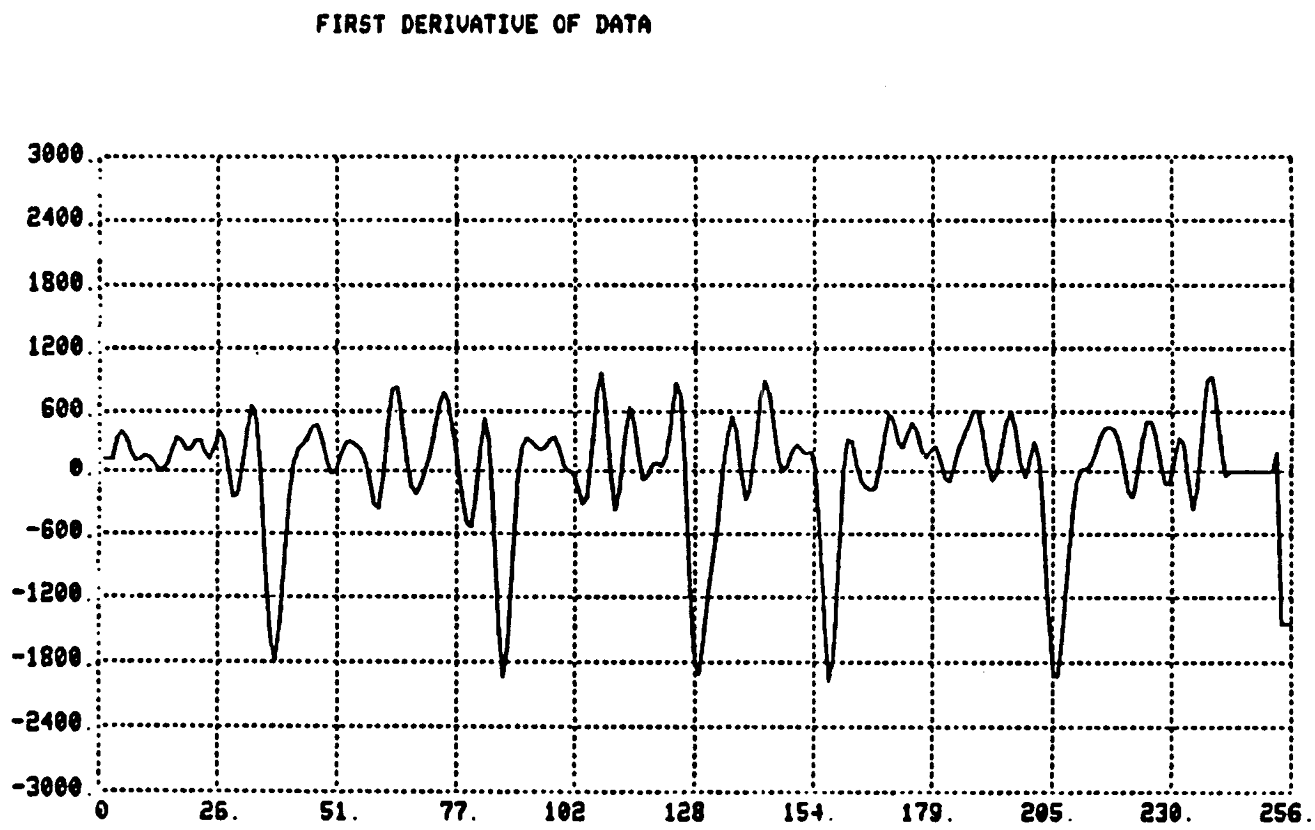

Fig. 4. First derivative of the filtered eye movements. The RMS value is 585 A/D counts for 2.08 seconds of data. The signal-to-noise ratio is 3.3:1.

