RESEARCH TRIANGLE INSTITUTE

Final Report

SIMULATION STUDIES FOR NOAA STABILIZED COMPENSATION PROGRAM

Prepared under Contract No. 2-35369

by

W. Howard Ruedger

Research Triangle Institute
Research Triangle Park
North Carolina 27709

for

Data Processing and Analysis Division
National Environmental Satellite Service
National Oceanic and Atmospheric Administration
Washington, D. C.

August 1973

RESEARCH TRIANGLE PARK, NORTH CAROLINA 27709

U.S. DEPARTMENT OF COMMERCE
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
NOAA

74 1106

FOREWORD

This report was prepared by the Research Triangle Institute, Research Triangle Park, North Carolina, under Contract 2-35369. The work has been administered by the Data Processing and Analysis Division, National Environmental Satellite Service, National Oceanic and Atmospheric Administration. Dr. John A. Leese is the NESS Technical Representative for the contract.

The program studies began on 21 April, 1972, and were completed on 20 May 1973. Participating RTI staff members were as follows:

W. H. Ruedger, Project Leader

G. S. Brown, Member of the Technical Staff

J. H. White, Member of the Technical Staff

ABSTRACT

This report describes the participation of the Research Triangle Institute in the NOAA/NESS program to achieve data quality assurance. Long-term observation of the quality of the data from the scanning radiometer systems aboard the ITOS type spacecraft has indicated that a requirement exists for a systematic means of maintaining data quality. Previous studies have further indicated that the alignment of the equipment used to compensate for spacecraft tape recorder flutter effects was particularly sensitive in affecting data quality. This equipment was selected as the subject of a pilot program to assure data quality by stabilized control. RTI has contributed to the overall objective of this pilot program by conducting analyses of the z-axis* compensation technique as it interacts with the data and by recommending diagnostic logic to survey data quality as the data are ingested from the Command and Data Acquisition Station (CDA).

---

*NOAA nomenclature referring to the flutter components introduced at the spacecraft tape recorder.

TABLE OF CONTENTS

ILLUSTRATIONS

TABLES

1.0 INTRODUCTION

This report describes the participation of the Research Triangle Institute in the NOAA/NESS program to achieve data quality assurance. Long-term observation of the quality of the data from the scanning radiometer systems aboard the ITOS type spacecraft has indicated that a requirement exists for a systematic means of maintaining data quality. Previous studies have further indicated that the alignment of the equipment used to compensate for spacecraft tape recorder flutter effects was particularly sensitive in affecting data quality. This equipment was selected as the subject of a pilot program to assure data quality by stabilized control. RTI has contributed to the overall objective of this pilot program by conducting analyses of the z-axis compensation technique as it interacts with the data and by recommending diagnostic logic to survey data quality as the data are ingested from the Command and Data Acquisition Station.

Figure 1-1 shows the overall flow of ITOS scanning radiometer data. During real time operation, the data are recorded on the spacecraft tape recorder for subsequent readout. During acquisition the tape recorder is played back at an increased speed, the data is frequency division multiplexed, and then transmitted to the ground station via an S-Band telemetry link. At the ground station the data is received, demultiplexed and recorded. During recorder playback, the data is z-axis corrected and fed to the long lines for transmittal to DAPAD where it is digitized and input to the data processing equipment.

Figure 1-2 indicates the overall philosophy of the program to stabilize data quality. The concept consists of monitoring specific parameters in the data as they are ingested into the Digital Data Handling System (DDHS), conducting analyses in real-time, evaluating these analyses and communicating informative diagnostics to the CDA if data quality is not being maintained. In the event a fault does exist at the CDA, it is anticipated that these diagnostics will be of assistance in localizing and correcting this condition. This program represents an on-going effort at NESS. As a result, this report documents only those areas in which RTI was directly involved. This involvement consisted of an investigation of the compensation technique, the development of a digital model of the hardware, and the use of this model to derive sensitivities to be used as guidelines in the development of the required diagnostic logic.

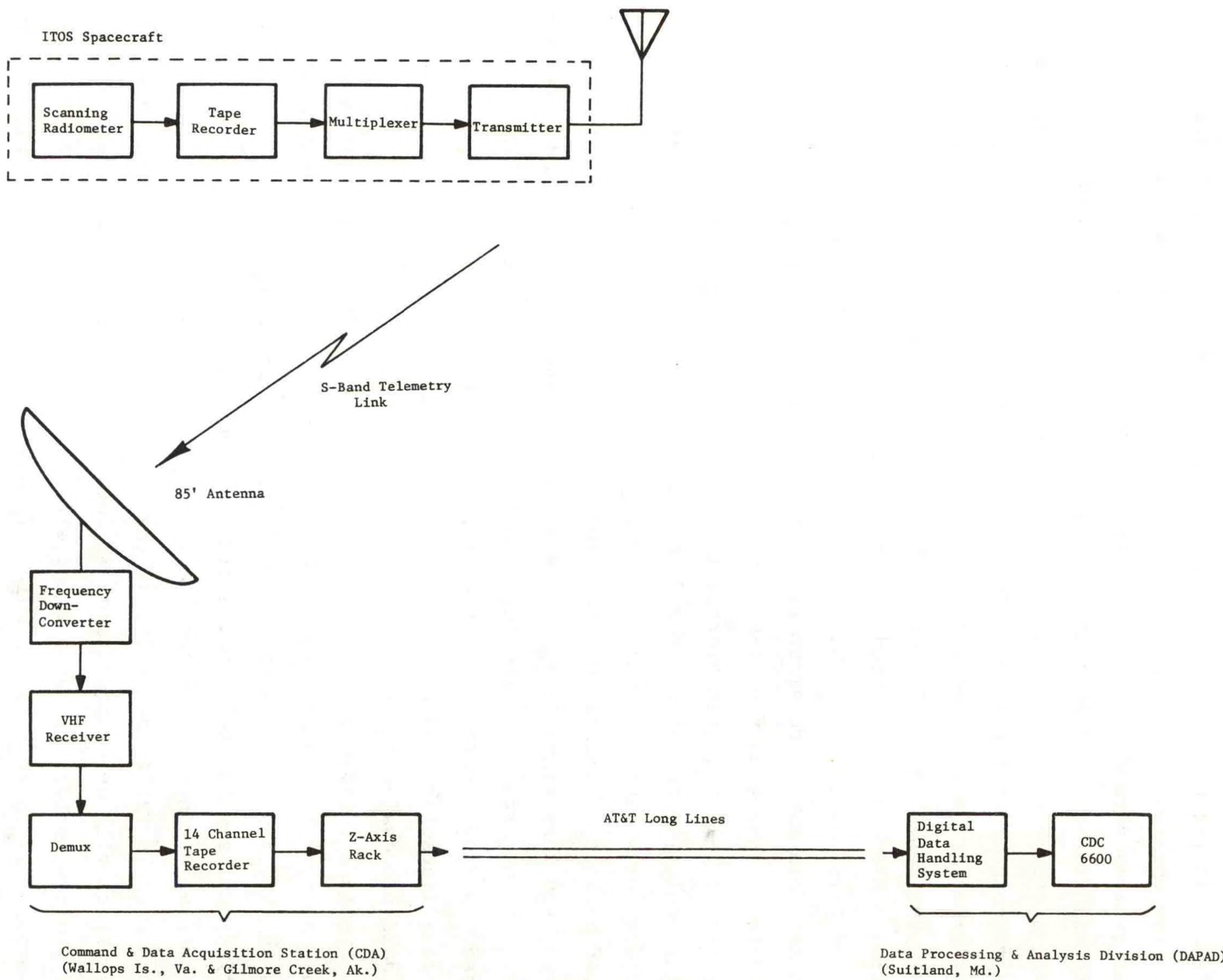


Figure 1-1. ITOS Scanning Radiometer Data Flow.

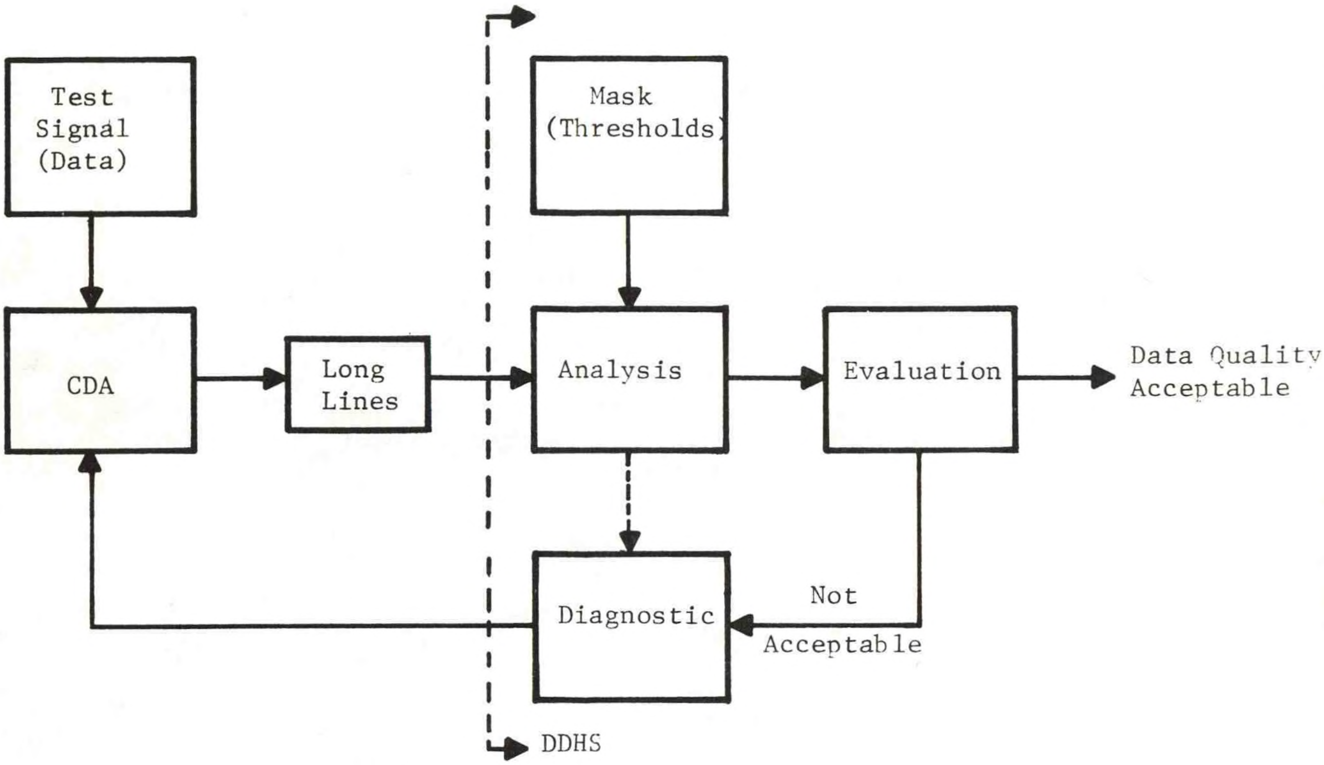


Figure 1-2. Overall Data Quality Assurance Philosophy.

The report has three main features. The initial sections present a tutorial discussion of frequency modulation theory to serve as background and also a tutorial discussion of the theory of operation of the z-axis compensation technique. Subsequent sections describe the model, pertinent analyses conducted with the model, and preliminary diagnostic logic recommendations based on these analyses. The computer algorithm is then documented in detail in the appendices. It is hoped that this approach will result in the report being a working document for NESS and RTI personnel as well as documenting the effort expended during the study.

2.0 FREQUENCY MODULATION CONSIDERATIONS

This section presents a tutorial introduction to the mathematical aspects of frequency modulation. The intent here is to indicate the effects of the modulation technique being non-linear. Of prime concern is the spectral representation of an FM modulated waveform, especially the facts that spectral superposition does not apply and that tone modulation has a distributed spectral representation which is tone amplitude dependent.

2.1 Theoretical Considerations

The "frequency" of a waveform is defined as the time-rate-of-change of phase and as such usually suggests a periodic phenomenon (i.e. in the spectral sense). On the other hand "frequency modulation" implies a time varying frequency which is contrary to the usual notion. It therefore becomes important in the context of frequency modulation to distinguish between "spectral" and "instantaneous" frequencies.

In general, a frequency modulated wave $x_c(t)$ can be written as

$$x_c(t) = A_c \cos\Theta_c(t) \tag{2-1}$$

where A_c is constant and $\Theta_c(t)$ is a function of the carrier frequency and some baseband signal $x(t)$. In order to distinguish between the two types of frequency mentioned in the preceding paragraph, $\Theta_c(t)$ can be written as

$$\Theta_c(t) = 2\pi f_c t + \phi(t)$$

where f_c is the carrier (spectral) frequency and $\phi(t)$ contains the baseband signal $x(t)$. Returning to the concept that frequency is the time-rate-of-change of phase, one can define "instantaneous" frequency as

$$\xi(t) = \frac{1}{2\pi}\frac{d}{dt}\Theta_c(t) = f_c + \frac{1}{2\pi}\frac{d\phi(t)}{dt} \quad .$$

If the baseband signal is now taken to be the change in instantaneous frequency as a function of time,

$$\xi(t) = f_c + f_D\, x(t)$$

(where f_D is called the "deviation" constant and is a system parameter), one can identify

$$\frac{d\phi(t)}{dt} = 2\pi\, f_D\, x(t)$$

or

$$\phi(t) = 2\pi f_D \int_0^t x(\tau)\, d\tau \quad .$$

Now the modulated wave given in eq. 2-1 can be written as

$$x_c(t) = A_c \cos[\omega_c t + 2\pi f_D \int_0^t x(\tau)\, d\tau] \qquad (2\text{-}2)$$

indicating the relation between a baseband signal and the modulated carrier.

The point of this discussion is to demonstrate that an intuitive notion of the relationship of the spectral distribution of signal information at baseband and its distribution after modulation is not available. This will be pursued in greater detail in the following section which deals with the spectrum of a frequency modulated wave.

2.2 Spectral Analysis of Modulated Carrier

The spectral analysis of a frequency modulated carrier is pertinent to the study in that data processing equipment bandwidths are continually of interest. For this reason a discussion of the behavior of the spectral distribution of the modulated carrier is included. The approach will be to initially examine the behavior of a carrier modulated by a single tone as a function of tone amplitude and frequency and to then extend this to a two tone situation. Multiple tone situations will not be discussed as they are simply mathematically complicated extensions of the two tone case and offer no additional insight.

Consider the baseband tone,

$$x(t) = A_m \cos\omega_m t \quad .$$

Equation 2-2, the modulated carrier takes on the form,

$$x_c(t) = A_c\cos(\omega_c t + 2\pi f_D \int_0^t A_m \cos\omega_m \tau d\tau)$$

$$= A_c\cos(\omega_c t + \frac{f_D A_m}{f_m} \sin\omega_m t) \quad .$$

Define,

$$\beta = \frac{f_D A_m}{f_m}$$

and obtain

$$x_c(t) = A_c\cos(\omega_c t + \beta\sin\omega_m t) \quad . \tag{2-3}$$

The parameter β is termed the modulation index and represents the maximum phase deviation produced by the tone. Rewriting equation 2-3 as

$$x_c(t) = A_c[\cos\omega_c t \cos(\beta\sin\omega_m t) - \sin\omega_c t \sin(\beta\sin\omega_m t)]$$

and writing $\cos(\beta\sin\omega_m t)$ and $\sin(\beta\sin\omega_m t)$ in terms of Bessel functions of the first kind, one can achieve a Fourier series type representation for $x_c(t)$. In particular,

$$x_c(t) = A_c J_o(\beta) \cos\omega_c t$$

$$+ \sum_{n \text{ odd}}^{\infty} A_c J_n(\beta)[\cos(\omega_c + n\omega_m)t - \cos(\omega_c - n\omega_m)t]$$

$$+ \sum_{n \text{ even}}^{\infty} A_c J_n(\beta)[\cos(\omega_c + n\omega_m)t + \cos(\omega_c - n\omega_m)t] \quad .$$

Notice that the spectrum of the modulated carrier consists of an infinite number of lines spaced at intervals of ω_m about the carrier with amplitudes that vary as Bessel functions of the modulation index. An illustrative line spectrum for a frequency modulated tone is shown as Figure 2-1.

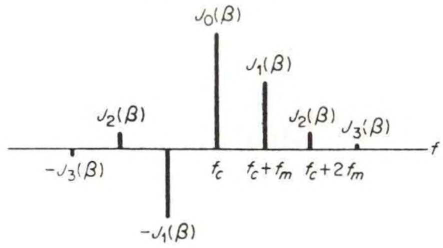


Figure 2-1. An FM line spectrum, single tone modulation.

Further, notice that for a given tone frequency, the modulation index β changes as the amplitude of the tone varies which produces a different spectral distribution. Figure 2-2 shows the behavior of Bessel functions versus their argument.

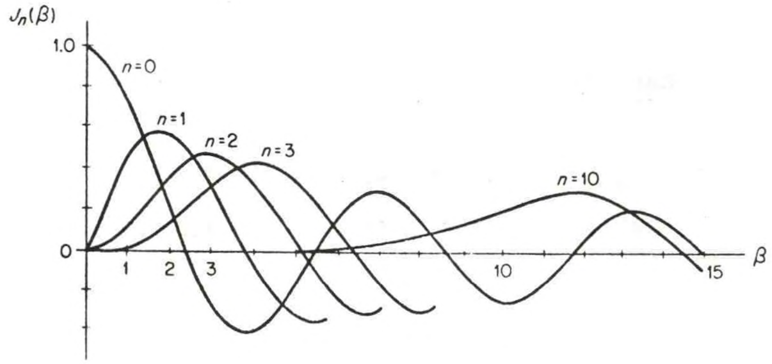


Figure 2-2. Bessel functions of fixed order plotted versus the argument β.

One may view the variation in spectral distribution as a function of amplitude and frequency by noting that the harmonics generated by β=2 are quite different from harmonics generated by β=4. This differential in β could be simply the result of a two-to-one increase in tone amplitude. Notice that for β=12 (which incidentially represents an unlikely situation), the maximum spectral line occurs at n=10 which represents a frequency ten times the tone frequency and would require an extremely wide channel bandwidth. Figure 2-3 indicates the effect of changing tone amplitude and frequency. The effect of amplitude only and frequency only changes are indicated (recall that $\beta \sim \frac{A_m}{f_m}$).

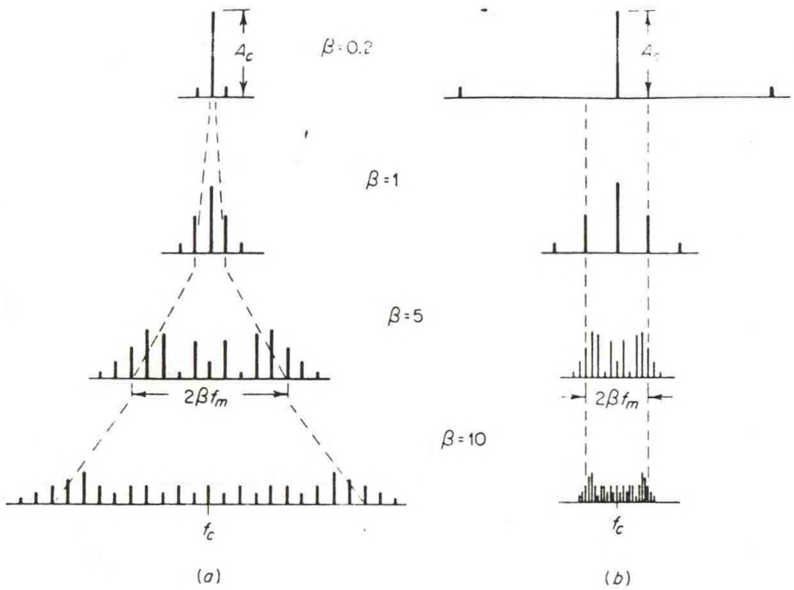


Figure 2-3. Tone-modulated FM line spectra showing the effects of tone amplitude and frequency. (a) frequency fixed, increasing amplitude; (b) amplitude fixed, decreasing frequency.

The preceding discussion has treated the behavior of the spectrum for only a single tone. It is of interest to consider the additional influence of a multiple tone signal as this represents a more realistic situation. The mathematics of investigating the general multitone situation is extremely cumbersome and, as a result, only the two tone case will be discussed here. This is adequate as all the pertinent effects will be revealed.

Following the mathematical procedure introduced for a single tone and omitting the intermediate details, for a signal of the form

$$x(t) = A_1 \cos\omega_1 t + A_2 \cos\omega_2 t$$

where f_1 and f_2 are not harmonically related, one can write the modulated carrier as,

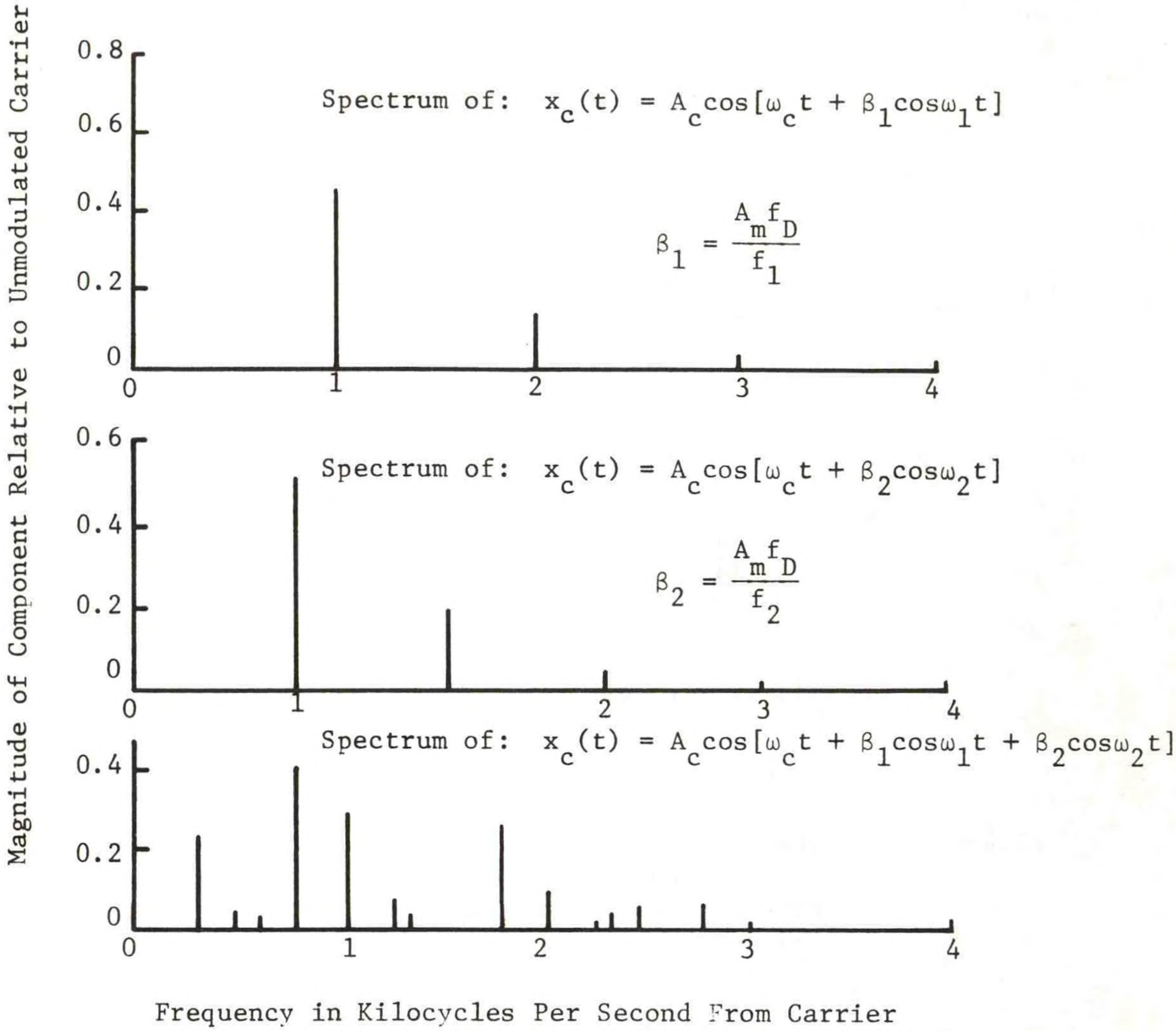


Figure 2-4. Spectra Showing the Failure of Superposition in Frequency Modulation.

$$x_c(t) = A_c \sum_n^\infty \sum_m^\infty J_n(\beta_1)\, J_m(\beta_2)\, \cos(\omega_c + n\omega_1 + m\omega_2)t \quad .$$

Again, without including the details, this may be interpreted in the spectral domain as being divided into four categories:

1. the carrier alone at f_c
2. sidebands at $f_c \pm nf_1$ due to f_1 alone
3. sidebands at $f_c \pm mf_2$ due to f_2 alone

and 4. sidebands at $f_c \pm nf_1 \pm mf_2$ due to the beat-frequency modulation between the two tones.

This last category is the result of the modulation process being non-linear and, as would be expected, gives rise to the feature of FM that spectral superposition does not hold and as such data interaction can occur if the modulated carrier is bandlimited to any extent. Figure 2-4 demonstrates this effect by presenting the single and sum spectra of two tones.

The preceding discussion has been brief, but has demonstrated the data dependence of the modulated subcarrier spectrum. The basic reason for including this discussion is to admit that bandpass filtering the modulated carrier may cause distortion of the baseband signal and interaction of spectral harmonics. Standard rules of thumb are common in selecting bandwidth requirements and are conditioned on the data characteristics. For example "Carson's Rule" provides that for most applications, a bandwidth of $\pm$ $(f_D + f_m)$ about the carrier frequency is adequate (f_D is the previously mentioned deviation constant and f_m is the highest frequency in the baseband signal). These topics are explored in greater detail in references 2-1 and 2-2.

3.0 Z-AXIS RACK OPERATION

This section presents a qualitative discussion of the operation of the z-axis rack. Included is an introduction to pulse-averaging discriminator operation and utilization of the output of the reference discriminator in order to compensate for tape recorder speed error.

3.1 Discriminator Operation

A block diagram of the basic pulse-averaging discriminator (see ref. 3-1) used in the z-axis rack is shown in Figure 3-1. The input bandpass filter shown is not necessary for the discussion but is included for completeness.

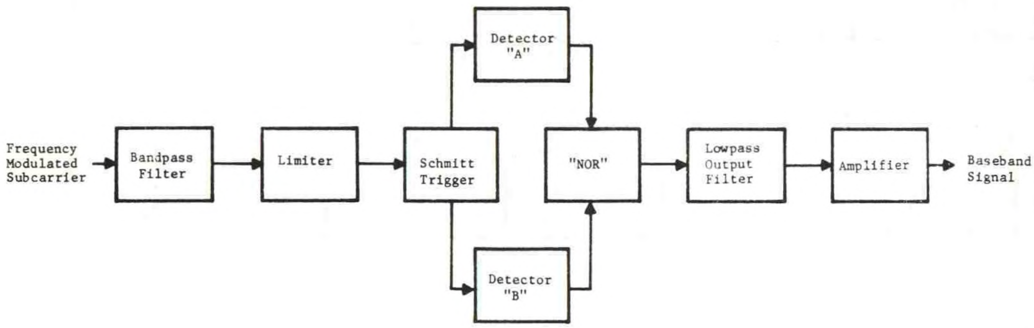


Figure 3-1. Basic Discriminator Block Diagram.

Figure 3-2 indicates waveforms encountered at various locations in the discriminator and may be used to describe the functional operation of this type of FM demodulator. Figure 3-2-a indicates a simple 290 Hz base-band tone which will be used to modulate a 5000 Hz sub-carrier. (It should be noted that the waveform parameters such as subcarrier frequency, etc., were chosen for ease of graphic presentation rather than as representative of the actual hardware configuration.) A portion of the modulated subcarrier is shown in Figure 3-2-b. This represents the input to the discriminator.

This frequency modulated subcarrier is first "hard limited" at the input in order to remove any residual amplitude modulation that may have been introduced in the satellite modulator, multiplexer and transmitter or in the ground station receiver and demultiplexer. The resultant waveform is shown in Figure 3-2-c. It should be remarked that limiting does not theoretically introduce base-band distortion in that all base-band information

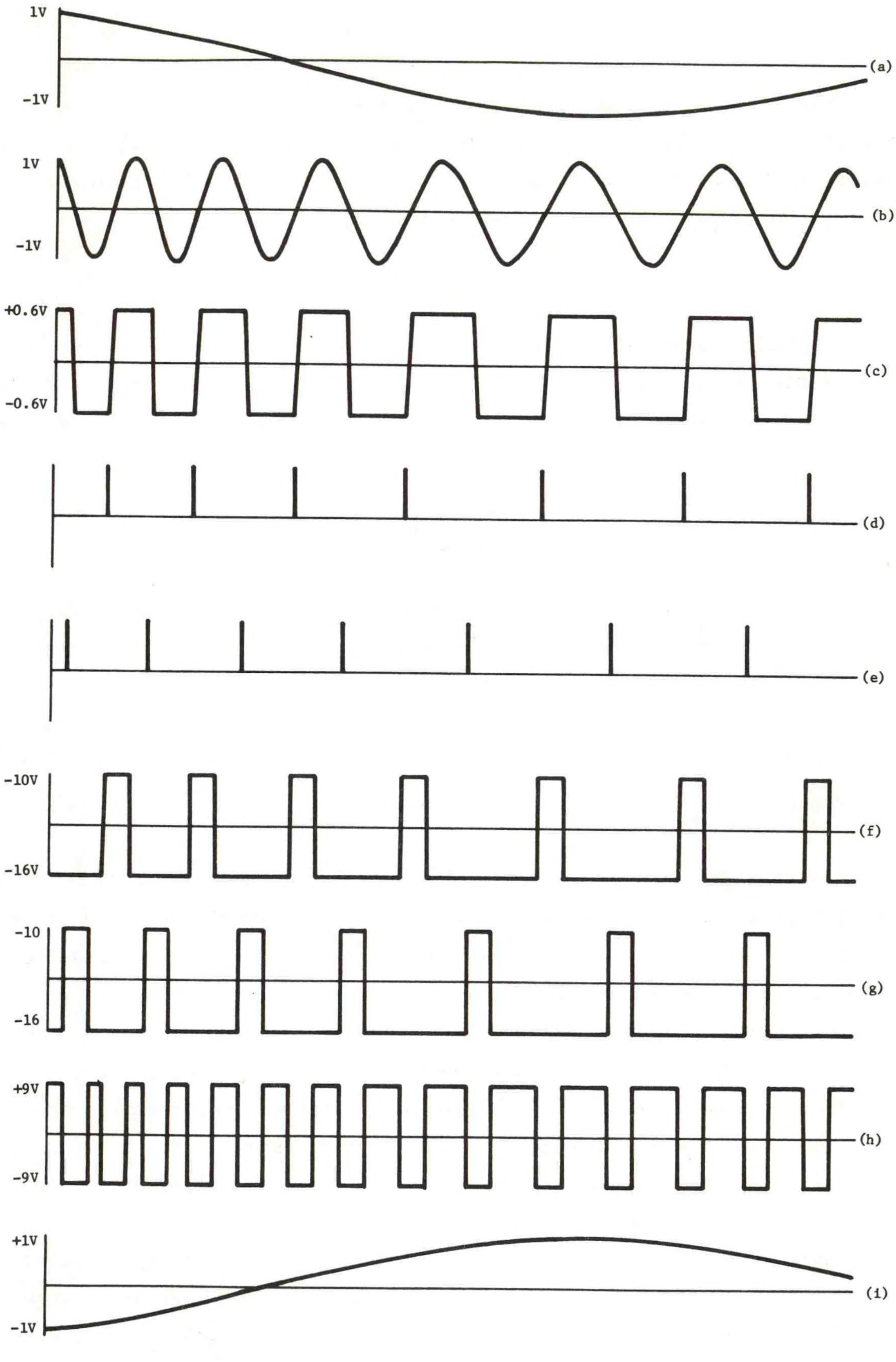
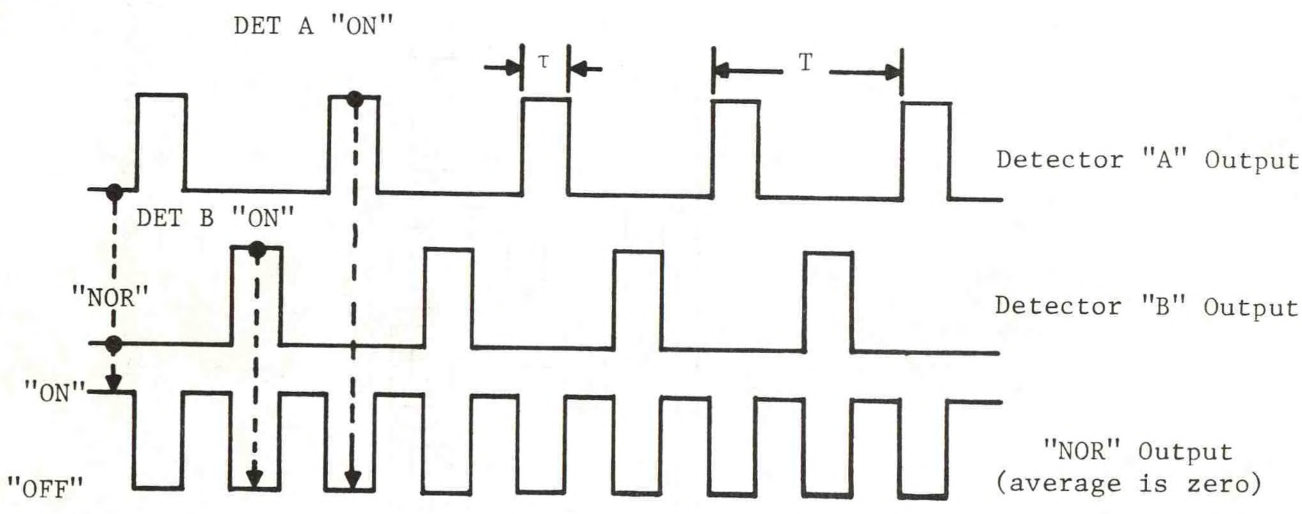


Figure 3-2. Basic Discriminator Operation - Waveforms.

is contained in the zero-crossings of the frequency modulated subcarrier. These crossings are accurately preserved in the process of limiting.

