AD-777 732

AN OCEAN WAVE MEASURING BUOY

SCRIPPS INSTITUTION OF OCEANOGRAPHY

PREPARED FOR
OFFICE OF NAVAL RESEARCH
ADVANCED RESEARCH PROJECTS AGENCY

JANUARY 1974

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
SIO Reference No. 74-6

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER
AD 777 732

4. TITLE (and Subtitle)
An Ocean Wave Measuring Buoy

5. TYPE OF REPORT & PERIOD COVERED
12/15/70-10/31/73
Special Technical

6. PERFORMING ORG. REPORT NUMBER
AOEL Report No. 56

7. AUTHOR(s)
Robert H. Stewart

8. CONTRACT OR GRANT NUMBER(s)
N00014-69-A-0200-6012

9. PERFORMING ORGANIZATION NAME AND ADDRESS
714-453-2000, X2476
Scripps Institution of Oceanography
University of California, San Diego
P.O. Box 1529, La Jolla, Ca. 92037

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
NR 294-016

11. CONTROLLING OFFICE NAME AND ADDRESS
Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, Virginia 22209

12. REPORT DATE
January 1974

13. NUMBER OF PAGES
64 pages

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)
Office of Naval Research
800 N. Quincy Street
Arlington, Virginia 22217

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this document is unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

DDC
RECEIVED
APR 25 1974
B

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Wave Buoy
Directional Spectra

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

We describe the design, construction, and performance of a surface wave-following buoy that can measure the directional spectrum of 0.06 to 0.5 Hz ocean waves with an accuracy of 10%, and an angular resolution of around 90°. The buoy is disc shaped, 1.5 m (5 feet) in diameter, weighs 150 kgm (300 lb), and is completely self-contained. Wave height is measured by an accelerometer mounted on the inner gimbal of a vertical gyro, wave slopes

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

Unclassified

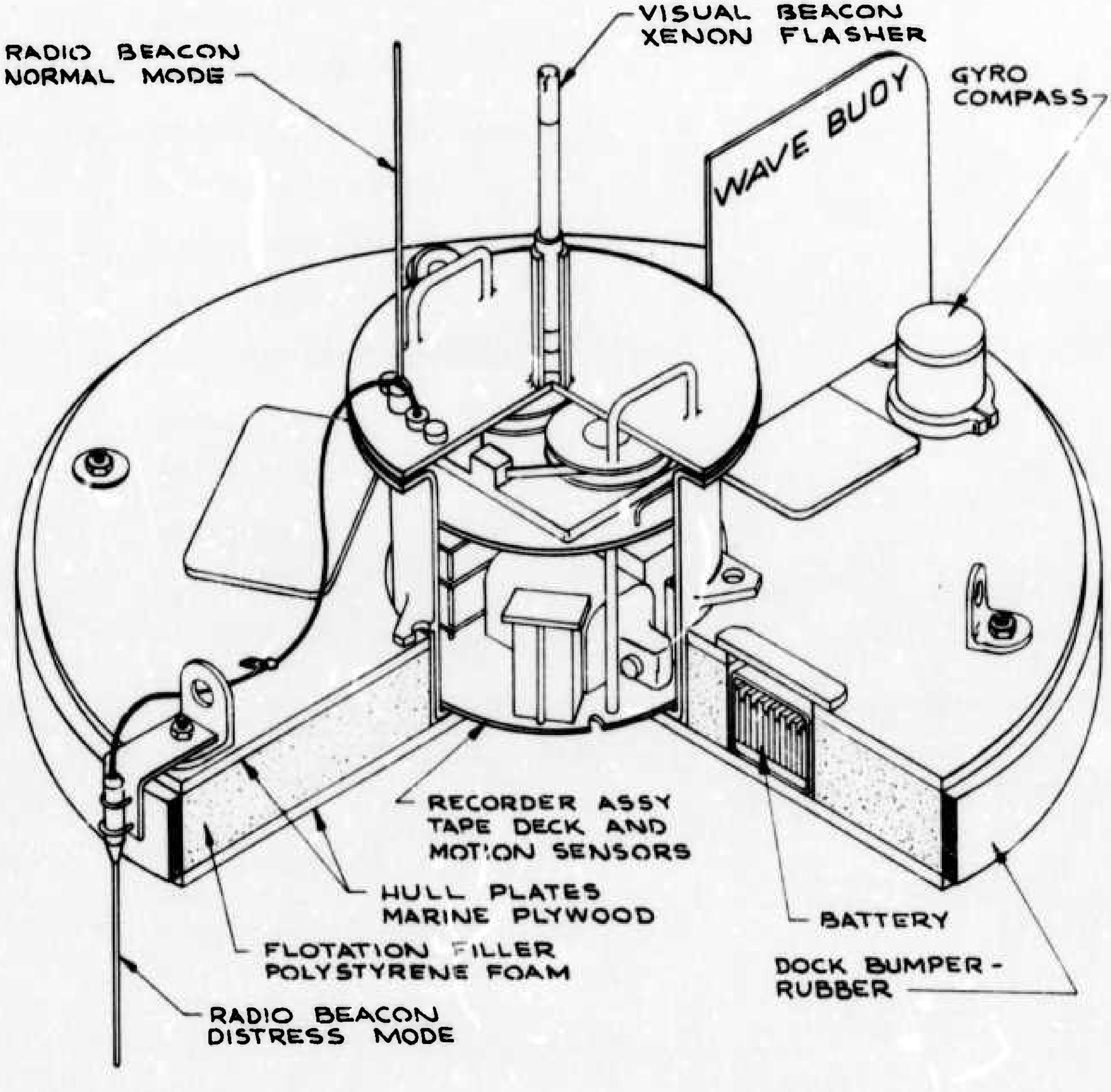
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

