

LEVEL II

12
10.9

ADA083021

CHARACTERIZATION OF TITAN III-D ACOUSTIC PRESSURE SPECTRA
BY LEAST-SQUARES FIT TO THEORETICAL MODEL

Eugene B. Hartnett
Eric Carleen

Boston College
Chestnut Hill, MA 02167

January 1980

Scientific Report No. 2

THIS DOCUMENT IS BEST QUALITY PRACTICAL.
THE COPY FURNISHED TO DDC CONTAINED A
SIGNIFICANT NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.

Approved for public release; distribution unlimited

AIR FORCE GEOPHYSICS LABORATORY
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
HANSCOM AFB, MASS. 01731

DTIC
ELECTRIC
S APR 14 1980
E

DDC FILE COPY

80 4 14 010

Qualified requestors may obtain additional copies from the Defense Documentation Center. All others should apply to the National Technical Information Service.

DISCLAIMER NOTICE

**THIS DOCUMENT IS BEST QUALITY
PRACTICABLE. THE COPY FURNISHED
TO DTIC CONTAINED A SIGNIFICANT
NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.**

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

19 REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM	
1. REPORT NUMBER AFGL-TR-80-0004	2. GOVT ACCESSION NO.	3. REPORT TYPE & CATALOG NUMBER Interim Rept.	4. TITLE (and Subtitle) CHARACTERIZATION OF TITAN III-D ACOUSTIC PRESSURE SPECTRA BY LEAST-SQUARES FIT TO THEORETICAL MODEL
5. AUTHOR(s) Eugene B. Hartnett/ Eric Carleen	6. CONTRACT OR GRANT NUMBER(s) F19628-78-C-0008	7. PERFORMING ORGANIZATION NAME AND ADDRESS Boston College Chestnut Hill, MA 02167	8. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62010F 7600-09-02
9. CONTROLLING OFFICE NAME AND ADDRESS Air Force Geophysics Laboratory Hanscom Air Force Base, MA 01731 Contract Manager: Henry A. Ossing/LWH	10. REPORT DATE January 1980	11. NUMBER OF PAGES 58	12. SECURITY CLASS. (of this report) Unclassified
13. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 591	14. DECLASSIFICATION/DOWNGRADING SCHEDULE	15. DISTRIBUTION STATEMENT (of this Report) Approved for public release, Distribution unlimited	
16. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)			
17. SUPPLEMENTARY NOTES			
18. KEY WORDS (Continue on reverse side if necessary and identify by block number) rocket plume acoustics, power spectral density, least squares fit			
19. ABSTRACT (Continue on reverse side if necessary and identify by block number) A theoretical model for the acoustic spectra of undeflected rocket plumes is fitted to computed spectra of a Titan III-D at varying times after ignition, by a least-squares method. Tests for the goodness of the fit are made.			

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

061150

11

TABLE OF FIGURES

Fig.		Page
1	Instrument Array	11
2	Periodogram of Noise (1 min. before launch)	12
3	Pressure Detector System Response	13
4a	Spectra 3.83 seconds after launch	14
4b	Chi-Squared test " " " "	15
5a	Spectra 5.39 " " "	16
5b	Chi-Squared test " " " "	17
6a	Spectra 6.95 " " "	18
6b	Chi-Squared test " " " "	19
7a	Spectra 8.51 " " "	20
7b	Chi-Squared test " " " "	21
8a	Spectra 10.07 " " "	22
8b	Chi-Squared test " " " "	23
9a	Spectra 11.63 " " "	24
9b	Chi-Squared test " " " "	25
10a	Spectra 13.19 " " "	26
10b	Chi-Squared test " " " "	27
11a	Spectra 14.75 " " "	28
11b	Chi-Squared test " " " "	29
12a	Spectra 16.31 " " "	30
12b	Chi-Squared test " " " "	31

INTRODUCTION

Boston College, in its investigation of the seismic effects of rocket launchings on structures in the immediate environment, is analyzing data taken during a Titan III-D launch at Vandenberg AFB in March 1979. The work reported herein deals with the spectral form of the induced surface pressure. A theoretical model is fitted to the power spectral density of the observed pressure at various times during launch. The results are presented along with tests of their validity.

Accession For	
NTIS GRA&I	☒
DDC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Classification Codes	
Dist	Author/Or Special
A	23 CP

CHARACTERIZING PRESSURE SPECTRA

The data are contained on a file (LAUNC2.MNR) which begins 1.64 seconds before ignition and continues for 61.44 seconds at 100 samples per second on each of 16 channels. The array of instruments corresponding to these channels is shown in Fig. 1. For this report only the pressure sensor on channel 10 which is 950 feet from the launch point was used.

To obtain estimates of the power spectral density at a given time after ignition 256 samples were read in beginning at that time. These samples had been digitized at 204.8 counts per volt. The power spectra were computed by the periodogram technique employing an FFT algorithm. This periodogram is such that $\frac{1}{N} \sum_{i=1}^N x_i^2 = \sum_{k=1}^N S_k$.

It was found that the noise caused by the system and ambient pressure fluctuations was white and assumed to be additive above the quantization level. It was estimated by averaging 20 periodograms starting 1 minute before launch and found to average $9.37(10^{-8})$ volts² per cell (Fig. 2). This was subtracted from the rocket spectra to yield our best estimate of rocket induced surface pressure observed through our system. To convert these spectra to the spectra of the pressure appearing at the input, they were divided by the squared magnitude of the system response (Fig. 3). That is $S_k^{in} = \frac{S_k^{out}}{H_k^2}$. These were scaled by $1/\Delta f$ at 0hz and

the Nyquist frequency and $2/\Delta f$ elsewhere so the resultant

power spectral density when integrated by trapezoidal rule from 0hz to the Nyquist frequency equals the mean square of the input. However since the value at 0hz is dependent on amplifier drift its value is discarded and to reduce contamination from errors in our estimate of system behavior near the Nyquist frequency only frequencies less than 40hz were considered.

From physical considerations (1) and experimental studies (2) it is believed that the power spectral density of the surface pressure caused by undeflected chemical rocket plumes is of the form:

$$P(w) = \frac{4}{\pi} \frac{P_{\max}}{W_0} \left\{ \frac{W}{W_0} + \frac{W_0}{W} \right\}^{-2}$$

$$\text{or } P(f) = \frac{4}{\pi} \frac{P_0}{f_0} \left\{ \frac{f}{f_0} + \frac{f_0}{f} \right\}^{-2}$$

It was desired to obtain values of P_0 and f_0 which minimized the sum of the squared differences between the observed power spectral density and the theoretical as described by the above equation. To find the sum of squared errors over the range of the power spectral density:

$$E = \sum_{k=1}^{129} (P'_k - P(f_k))^2$$

where P'_k is our estimate of the k^{th} value of the power spectral density and $f_k = 100(k-1)/256$ which is the frequency represented by the k^{th} values of the power spectral density.

For reasons already stated the full range of the power spectral density was not to be used so the limits of the

summation were changed accordingly.

$$E = \sum_{k=2}^{103} (P'_k - P(f_k))^2$$

Rather than finding both P_0 and f_0 by trial and error the derivative of E with respect to P_0 is taken and set to zero that is where the extremal would be found.

$$\frac{dE}{dP_0} = 0 = \sum_{k=2}^{103} 2 \left(P'_k - \frac{4P_0}{\pi f_0} \left\{ \frac{f_k}{f_0} + \frac{f_0}{f_k} \right\}^{-2} \right) \left(-\frac{4}{\pi} f_0 \left\{ \frac{f_k}{f_0} + \frac{f_0}{f_k} \right\}^{-2} \right)$$

$$P_0 = \frac{\pi f_0}{4} \frac{\sum_{k=2}^{103} P'_k \left(\frac{f_k}{f_0} + \frac{f_0}{f_k} \right)^{-2}}{\sum_{k=2}^{103} \left(\frac{f_k}{f_0} + \frac{f_0}{f_k} \right)^{-4}}$$

So for any value of f_0 the value of P_0 which insures minimum error in the least square sense is uniquely determined.

To find the value of f_0 which yields the smallest E (the 'best' f_0) an iteration scheme was devised. The initial search interval has f_2 and f_{103} as its endpoints in the belief that f_0 lies between them. Five trial values of f_0 are considered starting with the low endpoint of the search interval and increasing in equal steps to the high endpoint. For each trial value of f_0 , P_0 and E are computed by the formulae above and the best f_0 selected. A new search interval is defined with endpoints equal to the trial values adjacent to the best f_0 during previous search and a new estimate of the best f_0 found. The process is repeated until f_0 is determined to within the limits of the computer's accuracy. In this case single precision yields about seven significant digits.

The resultant f_0 and P_0 were used to generate plots which show power spectral density based on observed values and theoretical values vs. normalized frequency, f/f_0 .

Table I shows f_0 and P_0 for each segment.

In order to compare average observed power with theoretical power the theoretical power spectral density was integrated from $-\infty$ to ∞ .

$$\begin{aligned} P_t = \int_{-\infty}^{\infty} P(f) df &= \frac{4}{\pi} \frac{P_0}{f_0} \int_{-\infty}^{\infty} \left(\frac{f}{f_0} + \frac{f_0}{f} \right)^{-2} df \\ &= \frac{4}{\pi} P_0 f_0 \int_{-\infty}^{\infty} \frac{f^2}{(f^2 + f_0^2)^2} df \\ &= \frac{4P_0 f_0}{\pi} \left\{ -\frac{1}{2} \frac{f}{(f^2 + f_0^2)} \right\}_{-\infty}^{\infty} + \frac{1}{2} \int_{-\infty}^{\infty} \frac{1}{(f^2 + f_0^2)} df \quad (3) \end{aligned}$$

$$= \frac{2P_0 f_0}{\pi} \int_{-\infty}^{\infty} \frac{1}{(f^2 + f_0^2)} df = \frac{2P_0 f_0}{\pi} \left(\frac{\pi}{f_0} \right) = 2P_0 \quad (3)$$

The observed power ($\Delta \sum_{k=1}^9 P'_k$) and theoretical power are contained in Table I.

VALIDATION

Although the theoretical curves do not appear to closely fit the observed spectra it is known that the use of the periodogram with a large number of samples to estimate the spectrum produces results which fluctuate wildly (4). It was desired to test whether these fluctuations fall within expected limits when the theoretical power spectral density

is assumed true.

Oppenheim and Schafer (4) show the development of an expression for the variance of the estimated P'_k 's for a white Gaussian process and Hinich and Clay (5) state that the result is a good approximation for a wide variety of random processes. They state that the variance (σ^2) of the spectrum is approximately equal to its magnitude squared.

$$\sigma_k^2 = \frac{1}{M} \sum_{m=1}^M (P'_k{}^{(m)} - \bar{P}'_k)^2 = \bar{P}'_k{}^2$$

$$\text{or } \frac{\sigma_k^2}{\bar{P}'_k{}^2} = \frac{1}{M} \sum_{m=1}^M \frac{(P'_k{}^{(m)} - \bar{P}'_k)^2}{\bar{P}'_k{}^2}$$

A figure of merit is defined as $\frac{1}{102} \sum_{k=2}^{103} \frac{\sigma_k^2}{\bar{P}'_k{}^2}$

which with $M=1$ and $\bar{P}'_k$ assumed equal to $P(f_k)$ becomes

$$\frac{1}{102} \sum_{k=2}^{103} \frac{(P'_k - P(f_k))^2}{P^2(f_k)}$$

This quantity which should be approximately 1 is contained in Table I. In addition as has been stated in (5), for large N $\frac{P'_k}{P(f_k)}$ follows a chi-squared distribution with 2 degrees of freedom.

As an additional test of the validity of the fit this criteria was used. For each value of $A_k = \frac{2P'_k}{P(f_k)}$ its cumulative relative frequency was computed.

a_x = estimated probability that $A_k \leq x$.

The chi-squared distribution for 2 degrees of freedom is

$$a = \frac{1}{2} \int_0^{\mu} -\frac{x}{2} dx = 1 - e^{-\frac{\mu}{2}} \quad (6).$$

Therefore $\mu = -2 \ln(1-a)$

$$\text{and } \mu_x = -2 \ln(1-a_x)$$

For a good fit μ_x should be approximately equal to x .
 x has been plotted vs. μ_x for each segment.

RESULTS

Results are presented in Table I and plots which are figure 4 through 12. Plots (a) are of the observed spectra (points only) and fitted theoretical curves (solid lines). Plots (b) are A_k vs. μ_x as previously defined and labelled 'observed test statistic' and 'theoretical test statistic' respectively. The vertical axis is scaled to fit the maximum value of A_k although this point cannot be plotted since its corresponding theoretical value is infinitely large. The horizontal axis stops at 9.25 which is the 99th percentile for the 2 degree of freedom chi-squared distribution. The solid line represents $A_k = \mu_x$. When the observed points lie above the line it means we had larger values than we should have expected. When they lie below it means they are smaller than expected.

CONCLUSIONS

In order to determine bounds on the figure of merit for acceptable fits simulated data was used. A random number

subroutine was used to produce normally distributed variables. For each group of 256 numbers the FFT was taken and the magnitude multiplied by the square root of $P(f)$ for a set value of P_0 and f_0 . The inverse transform was taken and the result multiplied by a decaying exponential which modelled the envelope of the rocket data. Two hundred similarly produced groups of 256 variables were fitted with the theoretical power spectral density curve and statistics of the figures of merit accumulated.

The figures of merit for this simulation were found to have a mean of 1.07 and a standard deviation of .35. The minimum value obtained was .57 and the maximum was 2.98. It was decided to accept the fitted data if its figure of merit fell within these extrema.

The first fit, which began 3.83 seconds after ignition, produced a figure of merit of 214.6. This fit is rejected. It is believed that this early in the launch the plume was not undeflected and the theoretical curve does not apply. Figures of merit for subsequent fits fell within extrema criterion and these fits are accepted.

It was found that when f_0 was low (equal to 2.8 hz) the chi-squared test plot, while a straight line, lay above the $A_k = 11_x$ line, as happened on a number of the fitted segments, and the estimated value of f_0 was high (3.2hz). Evidently out estimates while acceptable according to the figure of merit criterion are biased toward the high frequencies.

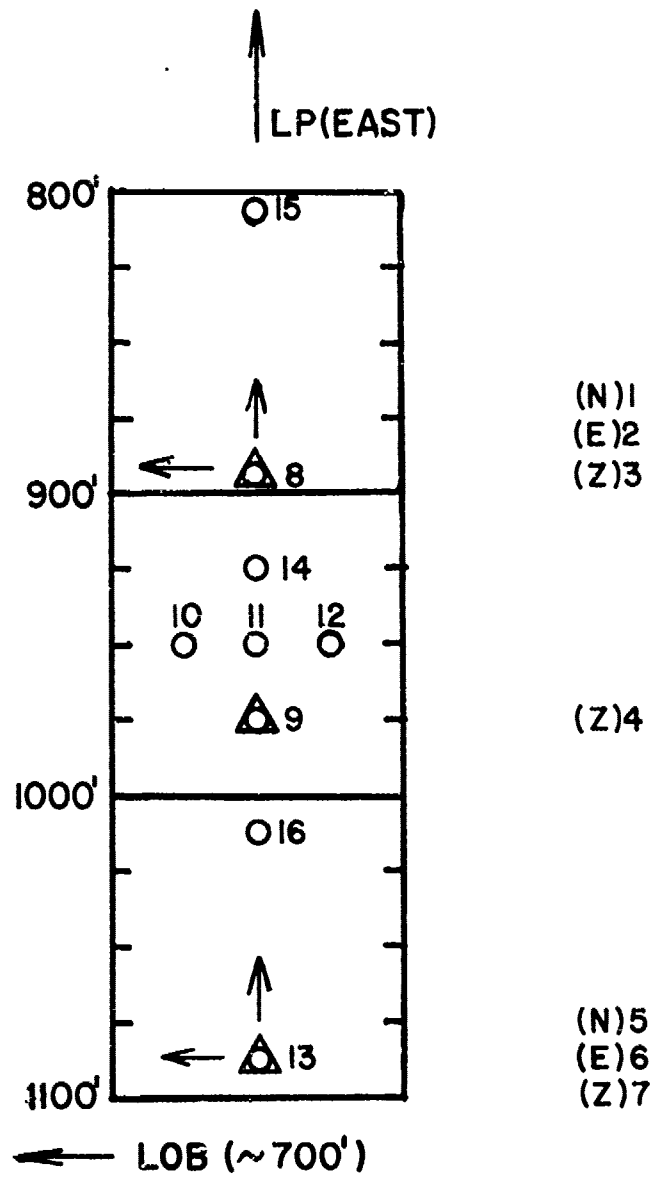
REFERENCES

1. A. Powell, Theory of Vortex Sound, Journal of the Acoustical Society of America, Vol. 36, No. 1, Jan. 1964.
2. Acoustic Loads Generated by the Propulsion System, NASA SP-8072 June 1971.
3. R.S. Burington, Handbook of Mathematical Tables and Formulas, Fifth Edition, McGraw-Hill Book Company, New York, 1973.
4. Alan V. Oppenheim and Ronald W. Schaffer, Digital Signal Processing, Prentice-Hall, Englewood Cliffs, N.J., 1975.
5. M.J. Hinich and C.S. Clay, The Application of the Discrete Fourier Transform in the Estimation of Power Spectra, Coherence, and Bispectra of Geophysical Data, Reviews of Geophysics, Vol. 6, No. 3, Aug. 1968.
6. D.B. Owen, Handbook of Statistical Tables, Addison-Wesley Publishing Co., Inc., Reading, Mass. 1962.