The limited subcarrier is then fed to a Schmitt Trigger for purposes of detecting the aforementioned zero crossings. The Schmitt Trigger outputs a pulse each time the limited waveform passes through zero. A separate pulse is output depending on whether the zero crossing was positive going or negative going. The upcrossings are indicated in Figure 3-2-d and the downcrossings in Figure 3-2-e. The upcrossings are fed to detector "A" and the downcrossings to detector "B" where the pulses are "stretched" to have a width which is one-fourth the period of the unmodulated subcarrier. This choice of pulse width is not arbitrary. It results from the constraints that an unmodulated subcarrier will provide zero at the output of the demodulator and that one of the effects of the "NOR" is to act as a frequency doubler. Figure 3-3 indicates the relationship between the subcarrier period and pulse width in the unmodulated situation. (This figure will be more informative if referred to again following the discussion of the "NOR" circuit and low pass output filter.)



T is period of the subcarrier

τ is the detector pulse width (= T/4)

Figure 3-3. Constraints on Detector Pulse Width.

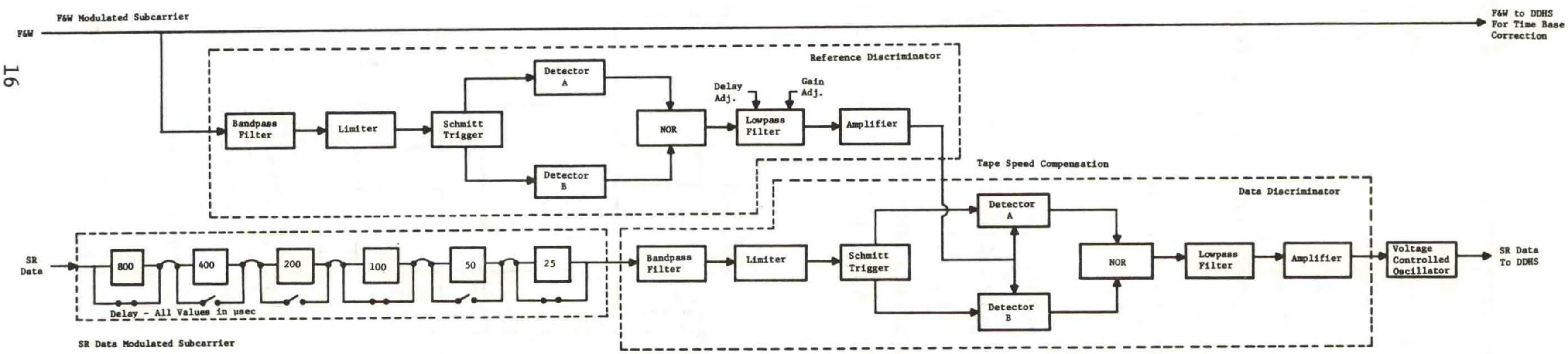


Figure 3-4. Z-Axis Rack Compensation Technique.

The outputs of detectors "A" and "B" are shown in Figures 3-2-f and 3-2-g, respectively.

The stretched pulses are fed to the "NOR" circuit. This circuit produces an "ON" condition if there is a pulse from neither detector "A" nor detector "B" as shown in Figure 3-2-h. Notice that the time the waveform spends in the "ON" condition varies with time while the time spent in the "OFF" condition does not. This waveform when low-pass filtered (i.e. averaged) will produce the waveform shown in Figure 3-2-i. Notice that the original base-band signal is recovered exactly except for a change in polarity. The polarity is reversed in the output amplifier.

3.2 Flutter Compensation Technique

The effect of tape speed error on the spacecraft tape recorder is to shift the data sub-carrier frequency to a higher frequency when the recorder is running faster than nominal and to a lower frequency when the tape recorder is running slower than nominal. Since flutter is a time varying error in recorder speed, its effect is to deviate the carrier from nominal in some time varying fashion. This is equivalent to frequency modulating the sub-carrier with the flutter variations. By recording an unmodulated sub-carrier on a separate track on the tape recorder, a flutter only signal may be recovered upon playback for use in correcting the flutter corrupted data subcarrier.

A detailed block diagram of the correction scheme is shown in Figure 3-4. Further design details may be found in ref. 3, but have been omitted here as they do not substantially contribute to the theory of operation of the compensation technique. The basic theory behind the technique is to translate the flutter only signal to baseband (demodulate the subcarrier) by means of a "reference" discriminator. This signal is then used to vary the pulse width in the data discriminator detectors. When the "NOR" output is subsequently integrated (in the low-pass output filter), the result is as if no flutter had been present.

Figure 3-5 indicates the "NOR" output waveforms for the uncompensated and compensated situations. Figure 3-5-a shows the uncompensated case as was also seen in Figure 3-2. Notice that in the "OFF" state (-9v), the duration per cycle is constant due to a pulse being present with nominal width.

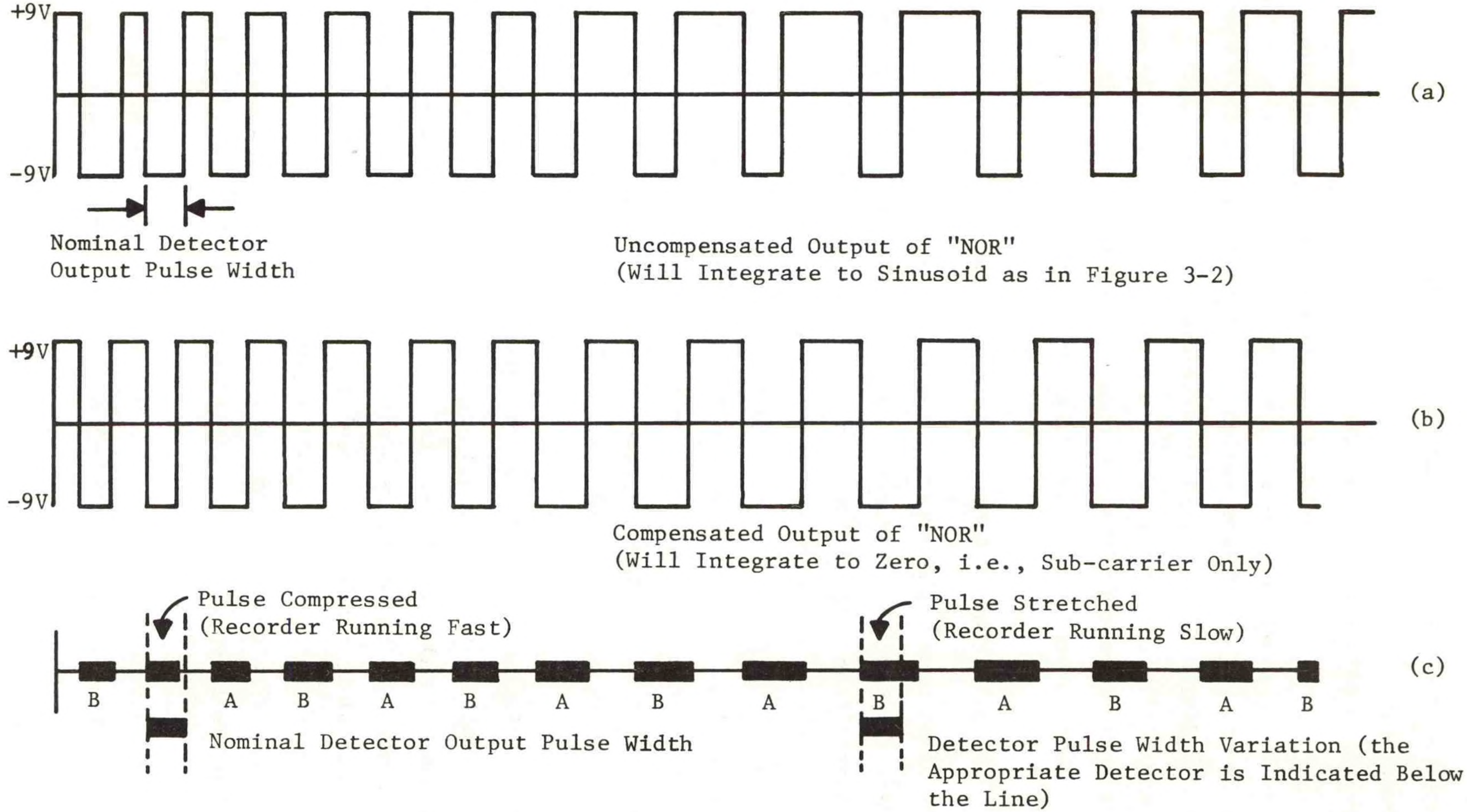


Figure 3-5. Compensation of Flutter only Condition.

In Figure 3-5-b is shown the output of the "NOR" after the detector pulse widths have been modified. For the case where the recorder is running faster than nominal, the pulse width is compressed while when the recorder is running slower than nominal, the pulse width is expanded. Note that although the frequency of the resultant waveform varies as a function of the flutter, the output of the low-pass filter (an integrator) will be zero indicating that the flutter effect has been removed. It should be emphasized that the present discussion is concerned with a flutter only situation. In the presence of joint flutter and data information, only the flutter components are (theoretically) affected; the data are preserved intact.

At this point it is logical to discuss the prominent features of rack alignment as they relate to overall performance. The salient features of interest here are, 1) the scaling between the reference discriminator output and the extent of detector pulse width modification, and 2) the phase (or time) synchronization between the reference discriminator output and the detector outputs in the data discriminator.

With regard to the reference discriminator output scaling, consider the compensated data channel detector output pulse width to be represented by

$$T_{nominal} + K \cdot V_{flutter}(t)$$

where $T_{nominal}$ is (as has been previously discussed) constrained to be one-fourth the sub-carrier period and $V_{flutter}(t)$ is the output voltage of the low-pass filter in the reference channel. The parameter "K" is then the scaling required to match this voltage to the correct percentage expansion and compression of the detector output pulses in the data channel to effectively remove the flutter. If "K" is zero, no compensation has taken place. On the other hand if "K" is less than the correct value, insufficient compensation will occur. Further, if "K" is greater than the correct value, the data channel will be overcompensated and the flutter components enhanced. This behavior can be viewed as linear up to a saturation point where the stretched pulses begin to overlap or the compressed pulses begin to vanish. (This saturation effect was not considered during the study as this situation is considered non-normal.)

Temporal synchronization is required as a result of two features in the ITOS SR data processing channel. First, the flutter and data signals are modulated on different subcarrier frequencies in the multiplex scheme and thus have different transport times through the MUX/DEMUX electronics. Second, the flutter baseband signal is obtained <u>after</u> the low-pass filter and amplifier in the reference channel discriminator and used to modify the data channel discriminator at the detector circuits. Thus the transport delay(s) of the "NOR", low-pass filter, and output amplifier of the reference discriminator must be incorporated. The majority of the synchronization is achieved by delaying the data channel prior to demodulation. A secondary adjustment is available in the reference discriminator. However, it appears to be a phase adjustment only and thus does not possess the range of the main delay. In order to effect proper cancellation, the time delay is adjusted to synchronize the reference channel and the data channel. In the event that the two channels become phased 180° with respect to each other, a flutter enhancement occurs instead of cancellation. The extent of misalignment to cause enhancement is not considered likely at the ITOS identified flutter frequencies but could be significant if higher flutter frequencies were encountered.

4.0 MODEL DESCRIPTION

The basic approach to the model is that of a time domain simulation rather than development of a numerical treatment of theoretical transfer functions. This is largely the result of the non-linear functions encountered in the limiter, Schmitt Trigger, detectors, and the "NOR" circuits. The only exceptions to this design philosophy are the input band-pass filters and the lowpass output filters, where the frequency domain transfer functions were analytically specified and Fourier-transform, multiplication, inverse transform techniques were employed to avoid time-consuming numerical convolutions. Figure 4-1 shows an overall block diagram of the model implementation. The various approaches to each of the functional blocks are described in the following paragraphs. Table 4-1 indicates the parameter options incorporated into the model.

4.1 Reference Channel

The reference channel subcarrier frequency (f_o) is set at 6.25 Hz in order to be compatible with ITOS-D at 4:1 playback. The deviation index was selected to be $\pm$ 15% about this. The input band-pass filter was modeled about these parameters. Initially a theoretical complex frequency plane description for the filter transfer function was attempted but was rejected due to design complexity, and an ideal (rectangular amplitude and linear phase response) filter substituted. This has the effect of producing better transmission characteristics within the band-pass but introduces possible distortion as the base-band information contained in the higher order sidebands of the modulated subcarrier is rejected. This is considered to be negligible for the tone modulation used in the analysis.

The input amplifier, limiter, and cascade amplifier parameters were selected to be compatible with the EMR discriminator described in ref. 3-1.

The Schmitt Trigger (zero crossing detector) was forced to trigger on a threshold of + 1 mv for upcrossings and -1 mv for downcrossings to account for quantization noise contained in the (model) digitized frequency modulated subcarrier. A distinct deviation from the EMR discriminator design was the substitution of +9 volt level for an "on" condition and a -9 volt level for an "off" condition in the pulse stretching detectors (normal levels are -10 volts for "on" and -16 volts for "off"). This deviation resulted from a requirement to change pulse <u>height</u> instead of width in the compensation scheme in order to achieve

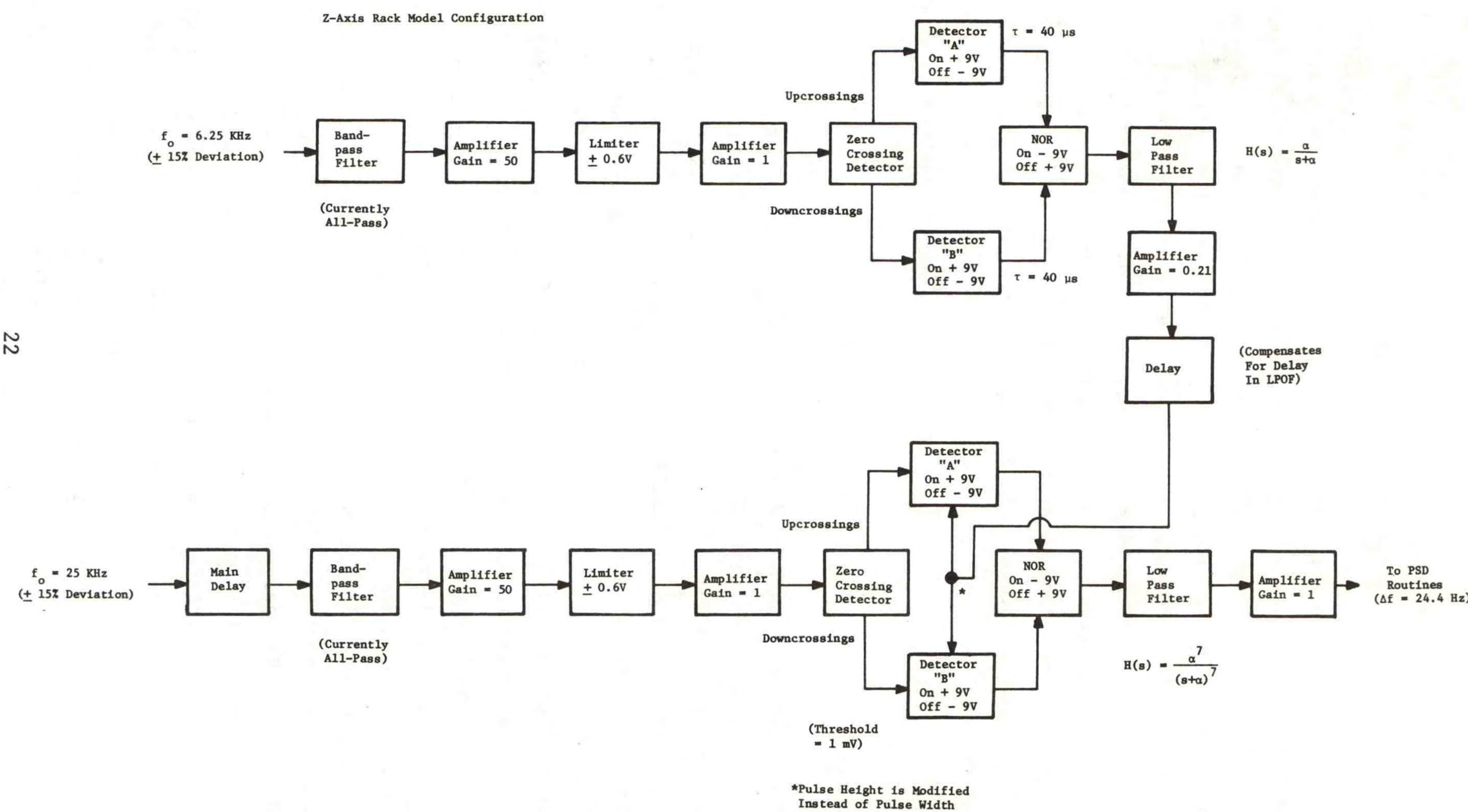


Figure 4-1. Overall Block Diagram of Model Implementation.

TABLE 4-1. PARAMETER OPTIONS

1. Transfer function of band-pass filter(s)
2. Input gain of limiter(s)
3. Clipping level
4. Output gain of limiter(s)
5. Inherent delay of limiter
6. Threshold of Schmitt Trigger(s)
7. Inherent delay of Schmitt Trigger
8. Pulse width scaling of detector (data only)
9. Pulse width of detectors
10. Output amplitude of detectors for "ON" condition ("ON" is pulse)
11. Output amplitude of detectors for "OFF" condition ("OFF" is no pulse)
12. Inherent delay of detectors
13. Output amplitude of "NOR" for "ON" condition
14. Output amplitude of "NOR" for "OFF" condition
15. Inherent delay of "NOR"
16. Transition voltage of "NOR"
17. Transfer function of low-pass filter(s)
18. Gain of output amplifier(s)
19. Main delay (error)

finer resolution (i.e., each pulse width represents a small finite number of words in the algorithm, thus a small amount of width compensation would be absorbed in time base quantization effects). The "NOR" circuit was modeled as a logical equivalent of the EMR circuitry with the specified levels of $\pm$ 9 volts. Notice that the utilization of a "NOR" has the effect of doubling the subcarrier frequency and phase inverting the output signal (see section 3.0).

The output low-pass filter was modeled in the complex frequency domain and transform-inverse transform techniques employed. The filter has a 6 db/octave roll-off and serves to integrate the output of the "NOR" circuit and to attenuate the contribution of the doubled subcarrier frequency.

The gain of the output amplifier was determined empirically for a single tone (see Section 5.0) and the polarity chosen to compensate for the phase inversion of the "NOR" circuit. Insofar as the amplifier is linear over its operating region the use of a single tone to determine the gain is justified.

An additional "delay" was incorporated into the model at the output of the reference discriminator to advance the reference signal with respect to the data subcarrier due to the delays encountered in the reference channel input bandpass filter and lowpass output filter. This would normally be achieved in the main delay in the data channel; however, in order to preserve the latter delay as a main delay "error", it was introduced at this point in the model.

4.2 Data Channel

The data discriminator is modeled essentially in the same manner as the reference discriminator with the following exceptions. First, the data channel subcarrier was shifted to 25 KHz instead of 22 KHz utilized in the hardware. This was necessitated by the requirement that the detector output pulse width be one-fourth the period of the subcarrier frequency and the requirement that this time span occupy an integer number of words in the computer algorithm. Since the percentage deviation remained at $\pm$ 15%, this does not reflect any significant restriction on the model capability. Second, since the base-band signals and subsequently modulated subcarriers are generated within the algorithm, they are initially in phase and the requirement for a main delay per se does not exist. The main delay actually incorporated into the model represents a delay "error" and is more meaningful in terms of overall rack operation. Third, the low-pass output filter was modeled (again in the complex frequency domain) as a seven pole passive filter with a roll-off of 42 dB per octave. This closely approximates the 40 dB/octave EMR specification. Fourth, the gain of the output amplifier was taken to be unity for convenience. This does not represent any compromise on model performance insofar as the areas of interest are represented by relative rather than absolute results.

4.3 Analysis Technique

The technique used for analysis of the results obtained from the model consisted primarily of the examination of flutter residuals and the sporadic harmonic behavior due to modulation non-linearities in the spectral domain. A sample output is shown in Appendix C. Additional comments regarding the specific analyses conducted are contained in the following section.

5.0 SENSITIVITIES DERIVED FROM MODEL

The model was exercised parametrically to determine sensitivity to significant parameters. While the capability exists to analyse each of the parameters listed in Table 4-1, this study concentrated on the main delay and reference discriminator output gain settings. The reasoning behind this choice is twofold: one, the other parameters are considered to provide second order effects and, two, these two parameters represent critical adjustments in the z-axis rack alignment. These two parameters were initially tested by observing the degree of cancellation of a single tone imposed on both the data and flutter channels. Subsequently, analyses were conducted with multiple tones and random noise backgrounds.

Figure 5-1 shows the flutter residual after cancellation of a 1.2 count* tone at 196 Hz as a function of delay error in micro-seconds. A change in delay of 12.5 μ sec would produce a change in flutter residual of approximately 0.02 counts at this frequency, representing a change slightly in excess of 1.5%. Since 25 μ sec is the resolution of the delay line, it would appear that the ability to achieve the proper delay is available (i.e., ± 12.5 μ sec) providing the required total delay is within the range of the delay line (1575 μ sec). Notice that since what is being affected in the compensation is a cancellation effect, as the tone frequency increases, the apparent total <u>phase</u> change due to 12.5 μ sec error also increases. At 800 Hz (approximately the highest flutter frequency identified on ITOS-D), 12.5 μ sec would produce a phase change equivalent to 50 μ sec at 200 Hz. From Figure 5-1, one can roughly anticipate a flutter residual of 0.08 counts or about 6% of peak. Thus based on pure tone cancellation, the delay line adjustment resolution is adequate for z-axis operation. It will be seen later that the non-linearities associated with the modulation technique make the establishment of the proper delay setting difficult in an operational environment.

---

*The term "count" represents the output of an eight-bit digitizer at DDHS which encodes the dynamic range of the sensor into a range of 256 levels. Strict details of this encoding are omitted here. It is sufficient to remark that low count values represent high temperatures while high count values represent low temperatures. Two counts are roughly equivalent to one degree Kelvin at the warm end and this scale factor is fairly linear over the temperature range of interest.

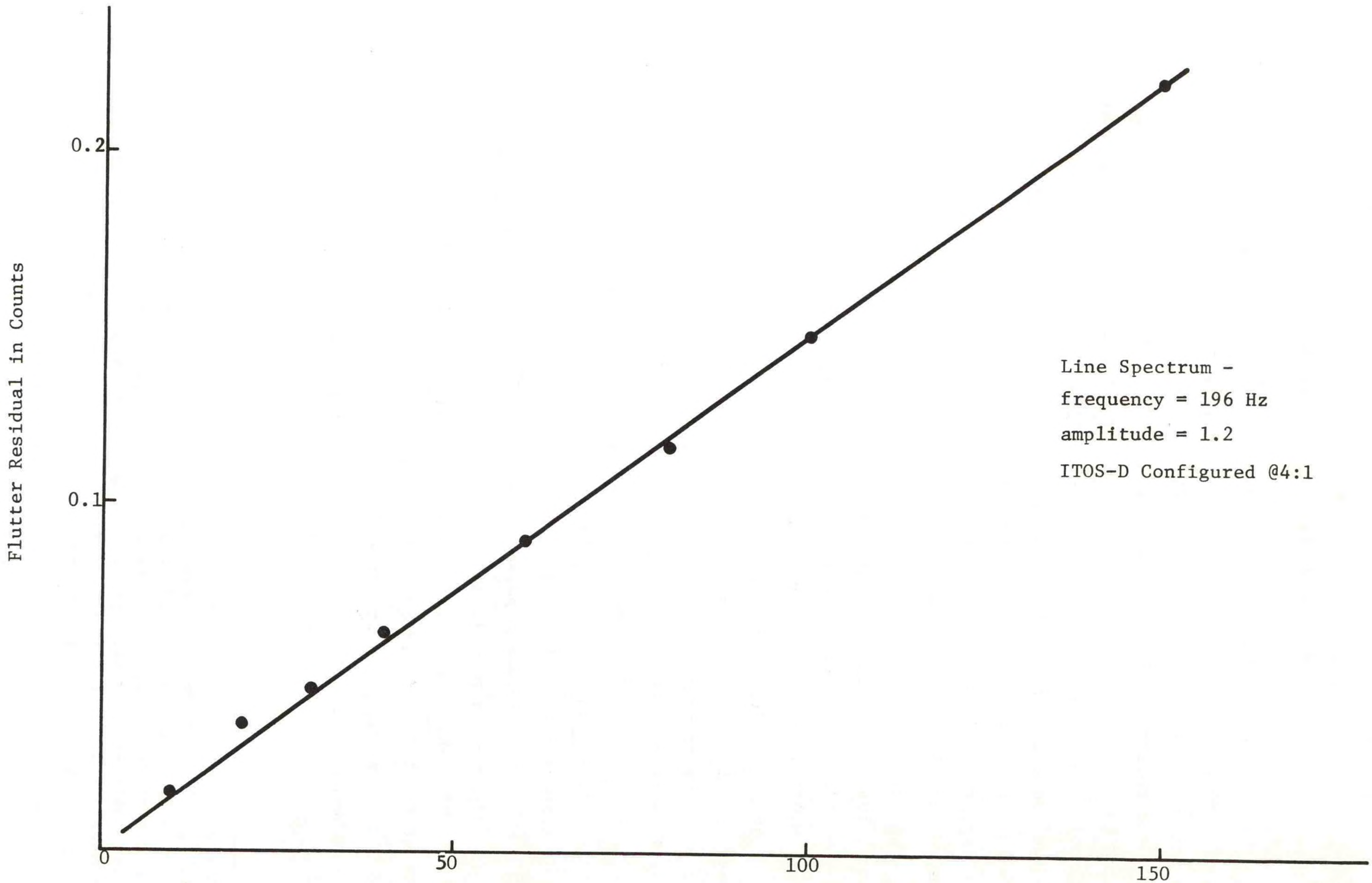


Figure 5-1. Flutter Residual versus Main Delay Error in μs.

Figure 5-2 indicates the sensitivity of flutter residual as a function of reference discriminator output gain. Notice that zero gain represents no compensation and the original tone amplitude of 1.2 counts is preserved. In the range of zero to -0.21 the data channel is undercompensated while in the range of -0.21 to -0.6 overcompensation occurs. The gain is negative in sign to account for the inversion which occurs in the "NOR" circuit as described in Section 3.0. In the actual hardware this is achieved in the output amplifier while in the model it is achieved mathematically as the sign on the value input for the gain parameter. It should be noted that the absolute value of gain was derived empirically in order to effect model performance and may not reflect the actual hardware gain of the output amplifier. The value is indicative of relative behavior. No attempt was made to examine the saturation effect described in Section 3.0. As in the case of the delay setting, it will be shown later that an optimum gain setting will be difficult to achieve in an operational situation.

Table 5-1 shows the effect of modulating the data and flutter channels with four tones nominally identified to have the same frequency and amplitude as ITOS-D flutter components, and subsequently observing rack operation as the main delay is perturbed from its nominal setting. Notice the tone at 782 Hz deteriorates back to the no compensation condition more rapidly than lower frequencies. This is caused by a given delay representing a larger phase error at the higher frequencies. As a result, the no-compensation and flutter enhancement situations are reached sooner at the higher frequency tones. Notice that this behavior is similarly observed for tones at 102 Hz and 417 Hz. Note that a contrary effect is observed for the tone at 156 Hz where a delay error actually improves the rack performance until an error of 100 μs is reached. This is considered to be due to the fact that due to the significant difference in tone amplitude, the tone is spectrally distributed differently and that in effect the non-linearity of the modulation technique has produced a phenomenon wherein the rack would appear to be exceeding the theoretical single tone performance. It should be commented that this type of data interaction was found to be prevalent in all multiple tone analyses. It is this type of behavior which causes the actual hardware to be difficult to align operationally.