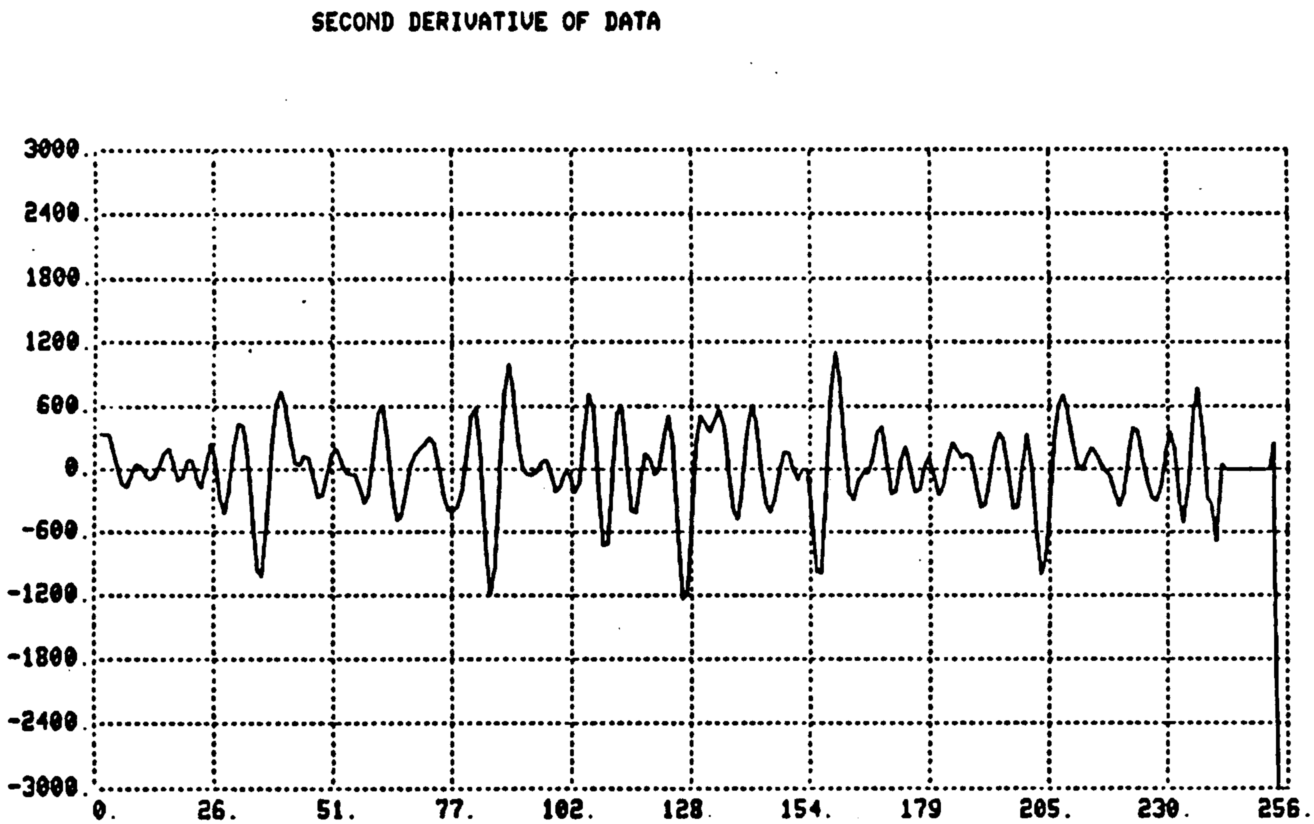

Fig. 5. The second derivative of the filtered eye movements. The RMS value was 485 x 10^2 A/D counts for 2.08 seconds of data. The signal-to-noise ratio is 2.3:1.

reversed (forward direction) until a zero crossing occurrs. This point was flagged as the end of the fast phase event.

In the next steps, the program uses the filtered eye position (EOG) waveform to perform a least squares linear regression on the slow phase velocity segment preceding the starting index of the current fast phase event. Then the points between start and end of the fast phase event are extrapotated and added to the position waveform.

Prior to output of the reconstructed slow phase EOG signal, a correction·of slow segment height is necessary. This is accomplished by obtaining the change of position (height) from the last point of the extrapolated EOG position (end of fast phase index) and the start of the following slow phase segment (end point + 1). Height corrections are cummulative and must carry the proper sign. The reconstructed slow phase EOG position waveform is shown in figure 6. The FPID program listing is given in the appendix.

Conclusion and Recommendations

Although the fast phase identification (FPID) and slow phase reconstruction appear to work, the program needs to be evaluated with various types of VOR and OKN data. At the present time only short segments of data can be analyzed. The program needs a circular buffer, so that long data records can be analyzed. Additionally, the reconstructed

