NASA

Technical Memorandum 83997

Interactive Digital Signal Processor

(NASA-TM-83997) INTERACTIVE DIGITAL SIGNAL PROCESSOR (NASA) 167 p HC A08/MF A01 CSCL 09B N84-34974

Unclas

G3/61 23127

W. H. Mish, R. M. Wenger, K. W. Behannon, and J. B. Byrnes

SEPTEMBER 1982
(Revised May 1984)

National Aeronautics and
Space Administration

Goddard Space Flight Center
Greenbelt, Maryland 20771

INTERACTIVE DIGITAL SIGNAL PROCESSOR

by

William H. Mish
Laboratory for Extraterrestrial Physics
NASA/Goddard Space Flight Center
Greenbelt, Maryland 20771

Ray M. Wenger
Computer Science Corporation
8728 Colesville Rd
Silver Spring, Maryland 20910

Kenneth W. Behannon
Laboratory for Extraterrestrial Physics
NASA/Goddard Space Flight Center
Greenbelt, Maryland 20771

James B. Byrnes
Laboratory for Extraterrestrial Physics
NASA/Goddard Space Flight Center
Greenbelt, Maryland 20771

September 1982
Revised May 1984

PREFACE TO THE SECOND EDITION

This second edition of the IDSP documentation, in addition to a number of minor corrections and updates to the original 1982 documentation, contains substantial new material. Since the Fall of 1982 five new operators have been added and are included in this edition.

RECPOL provides for rectangular to polar, and polar to rectangular rotations, ROTATE provides for coordinate rotations in conjuction with, or independent of, the EIG operator. MEM implements the Maximum Entropy Method for computing power spectra. And FILOPT and WTS aid in producing an optimal digital filter design in conjuction with FILDES.

Section 5.1 has been expanded to contain an example of the use of IDSP for the analysis of waves. This example demonstrates the use of operator EIG to perform an eigenfunction analysis on the components of a vector time series, Fourier transformed to the frequency domain, which reveals the polarization characteristics of the data as a function of frequency. In the process it was desirable to digitally filter the data and the example includes information on how to optimize, using FILOPT, a two band digital filter designed with the Remez exchange algorithm.

Section 5.2 has been expanded to discuss the use of FILOPT and WTS in the design of optimal digital filters and several examples are presented. See particularly Fig. 5.2-21 for an overview of the total filter design procedure.

ABSTRACT

The Interactive Digital Signal Processor (IDSP) is implemented on a VAX 11/780 under VMS. It consists of a set of time series analysis "Operators" each of which operates on an input file to produce an output file; the operators can be executed in any order that makes sense and recursively, if desired. The operators are the various algorithms that have been used in digital time series analysis work over the years. In addition, there is provision for user written operators to be easily interfaced to the system. The system can be operated both interactively and in batch mode.

In IDSP a file can consist of up to n (currently n=8) simultaneous time series. Thus storage for a file can be subdivided such that it is used, for example, entirely for one long single time series or for as many as n shorter time series, such as the components of a vector. An operator always operates simultaneously on all of the time series in a file.

IDSP currently includes over thirty standard operators that range from Fourier transform operations (FFT,FFTIN,WINDOW,SPECT), design and application of digital filters (FILDES,FILOPT,FILTER,WTS), eigenvalue analysis (EIG), to operators that provide graphical output (GRAPH,GRAFCK), allow batch operation (REDO), editing (CONCAT,EDHIST,EDIT,INTERP,SUBSET,SUBSER) and display information (SHOW, CMDHIS). The complete set of standard operators is listed below.

AVER, CMDHIS, CONCAT, COPY, DCL, DSTAT, EDHIST, EDIT, EIG, FILDES, FILOPT, FILTER, FFT, FFTIN, FLOP, GRAFCK, GRAPH, INTERP, MEM, MNFLD, NORM, RECPOL, REDO, ROTATE, SETUP, SHOW, SPECT, SUBSER, SUBSET, TRACE, WINDOW, WTS, STOP.

IDSP is being used extensively to process data sets obtained from scientific experiments onboard spacecraft such as Dynamics Explorer, ISEE, IMP and Voyager. In addition IDSP provides an excellent teaching tool for demonstrating the application of the various time series operators to artifically-generated signals.

IDSP is available from the Computer Software Management and Information Center (COSMIC), 112 Barrow Hall, University of Georgia, Athens, Georgia 30602, Program Number GSC-12862.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1.0) INTRODUCTION TO IDSP

Many, but of course not all, of the phenomena that we want to measure in the real world are of an analog nature. Often it is desirable, perhaps essential, that we be able to process these measurements on a digital device in order to be able to extract the desired information from the raw data. The process of mapping the analog phenomena to the digital world is called sampling. When a phenomenon is sampled as a function of time, what results is a digital time series.

The design of the instrumentation to perform this sampling is, of course, critical. How often the instrument samples the phenomenon (to avoid aliasing), the time between samples (is it constant?), the number of bits of digitization (precision), and the transfer function of the instrument all have a large affect on the ability of the researcher to analyze the data properly.

Fig. 1.0-1 is a schematic of this process, i.e., an analog phenomenon is sampled as a function of time and digitized, and what results is a digital time series.

Often the phenomenon of interest requires that multiple parameters of the phenomenon be sampled similtaneously. As an example: if the phenomenon of interest is a vector quantity, then it is required that all of the components of the vector be sampled. This, in turn, will result in there being not one but "n" digital time series generated.

The processing of these digital time series to extract information usually is done with a number of fairly standard operations that are applied to the data serially, but, of course, not always in the same order. It may also be very desirable to "experiment" with various operations to explore their effect on the results.

Some of the more common operations that come to mind are the editing of data to remove "bad" points, interpolation of the data during data drop-outs, digital filtering of data to remove unwanted signals, and discrete Fourier transformation of data so that it can be viewed in the frequency domain, to mention but a few.

It was the above mentioned concepts that dictated the design of the Interactive Digital Signal Processor software.

IDSP was designed and implemented to be extremely interactive and easy to use. Essentially, it consists of a set of "operators" each of which operates on an input file to produce an output file; the operators can be executed in any order that makes sense with respect to the task to be accomplished and recursively, if desired. The operators themselves are simply the various algorithms that have been used in digital time series analysis work over the years. Some specific examples are: an operator FILTER to filter a time series, an operator FFT to perform a discrete Fourier Transform on a time series, an operator INTERP to interpolate across gaps in a time series etc. In addition, there is provision for user written operators to be easily interfaced to the system.

In order for an operator to process an input file to produce an output file, it is necessary that the input file have a "name". In IDSP the name of the input file is simply the name of the last operator to process this file. Thus, if we have just interpolated a file, the name of this file is now INTERP. If we now filter INTERP, its name becomes FILTER. In this fashion a file progresses through the various operations. Note that when the data are first introduced into the system the name of the file is SETUP.

The only exception to this rule of the output file being named exactly for the operator that just processed it is the case where an operator produces more than a single output file. An example is the spectral density matrix operation SPECT which, for technical reasons, produces three output files SPECTD and SPECTOFF containing the diagonal elements of the matrix and off-diagonal elements, respectively and COHPH which contains the coherence and phase information arising from the off-diagonal terms.

So far we have treated the terms "time series" and files as if they are the same thing; however, there is a distinction. In IDSP a file can consist of up to n (currently n=8) simultaneous time series, as long as each tuple of the file has the same time tag (see Fig. 1.0-2). Thus storage for a file can be subdivided such that it is used, for example, entirely for one long single time series or for as many as n shorter time series (total space not to exceed 500,000 real points, operations involving FFT will be smaller due to working arrays). An operator always operates simultaneously and identically on all of the time series in a file regardless of whether there is one, two, ..., or n. This distinction is an important aspect of IDSP when one considers that many time series consist of multiple components, e.g., vector components.

To allow the user to keep track of what operations have been done to a file, an integral part of each file is history information that explicitly records the history of operations performed. The history information is very useful because the processing of a file can involve many operators in a nontrivial application, and it is quite easy to forget just what operations have been applied. Any time the user wishes to find out this history it is only necessary to type in SHOW "file name" and the history of the file to this point will be displayed on the terminal. Also when a file is graphed with the Operator GRAPH the history information is displayed along with the graph.

IDSP processes the time series according to two basic concepts: SPAN and INTERVAL (see Fig. 1.0-3). The Interval is defined as the basic time segment to be analyzed--the segment of data that is to be filtered or Fourier transformed, etc. by an operator. The Span is defined to be the total time under analysis--composed of one or more contiguous Intervals.

IDSP is designed to operate on a DEC VAX 11/780 and thus the notion of DIRECTORIES needs to be understood. Since a disk on the VAX can contain files belonging to many different users, each disk has a set of files called directories, i.e., a catalog of the files on that disk that belong to a particular user. When the user "Signs On" the VAX the user is automatically put into his/her default directory. IDSP uses one sub-directory of this default directory: [Userid.IDSP]. The sub-directory [Userid.IDSP] is used to store input parameters required by some of the operators, e.g., in order to design a filter there is an operator called FILDES which requires that certain design parameters be supplied; these are stored in sub-directory

[Userid.IDSP]. The same sort of thing is true of operator SETUP.

[Userid.IDSP] is also used to store the executable version of IDSP and all of the working files of IDSP and it is out of this sub-directory that execution takes place. The execution of an operator, as previously explained, generates an output data set. Only the latest version of each data set is retained by IDSP, e.g., multiple execution of the same operator over-writes previous versions of the data set unless the COPY operator is used to save the previous version.

<u>IDSP includes the following Operators:</u>

1. AVER--average every n points
2. CMDHIS--give listing of commands issued during current session
3. CONCAT--concatenate two different datasets
4. COPY--copy one dataset to another
5. DCL--allow user to enter VAX DCL commands and return to IDSP
   Decimation--reduce data as option when filtering with operator FILTER
6. DSTAT--compute dataset statistics
7. EDHIST--allow user to edit history of dataset
8. EDIT--display series and interactively edit on a HP2648A Graphics terminal.
9. EIG--rotate spectral matrices to eigenvalue coordinates
10. FILDES--design filter via Remez Exchange method
11. FILOPT--aids in producing an optimal filter design
12. FILTER--filter data
13. FFT--discrete Fast Fourier Transform
14. FFTIN--inverse discrete Fourier Transform
15. FLOP--interactively select printout device during execution
16. GRAFCK--check to see if subprocess executing graph job has finished
17. GRAPH--plot each series in the dataset
18. INTERP--interpolation
19. MEM--Maximum Entropy method of computing power spectra
20. MNFLD--transform a vector to mean field coordinates
21. NORM--normalize data
22. RECPOL--rectangular to polar; polar to rectangular rotations
23. REDO--repeat command sequence in batch jobs
24. ROTATE--arbitrary coordinate rotation
25. SETUP--call application interface INPUT, place data into proper configuration for analysis
26. SHOW--display history and optionally data of given dataset
27. SPECT--form spectral matrices
28. SUBSER--extract specified series from dataset
29. SUBSET--create a subset of a given dataset
30. TRACE--form a trace of specified dataset
31. WINDOW--choose data window, with option to pad with zeroes
32. WTS--compute the number of digital filter weights
33. STOP--termination procedure

In addition to the above standard operators provision has been made for the user to write their own operators and interface them to IDSP (see Appendix B for details).

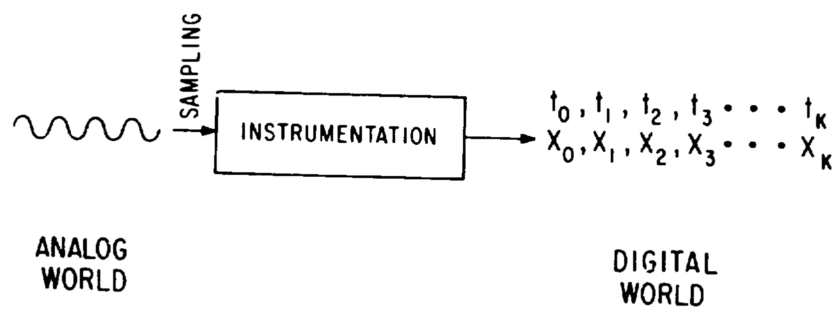
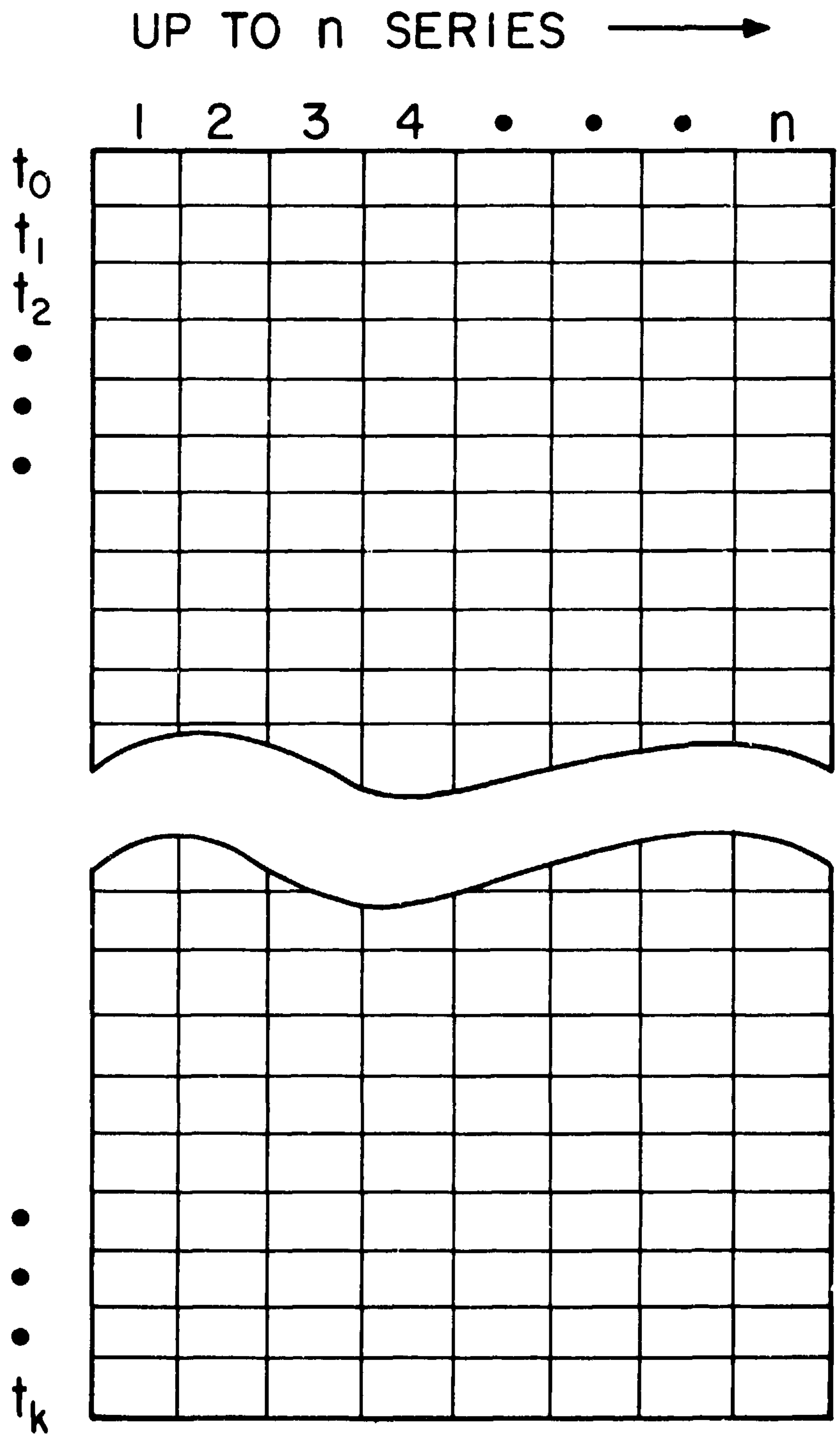


Fig. 1.0-1

A schematic of mapping an analog phenomenon to the digital world.



$$t_1 - t_0 = \Delta T = \bullet \bullet \bullet \bullet = t_k - t_{k-1}$$

Fig. 1.0-2

In IDSP a file can consist of up to n simultaneous time series as long as each tuple has the same time tag.

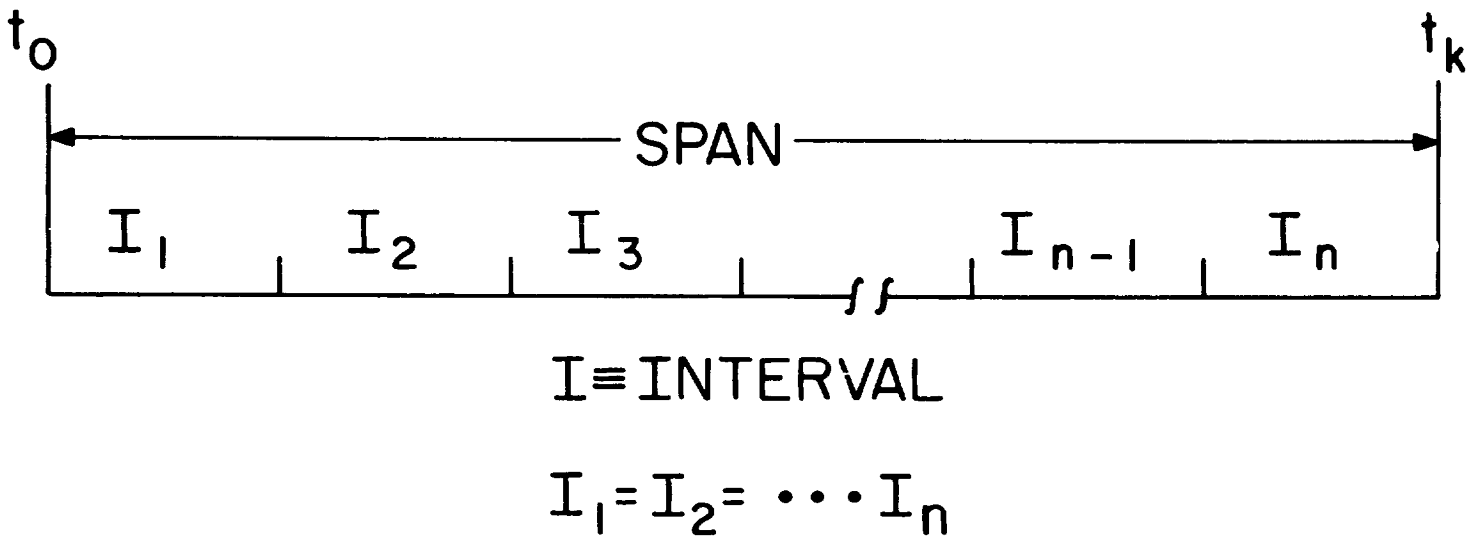


Fig. 1.0-3

IDSP processes the time series according to two basic concepts: SPAN and INTERVAL. The Interval is defined as the basic time segment to be analyzed-- the segment of data that is to be filtered or Fourier transformed, etc., by an operator. The Span is defined to be the total time under analysis-- composed of one or more contiguous Intervals.

2.0) AN EXAMPLE OF THE USE OF IDSP

The following example illustrates how to use IDSP. On the Dynamics Explorer Spacecraft there is a magnetic field experiment that provides a vector measurement of the ambient magnetic field every 0.5 sec. The data from one of the components of the vector is shown in Fig. 2.0-1 as a function of time. This plot was obtained by using operator GRAPH on data set SETUP.

As is evident there appears to be substantial periodic signal riding on top of the actual signal of interest. It was desired to remove this contamination from the signal of interest. First it was necessary to do a spectrum analysis on the data to determine the frequency(s) of the unwanted signals present. Fig 2.0-2 shows the results of using operators WINDOW, FFT, SPECT, and GRAPH on this time series, note that at 0.0324 and 0.175 Hz there are peaks in the spectrum resulting from the unwanted signals.

We now use operator FILDES to design a finite impulse response (FIR) filter to remove both unwanted frequencies. In this case we use a single filter that has two stopbands and three passbands for a total of 5 bands. Operator FILDES will permit you to design a FIR filter with up to 10 bands total.

The transfer function of this 5 band filter is shown in Fig. 2.0-3. This transfer function is obtained by using operator FFT on data set WINDOW which contains the windowed filter coefficients. The plot is obtained with operator GRAPH. Table 2.0-1 results from the execution of operator FILDES and presents the coefficients and other parameters of this 5 band filter. Fig. 2.0-4 shows the results of using operator FILTER to filter the original time series shown in Fig. 2.0-1 with this 5 band filter.

The IDSP command sequence without required and optional operands is shown below:

SETUP	input Dynamics Explorer data into IDSP
GRAPH SETUP	plot the raw data
WINDOW SETUP	window the raw data
FFT WINDOW	Fourier transform the windowed data
SPECT FFT	generate the spectral matrices
GRAPH SPECTD	plot the transformed data to obtain the power spectrum
FILDES	design the 5 band digital filter
WINDOW FILDES	window the filter coefficients
FFT WINDOW	Fourier transform the windowed coefficients to obtain the transfer function of the filter
GRAPH FFTMP	plot the filter transfer function
FILTER SETUP	digital filter the raw data
GRAPH FILTER	plot the filtered data

****

FINITE IMPULSE RESPONSE (FIR)
LINEAR PHASE DIGITAL FILTER DESIGN
REMEZ EXCHANGE ALGORITHM

BANDPASS FILTER

FILTER LENGTH = 251

***** IMPULSE RESPONSE *****

H(1)	=	0.15814120E-02	=	H(251)
H(2)	=	-0.34383922E-02	=	H(250)
H(3)	=	0.14056013E-02	=	H(249)
H(4)	=	0.24078514E-02	=	H(248)
H(5)	=	0.19100166E-02	=	H(247)
H(6)	=	0.11902003E-02	=	H(246)
H(7)	=	0.78469788E-03	=	H(245)
H(8)	=	0.82009403E-03	=	H(244)
H(9)	=	0.12092897E-02	=	H(243)
H(10)	=	0.17456150E-02	=	H(242)
H(11)	=	0.21647078E-02	=	H(241)
H(12)	=	0.22298915E-02	=	H(240)
H(13)	=	0.18296[illegible]54E-02	=	H(239)
H(14)	=	0.10444109E-02	=	H(238)
H(15)	=	0.12637804E-03	=	H(237)
H(16)	=	-0.60599874E-03	=	H(236)
H(17)	=	-0.92391594E-03	=	H(235)
H(18)	=	-0.80711755E-03	=	H(234)
H(19)	=	-0.45952279E-03	=	H(233)
H(20)	=	-0.20076071E-03	=	H(232)
H(21)	=	-0.28555829E-03	=	H(231)
H(22)	=	-0.75000722E-03	=	H(230)
H(23)	=	-0.13736545E-02	=	H(229)
H(24)	=	-0.17928836E-02	=	H(228)
H(25)	=	-0.17053958E-02	=	H(227)
H(26)	=	-0.10633392E-02	=	H(226)
H(27)	=	-0.13580504E-03	=	H(225)
H(28)	=	0.60757459E-03	=	H(224)
H(29)	=	0.73501718E-03	=	H(223)
H(30)	=	0.92054295E-04	=	H(222)
H(31)	=	-0.10959945E-02	=	H(221)
H(32)	=	-0.23258266E-02	=	H(220)
H(33)	=	-0.30748146E-02	=	H(219)
H(34)	=	-0.30839785E-02	=	H(218)
H(35)	=	-0.25047711E-02	=	H(217)
H(36)	=	-0.18128060E-02	=	H(216)
H(37)	=	-0.15360778E-02	=	H(215)
H(38)	=	-0.19327183E-02	=	H(214)
H(39)	=	-0.28102121E-02	=	H(213)
H(40)	=	-0.35964199E-02	=	H(212)
H(41)	=	-0.36401507E-02	=	H(211)
H(42)	=	-0.25890961E-02	=	H(210)
H(43)	=	-0.63115772E-03	=	H(209)
H(44)	=	0.15547265E-02	=	H(208)
H(45)	=	0.31153709E-02	=	H(207)
H(46)	=	0.34618520E-02	=	H(206)
H(47)	=	0.25875489E-02	=	H(205)
H(48)	=	0.10776789E-02	=	H(204)
H(49)	=	-0.20811461E-03	=	H(203)
H(50)	=	-0.61613688E-03	=	H(202)
H(51)	=	-0.81341735E-04	=	H(201)
H(52)	=	0.82465919E-03	=	H(200)
H(53)	=	0.11941771E-02	=	H(199)
H(54)	=	0.34294141E-03	=	H(198)
H(55)	=	-0.17857827E-02	=	H(197)
H(56)	=	-0.44503259E-02	=	H(196)
H(57)	=	-0.65120440E-02	=	H(195)
H(58)	=	-0.69681485E-02	=	H(194)
H(59)	=	-0.55483943E-02	=	H(193)
H(60)	=	-0.29113430E-02	=	H(192)
H(61)	=	-0.31874361E-03	=	H(191)
H(62)	=	0.98161516E-03	=	H(190)
H(63)	=	0.43110389E-03	=	H(189)
H(64)	=	-0.15216360E-02	=	H(188)
H(65)	=	-0.36827554E-02	=	H(187)
H(66)	=	-0.47876472E-02	=	H(186)
H(67)	=	-0.42325114E-02	=	H(185)
H(68)	=	-0.24825933E-02	=	H(184)
H(69)	=	-0.86959032E-03	=	H(183)
H(70)	=	-0.85432507E-03	=	H(182)
H(71)	=	-0.31593335E-02	=	H(181)
H(72)	=	-0.72583547E-02	=	H(180)
H(73)	=	-0.11520541E-01	=	H(179)
H(74)	=	-0.13979098E-01	=	H(178)
H(75)	=	-0.13352104E-01	=	H(177)
H(76)	=	-0.97541511E-02	=	H(176)
H(77)	=	-0.46772487E-02	=	H(175)
H(78)	=	-0.21605555E-03	=	H(174)
H(79)	=	0.20538503E-02	=	H(173)
H(80)	=	0.19452734E-02	=	H(172)
H(81)	=	0.77985239E-03	=	H(171)
H(82)	=	0.64122514E-03	=	H(170)
H(83)	=	0.31412258E-02	=	H(169)
H(84)	=	0.83715729E-02	=	H(168)
H(85)	=	0.14660291E-01	=	H(167)
H(86)	=	0.19336510E-01	=	H(166)
H(87)	=	0.20136783E-01	=	H(165)
H(88)	=	0.16507745E-01	=	H(164)
H(89)	=	0.10078511E-01	=	H(163)
H(90)	=	0.39618891E-02	=	H(162)
H(91)	=	0.12056866E-02	=	H(161)
H(92)	=	0.31835234E-02	=	H(160)
H(93)	=	0.88162953E-02	=	H(159)
H(94)	=	0.15116510E-01	=	H(158)
H(95)	=	0.18813241E-01	=	H(157)
H(96)	=	0.18231031E-01	=	H(156)
H(97)	=	0.14352880E-01	=	H(155)
H(98)	=	0.10422706E-01	=	H(154)
H(99)	=	0.10182290E-01	=	H(153)
H(100)	=	0.15647490E-01	=	H(152)
H(101)	=	0.25635440E-01	=	H(151)
H(102)	=	0.35953119E-01	=	H(150)
H(103)	=	0.41306213E-01	=	H(149)
H(104)	=	0.38028747E-01	=	H(148)
H(105)	=	0.26212333E-01	=	H(147)
H(106)	=	0.99809859E-02	=	H(146)
H(107)	=	-0.43794718E-02	=	H(145)
H(108)	=	-0.11643316E-01	=	H(144)
H(109)	=	-0.10513707E-01	=	H(143)
H(110)	=	-0.45787501E-02	=	H(142)
H(111)	=	-0.57229016E-03	=	H(141)
H(112)	=	-0.46508880E-02	=	H(140)
H(113)	=	-0.18505570E-01	=	H(139)
H(114)	=	-0.37433039E-01	=	H(138)
H(115)	=	-0.51723648E-01	=	H(137)
H(116)	=	-0.51059123E-01	=	H(136)
H(117)	=	-0.30053660E-01	=	H(135)
H(118)	=	0.77681956E-02	=	H(134)
H(119)	=	0.49811829E-01	=	H(133)
H(120)	=	0.78650512E-01	=	H(132)
H(121)	=	0.79160124E-01	=	H(131)
H(122)	=	0.35535777E-01	=	H(130)
H(123)	=	-0.15060277E-01	=	H(129)
H(124)	=	-0.84144048E-01	=	H(128)
H(125)	=	-0.13848756E+00	=	H(127)
H(126)	=	0.84695556E+00	=	H(126)

	BAND 1	BAND 2	BAND 3	BAND 4
LOWER BAND EDGE	0.0000000	0.0155000	0.0270000	0.0600000
UPPER BAND EDGE	0.0055000	0.0170000	0.0500000	0.1150000
DESIRED VALUE	1.0000000	0.0000000	1.0000000	0.0000000
WEIGHTING	1.0000000	8.0000000	1.0000000	8.0000000
DEVIATION	0.0103347	0.0012918	0.0103347	0.0012918
DEVIATION IN DB	0.0893056	-57.7758598	0.0893056	-57.7758598

	BAND 5
LOWER BAND EDGE	0.1250000
UPPER BAND EDGE	0.5000000
DESIRED VALUE	1.0000000
WEIGHTING	1.0000000
DEVIATION	0.0103347
DEVIATION IN DB	0.0893056

EXTREMAL FREQUENCIES--MAXIMA OF THE ERROR CURVE

0.0000000	0.0039683	0.0055000	0.0155000	0.0162441
0.0170000	0.0270000	0.0282401	0.0309683	0.0346885
0.0381607	0.0418809	0.0456012	0.0485773	0.0500000
0.0627282	0.0654563	0.0689285	0.0724007	0.0761210
0.0798412	0.0835614	0.0872816	0.0910019	0.0947221
0.0984423	0.1021625	0.1056347	0.1091070	0.1120831
0.1140673	0.1150000	0.1250000	0.1262401	0.1292163
0.1329365	0.1369047	0.1408730	0.1448412	0.1488094
0.1530257	0.1569939	0.1612102	0.1651784	0.1691467
0.1733629	0.1773311	0.1815474	0.1855156	0.1897319
0.1937001	0.1976684	0.2018846	0.2058529	0.2098211
0.2140374	0.2180056	0.2219738	0.2261901	0.2301583
0.2341266	0.2383428	0.2423111	0.2462793	0.2594956
0.2544638	0.2584320	0.2626483	0.2666165	0.2705848
0.2745530	0.2787693	0.2827375	0.2867057	0.2906740
0.2948902	0.2988585	0.3028267	0.3067949	0.3110112
0.3149794	0.3189477	0.3229159	0.3271322	0.3311004
0.3350686	0.3390369	0.3432531	0.3472214	0.3511896
0.3551579	0.3593741	0.3633423	0.3673106	0.3712788
0.3752471	0.3794633	0.3834316	0.3873998	0.3913680
0.3955843	0.3995525	0.4035208	0.4074890	0.4114572
0.4156735	0.4196417	0.4236100	0.4275782	0.4315464
0.4357627	0.4397309	0.4436992	0.4476674	0.4518837
0.4558519	0.4598202	0.4637884	0.4677566	0.4719729
0.4759411	0.4799094	0.4838776	0.4878458	0.4920621
0.4960303	0.5000000			

****

TABLE 2.0-1

ORIGINAL PAGE IS
OF POOR QUALITY

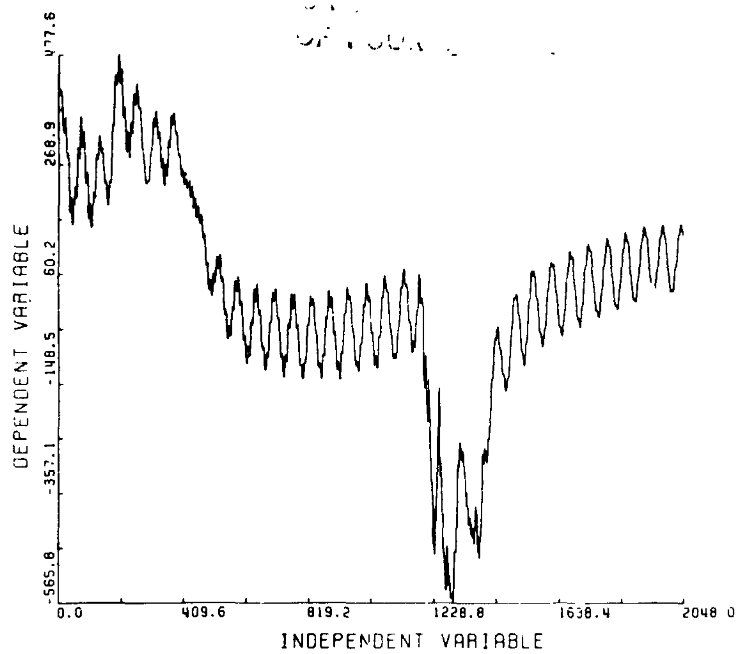


Fig. 2.0-1

On the Dynamics Explorer Spacecraft there is a magnetic field experiment that provides a vector measurement of the ambient magnetic field every 0.5 sec. The data from one of the components of the vector is plotted as a function of time.

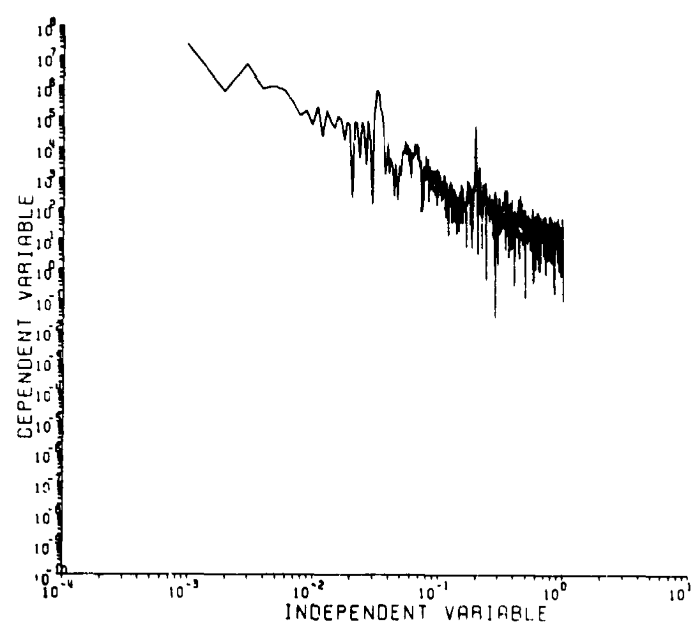


Fig. 2.0-2

Shows the spectrum resulting from application of operators WINDOW, FFT, SPECT, and GRAPH on the time series from Fig. 2.0-1, note that at 0.0324 and 0.175 Hz there are peaks in the spectrum resulting from the unwanted signals.

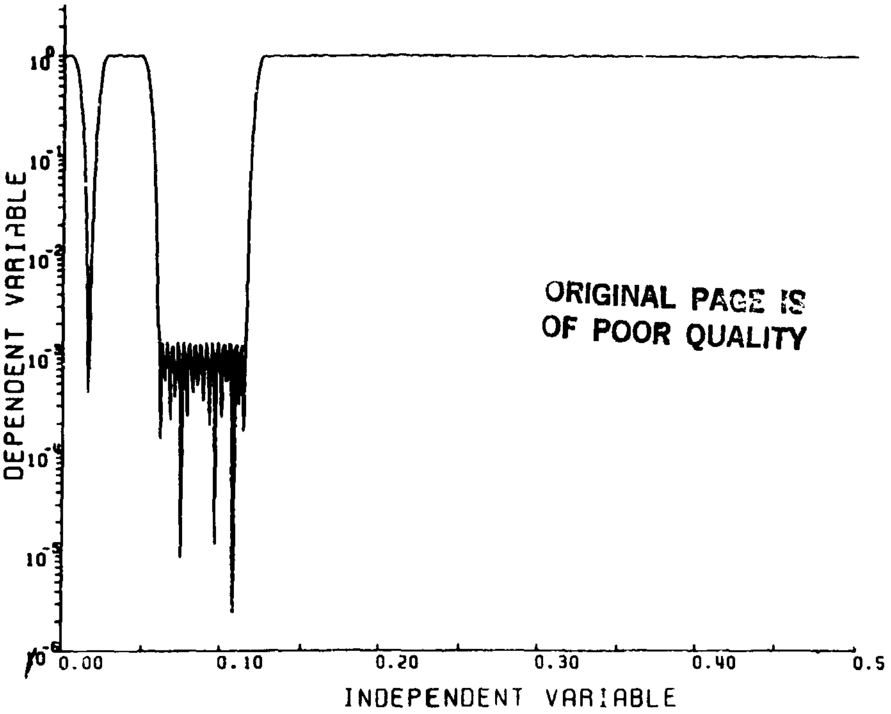


Fig. 2.0-3

The transfer function of the 5 band filter is shown. This transfer function is obtained by using operator FFT on data set WINDOW which contains the filter coefficients.

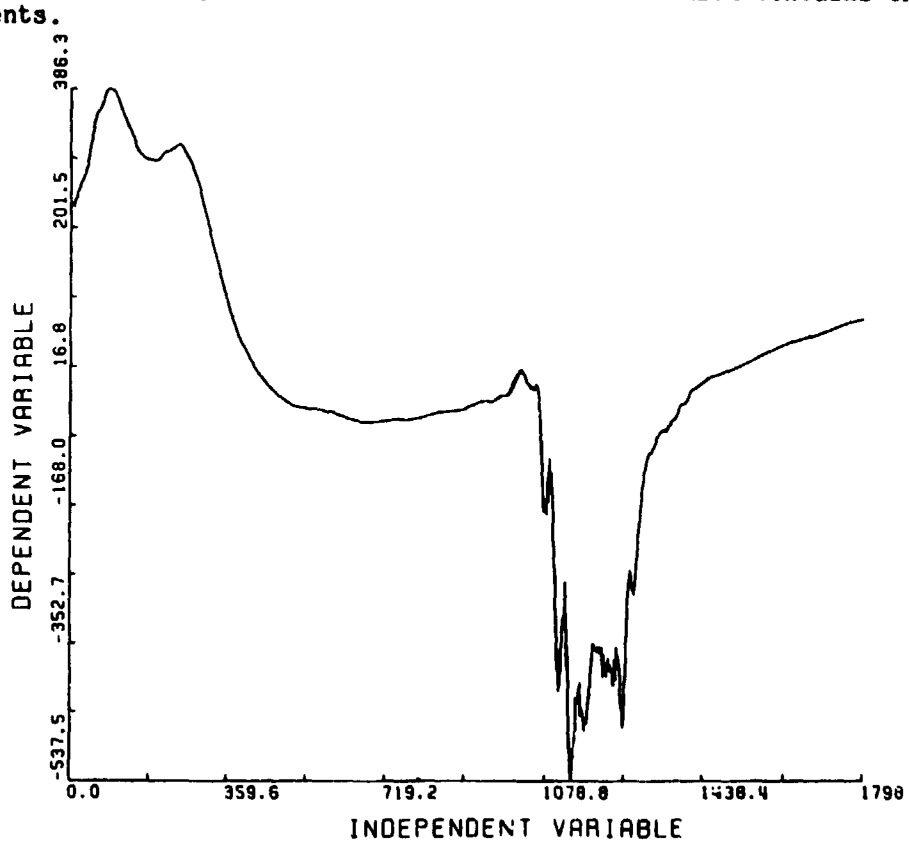


Fig. 2.0-4

Shows the results of using operator FILTER to filter the original time series shown in Fig. 2.0-1 with the 5 band filter shown in Fig. 2.0-3.

3.0) OVERVIEW OF THE IDSP SYSTEM DESIGN

1. The system must be interactive, yet capable of running in background mode. Basic functional operations are called operators. Each operator modifying a dataset writes results to a dataset identified with the operation performed. Other operators can then input those results by accessing the dataset associated with a previous operation. For routine operations, a command procedure can be written to batch process the data, using the operators in any meaningful order.

2. The user chooses the operations appropriate for the analysis of his particular data, and is able to specify which dataset will be the input for a given operator. If at any point in interactive mode the user feels a result is unsatisfactory, he can redo any operation using a dataset from an earlier operation as input. Or he may operate recursively by specifying the output from a given operator as input for that same operator. This structure allows the user to "interactively experiment" with the analysis of his data.

3. Each dataset includes records detailing the operations that have been performed on that dataset. This is included to help the user keep track of data set processing so that he does not misorder or duplicate operations (for example, inadvertantly filtering a dataset twice).

4. The system is able to process up to n (currently n=8) different time series concurrently with up to NPTS points each, as long as the total space required by the operation is less than a real array of length 500,000. NOTE: Because of the size of working arrays required, operations involving the FFT will be considerably smaller. For series involving more than 10,000 points, the user should check documentation under Design Considerations, APPENDIX C and also be aware that time of such long runs may be excessive.

5. Any given operator imposes the same operation on all of the series currently being processed. Each operator assumes that every series being processed involves the same time interval (see Fig. 1.0-2).

6. The system processes the time domain according to two basic parameters: span, interval. The interval is defined as the basic time segment to be analyzed--the segment of data which is to be filtered or Fourier transformed by a specified operation. The span is defined to be the total time under analysis--composed of one or more contiguous intervals (see Fig. 1.0-3).

7. At the outset (in operator SETUP), each series is required to have valid data at the endpoints of the desired interval. If necessary, the system successively reduces the interval until this requirement is met.

8. Comparisons between series of different types where different operators are used can be accomplished by making separate runs, storing the results, and using this as input for a final run. A cross correlation operator (not currently available as a system operator) could be used to analytically compare the two results. For meaningful comparison, each series must have the same number of points. A concatenation operator will be used when one desires comparisons between two different runs. This concatenation will be permitted only if the number of points in the two sets of series is identical. This composite dataset can then be accessed as input for comparison operators.

9. The user can write his own operators and interface them into the IDSP system by adhering to the conventions described in Writing User Defined Operators, APPENDIX B.

10. Each application accesses data by an input routine specific to the experimenter data of the application. This routine is called by the system, and is the interface between the system and the experimenter data. Because a different input routine is required if the application changes the user will need to relink IDSP with this different input routine in this event.

4.0) HOW TO START USING THE IDSP

The beginning user should do the following:

1. Create a subdirectory called [usrid.IDSP] where the user will place the input parameter files. Use the following sequence:

```
SET DEF [usrid]
CRE/DIR [usrid.IDSP]
SET DEF [usrid.IDSP]
```

2. Create datasets in [usrid.IDSP] which are required by the SETUP command: FOR051.DAT (see operator SETUP, Section 6); FOR052.DAT (see Writing Input Routines, APPENDIX A), and any other data sets required by the input routine.

3. ASSIGN experimenter data to FOR011 or other FORTRAN logical unit number as required by the specific input routine being linked (this step may be optional depending on the specific requirements of the INPUT routine being used). See Writing Input Routines, APPENDIX A.

4. Activate the IDSP system by: @SYS$IDSP:IDSP linknames, where "linknames" are the names of the dataset(s) containing the object versions of special routines the user wishes to link into the system, e.g. user written operators. The name of the dataset containing the desired applications interface INPUT routine is always required; datasets containing user defined operators are optional (see Writing User Defined Operators, APPENDIX B). Multiple datasets must be separated by commas. The user should indicate the device name where appropriate to avoid a search of the wrong device and subsequent abort of the link. If the user has already linked his desired routines in a previous run, he may skip the link step by specifying "NOLINK" for "linknames". WARNING: In order for the system to be properly initialized this execution step must be entered after every time the user logs on. (Otherwise lack of proper system assignments will cause the program to abort.)

Assuming the default device is DRA1:, IDSP will then do the appropriate link, make logical assignments, and set the default directory to DRA1:[usrid.IDSP]. When it is ready for the user to begin using the system, it will display: ENTER COMMAND. Files created by IDSP will be stored into this default directory.

5. Help in understanding the commands available may be obtained by typing 'HELP', or by typing 'HELP command', e. g., 'HELP SETUP'. The first non-HELP command of any session must be either SETUP (place experimenter data into proper configuration for analysis, see Section 6) or FILDES (design a filter and store coeficients, see Section 6) unless datasets have been saved from a previous session. The session is terminated by the command STOP which executes termination procedures.

ALL COMMANDS MUST BE TYPED IN UPPERCASE.

Commands have the following form: CMD REQ OPT where CMD is the command name, REQ are the required parameters (separated by spaces), and OPT are the optional parameters (separated by spaces). For any given command, the user will be prompted if required parameters are omitted. Unless optional parameters are entered on the first line, their default values will be used.

6. An example of a reasonable sequence of commands follows:

SETUP (obtain interval for analysis)
INTERP SETUP (interpolate missing points)
GRAPH INTERP (Versatec plot of data)
FFT INTERP (do Fourier transform)
SPECT FFT (form spectral matrices)
HELP SPECT (find out diagonal is stored in SPECTD)
SHOW SPECTD (see operation history of diagonal)
HELP GRAPH (recall graphing options)
GRAPH SPECTD (see graph of spectral terms)
STOP (end the IDSP session).