Table 5-2 indicates the effect of varying the reference discriminator gain on the rack performance with the four tones described in the previous paragraph. Notice that the phenomenon of data interdependence is much more

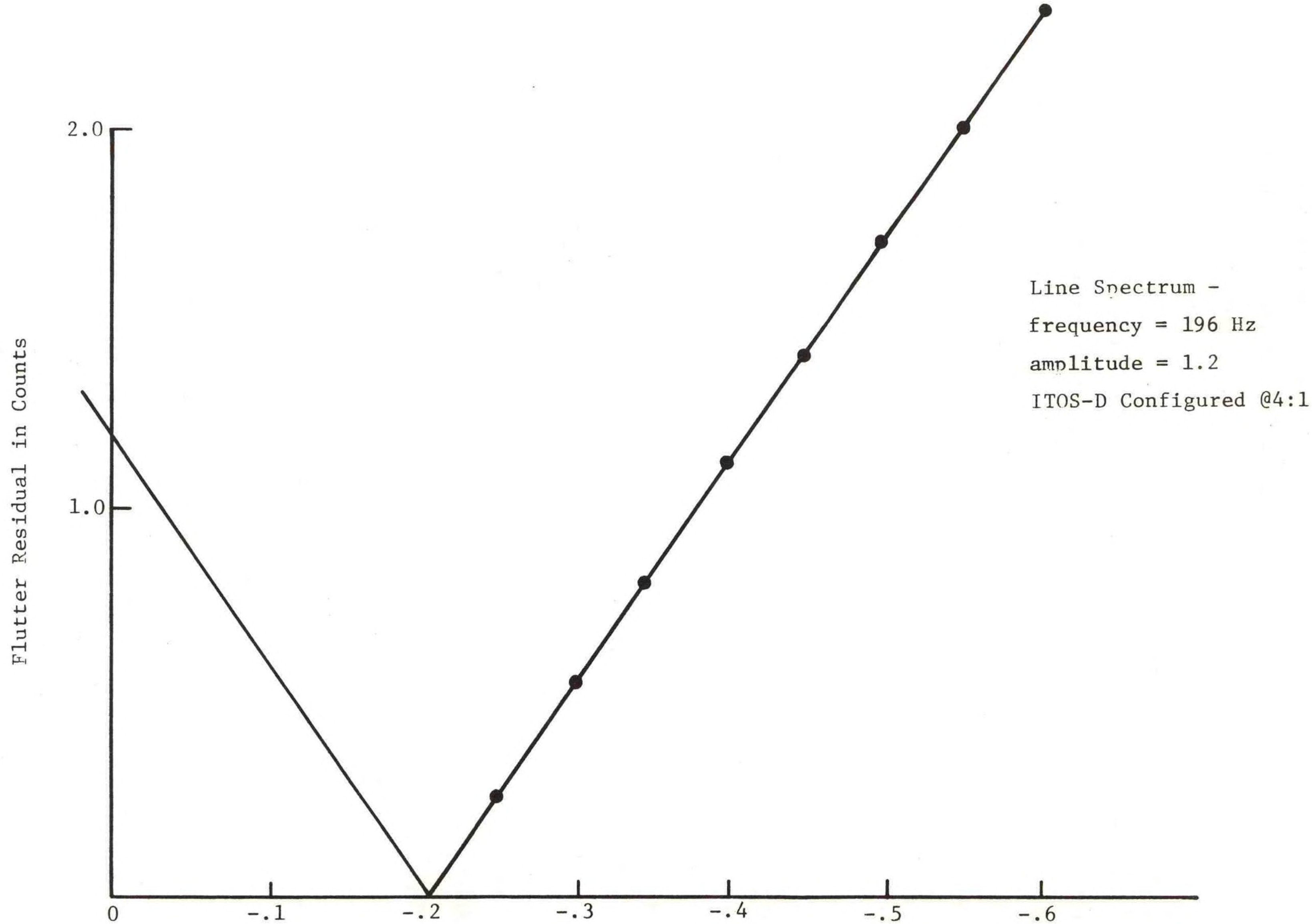


Figure 5-2. Flutter Residual versus (Model) Ref. Disc. Output Gain.

TABLE 5-1. TONES ON DATA AND FLUTTER CHANNELS - DELAY ERROR EFFECT
ITOS-D CONFIGURED AT 4:1

Delay Error μs	102 Hz	156 Hz	417 Hz	782 Hz
No Compensation	0.439	0.109	0.763	0.370
0	0.020	0.011	0.017	0.081
10	0.021	0.018	0.028	0.086
20	0.022	0.018	0.041	0.087
30	0.024	0.009	0.055	0.082
40	0.026	0.007	0.075	0.089
60	0.031	0.004	0.115	0.110
80	0.036	0.002	0.153	0.135
100	0.041	0.001	0.192	0.162
150	0.052	0.009	0.290	0.234
200	0.070	0.007	0.388	0.308

Values for no-compensation represent flutter tone amplitude in counts. The other values represent flutter residual in counts.

TABLE 5-2. TONES ON DATA AND FLUTTER CHANNELS - GAIN ADJUSTMENT EFFECT

Gain*	102 Hz	156 Hz	417 Hz	782 Hz
No compensation	0.476	.112	.780	.379
-0.17	.070	.011	.191	.133
-0.18	.047	.010	.157	.119
-0.19	.024	.012	.122	.105
-0.20	.014	.017	.088	.091
-0.21	.030	.022	.053	.078
-0.22	.053	.028	.021	.065
-0.23	.076	.034	.020	.054
-0.24	.100	.040	.053	.044
-0.25	.124	.047	.087	.038
-0.26	.147	.053	.121	.037

Values for no compensation represent flutter tone amplitude in counts. The other values represent flutter residual in counts.

*The values of gain are model dependent and should be interpreted only in a relative sense.

radical here as evidenced by the (underlined) gain at which each tone is most effectively cancelled. In this case it is virtually impossible to define an optimum setting. Note that the various values tend to cluster around the -0.21 value obtained for the single tone at 196 Hz shown in Figure 5-2.

In view of the fact that the delay and gain adjustments are set simultaneously in an operational situation, the difficulty in identifying an optimum setting for delay and gain is further emphasized. In the model case, the situation is somewhat simplified in that the frequency modulated signals are initially in time synchronization. The only differential delay up to the point of compensation is due to the input band-pass filters of the reference and data discriminators. Since these have been modeled as ideal filters with linear phase response, the transit delay may be calculated exactly (the delay is given by the slope of the phase response). Thus in the model, lack of knowledge about the delay setting does not influence the gain setting analysis. However, in the hardware situation, the alignment resorts to an iterative process. It has been observed in actual testing that the main delay setting is the better initial parameter to adjust in achieving the final alignment.

Table 5-3 indicates an analysis where no tone modulation was applied to either the flutter or data sub-carriers and noise alone imposed on the flutter sub-carrier only. The intent here was to demonstrate the fact that noise on the flutter channel would result in increased noise level on the data channel output via the flutter compensation technique. The notation is defined as follows:

σ^2_{NC} = the noise variance of the data channel output with no compensation

σ^2_{C} = the noise variance of the data channel output with compensation

σ^2_{n} = the contribution of noise in the data channel output due to noise on the flutter channel.

The data channel output has an apparent noise output due to quantization effects in the digital model. The important parameter to view here is σ^2_n as a function of the bandwidth of the low-pass output filter on the reference discriminator. As would be expected, as the bandwidth is reduced, the noise variance reduces nearly proportionately.

Table 5-3. Noise on Flutter Channel - White, Uniform Dist., $\sigma^2 = 0.11$.

Bandwidth of low-pass output filter	σ^2_{NC}	σ^2_C	σ^2_n
2 kHz	0.012	0.097	0.085
1 kHz	0.012	0.053	0.041
500 Hz	0.012	0.032	0.020
250 Hz	0.012	0.021	0.009
100 Hz	0.012	0.017	0.005

$\sigma^2_n = \sigma^2_{NC} - \sigma^2_C$ (since the noise samples are uncorrelated).

Table 5-4. Noise and Four Tones on Flutter Channel, Four Tones on Data Channel

dB Improvement

Bandwidth of low-pass output filter	102 Hz	156 Hz	417 Hz	782 Hz	σ^2_C
2 kHz	24.0	13.0	49.9	34.0	0.244
1 kHz	22.7	13.2	23.8	14.4	0.197
500 Hz	23.5	20.1	13.4	6.2	0.216
250 Hz	22.0	23.9	5.5	0.3	0.447
100 Hz	12.6	5.8	-1.0	0.0	1.055

$\sigma^2_{NC} = 1.119$

Values represent "dB Improvement" as defined in Appendix C.

Table 5-4 shows the effect of re-instituting the four tones on the data and flutter channels in the presence of a noise background. The values presented here represent "dB improvement" as described in Appendix C. The intent is to show the trade-off of low-pass output filter bandwidth reducing the total noise on the data channel versus flutter cancellation effectiveness. Notice that as the bandwidth is reduced the cancellation ability degrades beginning at the higher tone frequencies as would be expected. The -1.0 "dB improvement" at the 417 Hz tone for a 100 Hz filter is in reality a flutter enhancement and is likely due to the odd spectral distribution of the modulated carrier which gives rise to the data interaction described previously. The 2 KHz filter provides superior cancellation performance to the lower bandwidth filters in all cases. However a reduction to 1 KHz reduces the overall noise variance and still provides acceptable cancellation. This kind of optimization of bandwidth versus performance would be more emphatic if the flutter noise was not white noise but was concentrated about certain flutter harmonics. Figure 5-3 shows the spectral distribution for the first 100 lines ($\sim$2440 Hz) of the data channel output for the first three cases in Table 5-4. The flutter tones are indicated by the small diamonds below each plot. Of particular interest here is the behavior of the lines above the filter cut-off. For example, in Figure 5-3-c, wave numbers 90 and 93 have been significantly enhanced even when they are well in excess of the range of compensation. This again points out the strong data interaction due to the spectral distribution of the modulated carrier.

Finally, the effect of changing the roll-off rate of the low-pass output filter was investigated. Figure 5-4 shows the spectrum of Figure 5-3-b as compared with a similar situation where the order of the filter was increased from a single pole (roll-off rate of 6 dB/octave) to a two pole configuration (roll-off rate of 12 dB/octave). As would be expected, the <u>total</u> noise variance of the data channel was reduced from 0.197 to 0.170. This can be seen visually by inspecting Figure 5-4 in the region above the cutoff frequency (1000 Hz). However, notice that the cancellation effectiveness has decreased for the higher roll-off rate (note the increase in amplitude at those wave numbers marked with a diamond below each figure). This would lead one to the conclusion that significant information relative to effective cancellation is contained in wavenumbers greater than the cut-off frequency

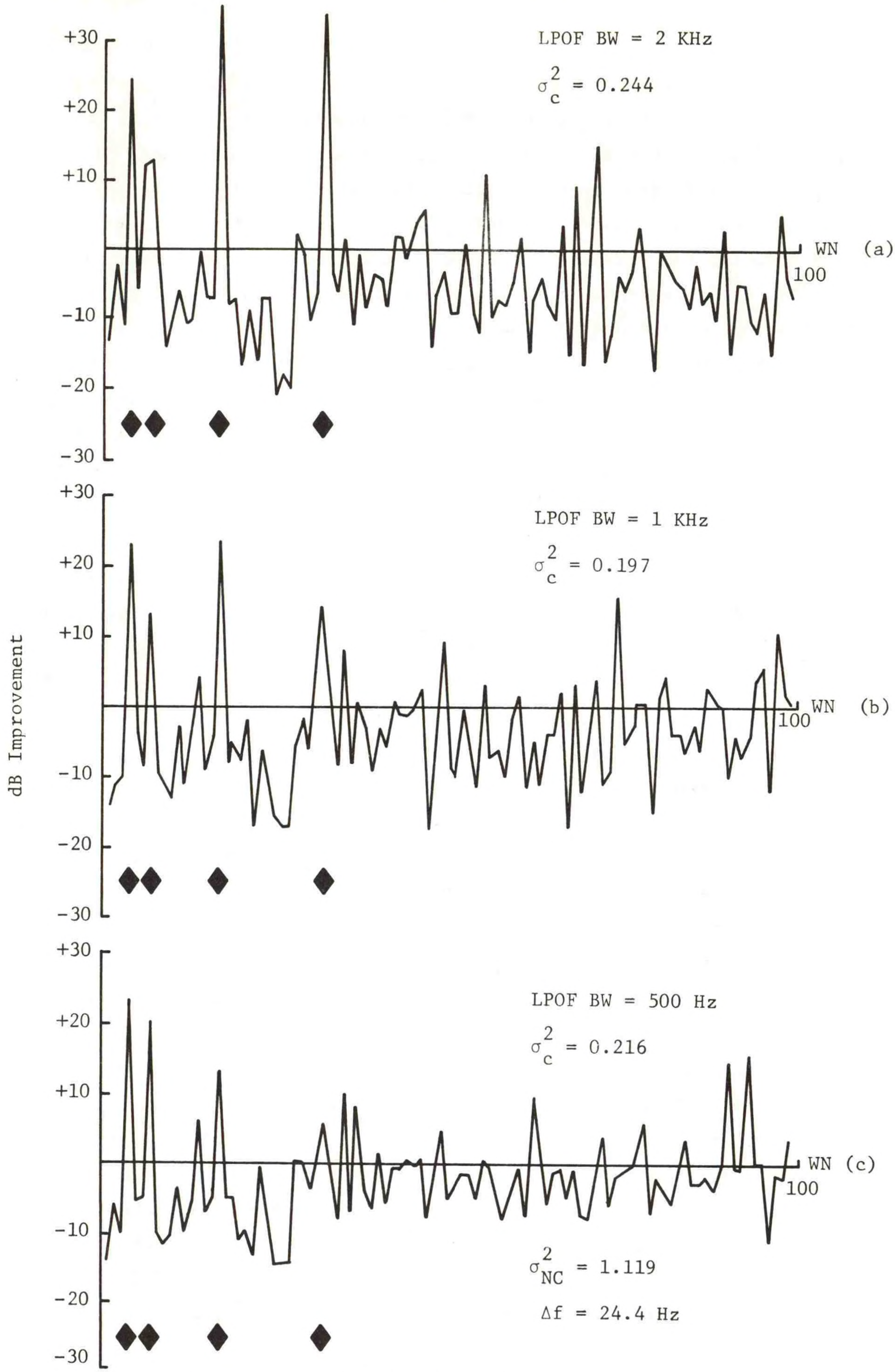


Figure 5-3. Data Channel Output Spectra for First Three Cases Shown in Table 5-4.

and that the lack of this information due to a higher rate of attenuation above cut-off noticeably affects performance.

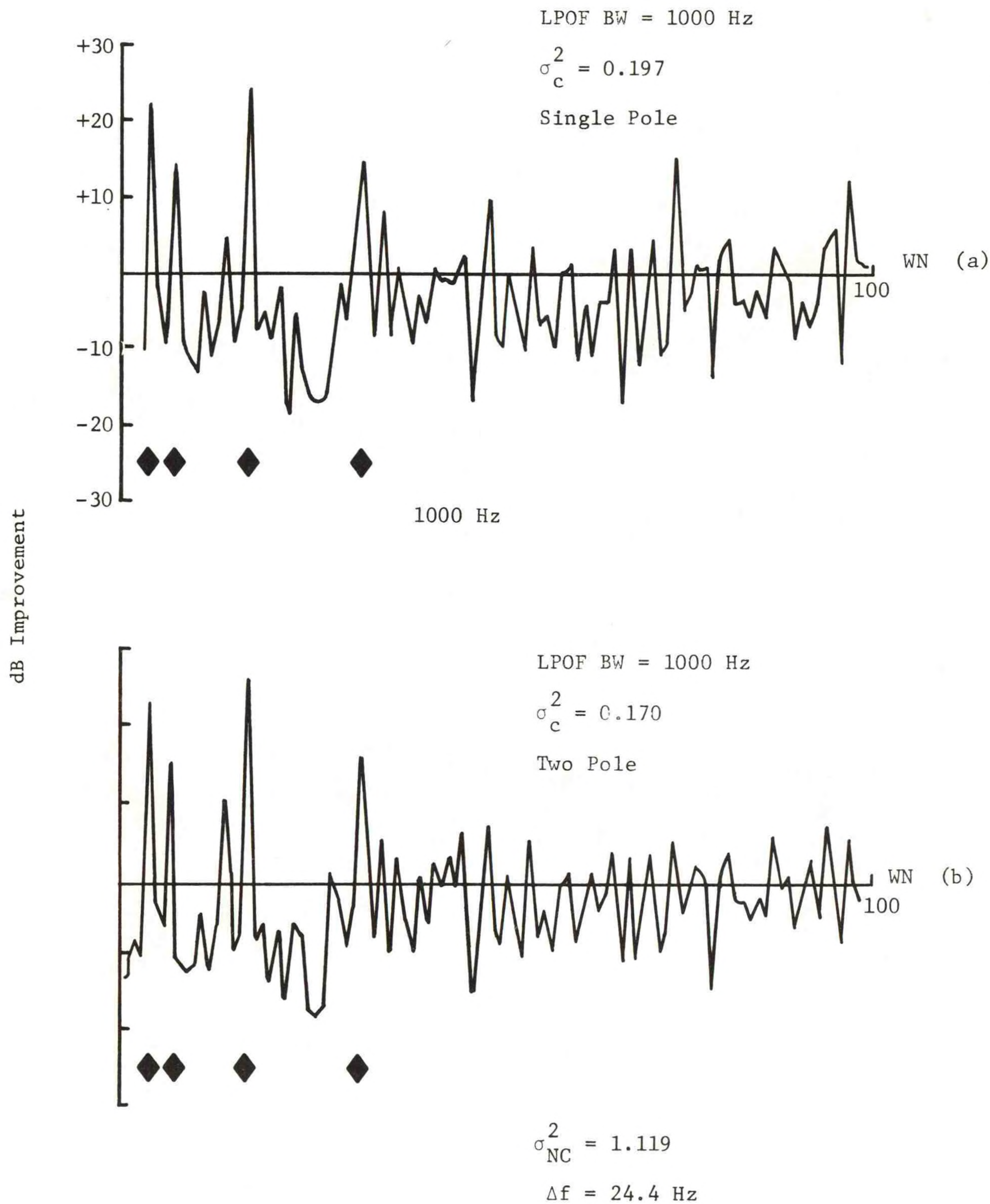


Figure 5-4. Effect of Changing Filter Roll-off.

6.0 PRELIMINARY RECOMMENDATIONS FOR DIAGNOSTIC IMPLEMENTATION

The philosophy of diagnostic implementation on-line at the DDHS is constrained to be a minimum interference technique. On a routine basis it must be rapid, repetitive, and not impose on computer storage requirements. Also, during those ingests where abnormal conditions are suspected, it must be of sufficient detail to be useful in correcting any system fault which may be present. These operational and performance constraints are best met here with an approach to the diagnostic implementation which is adaptive in the degree of detail as a function of the data quality observed.

Figure 6-1 shows a flow diagram of a recommended diagnostic procedure which would accommodate the above philosophy. As a simple measure of performance, an RMS would be computed on the space view each frame during ingest. This value would then be compared with a preset threshold. The threshold would represent the maximum noise level for which average or above average data quality could be assured. In the event the RMS was lower than the threshold, the ingest would continue and the operator advised of a normal condition at the termination of the ingest. In the event the RMS is greater than this threshold, it would be compared to a second (higher) threshold which would represent the maximum RMS noise which would assure the minimum acceptable data quality. If the RMS was lower than this threshold, a 64 sample power spectrum would be computed on the space view. The spectrum would be compared to a stored reference mask to see which flutter harmonics had changed and the result output for analysis. The ingest would continue and the operator advised of marginal data quality at the termination of the ingest. If, on the other hand, the RMS exceeded the maximum acceptable noise level, more detailed spectral analysis would be pursued. It would be ascertained whether nighttime visible data could be obtained within the remaining data to be ingested. If so, then a 1024 sample power spectrum would be computed and output for analysis and the ingest terminated. If not, the ingest would be terminated. The operator would be advised of the status at the termination of the ingest. A decision to reingest would then be made on the status observed and available spectral data.

During the course of the study, the complete diagnostic philosophy was not incorporated for on-line evaluation during ingest. However, an initial diagnostic configuration was developed and provided to NESS personnel for

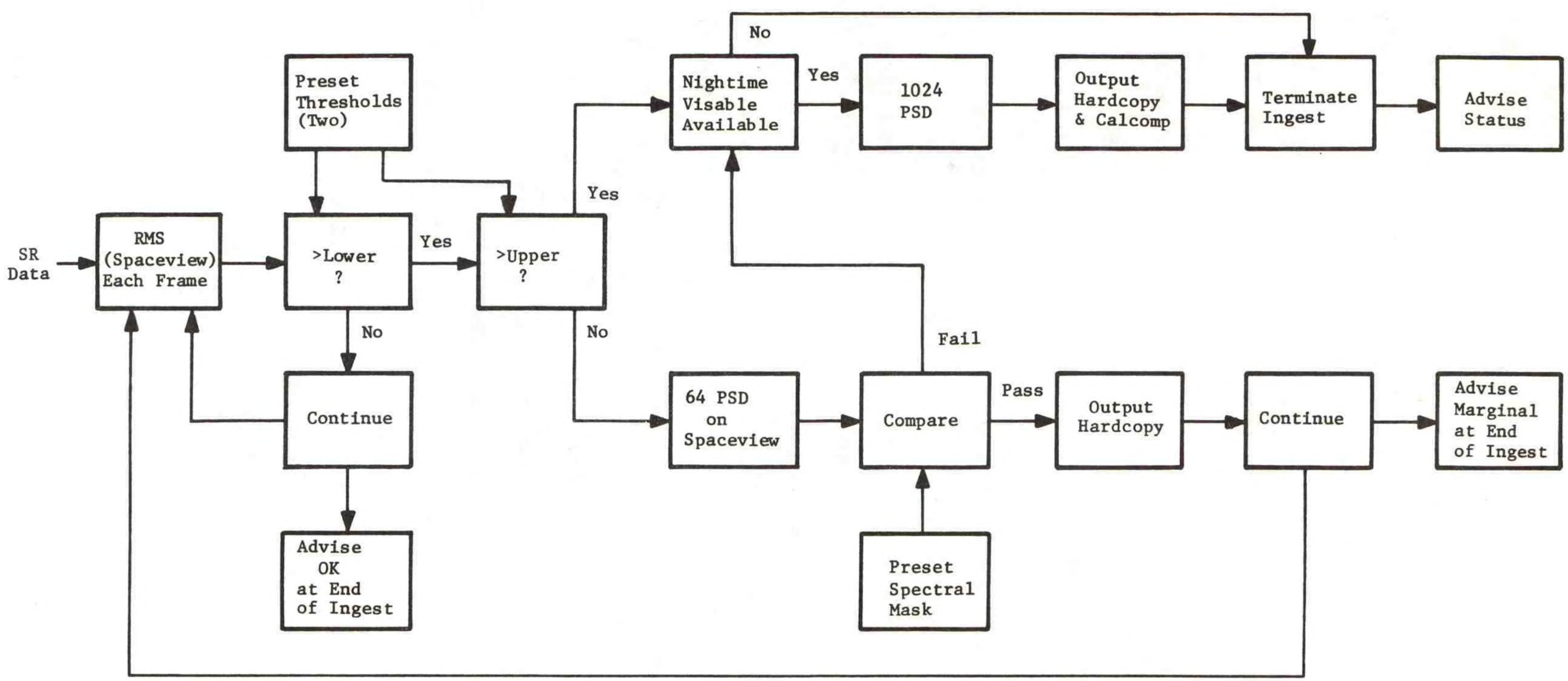
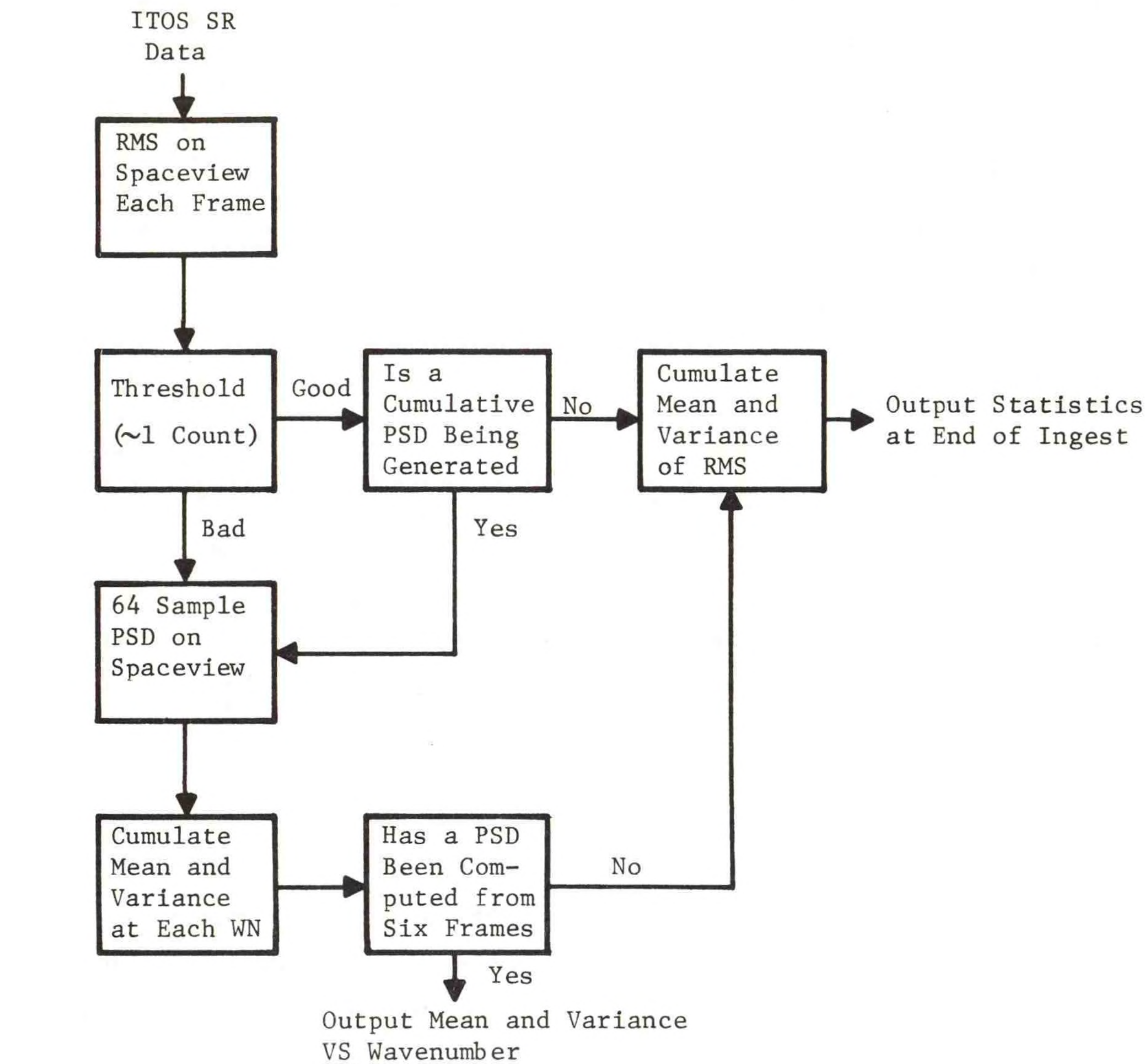


Figure 6-1. Preliminary Diagnostic Logic Flow Diagram.

preliminary simulation testing on the CDC 6600 at Suitland. The key elements of this configuration are shown in Figure 6-2. It is the intent of this type of approach to structure the final diagnostic configuration dynamically in order to maintain compatibility with system changes at NESS. In particular, it was not clear at the time of the conception of this configuration whether night-time visible data would be available, thus it has been deleted from the structure in such a way as to be included later if available.

The concept of performing a 64 sample power spectrum was given special attention in deriving the configuration shown. An analysis was performed to indicate the improvement in spectral stability which could be achieved by averaging. A vector of 1024 random samples with a flat spectrum was generated and its spectrum computed. This spectrum was then condensed to an equivalent 64 sample spectrum by summing the energy over groups of 16 wavenumbers. This result indicated a very flat spectrum as would be expected. The original 1024 sample data vector was then segmented into 16 independent 64-sample data sets and a spectrum computed for each. The results indicated these small sample spectra to be statistically unstable (i.e., flatness was not achieved). A cumulative average was then performed for the small sample spectra to determine the minimum number of spectra which could be averaged in order to reach a stable statistic. The average value at each wavenumber was computed with the number of spectra incorporated varying from 2 to 16. An apparently stable statistic was achieved at a minimum of six individual spectra. This cumulative average was then incorporated into the diagnostic logic.

The configuration shown in Figure 6-2 thus represents the basic cornerstone of the logic to achieve the final goal of stable compensation. It can be expanded and revised in accordance with ground system and satellite changes.



Note: Need to be sure not to compute PSD on all frames (i.e., upper threshold has not been incorporated) in the case of bad data.

Figure 6-2. Initial Diagnostic Configuration.

7.0 RECOMMENDED STANDARD OPERATING PROCEDURES

In order to provide a base for achieving stable overall system operation, a portion of the study was directed at preparing (in co-operation with Wallops CDA personnel) a "standardized operating procedure" for periodic alignment of the z-axis rack hardware. This effort resulted in an over detailed procedure which proved not to be viable on a day-to-day operational basis and was subsequently not formally adopted. It does represent a fairly complete technique for alignment and is included as Appendix E.

8.0 RECOMMENDATIONS FOR FUTURE EFFORT

It is recommended that the effort described in this report be continued toward the overall goal of achieving stable data quality assurance in an operational sense. Additional studies directed toward this end would include analytical investigations to quantitatively predict the effect of system changes (i.e. for future satellite configurations) on data quality and feasibility investigations to expand the scope of the diagnostic logic implementation. Additionally, close co-ordination and support is recommended during the actual operational implementation of this diagnostic logic at NESS.

9.0 APPENDICES

APPENDIX A

MODEL IMPLEMENTATION

This appendix presents a detailed block diagram of the digital computer algorithm which comprises the model developed during the study. A description of the model inputs follows as Appendix B, a sample output description as Appendix C, and a sample listing as Appendix D.

The block diagram indicates program control with the exception of the dotted line showing data flow for the flutter channel output. The subroutine call names are shown in parentheses. Power spectra are available as output at intermediate points in the program and are so indicated on the block diagram.