RECONSTRUCTED SLOW PHASE EOG POSITION DATA

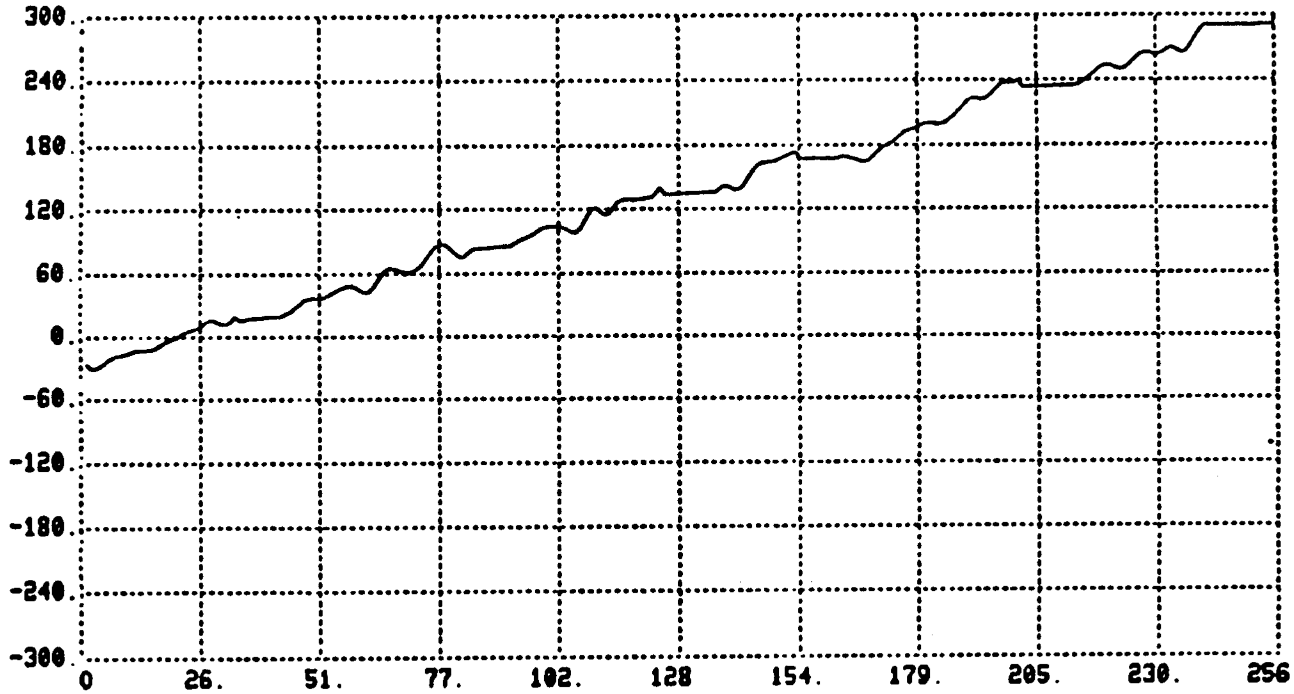

Fig. 6. The reconstructed slow phase horizontal eye movements.

wave must be written to a new data file prior to being used by the spectral programs written for NASA last summer.

REFERENCES

1. Bahill, A.T., et al., "Frequency Limitations of the Two-point Central Difference Differentiation Algorithm", Biol. Cybern. 45:1-4, 1982.

2. Baloh, R.W., et al., "Quantitative Measurement of Saccade Amplitude, Duration and Velocity", Neurology, 25(11):1065-1070, Nov. 1975.

3. Baloh, R.W., et al., "The Saccade Velocity Test", Neurology, 25(11):1071-1076, Nov. 1975.

4. Baloh, R.W., Et al., "Algorithm for Analysis of Saccadic Eye Movements Using a Digital Computer," Aviat., Space, and Environ. Med., 47(5):523-527, May 1976.

5. Baloh, R.W., et al., "Microcomputer Analysis of Eye Movements", Diagnosis 2(4):52-57, July 1984.

6. Hornbeck, R.W., Numberical Methods, QPI Series, Prentice-Hall, Inc. New Jersy, 1975.

7. Massoumnia, M.A., Detection of Fast Phase of Nystagmus Using Digital Filtering, Unpublished Master's Thesis, MIT, May 1983.

8. Wall, C. III, and Black, F.O., "Algorithms for the Clinical Analysis of Nystagmus Eye Movements", IEEE/BME Trans. Biomed. Eng., 28(9):638-646, Sept. 1981.

APPENDIX

FAST PHASE IDENTIFICATION

(FPID)

