The SETI Interpreter Program (SIP) – A Software Package for the SETI Field Tests

E. T. Olsen and A. Lokshin
Atmospheric Sciences Section

The SETI (Search for Extraterrestrial Intelligence) Interpreter Program is an interactive software package designed to allow flexible off-line processing of the SETI field test data on a PDP 11/44 computer. The user can write and immediately execute complex analysis programs using the compact SIP command language. The software utilized by the SETI Interpreter Program consists of FORTRAN - coded modules that are sequentially installed and executed.

I. Introduction

The objective of the SETI SR&T Program is the development of efficient means by which a large-scale microwave search for signals of extraterrestrial intelligent origin may be accomplished. A key element of the approach is the field test of hardware, search strategies, and signal-detection algorithms.

The SETI Science Team perceived the necessity for software that can be exploited by a large number of users to accomplish a wide range of analyses without requiring extensive new-code generation. The SETI Interpreter Program (hereafter referred to as the SIP) is the result.

A deliberate decision was made to sacrifice speed for flexibility, for it is planned to identify the optimum signal detection algorithms, which will then be implemented in custom high-speed hardware. The SIP is not expected to be capable of performing real-time data analysis so long as all processing is accomplished in software alone. Short bursts of raw data will be dumped on magnetic tape, which will then be analyzed in the field in near-real time or brought back to JPL or NASA/Ames.

The SETI SR&T Program will employ the Radio Frequency Surveillance System (RFISS) 65-kbin pipeline FFT device as well as its own 74-kbin multichannel spectrum analyzer (MCSA) to collect spectral data. The RFISS and the MCSA will both be used at DSS 13, the 26-meter R&D antenna at Goldstone, California; the MCSA will also be used at the 305-meter antenna at Arecibo Observatory in Puerto Rico.

II. Data Format

A necessary prerequisite to the development of software for data analysis is the specification of the format and content of the data records themselves. If such a standardization of the data records had not been made, the SIP would have been impossible, and all software would perforce be stand alone and thus entail a duplication of effort.

The SIP works with data in the standard SETI processed data format. This format defines a spectrum to be made up of 4-kword physical records, the first of which is a header record. Although the MCSA will be able to output data in several different formats, the SIP presently works with only one format, assuming data are unsigned 16-bit integers. This format was chosen because it is common to both the MCSA and the RFISS. At present, the SIP is configured to manipulate spectra as large as 32 kbins, but this limit is imposed only by the 512-kbyte memory of the PDP 11/44. When the memory is upgraded to a full 1 Mbyte, the SIP can be easily configured to handle 74-kbin spectra.

The MCSA will be equipped with an interface to allow direct data transfer to disk or tape of the PDP 11/44. The RFISS will be so equipped at some time in the future. During the next year, the only interface will be via magnetic tape transferred between the Modcomp in the RFISS van and the PDP 11/44. Software has been written to translate the Modcomp tape data into SETI processed data format on disk.

III. Features of the SETI Interpreter Program

The presently implemented SIP reads spectra in SETI processed data format from disk files and accepts commands from the user console. There are three major parts of the SIP:

(1) A memory-resident Task Manager, which parses keyboard input, keeps track of task names and the numerical parameters and character parameters to be passed to the tasks, and schedules the spawning of independent tasks.

(2) A memory-resident Data Region, amounting to 108 kwords, which is large enough to hold three 32-kword spectra and their respective 4-kword headers.

(3) A disk-resident SETI Task Library, which holds most of the executable tasks that the Task Manager will spawn. A few frequently utilized tasks are incorporated into the manager to speed the SIP execution.

All SIP software has been coded in FORTRAN 77, even the part of the task manager, which parses the keyboard input. Some readers may consider the choice of language unfortunate, arguing for an assembly code where high-speed computations are required and a character-string friendly code (like PASCAL, for example) where the parsing occurs. The answer is that the SETI program stipulated some time ago that all software should be coded in FORTRAN to forestall the building of another Tower of Babel when the attempt was made to merge the products of JPL, NASA Ames, and Stanford University.

The PDP 11/44 is not a virtual memory machine and so it is not possible for a program to address more than 32 kwords of memory. The SIP utilizes a 4-kword window that can be mapped around data and headers of the spectra, and so each task can really address no more than 28 kwords plus the window. It is this memory constraint that forced us to use disk-resident tasks, despite the fact that the time required to simply install a disk-resident task for execution is approximately one second.

Some readers who are familiar with PDP 11s may wonder why we went the route of spawning independent tasks, rather than utilizing the task overlay features of this machine. We actually started out along that road, but soon came to the point where the overlay tree became so impossibly complex that it was a major undertaking to modify a task or add a new task. We switched over to spawning independent tasks to make maintenance and modification of the SIP humanly possible!

Communication between separate spawned tasks is accomplished by means of a special common block and the headers of the spectra. Thus every task includes in its code a standard preamble that keeps it informed as to the state of the data and its analysis.

Due to the memory restriction, there are limits to the size of the task stack, numerical stack, character stack, and typed command line. A data analysis program may not exceed any one of the following constraints:

(1) A typed line may contain no more than 120 characters

(2) A command line (made up of one or more typed lines) may contain no more than 30 tasks

(3) A program may contain a maximum of 25 command lines, 100 tasks, 200 numerical parameters, 300 character parameters, 25 dynamically variable parameters, and 10 macro commands

(4) A macro command may contain a maximum of 15 tasks, 30 numerical parameters, and 100 character parameters.

The above tabulation of constraints has introduced some concepts that have not been previously defined. A "typed line" is whatever the investigator has typed in response to the SIP prompt. A "command line" is a unit that can be repeatedly executed. A "program" is a complete data analysis algorithm. A "macro command" is a convenient feature that allows the investigator to identify an often-used series of tasks with a name; in effect it is a user-defined task made up of many SIP tasks. A "dynamically variable parameter" is one of a special set of numerical parameters that can be changed by the program during execution and used as a numerical parameter input to a task.

The SIP has already undergone several revisions, for we identified new requirements as we began field-test data analyses. Some especially useful facilities we added are:

(1) Dynamically variable parameters and primitive logical IF and GO TO commands to allow program execution to be controlled by the results of the analysis

(2) Input of stored ASCII files in response to the SIP prompt. Thus we may use any text editor to compose a program file that can then be read and executed by the SIP

(3) The capability to heavily comment the ASCII file so that the analysis program is well documented

(4) Debugging features.

A full list of the tasks presently implemented and planned is given in the Appendix.

IV. An Example Analysis Program

The contemplated bimodal search strategy (see Ref. 1) for the microwave search includes an all-sky survey. Such a survey would rapidly scan the sky with an antenna, looking for excessive narrowband power coming from a direction fixed to the celestial sphere. Since the antenna is being moved, the power will vary because of changing sidelobe pickup, maser gain, and troposphere, among others. Thus the power in each spectrum bin will vary even though no narrowband signal is present. It will be necessary to use some reference spectrum to remove the changing ripples and overall bias of the newly acquired spectrum before applying a threshold to look for excessive power. This is a process that SETI personnel call "baselining," and it is a high-priority item in the field-test investigations. If we are fortunate (and careful), the ripples should be rather broad, and thus the required reference spectrum will be of very coarse resolution. In the worst case, the reference spectrum will have the same resolution as the newly acquired spectrum.