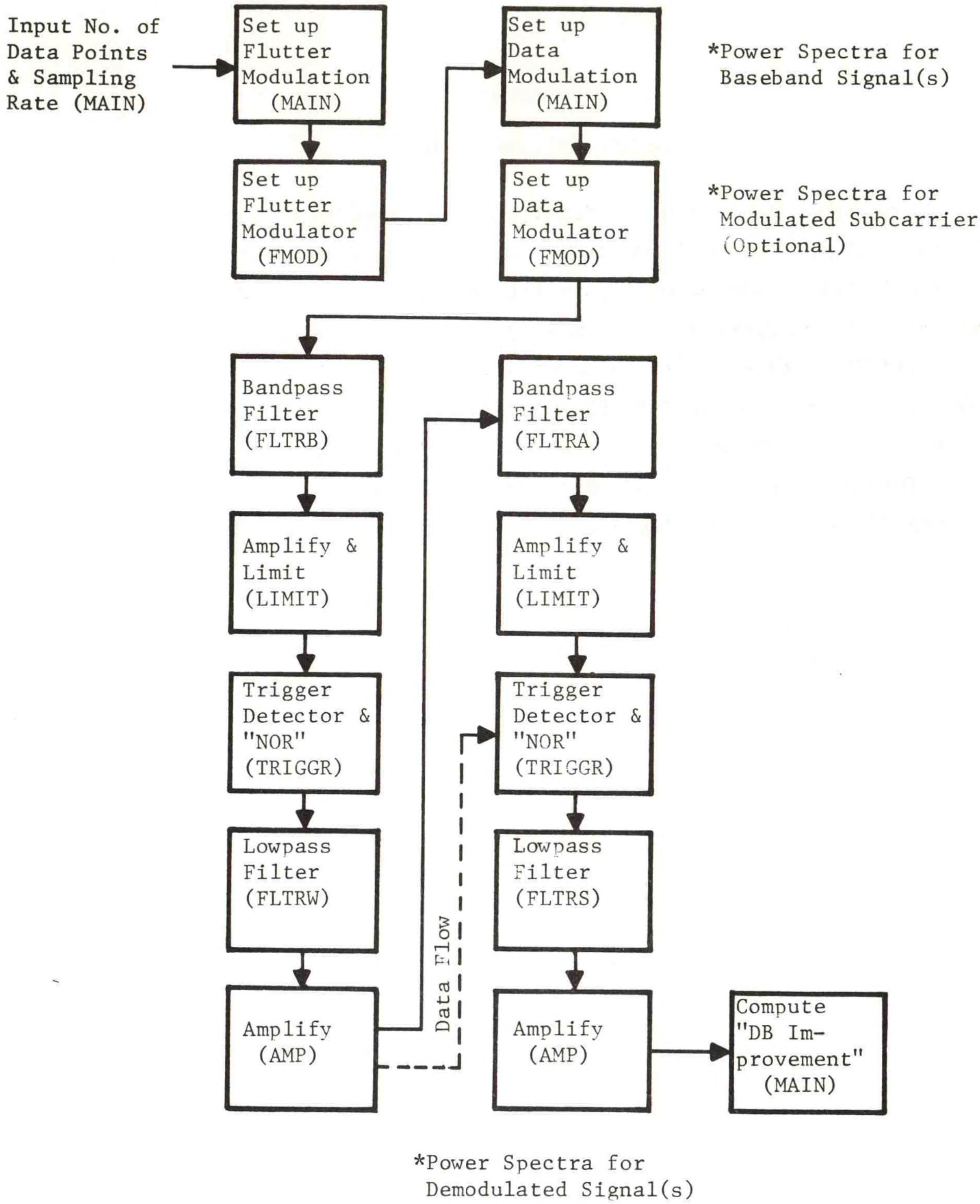


Figure A-1. Block Diagram of Model Implementation.

APPENDIX B

MODEL INPUT DESCRIPTION

This appendix presents the configuration of the data cards used with the model. A separate data deck must be supplied for each pass through the algorithm. The data deck must be complete within itself, and no data cards are common to multiple passes. With proper care this provides maximum program flexibility in terms of conducting analyses. For the first two passes the input bandpass filters are bypassed, and cards No. 3 and 4 as well as 22 and 23 must be removed. The parentheses following the card number indicate the calling routine. The cards are numbered sequentially but with gaps in the numbering system due to data deck modification during development. The deck was not renumbered in order to avoid confusion over the span of the study. The numbers to the left of the description indicate card column.

Data Deck Configuration

Flutter Channel

Card Number 1 (MAIN)

1-5 Number of data points to be analyzed (Integer Format)

6-25 Period of time between data points in seconds ("F" Format)

Card Number 2 (MAIN) - Integer Format - Diagnostic

1 Enter 1 if output is to be written on paper after program has executed (the reference and data channel output signals)

Card Number 3 (FLTRB) - Integer Format - Diagnostic

1 Enter 1 if array is to be written as it appears initially in the program.

2 Enter 1 if array is to be written after it is rearranged suitably for use by the Fourier transform subroutine.

3 Enter 1 if the Fourier spectrum is to be printed.

4 Enter 1 if the filtered Fourier spectrum is to be printed.

5 Enter 1 if the inverse Fourier transform output is to be printed.

6 Enter 1 if the array is to be printed after it is rearranged to be used by other subroutines.

Card Number 4 (FLTRB) - 3F10.3

1-10 Number of poles in filter

11-20 Maximum deviation frequency

21-30 Subcarrier frequency

Card Number 7 (LIMIT) - Integer Format - Diagnostic

1 Enter 1 if it is desired to write the data out after the flutter channel has been limited.

Card Number 8 (LIMIT) - "F" Format

1-10 The gain of the limiter stage before the flutter signal is limited. Must include decimal.

11-20 The gain of the limiter stage after the limiter must include decimal.

21-30 The amount of delay inherent in the limiter. Must include decimal.

31-40 The absolute value at which the signal is clipped. Must include decimal.

Card Number 9 (TRIGGR) - Integer Format - Diagnostic

1	Enter 1 if it is desired to output the data after the flutter channel has been examined for positive going and negative going zero crossings.
2	Enter 1 if it is desired to output the data which consists of the outputs of the A and B detectors.
3	Enter 1 if it is desired to output the data after the "NOR" circuit has "NORED" the outputs of the two detectors.

Card Number 10 (TRIGGR) - "F" Format

1-10	The triggering level of the Schmitt Trigger for increasing voltages.
11-20	The triggering level of the Schmitt Trigger for decreasing voltages.
21-30	The delay associated with the Schmitt Trigger circuit.

Card Number 11 (TRIGGR) - "F" Format

1-10	Enter 0.0
11-20	Pulse width of detector A in seconds.
21-30	The delay in seconds associated with the A detector.
31-40	The amplitude of the detector A pulse output.
41-50	The amplitude of the detector A output when no pulse is being output.

Card Number 12 (TRIGGR) - "F" Format

1-10	Enter 0.0.
11-20	Pulse width of detector B in seconds.
21-30	The delay in seconds associated with the detector.
31-40	The amplitude of the detector B pulse output.
41-50	The amplitude of the detector B output when no pulse is being output.

Card Number 13 (TRIGGR) - "F" Format

1-10	The amplitude of the "NOR" circuit output voltage when it is in the high state.
11-20	The amplitude of the "NOR" circuit output voltage when it is in the low state.
21-30	The delay in seconds associated with the "NOR" circuit.
31-40	The input level which differentiates the low level from the high level.

Card Number 14 (FLTRW) - Integer Format - Diagnostic

1	Enter 1 if flutter array is to be output as it appears upon entry to the filter subroutine.
2	Enter 1 if flutter array is to be output after it is rearranged (put in complex form) suitably for use by the Fourier transform subroutine.

3 Enter 1 if Fourier spectrum of input data is desired.

4 Enter 1 if the spectrum of the filtered data is desired.

5 Enter 1 if the inverse Fourier transform output is desired.

6 Enter 1 if the flutter array is to be output after the filtering operations have ended.

Card Number 15 (FLTRW) - Integer Format

1-3 The degree of the transfer polynomial numerator for the output flutter channel filter.

4-6 The degree of the transfer polynomial denominator for the output filter in the flutter channel.

Card Number 16 (FLTRW) - 5E16.8

1-80 The coefficients of the numerator polynomial of the output flutter channel filter, power increasing except for the constant term which is last. Additional cards may be used for high order polynomials.

Card Number 17 (FLTRW) - 5E16.8

1-80 The coefficients of the denominator polynomial of the output flutter channel filter, power increasing except for the constant term which is last. Additional cards may be used for high order polynomials.

Card Number 18 (AMPL) - Integer Format - Diagnostic

1 Enter 1 if the arrays are to be output after the flutter array is amplified.

Card Number 19 (AMPL) - "F" Format

1-10 Enter the gain of the amplification stage.

Card Number 20 (DELAY) - Integer Format - Diagnostic

1 Enter 1 if the arrays are to be output after the flutter array is delayed (advanced, see Card No. 21).

Card Number 21 (DELAY) - "F" Format

1-10 The time in seconds by which the flutter channel signal is advanced with respect to the signal (effectively a positive error in main delay setting).

Card Number 21A (DELAY) - Integer Format - Diagnostic

1 Enter 1 if the arrays are to be output after the signal array is delayed (advanced, see Card No. 21B).

Card Number 21B (DELAY) - "F" Format

1-10 The time in seconds by which the signal channel is advanced with respect to the flutter (effectively a negative error in main delay setting).

Data Channel

Card Number 22 (FLTRA) - Integer Format - Diagnostic

1 Enter 1 if signal and "A" array are to be written as they appear initially in the program. Useful only for debugging program.

2 Enter 1 if signal array is to be written after it is rearranged suitably for use by the Fourier transform subroutine.

3 Enter 1 if the Fourier spectrum of the signal data is to be printed.

4 Enter 1 if the filtered Fourier spectrum is to be printed.

5 Enter 1 if the inverse Fourier transform output is to be printed.

6 Enter 1 if the array is to be printed after it is rearranged to be used by other subroutines.

Card Number 23 (FLTRA) - 3F10.3

1-10 Number of poles in filter.

11-20 Maximum deviation frequency.

21-30 Subcarrier frequency.

Card Number 26 (LIMIT) - Integer Format - Diagnostic

1 Enter 1 if it is desired to write the signal data out after it has been limited.

Card Number 27 (LIMIT) - "F" Format

1-10 The gain of the limiter stage before the signal channel is limited. Must include decimal.

11-20 The gain of the limiter stage after the limiter. Must include decimal.

21-30 The amount of delay inherent in the limiter. Must include decimal.

31-40 The absolute value at which the signal is clipped. Must include decimal.

Card Number 28 (TRIGGR) - Integer Format - Diagnostic

1 Enter 1 if it is desired to output the data after the signal channel has been examined for positive going and negative going zero crossings.

2 Enter 1 if it is desired to output the data which consists of the outputs of the A and B detectors.

3 Enter 1 if it is desired to output the data after the "NOR" circuit has "NORED" the outputs of the two detectors.

Card Number 29 (TRIGGR) - "F" Format

1-10 The triggering level of the Schmitt Trigger for increasing voltages.

11-20 The triggering level of the Schmitt Trigger for decreasing voltages.

21-30 The delay associated with the Schmitt Trigger circuit.

Card Number 30 (TRIGGR) - "F" Format

1-10 The ratio of the detector A pulse width output to the magnitude of the wow and flutter channel, i.e., the "coefficient" determined by

$$\text{detector pulse width} = \text{"coefficient"} \times \text{flutter magnitude} + \text{nominal detector pulse width.}$$

11-20 The nominal pulse width detector A (see expression above).

21-30 The delay in seconds associated with the A detector.

31-40 The amplitude of the detector A pulse output.

41-50 The amplitude of the detector A output when no pulse is being output.

Card Number 31 (TRIGGR) - "F" Format

1-10 Ratio of the detector B pulse width to the magnitude of the wow and flutter signal. See corresponding entry in card number 30.

11-20 The nominal pulse width of detector B. See corresponding entry in card number 30.

21-30 The delay in seconds associated with the B detector.

31-40 The amplitude of the detector B pulse output.

41-50 The amplitude of the detector B output when no pulse is being output.

Card Number 32 (TRIGGR) - "F" Format

1-10 The amplitude of the "NOR" circuit output voltage when it is in the high state.

11-20 The amplitude of the "NOR" circuit output voltage when it is in the low state.

21-30 The delay in seconds associated with the "NOR" circuit.

31-40 The input level which differentiates the low level from the high level.

Card Number 33 (FLTRS) - Integer Format - Diagnostic

1 Enter 1 if signal array is to be output as it appears upon entry to the output filter subroutine. Useful for debugging only.

2 Enter 1 if flutter array is to be output after it is rearranged (put in complex form) suitably for use by the Fourier transform subroutine.

3 Enter 1 if the spectrum of data is desired.

4 Enter 1 if the spectrum of the filtered data is desired.

5 Enter 1 if the Fourier transform output is desired.

6 Enter 1 if the flutter array is to be output after the filtering operations have ended.

Card Number 34 (FLTRS) - Integer Format

1-3 The degree of the transfer polynomial numerator for the output signal channel filter.

4-6 The degree of the transfer polynomial denominator for the output filter in the signal channel.

Card Number 35 (FLTRS) - 5E16.8

1-80 The coefficients of the numerator polynomial of the output signal channel filter, power increasing except for the constant term which is last. Additional cards may be used for high order polynomials.

Card Number 36 (FLTRS) - 5E16.8

1-80 The coefficients of the denominator polynomial of the output signal channel filter, power increasing except for the constant term which is last. Additional cards may be used for high order polynomials.

Card Number 37 (AMPL) - Integer Format - Diagnostic

1 Enter 1 if the arrays are to be output after the signal array is amplified.

Card Number 38 (AMPL) - "F" Format

1-10 Enter the gain of the amplification stage.

APPENDIX C

STANDARD OUTPUT FORMAT

This appendix indicates the data output format from the model. Also available is a description of the input parameters (which has not been included here). As described in Appendix B, intermediate diagnostic output is also available as an option.

The basic parameter of interest in model performance is "dB Improvement" across the spectral band. This is defined as

$$20 \log_{10} \frac{A_n\text{(without compensation)}}{A_n\text{(with compensation)}}$$

where A_n is the amplitude of the "n"th harmonic in a spectral analysis of the data channel output. The spectrum for no compensation is generated during the first pass through the algorithm and stored. The above computation is provided for each subsequent pass desired.

Figure C-1 indicates the output format. The first column represents the wavenumber with wavenumber 1 being the "DC" term. The spectrum is computed using the Fast-Fourier transform method and incorporates 4096 data points at a sampling rate of 10 μ sec which corresponds to a Δf of 24.4 Hz at the DDHS. The second column represents the frequency in hertz at the spacecraft during record (i.e., real time), while the third column represents the frequency in hertz as seen at the DDHS during 4:1 playback from the CDA. (Recall that there is a 20.833:1 recorder speed increase during satellite readout.) The fourth column represents the spectral amplitude in counts for the no-compensation case and the fifth column the spectral amplitude in counts for the compensated case. Note that the flutter tones referred to in Section 5.0 appear at wavenumbers 5, 8, 18, and 33. The sixth column represents the "dB Improvement" mentioned above. In order to provide a feeling for what the magnitude of these numbers means, a 20 "dB Improvement" would indicate that the flutter harmonic has been reduced to one-tenth its original amplitude. Correspondingly, a 40 "dB Improvement" would indicate a reduction of the flutter harmonic amplitude to 1% of its original amplitude. Negative values indicate flutter enhancement. It should be noted that for the case presented,

	S/C	4/1	NO-Z	Z-COR	DB-IMP		S/C	4/1	NO-Z	Z-COR	DB-IMP
1	0.0	0.0	0.000	0.038	-146.753	61	281.3	1464.8	0.023	0.013	4.818
2	4.7	24.4	0.000	0.039	-69.889	62	285.9	1489.3	0.008	0.019	-7.807
3	9.4	48.8	0.001	0.040	-37.930	63	290.6	1513.7	0.007	0.011	-4.375
4	14.1	73.2	0.000	0.031	-38.040	64	295.3	1538.1	0.005	0.011	-7.124
5	18.8	97.7	0.485	0.015	30.176	65	300.0	1562.5	0.014	0.025	-5.188
6	23.4	122.1	0.001	0.041	-33.951	66	304.7	1586.9	0.016	0.026	-4.098
7	28.1	146.5	0.001	0.040	-30.674	67	309.4	1611.3	0.022	0.013	4.506
8	32.8	170.9	0.121	0.030	12.032	68	314.1	1635.7	0.007	0.019	-8.382
9	37.5	195.3	0.001	0.029	-33.030	69	318.8	1660.2	0.006	0.015	-7.157
10	42.2	219.7	0.002	0.035	-25.375	70	323.4	1684.6	0.014	0.018	-2.003
11	46.9	244.1	0.002	0.030	-25.342	71	328.1	1709.0	0.000	0.014	-33.305
12	51.6	268.6	0.003	0.029	-19.199	72	332.8	1733.4	0.027	0.014	5.356
13	56.3	293.0	0.001	0.038	-30.521	73	337.5	1757.8	0.006	0.007	-2.070
14	60.9	317.4	0.000	0.008	-26.555	74	342.2	1782.2	0.010	0.021	-6.614
15	65.6	341.8	0.001	0.034	-27.743	75	346.9	1806.6	0.009	0.012	-1.944
16	70.3	366.2	0.004	0.005	-2.632	76	351.6	1831.1	0.017	0.008	6.065
17	75.0	390.6	0.003	0.035	-21.237	77	356.3	1855.5	0.003	0.016	-14.304
18	79.7	415.0	0.788	0.020	31.856	78	360.9	1879.9	0.018	0.011	3.992
19	84.4	439.5	0.001	0.026	-33.242	79	365.6	1904.3	0.004	0.005	-1.152
20	89.1	463.9	0.003	0.045	-22.528	80	370.3	1928.7	0.009	0.024	-8.595
21	93.8	488.3	0.004	0.035	-18.316	81	375.0	1953.1	0.025	0.030	-1.627
22	98.4	512.7	0.001	0.009	-15.502	82	379.7	1977.5	0.008	0.014	-5.176
23	103.1	537.1	0.011	0.042	-11.884	83	384.4	2002.0	0.011	0.013	-1.712
24	107.8	561.5	0.002	0.035	-25.449	84	389.1	2026.4	0.009	0.017	-5.269
25	112.5	585.9	0.003	0.026	-18.051	85	393.8	2050.8	0.020	0.031	-3.889
26	117.2	610.4	0.009	0.044	-13.846	86	398.4	2075.2	0.023	0.021	0.434
27	121.9	634.8	0.008	0.042	-14.270	87	403.1	2099.6	0.016	0.021	-2.079
28	126.6	659.2	0.011	0.050	-12.825	88	407.8	2124.0	0.006	0.009	-2.894
29	131.3	683.6	0.005	0.019	-12.257	89	412.5	2148.4	0.011	0.017	-4.034
30	135.9	708.0	0.003	0.038	-21.526	90	417.2	2172.9	0.023	0.025	-0.853
31	140.6	732.4	0.006	0.018	-9.953	91	421.9	2197.3	0.012	0.016	-2.987
32	145.3	756.8	0.009	0.029	-9.967	92	426.6	2221.7	0.018	0.023	-2.077
33	150.0	781.3	0.387	0.075	14.289	93	431.3	2246.1	0.010	0.020	-5.841
34	154.7	805.7	0.004	0.026	-15.722	94	435.9	2270.5	0.011	0.009	2.093
35	159.4	830.1	0.012	0.011	1.231	95	440.6	2294.9	0.008	0.013	-3.735
36	164.1	854.5	0.008	0.017	-6.474	96	445.3	2319.3	0.026	0.021	1.699
37	168.8	878.9	0.012	0.006	6.387	97	450.0	2343.8	0.010	0.025	-7.557
38	173.4	903.3	0.005	0.022	-13.384	98	454.7	2368.2	0.031	0.029	0.710
39	178.1	927.7	0.015	0.023	-3.551	99	459.4	2392.6	0.010	0.008	1.980
40	182.8	952.1	0.003	0.018	-15.054	100	464.1	2417.0	0.016	0.015	0.642
41	187.5	976.6	0.017	0.015	0.929	101	468.8	2441.4	0.028	0.031	-1.018
42	192.2	1001.0	0.006	0.030	-14.072	102	473.4	2465.8	0.016	0.014	0.922
43	196.9	1025.4	0.016	0.015	0.996	103	478.1	2490.2	0.032	0.035	-0.873
44	201.6	1049.8	0.006	0.039	-16.687	104	482.8	2514.6	0.014	0.015	-0.747
45	206.3	1074.2	0.013	0.039	-9.367	105	487.5	2539.1	0.001	0.002	-7.928
46	210.9	1098.6	0.004	0.013	-9.961	106	492.2	2563.5	0.015	0.019	-1.963
47	215.6	1123.0	0.007	0.020	-8.919	107	496.9	2587.9	0.015	0.008	5.643
48	220.3	1147.5	0.008	0.032	-12.600	108	501.6	2612.3	0.002	0.011	-16.666
49	225.0	1171.9	0.011	0.025	-6.957	109	506.3	2636.7	0.014	0.015	-0.819
50	229.7	1196.3	0.010	0.010	0.748	110	510.9	2661.1	0.034	0.034	-0.074
51	234.4	1220.7	0.008	0.026	-9.951	111	515.6	2685.5	0.007	0.012	-4.105
52	239.1	1245.1	0.006	0.021	-10.841	112	520.3	2710.0	0.015	0.024	-4.027
53	243.8	1269.5	0.005	0.028	-15.906	113	525.0	2734.4	0.010	0.017	-4.500
54	248.4	1293.9	0.018	0.028	-3.824	114	529.7	2758.8	0.027	0.029	-0.441
55	253.1	1318.4	0.006	0.023	-12.314	115	534.4	2783.2	0.015	0.013	1.143
56	257.8	1342.8	0.029	0.045	-3.735	116	539.1	2807.6	0.005	0.003	4.263
57	262.5	1367.2	0.012	0.028	-7.704	117	543.8	2832.0	0.017	0.015	1.345
58	267.2	1391.6	0.013	0.016	-1.809	118	548.4	2856.4	0.022	0.024	-0.629
59	271.9	1416.0	0.003	0.027	-20.195	119	553.1	2880.9	0.011	0.007	3.671
60	276.6	1440.4	0.017	0.011	3.684	120	557.8	2905.3	0.019	0.022	-1.064

Figure C-1. Sample Model Output Format.

the reference discriminator output gain was slightly misaligned resulting in a small "DC" component to be introduced into the data channel during the compensation process. For this reason the -146.753 "dB Improvement" of wavenumber 1 should be ignored. Columns 7 through 12 represent a continuation of the data presented in 1 through 6. Since 4096 samples were used in the spectral analysis, 2048 harmonics were generated with a Nyquist frequency of 50,000 Hz. This is well beyond the data channel bandwidth and only frequencies out to approximately 3000 Hz (wavenumber 120) at the DDHS were output. This adequately covers the band of interest for analysis of the z-axis rack operation.

