FILE

Institute of
Oceanographic Sciences
Deacon Laboratory

INTERNAL DOCUMENT No. 343

SWALES Sonic Buoy - Sonic Anemometer spectral and raw data report

C H Clayson & R W Pascal

1994

Natural Environment Research Council

INSTITUTE OF OCEANOGRAPHIC SCIENCES
DEACON LABORATORY

INTERNAL DOCUMENT No. 343

SWALES Sonic Buoy - Sonic Anemometer spectral and raw data report

C H Clayson & R W Pascal

1994

Wormley
Godalming
Surrey GU8 5UB UK
Tel +44-(0)428 684141
Telex 858833 OCEANS G
Telefax +44-(0)428 683066

DOCUMENT DATA SHEET

AUTHOR

CLAYSON, C H & PASCAL, R W.

PUBLICATION DATE

1994

TITLE

SWALES Sonic Buoy - Sonic Anemometer spectral and raw data report.

REFERENCE

Institute of Oceanographic Sciences Deacon Laboratory, Internal Document, No. 343, 25pp. (Unpublished manuscript)

ABSTRACT

During the SWALES experiment in the autumn of 1993, the Sonic Buoy was deployed twice as part of an array of moored instrumentation. On the Sonic Buoy, 3 component wind speed samples were acquired by the Sonic Processor from a Gill Sonic Anemometer; using a sampling rate of 20.83 Hz. These data were spectrally processed to a form which allowed estimation of wind stress to be made, using the dissipation technique. The processed data were saved to an EPROM logger and a selected set of parameters was also sent to the Formatter Processor for amalgamation with the Multimet mean meteorological data. Raw anemometer data were continuously telemetered via a VHF link to shore and occasional records of raw data were also logged on the Sonic Buoy Raw Data Logger.

This data report briefly describes the processes employed in acquisition of the data. It then describes the processes for the recovery of the data from the various source media, the quality control procedures applied and, finally, the resulting output data files.

Appendices include comprehensive details of the software developed for the above processes and of the formats used for the input and output data.

KEYWORDS

ISSUING ORGANISATION

Institute of Oceanographic Sciences
Deacon Laboratory
Wormley, Godalming
Surrey GU8 5UB. UK.

Director: Colin Summerhayes DSc

Telephone Wormley (0428) 684141
Telex 858833 OCEANS G.
Facsimile (0428) 683066

Copies of this report are available from: ***The Library,*** *PRICE* *£0.00*

Index

SWALES SONIC BUOY - SONIC ANEMOMETER SPECTRAL AND RAW DATA REPORT

Equipment

The Sonic Processor is fully described in the handbook "Sonic Buoy Sonic Processor", ref.1. It consists of a PC-compatible processing system, using DSP ECAT and ECAT-X Eurocard-format boards. These are mounted on a motherboard of the RCA microboard format, together with a 16 Mbyte IOSDL EPROM logger and interfaces on an additional circuit board (SEROPT rev. 2).

Briefly, the Sonic Processor is designed to control and communicate with a Gill Ultrasonic anemometer, to acquire and spectrally process 12,288 samples of anemometer data at quarter-hour intervals, and to output a short parameters message to the Formatter (ref. 2). The processor also outputs the spectrum and parameters to the EPROM logger at quarter-hour intervals. The EPROM logger writes all spectra and parameters to a single file, which is set up at the time the equipment is being prepared for use.

The Onboard (as opposed to Telemetered) Raw Sonic Data Logger is fully described in the handbook "Sonic Buoy Sonic Raw Data Logger", ref. 3. The Raw Sonic Data Logger consists of a PC-compatible processing system, using a DSP GCAT 3000 and 2000 boards on an IOSDL BMPPROC2 motherboard. It is designed to acquire a "10 minute" sample of prompted raw data from the Sonic anemometer and to save this as a FASTCOM-format file on a 4 Mbyte Flash EEPROM memory card at intervals of 2 days.

Data Sources and Processing

The data parameters sent serially to the Formatter were combined with Multimet mean meteorological data by the Formatter and were telemetered via the ARGOS and Meteosat satellite systems to IOSDL. Details of the acquisition and processing of such data have been described in the report on the Formatter data, ref. 4, and will not be repeated here.

Figure 1. Sonic Buoy Onboard Sonic Data Acquisition/Processing

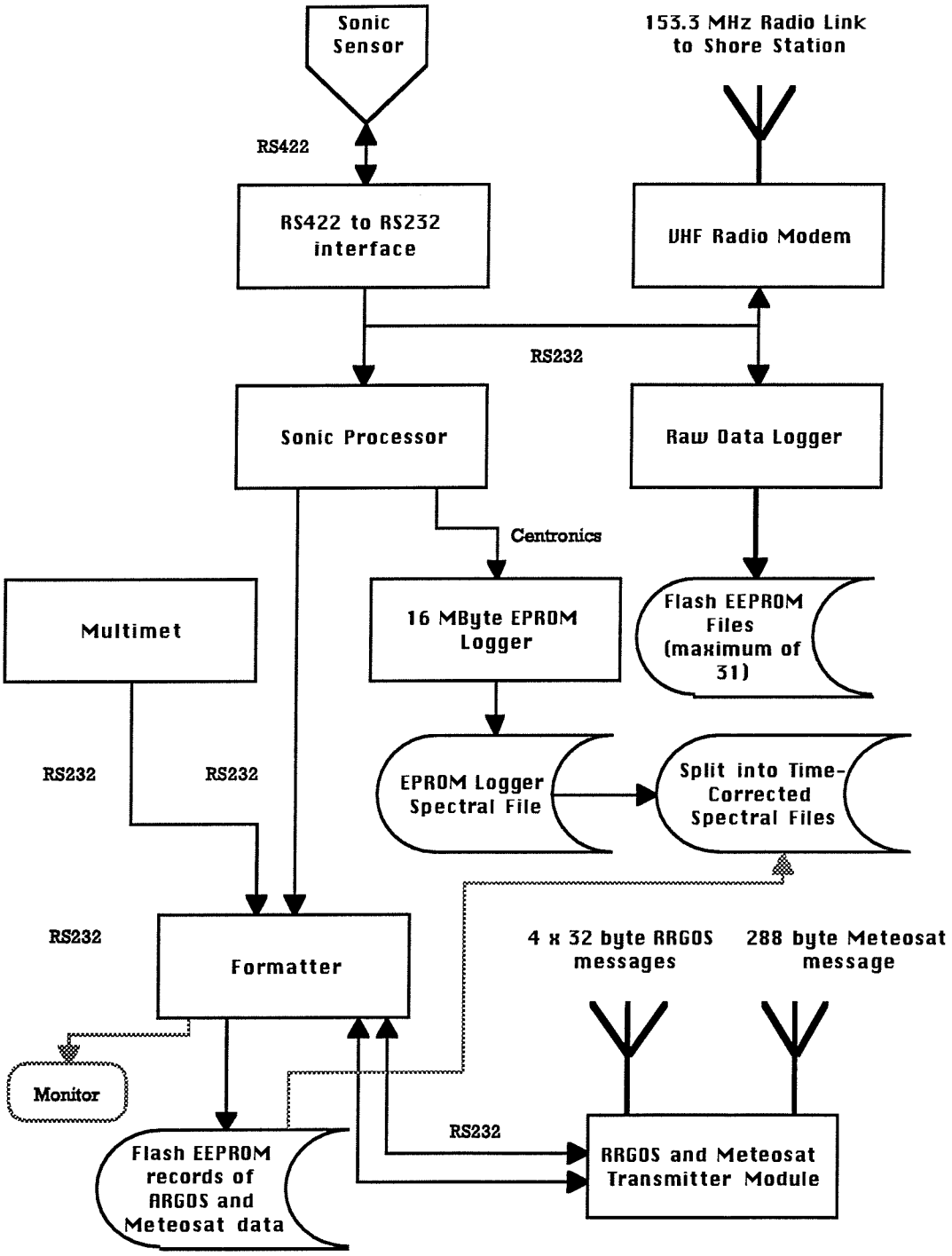


Figure 1, above, gives an overall view of the data sources and processing relevant to data from the Sonic anemometer.