5.0) APPLICATION NOTES

5.1) FOURIER TRANSFORMS, SPECTRAL DENSITY MATRIX, THE EIGENVECTOR SYSTEM AND MAXMIUM ENTROPY METHOD (OPERATORS FFT, FFTIN, SPECT, EIG, MEM)

5.1.1) OPERATOR FFT

We start by considering three related concepts of Fourier analysis, the Fourier Series, the Continuous Fourier Transform (Integral) and the Discrete Fourier Transform.

It can be shown that under rather liberal conditions (see Lanczos, 1956) an entirely "unpredictable" function f(t), generally normalized to the range $\pm\pi$, can be represented, (to any arbitrarily high degree of accuracy), by a sum of components which are periodic (sines and cosines). This sum or series is called a Fourier Series. The representation of f(t) as a Fourier series demands strict periodicity in the time domain. That is, when f(t) is represented by a Fourier Series it is assumed to repeat with period 2π outside of the fundamental range $\pm\pi$. The function f(t) may be a truly periodic function, or may exist in the finite interval $\pm\pi$ only, and we force periodicity on it in order to make the Fourier Series applicable for its representation. In the latter case periodicity is employed as a mathematical artifice.

Fourier found that decomposition of arbitrary functions into harmonic components remains possible even if the realm of the function f(t) extends beyond $\pm\pi$ to $\pm\infty$ (see Lanczos, 1966). This is called the Continuous Fourier Transform (see Eq. 5.1-1) which is neither periodic in the time or frequency domain due to the infinite limits. This new function, X(f), does not resemble the original function in any direct way but is merely associated with it somewhat as the logarithm of a number is associated with the original number. For the purpose of time series analysis this process maps the function from the time domain to the frequency domain.

If the data we have to work with are digitized observations taken at equidistant time intervals, ΔT, we employ the methods of Fourier Transforms but adapted to finite summation (Discrete Fourier Transform) instead of integration (Continuous Fourier Transform). The Discrete Fourier Transform is a special case of the Continuous Fourier Transform where it is assumed that the N samples of the original function f(t) are one period of a periodic waveform.

The Discrete Fourier transform, which is implemented as operator FFT, is of interest because it, under certain conditions to be discussed, approximates the continuous transform and thus allows us to Fourier transform discrete (digitized) data. The Discrete Fourier Transform implies periodicity in the time domain which results in periodicity in the frequency domain.

The relationship between the discrete and continuous Fourier transform is as follows:

The Continuous Fourier transform is defined as:

$$X(f)=1/(2\pi)\int_{-\infty}^{+\infty} x(t)e^{-i2\pi ft}dt \quad -\infty<f,t< +\infty \quad ; i=(-1)^{0.5} \qquad (5.1\text{-}1)$$

The equivalent Discrete Fourier transform is defined as:

$$X(j)= (1/N)\sum_{k=0}^{N-1} x(k)e^{-i2\pi jk/N} \quad j,k=0,1,2....,N-1 \qquad (5.1\text{-}2)$$

where N = the total number of data points in the original time series.

The validity of this relationship is a function of the particular waveform being analyzed.

According to Brigham, 1974, there are five cases of time domain functions to consider:

Case 1 Band-limited periodic waveform, truncation interval (rectangular window) equal to one (or multiple) period(s).

This example represents the only class of waveforms for which the discrete and continuous Fourier transform are exactly the same within a scaling constant.

Equivalence of the two tranforms requires:

(i) the time series of interest, x(t), must be periodic over T (see Fig. 5.1-1);

(ii) x(t) must be band-limited;

(iii) The sampling rate, f_f, must be at least two times the largest frequency component of x(t); It is defined as $f_f=1/(2\Delta T)$, where ΔT is a constant and is the time between samples; and

(iv) The truncation (rectangular window) must be non-zero over exactly one (or multiple) period(s) of x(t). This also implies that the time series x(t) should be a stationary series. This means its statistical properties should not change with time during the period of time spanned by the series (see Jenkins, 1968).

Case 2 Band-limited periodic waveforms, truncation interval not equal to integer period(s).

If a periodic, band-limited function is sampled and truncated to consist of other than an integer multiple of the period, the discrete and continuous fourier transforms will differ. The effect of truncation at other than a multiple of the period is to create a periodic function with sharp discontinuities (see Fig. 5.3-7 under the discussion of zero padding). These sharp discontinuities in the time domain result in additional frequency components in the frequency domain. This effect is termed leakage. Windowing with other than a rectangular window (see Section 5.3) can be employed to reduce this leakage.

Case 3 Another class are functions which are of finite duration in the time domain. If x(t) is time-limited (for an example see Matthaeus and Goldstein, 1982), its Fourier transform cannot be band-limited and sampling must result in aliasing. It is necessary to choose the sample interval ΔT such that aliasing is reduced to an acceptable range. For this class of functions, if N is chosen equal to the number of samples of the finite-length function, then the only error is that introduced by aliasing. Errors introduced by aliasing are reduced by choosing the sample interval, ΔT, sufficiently small and, in the limit (see Lanczos, 1956), the discrete Fourier transform will agree to within a constant with the continuous Fourier transform.

Case 4 General periodic waveforms, truncation interval one (multiple) period(s).

Periodic waveforms not band-limited but truncated to an interval of one (multiple) period(s) will result in the discrete and continuous Fourier transforms being the same with the only source of error being aliasing. If the truncation is not equal to an integer multiple of the period, then results are as described in Case 2.

Case 5 General waveforms, not time-limited or band-limited.

This important and common class of functions are neither time or band-limited. Sampling thus results in an aliased frequency function and time domain truncation introduces rippling in the frequency domain. As this class of functions is often encountered we would like to treat them as though they were either band-limited (Case 1) or time-limited (Case 3). The aliasing error can often be reduced to an acceptable level by decreasing ΔT, and the time domain truncation error can often be reduced by windowing with other than a rectangular window (see Section 5.3).

Graphical interpretation of operator FFT

For the following discussion see Eq. 5.1-2 and Fig. 5.1-1 which follows Bergland, 1969. Let (ΔT) be the time between samples in the time domain, so that the fundamental frequency is, $f_0 = 1/T$, and $f_s = Nf_0$. X(j), the discrete Fourier transform, is in general a complex series. The time series $x(k\Delta T)$ is assumed to be periodic in the time domain of period T. The Fourier coefficients $X(jf_0)$ are periodic over f_s by definition of Eq. 5.1-2. Each j should be interpreted as a harmonic number and each k a sample period number. Note that $x(N\Delta T)=x(0)$, $X(Nf_0)=X(0)$, actual frequency = jf_0, and actual time = $k\Delta T$.

When the series x(k) is composed of real numbers, as it often is, the real part of X(j) is symmetric (even function) about the Nyquist frequency and the imaginary part is antisymmetric (odd function) about the Nyquist frequency. Fig. 5.1-1 shows the relationship between $x(k\Delta T)$ and $X(jf_0)$. This also can be seen in the example shown in Figs. 5.1-2 and 5.1-3, where we show the real and imaginary parts of the results of discrete Fourier transforming $x(t) = 10 \text{ Cos } (2\pi * 15 * k * \Delta T) + 5 \text{ Sin } (2\pi * 20 * k * \Delta T)$ sampled at 120 times per sec. Note that the real part shown in Fig. 5.1-2 is symmetric about f_f (i.e., BIN 120/2 = 60) and the imaginary part shown in Fig. 5.1-3 is antisymmetric about f_f. Fig. 5.1-4 is the results of plotting the magnitude of this complex transform, which shows the expected peaks at BINS 15 and 20, respectively. Note: for convenience the range is thought of as $1 - (N/2)f_0$ to $(N/2)f_0$.

Operator FFT computes the $X(jf_0)$ complex coefficients; only (N/2)+1 coefficients are retained in the output data set (FFT) because of symmetry, where N is the number of points in the original time series (N=NPTS). Operator FFT also has the option of computing the magnitude and phase of the coefficients which are stored in data set FFTMP, if this option is executed.

5.1.2) OPERATOR SPECT

Often one wants to look at power spectra of a vector quantity $B_i(i=1,2,3)$. for this it is convenient to look at the matrix $G = G(jf_0) \equiv \langle B_i^*(jf_0), B_j(jf_0)\rangle$, which is an estimate of the Fourier transform of the 2-time correlation matrix $\langle B_i(t), B_j(t+\tau)\rangle$. For any vector quantity, the matrix G can be written in terms of the Power Spectral Densities (PSD), cospectra and quadrature spectra associated with the three vector component directions (x,y,z) as follows (see Otnes and Enochson, 1972):

Note: B_i^* = Complex Conj.

$$G = \begin{bmatrix} (\ S_x\ ) & (C_{xy}-iQ_{xy}) & (C_{xz}-iQ_{xz}) \\ (C_{yx}-iQ_{yx}) & (\ S_y\ ) & (C_{yz}-iQ_{yz}) \\ (C_{zx}-iQ_{zx}) & (C_{zy}-iQ_{zy}) & (\ S_z\ ) \end{bmatrix} \quad (5.1\text{-}3)$$

Where S is the power density and C is the cospectrum and Q is the quadrature spectrum. The sample cospectrum measures the covariance between in-phase components (i.e., between cosine components and sine components separately), and the sample quadrature spectrum measures the covariance between the out-of-phase components (i.e., between sine and cosine components, see Jenkins and Watts, 1968).

Now $C_{ij} = C_{ji}$ and $Q_{ij} = -Q_{ji}$ by definition, so we may write

$$G = \begin{bmatrix} (\ S_x\ ) & (C_{xy}-iQ_{xy}) & (C_{xz}-iQ_{xz}) \\ (C_{xy}+iQ_{xy}) & (\ S_y\ ) & (C_{yz}-iQ_{yz}) \\ (C_{xz}+iQ_{xz}) & (C_{yz}+iQ_{yz}) & (\ S_z\ ) \end{bmatrix} \quad (5.1\text{-}4)$$

or

$$G = \begin{bmatrix} (S_x) & (C_{xy}) & (C_{xz}) \\ (C_{xy}) & (S_y) & (C_{yz}) \\ (C_{xz}) & (C_{yz}) & (S_z) \end{bmatrix} - i \begin{bmatrix} 0 & Q_{xy} & Q_{xz} \\ -Q_{xy} & 0 & Q_{yz} \\ -Q_{xz} & -Q_{yz} & 0 \end{bmatrix} \quad (5.1\text{-}5)$$

Thus the PSD matrix is composed of a real, symmetric part and an imaginary, skew-symmetric part, and hence the matrix is hermitian.

We can also use the off-diagonal terms of the power density matrix to compute coherence and phase lag. The coherence function is defined as

$$\gamma_{ij} = (C^2_{ij} + Q^2_{ij})^{0.5}/(S_i * S_j)^{0.5} \quad (5.1\text{-}6)$$

and indicates whether the amplitude of the component at a particular frequency in one series is associated with a large or small amplitude at the same frequency in the other series.

and phase angle or phase lag in degrees is defined as

$$\phi_{ij} = (180/\pi)\tan^{-1}(Q_{ij}/C_{ij}) \qquad (5.1\text{-}7)$$

and indicates whether the frequency components in one series lead or lag the components at the same frequency in the other series.

Operator SPECT computes the PSD matrix by using up to the first four (4) complex series in the file generated by Operator FFT and in turn generates all possible permutations of the power and cross spectra using equation 5.1-8 below; note that i is the index for the individual elements of each time series, and j is the series index.

$$P(i,j,k) = XNORM*X(i,j)^{*}*X(i,k) \qquad (5.1\text{-}8)$$

Note: $X(i,j)^{*}$ = Complex Conj.,

where XNORM = 2/((2*NPTS-2)**2), thus folding the negative power and adding it to the positive. For i=1, XNORM = 1/((2*NPTS-2)**2).

Example

For the case of three (3) time series, each containing 11 points (NPTS=11), these matrices are generated in the following fashion. Note that the time series, after being Fourier transformed (Operator FFT), consists of 6 complex values of X(j). Note that for a given value of the time series index i, Eq. 5.1-8 produces from the Fourier transformed i^{th} elements of the three parallel series all possible permutations of auto and cross power spectra.

$P(1,1,1) = XNORM*X(1,1)^{*}*X(1,1)$
$P(1,1,2) = XNORM*X(1,1)^{*}*X(1,2)$
$P(1,1,3) = XNORM*X(1,1)^{*}*X(1,3)$
$P(1,2,2) = XNORM*X(1,2)^{*}*X(1,2)$
$P(1,2,3) = XNORM*X(1,2)^{*}*X(1,3)$
$P(1,3,3) = XNORM*X(1,3)^{*}*X(1,3)$
P(2,1,1) = XNORM*X(2,1)**X(2,1)

$$P(5,3,3) = XNORM * X(5,3)^{*} * X(5,3)$$
$$P(6,1,1) = XNORM * X(6,1)^{*} * X(6,1)$$
$$P(6,1,2) = XNORM * X(6,1)^{*} * X(6,2)$$
$$P(6,1,3) = XNORM * X(6,1)^{*} * X(6,3)$$
$$P(6,2,2) = XNORM * X(6,2)^{*} * X(6,2)$$
$$P(6,2,3) = XNORM * X(6,2)^{*} * X(6,3)$$
$$P(6,3,3) = XNORM * X(6,3)^{*} * X(6,3)$$

For the above calculations Operator SPECT would generate the set of PSD matrices shown below. Note that these Matrices are hermitian so we only compute the upper right portion:

$$i=6 \quad \begin{bmatrix} P(6,1,1) & P(6,1,2) & P(6,1,3) \\ & P(6,2,2) & P(6,2,3) \\ & & P(6,3,3) \end{bmatrix}$$

$$i=5 \quad \begin{bmatrix} P(5,1,1) & P(5,1,2) & P(5,1,3) \\ & P(5,2,2) & P(5,2,3) \\ & & P(5,3,3) \end{bmatrix}$$

$$i=4 \quad \begin{bmatrix} P(4,1,1) & P(4,1,2) & P(4,1,3) \\ & P(4,2,2) & P(4,2,3) \\ & & P(4,3,3) \end{bmatrix}$$

$$i=3 \quad \begin{bmatrix} P(3,1,1) & P(3,1,2) & P(3,1,3) \\ & P(3,2,2) & P(3,2,3) \\ & & P(3,3,3) \end{bmatrix}$$

$$i=2 \quad \begin{bmatrix} P(2,1,1) & P(2,1,2) & P(2,1,3) \\ & P(2,2,2) & P(2,2,3) \\ & & P(2,3,3) \end{bmatrix}$$

$$i=1 \quad \begin{bmatrix} P(1,1,1) & P(1,1,2) & P(1,1,3) \\ & P(1,2,2) & P(1,2,3) \\ & & P(1,3,3) \end{bmatrix}$$

Operator SPECT partitions the PSD matrices into data set SPECTD containing the diagonal components which are real and provide information about the power at each frequency in each separate time series and data set SPECTOFF containing the off-diagonal elements which are complex and contain information about the phase and coherence between time series. In the above example P(i,1,1), P(i,2,2), P(i,3,3) will be contained in SPECTD while the off-diagonal terms, P(i,1,2), P(i,1,3), P(i,2,3), will be contained in SPECTOFF. Operator SPECT uses equations 5.1-6 and 5.1-7 to compute the coherence and phase lag. These values are stored in complex data set COHPH. The real part of COHPH contains the coherence and the imaginary part contains the phase lag. See Fig. 5.1-5 for a schematic of Operator SPECT's input and output data sets.

5.1.3) STATISTICAL STABILITY

Part of Operator SPECT is an option to allow PSD matrices to be averaged together to improve statistical stability of estimates being made.

X(j) is a measure of the power in a frequency range bounded, for the j^{th} estimate, by the frequencies $(j\pm1/2)f_f/M$ and centered at jf_f/M where M is the number of spectral estimates to be calculated.

Associated with the computation of PSD is the degree of freedom υ, which for moderate to large values is given by:

$$\upsilon = 2N/M \qquad (5.1\text{-}9)$$

where N, as before, is the number of data points (NPTS) in the given time series. The quantity υ is related to the confidence interval for the estimates (see Sentman, 1974). for $\upsilon > 4$, confidence limits can be computed with good accuracy using the confidence factor k_c, where

$$k_c = \exp(2.3b/10(\upsilon-1)^{0.5}) \qquad (5.1\text{-}10)$$

and for 98% confidence, b = 29; for 96%, b = 25; for 90%, b = 20; and for 80%, b = 16, Using k_c one can calculate the error bar limits from

$$\text{upper limit} = (\text{PSD estimate})(k_c)^{0.5}$$

$$\text{lower limit} = (\text{PSD estimate})/(k_c)^{0.5}$$

For example, if N = 1500 samples and M = 60 spectral estimates, giving

$$\nu = 2N/M = 50$$

then k_c = exp(0.0329b) and, for 90% confidence, $(k_c)^{0.5}$ = 1.390 and $1/(k_c)^{0.5}$ = 0.719. The PSD estimate would thus be multiplied by the factors 1.390 and 0.719 to obtain error bar upper and lower limits, respectively.

Note the relationship between data set SPECTD, when not averaged, and the real part of data set FFTMP (see Section 6,FFT Operator); SPECTD = 2*(FFTMP/(2*J-1))**2, where J = the number of points in the FFT including zero frequency.

5.1.4) OPERATOR EIG, EIGENVECTOR SYSTEM

Users of spectral analysis techniques are often interested in applying them to the study of wave phenomena. In such cases it is usually necessary to investigate the fluctuation characteristics of the individual components of a vector, not only relative to a physically-defined set of coordinates but also relative to one which is defined by the directional properties of the fluctuations themselves. For the case of an ideal elliptically-polarized plane wave, for example, there is no fluctuation perpendicular to the wave front, and a maximum level of fluctuation along one direction parallel to the wave front. There is thus a preferred coordinate system which most perfectly reveals the nature of such a wave.

Such an analysis is implemented using an eigenvalue and eigenvector calculation. In such a calculation, the characteristic variance ellipse (see Fig. 5.1-6) is determined, to the extent possible; the principal axes of the ellipse are the directions of maximum, intermediate and minimum fluctuation, respectively. To obtain eigenvalues and eigenvectors one needs only to diagonalize the real part of the PSD matrix, i.e., the real part of Eqn. (5.1-5) For this purpose, IDSP employs the subroutine EIGRS, which is an IMSL (International Mathematical and Statistical Libraries Inc., see Edition 8, Vol. 2, Chapter E) FORTRAN routine which computes and returns the eigenvalues and associated eigenvectors of a real symmetric matrix. The eigenvalues are the variances in the three principal axis directions (see Fig. 5.1-6). EIGRS does not order the eigenvalues and eigenvectors according to maximum, intermediate and mimimum, however; this is accomplished in subroutine DIAG. The eigenvectors V_j are labeled with the indices JMAX, JMID, and JMIN, respectively, in the ordering process. When used in conjuction with spectral analysis, this process is carried out separately and independently for each spectral estimate (each frequency bin). Thus the directions of interest can be different at each frequency and Fig. 5.1-6 is for a particular frequency, f.

For the computation of parameters which characterize the polarization properties of the fluctuations (see Fig. 5.1-7), we transform to a coordinate system defined by the eigenvector system. This can be done in one of two ways, either using the standard eigenvector definition in terms of the principal axis system or using a special definition appropriate for data in "mean field coordinates", which are defined in the description of operator MNFLD (see Section 6). These systems are further described below.

1) <u>Standard Eigenvector System</u>- Under this option (the default mode), the standard eigenvector definition developed above is used.

From this definition a matrix R can be derived for transformation of data from the input coordinate system to the eigenvector system:

$$R = \begin{bmatrix} V(1,JMAX) & V(2,JMAX) & V(3,JMAX) \\ V(1,JMID) & V(2,JMID) & V(3,JMID) \\ V(1,JMIN) & V(2,JMIN) & V(3,JMIN) \end{bmatrix} \quad (5.1\text{-}11)$$

Where V(i,JMIN), i = 1,2,3, is the unit vector in the minimum variance direction.

This matrix is then used to transform both the real and imaginary parts of the

PSD matrix data sets (SPECTD and SPECTOFF). Polarization parameters are computed according to the formulation described under option (2) below.

2) <u>Mean Field Eigenvector System</u>- Use of this option is valid only if the data analyzed <u>have been</u> transformed to the <u>Mean Field Coordinate System</u> by use of operator MNFLD (see Section 6) prior to application of EIG. This will have set the switch MFIELD = 1, which automatically activates the appropriate eigenvector derivation procedure.

In this case, the ordering of eigenvalues and eigenvectors in DIAG will lead to the definition

$$\begin{aligned} k_1 &= V(1,JMIN) \\ k_2 &= V(2,JMIN) \\ k_3 &= V(3,JMIN). \end{aligned} \qquad (5.1\text{-}12)$$

The unit vector $\hat{k}$ gives the wave normal vector direction in the case of a wave analysis, and is here taken as the direction of minimum fluctuation, i.e., the direction given by the eigenvector assiciated with the minimum eigenvalue, and is one of the set of eigenvector coordinates.

For data in mean field coordinates, the matrix R can be derived which transforms the real and imaginary parts of the PSD matrix data sets (SPECTD and SPECTOFF) to (mean field) eigenvector coordinates. In this case, R is expressed in terms of the components (k_1,k_2,k_3) of $\hat{k}$ only:

$$R = \begin{bmatrix} \dfrac{-k_1k_3}{\sqrt{k_1^2+k_2^2}} & \dfrac{-k_2k_3}{\sqrt{k_1^2+k_2^2}} & \sqrt{k_1^2+k_2^2} \\ \dfrac{k_2}{\sqrt{k_1^2+k_2^2}} & \dfrac{-k_1}{\sqrt{k_1^2+k_2^2}} & 0 \\ k_1 & k_2 & k_3 \end{bmatrix} \qquad (5.1\text{-}13)$$

This matrix is then used to transform both the real and imaginary parts of the PSD matrix data sets (SPECTD and SPECTOFF). To compute polarization parameters (assuming a plane wave), one uses only the components of the resultant spectral density matrix corresponding to the plane perpendicular to the wave vector direction. Thus the analysis reduces to that of a 2 x 2 matrix.

For the polarization analysis we have followed the formulations of Fowler et al., 1967 and Rankin and Kurtz 1970. The two-dimensional power spectral matrix is given by

$$J = \begin{bmatrix} J_{xx} & J_{xy} \\ J_{yx} & J_{yy} \end{bmatrix} = \begin{bmatrix} R_{xx} & (R_{xy} - iI_{xy}) \\ (R_{yx} - iI_{yx}) & R_{yy} \end{bmatrix} \tag{5.1-14}$$

where R and I signify real and imaginary parts, respectively. Now $|J| = R_{xx}\, R_{yy} - [(R_{yx} - iI_{yx})\,(R_{xy} - iI_{xy})]$ or, since

$$R_{yx} = R_{xy} \text{ and } I_{yx} = - I_{xy},$$

then

$$|J| = R_{xx}\, R_{yy} - R^2_{xy} - I^2_{xy}. \tag{5.1-15}$$

Computing the characteristic equation

$$D = \frac{1}{2}\,\{(R_{xx} + R_{yy}) - [(R_{xx} + R_{yy})^2 - 4|J|]^{1/2}\}$$

from the characteristic root of the matrix J (see the derivation in Fowler et al., 1967), we can then define a matrix that represents the polarized part of the signal only:

$$P = \begin{bmatrix} P_{xx} & P_{xy} \\ P_{yx} & P_{yy} \end{bmatrix} = \begin{bmatrix} (J_{xx} - D) & J_{xy} \\ J_{yx} & (J_{yy} - D) \end{bmatrix} \tag{5.1-16}$$

We then have that $P_{xx} = R_{xx} - D$ and $P_{yy} = R_{yy} - D$, and we can calculate the degree of polarization, DEGPOL, which is the ratio of polarized power to total power in fluctuations, according to

ORIGINAL PAGE IS OF POOR QUALITY

$$\mathrm{DEGPOL} = \frac{\mathrm{Tr}|P|}{\mathrm{Tr}|J|} = \frac{P_{xx} + P_{yy}}{J_{xx} + J_{yy}}, \tag{5.1-17}$$

where Tr = trace. Further, we can compute, as previously shown, the <u>coherency</u>:

$$\mathrm{COH} = \left(\frac{J_{xy}\ J_{yx}}{J_{xx}\ J_{yy}}\right)^{1/2}$$

$$= \left(\frac{R^2{}_{xy} + I^2{}_{xy}}{R_{xx}\ R_{yy}}\right)^{1/2} \tag{5.1-18}$$

The <u>angle of polarization</u> θ, defined as the angle between the X axis of the coordinate frame and the major axis of the polarization ellipse, is given by

$$\tan 2\theta = \frac{2\mathrm{Re}(J_{xy})}{J_{xx} - J_{yy}}.$$

$$= \frac{2R_{xy}}{J_{xx} - J_{yy}},$$

where Re = real part.

Thus

$$\mathrm{THETA} = \theta = \frac{1}{2}\tan^{-1}\left(\frac{2R_{xy}}{J_{xx} - J_{yy}}.\right) \tag{5.1-19}$$

For data in the eigenvector coordinate frame, THETA = 0.

The <u>ellipticity</u> ELIP of the polarization ellipse is the ratio of the minor axis to the major axis of the ellipse. The parameter is computed as follows:

$$\sin 2\beta = \frac{2\mathrm{Im}(J_{xy})}{[(J_{xx} + J_{yy})^2 - 4|J|]^{1/2}}$$

$$= \frac{-2I_{xy}}{[(R_{xx} + R_{yy})^2 - 4|J|]^{1/2}},$$

where Im = imaginary part. Thus

$$\beta = \frac{1}{2}\sin^{-1}\left(\frac{-2I_{xy}}{[(R_{xx} + R_{yy})^2 - 4|J|]^{1/2}}\right) \qquad (5.1\text{-}20)$$

and

$$\text{ELIP} = \tan|\beta|. \qquad (5.1\text{-}21)$$

The sign of β gives the sense of the polarization relative to the vector field being analyzed. This is shown, along with the corresponding phase angle value, in the Table 5.1-1.

TABLE 5.1-1

	Left-handed Polarization	Right-handed Polarization
Phase lag ϕ_{xy}	90°	270°
$\hat{k} \cdot \vec{B}$ positive	β negative	β positive
$\hat{k} \cdot \vec{B}$ negative	β positive	β negative

From this table we see that we can check results on the sense of polarization relative to the vector field at a given frequency for consistency between the computed phase angle at that frequency and the sign of β (dependent on the sign of $\hat{k} \cdot \vec{B}$). Since β is not one of the output parameters available in data set EIGPARM (see last paragraph of this Section), the sign of β has been assigned to DEGPOL, which is otherwise a positive definite quantity. Only the absolute value of β can be obtained from the quantity ELIP. The value of $\hat{k}\cdot\vec{B}$ = BDOTK is an additional output quantity which depends for its sign on the sign of $\hat{k}$. The sense of $\hat{k}$ (whether it is positive or negative) is completely arbitrary as determined by subroutine EIGRS. Thus it is necessary to adopt a convention for the sign of $\hat{k}$. This is done by means of the optional parameter KCON. It allows the user of operator EIG to either force $\hat{k}$ always to be outward rather than inward-directed relative to the sun or to force $\hat{k}$ always to have a component in the $+\vec{B}$ rather than the $-\vec{B}$ direction (unless $\hat{k} \perp \vec{B}$). A more detailed description of KCON is given in Section 6.9.

A second unit wave vector $\hat{k}$ is independently computed using the method described by Means, 1972. This technique derives the components of $\hat{k}$ using only the imaginary part of the spectral density matrix. Good agreement between the results from the two methods has been taken as an indicator that the wave normal vector was well determined. The analysis output also includes the signal-to-noise ratio defined by Means, 1972 as

$$SNR = \frac{J_{xx} + J_{yy}}{J_{zz}}, \qquad (5.1\text{-}22)$$

where $J_{xx} + J_{yy}$ is the trace of the 2-dimensional matrix defined by equation (5.1-14) and J_{zz} is the additional diagonal element of a 3 x 3 matrix that includes the $\hat{k}$ direction.

In studies of magnetic turbulence, a useful quantity (because it is an invariant of ideal MHD turbulence) is the magnetic helicity. The magnetic helicity is a measure of the topographical linkage of the magnetic field and is closely related to the polarization (see Moffatt, 1978). Techniques of determining the magnetic helicity can be found in Matthaeus and Goldstein, 1982 and Matthaeus et al., 1982. If we denote by $H_m(f)$ the reduced magnetic helicity spectrum, then a parameterzation of H_m bounded by ± 1 can be defined as

$$SIGMA = fH_m(f)/tr|G| \qquad (5.1\text{-}23)$$

where f is frequency and G and Q_{xy} are from Eq. 5.1-3 and $H_m(f) = 2Q_{yx}(f)/f$, where it is assumed that the spectrum has been reduced to the z axis (see Matthaeus and Goldstein, 1982 and Matthaeus et al., 1982 for more details). Note: Do not use the mean field option if you want to compute SIGMA- results will be meaningless.

Data set EIGPARM contains scaler quantities DEGPOL, COH, ELIP, THETA, TRTOT, SNR, BDOTK, and SIGMA. EIG also creates data set EIGVEC which contains in series 1-3, the first three components of the eigenvector corresponding to the direction of minimum variance for that mode. Series 4-6 are the eigenvalues (minimum, intermediate and maximum, respectively). Also created by operator EIG is data set EIGVEC1 which contains the three components of the eigenvector corresponding to the intermediate direction (series 1-3) and the three eigenvector components corresponding to the maximum direction of variation (series 4-6). For a schematic of operator EIG and it data sets see Fig. 5.1-8.

5.1.5) EXAMPLE: THE USE OF IDSP IN THE ANALYSIS OF WAVES IN DEEP SPACE DATA

In this section we provide an example of the application of the IDSP to a specific scientific research task. In this case use of IDSP assisted in analyzing magnetic field measurements made by the Voyager 2 spacecraft 11 days after its closest approach to Saturn on August 27, 1981. At that time, low frequency waves were detected in the magnetic field, apparently in association with enhanced flows of charged particles from the direction of Saturn. Fig. 5.1-9 shows the magnitude B of the total magnetic field as well as that of each of the three orthogonal vector components of the field in a heliographic coordinate system during a 24-hour period. The data plotted are 48-second averages of initially 60 millisecond measurements. Large amplitude waves with a period of approximately 1.6 hours on average can be seen both in the magnitude and each of the components of the field. As noted on Fig. 5.1-9, higher frequency fluctuations of the field were detected also during a six hour period beginning at midday. They will be investigated later in this example. The analysis of both types of waves was accomplished via the IDSP and will be illustrated.

The data shown in Fig. 5.1-9 were processed using operators SETUP, INTERP and GRAPH. Using operator WINDOW, the effect of different types of windows on the data was examined. In the examples shown, the rectangular (default) window option was used. The data were spectrum-analyzed using the FFT and SPECT operators, with 3 points forming each spectral estimate in the latter operation (6 degrees of freedom). Fig. 5.1-10 shows part of the output of SPECT in the form of the power spectral density matrix diagonal terms plotted using GRAPH. They constitute the spectra of the field vector components and the field magnitude (data set SPECTD). Shown along with the PSD of B for comparison is the trace of the PSD matrix (calculated using operator TRACE), which is a composite of the individual component spectra. The peaks of each of the spectra, corresponding to the quasi-sinusoidal waves in the time domain, occur in a band centered at a frequency of 1.75×10^{-4} Hz.

The properties of the waves were studied further by means of the EIG operator. A subset of the results are shown in Fig. 5.1-11. Selected parameters were plotted from data set EIGPARM using operator GRAPH. Their values are shown for a restricted region of the frequency domain near the peak frequency, where in this case, a linear frequency scale has been chosen for display purposes. For reference, the corresponding segment of the PSD trace is plotted in the top panel. The most relevant quantities resulting from the eigenfunction analysis are shown below the trace: (1) Degree of polarization (DEGPOL), (2) cosine of the angle between the wave normal vector $\hat{k}$ and the magnetic field vector $\hat{B}$ ($\cos\alpha = \hat{k} \cdot \hat{B}$ = BDOTK where α is known only modulo 180°-see discussion following Table 5.1-1), and (3) the ellipticity of the polarization ellipse (ELIP). The definitions of these quantities were given in Eqns. 5.1-17, 5.1-21 and the text following those equations.

A vertical band one spectral estimate wide delineates the relevant region on each parameter curve in Fig. 5.1-11. Note that DEGPOL has been plotted with both positive and negative values. This is because this otherwise positive-definite quantity is multiplied by the sign of β (Eqn. 5.1-20). The wave is seen to be highly polarized ($\mathrm{DEGPOL}_{\mathrm{peak}} = 0.95$). An average value of BDOTK over the peak gives $\alpha = 112°$; thus the wave is propagating at a large angle to the magnetic field direction. Positive values of DEGPOL at the wave frequency, together with negative values of BDOTK, suggests left hand (LH) polarization in the spacecraft frame of reference (see Table 5.1-1 and following text). However, as indicated following Table 5.1-1, in running EIG a convention forces $\hat{k}$ to have a particular sense of direction. The user must be careful that this direction is physically reasonable in terms of its implications for the specific phenomenon being studied. In this case, as would be usual for studies of waves in the solar wind, the analyst set KCON = 1 (away from the sun). This also forced $\hat{k}$ to have a component toward Saturn, the source of the beam of ions exciting the waves. But, since the waves should have the same sense of direction as that of ion flow in the beam, $\hat{k}$ should in fact be directed away from Saturn. Thus we are led to the intepretation that in this case the wave was actually right hand (RH) polarized (corresponding to α = 180° - 112° = 68°). Finally, an average value of ELIP = 0.7 is found at the peak, indicating that the wave is much closer to being circularly polarized (ELIP = 1.0) than linearly polarized (ELIP = 0).

It is sometimes useful to replot the original data in a coordinate system defined by the set of eigenvectors corresponding to the directions of maximum, intermediate and minimum oscillation in the wave. This is accomplished using the ROTATE operator on the original (in this case INTERP) data set, with the new coordinate system provided by the EIGVEC and EIGVEC1 data sets. The specification given in this case was: ROTATE INTERP 1,2,3 0 NFREQ=16. The resulting data are shown in Fig. 5.1-12, with X_E being in the direction of maximum variation and Z_E in minimum variation direction.

As mentioned earlier, in Fig. 5.1-9 and 5.1-12, an interval of higher frequency fluctuations is seen superimposed on the 1.6 hour period waves. The magnetic field data for the 6.4 hour interval in which this second wave mode ocurred are plotted on an expanded time scale in Fig. 5.1-13. In this case higher time-resolution data (9.6 sec averages) are shown. Approximately 4 cycles of the low frequency oscillation are still seen modulating the higher frequency fluctuations, which can be seen to be of much higher amplitude in the vector components than in the field magnitude. To facilitate the study of higher frequency variations, we removed the low frequency excursions by high pass filtering the data set shown in Fig. 5.1-13. The procedure for designing and applying digital filters is discussed and illustrated in Section 5.2. The filter specifications that were entered into data set FOR058.DAT and used by operator FILDES for this task are as follows:

```
125,1,2,0
0.0,0.002,0.02,0.5
0.0,1.0
182.0,1.0
```

This is a filter with 125 coefficients, of the multiple passband/stopband type, with one stopband and one passband (for which the start and stop normalized frequencies are (0.0, 0.002) and (0.02, 0.5), respectively). The grid density is set at 16 (input specification of 0 gives default value of

16). Ideally there is complete attenuation in the stopband and no attenuation in the passband (0.0, 1.0), and, finally, a relative weighting of 182 in the stopband to 1 in the passband is used.

Fig. 5.1-14 illustrates the frequency response of the filter generated for this task. Because the higher frequency oscillation to be studied was still at a relatively low value of normalized frequency ($\sim$ 0.03), the passband response was designed to rise steeply to a value of 1 near zero normalized frequency. This requirement tightly constrained the values that could be given to most of the filter design parameters. The bandedge array values (EDGE(i), i=1,4) and band weight functions (WTX(1), WTX(2)) could be adjusted, however, within the given constraints, to permit to some degree the maximization of attenuation in the stopband and minimization of ripple amplitude in the passband. Because of the fact that the two factors work against one another for a given filter configuration as the weights are varied, reduction of passband ripple beyond a given point is limited by the onset of incomplete stopband attenuation. Conversely, stopband attenuation is limited by increasing distortion of the spectrum by the increase in passband ripple amplitude. Any such distortion is undesirable and may not be totally eradicable if the number of filter coefficients cannot be increased further and/or the bandwidths cannot be further adjusted, including the width of the transition band (see Fig. 5.2-1).

A study was performed (using operator FILOPT, see Section 6) to provide insight into the degree of tradeoff possible in this example between stopband (band 1) attenuation and passband (band 2) ripple. While all other design parameters were held fixed, WTX(1) and WTX(2) were varied over values ranging from (1,100) to (1000,1), which, since the weights actually represent a ratio, is equivalent to setting WTX(2) = 1 and varying WTX(1) over the range 0.01 to 1000. As described in Section 5.2, for a given set of parameter values, FILDES generates a filter and lists its characteristics (as illustrated in Table. 5.2-1). Among the output characteristics tabulated are the amplitude deviations in each band, i.e., the delta_1 and delta_2 values. Although the resulting values of each have been influenced by the full suite of input parameter values, and hence desired values may not be realized, the ratio is predetermined by the bandweights, i.e., by definition $\text{delta}_1/\text{delta}_2$ =

WTX(1)/WTX(2) (see Rabiner and Gold, 1975, pp 197,199).

From the amplitude deviations, the passband and stopband ripple amplitudes in dB are computed (see Section 5.2 for definitions).

In our tradeoff study we plotted the various passband and stopband ripple combinations output by FILDES as functions of the relative weight WTX(1)/WTX(2). We used the dB values since they are directly comparable to the desired performance specifications that enter as inputs into the filter planning process (see Section 5.2). The resulting curves are shown in Fig. 5.1-15. As can be seen, passband (PB) ripple is relatively flat over much of the region of steepest increase (negative) in stopband (SB) ripple. For larger relative weights (>160), however, PB ripple rises steeply, and the continued SB ripple growth becomes less steep.

For our high pass filter, the stopband minimum frequency response (and hence the level of attenuation) is that computed at zero frequency (see Fig. 5.1-14). This value is also a function of relative bandweight, and, although influenced by the stopband ripple amplitude, its variation with changes in relative weight is different from that of the ripple amplitude. The minimum response (maximum attenuation) dependence on weight for this case is illustrated in Fig. 5.1-16. The curves shown in Fig. 5.1-15 and 16 can be obtained by using INTERP on the output from the FILOPT operator and graphing the results.

Although there are two distinct minima in the curve, the second, occurring at a relative weight of 182, was found to provide the most complete attenuation of the low frequencies in the data. From Fig. 5.1-15 it is seen that for a weight of 182 the passband ripple has not yet grown large (0.13), but the stopband ripple is relatively large (-82), as desired. This was thus considered to be the optimum state for the given bandedge configuration. It is worth noting that prior to selection of bandedges (0.0,0.002,0.02,0.5), we investigated the configuration (0.0,0.001,0.01,0.5). For that case the ripple vs. relative weight diagram was similar in general form to that of Fig. 5.1-15, but differed considerably in detail. The stee increase in PB ripple began at a low value of relative weight (4.5). The optimum values of SB and

PB ripple were -40 and 0.35, respectively, considerably less effective than those possible with the final bandedge configuration chosen. Thus it is also important to optimize bandedge values as well as band-weighting factors. It should be noted that a set of desired stopband and passband ripple amplitudes will end up being the achieved characteristics of a particular filter configuration only if the total set of design parameters can accomodate the realization of those values.

Upon completion of the tradeoff and optimization procedure for our high pass filter design, operator FILTER was called to apply the resultant filter to the Voyager data. The filtered data set is plotted in Fig. 5.1-17. It is clear that the filter has successfully removed the modulation. The remaining fluctuation is quite variable in amplitude, and the frequency is evidently variable as well. This may be the result of a superposition of several fluctuations of different amplitudes and frequencies. The corresponding spectral characteristics are shown in Fig. 5.1-18, where 5 points were used to form each estimate (10 degrees of freedom). The rising trend of the PSD at low frequencies < 10^{-3} Hz in each case is the result of the filtering of the data. The weakly dominant frequencies in the fluctations (3.4 to 3.9 X 10^{-3} Hz; period = 4.3 to 4.9 min) are denoted by the vertical dashed line; as can be seen, this peak is only a single feature of a broad, highly structured shoulder on the overall spectrum, as was anticipated from the appearance of the data

Selected results of the eigenfunction analysis are plotted in Fig. 5.1-19 together with a portion of the trace of the PSD matrix. As in the case of the 1.6 hour period waves, the $\sim$ 5 min fluctuations are highly polarized (Maximum DEGPOL = 0.8) and elliptical (Maximum ELIP = 0.75), polarization in the spacecraft reference frame is righthanded (RH), as indicated by +DP and +BDOTK, and the waves are propagating at a fairly large angle to $\vec{B}$ (average BDOTK = cos $\alpha \sim 0.57$ so $\alpha = 55^{\circ}$). Note that in this case KCON = 0 was specified in running EIG, forcing $\hat{K}$ to have a component in the $+\vec{B}$ direction.

5.1.6) OPERATOR MEM (MAXIMUM ENTROPY METHOD)

The maximum entropy method (Operator MEM) for spectral estimates was introduced by Burg in 1967. Its purpose was to increase the spectral resolution when the length of the observed data record was short. Conventional spectrum analysis techniques usually window the autocorrelation, append zeros and then Fourier Transform to obtain the spectral estimate. Instead the MEM extrapolates the autocorrelation beyond the limited range of the data to derive the spectral estimates at the desired resolution with little or no interference from other frequencies.

The MEM power spectrum, P(f), can be described by the following expression

$$P(f) = \frac{P_m \Delta t}{\left(1 - \sum_{n=1}^{m} a_{mn} e^{-i 2\pi f n \Delta t}\right)^2}$$

P_m = output power of the prediction error filter

a_{mn}'s = m+1 prediction error filter coefficients

Δt = data sampling rate

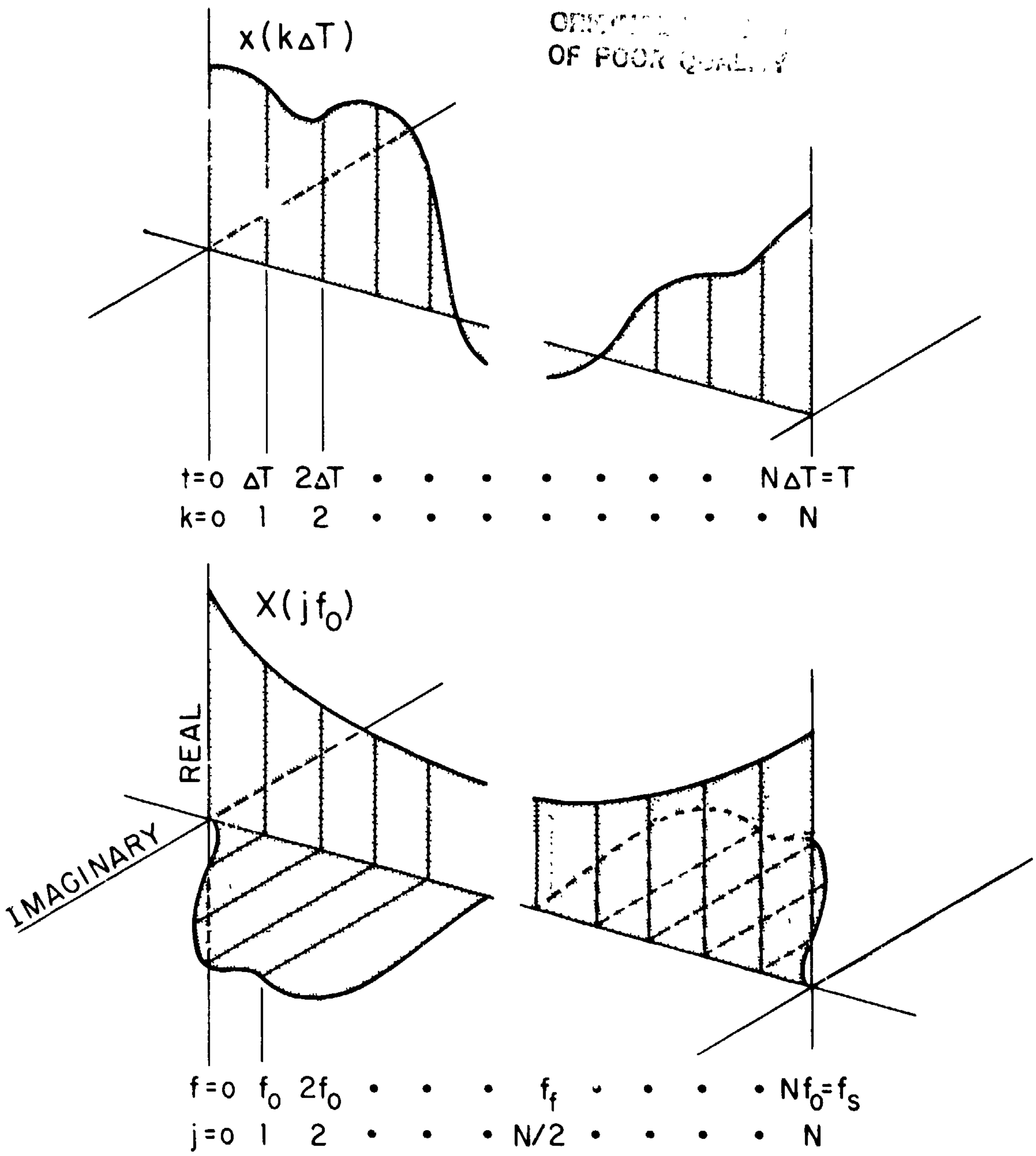
f = frequency

For details on how the filter coefficients are calculated see Anderson, 1974.