by the tilt of the buoy about the gyro's vertical axis, and buoy heading by a gyro stabilized compass. Data from the transducers are digitized and recorded on computer compatible magnetic tape. Wave spectra are calculated from the data by computer. The accuracy of the measurements is verified by calibration in a wave tank, and by the internal consistency of the data.

Disclaimer

*Amount of Contract:	$3,623,808.00
*Effective date of Contract:	December 15, 1968
*Contract Expiration Date:	December 31, 1973

*These quantities reflect the cumulative funding of all ARPA-sponsored activities of the Advanced Ocean Engineering Laboratory of Scripps Institution of Oceanography over this 5 year period.



● WAVE BUOY ●

CUTAWAY VIEW

FRONTISPIECE

ic

TABLE OF CONTENTS

AN OCEAN WAVE MEASURING BUOY

ABSTRACT

We describe the design, construction, and performance of a surface wave-following buoy that can measure the directional spectrum of 0.06 to 0.5 Hz ocean waves with an accuracy of 10%, and an angular resolution of around 90°. The buoy is disc shaped, 1.5 m (5 feet) in diameter, weighs 150 kgm (300 lb), and is completely self-contained. Wave height is measured by an accelerometer mounted on the inner gimbal of a vertical gyro, wave slopes by the tilt of the buoy about the gyro's vertical axis, and buoy heading by a gyro stabilized compass. Data from the transducers are digitized and recorded on computer compatible magnetic tape. Wave spectra are calculated from the data by computer. The accuracy of the measurements is verified by calibration in a wave tank, and by the internal consistency of the data.

1. INTRODUCTION

The wave measuring buoy described in this report was designed to provide directional spectra of ocean waves for comparison with radar scatter data. Initially we sought to purchase a buoy, but a quick survey of available instruments showed that none were capable of measuring a directional spectrum. Thus we were forced to design and construct one.