The Sonic Buoy was deployed for two separate periods:

Day 293.59	(1st deployed)	to Day 315	(recovered inverted)
Day 326.60	(re-deployed)	to Day 355	(recovered from rocks)

During the 1st deployment, the buoy overturned at day 313.58. During the 2nd deployment, the buoy systems progressively failed due to premature exhaustion of the buoy's primary batteries from approximately day 338.53.

EPROM Logger Data

The complete data set was recovered from the EPROM logger after each deployment and transferred via RS232 into the hex ASCII PC files given below:

SONIC2.TXT Day 285 1315 hrs to day 313 1345 hrs

SON325.TXT Day 325 1330 hrs to day 338 2015 hrs

These were decoded, by means of the QBASIC application DECODE.BAS (see Appendix A.1), into the ASCII files:

SONIC2.DOC

SONIC325.DOC

The ASCII spectral data files have the format described in Appendix B.

For convenience in handling, the SONIC2.DOC file was split into 5 shorter files; these were named:

SON293.DOC

SON297.DOC

SON301.DOC

SON305.DOC

SON309.DOC

SONIC325.DOC was corrected for the incorrect time stamps resulting from Sonic Processor clock shifts, as described in *Data Time Stamping* below. The corrected spectral data for the second deployment period were then split into 3 shorter ASCII files for convenience; these were named:

SON325.DOC

SON330.DOC

SON335.DOC

The program SONPARAMS.BAS (Appendix A.4) was then used to create 3 tabular files of Day, PSD, MWS, NWS, EWS, VWS and Fit_A, from these files, TAB325.DOC, TAB330.DOC and TAB335.DOC. Data from these files were used to fill in the gaps in the Formatter files. The resulting CricketGraph files are:

1st Deployment CG Final

2nd Deployment CG Final

PCMCIA Flash Card Raw Data

The complete data set was recovered from the 4 Mbyte Flash Memory Card after each deployment. The card contents were dumped to a PC disk file, using a Databook ThinCard drive and associated software. The resulting 4 Mbyte file was then split into 4 x 1 Mbyte files by the C application 4MTO1M.EXE (source code in Appendix A.2) to allow easier handling and transfer to other machines; the resulting PC files were named RAWSWAL.1MG, RAWSWAL.2MG, RAWSWAL.3MG, RAWSWAL.4MG

The format of the Flash Card data is described in Appendix C.

The raw data "records" available are:-

1st Deployment

Day Number	Time of END of prompted data (Formatter Clock Time)
288	1210
290	1210
292	1211
294	1211
296	1212
298	1212
300	1213
302	1213
304	1214
306	1215
308	1215
310	1216
312	1216

2nd Deployment

Day Number	Time of END of prompted data (Formatter Clock Time)
330	1216
332	1216
334	1217
336	1217
338	1218

Data Quality Checking

A simple check on the spectra was made by calculating the standard deviation of the PSD over the frequency range 2 - 4 Hz. This was typically in the range 0.07 - 0.12 for "good" spectra, but rose to as much as 0.4 intermittently. Spectra with standard deviations of more than 0.15 were flagged by the QBasic application SONSELEC.BAS (Appendix A.5), which also plotted the spectra to the VDU. It was found that the "bad" spectra were contaminated by a number of spikes, at about 0.13, 3.85 and 8.18 Hz; these contaminated spectra are shown in figures 2 - 5. This implies that there was an occasional problem with the sensor, possibly rain. No "bad" spectra were flagged during the second deployment of the buoy (with a different sensor).

Note that the selection of a standard deviation limit of 0.15 was fairly arbitrary, based on inspection of plots of standard deviation of PSD (2 - 4 Hz range) over the whole of the two deployment periods (figures 6 - 13).