The equation is solved by stepwise iterative increase of the matrix dimension from m-1 to m. The only remaining area of concern is choosing the number of filter coefficients, m. If m is too small the power spectrum tends to be very smooth, thus eliminating any additional resolution; if m is too large, spurious information may be introduced into the spectrum. The value $2N/\ln(2N)$ can be used as a default parameter. This was proposed by Berryman, 1978 based on empirical evidence. Otherwise a trial and error approach is a tedious alternative.

REFERENCES:

1) Anderson, N., Shortnotes: On the calculation of filter coefficients for maximum entropy spectral analysis, Geophysics, Vol. 39, No. 1 (February, 1974), pp. 69-72.
2) Bergland, G. D., A Guided Tour of the Fast Fourier Transform, IEEE Spectrum, July 1969.
3) Berryman, J. G., Choice of operator length for maximum entropy spectral analysis, Geophysics, Vol. 43, No. 7 (December 1978), pp. 1384-1391.
4) Brigham, E. O., The Fast Fourier Transform, Prentice-Hall, Inc, Englewood Cliffs, New Jersey, 1974.
5) Burg, J. P., Maximum entropy spectral analysis, presented at the 37th Annual International SEG Meeting, Oklahoma City, OK, October 31, 1967.
6) Fowler, R. A., B. J. Kotick and R. D. Elliot, Polarization Analysis of Natural and Artificially Induced Geomagnetic Micropulsations, J. Geophys. Res., 72, 2871, 1967.
7) Jenkins, G. M. and D. G. Watts, Spectral Analysis and its Applications, Holden-Day, San Francisco, 1968.
8) Lanczos, C., Applied Analysis, Prentice-Hall, Englewood Cliffs, NJ, 1956.
9) Lanczos, C. Discourse on Fourier Series, Hafner Publishing Co, New York, 1966.
10) Matthaeus, W. H., M. L. Goldstein, Measurements of the rugged invariants of magnetohydrodynamic turbulence in the solar wind, J. Geophys. Res. 87, 6011, 1982.
11) Matthaeus, W. H., M. L. Goldstein, and C. Smith, Evaluation of magnetic helicity in homogeneous turbulence, Phys. Rev. Lett, 48, 1256, 1982.
12) Means, J. D., Use of Three-Dimensional Covariance Matrix in Analyzing the Polarization Properties of Plane Waves, J. Geophys. Res., 77, 5551, 1972.
13) Moffatt, H. K., Magnetic Field Generation in Electrically Conducting Fluids, Cambridge University Press, NY, 1978.
14) Otnes, R. K. and L. Enochson, Digital Time Series Analysis, J. Wiley and Sons, New York, 1972.
15) Rabiner, L. R., B. Gold, Theory and Application of Digital Signal Processing, Prentice-Hall Inc., Englewood Cliffs, NJ, 1975.
16) Rankin, D. and R. Kurtz, Statistical Study of Micropulsation Polarization, J. Geophys. Res., 75, 5444, 1970.
17) Sentman, D. D., Basic Elements of Power Spectral Analysis, U of Iowa Preprint 74-5, The University of Iowa, Iowa City, Iowa 52242, Jan. 1974.



$f_0 = 1/T$; $f_s = Nf$; $T = N\Delta T$

FIG. 5.1-1

This drawing is after Bergland, 1969 and shows what happens when a time series x(t) is Fourier transformed. X(j) is, in general, a complex series. The time series $x(k\Delta T)$ is assumed to be periodic in the time domain of period T. The Fourier coefficients $X(jf_0)$ are periodic over f_s. Each j should be interpreted as a harmonic number and each k a sample period number. Actual frequency = jf_0. Actual time = $k\Delta T$. When x(k) series is composed of real numbers, as it often is, the real part of X(j) is symmetric (even function) about the Nyquist frequency $f_f = f_s/2$ and the imaginary part is antisymmetric (odd function) about the Nyquist frequency.

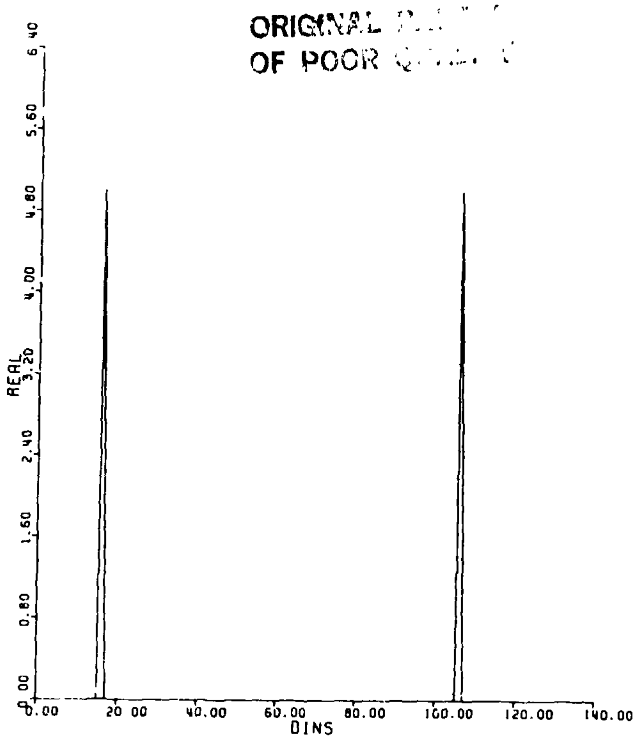


FIG. 5.1-2

Shows the real part of the results of Fourier transforming $x(t) = 10\ \text{Cos}\ (2\pi * 15 * k * \Delta T) + 5\ \text{Sin}\ (2\pi * 20 * k * \Delta T)$ sampled at 120 times per sec. Note that the real part is symmetric about the Nyquist frequency, f_r (i.e., BIN 120/2 = 60).

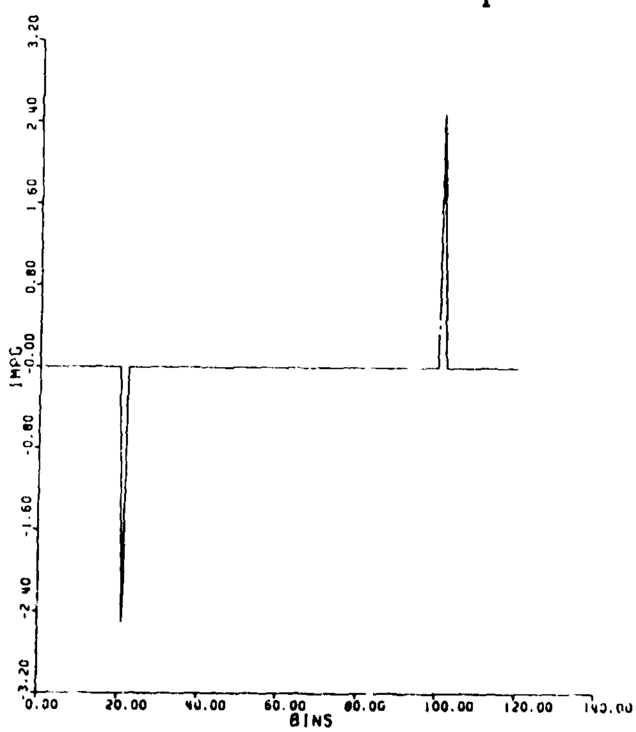


FIG. 5.1-3

Shows the imaginary part of the results of Fourier transforming $x(t) = 10\ \text{Cos}\ (2\pi * 15 * k * \Delta T) + 5\ \text{Sin}\ (2\pi * 20 * k * \Delta T)$ sampled at 120 times per sec. Note that the imaginary part is antisymmetric about the Nyquist frequency, f_r.

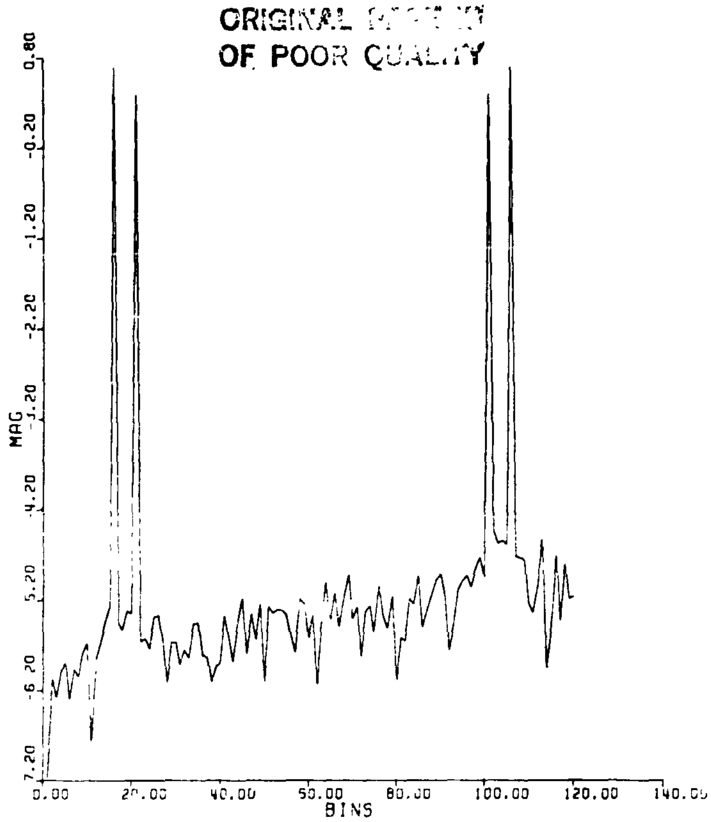


Fig. 5.1-4

Shows the results of plotting the magnitude of the Fourier transform of x(t) = 10 Cos (2π*15*k*ΔT) + 5 Sin (2π*20*k*ΔT) sampled at 120 times per sec, which shows the expected peaks at BINS 15 and 20, respectively and is symmetric about the Nyquist frequency.

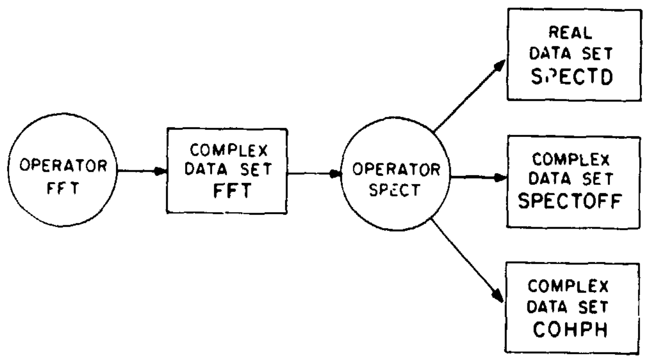


Fig. 5.1-5

A schematic of the operators FFT and SPECT and their associated data sets.

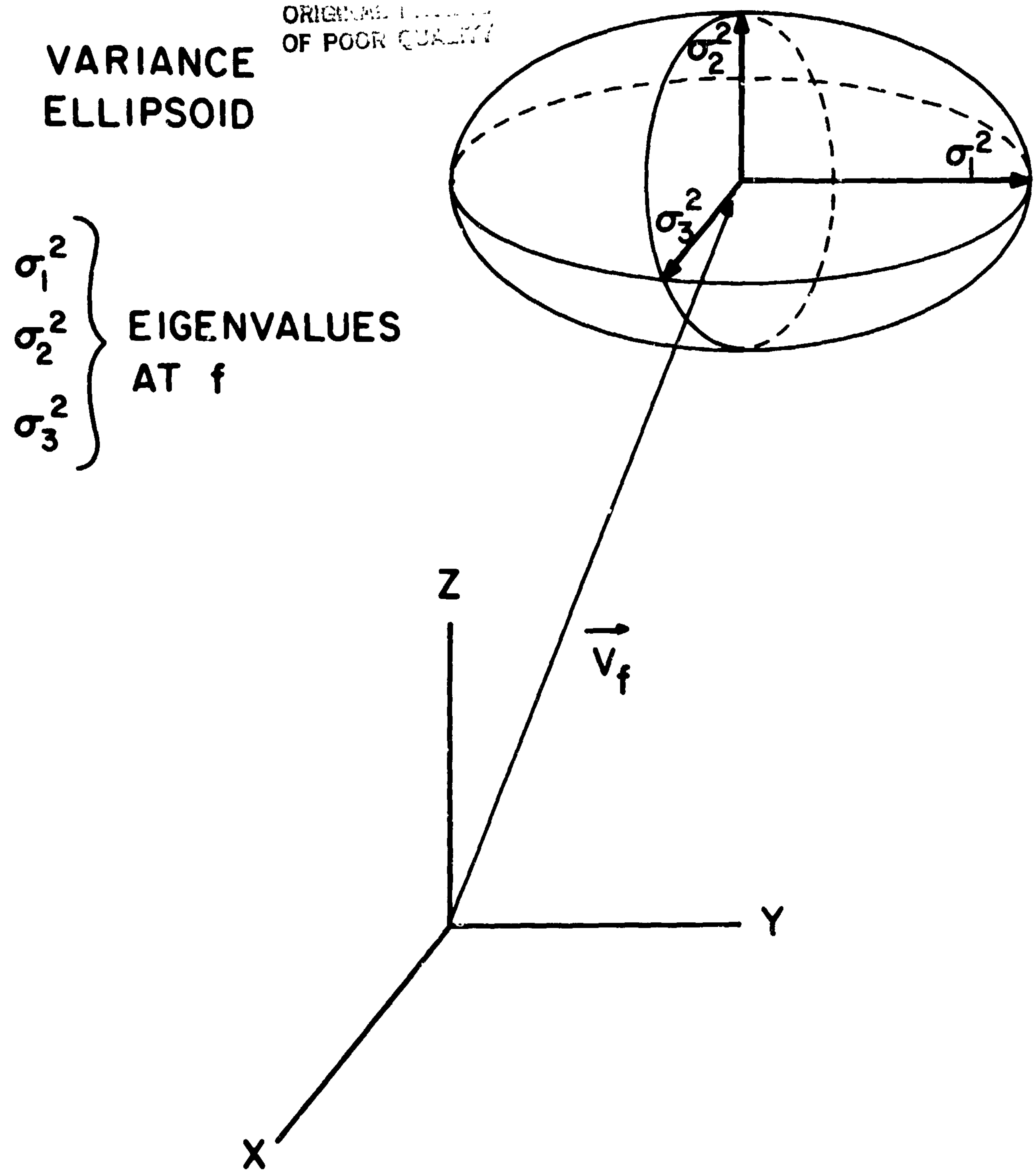
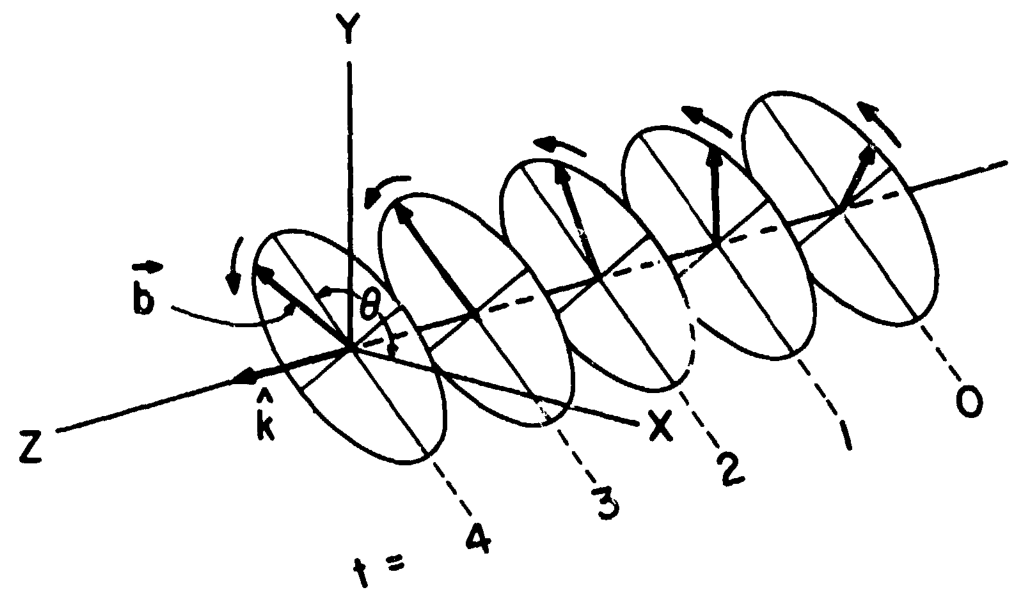


Fig. 5.1-6

In the study of wave phenomena it is common to investigate, using an eigenfunction analysis, the fluctuations characteristics of the individual components of the vector relative to the directional properties of the fluctuation themselves. In such calculation the eigenvalues determine the principal axes ($\sigma_1^2, \sigma_2^2, \sigma_3^2$) of the characteristic variance (polarization) ellipse at each frequency estimate. The eigenvectors define the coordinate system corresponding to the directions of maximum (X), intermediate (Y) and minimum (Z) oscillation in the wave at each frequency estimate.

ORIGINAL PAGE IS
OF POOR QUALITY



LEFT-HAND POLARIZATION

Fig. 5.1-7

Illustration of a plane, left-hand polarized wave, with front parallel to the X-Y plane, shown at a succession of times t = 0 to t = 4 as it propagates in the + Z direction (toward left). The perturbation vector $\vec{b}$ rotates CCW as viewed from upstream, its tip describing the polarization ellipse in each 360° rotation. Although the total field $\vec{B}$ is not shown, this case corresponds to $\hat{k} \cdot \vec{B}$ positive and β negative. For right-handed waves propagating in the + Z direction, the rotation sense would be CW. In the case shown, the spatial orientation of the ellipse remains constant during the time interval of the propagation; in general, it may vary with time.

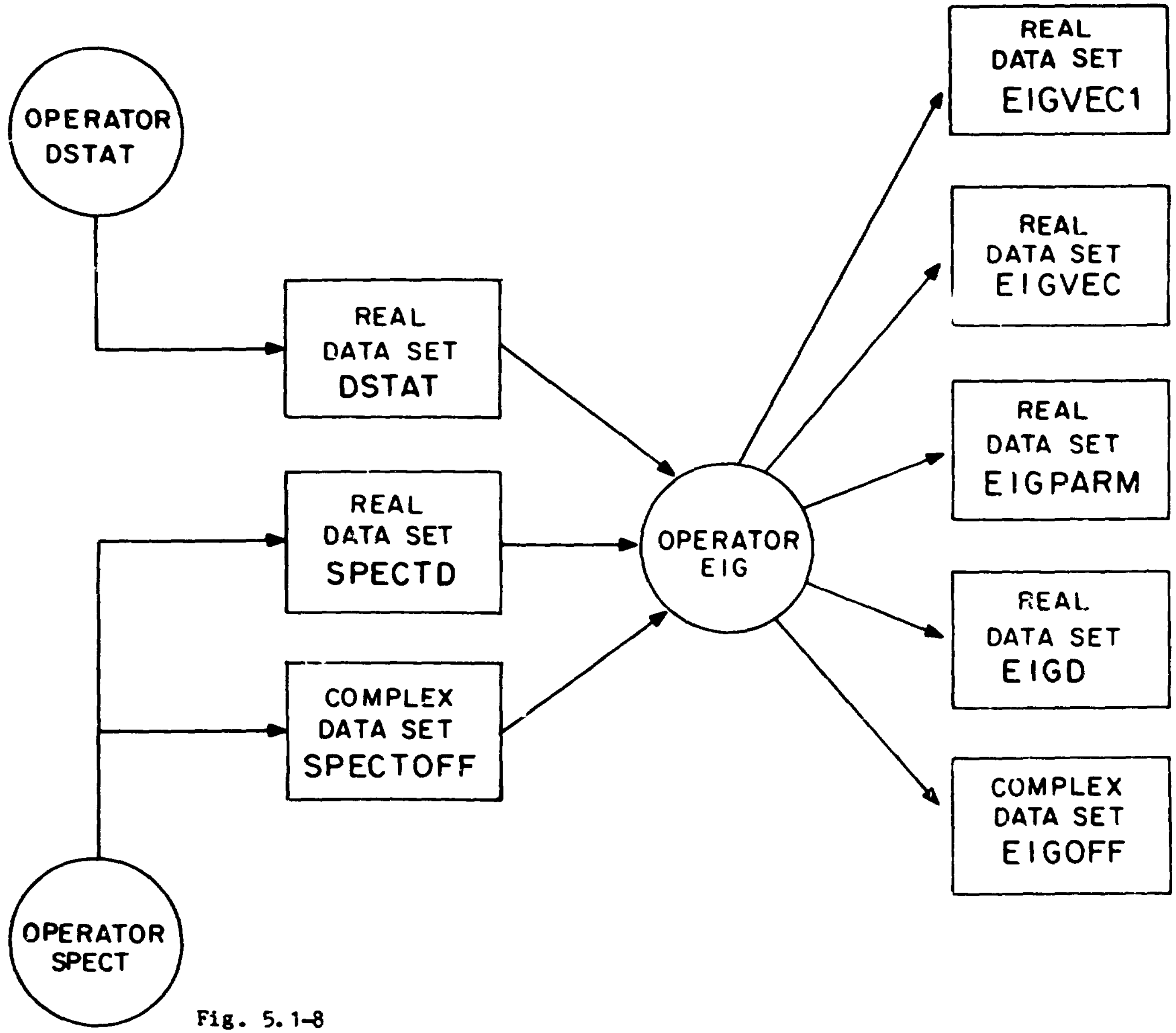


Fig. 5.1-8

A schematic of the operators EIG, SPECT, and DSTAT and the associated data sets.

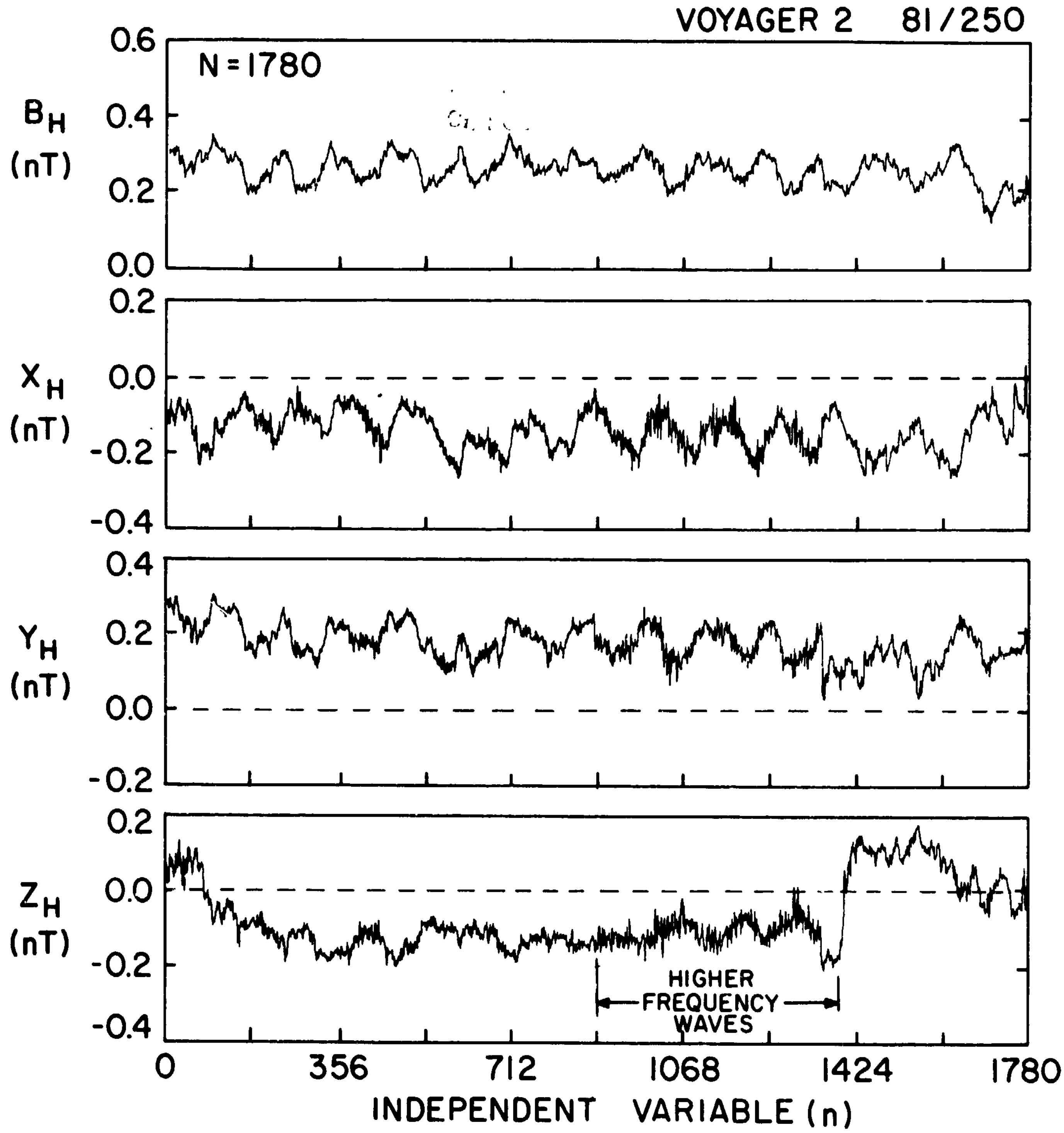


Fig. 5.1-9

Raw vector magnetic field time series from the Voyager 2 spacecraft consisting of 48-sec averages of field magnitude B_H, and vector components X_H, Y_H and Z_H in heliographic coordinates (ordinate units are nT = nanotestlas = 10**-5 Gauss; abscissa units used are sample number rather than time). The data cover a 24-hour period (1981/day 250) and show oscillation of field at two frequencies differing by a factor of 20. Note that at the lower frequency there are large amplitude variations in the field magnitude as well as in the components.

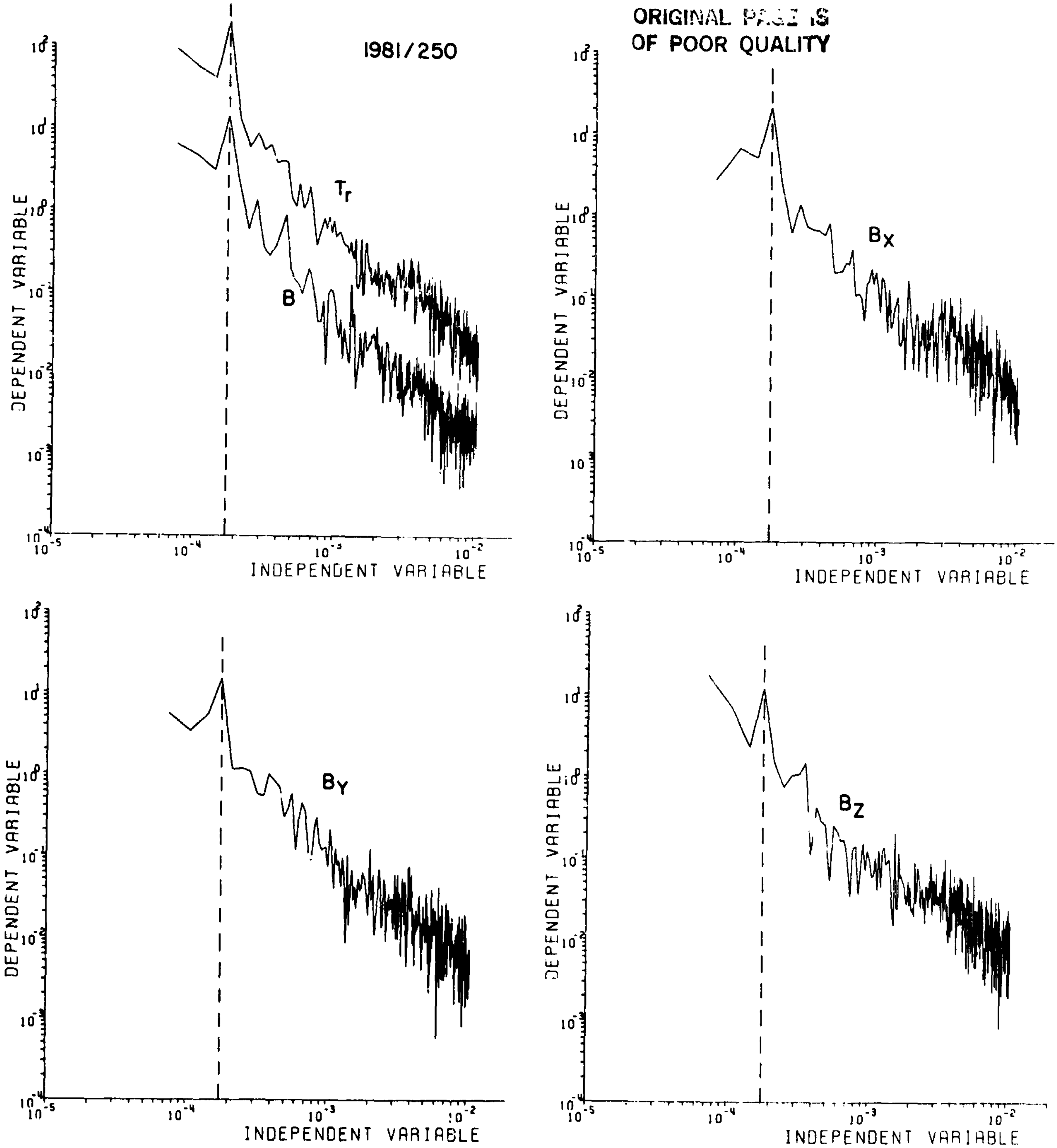


Fig. 5.1-10

Power spectra for each time series shown in Fig. 5.1-9, plus the trace (Tr) of the PSD matrix (upper left, with field magnitude B spectrum). Ordinate units are nT^2/Hz and abscissa units are Hz. In each of the spectra a pronounced peak is seen centered on 1.75 X 10**-4 Hz (dashed vertical line) corresponding to the low frequency, large amplitude variation prominent in the time series. This figure illustrates that there is relatively more power in directional fluctuations than in magnitude (field strength) fluctuations.

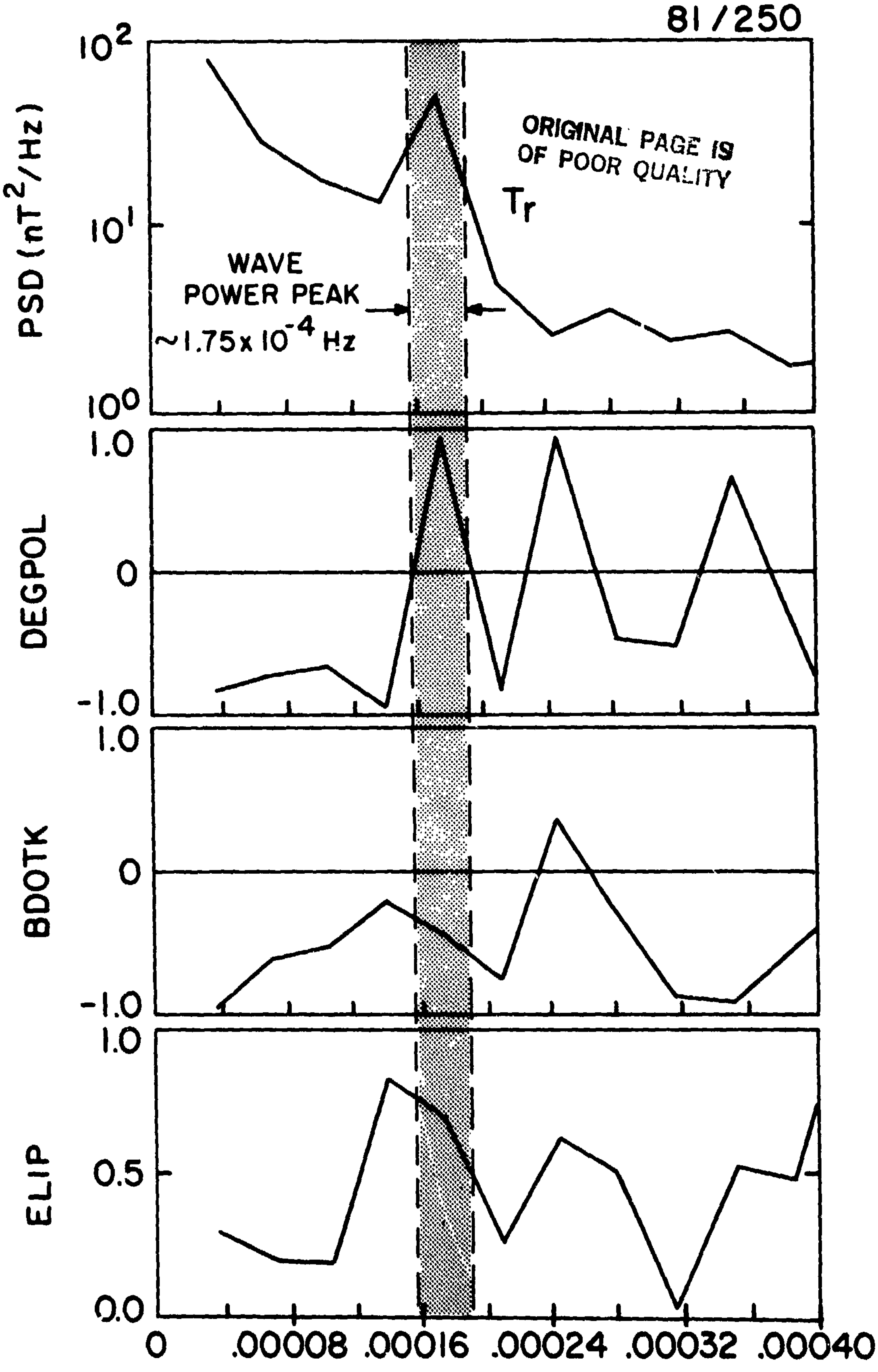


Fig. 5.1-11

Eigenfunction properties of fluctuations shown in Fig. 5.1-9 over a restricted range of frequency (4 X 10**-5 to 4 X 10**-4 Hz) that includes the peak wave frequency (delineated by the vertical hatched band). In the top panel the trace spectrum is repeated from Fig. 5.1-10 for reference. Also shown, in panels 2 through 4, respectively, are results from the application of EIG: degree of polarization, cosine of angle between $\hat{B}$ and $\hat{k}$, and wave ellipticity (see text for discussion).

ORIGINAL PAGE IS
OF POOR QUALITY

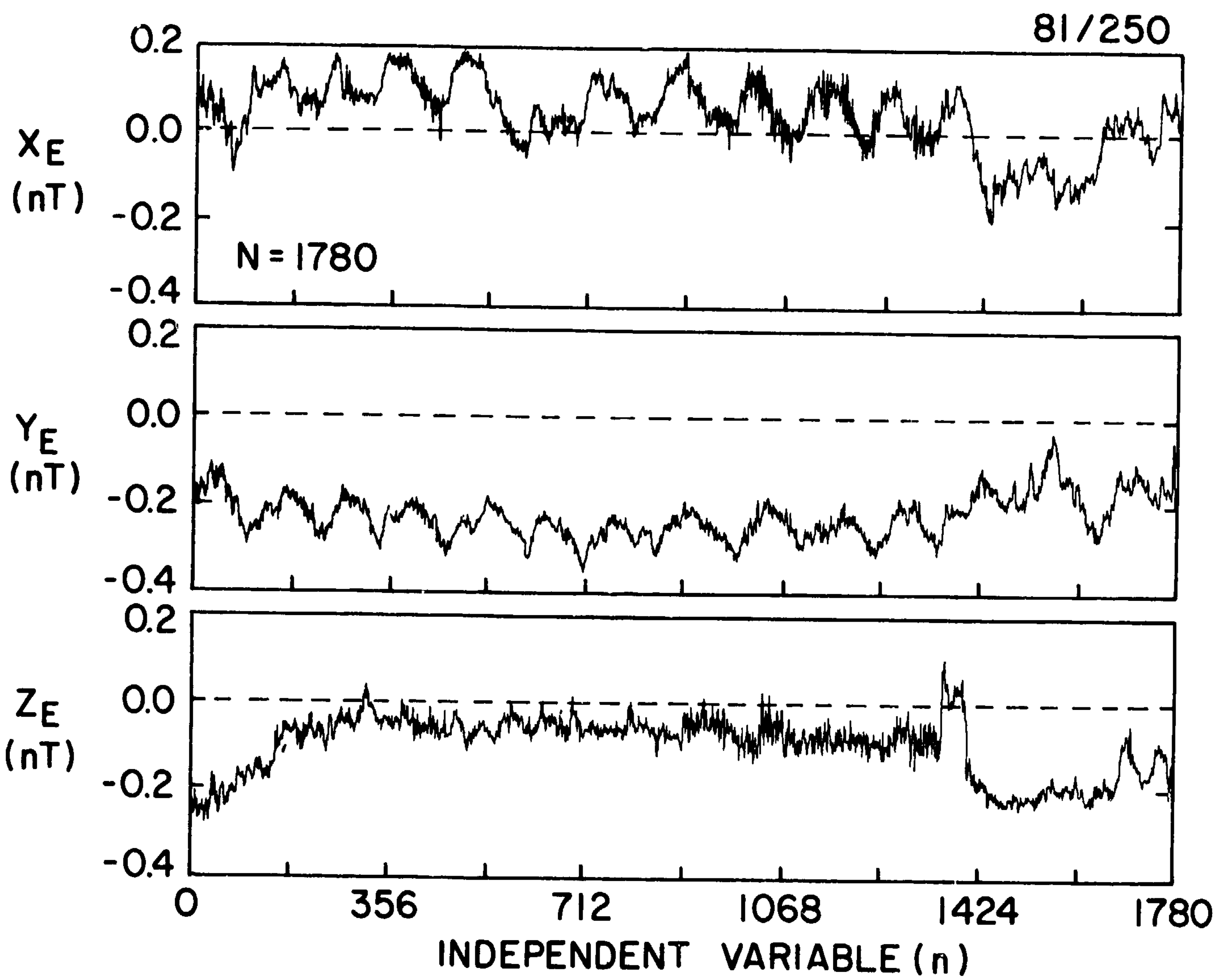


Fig. 5.1-12

Raw time series of Fig. 5.1-9 rotated into new coordinate system defined by the wave eigenvector set. Operator EIG generates sets of eigenvalues and eigenvectors for each spectral estimate. Those determining the coordinates of the data in this figure corresponded to the estimate containing the spectral peak denoted in Figs. 5.1-10 and 5.1-11. The X_E direction is that of the eigenvector associated with the largest eigenvalue, Z_E is that associated with the smallest, and Y_E is that associated with the intermediate value. Since the wave is nearly circularly polarized, the amplitudes of the oscillations in the Y_E direction are nearly as large as in the X_E direction and essentially zero in the Z_E direction. The magnitude of the field is not included since it is invariant under rotation.

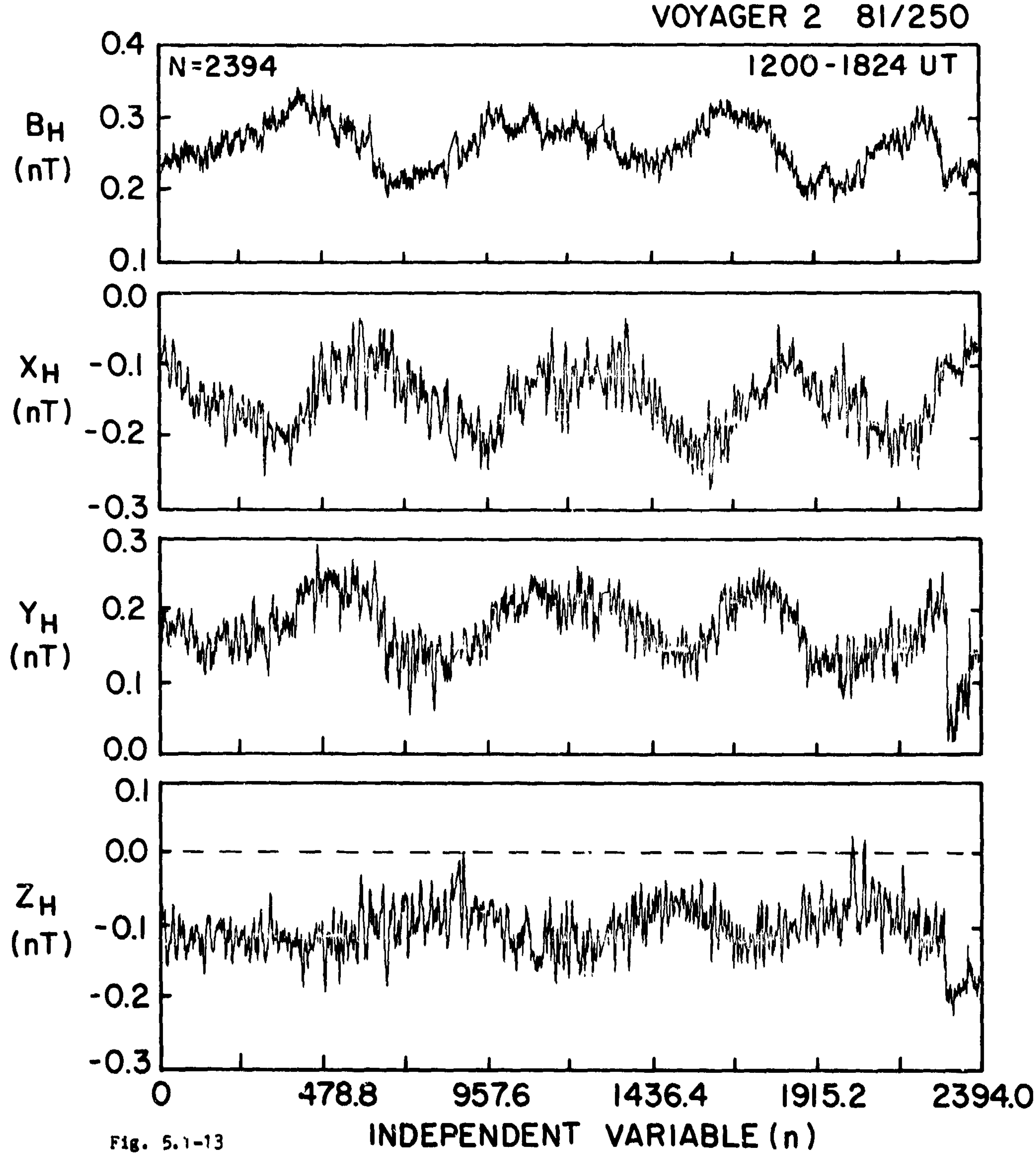


Fig. 5.1-13

Raw time series consisting of 9.6 sec averages for the portion of day 250 in which higher frequency fluctuations were seen also in Fig. 5.1-9 (1200-1824 UT). Note that in this case the amplitude of variations in the magnetic field magnitude (B_H) 's significantly lower than that of those in the vector components.