TABLE I

Time after Ignition (seconds)	f_o (hz)	P_o (psi ²)	$E((\frac{psi^2}{hz})^2)$	Observed Power (psi ²)	Theoretical ₂ Power (psi ²)	Figure of Merit
3.83	6.798	$8.477(10^{-5})$	$4.022(10^{-10})$	$7.931(10^{-5})$	$1.689(10^{-4})$	214.6
5.39	14.299	$6.741(10^{-4})$	$1.268(10^{-8})$	$3.875(10^{-4})$	$1.348(10^{-3})$	1.005
6.95	12.071	$1.606(10^{-3})$	$5.342(10^{-8})$	$1.007(10^{-3})$	$3.213(10^{-3})$	.620
8.51	11.190	$1.257(10^{-3})$	$6.848(10^{-8})$	$8.264(10^{-4})$	$2.514(10^{-3})$	1.381
10.07	7.548	$6.361(10^{-4})$	$1.648(10^{-8})$	$4.993(10^{-4})$	$1.272(10^{-3})$	1.528
11.63	5.235	$2.845(10^{-4})$	$5.418(10^{-9})$	$2.572(10^{-4})$	$5.690(10^{-4})$	1.813
13.19	6.097	$1.115(10^{-4})$	$7.147(10^{-10})$	$1.035(10^{-4})$	$2.230(10^{-4})$	1.405
14.75	4.390	$4.608(10^{-5})$	$2.198(10^{-10})$	$4.872(10^{-5})$	$9.216(10^{-5})$	2.59
16.31	2.855	$1.860(10^{-5})$	$9.203(10^{-11})$	$2.420(10^{-5})$	$3.719(10^{-5})$	1.032

VANDENBERG ARRAY



○ PRESSURE SENSORS (9)

△ SEISMOMETERS (7)

Fig. 1

VOLUME/CELL PERIODGRAM OF AMBIENT NOISE : 4IN. BEFORE LAUNCH (AVERAGE)

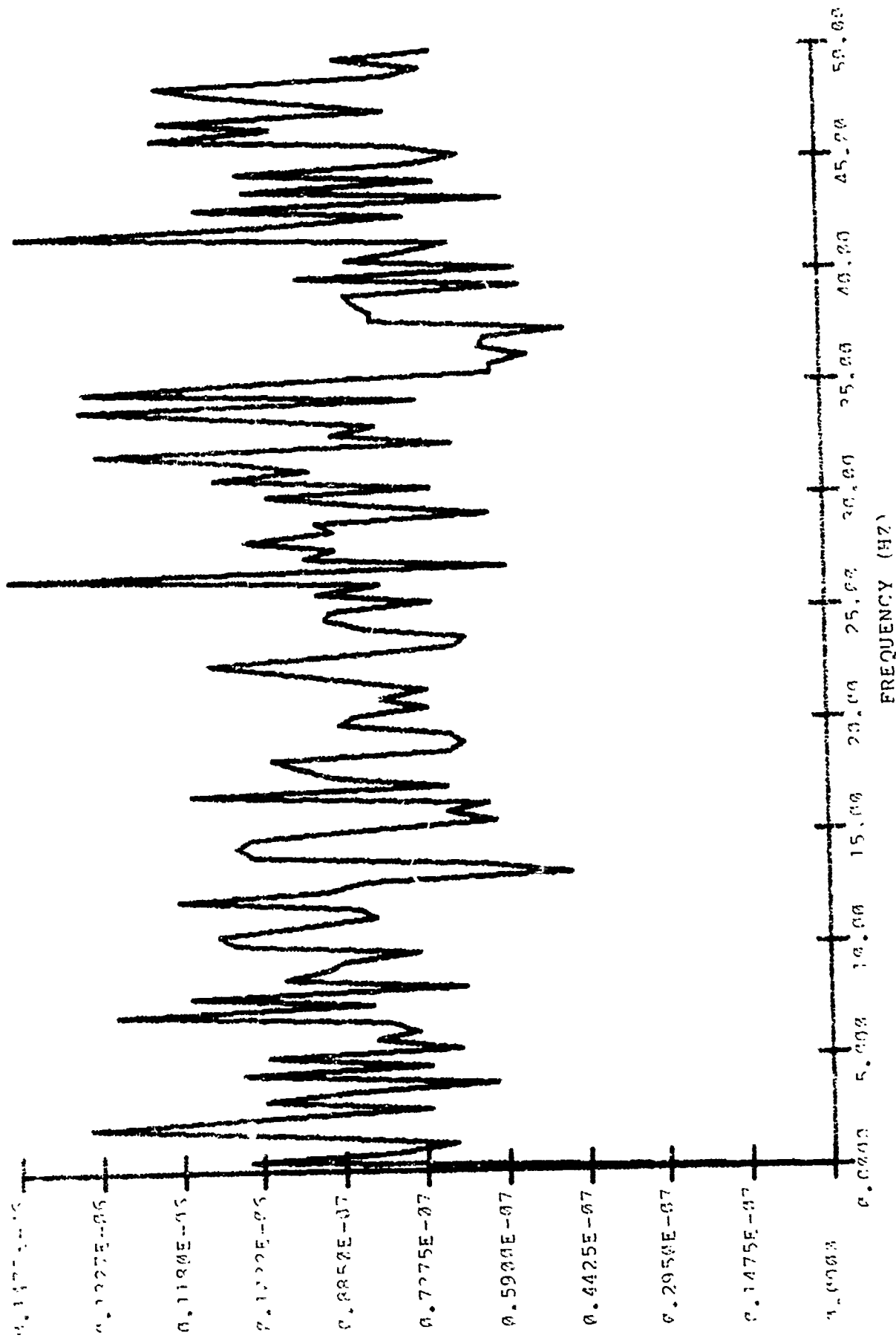


Fig. 2

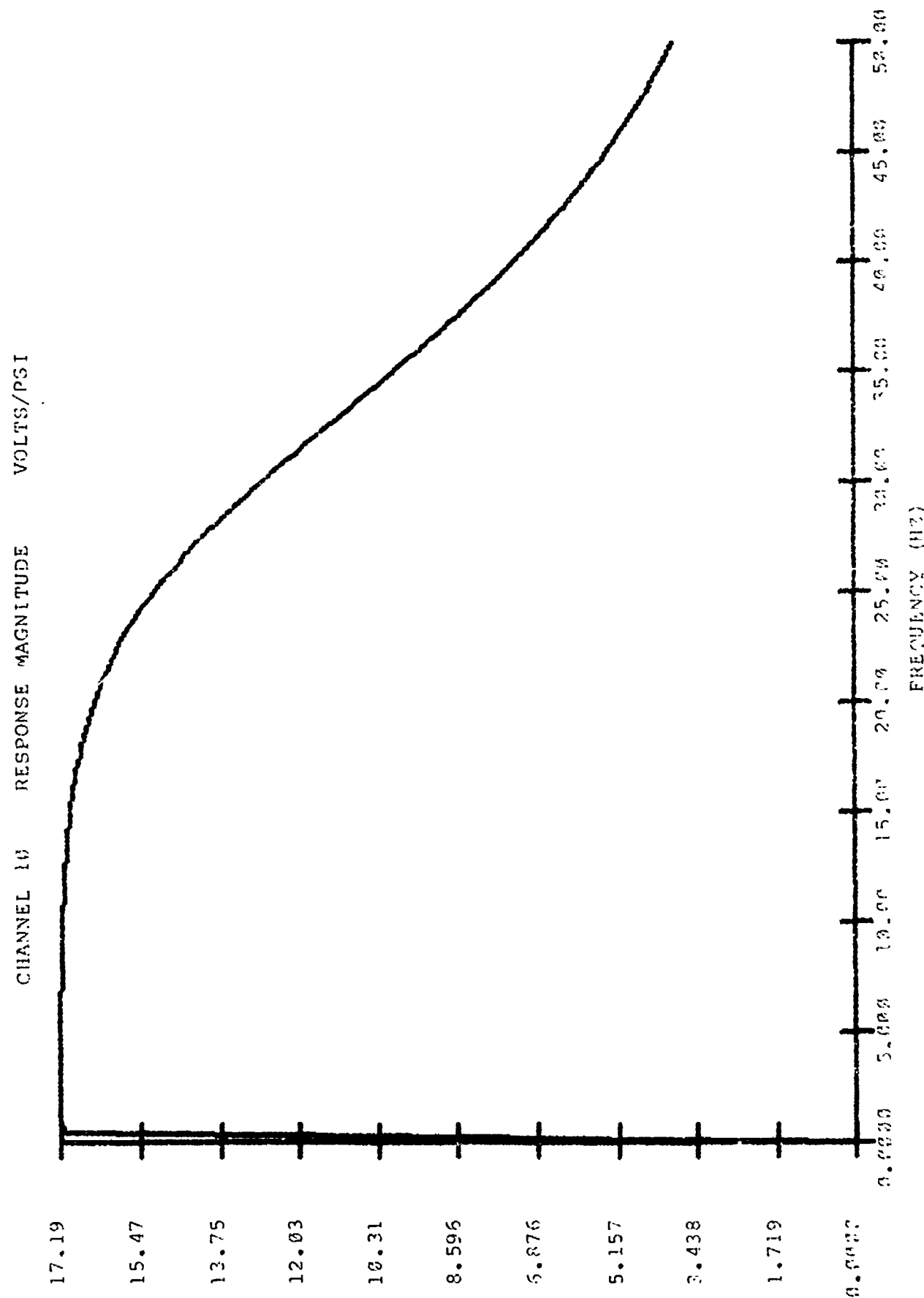


Fig. 3

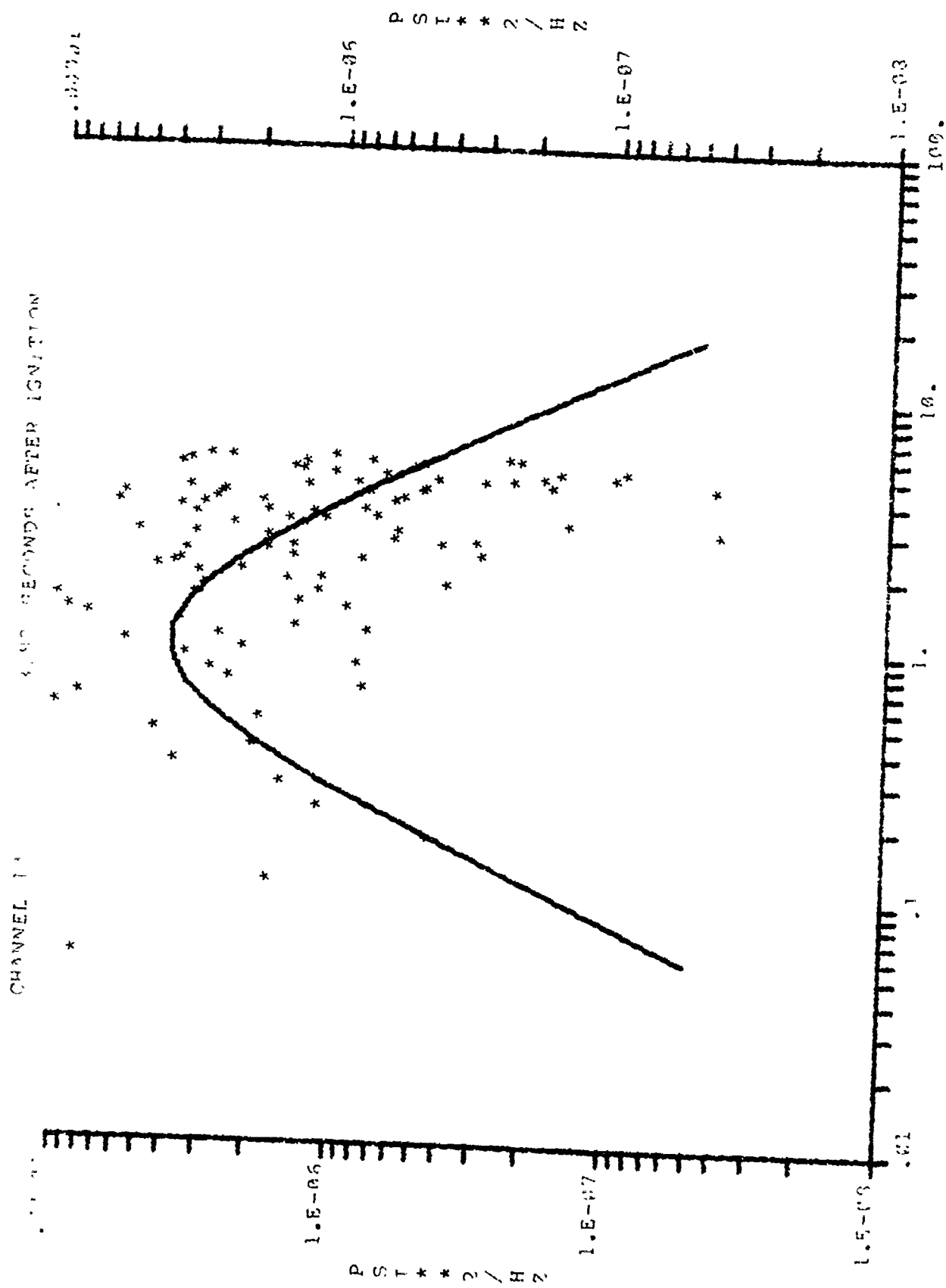
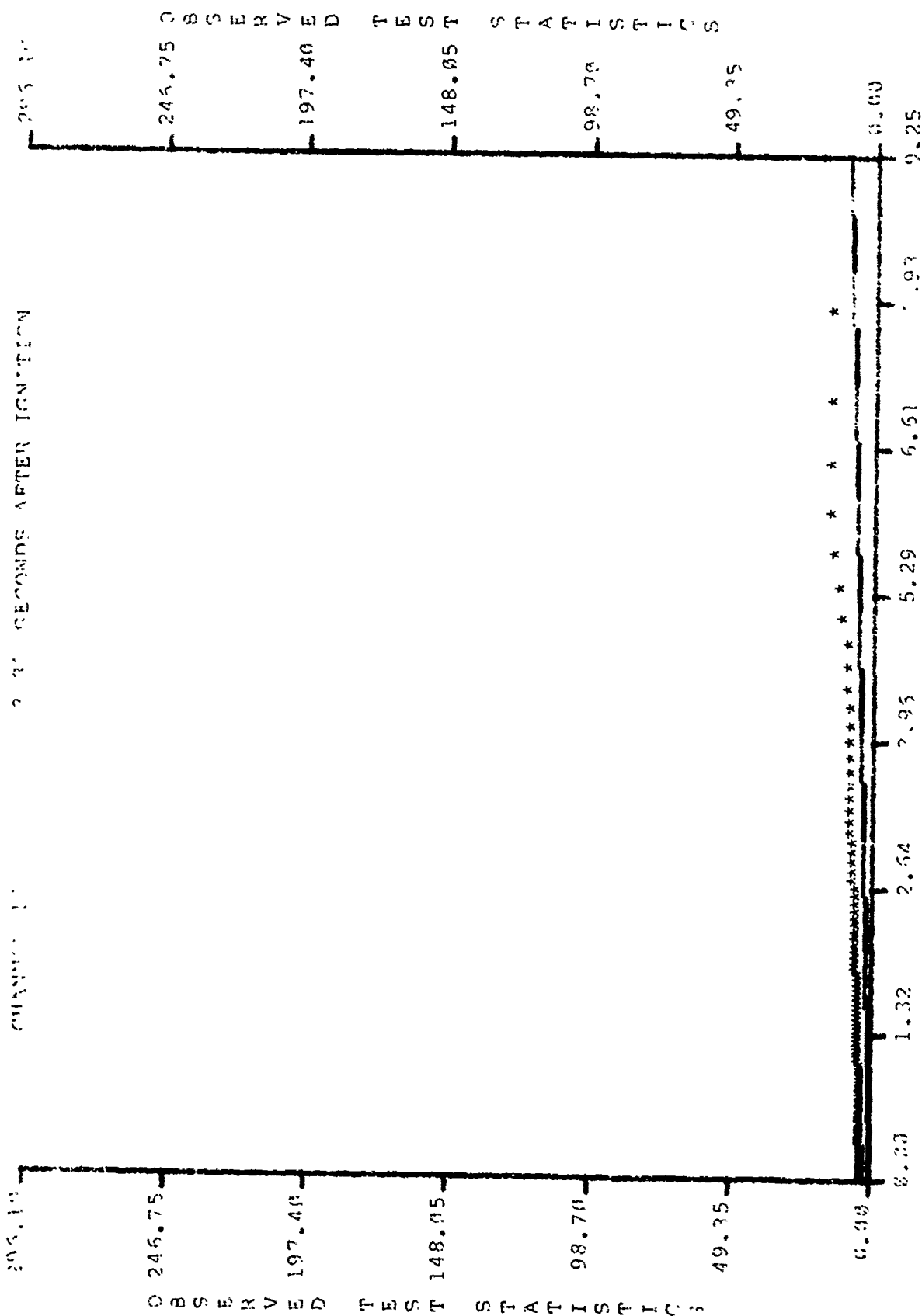