Figure 2. Suspect Spectra, day 302

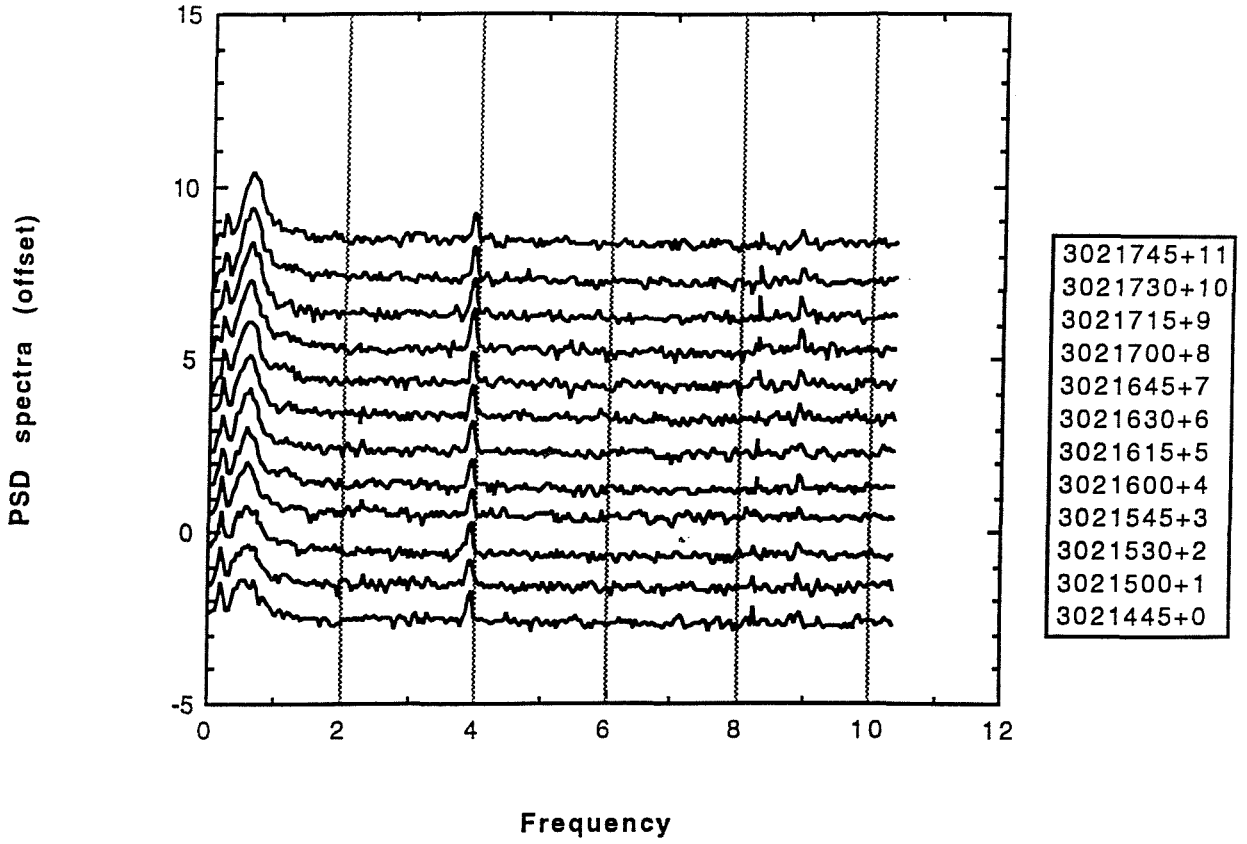


Figure 3. Suspect Spectra, day 309

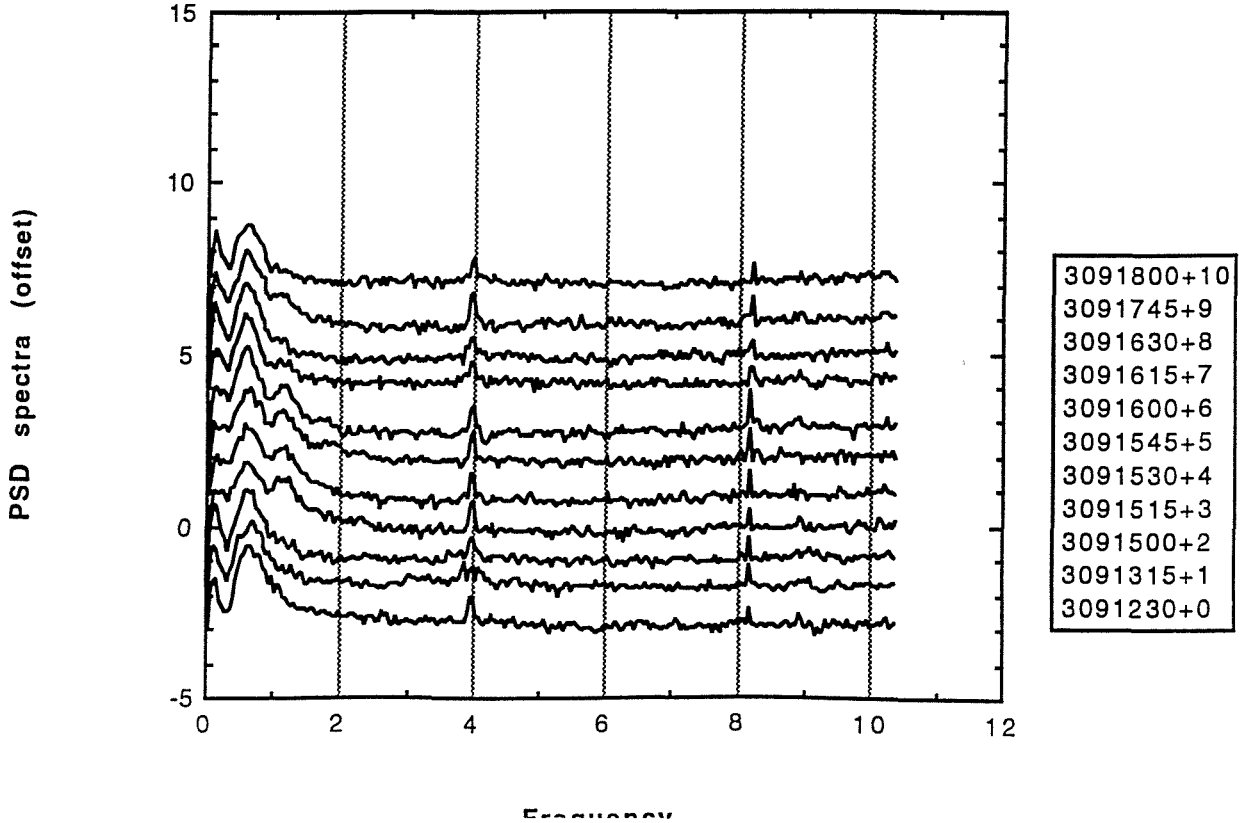


Figure 4. Suspect Spectra , day 311 0615-1100

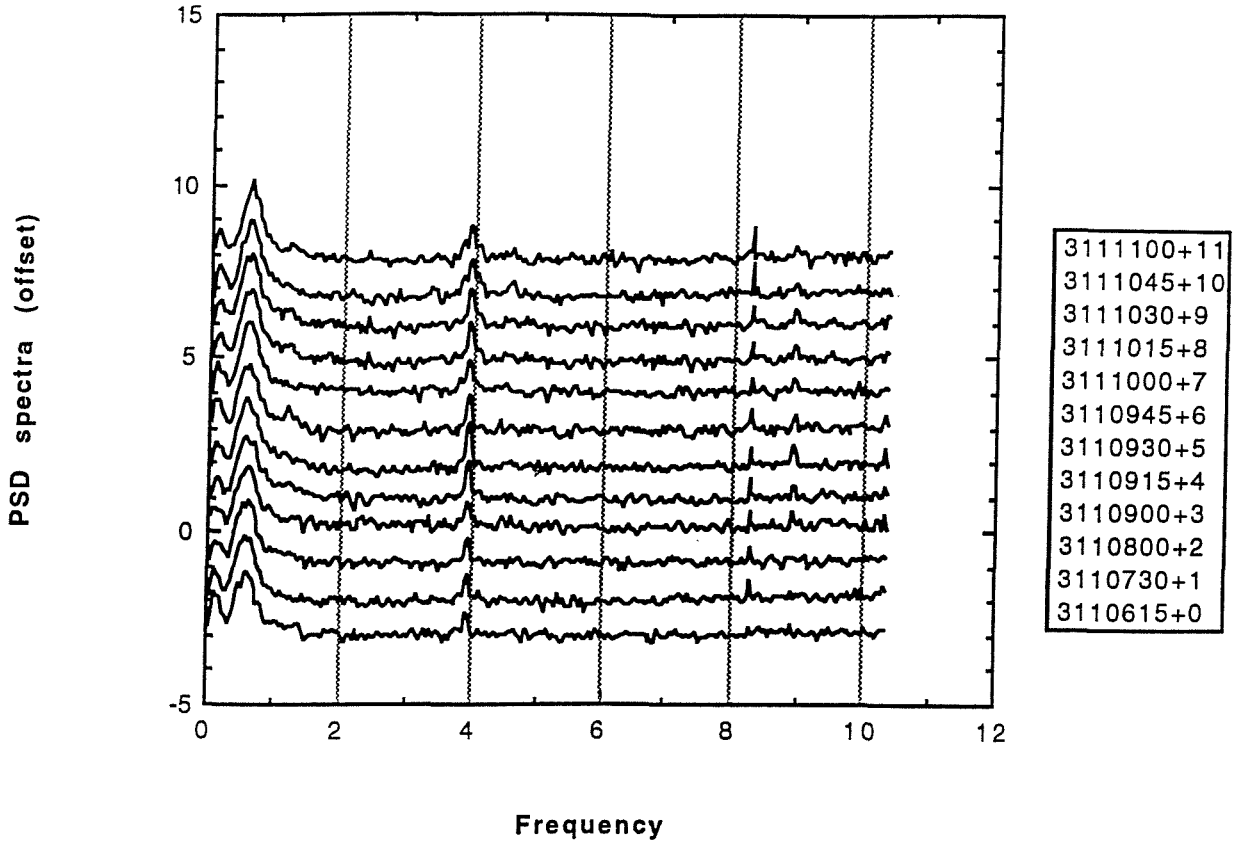


Figure 5. Suspect Spectra, day 311 1115-2115

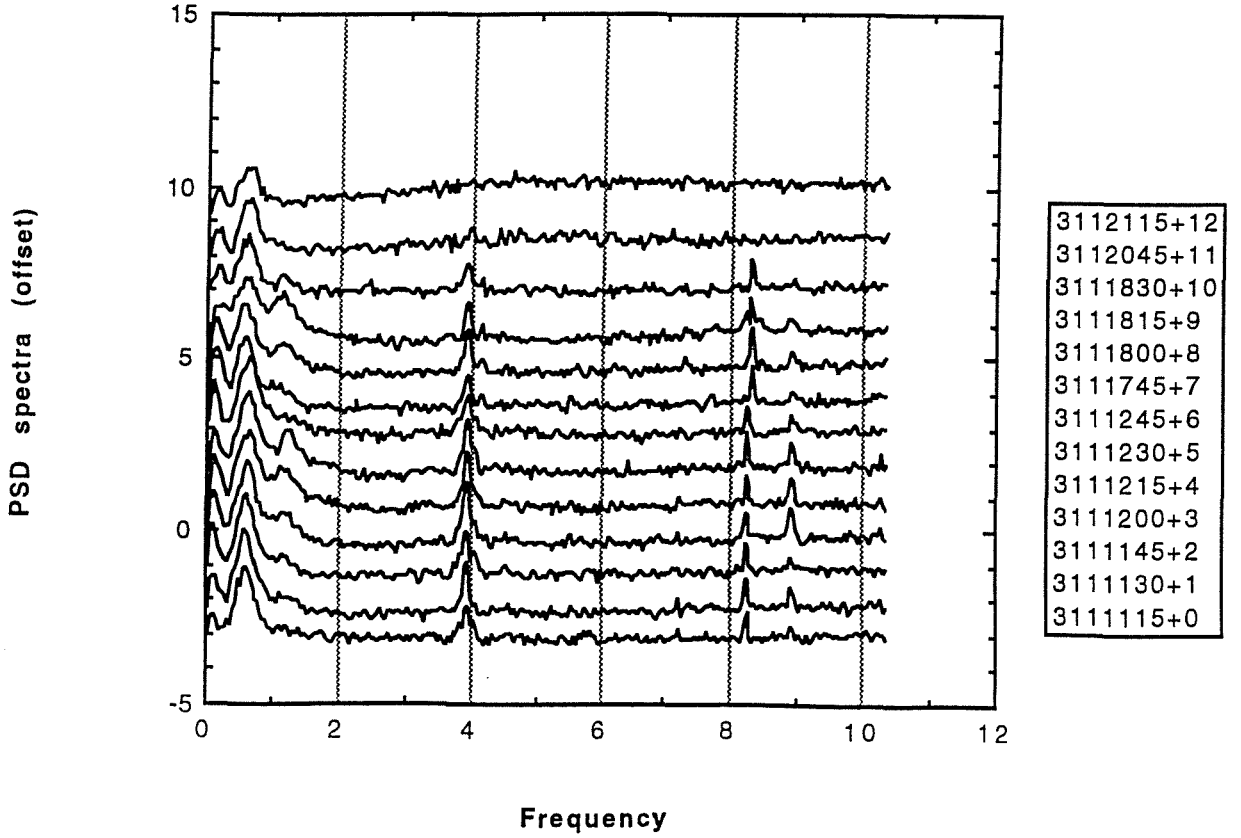


Figure 6. Data from "SON293.VAR"