ORIGINAL PAGE IS
OF POOR QUALITY

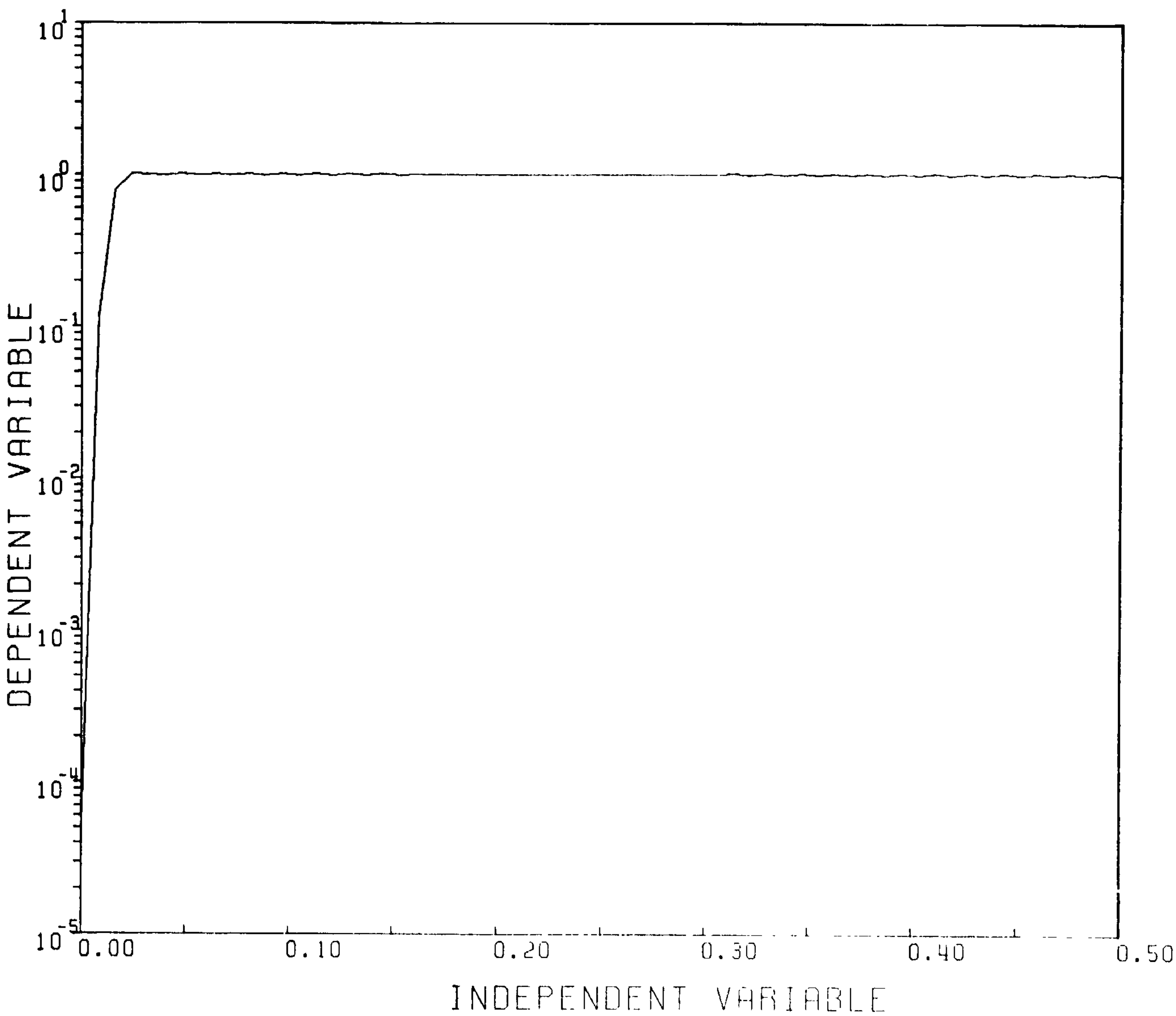


Fig. 5.1-14

The transfer function (frequency response) of the high pass filter designed with operator FILDES to remove the low frequency oscillations from the raw data. The function was obtained and plotted through the successive operators WINDOW FILDES, FFT WINDOW, and GRAPH FFTMP.

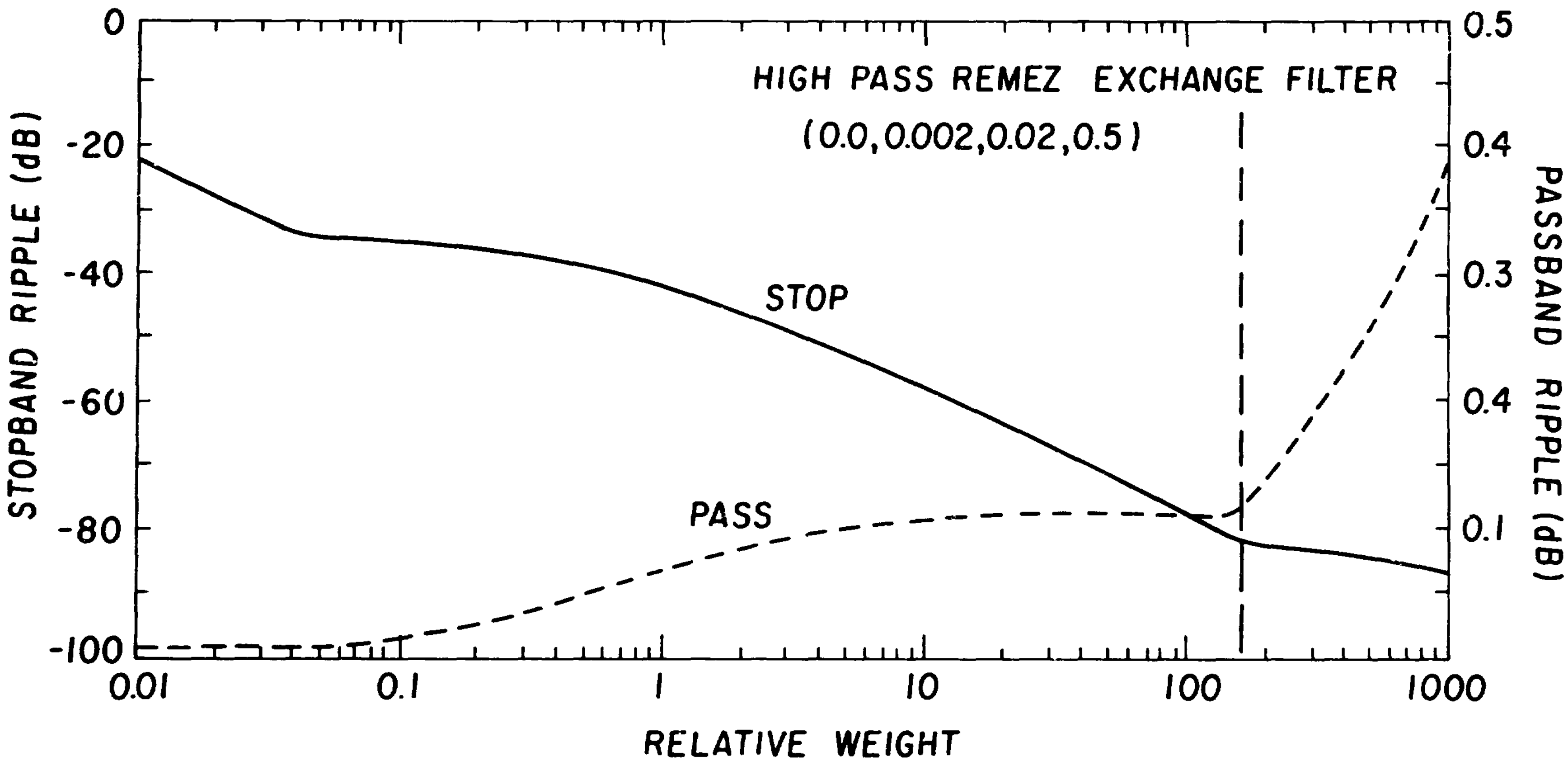


Fig. 5.1-15

Passband (PB) and Stopband (SB) ripple amplitude as a function of relative bandweight WTX(1)/WTX(2) for a 125-coefficient high pass filter with the given bandedges. The vertical dashed line delineates ripple characteristics corresponding to relative weight value of '82 used for filter shown in Fig. 5.1-14 (see text and Fig. 5.1-16).

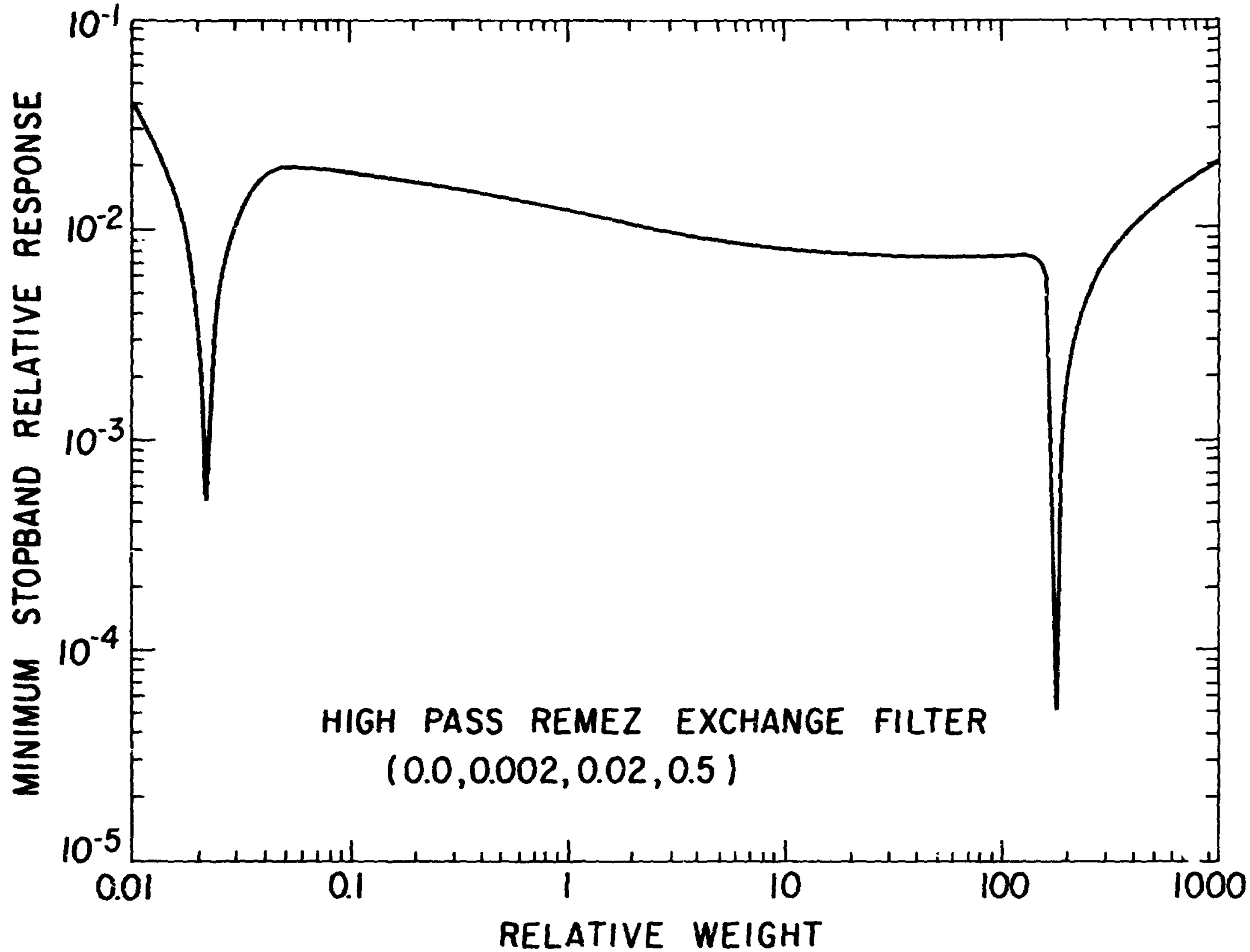


Fig. 5.1-16

Stopband minimum frequency response as a function of relative bandweight for filter described in Fig. 5.1-15 caption. The second (right-handed) minimum at WTX(1)/WTX(2) = 182 is maximum attenuation state for the given configuration.

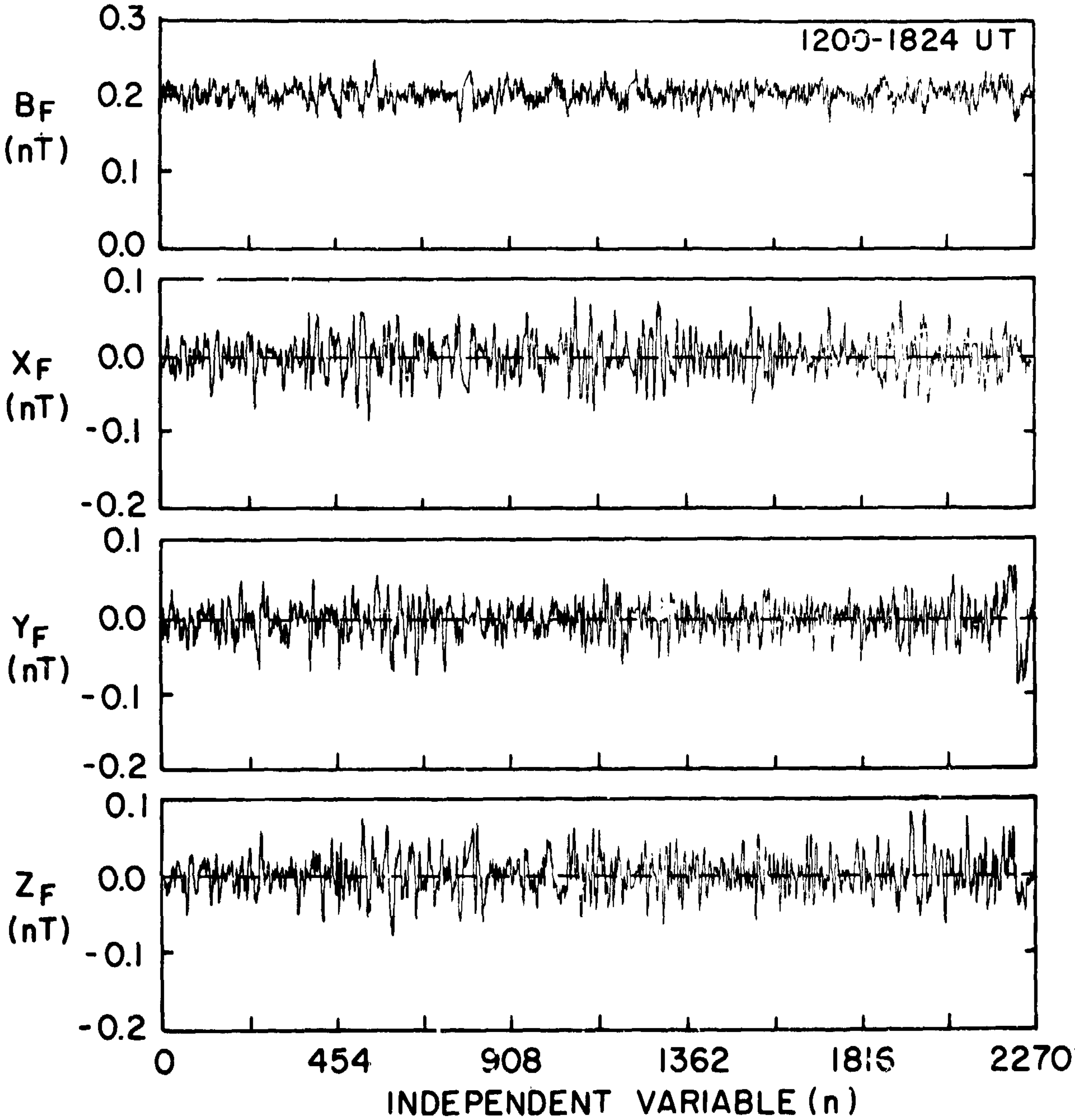


Fig. 5.1-17

Output from the application of the high pass filter to the time series shown in Fig. 5.1-13. The low frequency modulation of the data has been removed successfully.

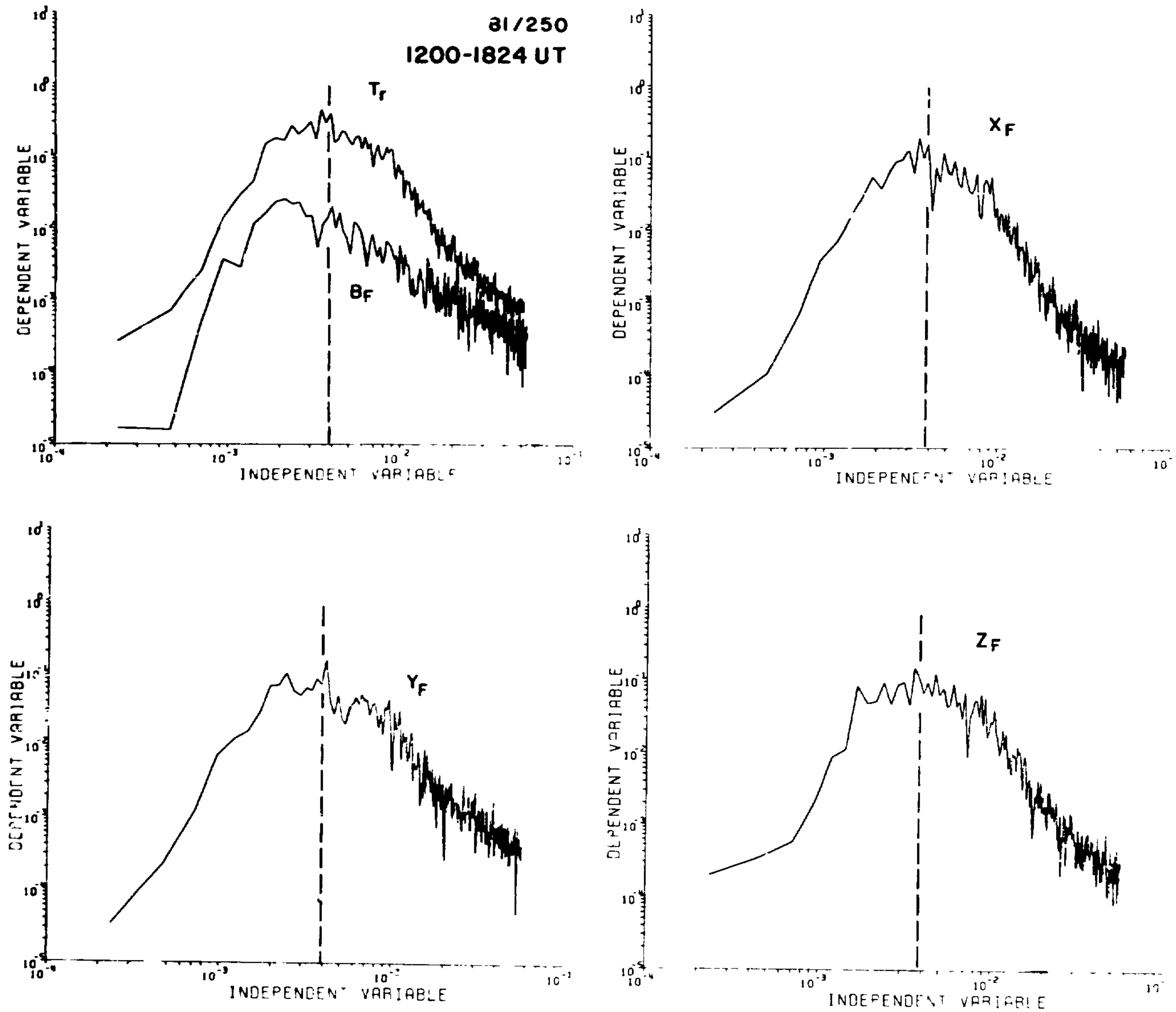


Fig. 5.1-18

Power spectra for filtered time series shown in Fig. 5.1-17. Suppression of power in low frequency variations results in a more distinctive display of the "high" frequency shoulder, with peak less prominent in this case but centered on a frequency of 3.4 X 10**-3 Hz (dashed vertical line).

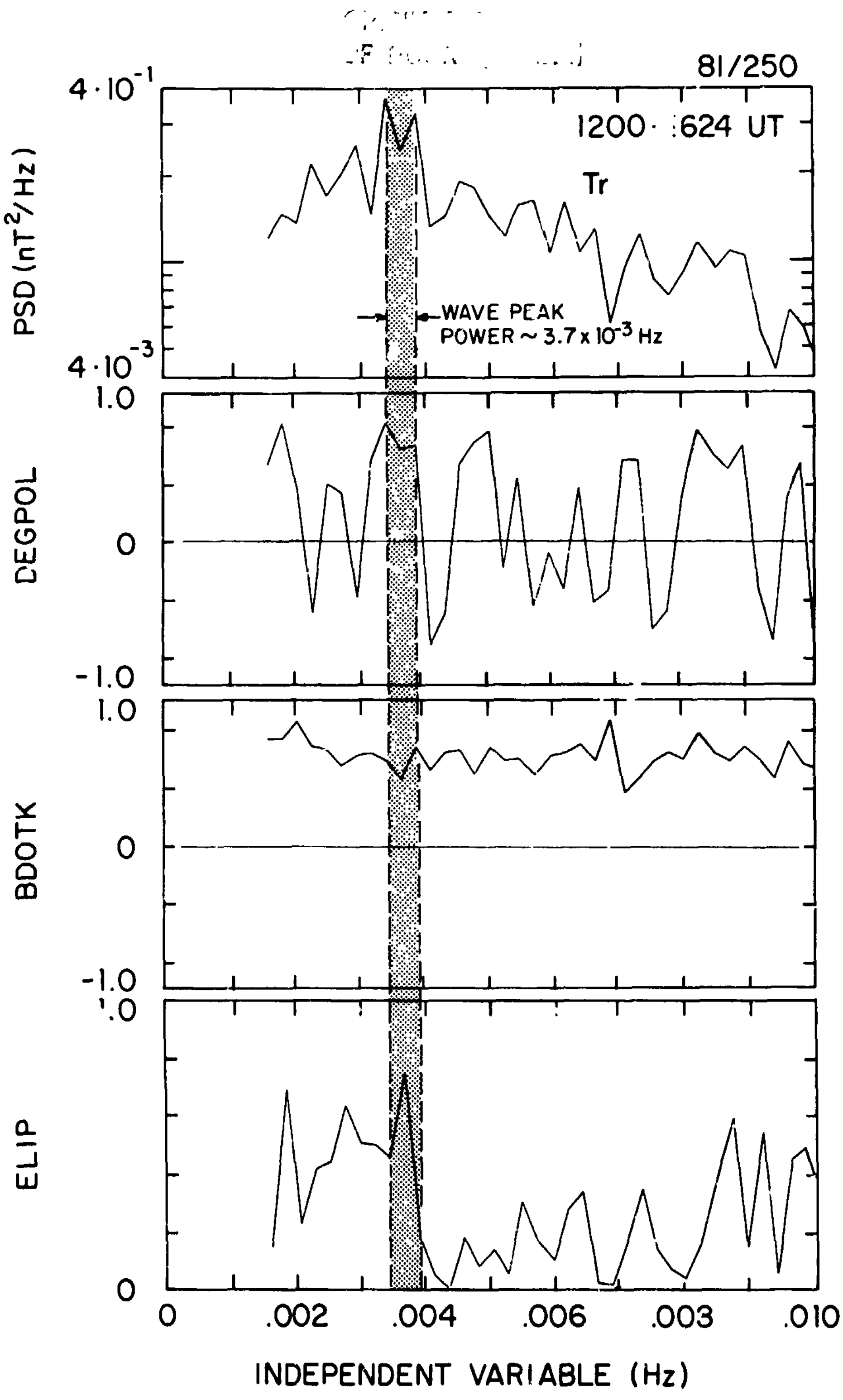


Fig. 5.1-19

Eigenfunction properties of the fluctuations shown in Fig. 5.1-17 in the frequency band 0.001 to 0.01 Hz. Parameters plotted are same as those in Fig. 5.1-11. Frequency of peak power in the fluctuations is delineated by the vertical hatched band. Comparison with Fig. 5.1-11 shows properties of waves at 3.4 X 10**-3 Hz are similar to those of the 1.75 X 10**-4 Hz waves.

5.2) DIGITAL FILTERS (OPERATORS FILDES, FILOPT, FILTER AND WTS)

5.2.1) INTRODUCTION TO DIGITAL FILTER DESIGN USING REMEZ EXCHANGE

Built into IDSP are operators that allow a user to design nonrecursive Finite Impulse Response (FIR) digital filters and then apply these filters to the time series. There is currently no capability to design Infinite Impulse Response (IIR) filters in IDSP.

Nonrecursive FIR filters have the following properties:

1) They are implemented in a nonrecursive fashion (operator FILTER), i.e., there is no feedback in the equation, (thus the name "Finite"; IIR filters must be implemented recursively).

2) Because of property (1) above, they are always stable. On the other hand the stability of IIR filters must always be checked.

3) They can be made to be linear phase operations, i.e., with zero phase distortion, by making the number, N, of filter coefficients odd and centering about sample number (N-1)/2. IIR filters are not of linear phase.

4) They are straightforward to design (operators FILOPT and FILDES).

5) FIR filters are not as efficient as IIR filter, i.e., for given design specifications IIR will require fewer multiplications. This efficiency can be important when large quantities of data are to be filtered and/or the filter has a large number of coefficients, or in the case where the filter is to be implemented in hardware. However, in IDSP these considerations are usually not important, so only FIR design is provided.

The FILOPT and FILDES operators use a computer program developed by McClellan, Parks and Rabiner (see Rabiner and Gold, 1975) that employs an algorithm called Remez exchange which is very efficient in producing an optimal (in the weighted Chebyshev sense) filter. This program has found wide application in the area of digital filtering and is considered state-of-the-art. The original program was obtained from the IEEE Library of Programs for Digital Signal Processing (see Programs for Digital Signal Processing, 1979) and restructured as operators for use in IDSP.

Before illustrating the use of these operators, we introduce the concept of normalized frequency: NORMALIZED FREQUENCY=(ACTUAL FREQUENCY)/(SAMPLE RATE), with units of cycles per sample. Thus the Nyquist frequency will always be 0.5 in normalized Hz. As an example, if we sample a 20 Hz signal 100 times/sec the normalized frequency is 0.2 Hz.

The important filter design specifications are shown in Fig. 5.2-1 (see Schaff, 1979) where the abscissa is plotted as normalized frequency.

$[0,f_p]$ is the normalized PASSBAND
$[f_p,f_s]$ is the normalized transition width, $delta_F$
$[f_s,0.5]$ is the normalized STOPBAND
$delta_1$ is defined as the amplitude deviation in the PASSBAND
$delta_2$ is defined as the amplitude deviation in the STOPBAND
PASSBAND RIPPLE is defined to be $20\log_{10}(1+delta_1)$ in dB
STOPBAND RIPPLE is defined to be $-20\log_{10}(delta_2)$ in dB

Typical values for the PASSBAND RIPPLE would be in the range of 1 to 0.01 dB and for the STOPBAND RIPPLE 20 to 90 dB.

5.2.2) DESIGN OF BANDPASS/BANDSTOP FILTERS

The following example illustrates how to go about designing a filter and in turn applying it to the time series. On the Dynamics Explorer Spacecraft there is a magnetic field experiment that provides a vector measurement of the ambient magnetic field every 0.5 sec. The data from one of the components of the vector is shown in Fig. 5.2-2 as a function of time. As is evident there appears to be a substantial periodic signal riding on top of the actual signal of interest. It was desired to remove this contamination from the signal of interest. First it was necessary to do a spectrum analysis on the data to determine the frequency(s) of the unwanted signals present in the data.

Fig 5.2-3 shows the results of using operators WINDOW, FFT, SPECT and GRAPH on this time series; note that at 0.0324 and 0.175 Hz there are peaks in the spectrum resulting from the unwanted signals.

The first order of business is to design a FIR filter to remove the unwanted 0.0324 Hz signal (0.0162 normalized Hz, e.g., 0.0324/2).

Referring to Fig. 5.2-1, once the designer has decided on the number of filter coefficients and the normalized transition width for a particular filter design the important parameters are 1 + $delta_1$ (the bandpass ripple) and $delta_2$ (the bandstop ripple) as a function of the weight ratio, $delta_1/delta_2$ = WTX(1)/WTX(2) (see Rabiner and Gold, 1975, pp 197,199). $Delta_2$ is a measure of how successfully we will attenuate the unwanted part of the frequency spectrum and we want to make this value as small as possible, while at the same time minimizing the bandpass ripple, 1 + $delta_1$. In order to aid in making the necessary tradeoff decisions we have developed an operator, FILOPT, which computes, using the Remez exchange algorithm, the values, in dB, for 1 + $delta_1$ and $delta_2$ over a range of $delta_1/delta_2$ ratios. Typically operator FILOPT (see Section 6) would be used to compute 1 + $delta_1$ and $delta_2$ over 4 decades of the ratio (from 0.1 to 1000). Because it is necessary to execute the Remez exchange algorithm over such a wide range of ratios it was decided to only do the computation for 9 values each decade and use operator INTERP to interpolate for the equidistant spacing required by IDSP. This dramatically decreases the amount of computation time required to execute FILOPT. The operator FILOPT produces an IDSP compatible data set consisting of elements of three time series for each weight ratio (see Fig. 5.2-4): series 1 = passband ripple = 20log(1+$delta_1$); series 2 = stopband ripple = 20log($delta_2$); and series 3 = the minimum value of $delta_2$ in the stopband. It is then necessary for the designer to interpolate (operator INTERP) data set FILOPT so as to produce the data set, INTERP, that can then be graphically displayed using operator GRAPH. The designer can then use these graphs to make a decision about what weight ratio will give the best stopband attenuation for the minimum of passband ripple.

Another approach, if there is some flexibility in the number of filter coefficients that can be used, and the designer wishes to prespecify the values for $delta_1$, $delta_2$ and the transition width is to estimate the number of filter coefficients from the following empirical formula which is valid for lowpass filters and has been implemented as operator WTS (see Section 6).

$$NFILT = D(delta_1, delta_2)/(delta\ F) - f(delta_1, delta_2)(delta\ F) + 1.0$$

$$delta\ F = ABS(f_s - f_p)$$

$$D(delta_1, delta_2) = [a_1(\log_{10} delta_1)^2 + a_2 \log_{10} delta_1 + a_3] \log_{10} delta_2 + [a_4(\log_{10} delta_1)^2 + a_5 \log_{10} delta_1 + a_6]$$

$$f(delta_1, delta_2) = b_1 + b_2(\log_{10} delta_1 - \log_{10} delta_2)$$

$a_1=5.309X10^{-3}$; $a_2=7.114X10^{-2}$; $a_3=-4.761X10^{-1}$; $a_4=-2.66X10^{-3}$; $a_5=-5.941X10^{-1}$
$a_6=-4.278X10^{-1}$; $b_1=11.01217$; $b_2=0.51244$

In our case we used FILOPT to determine the best tradeoff for $delta_1$ and $delta_2$. For this filter we decided to use 149 coefficients; f_p = 0.0055 and f_s = 0.0155 for the first transition width and f_s = 0.017 and f_p = 0.027 for the second transition width. We then use FILOPT to provide us with guidance on what is the best tradeoff for $delta_1$ and $delta_2$. Referring to the output of FILOPT shown in Fig. 5.2-4 we conclude that a ratio of WTX(1)/WTX(2) = 2.5 gives us small bandpass ripple and acceptable bandstop characteristics. We now have all of the design specifications for this first filter and can set up specifications to allow operator FILDES to be used to design the desired filter.

To do this we generate a data set on disk that has the following format:

First record:	NFILT,JTYPE,NBANDS,LGRID.	(4I5 format)
Next record(s):	EDGE	(4F15.9 format)
Next record(s):	FX	(4F15.9 format)
Next record(s):	WTX	(4F15.9 format)

These records can be placed into [usrid.IDSP]FOR058.DAT or a data set of another name, if the user does not want to enter design specifications interactively. In general it is probably better to generate the data set for purposes of documenting the design rather than entering them interactively.

NFILT-- FILTER LENGTH;

JTYPE-- TYPE OF FILTER

1 = MULTIPLE PASSBAND/STOPBAND FILTER,

2 = DIFFERENTIATOR,

3 = HILBERT TRANSFORM FILTER;

NBANDS-- NUMBER OF BANDS;

LGRID-- GRID DENSITY, WILL BE SET TO 16.

EDGE(2*NBANDS)-- BANDEDGE ARRAY, LOWER AND UPPER EDGES FOR EACH BAND WITH A MAXIMUM OF 10 BANDS.

FX(NBANDS)-- IDEALIZED PERFORMANCE ARRAY (OR DESIRED SLOPE IF A DIFFERENTIATOR) FOR EACH BAND.

WTX(NBANDS)-- WEIGHT FUNCTION ARRAY IN EACH BAND. FOR A DIFFERENTIATOR, THE WEIGHT FUNCTION IS INVERSELY PROPORTIONAL TO F.

In the case of this filter:

NFILT= 149; filter coefficients

JTYPE= 1; multiple PASSBAND/STOPBAND filter

NBAND= 3; one STOPBAND and two PASSBANDS

LGRID= 0; for purposes of plotting the transfer function-always set to 16 by the operator

BANDEDGE ARRAY (in normalized freq)= [0.0,0.0055]; [0.0155,0.017]; [0.027,0.5] Note that two records are required for the BANDEDGE ARRAY in this example because of the 4F15.9 format, and we have 6 numbers to enter.

FTX= 1.0,0.0,1.0; Idealized performance of the filter in each of the bands; in other words we wish no attenuation in the PASSBANDS and complete attenuation in the STOPBAND.

WTX= 1.0,2.5,1.0; the ratio of $delta_1/delta_2$.

Now that we have developed the design specifications for the filter, we can enter the specifications into a data set or enter them interactively; IDSP will do it either way. An example of the design specification data set is shown below. We then execute the operator FILDES, and it will produce a standard IDSP data set that contains the computed filter coefficients.

```
149,1,3,0
0.0,0.0055,0.0155,0.017
0.027,0.5
1.0,0.0,1.0
1.0,2.5,1.0
```

Looking at Table 5.2-1, which is also produced by the execution of the operator FILDES, we see the coefficients and design specifications printed out. "LOWER BANDEDGE & UPPER BANDEDGE" gives us the bandedge array values; "DESIRED VALUE" gives the idealized performance; "WEIGHTING" gives the ratio of $delta_1/delta_2$ and "DEVIATION & DEVIATION in dB" give the amplitude deviation resulting from the design, 1.0 + $delta_1$ and $delta_2$. Note Table 5.2-1 only shows 63 coefficients because of symmetry.

The next step is to look at the frequency response (transfer function) of the filter that was just designed to ensure that it is what we want. To do that remember that the FIR filter is simply a "black box" through which we are going to pass our time series, and what we want to know is the frequency response of this "black box" for all frequencies from DC up to the Nyquist frequency (0.5 normalized frequency). The frequency response of a LINEAR SHIFT INVARIANT SYSTEM, in this case our FIR filter, can be obtained by taking the Fourier transform of the impulse response of the linear system. All linear systems produce an output by convolving the input signal with the impulse response. In our case the input signal is our time series and the just calculated filter coefficients are the impulse response. If we take the Fourier transform of the filter coefficients we will obtain the frequency response of the filter. Note that it is only necessary to look at the amplitude part of the frequency response because we know by definition of an FIR filter that the phase is linear.

After execution of operator WINDOW (using the rectangular window and zero padding options to increase resolution of the plot) on data set FILDES (which contains the filter coefficients), we execute the operator FFT with the magnitude/phase option (which stores the results in magnitude/phase form) on data set WINDOW. We then execute the operator GRAPH on data set FFTMP to produce Fig. 5.2-5 which is the amplitude response for the filter from DC up to 0.5 normalized Hz. Note that we used operator WINDOW and zero padded to better define the transfer function for purposes of plotting. Fig. 5.2-6 shows the original time series after being filtered with the FIR filter that we just designed.

In a similar fashion we design the second filter, again using FILOPT to determine the optimal design, to remove the unwanted 0.175 Hz (or 0.0875 normalized Hz) signal. The amplitude response of this second filter is shown in Fig 5.2-7. The design specification data set appears below:

125,1,3,0
0.0,0.05,0.075,0.1
0.125,0.5
1.0,0.0,1.0
1.0,38.0,1.0

Finally, we filter the time series a <u>second</u> time to remove the 0.175 Hz signal and the resulting time series is shown in Fig. 5.2-8, substantially free of both unwanted signals.

It should be noted that for purposes of illustration we have designed two separate filters; however, both unwanted frequencies could have been removed with a <u>single</u> filter that has two stopbands and three passbands for a total of 5 bands. Operator FILOPT and FILDES will permit you to design a filter with up to a total of 10 bands and 512 coefficients.

The design tradeoff information of this 5 band filter from FILOPT is shown in Fig. 5.2-9. The transfer function is shown in Fig. 5.2-10 where a $delta_1/delta_2$= 8.0 was chosen as optimal. Table 2.0-1 results from the execution of operator FILDES and presents the coefficients and other parameters of this 5 band filter. Fig. 5.2-11 shows the results of filtering the original time series from Fig. 5.2-2 with this filter. Fig. 5.2-12 shows the power spectrum of this filtered data processed with the 5-band filter; Fig. 5.2-13 is the power spectrum of the data set filtered with the two 3-band filter. In both cases the contaimination signals have been removed.

Fig. 5.2-21 depicts the series of operations involved in the optimal design of a multiple band filter using IDSP.

5.2.3) USE OF A DIFFERENTIATING FILTER:

The spectral analysis of a finite duration time series may include errors due to "leakage". This error is most likely to modify regions of the spectrum which contain little power by leaking power from more energetic frequencies, through the influence of the side lobes of the data window (see Section 5.3 for a detail discussion of windows) used in the analysis. One way to determine if a particular spectral feature is real or due to leakage is to "pre-whiten" the series using a differentiating filter. In effect this multiplies the original spectrum by the "one" power of the frequency (F) which amplifies the high frequency relative to the low. For spectra which are dominated by low frequency power, the differentiator flattens, or "whitens" the spectrum and decreases the leakage effect on high frequency spectral features.

As an example of designing a differentiating filter consider the tangential (T) and normal (N) components of a magnetic field vector plotted as a function of time in Fig 5.2-14 and Fig. 5.2-15, respectively. Fig 5.2-16 shows the power spectrum of this vector. This spectrum decreases at approximately $F^{-5/3}$ for large F, so that spectral features in the higher F region may be influenced by leakage from the lower F region.

A differentiating filter was designed using the operator FILDES to prewhiten these data. This filter is 126 coefficients in length with a cutoff frequency at 0.5 and a slope of 1. The design specification data set is as follows:

126,2,1,16
0.0,0.5
1.0
1.0

The filter coefficients are shown in Table 5.2-2, and the transfer function is plotted in Fig. 5.2-17 The filtered versions of T and N are shown in Fig. 5.2-18 and 5.2-19, respectively. The low frequency features are visibly absent, and the data appear to be much more like white noise. This is borne out by the power spectrum shown in Fig. 5.2-20 which is now flat at large F.

REFERENCES

Programs for Digital Signal Processing, IEEE Press, 1979.

Rabiner, L. R., B. Gold, Theory and Application of Digital Signal Processing, Prentice-Hall Inc., Englewood Cliffs, N J, 1975.

Schaff, W. E., Course Notes: Modern Methods of Digital Signal Processing, Integrated Computer Systems, Inc, October 1979.

