CR-152004

John Foster
And Associates
Inc.

(NASA-CR-152004) POINT SOURCE DETECTION IN INFRARED ASTRONOMICAL SURVEYS (Foster (John) and Associates, Inc.) 74 p HC A04/MF A01 CSCL 03A

N77-34061

Unclas
G3/89 46973

NASA
National Aeronautics and Space Administration

Ames Research Center
Moffett Field, California

ARC 275 (Nov 75)

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California
94035

NASA

Reply to Attn of NAS 2-9432
ATL:202-3

September 16, 1977

NASA Representative
Scientific & Technical Information Facility
P.O. Box 8757
Baltimore/Washington International Airport,
Maryland 21240

Subject: Transmittal of Contractor Report: "Point Source Detection in Infrared Astronomical Surveys," dated March 20, 1977, Robert F. Pelzmann, Jr., PhD.

Reference: Program Code 358-41-06

The subject report prepared under Contract NAS 2-9432 has been reviewed at Ames and is recommended for release in STAR as CR-152004.

Ralph W. Lewis

Ralph W. Lewis
Chief, Library Branch

Enclosure:
1 cy subject report

cc:
NASA Hqrs., Code KSI (w/o encs.)

POINT SOURCE DETECTION

IN

INFRARED ASTRONOMICAL SURVEYS

By

Robert F. Pelzmann, Jr., Ph.D.

20 March 1977

Prepared under Contract No. NAS2-9432

for the

National Aeronautics and Space Administration
Ames Research Center
Moffett Field, California

NASA CR-152004

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

1.0 INTRODUCTION

This is the second report on data processing techniques useful for infrared astronomy data analysis systems. As with the first report (NASA CR-151943), the investigation is restricted to consideration of data from space-based telescope systems operating as survey instruments. In this report the theoretical background for specific point-source detection schemes is completed, and the development of specific algorithms and software for the broad range of requirements is begun.

Section 2 develops the detail detection tests and processing requirements for point-source surveys and evaluates the performance measurement processes. The details of peak detection decisions and correlation detection are covered for the case of general bandlimited white gaussian noise. For non-white noise, a modified correlation test and a matched filter test are presented. A technique for resampling the data which is equivalent to a matched filtering approach is discussed which automatically decorrelates the noise. Implementation of this Karhunen-Loeve filtering is necessarily complicated, but for some kinds of noise an acceptable approach.

Section 3 then reviews a basic processing task to indicate where computation is needed outside of the normal data stream. While the processing used in the primary data reduction task is important, the actual software depends heavily on the specific mission hardware and is best approached anew for each task using the theories of Section 2 of this report and of the previous report, and of several cited authors. For the general signal processing task, a routine for designing digital filters is given based on the theory of Section 2.5. The calibration of detector-filter systems is the most complicated of the tasks off the main processing line; a routine which provides this calibration for blackbody or other input spectra. Finally, the preliminary processing routine for a previous survey program is presented and briefly discussed to indicate how much processing can be done in a single pass of the data.

The Appendix in Section 4 presents an interesting game which can develop a fuller appreciation and understanding of the complexities of data analysis.

2.0 TECHNICAL ASPECTS

This section completes the task begun in the first report of reviewing the theoretical basis for the design of point-source survey data analysis software. The detection techniques for single-channel signal and noise processing are reviewed. The schemes reviewed include peak detection, optimal filtering, correlation, and Karhunen-Loeve filtering. The details of digital filtering, which is applicable to many aspects of data processing, are reviewed in the final section.

2.1 Detection of Signals in Noise

In most communication systems the errors (false detections and missed signals) are assumed to be of equal importance and with known probabilities. In more general detection problems, however, the *a priori* probabilities and costs of those errors are difficult to determine. The Neyman-Pearson test was first applied in such a case to radar detection with a peak measurement technique. The criterion can also be applied to more sophisticated detection methods, and in all cases, will give the highest probability of detection at a chosen false-alarm rate. The type of technique used depends on the amount of information available about the expected signal; generally, more information used will result in a higher detection probability at the chosen false-alarm rate. The likelihood ratio is the test used where the hypothesis is chosen if:

$$\lambda = \frac{p(s)}{p(n)} \geq \eta \qquad 2.1\text{-}1$$

and the counter-hypothesis (no signal) is chosen otherwise. Here p(s) is the probability density function of the data with a signal present and p(n) is the p.d.f. of the noise alone, and η is the decision level chosen to satisfy the false-alarm constraint.

Consider the case of a signal in white noise, such that the signal has a normalized mean value of one. The probability functions are:

$$p(s) = \frac{1}{\sqrt{4\pi}} e^{-(y-1)^2/4} \quad \text{and} \quad p(n) = \frac{1}{\sqrt{4\pi}} e^{-y^2/4} \tag{2.1-2}$$

Then the likelihood ratio test is:

$$\lambda(y) = e^{(y/2)-1} \geq \lambda_o \tag{2.1-3}$$

To determine the threshold λ_o, the false-alarm probability is found from:

$$P(f.a.) = \int_\gamma^\infty \frac{1}{\sqrt{4\pi}} e^{-y^2/4} dy \tag{2.1-4}$$

If we want a false-alarm rate of 10% or less, then $\gamma = 1.8$, and we choose the hypothesis if $y \geq 1.8$.

The probability of detection for a single test observation is:

$$P(det.) = \int_\gamma^\infty \frac{1}{\sqrt{4\pi}} e^{-(y-1)^2/4} dy = 0.285 \tag{2.1-5}$$

In terms of the likelihood ratio, note that $\lambda(\gamma) = \lambda_o = 1.9$ and we make a detection whenever $\lambda(y) \geq 1.9$. To improve this rather mediocre performance, several measurements may be tested. With the same false-alarm rate, we choose the decision level differently. If we take η independent samples, the signal-present probability distribution has unity mean and a variance of σ^2, and:

$$\begin{aligned} p_s(y_1, y_2, \ldots, y_n) &= \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left[-\frac{(y_1-1)^2}{2\sigma^2}\right] \times \ldots \\ &\quad \times \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left[-\frac{(y_n-1)^2}{2\sigma^2}\right] \end{aligned} \tag{2.1-6}$$

Similarly, the noise-only probability distribution is:

$$p_n(y) = \left(\frac{1}{2\pi\sigma^2}\right)^{n/2} \exp\left[-\frac{1}{2} \sum_1^n \left(\frac{y_i}{\sigma}\right)^2\right. \tag{2.1-7}$$

Taking the logarithm of the likelihood ratio, the decision test is:

$$\frac{1}{n}\sum_{i=1}^{n} y_i \geq \lambda_o' \qquad 2.1\text{-}8$$

where

$$\lambda_o' = \frac{1}{2} + \frac{\sigma^2}{n} \ln \lambda_o$$

Now the probability of each kind of error is different, and we choose λ_o' by evaluating P_1 (f.a.) + P_2 (missed signal), where

$$P_1(\text{f.a.}) = \int_{\lambda_o'}^{\infty} P_n(\bar{y})\, d\bar{y} = \int_{\lambda_o'}^{\infty} \left(\frac{n}{2\pi\sigma^2}\right)^{1/2} e^{-nz^2/2\sigma^2}\, dz$$

$$2.1\text{-}9$$

and

$$P_2(\text{m.s.}) = \int_{-\infty}^{\lambda_o'} P_s(\bar{y})\, d\bar{y} = \int_{-\infty}^{\lambda_o'} \left(\frac{n}{2\pi\sigma^2}\right)^{1/2} e^{-n(z-1)^2/2\sigma^2}\, dz$$

It now becomes clear that improving the performance of simple peak detection schemes becomes a complicated task even using very little information about the signal. If we use more of the information available, and some of the knowledge about the nature of the noise, a more successful detection scheme can be derived. When multiple detections are made on a single source, the above can be used to evaluate the detection probability.

2.2 Correlation Detection

Rather than make a detection test based on only the peaks of the data stream as in the previous example, consider how we might deal with detecting a signal that we know. Let r_k, k=1, ..., m be the sequential data samples. Assuming additive noise,

$$r_k = \left\{ {S_K \atop 0} \right\} + n_k \qquad 2.2\text{-}1$$

where S_R is the k^{th} value of our expected signal, and n_k is the noise sample. We may now derive a likelihood ratio test which uses this information.

First, assume that the noise is bandlimited white noise with power spectral density $S(\omega) = N_0/2$ for $|\omega| < \Omega$ and zero otherwise. The noise autocorrelation function then is given by:

$$R(\tau) = \frac{N_{o\Omega}}{2\pi} \frac{\sin(\Omega\tau)}{\Omega\tau} \qquad 2.2\text{-}2$$

This has its first zero at $\tau = \pi/\Omega$ so that if the received signal is sampled at intervals $\Delta t = \pi/\Omega$ the samples will be uncorrelated, and being gaussian they then will be statistically independent. The probability density functions of the two cases will be:

$$p_s(r) = \left(\frac{1}{2\pi\sigma_n^2}\right)^{m/2} \exp\left[-\sum_{k=1}^{m} \frac{(r_K-S_K)^2}{2\sigma_n^2}\right]$$

and 2.2-3

$$p_n(r) = \left(\frac{1}{2\pi\sigma_n^2}\right)^{m/2} \exp\left[-\sum_{k=1}^{m} \frac{(r_k)^2}{2\sigma_n^2}\right]$$

and the logarithm of the likelihood ratio test results in the decision test:

$$\sum_{k=1}^{m} \frac{r_K S_K}{\sigma_n^2} \geq \ln \lambda_o + \frac{1}{2} \sum_{k=1}^{m} \frac{S_R^2}{\sigma_n^2} \qquad 2.2\text{-}4$$

Now the left-hand side of 2.2-4 is just the normalized cross-correlation coefficient of the signal with its expected template. Furthermore, the variance of the noise σ_n^2 is just the noise auto-covariance function at zero frequency,

$$\sigma_n^2 = \frac{N_o\Omega}{2\pi} \qquad 2.2\text{-}5$$

ORIGINAL PAGE IS
OF POOR QUALITY

Since one of our two signals is zero (noise only), we may define the average signal energy E and the time cross-correlation coefficient ρ by:

$$E = \frac{1}{2n} \sum_{k=1}^{m} S_k^{\,2} \qquad 2.2\text{-}6$$

and

$$\bar{\rho} = 0$$

By extending equations 2.2-3 to infinite bandwidth $\Omega \to \infty$, the probability density functions for the signal case and the noise-only case can be derived as:

$$P_n(G) = \left[\frac{1}{2\pi N_o E}\right]^{1/2} \exp\left[- \frac{(G+E)^2}{2N_o E}\right]$$

$$2.2\text{-}7$$

$$P_s(G) = \left[\frac{1}{2\pi N_o E}\right]^{1/2} \exp\left[- \frac{(G-E)^2}{2N_o E}\right]$$

Since the false-alarm rate and the missed sources probabilities are equal when the samples are uncorrelated, the error rate is:

$$P_e = \int_{\gamma}^{\infty} \frac{1}{\sqrt{2\pi}} \; e^{-z^2/2} \, dz$$

where 2.2-8

$$\gamma = (E/N_o)^{1/2}$$

and thus we can determine an error rate based on the signal-to-noise ratio, independent of the shape of the signal. Figure 1 shows the error rate as a function of the signal-to-noise power ratio. Note that as long as the noise samples are uncorrelated, the error rate is also independent of the number of samples in the correlation sum. This apparently unreasonable result is directly related to the assumption of statistically independent samples. For bandlimited

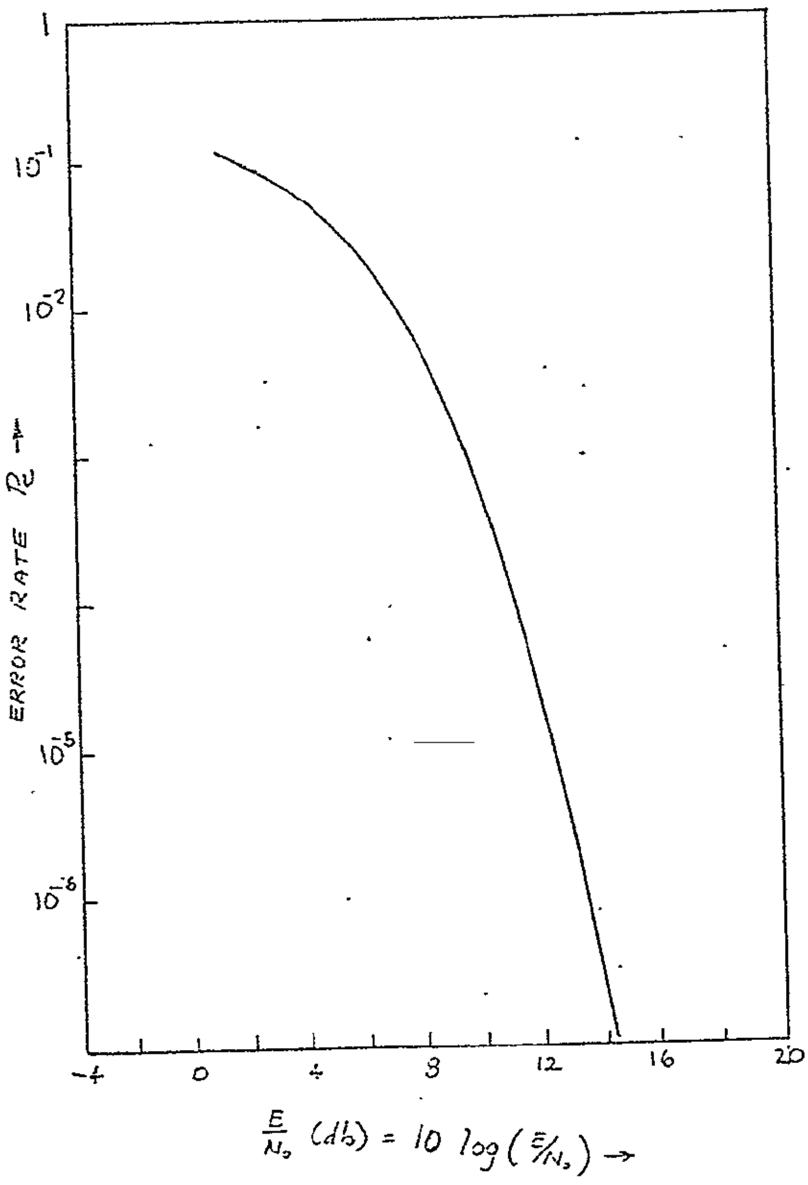


FIGURE 1. ERROR RATE PERFORMANCE FOR CORRELATION DETECTION

white noise there must be $m = \Omega t/\pi$ = constant independent samples in the interval 0 to T. Oversampling the signal may actually result in degraded performance, as will be discussed in section 2.4. If we choose the value of γ in 2.2-8 to achieve a desired error rate, then the probability of detection is

$$P_D = \int_{\gamma-(2E/N_o)^{1/2}}^{\infty} (2\pi)^{-1/2} \exp[-Z^{2/2}]\, dz \qquad 2.2\text{-}9$$

which is shown in Figure 2 as a function of signal-to-noise ratio and error rate.