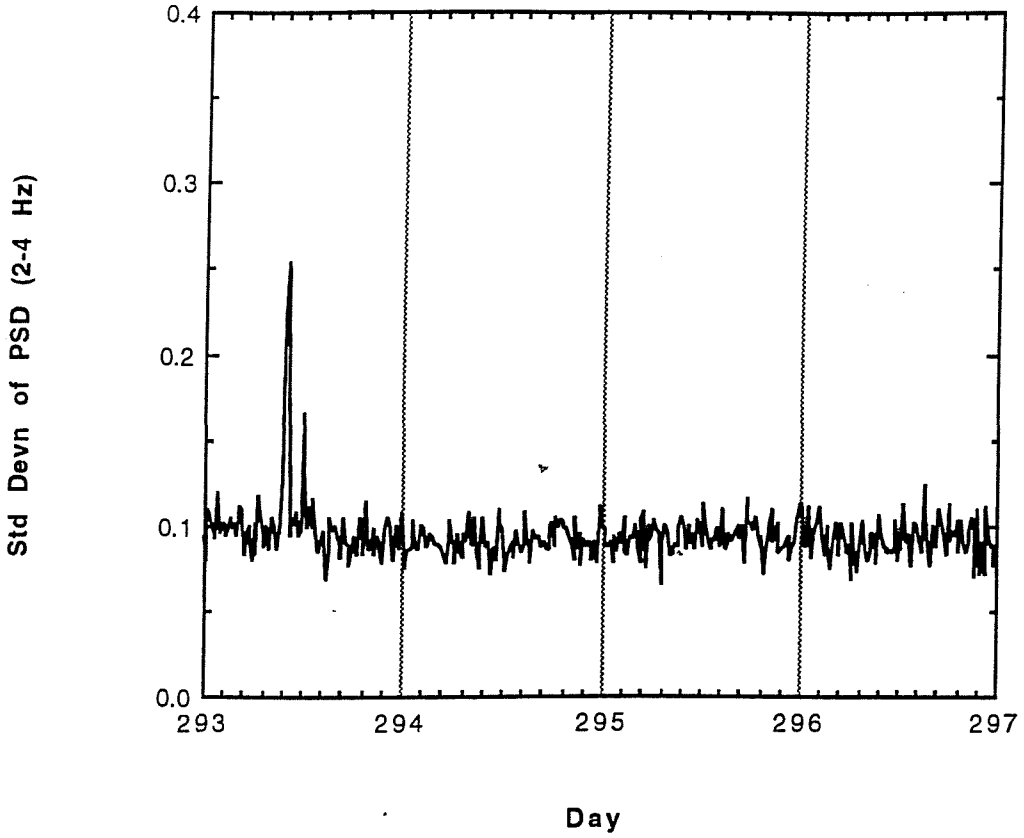


Figure 7. Data from "SON297.VAR"

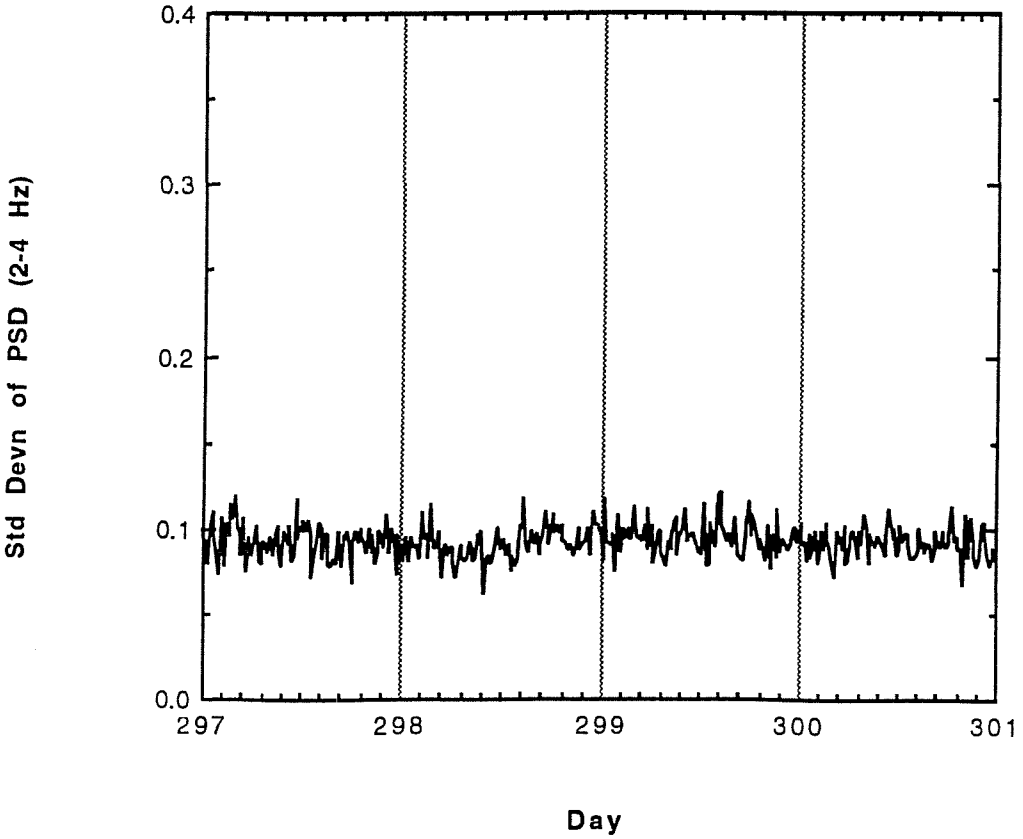


Figure 8. Data from "SON301.VAR"