APPENDIX D

SAMPLE PROGRAM LISTING

This appendix includes a sample listing of the digital algorithm used during this study. The model was programmed in Fortran IV - G level for an IBM Systems 370/65 located at the Triangle Universities Computation Center in the Research Triangle Park, N. C. The system log and source module map have been retained in the listing for completeness.

```
123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
1                                                                                                                      1
1 *****     *     *   ******    *****       ****    ******    *****               *     *             *     *          1
1 *    *    *     *   *         *    *     *    *   *         *    *              *     *             *     *          1
1 *    *    *     *   *         *     *   *         *         *    *              *     *             *     *          1
1 *****     *     *   *****     *     *   *         *****     *****               *  *  *             *******          1
1 *   *     *     *   *         *     *   *   ***   *         *   *               * * * *             *     *          1
1 *    *     *   *    *         *    *     *    *   *         *    *              **   **             *     *          1
1 *     *     ***     *******   *****       ****    *******   *     *             *     *             *     *          1
1                                                                                                                      1
1                                                                                                                      1
1 *     *     ***     *****     ******    *                                 *****       *******     ***                1
1 **   **    *   *    *    *    *         *                                 *    *         *         *                 1
1 * * * *   *     *   *     *   *         *                                 *    *         *         *                 1
1 *  *  *   *     *   *     *   *****     *                                 *****          *         *                 1
1 *     *   *     *   *     *   *         *                                 *   *          *         *                 1
1 *     *    *   *    *    *    *         *                                 *    *         *         *                 1
1 *     *     ***     *****     *******   *******                           *     *        *        ***                1
1                                                                                                                      1
123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
```

```
PROGRAMMER=RUEDGER,W H          ACCOUNT=RTI.C43.PO1217

      JOB TURN-AROUND                   JOB PARAMETERS                 TIME USED INCLUDES          MISC. JOB VALUES

             DATE         TIME              SPECIFIED    USED                     NO.   EQUIVALENT
ENTERED   11/27/73   17:13:53.9    TIME       3:00.0    2:32.1   CPU                      1:49.3   LINES    IN       1722
EXECUTED  11/28/73    0:31:51.0    PAGES        150       100    UR EXCPS         6584    0:26.4   LINES    OUT      4847
RETURNED  11/28/73    1:24:13.9    CARDS          0         0    TAPE EXCPS          0    0:00.0   CORE TIME        57045 K-SEC
JOB ENTERED ON TUESDAY             PLOTS          0         0    DISK EXCPS        819    0:16.4   PRIORITY           00
                                                                 ARM SEEKS         166    0:00.0   APPROX. COST     $30.51
                                                                 ARM MOTION       1878    0:00.0

                      J O B    C O N S O L E    L O G

N  0.31.43 JOB 779 -- MODEL     -- BEGINNING EXEC - INIT 4 - CLASS A
L* 0.31.51 JOB 779 ++++++S=000,T=03,J=MODEL      0:31:51 RUEDGER,W H
 * 0.31.51 JOB 779 IEF281I 263    NOW OFFLINE
N  0.39.14 JOB 779 END EXECUTION.
```

```
//MODEL JOB RTI.C43.P01217,'RUEDGER,W H',MSGLEVEL=1,REGION=375K,        1
//            T=3,P=150,PRTY=0
// EXEC FTGCLG
      ***        FORTRAN "G" LEVEL COMPILE, LINKEDIT, AND EXECUTE              00000010
      XXC          EXEC PGM=FORTRANG                                            00000020
      XXSYSLIN     DD DSN=&&OBJSET,SPACE=(80,(1000,500)),UNIT=SYSDA,            00000030
      XX              DISP=(MOD,PASS),DCB=(RECFM=FB,LRECL=80,BLKSIZE=400)       00000040
      XXSYSPRINT  DD SYSOUT=A                                                   00000050
      XXSYSPUNCH  DD DSN=NULLFILE,DISP=SHR                                      00000060
//C.SYSIN DD *
         IEF236I ALLOC. FOR MODEL    C
         IEF237I 136   ALLOCATED TO SYSLIN
         IEF237I 56B   ALLOCATED TO SYSPRINT
         IEF237I 503   ALLOCATED TO SYSIN
IEF142I - STEP WAS EXECUTED - COND CODE 0000
         IEF285I   SYS73331.T234925.RV000.MODEL.OBJSET              PASSED
         IEF285I   VOL SER NOS= SPARE9.
C       CORE=376K TIME---0:26.1  CPU---0:09.6  I/O---0:16.5     RC-----0
C       USED=098K UR---3056 DISK---214 SEEK----18 MO---41C TAPE-----0

      XXL          EXEC PGM=LINKEDIT,PARM='MAP,LIST',COND=(5,LT,C)              00000070
      XXSYSLIB     DD DISP=SHR,DSNAME=SYS1.FORTLIB                              00000080
      XX           DD DISP=SHR,DSNAME=SYS1.SUBLIB                               00000090
      XXSYSLIN     DD DSN=&&OBJSET,DISP=(MOD,DELETE),DCB=RECFM=FB               00000100
      XX           DD DDNAME=SYSIN                                              00000110
      XXALLCCATE  DD DSN=&&LOADSET,DISP=(MOD,PASS),SPACE=(1024,(300,50,1)),     00000120
      XX              UNIT=DISK                                                 00000130
      XXSYSLMOD    DD DSN=&&LOADSET(USERPROG),DISP=(OLD,PASS),                  00000140
      XX              VOLUME=REF=*.ALLOCATE                                     00000150
      XXSYSUT1     DD DSN=&&UT1,SPACE=(1024,(480,240)),UNIT=SYSDA               00000160
      XXSYSPRINT  DD SYSOUT=A                                                   00000170
         IEF236I ALLOC. FOR MODEL    L
         IEF237I 131   ALLOCATED TO SYSLIB
         IEF237I 171   ALLOCATED TO
         IEF237I 136   ALLOCATED TO SYSLIN
         IEF237I 136   ALLOCATED TO ALLOCATE
         IEF237I 136   ALLOCATED TO SYSLMOD
         IEF237I 136   ALLCCATED TO SYSUT1
         IEF237I 562   ALLOCATED TO SYSPRINT
IEF142I - STEP WAS EXECUTED - COND CODE 0000
         IEF285I   SYS1.FORTLIB                                     KEPT
         IEF285I   VOL SER NOS= HYPERD.
         IEF285I   SYS1.SUBLIB                                      KEPT
         IEF285I   VOL SER NOS= TUC111.
         IEF285I   SYS73331.T234925.RV000.MODEL.OBJSET              DELETED
         IEF285I   VOL SER NOS= SPARE9.
         IEF285I   SYS73331.T234925.RV000.MODEL.LOADSET             PASSED
         IEF285I   VOL SER NOS= SPARE9.
         IEF285I   SYS73331.T234925.RV000.MODEL.LOADSET             PASSED
         IEF285I   VOL SER NOS= SPARE9.
         IEF285I   SYS73331.T234925.RV000.MODEL.UT1                 DELETED
         IEF285I   VOL SER NOS= SPARE9.
L       CORE=376K TIME---0:11.8  CPU---0:01.0  I/O---0:10.9     RC-----0
L       USED=108K UR----71 DISK---530 SEEK---118 MO--1101 TAPE-----0

      XXG          EXEC PGM=*.L.SYSLMOD,COND=((5,LT,C),(5,LT,L))                00000180
      XXFT03F001  DD SYSOUT=A,DCB=(RECFM=FBA,BLKSIZE=133,LRECL=133)             00000190
      XXFT02F001  DD DDNAME=SYSPUNCH                                            00000200
      XXFT01F001  DD DDNAME=SYSIN                                               00000210
//G.SYSIN DD *
//
         IEF236I ALLOC. FOR MODEL    G
         IEF237I 136   ALLOCATED TO PGM=*.DD
```

```
          IEF237I 562   ALLOCATED TO FT03F001
          IEF237I 504   ALLOCATED TO FT01F001
IEF142I - STEP WAS EXECUTED - COND CODE 0000
          IEF285I   SYS73331.T234925.RV000.MODEL.LOADSET        PASSED
          IEF285I   VOL SER NOS= SPARE9.
G        CORE=376K TIME---1:53.5  CPU---1:38.8  I/O---0:14.7   RC-----0
G        USED=350K UR--3380 DISK----59 SEEK----19 MO---204 TAPE-----0

          IEF285I   SYS73331.T234925.RV000.MODEL.LOADSET        DELETED
          IEF285I   VOL SER NOS= SPARE9.
          IEF285I   SYS73331.T234925.RV000.MODEL.LOADSET        KEPT
          IEF285I   VOL SER NOS= SPARE9.
MODEL    TIME---2:31.6
```

```
          C***************  MAIN PROGRAM  *********************             ***
 0001           COMMON SIGNAL(4096),A(4096),WAF(4096),SPEC(120),DUMF,DUMD
 0002           DIMENSION FIRST(120)
 0003           DIMENSION INV(1024),S(1024)
 0004         1 FORMAT(1H1)
 0005        50 FORMAT(32E15.8)
 0006        51 FORMAT(1X,10F12.7)
          C*********  PROGRAM TO GENERATE TEST DATA  ***********
          C****
          C     MODEL CURRENTLY CONFIGURED FOR ITOS-D AT FOUR TO ONE
          C     PLAYBACK
          C    LET SAMPLE TIME BE 10 MICROSECONDS
          C****
 0007           DO 5000 KKK=1,11
 0008           PI = 3.1415926
 0009           TPI=2.0*PI
 0010           READ(1,48) NUMBER,PERIOD
 0011        48 FORMAT(I5,F20.4)
 0012           READ(1,49) IWRIT
 0013        49 FORMAT(40I1)
 0014           NA = NUMBER / 32
 0015           NE = 0
          C     DO 60 ND = 1, NA
          C     NB = NE + 1
          C     NE = NE + 32
          C  60 READ(8,50)  (SIGNAL(NI), NI = NB,NE)
          C     NE = 0
          C     DO 61 ND = 1,NA
          C     NB = NE + 1
          C     NE = NE + 32
          C  61 READ(8,50)  (WAF(NI),NI = NB,NE)
 0016           IX = NUMBER
 0017           I = 0
 0018        62 IX = IX / 2
 0019           I = I + 1
 0020           IF(IX.LE.1) GO TC 63
 0021           GO TO 62
 0022        63 NNUMA = IX
 0023           NNUMB=NUMBER/4
 0024           NUMBER=4096
 0025           NNUMA=12
 0026           NNUMB=1024
 0027           NEIGHT=NUMBER/8
          C     ZERO ALL DATA ARRAYS
 0028           DO 1000 KZ=1,NUMBER
 0029           SIGNAL(KZ)=0.0
 0030           A(KZ)=0.0
 0031           WAF(KZ)=0.0
 0032      1000 CONTINUE
          C
          C     SET UP FLUTTER MODULATION
 0033           FMAX=0.0
 0034           FMIN=0.0
 0035           IX=7
 0036           DO 10 JA=1,NUMBER
 0037           CONSA=(FLOAT(JA)/512.0)*PI
```

```
FORTRAN IV G LEVEL  21                  MAIN              DATE = 73332           00/31/53                PAGE 0002

 0038              CONSA=AMOD(CONSA,TPI)
 0039              CONSB=(FLOAT(JA)/292.571)*PI
 0040              CONSB=AMOD(CONSB,TPI)
 0041              CONSC=(FLOAT(JA)/120.471)*PI
 0042              CONSC=AMOD(CONSC,TPI)
 0043              CONSD=(FLOAT(JA)/64.0)*PI
 0044              CONSD=AMOD(CONSD,TPI)
 0045              A(JA)=0.4*COS(CONSA)+0.1*COS(CONSB)+0.65*COS(CONSC)
                  1+0.32*COS(CONSD)
 0046              CALL RANDU(IX,IY,FLY)
 0047              ARND=2.0*(FLY-0.5)
             C**** REMOVES RANDOM NOISE FIELD
 0048              ARND=0.0
             C****
 0049              A(JA)=A(JA)+ARND
 0050              IX=IY
 0051              FMAX=AMAX1(FMAX,A(JA))
 0052              FMIN=AMIN1(FMIN,A(JA))
 0053           10 CONTINUE
             C**** GENERATE APPENDAGE FOR TRIGGER
 0054              NP1=NUMBER+1
 0055              CALL RANDU(IX,IY,FLY)
 0056              ARND=2.0*(FLY-0.5)
             C**** REMOVES RANDOM NOISE FIELD
 0057              ARND=0.0
             C****
 0058              CONSA=(FLOAT(NP1)/512.0)*PI
 0059              CONSA=AMOD(CONSA,TPI)
 0060              CONSB=(FLOAT(NP1)/292.571)*PI
 0061              CONSB=AMOD(CONSB,TPI)
 0062              CONSC=(FLOAT(NP1)/120.471)*PI
 0063              CONSC=AMOD(CONSC,TPI)
 0064              CONSD=(FLOAT(NP1)/64.0)*PI
 0065              CONSD=AMOD(CONSD,TPI)
 0066              DUMF=0.4*COS(CONSA)+0.1*COS(CONSB)+0.65*COS(CONSC)
                  1+0.32*COS(CONSD)
 0067              DUMF=DUMF+ARND
 0068              IF(KKK.GT.1) GO TO 400
 0069              WRITE(3,110)
 0070          110 FORMAT('1AMPLITUDE SPECTRA FOR FLUTTER INPUT')
 0071              CALL PSD(A,NUMBER,NNUMA)
 0072              WRITE(3,210) FMAX
 0073          210 FORMAT(' MAXIMUM FLUTTER AMPLITUDE IS',F10.3)
 0074              WRITE(3,220) FMIN
 0075          220 FORMAT(' MINIMUM FLUTTER AMPLITUDE IS',F10.3)
 0076          400 CONTINUE
             C     SET UP FLUTTER MODULATOR
 0077              FO=6250.0
 0078              DEVIA=937.5
 0079              CALL FMOD(FO,DEVIA,1,NUMBER)
             C     SET UP DATA CHANNEI MODULATION
 0080              SMAX=0.0
 0081              SMIN=0.0
 0082              DO 30 JA=1,NUMBER
 0083              CONSA=(FLOAT(JA)/512.0)*PI
 0084              CONSA=AMOD(CONSA,TPI)
```

```
FORTRAN IV G LEVEL  21                MAIN              DATE = 73332         00/31/53              PAGE 0003

  0085                 CONSB=(FLOAT(JA)/292.571)*PI
  0086                 CONSB=AMOD(CONSB,TPI)
  0087                 CONSC=(FLOAT(JA)/120.471)*PI
  0088                 CONSC=AMOD(CONSC,TPI)
  0089                 CONSD=(FLOAT(JA)/64.0)*PI
  0090                 CONSD=AMOD(CONSD,TPI)
  0091                 CONSE=(FLOAT(JA)/256.0)*PI
  0092                 CONSE=AMOD(CONSE,TPI)
  0093                 CONSF=(FLOAT(JA)/49.951)*PI
  0094                 CONSF=AMOD(CONSF,TPI)
  0095                 A(JA)=0.4*COS(CONSA)+0.1*COS(CONSB)+0.65*COS(CONSC)
                      1+0.32*COS(CONSD)
             C        2+0.75*COS(CONSE)+0.25*COS(CONSF)
  0096                 SMAX=AMAX1(SMAX,A(JA))
  0097                 SMIN=AMIN1(SMIN,A(JA))
  0098              30 CONTINUE
             C**** GENERATE APPENDAGE FOR TRIGGER
  0099                 CONSA=(FLOAT(NP1)/512.0)*PI
  0100                 CONSA=AMOD(CONSA,TPI)
  0101                 CONSB=(FLOAT(NP1)/292.571)*PI
  0102                 CONSB=AMOD(CONSB,TPI)
  0103                 CONSC=(FLOAT(NP1)/120.471)*PI
  0104                 CONSC=AMOD(CONSC,TPI)
  0105                 CONSD=(FLOAT(NP1)/64.0)*PI
  0106                 CONSD=AMOD(CONSD,TPI)
  0107                 DUMD=0.4*COS(CONSA)+0.1*COS(CONSB)+0.65*COS(CONSC)
                      1+0.32*COS(CONSD)
             C****
  0108                 IF(KKK.GT.1) GO TO 500
  0109                 WRITE(3,120)
  0110             120 FORMAT('1AMPLITUDE SPECTRA FOR SIGNAL INPUT')
  0111                 CALL PSD(A,NUMBER,NNUMA)
  0112                 WRITE(3,230) SMAX
  0113             230 FORMAT(' MAXIMUM SIGNAL AMPLITUDE IS',F10.3)
  0114                 WRITE(3,240) SMIN
  0115             240 FORMAT(' MINIMUM SIGNAL AMPLITUDE IS',F10.3)
  0116             500 CONTINUE
             C         SET UP DATA MODULATOR
  0117                 FO=25000.0
  0118                 DEVIA=3375.0
  0119                 CALL FMOD(FO,DEVIA,0,NUMBER)
  0120              12 CONTINUE
             C         IF(KKK.GT.1) GO TO 525
             C         WRITE(3,125)
             C     125 FORMAT('1AMPLITUDE SPECTRA FOR MODULATED CARRIER')
             C         CALL PSD(SIGNAL,NUMBER,NNUMA)
             C     525 CONTINUE
             C         CALL WRITE(NEIGHT)
  0121                 IF(KKK.LE.2) GO TO 535
  0122                 CALL FLTRB(PERIOD,NUMBER,NNUMA,NNUMB,INV,S)
  0123             535 CONTINUE
  0124                 WRITE(3,1)
  0125                 CALL LIMIT(PERIOD,1,NUMBER)
  0126                 CALL TRIGGR(PERIOD,1,NUMBER)
  0127                 CALL FLTRW(PERIOD,NUMBER,NNUMA,NNUMB,INV,S)
  0128                 CALL AMPL(1,NUMBER)
```

```
FORTRAN IV G LEVEL  21                MAIN               DATE = 73332          00/31/53                 PAGE 0004

 0129              WRITE(3,130)
 0130          130 FORMAT('1AMPLITUDE SPECTRA FOR FLUTTER OUTPUT')
 0131              CALL PSD(WAF,NUMBER,NNUMA)
 0132              WRITE(3,1)
 0133              IF(KKK.GT.1) GO TO 600
 0134              DO 550 I=1,NUMBER
 0135              WAF(I)=0.0
 0136          550 CONTINUE
 0137          600 CONTINUE
 0138              CALL DELAYP(PERIOD,0,NUMBER)
 0139              CALL DELAYP(PERIOD,1,NUMBER)
 0140              IF(KKK.LE.2) GO TO 635
 0141              CALL FLTRA(PERIOD,NUMBER,NNUMA,NNUMB,INV,S)
 0142          635 CONTINUE
 0143              CALL LIMIT(PERIOD,0,NUMBER)
 0144              CALL TRIGGR(PERIOD,0,NUMBER)
 0145              CALL FLTRS(PERIOD,NUMBER,NNUMA,NNUMB,INV,S)
 0146              CALL AMPL(0,NUMBER)
 0147              IF(IWRIT.EQ.0) GO TO 66
 0148              NE = 0
 0149              DO 65 ND = 1, NA
 0150              NB = NE + 1
 0151              NE = NE + 32
             C     WRITE(6,50)  (SIGNAI(II),II=NB,NE)
 0152           65 WRITE(3,51)  (SIGNAL(II),II=NB,NE)
 0153           66 CONTINUE
 0154              WRITE(3,140)
 0155          140 FORMAT('1AMPLITUDE SPECTRA FOR SIGNAL OUTPUT')
 0156              CALL PSD(SIGNAL,NUMBER,NNUMA)
 0157              IF(KKK.GT.1) GO TO 700
 0158              DO 750 I=1,120
 0159              FIRST(I)=SPEC(I)
 0160          750 CONTINUE
 0161              GO TO 800
 0162          700 CONTINUE
 0163              WRITE(3,720)
 0164          720 FORMAT('1          S/C       4/1        NO-Z        Z-COR      DB-IMP
                  1          S/C       4/1        NO-Z        Z-COR      DB-IMP')
 0165              DF=1/(NUMBER*PERIOD)
 0166              DO 850 I=1,60
 0167              FGD1=(I-1)*DF
 0168              FGD2=FGD1+60.0*DF
 0169              FSC1=FGD1*(4.0/20.833)
 0170              FSC2=FGD2*(4.0/20.833)
 0171              I2=I+60.0
 0172              DB1=20.*ALOG10(FIRST(I)/SPEC(I))
 0173              DB2=20.0*ALOG10(FIRST(I2)/SPEC(I2))
 0174              WRITE(3,730) I,FSC1,FGD1,FIRST(I),SPEC(I),DB1,
                  1I2,FSC2,FGD2,FIRST(I2),SPEC(I2),DB2
 0175          730 FORMAT(I4,2F8.1,3F10.3,10X,I4,2F8.1,3F10.3)
 0176          850 CONTINUE
 0177          800 CONTINUE
 0178         5000 CONTINUE
 0179              CALL EXIT
 0180              END
```

```
                         COMMON BLOCK /          / MAP SIZE    C1E8
SYMBOL    LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION
SIGNAL         0       A          4000       WAF        8000       SPEC       C000       DUMF       C1E0
DUMD        C1E4

                         SUBPROGRAMS CALLED
SYMBOL    LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION
IBCOM#       13C       RANDU       140       PSD         144       FMCD        148       FLTRB       14C
LIMIT        150       TRIGGR      154       FLTRW       158       AMPL        15C       DELAYP      160
FLTRA        164       FLTRS       168       EXIT        16C       COS         170       AMAX1       174
AMIN1        178       ALOG10      17C

                         SCALAR MAP
SYMBOL    LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION
KKK          3E0       PI          3E4       TPI         3E8       NUMBER      3EC       PERIOD      3F0
IWRIT        3F4       NA          3F8       NE          3FC       IX          400       I           404
NNUMA        408       NNUMB       40C       NEIGHT      410       KZ          414       FMAX        418
FMIN         41C       JA          420       CONSA       424       CONSB       428       CONSC       42C
CONSD        430       IY          434       FLY         438       ARND        43C       NP1         440
FO           444       DEVIA       448       SMAX        44C       SMIN        450       CONSE       454
CONSP        458       ND          45C       NB          460       II          464       DF          468
FGD1         46C       FGD2        470       FSC1        474       FSC2        478       I2          47C
DB1          480       DB2         484

                         ARRAY MAP
SYMBOL    LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION
FIRST        488       INV         668       S          1668

                         FORMAT STATEMENT MAP
SYMBOL    LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION
     1      2668           50     266D           51     2674           48     267D           49     2684
   110      268A          210     26B2          220     26D6          120     26FA          230     2721
   240      2744          130     2767          140     2790          720     27B8          730     282A

  *OPTICNS IN EFFECT*  ID,EBCDIC,SOURCE,NOLIST,DECK,LOAD,MAP
  *OPTIONS IN EFFECT*  NAME = MAIN    , LINECNT =       58
  *STATISTICS*    SOURCE STATEMENTS =       180,PROGRAM SIZE =     14296
  *STATISTICS*  NO DIAGNOSTICS GENERATED
```

```
FORTRAN IV G LEVEL  21                MAIN            DATE = 73332         00/31/53              PAGE 0001

                    C
                    C
                    C
      0001                SUBROUTINE FMOD(C,B,MODE,NUMBER)
      0002                COMMON SIGNAL(4096),Z(4096),WAF(4096),SPEC(120),DUMF,DUMD
      0003                REAL*8 A(4096),FO,DEVIA,SUM,TEMP,CONS
      0004                REAL*8 PI,TPI
      0005                REAL*8 DDUMF,DDUMD,NDP1
      0006                PI=3.141592653589793D0
      0007                TPI=2.D0*PI
      0008                FO=C
      0009                DEVIA=B
      0010                DDUMF=DUMF
      0011                DDUMD=DUMD
                    C     NP1=NUMBER+1
                    C     NDP1=NP1
                    C**** "FIX" FOR APPENDAGE
      0012                NDP1=NUMBER
                    C****
      0013                DO 60 J=1,NUMBER
      0014             60 A(J)=Z(J)
      0015                SUM=0.D0
      0016                DO 40 JE=1,NUMBER
      0017                SUM=SUM+A(JE)*0.00001D0
      0018                TEMP=SUM
                    C**** "FIX" TO LET UNMODULATED CARRIER HAVE ZERO CROSSING IN FIRST
                    C     ELEMENT OF ARRAY
      0019                JM=JE-1
      0020                CONS=(FO*PI*0.00002D0*JM)+DEVIA*TEMP*TPI
                    C****
      0021                CONS=DMOD(CONS,TPI)
      0022                IF(MODE.EQ.1) GO TO 45
      0023                SIGNAL(JE)=DSIN(CONS)
      0024                GO TO 40
      0025             45 WAF(JE)=DSIN(CONS)
      0026             40 CONTINUE
                    C**** TRIGGER APPENDAGE
      0027                IF(MODE.EQ.1) GO TO 55
      0028                CONS=(FO*PI*0.00002D0*NDP1)+DEVIA*(TEMP+DDUMD)*TPI
      0029                DUMD=DSIN(CONS)
      0030                GO TO 50
      0031             55 CONS=(FO*PI*0.00002D0*NDP1)+DEVIA*(TEMP+DDUMF)*TPI
      0032                DUMF=DSIN(CONS)
      0033             50 CONTINUE
                    C****
      0034                RETURN
      0035                END
```

```
FORTRAN IV G LEVEL  21                FMOD             DATE = 73332         00/31/53            PAGE 0002

                                   COMMON BLOCK /            / MAP SIZE    C1E8
   SYMBOL      LOCATION    SYMBOL     LOCATION     SYMBOL     LOCATION     SYMBOL     LOCATION     SYMBOL     LOCATION
   SIGNAL           0      Z              4000     MAP            8000     SPEC          C000      DUMF          C1E0
   DUMD          C1E4

                                   SUBPROGRAMS CALLED
   SYMBOL      LOCATION    SYMBOL     LOCATION     SYMBOL     LOCATION     SYMBOL     LOCATION     SYMBOL     LOCATION
   DSIN            D4

                                   SCALAR MAP
   SYMBOL      LOCATION    SYMBOL     LOCATION     SYMBOL     LOCATION     SYMBOL     LOCATION     SYMBOL     LOCATION
   PI              F0      TPI              F8     PO              100     DEVIA          108      DDUMF          110
   DDUMD          118      NDP1            120     SUM             128     TEMP           130      CONS           138
   C              140      B               144     NUMBER          148     J              14C      JE             150
   JM             154      MODE            158

                                   ARRAY MAP
   SYMBOL      LOCATION    SYMBOL     LOCATION     SYMBOL     LOCATION     SYMBOL     LOCATION     SYMBOL     LOCATION
   A              160

      *OPTIONS IN EFFECT*  ID,EBCDIC,SOURCE,NOLIST,DECK,LOAD,MAP
      *OPTICNS IN EFFECT*  NAME = FMOD     , LINECNT =       58
      *STATISTICS*    SOURCE STATEMENTS =         35,PROGRAM SIZE =     33918
      *STATISTICS*  NO DIAGNOSTICS GENERATED
```

```
0001          SUBROUTINE FLTRB(PERIOD,NUMBER,NNUMA,NNUMB,INV,S)
         C    THIS SUBROUTINE CONVERTS THE WOW AND FLUTTER DATA CHANNEL FROM
         C    THE TIME DOMAIN TO THE FREQUENCY DOMAIN.  IT THEN MULTIPLIES
         C    THE RESULT BY A TRANSFER FUNCTION WHICH IS INPUT AS A QUOTENT
         C    OF TWO POLYNOMIALS.  THEN THE CHANNEL IS CONVERTED BACK TO THE
         C    TIME DOMAIN
0002          COMMON SIGNAL(4096),WAF(4096)
0003          COMPLEX WAF,XNUM,XDENOM,QUOTNT
0004          DIMENSION M(3),INV(NNUMB),S(NNUMB),COFN(10),COFD(10)
0005          DIMENSION QUOTNT(2048)
0006          DIMENSION AMP(4096),PHASE(4096)
0007          DIMENSION CO(4096),QA(4096)
0008      200 FORMAT(6I1)
0009      201 FORMAT(1X,10F12.7)
0010      210 FORMAT(' BANDPASS FILTER ROUTINE')
0011      221 FORMAT(' ARRAY BEFORE DATA IS REARRANGED PROPERLY')
0012      222 FORMAT('1 DATA HAS NOW BEEN REARRANGED IN PROPER CELLS OF ARRAY RE
             1ADY FOR TAKING FOURIER TRANSFORM')
0013      223 FORMAT('1 OUTPUT OF FOURIER TRANSFORM')
0014      224 FORMAT('1DATA AFTER MULTIPLICATION BY TRANSFER FUNCTION')
0015      225 FORMAT('1DATA AFTER INVERSE TRANSFORMATION')
0016      226 FORMAT('1DATA AFTER REPACKING TO OBTAIN 3 REAL ARRAYS')
0017      230 FORMAT(2I3)
0018      231 FORMAT(5E16.8)
0019      232 FORMAT(1H1,2I3)
0020      233 FORMAT(1H ,5F16.8)
0021          WRITE(3,210)
0022          READ(1,200) IWRI1,IWRI2,IWRI3,IWRI4,IWRI5,IWRI6
0023          IF(IWRI1.EQ.1) WRITE(3,221)
0024          IF(IWRI1.EQ.1) WRITE(3,201) (WAF(IST),IST=1,NUMBER)
0025          M(1) = NNUMA
0026          M(2) = 0
0027          M(3) = 0
         C    NOTE:: WAF ARRAY IS DECLARED COMPLEX, SO THERE ARE ONLY TWO
         C    ARRAYS INSTEAD OF THREE IN THE COMMON SINCE THE COMPLEX ARRAY
         C    IS THE SAME SIZE AS TWO STANDARD REAL ARRAYS.  HOWEVER, THE DATA
         C    (ALL REAL DATA) IN THE COMPLEX ARRAY IS ONLY STORED IN THE LAST
         C    HALF OF THE ARRAY(IN BLTH REAL AND COMPLEX CELLS).   THEREFORE
         C    THE DATA MUST BE SPREAD OUT AMONG  THE REAL CELLS THROUGHOUT THE
         C    ARRAY
         C    AN ATTEMPT HAS BE MADE TO ACCOMPLISH THAT HERE.
0028          L = NUMBER / 2
0029          LL = L + 1
0030          DO 250 IIA = 1, L
0031          IIB = IIA + L
0032          IIC = IIA * 2
0033          IID = IIC - 1
0034          G = REAL(WAF(IIB))
0035          WAF(IID) = CMPLX(G,0.0)
0036          G = AIMAG(WAF(IIB))
0037      250 WAF(IIC) = CMPLX(G,0.0)
0038          IF(IWRI2.EQ.1) WRITE(3,222)
0039          IF(IWRI2.EQ.1) WRITE(3,201) (WAF(IST),IST=1,NUMBER)
         C    TAKE THE FOURIER TRANSFORM
0040          IFSET = 1
0041          CALL HARM(WAF,M,INV,S,IFSET,IFERR)
```

```
FORTRAN IV G LEVEL  21                 FLTRB              DATE = 73332          00/31/53               PAGE 0002

 0042                WRITE(3,17) IPERR
 0043             17 FORMAT('     IFERR =    ',I3)
 0044                IF(IWRI3.EQ.1) WRITE(3,223)
 0045                IF(IWRI3.EQ.1) WRITE(3,201) (WAF(IST),IST=1,NUMBER)
                C****
 0046                READ(1,234) POL,DEV,FO
 0047            234 FORMAT(3F10.3)
 0048                WRITE(3,235) DEV,FC
 0049            235 FORMAT(' DEV=',F10.3,' FO=',F10.3)
 0050                DELF=1.0/(PERIOD*NUMBER)
                C****
                C      "1" IN 1.5 ACCOUNTS FOR WN 1 BEING DC
                C****
 0051                NL=(1.0-DEV)*FO/DELF+1.5
 0052                NH=(1.0+DEV)*FO/DELF+1.5
 0053                NLP1=NL+1
 0054                NHP1=NH+1
 0055                BW=(NH-NL)/2
                C****
 0056                NHALF=NUMBER/2
 0057                DO 400 J=1,NL
 0058                AMAG=0.0
 0059                AFAZE=(POL/4.0)*3.14159
 0060                CO(J)=AMAG*COS(AFAZE)
 0061                QA(J)=AMAG*SIN(AFAZE)
 0062                QUOTNT(J)=CMPLX(CO(J),QA(J))
 0063            400 CONTINUE
 0064                DO 410 J=NLP1,NH
 0065                AMAG=1.0
 0066                AFAZE=(POL/4.0)*3.14159*(FO/DELF+1-J)/BW
 0067                CO(J)=AMAG*COS(AFAZE)
 0068                QA(J)=AMAG*SIN(AFAZE)
 0069                QUOTNT(J)=CMPLX(CO(J),QA(J))
 0070            410 CONTINUE
 0071                DO 420 J=NHP1,NHALF
 0072                AMAG=0.0
 0073                AFAZE=(POL/4.0)*3.14159
 0074                AFAZE=-AFAZE
 0075                CO(J)=AMAG*COS(AFAZE)
 0076                QA(J)=AMAG*SIN(AFAZE)
 0077                QUOTNT(J)=CMPLX(CO(J),QA(J))
 0078            420 CONTINUE
                C****
                C      WRITE(3,233)
                C      WRITE(3,240)  (QUOTNT(J),J=1,512)
                C 240 FORMAT(1X,8F15.8)
                C****
 0079                DO 425 J=1,NHALF
 0080                WAF(J)=WAF(J)*QUCTNT(J)
 0081            425 CONTINUE
 0082                DO 430 J=2,NHALF
 0083                NNO=NUMBER-J+2
 0084            430 WAF(NNO)=CONJG(WAF(J))
 0085                MHP1=NHALF+1
 0086                WAF(MHP1)=CMPLX(0.0,0.0)
                C****
```

```
0087          ZERO=0.0
0088          DO 500 J=1,NHALF
0089          X=CO(J)
0090          Y=QA(J)
0091          AMP(J)=SQRT(X**2+Y**2)
0092          IF(X.EQ.ZERO.AND.Y.GT.ZERO)  GO TO 550
0093          IF(X.EQ.ZERO.AND.Y.EQ.ZERO)  GO TO 545
0094          IF(X.EQ.ZERO.AND.Y.LT.ZERO)  GO TO 540
0095          GO TO 560
0096      540 PHASE(J)=270.0
0097          GO TO 500
0098      545 PHASE(J)=0.0
0099          GO TO 500
0100      550 PHASE(J)=90.0
0101          GO TO 500
0102      560 PHASE(J)=ATAN(Y/X)
0103          PHASE(J)=57.296*PHASE(J)
0104          IF(X.LT.ZERO)  GO TC 520
0105          IF(X.GT.ZERO.AND.Y.LT.ZERO)  GO TO 530
0106          GO TO 500
0107      520 PHASE(J)=PHASE(J)+180.0
0108          GO TO 500
0109      530 PHASE(J)=PHASE(J)+360.0
0110          GO TO 500
0111      500 CONTINUE
        C     WRITE(3,610)  (AMP(J),J=1,512)
        C     WRITE(3,233)
        C     WRITE(3,610)  (PHASE(J),J=1,512)
0112      610 FORMAT(10F10.3)
        C****
0113          IF(IWRI4.EQ.1)  WRITE(3,224)
0114          IF(IWRI4.EQ.1)  WRITE(3,201) (WAF(IST),IST=1,NUMBER)
        C    TAKE THE INVERSE TRANSFORM.
0115          IFSET = -1
0116          CALL HARM(WAF,M,INV,S,IPSET,IFERR)
0117          WRITE(3,17) IFERR
0118          IF(IWRI5.EQ.1)  WRITE(3,225)
0119          IF(IWRI5.EQ.1)  WRITE(3,201) (WAF(IST),IST=1,NUMBER)
        C    COMPRESS THE DATA BACK INTO ONE HALF THE ARRAY SO IT WILL BE IN
        C    THE PROPER FORM FOR A THREE ARRAY COMMON.
0120          DO 271 IIH = 1,L
0121          IIJ = L - IIH + 1
0122          IIM = L + IIJ
0123          IIK = IIJ * 2
0124          IIL = IIK - 1
0125          A = REAL(WAF(IIK))
0126          B = REAL(WAF(IIL))
0127      271 WAF(IIM) = CMPLX(B,A)
0128          IF(IWRI6.EQ.1)  WRITE(3,226)
0129          LIST=L+1
0130          IF(IWRI6.EQ.1)  WRITE(3,201) (WAF(IST),IST=LIST,NUMBER)
0131          RETURN
0132          END
```

COMMON BLOCK / / MAP SIZE C000

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
SIGNAL	0	WAP	4000						

SUBPROGRAMS CALLED

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
IBCOM#	160	HARM	164	CMPY#	168	COS	16C	SIN	170
SQRT	174	ATAN	178						

SCALAR MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
NNUMB	220	IWRI1	224	IWRI2	228	IWRI3	22C	IWRI4	230
IWRI5	234	IWRI6	238	IST	23C	NUMBER	240	NNUMA	244
L	248	LL	24C	IIA	250	IIB	254	IIC	258
IID	25C	G	260	IFSET	264	IFERR	268	POL	26C
DEV	270	FO	274	DELF	278	PERIOD	27C	NL	280
NH	284	NLP1	288	NHP1	28C	BW	290	NHALF	294
J	298	AMAG	29C	AFAZE	2A0	NNO	2A4	MHP1	2A8
ZERO	2AC	X	2B0	Y	2B4	IIH	2B8	IIJ	2BC
IIM	2C0	IIK	2C4	IIL	2C8	A	2CC	B	2D0
LIST	2D4								

ARRAY MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
M	2D8	INV	2E4	S	2E8	COFN	2EC	COFD	314
QUOTNT	340	AMP	4340	PHASE	8340	CO	C340	QA	10340