Of course, the reference spectrum will also vary with time. The current plans require that the reference spectrum be updated with some smoothing in time. In the worst case, each spectrum bin will be baselined using a value that reflects its recent history. The example that follows shows how this algorithm may be implemented in the SIP for testing.

Figure 1 is an overall logical flow diagram of the algorithm, which does not include SIP peculiar steps. A new spectrum is acquired and it is compared to a reference spectrum. If any bins exceed a preset threshold, the hits are reported. The reference spectrum is then updated and the cycle begins anew.

Figure 2 is the flow diagram, which includes the SIP peculiar steps. Recall that there are presently three spectra allowed in memory (V1, V2, and V3) and up to 25 dynamical variables ($P(n)$).

Figure 3 is the analysis program itself as read by the SIP. These are four ASCII files that were composed using a text editor and stored. The first file initializes some parameters, the second file defines macros, the third file makes a first guess at the reference spectrum, and the fourth file is the actual analysis program. If a first guess at the reference spectrum is already on disk, the third file need not be executed. These four files demonstrate how a program file may be documented for future users. As an indication of the (lack of) speed of the SIP, the master loop of the fourth file requires approximately one minute and twenty seconds. Note that this program contains an extra calculation that keeps track of the number of times a particular bin exceeded threshold. This is an attempt to identify local radio frequency interference (RFI) that is not variable in frequency.

This analysis program has in fact been executed using some data taken with the RFISS at X-band using DSS 14. Figure 4 shows what an individual raw spectrum looks like. Figure 5 shows the first guess baseline. Figure 6 is a representative report on an individual spectrum. Figure 7 is the accumulated RFI report. Figure 8 is the plot output of this accumulated RFI report.

V. Software Portability

The SIP would be portable to any machine supporting FORTRAN 77 with minor changes except for two PDP 11/44 system features that have been incorporated in it. The first is the spawning of independent tasks. Any target machine must allow an executable element resident on the disk to be installed, run, and removed under software calls to the system. The second and more serious is the memory mapping which is built into the SIP. Any machine of the PDP 11 series that supports memory mapping is a good target, but even a VAX requires some major modifications of the software. The SETI program does plan to make the SIP run on a VAX, but we have not yet scoped the magnitude of the task.

The SIP Task Manager can be easily modified to allow a totally different library of tasks and data structure to be installed. Thus any potential user can customize the SIP for a specialized analysis task, building upon the basic structure already in place.

VI. Future Work

As the reader can see by looking at the tasks listed in the Appendix, not all desired tasks have been implemented yet. These and tasks yet to be identified shall be added to those currently available as time and need dictate. Once the memory of the PDP 11/44 is upgraded to 1 Mbyte, the SIP data area in memory will be set up to handle four vectors to minimize disk I/O. There will very likely be two installed versions of the SIP. The smaller should handle up to 32-kbin spectra so that two investigators may execute in the PDP 11/44 concurrently. The larger will handle 74-kbin spectra, and only one investigator will fit in the memory at a time.

References

1. Gulkis, S., Olsen, E. T., and Tarter, J., "A Bimodal Search Strategy for SETI," *Strategies for the Search for Life in the Universe*, edited by M. D. Papagiannis. The Proceedings of the IAU General Assembly, Vol. 83, pp, 93-105. D. Reidel, Dodrecht, Holland, the Netherlands, 1980.

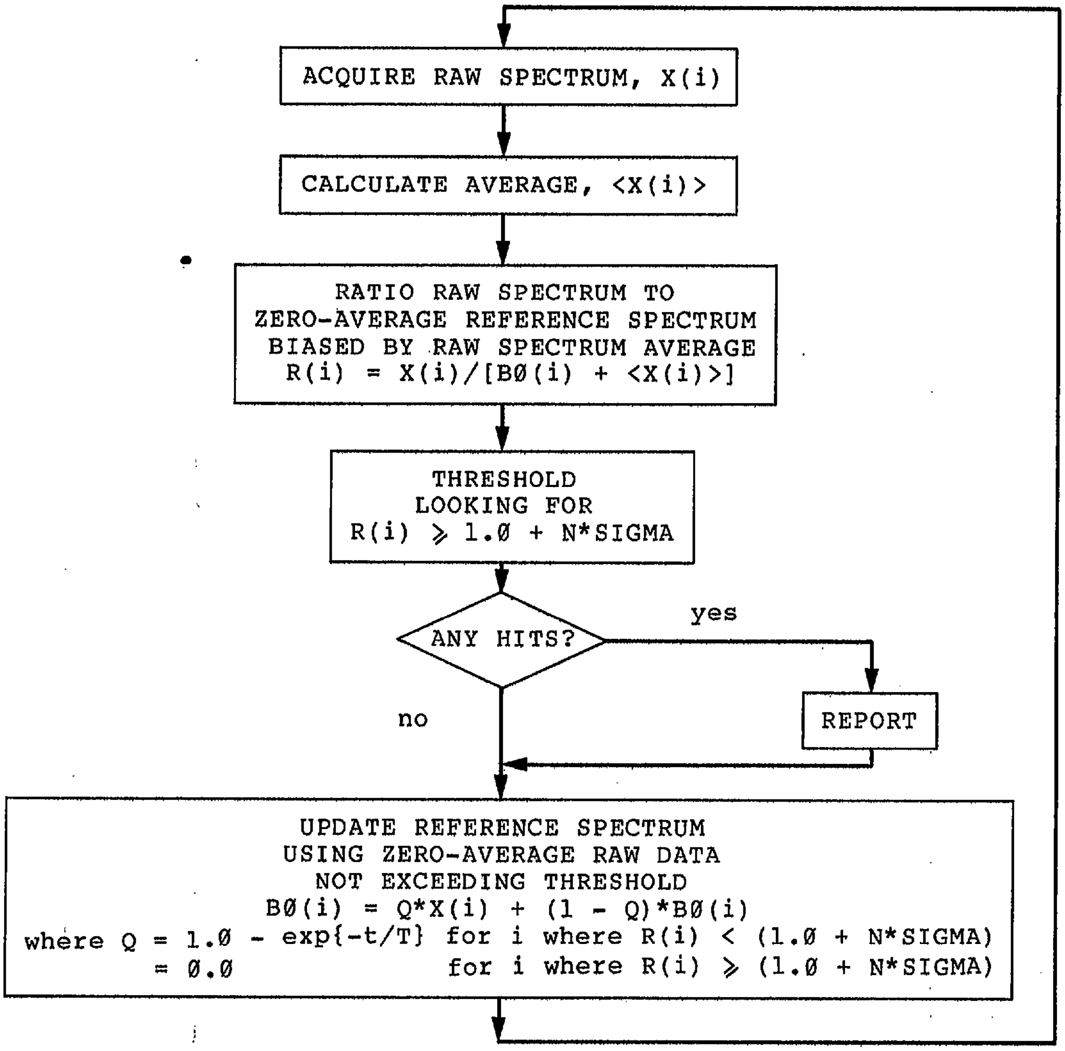

ACQUIRE RAW SPECTRUM, X(i)
CALCULATE AVERAGE, <X(i)>
RATIO RAW SPECTRUM TO
ZERO-AVERAGE REFERENCE SPECTRUM
BIASED BY RAW SPECTRUM AVERAGE
R(i) = X(i)/[BØ(i) + <X(i)>]
THRESHOLD
LOOKING FOR
R(i) ⩾ 1.Ø + N*SIGMA
ANY HITS?
yes
REPORT
no
UPDATE REFERENCE SPECTRUM
USING ZERO-AVERAGE RAW DATA
NOT EXCEEDING THRESHOLD
BØ(i) = Q*X(i) + (1 - Q)*BØ(i)
where Q = 1.Ø - exp{-t/T} for i where R(i) < (1.Ø + N*SIGMA)
= Ø.Ø for i where R(i) ⩾ (1.Ø + N*SIGMA)