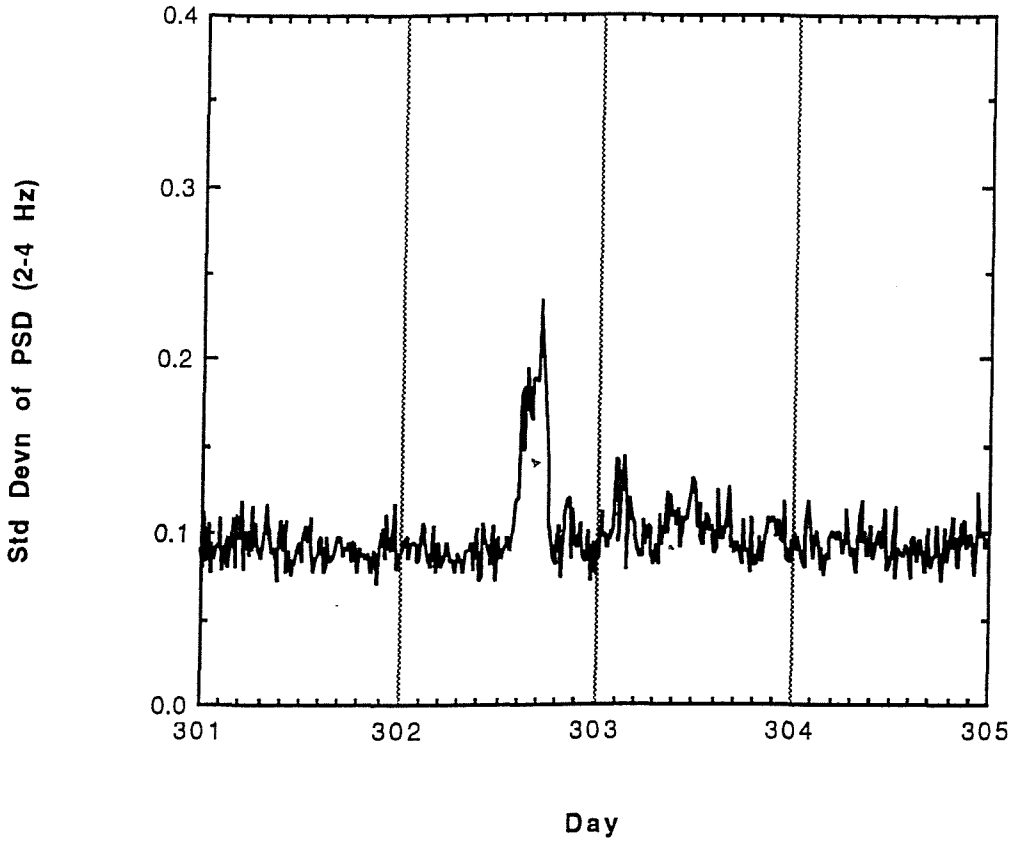


Figure 9. Data from "SON305.VAR"

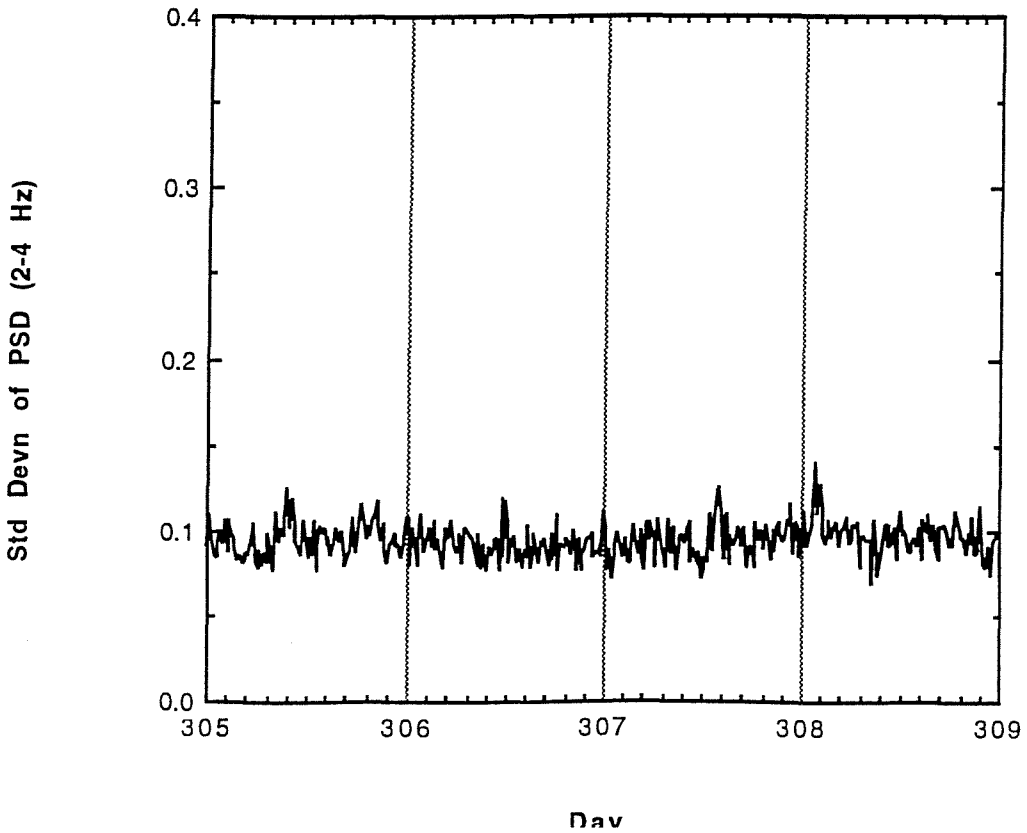


Figure 10. Data from "SON309.VAR"

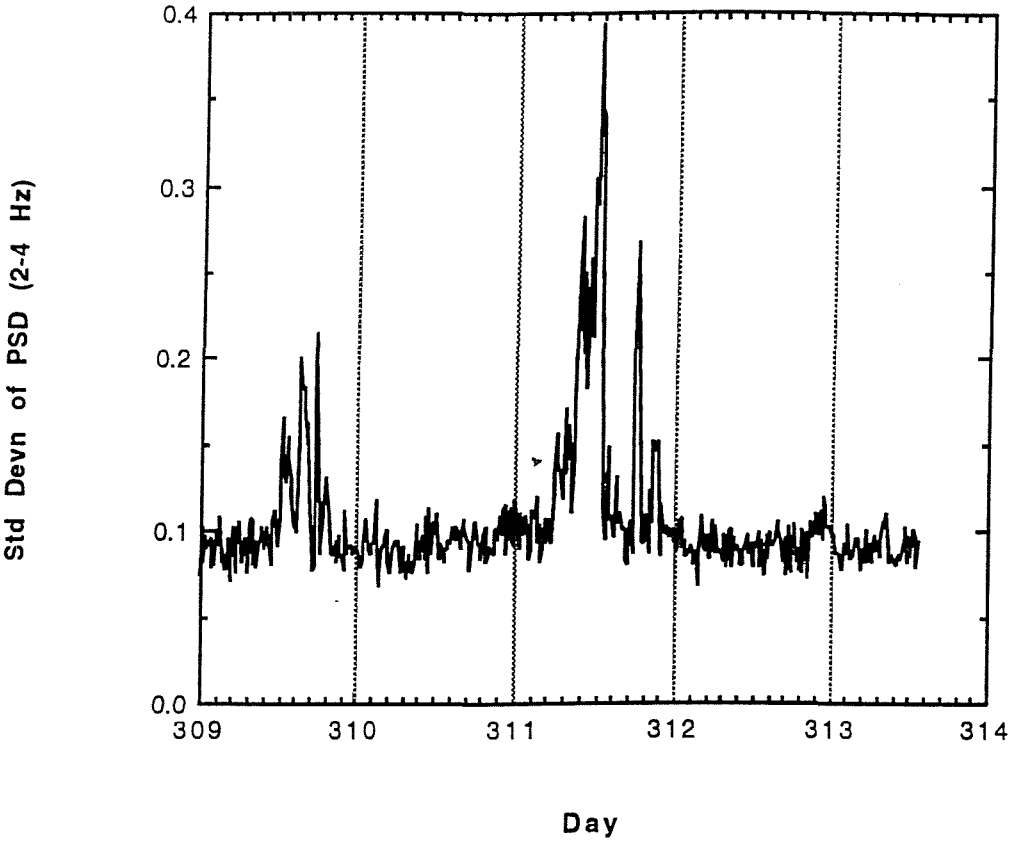


Figure 11. Data from "SON325.VAR"