PROGRAM LISTING

```
          PROGRAM FPID
    C
          INTEGER Z(512),DSKINC,NBLK,IPOSN(16),MAX(16),ICHNUM(16),LENGTH
          REAL AMPL, PERIOD, INTRAT, HEOG(512), XT(512), COEF(15),
         1 YHEOG(512), SDEOG(512), SPEOG(512), SPYHFOG(512), FDEOG(512)
    C
          EQUIVALENCE (AMPL,Z(150)),(PERIOD,Z(152)),(INTRAT,Z(159)),
         1 (DSKINC,Z(167));(NBLK,Z(172)),(ICHNUM,Z(177)),
         2 (IPOSN,Z(225)),(MAX,Z(241))
    C
          COEF(1) = .0037165603
          COEF(2) = .020235427
          COEF(3) = .013399956
          COEF(4) = -.040643737
          COEF(5) = -.066073574
          COEF(6) = .061152663
          COEF(7) = .30294424
          COEF(8) = .43047285
          NEH = 8
          DO 5 N = 1, 7
          COEF(16-N) = COEF(N)
    5     CONTINUE
    C
          HHT = 0.
          THT = 0.
          ISLG = 0
    C
          INTEGER BUFR2(1024), IERR
          BYTE FILNAM(12)
    10    DO 20 K=1,12
    20    FILNAM(K)=0
          TYPE 30
    30    FORMAT (' ENTER COMPLETE FILENAME'/'  FILENAME ?',$)
          ACCEPT 40,FILNAM
    40    FORMAT (12A1)
    C
          CALL DISKIO(FILNAM,-3,Z,256,0,NDUMMY,IERR)
          IF (IERR .NE. 0) TYPE *,' ERROR CODE',IERR,' DURING READING'
    C
    42    TYPE *,' CHAN NO.'
          ACCEPT *,K
    45    TYPE *,' ENTER THE NUMBER OF DATA POINTS FOR EOG'
          ACCEPT *,LENGTH
    C     TYPE *,'   DO YOU WANT FILES LEFT OPEN? (1=YES)'
    C     ACCEPT *,LOPEN
          LOPEN = 1
          TYPE *,' ENTER STARTING BLOCK NUMBER'
          ACCEPT *,IBLK
    C     TYPE *,' DO YOU WANT TO PLOT EVERY POINT? ENTER STEP SIZE!'
    C     ACCEPT *,ISTEP
          ISTEP = 1
    C
          NWRDS = DSKINC* 256
          ITEMP = 0
```

```
48     TYPE 50, NBLK, NWRDS, IBLK,DSKINC
50     FORMAT (' NBLK=',I5,' NWRDS=',I5,' IBLK=',I5,'DSKINC=',I5)
       TYPE 51,MAX(K),IPOSN(K)
51     FORMAT (' MAX(K) = ',I5,' IPOSN(K) = ',I5)
C
       IRMODE=3
       IF (LOPEN .EQ. 1) IRMODE = -3
       TYPE *,'   NOW READING FROM DISK'
C
52     CALL DISKIO(FILNAM,IRMODE,BUFR2,NWRDS,IBLK,NDUMMY,IERR)
       IF (IERR .NE. 0) TYPE *,' ERROR CODE',IERR,' DURING READ'
C
       NPTS = (MAX(K)-IPOSN(K)+1)/ISTEP
       DO 57 L = 1,NPTS
       HEOG(L+ITEMP) = BUFR2(IPOSN(K) + L*ISTEP)
57     CONTINUE
C
       DO 950 I = 1, 26
       TYPE 951, I, HEOG(I)
951    FORMAT (' I = ',I4,' HEOG = ',F12.5)
950    CONTINUE
       DO 960 I = LENGTH-26, LENGTH
       TYPE 961, I, HEOG(I)
961    FORMAT (' I = ',I4,' HEOG = ',F12.5)
960    CONTINUE
       TYPE 962, NPTS, ITEMP
962    FORMAT (' NPTS = ',I8,' ITEMP = ',I8)
       TYPE *,' WAITING TYPE 1'
       ACCEPT*,ISN
C
       IBLK = IBLK + DSKINC
       ITEMP = ITEMP + NPTS
C
       IF (ITEMP .LT. LENGTH) GO TO 52
       XLGT = FLOAT(LENGTH)
       TYPE 50,NBLK,NWRDS,IBLK,DSKINC
C
       DO 60 J = 1,LENGTH
       XT(J) = FLOAT(J)
60     CONTINUE
       TYPE *,' SET YMIN!'
       ACCEPT *,YMIN
       TYPE *,' SET YMAX!'
       ACCEPT *,YMAX
C
       ILFT = 600
       IRIT = 3500
       IBOT = 1000
       ITOP = 2500
       XMIN = 0.
       XMAX = XLGT
C
       TYPE *,' DO YOU WANT TO REMOVE D.C.? (YES=1)'
       ACCEPT *, IYES
```

```
           IF (IYES .NE. 1) GO TO 162
     C
     C     COMPUTE THE MEAN OF THE DATA
           XSUM = 0.0
           DO 58 I = 1, LENGTH
           XSUM = XSUM + HEOG(I)
     58    CONTINUE
           XMEAN =XSUM/XLGT
           DO 59 I = 1, LENGTH
           HEOG(I) = HEOG(I) - XMEAN
     59    CONTINUE
     C
     C
           CALL TSXCHK
           CALL GRINIT(4014,4631,1)
           CALL CHRSIZ(2)
           CALL ERASE
           CALL GRID(10,10,ILFT,IRIT,IBOT,ITOP,97)
           CALL ANOTAT(10,10,ILFT,IRIT,IBOT,ITOP,XMIN,XMAX,YMIN,YMAX)
           CALL XYPLOT(XT,HEOG,LENGTH,ILFT,IRIT,IBOT,ITOP,XMIN,XMAX,
          1 YMIN,YMAX,1,0)
     C
           CALL MPLOT(ILFT+500,ITOP+300,-1)
           TYPE 61
     61    FORMAT('+RAW DATA ')
     C
           TYPE *,' DO YOU WANT A HARD COPY? (YES=1)'
           ACCEPT *,IS
           IF (IS .NE. 1) GO TO 62
     C
           CALL COPY(0)