Fig. 1. A candidate signal-detection algorithm for an all-sky SETI. Because of the rapid movement of the antenna, the reference spectrum must be continuously updated

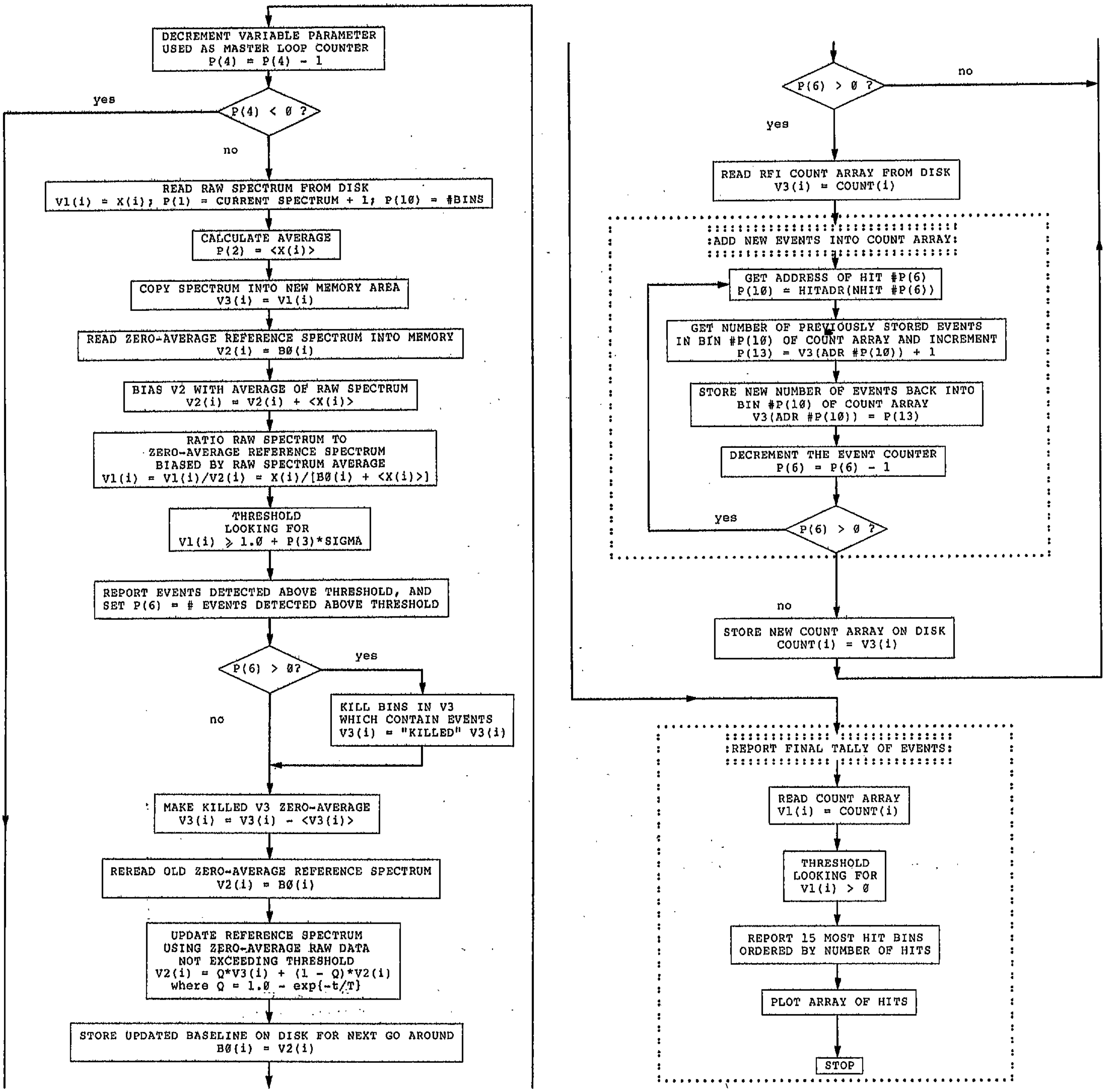
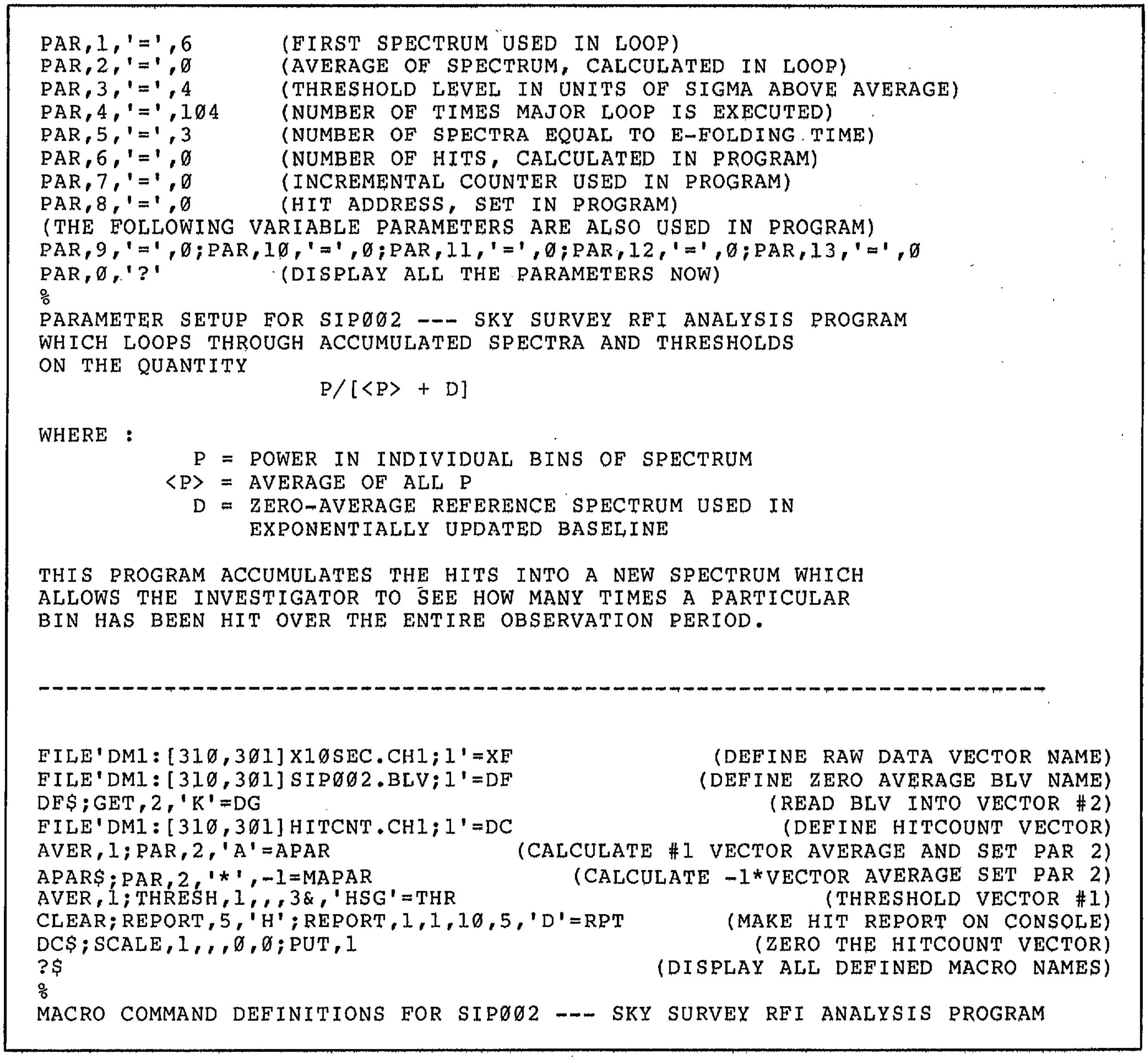
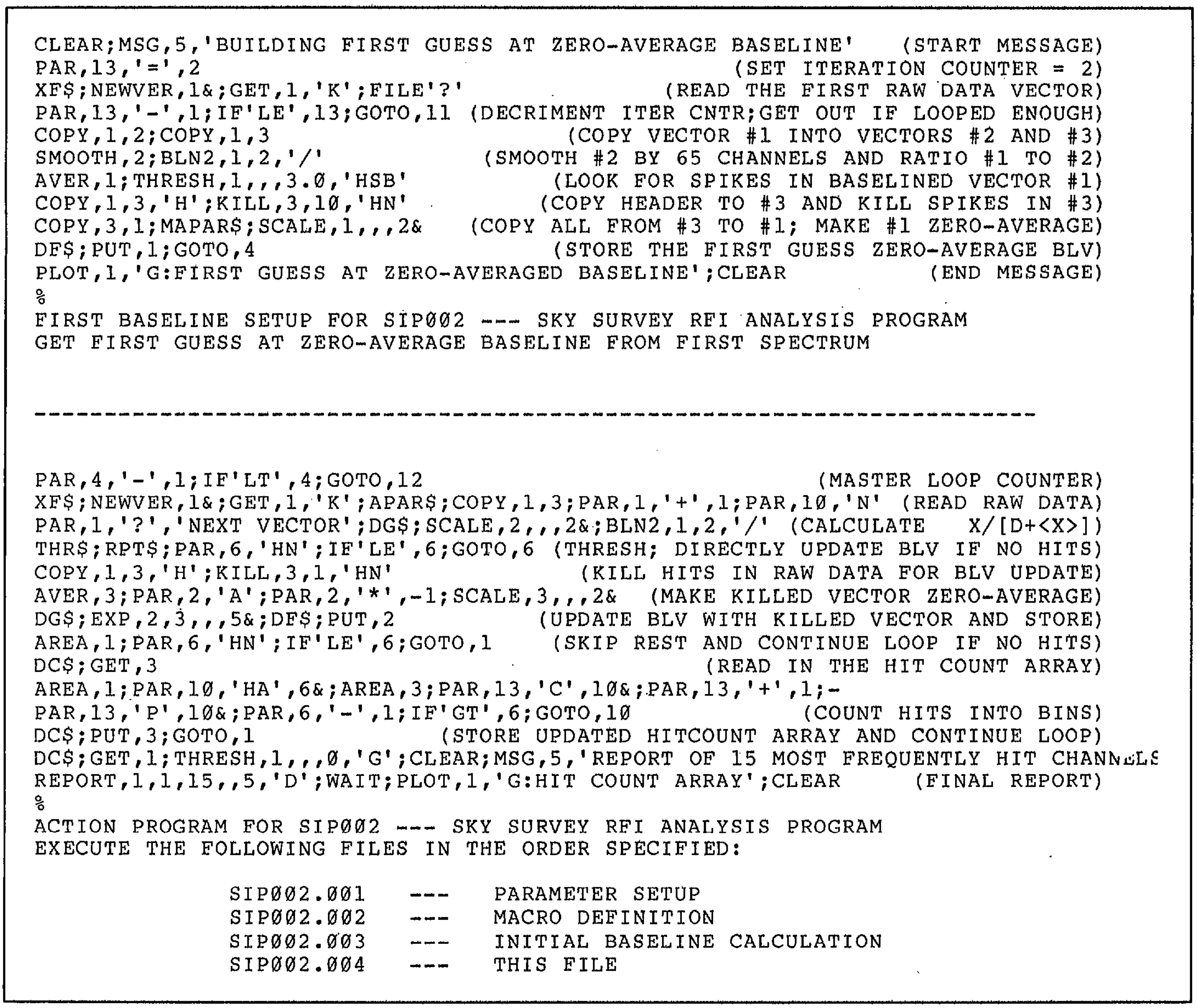