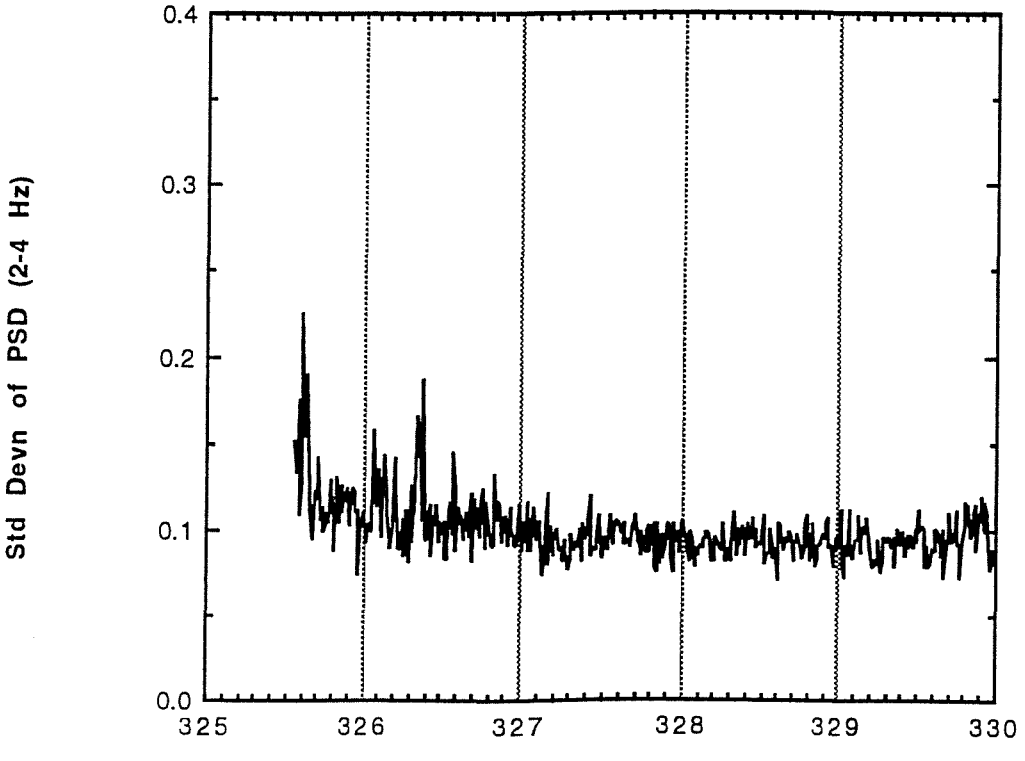


Figure 12. Data from "SON330.VAR"

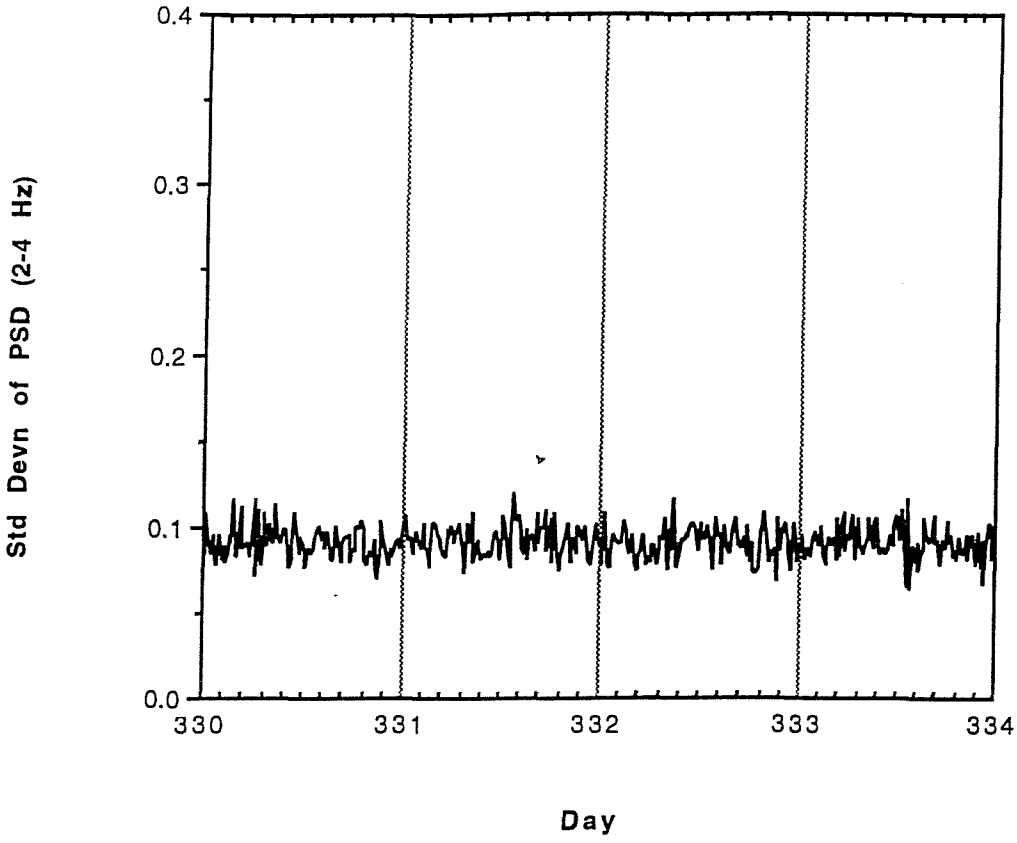
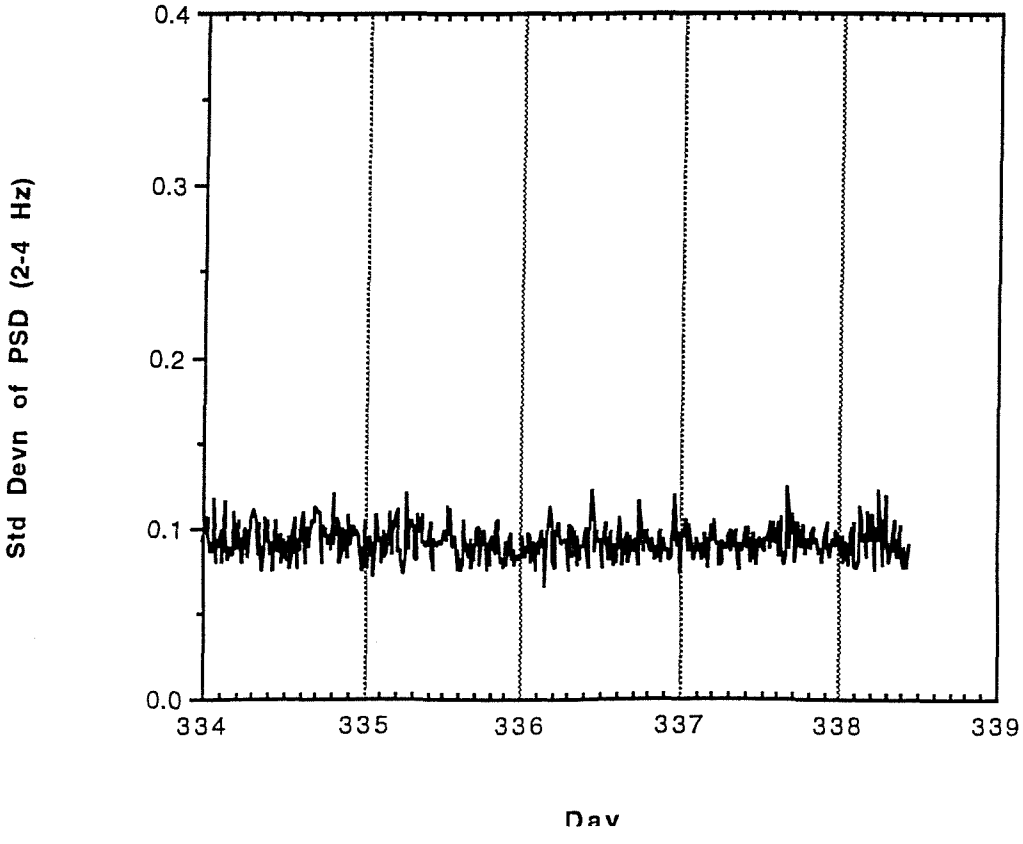


Figure 13. Data from "SON334.VAR"



Bad data parameters have not been removed from the tabular or spectral data files, but the flagged spectral file headers are listed below; those in brackets are prior to the respective deployments.

(2931000FFTSpd	2931015FFTSpd	2931215FFTSpd)	
3021445FFTSpd	3021500FFTSpd	3021530FFTSpd	3021545FFTSpd
3021600FFTSpd	3021615FFTSpd	3021630FFTSpd	3021645FFTSpd
3021700FFTSpd	3021715FFTSpd	3021730FFTSpd	3021745FFTSpd
3091230FFTSpd	3091315FFTSpd	3091500FFTSpd	3091515FFTSpd
3091530FFTSpd	3091545FFTSpd	3091600FFTSpd	3091615FFTSpd
3091630FFTSpd	3091745FFTSpd	3091800FFTSpd	
3110615FFTSpd	3110730FFTSpd	3110800FFTSpd	3110900FFTSpd
3110915FFTSpd	3110930FFTSpd	3110945FFTSpd	3111000FFTSpd
3111015FFTSpd	3111030FFTSpd	3111045FFTSpd	3111100FFTSpd
3111115FFTSpd	3111130FFTSpd	3111145FFTSpd	3111200FFTSpd
3111215FFTSpd	3111230FFTSpd	3111245FFTSpd	3111745FFTSpd
3111800FFTSpd	3111815FFTSpd	3111830FFTSpd	3112045FFTSpd
3112115FFTSpd			
(3251330FFTSpd	3251400FFTSpd	3251445FFTSpd	3251500FFTSpd
3251515FFTSpd	3251530FFTSpd)		
(3260145FFTSpd	3260200FFTSpd	3260830FFTSpd	3260845FFTSpd
3260930FFTSpd)			

The raw data files have not all been extracted from the FlashCard data files and replayed through the processing software; however, they appear to be satisfactory.