If we have chosen the normalized signal template properly, our detection test simultaneously makes a best estimate of the signal amplitude. If the signal model is written as a function of a constant amplitude factor A, then the maximum likelihood estimate of that amplitude is the solution of:

$$\sum_{i=1}^{m} [r_i - s_i (A)] \frac{\partial s(A)}{\partial A} = 0 \qquad 2.2\text{-}10$$

or, writing $s = A s'$, we want the solution of:

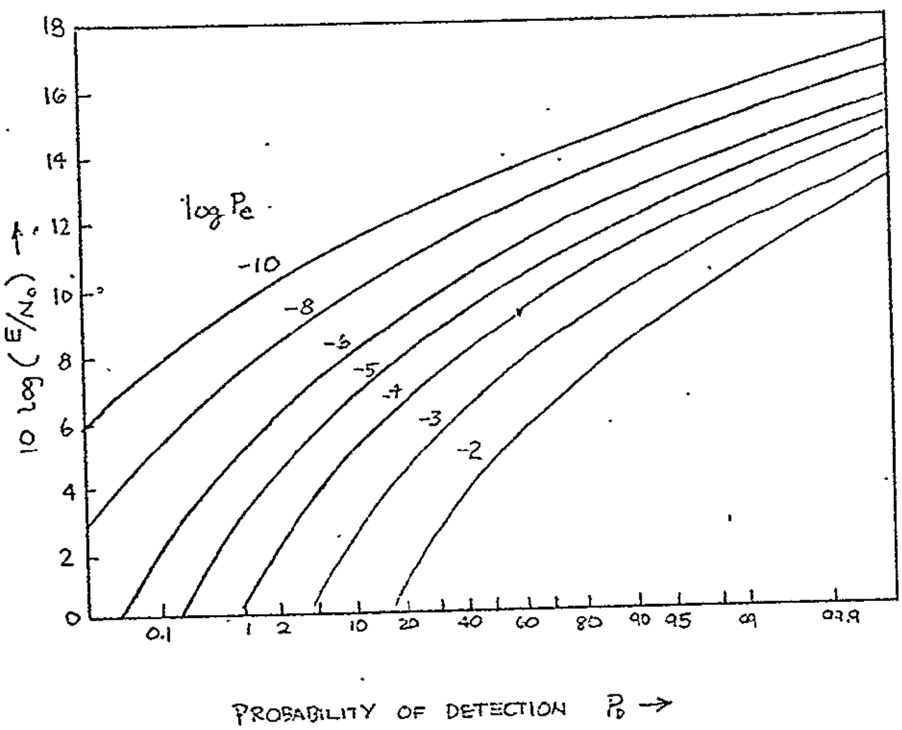
$$\sum_{i=1}^{m} (r_i - \hat{A} S_i)S_i = 0 \qquad 2.2\text{-}11$$

That solution is

$$\hat{A} = \frac{\sum_{i=1}^{m} (r_i S_i)}{\sum_{i=1}^{m} S_i^{2}} \qquad 2.2\text{-}12$$

and now if S_i was normalized such that $\sum S_i^{2} = 1$, and we re-arrange the terms in 2.2-4, we have the detection test and amplitude estimate simultaneously:

$$\hat{A} = \sum_{k=1}^{m} r_k S_R \geq \sigma_n^2 \ln \lambda_o + \frac{1}{2} \qquad 2.2\text{-}13$$



log Pe
-10
-8
-6
-5
-4
-3
-2
10 log (E/No)
0
2
4
6
8
10
12
14
16
18
0.1
1
2
10
20
40
60
80
90
95
99
99.9
PROBABILITY OF DETECTION PD

Figure 2.

Now it is clear how the correlation test is a better detector than a peak test. The correlation test takes an average of the signals weighted by the expected response as a best esimate of the amplitude. Because it is using n samples of the signal, the improvement in error rate can be as much as $\sqrt{n}$. The uncertainty in the estimate is determined from the noise autocovariance function, as in section 2.3 as:

$$\sigma_A^2 \quad \int_0^T \int_0^t s(\tau)\, s(z)\, R_n(z-\tau)\, dz d\tau \qquad 2.2\text{-}14$$

If we have N multiple pulses available from a single source, the decision test 2.2-4 can be modified to:

$$\sum_{i=1}^{N} \sum_{k=1}^{m} r_k\, s_k \geq \sigma_n^2 \ln \lambda_o' + \frac{1}{2} \sum_{i=1}^{N} \sum_{k-1}^{m} s_k^2$$

The false-alarm rate given by 2.2-8 is not changed, since we are designing our test for a chosen error performance. However, the detection probability improves; the new detection rate is given by 2.2-9 by replacing E with E':

$$E' = \sum_{i=1}^{N} E_i$$

And since all signal energies for a given source are equal, there will be a 3 dB increase in the equivalent performance for each doubling of the number of signals.

As a final note to this discussion, the signal-to-noise ratio used here is the more useful signal power-to-noise power ratio, not the typical peak-to-rms value which has little physical meaning.

2.3 Matched Filters and Non-White Noise

The cross-correlation term on the left-hand side of equation 2.2-4 can be replaced by the equivalent matched filter. If the filter's transfer function is $h(t_i)$, then the output of the filter is

$$e(t_k) = \sum_{i=1}^{m} h_i \, r_{k-i} \qquad 2.3\text{-}1$$

and by inspection the filter output matches the correlator output if

$$h_i = S_{m-i}$$

That is, the matched filter is the time-reversed image of the signal expected. It is important to note, however, that because of the time reversal, the matched filter and the correlator output are equal only at time T, where the entire signal train (in samples) is within the bounds of the filter or of the correlator.

The matched filter representation is well suited to the case of non-white noise. We will show that the optimal detector for non-white noise replaces the left-hand side of 2.2-4 with a filter which is the product of the white-noise matched filter and a pre-whitening filter described in terms of the autocovariance function of the noise. To avoid confusing subscripts, we shall write the filter transformations in terms of time integrals which are the generalized extensions of the summations in section 2.2. The output of the filter at time T is:

$$e(T) = \int_0^T h(\tau)\, r\,(T-\tau)\, d\tau \qquad 2.3\text{-}2$$

The signal and noise components are easily identified as

$$S(T) = \int_0^T h(\tau)\, S(T-\tau)\, d\tau$$

2.3-3

$$N(T) = \int_0^T h(\tau)\, n\,(T-\tau)\, d\tau$$

The noise power then can be written in terms of the autocovariance function as

$$\sigma_n^2 = \int_0^T \int_0^T h(\tau)\, h(z)\, R_n(z-\tau)\, dz d\tau \qquad 2.3\text{-}4$$

The optimum signal-to-noise ratio can be found by minimizing the Lagrangian:

$$L = \int_0^T \int_0^T h(\tau)\, h(z)\, R_n(z-\tau) dz d\tau - \mu \int_0^T h(\tau) s(T-\tau)\, d\tau \qquad 2.3\text{-}5$$

The resulting variation yields:

$$\int_0^T h_o\,(z) R_n(\tau-z) dz = s(T-\tau) \qquad 2.3\text{-}6$$

The filter which satisfies this relation will maximize the signal-to-noise ratio for a known signal in any noise with autocovariance function $R_n(\tau)$. Equation 2.3-6 is, of course, a Fredholm integral equation of the first kind which is solvable only for a restricted group of covariance functions $R_n(\tau)$. If, however, we can adequately approximate the integration by replacing the 0 to T limits with $-\infty$ to $+\infty$, then the Fourier transform of 2.3-6 gives immediately

$$H(s) = \frac{S^*(s)e^{-sT}}{S_n(s)}$$

where $s = iw$ and $S_n(s)$ is the actual power spectral density function of the noise. This matched filter is then just the white-noise matched filter co nvolved with the actual noise spectrum. This result was derived for the limit $T \to \pm\infty$, but a detailed derivation shows that it holds wherever the data samples are uncorrelated, which was determined from the zeros of the noise autocovariance function.

As in the previous section, a best estimate of the signal amplitude exists in the presence of non-white noise. That estimate is given by:

$$\hat{A} = \frac{\int_0^T h(\tau) r(\tau) d\tau}{\int_0^T h(\tau) s(\tau) d\tau} \qquad 2.3\text{-}7$$

where $h(\tau)$ is the solution of:

$$s(t) = \int_0^T R_n(t-\tau)\ h(\tau)\ d\tau \qquad 2.3\text{-}8$$

Comparing this result with 2.3-6, we see that the optimal whitening filter is the best weighting function for the correlation detector and the amplitude estimate in the presence of non-white noise.

2.4 Karhunen-Loeve Filtering

The emphasis in the preceding section was an additive white noise. Since this is often invalid, we derived a test based on the noise auto-covariance function. For the white noise case we considered a flat bandlimited spectrum and found that appropriate uniformly spaced amplitude samples were statistically independent. For colored noise we considered the continuous sampling limit and wrote the detection equations as integral relationships. However, uniformly spaced samples in colored noise are correlated and the sampled case is difficult to evaluate explicitly. There is, however, another method which can be used to generate statistically independent samples. While these are not amplitude samples, they can be used to construct the same detection and performance tests as previously described. The approach used will be to expand the signal in a series of functions which are orthogonal over the region 0 to T.

The functions we seek are a set of $f_i(t)$'s with the normality condition:

$$\int_0^T f_i(t)\ f_j^*(t)\ dt = \begin{cases} 1 & i=j \\ 0 & i\neq j \end{cases} \qquad 2.4\text{-}1$$

and given these functions, the new samples r_k' of the data are given by:

$$r_k = \int_0^T r(t)\ f_k(t) dt \qquad 2.4\text{-}2$$

we also need the re-sampled signal template:

$$s_k = \int_0^T s(t)\ f_k\ (t) dt \qquad 2.4\text{-}3$$

The eigenfunctions $f_k(t)$ are the solutions of the integral equation:

$$\lambda_j\ f_j\ (t) = \int_0^T f_j\ (x)\ R_n\ (t-x) dx \qquad 2.4\text{-}4$$

Now we may write the probability density functions for the new sample set, as:

$$p_s = \prod_{k=1}^{N} \left(\frac{1}{2\pi\lambda_k}\right)^{1/2} \exp\left[\frac{(r_k - s_k)^2}{-2\lambda_k}\right] \qquad 2.4\text{-}5$$

$$p_n = \prod_{k=1}^{N} \left(\frac{1}{2\pi\lambda_k}\right)^{1/2} \exp\left[\frac{r_k^2}{-2\lambda_k}\right]$$

and the detection test becomes:

$$\sum_{k=1}^{N} \frac{s_k r_k}{\lambda_k} \geq \ln \lambda_0 + \frac{1}{2} \sum_{k=1}^{N} \frac{s_r^2}{\lambda_k} \qquad 2.4\text{-}5$$

which is identical to 2.2-4 except that σ_n^2 has been replaced by the eigenvalues λ_k, and the signal samples have been transformed by a weighting function similar to the whitening filter of section 2.3. In this case, however, equations 2.4-2 through 2.4-4 can be written as sums over the time sampled values with no loss of generality, hence, with no degradation in performance caused by correlated samples.

2.5 Digital Filtering

The transformations of sections 2.2 through 2.4 can be written as filter transfer functions. Additionally, empirical methods can be used to synthesize a desired transfer function and the equations of those sections can then be used to evaluate the error rate and detection performance. This latter course is often followed when the sampling rate is constrained by some considerations other than those requiring uncorrelated noise samples. Typically the desired filtering is matched to the sample rate and the signal dwell time by the Nyquist theorem and we wish to evaluate the detection performance of such systems. Additionally, it may be desirable to further filter the data to improve the signal-to-noise ratio based on the observed noise spectrum. In this section we will discuss how such a transfer function could be synthesized and then derive the algorithm for converting that analog transfer function to a digital difference equation.

Given an analog impulse response function H(S), the difference equation for the filter function can be derived. Also, given the nominal characteristics desired, the transfer function can be synthesized. Both of these techniques are described below.

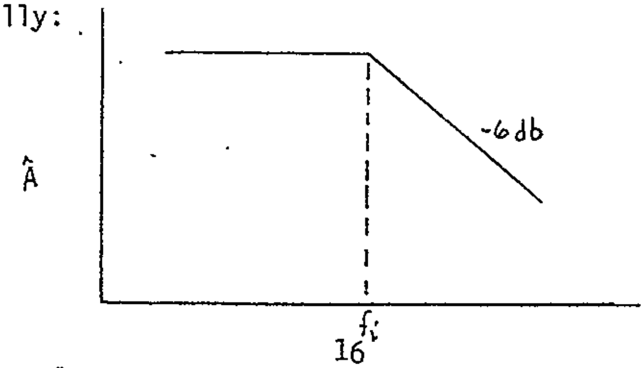
The frequency response can generally be described as a series of first-order filters. The transfer function of a low-pass filter is:

$$H_i(S) = \left(\frac{a_i}{a_i + S}\right) G_L \qquad 2.5\text{-}1$$

where $a_i = 2\pi f_i$, G_L is the gain of the filter.

f_i = the corner frequency of the filter (Hz).

Graphically:

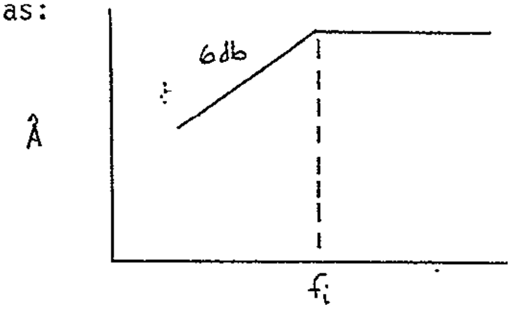


That is, the response of a low-pass filter is flat for $f < f_i$, and falls at 6 dB per octave (linearly on a log A-log f graph).

For a high-pass filter,

$$H_i(S) = \left(\frac{S}{a_i + S}\right) G_H \qquad 2.5\text{-}2$$

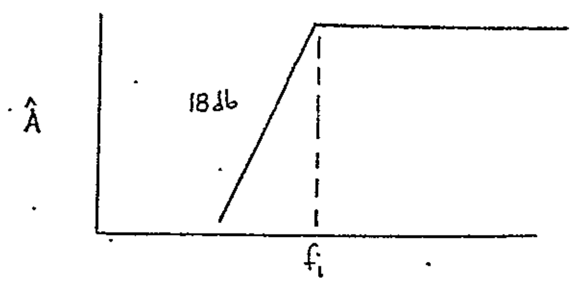
which appears as:



and the slope is the same as before. Higher order forms of these filters have transfer functions which are powers of the above H(S)'s, with the exponent n equal to the order of the filter. That is, a 3rd order high-pass filter is:

$$H(S) = \left(\frac{S}{a_i + S}\right)^3 G_H \qquad 2.5\text{-}3$$

and its response slope increases by a factor of n (3 in this example):



Finally, a circuit which can be described by a series of such filters has a transfer function which is a product of the elemental $H_i(S)$ terms, and a response curve which is a series of line segments with n(± 6 dB) quantum slope changes at each characteristic frequency.

A representative example is demonstrated by the following. The filter consists of a first-order high-pass filter of $f_i = 4$ Hz, and a second-order low-pass filter of $f_2 = 40$ Hz. In addition, the detector acts as a low-pass filter of order 1 at $f_3 = 1$ Hz. The overall transfer function is then:

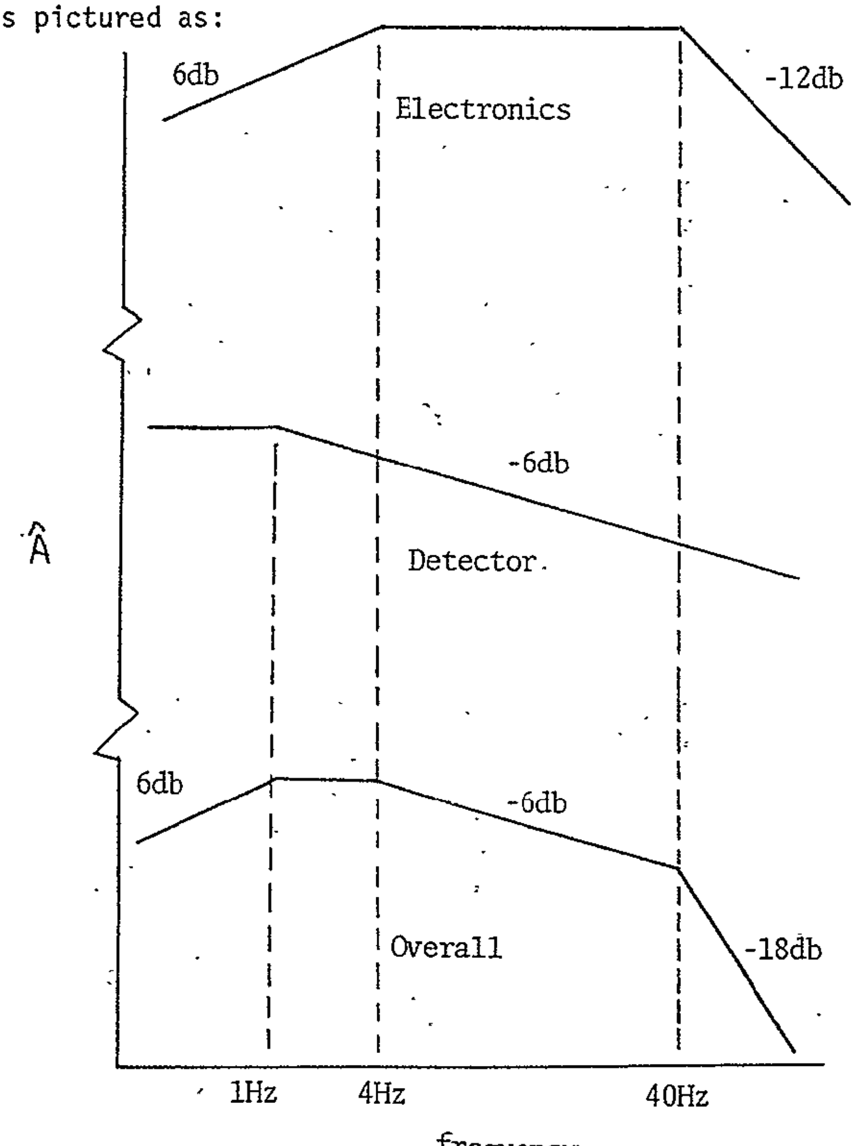
$$H(S) = \left(\frac{S}{a_3 + S}\right) \left(\frac{a_2}{a_2 + S}\right) \left(\frac{a_1}{a_1 + S}\right)^2 G \qquad 2.5\text{-}4$$