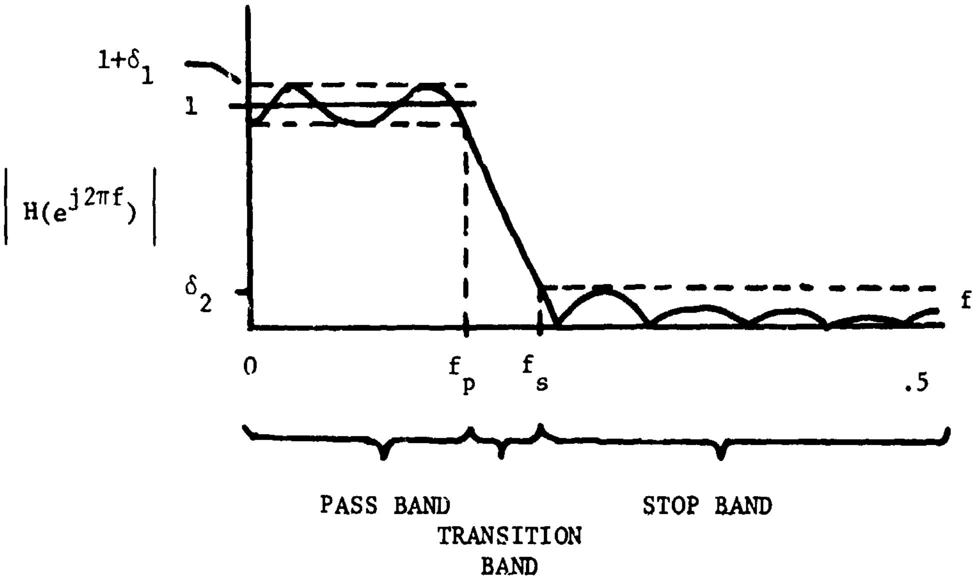
```
************************************************************
        FINITE IMPULSE RESPONSE (FIR)
      LINEAR PHASE DIGITAL FILTER DESIGN
          REMEZ EXCHANGE ALGORITHM

              BANDPASS FILTER

            FILTER LENGTH = 149

        ***** IMPULSE RESPONSE *****
      H(  1) =  0.59528281E-02 = H(149)
      H(  2) = -0.29914447E-02 = H(148)
      H(  3) = -0.26762583E-02 = H(147)
      H(  4) = -0.26017630E-02 = H(146)
      H(  5) = -0.26937625E-02 = H(145)
      H(  6) = -0.29003518E-02 = H(144)
      H(  7) = -0.31854392E-02 = H(143)
      H(  8) = -0.35189979E-02 = H(142)
      H(  9) = -0.38829341E-02 = H(141)
      H( 10) = -0.42572655E-02 = H(140)
      H( 11) = -0.46299286E-02 = H(139)
      H( 12) = -0.49843024E-02 = H(138)
      H( 13) = -0.53115957E-02 = H(137)
      H( 14) = -0.55976524E-02 = H(136)
      H( 15) = -0.58347476E-02 = H(135)
      H( 16) = -0.60097934E-02 = H(134)
      H( 17) = -0.61165341E-02 = H(133)
      H( 18) = -0.61437651E-02 = H(132)
      H( 19) = -0.60864589E-02 = H(131)
      H( 20) = -0.59347842E-02 = H(130)
      H( 21) = -0.56859865E-02 = H(129)
      H( 22) = -0.53334623E-02 = H(128)
      H( 23) = -0.48777661E-02 = H(127)
      H( 24) = -0.43144981E-02 = H(126)
      H( 25) = -0.36470189E-02 = H(125)
      H( 26) = -0.28742868E-02 = H(124)
      H( 27) = -0.20048874E-02 = H(123)
      H( 28) = -0.10409054E-02 = H(122)
      H( 29) =  0.54370380E-05 = H(121)
      H( 30) =  0.11301419E-02 = H(120)
      H( 31) =  0.23149711E-02 = H(119)
      H( 32) =  0.35534513E-02 = H(118)
      H( 33) =  0.48225583E-02 = H(117)
      H( 34) =  0.61177318E-02 = H(116)
      H( 35) =  0.74068061E-02 = H(115)
      H( 36) =  0.86910734E-02 = H(114)
      H( 37) =  0.99259838E-02 = H(113)
      H( 38) =  0.11119566E-01 = H(112)
      H( 39) =  0.12242499E-01 = H(111)
      H( 40) =  0.13257143E-01 = H(110)
      H( 41) =  0.14175730E-01 = H(109)
      H( 42) =  0.14966194E-01 = H(108)
      H( 43) =  0.15614625E-01 = H(107)
      H( 44) =  0.16096387E-01 = H(106)
      H( 45) =  0.16410498E-01 = H(105)
      H( 46) =  0.16543033E-01 = H(104)
      H( 47) =  0.16449578E-01 = H(103)
      H( 48) =  0.16237279E-01 = H(102)
      H( 49) =  0.15784621E-01 = H(101)
      H( 50) =  0.15127735E-01 = H(100)
      H( 51) =  0.14271768E-01 = H( 99)
      H( 52) =  0.13220074E-01 = H( 98)
      H( 53) =  0.11983175E-01 = H( 97)
      H( 54) =  0.10570571E-01 = H( 96)
      H( 55) =  0.89974189E-02 = H( 95)
      H( 56) =  0.72780177E-02 = H( 94)
      H( 57) =  0.54319571E-02 = H( 93)
      H( 58) =  0.34783420E-02 = H( 92)
      H( 59) =  0.14410270E-02 = H( 91)
      H( 60) = -0.65731985E-03 = H( 90)
      H( 61) = -0.27894827E-02 = H( 89)
      H( 62) = -0.49299193E-02 = H( 88)
      H( 63) = -0.70494656E-02 = H( 87)
      H( 64) = -0.91231456E-02 = H( 86)
      H( 65) = -0.11123234E-01 = H( 85)
      H( 66) = -0.13025748E-01 = H( 84)
      H( 67) = -0.14802608E-01 = H( 83)
      H( 68) = -0.16431367E-01 = H( 82)
      H( 69) = -0.17889120E-01 = H( 81)
      H( 70) = -0.19159628E-01 = H( 80)
      H( 71) = -0.20223662E-01 = H( 79)
      H( 72) = -0.21065198E-01 = H( 78)
      H( 73) = -0.21675173E-01 = H( 77)
      H( 74) = -0.22042712E-01 = H( 76)
      H( 75) =  0.97762916E+00 = H( 75)

                      BAND  1        BAND  2       BAND  3
LOWER BAND EDGE      0.0000000      0.0155000     0.0270000
UPPER BAND EDGE      0.0055000      0.0170000     0.5000000
DESIRED VALUE        1.0000000      0.0000000     1.0000000
WEIGHTING            1.0000000      2.5000000     1.0000000
DEVIATION            0.0151031      0.0060413     0.0151031
DEVIATION IN DB      0.1302031    -44.3774643     0.1302031

EXTREMAL FREQUENCIES--MAXIMA OF THE ERROR CURVE
    0.0000000   0.0041667   0.6163333   0.0170000   0.0270000
    0.0290833   0.0336667   0.0399167   0.0461667   0.0528333
    0.0595000   0.0661666   0.0728333   0.0795000   0.0865833
    0.0932499   0.0999166   0.1069999   0.1136666   0.1203332
    0.1274166   0.1340832   0.1407499   0.1478332   0.1544999
    0.1611665   0.1682499   0.1749165   0.1815832   0.1886665
    0.1953332   0.2019998   0.2086665   0.2157498   0.2224165
    0.2290832   0.2357498   0.2428331   0.2494998   0.2561665
    0.2632498   0.2699164   0.2765831   0.2832498   0.2903331
    0.2969998   0.3036664   0.3103331   0.3174164   0.3240831
    0.3307497   0.3374164   0.3444997   0.3511664   0.3578330
    0.3644997   0.3715830   0.3782497   0.3849163   0.3915830
    0.3986663   0.4053330   0.4119996   0.4186663   0.4257496
    0.4324163   0.4390830   0.4457496   0.4528329   0.4594996
    0.4661663   0.4728329   0.4794996   0.4865829   0.4932496
    0.5000000

*****************  ********************************************
```

TABLE 5.2-1

```
*****************************************************
              FINITE IMPULSE RESPONSE (FIR)
          LINEAR PHASE DIGITAL FILTER DESIGN
              REMEZ EXCHANGE ALGORITHM

                    DIFFERENTIATOR

                 FILTER LENGTH = 126

            ***** IMPULSE RESPONSE *****
          H( 1) =  0.13726675E-03 = -H(126)
          H( 2) = -0.16789764E-03 = -H(125)
          H( 3) =  0.46877274E-04 = -H(124)
          H( 4) = -0.28490012E-04 = -H(123)
          H( 5) =  0.23145527E-04 = -H(122)
          H( 6) = -0.21183154E-04 = -H(121)
          H( 7) =  0.20456233E-04 = -H(120)
          H( 8) = -0.20280140E-04 = -H(119)
          H( 9) =  0.20408930E-04 = -H(118)
          H(10) = -0.20737123E-04 = -H(117)
          H(11) =  0.21182379E-04 = -H(116)
          H(12) = -0.21710353E-04 = -H(115)
          H(13) =  0.22335353E-04 = -H(114)
          H(14) = -0.23029677E-04 = -H(113)
          H(15) =  0.23785062E-04 = -H(112)
          H(16) = -0.24620294E-04 = -H(111)
          H(17) =  0.25514664E-04 = -H(110)
          H(18) = -0.26486456E-04 = -H(109)
          H(19) =  0.27540846E-04 = -H(108)
          H(20) = -0.28666873E-04 = -H(107)
          H(21) =  0.29886767E-04 = -H(106)
          H(22) = -0.31212112E-04 = -H(105)
          H(23) =  0.32635620E-04 = -H(104)
          H(24) = -0.34174525E-04 = -H(103)
          H(25) =  0.35854351E-04 = -H(102)
          H(26) = -0.37678623E-04 = -H(101)
          H(27) =  0.39653518E-04 = -H(100)
          H(28) = -0.41857053E-04 = -H( 99)
          H(29) =  0.44097524E-04 = -H( 98)
          H(30) = -0.46758190E-04 = -H( 97)
          H(31) =  0.49797836E-04 = -H( 96)
          H(32) = -0.52752290E-04 = -H( 95)
          H(33) =  0.5598259?E-04 = -H( 94)
          H(34) = -0.59666606E-04 = -H( 93)
          H(35) =  0.63821462E-04 = -H( 92)
          H(36) = -0.68460155E-04 = -H( 91)
          H(37) =  0.73613737E-04 = -H( 90)
          H(38) = -0.79358404E-04 = -H( 89)
          H(39) =  0.85837455E-04 = -H( 88)
          H(40) = -0.93181690E-04 = -H( 87)
          H(41) =  0.10149305E-03 = -H( 86)
          H(42) = -0.11099881E-03 = -H( 85)
          H(43) =  0.12194661E-03 = -H( 84)
          H(44) = -0.13460728E-03 = -H( 83)
          H(45) =  0.14939171E-03 = -H( 82)
          H(46) = -0.16677857E-03 = -H( 81)
          H(47) =  0.18742082E-03 = -H( 80)
          H(48) = -0.21219914E-03 = -H( 79)
          H(49) =  0.24227699E-03 = -H( 78)
          H(50) = -0.27927748E-03 = -H( 77)
          H(51) =  0.32552367E-03 = -H( 76)
          H(52) = -0.38435805E-03 = -H( 75)
          H(53) =  0.46078526E-03 = -H( 74)
          H(54) = -0.56260586E-03 = -H( 73)
          H(55) =  0.70245465E-03 = -H( 72)
          H(56) = -0.90188818E-03 = -H( 71)
          H(57) =  0.12003481E-02 = -H( 70)
          H(58) = -0.16759943E-02 = -H( 69)
          H(59) =  0.25029865E-02 = -H( 68)
          H(60) = -0.41368390E-02 = -H( 67)
          H(61) =  0.81069879E-02 = -H( 66)
          H(62) = -0.22517066E-01 = -H( 65)
          H(63) =  0.20264359E+00 = -H( 64)

                             BAND  1
LOWER BAND EDGE            0.0000000
UPPER BAND EDGE            0.5000000
DESIRED SLOPE              1.0000000
WEIGHTING                  1.0000000
DEVIATION                  0.0014572

EXTREMAL FREQUENCIES--MAXIMA OF THE ERROR CURVE
    0.0004960  0.0079365  0.0158730  0.0243056  0.0322421
    0.0401786  0.0481151  0.0565477  0.0644842  0.0724207
    0.0803572  0.0882937  0.0967262  0.1046626  0.1125991
    0.1205356  0.1289681  0.1369046  0.1448411  0.1527776
    0.1612101  0.1691465  0.1770830  0.1850195  0.1929560
    0.2013885  0.2093250  0.2172614  0.2251979  0.2336304
    0.2415669  0.2495034  0.2574399  0.2658724  0.2738088
    0.2817453  0.2896818  0.2976183  0.3060508  0.3139873
    0.3219237  0.3298602  0.338_927  0.3462292  0.3541657
    0.3621022  0.3705347  0.3784711  0.3864076  0.3943441
    0.4027766  0.4107131  0.4186496  0.4265860  0.4350185
    0.4429550  0.4508915  0.4588280  0.4667645  0.4747009
    0.4826374  0.4900779  0.4970223  0.5000000
*****************************************************************
```

Table 5.2-2

ORIGINAL PAGE IS
OF POOR QUALITY



δ_1 = AMPLITUDE DEVIATION IN PASS BAND

δ_2 = AMPLITUDE DEVIATION IN STOP BAND

ΔF = NORMALIZED TRANSITION WIDTH

$\{\delta_1, \delta_2, f_p, f_s\}$ = DESIGN TOLERANCES, OR SPECIFICATIONS.

Fig. 5.2-1

FIR filter design specifications after Schaff, 1979.

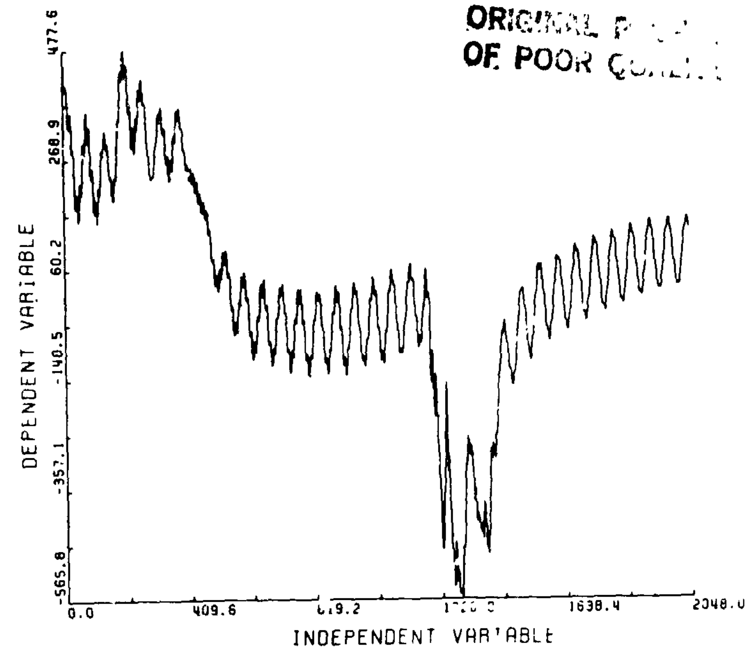


Fig. 5.2-2

Raw magnetic field time series from the DE spacecraft sampled at twice/sec, showing contamination riding on top of the desired signal.

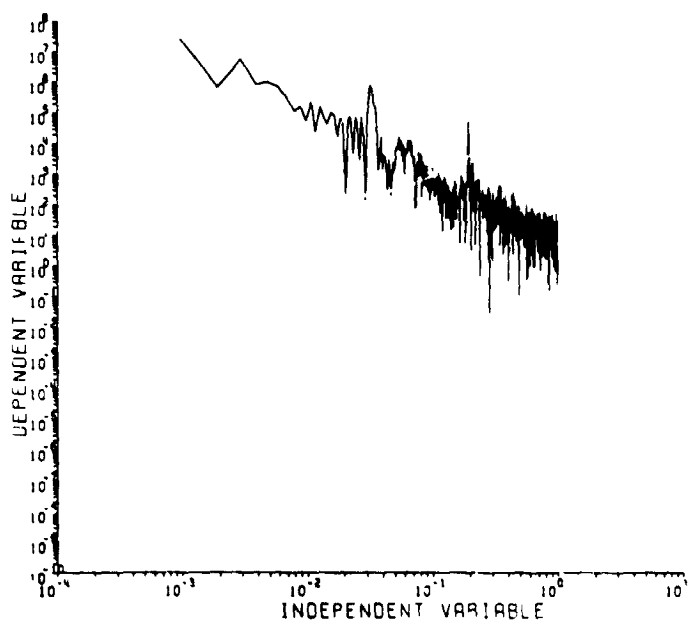


Fig. 5.2-3

Power spectrum of the raw time series shown in Fig. 5.2-2 showing power at 0.0324 and 0.175 Hz.

ORIGINAL PAGE IS OF POOR QUALITY

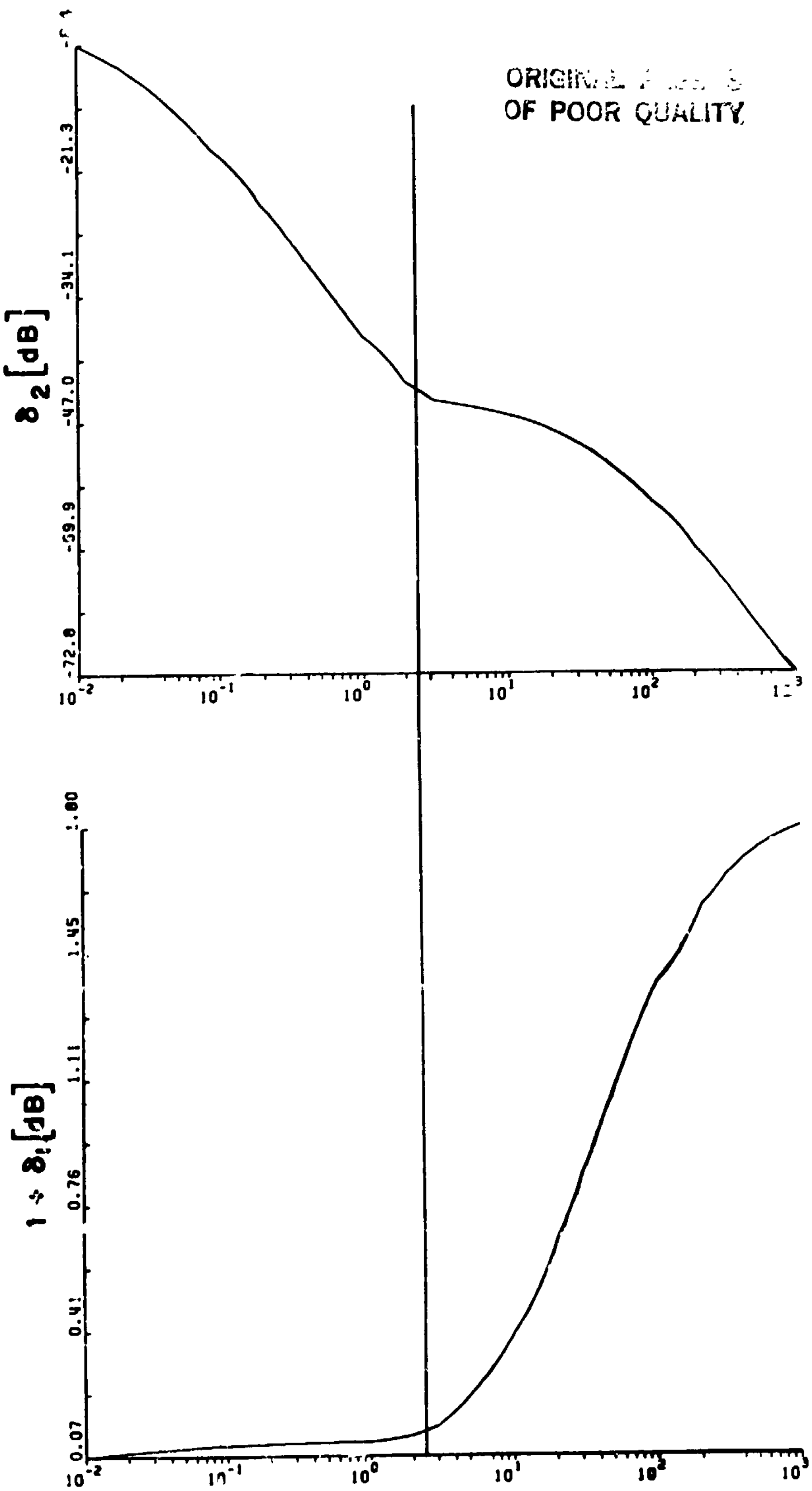


Fig. 5.2-4

The output of operator FILOPT showing 20log(1 + $delta_1$), and 20log($delta_2$) plotted against the weight ratio, $delta_1$/$delta_2$. From these plots we concluded that the best tradeoff was a weight ratio of 2.5, which in turn was used to design the filter shown in Fig. 5.2-5.

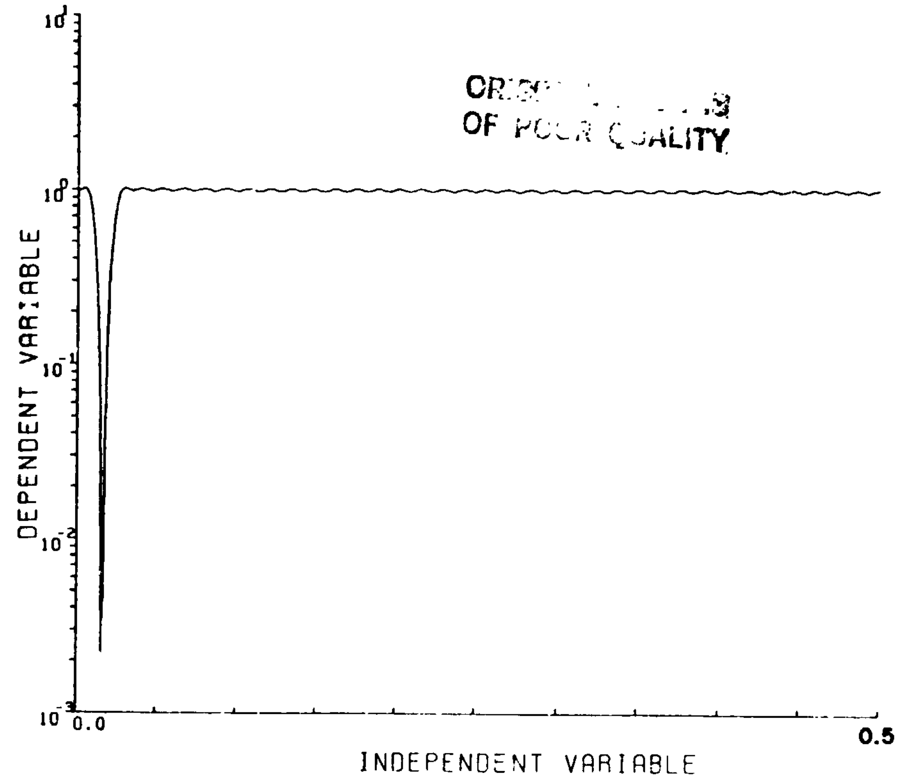


Fig. 5.2-5

Transfer function of a notch filter designed to remove the unwanted 0.0324 Hz signal from the original time series.

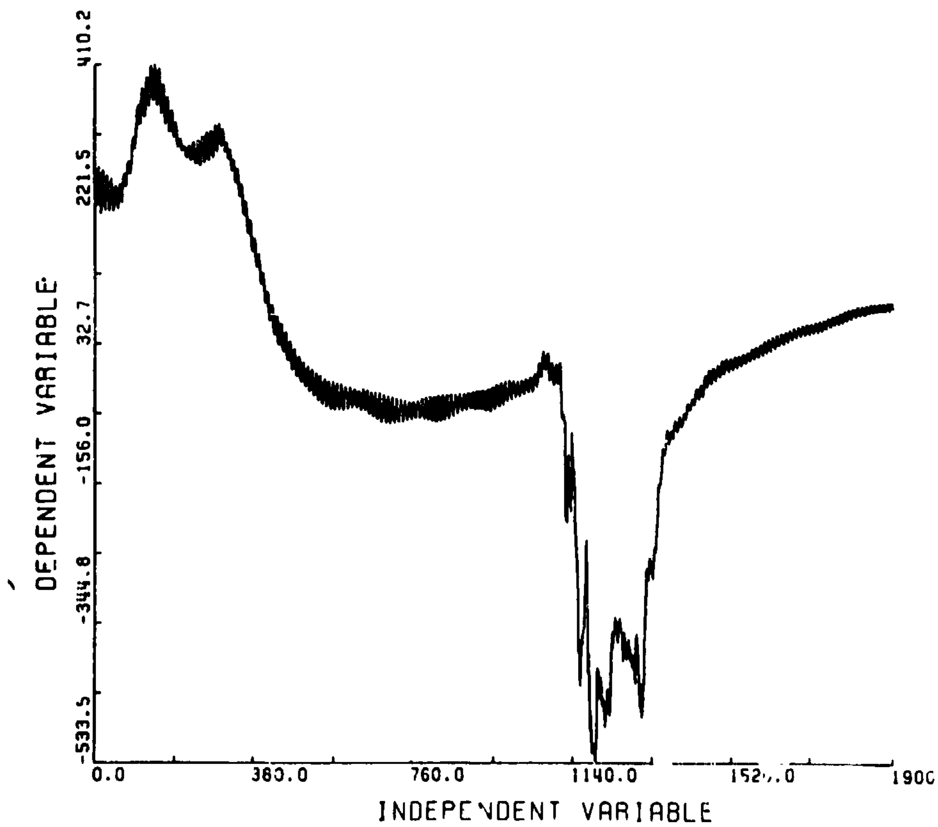


Fig. 5.2-6

Time series after applying the filter shown in Fig. 5.2-5.

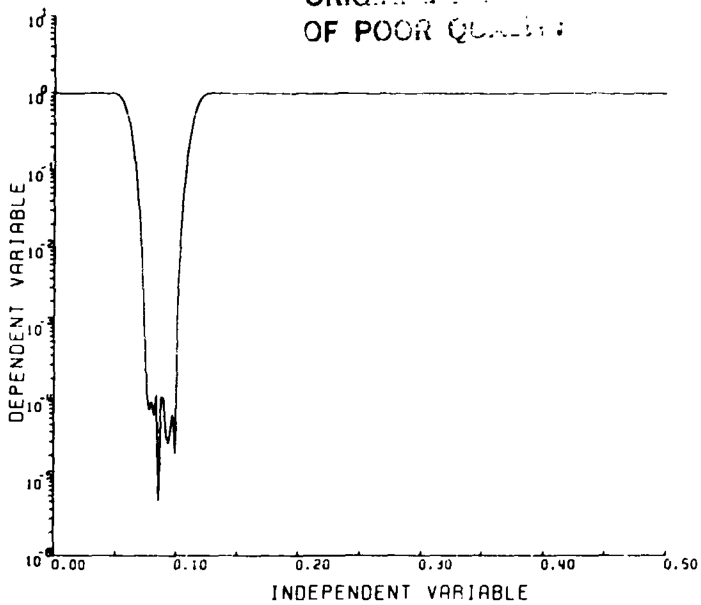


Fig. 5.2-7

Transfer function of a notch filter designed to remove the unwanted 0.175 Hz signal from the time series.

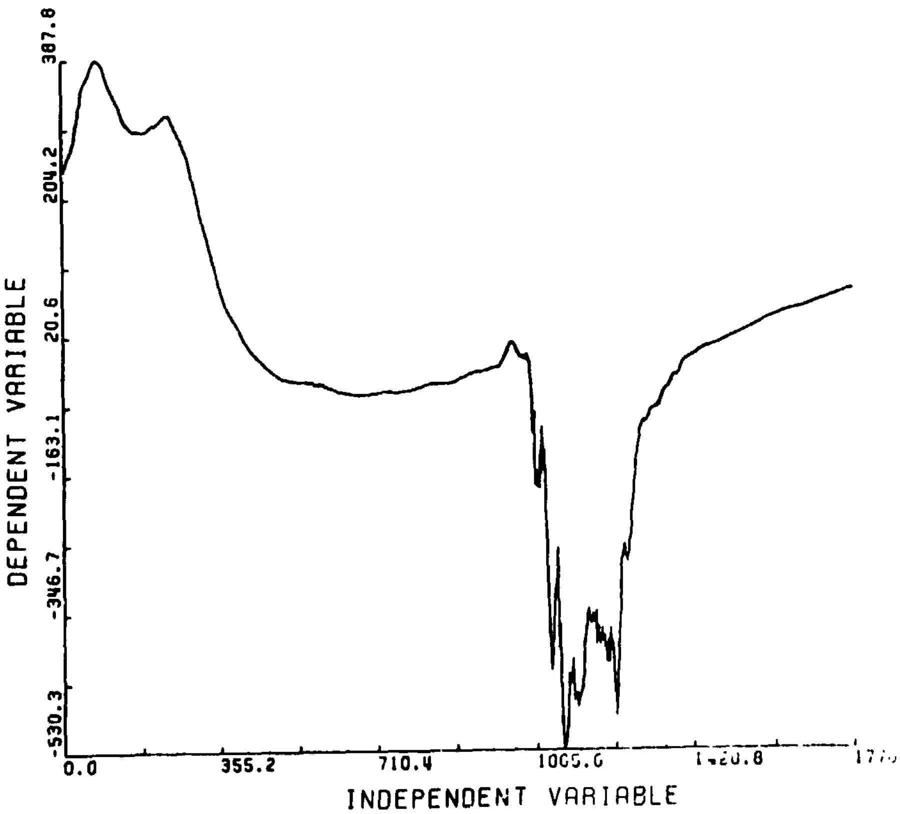


Fig. 5.2-8

Time series after application of both the filter shown in Fig. 5.2-5 and Fig. 5.2-7. Note that the time series is substantially free of the unwanted signals.

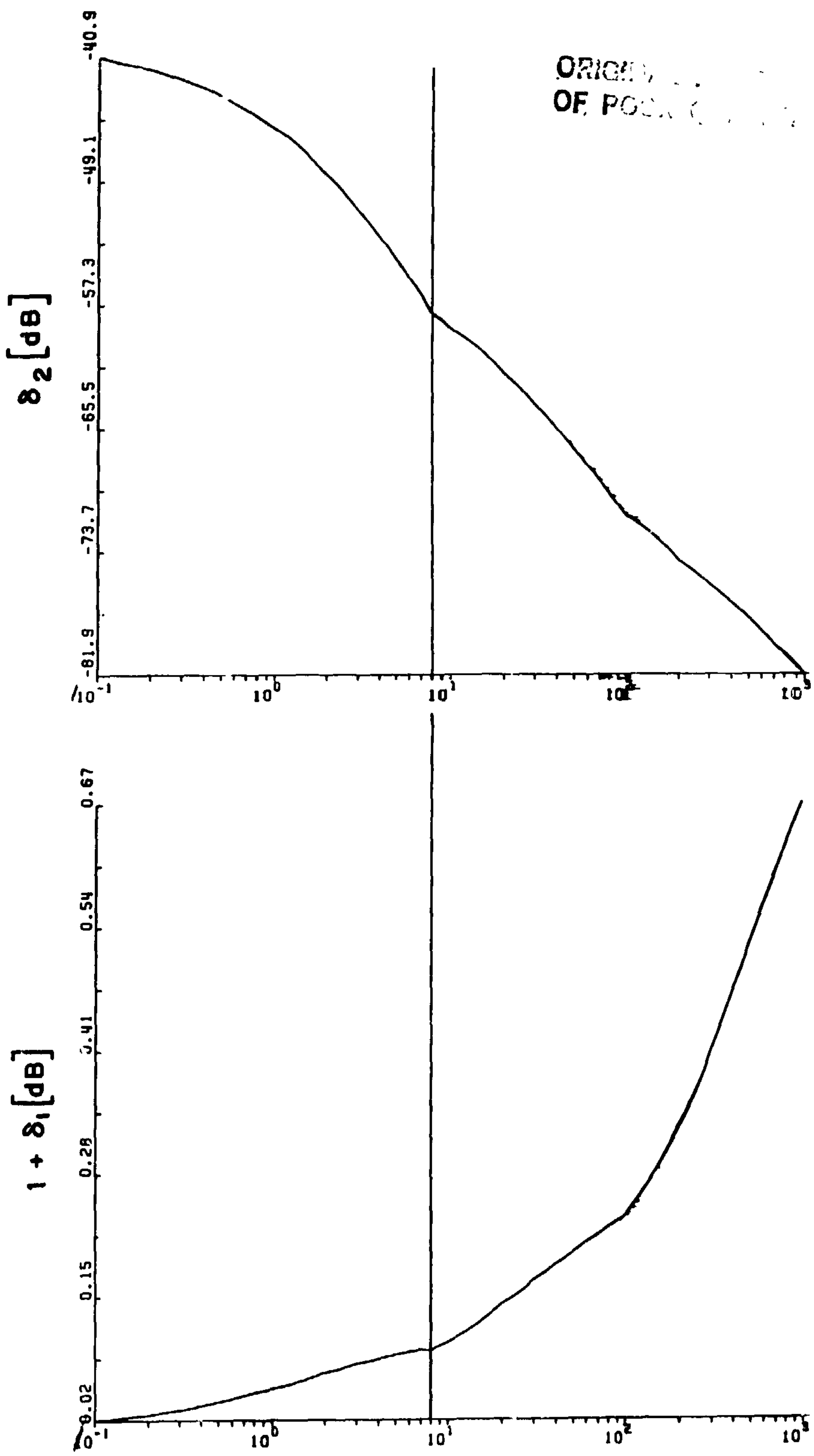


Fig. 5.2-9 RELATIVE WEIGHT

The output of operator FILOPT showing 20log(1 + delta$_1$), and 20log(delta$_2$) plotted against the weight ratio, delta$_1$/delta$_2$. From these plots we concluded that the best tradeoff was a weight ratio of 8, which in turn was used to design the filter shown in Fig. 5.2-10.

ORIGINAL PAGE IS
OF POOR QUALITY

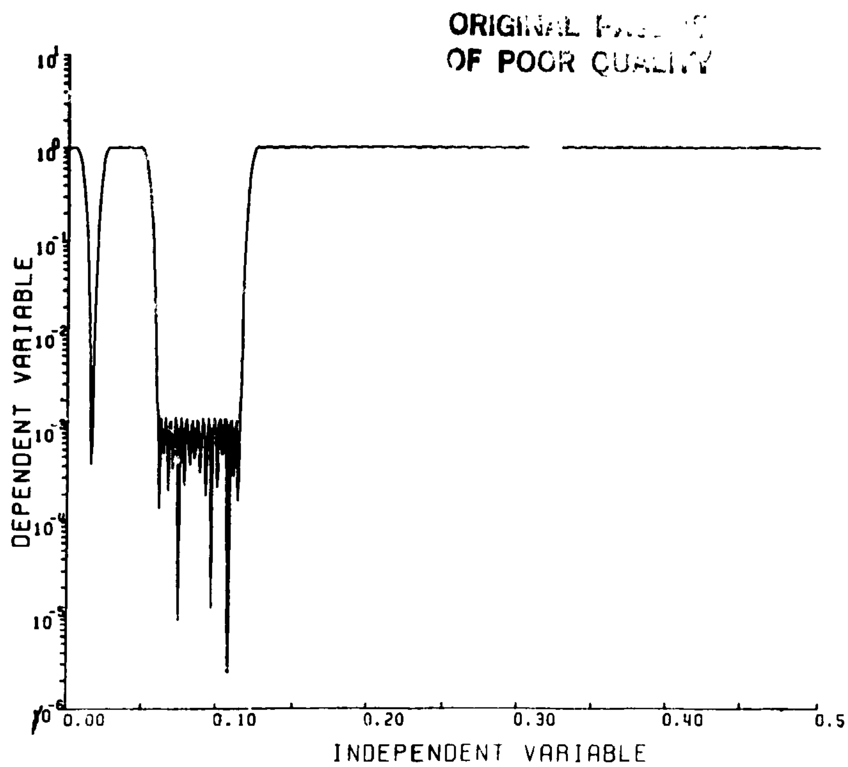


Fig. 5.2-10

The transfer function of the 5 band filter is shown. This transfer function is obtained by using operator FFT on data set WINDOW which contains the filter coefficients.

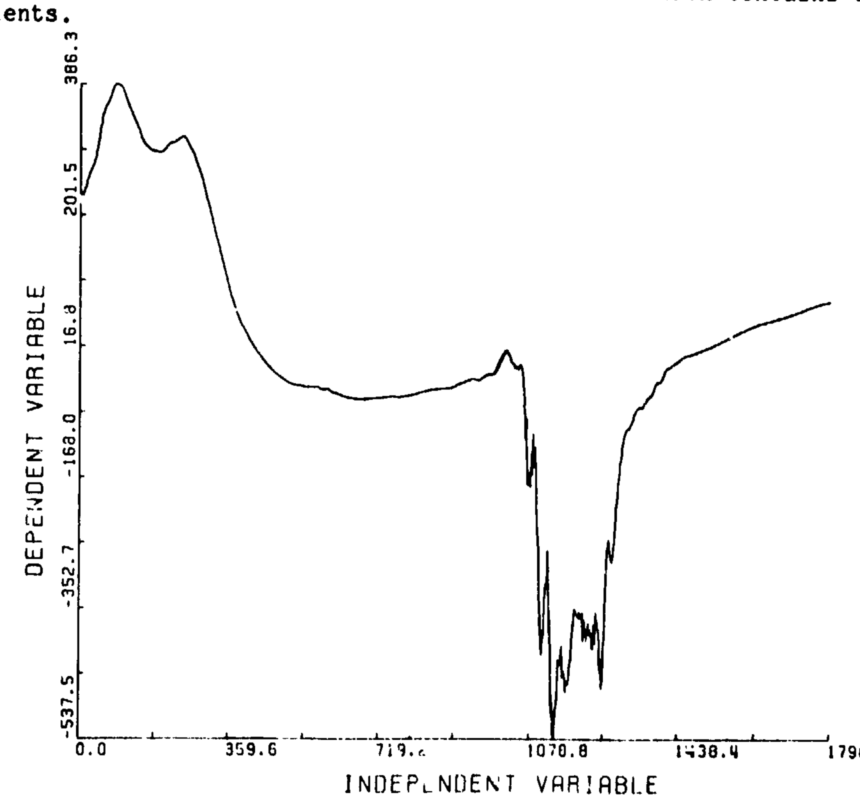


Fig. 5.2-11

Shows the results of using operator FILTER to filter the original time series shown in Fig. 5.2-2 with the 5 band filter shown in Fig. 5.2-10.

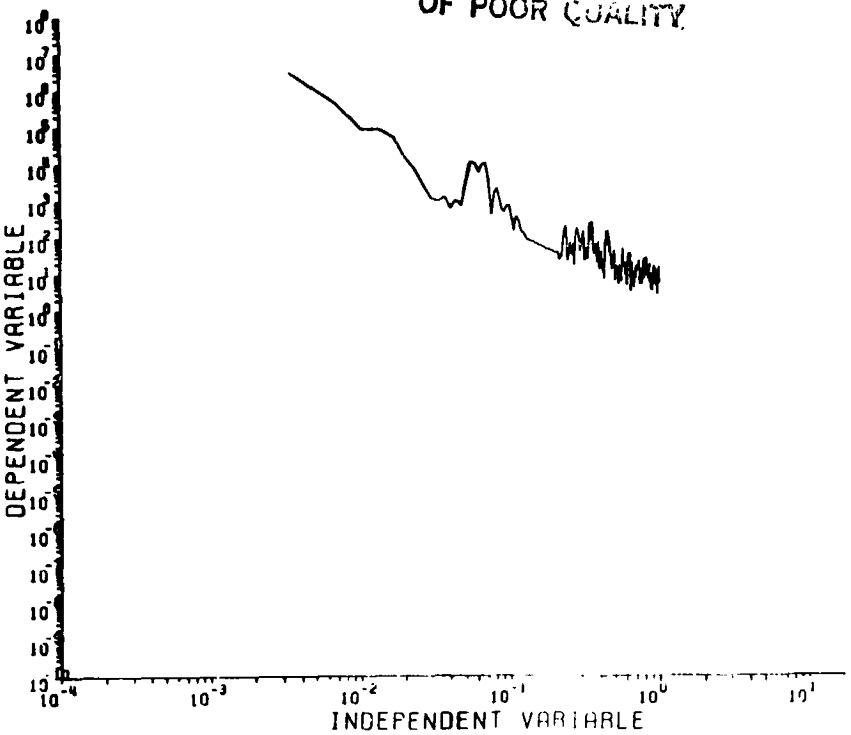


Fig. 5.2-12

Shows the power spectrum of the data after being filtered with the 5-band filter, note that the peaks in the power spectrum at 0.0324 Hz and 0.175 Hz have been removed.

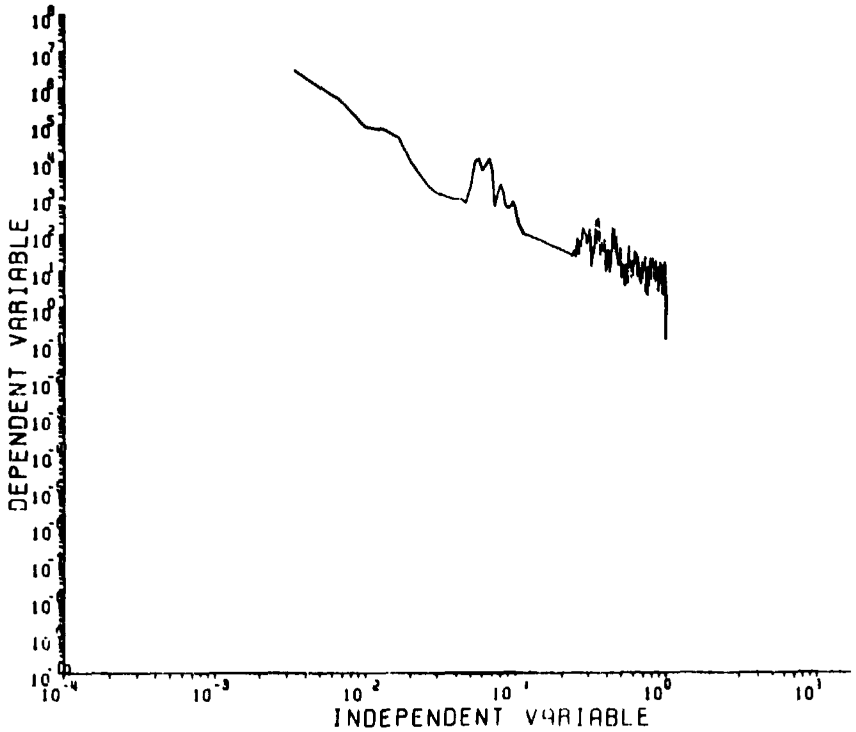


Fig. 5.2-13

Shows the power spectrum of the data after being filtered with the two 3-band filters, note that the peaks in the power spectrum at 0.0324 Hz and 0.175 Hz have been removed.

Fig. 5.2-14

Shows the tangential (T) component of a magnetic field vector plotted as a function of time.

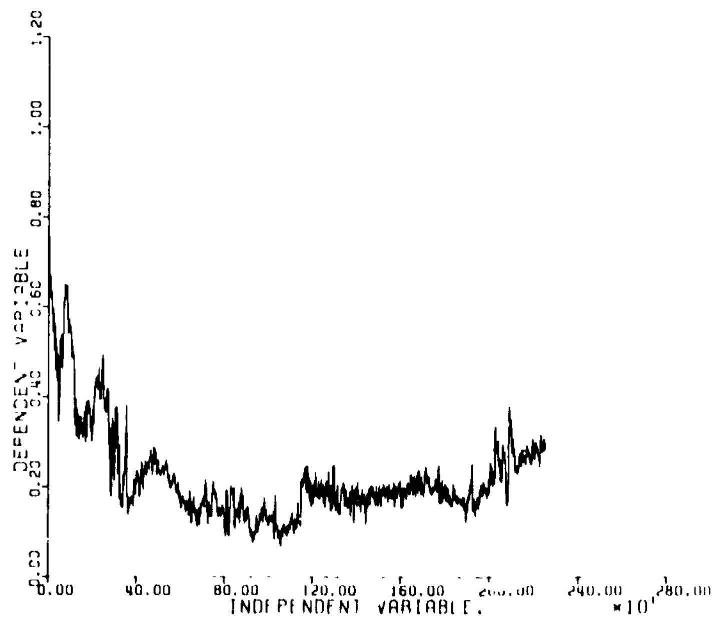


Fig. 5.2-15

Shows the normal (N) component of a magnetic field vector plotted as a function of time.

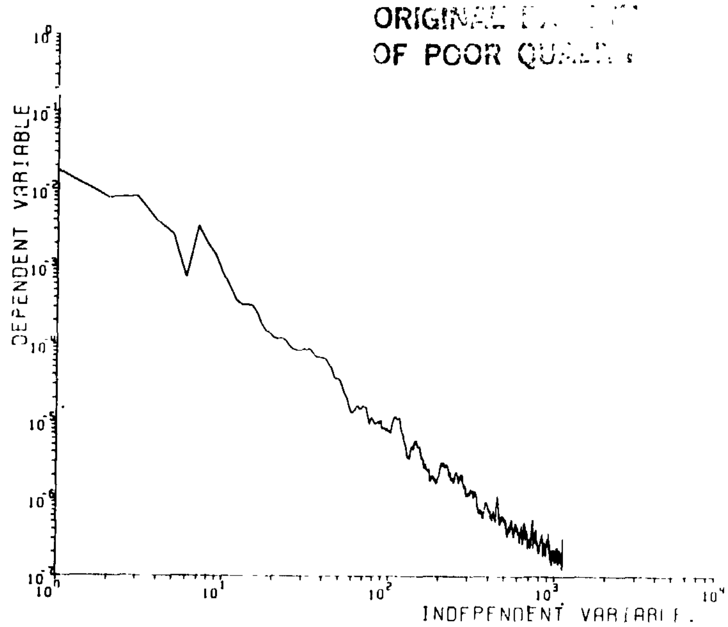


Fig. 5.2-16

Shows the power spectrum of the magnetic field vector plotted in Fig. 5.2-14 and 5.2-15.

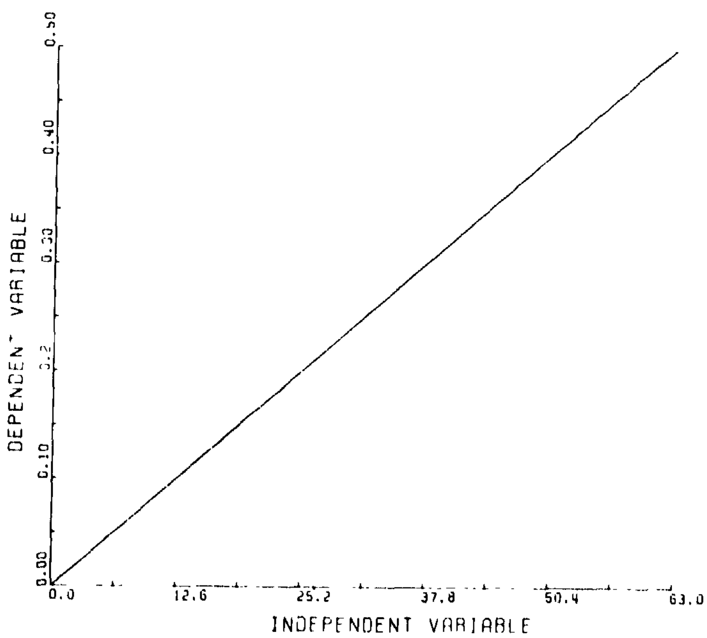


Fig. 5.2-17

Shows the transfer function of the differentiating filter.

ORIGINAL ... OF POOR ...

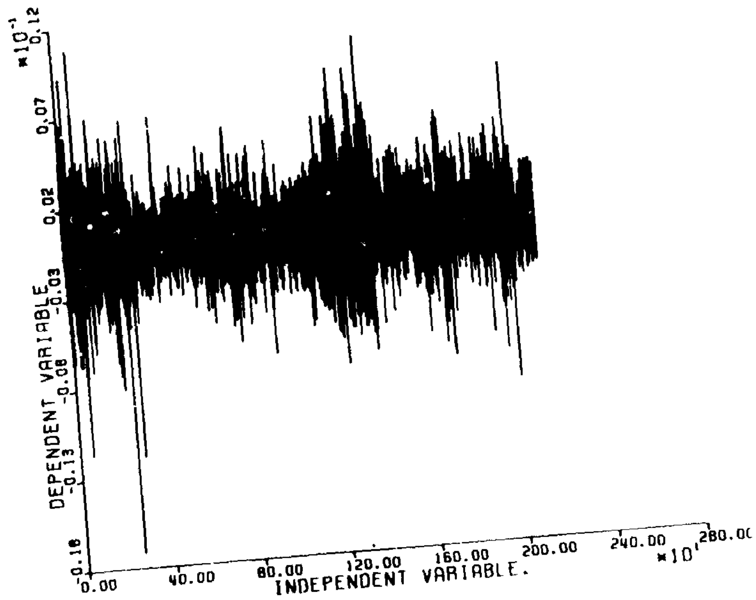


Fig. 5.2-18

Filtered version of T originally plotted in Fig. 5.2-14.

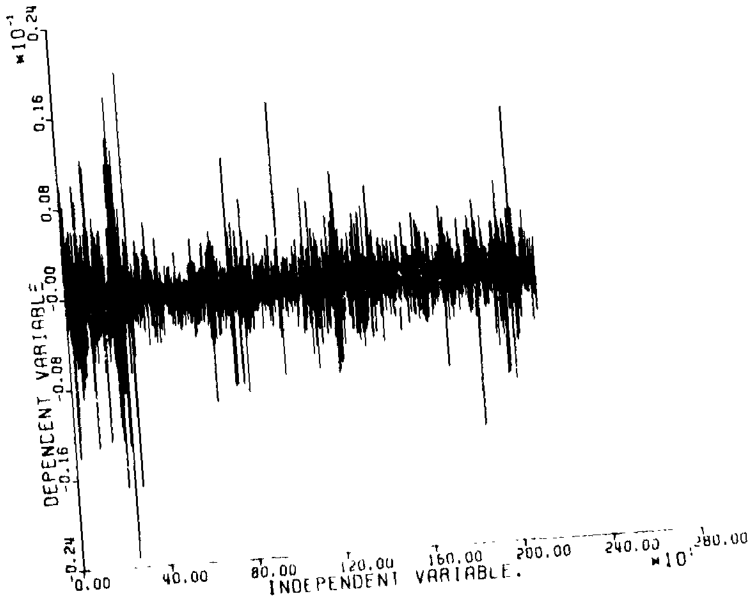


Fig. 5.2-19

Filtered version of N originally plotted in Fig. 5.2-15.