Summary of Data Produced

Spectral Data Files

a) original translations of EPROM logger data

SONIC2.DOC

SONIC325.DOC

b) time corrected and split into shorter files

SON293.DOC	days 293.0000 TO 296.9896
SON297.DOC	days 297.0000 to 300.9896
SON301.DOC	days 301.0000 to 304.9896
SON305.DOC	days 305.0000 to 308.9896
SON309.DOC	days 309.0000 to 313.5625
SON325.DOC	days 325.5625 to 329.9896
SON330.DOC	days 330.0000 to 333.9896

SON334.DOC days 334.0000 to 338.4375

(parameters from these were then combined with Formatter tabular data resulting in the CricketGraph files

1st Deployment CG Final

2nd Deployment CG Final)

Raw Data Files

1st Deployment:

RAWSWAL.1MG

RAWSWAL.2MG

RAWSWAL.3MG

RAWSWAL.4MG

2nd Deployment:

RAWSWAL1.1MG

RAWSWAL1.2MG

RAWSWAL1.3MG

RAWSWAL1.4MG

Data Time Stamping

The Sonic Processor clock performed correctly during the first deployment, but during the second deployment, the operation of the Real Time Clock to System Clock update program RTCN.EXE occasionally caused time jumps; the reason for this has not yet been found.

The Sonic parameters logged by the Formatter during the second deployment to Flash EEPROM have been time corrected, using the Formatter clock time/date. It was then possible to compare these parameters with the parameters appended to the EPROM spectral data and hence to correct the time stamps of the spectral data. The QBasic application CORRSON.BAS (Appendix A.3) was developed for this purpose. The data file SONIC325.DOC was split into 9 files with constant Sonic timeshifts, CORRSON was then used to correct the time stamps of these files. Finally, the 9 files were combined into 3 spectral files, each of about 5 days duration, named SON325.DOC, SON330.DOC and SON334.DOC

Acknowledgements

The SWALES data set was the result of the concerted efforts of many, including the IOSDL Centre for Ocean Technology Development members of the Met Team, the IOSDL Moorings Team and the JRC members of the Met Team. The experimental work was funded by the

MAFF Flood and Coastal Defence Division under commission FD0603; analysis of the data will be under commission FD0601.

References

1. Clayson, C.H. and Pascal, R.W. 1994, Sonic Buoy - Sonic Processor Handbook, IOSDL Internal Document No. 340, 69pp.

2. Clayson, C.H. 1994, Sonic Buoy - Formatter Handbook, IOSDL Internal Document No. 339, 126pp.

3. Clayson, C.H. 1994, Sonic Buoy Raw Sonic Data Logger, IOSDL Internal Document (in course of preparation)

4. Clayson, C.H. 1994, SWALES Sonic Buoy - Meteorological Data Report, IOSDL Internal Document No. 342, 42pp.

Appendix A.1 BASIC program DECODE.BAS

```
REM QBasic program DECODE.BAS for PC
REM program to convert hex ASCII EPROM logger output file
REM to ASCII file
REM
REM RWP

5 CLS
10 OPEN "d:\sonic.txt" FOR INPUT AS #1
12 OPEN "d:\sonic.doc" FOR OUTPUT AS #2
15 a$ = INPUT$(48, 1)
16 DO
18 FOR N = 1 TO 15
20      FOR i = 1 TO 128
30              a$ = INPUT$(2, 1)
35              a$ = "&H" + a$
37              a = VAL(a$)
40              PRINT #2, CHR$(a);
50      NEXT:
60      x$ = INPUT$(4, 1)
70 NEXT N
80 LOOP UNTIL (EOF(1))
90 CLOSE #1: CLOSE #2
100 END
```

Appendix A.2 Source Code of 4MTO1M.C

```
/* Source Code of 4MTO1M.C for PC
for converting a 4 Mbyte flashcard file
(produced by reading card on thincard drive with batch file T.BAT)
to 4 separate 1 Mbyte files
Author CHC 16th February 1994 */

#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
```

```
main()
{

FILE * fin;
FILE * fout;

int c;

long n;

if ( (fin = fopen("c:\\thincard\\test", "rb")) != NULL)
     {
     if ( (fout = fopen("test.1mg", "wb+")) != NULL)
          {
          printf("Converting 1st Mbyte\n");
          for (n = 0; n < 1048576L; n++)              /* copy 1st meg to testfile */
               {
               c = fgetc(fin);
               /* if ((n > 262143L) && (n < 524288L)) for txfr of sonic raw data */

                    {
                    fputc(c, fout);
                    }
               }
          printf("Transfer OK\n");
          }
     else
          {
          printf("Failed to open O/P File\n");
          }
     fclose(fout);
     printf("Press a key to continue\n");
     getch();
     /* copy 2nd meg to test.2mg */

     if ( (fout = fopen("test.2mg", "wb+")) != NULL)
          {
          printf("Converting 2nd Mbyte\n");
          for (n = 0; n < 1048576L; n++)
               {
               c = fgetc(fin);
               fputc(c, fout);
               }
          printf("Transfer OK\n");
          }
     else
          {
          printf("Failed to open O/P File\n");
          }
     fclose(fout);
     printf("Press a key to continue\n");
     getch();

     /* copy 3rd meg to test.3mg */

     if ( (fout = fopen("test.3mg", "wb+")) != NULL)
          {
          printf("Converting 3rd Mbyte\n");
          for (n = 0; n < 1048576L; n++)
               {
               c = fgetc(fin);
               fputc(c, fout);
               }
```

```
            printf("Transfer OK\n");
            }
        else
            {
            printf("Failed to open O/P File\n");
            }
        fclose(fout);
        printf("Press a key to continue\n");
        getch();

        /* copy 4th meg to test.4mg */

        if ( (fout = fopen("test.4mg", "wb+")) != NULL)
            {
            printf("Converting 4th Mbyte\n");
            for (n = 0; n < 1048576L; n++)
                {
                c = fgetc(fin);
                fputc(c, fout);
                }
            printf("Transfer OK\n");
            }
        else
            {
            printf("Failed to open O/P File\n");
            }
        fclose(fout);


    }
else
    {
    printf("Failed to open I/P File\n");
    }

return 0;
}
```

Appendix A.3 Listing of CORRSON.BAS

```
REM QBasic program CORRSON.BAS for PC
REM program to correct times of sonic eprom data
REM
REM CHC 4th March 1994

CLS
INPUT "Enter file to be corrected : ", f$
INPUT "Enter start date as string jjjhhmm : ", d$

OPEN f$ FOR RANDOM AS #1 LEN = 1920
FIELD #1, 1920 AS r$
n% = 1

WHILE NOT EOF(1)
    GET #1, n%
    PRINT MID$(r$, 1, 7); " "; MID$(r$, 1806, 7)
    IF MID$(r$, 1, 7) <> MID$(r$, 1806, 7) THEN
        PRINT "Duff info": END
    END IF
    MID$(r$, 1, 7) = d$: MID$(r$, 1806, 7) = d$
    days% = VAL(LEFT$(d$, 3))
```

```
        hours% = VAL(MID$(d$, 4, 2))
        minutes% = VAL(MID$(d$, 6, 2))
        IF minutes% = 45 THEN
            IF hours% = 23 THEN
                days% = days% + 1: hours$ = "00": minutes$ = "00"
            ELSE
                hours% = hours% + 1: minutes$ = "00"
                hours$ = STR$(hours%)
                IF LEN(hours$) = 2 THEN
                    hours$ = "0" + RIGHT$(hours$, 1)
                ELSE
                    hours$ = RIGHT$(hours$, 2)
                END IF
            END IF
        ELSE
            minutes$ = STR$(minutes% + 15)
            IF LEN(minutes$) = 2 THEN
                minutes$ = "0" + RIGHT$(minutes$, 1)
            ELSE
                minutes$ = RIGHT$(minutes$, 2)
            END IF
            hours$ = STR$(hours%)
            IF LEN(hours$) = 2 THEN
                hours$ = "0" + RIGHT$(hours$, 1)
            ELSE
                hours$ = RIGHT$(hours$, 2)
            END IF

        END IF
        days$ = RIGHT$(STR$(days%), 3)

        d$ = days$ + hours$ + minutes$
        PRINT d$
        PUT #1, n%
        n% = n% + 1
    WEND
    CLOSE
    END
```