$a_3 = 2\pi \cdot 4$ Hz

$a_2 = 2\pi \cdot 1$ Hz

$a_1 = 2\pi \cdot 40$ Hz

which is pictured as:



The definition of the corner frequency is important here. If we have defined the f_i points as the traditional 3 dB or half-power response frequencies, then the values used in 2.5-1 and 2.5-2 for a_i must be altered somewhat. Note, for example, that for a single order filter (m=1), we have the half-power response at

$$H(S)\,H^*(S) = \frac{1}{2} = \frac{a_i^2}{a_i^2 + S^2} \Rightarrow S = a_i \qquad 2.5\text{-}5$$

as expected. For an m^{th} order filter, we find

$$H(S)\,H^*(S) = \frac{1}{2} = \left(\frac{a_i^2}{a_i^2 + S^2}\right)^m \Rightarrow S^2 = (\sqrt[m]{2} - 1)\, a_i^2 \qquad 2.5\text{-}6$$

and similarly, for a high-pass filter:

$$S = a_i/(\sqrt[m]{2}-1)^{1/2} \qquad 2.5\text{-}7$$

Given the transfer function H(S), the difference equation can now be determined as follows. First, transform the frequencies. Since we desire the digital equivalent frequency, determine A_i by:

$$A_i = \mathrm{TAN}\left(\frac{2\pi f_i T}{2}\right) = \mathrm{TAN}\left(\frac{a_i T}{2}\right) \qquad 2.5\text{-}8$$

where T is the sampling interval, = 1/SR (SR is the sample rate in Hz). Second, transform the H(S) function to an H(Z) function by the substitution:

$$S \rightarrow \frac{Z-1}{Z+1} \qquad 2.5\text{-}9$$

This transformation preserves the square-magnitude response of the system except for a warping of the frequency scale as given by the first relationship (2.5-5). The advantage of this transformation is that aliasing is not introduced by the digital representation, thus avoiding the necessity of "guard" filters (anti-aliasing) which would result in a digital filter of higher order than the original analog transfer function.

The technique then is to express H(Z) in terms of a polynomial in Z, as:

$$H(Z) = G * \frac{P(Z)}{Q(Z)} \qquad 2.5\text{-}10$$

with the order of Q(Z) equal to or greater than P(Z). Most easily, the substitution used is:

$$(A_i + S) \rightarrow \frac{(A_1 + 1)(Z + U_i)}{(Z+1)} \qquad 2.5\text{-}11$$

where $U_i = \frac{(A_i - 1)}{(A_i + 1)}$ and $P_i = A_i + 1$

The numerator and denominator of H(Z) are then divided by Z^n, where n is the order of the denominator, resulting in a transfer function which is a ratio of two polynomials of equal order in powers of Z^{-1}. Equation 2.5-9 is used for S terms in the numerator. The general form of this H(Z) is:

$$H(Z) = \frac{Z^{-n}(Z+1)^{n-m}\,(Z-1)^m \prod_{i=1}^{(n-m)} A_i}{\left(1 + \sum_{i=1}^{n} S_i\, Z^{-i}\right) \prod_{i=1}^{n} P_i} \prod_{i=1}^{n} G_i \qquad 2.5\text{-}12$$

where:

A_i is as defined previously;

n = the number of elemental filters;

m = the number of high-pass filters, and the A_i's are ordered with the low-pass frequencies first.

$P_i = A_i + 1$

and

S_i are the expanded coefficients of the product function:

$$\prod_{i=1}^{n} (U_i + Z)$$

such that:

$$S_1 = \sum_{i=1}^{n} U_i$$

$$S_2 = \sum_{j=2}^{n} U_j \left(\sum_{i=1}^{j=1} U_i \right) \qquad 2.5\text{-}13$$

$$S_3 = \sum_{j=3}^{n} U_R \left[\sum_{j=2}^{k-1} U_j \left(\sum_{i=1}^{j-1} U_i \right) \right]$$

$$S_r = \sum_{m=r}^{n} U_m \sum_{\ell=r-1}^{m-1} U_\ell \sum_{k=4-2}^{\ell=1} [\ldots . \left(\sum_{i=1}^{j-1} A_i \right)]$$

Again from our example 2.5-4, the H(S) transforms to:

$$H(Z) = \frac{Z^{-4}\ (Z-1)(Z+1)^3}{Z^{-4}(Z+U_1)^2(Z+U_2)(Z+U_3)} \ \frac{A_1^{\ 2}\ A_2}{P_1^{\ 2}\ P_2\ P_3}\ G \qquad 2.5\text{-}14$$

The Z-transform corresponds exactly with 2.5-12 if we consider the four elements of the filter as having frequencies (in the original form) of 1, 4, 40, and 40 Hz. That is, U_1 is repeated, but treated as if it were two different terms. Ignoring the constant factor temporarily:

$$H(Z) = \frac{1-2Z^{-1} - 2Z^{-3} - Z^{-4}}{1 + \sum_{i=1}^{4} S_i\ Z^{-1}} \qquad 2.5\text{-}15$$

where $S_1 = 2U_1 + U_2 + U_3$

$$S_2 = U_1U_1 + 2U_1U_2 + 2U_1U_3 + U_2U_3 \qquad 2.5\text{-}16$$

$$S_3 = U_1U_1U_2 + U_1U_1U_3 + 2U_1U_2U_3$$

$$S_4 = U_1^2 U_2U_3$$

Now that we have the Z-transform of the H(S) transfer function, the difference equation for the system can be written. Noting that Z^{-1} is the unit delay function, and writing H(Z) as:

$$H(Z) = \frac{1 + \sum_{i=1}^{n} T_i Z^{-i}}{1 + \sum_{i=1}^{n} S_i Z^{-i}} = \frac{Y(Z)}{X(Z)} \qquad 2.5\text{-}17$$

where Y(Z) is the Z-transform of the output, and X(Z) is the input transform. Inverting the transform we find:

$$Y_j = X_j + \sum_{i=1}^{n} T_i X_{j-i} - \sum_{i=1}^{n} S_i Y_{j-i} \qquad 2.5\text{-}18$$

where Y_j is the jth sample of the output series, and X_j is the corresponding jth input value. Continuing our example,

$$Y_n=X_n-2X_{n-1}-2X_{n-3}-X_{n-4}-S_iY_{n-1}-S_2Y_n-2-S_3Y_{n-3}-S_4Y_{n-4} \qquad 2.5\text{-}19$$

It is interesting to note that because the transformation 2.5-9 is bilinear, the difference equation will always be the same order in X_{n-i} and Y_{n-i} terms (except for canceling of some Z^i terms by the expansion of $(Z+1)^m(Z-1)^\ell$).

To find the T_i coefficients of equations 2.5-17 and 2.5-18, we must expand

$$(1+Z)^{n-m}(1-Z)^m = 1+T_1 Z+\ldots + T_nZ^n \qquad 2.5\text{-}20$$

but the gain constant in 2.2-12 of

$$\prod_{i=1}^{m} A_i \prod_{i=1}^{n} \left(\frac{G_i}{P_i}\right) = COEFF \qquad 2.5\text{-}21$$

must be included as a factor of all the X_i terms.

Expanding 2.5-20 can be done using the binomial expansion subroutine attached to expand $(1+Z)^{n-m}$ and $(1-Z)^m$, and the polynomial product routine to find the resulting terms. Then if we redefine the subscript of T by one to absorb the 1 on the r.h.s. of 2.5-20, we can include the factor 2.5-21 easily into the definition of T_i:

$$(1+Z)^{n-m}(1-Z)^m - T_1 + T_2 Z + \ldots + T_{n+1} Z^n \qquad 2.5\text{-}22$$

if we also redefine the S_i and set $S_1 = 0$, then the relations 2.2-17 and 2.2-18 can be written:

$$Y_i = \sum_{i=1}^{n+1} T_i X_{j-i+1} - \sum_{i=1}^{n+1} S_i Y_{j-i+1} \qquad 2.5\text{-}23$$

$$= \sum_{i=1}^{n+1} (T_i X_{j-i+1} - S_i Y_{j-i+1})$$

This redefinition of S_i has been included in the attached algorithm (see section 3.1).

ORIGINAL PAGE IS
OF POOR QUALITY

3.0 ALGORITHMS AND SOFTWARE

Section 3 of the first report covered in detail the basic survey point-source processing scheme. However, in order to successfully complete the sky survey and to define the detection test gates, a number of peripheral routines are needed. This section will discuss several of the most important of these routines. Some routines are simple, such as coordinate transformations used for positional matching. Others, although complex and worthy of discussion, are very specifically written for each mission. These routines generally are part of the primary data processing system and are assembled from the formula and algorithms of section 2 of this report and section 3 of the previous paper. Yet, other routines which are a part of the data base merging are decisions and tests for specific types of astronomical sources and depend on the sensor bandpasses and sensitivities and on the spectral characteristics of the sources being searched for. Some routines which are only peripheral to the primary detection scheme are so basic and important that they are worthy of individual discussion here. The complete set of programs designed and tested on the Celestial Mapping Program (CMP) data will be published at a later date when that task is completed.

Two basic programs will be covered here and one front-end detection scheme used on a previous survey program. First, we will discuss an application of the digital filter design scheme of section 2.5. Then the calibration of infrared detectors is discussed and a routine to evaluate the spectral response of a detector plus filter combination to a variety of stellar spectra.

3.1 A Digital Filter Design Aid

The discussion of section 2.5 covered the algorithm for digital filter design; here, we consider the specific use of the following routines. The program attached does two things. First, the coefficients of the difference equation

$$Y_j = \sum_{i=1}^{N+1} (T_i X_{j-i+1} - S_i Y_{j-i+1})$$

where the X_i's are the input data, Y_i's the output data, and S_i and T_i the filter coefficients determined from the desired frequency response.

Second, the routine creates a pair of sample response sequences. One is the impulse response function of the filter. If the filter characteristics were chosen to duplicate the response of a detector and its electronics, then this impulse response will model the radiation-hit response. The other response is the system reaction to a square wave. Since the duration of the square wave is equal to the point-source dwell time, the response is approximately the same as a source signature.

```
      PROGRAM TRNSFN(INPUT,OUTPUT)
      COMPLEX C,SUM,COEFF,T
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ****
C                                                                        ****
C           THE FOLLOWING DIMENSION STATEMENT IS USED TO CREATE          ****
C          A SAMPLE RESPONSE SEQUENCE FOR A RECTANGULAR INPUT            ****
C          EQUAL IN LENGTH TO THE STAR DWELL TIME, AND THE IMPULSE       ****
C          RESPONSE SEQUENCE.                                            ****
C                                                                        ****
      DIMENSION XX(200),XP(200),YY(200),YP(200),TIM(200)                 ****
C                                                                        ****
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ****
C
C     THE ROUTINE IS DIMENSIONED FOR TRANSFER FUNCTIONS OF TOTAL ORDER
C     9 OR LESS.  FOR HIGHER ORDERS, CHANGE THE DIMENSIONS OF ALL THE
C     FOLLOWING TO N+1, WHERE N IS THE DESIRED ORDER.
C
      DIMENSION C(10),SUM(10),T(10),X(10),Y(10),Z(10),RS(10),TR(10)
C
      PRINT 99
   99 FORMAT(1HR)
      A1=1HC  $  A2=1HS  $  A3=1HT  $  A4=1HF
C
C     N AND M DESCRIBE THE ORDER OF THE TRANSFER FUNCTION.
C     N = THE TOTAL ORDER
C     M = THE ORDER OF THE HIGH-PASS FUNCTIONS
C
C     THE PROGRAM WILL LOOP FOR NEW TRANSFER FUNCTIONS.  TO END,
C     SET M = 0.  (THE LAST DATA CARD CAN BE A BLANK TO STOP)
C
   50 CONTINUE
      READ 100,N,M
  100 FORMAT(2I5)
      IF(N.EQ.0) GO TO 51
C
C     THE SAMPLE RATE SR GOVERNS THE FREQUENCY WARPING FOR THE S TO Z
C     TRANSFORMATION.
C     THE DWELL TIME IS DETERMINED FROM THE SCAN RATE SCNRTE AND
C     THE DETECTOR SIZE SIZE.
C
C     SIZE AND SCNRTE ARE VARIABLES USED ONLY FOR GENERATING
C     THE TEST CASES.
C
      READ 101,  SR,SIZE,SCNRTE
```

```
  101 FORMAT(3F10.0)
      LN=INT((SR*SIZE/SCNRTE)+0.5)
C
C     LIST THE INPUT PARAMETERS
C
      PRINT 201,N,M, SR,LN
  201 FORMAT(1H1,//,20X,*N = *,I5,*      M = *,I5,*      SAMPLE RATE = *,F
     110.4,*      DWELL = *,I5,//)
C
C     THE CORNER FREQUENCIES FOR THE FILTER ELEMENTS CAN BE SPECIFIED
C     AS COMPLEX (REAL + IMAGINARY)
C     THEY SHOULD BE ORDERED WITH THE M HIGH-PASS ELEMENTS LAST
```

```
      C      EACH ELEMENT HAS A (COMPLEX)GAIN WHICH CANNOT BE ZERO
      C      THE GAIN CAN BE SET AS (1.00,0.00)
      C      THE ORDER MUST BE EGUAL TO OR GREATER THAN ONE
      C
      C      THE FREQUENCY CARDS ARE FORMATTED (4F10.6,I5) AS FOLLOWS:
      C      FREQ(REAL),FREQ(IMAG),GAIN(REAL),GAIN(IMAG),ORDER
      C
             PI=3.141592653579
             COEFF=CMPLX(1.0,0.0)
             NT=0
             DO 1 I=1,N
             READ 102,WDR,WDI,GAINR,GAINI,NORD
         102 FORMAT(4F10.6,I5)
             PRINT 205, I,WDR,WDI,NORD
         205 FORMAT(1H ,10X,*WD(*,I2,*) = *,2F10.6,5X,*ORDER = *,I5)
             NN=NORD
             IF((NT+M).GE.N) GO TO 7
             WDR=WDR/(2.**(1./NORD)-1.)
             WDI=WDI/(2.**(1./NORD)-1.)
             GO TO 8
           7 WDR=WDR*(2.**(1./NORD)-1.)
             WDI=WDI*(2.**(1./NORD)-1.)
           8 CONTINUE
             WAI=TAN(WDI*PI/ SR)
             WAR=TAN(WDR*PI/ SR)
      C
      C      LIST THE FREQUENCIES
      C
         131 CONTINUE
             NT=NT+1
             PRINT 202, NT,WDR,WDI,I,WAR,WAI,NT,GAINR,GAINI
         202 FORMAT(1H ,10X,*WD(*,I2,*) = *,2F10.6,5X,*WA(*,I2,*) = *,2F10.6,5X
            1,*GAIN(*,I2,*) = *,2F10.6)
             C(NT)=CMPLX(WAR-1.,WAI)/CMPLX(WAR+1.,WAI)
             COEFF=COEFF*CMPLX(GAINR,GAINI)/CMPLX(WAR+1.,WAI)
             IF(NT.LE.(N-M))  COEFF=COEFF*CMPLX(WAR,WAI)
             NN=NN-1
             IF(NN.GT.0) GO TO 131
             IF(NT.EQ.N) GO TO 141
           1 CONTINUE
         141 CONTINUE
      C
      C      LIST THE C(I) TERMS AND THE VALUE OF COEFF
      C
             IB=0
```

```
  203 FORMAT(1H0,//,((10X,A1,*(*,I2,*) = *,2F12.8,/)) )
C
C     DETERMINE THE S(I) COEFFICIENTS OF THE Y(J-I) TERMS
C
      DO 2 I=1,N
    2 SUM(I)=CMPLX(0.0,0.0)
      DO 4 I=1,N
      SUM(1)=SUM(1)+C(I)
      IF(I.EQ.N) GO TO 4
      DO 3 J=2,N
      IF(I+J-1.LE.N) SUM(J)=SUM(J)+SUM(J-1)*C(I+J-1)
```

```
    3   CONTINUE
    4   CONTINUE
        DO 5 I=1,N
    5   SUM(N+2-I)=SUM(N+1-I)
        SUM(1)=CMPLX(0.0,0.0)
C
C       LIST THE S(I) COEFFICIENTS
C
        NN=N+1
        PRINT 203, (A2,I,SUM(I),I=1,NN)
C
C       EXPAND THE NUMERATOR TO DETERMINE THE T(I) COEFFICIENTS OF X(J-I)
C
C       NOTE THAT T(I) HAS EXACTLY (N-M+1)+(M+1)-1 = N+1 TERMS
C
        NP=N-M
        CALL BINEXP(X,NP,+1.00)
        IDIMX=NP
        NP=M
        CALL BINEXP(Y,NP,-1.00)
        IDIMY=NP
        CALL PMPY(X,IDIMX,Y,IDIMY,Z,IDIMZ)
        DO 6 I=1,IDIMZ
        T(I)=Z(I)*COEFF
        TR(I)=REAL(T(I))
        RS(I)=REAL(SUM(I))
    6   CONTINUE
C
C       NOW LIST THE T(I) COEFFICIENTS OF THE Y(J-I) TERMS
C
        PRINT 203, (A3,I,T(I),I=1,IDIMZ)
C
C       NOW ALL THE FACTORS ARE DETERMINED.
C       WE MAY NOW CALCULATE THE RESPONSE TO AN ARBITRARY INPUT SEQUENCE
C       OF X(I) VALUES AS
C
C       Y(J) = SUM(I=1 TO N+1) OF (( T(I)*X(J-I+1)-S(I)*Y(J-I+1) ))
C
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ****
C                                                                     ****
C           THE FOLLOWING CARDS MARKED  ****  CREATE A SAMPLE RESPONSE ****
C           SEQUENCE. THE INPUT RECTANGULAR PULSE IS CREATED IN XX(I)  ****
C           WITH A LENGTH EQUAL TO THE STAR DWELL TIME FOR THE INSTRUMENT ****
C          INPUT.  THE RESPONSE SEQUENCE IS IN YY(I).  THE IMPULSE     ****
```

```
C           OUTPUT IS NUMBERED SEQUE  ALLY AND THE SAMPLE TIME IS          ****
C           GIVEN IN MILLISECONDS,                                         ****
C                                                                          ****
        DO 11 I=1,200                                                      ****
        TIM(I)=0,0                                                         ****
        IF(I,GE,10) TIM(I)=1000,*(I-20)/ SR
        XX(I)=0,00  $  XP(I)=0,00                                          ****
        YY(I)=0,00  $  YP(I)=0,00                                          ****
   11   CONTINUE                                                           ****
        LL=19+LN
        DO 12 I=20,LL
```

```
   12  XX(I)=1.00                                                      ****
C                                                                      ****
C
C      GENERATE GAUSSIAN NOISE AND ADD IT TO THE INPUT DATA.
C
       CALL RANSET(863211)
       A=0.
       NOISE=1  $  SNR=100.
       NOISE=1  $  SNR=10.
       NOISE=0
       IF(NOISE.EQ.0) GO TO 123
       DO 122 I=10,200
       GNOISE=0.0
       DO 121 J=1,12
  121  GNOISE = GNOISE+ RANF(A)
       GNOISE = (GNOISE-6.0)/SNR
       XX(I)=XX(I)+GNOISE
  122  CONTINUE
  123  CONTINUE
C
       XP(10)=1.00                                                     ****
       IL=N+1                                                          ****
       JF=N+2                                                          ****
       YYM=-9999.  $YPM=-9999.
       DO 14 J=JF,200                                                  ****
       DO 13 I=1,IL                                                    ****
       YY(J)=YY(J)+TR(I)*XX(J-I+1)-RS(I)*YY(J-I+1)                     ****
       YP(J)=YP(J)+TR(I)*XP(J-I+1)-RS(I)*YP(J-I+1)                     ****
   13  CONTINUE                                                        ****
       IF(YP(J).GT.YPM) YPM=YP(J)                                      ****
       IF(YY(J).GT.YYM) YYM=YY(J)                                      ****
   14  CONTINUE                                                        ****
       DO 15 J=1,200                                                   ****
       YY(J)=YY(J)/YYM  $  YP(J)=YP(J)/YPM                             ****
   15  CONTINUE                                                        ****
       PRINT 300, YYM,YPM                                              ****
  300  FORMAT(1H1,10X,*YYMAX = *,F10.6,*      YPMAX = *,F10.6,//)      ****
       PRINT 111, (I,TIM(I),XX(I),YY(I),XP(I),YP(I),I=1,200)           ****
  111  FORMAT(1H ,*     I              MSEC             X             Y ****
      1       XP               YP*,/,((3X,I3,5(5X,F10.3))))            ****
C                                                                      ****
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ****
C
       GO TO 50
   51  CONTINUE
```

```
      STOP
      END

      SYMBOLIC REFERENCE MAP (R=1)

 ENTRY POINTS
    45  TRNSFN
```