DECREMENT VARIABLE PARAMETER USED AS MASTER LOOP COUNTER
P(4) = P(4) - 1
P(4) < Ø ?
yes
no
READ RAW SPECTRUM FROM DISK
V1(i) = X(i); P(1) = CURRENT SPECTRUM + 1; P(1Ø) = #BINS
CALCULATE AVERAGE
P(2) = <X(i)>
COPY SPECTRUM INTO NEW MEMORY AREA
V3(i) = V1(i)
READ ZERO-AVERAGE REFERENCE SPECTRUM INTO MEMORY
V2(i) = BØ(i)
BIAS V2 WITH AVERAGE OF RAW SPECTRUM
V2(i) = V2(i) + <X(i)>
RATIO RAW SPECTRUM TO ZERO-AVERAGE REFERENCE SPECTRUM BIASED BY RAW SPECTRUM AVERAGE
V1(i) = V1(i)/V2(i) = X(i)/[BØ(i) + <X(i)>]
THRESHOLD LOOKING FOR
V1(i) ≥ 1.Ø + P(3)*SIGMA
REPORT EVENTS DETECTED ABOVE THRESHOLD, AND SET P(6) = # EVENTS DETECTED ABOVE THRESHOLD
P(6) > Ø?
yes
no
KILL BINS IN V3 WHICH CONTAIN EVENTS
V3(i) = "KILLED" V3(i)
MAKE KILLED V3 ZERO-AVERAGE
V3(i) = V3(i) - <V3(i)>
REREAD OLD ZERO-AVERAGE REFERENCE SPECTRUM
V2(i) = BØ(i)
UPDATE REFERENCE SPECTRUM USING ZERO-AVERAGE RAW DATA NOT EXCEEDING THRESHOLD
V2(i) = Q*V3(i) + (1 - Q)*V2(i)
where Q = 1.Ø - exp{-t/T}
STORE UPDATED BASELINE ON DISK FOR NEXT GO AROUND
BØ(i) = V2(i)
P(6) > Ø ?
no
yes
READ RFI COUNT ARRAY FROM DISK
V3(i) = COUNT(i)
:ADD NEW EVENTS INTO COUNT ARRAY:
GET ADDRESS OF HIT #P(6)
P(1Ø) = HITADR(NHIT #P(6))
GET NUMBER OF PREVIOUSLY STORED EVENTS IN BIN #P(1Ø) OF COUNT ARRAY AND INCREMENT
P(13) = V3(ADR #P(1Ø)) + 1
STORE NEW NUMBER OF EVENTS BACK INTO BIN #P(1Ø) OF COUNT ARRAY
V3(ADR #P(1Ø)) = P(13)
DECREMENT THE EVENT COUNTER
P(6) = P(6) - 1
yes
P(6) > Ø ?
no
STORE NEW COUNT ARRAY ON DISK
COUNT(i) = V3(i)
:REPORT FINAL TALLY OF EVENTS:
READ COUNT ARRAY
V1(i) = COUNT(i)
THRESHOLD LOOKING FOR
V1(i) > Ø
REPORT 15 MOST HIT BINS ORDERED BY NUMBER OF HITS
PLOT ARRAY OF HITS
STOP