Fig. 4a

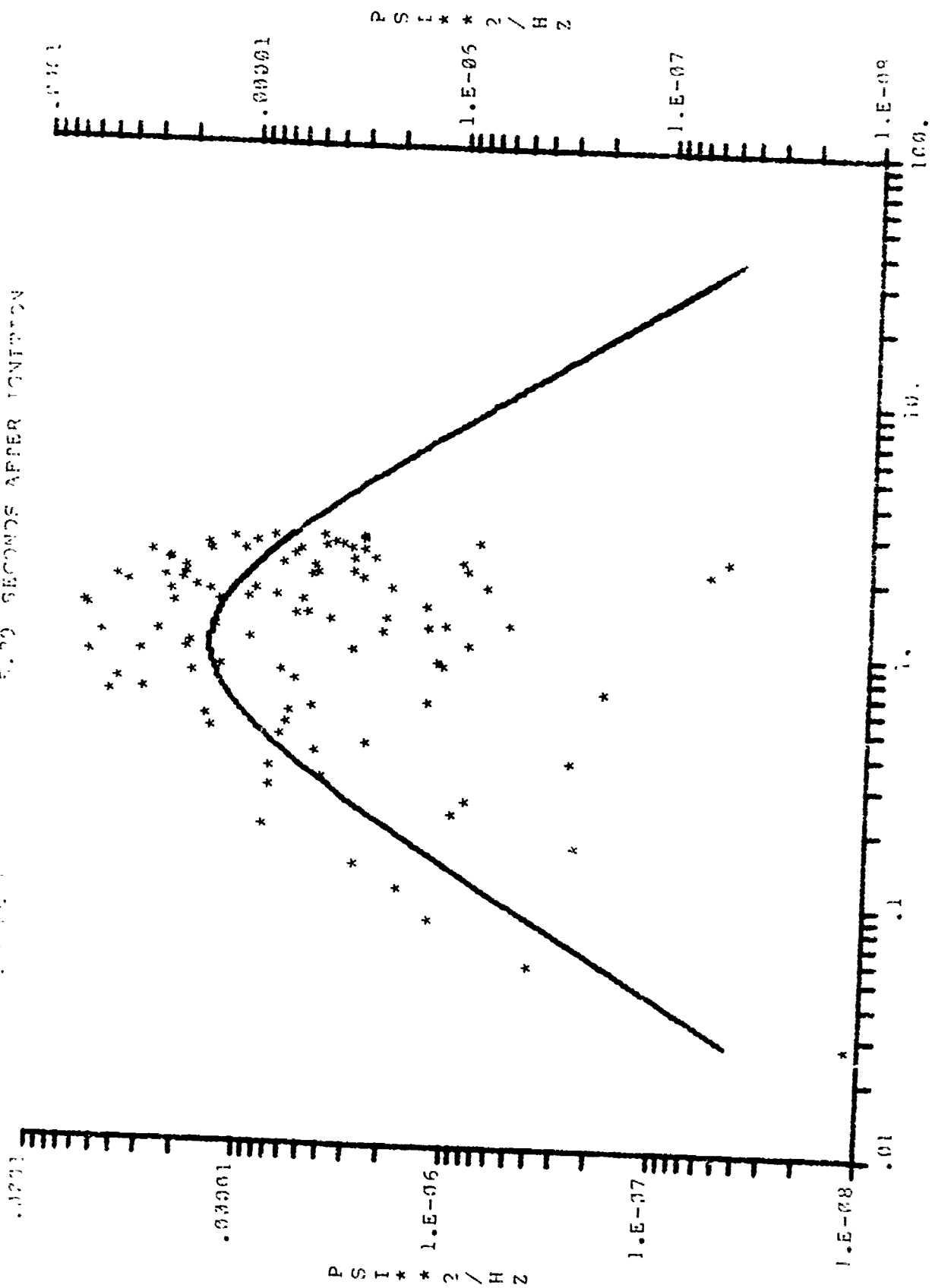


THEORETICAL TEST STATISTICS

Fig. 4b

5.00 SECONDS AFTER IGNITION

CHANNEL 1



NORMALIZED FREQUENCY

Fig. 5a

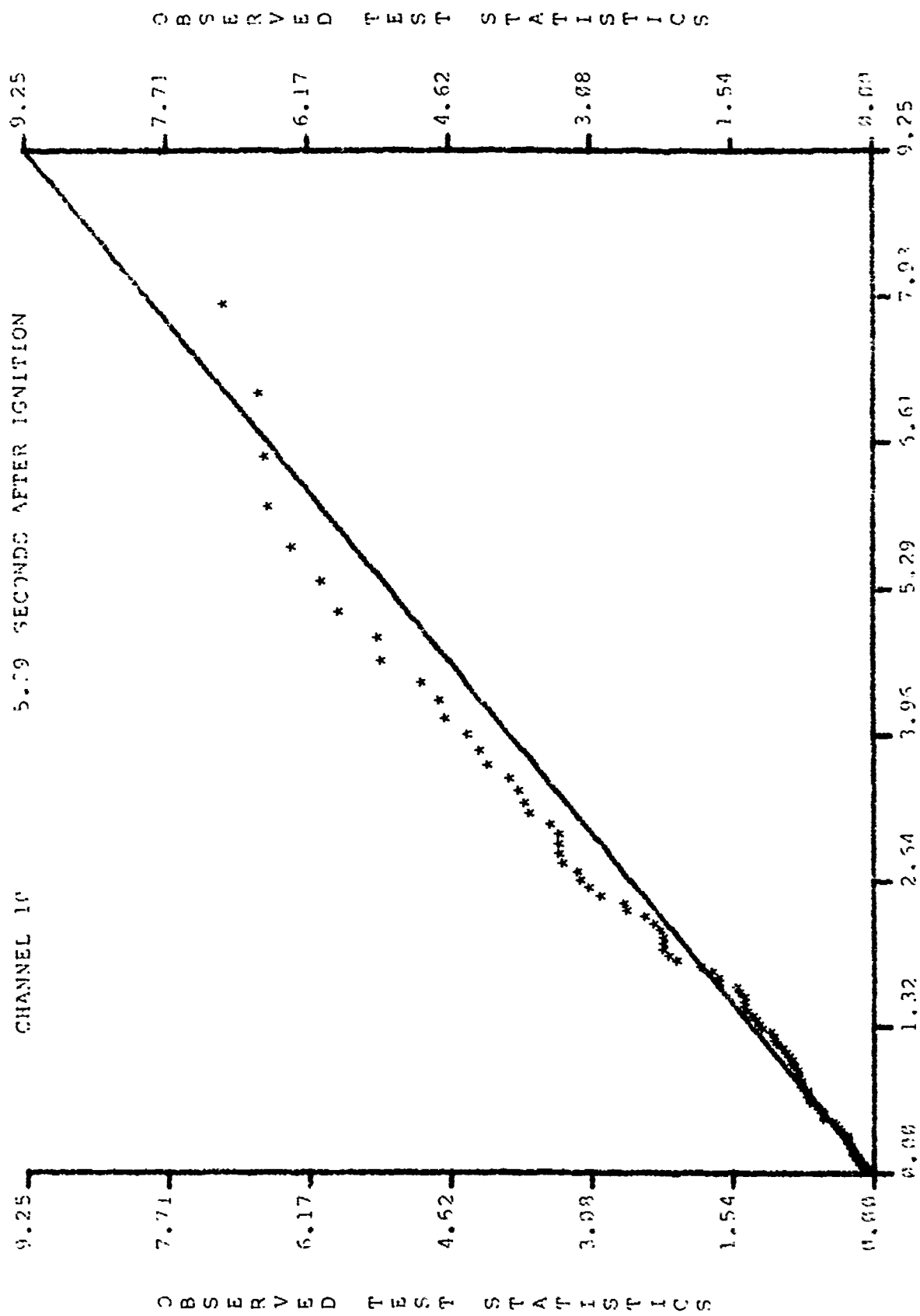


Fig. 5b

CHANNEL 10 5.95 SECONDS AFTER IGNITION

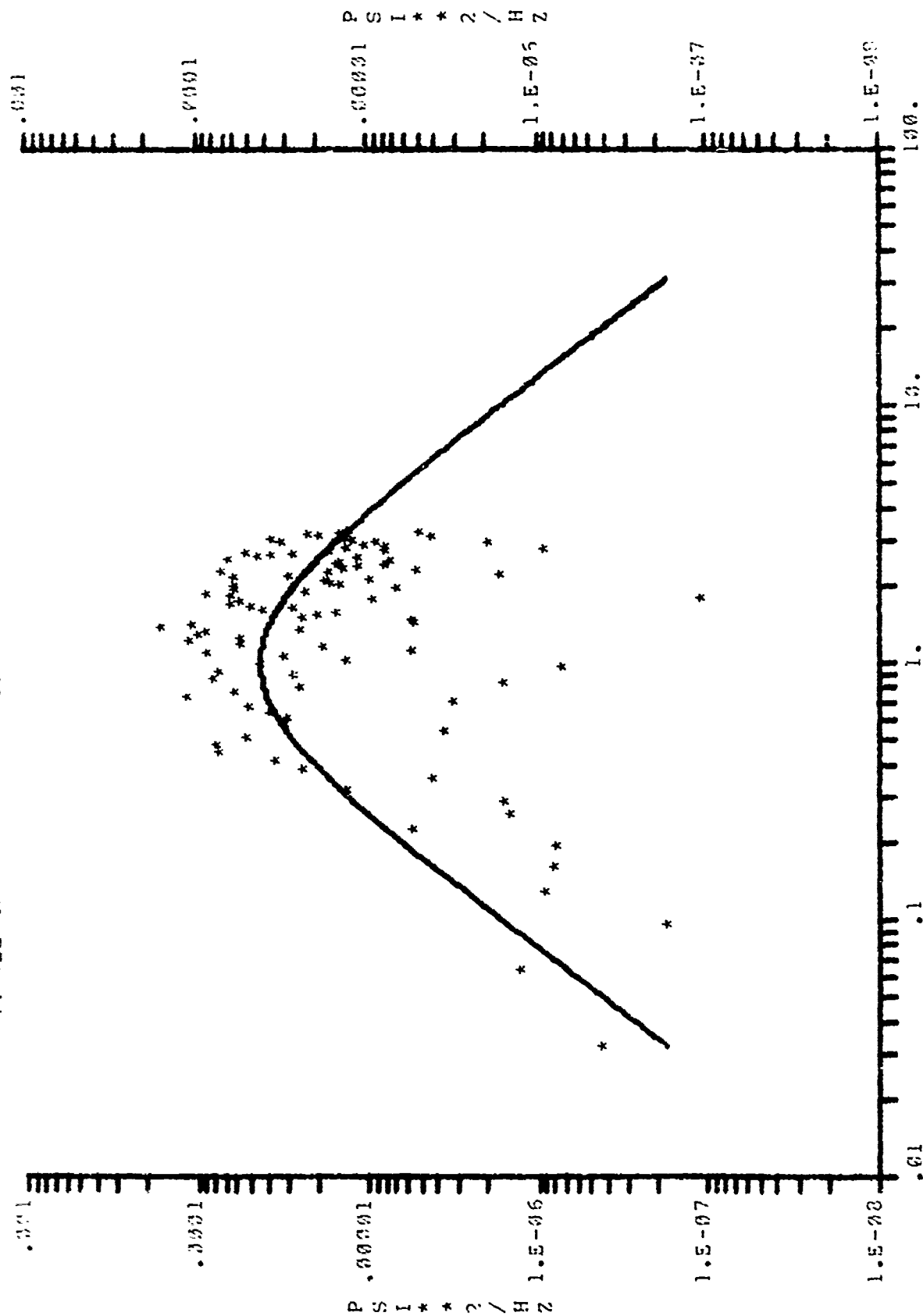
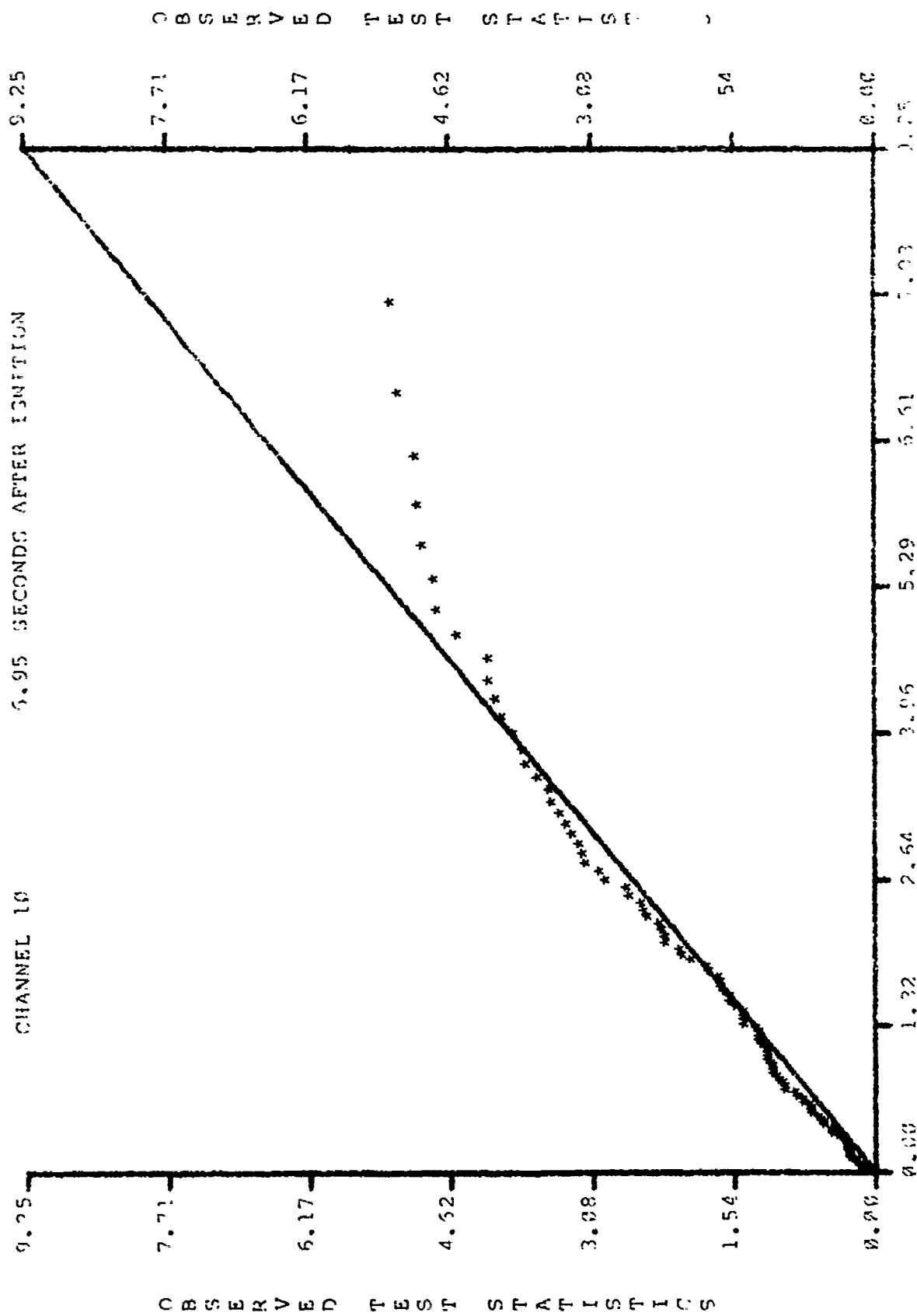