     C
     62    DO 500 I = 3, LENGTH-2
           YHEOG(I) = .11*(HEOG(I-2)+HEOG(I+2))+.22*(HEOG(I-1)+HEOG(I+1))
          1 + .33*HEOG(I)
     500   CONTINUE
           YHEOG(1) = YHEOG(3)
           YHEOG(2) = YHEOG(3)
           YHEOG(LENGTH) = YHEOG(LENGTH-2)
           YHEOG(LENGTH-1) = YHEOG(LENGTH)
     C
           DO 510 I = 1, LENGTH
           HEOG(I) = YHEOG(I)
     510   CONTINUE
     C
     162   NCO = 15
           TYPE *,' FILTERING DATA WITH 15-POINT FIR FILTER!'
     C
           DO 80 I = 2, LENGTH+NCO
           SUM =0.0
           DO 70 J = 1,NCO
           L = J
           IF (J .GE. I) GO TO 70
           IF (I-J .GT. LENGTH) GO TO 70
```

```
                H = COEF(J) * HEOG(I-J)
                SUM = SUM + H
         70     CONTINUE
                YHEOG(I-1) = SUM
         80     CONTINUE
        C
                J2 = 14
                DO 81 J = 15, LENGTH+J2
                YHEOG(J-J2) = YHEOG(J)
         81     CONTINUE
        C
                DO 630 I = 1, 14
                YHEOG(LENGTH-I-1)=YHEOG(LENGTH-14)
        630     CONTINUE
        C
                CALL TSXCHK
                CALL GRINIT(4014,4631,1)
                CALL CHRSIZ(3)
                CALL ERASE
                CALL GRID(10,10,ILFT,IRIT,IBOT,ITOP,97)
                CALL ANOTAT(10,10,ILFT,IRIT,IBOT,ITOP,XMIN,XMAX,YMIN,YMAX)
                CALL XYPLOT(XT,YHEOG,LENGTH,ILFT,IRIT,IBOT,ITOP,XMIN,XMAX,
               1 YMIN,YMAX,1,0)
        C
                CALL MPLOT(ILFT+500,ITOP+300,-1)
                TYPE 261
        261     FORMAT('+ FIR FILTERED DATA ')
        C
                TYPE *,' DO YOU WANT A HARD COPY? (YES=1)'
                ACCEPT *,IS
                IF (IS .NE. 1) GO TO 83
        C
                CALL COPY(0)
        C
         83     H = FLOAT(ISTEP)/120.
                L = LENGTH
                TYPE *,' SET DESIRED HALF ORDER OF DIFFERENTIATOR:'
                ACCEPT *,NDIF
                GO TO (84,86,88),NDIF
        C
         84     DO 85 I = 2, LENGTH
                FDEOG(I) = (YHEOG(I+1) - YHEOG(I-1))/(H * 2.)
         85     CONTINUE
                FDEOG(1) = FDEOG(2)
                FDEOG(L) = FDEOG(L-1)
                GO TO 1000
        C
         86     DO 87 I = 3, L-2
                FDEOG(I) = ((YHEOG(I-2) - YHEOG(I+2)) + (8.*(YHEOG(I+1) -
               1 YHEOG(I-1))))/(H*12.)
         87     CONTINUE
                FDEOG(1) = FDEOG(3)
                FDEOG(2) = FDEOG(3)
                FDEOG(L) = FDEOG(L-2)
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
      FDEOG(L-1) = FDEOG(L-2)
C
1000  DO 200 I = 3, L-2
      SDEOG(I) = ((-1.)*(YHEOG(I-2) + YHEOG(I+2)) + (16.*(YHEOG(I+1) +
     1 YHEOG(I-1))) - 30.*YHEOG(I))/(12.*H**2.)
200   CONTINUE
      SDEOG(1) = SDEOG(3)
      SDEOG(2) = SDEOG(3)
      SDEOG(L) = SDEOG(L-2)
      SDEOG(L-1) = SDEOG(L-2)
C
      REAL XMFSD
      TYPE *,' SETTING MULTIPLYING FACTOR FOR  2ND DERIVATIVE'
C     ACCEPT *,XMFSD
      XMFSD = .01
C
      DO 400 I = 1, L
      SDEOG(I) = XMFSD*SDEOG(I)
400   CONTINUE
C
      FDRMS = 0.
      SSFD = 0.
      DO 700 I = 1, LENGTH-20
      SSFD = SSFD + FDEOG(I)**2.
700   CONTINUE
      SSMFD = SSFD/FLOAT(LENGTH-20)
      FDRMS = SQRT(SSMFD)
      TYPE 725,FDRMS
725   FORMAT (' RMS OF 1ST DERV. =',F8.3)
C
      SDRMS = 0.
      SSSD = 0.
      DO 750 I = 1, LENGTH-20
      SSSD = SSSD + SDEOG(I)**2.
750   CONTINUE
      SSMSD = SSSD/FLOAT(LENGTH-20)
      SDRMS = SQRT(SSMSD)
      TYPE 775,SDRMS
775   FORMAT (' RMS OF 2ND DERV. =',F8.3)
C
      TYPE *,' WAITING! TYPE 1 TO CONTINUE.'
      ACCEPT*, JES
      GO TO 90
C
88    DO 89 I = 4, L-3
      FDEOG(I) = ((YHEOG(I+3) - YHEOG(I-3)) + (9.*(YHEOG(I-2) -
     1 YHEOG(I+2))) + (45.*(YHEOG(I+1) - YHEOG(I-1))))/(H*60.)
89    CONTINUE
      FDEOG(1) = FDEOG(4)
      FDEOG(2) = FDEOG(4)
      FDEOG(3) = FDEOG(4)
      FDEOG(L) = FDEOG(L-3)
      FDEOG(L-1) = FDEOG(L-3)
      FDEOG(L-2) = FDEOG(L-3)
```

```
      GO TO 1000
C
90    YMIN = YMIN*10.
      YMAX = YMAX*10.
      CALL TSXCHK
      CALL GRINIT(4014,4631,1)