Fig. 2. Implementation in the SIP of the candidate all-sky SETI signal-detection algorithm shown in Fig. 1. Note that an extra report has been added that shows the investigator how many times the power in a bin has exceeded the threshold



```
PAR,1,'=',6         (FIRST SPECTRUM USED IN LOOP)
PAR,2,'=',Ø         (AVERAGE OF SPECTRUM, CALCULATED IN LOOP)
PAR,3,'=',4         (THRESHOLD LEVEL IN UNITS OF SIGMA ABOVE AVERAGE)
PAR,4,'=',1Ø4       (NUMBER OF TIMES MAJOR LOOP IS EXECUTED)
PAR,5,'=',3         (NUMBER OF SPECTRA EQUAL TO E-FOLDING TIME)
PAR,6,'=',Ø         (NUMBER OF HITS, CALCULATED IN PROGRAM)
PAR,7,'=',Ø         (INCREMENTAL COUNTER USED IN PROGRAM)
PAR,8,'=',Ø         (HIT ADDRESS, SET IN PROGRAM)
(THE FOLLOWING VARIABLE PARAMETERS ARE ALSO USED IN PROGRAM)
PAR,9,'=',Ø;PAR,1Ø,'=',Ø;PAR,11,'=',Ø;PAR,12,'=',Ø;PAR,13,'=',Ø
PAR,Ø,'?'           (DISPLAY ALL THE PARAMETERS NOW)
%
PARAMETER SETUP FOR SIPØØ2 --- SKY SURVEY RFI ANALYSIS PROGRAM
WHICH LOOPS THROUGH ACCUMULATED SPECTRA AND THRESHOLDS
ON THE QUANTITY
                    P/[<P> + D]

WHERE :
           P = POWER IN INDIVIDUAL BINS OF SPECTRUM
         <P> = AVERAGE OF ALL P
           D = ZERO-AVERAGE REFERENCE SPECTRUM USED IN
               EXPONENTIALLY UPDATED BASELINE

THIS PROGRAM ACCUMULATES THE HITS INTO A NEW SPECTRUM WHICH
ALLOWS THE INVESTIGATOR TO SEE HOW MANY TIMES A PARTICULAR
BIN HAS BEEN HIT OVER THE ENTIRE OBSERVATION PERIOD.


-----------------------------------------------------------------------


FILE'DM1:[31Ø,3Ø1]X1ØSEC.CH1;1'=XF              (DEFINE RAW DATA VECTOR NAME)
FILE'DM1:[31Ø,3Ø1]SIPØØ2.BLV;1'=DF             (DEFINE ZERO AVERAGE BLV NAME)
DF$;GET,2,'K'=DG                                   (READ BLV INTO VECTOR #2)
FILE'DM1:[31Ø,3Ø1]HITCNT.CH1;1'=DC                  (DEFINE HITCOUNT VECTOR)
AVER,1;PAR,2,'A'=APAR                (CALCULATE #1 VECTOR AVERAGE AND SET PAR 2)
APAR$;PAR,2,'*',-1=MAPAR                 (CALCULATE -1*VECTOR AVERAGE SET PAR 2)
AVER,1;THRESH,1,,,3&,'HSG'=THR                           (THRESHOLD VECTOR #1)
CLEAR;REPORT,5,'H';REPORT,1,1,1Ø,5,'D'=RPT     (MAKE HIT REPORT ON CONSOLE)
DC$;SCALE,1,,,Ø,Ø;PUT,1                             (ZERO THE HITCOUNT VECTOR)
?$                                            (DISPLAY ALL DEFINED MACRO NAMES)
%
MACRO COMMAND DEFINITIONS FOR SIPØØ2 --- SKY SURVEY RFI ANALYSIS PROGRAM
```