Fig. 6a



THEORETICAL TEST STATISTICS

Fig. 6b

CHANNEL 12

3.51 SECONDS AFTER IGNITION

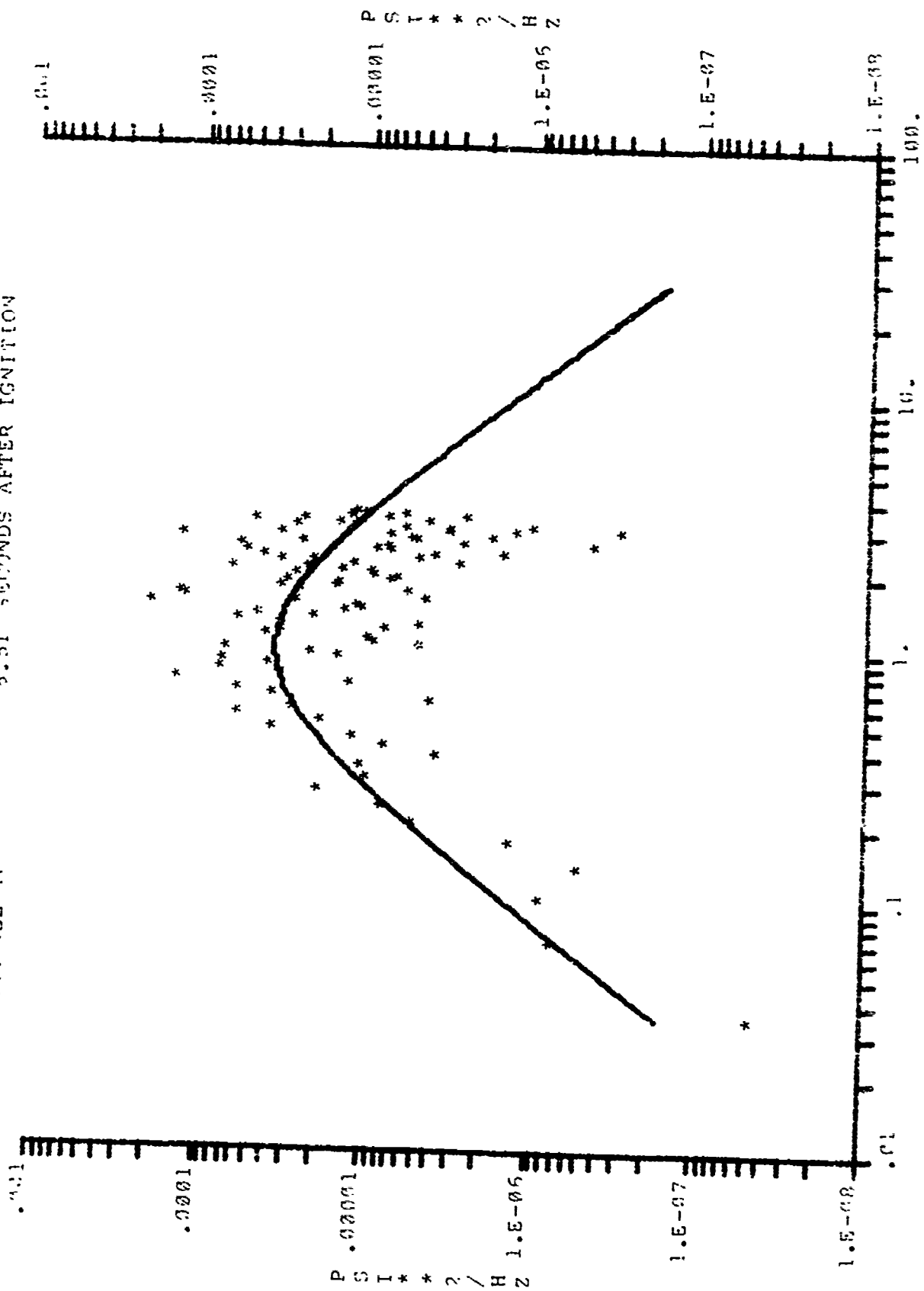


Fig. 7a

CHANNEL 13 13.07 SECONDS AFTER IGNITION

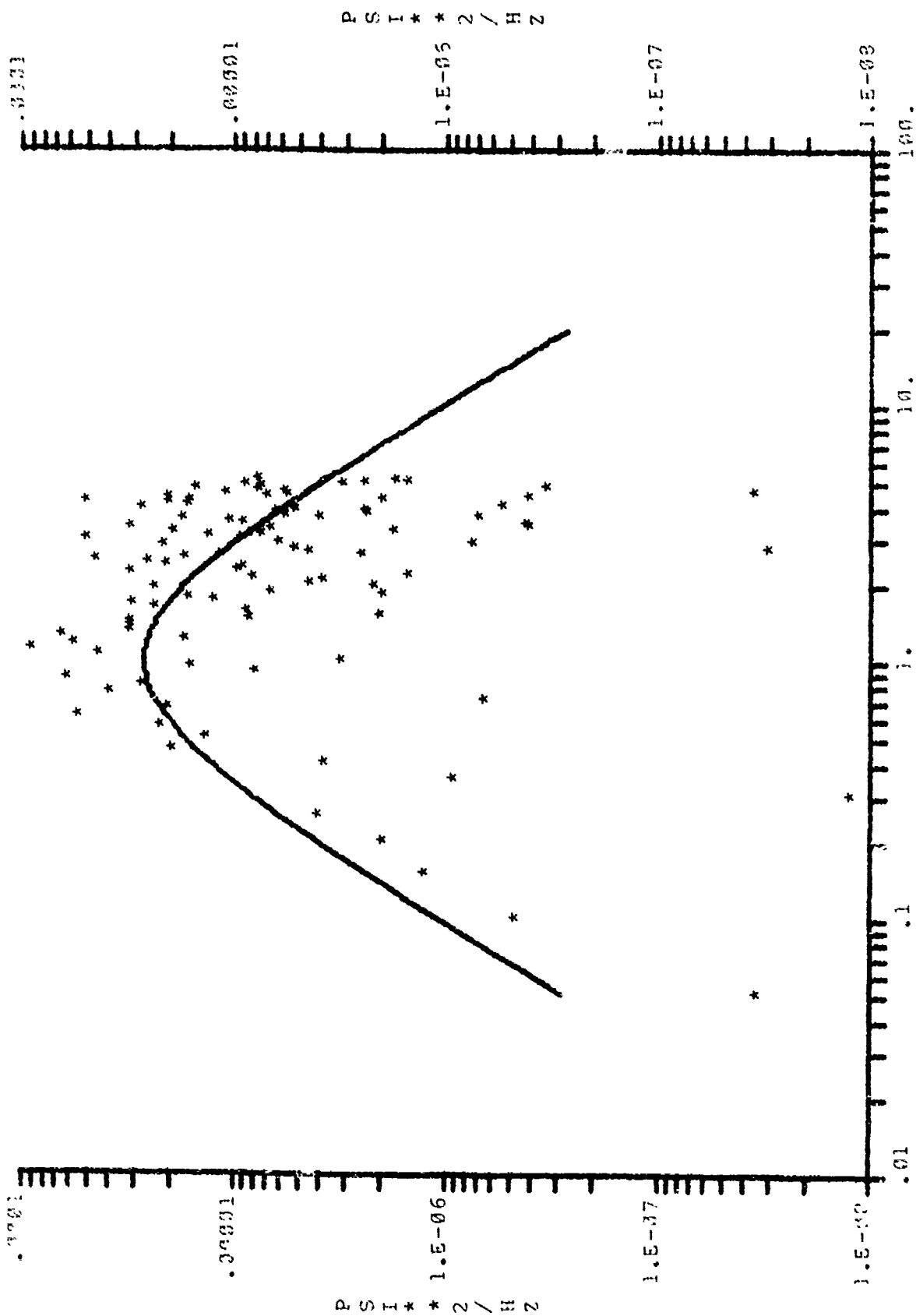
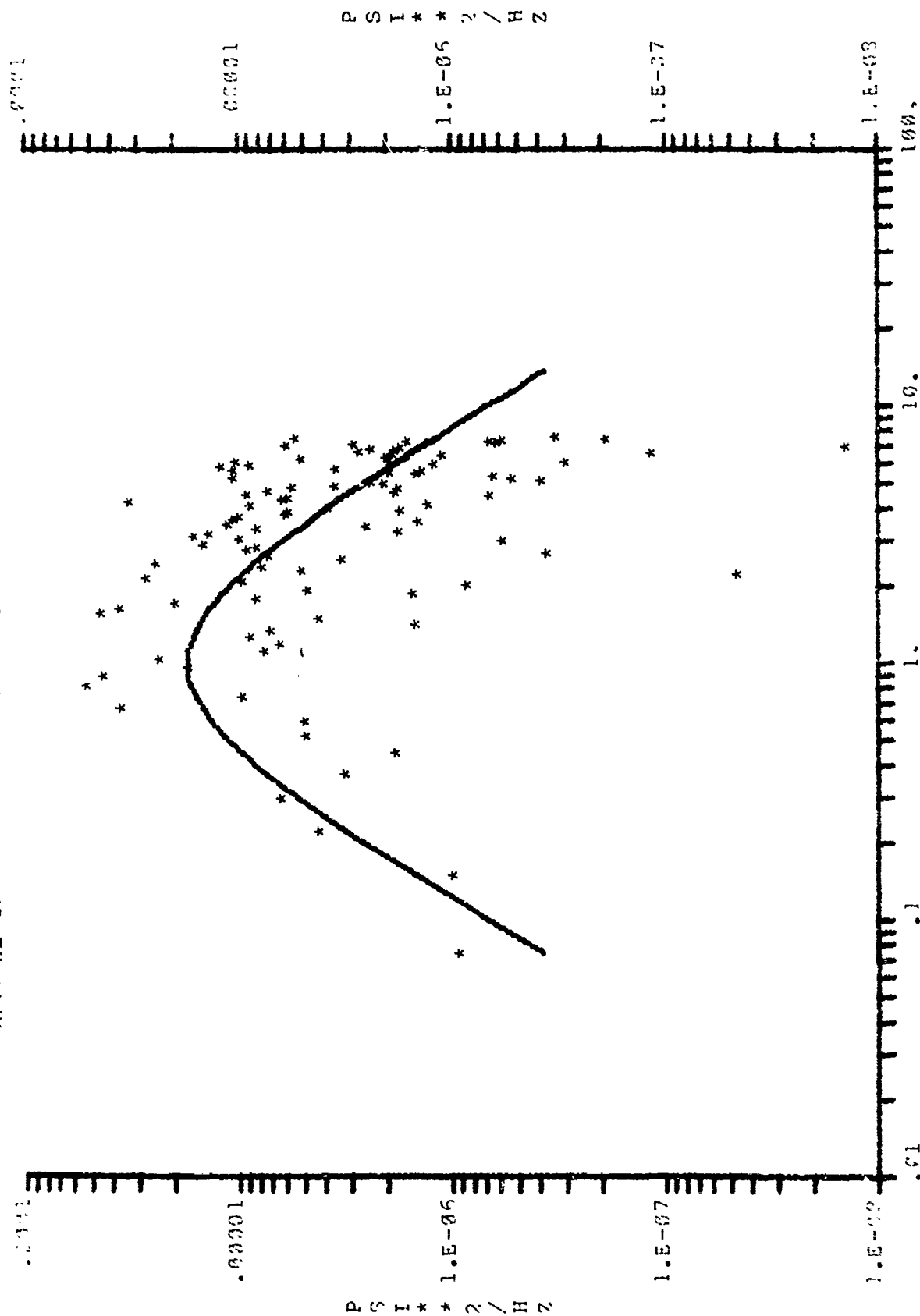