      CALL CHRSIZ(3)
      CALL ERASE
      CALL GRID(10,10,ILFT,IRIT,IBOT,ITOP,97)
      CALL ANOTAT(10,10,ILFT,IRIT,IBOT,ITOP,XMIN,XMAX,YMIN,YMAX)
      CALL XYPLOT(XT,FDEOG,LENGTH,ILFT,IRIT,IBOT,ITOP,XMIN,XMAX,
     1 YMIN,YMAX,1,0)
C
      CALL MPLOT(ILFT+500,ITOP+300,-1)
      TYPE 282
282   FORMAT ('+ FIRST DERIVATIVE OF DATA')
      TYPE *,' DO YOU WANT A HARD COPY? (YES=1)'
      ACCEPT *,IS
      IF (IS .NE. 1) GO TO 283
C
      CALL COPY(0)
C
283   TYPE *,' DO YOU WANT A SINGLE CURVE PLOT? (YES=1)'
      ACCEPT *, IYES
      IF (IYES .NE. 1) GO TO 210
      CALL TSXCHK
      CALL GRINIT(4014,4631,1)
      CALL CHRSIZ(3)
      CALL ERASE
      CALL GRID(10,10,ILFT,IRIT,IBOT,ITOP,97)
      CALL ANOTAT(10,10,ILFT,IRIT,IBOT,ITOP,XMIN,XMAX,YMIN,YMAX)
      GO TO 250
210   IBOT = 1250
      ITOP = 2750
250   CALL XYPLOT(XT,SDEOG,LENGTH,ILFT,IRIT,IBOT,ITOP,XMIN,XMAX,
     1 YMIN,YMAX,1,0)
C
      CALL MPLOT(ILFT+500,ITOP+300,-1)
      TYPE 382
382   FORMAT ('+ SECOND DERIVATIVE OF DATA')
      TYPE *,' DO YOU WANT A HARD COPY? (YES=1)'
      ACCEPT *,IS
      IF (IS .NE. 1) GO TO 92
C
      CALL COPY(0)
C
92    TYPE *,' ANOTHER DIFFERENTIATOR? (YES=1)'
      ACCEPT *,IS
      IF (IS .EQ. 1) GO TO 83
C
      CALL ERASE
      TYPE *,' IDENTIFICATION OF FAST PHASES'
      FDRMSTH = FDRMS*2.
      TYPE 410, FDRMSTH
```

```
0270   410    FORMAT (' 1ST DERV. RMS THRESHHOLD =',F9.3)
       C
0271   110    LTERM = LENGTH - ISLG
0272          DO 112 J = 1, LTERM
0273          IF (ABS(FDEOG(ISLG+J)) .GT. ABS(FDRMSTH)) GO TO 115
0275   112    CONTINUE
       C
0276   115    ITLST = J + ISLG
0277          TYPE 411,ITLST
0278   411    FORMAT (' INDEX TSH EXCD AT I =',I5)
       C
0279          IF (FDEOG(ITLST) .GE. 0.) GO TO 120
0281          TYPE *,' NEGATIVE VALUES START'
0282          DO 117 L = 1, 10
0283          IF (FDEOG(ITLST - L) .GE. 0.) GO TO 124
0285   117    CONTINUE
0286          TYPE *,' POSITIVE VALUES START'
0287   120    DO 122 L = 1, 10
0288          IF (FDEOG(ITLST - L) .LT. 0.) GO TO 128
0290   122    CONTINUE
0291          TYPE *,' NEGATIVE VALUES END'
0292   124    ISTRT = ITLST - L
0293          TYPE 412,ISTRT
0294   412    FORMAT (' STRT NEG INDEX AT I = ',I4)
       C
0295          DO 126 K = 1, 30
0296          IF (FDEOG(ITLST + K) .GE. 0.) GO TO 135
0298   126    CONTINUE
0299          TYPE *,' POSITIVE VALUES END'
0300   128    ISTRT = ITLST - L
0301          TYPE 414, ISTRT
0302   414    FORMAT (' STRT POS INDEX AT I = ',I4)
       C
0303          DO 130 LN = 1, 30
0304          IF (FDEOG(ITLST + LN) .LT. 0) GO TO 134
0306   130    CONTINUE
0307   134    ISTP = ITLST + LN
0308          TYPE 415, ISTP
0309   415    FORMAT (' END POS INDEX AT I = ',I4)
0310          GO TO 136
       C
0311   135    ISTP = ITLST + K
0312          TYPE 416, ISTP
0313   416    FORMAT (' END NEG INDEX AT I = ',I4)
       C
       C      LEAST SQUARES FIT
0314   136    XN = 10.
0315          DXSUM =  0.
0316          DXSSUM = 0.
0317          DYSUM = 0.
0318          DXYSUM = 0.
0319          DO 140 ISL = ISTRT-10, ISTRT
0320          FI2 = FLOAT(ISL)
0321          DXSUM = DXSUM + FI2
```

```
              DXSSUM = DXSSUM + FI2*FI2
              DYSUM = DYSUM + YHEOG(ISL)
              DXYSUM = DXYSUM + FI2*YHEOG(ISL)
        140   CONTINUE
        C
              DENOM = (XN*DXSSUM - DXSUM*DXSUM)
              SLOPE = (XN*DXYSUM - DXSUM*DYSUM)/DENOM
              YINT = (DYSUM*DXSSUM - DXSUM*DXYSUM)/DENOM
              TYPE 417, SLOPE, YINT
        417   FORMAT (' SLOPE =',F12.5,' YINT =',F12.5)
        C
              DO 142 ILF = ISTRT, ISTP
              XILF = FLOAT(ILF)
              YHEOG(ILF) = SLOPE*XILF + YINT
              TYPE *, ILF, YHEOG(ILF)
        142   CONTINUE
        C
              TYPE *,' TO CONTINUE TYPE: 1 '
              ACCEPT*,IX
              CALL ERASE
        C
              TYPE *,' RECONSTRUCT SLOW PHASE'
        C
              HHT = HHT + YHEOG(ISTP) -YHEOG(ISTP+1)
              TYPE 420, HHT
        420   FORMAT(' HEIGHT CORRECTION =',F12.5)