FORMAT STATEMENT MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
200	14340	201	14346	210	1434F	221	1436B	222	14398
223	143F6	224	14417	225	1444A	226	14470	230	144A1
231	144A7	232	144AE	233	144B7	17	144C1	234	144D5
235	144DC	610	144F1						

```
*OPTIONS IN EFFECT*  ID,EBCDIC,SOURCE,NOLIST,DECK,LOAD,MAP
*OPTICNS IN EFFECT*  NAME = FLTRB    , LINECNT =      58
*STATISTICS*    SOURCE STATEMENTS =        132,PROGRAM SIZE =      86516
*STATISTICS*  NO DIAGNOSTICS GENERATED
```

```
0001                SUBROUTINE FLTRA(PERIOD,NUMBER,NNUMA,NNUMB,INV,S)
             C      THIS SUBROUTINE DOES THE SAME AS THE FLTRB SUBROUTINE ONLY IT
             C      OPERATES ON THE SIGNAL ARRAY INSTEAD OF THE WAF.
0002                COMMON SIGNAL(4096),WAF(4096)
0003                COMPLEX SIGNAL,XNUM,XDENOM,QUOTNT
0004                DIMENSION M(3),INV(NNUMB),S(NNUMB),COFN(10)
0005                DIMENSION COFD(10)
0006                DIMENSION QUOTNT(2048)
0007                DIMENSION AMP(4096),PHASE(4096)
0008                DIMENSION CO(4096),QA(4096)
0009            200 FORMAT(6I1)
0010            201 FORMAT(1X,10F12.7)
0011            210 FORMAT(' BANDPASS FILTER ROUTINE')
0012            221 FORMAT(' ARRAY BEFCRE DATA IS REARRANGED PROPERLY')
0013            222 FORMAT('  DATA HAS NOW BEEN REARRANGED IN PROPER CELLS CF ARRAY RE
               1ADY FOR TAKING FOURIER TRANSFORM')
0014            223 FORMAT('  OUTPUT OF FOURIER TRANSFORM')
0015            224 FORMAT(' DATA AFTER MULTIPLICATION BY TRANSFER FUNCTION')
0016            225 FORMAT(' DATA AFTER INVERSE TRANSFORMATION')
0017            226 FORMAT(' DATA AFTER REPACKING TO OBTAIN 3 REAL ARRAYS')
0018            230 FORMAT(2I3)
0019            231 FORMAT(5E16.8)
0020            232 FORMAT(1H ,2I3)
0021            233 FORMAT(1H ,5F16.8)
0022                WRITE(3,210)
0023                READ(1,200) IWRI1,IWRI2,IWRI3,IWRI4,IWRI5,IWRI6
0024                IF(IWRI1.EQ.1) WRITE(3,221)
0025                IF(IWRI1.EQ.1) WRITE(3,201) (SIGNAL(IST),IST=1,NUMBER)
0026                M(1) = NNUMA
0027                M(2) = 0
0028                M(3) = 0
             C      NOTE:  HERE AS IN THE FLTRB SUBROUTINE THE DATA MUST BE
             C      REDISTRIBUTED AMOUNG THE REAL AND COMPLEX CELLS OF THE SIGNAL
             C      ARRAY.  THE ONLY DIFFERENCE IS THAT THE DATA MUST BE SHIFTED FROM
             C      THE LOWER HALF OF THE ARRAY UPWARD INSTEAD OF DOWNWARD.
0029                L = NUMBER / 2
0030                LL = L + 1
0031                DO 250 IIA = 1, L
0032                IIB = L - IIA + 1
0033                IIC = IIB * 2
0034                IID = IIC - 1
0035                G = REAL(SIGNAL(IIB))
0036                H = AIMAG(SIGNAL(IIB))
0037                SIGNAL(IID) = CMPLX(G,0.0)
0038            250 SIGNAL(IIC) = CMPLX(H,0.0)
0039                IF(IWRI2.EQ.1) WRITE(3,222)
0040                IF(IWRI2.EQ.1) WRITE(3,201) (SIGNAL(IST),IST=1,NUMBER)
             C      TAKE THE FOURIER TRANSFORM
0041                IPSET = 1
0042                CALL HARM(SIGNAL,M,INV,S,IPSET,IFERR)
0043                WRITE(3,17) IFERR
0044             17 FORMAT('        IFERR =    ',I3)
0045                IF(IWRI3.EQ.1) WRITE(3,223)
0046                IF(IWRI3.EQ.1) WRITE(3,201) (SIGNAL(IST),IST=1,NUMBER)
             C****
0047                READ(1,234) POL,DEV,FO
```

```
0048      234 FORMAT(3F10.3)
0049          WRITE(3,235) DEV
0050      235 FORMAT(' DEVIATION EQUALS',F10.3)
0051          DELF=1.0/(PERIOD*NUMBER)
        C****
        C     "1" IN 1.5 ACCOUNTS FOR WN 1 BEING DC
        C****
0052          NL=(1.0-DEV)*FO/DELF+1.5
0053          NH=(1.0+DEV)*FO/DELF+1.5
0054          NLP1=NL+1
0055          NHP1=NH+1
0056          BW=(NH-NL)/2
        C****
0057          NHALF=NUMBER/2
0058          DO 400 J=1,NL
0059          AMAG=0.0
0060          AFAZE=(POL/4.0)*3.14159
0061          CO(J)=AMAG*COS(AFAZE)
0062          QA(J)=AMAG*SIN(AFAZE)
0063          QUOTNT(J)=CMPLX(CO(J),QA(J))
0064      400 CONTINUE
0065          DO 410 J=NLP1,NH
0066          AMAG=1.0
0067          AFAZE=(POL/4.0)*3.14159*(FO/DELF+1-J)/BW
0068          CO(J)=AMAG*COS(AFAZE)
0069          QA(J)=AMAG*SIN(AFAZE)
0070          QUOTNT(J)=CMPLX(CO(J),QA(J))
0071      410 CONTINUE
0072          DO 420 J=NHP1,NHALF
0073          AMAG=0.0
0074          AFAZE=(POL/4.0)*3.14159
0075          AFAZE=-AFAZE
0076          CO(J)=AMAG*COS(AFAZE)
0077          QA(J)=AMAG*SIN(AFAZE)
0078          QUOTNT(J)=CMPLX(CO(J),QA(J))
0079      420 CONTINUE
        C****
        C     WRITE(3,233)
        C     WRITE(3,240) (QUOTNT(J),J=1,512)
        C 240 FORMAT(1X,8F15.8)
        C****
0080          DO 425 J=1,NHALF
0081          SIGNAL(J)=SIGNAL(J)*QUOTNT(J)
0082      425 CONTINUE
0083          DO 430 J=2,NHALF
0084          NNO=NUMBER-J+2
0085      430 SIGNAL(NNO)=CONJG(SIGNAL(J))
0086          MHP1=NHALF+1
0087          SIGNAL(MHP1)=CMPLX(0.0,0.0)
        C****
0088          ZERO=0.0
0089          DO 500 J=1,NHALF
0090          X=CO(J)
0091          Y=QA(J)
0092          AMP(J)=SQRT(X**2+Y**2)
0093          IF(X.EQ.ZERO.AND.Y.GT.ZERO) GO TO 550
```

```
0094          IF(X.EQ.ZERO.AND.Y.EQ.ZERO) GO TO 545
0095          IF(X.EQ.ZERO.AND.Y.LT.ZERO) GO TO 540
0096          GO TO 560
0097      540 PHASE(J)=270.0
0098          GO TO 500
0099      545 PHASE(J)=0.0
0100          GO TO 500
0101      550 PHASE(J)=90.0
0102          GO TO 500
0103      560 PHASE(J)=ATAN(Y/X)
0104          PHASE(J)=57.296*PHASE(J)
0105          IF(X.LT.ZERO) GO TO 520
0106          IF(X.GT.ZERO.AND.Y.LT.ZERO) GO TO 530
0107          GO TO 500
0108      520 PHASE(J)=PHASE(J)+180.0
0109          GO TO 500
0110      530 PHASE(J)=PHASE(J)+360.0
0111          GO TO 500
0112      500 CONTINUE
        C     WRITE(3,610) (AMP(J),J=1,512)
        C     WRITE(3,233)
        C     WRITE(3,610) (PHASE(J),J=1,512)
0113      610 FORMAT(10F10.3)
        C****
0114          IF(IWRI4.EQ.1) WRITE(3,224)
0115          IF(IWRI4.EQ.1) WRITE(3,201) (SIGNAL(IST),IST=1,NUMBER)
        C    TAKE THE INVERSE FOURIER TRANSFORM
0116          IFSET = -1
0117          CALL HARM(SIGNAL,M,INV,S,IFSET,IFERR)
0118          WRITE(3,17) IFERR
0119          IF(IWRI5.EQ.1) WRITE(3,225)
0120          IF(IWRI5.EQ.1) WRITE(3,201) (SIGNAL(IST),IST=1,NUMBER)
        C    COMPRESS THE DATA BACK INTO ONE HALF THE ARRAY SO IT WILL BE IN THE
        C    PROPER FORM FOR A THREE ARRAY COMMON.
0121          DO 271 IIH = 1,L
0122          IIJ = 2 * IIH
0123          IIK = IIJ  - 1
0124          A = REAL(SIGNAL(IIK))
0125          B = REAL(SIGNAL(IIJ))
0126      271 SIGNAL(IIH) = CMPLX(A,B)
0127          IF(IWRI6.EQ.1) WRITE(3,226)
0128          IF(IWRI6.EQ.1) WRITE(3,201) (SIGNAL(IST),IST=1,L)
0129          RETURN
0130          END
```

COMMON BLOCK / / MAP SIZE C000

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
SIGNAL	0	WAF	8000						

SUBPROGRAMS CALLED

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
IBCOM#	160	HARM	164	CMPY#	168	COS	16C	SIN	170
SQRT	174	ATAN	178						

SCALAR MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
NNUMB	220	IWRI1	224	IWRI2	228	IWRI3	22C	IWRI4	230
IWRI5	234	IWRI6	238	IST	23C	NUMBER	240	NNUMA	244
L	248	LL	24C	IIA	250	IIB	254	IIC	258
IID	25C	G	260	H	264	IFSET	268	IFERR	26C
POL	270	DEV	274	FO	278	DELP	27C	PERIOD	280
NL	284	NH	288	NLP1	28C	NHP1	290	BW	294
NHALF	298	J	29C	AMAG	2A0	AFAZE	2A4	NNO	2A8
NHP1	2AC	ZERO	2B0	X	2B4	Y	2B8	IIH	2BC
IIJ	2C0	IIK	2C4	A	2C8	B	2CC		

ARRAY MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
M	2D0	INV	2CC	S	2B0	COFN	2B4	COFD	30C
QUOTNT	338	AMP	4338	PHASE	8338	CO	C338	QA	10338