3.2 Infrared Filter Calibration

The calibration of infrared brightnesses is the single most difficult aspect of a scanning sky survey. For ground-based point-and-integrate systems, it is possible, and in fact common, to make all measurements of the same signal-to-noise ratio by varying the integration. Since the amplitude uncertainty was shown to be a function of the signal-to-noise value in section 2.2, it is clear that uniform photometric accuracy is readily achieved. For survey instruments, the uncertainty of the initial measurements is inversely related to the signal-to-noise value, applying a fundamental limit to the accuracy of the survey measurements which varies both with brightness of the source and its location in the sky.

Further complicating the problem is the fact that the sources have a wide variety of spectrum so that the broad band detectors typical of infrared survey instruments do not have a well-defined intrinsic calibration. It is possible to calibrate the detector voltage in terms of the watts per cm^2 it receives. However, if the survey is measure sources in several colors, or if the calibrations are to be derived from measurements made in a difficult wavelength region, the measurements must be referred to a spectral intensity. The wavelength bandwidth that is needed, however, is dependent on the spectrum of the source being measured. Furthermore, the effective wavelength of that measurement varies with the input spectrum.

The units of the brightness measurement are another problem. The most useful form is the brightness magnitude, defined by

$$m = -2.5 \log_{10} \left(\frac{B}{B_0}\right) \qquad 3.2\text{-}1$$

where B is the observed brightness and B_0 is the reference value. This reference is different for every filter, since it is defined as the response that filter-detector would observe from a particular "standard" star - the archetype is α-Lyra, which is defined as a

10,000°K blackbody source with an angular diameter of 1.5697 E-16 steradians. The great benefit of this magnitude measurement is that we skirt the question of effective bandwidth. These magnitude measurements still need an effective wavelength, but for blackbody spectra at wavelengths less than 50 micrometers, the effective wavelength varies only very slowly until the source temperature falls below 500°K. Finally, the magnitude measurements defined by 3.2-1 can be used inversely to find the equivalent blackbody color temperature if measurements are available in two or more bands.

Figures 3, 4, and 5 show the variations in bandwidth, effective wavelength, and magnitude difference for three infrared filters similar to ones commonly used in previous surveys and measurements. The results were derived from the attached filter calibration routine which is self-explanatory.

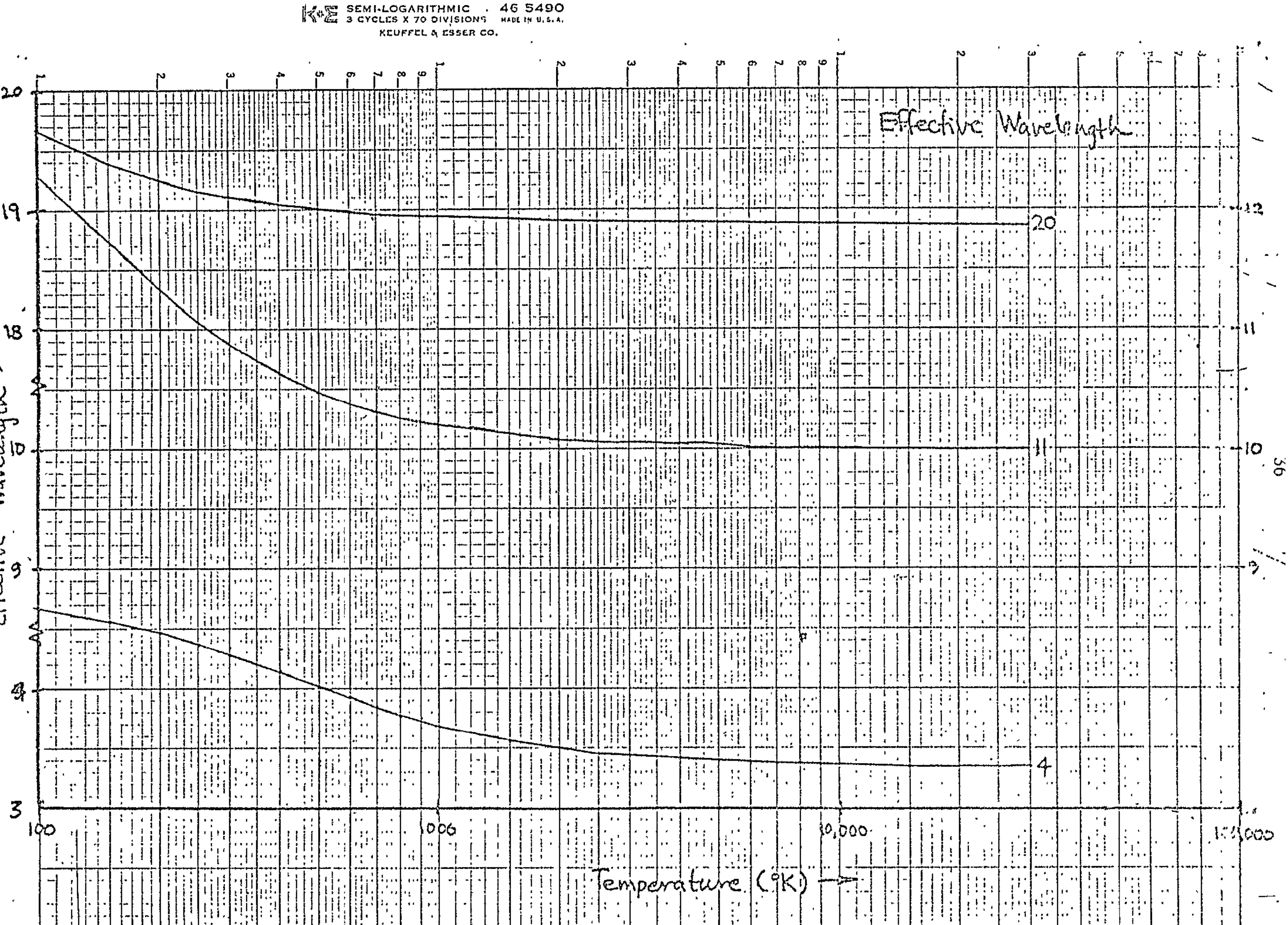

K+E SEMI-LOGARITHMIC 46 5490
3 CYCLES X 70 DIVISIONS MADE IN U.S.A.
KEUFFEL & ESSER CO.
Effective Wavelength
20
11
4
100
1000
10,000
100,000
Temperature (°K) →

Figure 4.

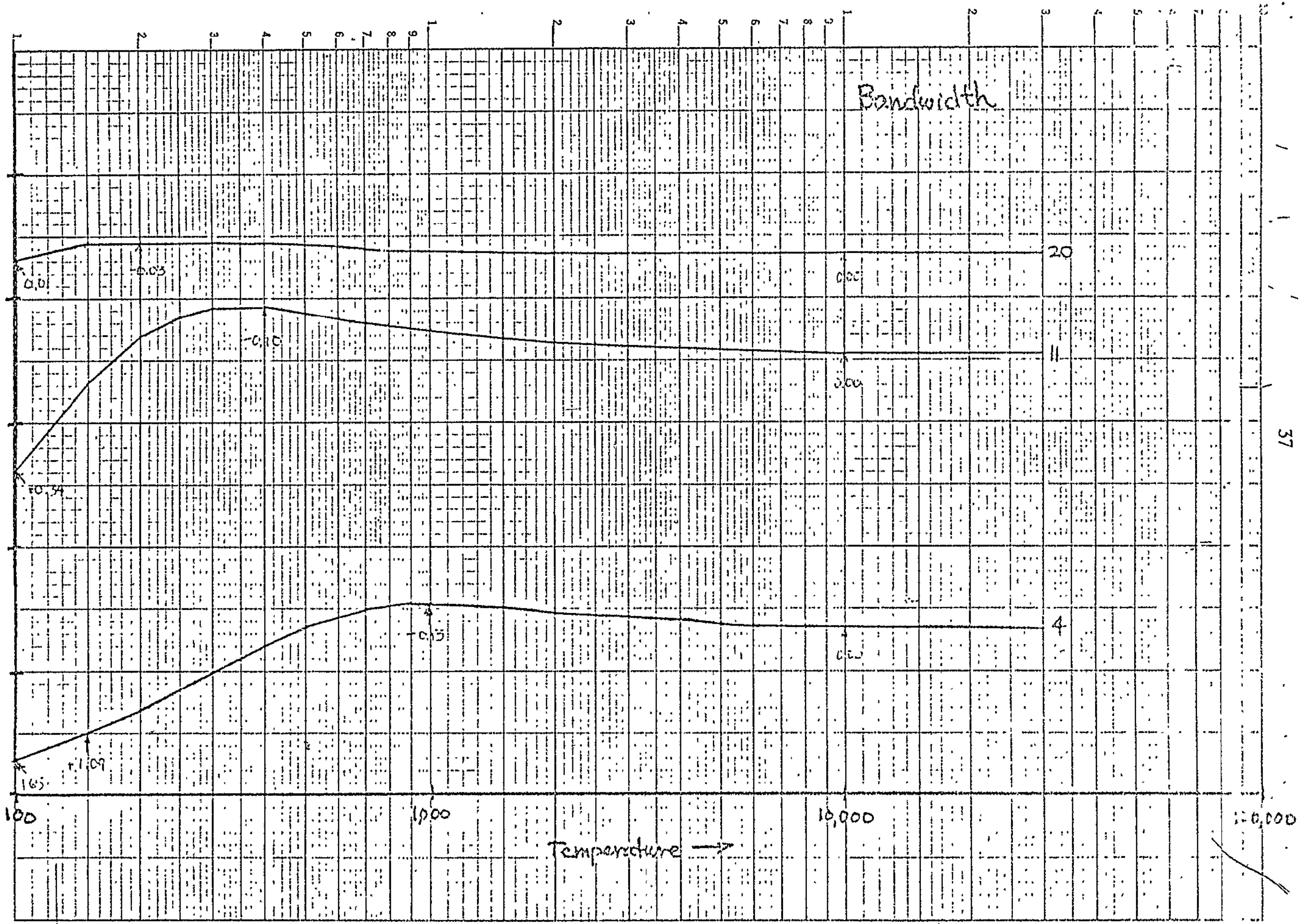


Figure 5.