        C
              TYPE 421, ISLG, ISTP
        421   FORMAT (' STRT SP INDX = ,'I4,'STP SP INDX = ',I4)
        C
              DO 145 I = ISLG+1, ISTP
              SPEOG(I) = YHEOG(I) + THT
              TYPE *, I, SPEOG(I)
        145   CONTINUE
        C
              THT = HHT
              ISLG = ISTP
              TYPE 425,ISLG
        425   FORMAT(' NEW SEARCH STARTS AT I = ',I4)
              TYPE *,' TO CONTINUE TYPE: 1'
              ACCEPT *, IXS
        C
              IF (ISLG .LT. LENGTH) GO TO 110
        C
              TYPE *,' PLOT RECONSTRUCTED SLOW WAVE? (YES = 1)'
              ACCEPT *, IYES
              IF (IYES .NE. 1) GO TO 149
        C
              YMIN = YMIN*.1
              YMAX = YMAX*.1
        C
        900   CALL TSXCHK
              CALL GRINIT(4014,4631,1)
              CALL CHRSIZ(3)
```

```
0366          CALL ERASE
0367          CALL GRID(10,10,ILFT,IRIT,IBOT,ITOP,97)
0368          CALL ANOTAT(10,10,ILFT,IRIT,IBOT,ITOP,XMIN,XMAX,YMIN,YMAX)
0369          CALL XYPLOT(XT,SPEOG,LENGTH,ILFT,IRIT,IBOT,ITOP,XMIN,XMAX,
             1 YMIN,YMAX,1,0)
        C
0370          CALL MPLOT(ILFT+500,ITOP+300,-1)
0371          TYPE 882
0372    882   FORMAT ('+ RECONSTRUCTED SLOW PHASE EOG POSITION DATA')
0373          TYPE *,' DO YOU WANT A HARD COPY? (YES=1)'
0374          ACCEPT *,IS
0375          IF (IS .NE. 1) GO TO 149
        C
0377          CALL COPY(0)
        C
        C
        C262  TYPE *,' DO YOU WANT 5-POINT SMOOTHING? (YES=1)'
        C     ACCEPT *,IYES
        C     IF (IYES .NE. 1) GO TO 149
        C
        C     DO 600 I = 3, LENGTH-2
        C     SPYHEOG(I) = .11*(SPEOG(I-2)+SPEOG(I+2))+.22*(SPEOG(I-1)
        C     1 +SPEOG(I+1)) + .33*SPEOG(I)
        C600  CONTINUE
        C     SPYHEOG(1) = SPYHEOG(3)
        C     SPYHEOG(2) = SPYHEOG(3)
        C     SPYHEOG(LENGTH) = SPYHEOG(LENGTH-2)
        C     SPYHEOG(LENGTH-1) = SPYHEOG(LENGTH)
        C
        C375  CALL TSXCHK
        C     CALL GRINIT(4014,4631,1)
        C     CALL CHRSIZ(2)
        C     CALL ERASE
        C     CALL GRID(10,10,ILFT,IRIT,IBOT,ITOP,97)
        C     CALL ANOTAT(10,10,ILFT,IRIT,IBOT,ITOP,XMIN,XMAX,YMIN,YMAX)
        C     CALL XYPLOT(XT,SPYHEOG,LENGTH,ILFT,IRIT,IBOT,ITOP,XMIN,XMAX,
        C     1 YMIN,YMAX,1,0)
        C
        C     CALL MPLOT(ILFT+500,ITOP+300,-1)
        C     TYPE 82
        C82   FORMAT ('+ FIR FILTERED AND 5-PT. SMOOTHED DATA')
        C     TYPE *,' DO YOU WANT A HARD COPY? (YES=1)'
        C     ACCEPT *,IS
        C     IF (IS .NE. 1) GO TO 149
        C
        C     CALL COPY(0)
        C
0378    149   TYPE *,' DO YOU WANT A 15-PT FIR FILTER? (YES=1)'
0379          ACCEPT *,IYES
0380          IF (IYES .NE. 1) GO TO 150
        C
0382          DO 100 I = 2, LENGTH+NCO
0383          DSUM =0.0
0384          DO 95 J = 1, NCO
```

```
          IF (J .GE. I) GO TO 95
          IF (I-J .GT. LENGTH) GO TO 95
          DH = COEF(J) * SPEOG(I-J)
          DSUM = DSUM + DH
 95       CONTINUE
          SPYHEOG(I-1) = DSUM
 100      CONTINUE
C
          DO 101 M = 15, LENGTH+J2
          SPYHEOG(M-J2) = SPYHEOG(M)
 101      CONTINUE
C
          DO 102 I = 1, 14
          SPYHEOG(LENGTH-I-1) = SPYHEOG(LENGTH-14)
 102      CONTINUE
C
          CALL TSXCHK
          CALL GRINIT(4014,4631,1)
          CALL CHRSIZ(3)
          CALL ERASE
          CALL GRID(10,10,ILFT,IRIT,IBOT,ITOP,97)
          CALL ANOTAT(10,10,ILFT,IRIT,IBOT,ITOP,XMIN,XMAX,YMIN,YMAX)
          CALL XYPLOT(XT,SPYHEOG,LENGTH,ILFT,IRIT,IBOT,ITOP,XMIN,XMAX,
         1 YMIN,YMAX,1,0)
C
          CALL MPLOT(ILFT+500,ITOP+300,-1)
          TYPE 104
 104      FORMAT ('+ 15-PT FILTERED SLOW PHASE EOG DATA')
          CALL COPY(0)
C
 150      IBLK = IBLK - 3
          IF( IBLK .LT. NBLK) GO TO 48
C
          TYPE *,' ANOTHER CHANNEL OF DATA? (YES=1)'
          ACCEPT *,JES
          IF (JES .EQ. 1) GO TO 42
C
          TYPE *,' TRY ANOTHER FILE? (1=YES) ? '
          ACCEPT *, MORE
          IF (MORE .EQ. 1) GO TO 10
C
          STOP
          END
```