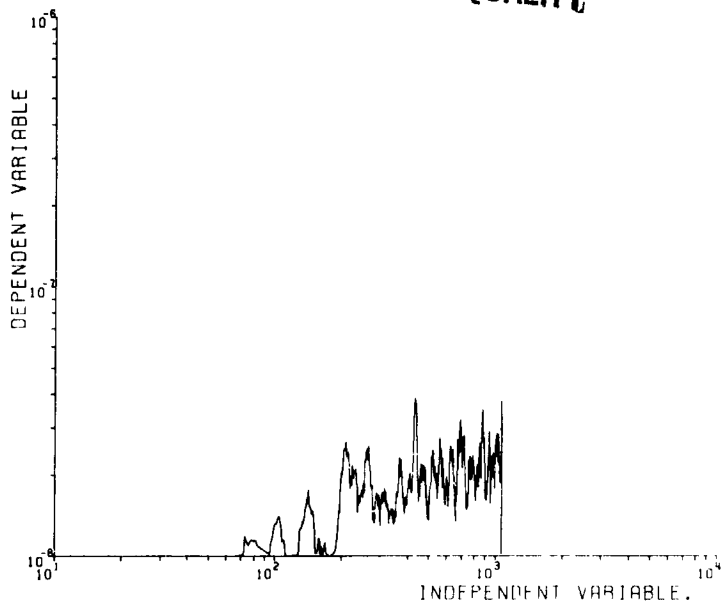


Fig. 5.2-20

The low frequency features are visibly absent and the data appears to be much more like white noise. This is borne out by the power spectrum shown in this plot which is flat at large F.

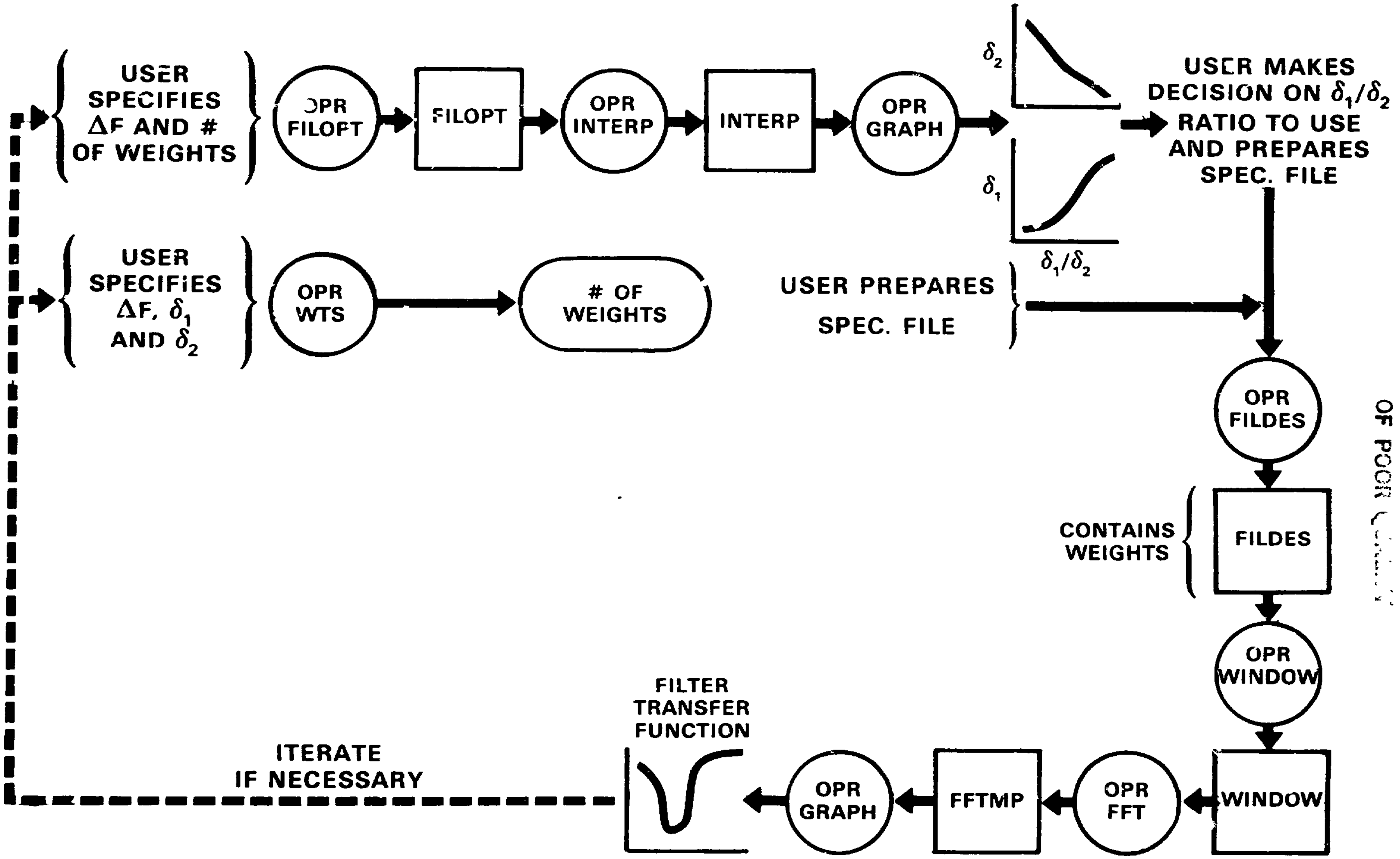


Fig. 5.2-21

Depicts the series of operations and options involved in the optimal design of a multiple band filter.

5.3) THE EFFECT OF WINDOWING, ZERO PADDING, AND OBSERVATION TIME ON THE ESTIMATION OF SPECTRA OF SINUSOIDS (OPERATORS SETUP, WINDOW, FFT, AND SPECT)

There are two fundamental issues in estimating spectra (see operators WINDOW, FFT, AND SPECT) of sinusoids:

Resolution- How close in frequency can multiple sinusoids be spaced and still be uniquely identified in the spectrum.

Dynamic Range- How large can the differences in amplitudes of the sinusoids be and still be uniquely identified in the spectrum.

The choice of the weighting function (window, see operator WINDOW) determines dynamic range and observation time (the length of the data record, see operator SETUP) determines resolution.

The procedure is as follows:

First we introduce the concept of normalized frequency which is defined as: NORMALIZED FREQUENCY=(ACTUAL FREQUENCY)/(SAMPLE RATE), which has units of cycles per sample. So the Nyquist frequency will always be 0.5 in normalized Hz.

a.) Select weighting function (window) based on the desired dynamic range. See Table 5.3-1 below. The main lobe widths in Table 5.3-1 are given in normalized Hz. Reference 2 makes a strong case for using the 6 dB point as a performance indicator rather than the traditional 3 dB point. In other words if N=100 the main lobe width of the rectangular window is 0.0121 normalized Hz. Note that all windows have a main lobe width proportional to 1/N normalized Hz.

b.) Select the number of data points, N, based on the desired resolution.

TABLE 5.3-1

WINDOW	PEAK SIDELOBE (dB)	MAINLOBE WIDTH (normalized Hz) (6 dB point)
RECTANGULAR	-13	1.21/N
HANNING	-32	2.00/N
HAMMING	-43	1.81/N

The following discussion explains how to go about this process:

DYNAMIC RANGE (WINDOWING)

An infinitely long pure sinusoid has as its Fourier transform a discrete line in the frequency domain. In practice, of course, we have a finite length data record of which we wish to take the Fourier transform. In effect we see the signal through a finite "window". If you do nothing but simply take N samples of a signal, you are windowing the signal by uniform (rectangular) weights. In the frequency domain the spectrum of this data record is the spectrum of this rectangular weighting function shifted by the sinusodial frequency. In the time domain you multiply the pure sinusoid by a rectangular window; in the frequency domain this becomes a convolution.

For the "do nothing" (rectangular window) case the Fourier transform of this window is plotted in Fig. 5.3-1 for a sinusoid with amplitude 50 and frequency 20 Hz with one unit of noise added and a observation time of 0.5 sec.

Note the 6 dB width of the main lobe is 1.21/N in normalized Hz. Therefore, if you do nothing but take N samples of a sinusoid and Fourier transform it you will get the function in Fig. 5.3-1 centered at the frequency of the sinusoid, which in this case is 20 Hz (0.2 normalized Hz).

As seen in Fig. 5.3-1 there is considerable energy in the side lobes. To reduce the size of the side lobes a different weighting function (window) whose side lobes are lower is used. However, the price paid for lower side lobes is an increase in the width of the main lobe. Using Table 5.3-1 and knowing the dynamic range of the signals of interest allows us to select the proper "window".

As an example suppose we are trying to resolve two sine waves with a ratio in amplitude of 1/20, further suppose in the frequency domain that the separation in frequency is such that the smaller amplitude sine wave falls at the peak of the first side lobe of the higher amplitude sine wave. From Table 5.3-1, if we use a rectangular window, the side lobe will "mask" the smaller amplitude sine wave. Since a factor of 20 in amplitude is $20\log_{10}(1/20)=-26$ dB, the peak side lobe for the rectangular window at -13 dB will thus mask the smaller amplitude; however, the "hanning" window is at -32 dB and thus the smaller signal should be visible. We have improved our dynamic range by a change of window! Note, however, that the 6 dB main lobe which was 1.21/N normalized Hz wide in the case of the rectangular window (see Table 5.3-1) has now widened to 2.00/N normalized Hz with an attendent loss of resolvability.

RESOLUTION

Suppose there are two sinusoids present in the data record. In the frequency domain, after taking the Fourier transform (see operators FFT and SPECT), we will have one or two peaks depending on how close in frequency they are relative to the main lobe of the particular window that was used. As the frequencies get closer together, the main lobes go from a bimodal to unimodal curve. A commonly accepted criterion for resolution is that 1/2 of the main lobe width separation is needed to resolve two sinusoids. Using Table 5.3-1 we can determine the main lobe width for the particular window used.

EXAMPLE

As an example of the complete process of selecting dynamic range and resolution, consider a signal sampled at 100 times per second, i.e., sample rate =100. Suppose two unknown frequencies are present in the data separated by at least 2 Hz, and we know that the amplitude of one frequency can be no greater than 50 times the amplitude of the other frequency. For this example we have used a time series with two frequencies (20.5 Hz and 18 Hz or 0.205 and 0.18 normalized Hz) with amplitudes 50 and 1 units, respectively. We now need to answer the following questions:

a.) What is the dynamic range of the signals, thus what window should be employed?

b.) What is the desired resolution, thus determining the length of the data record, i.e, the observation time?

The answer to question (a) is that the dynamic range is $20\log_{10}(1/50)=-34$ dB; thus we need to use a Hamming window with a peak side lobe is down at -43 dB. Fig. 5.3-2 shows the Fourier transform of such a time series which has been transformed using a "hanning" window. Note the transform does not show the presence of the lower amplitude frequency. However, Fig. 5.3-3 is the same time series transformed using a "Hamming" window which clearly shows the lower amplitude frequency. Both Fig. 5.3-2 and 5.3-3 show the widening of the main lobes as the result of using the hanning and Hamming window.

The answer to question (b) is that the resolution necessary is 2.5 parts in 100 or 0.025. Using the criterion that 1/2 of the main lobe width separation is needed to resolve two sinusoids, from Table 5.3-1 the main lobe width of the Hamming window is 1.81/N normalized Hz. So 1/2(1.81/N)=0.025, and, solving for N, we get 36 data points or a 0.36 sec observation time. Note in Fig. 5.3-4 with an observation time of 0.25 seconds we are not able to resolve the lower amplitude frequency even though a Hamming window was used. Whereas in Fig 5.3-3 with a 0.5 sec observation time the lower signal is still seen.

What this amounts to in practical terms is that if there are several features "close" together we must have a "longer" observation time or the convolving windows will not let us distingish them from each other. The "closer" the two features are to each other the "longer" the observation time required to resolve them. The resolution limit using Fourier techniques and a rectangular window is $1/T_0$ where T_0 represents the observation time.

<u>ZERO PADDING</u> (Zero padding is an option in Operator WINDOW)

The purpose of zero padding is to better define, for purposes of appearance and accuracy, the plotting of the transform. Zero padding, as the name implies, is adding additional zeros to the time series before the Fourier transform is computed so that the transform will be evaluated at a larger number of normalized frequencies than it would be if only the original N data points were used.

When zero padding and using a rectangular window, one has to exercise caution because padding may introduce a discontinuity into the time series, i.e., the series can abruptly go to zero at the first padded point (see Fig. 5.3-5), this in turn will introduce additional unwanted spectral contributions (leakage). It should be noted that the Fourier transform implicitly assumes that the time domain function observed during T_0 is repeated with period T_0 outside of the observation limits. Consequently when zero padding we have modified the original time domain function and consquently would expect this modified function to have a different transform and no longer approximate the original continuous Fourier transform.

In Fig. 5.3-6 we show the transform of a 10 Hz time series, using a rectangular window, without zero padding and, in Fig. 5.3-7, the transform after padding; notice the much higher leakage levels introduced by the discontinuity due to the padding. Recall that windows are applied to reduce the order of the discontinuity at the boundary of the periodic extension, so in the case of a time series windowed with other than a rectangular window, zero padding will not introduce a discontinuity because the window tapers the time series to zero prior to padding. This tapering can be seen in Fig. 5.3-8 where we show the time series of a 10 Hz signal after application of a Hamming window. Fig. 5.3-9 and 5.3-10 show the transform of a 10 Hz time series after application of a Hamming window without zero padding and with zero padding, respectively. Note that there is only a small increase in the leakage introduced by the padding.

Note that zero padding does not improve resolution since only an increase in observation time (T_0) will do that. But zero padding does result in additional Fourier coefficients being computed which better defines the sampled continuous Fourier transform.

References

1) Hamming R. W., Digital Filters, Prentice Hall, Inc. Englewood Cliff, NJ 07632, 1977.

2) Harris F. J., On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform, Proceeding of the IEEE, Vol 66 No 1, Jan. 1978.

3) Schaff, W. E., Course Notes: Modern Methods of Digital Signal Processing, Integrated Computer Systems, Inc, October 1979.

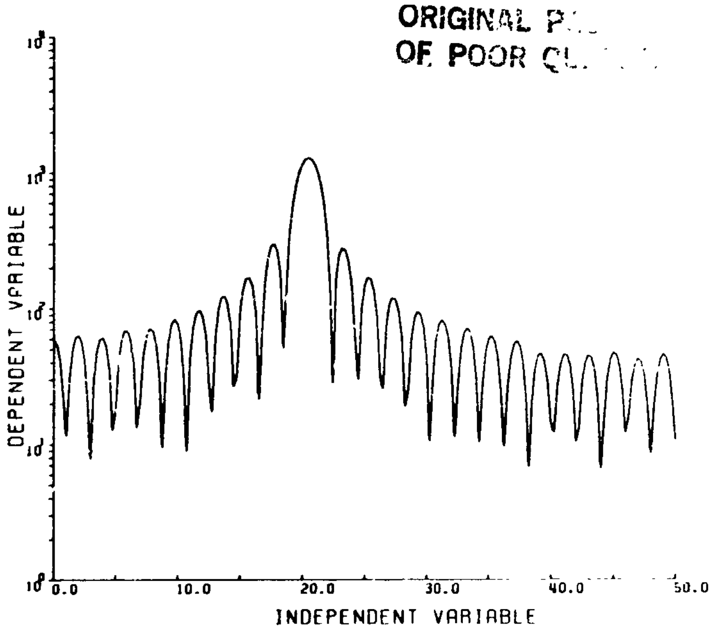


Fig 5.3-1

Transform of a sinusoid of amplitude 50 units, and frequency 20 Hz with one unit of random noise added using a rectangular ("do nothing") window. Observation time of 0.5 sec.

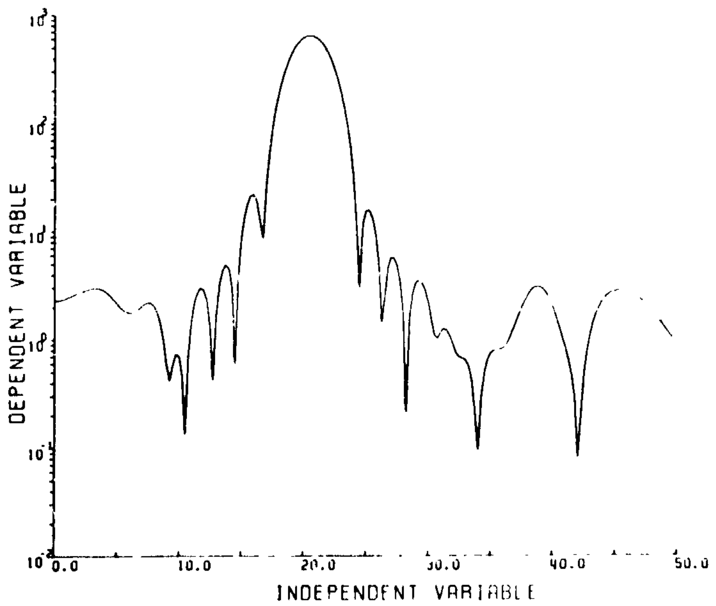


Fig. 5.3-2

Transform of a time series containing two frequencies (20.5 and 18 Hz) with amplitudes 50 and 1 units, respectively, and one unit of random noise using a hanning window. Observation time of 0.5 sec. Demonstrates the inability of the hanning window to resolve the two frequencies present. Note the width of the main lobe has increased relative to the main lobe width of the rectangular window.

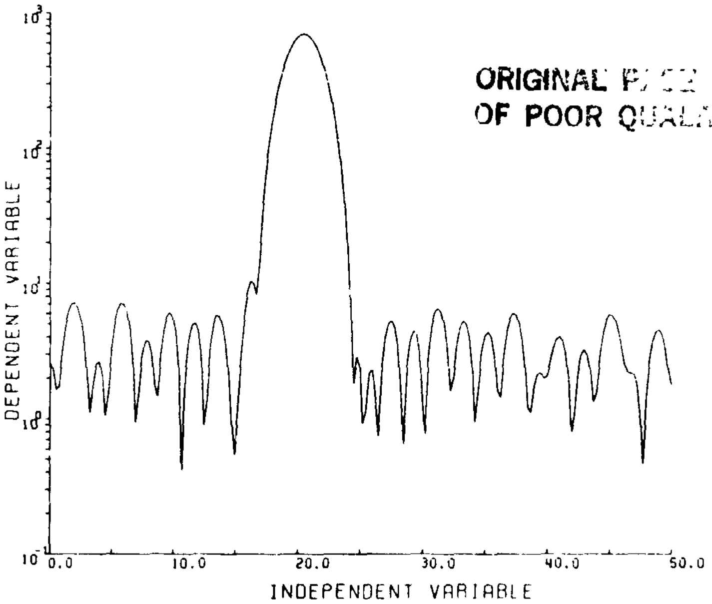


Fig. 5.3-3

Transform of a time series containing two frequencies (20.5 and 18 Hz) with amplitudes 50 and 1 units, respectively, and one unit of random noise using a Hamming window. Observation time of 0.5 sec. Demonstrates the ability of the Hamming window to resolve the two discrete frequencies.

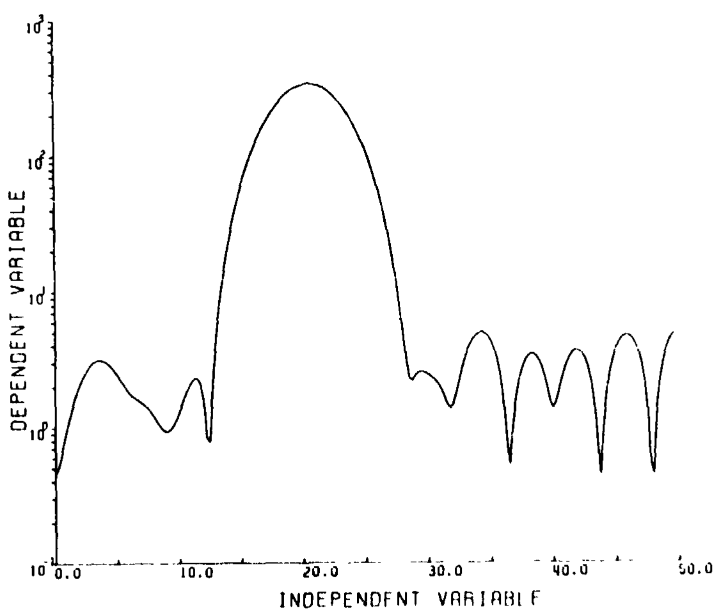


Fig. 5.3-4

Transform of a time series containing two frequencies (20.5 and 18 Hz) with amplitudes 50 and 1 units respectively and one unit of random noise using a Hamming window. Observation time of 0.25 sec. Demonstrates the inability, even using the Hamming window, to resolve the two frequencies present due to inadequate observation time.

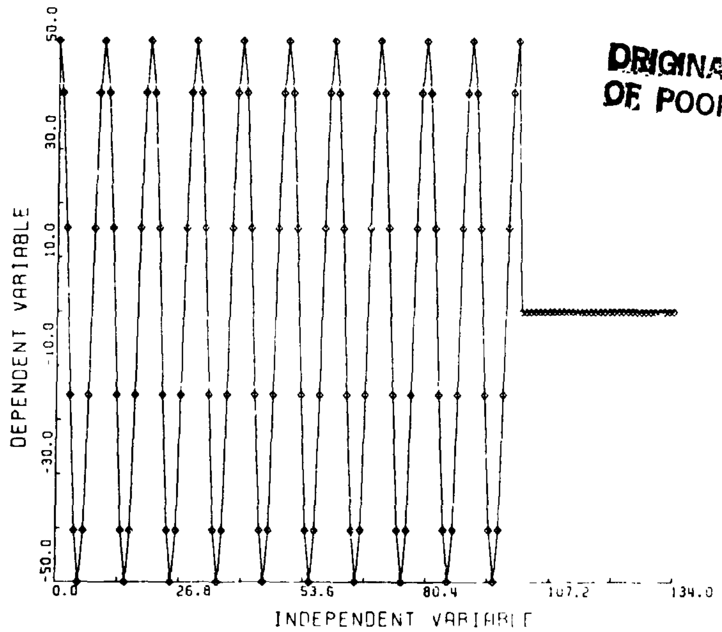


Fig. 5.3-5

Plot of a 10 Hz time series showing the discontinuity introduced by zero padding. The Fourier transform implicitly assumes that the time domain function observed during T_0 is repeated with period T_0 outside of these limits. Consquently zero padding will modify the original function and result in a different transform.

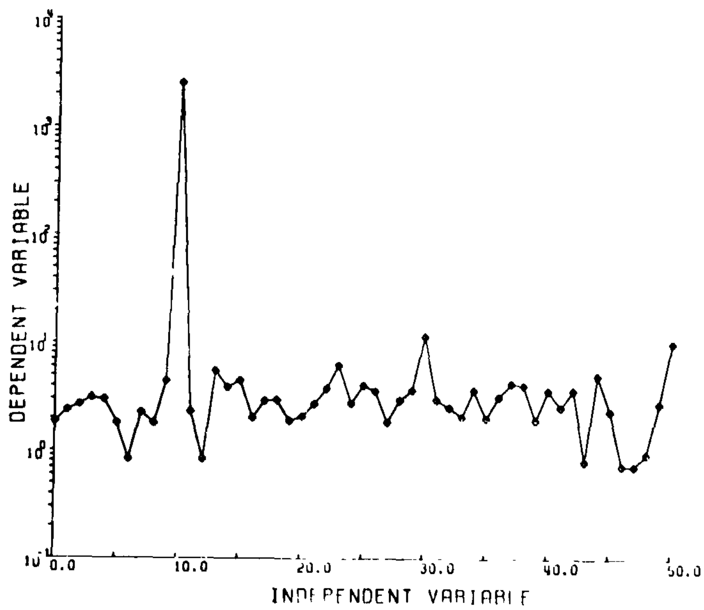


Fig. 5.3-6

Transform of a 10 Hz time series containing 1 unit of noise with no zero padding using a rectangular window.

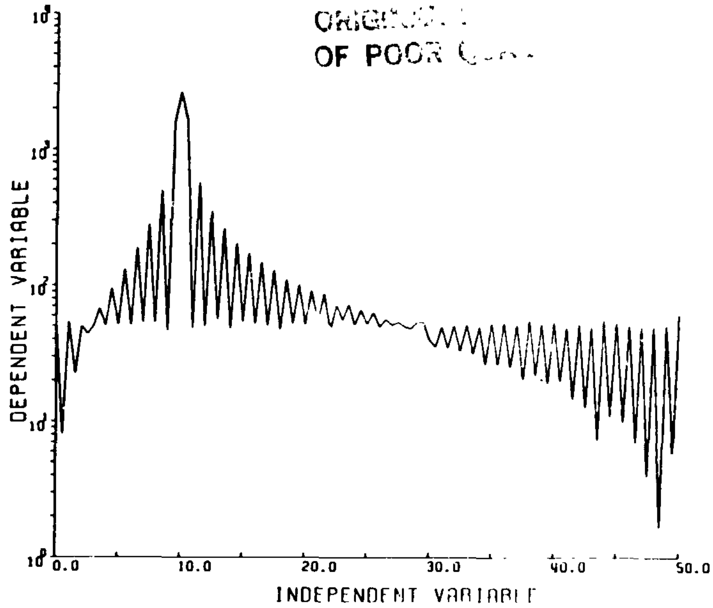


Fig. 5.3-7

Transform of a 10 Hz time series containing 1 unit of noise zero padded to 200 points using a rectangular window. Note the increase in leakage due to zero padding.

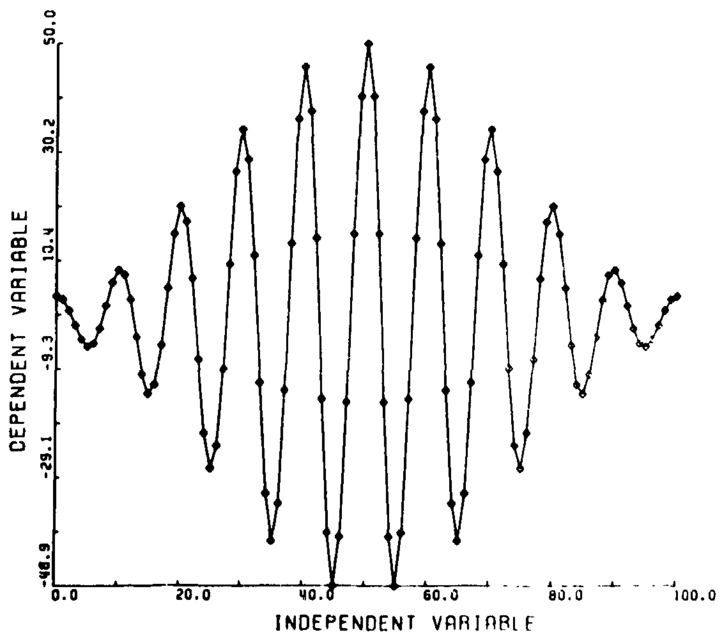


Fig. 5.3-8

Plot of a 10 Hz time series after being windowed by a Hamming window, note how the ends are tapered to zero.

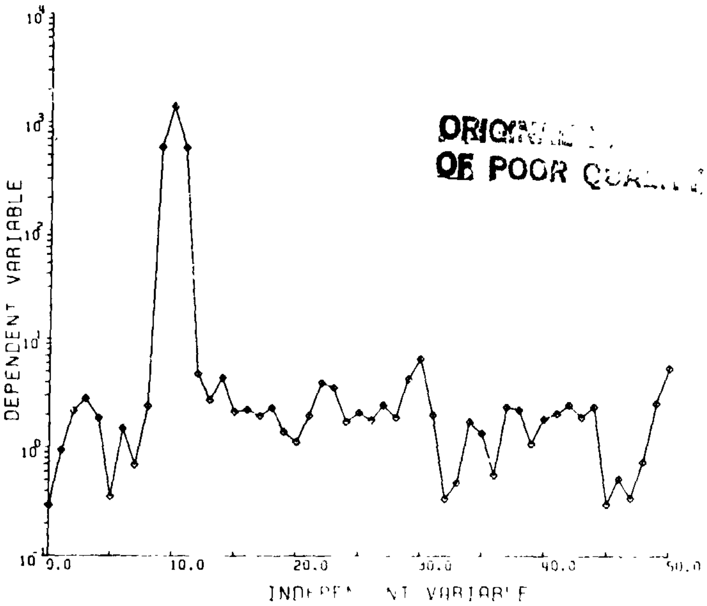


Fig. 5.3-9

Transform of a 10 Hz time series containing 1 unit of noise using a Hamming window and no zero padding.

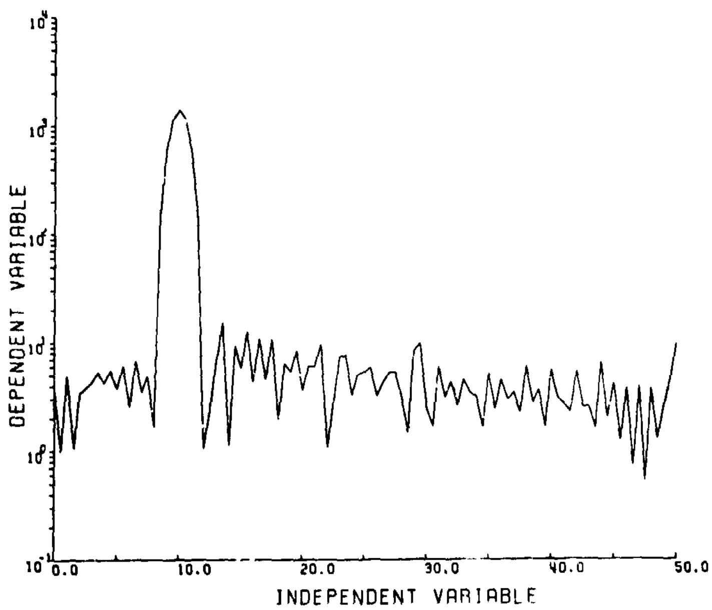


Fig. 5.3-10

Transform of a 10 Hz time series containing 1 unit of noise using a Hamming window zero padded out to 200 points, note that the leakage level is about the same as shown in Fig 5.3-9.

6.0 IDSP OPERATOR DESCRIPTIONS

Usage of AVER Command

The user specifies 'AVER dsname n' where n must be the number of points which he wants averaged into each box. The last box may not be full depending on how n divides the number of available points. If the user specifies optional parameter ZERO, the first box in the average contains only one point. (This option can be used when one wants to avoid having the zero frequency averaged in with anything.) The following example utilizes this option, with the rest of the points of dataset SPECTOFF averaged into 8 points per average:

AVER SPECTOFF 8 ZERO

Usage of CMDHIS Command

This command causes a printout of user responses to the system query "ENTER COMMAND" that were entered during the current analysis session. It has no operands. The file that contains these commands is FOR022.DAT and can be printed after the session ends.

Usage of CONCAT Command

If two datasets have the same number of points (NPTS), if both are of the same type (real or complex), and if the sum of their series is less than or equal to eight, they can be concatenated into one dataset of NPTS points and (NSER1 + NSER2) series. The form of the command is:

```
CONCAT data1 data2
```

For inverse operator, see SUBSER.

Usage of the COPY Command

The IDSP system destroys datasets in its default working directory [usrid.IDSP] by overwriting the old dataset when a given operation is repeated. After any operation, any dataset can be copied into a file of some other name. The first example copies the current version of WINDOW into WIND6 in the working directory; the second copies it into a different directory file [usrid.GAMMA]WIND7.DAT.

```
COPY WINDOW WIND6

COPY WINDOW [usrid.GAMMA]WIND7
```

Usage of the DCL Command

This operator allows the user to temporarily exit the IDSP process and execute VAX Command Language Commands via a subprocess. He will receive the standard "$" prompt. When finished the user should type: "STOP/ID=0" or "LO" This will return control to IDSP by terminating the subprocess. NOTE: Typing "L[illegible] produces a message indicating the subprocess was logged off. It is no [illegible] for alarm. The user is now back in IDSP.

WARNING: Do not do something foolish in DCL like Control Y, or a delete, [illegible] a purge. When in DCL your LOGIN constraints, such as "decide" for delete o[illegible] KEEP=2 for purge are <u>not</u> recognized.

Usage of DSTAT Command

To obtain statistics of a dataset, type "DSTAT dataset- name". Any data flagged as BAD (see SETUP) will not be included in the statistical calculation; the percent of bad data per series will be calculated. Statistics calculated for each series:

0	XM	Mean	$\Sigma X_i/n$
1	XMABS	Mean of absolute values	$\Sigma \lvert X_i \rvert/n$
2	XMSQ	Mean of values squared	$\Sigma X_i^2/n$
3	XRMS	Root Mean Square	SQRT(XMSQ)
4	VAR	Unbiased variance (division by n-1)	$\Sigma(X_i-XM)^2/(n-1)$
5	NBAD	Number of bad data per series	
6	PER	Percent bad data per series	

Usage of the EDHIST Command

The EDHIST operator allows the user to edit the history elements in the header of a specified dataset. The form of the command is: EDHIST dsname where "dsname" is the name of the input dataset.

This operator has several editing commands listed below:

D: Delete the current line.
R: Replace the current line.
I: Insert a line before the current line.
<cr>: Save the current line.
E: End of editing session.

Usage of EDIT Command

The interactive user at a HP-2648A Graphics Terminal can dynamically flag data points as BAD (set equal to -99.9) in any real dataset. The EDIT operator causes a display of up to 500 points to the screen and places the user into interactive usage of interactive graphics commands. (To see a list of these commands at this stage, type "HELP".) The only commands that have any effect on the dataset stored by EDIT are those that perform deletions—the EDIT operator will flag them as BAD as soon as the user types "EXIT" to return to the EDIT operator. If the series had more than 500 points, the next group of points will then be displayed. As soon as editing is finished on one series, EDIT moves to the next one. The form of the command is:

EDIT dataset SER=j ALL

where "SER=j" is optional if the user wants only to edit the jth series. "ALL" is optional and specifies that if the ith point is deleted in any series, then it is deleted in all the series.

More extensive documentation on the HP interactive graphics commands can be found in Carleton, T. P., LEP VAX Users Guide, NASA TM 84931, October 1982.

Usage of the EIG Command

This operator takes as input spectral matrices and does rotation to an eigenvalue coordinate system (see Section 5.1 for details). Statistics for some of the computations are extracted from a dataset created previously by operator DSTAT, operating on a time series data set such as the output of INTERP or FILTER, i.e., a pre-FFT data set. The form of the command is:

EIG SPECTD SPECTOFF DSTAT MF KCON

where "SPECTD" is the dataset containing the diagonal terms of the spectral matrix; "SPECTOFF" contains the corresponding off-diagonal terms; and "DSTAT" contains the dataset of series statistics. "MF" is optional and is used only if the user wants the analysis to assume that the original series have been placed into a Mean-Field coordinate system (see operator MNFLD). The diagonals of the rotated system are stored in EIGD, the off-diagonals are stored in EIGOFF, and various scaler parameters are stored in EIGPARM, and the eigenvectors and eigenvalues associated with the minimum, intermediate and maximum directions are stored in EIGVEC and EIGVEC1.

KCON is an optional parameter of EIG which can be used to select a sign convention for the wave normal vector $\hat{k}$ determined in the polarization analysis (but with arbitrary sign). KCON can have the values 1, -1 and 0, with KCON = 1 the default state. In this case, the component k_1 is forced to be positive by inversion of $\hat{k}$ (multiplying all components by -1) if found to be negative. For coordinate systems centered at the sun, that will force $\hat{k}$ to have a component outward (along the radius vector direction) from the sun. For data coordinate systems defined with X positive toward the sun, the user should choose KCON = -1; this forces k_1 to be in the negative X direction by inversion of $\hat{k}$, again insuring that $\hat{k}$ have a component outward from the sun. If KCON = 0 is chosen, a different convention, based on the direction of average $\vec{B}$, is applied. In this case the quantity BDOTK will be tested; if BDOTK < 0, then $\hat{k}$ is inverted and BDOTK recomputed. This forces $\hat{k}$ to have a component in the $+\vec{B}$ direction.

Data set EIGPARM contains scaler quantities DEGPOL, COH, ELIP, THETA, TRTOT, SNR, BDOTK, and SIGMA. EIG also creates data set EIGVEC which contains in series 1-3, the first three components of the eigenvector corresponding to the direction of minimum variance for that mode. Series 4-6 are the eigenvalues (minimum, intermediate and maximum, respectively). Also created by operator EIG is data set EIGVEC1 which contains the three components of the eigenvector corresponding to the intermediate direction of variance (series 1-3) and the three eigenvectors corresponding to the maximum direction of variation (series 4-6).

WARNING: If the "MF" option has been specified, the normalized magnetic helicity (SIGMA) in EIGPARM will not be correct.

For a schematic of operator EIG and its relationships to operators DSTAT and SPECT and the associated data sets see Fig. 5.1-8.

Usage of FFT Command

The IMSL routine FFTRC is used to compute the Fourier Transform, X, of every series, A, in the dataset according to the following formula:

X(K+1) = SUM FROM J = 0 TO N-1

OF A(J+1)*CEXP((0.0,(2.0*PI*J*K)/N)) FOR K=0,1,...,N/2 AND PI=3.1415...

Only N/2+1 points are stored because of symmetry. (If the number of points in each input series was odd, it is forced to be even by deleting the final point of each series before taking the transform.) The default normalization factor is 1. The user can choose his own normalization by specifying the optional parameter: NORM=x where x is the value he desires. The first of the following examples has default normalization; the second has normalization .000005.

FFT INTERP

FFT WINDOW NORM=5.E-6

WARNING: If N/2*2 has a large prime factor, the "FFT" will be VERY slow.

If the user wishes to obtain the magnitude and phase of the FFT, he can specify the optional parameter "MP" which will also produce a complex dataset FFTMP where the magnitude is stored as the real part, and the phase in degrees is stored as the imaginary part.

For inverse transform, see FFTIN.

LIMITATION

Because of the size of working arrays required, operations involving the FFT are limited to a much smaller size than other operations (such as SETUP). For series involving more than 10,000 points, the user should check documentation under "DESIGN CONSIDERATIONS" and also be aware that the time for such long runs may be excessive.

Usage of FFTIN Command

The IMSL routine FFTCC is used to compute the inverse Fourier Transform, X, of every series, A, in the dataset, each containing NPTS. First, a working array is constructed containing the complete transform, using the conjugate symmetry of the data:

```
N = (NPTS-1)*2

B(N+2-I) = CONJG(A(I)),     FOR I=2,...,N/2
```

Then X is computed according the the following formula:

```
X(K+1) = SUM FROM J = 0 TO N-1

OF B(J+1)*CEXP((0.0,(-2.0*PI*J*K)/N))   FOR K=0,1,...,N-1 AND PI=3.1415...
```

The default normalization factor is (1/N). The user can choose his own normalization by specifying the optional parameter: NORM=x where x is the value he desires. The first of the following examples has default normalization; the second has normalization .000005.

```
FFTIN SPECTOFF

FFTIN SPECTD NORM=5.E-6
```

For the forward transform, see FFT.

<u>LIMITATION</u>--see FFT

Usage of FILDES

For data having sample spacing T, the highest observable frequency (called the Nyquist frequency) is FN = 1/(2*T). To formulate filter design parameters, use a normalized frequency scale S = [0.0,0.5] where 0.5 corresponds to the Nyquist frequency (FN). Thus, given any frequency F that the user wishes to input as a filter design specification, he first scales it by the formula:

FS = 0.5 * (F / FN) or equivalently

FS = F * T

Hence, if the user wishes to enter a frequency parameter corresponding to a frequency of 20 (as in cos(2*PI*20*t)), and if his data spacing is T = 0.01, then FN = 50, and FS = 0.2.

To execute the command, type a sequence of the form "FILDES r DSN=name" where "r" is a required parameter specifying the source device (T = terminal, F = disk file) for the filter design specifications. When reading from a file, the user can specify the optional parameter DSN to equal the name of the particular file he wishes to access. When this parameter is omitted, the latest version of FOR058.DAT is always used. The following example executes the design according to the fourth version of the design spec file:

FILDES F DSN=LOWPASS1.DAT;4

The filter coefficients will be stored in a dataset called FILDES. The interactive user who desires to see the effect of several different filters should copy FILDES into some other dataset name (see COPY) before executing FILDES again. Then he can (as an option to the FILTER command) choose which filter he wants applied to his data.

Filter design by Remez exchange technique. Maximum filter length is 512. Input data records as shown below:

First record: NFILT,JTYPE,NBANDS,LGRID. (4I5 format)
Next record(s):EDGE (4F15.9 format)
Next record(s): FX (4F15.9 format)
Next record(s): WTX (4F15.9 format)

These records can be placed into [usrid.IDSP]FOR058.DAT if the user does not want to enter design specifications interactively.

NFILT-- FILTER LENGTH

JTYPE-- TYPE OF FILTER

1 = MULTIPLE PASSBAND/STOPBAND FILTER
2 = DIFFERENTIATOR
3 = HILBERT TRANSFORM FILTER

NBANDS-- NUMBER OF BANDS (LIMIT OF 10)

LGRID-- GRID DENSITY, WILL BE SET TO 16 UNLESS SPECIFIED OTHERWISE BY A POSITIVE CONSTANT.

EDGE(2*NBANDS)-- BANDEDGE ARRAY, LOWER AND UPPER EDGES FOR EACH BAND WITH A MAXIMUM OF 10 BANDS.

FX(NBANDS)-- DESIRED FUNCTION ARRAY (OR DESIRED SLOPE IF A DIFFERENTIATOR) FOR EACH BAND.

WTX(NBANDS)-- WEIGHT FUNCTION ARRAY IN EACH BAND. FOR A DIFFERENTIATOR, THE WEIGHT FUNCTION IS INVERSELY PROPORTIONAL TO F.

SAMPLE INPUT DATA SETUP:

```
32,1,3,0
0.0,0.1,0.2,0.35
0.425,0.5       (IN 4F15.9 FORMAT)
0.0,1.0,0.0
10.0,1.0,10.0
```

THIS DATA SPECIFIES A LENGTH 32 BANDPASS FILTER WITH STOPBANDS 0 TO 0.1 AND 0.425 TO 0.5, AND PASSBAND FROM 0.2 TO 0.35 WITH WEIGHTING OF 10 IN THE STOPBANDS AND 1 IN THE PASSBAND. THE GRID DENSITY DEFAULTS TO 16.

THE FOLLOWING INPUT DATA SPECIFIES A LENGTH 32 FULLBAND DIFFERENTIATOR WITH SLOPE 1 AND WEIGHTING OF 1/F. THE GRID DENSITY WILL BE SET TO 20.

```
32,2,1,20
0,0.5
1.0
1.0
```

If the user wishes to peruse the printout describing the filter he can route the output to disk via the FLOP command before executing the FILDES command.

LIMITATIONS: Avoid designing a differentiator with an ODD number of coefficients. At times the design algorithm complains about possible convergence problems. When designing a filter, ALWAYS MAKE A PLOT OF THE TRANSFER FUNCTION to ensure that the filter behaves as you desire. Section 5.1.5, and particularly Section 5.2.2, discuss a technique for the systematic determination of the stopband ripple and passband ripple using operator FILOPT. Number of bands is limited to 10 for bandpass/bandstop filter. The maximum number of coefficients is 512 for any type of filter.