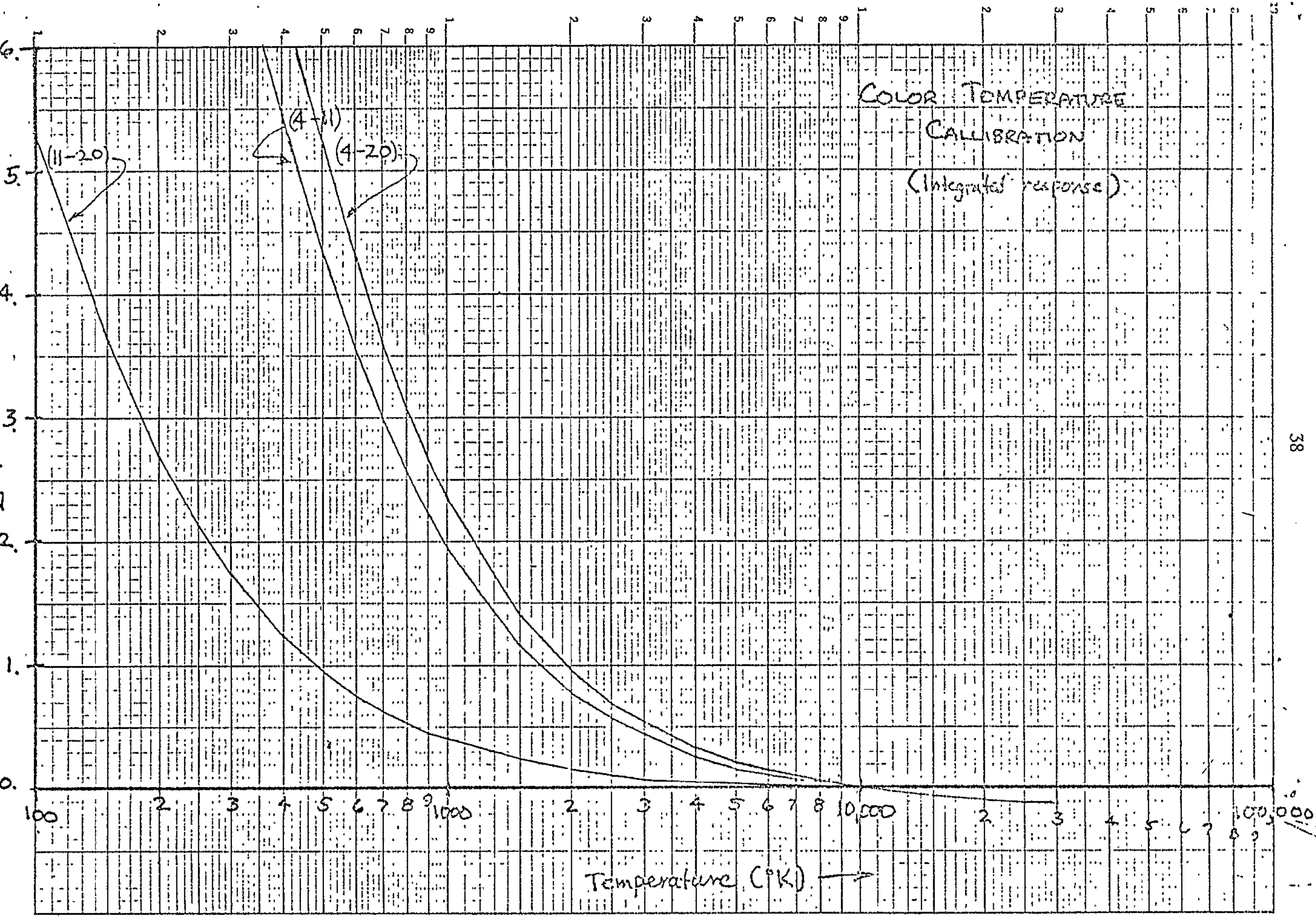

COLOR TEMPERATURE
CALLIBRATION
(Integrated response)
(11-20)
(4-11)
(4-20)
6.
5.
4.
3
2.
1.
0.
100
1000
10,000
100,000
Temperature (°K)

Figure 6.

```
?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?*?
PELZM,
ACCOUNT,SPTHEM,T3025,
MAP,OFF,
FTN.
REWIND,PUNCH,
COPY,PUNCH,OUTPUT,
EXIT,

      PROGRAM INTGRT (INPUT,OUTPUT,PUNCH,TAPE5=INPUT,TAPE6=OUTPUT)
C     PLANCK2  (MODIFIED)
C--------------INTEGRATION CAN BE DONE WITH OR WITHOUT THE PLANCK FUNCTION,
C     PROGRAM FOR INTEGRATING THE PLANCK BLACKBODY RADIATION FUNCTION
C     OVER AN INTERVAL DETERMINED AND ATTENUATED BY FILTER-SENSITIVITY
C     TYPE FUNCTIONS,
C     DATA SHOULD BE IN ORDER OF INCREASING WAVELENGTH.
C     NFUNCT   = NO, OF RESPONSE FUNCTION
C     IRITE    = 1 - WRITE WAVELENGTH, INTENSITY, RADIANCE, PLANCK INTENSITY,
C              = 0 - DO NOT WRITE,
C     NFLUX    = 1 - PLANCK FUNCTION TO BE CALCULATED,
C              = 0 - PLANCK FUNCTION NOT TO BE CALCULATED,
C     IWAVTR   = 1 - TRANSMISSION DATA DECK CONSISTS OF WAV, AND TRANS,
C              = 0 - TRANSMISSION DATA DECK CONSISTS OF TRANS, DATA ONLY,
C     NORMAL   = 1 - NORMALIZED FUNCTION1(I) * FUNCTION2(I)
C              = 0 - DO NOT NORMALIZE
C     IPLOT    = 1 PLOT
C              = 0 DO NOT PLOT
C     NTRANS   = NO, OF SETS OF TRANSMISSION DATA PER RESPONSE FUNCTION,
C     M        = NO, OF FUNCTIONS (1 OR 2),
C     N        = NO, OF WAVELENGTH PER FUNCTION (ODD INTEGER),
C     L        = NO. OF TEMP. TO BE CALCULATED (IF THERE ARE NONE L=1),
C     EXPER    = NAME OF EXPERIMENT, DATA, ETC, FORMAT 4A6,
C     TYPE     = IDENTIFYING NAME OF DATA, FORMAT 4A6,
C         CARD 1  NFUNCT,IRITE,NFLUX,IWAVTR,NORMAL,IPLOT  (6I3)
C         CARD 2  NTRANS                                  (1I3)
C         CARD 3  M,N,L,                                  (1I1,2I3)
C         CARD 4  TEMP  (I=1,L)                           (9F8,2)
C         CARD 5  EXPER (K=1,4) - WAV,RES (I=1,N)         (4A6/(12F6,4))
C         CARD 6  TYPE (K=1,4) - WAV,TRAN OR TRAN (I=1,N) (4A6/(12F6,4))
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C    REFERENCE FLUX  IS THE INTEGRATED FLUC OF A 10,000 DEGREE B.B, OF
C         SIZE 1,5697E-16 STR (ALPHA LYRA) . . .  THE ZERO MAGNITUDE REF
C         AND IS INTEGRAL OF B(LAMBDA,10,000)*R(LAMBDA)*DLAMBDA*OMEGA
C
```

39

ORIGINAL PAGE IS
OF POOR QUALITY

```
C     DEFINED WAVELENGTH IS THE FLAT RESPONSE EFFECTIVE WAVELENGTH
C
C     BDWDTH ZERO IS THE FLAT RESPONSE BANDWIDTH
C
C     WAVELENGTH IS THE TRUE EFFECTIVE WAVELENGTH FOR B,B, SPECTRA AT
C         THE GIVEN TEMPERATURE
C         THE TRUE EFFECTIVE WAVELENGTH IS THE INTEGRAL OF B(LAMBDA,T)*
C         R(LAMBDA)*LAMBDA*DLAMBDA DIVIDED BY THE INTEGRAL OF B(LAMBDA,T)
C         *R(LAMBDA)*DLAMBDA
C
C     BDWDTH IS THE TRUE BANDWIDTH AT THE TRUE EFFECTIVE WAVELENGTH FOR
C         THE GIVEN TEMPERATURE
C         THE TRUE EFFECTIVE BANDWIDTH  IS THE INTEGRAL OF B(LAMBDA,T)*
C         R(LAMBDA)*DLAMBDA DIVIDED BY B(LAMBDA EFF,T)
C
C     BDWDTHDF IS THE PROPER BANDWIDTH FOR THE DEFINED EFFECTIVE WAVELENGTH
C
C     BR MAG IS -2,5*LOG(B(LAMBDA EFF,T)/B0(LAMBDA 0,10))  WHERE B(LAMBDA,T
C         IS THE B,B, EMISSION
C
C     COL MAG IS -2,5*LOG(INT FLUX/REFERENCE FLUX)
C
C     INT FLUX IS THE INTEGRATED FLUX ON THE DETECTOR  IN W*CM-2*STR-1
C
C     VIEW FLUX IS THE INTEGRATED FLUX MULTIPLIED BY THE FIELD OF VIEW
C          = 4,56-06 STR  AND IS IN W*CM-2
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      REAL MAGFAC
      DIMENSION WAV(150),RES(150),TRAN(150),TEMP(150),W(150),WX(150),WXX
     1(150),RESP(150),WWX(150),TYPE(4),EXPER(4)
      NJ=0
      OMEGA=1,5697E-16
      PI=3,141592654
      ICOL=0
      L=0
      READ (5,94) NFUNCT,IRITE,NFLUX,IWAVTR,NORMAL,IPLOT
 94   FORMAT(6I3)
 95   NJ=NJ+1
      NFLUX=0
      DO 92 I=1,4
92    TYPE(I)=6H
      LOLD=L
      ICOL=ICOL+1
```

```
      NN=0
      READ (5,99) NTRANS
99    FORMAT(1I3)
      READ (5,100) M,N,L
100   FORMAT(1I1,2I3)
      TEMP(1)=10000.
C     INSERT TEMP DEFINING CARDS; L=XX, TEMP(2)=XX,ETC,
      L=70
      TEMP(2)=100.
      DO 20 I=3,42
20    TEMP(I)=TEMP(I-1)+10.
      DO 21 I=43,52
21    TEMP(I)=TEMP(I-1)+50.
      DO 22 I=53,70
22    TEMP(I)=TEMP(I-1)+500.
      N1=1  $  N2=N
      READ(5,101) (EXPER(K),K=1,4),(WAV(I),RES(I),I=N1,N2)
 101  FORMAT(4A6/(12F6.4))
      L1=N+1
      DO 500 I=L1,150
      WAV(I)=0.00
      RES(I)=0.00
      TRAN(I)=0.00
      RESP(I)=0.00
 500  CONTINUE
      N2=N
98    NN=NN+1
      IF(M-1) 1,1,3
 3    IF(IWAVTR.EQ.1) GO TO 601
      READ (5,112) (TYPE(K),K=1,4),(TRAN(I),I=1,N)
 112  FORMAT(4A6/(12F6.4))
      GO TO 600
 601  READ (5,112) (TYPE(K),K=1,4),(WAV(I),TRAN(I),I=1,N)
 600  CONTINUE
      DO 2 I=1,N,1
 2    RESP(I)=RES(I)*TRAN(I)
      GO TO 11
1     DO 9 I=1,N,1
      TRAN(I)=0.
9     RESP(I)=RES(I)
887   CONTINUE
 11   CONTINUE
      IF(NORMAL.EQ.1) GO TO 700
      BIGRP=1.
      N1=1
```

```
220   NSPACE=0
      LINE=0
      WRITE (6,212) (EXPER(K),K=1,4),(TYPE(K),K=1,4)
212   FORMAT(1H1,37X,4A6/  41X,19HTABLE OF INPUT DATA/  38X,4A6//20X,
     110HWAVELENGTH,3X,11HFUNCTION(1),3X,11HFUNCTION(2),5X,10HWAVELENGTH
     2,3X,11HFUNCTION(1),3X,11HFUNCTION(2)//)
      IBG=N1  $  IND=N2
      DO 114 I=IBG,IND
      NSPACE=NSPACE+1
      LINE=LINE+1
      IF(5-NSPACE)204,206,206
206   WRITE (6,210) WAV(I),RES(I),TRAN(I),WAV(I+35),RES(I+35),TRAN(I+35)
210   FORMAT(22X,F5.2,7X,F7.4,7X,F7.4,10X,F5.2,7X,F7.4,7X,F7.4)
      GO TO 114
204   NSPACE=1
      IF(35-LINE)219,214,214
214   WRITE (6,211) WAV(I),RES(I),TRAN(I),WAV(I+35),RES(I+35),TRAN(I+35)
211   FORMAT(1H0,21X,F5.2,7X,F7.4,7X,F7.4,10X,F5.2,7X,F7.4,7X,F7.4)
      GO TO 114
219   N1=I+LINE-1
      IF(N2-N1)250,220,220
114   CONTINUE
      GO TO 250
700   BIGRP=RESP(1)
      DO 30 I=2,N
      IF(BIGRP-RESP(I))32,30,30
32    BIGRP=RESP(I)
30    CONTINUE
      DO 33 I=1,N
      RESP(I)=RESP(I)/BIGRP
33    CONTINUE
      N1=1
720   NSPACE=0
      LINE=0
      WRITE (6,712) (EXPER(K),K=1,4),(TYPE(K),K=1,4)
712   FORMAT(1H1,46X,4A6/  50X,19HTABLE OF INPUT DATA/  47X,4A6//10X,
     110HWAVELENGTH,3X,11HFUNCTION(1),3X,11HFUNCTION(2),3X,
     211HFUNCT.(1X2),5X,10HWAVELENGTH,3X,11HFUNCTION(1),3X,
     311HFUNCTION(2),3X,11HFUNCT.(1X2)/51X,10HNORMALIZED,47X,
     410HNORMALIZED//)
      IBG=N1  $  IND=N2
      DO 614 I=IBG,IND
      NSPACE=NSPACE+1
      LINE=LINE+1
      IF(5-NSPACE)704,706,706
```

```
 706  WRITE (6,710) WAV(I),RES(I),TRAN(I),RESP(I),WAV(I+35),RES(I+35),
     1TRAN(I+35),RESP(I+35)
 710  FORMAT(12X,F5.2,3(7X,F7.4),10X,F5.2,3(7X,F7.4))
      GO TO 614
 704  NSPACE=1
      IF(35-LINE)719,714,714
 714  WRITE (6,711) WAV(I),RES(I),TRAN(I),RESP(I),WAV(I+35),RES(I+35),
     1TRAN(I+35),RESP(I+35)
 711  FORMAT(1H0,11X,F5.2,3(7X,F7.4),10X,F5.2,3(7X,F7.4))
      GO TO 614
 719  N1=I+LINE-1
      IF(N2-N1)750,720,720
 614  CONTINUE
 750  CONTINUE
1111  FORMAT(F10.7,F10.8)
250   IF(NFLUX.EQ.0) GO TO 307
      WRITE(6,107) (TEMP(I),I=1,L)
107   FORMAT(// 55X,11HTEMPERATURE//(9F13.2))
 809  J=0
6     J=J+1
      IF(NFLUX.EQ.0) GO TO 307
      DO 4 I=1,N,1
4     CALL SPCTRM (TEMP(J),WAV(I),RESP(I),W(I),WX(I),WXX(I),ZINT)
      GO TO 301
 307  DO 302 I=1,N
      WX(I)=RESP(I)
      WXX(I)=WX(I)/PI
 302  CONTINUE
      IF(IRITE.EQ.0) GO TO 300
 301  NSPACE=5
      LINE=45
      DO 216 I=1,N
      NSPACE=NSPACE+1
      LINE=LINE+1
      IF(5-NSPACE)240,242,242
 242  WRITE (6,120) WAV(I),WX(I),WXX(I),W(I)
 120  FORMAT(30X,1PE12.5,4X,1PE12.5,5X,1PE12.5,9X,1PE12.5)
      GO TO 216
 240  NSPACE=1
      IF(45-LINE)244,246,246
 246  WRITE (6,122) WAV(I),WX(I),WXX(I),W(I)
 122  FORMAT(1H0,29X,1PE12.5,4X,1PE12.5,5X,1PE12.5,9X,1PE12.5)
      GO TO 216
 244  LINE=1
      IF(IRITE.EQ.1) GO TO 242
```