FORTRAN IV Storage Map for Program Unit FPID

Local Variables, .PSECT $DATA, Size = 042652 (8917. words)

Name	Type	Offset		Name	Type	Offset		Name	Type	Offset	
AMPL	R*4	000454	Eqv	DENOM	R*4	042516		DH	R*4	042550	
DSKINC	I*2	000516	Eqv	DSUM	R*4	042544		DXSSUM	R*4	042474	
DXSUM	R*4	042470		DXYSUM	R*4	042504		DYSUM	R*4	042500	
FDRMS	R*4	042414		FDRMST	R*4	042446		FI2	R*4	042512	
H	R*4	042404		HHT	R*4	042262		I	I*2	042320	
IBLK	I*2	042302		IBOT	I*2	042346		IERR	I*2	042250	
ILF	I*2	042532		ILFT	I*2	042342		INTRAT	R*4	000476	Eqv
IRIT	I*2	042344		IRMODE	I*2	042312		IS	I*2	042374	
ISL	I*2	042510		ISLG	I*2	042272		ISN	I*2	042322	
ISTEP	I*2	042304		ISTP	I*2	042462		ISTRT	I*2	042456	
ITEMP	I*2	042310		ITLST	I*2	042454		ITOP	I*2	042350	
IX	I*2	042540		IXS	I*2	042542		IYES	I*2	042362	
J	I*2	042330		JES	I*2	042444		J2	I*2	042410	
K	I*2	042274		L	I*2	042316		LENGTH	I*2	042246	
LN	I*2	042460		LOPEN	I*2	042300		LTERM	I*2	042452	
M	I*2	042554		MORE	I*2	042556		N	I*2	042260	
NBLK	I*2	000530	Eqv	NCO	I*2	042376		NDIF	I*2	042412	
NDUMMY	I*2	042276		NEH	I*2	042256		NPTS	I*2	042314	
NWRDS	I*2	042306		PERIOD	R*4	000460	Eqv	SDRMS	R*4	042430	
SLOPE	R*4	042522		SSFD	R*4	042420		SSMFD	R*4	042424	
SSMSD	R*4	042440		SSSD	R*4	042434		SUM	R*4	042400	
THT	R*4	042266		XILF	R*4	042534		XLGT	R*4	042324	
XMAX	R*4	042356		XMEAN	R*4	042370		XMFSD	R*4	042252	
XMIN	R*4	042352		XN	R*4	042464		XSUM	R*4	042364	
YINT	R*4	042526		YMAX	R*4	042336		YMIN	R*4	042332	

Local and COMMON Arrays:

Name	Type	Section	Offset	------Size-----	Dimensions
BUFR2	I*2	$DATA	036076	004000 (1024.)	(1024)
COEF	R*4	$DATA	012002	000074 (30.)	(15)
FDEOG	R*4	$DATA	032076	004000 (1024.)	(512)
FILNAM	L*1	$DATA	042076	000014 (6.)	(12)
HEOG	R*4	$DATA	002002	004000 (1024.)	(512)
ICHNUM	I*2	$DATA	000542	000040 (16.)	(16)
IPOSN	I*2	$DATA	000702	000040 (16.)	(16)
MAX	I*2	$DATA	000742	000040 (16.)	(16)
SDEOG	R*4	$DATA	016076	004000 (1024.)	(512)
SPEOG	R*4	$DATA	022076	004000 (1024.)	(512)
SPYHEO	R*4	$DATA	026076	004000 (1024.)	(512)
XT	R*4	$DATA	006002	004000 (1024.)	(512)
YHEOG	R*4	$DATA	012076	004000 (1024.)	(512)
Z	I*2	$DATA	000002	002000 (512.)	(512)

Subroutines, Functions, Statement and Processor-Defined Functions:

Name	Type	Name	Type	Name	Type	Name	Type	Name	Type
ABS	R*4	ANOTAT	R*4	CHRSIZ	R*4	COPY	R*4	DISKIO	R*4
ERASE	R*4	FLOAT	R*4	GRID	R*4	GRINIT	R*4	MPLOT	I*2
SQRT	R*4	TSXCHK	R*4	XYPLOT	R*4				