Fig. 3. The four files that comprise the actual SIP statements implementing the all-sky signal- detection algorithm shown in Fig. 2. The time required to execute one cycle of the major loop is approximately one minute, twenty seconds



```
CLEAR;MSG,5,'BUILDING FIRST GUESS AT ZERO-AVERAGE BASELINE'   (START MESSAGE)
PAR,13,'=',2                                      (SET ITERATION COUNTER = 2)
XF$;NEWVER,1&;GET,1,'K';FILE'?'              (READ THE FIRST RAW DATA VECTOR)
PAR,13,'-',1;IF'LE',13;GOTO,11 (DECRIMENT ITER CNTR;GET OUT IF LOOPED ENOUGH)
COPY,1,2;COPY,1,3                     (COPY VECTOR #1 INTO VECTORS #2 AND #3)
SMOOTH,2;BLN2,1,2,'/'           (SMOOTH #2 BY 65 CHANNELS AND RATIO #1 TO #2)
AVER,1;THRESH,1,,,3.Ø,'HSB'          (LOOK FOR SPIKES IN BASELINED VECTOR #1)
COPY,1,3,'H';KILL,3,1Ø,'HN'         (COPY HEADER TO #3 AND KILL SPIKES IN #3)
COPY,3,1;MAPAR$;SCALE,1,,,2&   (COPY ALL FROM #3 TO #1; MAKE #1 ZERO-AVERAGE)
DF$;PUT,1;GOTO,4                     (STORE THE FIRST GUESS ZERO-AVERAGE BLV)
PLOT,1,'G:FIRST GUESS AT ZERO-AVERAGED BASELINE';CLEAR          (END MESSAGE)
%
FIRST BASELINE SETUP FOR SIPØØ2 --- SKY SURVEY RFI ANALYSIS PROGRAM
GET FIRST GUESS AT ZERO-AVERAGE BASELINE FROM FIRST SPECTRUM



------------------------------------------------------------------------


PAR,4,'-',1;IF'LT',4;GOTO,12                            (MASTER LOOP COUNTER)
XF$;NEWVER,1&;GET,1,'K';APAR$;COPY,1,3;PAR,1,'+',1;PAR,1Ø,'N' (READ RAW DATA)
PAR,1,'?','NEXT VECTOR';DG$;SCALE,2,,,2&;BLN2,1,2,'/' (CALCULATE   X/[D+<X>])
THR$;RPT$;PAR,6,'HN';IF'LE',6;GOTO,6 (THRESH; DIRECTLY UPDATE BLV IF NO HITS)
COPY,1,3,'H';KILL,3,1,'HN'             (KILL HITS IN RAW DATA FOR BLV UPDATE)
AVER,3;PAR,2,'A';PAR,2,'*',-1;SCALE,3,,,2&  (MAKE KILLED VECTOR ZERO-AVERAGE)
DG$;EXP,2,3,,,5&;DF$;PUT,2          (UPDATE BLV WITH KILLED VECTOR AND STORE)
AREA,1;PAR,6,'HN';IF'LE',6;GOTO,1    (SKIP REST AND CONTINUE LOOP IF NO HITS)
DC$;GET,3                                       (READ IN THE HIT COUNT ARRAY)
AREA,1;PAR,1Ø,'HA',6&;AREA,3;PAR,13,'C',1Ø&;PAR,13,'+',1;-
PAR,13,'P',1Ø&;PAR,6,'-',1;IF'GT',6;GOTO,1Ø            (COUNT HITS INTO BINS)
DC$;PUT,3;GOTO,1             (STORE UPDATED HITCOUNT ARRAY AND CONTINUE LOOP)
DC$;GET,1;THRESH,1,,,Ø,'G';CLEAR;MSG,5,'REPORT OF 15 MOST FREQUENTLY HIT CHANNELS
REPORT,1,1,15,,5,'D';WAIT;PLOT,1,'G:HIT COUNT ARRAY';CLEAR     (FINAL REPORT)
%
ACTION PROGRAM FOR SIPØØ2 --- SKY SURVEY RFI ANALYSIS PROGRAM
EXECUTE THE FOLLOWING FILES IN THE ORDER SPECIFIED:

              SIPØØ2.ØØ1   ---   PARAMETER SETUP
              SIPØØ2.ØØ2   ---   MACRO DEFINITION
              SIPØØ2.ØØ3   ---   INITIAL BASELINE CALCULATION
              SIPØØ2.ØØ4   ---   THIS FILE
```

Fig. 3 (contd)

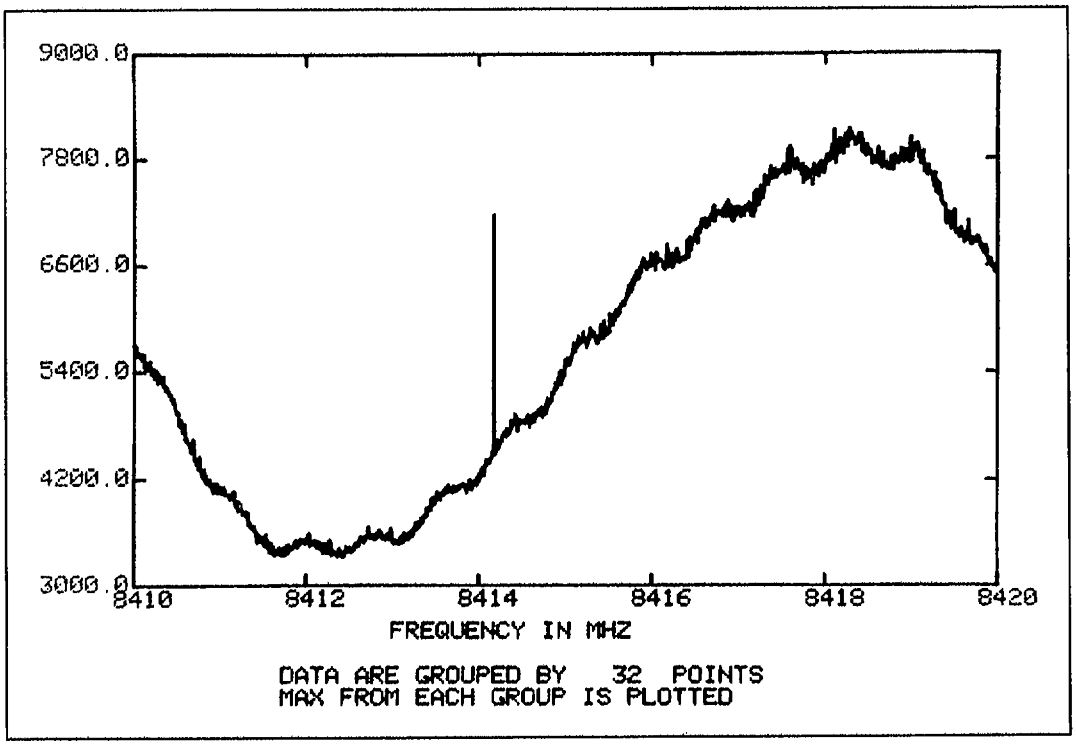


Fig. 4. A 32-kbin spectrum before application of a baseline. Note the RFI at 8414.3 MHz