43

```
      GO TO 300
216   CONTINUE
300   H=WAV(2)-WAV(1)
      XINT=0.0
      KN=N-2
      DO 5 I=1,KN,2
      XINT=XINT+(H/3.0)*(WX(I)+4.0*WX(I+1)+WX(I+2))
5     CONTINUE
      IF(NFLUX.EQ.1) GO TO 140
      WRITE (6,142) XINT
      BWZERO=XINT
142   FORMAT(///1X,11HBANDWIDTH =1PE12.5,8H MICRONS)
      BC=BC2=0.
      GO TO 144
140   XINT=XINT/PI
      XIRP=XINT
      YINT=0.0
      DO 7 I=1,KN,2
      YINT=YINT+(H/3.0)*(W(I)+4.0*W(I+1)+W(I+2))
7     CONTINUE
      YINT=YINT/PI
      EFX=XINT*BIGRP/YINT
      TEFX=XINT*BIGRP/ZINT
      BC=2.5*ALOG10(1/EFX)
      BC2=2.5*ALOG10(1/TEFX)
144   DO 126 I=1,N
126   WWX(I)=WX(I)*WAV(I)
      XXINT=0.0
      DO 128 I=1,KN,2
      XXINT=XXINT+(H/3.0)*(WWX(I)+4.0*WWX(I+1)+WWX(I+2))
128   CONTINUE
      XXINT=XXINT/PI
      EFFT=XXINT/XINT
      IF(NFLUX.EQ.1) GO TO 146
      EFFT=PI*EFFT
      WRITE (6,130) EFFT
130   FORMAT(//1X,23HEFFECTIVE WAVELENGTH = 1PE12.5)
146   CONTINUE
      IF(NFLUX.EQ.0) WAVDEF=EFFT
      IF(NFLUX.EQ.0) GO TO 8
      ZNON=0.
      CALL SPCTRM (TEMP(J),EFFT,ZNON,BWAV,ZNON,ZNON,ZNON)
      BWAV=BWAV/PI
      IF(J.NE.1) GO TO 4451
      BWREF=BWAV
```

ORIGINAL PAGE IS OF POOR QUALITY

```
      AHREF=XIRP*OMEGA
      PUNCH 222, AHREF
222   FORMAT(5X,*REFERENCE FLUX = *,1PE12.5)
      PUNCH 223,WAVDEF,BWZERO
223   FORMAT(5X,*DEFINED WAVELENGTH = *,F12.8,*  ZERO BANDWIDTH = *,F12.
     18)
      PUNCH 221
221   FORMAT(1X,*NO.      TEMP   WAVLNTH   BDWDTH   BWTHDF   BR MAG  COL
     1MAG  INT.FLUX  VIEW FLUX*)
1451  CONTINUE
      ZNON=0.
      CALL SPCTRM (TEMP(J),WAVDEF,ZNON,BWAVDF,ZNON,ZNON,ZNON)
      BWAVDF=BWAVDF/PI
      IF(BWAV.EQ.0.0) GO TO 4441
      DELWAV=XINT/BWAV
      DLWVDF=XINT/BWAVDF
4441  CONTINUE
      MAGFAC=BWAV/BWREF
      MAGFAC=ALOG10(MAGFAC)
      MAGFAC=-2.5*MAGFAC
      BRMAG=-2.5*ALOG10(XIRP*OMEGA/AHREF)
      AIRP=XIRP*4.5E-06                                               LT 5
      PUNCH 200,ICOL,TEMP(J),EFFT,DELWAV,DLWVDF,MAGFAC,BRMAG,XIRP,AIRP
200   FORMAT(1X,I3,2X,6F9.3,2E10.3)
      IF(J-L) 6,8,8
8     CONTINUE
      IF(NFLUX.EQ.1) GO TO 888
      NFLUX=1
      GO TO 887
888   CONTINUE
      IF(NN-NTRANS)98,97,97
97    IF(NJ-NFUNCT)95,96,96
96    CONTINUE
      STOP
      END
      SUBROUTINE SPCTRM (TEMP,WAV,RESP,W,WX,WXX,WINTG)
      INSERT DESIRED EMISSIVITY HERE AS EMS = XXXX
      EMS=1.00
      PI=3.141592654
      IF(WAV.EQ.0.) GO TO 10
      AI=1.43879/(WAV*TEMP/10000.)
      IF(AI.GT.88.) GO TO 10
      BI=EXP(AI)
      W=(3.741832E-16)/(((WAV/10000.)**5)*(BI-1.))*EMS
      GO TO 12
```

```
0     W=(3.741832E-16)*EXP(-AI)/(((WAV/10000.)**5))*EMS
2     WX=RFSP*W
      WXX=WX/PI
      WINTG=(5.6686E-12)*(TEMP**4)/PI *EMS
      RETURN
      END
```

3.3 A Point-Source Detection Routine

The last routine presented here is a program developed for an early sky survey. The purpose in reviewing it here is to illustrate both the breadth of processing which can be done in a single pass of the data and also the complexity of the required software. The routine unpacks and de-commutates the data and checks for errors and gaps. Three background channels are processed, the running noise computed, and the data plotted. Within the basic detection loop, the data is tested for signal peaks, correlation peaks, and signal length, and the radiation hits are separated from the data. Estimates of the amplitude and bias level are made, and the position of the position of the signal is found from the time of detection.

Inspection of a sample portion of the preliminary detection list reveals some of the basic problems which the following merging routines will need to deal with. The most complex problem is that the correlation coefficient does not track well with some of the other measurements of a good signal. For example, source number 22 has a good correlation coefficient, but the estimated amplitude is less than half the peak height, and the amplitude estimate peak is significantly shifted in time from the data peak and also from the peak correlation coefficient. The correlation coefficient is below a reasonable error gate, but the same is true for signal number 10, where except for a slightly low ρ, the signal is very good. This pattern persisted throughout the data, and a careful study revealed that the poor ρ values were the result of an uncertainty in the detector bias. Since the sensor used bidirectional logarithmic amplifiers, the uncertainty in bias led to a possible error in de-compressing the amplifier functions which would tend to warp the signal shape significantly and degrade the value of the correlation coefficient, and also warping the noise spectrum.

The software presented here was not designed to minimize its use of computer resources which would probably have resulted in a separation of the multiple functions of this routine. Furthermore, using a maximum sensitivity test which allowed a 10% error rate, the resulting data was not significantly compressed. Of course, making multiple measurements of the signal quality and not immediately testing on them. This allowed manual inspection of the data quality and careful adjustment of the tests which followed providing a sound study basis for a larger detection scheme.

```
      PROGRAM HISTR (TAPE1,TAPE2,TAPE3,TAPE4,INPUT,OUTPUT,TAPE5=INPUT,
     1TAPE6=OUTPUT,TAPE7,TAPE8,TAPE9,TAPE10)
      DIMENSION RMS(65),SYSN(65),BKFAC(3)
      DIMENSION BKMN(3),BKSIG(3)
      DIMENSION            BIAS(65,78),BIMAX(65)
      DIMENSION ICOLCNT(3)
      DIMENSION PROGID(3),ICHNM(54),X(39),Y(39)
      DIMENSION KAJ(65),KRJ(65),AMAXJ(65),RMAXJ(65),RSQ(65,78),AMP(65,78
     1),BNEX(65)
      DIMENSION YY(50),DMX(65,39)
      DIMENSION IA(507),IB(2535),MV(65,39),TM(39),CO(21),        XBAR(6
     15),SIGMO(65),KPRIME(65),RMS(65),KSTAR(65),REFLVL(65)           ,KPE
     2AK(65),USNTIM(65),PKTIM(65),YN(65),YNM1(65),YNM2(65)
      DIMENSION AVRVAL(65),ZNO(65)
      DIMENSION XNM1(65),XNM2(65),XN(65)
      DIMENSION IOFST(65)
      REAL  DTAMAX(65),DTA(65,39),USN(65),MEAN,NOISE
      REAL FACTOR(65)
      INTEGER START,FSTREC,REJECT,OBJECT,CHAN(65),PHREC,KRPCNT(65),COUNT
     1(65)
      LOGICAL MARKER(65),BEGIN
      LOGICAL BEGIN2
      EQUIVALENCE (IB,MV)
      COMMON Z(38),F(38),WORK(38),ARG(38),VAL(38),DET(1117)
C
C     INITIALIZE DATA AND DO PRELIMINARY CALCULATIONS
C
      DATA (SYSN(I),I=1,65)/0.,0.,0.,0.,0.,0.,0.,.2180,.2344,.1935,.0730
     1,.1584,.2820,.2232,.2012,.1241,.1531,.2769,.1384,.1159,.2232,.2705
     2,.1335,.0910,.2797,0.,.2662,.2609,.2043,.2395,.1818,.0921,.3359,.1
     3312,.2713,.4505,.4172,.1746,.1530,.2512,.1759,.1952,.2278,.1776,0.
     4,.2553,.3145,.1990,.2260,.2336,.2119,.1582,.2653,.2839,.1735,.2186
     5,.2098,.2071,.1594,.2146,.3406,.3332,.2717,0.,0./
C
C


C
C
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C
      DATA (IOFST(I),I=1,65)/0,0,0,0,0,0,0,514,513,523,539,467,517,516,5
     101,520,525,545,476,487,510,490,502,533,535,0,512,433,482,497,468,4
     282,471,439,475,499,507,494,446,497,438,513,521,498,0,495,533,502,5
     330,513,435,467,489,491,518,470,523,530,478,472,504,513,542,0,0/
C
C
      DATA (CO(I),I=1,21) /+0.7239328E+02,-0.1701253E+03,+0.1507010E+03,
     1-0.5731479E+02,+0.8162575E+01,-0.1163058E+02,+0.1525229E+02,-0.499
     2234?4E+01,+0.8706304E+00,-0.2151428E+01,+0.1230007E+02,-0.2362990E+
     302,+0.3? ?-512E+02,-0.1301119E+02,+0.2655270E+01,-0.2449890E+02,+0.
     4411507?E+03,-0.2140658E+03,+0.1972825E+03,-0.8847811E+02,+0.156208
     58E+02/
```

```
C
      DATA (CHAN(I),I=1,65) / 0,0,0,0,0,0,0,101,102,103,104,105,106,107,
     1108,207,110,111,112,113,114,115,116,117,118,0,201,202,203,204,205,
     2206,209,308,212,210,211,109,213,214,215,216,217,218,0,301,302,303,
     3304,305,306,307,303,208,310,311,312,313,314,315,316,317,318,0,0/
C
      DATA (PROGID(I),I=1,3)/10HPELZMANN 2,10H121      SDG,10HNAL TRACES/
C
      DATA(ICHNM(I),I=1,54)/8,9,10,11,12,13,14,15,35,17,18,19,20,21,22,
     123,24,25,27,28,29,30,31,32,16,34,33,36,37,35,39,40,41,42,43,44,46,
     247,48,49,50,51,52,53,34,55,56,57,58,59,60,61,62,63/
C
      READ (5,44) (YY(I),I=1,40)
   44 FORMAT(3F25.14)
      PRINT 46
      PRINT 45, (I,YY(I),I=1,40)
   45 FORMAT(1X,I10,E25.14)
      PRINT 46
   46 FORMAT (1H1)
C
      IPLOT=0
      SYMHT=0.08
      DX=0.220
      FSTAZ=-28.70
      TJOB=2000.
      TJOB=200.
      SCNTIM=171.0 $ SCNLEN=110.0
      DY=4.0
      TANX=0.
      SUMYY=0.
      SUMY=0.
      DO 21 I=1,40
      YY(I)=YY(I)/100.
      SUMY=SUMY+YY(I)
      SUMYY=SUMYY+YY(I)**2
   21 NY=I
      YDENOM=NY*SUMYY-SUMY*SUMY
      SCNEND=SCNTIM+SCNLEN
      XMAX=SCNLEN/DX
      XMAX=XMAX*1.308
      NSYMS=INT(XMAX/2.)
      OFFSET=0.5
      YMAX=12.0
      ICOLOR=1
      ZFACT=1.0
      ZLONG=3*(XMAX+10.)
      IF(IPLOT.EQ.0) GO TO 7
      CALL PLTID3 (PROGID,ZLONG,YMAX,ZFACT)
      CALL NEWPEN(3)
    7 CONTINUE
      CALL DTAEXP
      PRINT 46
    4 CONTINUE
      DO 1 J=1,65
      SYSN(J)=SYSN(J)*SYSN(J)
      QMS(J)=0.
      AMAXJ(J)=RMAXJ(J)=-5000.
```