Fig. 8a

CHANNEL 17 11.52 SECONDS AFTER IGNITION



NORMALIZED FREQUENCY
Fig. 9a

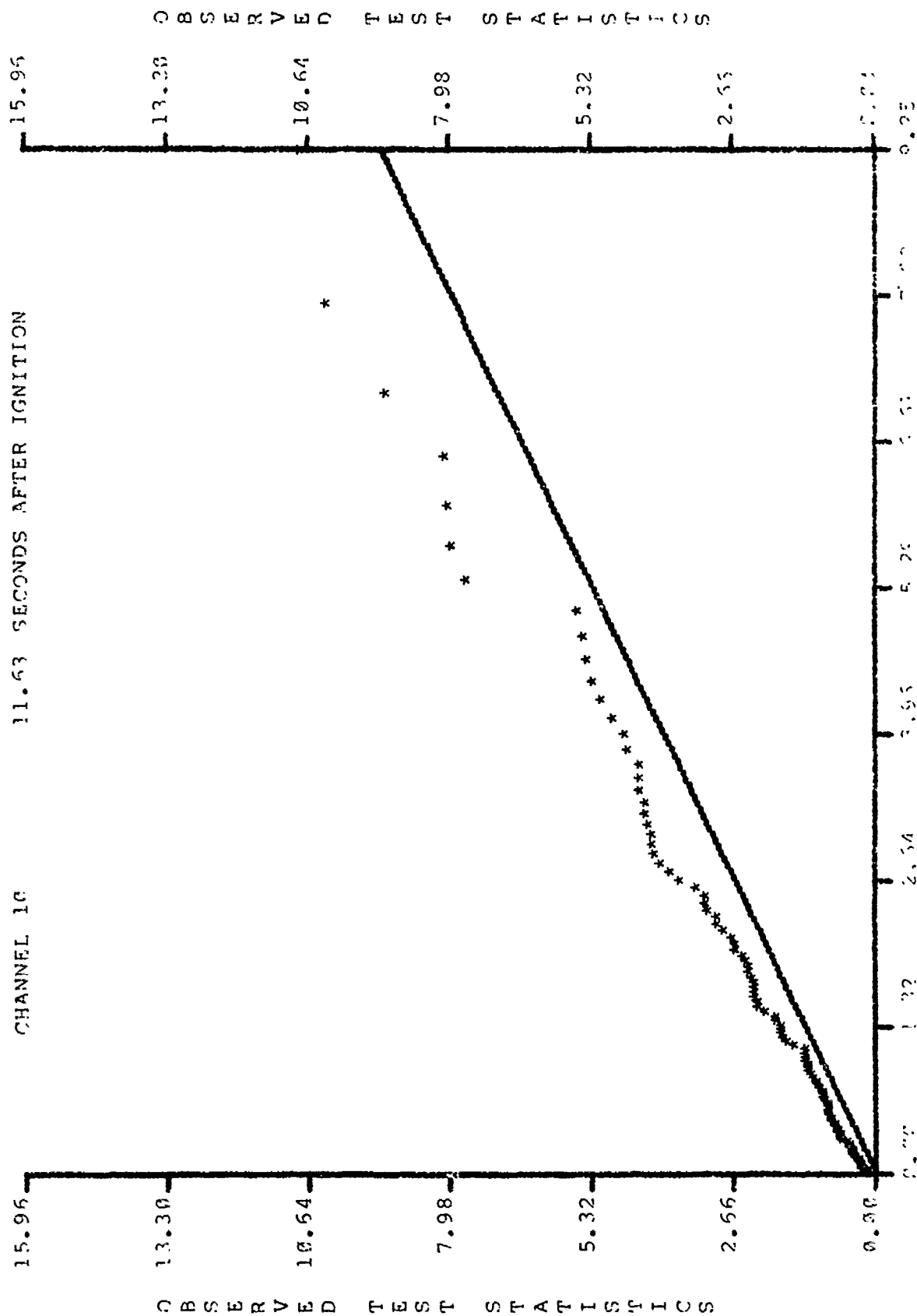


Fig. 9b

CHANNEL 1' 12.19 SECONDS AFTER IGNITION

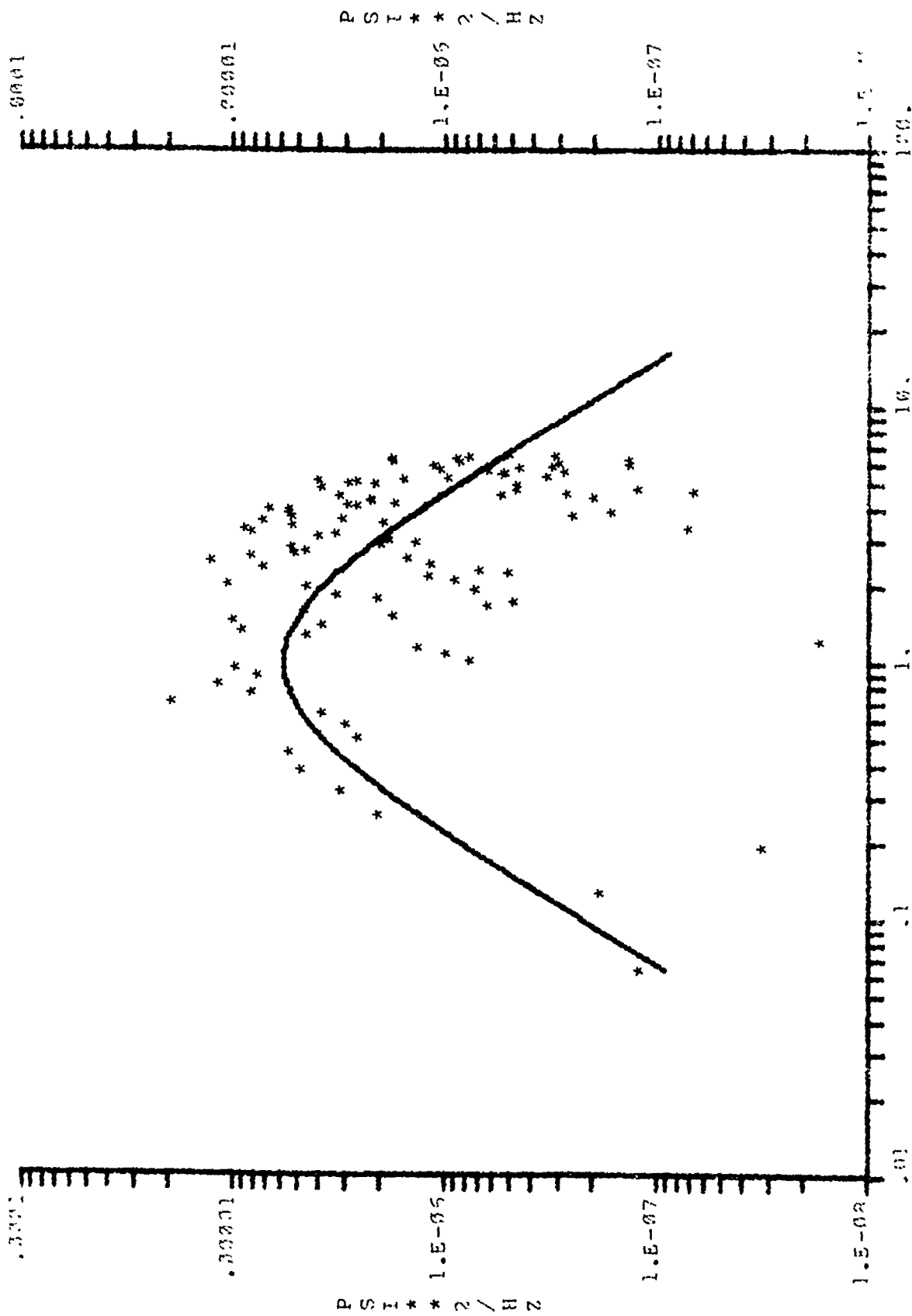
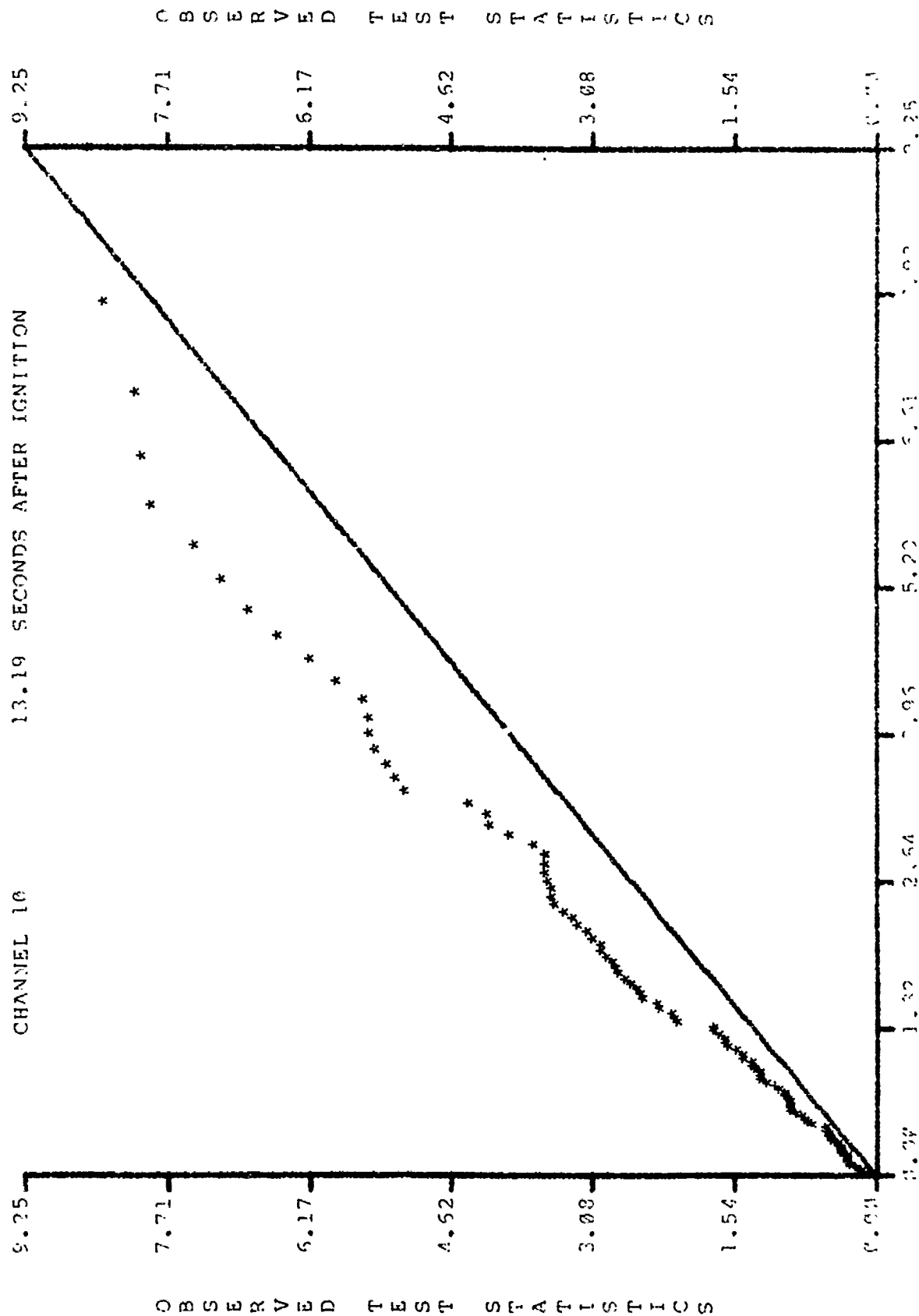
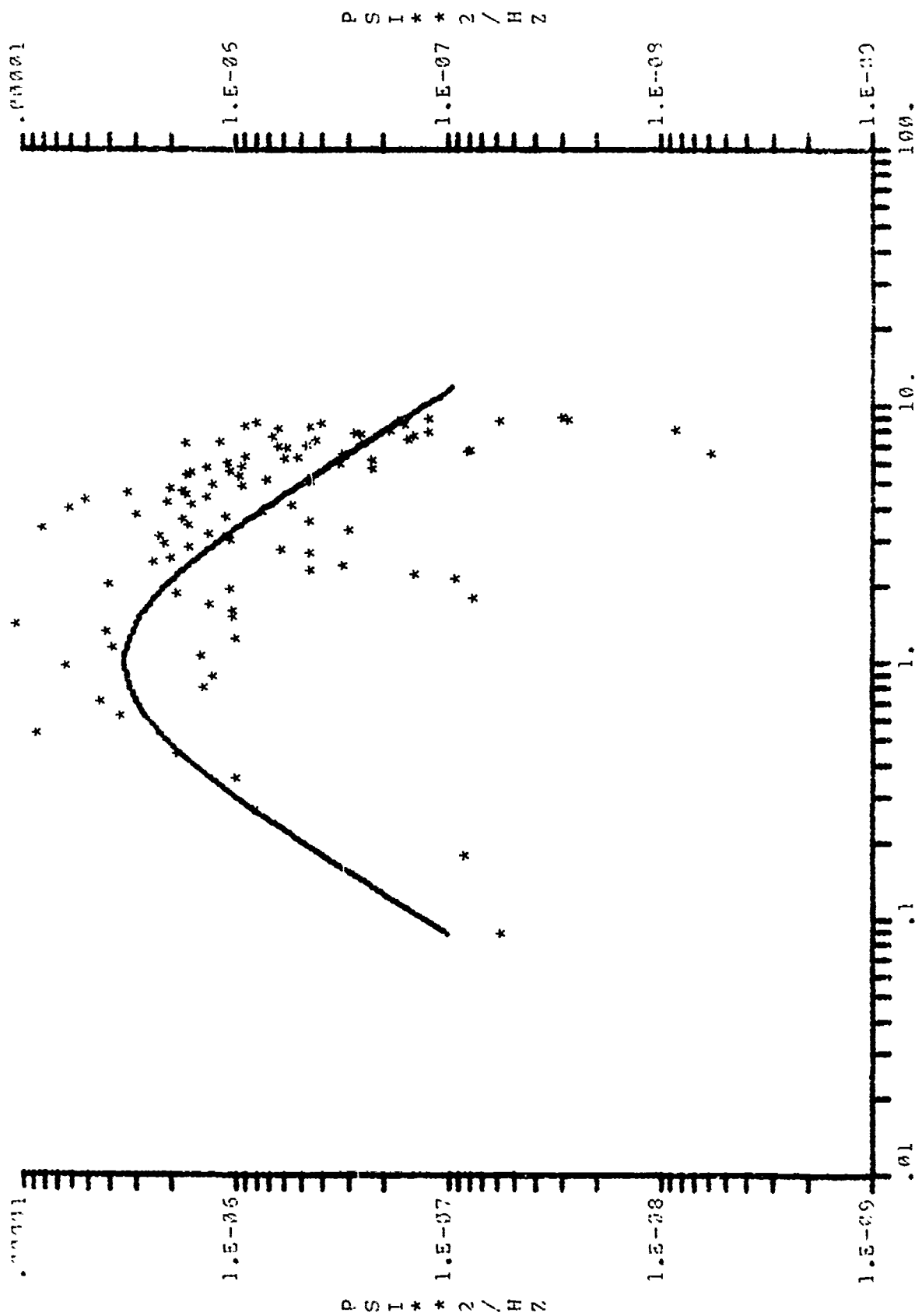