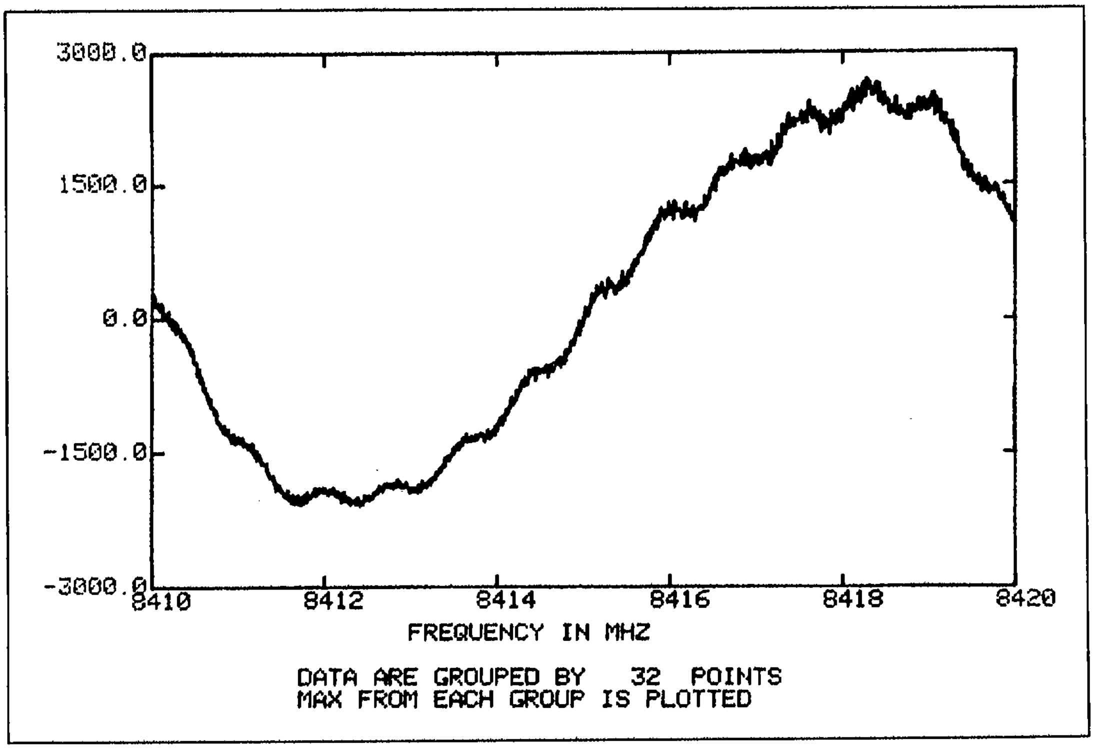
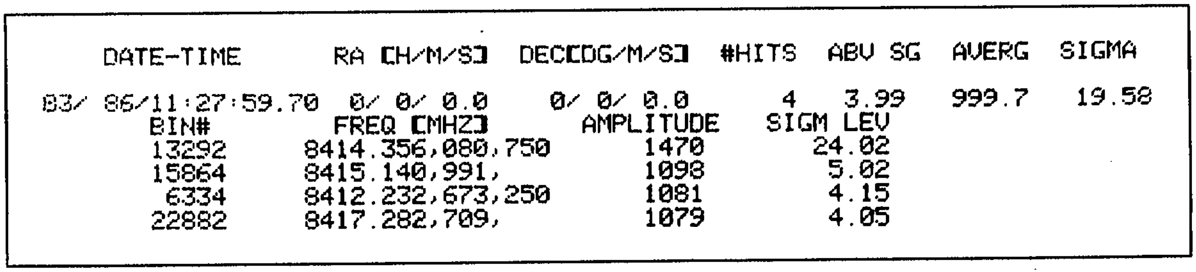
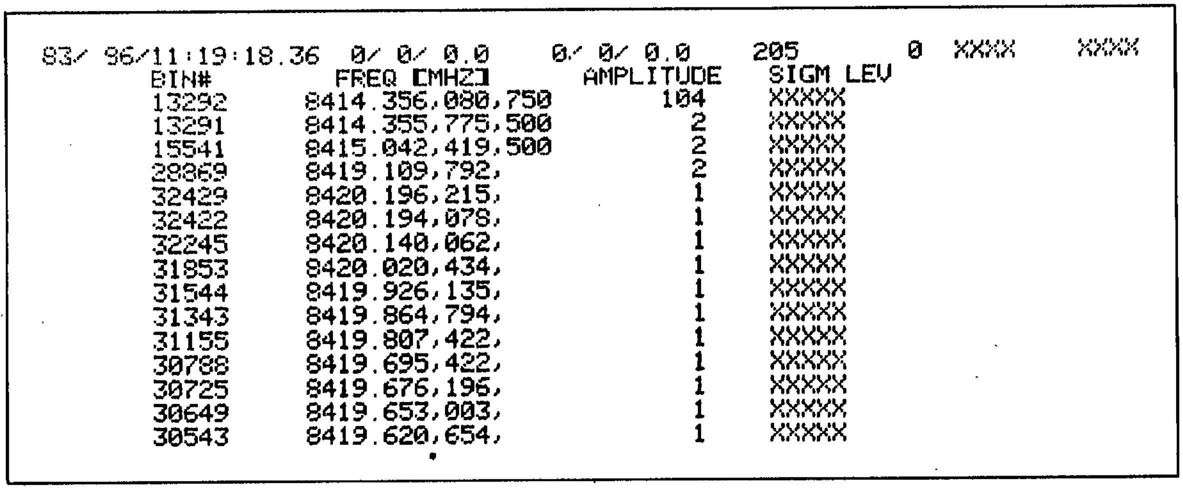


Fig. 5. The first derived zero-average reference spectrum. Note that the structure present in Fig. 4 is preserved, with the exception of the narrowband feature at 8414.3 MHz



```
  DATE-TIME         RA [H/M/S]  DEC[DG/M/S]  #HITS  ABV SG  AVERG  SIGMA

83/ 86/11:27:59.70  0/ 0/ 0.0    0/ 0/ 0.0     4    3.99   999.7   19.58
      BIN#          FREQ [MHZ]         AMPLITUDE   SIGM LEV
      13292      8414.356,080,750        1470        24.02
      15864      8415.140,991,           1098         5.02
       6334      8412.232,673,250        1081         4.15
      22882      8417.282,709,           1079         4.05
```

Fig. 6. A representative report after thresholding a baselined spectrum at a level of 4*SIGMA above its average



```
83/ 96/11:19:18.36  0/ 0/ 0.0    0/ 0/ 0.0    205        0  XXXX     XXXX
      BIN#          FREQ [MHZ]         AMPLITUDE   SIGM LEV
      13292      8414.356,080,750        104      XXXXX
      13291      8414.355,775,500          2      XXXXX
      15541      8415.042,419,500          2      XXXXX
      28869      8419.109,792,             2      XXXXX
      32429      8420.196,215,             1      XXXXX
      32422      8420.194,078,             1      XXXXX
      32245      8420.140,062,             1      XXXXX
      31853      8420.020,434,             1      XXXXX
      31544      8419.926,135,             1      XXXXX
      31343      8419.864,794,             1      XXXXX
      31155      8419.807,422,             1      XXXXX
      30788      8419.695,422,             1      XXXXX
      30725      8419.676,196,             1      XXXXX
      30649      8419.653,003,             1      XXXXX
      30543      8419.620,654,             1      XXXXX
```

Fig. 7. The accumulated RFI report showing the 15 most frequently hit bins after baselining and thresholding 104 spectra at the level of four standard deviations above their means. The interference at 8414.3 MHz was detected in all spectra