A number of techniques exist for determining the directional distribution of a wave field. Essentially, the measurement requires a coherent sample of the wave field over many wavelengths. The directional resolution is proportional to the number of wavelengths used. The measurement of the ocean wave field is particularly difficult because it is difficult to establish a fixed reference point to which wave heights can be related. Wave data can be measured at several places using an array on a stable platform, or, alternately, the surface elevation and two components of slope (tilt) can be measured at one point using an inertial reference.

A buoy that measures wave height and tilt (commonly called a pitch-and-roll buoy) is particularly simple, and can be small and easily handled. We have chosen this technique.

Several pitch-and-roll buoys have been built in the past. The first were designed and built by the National Institute of Oceanography in England. One was contained in a 5 feet 6 inch cast aluminum, ellipsoidal hull (Longuett-Higgins, Cartwright, and Smith, 1963). Later versions used 54" torus shaped hulls. Still later, another version was built by Hudson Laboratories of Columbia University (Saenger 1969a, b; Goldberg and Goldberg, 1969; Jordan, 1969). All of these buoys required a ship to stand by for recording data and supplying power. Our design borrows heavily from these previous instruments and is an extension of their concept: It is completely self contained and can operate unattended.

2. PRINCIPLE OF OPERATION

The hull of a pitch-and-roll wave measuring buoy is disc shaped, of shallow draft, and is radially symmetric about an axis normal to the water surface. Such a hull will follow the surface of a wave provided the wavelength is sufficiently long; and its high frequency response will be independent of wave direction. The buoy contains an accelerometer mounted on the axis of a vertical gyro and a gyro stabilized compass. The vertical component of acceleration, when doubly integrated, gives the sea surface height. The buoy tilts, when referenced to the vertical axis of the gyro and to the compass heading, gives the wave slopes in a fixed (North centered) coordinate system. These variables can be related to the ocean-wave directional spectrum and its lower-order moments.

Let

$$(\zeta_1, \zeta_2, \zeta_3) = (\zeta, \partial\zeta/\partial x, \partial\zeta/\partial y) \qquad (2.1)$$

be the wave height and slopes measured by the buoy as a function of time in a coordinate system with x,y pointing North and East (Fig. 1b). The spectrum $F_i(k_x,k_y)$ of $\zeta_i(x,y,t_o)$ at some instant t_o is so defined that