ORIGINAL PAGE IS OF POOR QUALITY

```
      DTAMAX(J)=-5000.
      KRJ(J)=KAJ(J)=0
      ONEX(J)=0.
      PRECNT(J)=0
      COUNT(J)=0
      KPRIME(J)=0
      YN(J)=1.
      YNM1(J)=0.
      YNM2(J)=0.
      KPEAK(J)=0
      KSTAR(J)=0
      MARKER(J)=.TRUE.
      XN(J)=XNM1(J)=XNM2(J)=0.
1     XBAR(J)=0.0
      PI=3.1415926535
      SAMPR=350.
      FREQA=1.0
      FREQB=4.0
      ALPHA=2.*PI*FREQA
      BETA=2.*PI*FREQB
      TANA=TAN(ALPHA/(2.*SAMPR))
      TANB=TAN(BETA/(2.*SAMPR))
      OPA=1.+TANA
      OMA=1.-TANA
      OPB=1.+TANB
      OMB=1.-TANB
      DO 9 I=1,3
9     ICOLCNT(I)=0
      BEGIN2=.TRUE.
      ERR=1.00
      NSTAR=0
      NRJCT=0
      DECP=-16.671591*PI/180.
      RAP=6.7337238*PI/12.
      CODECP=COS(DECP)
      SIDECP=SIN(DECP)
      BEGIN=.TRUE.
      START=-1
      REJECT=0
      NU=5
      OBJECT=0
      PHREC=819
      IF(IFLOT.EQ.0) GO TO 43
      CALL AXIS (2.0,0.0,7HAZIMUTH,-7,XMAX,0.0,FSTAZ, DX,10.0)
      CALL AXIS (2.0,0.0,6HZENITH,6,9.0,90.0,91.2,-DX,20.0)
      CALL AXIS (1.0,3.0,12HSIGNAL SCALE,12,5.0,90.0,-2.*DY,DY,20.0)
      CALL PLOT(2.0,0.0,-3)
      XSYM=0.
      DO 11 J=1,NSYMS
      YSYM=0.
      XSYM=XSYM+2.
      DO 11 K=1,18
      YSYM=YSYM+0.5
11    CALL SYMBOL (XSYM,YSYM,SYMHT,3,0.0,-1)
      YLEAST=-0.5*DY
      YMOST=9.5*DY
43    CONTINUE
```

```
          DO 3 J=1,65
          XN(J)=XNM1(J)=XNM2(J)=0.
          YN(J)=0.
          YNM1(J)=0.
          YNM2(J)=0.
    3     CONTINUE
          DO 22 J=1,65
          DO 22 K=1,39
          AMP(J,K)=RSC(J,K)=AMP(J,K+39)=RSC(J,K+39)=0.
          BIAS(J,K)=BIAS(J,K+39)=0.
   22     DTA(J,K)=DNX(J,K)=0.
    C
    C     READ IN A PHYSICAL RECORD, 65 BY 39 ELEMENTS. (12 bits each)
    C
    5     CONTINUE
          CLTIM=SECOND(A)
          IF((TJOB-CLTIM).LE.40.) GO TO 999
          PHREC=PHREC+1
          BUFFER IN (1,1) (IA(1),IA(507))
          CALL RECALL
          IF(UNIT(1)) 10,20,30
   10     CONTINUE
          DO 40 I=1,507
          DO 40 J=1,5
          K=5*(I-1)+J
          IB(K)=MBYTEX(IA(I),J)
   40     CONTINUE
    C
    C     FIND THE AVERAGE BLOCK TIME
    C
          SQ=0.0
          SQT=0.0
          DO 49 K=1,39
          N=MV(1,K)*4096+MV(2,K)
          IF(N.GT.40000000B) N=N-77777777B
          TM(K)=N/1000.
          SQ=SQ+TM(K)
          SQT=SQT+K*TM(K)
   49     CONTINUE
          TA=TANX
          TANX=4.*(SQ-3.*SQT/80.)/39.
    C
    C     CONTINUE THE CALCULATION FOR TIMES WITHIN THE CHOSEN LIMITS
    C
          IF (TA.LT.SCNTIM) GO TO 5
          START=START+1
          IF(START.GT.0) GO TO 41
          FSTREC=PHREC
          FSTTIM=TA
   41     IF(TA.GT.SCNEND) GO TO 999
    C
    C     CORRECT OFFSET, RESCALE, AND LOG EXPAND FOR DATA, AND FIND THE
    C     AVERAGE AND DEVIATION ON WORDS 8 THRU 64.  EXCLUDE THE BACKGROUND
    C     CHANNELS ON WORDS 26, 45, AND 64.
    C
          DO 70 J=8,64
          IF((J.EQ.26).OR.(J.EQ.45).OR.(J.EQ.64)) GO TO 69
```

```
      SUM=0.
      SUMSQ=ONEX(J)
      ONEX(J)=0.
      DO 60 K=1,39
      DTA(J,K)=DNX(J,K)
      ILVL=MV(J,K)-IOFST(J)
      IF(ILVL.LT.-559) ILVL=-559
      IF(ILVL.GT.560) ILVL=560
      IF(ILVL.LT.0) NLVL=561-ILVL
      IF(ILVL.GE.0) NLVL=ILVL+1
      IF(ILVL.LT.0) DNX(J,K)=-DET(NLVL)
      IF(ILVL.GE.0) DNX(J,K)=DET(NLVL)
C
      IF (ICOLOR.NE.1) GO TO 60
C
      IF(NLVL.EQ.0.OR.NLVL.EQ.562) NLVL=NLVL+1
      IF(NLVL.EQ.561.OR.NLVL.EQ.1119) NLVL=NLVL-1
      QTERR=((DET(NLVL+1)-DET(NLVL-1))/2.)*ERR
      ONEX(J)=ONEX(J)+QTERR*QTERR
60    CONTINUE
      IF(ICOLOR.NE.1) GO TO 15
      DO 33 K=1,39
      RSQ(J,K)=RSQ(J,K+39)
      BIAS(J,K)=BIAS(J,K+39)
      AMP(J,K)=AMP(J,K+39)
      SUMX=SUMXY=SUMXX=0.
      DO 31 KK=K,39
      SUMX=SUMX+DTA(J,KK)
      SUMXX=SUMXX+DTA(J,KK)**2
31    SUMXY=SUMXY+DTA(J,KK)*YY(KK-K+1)
      DO 32 KL=1,K
      SUMX=SUMX+DNX(J,KL)
      SUMXX=SUMXX+DNX(J,KL)**2
32    SUMXY=SUMXY+DNX(J,KL)*YY(40-K+KL)
      XDENOM=40.*SUMXX-SUMX*SUMX
      RNUM=40.*SUMXY-SUMX*SUMY
      IF((XDENOM.EQ.0.).OR.(YDENOM.EQ.0.)) GO TO 36
      RSQ(J,K+39)=RNUM*RNUM/(XDENOM*YDENOM)
      GO TO 37
36    RSQ(J,K+39)=0.
37    IF(YDENOM.EQ.0.) GO TO 38
      AMP(J,K+39)=RNUM/YDENOM
      GO TO 39
38    AMP(J,K+39)=0.
39    IF(YDENOM.EQ.0.) GO TO 42
      BIAS(J,K+39)=(SUMYY*SUMX-SUMY*SUMXY)/YDENOM
      GO TO 33
42    BIAS(J,K+39)=0.
33    CONTINUE
15    CONTINUE
      IF(BEGIN2) GO TO 70
C
C     FOLLOWING IS THE DIGITAL FILTER ROUTINE
C
      DO 14 K=1,39
      XN(J)=DTA(J,K)
      YN(J)=0.312*0.312*DTA(J,K)+1.376*YNM1(J)-0.473344*YNM2(J)
```

```
      XNM2(J)=XNM1(J)
      XNM1(J)=YN(J)
      YNM2(J)=YNM1(J)
      YNM1(J)=YN(J)
      DTA(J,K)=YN(J)
      SUM=SUM+DTA(J,K)
14    CONTINUE
      IF(ICCLOR.NE.1) GO TO 70
C
      XBAR(J)=SUM/39.0
      DO 65 K=1,39
      DEV=DTA(J,K)-XBAR(J)
      SUMSQ=SUMSQ+DEV*DEV
65    CONTINUE
      SIGSQ(J)=SUMSQ/39.0
C
C     SET MSN EQUAL TO NOISE IF READING FIRST RECORD.  OTHERWISE USE
C     THE NOISE FORMULA
C     SET THE REFERENCE LEVEL EQUAL TO THE CURRENT AVERAGE IF READING
C     THE FIRST RECORD.  OTHERWISE USE THE REFLVL FORMULA.
C
      IF (BEGIN) REFLVL(J)=XBAR(J)
      IF(BEGIN) MSN(J)=SIGSQ(J)
      IF(((SIGSQ(J)-3.0*MSN(J)).GT.0.).OR.(.NOT.MARKER(J))) GO TO 66
      MSN(J)=0.9*MSN(J)+0.1*SIGSQ(J)
67    REFLVL(J)=0.9*REFLVL(J)+0.1*XBAR(J)
66    CONTINUE
C
C     SEARCH FOR RAPID RISE AND SOURCE SIGNALS IN THE CURRENT WORD OF THE
C     CURRENT BLOCK OF DATA.  DISCARD ALL SIGNALS WHICH MEET THE RRP TEST.
C
      RMS(J)=3.*SQRT(MSN(J))
C
C
      DO 91 K=1,39
      IF (MARKER(J)) GO TO 99
      IF(DTA(J,K).GT.DTAMAX(J)) KPEAK(J)=K
      DTAMAX(J)=AMAX1(DTA(J,K),DTAMAX(J))
      IF((DTA(J,K)-REFLVL(J)-RMS(J)).LT.0.0) KPRIME(J)=K
      GO TO 101
99    IF((DTA(J,K)-REFLVL(J)-RMS(J)).GT.0.0) GO TO 100
      GO TO 101
100   MARKER(J)=.FALSE.
      KSRB=K+28
      KSRE=K+39
      DO 35 I=KSRB,KSRE
      IF(RSQ(J,I).LT.RMAXJ(J)) GO TO 34
      KRJ(J)=I-28
      RMAXJ(J)=RSQ(J,I)
34    CONTINUE
      IF(AMP(J,I).LT.AMAXJ(J)) GO TO 35
      KAJ(J)=I-28
      AMAXJ(J)=AMP(J,I)
      BIMAX(J)=BIAS(J,I)
35    CONTINUE
      AVEVAL(J)=REFLVL(J)
      ZNO(J)=RMS(J)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      KSTAF(J)=K
      DTAMAX(J)=DTA(J,K)
      KPEAK(J)=K
      BGNTIM(J)=TA+(KSTAF(J)+J/70.)/350.
101   CONTINUE
      IF(KPRIME(J).EQ.0) GO TO 90
      IF(KPEAK(J).EQ.KSTAR(J)) GO TO 102
      IF(ZNO(J).EQ.0.) GO TO 105
      IF((1*(DTAMAX(J)-AVRVAL(J))/(ZNO(J)*(KPEAK(J)-KSTAR(J)))).LT.9.)
     1 GO TO 102
105   CONTINUE
      PKTIM(J)=TA+(KPEAK(J)+J/70.)/350.
      RRPCNT(J)=RRPCNT(J)+1
      NRJCT=NRJCT+1
      WRITE(3,2) PHREC,J,KSTAR(J),KPEAK(J),KPRIME(J),DTAMAX(J),BGNTIM(J)
     1,PKTIM(J),AVRVAL(J),ZNO(J),NRJCT,RRPCNT(J),KAJ(J),AMAXJ(J),KRJ(J),
     2RMAXJ(J),BIMAX(J)
2     FORMAT(5I5,5E12.5,2I5,2(I5,E12.5),E12.5)
      GO TO 103
C
C     IF A SOURCE HAS BEEN FOUND WHICH EXCEEDS THE REFERENCE LEVEL BY
C     THREE SIGMA, CALCULATE THE BEGINING TIME AND THE TIME OF PEAK,
C     AND OUTPUT THE STAR DATA TO THE RECORDING FILE.
C
102   PKTIM(J)=TA+(KPEAK(J)+J/70.)/350.
      COUNT(J)=COUNT(J)+1
      NSTAR=NSTAR+1
      WRITE(2,2) PHREC,J,KSTAR(J),KPEAK(J),KPRIME(J),DTAMAX(J),BGNTIM(J)
     1,PKTIM(J),AVRVAL(J),ZNO(J),NSTAR,COUNT(J),KAJ(J),AMAXJ(J),KRJ(J),R
     2MAXJ(J),BIMAX(J)
103   MARKER(J)=.TRUE.
      KAJ(J)=KRJ(J)=0
      RMAXJ(J)=AMAXJ(J)=-5000.
      KSTAR(J)=0
      KPRIME(J)=0
      KPEAK(J)=0
      DTAMAX(J)=-5000.
90    IF((.NOT.MARKER(J)).AND.(K.EQ.39)) KSTAR(J)=KSTAR(J)-39
      IF((.NOT.MARKER(J)).AND.(K.EQ.39)) KPEAK(J)=KPEAK(J)-39
      IF((.NOT.MARKER(J)).AND.(K.EQ.39)) KAJ(J)=KAJ(J)-39
      IF((.NOT.MARKER(J)).AND.(K.EQ.39)) KRJ(J)=KRJ(J)-39
91    CONTINUE
      GO TO 0
69    CONTINUE
C
C     BACKGROUND DATA CHANNELS, IN MV(J,K)
C     J=26,45,64... K=1 TO 39
C
      SUMBK=0.
      SUSQBK=0.
      IBK=(J-7)/19
      DO 16 K=1,39
      POINT=MV(J,K)/.2048
      VAL=POINT*BKFAC(IBK)
      TIME=TA+(K+J/70.)/350.
      IF(J.EQ.26) WRITE(8) VAL, TIME
      IF(J.EQ.45) WRITE(9) VAL,TIME
```

```
      IF(J.EQ.64) WRITE(10) VAL,TIME
      SUMBK=SUMBK+POINT
      SUSQBK=SUSQBK+POINT*POINT
 15   CONTINUE
      BKMN(IBK)=SUMBK*BKFAC(IBK)/39.
      SUSQBK=SUSQBK*BKFAC(IBK)*BKFAC(IBK)/39.
      BKSIG(IBK)=SUSQBK-BKMN(IBK)*BKMN(IBK)
      IF(BKSIG(IBK).LT.0.) BKSIG(IBK)=0.
      BKSIG(IBK)=SQRT(BKSIG(IBK))
 70   CONTINUE
      IF(.NOT.BEGIN2) BEGIN=.FALSE.
      BEGIN2=.FALSE.
      IF(IPLOT.EQ.0) GO TO 2002
      ISTCH=(ICOLCR-1)*18+1
      IENDCH=ISTCH+17
      DO 2001 LL=ISTCH,IENDCH
      M=MOD((LL-1),18)+1
      DO 3001 KK=1,39
      L=ICHNM(LL)
      R=TA+(KK+L/70.)/350.
      X(KK)=-251.3790+1.3082-05*R  -2.771243E-05*R*R    -0.0326111111110
     1+1.87547559780*(-0.138*((CHAN(L)/100)-2)+(0.0353/2.)-((MOD((MOD(CH
     2AN(L),100)),2))*0.0353))-0.105-FSTAZ
      Y(KK)=OFFSET*M*DY+DTA(L,KK)
      IF(Y(KK).LT.YLEAST) Y(KK)=YLEAST
      IF(Y(KK).GT.YMOST) Y(KK)=YMOST
 3001 CONTINUE
      CALL LINE (X,Y,39,1,0,1,0.0, DX,-1,0,DY ,0.05)
 2001 CONTINUE
 2002 CONTINUE
      IF(ICCLOR.NE.1) GO TO 5
      WRITE(4,1001) ((BKMN(K),BKSIG(K)),K=1,3),PHREC
 1001 FORMAT(6(5X,1PE11.4),5X,I5)
      IF((MOD(PHREC,NU)).NE.0) GO TO 5
      DO 50 I=1,65
      DEL=RMS(I)*RMS(I)-SYSN(I)
      IF(DEL.LT.0.) DEL=0.
 50   QMS(I)=SQRT(DEL)
      WRITE(6,1002) PHREC,TA,(CHAN(I),RMS(I),QMS(I),I=8,25),(CHAN(I),RMS
     1(I),QMS(I),I=27,44),(CHAN(I),RMS(I),QMS(I),I=46,63)
      WRITE(7,1002) PHREC,TA,(CHAN(I),RMS(I),QMS(I),I=8,25),(CHAN(I),RMS
     1(I),QMS(I),I=27,44),(CHAN(I),RMS(I),QMS(I),I=46,63)
 1002  FORMAT(1H0,4X,I4,4X,F8.2,18(3(*   ]   *,I3,-X,E10.4,5X,E10.4),/,19X
     1))
      GO TO 5
 30   WRITE(6,1003) PHREC
 1003 FORMAT(1H0,5X,*PARITY ERROR ON RECORD *,I10,*  CONTINUE READING*,/
     1)
      GO TO 5
 999  ILSTRC=PHREC
      WRITE(6,1004) PHREC,TA
 1004 FORMAT(1H0,5X,*TIME MAXIMUM REACHED AT RECORD *,I10,5X,*  FINAL TI
     1ME IS *,E10.4,///)
 20   IF(ICCLOR.NE.1) GO TO 18
      ENDFILE 4
      REWIND 4
      PRINT -6.
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
80    READ(4,1001) ((BKMN(K),BKSIG(K)),K=1,3),IBKREC
      IF(EOF(4)) 82,81
81    WRITE(7,1001) ((BKMN(K),BKSIG(K)),K=1,3),IBKREC
      GO TO 80
82    CONTINUE
      ENDFILE 7
      REWIND 7
83    READ (7,1012) PHREC,TA,(CHAN(I),RMS(I),OMS(I),I=8,25),(CHAN(I),RMS
     1(I),OMS(I),I=27,44),(CHAN(I),RMS(I),OMS(I),I=46,63)
      IF(EOF(7)) 85,84
84    WRITE(4,1012) PHREC,TA,(CHAN(I),RMS(I),OMS(I),I=8,25),(CHAN(I),RMS
     1(I),OMS(I),I=27,44),(CHAN(I),RMS(I),OMS(I),I=46,63)
      GO TO 83
85    ENDFILE 4
C
C     NOW COUNT UP THE OBJECTS DETECTED, AND THE REJECTS, AND LIST TABLES
C
      DO 200 I=1,65
      J=CHAN(I)/100
      IF(J.NE.0) ICOLCNT(J)=ICOLCNT(J)+COUNT(I)
      REJECT=REJECT+RRPCNT(I)
200   OBJECT=OBJECT+COUNT(I)
      ISPACE=ILSTRC-FSTREC+5
18    IF(IPLOT.NE.0) GO TO 17
      WRITE(6,1010)
1010  FORMAT(1H0,5X,*END OF INPUT FILE*,5X,*NO PLOTS*)
      GO TO 19
17    WRITE(6,1009) ICOLOR,ICOLCNT(ICOLOR)
1009  FORMAT(1H0,5X,*END OF INPUT FILE*,5X,*PLOT COLOR NUMBER*,I3,5X,I5,
     1* STARS*)
19    CONTINUE
      IF(ICCLOP.NE.1) GO TO 402
      WRITE(6,1005) FSTREC,PHREC
1005  FORMAT(1H1,5X,*THE FIRST RECORD IS *,I10,5X,*THE LAST RECORD IS *,
     1I10)
      WRITE(6,1011) FSTTIM,TA
1011  FORMAT(1H ,5X,*THE DATA BEGINS AT T = *,F10.4,* SECONDS AND ENDS
     1AT T = *,F10.4,* SECONDS*)
      WRITE(6,1015) OBJECT,REJECT
1015  FORMAT(1H0,5X,*SOURCES FOUND \ *,I10,5X,*SIGNALS REJECTED \ *,I10)
      WRITE(6,1014)
1014  FORMAT(1H0,20X,*CHANNEL NOMENCLATURE\ *,/,43X,*100IS ARE SHORT WAV
     1ELENGTH*,/,43X,*200IS ARE MEDIUM WAVELENGTH*,/,43X,*300IS ARE LONG
     2 WAVELENGTH*,/)
      WRITE(6,1007)
1007  FORMAT(1H0,5X,*OBJECTS DETECTED ON EACH CHANNEL*)
      WRITE(6,1006) (CHAN(I),COUNT(I),I=8,25),(CHAN(I),COUNT(I),I=27,44)
     1,(CHAN(I),COUNT(I),I=46,63)
1006  FORMAT(9(4X,I3,*---*,I4))
      WRITE(6,1008)
1008  FORMAT(1H0,5X,*SIGNALS REJECTED ON EACH CHANNEL*)
      WRITE(6,1006) (CHAN(I),RRPCNT(I),I=8,25),(CHAN(I),RRPCNT(I),I=27,4
     14),(CHAN(I),RRPCNT(I),I=46,63)
      WRITE(6,1000) (ICOLCNT(I),I=1,3)
1000  FORMAT(1H0,5X,*SOURCES ON EACH COLOR*,/,27X,*SHORT----*,I5,/,27X,*
     1MEDIUM---*,I5,/,27X,*LONG-----*,I5)
C
```

```
C     REWIND THE DETECTION FILE AND LIST THE SOURCES FOUND
C
      ENDFILE 2
      ENDFILE 3
      REWIND 2
      REWIND 3
      NT=2
      PRINT 46
      WRITE(6,1016)
1016  FORMAT(1H0,58X,*STARS DETECTED*,//)
400   WRITE(6,1021)
1021  FORMAT(1X,*NUMBER CHAN  ST  PK   E   A   R  LNTH  TIME     AMPL   H
     1EIGHT  SHIFT  MEAN  COR COF   SNR   NOISE  BRIGHT  ZEN  AZIMUTH RA
     2(HR)   DECL*,/)
300   READ(NT,2) IREC,IWD,ISAMP,IPEAK,IEND,PEAK,BGN,PK,MEAN,NOISE,ISTTL,
     1ISTCH,KACH,ACH,KRCH,RCH,SHIFT
      IF(EOF(NT)) 302,301
301   HT=PEAK-MEAN
      RCH=SQRT(RCH)
      IF(NOISE.EQ.0.) GO TO 12
      SNR=3.*HT/NOISE
      GO TO 13
12    SNR=0.
13    CONTINUE
      BR=HT*FACTOR(IWD)
      ZRS=(IPEAK-ISAMP)*1.30814/350.
      ZE=91.5039700924-0.1395353564-76*((MOD(CHAN(IWD),100))-1)-0.233
      ZERAD=ZE*PI/180.
      TI=PK
      PK =-251.3790+1.3082405*PK -2.7712-3E-05*PK*PK  -0.0326111111113+1
     1.0.5547559755*(-0.138*((CHAN(IWD)/100)-2)+(0.035372.)-((MOD((MOD(CH
     2AN(IWD),100)),2))*0.0353))-0.195
      AZRAD=PK2*PI/180.
      PK2=PK
      IF (PK.LT.0.) PK2=PK2+360.
      PK=PK+360.
      COSZE=COS(ZERAD)
      SINZE=SIN(ZERAD)
      COSAZ=COS(AZRAD)
      SINAZ=SIN(AZRAD)
      SINDEC=COSZE*SIDECP+SINZE*CODECP*COSAZ
      DEC=ASIN(SINDEC)
      COSDEC=SQRT(1.-SINDEC*SINDEC)
      SINHA=-SINAZ*SINZE/COSDEC
      COSHA=(COSZE-SIDECP*SINDEC)/(CODECP*COSDEC)
      HA=ATAN2(SINHA,COSHA)
      RA=RAP-HA
      DECDEG=DEC*180./PI
      RAHMS=RA*12./PI
      WRITE(6,1022) ISTTL,ISTCH,CHAN(IWD),ISAMP,IPEAK,IEND,KACH,KRCH,ZRS
     1,TI,ACH,HT,SHIFT,MEAN,RCH,SNR,NOISE,BR,ZE,PK,RAHMS,DECDEG
1022  FORMAT(1X,I4,7(1X,I3),1X,F5.3,1X,F6.2,2(1X,F7.2),2(1X,F6.2),1X,F7.
     14,1X,F7.2,1X,F6.3,1X,F7.2,1X,F5.2,1X,F7.3,1X,F6.3,1X,F7.3)
      WRITE(-,2022) ISTTL,ISTCH,CHAN(IWD),ISAMP,IPEAK,IEND,KACH,KRCH,ZRS
     1,TI,ACH,HT,SHIFT,MEAN,RCH,SNR,NOISE,BR,ZE,PK,RAHMS,DECDEG
2022  FORMAT(I4,7I3,2F8.4,2F6.3,5F8.4,F8.3,4F8.4)
      GO TO 300
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      92  ENDFILE 4
          IF(NT.EQ.3) GO TO 402
          PRINT 45
          WRITE(6,1017)
    1017  FORMAT(1H0,/3X,*STARS REJECTED*,//)
          NT=3
          GO TO 402
     402  CONTINUE
          ENDFILE 4
          REWIND 4
          IF(IPLOT.EQ.0) GO TO 6
          ICOLOR=ICOLOR+1
          IF(ICOLOR.GT.3) GO TO 6
          XNEW=XMAX+10.
          DO 8 I=1,ISPACE
       8  BACKSPACE 1
          CALL PLOT (XNEW,0.0,-3)
          IF(ICOLOR.EQ.2) DY=DY*2.0
          GO TO
       6  CONTINUE
          ENDFILE 6
          IF(IPLOT.EQ.0) GO TO 777
          CALL ENDPLT
     777  CONTINUE
          STOP
          END
```

```
EVERITY  DETAILS    DIAGNOSIS OF PROBLEM

  I     +600 360  TOTAL RECORD LENGTH IS GREATER THAN 137 CHARACTERS.  IT MAY EXCE
  I     VAL       ARRAY NAME OPERAND NOT SUBSCRIPTED, FIRST ELEMENT WILL BE USED.
```

```
MBOLIC REFERENCE MAP (R=3)

NTS      DEF LINE    REFERENCES
HISTR        1

     SN  TYPE             RELOCATION
       * REAL                               REFS     187
H        REAL                               REFS     563      567      DEFINED
PHA      REAL                               REFS     135      DEFINED  133
AXJ      REAL             ARRAY             REFS       8      336      357
                                             375
P        REAL             ARRAY             REFS       8      253      336
                                             270      272
G        REAL             ARRAY             REFS      24
RVAL     REAL             ARRAY             REFS      14      351      357
RAD      REAL                               REFS     552      553      DEFINED
GIN      LOGICAL                            REFS      21      307      308      DEF
GIN2     LOGICAL                            REFS      22      279      410      DEF
```