Fig. 10a



CHANNEL 1-1 14.75 SECONDS AFTER IGNITION



NORMALIZED FREQUENCY
Fig. 11a

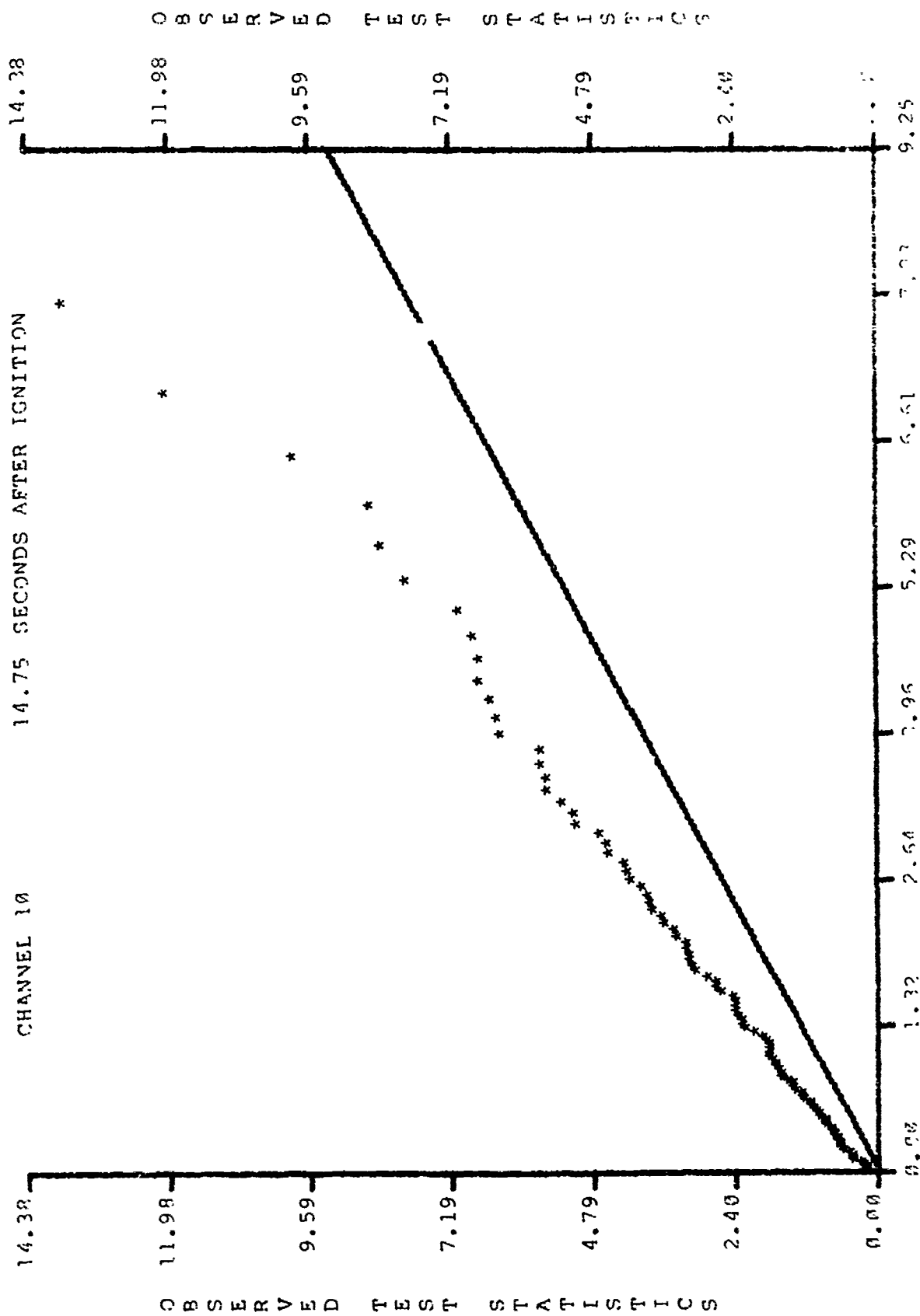


Fig. 11b

CHANNEL 19 16.31 SECONDS AFTER IGNITION

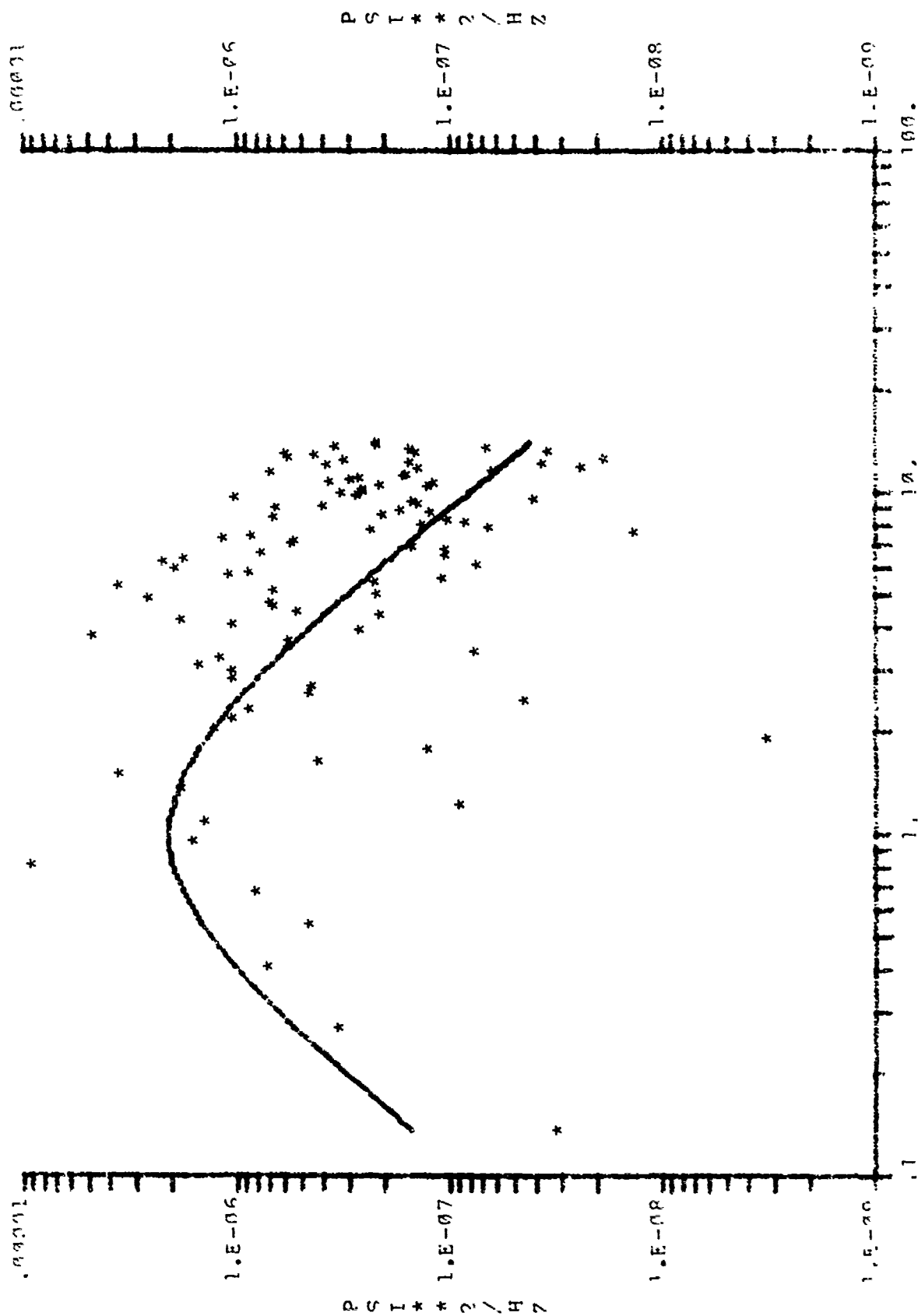
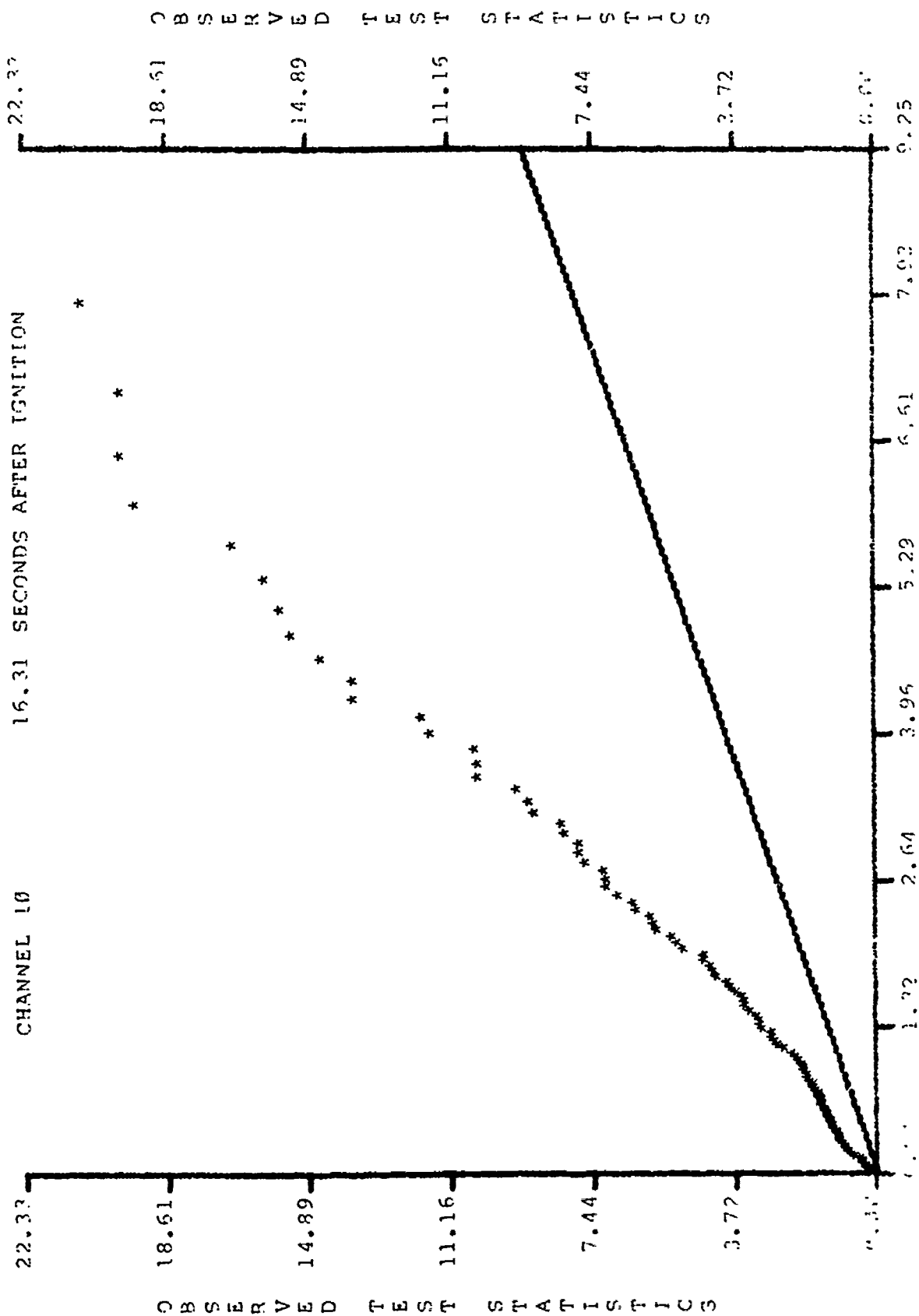


Fig. 12a



THEORETICAL TEST STATISTICS
Fig. 12b

APPENDIX

Listing of computer program follows

```

0001      PROGRAM POWER4
      C      THIS IS POWER2 WITH MUCH OF THE OUTPUT LEFT OUT.
      C      LINK THIS WITH PLLIB3 FOR PLOTTING.
      C      LINK WITH MISC FOR HISTOGRAM AND ASSOCIATED DATA TABLES,
      C      DATMIN AND DATMAX FUNCTIONS, AND SEARCH SUBROUTINE.
      C      LINK WITH FFTSUB FOR FAST FOURIER TRANSFORM SUBROUTINES.
      C
0002      DIMENSION WORK(102), PTHEOR(300), ASCALE(9), IN(20), SUMMY(256),
      +          CONV(103), NSKIP(12), RESID(102), AJUNK(25), PMULT(10),
      +          STATS(102), THSTAT(102), DAT(3), VARPLT(100), SEPLET(100)
0003      LOGICAL*1 LABEL(5)
0004      COMMON PI, W(300), PPRIME(256)
0005      DATA PI/3.1415926535898/
0006      DATA ASCALE/175.9,131.4,204.8,171.4,203.9,153.9,152.9,204.,203.2/
0007      DATA NSKIP/0,235,391,547,703,859,1015,1171,1327,1483,1639,1795/
0008      DATA PMULT/.575,1.,2.,2.773/
      C
0009      TYPE 10
0010  10      FORMAT ( ' TYPE DATA FILE NAME: ' /)
0011      CALL ASSIGN (1, ' ', -1, 'RDO')
0012      DEFINE FILE 1 (5144, 20, U, INDEX)
0013      CALL ASSIGN(2, 'DK:VF2PRS.DAT',13, 'RDO')
0014      DEFINE FILE 2 (257,6,U,JNDEX)
0015      TYPE 15
0016  15      FORMAT(' TYPE FILE NAME FOR OUTPUT OF RESIDUALS AND THEORETICAL'
      +          / ' VALUES: ' /)
0017      CALL ASSIGN(3, ' ', -1)
0018      DEFINE FILE 3 (102,50,U,KNDEX)
      C
0019      DELTAF = 100./256.
0020      RNOYZ = .935E-7
      C
0021      JNDEX = 1
0022      DO 20 I=1,103
0023          READ (2,JNDEX)XXX,CONV(I),YYY
0024  20      JNDEX = JNDEX + 1
      C
0025      TYPE*, 'CHANNEL NUMBER = ?'
0026      ACCEPT*, ICHAN
0027      TYPE*, 'ISTART = ?'
0028      ACCEPT*, ISTART
      C
0029      CALL DATE (DAT)
      C
      C
0030      DO 1000 ISKIP = ISTART,12
      C
0031      PRINT 25, DAT
0032  25      FORMAT(' PROGRAM POWER4'10X,3A4, //
      +          ' FILES USED:'13X'LAUNC2.MNR INPUT'/
      +          25X'VF2PRS.DAT INPUT' / 25X'DATA.LC2 OUTPUT' //)
0033      PRINT 30, ICHAN, NSKIP(ISKIP), RNOYZ/DELTAF
0034  30      FORMAT('*****CHANNEL', I3, 10X, I5, ' OBSERVATIONS SKIPPED.')

```

```

      +      ' NOISE SUBTRACTED =',1PE11.4)
C
C      READ IN NEW OBSERVED DATA
C
0035      INDEX = NSKIP(ISKIP) + 1
0036      DO 50 I=1,256
0037          DUMMY(I) = 0.
0038          READ (1,INDEX)IN
0039      50      PPRIME(I) = IN(ICHAN)/ASCALE(ICHAN-7)
C
C      SUBROUTINE FFTFP CONVERTS ARRAY DATA INTO SPECTRUM
0040      CALL FFTFP (PPRIME,DUMMY,256,2,0)
C
C      CONVERT SPECTRUM INTO PSD (PPRIME ARRAY). DROP FIRST POINT.
C      GENERATE FREQUENCY AND CONVERT TO OMEGA (W ARRAY).
C
C      THE END OF THE LOOP IS 102 INSTEAD OF 128, SO THAT THE MAXIMUM
C      FREQUENCY IS NOW 40 HZ, INSTEAD OF 50 HZ.
C
0041      PSUM = 0.
0042      DO 50 I=1,102
0043          PPRIME(I)=2.*(PPRIME(I+1)-RNOYZ)/(DELTA*CONV(I+1)*CONV(I+1))
0044          IF(PPRIME(I) .LE. 0.)PRINT 58,FLOAT(I)*DELTA,PPRIME(I)
0045      50      FORMAT(' *****WARNING      AT', F5.1,' HZ  PSI**2/HZ =',G11.4)
0046          PSUM = PSUM + PPRIME(I)
0047          W(I) = FLOAT(I) * DELTA
C
0048      IF(ISTART .NE. ISKIP) GOTO 75
0049      KNDEX = 1
0050      DO 70 I=1,102
0051          READ (3,KNDEX)AJUNK
0052          AJUNK(25) = W(I)
0053          KNDEX = KNDEX + 1
0054      70      WRITE(3,KNDEX)AJUNK
C
C      FIND THE TRIAL FREQUENCY THAT GENERATES THEORETICAL VALUES THAT
C      BEST MATCH THE OBSERVED VALUES (CRITERION: LEAST SUM OF SQUARED
C      ERRORS). STORE ANSWER IN W0. STORE ASSOCIATED BEST P IN P0.
C
0055      75      CALL SEARCH(W0,SSE,A,0,W(1),W(102),0.,7,WORK,PPRIME,102,0,0.
0056          7.,5.,'TRIAL FREQUENCY',15,VARPLT,SSEPLT,103)
0057      P0 = PNOT(W0)
C
C      USING THE ESTIMATES OF P AND W, CALCULATE THE CURVE (PTHEOR).
C      THEN CALCULATE THE RESIDUALS AND THE FIGURE OF MERIT (DOLLY)
C
0058      CALL FUNCT(W0,PTHEOR)
0059      RESBAR = 0.
0060      DOLLY = 0.
0061      DO 55 I=1,102
0062          RESID(I) = PPRIME(I) - PTHEOR(I)
0063          RESBAR = RESBAR + ABS(RESID(I))
0064      55      DOLLY = DOLLY + (RESID(I)/PTHEOR(I))**2
0065      RESBAR = RESBAR/102.
0066      DOLLY = DOLLY/102.

```

```

C
C      PRINTOUT RESULTS
C
0058      PRINT 575, W0, P0, SSE, DELTAF*PSUM, 2.*P0, RESBAR, DOLLY
0059  575  FORMAT (///' F-NOT ESTIMATE (HZ)', T32, 1PE15.7 /
+           ' P ESTIMATE', T32, 1PE15.7 /
+           ' SUM OF SQUARED ERRORS (SSE)', T32, 1PE15.7 /
+           ' TOTAL OBSERVED POWER', T32, 1PE15.7 /
+           ' TOTAL THEORETICAL POWER', T32, 1PE15.7 /
+           ' AVERAGE ABSOLUTE RESIDUAL', T32, 1PE15.7 /
+           ' FIGURE OF MERIT', T32, 1PE15.7, //// )

C
C
C      PRINTOUT RESIDUALS AND THEORETICAL VALUES WITH FREQUENCY.
C      STORE THEM IN DATA FILE 3.
C
C      THE FOLLOWING PRINTOUT IS OPTIONAL
CC      PRINT 590
0070  590  FORMAT(' FREQ      THEORETICAL', 2( 19X 'FREQ      THEORETICAL') /
+           ' (HZ)      PSI**2/HZ' 5X 'RESIDUAL', 2(16X ' (HZ)      PSI**2/HZ'
+           5X 'RESIDUAL'))
CC      DO 600 I=1, 34
CC600      PRINT 610, ((W(J), PTHEOR(J), RESID(J)), J=1, (1+58), 34)
0071  610  FORMAT(1X, F5.2, 2(5X, 1PE9.2), 0P, 2(15X, F5.2, 2(5X, 1PE9.2), 0P))
0072      PRINT 625
0073  625  FORMAT('1')
0074      KINDEX = 1
0075      DO 650 I=1, 102
0076          READ(3'KINDEX) AJUNK
0077          LDEX = ISKIP*2 - 1
0078          AJUNK(LDEX) = PTHEOR(I)
0079          AJUNK(LDEX + 1) = RESID(I)
0080          KINDEX = KINDEX - 1
0081  650  WRITE(3'KINDEX) AJUNK

C
C      OPTIONAL PLOT
C
C
C      PLOT PPRIME(OBSERVED) VS PTHEOR(THEORETICAL)
C      INCLUDE 4 COMPARISON LINES:
C          1) ( .575 * PTHEOR) VS PTHEOR
C          2) (1.000 * PTHEOR) VS PTHEOR
C          3) (2.000 * PTHEOR) VS PTHEOR
C          4) (2.773 * PTHEOR) VS PTHEOR
C
C      PLOTTING INFO:
0082      PTMIN = AMIN1(PTHEOR(1), PTHEOR(102))
0083      PPMIN = DATMIN(PPRIME, 102)
0084      YMIN = AMIN1(PPMIN, .5*PTMIN)
0085      PTMAX = DATMAX(PTHEOR, 102)
0086      PPMAX = DATMAX(PPRIME, 102)
0087      YMAX = AMAX1(PPMAX, 2.773*PTMAX)
0088      DELTAY = (YMAX - YMIN)/6.
0089      DELTAX = (PTMAX - PTMIN)/7.