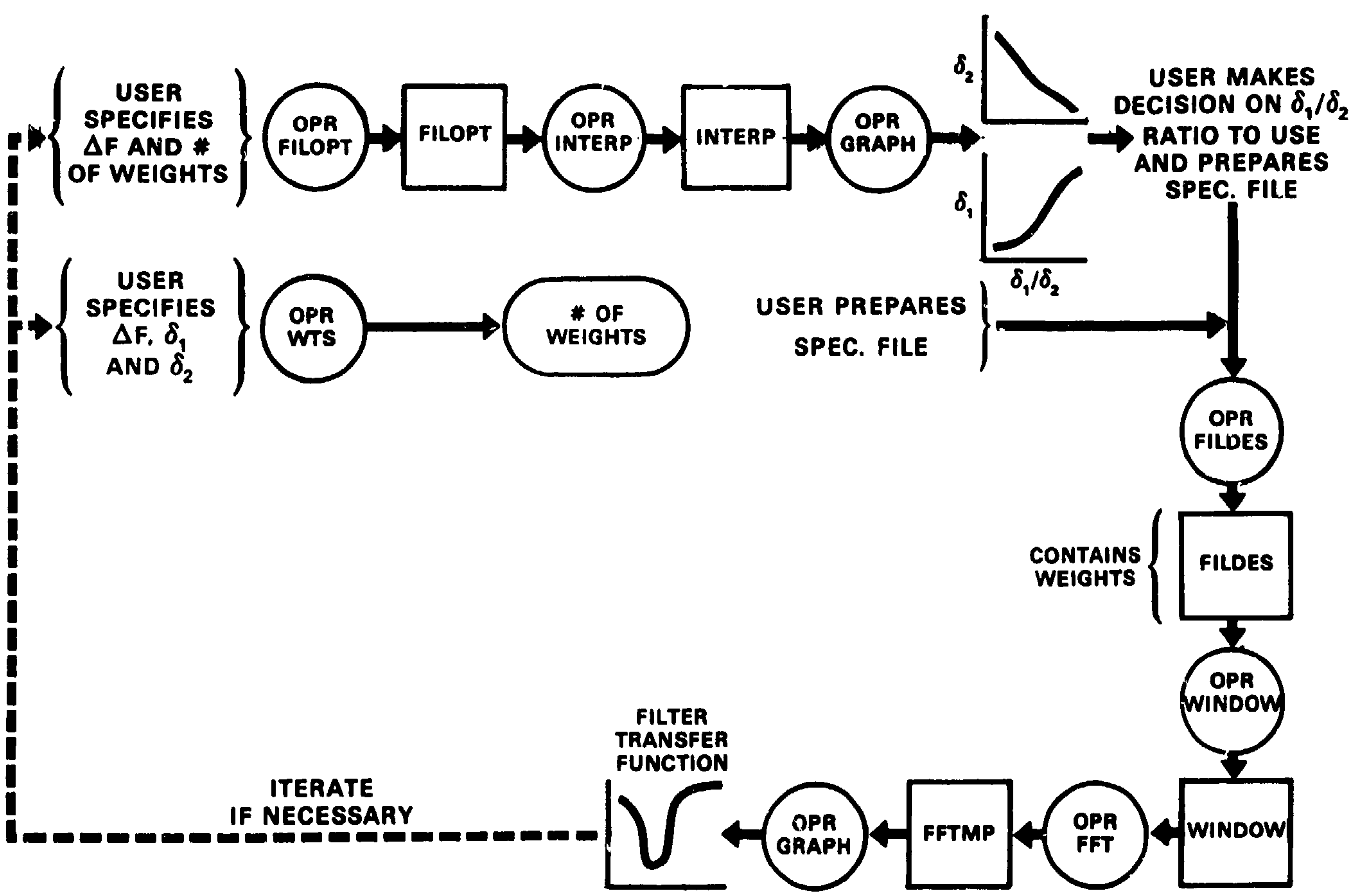
ORIGINAL PAGE IS
OF POOR QUALITY

To obtain a plot of the transfer function of the filter, window the filter coefficients with a rectangular window and pad with enough zeroes to isolate the function. Specify:

```
WINDOW FILDES 0 PAD=512

FFT WINDOW MP
```

This creates a complex dataset FFTMP where the magnitude is stored as the real part, the phase (in degrees) as the imaginary part. FFTMP can then be plotted using the GRAPH operator.

More information available in Chapter 5 of IEEE PROGRAMS FOR DIGITAL SIGNAL PROCESSING (NASA/GSFC Library: QA 76.5.P77). See also Rabiner and Gold, THEORY AND APPLICATIONS OF DIGITAL SIGNAL PROCESSING, Chapter 3.



Usage of FILOPT

This operator assists the user in determining the optimum ratio of $delta_1$ to $delta_2$ (See Fig. 5.2-1 for definitions of $delta_1$ and $delta_2$) to use in designing a filter with the Remez exchange algorithm.

To execute the command, type a sequence of the form "FILOPT r DSN=name" where "r" is a required parameter specifying the source device (T=terminal, F=disk file) for the filter design specifications. When reading from a file, the user can specify the optional parameter DSN to equal the name of the particular file he wishes to access. When this parameter is omitted, the latest version of FOR059.DAT is always used. The following example executes the algorithm according to the fifth version of the design specification file:

```
FILOPT F DSN=LOWPASS.DAT;5
```

Output is stored in a data set called FILOPT. THis file contains the following three series of information:

1 - pass band ripple in db - defined as 20*log(1+delta1)
2 - stop band ripple in db - defined as 20*log($delta_2$)
3 - lowest limit of the stop band

The values are calculated nine times per decade for a maximum of five decades. The resolution of the output file is equal to the value of the smallest decade. All intermediate points are written as fill data (-99.9) to allow for subsequent interpolation in order to meet the equidistance spacing requirement of IDSP compatible data sets.

Each filter is designed by the Remez exchange techique. Maximum filter length is 512. Input data records as shown below:

```
First record:   NFILT,NBANDS,LGRID
Next record(s): EDGE (4F15.9 format)
Next record(s): FX (4f15.9 format)
Last record:    IDEC,NDEC
```

These records can be placed into [usrid.idsp]FOR059.DAT if the user does not want to enter design specifications interactively.

NFILT - filter length
NBANDS - number of bands
LGRID - grid density, will be set to 16 unless specified otherwise by a positive constant less than 16

EDGE(2*nbands) - bandedge array, lower and upper edges for each band with a maximum of 10 bands

FX(nbands) - desired function array for each band

IDEC - starting decade for calculation (power of 10)
NDEC - number of decades (maximum of five)

ORIGINAL PAGE IS
OF POOR QUALITY

SAMPLE INPUT DATA SETUP:

```
32,3,0
0.0,0.1,0.2,0.35
0.425,0.5
0.0,1.0,0.0
-1,4
```

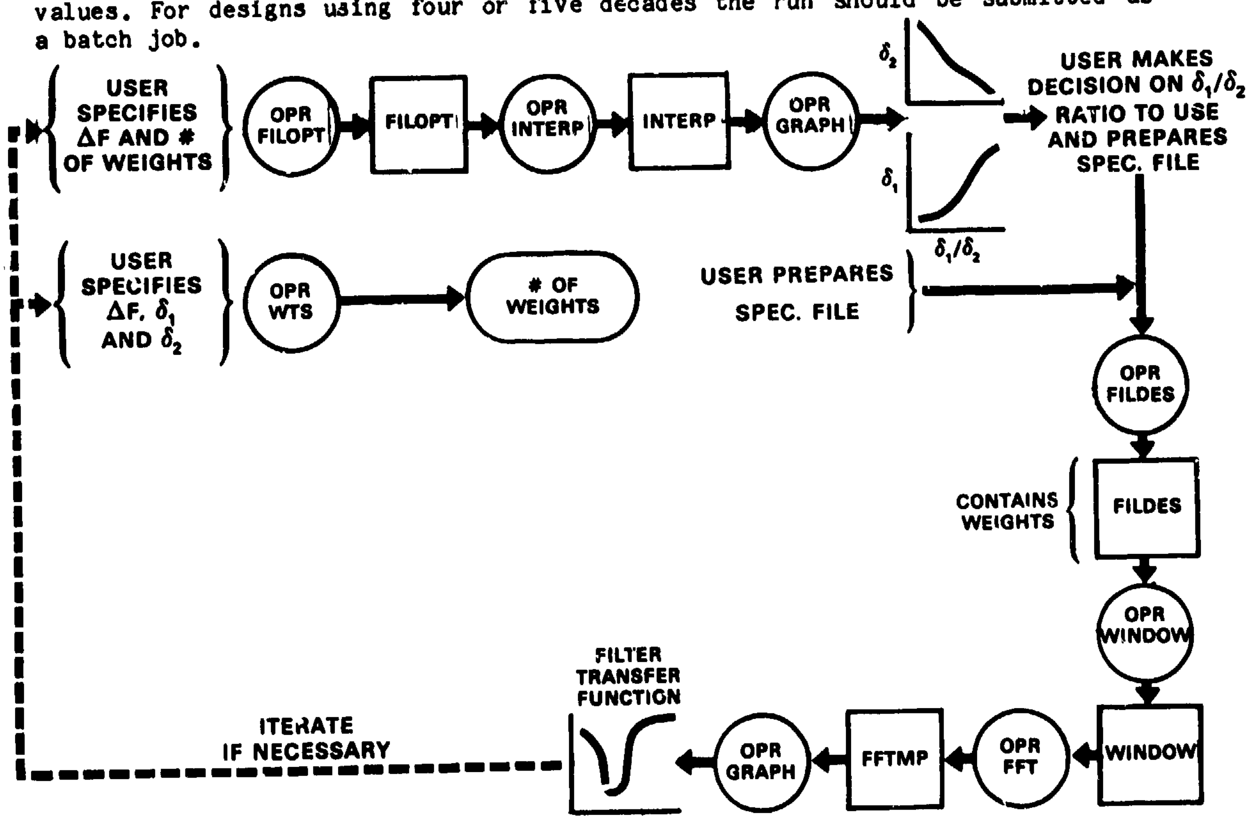
This data specifies a length 32 bandpass filter with stopbands 0.0 to 0.1 and 0.425 to 0.5, and passband from 0.2 to 0.35. The weighting ratio starts with the value 0.1 and ends with the value 1000.0 for the stopbands and a value 1.0 in the passband. The grid density defaults to 16.0.

LIMITIONS: There are some problems in the filter design program that result in a lack of convergence message. The maximum number of coefficients is 512.

To obtain a plot of the FILOPT dataset use operator INTERP to interpolate the fill data points and plot the values using the GRAPH operator. Specify:

```
INTERP FILOPT
GRAPH INTERP
```

Warning: FILOPT operator executes the Remez exchange algorithm up to 50 times and creates a dataset containing three series of up to 100,000 points in length. This procedure takes a large amount of time and space to accomplish. Also, the GRAPH operator uses a considerable amount of time to plot these values. For designs using four or five decades the run should be submitted as a batch job.



6.13-2

Usage of FILTER Command

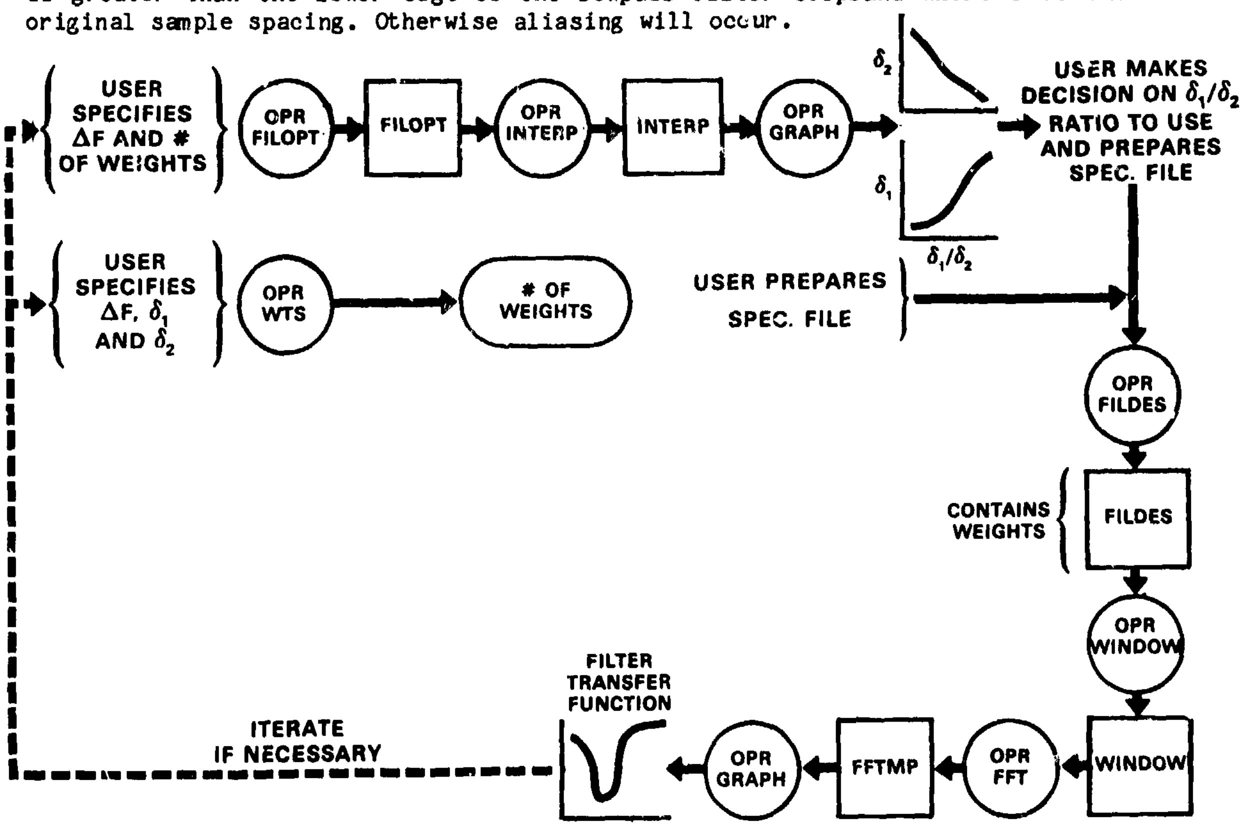
Subroutine FILTER extracts the filter coefficients from a dataset called FILDES unless the user specifies that they exist elsewhere by including the optional parameter "DSN=dataset". The dataset of filter coefficients must be created by FILDES before FILTER is executed. Filtering makes contiguous intervals noncontiguous. To obtain contiguous filtered intervals, use SETUP to create one large interval, filter it, and then use the SUBSET operator to produce the set of contiguous intervals desired. To filter the dataset WINDOW with the filter coefficients currently stored in FILDES, the user would specify:

```
FILTER WINDOW
```

If he wished to use filter coefficients stored in N91HP04 the user would specify:

```
FILTER WINDOW DSN=N91HP04
```

To avoid accidental aliasing problems, decimation can be accomplished only as an option of the FILTER command. The user should enter "FILTER dataset DEC=n" where "n" specifies that only the 0th, (n)th, (2n)th,. . . points will be kept after filtering. If this option is omitted, every point is kept (no decimation). WARNING: The decimation rate "n" must be chosen so that 1/(2*n*T) is greater than the lower edge of the lowpass filter stopband where T is the original sample spacing. Otherwise aliasing will occur.



Usage of FLOP Command

Output for a given operation is routed according to the state of a system variable IOUT. The interactive user by default receives output at the terminal. If he wishes to have hard copy of some operation, he simply types the command "FLOP". From that point, printout goes to [usrid.IDSP]IDSPOUT.DAT which the user can access according to his needs after the session ends. Output to the terminal can again be restored by typing "FLOP". Interactive user prompts are unaffected by this command.

Usage of GRAFCK Command

This command allows the user to check on the status of the subprocess that he initiated via the GRAPH command. The form of the command is:

GRAFCK n

where "n" is optional. If "n" is omitted, only one status check is made. Otherwise, a status check will be make every 15 seconds until either the subprocess has finished or "n" checks have been made. If the subprocess failed, a listing of the log file will be printed. Note that under some conditions GRAFCK will claim that the graph terminated normally even when the subprocess has actually failed. A look at the file (GRAF.LOG) will describe any problems in detail.

Usage of GRAPH Command

Plot all the series of any dataset on the VERSATEC plotter in the specified type plot (0 = linear(x)-linear(y), 1 = linear(x)-log(y), 2 = log(x)-linear(y), 3 = log(x)-log(y)).

Specify line-type desired by (-1 = points plotted as symbols only, 0 = points connected by lines only, 1 = points plotted as symbols connected by lines).

If only the j-th series is desired, it alone can be plotted by specifying the optional parameter SFR=j.

Both real and imaginary parts of complex datasets are plotted on separate plot frames. The user can specify one or the other by the optional parameter "REAL" or "IMAG".

The length of the x-axis can be set by specifying "XLEN=x" where x is in interval [1,100] (units of inches). The range of values desired in the plot can be specified via the option SIZE. If this option is chosen, the next record in the command stream must specify XL,XH,YL,YH in 4G10.0.

To obtain nice labeling in linear plots, specify limits so that the desired range is neatly divisible by five. Logarithmic plots are forced to begin and end on decades. Points that fall outside the plot window are plotted at the boundary. The independent variable is by default plotted as indices where i = 0, 1, 2, . . . , NPTS - 1.

If the user wants it in terms of the actual delta spacing of the dataset, these indices will be multiplied by delta if the user specifies option "DELTA".

If plots are specified to be logarithmic in the first variable, the first point of the plot is automatically omitted.

In the first example, a log-linear, symbols-only plot is given for every series of dataset WINDOW; in the second, only the third series of dataset INTERP is given a linear-linear, symbol-line plot; in the third, only the real part of dataset FFT is given a linear-log, symbol plot with actual data spacing in the independent variable and range limits for X and then for Y:

```
GRAPH WINDOW 2 -1

GRAPH INTERP 0 1 SER=3

GRAPH FFT 1 -1 REAL DELTA SIZE
0.,1.25,10.,1.E5
```

If the user wishes to stack several graph commands in rapid succession and then plot them all at once, he should specify the option "HOLD" on all of them except the last graph command executed. (WARNING: When this option is used a dataset name should usually appear only ONCE in the group being held for one graphing job. Remember that only the last version of the repeated dataset will exist at graphing execution. Depending on the command sequence this final version may have parameters incompatible with an earlier version, using the job to fail.)

Special effects can be accomplished by combining other operators with the GRAPH command. For example, suppose the user wanted a logarithmic plot of only the negative values of a dataset SPECTOFF. He could enter:

NORM SPECTOFF -1

GRAPH NORM SIZE

where in the size parameter he would specify how closely he wished the plot to approach zero. The results that are "off scale" would correspond to non-negative values in the original dataset.

The actual plotting algorithm accomplishes a VERSATEC plot via a subprocess which IDSP creates for the user. The first five characters of this process name will be the USRID. To check the status of this subprocess use command GRAFCK.

A log file of each graphing job is written to GRAFJB.LOG so that the user can examine it in case his results were different from his expectations. The user can easily delete all such files by the DCL command: "DEL GRAFJB.LOG.*".

If the user has an HP2648A terminal, he can have graphics output sent to the screen by entering the option "DEV=HP". Up to 500 points of a given series will be displayed. Then the user can utilize the commands inside the HP-graphics package (for help, type "HELP" after points are displayed). When he has seen enough of those points, the user types "EXIT" to return him to the next segment of graphic display. He will then see his current graphics display plus an over-laid alphanumeric display describing the graphics. Graphic display can be turned on and off via key: "SHIFT GDSP"; alphanumeric via: "SHIFT ADSP".

When the option DEV is used, the following options are ignored: XLEN, HOLD. Also even though the line-type parameter is required, the present configuration displays points only. (Inside the HP-package, these can be connected by a line by entering "DATA LINE".)

Usage of the INTERP Command

Interpolation is linear. If any flag values are found, they are replaced by interpolated values. The example below results in linear interpolation:

INTERP SETUP

Usage of MEM command

Maximum Entropy Method of computing Power Spectra (Anderson's derivation, Geophysics, Vol 39, Feb., 1974)

This operator handles both real and complex datasets as input. The power spectrum is output as a real dataset (MEM.DAT). The subroutine COEFF calculates the filter coefficients A_m and the subroutine CPSPEC calculates the power spectrum at frequencies from zero to f_{nq} (Nyquist frequency) with the desired frequency spacing (FRQSPC). The default normalization factor (NORM) is 1. The default number of filter coefficients (NCOEF) is equal to 2N/ln2N (reference: Berryman, Geophysics Vol 43, Dec. 1978). The default number of frequency points (NFP) is equal to N/2 + 1, where N is the number of points in the original time series.

The optional parameters may be input as follows:

NORM =X; where X is the desired normalization factor (real)
NCOEF=Y; where Y is the number of filter coefficients (integer)
NFD =Z; where Z is the number of frequency points (integer)

EXAMPLE:

MEM SETUP

uses defaults values

MEM SETUP NCOEF=50 NFP=75

NCOEF = 50, NFP = 75, default normalization, NORM = 1

MEM SETUP NORM=5.E-6

uses given normalization factor and gets default values for NCOEF and NFP

WARNING: The MEM operator is recommended for datasets with less than 1000 points, increasing the number of points, NCOEF or NFP will cause MEM to be slower.

LIMITATIONS: Because of the size of the working arrays required, operations involving MEM are limited to a much smaller size than some other operators. Therefore, datasets must contain less than 15,000 complex or real points/series for up to 8 series.

Usage of the MNFLD Command

The operator MNFLD performs a transformation from the coordinate system of an input vector data set to mean field coordinates. It initially computes the mean field vector from the input data. In the new coordinate system, ZMF is taken in the direction of the mean field. XMF is chosen to lie in one of the original coordinate planes, determined by the intended use of the transformed data. The desired plane is selected in response to a prompt: 1 = XY, 2 = YZ, 3 = XZ, which appears after the operator is commanded: MNFLD SETUP, MNFLD INTERP, etc. For whatever choice is given for XMF, YMF is properly oriented to complete the right-handed coordinate set. The transformation is only valid for matrix size = 3.

If spectral analysis is to be performed at a later stage, note that transformation to mean field coordinates permits specification of the MF option on the subsequent call to the operator EIG, for example,

EIG SPECTD SPECTOFF DSTAT MF.

This sets a flag MFIELD = 1 internal to EIG and its supporting subroutines, and enables application of a special eigenvector coordinate system definition appropriate for mean field data (see operator EIG documentation).

If the magnitude of the projection of the mean field vector on the plane chosen for the new X-axis is less than a specified lower limit (currently coded into subroutine MFCOMP, which is called by MNFLD, as the quantity BLO = 0.01 times the total mean field magnitude), then the rotation matrix for the transformation is specified rather than computed (see matrix definitions following statement numbers 61 and 81 in a listing of subroutine MFCOMP for the cases where XZ and YZ planes are selected, respectively). If the input data has a mean vector already in the ZMF direction (with an XY plane projection less than or equal to BLO), then no transformation is carried out, and the message "NO ROTATION - DATA ALREADY IN MF COORD" appears. A listing of the mean field vector, and the transformation matrix used will appear on the terminal screen (or the above "NO ROTATION" message) unless a FLOP is commanded, which routes the output listings to IDSPOUT.DAT for later hardcopy display availability.

Usage of NORM Command

The NORM operator multiplies every point in the dataset by the specified factor. In the following example, dataset FFT is normalized by 0.5E-4.

```
NORM FFT .00005
```

Usage of the RECPOL Command

The operator RECPOL will convert from rectangular to polar or polar to rectangular coordinates depending on parameter ITYPE.

If ITYPE=1: rectangular to polar, if ITYPE=0: polar to rectangular.

The data set created as output will be RECPOL.DAT. The location of x,y,z in the input data set will need to be supplied if converting from rectangular to polar, while the location of r, theta, phi in the input data set will need to be supplied if converting from polar to rectangular. If polar to rectangular the order of output data set will always be IX=x, IY=y, IZ=z. If rectangular to polar the order of output data set will always be IX=r, IY=theta, IZ=phi.

EXAMPLE OF INVOKING RECPOL:

RECPOL datasetname

The user will then be prompted to enter whether he/she wants to go from polar to rectangular or rectangular to polar. (ENTER A 0 IF WANT TO GO POLAR TO RECTANGULAR, ENTER A 1 IF WANT TO GO RECTANGULAR TO POLAR.) on the same line the user will enter the locations for x,y,z or for r,theta,phi in the input dataset, as appropriate. The locations will be entered as a number between [1, #SERIES].

EXAMPLE: 0,1,2,3

Usage of REDO Command

This command allows one to repeat a sequence of commands in batch jobs by rewinding the file containing the sequence. User specifies 'REDO'.

Usage of the ROTATE command

The operator rotate performs a coordinate rotation using data sets EIGVEC and EIGVEC1, which contain the direction cosines (eigenvectors) of the variance ellipsoid computed by operator EIG. i.e., the principal axis of the variance ellipsoid at a particular frequency (NFREQ). Thus this operator assumes EIG hasalready been executed and the data sets EIGVEC and EIGVEC1 exist, unless the option to enter the direction cosines interactively is selected, in which case the direction cosines are entered from the terminal.

If EIGVEC and EIGVEC1 are used for rotation then the first three series of the EIGVEC dataset at NFREQ supply the direction of minimum variance. The first three series of the EIGVEC1 dataset at NFREQ supply the intermediate direction and series 4-6 of the EIGVEC1 dataset at NFREQ supply the maximum direction of variation.

If the rotation is to be performed from terminal input, the user will be prompted for the direction cosines. NFREQ is not required and thus the user will not be prompted for NFREQ.

The name of the data set created by this operator is ROTATE.DAT.

TO INVOKE OPERATOR:

ROTATE datasetname

FOR EXAMPLE:

ROTATE IS120105 THEN FOLLOW PROMPTS.

THE PROMPTS WILL BE AS FOLLOWS:

The first input will be to indicate where in the input dataset to find x,y,z (each should be a number between 1 and the number of series). The next prompt will indicate whether the direction cosines will be input from terminal or obtained from datasets EIGVEC and EIGVEC1. If the direction cosines will be obtained from EIGVEC and EIGVEC1 the final prompt will be for the desired frequency (NFREQ). If the direction cosines will be input from the terminal, there will be three prompts to obtain the direction cosines.

EXAMPLE: (DIRECTION COSINES INPUT FROM TERMINAL)

1,2,3 1 .9890,0.2,.1439 .5567,.23,.89 .1656,-.2367,.3456

EXAMPLE: (DIRECTION COSINES FROM EIGVEC AND EIGVEC1)

1,2,3 0

Usage of SETUP Command

SETUP receives as parameters the endpoints of the span of time that the user wants analyzed (IT1,IT2), and the interval size (SIZE) he wants processed, specified in units according to IUNIT. (Interval: basic time segment to be analyzed. Span: Total time composed of contiguous intervals, see Fig. 1.0-3.) IT1 and IT2 are converted from date/time format into REAL*8 variables BEGTIM and ENDTIM in units of seconds.

At the end of the routine BEGTIM is right adjusted by the amount of processing that has occurred to prepare for a subsequent run. If the left endpoint is greater than the right, attempt will be made to input parameters for a new span.

User data are provided by calls to subroutine INPUT. Data gaps are filled with flag value BAD (-99.9). Analysis is based on the time sampling rate (DELTIM) that was found when the buffer was filled with user data.

Variables:

IT1(6) = start date/time of span desired [4 digit year; day (Jan 1 = day 1); hour; min; sec; msec].

IT2(6) = end date/time of span desired.

IUNIT = parameter indicating units of SIZE: (1 = seconds, 2 = minutes, 3 = hours, 4 = days).

KSER = n--user specification of which series desired from INPUT block: If n is positive, choose the first n series in block; if n is negative, choose the indices represented by each digit, starting at the rightmost digit. For example, n = 3 chooses series 1, 2, 3; n = -351 chooses series 1, 5, 3 in that order. For factors influencing choice of specification order, see SPECT (operator which calculates spectral matrices).

SIZE = size of interval to be processed (see IUNIT parameter).

Those records must be placed into FOR051.DAT before processing begins, so that SETUP can access required parameters:

IT1(6) format(6I5)

KSER,SIZE,IUNIT format(I10,F10.3,I10)

IT2(6) format(6I5).

SETUP attempts to collect data for analysis beginning with the time specified as the beginning of the span. The first such point that contains all good data defines the start time of the interval being setup. The stop time of the span (IT2) is an absolute maximum. The endpoint of the interval is then defined as:

B = MIN(A+SIZE*function(IUNIT),IT2).

All points found with corresponding times less than B will be included in the current interval by SETUP. The point corresponding to B will go into the next interval. Any interval may be shortened to insure that good data exists at the endpoint of the interval.

For additional details on assignments and files required, see Section 4 and Appendix A and Appendix F.

Additional span(s) subsequent to the first may be accessed by immediately adding the appropriate three records for each as specified above.

Ordinarily the user need not concern himself with the size of arrays required. In response to the command "SETUP", the system sets up the next interval for processing. If the user also wishes to see a display of the intermediate data buffer as it is being filled, he should specify:

SETUP 1

If the user has his parameters stored in a file different from FOR051.DAT, he can access them by specifying the optional paramter "DSN=file", for example:

SETUP DSN=V2PLASMA.DAT

NOTE: Do not Control Y out of IDSP if processing multiple intervals. This will cause all of the data statements to be reinitialized and IDSP will start the span at the beginning.

Usage of SHOW Command

Output for every IDSP operation is placed into a dataset identified as 'operation.DAT'. History of any given dataset can be obtained by typing 'SHOW operation'; History and data by 'SHOW operation D'. Interactive users can obtain hard copy of any dataset by routing output to disk via the FLOP command, and then executing 'SHOW operation D'. "D" can be any nonblank character.

Usage of SPECT Command

SPECT uses the output of the Fourier Transform to compute the spectral density matrix (PSD). This operator uses as input the first JSER series of the requested dataset where JSER = min(NSER,4), and NSER is the number of series in the dataset. All other series are ignored.

Let the ith point of each of the four series used as input to this operator be represented by A(i,j), j=1,2,3,4. Then PSD(i,j,k) = XNORM*Conjg(A(i,j))*A(i,k) gives the raw j-kth element of the ith spectral matrix. For an input dataset containing NPTS and for i = 1, 2, . . . , NPTS-1 XNORM = 2./((2*NPTS-2)**2), thus folding the negative power and adding it to the positive. For i = 0, XNORM = 1./((2*NPTS-2)**2).

The user must specify how many of these matrices he wishes to have added together to make spectral estimates. He is required to specify an odd integer, thus centering the effective frequency of the estimate over the center point used in estimation. To retain the raw spectral estimates, the user must specify one point per estimate. Each estimate is actually a simple sum.
Suppose the user specified n points per estimate where n=2k+1. The special case of zero frequency is accomplished by retaining the 0-th point without modification, and ignoring the next k points.

If the number of spectral matrices was such that the final estimate is not "full", the available points are used and a message is written to the history of the dataset, detailing how many points were used.

By default, the system delivers results in units/HERTZ by dividing by the frequency width of the estimate. The 0-th estimate is skipped (left unchanged). If the user wishes his results in terms of power, he should specify the optional parameter POWER.

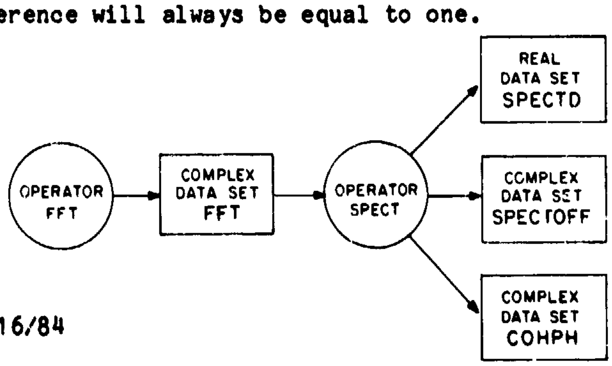
The following example will form spectral matrices using dataset FFT as input with 7 elements forming each estimate:

SPECT FFT 7

The diagonal terms 11,22,33,... are stored in SPECTD. The off diagonal terms are stored in SPECTOFF by taking in order the rows of the upper triangular portion of the spectral matrix. The history record details which matrix element is associated with which series in dataset SPECTOFF.

In a similar fashion, the coherence and phase arising from the off-diagonal terms are stored as the real and imaginary parts of complex dataset COHPH, respectively. WARNING: Because of the underlying mathematics, if only one point is used to form each estimate, the coherence will always be equal to one.

LIMITATION—see FFT.



Usage of the STOP Command

The STOP command is necessary to terminate the analysis session. A check will be made to insure that any subprocesses initiated for graphing have finished. If necessary, IDSP will wait until they finish.

Usage of SUBSER Command

This command allows the user to form a subset of the specified dataset. The form of the command is:

SUBSER dataset n1 n2

where n1 is the index of the first series desired, and n2 is the number of series to be placed into the subset.

If the dataset WINDOW has 7 series, then the following command will place series 2, 3, and 4 into dataset SUBSER:

SUBSER WINDOW 2 3

For an inverse operator, see CONCAT.

Usage of SUBSET Command

The user can extract a subset of NP points from a specified dataset beginning at index I1 where I1 is in the interval [0;NPTS-1] and NPTS is the number of points in the parent dataset. This capability allows one to obtain contiguous intervals of data that has been interpolated and filtered.

The following example extracts contiguous subsets of 500 points each from the dataset FILTER which has 1500 points or more. Note that the first argument is the index of the starting point, and the second argument is the number of points to be extracted. In the example we are extracting 500 points each time.

```
SETUP

INTERP SETUP

FILTER INTERP

SUBSET FILTER 0 500
do chosen operations on SUBSET(Note that
unless COPY has been used the previous SUBSET
has been lost)

SUBSET FILTER 500 500
do chosen operations on SUBSET

SUBSET FILTER 1000 500
```

If you wish to save these subsets, see COPY.

Usage of TRACE Command

This command allows the user to form a trace of the specified dataset. The form of the command is:

TRACE dataset n1 n2

where n1 is the index of the first series desired, and n2 is the number of series to be placed into the trace. If the dataset WINDOW has 7 series, then the following command will sum series 2, 3, and 4 and place the result into dataset TRACE:

TRACE SPECTD 2 3

Normally operator TRACE is used on data set SPECTD

Usage of WINDOW Command

WINDOW windows every series in the dataset according to user specifications. (0 = Rectangular, 1 = 10% cosine taper, 2 = Hanning, 3 = Hamming.)

The dataset can be padded with zeroes out to point number "n" (after windowing) by specifying the optional parameter "PAD=n". For example, a Hanning window of dataset INTERP is accomplished by:

WINDOW INTERP 2

A rectangular window of dataset FILDES, then padded with zeroes out to point 2048 is accomplished by:

WINDOW FILDES 0 PAD=2048

Usage of WTS

LOW PASS FILTER WEIGHTS ESTIMATION OPERATOR

WTS Calculates an estimate for the number of filter weights necessary to design a given filter using the Remez exchange algorithm. To execute, input the start and stop frequencies of the transition band, sampling frequency of the data, and the desired pass band ripple and stop band ripple.

SAMPLE INPUTS:
First record: F1,F2,FS
Second record: PBR,SBR

F1 - Start frequency of the transition band
F2 - Stop frequency of the transition band
FS - Sampling frequency of the data

PBR - Pass band ripple ($20*\log(1+delta_1)$)
SBR - Stop band ripple ($20*\log(delta_2)$)

Two types of ouput can be generated by WTS. First is a printout that can be sent to the terminal screen or to a file named [usrid.idsp]idspout.dat. To sent the output to a file the user must execute the command FLOP just before the execution of this operator. This output contains the input parameters, the normalized band edges, $delta_1$, $delta_2$, the ratio between $delta_1$ and $delta_2$, and the number of filter weights calculated.

SAMPLE OUTPUT:
TRANSITION BAND FREQUENCIES = 0.6 - 0.8
SAMPLING FREQUENCY = 4.4

PASS RIPPLE = 1.0 DB
STOP RIPPLE = 50.0 DB

BAND EDGES = 0.0000 0.1364 0.1818 0.5000

$DELTA_1$ = 0.12202
$DELTA_2$ = 0.00316
$DELTA_1/DELTA_2$ = 38.58564

NUMBER OF FILTER WEIGHTS = 32.459

The second output contains the necessary input parameters for the FILDES command. These values will be stored in the current version of FOR058.DAT.

SAMPLE OUTPUT:
33,1,2,0
0.0,0.1361,0.1818,0.5
1.0,0.0
1.0,38.586

LIMITION: WTS will produce a FILDES input dataset with an odd number of filter weights (maximum of 511 weights).

BIBLIOGRAPHY

1) Anderson, N., Shortnotes: On the calculation of filter coefficients for maximum entropy spectral analysis, Geophysics, Vol. 39, No. 1 (February, 1974), pp. 69-72.
2) Bergland, G. D., A Guided Tour of the Fast Fourier Transform, IEEE Spectrum, July 1969.
3) Berryman, J. G., Choice of operator length for maximum entropy spectral analysis, Geophysics, Vol. 43, No. 7 (December 1978), pp. 1384-1391.
4) Brigham, E. O., The Fast Fourier Transform, Prentice-Hall, Inc, Englewood Cliffs, New Jersey, 1974.
5) Burg, J. P., Maximum entropy spectral analysis, presented at the 37th Annual International SEG Meeting, Oklahoma City, OK, October 31, 1967.
6) Carleton, T. P., LEP VAX Users Guide, NASA/GSFC TM 84931, October 1982.
7) Espenak, F., K. W. Behannon, W. H. Mish, J. A. Jones, LEP Generalized Spectral Analysis System, NASA/GSFC X-694-77-250, Revised April 1979.
8) Fowler, R. A., B. J. Kotick and R. D. Elliot, Polarization Analysis of Natural and Artificially Induced Geomagnetic Micropulsations, J. Geophys. Res., 72, 2871, 1967.
9) Hamming R. W., Digital Filters, Prentice Hall, Inc. Englewood Cliff, NJ 07632, 1977.
10) Harris F. J., On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform, Proceeding of the IEEE, Vol 66 No 1, Jan. 1978.
11) IMSL Library, Reference Manual, Edition 9, IMSL, Sixth Floor GNB Bldg, 7500 Bellaire Boulevard, Houston Tx 77036, 1982
12) Jenkins, G. M. and D. G. Watts, Spectral Analysis and its Applications, Holden-Day, San Francisco, 1968.
13) Lanczos, C., Applied Analysis, Prentice-Hall, Englewood Cliffs, NJ, 1956.
14) Lanczos, C., Discourse on Fourier Series, Hafner Publishing Co, New York, 1966.
15) Matthaeus, W. H., M. L. Goldstein, Stationarity of Magnetohydrodynamic Fluctuations in the Solar Wind, J. Geophys. Res. 87, NO. A12,10347, December 1, 1982.
16) Matthaeus, W. H., M. L. Goldstein, Measurements of the rugged invariants of magnetohydrodynamic turbulence in the solar wind, J. Geophys. Res. 87, 6011, 1982.
17) Matthaeus, W. H., M. L. Goldstein, and C. Smith, Evaluation of magnetic helicity in homogeneous turbulence, Phys. Rev. Lett, 48, 1256, 1982.
18) Means, J. D., Use of Three-Dimensional Covariance Matrix in Analyzing the Polarization Properties of Plane Waves, J. Geophys. Res., 77, 5551, 1972.
19) Mish, W. H., Editor, Commonly Used Digital Tape, Disk and Card Formats, NASA/GSFC, X-694-76-242, Revised Feb. 1981.
20) Moffatt, H. K., Magnetic Field Generation in Electrically Conducting Fluids, Cambridge University Press, NY, 1978.
21) Otnes, R. K. and L. Enochson, Digital Time Series Analysis, J. Wiley and Sons, New York, 1972.
22) Programs for Digital Signal Processing, IEEE Press, 1979.
23) Rabiner, L. R., B. Gold, Theory and Application of Digital Signal Processing, Prentice-Hall Inc., Englewood Cliffs, N J, 1975.
24) Rankin, D. and R. Kurtz, Statistical Study of Micropulsation Polarization, J. Geophys. Res., 75, 5444, 1970.
25) Schaff, W. E., Course Notes: Modern Methods of Digital Signal Processing, Integrated Computer Systems, Inc, October 1979.
26) Sentman, D. D., Basic Elements of Power Spectral Analysis, U of Iowa Preprint 74-5, The University of Iowa, Iowa City, Iowa 52242, January 1974.

ACKNOWLEDGEMENTS

The authors would like to acknowledge and thank M. Goldstein and W. Matthaeus for aiding in the initial debugging of the system, for their theoretical contributions, operator development, helpful suggestions, and careful review of the manuscript. We thank F. Ottens for his contributions to operator development, maintenance, and the Appendices.

APPENDIX A Writing INPUT Routines

The user interfaces his particular data to the IDSP system via an input routine which delivers a block of data to the system whenever it is needed.

The routine must be specified as SUBROUTINE INPUT with argument list: (BLOCK,TIME,MAXBLK,MEMBER,NSER,DELTIM,PCNT,BAD,EOF).

The routine arguments which subroutine INPUT must supply are as follows:

BLOCK = two-dimensional real*4 array containing up to 256 points in each of up to 8 series. This array must have dimension (256,8).

TIME = REAL*8 array containing the time associated with each point, specified in seconds since the beginning of 1960. This array must have dimension (256). To convert calandar (Jan 1 = day 1) date-time format into seconds since 1960, a special subroutine has been supplied. Fill real*8 variable XTIME with the converted time by coding:

```
CALL DATTIM (XTIME,YEAR,DAY,HOUR,MIN,SEC,MSEC)
```

where all variables except XTIME are integer*2. YEAR is specified as a four-digit integer. To convert decimal days (Jan 1 = day 0) to seconds, use a supplied subroutine:

```
CALL DECTIM (XTIME,YEAR,DDAY)
```

where real*8 XTIME is returned from inputs integer*2 YEAR (4 digits) and real*8 DDAY(Decimal Day.Fraction of Day).

MEMBER(INTEGER*4) = number of points INPUT is delivering on this particular call

NSER(INTEGER*4) = number of series being delivered to the system (max = 8)

DELTIM(REAL*4) = time interval in seconds between adjacent points

PCNT(REAL*4) = percent variation user is willing to allow in DELTIM (enter 1% as .01)

EOF = logical variable which the INPUT routine must set to .TRUE. whenever an end of file occurs during read, or when the user desires to inform IDSP not to continue calling the INPUT routine.

The following are system arguments and must not be altered:

MAXBLK(INTEGER*4) = maximum number of points allowed in each series = 256

BAD(REAL*4) = IDSP flag for bad data. Any bad data detected by the input routine should be set to BAD (e. g. BLOCK(25,2) = BAD). Missing data should also be set to BAD.

In addition, the first call to INPUT should cause character variable HISTIN to be initialized to a string of up to 80 characters describing the input dataset. This variable must be CHARACTER*80 and be specified:

COMMON /HIST1/ HISTIN

For examples showing dimensions and variable declarations, see SYS$IDSP:INPUT.FOR, INPUTVOY.FOR, which appear in this APPENDIX.