FORMAT STATEMENT MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
200	14338	201	1433E	210	14347	221	14363	222	14390
223	143EE	224	1440F	225	14442	226	14468	230	14499
231	1449F	232	144A6	233	144AF	17	144B9	234	144CE
235	144D5	610	144ED						

```
*OPTIONS IN EFFECT*  ID,EBCDIC,SOURCE,NOLIST,DECK,LOAD,MAP
*OPTIONS IN EFFECT*  NAME = FLTRA    , LINECNT =       58
*STATISTICS*    SOURCE STATEMENTS =        130,PROGRAM SIZE =     86468
*STATISTICS*  NO DIAGNOSTICS GENERATED
```

```
FORTRAN IV G LEVEL  21                MAIN              DATE = 73332        00/31/53             PAGE 0001

                 C
 0001                  SUBROUTINE TRIGGR(PERIOD,MODE,NUMBER)
 0002                  COMMON SIGNAL(4096),A(4096),WAF(4096),SPEC(120),DUMF,DUMD
 0003              400 FORMAT(7F10.5)
 0004              401 FORMAT(3I1)
 0005              402 FORMAT('   PULSE WIDTH COEF SHOULD BE ZERO. PROGRAM HALTS ')
 0006              403 FORMAT('   NEGATIVE VALUE FOR MODE       ')
 0007                  READ(1,401) IWR1,IWR2,IWR3
 0008                  READ(1,400) ETRIGU,ETRIGD,TRGDEL
 0009                  READ(1,400) PWCOFA,PWCONA,PWDELA,PPAMA,PNAMA
 0010                  READ(1,400) PWCOFB,PWCONB,PWDELB,PPAMB,PNAMB
 0011                  READ(1,400) ENORHI,ENORLO,XNORDE,ETRIP
                 C****
 0012                  PWCOFA=0.0
 0013                  PWCOFB=0.0
 0014                  PNAMA=-9.0
 0015                  PNAMB=-9.0
 0016                  PPAMA=9.0
 0017                  PPAMB=9.0
 0018                  ETRIP=0.0
 0019                  NEIGHT=NUMBER/8
                 C****
 0020                  WRITE(3,1100) ETRIGU,ETRIGD,TRGDEL
 0021                  WRITE(3,1101) PWCOFA,PWCONA,PWDELA,PPAMA,PNAMA
 0022                  WRITE(3,1102) PWCOFB,PWCONB,PWDELB,PPAMB,PNAMB
 0023                  WRITE(3,1103) ENORHI,ENORLO,XNORDE,ETRIP
 0024             1100 FORMAT(' TRIGGER SUBROUTINE'/' TRIGGER SECTION'/'      TRIGGERS ON'
                      1,F7.3,' VOLTS FOR INCREASING VOLTAGES AND ', F7.3,' VOLTS FOR DECR
                      2EASING'/'      DELAY IS',F15.10,' SECONDS')
 0025             1101 FORMAT(' DETECTOR SECTION'/'    DETECTOR A'/'      PULSE WIDTH IS',F
                      110.7,'  TIMES THE FLUTTER SIGNAL PLUS ',F10.7,'SECONDS'/'      DELA
                      2Y IS', F10.7,' PEAK PULSE VALUE IS',F7.3,'AND RESIDUAL VALUE IS',
                      3F7.3)
 0026             1102 FORMAT(' DETECTOR SECTION'/'    DETECTOR B'/'      PULSE WIDTH IS',F
                      110.7,'  TIMES THE FLUTTER SIGNAL PLUS ',F10.7,'SECONDS'/'      DELA
                      2Y IS', F10.7,' PEAK PULSE VALUE IS',F7.3,'AND RESIDUAL VALUE IS',
                      3F7.3)
 0027             1103 FORMAT(' NOR CIRCUIT SECTION'/'    OUTPUT VOLTAGES'/'      HIGH-- ',
                      1F7.3/'      LOW-- ',F7.3/'      DELAY-- ',F7.3/'      TRANSITION VOLT
                      2AGE-- ',F7.3)
                 C     MODE = 0 OPERATE ON SIGNAL ARRAY
                 C     MODE = 1 OPERATE ON WOW AND FLUTTER ARRAY
 0028                  IF(MODE.EQ.1.AND.PWCOFA.GT.0.0) GO TO 495
 0029                  IF(MODE.EQ.1.AND.PWCOFB.GT.0.0) GO TO 495
                 C     SCHMITT TRIGGER
                 C     THIS SECTION SCANS THE DATA STREAM FOR CROSSINGS OF THE TRIGGERING
                 C     VOLTAGE.  IF THE VOLTAGE CROSSES ETRIGU INCREASING, THEN A VALUE
                 C     OF 100 IS PUT ON THE DATA STREAM AT THAT POINT.  LIKEWISE IF THE
                 C     VOLTAGE CROSSES ETRIGD DECREASING, THEN A VALUE OF 100 IS PUT ON
                 C     THE A DATA STREAM AT THAT POINT.
 0030                  IT = NUMBER - 1
 0031                  DO 458 IJ = 1, IT
 0032                  IL = IJ + 1
 0033                  A(IJ) = 0.0
 0034                  IF(MODE.GE.1) GO TO 455
                 C     THIS SECTION IS USED IF THE DATA STREAM IS THE SIGNAL STREAM(NOT
```

```
FORTRAN IV G LEVEL  21                    TRIGGR              DATE = 73332          00/31/53                  PAGE 0002

                    C     THE WOW AND FLUTTER STREAM)
 0035                     IF(SIGNAL(IJ).LT.ETRIGU.AND.SIGNAL(IL).GT.ETRIGU)  GO TO 452
 0036                     IF(SIGNAL(IJ).GT.ETRIGD.AND.SIGNAL(IL).LT.ETRIGD)  GO TO 454
 0037                     SIGNAL(IJ) = 0.0
 0038                     GO TO 456
 0039                 452 SIGNAL(IJ)  =  100.0
 0040                     GO TO 456
 0041                 454 A(IJ)  = 100.0
 0042                     SIGNAL(IJ)  = 0.0
 0043                     GO TO 456
                    C     THIS SECTION IS USED IF THE DATA STREAM TO BE USED IS THE WOW AND
                    C     FLUTTER STREAM
 0044                 455 IF(WAF(IJ).LT.ETRIGU.AND.WAF(IL).GT.ETRIGU)  GO TO 457
 0045                     IF(WAF(IJ).GT.ETRIGD.AND.WAF(IL).LT.ETRIGD)  GO TO 459
 0046                     WAF(IJ)  = 0.0
 0047                     GO TO 456
 0048                 457 WAF(IJ)  =  100.0
 0049                     GO TO 456
 0050                 459 A(IJ)  =  100.0
 0051                     WAF(IJ)  =  0.0
 0052                 456 CONTINUE
 0053                 458 CONTINUE
                    C**** APPENDAGE
 0054                     A(NUMBER)=0.0
 0055                     IF(MODE.EQ.1)  GO TC 555
 0056                     IF(SIGNAL(NUMBER).LT.ETRIGU.AND.DUMD.GT.ETRIGU)  GO TO 552
 0057                     IF(SIGNAL(NUMBER).GT.ETRIGD.AND.DUMD.LT.ETRIGD)  GO TO 554
 0058                     SIGNAL(NUMBER)=0.0
 0059                     GO TO 556
 0060                 552 SIGNAL(NUMBER)=100.0
 0061                     GO TO 556
 0062                 554 A(NUMBER)=100.0
 0063                     SIGNAL(NUMBER)=0.0
 0064                     GO TO 556
 0065                 555 CONTINUE
 0066                     IF(WAF(NUMBER).LT.ETRIGU.AND.DUMF.GT.ETRIGU)  GO TO 557
 0067                     IF(WAF(NUMBER).GT.ETRIGD.AND.DUMF.LT.ETRIGD)  GO TO 559
 0068                     WAF(NUMBER)=0.0
 0069                     GO TO 556
 0070                 557 WAF(NUMBER)=100.0
 0071                     GO TO 556
 0072                 559 A(NUMBER)=100.0
 0073                     WAF(NUMBER)=0.0
 0074                 556 CONTINUE
                    C****
 0075                     IF(IWR1.EQ.1)  CALL WRITE(NUMBER)
                    C     AT THIS POINT DATA IS CONTAINED ON THE TWO VECTORSWHICH GIVE THE
                    C     PULSES REPRESENTING ZERO CROSSINGS,  ONE VECTRR OF UPWARD CROSSINGS
                    C     AND ONE FOR DOWNWARC.
                    C     DETECTORS A AND B
                    C     THIS SECTION SCANS THE DATA STREAM AND THE A STREAM FOR PULSES AND
                    C     FOR EACH PULSE IT OUTPUTS A PULSE OF SPECIFIED DURATION.  IF THE
                    C     PULSE IS TO BE PUT ON THE SIGNAL CHANNEL, ITS DURATION IS
                    C     DETERMINED BY THE MAGNITUDE OF THE WAF CHANNEL AT THAT TIME.  IF IT
                    C     IS TO BE BUT ON THE WAF CHANNEL ITS DURATION IS FIXED.
                    C****
```

```
                C     LOGIC MODIFIED TO CHANGE PULSE HEIGHT
                C****
      0076            DO 460 IK = 1, NUMBER
      0077            IM = NUMBER - IK + 1
      0078            IF(MODE.GE.1) GO TO 468
                C     CHECK FOR INPUT PULSES
      0079            IF(SIGNAL(IM).GT.80.0) GO TO 475
      0080            SIGNAL(IM) = PNAMA
      0081            GO TO 469
      0082        468 IF(WAF(IM).GT.80.0) GO TO 476
      0083            WAF(IM) = PNAMA
      0084        469 CONTINUE
      0085            IF(A(IM).GT.80.0) GO TO 480
      0086            A(IM) = PNAMB
      0087            GO TO 490
                C     SET LINE TO NO PULSE VALUE
      0088        475 SIGNAL(IM) = PNAMA
      0089            A(IM) = PNAMB
      0090            GO TO 477
      0091        476 WAF(IM) = PNAMA
      0092            A(IM) = PNAMB
      0093        477 IPWID=((PWCOFA*WAF(IM)+PWCONA)/PERIOD)+0.5
                C     COMPUTE PULSE HEIGHT FOR A DECTOR
                C****
      0094            TEMPA=-PPAMA*WAF(IM)+PPAMA
                C****
                C     PUT IN PULSES IN DATA LINE
      0095            IDELAY=(PWDELA/PERIOD)+0.5
      0096            IN = IDELAY + IM
      0097            IO=IN+IPWID-1
      0098            IO=MINO(IO,NUMBER)
      0099            DO 472 IP = IN,IO
      0100            IF(MODE.GE.1) GO TC 471
                C****
      0101            SIGNAL(IP)=TEMPA
                C****
      0102            GO TO 472
      0103        471 WAF(IP) = PPAMA
      0104        472 CONTINUE
      0105            GO TO 489
      0106        480 IPWID=((PWCOPB*WAF(IM)+PWCONB)/PERIOD)+0.5
                C     COMPUTE PULSE HEIGHT FOR B DETECTOR
                C****
      0107            TEMPB=-PPAMB*WAF(IM)+PPAMB
                C****
      0108            A(IM) = PNAMB
      0109            IDELAY=(PWDELB/PERIOD)+0.5
      0110            IN = IDELAY + IM
      0111            IO=IN+IPWID-1
      0112            IO=MINO(IO,NUMBER)
                C     PUT PULSE ON LINE (A ARRAY)
      0113            DO 482 IP = IN,IO
                C****
      0114            IF(MODE.GE.1) GO TC 481
      0115            A(IP)=TEMPB
      0116            GO TO 482
```

```
FORTRAN IV G LEVEL  21                  TRIGGR             DATE = 73332          00/31/53              PAGE 0004
  0117             481 A(IP)=PPAMB
  0118             482 CONTINUE
                 C****
  0119                 GO TO 489
  0120             488 WRITE(3,403)
  0121             489 CONTINUE
  0122             490 CONTINUE
  0123             460 CONTINUE
                 C     NOR CIRCUIT
                 C     THIS SECTION NORS THE OUTPUT OF THE A AND B DETECTORS
  0124                 IF(IWR2.EQ.1) CALL WRITE(NUMBER)
  0125                 IXDEL =(XNORDE / PERIOD)+0.5
  0126                 DO 492 IR = 1, NUMBER
  0127                 IS = NUMBER - IR + 1
  0128                 IT = IS + IXDEL
  0129                 IF(IT.GT.NUMBER) GO TO 492
  0130                 IF(MODE) 492,499,497
                 C****
  0131             499 IF(SIGNAL(IS).GT.ETRIP.OR.A(IS).GT.ETRIP) GO TO 600
  0132                 IF(SIGNAL(IS).LT.ETRIP.AND.A(IS).LT.ETRIP) SIGNAL(IT)=
                      1-AMIN1(SIGNAL(IT),A(IT))
  0133                 GO TO 492
  0134             600 SIGNAL(IT)=-AMAX1(SIGNAL(IT),A(IT))
  0135                 GO TO 492
  0136             497 IF(WAF(IS).GT.ETRIP.OR.A(IS).GT.ETRIP) GO TO 601
  0137                 IF(WAF(IS).LT.ETRIF.AND.A(IS).LT.ETRIP) WAF(IT)=
                      1-AMIN1(WAF(IT),A(IT))
  0138                 GO TO 492
  0139             601 WAF(IT)=-AMAX1(WAF(IT),A(IT))
                 C****
  0140             492 CONTINUE
  0141                 GO TO 494
  0142             495 WRITE(3,402)
  0143                 CALL EXIT
  0144             494 CONTINUE
  0145                 IF(IXDEL.EQ.0) GO TO 496
  0146                 DO 440 IT = 1, IXDEL
  0147                 IF(MODE) 440, 442, 443
  0148             442 SIGNAL(IT) = ENORLC
  0149                 GO TO 440
  0150             443 WAF(IT) = ENORLO
  0151             440 CONTINUE
  0152             496 CONTINUE
  0153                 IF(IWR3.EQ.1) CALL WRITE(NUMBER)
  0154                 RETURN
  0155                 END
```

```
                              COMMON BLOCK /         / MAP SIZE   C1E8
 SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION
 SIGNAL        0       A           4000      WAP         8000      SPEC        C000      DUMP        C1E0
 DUMD       C1E4

                              SUBPROGRAMS CALLED
 SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION
 IBCOM#       23C      WRITE        240      EXIT         244      MINO         248      AMIN1        24C
 AMAX1        250

                              SCALAR MAP
 SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION
 IWR1         290      IWR2         294      IWR3         298      ETRIGU       29C      ETRIGD       2A0
 TRGDEL       2A4      PWCOFA       2A8      PWCONA       2AC      PWDELA       2B0      PPAMA        2B4
 PNAMA        2B8      PWCOFB       2BC      PWCONB       2C0      PWDELB       2C4      PPAMB        2C8
 PNAMB        2CC      ENORHI       2D0      ENORLO       2D4      XNORDE       2D8      ETRIP        2DC
 NRIGHT       2E0      NUMBER       2E4      MODE         2E8      IT           2EC      IJ           2F0
 IL           2F4      IK           2F8      IM           2FC      IPWID        300      PERIOD       304
 TEMPA        308      IDELAY       30C      IN           310      IO           314      IP           318
 TEMPB        31C      IXDEL        320      IR           324      IS           328

                              FORMAT STATEMENT MAP
 SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION
   400        32C        401        333        402        339        403        36F       1100        391
  1101        42D       1102        4DF       1103        591

    *OPTIONS IN EFFECT*  ID,EBCDIC,SOURCE,NOLIST,DECK,LOAD,MAP
    *OPTIONS IN EFFECT*  NAME = TRIGGR   , LINECNT =       58
    *STATISTICS*    SOURCE STATEMENTS =       155,PROGRAM SIZE =       4980
    *STATISTICS*  NO DIAGNOSTICS GENERATED
```

```
                C
                C
                C
                C
0001                  SUBROUTINE DELAYP(PERIOD,MODE,NUMBER)
0002                  COMMON SIGNAL(4096),A(4096),WAF(4096)
0003                  DIMENSION TEMP(500)
                C     PROGRAM SCANS APPROPRIATE DATA STREAM AND MOVES ALL DATA POINTS
                C     AN APPROPRIATE AMOUNT LATER IN TIME.
0004              700 FORMAT(F10.5)
0005              401 FORMAT(1I1)
0006                  READ(1,401)IWR1
0007                  READ(1,700) DELAY
                C     COMPUTE DELAY IN TERMS OF SAMPLE POINTS.
0008                  IF(MODE) 750,721,722
0009              722 WRITE(3,1104) DELAY
0010             1104 FORMAT(' DELAY SUBROUTINE,  DELAYS DATA IN SIGNAL STREAM',F10.7, '
                     1SECONDS ')
0011                  GO TO 723
0012              721 WRITE(3,1105) DELAY
0013             1105 FORMAT(' DELAY SUBROUTINE DELAYS DATA IN WAF STREAM ',F10.7,' SECO
                     1NDS')
0014              723 CONTINUE
0015                  IVV =(DELAY / PERICD)+0.5
0016                  IU = NUMBER- IVV
                C     FILL UNSHIFTED DATA WITH CYCLIC ARRAY
0017                  IF(IVV.EQ.0) GO TO 726
0018                  DO 725 IV=1,IVV
0019                  IY=IU+IV
0020                  IF(MODE)753,730,729
0021              729 TEMP(IV)=SIGNAL(IY)
0022                  GO TO 725
0023              730 TEMP(IV)=WAF(IY)
0024              725 CONTINUE
0025              726 CONTINUE
                C     DELAY REST OF ARRAY
0026                  DO 750 IV = 1, IU
0027                  IW = IU - IV + 1
0028                  IX = IW + IVV
0029                  IF(MODE)753,752,751
0030              751 SIGNAL(IX) = SIGNAL(IW)
0031                  GO TO 750
0032              752 WAF(IX) = WAF(IW)
0033                  GO TO 750
0034              753 WRITE(3,705)
0035              705 FORMAT('    MODE VALUE IS NEGATIVE')
0036              750 CONTINUE
                C     PUT CYCLIC DATA IN FRONT OF ARRAY
0037                  IF(IVV.EQ.0) GO TO 761
0038                  DO 760 I=1,IVV
0039                  IF(MODE)753,764,762
0040              762 SIGNAL(I)=TEMP(I)
0041                  GO TO 760
0042              764 WAF(I)=TEMP(I)
0043              760 CONTINUE
0044              761 CONTINUE
```

```
FORTRAN IV G LEVEL  21              DELAYP              DATE = 73332        00/31/53            PAGE 0002

  0045              IF(IWR1.EQ.1) CALL WRITE(NUMBER)
  0046              RETURN
  0047              END
```

```
FORTRAN IV G LEVEL  21              DELAYP            DATE = 73332          00/31/53              PAGE 0003

                                       COMMON BLOCK /        / MAP SIZE   C000
SYMBOL    LOCATION      SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION
SIGNAL         0        A           4000       WAP         8000

                                       SUBPROGRAMS CALLED
SYMBOL    LOCATION      SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION
IBCOM#       12C        WRITE        130

                                       SCALAR MAP
SYMBOL    LOCATION      SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION
IWR1         138        DELAY        13C       MODE         140       IVV          144       PERIOD       148
IU           14C        NUMBER       150       IV           154       IY           158       IW           15C
IX           160        I            164

                                       ARRAY MAP
SYMBOL    LOCATION      SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION
TEMP         168

                                       FORMAT STATEMENT MAP
SYMBOL    LOCATION      SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION
  700        938          401        93D         1104       943         1105       984          705       9C1

    *OPTIONS IN EFFECT*  ID,EBCDIC,SOURCE,NOLIST,DECK,LOAD,MAP
    *OPTIONS IN EFFECT*  NAME = DELAYP  , LINECNT =       58
    *STATISTICS*    SOURCE STATEMENTS =        47,PROGRAM SIZE =      3358
    *STATISTICS*  NO DIAGNOSTICS GENERATED
```

```
FORTRAN IV G LEVEL  21                -AMPL              DATE = 73332          00/31/53                PAGE 0001

0001                 SUBROUTINE AMPL(MODE,NUMBER)
              C     PROGRAM SIMPLY MULTIPLIES ALL VALUES OF THE APPROPRIATE DATA
              C     STREAM BY A CONSTANT
0002                 COMMON SIGNAL(4096),A(4096),WAF(4096)
0003             401 FORMAT(1I1)
0004             600 FORMAT('  MODE VALUE IS NEGATIVE')
0005             601 FORMAT(1F10.5)
0006                 READ(1,401)IWR1
0007                 READ(1,601) GAIN
0008                 IF(MODE) 655,621,622
0009             621 WRITE(3,1106) GAIN
0010            1106 FORMAT(' AMPLIFIER SUBROUTINE, PROVIDES GAIN OF ',F10.2,'FOR SIGNA
                    1L DATA')
0011                 GO TO 623
0012             622 WRITE(3,1107) GAIN
0013            1107 FORMAT(' AMPLIFIER SUBROUTINE, PROVIDES GAIN OF ',F10.2,'FOR WAF D
                    1ATA')
0014             623 CONTINUE
0015                 DO 655 IU = 1, NUMBER
0016                 IF(MODE) 651,652,653
0017             651 WRITE(3,600)
0018                 GO TO 655
0019             652 SIGNAL(IU) = GAIN* SIGNAL(IU)
0020                 GO TO 655
0021             653 WAF(IU) = GAIN * WAF(IU)
0022             655 CONTINUE
0023                 IF(IWR1.EQ.1) CALL WRITE(NUMBER)
0024                 RETURN
0025                 END
```

```
FORTRAN IV G LEVEL  21              AMPL            DATE = 73332       00/31/53            PAGE 0002

                                   COMMON BLOCK /          / MAP SIZE    C000
   SYMBOL     LOCATION       SYMBOL     LOCATION      SYMBOL     LOCATION      SYMBOL     LOCATION      SYMBOL     LOCATION
   SIGNAL         0          A            4000        WAF          8000

                                   SUBPROGRAMS CALLED
   SYMBOL     LOCATION       SYMBOL     LOCATION      SYMBOL     LOCATION      SYMBOL     LOCATION      SYMBOL     LOCATION
   IBCOM#        DC          WRITE         E0

                                   SCALAR MAP
   SYMBOL     LOCATION       SYMBOL     LOCATION      SYMBOL     LOCATION      SYMBOL     LOCATION      SYMBOL     LOCATION
   IWR1          E8          GAIN          EC         MODE          F0         IU            F4         NUMBER        F8

                                   FORMAT STATEMENT MAP
   SYMBOL     LOCATION       SYMBOL     LOCATION      SYMBOL     LOCATION      SYMBOL     LOCATION      SYMBOL     LOCATION
     401         FC            600         102          601         11E          1106         125          1107         165

     *OPTIONS IN EFFECT*  ID,EBCDIC,SOURCE,NOLIST,DECK,LOAD,MAP
     *OPTIONS IN EFFECT*  NAME = AMPL     , LINECNT =       58
     *STATISTICS*     SOURCE STATEMENTS =        25,PROGRAM SIZE =       868
     *STATISTICS*  NO DIAGNOSTICS GENERATED
```

```
FORTRAN IV G LEVEL  21                 MAIN              DATE = 73332          00/31/53                PAGE 0001
                  C
     0001               SUBROUTINE LIMIT(PERIOD,MODE,NUMBER)
     0002               COMMON SIGNAL(4096),A(4096),WAF(4096)
     0003           300 FORMAT(4F10.3)
     0004           301 FORMAT('     NEGATIVE VALUE FOR MODE  ')
     0005           401 FORMAT(1I1)
     0006               READ(1,401)IWR1
     0007               READ(1,300) GAIN1, GAIN2, DELAY, ECLIP
     0008               WRITE(3,1107) GAIN1,GAIN2,ECLIP,DELAY
     0009          1107 FORMAT(' SUBROUTINE LIMIT'/'    MODELS LIMITER WITH GAIN OF ',F7.3,
                       1'BEFORE IT AND ', F7.3,' AFTER'/'    CLIPPING VALUE IS PLUS OR MINU
                       2S ',F7.3,' VOLTS'/'      DELAY IS ',F12.8)
                  C     GAIN 1--GAIN BEFORE THE CLIPPING STAGE
                  C     GAIN2--GAIN AFTER THE CLIPPING STAGE
     0010               TOTAL = 0.0
                  C     COMPUTE AVERAGE SIGNAL LEVEL
     0011               IF(MODE.GT.0) GO TC 351
     0012               DO 350 IA = 1, NUMBER
     0013           350 TOTAL = SIGNAL(IA) + TOTAL
     0014               GO TO 353
     0015           351 CONTINUE
     0016               DO 352 IA = 1, NUMBER
     0017           352 TOTAL = WAF(IA) + TOTAL
     0018           353 CONTINUE
     0019               MEAN = TOTAL / NUMBER
     0020               IDELAY =(DELAY / PERIOD)+0.5
     0021               ETEST = ECLIP / GAIN1
     0022               ENEG = -ETEST
                  C     SCAN DATA STREAM ANC LIMIT AMPLIFIED WAVEFORM AT PLUS OR MINUS
                  C     ECLIP AND THEN AMPLIFY IT.
     0023               DO 355 IB = 1, NUMBER
     0024               IC=NUMBER-IB+1
     0025               IF(MODE) 340, 371, 372
     0026           371 ESIG = SIGNAL(IC) - MEAN
     0027               GO TO 375
     0028           372 ESIG = WAF(IC)    - MEAN
     0029               GO TO 375
     0030           340 WRITE(3,301)
     0031           375 IF(ESIG.GT.ETEST) GO TO 360
     0032               IF(ESIG.LT.ENEG) GC TO 362
     0033               ESIG = ESIG * GAIN1 * GAIN2
     0034               GO TO 364
     0035           360 ESIG = ECLIP * GAIN2
     0036               GO TO 364
     0037           362 ESIG = -ECLIP * GAIN2
     0038           364 ID=IC+IDELAY
     0039               IF(ID.GT.NUMBER) GO TO 366
     0040               IF(MODE.GT.0) GO TO 367
     0041               SIGNAL(ID) = ESIG
     0042               GO TO 366
     0043           367 WAF(ID) = ESIG
     0044           366 CONTINUE
     0045           355 CONTINUE
     0046               IF(IDELAY.LE.0) GO TO 390
                  C     ZERO OUT THE INITIAL PART OF THE DATA STREAM WHICH WAS DELAYED
     0047               IF(IDELAY.EQ.0) GO TO 390
```

```
FORTRAN IV G LEVEL  21                    LIMIT              DATE = 73332          00/31/53                PAGE 0002

                DO 370 IE = 1, IDELAY
                IF(MODE.GE.1) GO TO 369
                SIGNAL(IE) = 0.0
                GO TO 370
            369 WAF(IE) = 0.0
            370 CONTINUE
            390 CONTINUE
                IF(IWR1.EQ.1) CALL WRITE(NUMBER)
                RETURN
                END
```

```
FORTRAN IV G LEVEL  21                  LIMIT             DATE = 73332          00/31/53               PAGE 0003

                                 COMMON BLOCK /          / MAP SIZE    C000
 SYMBOL     LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION
 SIGNAL         0        A           4000       MAP         8000

                                 SUBPROGRAMS CALLED
 SYMBOL     LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION
 IBCOM#       140        WRITE        144

                                 SCALAR MAP
 SYMBOL     LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION
 IWR1         14C        GAIN1        150       GAIN2        154        DELAY        158        ECLIP        15C
 TOTAL        160        MODE         164       IA           168        NUMBER       16C        MEAN         170
 IDELAY       174        PERIOD       178       ETEST        17C        ENEG         180        IB           184
 IC           188        ESIG         18C       ID           190        IE           194

                                 FORMAT STATEMENT MAP
 SYMBOL     LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION
    300       198           301       19F          401       1C0          1107       1C6


   *OPTIONS IN EFFECT*  ID,EBCDIC,SOURCE,NOLIST,DECK,LOAD,MAP
   *OPTIONS IN EFFECT*  NAME = LIMIT    , LINECNT =       58
   *STATISTICS*    SOURCE STATEMENTS =       57,PROGRAM SIZE =      1696
   *STATISTICS*  NO DIAGNOSTICS GENERATED
```

```
FORTRAN IV G LEVEL  21              WRITE              DATE = 73332          00/31/53              PAGE 0001

                 SUBROUTINE WRITE(NUMBER)
                 COMMON SIGNAL(4096),A(4096),WAF(4096)
              19 FORMAT('1')
              20 FORMAT(5H SIG  ,10F10.5)
              21 FORMAT(5H A    ,10F10.5)
              22 FORMAT(5H WAF  ,10F10.5)
                 WRITE(3,19)
                 WRITE(3,20)  (SIGNAL(LL),LL=1,NUMBER)
                 WRITE(3,21)  (A(LL),LL=1,NUMBER)
                 WRITE(3,22)  (WAF(LL),LL=1,NUMBER)
                 RETURN
                 END
```

```
FORTRAN IV G LEVEL  21              WRITE              DATE = 73332        00/31/53            PAGE 0002

                                   COMMON BLOCK /           / MAP SIZE    C000
SYMBOL     LOCATION     SYMBOL     LOCATION     SYMBOL     LOCATION     SYMBOL     LOCATION     SYMBOL     LOCATION
SIGNAL         0        A             4000      WAP           8000

                                   SUBPROGRAMS CALLED
SYMBOL     LOCATION     SYMBOL     LOCATION     SYMBOL     LOCATION     SYMBOL     LOCATION     SYMBOL     LOCATION
IBCOM#        A8

                                   SCALAR MAP
SYMBOL     LOCATION     SYMBOL     LOCATION     SYMBOL     LOCATION     SYMBOL     LOCATION     SYMBOL     LOCATION
LL            AC        NUMBER        E0

                                   FORMAT STATEMENT MAP
SYMBOL     LOCATION     SYMBOL     LOCATION     SYMBOL     LOCATION     SYMBOL     LOCATION     SYMBOL     LOCATION
    19        B4            20        E9            21        C7            22        D5

   *OPTIONS IN EFFECT*  ID,EBCDIC,SOURCE,NOLIST,DECK,LOAD,MAP
   *OPTIONS IN EFFECT*  NAME = WRITE    , LINECNT =       58
   *STATISTICS*    SOURCE STATEMENTS =        12,PROGRAM SIZE =        600
   *STATISTICS*  NO DIAGNOSTICS GENERATED
```

```
FORTRAN IV G LEVEL  21              FLTRW            DATE = 73332        00/31/53            PAGE 0001

 0001                SUBROUTINE FLTRW(PERIOD,NUMBER,NNUMA,NNUMB,INV,S)
               C     THIS SUBROUTINE CONVERTS THE WOW AND FLUTTER DATA CHANNEL FROM
               C     THE TIME DOMAIN TO THE FREQUENCY DOMAIN.  IT THEN MULTIPLIES
               C     THE RESULT BY A TRANSFER FUNCTION WHICH IS INPUT AS A QUOTENT
               C     OF TWO POLYNOMIALS.  THEN THE CHANNEL IS CONVERTED BACK TO THE
               C     TIME DOMAIN
 0002                COMMON SIGNAL(4096),WAF(4096)
 0003                COMPLEX WAF,XNUM,XDENOM,QUOTNT
 0004                DIMENSION M(3),INV(NNUMB),S(NNUMB),COFN(10),COFD(10)
 0005                DIMENSION QUOTNT(32)
 0006            200 FORMAT(6I1)
 0007            201 FORMAT(1X,10F12.7)
 0008            210 FORMAT(' FILTER SUBROUTINE FOR WAF CHANNEL')
 0009            221 FORMAT(' ARRAY BEFORE DATA IS REARRANGED PROPERLY')
 0010            222 FORMAT('  DATA HAS NOW BEEN REARRANGED IN PROPER CELLS OF ARRAY RE
                    1ADY FOR TAKING FOURIER TRANSFORM')
 0011            223 FORMAT('  OUTPUT OF FOURIER TRANSFORM')
 0012            224 FORMAT(' DATA AFTER MULTIPLICATION BY TRANSFER FUNCTION')
 0013            225 FORMAT(' DATA AFTER INVERSE TRANSFORMATION')
 0014            226 FORMAT(' DATA AFTER REPACKING TO OBTAIN 3 REAL ARRAYS')
 0015            230 FORMAT(2I3)
 0016            231 FORMAT(5E16.8)
 0017            232 FORMAT(1H ,2I3)
 0018            233 FORMAT(1H ,5F16.8)
 0019                WRITE(3,210)
 0020                READ(1,200) IWRI1,IWRI2,IWRI3,IWRI4,IWRI5,IWRI6
 0021                IF(IWRI1.EQ.1) WRITE(3,221)
 0022                IF(IWRI1.EQ.1) WRITE(3,201) (WAF(IST),IST=1,NUMBER)
 0023                M(1) = NNUMA
 0024                M(2) = 0
 0025                M(3) = 0
               C     NOTE:: WAF ARRAY IS DECLARED COMPLEX, SO THERE ARE ONLY TWO
               C     ARRAYS INSTEAD OF THREE IN THE COMMON SINCE THE COMPLEX ARRAY
               C     IS THE SAME SIZE AS TWO STANDARD REAL ARRAYS.  HOWEVER, THE DATA
               C     (ALL REAL DATA) IN THE COMPLEX ARRAY IS ONLY STORED IN THE LAST
               C     HALF OF THE ARRAY(IN BLTH REAL AND COMPLEX CELLS).   THEREFORE
               C     THE DATA MUST BE SPREAD OUT AMONG  THE REAL CELLS THROUGHOUT THE
               C     ARRAY
               C     AN ATTEMPT HAS BE MADE TO ACCOMPLISH THAT HERE.
 0026                L = NUMBER / 2
 0027                LL = L + 1
 0028                DO 250 IIA = 1, L
 0029                IIB = IIA + L
 0030                IIC = IIA * 2
 0031                IID = IIC - 1
 0032                G = REAL(WAF(IIB))
 0033                WAF(IID) = CMPLX(G,0.0)
 0034                G = AIMAG(WAF(IIB))
 0035            250 WAF(IIC) = CMPLX(G,0.0)
 0036                IF(IWRI2.EQ.1) WRITE(3,222)
 0037                IF(IWRI2.EQ.1) WRITE(3,201) (WAF(IST),IST=1,NUMBER)
               C     TAKE THE FOURIER TRANSFORM
 0038                IFSET = 1
 0039                CALL HARM(WAF,M,INV,S,IFSET,IFERR)
 0040                WRITE(3,17) IFERR
 0041             17 FORMAT('     IFERR =     ',I3)
```

```
FORTRAN IV G LEVEL  21                FLTRW              DATE = 73332          00/31/53                PAGE 0002

 0042              IF(IWRI3.EQ.1) WRITE(3,223)
 0043              IF(IWRI3.EQ.1) WRITE(3,201) (WAF(IST),IST=1,NUMBER)
 0044              DO 260 IIG = 1, 10
 0045              COFN(IIG) = 0.0
 0046          260 COFD(IIG) = 0.0
 0047              N = 0
 0048              NO = 0
             C     READ IN TRANSFER FUNCTION POLOYNOMIALS (2)
 0049              READ(1,230) IDGRN,IDGRD
 0050              READ(1,231) (COFN(IIF),IIF = 1, IDGRN),CONSN
 0051              READ(1,231) (COFD(IIG),IIG = 1, IDGRD), CONSD
 0052              WRITE(3,232) IDGRN, IDGRD
 0053              WRITE(3,1226)
 0054         1226 FORMAT('  TRANSFER POLYNOMIALS '/10X,'NUMERATOR ',30X,'DENOMINATOR
                  1')
 0055              DO 228 IABD = 1, 10
 0056              IABC = 11- IABD
 0057          228 WRITE(3,227)COFN(IABC),IABC,COFD(IABC),IABC
 0058          227 FORMAT(5X,E20.5 ,'*(S**',I2,')',6X,E20.5 ,'*(S**',I2,')')
 0059              WRITE(3,229) CONSN,CONSD
 0060          229 FORMAT(5X,F20.15,14X,F20.5)
             C     COMPUTE THE VALUE OF THE TRANSFER FUNCTION AT EACH FREQUENCY
 0061          255 N = N + 1
 0062              RFREQ = ((N-1)*6.2831856)/ (PERIOD * NUMBER)
 0063              RFREQ2 = RFREQ*RFREQ
 0064              AREAL = - COFN(10)
 0065              AREAL = RFREQ2 * AREAL + COFN(8)
 0066              AREAL = RFREQ2 * AREAL - COFN(6)
 0067              AREAL = RFREQ2 * AREAL + COFN(4)
 0068              AREAL = RFREQ2 * AREAL - COFN(2)
 0069              AREAL = RFREQ2 * AREAL + CONSN
 0070              AIMIG = COFN(9)
 0071              AIMIG = RFREQ2 * AIMIG - COFN(7)
 0072              AIMIG = RFREQ2 * AIMIG + COFN(5)
 0073              AIMIG = RFREQ2 * AIMIG - COFN(3)
 0074              AIMIG = RFREQ2 * AIMIG + COFN(1)
 0075              AIMIG = AIMIG * RFREQ
 0076              XNUM = CMPLX(AREAL,AIMIG)
 0077              AREAL = -COFD(10)
 0078              AREAL = RFREQ2 * AREAL + COFD(8)
 0079              AREAL = RFREQ2 * AREAL - COFD(6)
 0080              AREAL = RFREQ2 * AREAL + COFD(4)
 0081              AREAL = RFREQ2 * AREAL - COFD(2)
 0082              AREAL = RFREQ2 * AREAL + CONSD
 0083              AIMIG = COFD(9)
 0084              AIMIG = RFREQ2 * AIMIG - COFD(7)
 0085              AIMIG = RFREQ2 * AIMIG + COFD(5)
 0086              AIMIG = RFREQ2 * AIMIG - COFD(3)
 0087              AIMIG = RFREQ2 * AIMIG + COFD(1)
 0088              AIMIG = AIMIG * RFREQ
 0089              XDENOM = CMPLX(AREAL,AIMIG)
 0090              NO = NO + 1
 0091              QUOTNT(NO) = XNUM / XDENOM
             C     MULTIPLY THE VALUE OF THE TRANSFER FUNCTION TIMES THE VALUE
             C     OF THE DATA SPECTRUM.
 0092              WAF(N) = WAF(N) * QUOTNT(NO)
```

```
0093            IF(N.EQ.1) GO TO 257
0094            NNO = NUMBER - N + 2
0095            IF(N.EQ.NNO) GO TO 257
0096            WAF(NNO) = CONJG(WAF(N))
0097        257 CONTINUE
0098            IF(NO.EQ.32) GO TO 265
0099            IF(N.GE.LL) GO TO 270
0100            GO TO 255
0101        265 CONTINUE
          C     WRITE(3,238) XNUM,XDENOM,RFREQ,RFREQ2
0102        238 FORMAT('  NUMERATOR =  ',2E16.6,'    DENOMINATOR =  ',2E16.6/'  F
               1REQUENCY =  ',F14.2,'   SQUARED =  ',F14.2)
          C     WRITE(3,240) (QUCTNT(NI),NI=1,32)
0103        240 FORMAT(1X,8E15.8)
          C     WRITE(3,239) N
0104        239 FORMAT(1H ,I5)
0105            NO = 0
0106            IF(N.GE.LL) GO TO 270
0107            GO TO 255
0108        270 CONTINUE
0109            IF(IWRI4.EQ.1) WRITE(3,224)
0110            IF(IWRI4.EQ.1) WRITE(3,201) (WAF(IST),IST=1,NUMBER)
          C    TAKE THE INVERSE TRANSFORM.
0111            IFSET = -1
0112            CALL HARM(WAF,M,INV,S,IFSET,IFERR)
0113            WRITE(3,17) IFERR
0114            IF(IWRI5.EQ.1) WRITE(3,225)
0115            IF(IWRI5.EQ.1) WRITE(3,201) (WAF(IST),IST=1,NUMBER)
          C    COMPRESS THE DATA BACK INTO ONE HALF THE ARRAY SO IT WILL BE IN
          C    THE PROPER FORM FOR A THREE ARRAY COMMON.
0116            DO 271 IIH = 1,L
0117            IIJ = L - IIH + 1
0118            IIM = L + IIJ
0119            IIK = IIJ * 2
0120            IIL = IIK - 1
0121            A = REAL(WAF(IIK))
0122            B = REAL(WAF(IIL))
0123        271 WAF(IIM) = CMPLX(B,A)
0124            IF(IWRI6.EQ.1) WRITE(3,226)
0125            LIST=L+1
0126            IF(IWRI6.EQ.1) WRITE(3,201) (WAF(IST),IST=LIST,NUMBER)
0127            RETURN
0128            END
```

```
FORTRAN IV G LEVEL  21                -PLTRW              DATE = 73332         00/31/53            PAGE 0004

                                COMMON BLOCK /          / MAP SIZE    C000
SYMBOL     LOCATION     SYMBOL     LOCATION      SYMBOL     LOCATION       SYMBOL     LOCATION       SYMBOL     LOCATION
SIGNAL         0        WAP          4000


                                SUBPROGRAMS CALLED
SYMBOL     LOCATION     SYMBOL     LOCATION      SYMBOL     LOCATION       SYMBOL     LOCATION       SYMBOL     LOCATION
IBCOM#        12C       HARM          130        CDVD#         134         CMPY#         138


                                SCALAR MAP
SYMBOL     LOCATION     SYMBOL     LOCATION      SYMBOL     LOCATION       SYMBOL     LOCATION       SYMBOL     LOCATION
XNUM          1B8       XDENOM        1C0        NNUMB         1C8         IWRI1         1CC         IWRI2         1D0
IWRI3         1D4       IWRI4         1D8        IWRI5         1DC         IWRI6         1E0         IST           1E4
NUMBER        1E8       NNUMA         1EC        L             1F0         IL            1F4         IIA           1F8
IIB           1FC       IIC           200        IID           204         G             208         IFSET         20C
IPERR         210       IIG           214        N             218         NO            21C         IDGRN         220
IDGRD         224       IIF           228        CONSN         22C         CONSD         230         IABD          234
IABC          238       RFREQ         23C        PERIOD        240         RFREQ2        244         AREAL         248
AIMIG         24C       NNO           250        IIH           254         IIJ           258         IIM           25C
IIK           260       IIL           264        A             268         B             26C         LIST          270


                                ARRAY MAP
SYMBOL     LOCATION     SYMBOL     LOCATION      SYMBOL     LOCATION       SYMBOL     LOCATION       SYMBOL     LOCATION
M             274       INV           280        S             284         COFN          288         COFD          2B0
QUOTNT        2D8


                                FORMAT STATEMENT MAP
SYMBOL     LOCATION     SYMBOL     LOCATION      SYMBOL     LOCATION       SYMBOL     LOCATION       SYMBOL     LOCATION
   200        3D8          201        3DE           210        3E7            221        40D            222        43A
   223        498          224        4B9           225        4EC            226        512            230        543
   231        549          232        550           233        559             17        563           1226        577
   227        5B0          229        5D4           238        5E0            240        63B            239        644

   *OPTIONS IN EFFECT*  ID,EBCDIC,SOURCE,NOLIST,DECK,LOAD,MAP
   *OPTIONS IN EFFECT*  NAME = PLTRW    , LINECNT =       58
   *STATISTICS*    SOURCE STATEMENTS =        128,PROGRAM SIZE =       4384
   *STATISTICS*  NO DIAGNOSTICS GENERATED
```

```
FORTRAN IV G LEVEL  21                    FLTRS              DATE = 73332          00/31/53               PAGE 0001

 0001              SUBROUTINE FLTRS(PERIOD,NUMBER,NNUMA,NNUMB,INV,S)
            C      THIS SUBROUTINE DOES THE SAME AS THE FLTRW SUBROUTINE ONLY IT
            C      OPERATES ON THE SIGNAL ARRAY INSTEAD OF THE WAF.
 0002              COMMON SIGNAL(4096),WAF(4096)
 0003              COMPLEX SIGNAL,XNUM,XDENOM,QUOTNT
 0004              DIMENSION M(3),INV(NNUMB),S(NNUMB),COFN(10)
 0005              DIMENSION COFD(10)
 0006              DIMENSION QUOTNT(32)
 0007          200 FORMAT(6I1)
 0008          201 FORMAT(1X,10F12.7)
 0009          210 FORMAT(' FILTER SUBROUTINE FOR SIGNAL CHANNEL ')
 0010          221 FORMAT(' ARRAY BEFORE DATA IS REARRANGED PROPERLY')
 0011          222 FORMAT('  DATA HAS NOW BEEN REARRANGED IN PROPER CELLS OF ARRAY RE
              1ADY FOR TAKING FOURIER TRANSFORM')
 0012          223 FORMAT('  OUTPUT OF FOURIER TRANSFORM')
 0013          224 FORMAT(' DATA AFTER MULTIPLICATION BY TRANSFER FUNCTION')
 0014          225 FORMAT(' DATA AFTER INVERSE TRANSFORMATION')
 0015          226 FORMAT(' DATA AFTER REPACKING TO OBTAIN 3 REAL ARRAYS')
 0016          230 FORMAT(2I3)
 0017          231 FORMAT(5E16.8)
 0018          232 FORMAT(1H ,2I3)
 0019          233 FORMAT(1H ,5F16.8)
 0020              WRITE(3,210)
 0021              READ(1,200) IWRI1,IWRI2,IWRI3,IWRI4,IWRI5,IWRI6
 0022              IF(IWRI1.EQ.1) WRITE(3,221)
 0023              IF(IWRI1.EQ.1) WRITE(3,201) (SIGNAL(IST),IST=1,NUMBER)
 0024              M(1) = NNUMA
 0025              M(2) = 0
 0026              M(3) = 0
            C      NOTE:  HERE AS IN THE FLTRW SUBROUTINE THE DATA MUST BE
            C      REDISTRIBUTED AMOUNG THE REAL AND COMPLEX CELLS OF THE SIGNAL
            C      ARRAY.  THE ONLY DIFFERENCE IS THAT THE DATA MUST BE SHIFTED FROM
            C      THE LOWER HALF OF THE ARRAY UPWARD INSTEAD OF DOWNWARD.
 0027              L = NUMBER / 2
 0028              LL = L + 1
 0029              DO 250 IIA = 1, L
 0030              IIB = L - IIA + 1
 0031              IIC = IIB * 2
 0032              IID = IIC - 1
 0033              G = REAL(SIGNAL(IIB))
 0034              H = AIMAG(SIGNAL(IIB))
 0035              SIGNAL(IID) = CMPLX(G,0.0)
 0036          250 SIGNAL(IIC) = CMPLX(H,0.0)
 0037              IF(IWRI2.EQ.1) WRITE(3,222)
 0038              IF(IWRI2.EQ.1) WRITE(3,201) (SIGNAL(IST),IST=1,NUMBER)
            C      TAKE THE FOURIER TRANSFORM
 0039              IFSET = 1
 0040              CALL HARM(SIGNAL,M,INV,S,IFSET,IFERR)
 0041              WRITE(3,17) IFERR
 0042           17 FORMAT('      IFERR =    ',I3)
 0043              IF(IWRI3.EQ.1) WRITE(3,223)
 0044              IF(IWRI3.EQ.1) WRITE(3,201) (SIGNAL(IST),IST=1,NUMBER)
 0045              DO 260 IIG = 1, 10
 0046              COFN(IIG) = 0.0
 0047          260 COFD(IIG) = 0.0
 0048              N = 0
```

```
0049            NO = 0
          C     READ IN TRANSFER FUNCTION POLOYNOMIALS (2)
0050            READ(1,230) IDGRN,IDGRD
0051            READ(1,231)  (COFN(IIF),IIF = 1, IDGRN),CONSN
0052            READ(1,231)  (COFD(IIG),IIG = 1, IDGRD), CONSD
0053            WRITE(3,232) IDGRN, IDGRD
0054            WRITE(3,1226)
0055       1226 FORMAT('  TRANSFER POLYNOMIALS '/10X,'NUMERATOR ',30X,'DENOMINATOR
               1')
0056            DO 228 IABD = 1, 10
0057            IABC = 11- IABD
0058        228 WRITE(3,227)COFN(IABC),IABC,COFD(IABC),IABC
0059        227 FORMAT(5X,E20.5 ,'*(S**',I2,')',6X,E20.5 ,'*(S**',I2,')')
0060            WRITE(3,229) CONSN,CONSD
0061        229 FORMAT(5X,F20.15,14X,F20.5)
          C     COMPUTE THE VALUS OF THE TRANSFER FUNCTION AT EACH FRFQUENCY.
0062        255 N = N + 1
0063            RFREQ = ((N-1)*6.2831856)/ (PERIOD * NUMBER)
0064            RFREQ2 = RFREQ*RFREQ
0065            AREAL = - COFN(10)
0066            AREAL = RFREQ2 * AREAL + COFN(8)
0067            AREAL = RFREQ2 * AREAL - COFN(6)
0068            AREAL = RFREQ2 * AREAL + COFN(4)
0069            AREAL = RFREQ2 * AREAL - COFN(2)
0070            AREAL = RFREQ2 * AREAL + CONSN
0071            AIMIG = COFN(9)
0072            AIMIG = RFREQ2 * AIMIG - COFN(7)
0073            AIMIG = RFREQ2 * AIMIG + COFN(5)
0074            AIMIG = RFREQ2 * AIMIG - COFN(3)
0075            AIMIG = RFREQ2 * AIMIG + COFN(1)
0076            AIMIG = AIMIG * RFREQ
0077            XNUM = CMPLX(AREAL,AIMIG)
0078            AREAL = -COFD(10)
0079            AREAL = RFREQ2 * AREAL + COFD(8)
0080            AREAL = RFREQ2 * AREAL - COFD(6)
0081            AREAL = RFREQ2 * AREAL + COFD(4)
0082            AREAL = RFREQ2 * AREAL - COFD(2)
0083            AREAL = RFREQ2 * AREAL + CONSD
0084            AIMIG = COFD(9)
0085            AIMIG = RFREQ2 * AIMIG - COFD(7)
0086            AIMIG = RFREQ2 * AIMIG + COFD(5)
0087            AIMIG = RFREQ2 * AIMIG - COFD(3)
0088            AIMIG = RFREQ2 * AIMIG + COFD(1)
0089            AIMIG = AIMIG * RFREQ
0090            XDENOM = CMPLX(AREAL,AIMIG)
0091            NO = NO + 1
0092            QUOTNT(NO) = XNUM / XDENOM
          C     MULTIPLY THE VALUE OF THE TRANSFER FUNCTION TIMES THE VALUF OF
          C     THE DATA SPECTRUM
0093            SIGNAL(N) = SIGNAL(N) * QUOTNT(NO)
0094            IF(N.EQ.1) GO TO 257
0095            NNO = NUMBER - N + 2
0096            IF(N.EQ.NNO) GO TC 257
0097            SIGNAL(NNO) = CONJG(SIGNAL(N))
0098        257 CONTINUE
0099            IF(NO.EQ.32) GO TO 265
```

```
FORTRAN IV G LEVEL  21                 FLTRS              DATE = 73332         00/31/53              PAGE 0003

 0100                 IF(N.GE.LL) GO TO 270
 0101                 GO TO 255
 0102             265 CONTINUE
                C     WRITE(3,240) (QUCTNT(NI),NI=1,32)
 0103             240 FORMAT(1X,8E15.8)
                C     WRITE(3,239) N
 0104             239 FORMAT(1H ,I5)
 0105                 NO = 0
 0106                 IF(N.GE.LL) GO TC 270
 0107                 GO TO 255
 0108             270 CONTINUE
 0109                 IF(IWRI4.EQ.1) WRITE(3,224)
 0110                 IF(IWRI4.EQ.1) WRITE(3,201) (SIGNAL(IST),IST=1,NUMBER)
                C    TAKE THE INVERSE FOURIER TRANSFORM
 0111                 IFSET = -1
 0112                 CALL HARM(SIGNAL,M,INV,S,IFSET,IFERR)
 0113                 WRITE(3,17) IFERR
 0114                 IF(IWRI5.EQ.1) WRITE(3,225)
 0115                 IF(IWRI5.EQ.1) WRITE(3,201) (SIGNAL(IST),IST=1,NUMBER)
                C    COMPRESS THE DATA BACK INTO ONE HALF THE ARRAY SO IT WILL BE IN THE
                C    PROPER FORM FOR A THREE ARRAY COMMON.
 0116                 DO 271 IIH = 1,L
 0117                 IIJ = 2 * IIH
 0118                 IIK = IIJ  - 1
 0119                 A = REAL(SIGNAL(IIK))
 0120                 B = REAL(SIGNAL(IIJ))
 0121             271 SIGNAL(IIH) = CMPLX(A,B)
 0122                 IF(IWRI6.EQ.1) WRITE(3,226)
 0123                 IF(IWRI6.EQ.1) WRITE(3,201) (SIGNAL(IST),IST=1,L)
 0124                 RETURN
 0125                 END
```

COMMON BLOCK / / MAP SIZE C000

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
SIGNAL	0	WAF	8000						

SUBPROGRAMS CALLED

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
IBCOM#	12C	HARM	130	CDVD#	134	CMPY#	138		

SCALAR MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
XNUM	1B8	XDENOM	1C0	NNUMB	1C8	IWRI1	1CC	IWRI2	1D0
IWRI3	1D4	IWRI4	1D8	IWRI5	1DC	IWRI6	1E0	IST	1E4
NUMBER	1E8	NNUMA	1EC	L	1F0	LL	1F4	IIA	1F8
IIB	1FC	IIC	200	IID	204	G	208	H	20C
IFSET	210	IFERR	214	IIG	218	N	21C	NO	220
IDGRN	224	IDGRD	228	IIF	22C	CONSN	230	CONSD	234
IABD	238	IABC	23C	BFREQ	240	PERIOD	244	BFREQ2	248
AREAL	24C	AIMIG	250	NNO	254	IIH	258	IIJ	25C
IIK	260	A	264	B	268				

ARRAY MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
M	26C	INV	278	S	27C	COFN	280	COFD	2A8
QUOTNT	2D0								