```

```

C
0093      GOTO 715 !PLOT IS SKIPPED.
C
0094      CALL PLOTS(6)
0097      CALL PLOT(1.,-6., 3)
0098      CALL AXIS1(2.,0., 'OBSERVED PSI**2/HZ ',18,6.,93.,YMIN,DELTAY)
0099      CALL AXIS1(2.,0., 'THEORETICAL PSI**2/HZ ',-21.,0.,0.,PTMIN,DELTAX)
0099      CALL AXIS1(7.,0., 'OBSERVED PSI**2/HZ ',-18,6.,93.,YMIN,DELTAY)
0099      CALL LINE(PTHEOR,PTMIN,DELTAX,PPRIME,YMIN,DELTAY,102,1,-1,'*')
C
      PLOT THE 3 COMPARISON LINES
0097      DO 710 I=1,4
0099      CALL PLOT(0.,(PMULT(I)*PTMIN - YMIN)/DELTAY,3)
0099      CALL PLOT(7.,(PMULT(I)*PTMAX - YMIN)/DELTAY,2)
0100      PRINT 625
C
C      CALCULATE AND PLOT TEST STATISTICS VS FREQUENCY
C
0101      DO 720 I=1,102
0102      STATS(I) = 2.*PPRIME(I)/PTHEOR(I)
C
C      OPTIONAL PLOT
C
0103      GOTO 725
0104      CALL DOPLOT(7.,6.,W,STATS,102,'FREQUENCY (HZ) ',14,
      'OBSERVED TEST STATISTICS',24,-1,'*',1,1)
0105      PRINT 625
C
C      OPTIONAL HISTOGRAM OF TEST STATISTICS
C
C
C      CALL HIST(7.,6.,0,'OBSERVED TEST STATISTICS',24,STATS,102,0,1,0)
C
C      ORDER DATA (NOT NECESSARY IF HIST WAS USED):
C
0105      DO 735 I=1,101
0107      MIN = I
0108      DO 730 J=1,102
0109      IF(STATS(J) .LT. STATS(MIN)) MIN = J
0111      CONTINUE
0112      S = STATS(I)
0113      STATS(I) = STATS(MIN)
0114      STATS(MIN) = S
C
C
C      CALCULATE THEORETICAL TEST STATISTICS BASED ON INVERSE OF
C      CHI SQUARE DISTRIBUTION.
C      THIS IS DONE FOR THE FIRST 101 OF THE 102 POINTS; THE
C      THEORETICAL VALUE FOR THE LAST POINT IS INFINITE.
C
0115      DO 740 I=1,101
0116      OBSPRB = 1-102.
0117      THESTAT(I) = -2.*ALOG(1.-OBSPRB)
C
C      PLOT THEORETICAL VS OBSERVED VALUES.
C

```

```

0118 DELTAY = AMAX1(STATS(102),THSTAT(101))/6.
0119 DELTAX = THSTAT(101)/7.
0120 CALL PLOTS(5)
0121 CALL RED
0122 CALL PLOT(1.,-6.5,-3)
0123 CALL AXIS1(0.,0.,'OBSERVED TEST STATISTICS',24,6.,90.,0.,DELTAY)
0124 CALL AXIS1(0.,0.,'THEORETICAL TEST STATISTICS',-27,7.,0.,0.,
+      DELTAX)
0125 CALL AXIS1(7.,0.,'OBSERVED TEST STATISTICS',-24,6.,90.,0.,DELTAY)
0126 CALL LINE(THSTAT,0.,DELTAX,STATS,0.,DELTAY,101,1,-1,'*')
0127 CALL PLOT( THSTAT(1)/DELTAX , THSTAT(1)/DELTAY ,3)
0128 CALL PLOT( THSTAT(101)/DELTAX , THSTAT(101)/DELTAY ,2)
C      NSKIP EQUALS SECONDS*100 PASSED
0129 TIME = FLOAT(NSKIP(1SKIP))/100. - 1.64
0130 ENCODE (5,750,LABEL(1))TIME
0131 750 FORMAT (F6.2)
0132 CALL SYMBOL (1.,6.,0,'CHANNEL 10',0.,10)
0133 CALL SYMBOL (3.,6.,0,LABEL,0.,6)
0134 CALL SYMBOL (3.75,6.,0,'SECONDS AFTER IGNITION',0.,22)
0135 PRINT 625
C
C      THE FOLLOWING EXTENDS THE PTHEOR AND W ARRAYS SO THAT THE
C      THEORETICAL CURVE IS SYMMETRIC
C      NPOINT = THE NUMBER OF POINTS NECESSARY FOR A SYMMETRIC CURVE.
C
0136 NPOINT = 101
0137 INDEX1 = 102
0138 775 NPOINT = NPOINT + 1
0139 W(NPOINT) = FLOAT(INDEX1)*DELTAF
0140 PTHEOR(NPOINT) = PCURVE(W0,P0,W(NPOINT))
0141 INDEX1 = INDEX1 + 10
0142 IF(PTHEOR(NPOINT) .GT. PTHEOR(1)) GOTO 775
C
C      NORMALIZE W BY DIVIDING BY W0
0144 DO 300 I=1,NPOINT
0145 800 W(I) = W(I)/W0
C
C      LOG-LOG PLOT OF BOTH OBSERVED AND THEORETICAL VALUES VS
C      FREQUENCY.
C
0146 PMIN = AMIN1(PPMIN,PTHEOR(1),PTHEOR(NPOINT))
0147 PMAX = AMAX1(PPMAX,PTMAX)
0148 CALL PLOTS(6)
0149 CALL RED
0150 CALL PLOT (1.,-6.5,-3)
0151 CALL LAXIS(0.,0.,'PSI**2/HZ',9,6.,90.,PMAX,PMIN,PDMIN,PDELTA)
0152 CALL LAXIS(7.,0.,'PSI**2/HZ',-9,6.,90.,PMAX,PMIN,PDMIN,PDELTA)
0153 CALL LAXIS(0.,0.,'NORMALIZED FREQUENCY',-20,7.,0.,W(NPOINT),
+      W(1),WDMIN,WDELTA)
0154 CALL LOGLOG (W,PTHEOR,NPOINT,1,WDMIN,WDELTA,PDMIN,PDELTA,0,'.')
0155 CALL LOGLOG (W,PPRIME,102,1,WDMIN,WDELTA,PDMIN,PDELTA,-1,'*')
0156 CALL SYMBOL (1.,6.2,0,'CHANNEL 10',0.,10)
0157 CALL SYMBOL (3.,6.2,0,LABEL,0.,6)
0158 CALL SYMBOL (3.75,6.2,0,'SECONDS AFTER IGNITION',0.,22)

```

```
0159      PRINT 525
      C
0160      GOTO 1000
      C
      C      CONVERT W INTO REGULAR FREQUENCY AND PLOT ABOVE CURVES ON LINEAR
      C      AXES.
      C
0161      DO 950 I=1,102
0162 950      W(I) = W(I)*W0
0163      WINC = (W(102) - W(1))/7.
0164      CALL PLOT (0.,-6.,-3)
0165      PMIN = AMIN1(PPMIN,PTMIN)
      C      PMAX IS THE SAME AS ABOVE, SO:
0166      PINC = (PMAX - PMIN)/6.
0167      CALL AXIS1 (0.,0.,'PSI**2/HZ',9,6.,90.,FMIN,PINC)
0168      CALL AXIS1 (7.,0.,'PSI**2/HZ',-9,6.,90.,PMIN,PINC)
0169      CALL AXIS1 (0.,0.,'FREQUENCY',-9,7.,0.,W(1),WINC)
0170      CALL LINE (W,W(1),WINC,PTHEOR,PMIN,PINC,102,1,0,'.')
0171      CALL LINE (W,W(1),WINC,PPRIME,PMIN,PINC,102,1,-1,'*')
0172      CALL SYMBOL (1.,6.2,0,'CHANNEL 10',0.,10)
0173      CALL SYMBOL (3.,6.2,0,LABEL,0.,6)
0174      CALL SYMBOL (3.75,6.2,0,'SECONDS AFTER IGNITION',0.,22)
0175      PRINT 525
0176 1000 CALL CLOSE(6)
      C
0177      STOP
0178      END
```

C
C
C
C
C

THIS FUNCTION RETURNS THE BEST P-NOT VALUE FOR A GIVEN
OMEGA-NOT. CRITERION IS LEAST SSE.

```

0001  FUNCTION PNOT(WNOT)
0002  COMMON PI, W(300), PPRIME(256)
0003  DENSUM = 0.
0004  TOPSUM = 0.
0005  DO 100 I=1,102
0006      WSUM = W(I)/WNOT + WNOT/W(I)
0007      DENSUM = DENSUM + 1./(WSUM**4)
0008  100  TOPSUM = TOPSUM + PPRIME(I)/(WSUM*WSUM)
0009  PNOT = PI * WNOT * TOPSUM/(4. * DENSUM)
0010  RETURN
0011  END
    
```

```
C
C
C      THIS FUNCTION RETURNS A P(I) VALUE FOR A GIVEN W(I)
C      W(I) IN OMEGA FORM.
C
0001      FUNCTION PCURVE (WNOT,P0,WI)
0002      COMMON PI
0003      WSUM = WI/WNOT + WNOT/WI
0004      PCURVE = 4. * P0/(PI * WNOT * WSUM * WSUM)
0005      RETURN
0006      END
```

```
C
C
0001      SUBROUTINE FUNCT (VAR,WORK)
0002      DIMENSION WORK(102)
0003      COMMON PI,W(300)
0004      P0 = PNOT (VAR)
0005      DO 100 I=1,102
0006      100   WORK(I) = PCURVE (VAR,P0,W(I))
0007      RETURN
0008      END
```

```

0001 SUBROUTINE FFTF (XREAL,XIMAG,N,M,IF)
      C
      C IF=0 FORWARD TRANSFORM
      C IF=1 INVERSE TRANSFORM
      C
      C M=0 XREAL AND XIMAG RETURNED AS REAL AND IMAG. FOR FORWARD XFORMS
      C M=1 " " " " " MAGNITUDE AND PHASE " " "
      C (PHASE IN DEGREES)
      C M=2 XREAL RETURNED AS 'PSD'; XIMAG =0.
      C HERE 'PSD' MEANS SUM OF N VALUES OF XREAL = MEAN SQUARE OF INPUT
      C
      C FOR INVERSE TRANSFORM M DEFINITIONS APPLY TO INPUT DATA
      C XREAL AND XIMAG RETURNED AS REAL AND IMAGINARY
      C
      C FOR FORWARD TRANSFORMS XREAL AND XIMAG INPUT AS REAL AND IMAGINARY
      C
0002 DIMENSION XREAL(1),XIMAG(1)
0003 PI=3.14159
0004 DTOR=PI/180.
0005 IF(IF.EQ.0)GO TO 6
      C
      C MUST PREPARE FOR INVERSE TRANSFORM
      C
0007 IF(M.EQ.0)GO TO 2
0009 IF(M.EQ.2)GO TO 4
      C
      C INPUT IS MAGNITUDE AND PHASE
      C
0011 DO 1 I=1,N
0012 FMAG=XREAL(I)/N
0013 XREAL(I)=FMAG*COS(XIMAG(I)*DTOR)
0014 XIMAG(I)=-FMAG*SIN(XIMAG(I)*DTOR)
0015 1 CONTINUE
0016 GO TO 5
      C
      C INPUT IS REAL AND IMAGINARY
      C
0017 2 DO 3 I=1,N
0018 XREAL(I)=XREAL(I)/N
0019 XIMAG(I)=-XIMAG(I)/N
0020 3 CONTINUE
0021 GO TO 5
      C
      C INPUT IS 'PSD'
      C
0022 4 FACT=FLOAT(N)*N
0023 DO 5 I=1,N
0024 XREAL(I)=XREAL(I)/FACT
0025 5 CONTINUE
      C
0026 6 CALL FFTB (XREAL,XIMAG,N)
      C
0027 IF(IF.EQ.1)RETURN
      C

```

```
      C      TRANSFORM WAS FORWARD
      C
0029      IF(M.EQ.0)RETURN
0031      IF(M.EQ.2)GO TO 8
      C
      C      DESIRE OUTPUT IN MAGNITUDE AND PHASE
      C
0033      DO 7 I=1,N
0034      XMAG=SQRT(XREAL(I)*XREAL(I)+XIMAG(I)*XIMAG(I))
0035      XIMAG(I)=ATAN2(XIMAG(I),XREAL(I))/DTOR
0036      XREAL(I)=XMAG
0037 7      CONTINUE
0038      RETURN
      C
      C      DESIRE 'PSD'
      C
0039 8      FACT=FLOAT(N)*N
0040      DO 9 I=1,N
0041      XREAL(I)=(XREAL(I)*XREAL(I)+XIMAG(I)*XIMAG(I))/FACT
0042      XIMAG(I)=0.
0043 9      CONTINUE
0044      RETURN
0045      END
```

```
0001      SUBROUTINE FFTB (XREAL,XIMAG,N)
0002      DIMENSION XREAL(N),XIMAG(N)
0003      NU=LOG2(N)
0004      N2=N/2
0005      NU1=NU-1
0006      K=0
0007      DO 100 L=1,NU
0008 102      DO 101 I=1,N2
0009          P=IBITR(K/2**NU1,NU)
0010          ARG=6.283185*P/FLOAT(N)
0011          C=COS(ARG)
0012          S=SIN(ARG)
0013          K1=K+1
0014          K1N2=K1+N2
0015          TREAL=XREAL(K1N2)*C+XIMAG(K1N2)*S
0016          TIMAG=XIMAG(K1N2)*C-XREAL(K1N2)*S
0017          XREAL(K1N2)=XREAL(K1)-TREAL
0018          XIMAG(K1N2)=XIMAG(K1)-TIMAG
0019          XREAL(K1)=XREAL(K1)+TREAL
0020          XIMAG(K1)=XIMAG(K1)+TIMAG
0021 101      K=K+1
0022          K=K+N2
0023          IF(K.LT.N)GO TO 102
0025          K=0
0026          NU1=NU1-1
0027 10J      N2=N2/2
0028          DO 103 K=1,N
0029              I=IBITR(K-1,NU)+1
0030              IF(I.LE.K)GO TO 103
0032              TREAL=XREAL(K)
0033              TIMAG=XIMAG(K)
0034              XREAL(K)=XREAL(I)
0035              XIMAG(K)=XIMAG(I)
0036              XREAL(I)=TREAL
0037              XIMAG(I)=TIMAG
0038 103      CONTINUE
0039      RETURN
0040      END
```

```

0001      FUNCTION IBITR(J,NU)
0002      J1=J
0003      IBITR=0
0004      DO 200 I=1,NU
0005      J2=J1/2
0006      IBITR=IBITR*2+(J1-2*J2)
0007 200  J1=J2
0008      RETURN
0009      END
    
```

```
0001      FUNCTION LOG2(N)
0002      N1=N
0003      J=1
0004      LOG2=0
0005  1      IF(J.EQ.N1)RETURN
0007      IF(J.GT.N1)GO TO 2
0009      J=J*2
0010      LOG2=LOG2+1
0011      GO TO 1
0012  2      TYPE 1000,N1
0013  1000  FORMAT (1X,I5,' IS NOT A POWER OF 2')
0014      STOP
0015      END
```

```
0001      SUBROUTINE HANN (RIN,N)
0002      DIMENSION RIN(1),TEMP(256)
0003      M=N-1
0004      DO 1 I=2,M
0005      TEMP(I)=(RIN(I-1)+RIN(I+1))/4+RIN(I)/2
0006 1      CONTINUE
0007      RIN(1)=(RIN(1)+RIN(2))/2
0008      RIN(N)=(RIN(N)+RIN(M))/2
0009      DO 2 I=2,M
0010      RIN(I)=TEMP(I)
0011 2      CONTINUE
0012      RETURN
0013      END
```

```
0001      SUBROUTINE HAN2 (RIN,N)
0002      DIMENSION RIN(1),TEMP(256)
0003      M=N-1
0004      DO 1 I=2,M
0005      TEMP(I)=(RIN(I-1)+RIN(I+1))/3.+RIN(I)/3.
0006 1      CONTINUE
0007      RIN(1)=(RIN(1)+RIN(2)+RIN(3))/3
0008      RIN(N)=(RIN(N)+RIN(M)+RIN(M-1))/3
0009      DO 2 I=2,M
0010      RIN(I)=TEMP(I)
0011 2      CONTINUE
0012      RETURN
0013      END
```

SUBROUTINE SEARCH (THETA, SSETHE, AMPTHE, IAMP, ALOW, HIGH, ACCTHE,
IACCSE, WORK, STAN, NPTS, IPRINT, IPLOT, XSIZE, YSIZE, VARNAM, NCHAR,
VARPLT, SSEPLT, NPLTPT)

PURPOSE

DOES A "STAB" SEARCH FOR AN UNKNOWN PARAMETER.

ARGUMENTS

THETA
VALUE OF PARAMETER RETURNED AFTER THE SEARCH.

SSETHETA
SSE ASSOCIATED WITH THETA, RETURNED BY PROGRAM.

AMPLTHE
AMPLITUDE RATIO FOUND AND RETURNED. IF NO RATIO
CALCULATIONS ARE DESIRED, SET IAMP = 0

IAMP
SEE ABOVE.

LOW
USER-SUPPLIED. LOW END OF RANGE IN WHICH THETA IS
EXPECTED TO APPEAR.

HIGH
USER-SUPPLIED. HIGH END OF RANGE FOR THETA.
IF THE MINIMUM SSE IS FOUND AT EITHER ALOW OR HIGH,
THE PROGRAM WILL EXPAND THE RANGE UNTIL A NON-BOUNDARY
MINIMUM SSE IS FOUND.

ACCTHE
USER-SUPPLIED. THE SEARCH ENDS WHEN EITHER THE
PARAMETER OR ITS SSE HAS BEEN FOUND TO SPECIFIED
PRECISIONS.

WHEN THE DIFFERENCE BETWEEN SUCCESSIVE ESTIMATES OF THE
PARAMETER IS LESS THAN OR EQUAL TO ACCTHE, THE SEARCH
IS STOPPED.

SETTING ACCTHE = 0. RESULTS IN THE COMPUTER SEARCHING TO THE LIMITS OF ITS OWN ACCURACY.