FOLDOUT FRAME

NUMBER	[illegible]	S	PK	E	A
1	1 [illegible]	1	[illegible]	[illegible]	1
2	1 111	20	[illegible]	[illegible]	[illegible]
3	1 [illegible]	[illegible]	[illegible]	[illegible]	[illegible]
4	1 203	0	10	27	6
5	1 209	[illegible]	[illegible]	18	[illegible]
6	1 [illegible]	4	[illegible]	[illegible]	[illegible]
7	1 [illegible]	[illegible]	2	[illegible]	[illegible]
8	1 [illegible]	[illegible]	[illegible]	33	11
9	1 [illegible]	1	1	[illegible]	1
10	1 [illegible]	-33	[illegible]	1	[illegible]
11	1 [illegible]	[illegible]	[illegible]	[illegible]	[illegible]
12	1 304	-33	-13	6	-22
13	1 [illegible]	[illegible]	[illegible]	[illegible]	[illegible]
14	1 307	-39	-21	1	-25
15	2 [illegible]	[illegible]	[illegible]	[illegible]	[illegible]
16	1 118	11	13	17	11
17	2 [illegible]	[illegible]	[illegible]	[illegible]	[illegible]
18	1 115	24	27	30	[illegible]
19	1 [illegible]	32	[illegible]	[illegible]	32
20	1 113	19	19	21	15
21	2 [illegible]	[illegible]	[illegible]	[illegible]	[illegible]
22	1 318	19	21	25	19
23	2 [illegible]	12	[illegible]	[illegible]	12
24	1 116	20	[illegible]	31	21
25	2 118	20	[illegible]	[illegible]	[illegible]
26	1 102	-3	[illegible]	1	-3
27	1 [illegible]	[illegible]	[illegible]	22	[illegible]
28	3 115	-9	[illegible]	1	-8
29	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
30	2 218	14	22	33	19
31	2 [illegible]	2	[illegible]	[illegible]	[illegible]
32	3 111	27	32	38	27
33	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
34	2 104	18	20	22	18
35	2 [illegible]	15	[illegible]	18	[illegible]
36	2 102	21	24	28	21
37	1 103	22	[illegible]	35	[illegible]
38	1 108	10	12	15	10
39	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
40	2 116	19	21	24	19
41	2 [illegible]	[illegible]	[illegible]	[illegible]	[illegible]
42	1 106	9	10	11	9
43	2 202	9	[illegible]	[illegible]	[illegible]
44	2 203	13	[illegible]	31	[illegible]
45	2 304	[illegible]	[illegible]	22	[illegible]
46	1 101	6	13	15	6
47	2 113	1		[illegible]	[illegible]
48	1 206	-4	-1	3	-4
49	3 [illegible]		[illegible]	[illegible]	[illegible]
50	2 103	[illegible]	[illegible]	[illegible]	[illegible]
51	3 [illegible]	[illegible]	[illegible]	34	[illegible]
52	4 114	[illegible]	[illegible]	35	[illegible]
53	1 213	[illegible]	[illegible]	2	-2
54	1 110	8	11	15	12
55	3 203	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

ORIGINAL PAGE IS OF POOR QUALITY

NUMBER	CHAN		ST[illegible]K		E	A
1	1	[illegible]	[illegible]	[illegible]	29	[illegible]
2	1	[illegible]	[illegible]	[illegible]	1	[illegible]
3	1	21[illegible]	[illegible]		14	[illegible]
4	1	111	11	[illegible]	[illegible]	11
5	1	205	[illegible]	[illegible]	33	[illegible]
6	1	214	-7	[illegible]	9	[illegible]
7	1	[illegible]		[illegible]	19	[illegible]
8	1	1[illegible]	[illegible]	[illegible]	28	13
9	1	[illegible]	[illegible]	-7	1	[illegible]
10	2	112	-13	[illegible]	3	-11
11	1	115	[illegible]	-1	[illegible]	-3
12	2	107	[illegible]	[illegible]	37	27
13	1	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
14	1	101	11	[illegible]	22	12
15	1	[illegible]	11	[illegible]	22	[illegible]
16	1	205	18	22	36	21
17	2	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
18	1	215	-7	[illegible]	4	[illegible]
19	2	[illegible]	[illegible]	[illegible]	30	[illegible]

4.0 APPENDIX: A SIGNAL PROCESSING GAME

The aim of this game is to develop skills in signal processing. The input data for this game are the recorded data $U_R(t)$. It is assumed that the non-uniform scanning velocity has been corrected for already. The time coordinate is given in discrete numbers t = 0, 1, 2...63. We may consider $U_T(t)$ as being about one quarter of a single horizontal scan (β = constants).

The rules of the games are as follows. The "investigator" gets the sheet "Recorded Data $U_R(t)$" and the sheet "Problem #1." After solving this problem he will give the solution to the "monitor" and to the "game constructor." Now he may start on problem #2, and so on. But it is <u>important</u> that the investigator does not get the next problem sheet before he has finished the previous problem. The reason is that the formulation of the later problems contains parts of the answers to the earlier problems. This has to do with the basic structure of this simulation game: for performing any meaningful signal processing operation one must have some knowledge about the original signal and/or the noise. For example, in problem #1 the investigator is told that the noise <u>is</u> additive and non-negative. In the later problems, the investigator will be supplied with even more <u>a priori</u> information. Naturally, this should enable him to extract the signals better and better. But the methods for doing this increase in complexity.

On the very last pages, following the problems, the design of the "recorded data" is explained, and the true original signal is unveiled. Obviously those pages should not be given to the investigator before he has solved all the problems.

RECORDED DATA $U_R(t)$

t is the discrete time variable running from t=0 to t=63

t	$U_R(t)$	t	$U_R(t)$	t	$U_R(t)$	t	$U_R(t)$
0	55	16	110	32	39	48	383
1	25	17	184	33	56	49	10
2	85	18	29	34	05	50	69
3	61	19	51	35	15	51	58
4	20	20	42	36	95	52	52
5	95	21	78	37	09	53	66
6	07	22	09	38	81	54	79
7	00	23	15	39	21	55	134
8	62	24	13	40	81	56	94
9	79	25	50	41	399	57	102
10	148	26	99	42	312	58	108
11	105	27	54	43	348	59	94
12	125	28	99	44	303	60	56
13	125	29	35	45	383	61	67
14	173	30	98	46	317	62	51
15	181	31	02	47	317	63	63

Problem #1

Given are the recorded data $U_R(t)$ with $t = 0, 1 \ldots 63$. Wanted are the original data $U_0(t)$, which represent the "one-dimensional equivalent object radiation" $S_E(\alpha)$. We assume that the known influences of the telescope $[M(x',y');\ R(x',\ y')]$ and of the electrical system $[G(t)]$ have been compensated already or are negligible. But the recorded signal $U_R(t)$ is corrupted by additive noise $N(t)$:

$$U_R(t) = U_0(t) + N(t)$$

The only features known about the original signal $U_0(t)$ and about the noise $N(t)$ are that they are non-negative:

$$U_0(t) \geq 0; \qquad\qquad N(t) \geq 0.$$

Furthermore, the noise $N(t)$ is stationary, which means that the noise properties are not "drifting." In other words, short-term average features of the noise remain the same from the beginning to the end of the observation.

Try to utilize the given a priori information for computing a new signal $U_1(t)$ from $U_R(t)$, which somehow is better than $U_R(t)$ as an approximated representation of $U_0(t)$. Plot $U_1(t)$ as a continuous curve, and also $U_R(t)$ for comparison.

Problem #2

Given are the facts:

$$U_0(t) \geq 0; \qquad N(t) \geq 0; \qquad \overline{N} = 50.$$

By $\overline{N}$ we mean the linear average of the noise. This $\overline{N}$ can be visualized as the dark current of the photoreceiver as measured with an instrument which rejects high frequencies.

Based on these facts, try to compute a better signal $U_2(t)$ from $U_R(t)$. Plot both $U_2(t)$ and $U_R(t)$.

Problem #3

Given are the same facts as in the previous problem. In addition, it is known that the noise is approximately "white."

$$N(t) = \bar{N} + n(t); \qquad \tilde{n}(\nu) = \int n(t)\, e^{-2\pi i \nu t}\, d\nu;$$

$$\nu = m/64; \quad m = -32, -31, \ldots -1, 0, +1, \ldots +30, +31;$$

$|\tilde{n}(\nu)|^2$ constant. The amount of this "constant" is not known. Try to deduce it from the recorded data $U_R(t)$. You might have to make an intelligent guess.

Problem #4

Given are the same facts as in the previous problems, including the "constant" which describes the noise power level.

$$|\tilde{n}(\nu)|^2 \approx \frac{16}{3} * 10^4 \text{ in } -\frac{1}{2} \leq \nu \leq +\frac{1}{2}.$$

Now that $|\tilde{n}(\nu)|^2$ is known and $U_R(\ )$ is computable, can you apply the Wiener-filter theory, at least in a guessed approximation? Try it and compute $U_4(t)$. Plot $U_4(t)$ and $U_R(t)$. Hing: represent $|\tilde{U}_0(\nu)|^2$ by a gaussian function of suitable peak power and width. Signal processing specialists always try it with a gaussian function if they don't know a better way.

$$|\tilde{U}_o(\nu)|^2 \approx P_4 \, e^{-\pi(\nu/\nu_4)^2}.$$

Problem #5

Try the same approach as in the previous problem, but with a guessed sinc^2-shaped $|U_0(\nu)|^2$

$$|U_0(\nu)|^2 \approx P_5 \ \mathrm{sinc}^2(\nu/\nu_5); \qquad \mathrm{sinc}\ z = \frac{\sin \pi z}{\pi z}$$

Plot the result $U_5(t)$ and also $U_R(t)$ for comparison.

Problem #6

Try the same approach as in the previous problem, but a somewhat different guess for $|\tilde{U}_0(\nu)|^2$

$$|\tilde{U}_0(\nu)|^2 = P_6 \operatorname{sinc}^2(\nu/\nu_6) + (P_0 - P_6)\, \delta_0;$$

Herein δ_0 means a function which is equal to 1 for $\nu = 0$ and equal to 0 for $\nu \neq 0$. Plot $U_6(t)$ and $U_R(t)$.

Problem #7

Based on all of the accumulated experience, try your own signal processing approach or simply guess what $U_0(t)$ might have been. Call it $U_7(t)$. Plot $U_7(t)$ and $U_R(t)$.