FORMAT STATEMENT MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
200	3D0	201	3D6	210	3DF	221	409	222	436
223	494	224	4B5	225	4B8	226	50E	230	53F
231	545	232	54C	233	555	17	55F	1226	574
227	5AD	229	5D1	240	5DD	239	5E6		

```
*OPTIONS IN EFFECT*  ID,EBCDIC,SOURCE,NOLIST,DECK,LOAD,MAP
*OPTIONS IN EFFECT*  NAME = FLTRS    , LINECNT =       58
*STATISTICS*    SOURCE STATEMENTS =       125,PROGRAM SIZE =      4256
*STATISTICS*  NO DIAGNOSTICS GENERATED
```

```
FORTRAN IV G LEVEL  21                -MAIN              DATE = 73332         00/31/53              PAGE 0001

                   C
 0001                    SUBROUTINE PSD(DIN,NUMBER,NNUMA)
 0002                    COMMON SIGNAL(4096),A(4096),WAF(4096),SPEC(120)
 0003                    DIMENSION DATA(4100),INV(512),S(512),AMP(2048)
 0004                    DIMENSION DIN(4096)
 0005                210 FORMAT('                  1            2            3            4
                        1    5            6          7            8            9            10')
 0006                310 FORMAT(I10,10F11.5)
 0007                410 FORMAT(I10)
 0008                    DO 50 I=1,NUMBER
 0009                 50 DATA(I)=DIN(I)
 0010                    M=NNUMA-1
 0011                    CALL RHARM(DATA,M,INV,S,IFERR)
 0012                    WRITE(3,410) IFERR
 0013                    DO 100 J=1,NUMBER,2
 0014                    JP1=J+1
 0015                    J2=JP1/2
 0016                    AMP(J2)=SQRT(DATA(J)**2+DATA(JP1)**2)
 0017                100 CONTINUE
 0018                    WRITE(3,210)
 0019                    NWAVE=NUMBER/4
 0020                    NHALF=(NWAVE/20)+1
 0021                    DO 200 J=1,NHALF
 0022                    IBEG=10*J-9
 0023                    IEND=10*J
 0024                    IF(IEND.GT.J2) IEND=J2
 0025                    WRITE (3,310) IBEG,(AMP(K),K=IBEG,IEND)
 0026                200 CONTINUE
 0027                    SUM=0.0
 0028                    ND2=NUMBER/2
 0029                    DO 300 J=2,ND2
 0030                    SUM=SUM+AMP(J)*AMP(J)
 0031                300 CONTINUE
 0032                    WRITE(3,510) SUM
 0033                510 FORMAT(' VARIANCE EQUALS',F12.4)
 0034                    SUM=0.0
 0035                    ND8=NUMBER/8
 0036                    DO 500 J=2,ND8
 0037                    SUM=SUM+AMP(J)*AMP(J)
 0038                500 CONTINUE
 0039                    WRITE(3,510) SUM
 0040                    DO 400 J=1,120
 0041                    SPEC(J)=AMP(J)
 0042                400 CONTINUE
 0043                    RETURN
 0044                    END
```

```
FORTRAN IV G LEVEL  21                    PSD              DATE = 73332          00/31/53           PAGE 0002

                                  COMMON BLOCK /          / MAP SIZE   C1E0
  SYMBOL    LOCATION     SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION
  SIGNAL        0        A           4000        MAP         8000        SPEC        C000

                                  SUBPROGRAMS CALLED
  SYMBOL    LOCATION     SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION
  RHARM        CC        IBCOM#       D0         SQRT         D4

                                  SCALAR MAP
  SYMBOL    LOCATION     SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION
  I            F0        NUMBER       F4         M            F8         NNUMA        FC         IFERR       100
  J           104        JP1         108         J2          10C         NWAVE       110         NHALF       114
  IBEG        118        IEND        11C         K           120         SUM         124         ND2         128
  ND8         12C

                                  ARRAY MAP
  SYMBOL    LOCATION     SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION
  DATA        130        INV         4140        S           4940        AMP         5140        DIN         7140

                                  FORMAT STATEMENT MAP
  SYMBOL    LOCATION     SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION
    210       7144         310       71EE          410       71C7          510       71CB

     *OPTIONS IN EFFECT*  ID,EBCDIC,SOURCE,NOLIST,DECK,LOAD,MAP
     *OPTIONS IN EFFECT*  NAME = PSD      , LINECNT =       58
     *STATISTICS*    SOURCE STATEMENTS =          44,PROGRAM SIZE =      30124
     *STATISTICS*  NO DIAGNOSTICS GENERATED

     *STATISTICS*  NO DIAGNOSTICS THIS STEP 2
```

```
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED MAP,LIST
         DEFAULT OPTION(S) USED -  SIZE=(100352,16384)

                                              MODULE MAP

  CONTROL SECTION                        ENTRY

    NAME     ORIGIN  LENGTH                NAME     LOCATION    NAME     LOCATION    NAME     LOCATION    NAME     LOCATION
  MAIN           00    37D8
  FMOD         37D8    847E
  FLTRB        BC58   151F4
  FLTRA       20E50   151C4
  TRIGGR      36018    1374
  DELAYP      37390     D1E
  AMPL        380B0     364
  LIMIT       38418     6A0
  WRITE       38AB8     258
  FLTRW       38D10    1120
  FLTRS       39E30    10A0
  PSD         3AED0    75AC
  IHCSLOG *   42480     1B6
                                       ALOG10      42480    ALOG        42498
  IHCFMAXB*   42638      C9
                                       MAX1        42638    MIN1        4264E    AMAX1       42664    AMIN1       4267A
  IHCCSAS *   42708      C4
                                       CMPY#       42708    CDVD#       42722
  IHCSSCN *   427D0     1D9
                                       COS         427D0    SIN         427E8
  IHCLSCN *   429B0     26C
                                       DCOS        429B0    DSIN        429CA
  IHCFEXIT*   42C20      1C
                                       EXIT        42C20
  HARM    *   42C40    1474
  IHCECOMH*   440B8     F61
                                       IBCOM#      440B8    FDIOCS#     44174    INTSWTCH    44FFE
  IHCCOMH2*   45020     65D
                                       SEQDASD     45398
  RANDU   *   45680     15E
  IHCFCVTH*   457E0    119D
                                       ADCON#      457E0    FCVAOUTP    4588A    FCVLOUTP    4591A    FCVZOUTP    45A6A
                                       FCVIOUTP    45E18    FCVEOUTP    4631A    FCVCOUTP    46534    INT6SWCH    4681B
  IHCEFNTH*   46980     542
                                       ARITH#      46980    ADJSWTCH    46D1C
  IHCSATN2*   46EC8     1CB
                                       ATAN2       46EC8    ATAN        46EDC
  IHCEFIOS*   47098     EBC
                                       FIOCS#      47098    FIOCSBEP    4709E
  IHCFIOS2*   47F58     52E
  IHCFIXPI*   48488     14F
                                       FIXPI#      48488
  IHCERRM *   485D8     5D4
                                       ERRMON      485D8    IHCERRE     485F0
```

```
  NAME     ORIGIN  LENGTH           NAME    LOCATION    NAME    LOCATION    NAME    LOCATION    NAME    LOCATION

 IHCFMAXI*  48BB0      C9
                                   MAXO      48BB0    MINO       48BC6   AMAXO      48BDC   AMINO      48BF2
 RHARM   *  48C80     62C
 IHCSSQRT*  492B0     145
                                   SQRT      492B0
 IHCETRCH*  493F8     28E
                                   IHCTRCH   493F8    ERRTRA     49400
 IHCUATBL*  49688     638
 IHCUOPT *  49CC0     3A0
$BLANKCCM   4A060    C1E8

ENTRY ADDRESS       00
TOTAL LENGTH     56248

****USERPROG  DOES NOT EXIST BUT HAS BEEN ADDED TO DATA SET
```

APPENDIX E

STANDARD OPERATING PROCEDURES

As mentioned in Section 6.0, a detailed standard operating procedure for z-axis rack alignment was developed for review by CDA personnel. This appendix documents that procedure.

Abbreviations

SR 1 = Scanning Radiometer Number 1

SR 2 = Scanning Radiometer Number 2

IR = Infrared

VIS = Visible

F&W = Flutter and Wow

SRR = Scanning Radiometer Recorder

S/C = Spacecraft

MUX = Multiplexer

DEMUX = Demultiplexer

RT = Real Time (CDA Tape Recorder set to 60 ips)

ST = Slow Time (CDA Tape Recorder set to 30 ips)

A7 Control Panel = Z-Axis Compensation Rack main control panel

A7J3, A7J4, ..., A7J16 = Connectors or test points 3, 4, ..., 16 on control panel A7

DVM = Digital Voltmeter (dc)

Note: The nomenclature RT F&W Discriminator #1, ST F&W Discriminator #2, etc. conforms to the RCA schematic for Rack 48 (last revision dated 5 December 1970). Connector designations are also based on this schematic.

List of Test Equipment

1 ea.	H-P 302 Wave Analyzer
1 ea.	H-P 240B Audio Oscillator
2 ea.	H-P 5245L Counter
1 ea.	H-P 3430A Digital Voltmeter (dc)
1 ea.	Ballentine 320 ac VTVM.

Z-Axis Alignment Standard Operating Procedures - Perform the following Daily Checks:

(1) F&W Discriminator Zero Balance

(2) Data Discriminator and VCO Balance

Initial Set-Up For Delay Adjustment - (This initial set-up should only have to be done once. If pre-pass checks start to vary, this operation should be redone after notifying Suitland)

1. ST Mode Reference Data (Figure E-1)
   a. Put test tape on recorder and set for ST/Ch A playback and 15 ips.
   b. Connect H-P 302A Wave Analyzer to DD#1 and tune to f_{fL} (check with audio osc. and frequency counter). Turn off all F&W discriminators.
   c. Adjust all delay line controls for minimum delay and set control panel for ST #1 operation.
   d. Playback about 3 min. of tape and record 302A reading. Rewind tape to its starting point.
   e. Tune HP 302A to f_{fH} (check with audio osc. and frequency counter) and repeat step 1.d.
   f. Connect HP 302A to DD #2 and repeat steps 1.b. through 1.e. with control panel set to ST #2 operation.
   g. Set tape recorder for ST/Ch B playback and 15 ips.
   h. Repeat steps 1.b. through 1.f.
2. ST Mode Alignment (Figure E-1 with F&W Discriminators turned on).
   a. Put test tape on recorder and set for ST/Ch A playback and 15 ips.
   b. Set control panel to ST #1 operation and all F&W discriminators turned on.
   c. Connect HP 302A tp DD #1 and tune to f_{fH} (check).
   d. Start tape recorder and adjust delay unit #1 until a minimum reading is obtained on the 302A. Record reading.
   e. Rewind test tape to its start point.
   f. Adjust 302A to f_{fL} and check.
   g. Start tape recorder and record 302A readings. Let tape play through to the end. Rewind tape.
   h. Set system for ST/Ch B playback.
   i. Repeat steps 1.a. through 1.f. but DO NOT ADJUST DELAY LINE (just record readings on 302A).

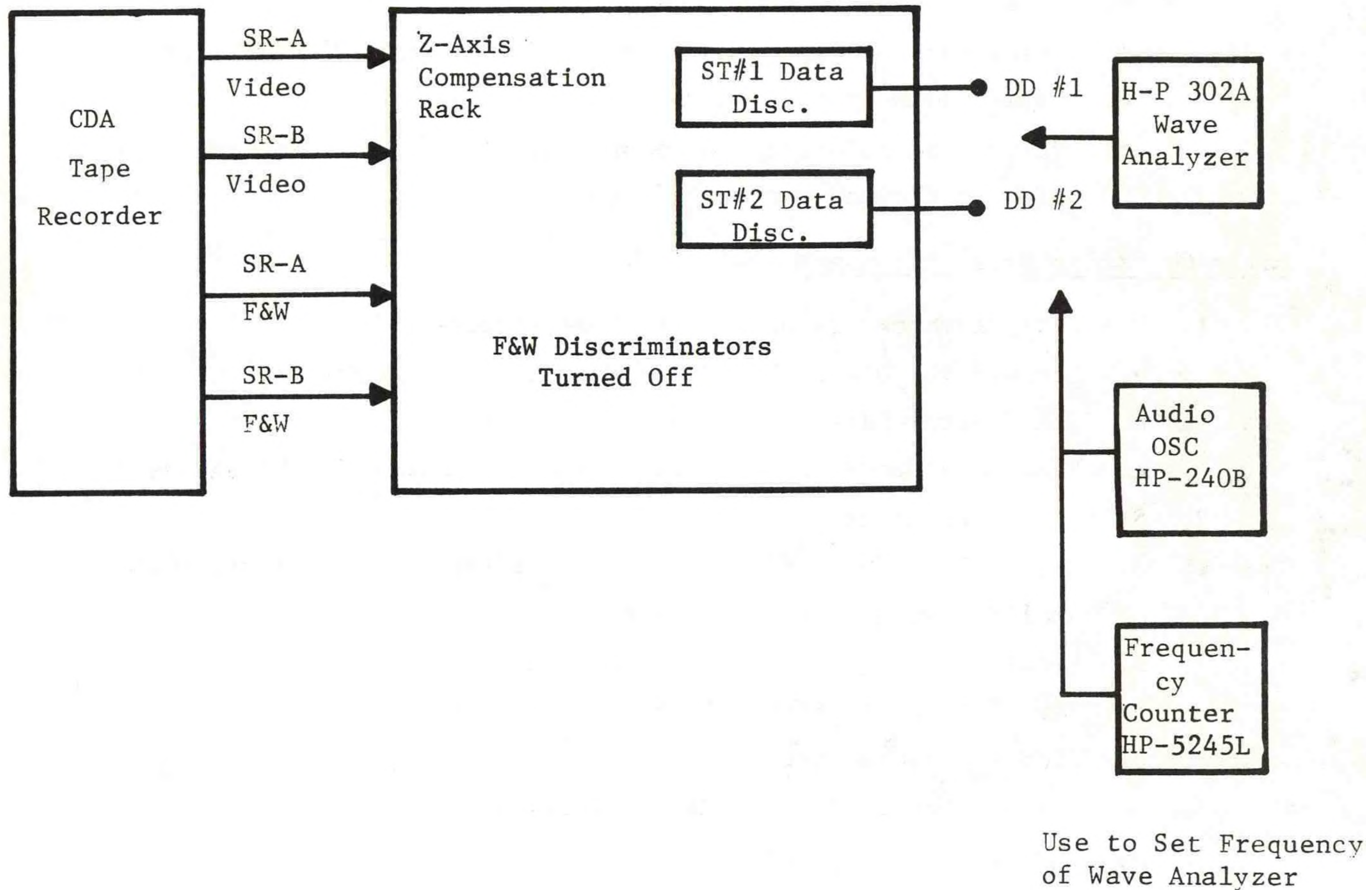


Figure E-1. Initial Set-up for Delay Adjustment.

j. Set system for ST/Ch A playback and 15 ips.

k. Set control panel to ST #2 operation and all F&W discriminators turned on.

l. Connect 302A to DD #2 and tune to f_{fH} (check).

m. Start tape recorder and adjust delay unit #2 until a minimum reading is obtained on the 302A. Record reading.

n. Repeat steps 1.d. through 1.g.

o. Tune 302A to f_{fH} and check.

p. Start tape recorder and record 302A reading. Rewind tape.

q. Adjust 302A to f_{fL} and check.

r. Start tape recorder and record 302A readings. Let tape play to the end and rewind tape.

Daily Check for Delay Adjustment

1. For data playback, connect as follows (Figure E-2).
   a. Connect HP 302A to DD #1 and tune to f_{fH} (check).
   b. Set Control Panel to ST #1/Ch A operation.
   c. Place test tape on tape recorder and set recorder to 15 ips.
2. Start tape recorder
   a. Adjust Delay Unit #1 for minimum reading on 302A (start with smallest delay increment, i.e. 25 μs).
   b. Record reading.
   c. Tune 302A to f_{fL} (check) and record reading.
   d. Stop and rewind tape.
3. Set Control Panel to ST #1/Ch B operation.
   a. Tune 302A to f_{fH} (check).
   b. Start tape recorder and record 302A reading. Stop tape.
   c. Tune 302A to f_{fL} (check).
   d. Start tape recorder and record 302A reading.
   e. Stop and rewind tape.
4. Set Control Panel to ST #2/Ch A operation.
   a. Connect 302A to DD #2 and tune to f_{fH} (check).
   b. Start tape and adjust Delay Unit #2 for minimum reading on 302A (start with smallest delay increment, i.e., 25 μs).
   c. Record reading.
   d. Tune 302A to f_{fL} (check) and record reading.
   e. Stop and rewind tape.

5. Set Control Panel to ST #2/Ch B operation.

   a. Repeat steps 3.a through 3.e.

6. Inform DDHS of any changes in Delay Unit Adjustments.

ST OPERATING MODE

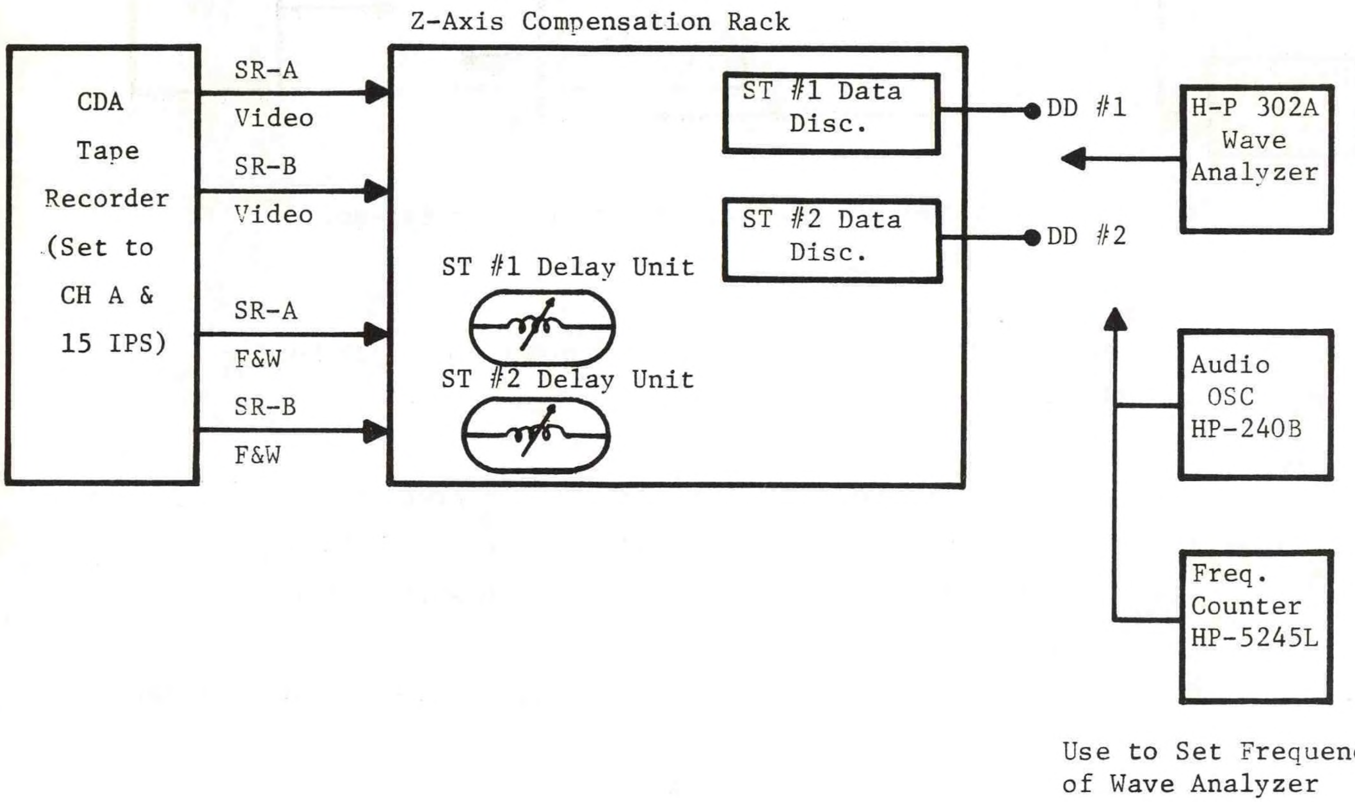


Figure E-2. Daily Check for Delay Adjustment.

OTHER DAILY CHECKS

1. F&W Discriminator Zero Balance (Figure E-3)

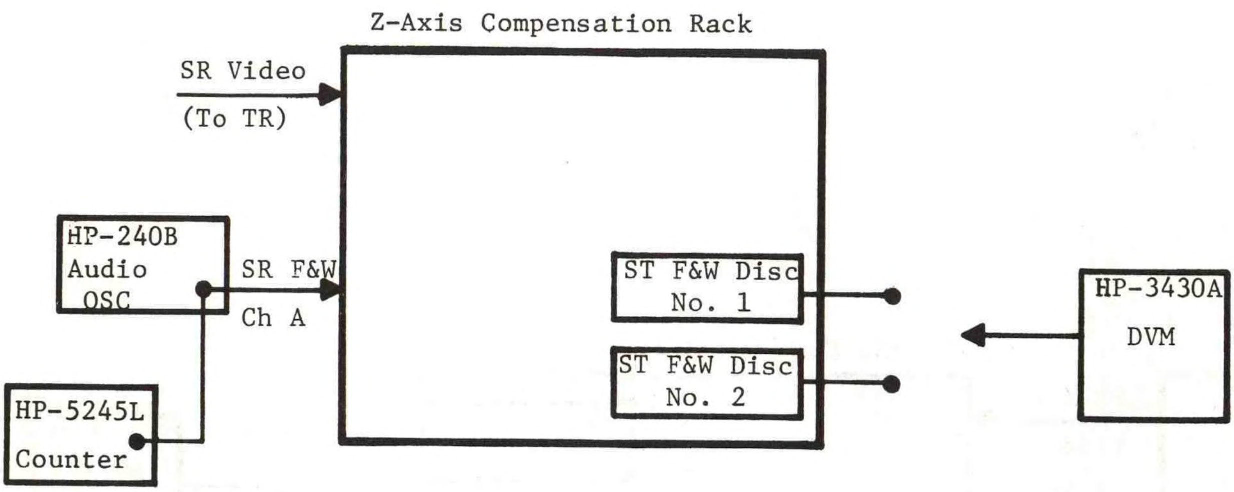


Figure E-3. F&W Discriminator Zero Balance Test Set-Up.

a. Set Audio Oscillator to 1 v. (rms) output and 6.25 KHz. $\pm$ 20 Hz.

b. Connect DVM to ST F&W Disc. No. 1.

c. Switch Control Panel to ST #1 Channel A operation.

d. Adjust BAL control on front panel of Model 287T Channel Selector Plug-In of ST #1 F&W Reference Discriminator for 0 $\pm$ 0.005 volts dc on DVM.

e. Set BANDEDGE VOLTS Control on RT #1 F&W Reference Discriminator fully clockwise.

f. Connect DVM to ST F&W Disc. No. 2.

g. Switch Control Panel to ST #2.

h. Repeat 1.d. and 1.e. for ST #2 F&W Reference Discriminator.

2. Data Discriminator and VCO Balance (Figure E-4)

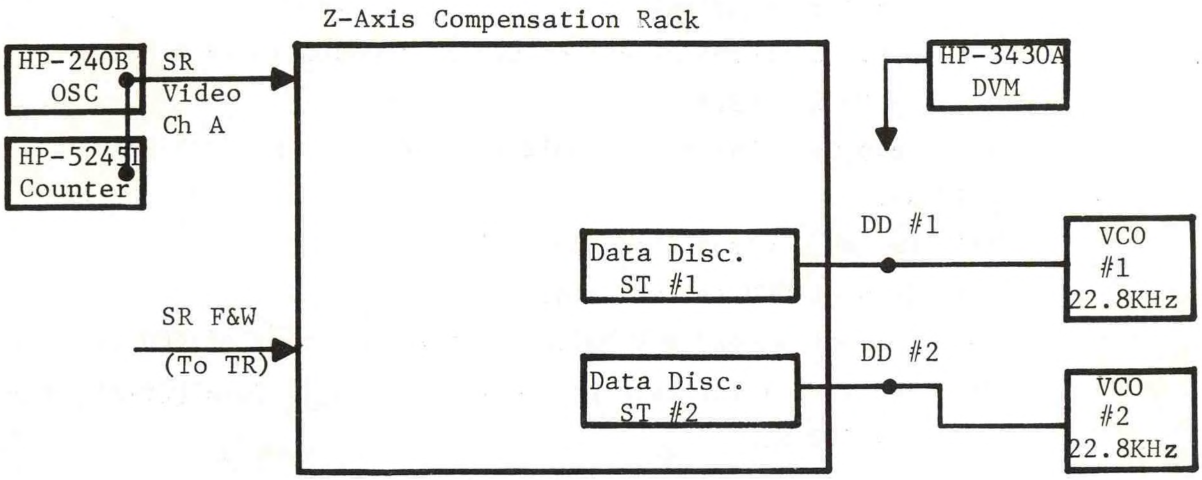


Figure E-4. Data Discriminator and VCO Balance Test Set-Up.

a. Turn ST #1 and ST #2 F&W Discriminators off. Use DVM on their outputs (see 1. for connectors) to verify 0 volts dc output.

b. Set Audio Oscillator to 1 volt (rms) output and 22.8 KHZ. ± 20 Hz.

(1) Switch Control Panel to ST #1.

(2) Connect DVM to DD #1.

(3) Connect second H-P 5245L Counter to output of VCO #1.

(4) Adjust BAL control of ST #1 Data Discriminator for 0 ± .005 volts dc on DVM.

(5) Adjust FREQ. Control of VCO #1 for a frequency output of 22.8 KHz. ± 20 Hz. on second counter.

(6) Set Audio Oscillator to 1 volt (rms) output and 26.4 KHz ± 20 Hz. (UBE)

(7) Adjust BANDEDGE VOLTS control on ST #1 Data Discriminator for + 5.0 volts dc on the DVM.

(8) Adjust INPUT control of VCO #1 for an output frequency of 26.4 KHz. ± 20 Hz. (UBE) on second counter.

(9) Set Audio Oscillator to 1 volt (rms) output and 19.2 KHz. ± 20 Hz. (LBE)

(10) Reading on DVM should be approximately -5.0 volts (record reading).

(11) Record frequency out of VCO #1 (should be 19.2 KHz. ± 20 Hz. LBE).

c. Set Audio Oscillator to 1 volt (rms) output and 22.8 KHz. ± 20 Hz.

(1) Switch Control Panel to ST #2.

(2) Connect DVM to connector DD #2.

(3) Connect second H-P 5245L counter to output of VCO #2.

(4) Repeat 2.b.(4) through 2.b.(11) on ST #2 Data Discriminator and VCO #2.

Master Test Tape Recording

1. In order to align the z-axis rack, it is required that only a single tone be recorded on the visible and/or IR tracks of the SRR. This means that the portion of the SR Processor which multiplexes the SR output with telemetry, step wedge, etc., data must be defeated.
2. The two equipments which will lead to the greatest time delay difference between the SR IR/Visible channel and the F&W channel are the spacecraft multiplexer and the ground-station demultiplexer. In recording the master tape, it is required that MUX, DEMUX and 14-track tape recorder be as similar as possible to the actual operational equipment.
3. The diagram in Figure E-5 indicates roughly how the tape is to be made. No Visible, Data or telemetry are shown since none of these signals are desired. No particular attention has been given to levels; however, attenuators or amplifiers may be necessary. The four cables from the output of the DEMUX to the 14-track tape recorder should all be of equal length. If possible, a virgin tape should be used on the 14-track recorder and it should be completely filled (end-to-end on tracks 5, 7, 9, 11) with test tone data. The two most important points to remember about this test tape are

(a) Continuous tones must be recorded on tracks 5, 7, 9, 11 of the 14-track recorder. Thermal vacuum data are not sufficient since they are not constant in amplitude due to the time-multiplexed data on the SR video output.

(b) Short of changing the spacecraft connections to eliminate unwanted data, the system should be as close as possible to operational conditions.

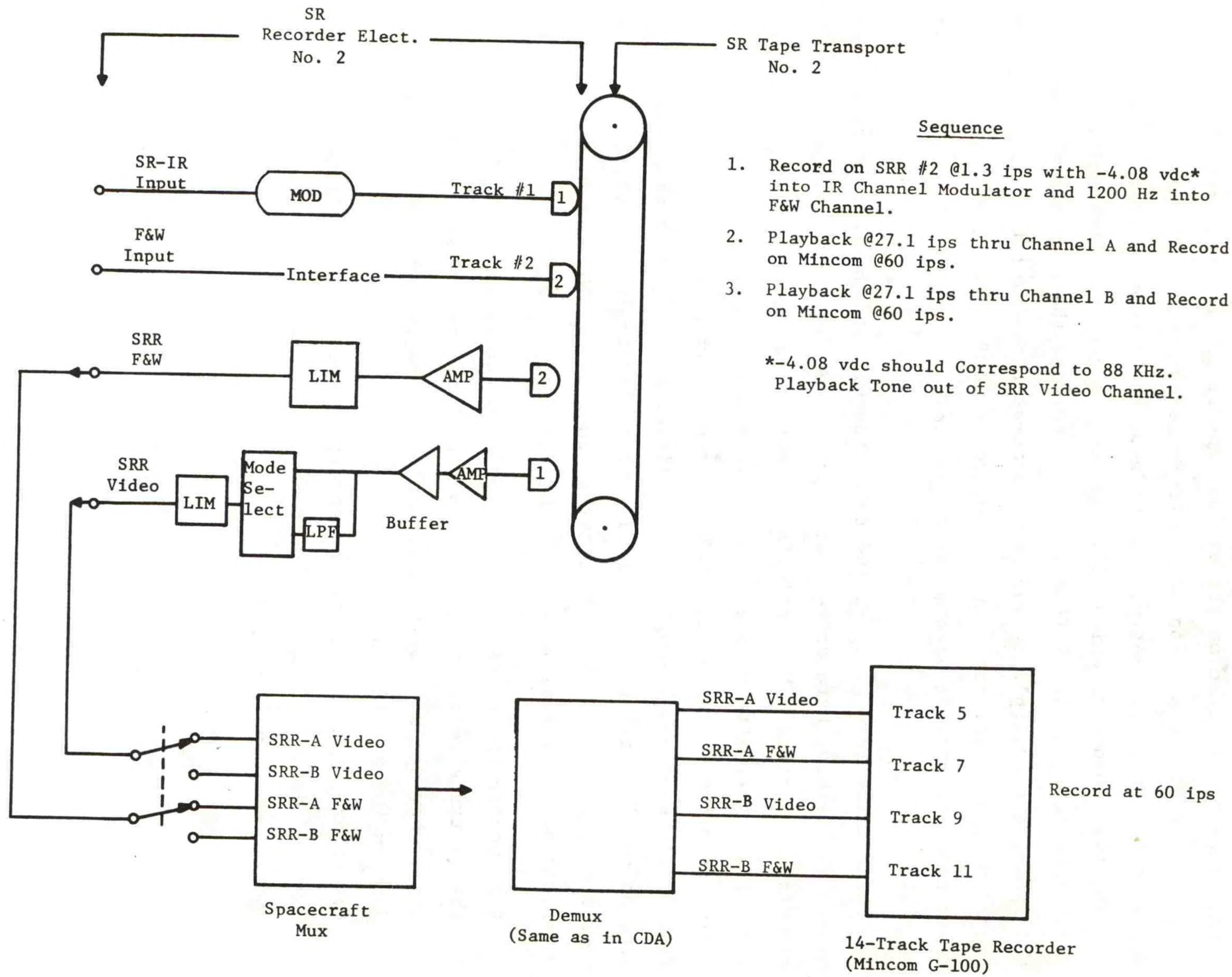


Figure E-5. Master Tape Recording Set-Up.

10.0 REFERENCES

2-1. Carlson, A. B.: Communication Systems - An Introduction to Signal and Noise in Electrical Communication. McGraw-Hill, 1968.

2-2. Black, H. S.: Modulation Theory. Van Nostrand, 1962.

3-1. EMR Model 287A-02 Subcarrier Discriminator (Revision E) - Instruction Manual, EMR Telemetry, April 1970.