IACCSE
WHEN SUCCESSIVE SSE'S AGREE TO IACCSE DIGITS THE SEARCH
IS STOPPED. MAXIMUM = 7

WORK VECTOR. LENGTH = NPTS.

```

C      STAN
C      USER-SUPPLIED DATA VECTOR CONTAINING THE POINTS THE
C      PROGRAM TRIES TO MATCH.
C
C      NPTS
C      NUMBER OF POINTS IN STAN.
C
C      IPRINT
C      UNLESS SET = 0, USER GETS PRINTOUT OF INTERMEDIATE
C      RESULTS OF THE SEARCH.
C
C      IPLOT
C      PLOTTING CONTROL CHARACTER
C      =0      NO PLOT
C      =1      PLOT WITH REGULAR AXES
C      =2      SEMILOG PLOT (SSE'S ON LOG AXIS)
C      =3      BOTH PLOTS
C
C      XSIZE
C      SIZE OF X-AXIS OF THE OPTIONAL PLOT.
C
C      YSIZE
C      SIZE OF Y-AXIS OF THE OPTIONAL PLOT.
C
C      VARNAM
C      HOLLERITH STRING NAME OF PARAMETER.
C
C      NCHAR
C      NUMBER OF CHARACTERS IN VARNAM.
C
C      VARPLT
C      ARRAY USED FOR PLOTTING.  LENGTH = NPTPLT
C      UPON RETURN CONTAINS ALL ESTIMATES OF THE VARIABLE,
C      ORDERED SMALL TO LARGE.
C
C      SSEPLT
C      ARRAY USED FOR PLOTTING.  LENGTH = NPTPLT
C      UPON RETURN CONTAINS SSE FOR EACH VARIABLE ESTIMATE
C
C      NPTPLT
C      SIZE OF PLOTTING ARRAYS.
C      IF THE INITIAL RANGE (HIGH - ALOW) HOLDS THE BEST VALUE
C      OF THE UNKNOWN VARIABLE, THEN THE FOLLOWING FORMULA SHOULD
C      PROVIDE AN UPPER BOUND ON NPTPLT:
C
C      
$$NPTPLT \leq 3 + 2 * (\log(RANGE/ACCTHE)) / \log(2)$$

C
C      THE USER MUST WRITE A SUBROUTINE FUNCT(VAR,WORK) THAT GENERATES
C      A VECTOR "WORK" OF ESTIMATED DATA POINTS, GIVEN THAT THE UNKNOWN
C      PARAMETER EQUALS "VAR".
C
4001  SUBROUTINE SEARCH(THETA,SSETHE,AMPTHE,IAMP,ALOW,HIGH,ACCTHE,
+      IACCSE,WORK,STAN,NPTS,IPRINT,IPLOT,XSIZE,YSIZE,VARNAM,NCHAR,
+      VARPLT,SSEPLT,NPTPLT)

```

```

0002      DIMENSION VAR(5),SSE(5),AMP(5),WORK(NPTS),STAN(NPTS),
      1      VARPLT(NPLTPT),SSEPLT(NPLTPT)
0003      LOGICAL*1 ACCSE1(14),ACCSE2(14),VARNAM(NCHAR),ALPH(27)
0004      DATA ALPH/' ','A','B','C','D','E','F','G','H','I','J','K',
      1      'L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z'/
      C
0005      IF(1ACCSE .GT. 7) STOP 'SSE ACCURACY CONSTANT MUST BE <= 7'
      C
0007      DIFOLD = 0. ! USED IN PRECISION CHECK SECTION
0008      IROUND = 1 ! MARKS THE ROUND JUST COMPLETED
0009      NPOINT = 0 ! MARKS THE NUMBER OF POINTS ENTERED INTO
      C      THE PLOTTING ARRAYS
0010      ISTOP = 0 ! IF THE PLOTTING ARRAYS ARE FILLED, SSESUB
      C      SUBROUTINE SETS THIS EQUAL TO 1, WHICH IN
      C      TURN CAUSES THE SEARCH TO STOP.
0011      NOTE = 1 ! INDEX OF THE "ALPH" ARRAY. USED WHEN THE
      C      SEARCH RANGE IS EXTENDED.
0012      ADD = (HIGH - ALOW)*.25
      C
      C      SET UP INITIAL VAR(1), VAR(3), AND VAR(5) WITH SSE'S
      C
0013      VAR(5) = HIGH
0014      CALL SSESUB(VAR(5),SSE(5),AMP(5),1AMP,WORK,STAN,NPTS,1PLOT,
      1      NPOINT,VARPLT,SSEPLT,NPLTPT,ISTOP)
      C
0015      VAR(3) = ALOW + .5*(HIGH-ALOW)
0016      CALL SSESUB(VAR(3),SSE(3),AMP(3),1AMP,WORK,STAN,NPTS,1PLOT,
      1      NPOINT,VARPLT,SSEPLT,NPLTPT,ISTOP)
      C
0017      IF(
0018      VAR(1) = VAR(3) - (VAR(5) - VAR(3))
      C
0018      IF(VAR(3) .NE. VAR(1)) GOTO 125 !PRECISION CHECK REQUIRED BECAUSE
      C      RANGE EXTENSION SENDS CONTROL
      C      BACK HERE.
0020      ISTART = 2
0021      GOTO 434
0022      125 CALL SSESUB(VAR(1),SSE(1),AMP(1),1AMP,WORK,STAN,NPTS,1PLOT,
      1      NPOINT,VARPLT,SSEPLT,NPLTPT,ISTOP)
      C
      C
      C      SET UP VAR(2) WITH SSE(2)
      C
0023      150 VAR(2) = (VAR(3) + VAR(1))* .5 + VAR(1)
0024      CALL SSESUB(VAR(2),SSE(2),AMP(2),1AMP,WORK,STAN,NPTS,1PLOT,
      1      NPOINT,VARPLT,SSEPLT,NPLTPT,ISTOP)
      C
      C      IF EITHER OF SSE(1) OR SSE(2) IS THE MINIMUM, THEN THERE IS NO
      C      NEED TO FIND SSE(4). IF SSE(1) IS MIN, EXTEND THE RANGE DOWNWARD.
      C      IF SSE(2) IS MIN, GO ON TO PRECISION CHECKS BEFORE DOING ANOTHER
      C      ROUND. OTHERWISE, COMPUTE VALUES FOR VAR(4).
      C
0025      IF((SSE(1) .GT. SSE(2)) .OR. (SSE(1) .GT. SSE(3)) .OR.
      1      (SSE(1) .GT. SSE(5))) GOTO 200
0027      IF(1PRINT .EQ. 0) GOTO 170
0028      PRINT 160, IROUND,ALPH(NOTE),VAR,SSE

```

```

0030 150  FORMAT(' ROUND',I3,A1 / ' VARIABLE:' 5G15.5 / ' SSE:'5X, 5G15.5)
0031      NOTE = NOTE + 1
0032      IF(NOTE .EQ. 28) NOTE = 27
0033      CALL CLOSE(5)
0034
0035 170  IF(ISTOP .NE. 0) GOTO 650
      C  RESET ARRAYS AND DO ANOTHER ROUND:
0037      CALL RESET(VAR,1,3,4,5)
0038      CALL RESET(SSE,1,3,4,5)
0039      CALL RESET(AMP,1,3,4,5)
0040      GOTO 100
      C
0041 200  IF((SSE(2) .GT. SSE(3)) .OR. (SSE(2) .GT. SSE(5))) GOTO 300
      C  HERE IT IS KNOWN THAT VAR(2) IS THE BEST VALUE SO FAR
0043      ISTART = 1
0044      GOTO 420
      C
      C  SET UP VAR(4) WITH SSE(4)
      C
0045 300  VAR(4) = (VAR(5) - VAR(3))* .5 + VAR(3)
0046      CALL SSESUB(VAR(4),SSE(4),AMP(4),IAMP,WORK,STAN,NPTS,IPLOT,
      +          NPOINT,VARPLT,SSEPLT,NPLTPT,ISTOP)
      C
      C  AS ABOVE, SEE IF THE ENDPOINT OF THE SEARCH INTERVAL (HERE, VAR(5))
      C  YIELDS A BOUNDARY MIN SSE.  IF SO, RESET THE VARIABLES SO THAT
      C  THE RANGE IS EXTENDED UPWARDS.  OTHERWISE, FIND THE STARTING POINT
      C  FOR THE NEXT ROUND, AND GO TO PRECISION CHECK SECTION.
      C
0047      IF( (SSE(5) .GT. SSE(3)) .OR. (SSE(5) .GT. SSE(4)) ) GOTO 400
0049      IF(IPRINT .EQ. 0) GOTO 350
0051      PRINT 160, IROUND,ALPH(NOTE),VAR,SSE
0052      NOTE = NOTE + 1
0053      IF(NOTE .EQ. 28) NOTE = 27
0055      CALL CLOSE(5)
0056 350  IF(ISTOP .NE. 0) GOTO 650
0058      CALL RESET(VAR,3,1,2,3)
0059      CALL RESET(SSE,3,1,2,3)
0060      CALL RESET(AMP,3,1,2,3)
0061      VAR(5) = VAR(3) + (VAR(3) - VAR(1))
0062      IF(VAR(5) .NE. VAR(3)) GOTO 375
0064      ISTART = 2
0065      GOTO 430
0066 375  CALL SSESUB(VAR(5),SSE(5),AMP(5),IAMP,WORK,STAN,NPTS,IPLOT,
      +          NPOINT,VARPLT,SSEPLT,NPLTPT,ISTOP)
0067      GOTO 300
      C
0068 400  ISTART = 2
0069      IF(SSE(4) .LE. SSE(3)) ISTART = 3
0071 420  IF(IPRINT .EQ. 0) GOTO 430
0073      PRINT 160, IROUND,ALPH(NOTE),VAR,SSE
0074      CALL CLOSE(5)
      C
      C  PRECISION CHECKS
      C
      C  FIRST:      IF THE DIFFERENCE BETWEEN SUCCESSIVE ESTIMATES OF THE

```

```

      C      UNKNOWN VARIABLE IS LESS THAN OR EQUAL TO "ACCTHE"
      C      (USER-SUPPLIED CONSTANT) THEN EXIT.
      C
      C      THE DIFFERENCE MAY NEVER BE LESS THAN ACCTHE BECAUSE OF
      C      ROUNDING, IN THIS CASE DIFF REMAINS CONSTANT OVER ROUNDS.
      C      DIFOLD IS A CHECK ON THIS.
      C
0075 430 DIFF = VAR(2) - VAR(1)
0076      IF(DIFF .GT. ACCTHE) GOTO 432
0078      PRINT 431,ACCTHE
0079 431 FORMAT('0***** SEARCH STOPPED      DIFFERENCE BETWEEN ESTIMATES <=',
+           G15.5)
0080      GOTO 700
0081 432 IF(DIFF .NE. DIFOLD) GOTO 440
0083 434 PRINT 435,VARNAM
0084 435 FORMAT('0*****SEARCH STOPPED*****'/
+           ' DOUBLE PRECISION REQUIRED FOR FURTHER REFINEMENTS OF',
+           ' ESTIMATES OF ',30A1)
0085      GOTO 700
      C
      C      SECOND:  IF THE SSE'S AGREE TO A USER-SUPPLIED NUMBER OF
      C      DIGITS (IACCSE) THEN EXIT.
      C
0096 440 DIFOLD = DIFF
0097      ENCODE (14,450,ACCSEL(1))SSE(ISTART)
0098 450 FORMAT(1PE14.7)
0099      DO 550 I=(ISTART+1),(ISTART+2)
0100          ENCODE(14,450,ACCSE2(1))SSE(I)
0101          DO 500 J=12,14
0102              IF(ACCSEL(J) .NE. ACCSE2(J)) GOTO 600
0104 500      CONTINUE
0105          DO 550 J=2,(IACCSE + 2)
0106              IF(ACCSEL(J) .NE. ACCSE2(J)) GOTO 600
0108 550      CONTINUE
0109      PRINT 575,IACCSE
0110 575 FORMAT('0***** SEARCH STOPPED      SUM OF SQUARED ERRORS MATCH TO',
+           12 ' DIGITS.')
0111      GOTO 700
      C
      C      SET UP ARRAYS FOR THE NEXT ROUND
0112 600 IF(ISTOP .NE. 0) GOTO 650
0114      CALL RESET(VAR,ISTART,1,3,5)
0115      CALL RESET(SSE,ISTART,1,3,5)
0116      CALL RESET(AMP,ISTART,1,3,5)
0117      IROUND = IROUND + 1
0118      NOTE = 1
0119      GOTO 150
      C
0110 650 PRINT*, '***** SEARCH STOPPED      PLOTTING ARRAYS FILLED *****'
0111 700 THETA = VAR(ISTART+1)
0112      SSETHE = SSE(ISTART+1)
0113      AMPTHE = AMP(ISTART+1)
      C
0114      IF(IPLOT .EQ. 0) RETURN

```

```
C
C      ORDER THE VARIABLE ESTIMATES SO THAT A LINE CAN BE DRAWN BETWEEN
C      DATA POINTS ON THE GRAPH,  SHOWING BEHAVIOR OF SSE'S.
C
0116 800  FORMAT ('1')
0117      DO 900 I=1,(NPOINT - 1)
0118          MIN = I
0119          DO 850 J=I,NPOINT
0120              IF(VARPLT(J) .LT. VARPLT(MIN)) MIN = J
0122 850  CONTINUE
0123          S1 = VARPLT(I)
0124          VARPLT(I) = VARPLT(MIN)
0125          VARPLT(MIN) = S1
0126          S1 = SSEPLT(I)
0127          SSEPLT(I) = SSEPLT(MIN)
0128 900  SSEPLT(MIN) = S1
C
0129      IF(IPLOT .EQ. 2) GOTO 950
C
0131      PRINT 800
C      REGULAR PLOT OF SSE VS VARIABLE ESTIMATES.
0132      CALL DOPLT(XSIZE,YSIZE,VARPLT,SSEPLT,NPOINT,VARNAM,NCHAR,'SSE',
+          3,0,'.',1,1)
C
0133      IF(IPLOT .EQ. 1) RETURN
C
0135 950  PRINT 800
C      SEMILOG PLOT OF SSE VS VARIABLE ESTIMATES.
0136      CALL PLTLGY(XSIZE,YSIZE,VARPLT,SSEPLT,NPOINT,VARNAM,NCHAR,'SSE',
+          3,0,'.',1)
0137      RETURN
0138      END
```

```
C
C      SUBROUTINE SSESUB
C
C      PURPOSE
C          CALLED BY SEARCH TO FIND AMPLITUDE RATIO(AMP) AND SSE FOR A
C          GIVEN VALUE OF THE UNKNOWN PARAMETER(VAR)
C
0001      SUBROUTINE SSESUB(VAR,SSE,AMP,IAMP,WORK,STAN,NPTS,IPLT,
+          NPOINT,VARPLT,SSEPLT,NPLTPT,ISTOP)
0002      DIMENSION WORK(NPTS),STAN(NPTS),VARPLT(NPLTPT),SSEPLT(NPLTPT)
C
0003      CALL FUNCT(VAR,WORK)
C
0004      A = 1.
0005      IF(IAMP .EQ. 0.) GOTO 200
C      FIND AMPLITUDE RATIO
0007      SUMSQ = 0.
0008      SUMX = 0.
0009      DO 100 I=1,NPTS
0010          SUMX = SUMX + STAN(I)*WORK(I)
0011      100    SUMSQ = SUMSQ + WORK(I)*WORK(I)
0012      AMP = SUMX/SUMSQ
0013      A = AMP
C
C      FIND SSE
0014      200    SSE = 0.
0015      DO 300 I=1,NPTS
0016          ERROR = STAN(I) - A*WORK(I)
0017      300    SSE = SSE + ERROR**2
C
C      FILL PLOTTING ARRAYS IF NECESSARY
C
0018      IF(IPLT .EQ. 0) RETURN
0019      NPOINT = NPOINT + 1
0020      VARPLT(NPOINT) = VAR
0021      SSEPLT(NPOINT) = SSE
0022      IF(NPOINT .GE. (NPLTPT - 1) ) ISTOP = 1
C      IF ISTOP = 1, ARRAYS ARE FILLED AND SEARCH STOPS UPON RETURN.
C
0023      RETURN
0024      END
```

C
C
C
C
C
C

SUBROUTINE RESET

PURPOSE

CALLED BY SEARCH TO REASSIGN ELEMENTS OF ARRAYS.

```
0001 SUBROUTINE RESET(A,I,J,K,L)
0002 DIMENSION A(5)
0003 X = A(I)
0004 Y = A(I+1)
0005 Z = A(I+2)
0006 A(J) = X
0007 A(K) = Y
0008 A(L) = Z
0009 RETURN
0010 END
```