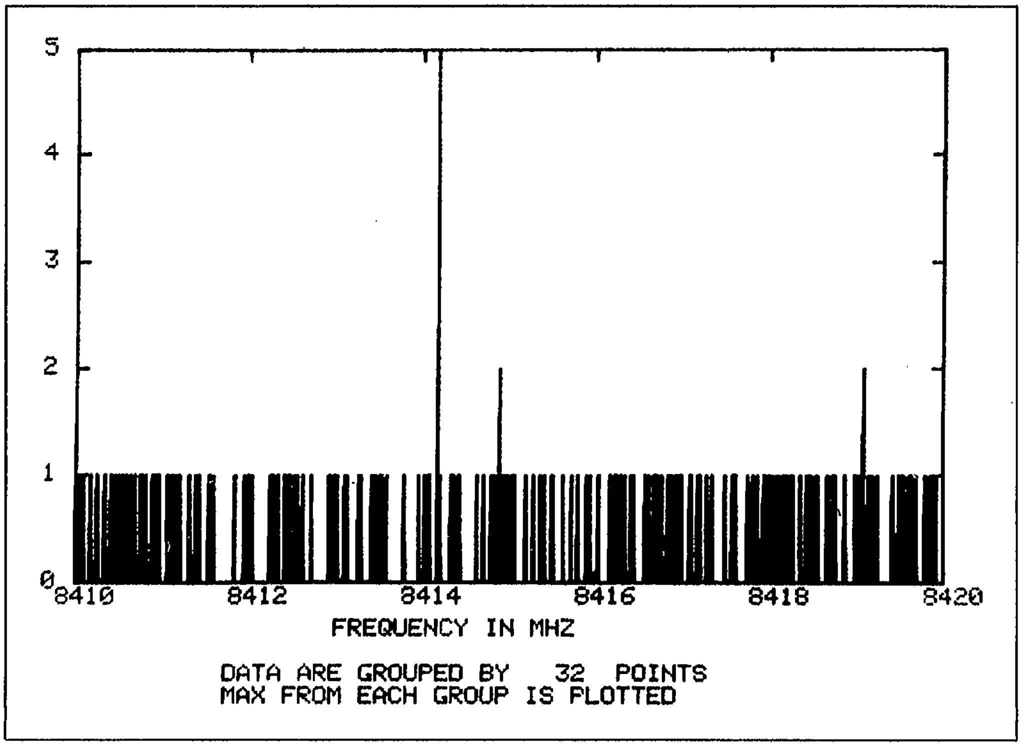


Fig. 8. The accumulated RFI report for the entire 10-MHz frequency interval. The RFI at 8414.3 MHz has been arbitrarily set to 5 so that the distribution of the events due to random noise is visible. A power spectral analysis shows the events to be randomly distributed over the 32 Kbins

Appendix

Alphabetical Listing of SIP Tasks

TASK #	TASK NAME AND PARAMETERS AND SUMMARY DESCRIPTION
1	ACCUM[,NAREA,NBIN1,NBIN2][,'CHAR'] Performs floating point accumulation and averaging of spectra.
2	AREA,NAREA Sets spectrum area for PAR command.
3	AVER,[NAREA,NBIN1,NBIN2,LU][,'?'] Calculates statistics of spectrum.
4	BELL Sounds tone at user console.
5	BLN2[,NAREA1,NAREA2,NBIN1,NBIN2,SCALE][,'CHAR'] Calculates quotient or difference of two spectra.
6	BYE Exits SIP.
7	CLEAR Erases Tektronix terminal screen.
8	CLIP[,NAREA,NBIN1,NBIN2,R1][,R2][,'CHARs'] Limits range of amplitudes in spectra.
9	CMT[,NAREA,NWORD,LU],'CHARACTER_STRING' Places ASCII comment into spectrum header.
10	COPY[,NAREA1,NAREA2][,'CHARs'] Copies spectrum from one memory area to another.
11	CURRNT[,LU] Displays SIP parameters for debugging purposes.
12	DATE[,LU] Displays current date and time on user console.
13	DB[,NAREA,NBIN1,NBIN2,NSCALE] Makes spectrum amplitudes logarithmic.
14	EXP[,NAREA1,NAREA2,NBIN1,NBIN2][,R][,'CHAR'] Exponentially updates baseline spectrum using current spectrum.
15	FFT[,NAREA,NBIN,N] Calculates power spectrum of spectrum amplitudes.

TASK #	TASK NAME AND PARAMETERS AND SUMMARY DESCRIPTION
16	FILE'NEWFILENAME'[,LU] Defines disk file name used by GET and PUT.
17	*FIT[,NAREA,NBIN1,NBIN2,NORDER,LU][,'CHAR'] Makes Least-squares fit to features of spectrum.
18	FRQ[,NAREA,R,LU][,'CHAR'] Makes conversion between bin number and frequency and displays on user console.
19	GET[,NAREA,NBLK1,NBLKS][,'K'] Reads spectrum from disk file into memory.
2Ø	GOTO,LINE Passes program control to new command line.
21	GROUP[,NAREA,NBIN1,NBIN2,NGROUP][,'CHAR'] Compresses spectrum by combining adjacent bins.
22	HELP Invokes facility which describes commands in detail.
23	HIST[,NAREA,NBIN1,NBIN2] Invokes statistical analysis package.
24	HRDCPY Triggers Tektronix console hardcopy.
25	IF,NPAR,'COND'[,VALUE] Logically compares value of dynamical variable to a value and skips rest of command line if false.
26	KILL[,NAREA,NVAL][,'CHARs'][,NBIN1,NBIN2,NBIN3,...] Adjusts amplitude in identified bins.
27	LEVEL,R[,'S'] Sets amplitude for CLIP, KILL, and THRESH
28	MSG,LU,'CHARACTER_STRING' Sends message to console.
29	NEWVER[,NVAL,'CHAR'] Changes file name version.
3Ø	PAR,NPARM[,R1][,'OP1'][,R2,'OP2', ,RN,'OPN'] Calculate value and set dynamical variable equal to it.
31	PEEK,NAREA,NBIN1,NBIN2[,LU][,'H'] Displays amplitudes of bins or values in header of spectrum on user console.

TASK #	TASK NAME AND PARAMETERS AND SUMMARY DESCRIPTION
32	PICK[NAREA,NBIN1,NBINS,NSTEP] Compresses spectrum by selecting equally spaced bins.
33	PLOT[,NAREA,R1,R2,'CHARs:heading character string'] Display spectrum on Tektronix terminal.
34	POKE,NAREA,NBIN1[,NBIN2],K1[,K2, ,Kn][,'H'] Places values into bins or header of spectrum.
35	PUT[,NAREA,NBLK1,NBLKS] Writes spectrum from memory into disk file.
36	RANGE,NBIN1,NBIN2 Specifies range of bins in spectrum over which all other tasks will operate.
37	RASTR,NAREA,NBIN,N Places time history of a bin into single spectrum.
38	REPORT[,NAREA,NHIT1,NHIT2,LU][,'CHAR'] Writes report of events crossing threshold on console.
39	SCALE[,NAREA,NBIN1,NBIN2,ADD,SCALE] Rescales amplitudes of bins of spectrum.
4Ø	SIGNAL[,NAREA,NBIN1,NBIN2,LU][,'CHAR'] Generates artificial signal and writes into spectrum.
41	SKIP[,NSKIP,LU][,'P'] Controls line printer pagination.
42	SMOOTH[,NAREA,NBIN1,NBIN2,NHALF] Smooths amplitudes in bins of spectrum with boxcar filter.
43	THRESH[,NAREA,NBIN1,NBIN2,T1][,T2][,'CHARs'] Applies threshold to spectrum and records events which exceed it.
44	TIME[,LU] Times SIP tasks.
45	WAIT Pauses SIP execution until console key is pressed.

* Task not yet implemented.
Parameters enclosed by [ ] may be omitted
if default values apply.