Appendix A.4 Listing of SONPARAM.BAS

```
REM QBasic program SONPARAM.BAS for PC
REM program to extract tabular data from EPROM logger data file
REM
REM CHC 4th March 1994

CLS
INPUT "Enter EPROM data file to be converted : ", f$
INPUT "Enter output tabular file name", g$

OPEN f$ FOR RANDOM AS #1 LEN = 1920
FIELD #1, 1920 AS r$
OPEN g$ FOR OUTPUT AS #2
n% = 1

WHILE NOT EOF(1)
    GET #1, n%
    PRINT MID$(r$, 1, 7); " "; MID$(r$, 1806, 7)
    IF MID$(r$, 1, 7) <> MID$(r$, 1806, 7) THEN
        PRINT "Duff info": END
    END IF
```

```
    day = VAL(LEFT$(r$, 3)) + VAL(MID$(r$, 4, 2)) / 24 + VAL(MID$(r$, 6, 2)) / 1440
    psd = VAL(MID$(r$, 1884, 8))
    mws = VAL(MID$(r$, 1843, 5))
    nws = VAL(MID$(r$, 1849, 6))
    ews = VAL(MID$(r$, 1856, 6))
    vws = VAL(MID$(r$, 1863, 6))
    fit = VAL(MID$(r$, 1893, 8))
    PRINT #2, day; CHR$(9); psd; CHR$(9); mws; CHR$(9); nws; CHR$(9); ews; CHR$(9); vws;
CHR$(9); fit
    n% = n% + 1
WEND
CLOSE
END
```

Appendix A.5 Listing of SONSELEC.BAS

```
REM QBasic program SONSELEC.BAS for PC
REM
REM program to select suspect Sonic spectra from EPROM logger data
REM based on exceeding a std devn of 0.15 for PSD over range 2 - 4 Hz
REM
REM CHC 11/04/94

CLS
INPUT "Enter EPROM data file to be examined : ", f$
INPUT "Enter output file for summary parameters : ", g$

OPEN f$ FOR RANDOM AS #1 LEN = 1920
OPEN g$ FOR OUTPUT AS #2
FIELD #1, 1920 AS r$
n% = 1
SCREEN 9

h1% = 512 * 2 / 20.833
h2% = 512 * 4 / 20.833


WHILE NOT EOF(1)
    GET #1, n%
    PRINT MID$(r$, 1, 7); " "; MID$(r$, 1806, 7)
    IF MID$(r$, 1, 7) <> MID$(r$, 1806, 7) THEN
        PRINT "Duff info": END
    END IF

    day = VAL(LEFT$(r$, 3)) + VAL(MID$(r$, 4, 2)) / 24 + VAL(MID$(r$, 6, 2)) / 1440
    s = 0
    FOR h% = 1 TO 255
        en = VAL(MID$(r$, 14 + 7 * h%, 6))
        IF h% = 1 THEN
            PSET (2 * h%, -40 * en)
        ELSE
            LINE -(2 * h%, -40 * en)
        END IF
        IF (h% >= h1%) AND (h% <= h2%) THEN s = s + en

    NEXT
    s = s / (h2% - h1% + 1)
    ss = 0
    FOR h% = h1% TO h2%
        en = VAL(MID$(r$, 14 + 7 * h%, 6))
```

```
        ss = ss + (en - s) * (en - s)
    NEXT
    ss = SQR(ss / (h2% - h1% + 1))
    IF (ss > .15) THEN PRINT #2, r$
    CLS
    n% = n% + 1
WEND
CLOSE
END
```

Appendix B Spectral Data Format

The format of each spectral data set within the ASCII files consists of a spectrum header, followed by the mean wind speed reading and 255 estimates of the form $\log_{10}(PSD*freq^{5/3})$. This is followed by a parameters header, followed by the computed parameters.

jjjhhmmFFTSpd<CR>	(where jjj is Julian Day, hh is hour, mm is minute of start)
mw.ws<CR>	(where mw.ws is mean resultant wind speed)
255 lines with the format:	
+h.est<CR>	(where h.est are spectral estimates for the 2nd to the 256th line)
jjjhhmmPSDSpd<CR>	(where jjj is Julian Day, hh is hour, mm is minute of start)
IDID<CR>	(Sonic Sensor ID)
001<CR>	(Records per file)
F1.F1<CR>	(Lower frequency for averaging range)
F2.F2<CR>	(Upper frequency for averaging range)
1<CR>	(Sonic Mode)
mw.ws<CR>	(where mw.ws is mean resultant wind speed)
+nm.ws<CR>	(where +nm.ws is mean wind speed from North)
+em.ws<CR>	(where +em.ws is mean wind speed from East)
+vm.ws<CR>	(where +vm.ws is mean wind speed upwards)
cme.an<CR>	(where cme.an is mean speed of sound)
hea.dg<CR>	(where hea.dg is mean buoy heading)
+p.sdpsd<CR>	(where +p.sdpsd is mean PSD over range F1 to F2)
+a.lalal<CR>	(where +a.lalal is least squares fit 'a' coefficient)
+b.bbbbbe+bbb<CR>	(where +b.bbbbbe+bbb is ls fit 'b' coefficient)
END<CR><LF><CR>	

making a total of 1920 characters, which are transferred to the EPROM logger in 15 blocks of 128 bytes.

Appendix C Raw Data Files Format

The directory entries in the first 256 kbytes of the card memory give the Raw Data Logger date/time and memory location for the start of each "record".

Each 32 byte directory entry is of the form:

vjjjhhmmbfflnn0vjjjhhmmbfflnn0

where

v is a marker
jjj (ASCII) is the Julian day number (range 001 - 366)
hh (ASCII) is the hour (range 00 - 23) of the END of the record

mm (ASCII) is the minute (range 00 - 59) of the END of the record
b (binary) is the memory block for the start of the data (range 4 - 63, each block is 65536 bytes, the directory being in blocks 0 - 3)
ff (binary) is the 16 bit offset within that block for the start of the data
ll (binary) is the 16 bit length of the data (modulus 65536)
nn (binary) is the 16 bit record number

The "records" are in the standard FASTCOM file format, consisting of a header and at least 12288 samples of data, i.e.

Header (44 bytes):

Mode<sp>1<LF>

Analog<sp>1<LF>

Time<sp>hh:mm:ss<sp>Date<sp>mm/dd/yy<LF>

Data: at least 12288 x 10 byte samples, each consisting of::

3 velocitycomponents (U, V, W) each of 2 bytes (16 bit binary integers), 1 * 2byte velocity of sound, C (16 bit binary integer), 2byte compass reading, H (16 bit binary integer)

where U, V and W normally have the range -6000 to +6000 for -60 m/s to +60 m/s, with a value of -10000 being used if there is a fault condition

C normally has the range 0 to +18500 for 0 m/s to 370 m/s, with a value of -10000 being used if there is a fault condition

H has the nominal range 2048 to 4088 for a compass output of 0 to 255 (0^{o} to 358.6^{o} clockwise relative to magnetic North)

The number of samples is longer than the nominal length of 12288 since the Raw Data Logger's logging cycle is turned on and off by the Prompted and Unprompted commands to the anemometer. The data are acquired in blocks of 20, 21 or 22 samples (on average 20.833). The Unprompted command is sent by the Sonic Processor after a block has been acquired which brings the Prompted total up to at least 12288. Consequently the raw data records vary slightly in length and are always at least 12288 samples in length.