If the user desires to test simulated data, an easy interface is accomplished by writing an input routine similar to INPUTGEN.FOR, located in DRC3:[IDSP], replacing the statements that do the actual data generation. The user could make a copy of this routine and modify it to suit his needs.

```
c   Subroutine INPUT (stored in INPUT.FOR)
c   IDSP
C   R. M. Wenger (CSC) 3/27/81
c
c   reads data created by GENDAT.FOR
c   Used for testing purposes.
c*********************************************************************
c
      SUBROUTINE INPUT(BLOCK,TIME,MAXBLK,MEMBER,NSER,DELTIM,PCNT,BAD,
     1   EOF)
      COMMON /HIST1/ HISTIN
      CHARACTER*80 HISTIN
      REAL*4 BLOCK(256,8)
      REAL*8 TIME(256),START
      LOGICAL INIT,EOF
      DATA INIT/ .FALSE./
      IF(INIT) GO TO 75
      OPEN(UNIT=11,TYPE='OLD',NAME='[ZBRMW.IDSP]FOR011.DAT',
     1   FORM='UNFORMATTED',READONLY)
      READ(11)HISTIN
      READ(11) DELTIM
      INIT = .TRUE.
      EOF = .FALSE.
      PCNT = .005
      PI = 3.1415926
      NSER = 5
C
   75 CONTINUE
      MEMBER = 0
      DO 100 I=1,8
      READ(11,END=990)(BLOCK(I,J),J=1,NSER),TIME(I)
      MEMBER = MEMBER + 1
  100 CONTINUE
      RETURN
  990 EOF = .TRUE.
      RETURN
      END
```

```
c-----------------------------------------------------------------------
c    Subroutine INPUTV1 (stored in INPUTVOY.FOR)
c    IDSP
c
c    R. M. Wenger (CSC) 11/12/80.
c
c       INPUTV provides input from Voyager1 summary data.
c    structure:
c      set PCNT according to spacecraft timing variations
c      set MEMBER = 0
c      on first execution, decide which data by record of FOR052.DAT
c      loop 1:
c      if member + number of type .gt. MAXBLK goto Done:
c      read record
c      if EOF, skip 'extract', set EOF to true.return
c      get begin time of block in decimal day units
c      depending on type of average desired,
c      call 'extract' as a function of the desired array
C      increment MEMBER
c      go to loop 1:
c-----------------------------------------------------------------------
c
      SUBROUTINE INPUT(BLOCK,TIMBLK,MAXBLK,MEMBER,NSERI,DELTIM,PCNT,BAD,
     1   EOF)
      COMMON /HIST1/ HISTIN
      CHARACTER*80 HISTIN
      REAL*4 BLOCK(256,8)
      REAL*8 TIMBLK(256)
      INTEGER*4 MEM(3)
      LOGICAL EOF,INIT
      DATA INIT /.FALSE./
      DATA MEM/ 1, 5, 25/
C
      INCLUDE 'U2LJMVOY.FOR/LIST'
C
      IF(INIT) GO TO 100
      OPEN(UNIT=11,TYPE='OLD',READONLY,
     1 ACCESS='SEQUENTIAL',RECORDTYPE='VARIABLE',
     2 FORM='UNFORMATTED',BLOCKSIZE=892,RECORDSIZE=496)
      READ(52,12) HISTIN
      READ(52,11) IAV
      PCNT = .005
      EOF = .FALSE.
      INIT = .TRUE.
  100 CONTINUE
C
      MEMBER = 0
  200 CONTINUE
      IF(MEMBER + MEM(IAV) .GT. MAXBLK) RETURN
      READ(11,END=600,ERR=650)(R4D(I),I=1,496)
      IF(IAV .EQ. 1) CALL EXTR(M48,BLOCK,TIMBLK,IDATE,NSERI,MEM(IAV),
     1 IAV,DELTIM,MEMBER,BAD)
      IF(IAV .EQ. 2) CALL EXTR(M96,BLOCK,TIMBLK,IDATE,NSERI,MEM(IAV),
     1 IAV,DELTIM,MEMBER,BAD)
```

```
      IF(IAV .EQ. 3) CALL EXTR(M192,BLOCK,TIMBLK,IDATE,NSERI,MEM(IAV),
     1 IAV,DELTIM,MEMBER,BAD)
      MEMBER = MEMBER + MEM(IAV)
      GO TO 200
  600 EOF = .TRUE.
      RETURN
  650 WRITE(6,61)
      IERR = 3
      STOP
   11 FORMAT(I5)
   12 FORMAT(A)
   61 FORMAT(1X,'****NOTICE: ERROR READING INPUT FILES. SESSION STOPS*')
      END
```

```
c-----------------------------------------------------------------------
c   Subroutine EXTR
c
c      EXTR gets data specified in array POINTS, and places it into
c   BLOCK; it places associated times into TIMBLK.  NPOINT is number
c   of points in passed array POINTS.
c        inside extract:
c          convert flag data to BAD
c          place data into BLOCK
c          calculate times (as fct of type) and place into TIMBLK
c          deliver DELTIM (in seconds)
c-----------------------------------------------------------------------
c
      SUBROUTINE EXTR(POINTS,BLOCK,TIMBLK,IDATE,NSERI,NPOINT,
     1   IAV,DELTIM,MEMBER,BAD)
      REAL*4 POINTS(275),BLOCK(256,8),DT(3)
      REAL*8 TIMBLK(256),TIME
      INTEGER*2 IDATE(6),IYR
      DATA FLAG/999./
      DATA DT/ 48., 9.6, 1.92/
      NSERI = 7
      DELTIM = DT(IAV)
C
      IYR = IDATE(1) + 1900
      CALL DATTIM(TIME,IY     TE(2),IDATE(3),IDATE(4),IDATE(5)
     1   ,IDATE(6))
      TIME = TIME - DELTIM
      DO 200 I=1,NPOINT
      TIME = TIME + DELTIM
      TIMBLK(MEMBER+I) = TIME
      NGET = I
      DO 150 J=1,NSERI
      BLOCK(MEMBER+I,J) = POINTS(NGET)
      IF(POINTS(NGET) .EQ. FLAG) BLOCK(MEMBER+I,J) = BAD
      NGET = NGET + NPOINT
  150 CONTINUE
  200 CONTINUE
      RETURN
      END
```

APPENDIX B-WRITING USER DEFINED OPERATORS

The user can write operators of his own and interface them into the system. For how to write them, see below. Object versions of user operators can be interfaced into the IDSP system via the link step (see How to START Using IDSP Section 4.0). The sequence has been designed so that the users library is searched ahead of the IDSP system.

The user can write his own operators and interface them into the IDSP system by adhering to the following conventions:

1. The name of the subroutine must be "Ui" where i=0,1,2,...,19.

2. Returns must always be "RETURN 1"

3. Use an INCLUDE statement to get current system initialization. Usage of the INCLUDE statement helps insure compatibility with IDSP system updates. Failure to use INCLUDE may result in subroutine incompatibility as IDSP is updated.

4. CMD is a character string containing user parameters specified at execution according to the desire of this operator. See "OPERATOR PARAMETERS VIA CHARACTER STRING" in APPENDIX C.

5. HISTO is an array of character strings in which the user describes in words the effect of this operation on the data.

6. HIST is the system history of operations that is retrieved with the given dataset when it is fetched. That history is actually a composite of all HISTO's that have detailed the meaning of each operation affecting the given dataset. Any new operation history in HISTO is appended to HIST by call to subroutine HISTUP. This subroutine takes NHISTO elements of array HISTO, appends them to the NHIST elements of array HIST and updates NHIST = NHIST + NHISTO. It also includes a safety check to prevent NHIST from getting too large.

7. IDGET, IDSAV are variables containing the names of the datasets respectively that the user wishes to fetch, and then save after appropriate operations.

8. Details on data structures involving subroutines FETCH and SAVE can be found in APPENDIX C.

9. For maximum utility in interactive mode, all printout except prompts should be written to logical unit number IOUT. This variable must not be altered by the subroutine. All user prompts should be written to logical unit FOR006.

10. DEG, PI, BAD are system parameters and must not be altered and are respectively: 180/3.141593, 3.141593, -99.9 (BAD is flag for bad data).

11. IDSP1 contains the space available to the user for working arrays. LIMSPC and MAXSER are system parameters (500,000 and 8 respectively) and must not be altered.

12. The dimensions needed for arrays are calculated by subroutine LIMSET and

are stored in array LIM. This subroutine must be called so that rrays can be properly dimensioned. For description of parameters, see LIMSET.FOR, APPENDIX C.

The operator NORM is an example requiring one input dataset (see NORM.FOR); The operator EIG is an example requiring more than one input dataset (see EIG.FOR).

LIST OF LOGICAL NAME ASSIGNMENTS

SYS$IDSP:IDSP = DRC3:[IDSP]IDSP

APPENDIX C-DESIGN CONSIDERATIONS FOR IDSP

USER DATA STRUCTURE AND MANIPULATION

1. Assume input data from spacecraft is real; usage of complex values occurring only after Fourier Transform.

2. Use a single set of input/output routines to store or retrieve either real or complex values, with array passed as a single parameter.

3. Each operator saves output into dataset called 'oper'.DAT.

4. Dataset names have arbitrary length up to maximum total of thirty characters.

5. User data is stored in a two-dimensional array where the first dimension refers to the number of points per series to facillitate operations with each series as a unit.

6. Output from each operator is written unformatted to disk as data(npts,nser) regardless of whether the data are real or complex.

7. Each operator module can set up variable domain array sizes according to the needs of the dataset being accessed and the working array requirements of the operation. The parameters NSER and NPTS are extracted from the dataset and used by the subroutine LIMSET to set up an array LIM defining the dimensions. The space available for these large arrays is found in IDSP1/SPACE. Each module splits this up as needed. Then these arrays and dimensions are passed to a routine that does the actual operation. Because of extra space requirements, FFT operations use a different routine LIMFFT to apportion space needed. For how these requirements are calculated, and limits involved, see a listing of the subroutine.

8. To facilitate implementation, each operator uses an "INCLUDE" statment: "INCLUDE 'OPER.FOR'" and each subprocessor uses "INCLUDE 'OPER1.FOR'". Each routine can add any other statments that are specific to its needs.

9. Each operator module can specify what format it desires to have complex arrays in memory. If JTYPE = 1, DATA is real and must be dimensioned REAL DATA(LIM(1),LIM(2)); parameter JFORM is ignored. If JTYPE = 2, DATA is complex, and will be retrieved according to parameter JFORM. In the call to SUBROUTINE FETCH, if the JFORM parameter = 1, data will be returned to array DATA as a complex variable; DATA must be specified COMPLEX DATA(LIM(1),LIM(2)). If JFORM = 2, data will be returned to array DATA as two real arrays, with the real part of each series stored in DATA(npts,j) and the imaginary part stored in DATA(npts,nser+j) where DATA must be specified REAL DATA(LIM(1),LIM(2)). Similarly, data are accessed by SUBROUTINE SAVE according to identification of data type and storage format given by parameters JTYPE and JFORM. When the calling routine is dimensioned in this fashion, the input/output subroutines will correctly save or fetch any of the three types of formats, retaining the flexibility of passing a single data array through an argument list. NEVER allow dimensions and NPTS and NSER to take dual roles. Such a structure encourages VERY SUBTLE ERRORS.

10. Due to special working arrays needed for operators FFT, FFTIN, and SPECT, a special subroutine (LIMFFT) was written to define locations and lengths of necessary arrays to make all these operators compatible. For any dataset to be input to any one of these three operators, it must be of size compatible with all three operators.

11. All input/output of user datasets is controlled by calls to subroutines whose listing can be found in SYS$IDSP:IO.FOR. Further details on subroutine parameters are found in the comments of this listing.

12. With each dataset are stored arrays XPARM and IPARM as follows: XPARM(1) = current data spacing XPARM(2-3) = time in seconds associated with first point (REAL*8). XPARM(4-5) = time in seconds associated with last point (REAL*8). IPARM(1) = pointer indicating current domain of dataset (1 = time (seconds); 2 = frequency (Hertz)). IPARM(2) = version number of system (see FIXVER in IO.FOR).

MISCELLANEOUS DETAILS Logical File Names:

The following FORTRAN logical unit numbers are used by IDSP and its associated routines.

1) User input parameters: 51-79

LOGICAL UNIT #S	USE
51-79	input to user INPUT routines
51	input to SETUP routine
58	input to FILDES (design filter)
57	character strings specifying operations

2) User data: 10-19

LOGICAL UNIT #S	USE
11	experimenter data test routine read by DRC3:[IDSP]INPUT.FOR
13	INPUT/OUTPUT parameters
17	give interactive user hardcopy

3) System Character strings: 20-29

LOGICAL UNIT #S	USE
21	HELP files
22	command history
23	submit plots
24	parameters needed for separate plot jobs
25	cluster names for plot; also used to associate Command Event Flag Cluster with a process

4) Error Conditions: 90-99

LOGICAL UNIT #S	USE
91	error condition in Versaplot batch job

Event Flags

Event flags 64 and 65 are reserved for testing the status of graphing jobs.

Event flag 1 is used in GRAFCK. See operators GRAFCK, GRAPH. See programs GRAFGOOD, GRAFGOOF, and GRAFJB.

IDEAS

1. Edit wild points using IMSL routine ICSMOU

OPERATOR PARAMETERS VIA CHARACTER STRING

During execution, parameters for a given operation are transmitted to the system via a set of substrings in the character variable CMD. The command string for any given operator is:

command p1 p2 p3 . . . .

where "command" is the name of the operation desired, and the "pi" are parameters used by the operator. The different substrings in the variable CMD are always separated by blanks. Each operator is of the form "Subroutine opera(*,CMD)". By the time execution is inside this subroutine, the "command " has been stripped off and CMD is left as:

p1 p2 p3 . . .

Access to these parameters is facilitated by calling one of two subroutines: CALL STRING(CMD,STR1) or CALL STRNUM(CMD,NUM). Both return the value of the leftmost parameter to the second argument, and return CMD with the first parameter and first blank stripped off, i. e.,

p2 p3 . . .

Use STRING when you wish to interpret the parameter as characters; use STRNUM when you wish to interpret it as an integer. Usually "p1" is the name of the dataset desired as input for the operation. Hence, many operators CALL ASK(CMD,IDGET). This routine calls STRING to obtain the dataset name. If CMD was blank (interactive user forgot to enter dataset name before carriage return), the user is prompted to enter the dataset name. If the user enters an invalid value for an optional parameter, the system will do one of two things: 1) issue prompt for new entry, 2) ignore the entry and use the default.

APPENDIX D-Execution of Batch Jobs

Batch jobs require everything that interactive jobs require plus a command file called IDSPBAT.COM which contains the commands desired. This file must exist in the same directory that contains the other user parameter files.

1) Create, for example, the following file called IDSPBAT.COM

```
$ CREATE FOR057.DAT
SETUP
INTERP SETUP
SHOW INTERP D
FFT INTERP
SHOW FFT D
STOP
$ EXIT
```

2) On the terminal execute the command file IDSPBAT.COM as follows:

```
@IDSPBAT.COM
```

This will create the file FOR057.DAT which will be read by the batch job.

3) Create a command file, for example BATCH.COM, containing the following:

```
$ASSIGN  your data  FOR011
any other assign statements your job may require
$@SYS$IDSP:IDSP  linknames or NOLINK
```

4) Submit the command file, for example BATCH.COM

APPENDIX E-Conventions on DATASET NAMES

When a dataset must be specified within a command, it can be entered in simple form, or can be fully qualified. Unless explicitly specified otherwise, data type ".DAT" is assumed. Simple form defaults to directory [usrid.IDSP]; for example dataset=INTERP actually specifies [usrid.IDSP]INTERP.DAT as the dataset name. The user can access or copy into other directories by specifying something such as [usrid.ALPHA]SETUP45 or [usrid.BETA]FFT13.SAV as the dataset name. The limit on the total number of characters for a dataset name is 30.

APPENDIX F-Existing input routines

Each application accesses data by an input routine specific to the experimenter data of the application. This routine is called by the system, and is the interface between the system and the experimenter data. Thus, user data is accessed by calls to subroutine INPUT which provides a block of up to 256 points for each of up to 8 series. With each point is also delivered an associated time, the same for all 8 series. Examples of existing input routines with required logical assignments identified by "$ASSIGN":

NOTE: The formats of many of the tapes referred to in this Appendix can be found in Mish, W. H., Commonly used digital tape, disk and card formats, NASA/GSFC X-694-76-242, Revised February 1981.

1) INPUTDET.FOR (reads detail data from Voyager) Prior to execution of IDSP the data has to be read onto disk from a Voyager Detail tape. This disk file is then read by INPUTDET.FOR.

$ASSIGN experimenterdata FOR011

Magnetic field components BX, BY, BZ are returned as series 1, 2, 3. Parameters must be stored in FOR052.DAT as indicated below. First record: string of up to 80 characters describing input file. Second record: year of data being accessed (4 digits)

2) INPUTPLS.FOR (reads plasma data from special disk dataset prepared by writing only needed items--see RAYPLS.FOR)

$ASSIGN experimenterdata FOR011

Eight series refer respectively to plasma data words 96-102, word 38. See p 1.15-15 of COMMONLY USED DIGITAL TAPE, DISK AND CARD FORMATS REV FEB 1981 for further details. Parameters must be stored in FOR052.DAT as indicated below. First record: string of up to 80 characters describing input file.

3) INPUTVOY.FOR (reads data in Voyager Summary Tape format)

$ASSIGN experimenterdata FOR011

Parameters must be stored in FOR052.DAT as indicated below. First record: string of up to 80 characters describing input file. Second record: (Specify in I5 type of averaged data desired: 1 = 48 sec, 2 = 9.6 sec, 3 = 1.92 sec)

4) ISEEINPUT.FOR: THIS INPUT ROUTINE CAN BE USED TO READ THE ISEE-1 OR ISEE-3 DATA POOL TAPE AND OBTAIN PARAMETERS AS REQUESTED BY THE USER FOR USE WITH THE IDSP PACKAGE. THE VARIABLES, FREQUENCY OF OCCURRENCE,NUMBER OF SERIES, NUMBER OF INTERVALS DESIRED ARE ENTERED ON A FILE FOR054.DAT. THE USER ENTERS THE VARIABLES DESIRED BY ENTERING THE WORD NUMBERS AS GIVEN IN "COMMONLY USED DIGITAL TAPE, DISK AND CARD FORMATS" PAGES 1.16-10 TO 1.16-13 FOR ISEE-3, AND PAGES 1.13-20 TO 1.13-27 FOR ISEE-3. FORTRAN LOGICAL UNIT 12 WILL BE USED TO READ THE ISEE DATA POOL INFORMATION. SEE DRA1:[U2FWO.IDSP] FOR A SAMPLE FOR054.DAT FILE.

5) RUSLINPUT.FOR: THIS INPUT ROUTINE CAN BE USED TO READ THE ISEE-1 AND ISEE-2 RUSSELL DATA WHICH RESIDES ON DISK AS DRC3:[U2DHF]RUSLISEE1.DAT---ISEE-1 RUSSELL MAGNETIC FIELD DATA, AND DRC3:[U2DHF]RUSLISEE2.DAT---ISEE-2 RUSSELL MAGNETIC FIELD DATA. THE USER ENTERS THE VARIABLES DESIRED, NUMBER OF SERIES, AND NUMBER OF INTERVALS DESIRED ON A FILE FOR068.DAT. THE USER ENTERS THE VARIABLE NUMBERS AS FOLLOWS: 09 IF WANT MAG FIELD BX COMPONENT IN MILLIGAMMAS (GSM), 10 IF WANT MAG FIELD BY COMPONENT IN MILLIGAMMAS (GSM), 11 IF WANT MAG FIELD BZ COMPONENT IN MILLIGAMMAS (GSM), 12 IF WANT MAG FIELD MAGNITUDE BT IN MILLIGAMMAS. FORTRAN LOGICAL UNIT 12 WILL BE USED TO READ THE RUSSELL DATA. SEE DRA1:[U2FWO.IDSP] FOR A SAMPLE FOR068.DAT FILE.

6) HELIINPUT.FOR: THIS INPUT ROUTINE CAN BE USED TO READ THE HELIOS 1 AND 2 HOURLY AVERAGE TAPE RECEIVED FROM LARRY KLEIN. THE USER ENTERS THE VARIABLE NUMBERS, NUMBER OF SERIES, AND THE NUMBER OF INTERVALS ON A FILE CALLED FOR070.DAT USING A FORMAT DESCRIPTION WRITE-UP SUPPLIED BY LARRY KLEIN. A SAMPLE FILE FOR070.DAT EXISTS IN DRA1:[U2FWO.IDSP]. EXAMPLE:12 ENTERED AS VARIABLE NUMBER INDICATES DENSITY DESIRED. FILE FOR070.DAT WILL ALSO CONTAIN INFORMATION TO BE SUPPLIED ABOUT THE INPUT TAPE---I.E. TAPE DRIVE, FORMAT TYPE, BLOCKSIZE, RECORDSIZE, FILE START, FILE STOP, AND WHETHER THE TAPE IS LABELLED OR NON-LABELLED.

7) BYRNESIN.FOR: THIS INPUT ROUTINE CAN BE USED TO READ THE DATA FROM A TAPE SUPPLIED BY JIM BYRNES CONSISTING OF DE MAGNETIC FIELD DATA. A FILE CALLED FOR060.DAT (SEE DRA1:[U2FWO.IDSP] FOR AN EXAMPLE) WILL BE USED TO ENTER THE TAPE DRIVE, RECORD FORMAT, WHETHER LABELLED OR NOT, BLOCKSIZE, RECORDSIZE, START FILE, AND STOP FILE. THIS FILE WILL ALSO CONTAIN A SECOND RECORD INDICATING THE NUMBER OF INTERVALS DESIRED.

8) IMFINPUT.FOR: THIS INPUT ROUTINE CAN BE USED TO READ THE ISEE-3 1 HOUR AVERAGE IMF TAPE RECEIVED FROM JOE KING. THE TAPE DRIVE USED, RECORD FORMAT, WHETHER SL OR NL, BLOCKSIZE,RECORDSIZE, START FILE, STOP FILE, WILL APPEAR ON THE FIRST RECORD OF FILE FOR061.DAT. THE SECOND RECORD WILL CONSIST OF THE VARIABLE NUMBERS, NUMBER OF SERIES, AND THE NUMBER OF INTERVALS. THE VARIABLE NUMBERS CAN BE OBTAINED FROM A FORMAT DESCRIPTION OF THE TAPE. EXAMPLE: 08 INDICATES THAT BX IN GSE IS DESIRED. A SAMPLE FOR061.DAT FILE CAN BE SEEN IN DRA1:[U2FWO.IDSP].

9) IMP8INPUT.FOR: THIS INPUT ROUTINE CAN BE USED TO READ THE IMP H or J 15.36 SECOND SUMMARY TAPE. VARIABLES DESIRED, NUMBER OF SERIES, NUMBER OF INTERVALS WILL BE ENTERED IN A FILE CALLED FOR053.DAT. THIS FILE WILL ALSO CONTAIN INFORMATION SUCH AS TAPE DRIVE, RECORD FORMAT, SL OR NL, BLOCKSIZE, RECORDSIZE, START FILE, STOP FILE. TO OBTAIN THE VARIABLE NUMBER, THE DOCUMENT "COMMONLY USED DIGITAL TAPE, DISK AND CARD FORMATS" PAGES 1.7-13 TO 1.7-16 SHOULD BE CONSULTED. EXAMPLE: 10 ENTERED AS A VARIABLE NUMBER INDICATES THAT FIELD MAGNITUDE (F1) IS DESIRED. AN EXAMPLE OF AN FOR053.DAT FILE CAN BE SEEN IN DRA1: [U2FWO.IDSP].

10) OMNIINPUT.FOR: THIS INPUT ROUTINE CAN BE USED TO READ THE COMPOSITE OMNITAPE INFORMATION WHICH EXISTS ON DRC3:[OMNITAPE] AND OBTAIN PARAMETERS AS REQUESTED BY THE USER FOR USE WITH THE IDSP PACKAGE. THE VARIABLES, FREQUENCY OF OCCURRENCE, NUMBER OF SERIES, NUMBER OF INTERVALS DESIRED ARE ENTERED ON A FILE FOR056.DAT. THE USER ENTERS THE VARIABLES DESIRED BY ENTERING THE WORD NUMBERS AS GIVEN IN "COMMONLY USED DIGITAL TAPE, DISK AND CARD FORMATS" PAGES 1.14-1 TO 1.14-3. FORTRAN LOGICAL UNIT 12 WILL BE USED TO READ THE OMNIDATA RECORDS. SEE DRA1:[U2FWO.IDSP] FOR A SAMPLE FOR056.DAT FILE.

11) VOYAINPUT.FOR: THIS INPUT ROUTINE CAN BE USED TO READ THE VOYAGER 1 AND 2 HOURLY AVERAGE TAPE RECEIVED FROM LARRY KLEIN. THE USER ENTERS THE VARIABLE NUMBERS, NUMBER OF SERIES, AND THE NUMBER OF INTERVALS ON A FILE CALLED FOR069.DAT USING A FORMAT DESCRIPTION WRITE-UP SUPPLIED BY LARRY KLEIN. A SAMPLE FILE FOR069.DAT EXISTS IN DRA1:[U2FWO.IDSP]. EXAMPLE:12 ENTERED AS VARIABLE NUMBER INDICATES DENSITY DESIRED. FILE FOR069.DAT WILL ALSO CONTAIN INFORMATION TO BE SUPPLIED ABOUT THE INPUT TAPE---I.E. TAPE DRIVE, FORMAT TYPE, BLOCKSIZE, RECORDSIZE, FILE START, FILE STOP, AND WHETHER THE TAPE IS LABELLED OR NON-LABELLED.

12) ISEEIMPIN.FOR: THIS INPUT ROUTINE CAN BE USED TO READ THE ISEE/IMP TAPE GENERATED BY JOE KING. THE USER ENTERS THE VARIABLES DESIRED ON A FILE FOR055.DAT. 01=BX OR BPAR, 02= BY OR BPER1, 3=BZ OR BPER2, 04=BM FOR ISEE-3. 05=BX OR BPAR, 06=BY OR BPER1, 07=BZ OR BPER2, 08=BM FOR IMP. THIS FILE WILL ALSO CONTAIN THE NUMBER OF SERIES CHOSEN. ALSO ON A SEPARATE RECORD THIS FILE CONTAINS INFORMATION ABOUT THE TAPE DRIVE USED, RECORD FORMAT, SL OR NL, BLOCKSIZE, RECORDSIZE, START FILE, AND STOP FILE. A TAPE FORMAT DESCRIPTION MAY BE OBTAINED FROM JOE KING. AN EXAMPLE OF A FOR055.DAT FILE MAY BE FOUND IN DRA1:[U2FWO.IDSP].

13) ADOLVOYG.FOR: This input routine can be used to read the Voyager Conjoint data tape. The following parameters will be read: F2, B1, B2, B3. The user enters the total number of spans to process and whether he/she desires 1.92s, 9.6s, or 48s data on a file called FOR067.DAT. A sample FOR067.DAT file exists on DRA1:[U2FWO.IDSP].

14) VOYNEWINP.FOR: This input routine can be used to read the revised Voyager 1 and 2 hourly average tape received from Larry Klein. The user enters the variable numbers, number of series, and the number of intervals on a file called FOR069.DAT using a format description write-up supplied by Larry Klein. A sample file FOR069.DAT exists in DRA1:[U2FWO.IDSP]. File FOR069 will also contain information to be supplied about the input tape, i.e., tape drive, format type, blocksize, recordsize, file start time, file stop time, and whether the tape is labelled or non-labelled.

15) YSFAHDATA.FOR: This input routine reads the file created by Fred Herrero named DRA1:[YSFAH.IDSP]NACSIN.DAT, Contact Fred Herrero for details.

<u>Detail on writing such a routine can be found under "Writing INPUT Routines. APPENDIX A".</u>

APPENDIX G-Changes incorporated in Version 1 of IDSP

The current supported version is identified by a LOGICAL assignment that is system generated for IDSP (see document on how to START using IDSP). The old version (Version 0) of IDSP will be saved but nonaccessable to the general user except through special arrangements with the systems management.

New operators include EDHIST and DCL.

Base year for internal times was changed from 1971 to 1960. This time change was made global for the IDSP system and a special common /IDSP7/ declared for it. The following files were affected:

START.FOR
SHARE.FOR
SETUP.FOR

IDSP was fixed to recognize datasets created under Version 0 and convert them internally to Version 1. All results become Version 1. This process is transparent to the user. A new subroutine was added to the file IO.FOR called FIXVER.

The file LIMSET.FOR was modified to clarify a comment.

The file INTERP.FOR was modified to correct a typographical error that caused problems for large datasets.

The file FILTER.FOR was modified to correct a timing error that was being placed into the output datasets.

The file SETUP.FOR was modified to solve several problems that had arisen.

The operator GRAPH was modified to execute graph jobs via a subprocess rather than a batch job. This was done to avoid system bottleneck when the batch queue was full. The STOP operator was then modified to make sure all graphing had finished before exiting IDSP.

Appropriate changes were also made in IDSP initialization. Files affected were:

GRAPH.FOR
START.FOR
STOP.FOR

The file GRAFCK.FOR was modified to reflect fact that the log file of the graph job is now written to the default directory.

The file GRAFJB.FOR was modified to include the name of the dataset being graphed in the file GRAFJB.LOG.

Appropriate documentation and executive modules were also modified to reflect these changes.

APPENDIX H-Existing user operators

W. Matthaeus and M. Goldstein have written several user operators which may be of general use. Several of these are described below. As of 10/14/82 these operators do not utilize the variable dimensioning aspect of IDSP. In most cases the data sets are dimensioned at (5002,8) although, as noted below, a couple of the operators use larger dimensions. The easiest way to use these operators is to link to DRA1:[YSWHM.IDSPUOP]USROP.OLB.

1. BAD POINT EDITOR (operator U10) This operater allows one to scan data sets and remove badpoints without the necessity of using a graphics terminal. The program is fully interactive and the various options available are spelled out when one types "U10". The program allows the user to set his own criterion for selecting bad points based on either the magnitude of the data or the standard deviation. Automatic editing based on those criteria is possible. In the "SCROLL" mode, four series at a time are displayed on the screen, the potential bad point is centered and five points before and after the selected point are shown. After editing the data the dataset can be rewritten on disk with the flagged points assigned the value "-99.9". Datasets can be edited rapidly with this program.

2. SPECTRAL ANALYSIS ROUTINES (operators U11, U2, U3, U4, U5)

A. Spectral Analysis Using Fast Fourier Transforms (operator U11) This routine is similar to the SPECT operator of IDSP, but provides the user with more information about the statistics of the results. This operator is dimensioned at (20600,8); it operates on the first three series of the input dataset (usually FFT). Two output datasets are produced which contain the diagonal and off-diagonal elements of the spectral tensor as well as series which contain the magnetic helicity. Unlike the IDSP operator, this operator uses running averages to achieve whatever statistical weight the uses desires. Also all the information produced by the operator is written on dataset FOR010.DAT and can be printed after the IDSP session. All power estimates have the dimensions of POWER in $(nT)^2$. It is assumed that the input data before transforming was a magnetic field time series in units of nT. Printed on the screen, the header of the output datasets, and on the FOR010.DAT dataset are the correlation length, the magnetic helicity length and several other length scales useful in MHD turbulence studies. Because this routine was written for analysis of solar wind data, the Taylor "frozen-in-flow" hypothesis is assumed and the solar wind velocity is used to convert from frequency to wavenumber. The output datasets contain the "raw" (one spectral estimate) power and magnetic helicity as well as the "smoothed" spectra.

B. Blackman-Tukey "Mean Lagged Product Technique" (operators U2 U3)

Correlation Functions (operator U2)

The user provides a time series which, at his option, may have been previously interpolated using the INTERP command. Following the interactive prompts this operator produces two datasets which contain the diagonal and off-diagonal elements of the correlation matrix. A maximum lag of 10% of the total data interval is the default assumption, but the user can readily request whatever he needs.

Spectral Matrix (operator U3) The input to this operator is the output of operator U2. This operator WINDOWS the correlation matrix (Hanning is the default), then zero pads (a factor of ten is the default), and finally computes the fast Fourier transform. If the maximum lag used in operator U2 is not 10% of the data length, then the amount of zero padding should be changed accordingly. One should know the solar wind velocity so that frequencies will be correctly changed to wavenumbers. The output data sets (including FOR010.DAT) contain the power spectrum and magnetic helicity.

3. SPECTRAL ANALYSIS OF PLASMA DATA (operators U4 U5)

Cross helicity (operator U4)

A quantity of great interest in MHD turbulence is the cross helicity. These operators permit one to determine the cross helicity using plasma and magnetic field data from the conjoint Voyager Summary tape. Operator U4 converts magnetic field measurements from units of nT to units of Alfven speed. The magnetic field is divided by SQRT(4*PI*rho) where "rho" is the mass density. One can either use the instantaneous proton density on the Summary tape or the average. A contribution from alpha particles can be included.

Plasma Spectra Operator (operator U5)

U5 uses the FFT technique to calculate the power spectrum of the velocity, density, cross helicity and total energy using the eight series that have previously been operated on by U4. Three output datasets are produced.

4. MISC. OPERATORS

Smoothing (operators U12 U13) Under some circumstances it is useful to be able to average datasets using a running average. For example, if one wants to perform an eigenvalue analysis one can use SPECT with only one spectral estimate, then smooth the spectral matrices using U12 (Real datasets) and U13 (Complex datasets) to produce spectra fully equivalent to the output of U11 but in a form that the EIG operator can handle.

Operator U9 is a graph command very similar to the IDSP GRAPH but which allows added flexibility in choosing some of the graph parameters. The routine also uses smaller lettering in the headers. The graphs are produced interactively; no subprocess is created.

APPENDIX I - INSTALLATION PROCEDURE

THE IDSP TAPE WAS CREATED USING THE DCL COPY COMMAND ON A VAX 11/780 UNDER VMS V2.3. THE VOLUME LABEL IS IDSPTP. THE MAGNETIC TAPE CONTAINS 124 FILES, DENSITY= 1600 BPI. THE TAPE CONTAINS THE FOLLOWING TYPE FILES COPIED FROM DISK:

1) FORTRAN SOURCE CODE FOR IDSP ROUTINES, AND IEEE DIGITAL SIGNAL PROCESSING (*.FOR),

2) COMMAND PROCEDURE FILES WHICH ARE PART OF THE IDSP PACKAGE (*.COM),

3) DOCUMENTATION FILES WHICH ARE PART OF THE IDSP PACKAGE (*.DOC),

4) OBJECT MODULE LIBRARY OF THE IEEE DIGITAL SIGNAL PROCESSING (EFF,WATE,D2,GEE2,ERROR2,REMEZ,OUCH,I1MACH) (DSP.OLB),

5) OBJECT MODULE LIBRARIES OF HP 2648A INTERACTIVE GRAPHICS ROUTINES WRITTEN BY MR. THURSTON CARLETON OF NASA/GSFC AND USED BY IDSP: (HPSUB.OLB AND COMMAND.OLB). NOTE THAT THE SOURCE CODE FOR MR. CARLETON'S HP ROUTINES ARE NOT SUPPLIED, BUT IF YOU HAVE A NEED FOR THEM THEY CAN BE SUPPLIED.

6) OBJECT MODULE LIBRARY OF IDSP ROUTINES: (IDSP.OLB),

7) OBJECT MODULE LIBRARY OF IMSL SINGLE PRECISION ROUTINES (EIGRS,FFTCC,UGETIO,USPKD,EHOBKS,EHOUSS,EQRT2S,FFT2C,UERTST,AND FFTRC) (IMSLIBS.OLB),

8) TWO DATA FILES (FOR051.DAT, AND FOR060.DAT) USED BY THE SAMPLE INPUT ROUTINE SUPPLIED (IDSPIN.FOR).

9) AN OBJECT FILE OF THE IDSP.FOR ROUTINE USED BY IDSPLINK.COM AT TIME LINK IS DONE THE FILE NAME IS IDSP.OBJ.

IN SUMMARY THE TAPE HAS 6 TYPE DISK FILES COPIED ONTO IT: *.COM, *.DOC, *.FOR, *.OBJ, *.DAT, AND *.OLB FILES.

NOTE THAT NONE OF THE PROPRIETARY VERSAPLOT SOFTWARE HAS BEEN PROVIDED. THEREFORE THE IDSP OPERATORS WHICH MAKE USE OF THE VERSATEC PLOTTERS WILL NOT WORK. HOWEVER IF YOUR INSTALLATION HAS THE SOFTWARE AVAILABLE THEN THE OPERATORS SHOULD BE USEABLE. NOTE THAT COMMAND PROCEDURE GRAFJB.COM LINKS WITH THE VERSAPLOT SOFTWARE SO UNLESS THE SOFTWARE IS AVAILABLE THIS COMMAND PROCEDURE WILL NOT WORK. ALSO, COMMAND PROCEDURE GRAFCM.COM WILL NOT WORK UNLESS THE VERSAPLOT SOFTWARE IS AVAILABLE. IN COMMAND PROCEDURE GRAFCM.COM THE SYMBOL PLOT IS USED. AT OUR INSTALLATION THE SYMBOL PLOT HAS BEEN DEFINED AS FOLLOWS: @SYS$DISK:[SYSMGR]PLOT WHERE THE LOGICAL NAME SYS$SYSDISK HAS BEEN EQUATED TO DEVICE DBA0: (PLOT EXECUTES A SYSTEM COMMAND PROCEDURE WHICH ALLOWS THE PLOT VECTOR FILES TO BE WRITTEN UNDER A SCRATCH UIC SO THAT USERS DO NOT EXCEED QUOTAS WHEN PLOTTING).

TO INSTALL IDSP THE FOLLOWING PROCEDURE COULD BE USED:

1) CREATE A DIRECTORY (NOT SUBDIRECTORY) ON A DISK DRIVE OF YOUR CHOICE. FOR PURPOSE OF EXAMPLE WE ASSUME THAT A DIRECTORY [IDSPTAPE] HAS BEEN CREATED ON DRIVE DRA2. THIS DIRECTORY SHOULD BE ABOUT 3000 BLOCKS IN SIZE.

2) EQUATE THE LOGICAL NAME SYS$IDSP TO THE DEVICE AND THE DIRECTORY CREATED IN ITEM #1. EXAMPLE:

   ASS DRA2:[IDSPTAPE] SYS$IDSP.

   FOR THE PURPOSES OF EXAMPLE WE WILL ASSUME THAT THE LOGICAL NAME SYS$IDSP HAS BEEN EQUATED TO DISK DRA2 AND DIRECTORY [IDSPTAPE].

3) MOUNT THE TAPE ON A TAPE DRIVE:

   MOUNT MTA1: IDSPTP

   WHERE MTA1: IS THE DEVICE NAME FOR THE TAPE DRIVE AND IDSPTP IS THE VOLUME LABEL. A 1600 BPI COMPATIBLE TAPE DRIVE SHOULD BE USED.

4) SET THE DEFAULT DIRECTORY TO SYS$IDSP. THIS ASSUMES THE DIRECTORY EXISTS.

5) COPY THE TAPE ONTO SYS$IDSP AS FOLLOWS:

   COPY MTA1:*.*.* SYS$IDSP:*.*.*

   MTA1: IS THE TAPE DRIVE. WHEN THE COPY IS COMPLETE (SHOULD BE 124 FILES),

   DISMOUNT MTA1:

   (OR THE APPROPRIATE TAPE DRIVE).

6) EDIT COMMAND PROCEDURE IDSP.COM IN SYS$IDSP AS FOLLOWS: SET UDISK EQUAL TO THE DISK DRIVE DEVICE NAME TO WHICH THE LOGICAL NAME SYS$IDSP HAS BEEN EQUATED (LOCATED AT STATEMENT 1800 OF THE COMMAND PROCEDURE). AFTER MODIFYING IDSP.COM RESAVE IT.

7) CREATE SUBDIRECTORY IDSP WITHIN THE DIRECTORY EQUATED TO BY SYS$IDSP AS FOLLOWS:

   CREATE/DIR DRA2:[IDSPTAPE.IDSP]

   THIS ASSUMES THAT THE DIRECTORY WAS IDSPTAPE AND THE DEVICE WAS DRA2.

8) SET THE DEFAULT TO THE IDSP SUBDIRECTORY CREATED IN IN ITEM #7.

9) COPY IDSPIN.FOR, FOR051.DAT, AND FOR060.DAT INTO THE JUST CREATED SUBDIRECTORY ([*.IDSP] WHERE * IS THE DIRECTORY NAME).

```
COPY SYS$IDSP:IDSPIN.FOR    IDSPIN.FOR
COPY SYS$IDSP:FOR051.DAT    FOR051.DAT
COPY SYS$IDSP:FOR060.DAT    FOR060.DAT
```

FOR TEST PURPOSES, COMPILE IDSPIN.FOR TO OBTAIN IDSPIN.OBJ. IDSPIN IS AN "INPUT" ROUTINE THAT HAS BEEN INCLUDED TO ALLOW THE JUST INSTALLED IDSP SYSTEM TO BE TESTED. ALSO INCLUDED ARE THE RELATED DATASETS FOR051.DAT AND FOR060.DAT. FOR060.DAT IS READ BY IDSPIN. FOR051.DAT IS READ BY THE SETUP OPERATOR (SEE TM 83997 PAGES 6.24-1&2 FOR A DESCRIPTION OF FOR051.DAT). IDSPIN.FOR IS CURRENTLY SET UP TO PROVIDE PERIODIC SIMULATED DATA TO IDSP. THE CHARACTERISTICS OF THE DATA CAN BE CHANGED BY CHANGING FOR060.DAT AND/OR IDSPIN.FOR ITSELF. OF COURSE, IF YOU MODIFY IDSPIN.FOR YOU WILL HAVE TO RECOMPILE AND RELINK.

10) EXECUTE THE COMMAND PROCEDURE IDSP.COM FOUND IN THE DIRECTORY EQUATED TO BY SYS$IDSP. SEE PAGE 4.0-1 (ITEM #4) OF THE TM 83997 ON THE IDSP PACKAGE FOR INSTRUCTIONS ON EXECUTING THE COMMAND PROCEDURE. EXECUTED AS FOLLOWS: (AN EXAMPLE FOLLOWS):

```
@SYS$IDSP:IDSP DRA2:[IDSPTAPE.IDSP]IDSPIN
```

11) AFTER THE IDSP SYSTEM HAS BEEN LINKED, YOU WILL SEE "ENTER COMMAND" ON THE SCREEN. THE FIRST COMMAND ENTERED IN THIS TEST SHOULD BE "SETUP" **(NOTE THE IDSP COMMANDS SHOULD BE IN UPPERCASE)** WHICH WILL READ THE TEST DATA INTO IDSP. THEN DO A "SHOW SETUP D" TO DISPLAY THE DATA. CONTINUE TO EXECUTE VARIOUS IDSP COMMANDS, PERHAPS THE COMMAND SEQUENCE SHOWN BELOW (SEE SECTION 6 OF TM 83997 FOR A DESCRIPTION OF THESE OPERATORS). IT IS ALSO INSTRUCTIVE TO EXECUTE THE IDSP "DCL" COMMAND AND THEN DO A "DIR" TO SEE THE FILES THAT HAVE BEEN GENERATED.
(NOTE TO EXIT IDSP ENTER THE COMMAND STOP).

```
SETUP
SHOW SETUP D
DSTAT SETUP
WINDOW SETUP 0
FFT WINDOW
SPECT FFT 1
EIG SPECTD SPECTOFF DSTAT
  .
  .
ETC.
  .
  .
STOP
```

ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX J - AN INPUT ROUTINE FOR GENERATION OF TEST DATA

```
      SUBROUTINE INPUT(BLOCK,TIME,MAXBLK, MEMBER,NSER
     1,DELTIM,PCNT,BAD,EOF)
      COMMON/HIST1/HISTIN
      CHARACTER*80 HISTIN
      REAL*4 BLOCK(256,8)
      REAL*8 TIME(256),XTIME,DSEED
      DATA TWOPI/6.2831853/,F1/20./,F2/18./,A1/50./,A2/1./
     1PHI1/0.0/,PHI2/0.0/,A3/10./
      LOGICAL EOF,INIT/.TRUE./
      HISTIN='DYANMIC RANGE AND RESOLUTION TEST'
      IF(INIT.EQ. FALSE.)GOTO200
      READ(60,1000)A1,A2,A3,F1,F2,F3,PHI1,PHI2,PHI3
 1000 FORMAT(9F5.0)
      MEMBER=256
      NSER=3
      DSEED=123457.D0
      DELTIM=.040
      PCNT=0.001
      T=0.0
      CALL DATTIM(XTIME,1981,90,0,0,0,0)
  200 CONTINUE
      DO 100 I=1,MEMBER
      TIME(I)=T+XTIME
      BLOCK(I,1)=A1*COS((TWOPI*F1)*T+PHI1)
     1+           A2*COS((TWOPI*F2)*T+PHI2)
     2+           A3*COS((TWOPI*F3)*T+PHI3)
C       RAND=( A3*GGUBFS(DSEED))
C       RAND=SIGN(RAND,BLOCK(I,1))
C       BLOCK(I,1)=BLOCK(I,1)+RAND
      BLOCK(I,2)=A2*SIN((TWOPI*F2)*T)-A2*SIN((TWOPI*F3)*T)
      BLOCK(I,3)=1.0
      BLOCK(I,4)=BLOCK(I,1)
      T=T+DELTIM
  100 CONTINUE
      INIT=.FALSE.
      RETURN
      END
```

```
FOR051.DAT
1981,90,0,0,0,0
3,20.,1
1981,90,0,2,4,0
```

```
FOR060.DAT
0.10 0.20 0.40 4.70 7.80 10.9
```

ABSTRACT

The Interactive Digital Signal Processor (IDSP) is implemented on a VAX 11/780 under VMS. It consists of a set of time series analysis "Operators" each of which operates on an input file to produce an output file; the operators can be executed in any order that makes sense and recursively, if desired. The operators are the various algorithms that have been used in digital time series analysis work over the years. In addition, there is provision for user written operators to be easily interfaced to the system. The system can be operated both interactively and in batch mode.

In IDSP a file can consist of up to n (currently n=8) simultaneous time series. Thus storage for a file can be subdivided such that it is used, for example, entirely for one long single time series or for as many as n shorter time series, such as the components of a vector. An operator always operates simultaneously on all of the time series in a file.

IDSP currently includes over thirty standard operators that range from Fourier transform operations (FFT,FFTIN,WINDOW,SPECT), design and application of digital filters (FILDES,FILOPT,FILTER,WTS), eigenvalue analysis (EIG), to operators that provide graphical output (GRAPH,GRAFCK), allow batch operation (REDO), editing (CONCAT,EDHIST,EDIT,INTERP,SUBSET,SUBSER) and display information (SHOW, CMDHIS). The complete set of standard operators is listed below.

AVER, CMDHIS, CONCAT, COPY, DCL, DSTAT, EDHIST, EDIT, EIG, FILDES, FILOPT, FILTER, FFT, FFTIN, FLOP, GRAFCK, GRAPH, INTERP, MEAN, MNFLD, NORM, RECPOL, REDO, ROTATE, SETUP, SHOW, SPECT, SUBSER, SUBSET, TRACE, WINDOW, WTS, STOP.

IDSP is being used extensively to process data sets obtained from scientific experiments onboard spacecraft such as Dynamics Explorer, ISEE, IMP and Voyager. In addition IDSP provides an excellent teaching tool for demonstrating the application of the various time series operators to artifically-generated signals.

IDSP is available from the Computer Software Management and Information Center (COSMIC), 112 Barrow Hall, University of Georgia, Athens, Georgia 30602, Program Number GSC-12862.