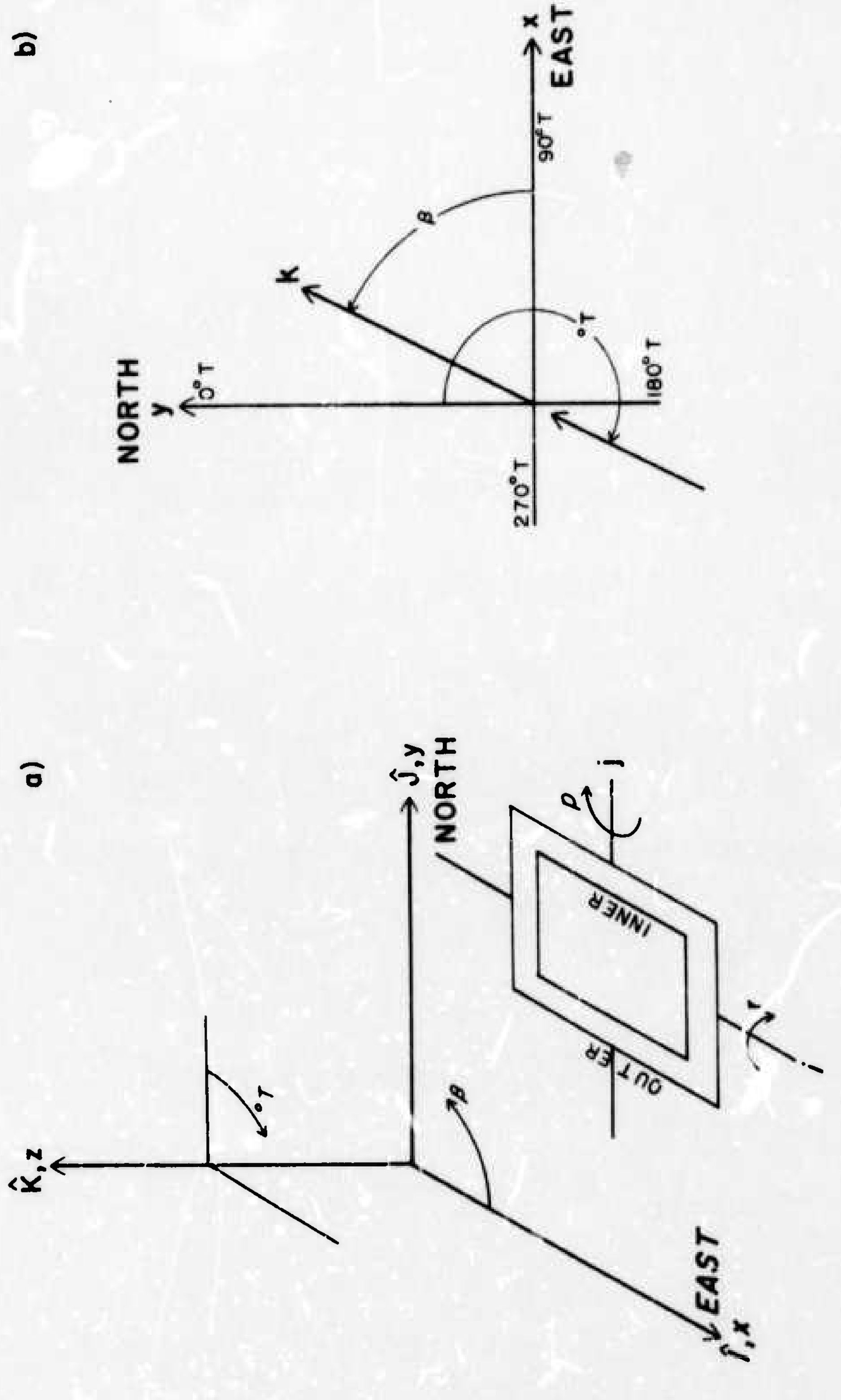


Figure 1. Coordinate system: a) of vertical gyro, b) for analysis of data.

$$\langle\zeta_i^2\rangle_{\text{space}} = \int\!\!\int_{-\infty}^{\infty} F_i(k_x,k_y)dk_x dk_y = \int_0^{\infty}\!\!\int_0^{2\pi} F_i(k,\beta)\, k\, dk\, d\beta \, . \qquad (2.2)$$

Similarly, for time series $\zeta_i(x,y;t)$ at one point $x_o y_o$,

$$\langle\zeta_i^2\rangle_{\text{time}} = \int_0^{\infty}\!\!\int_0^{2\pi} F_i(\omega,\beta)\, d\omega\, d\beta \, , \qquad (2.3)$$

where ω is a radian frequency, k is the wave number. By the ergodic theorem the space and time averages can be equated:

$$F_i(\omega,\beta)\, d\omega = F_i(k,\beta)\, k\, dk \, . \qquad (2.4)$$

For deep-water waves, $\omega^2 = gk$, where g is the acceleration of gravity, and the required Jacobian is

$$J \equiv \frac{F_i(k,\beta)}{F_i(\omega,\beta)} = \frac{1}{k}\frac{d}{dk} = \frac{g^2}{2\omega^3} \, . \qquad (2.5)$$

It is convenient to express all measured spectra in terms of $F_i(k_x,K_y)$, $F_i(\omega,\beta)$, etc., e.g. the contributions (per unit wavenumber space, per unit frequency-radian, etc.) to the mean-square surface elevation. We omit the subscript "i" and refer to F( ) as simply the wave spectrum.

The Cartesian moments are written

$$M_{pq} = \int\!\!\int_{-\infty}^{\infty} k_x^p\, k_y^q\, F(k_x,k_y)\, dk_x\, dk_y = \int_0^{\infty} k^{p+q}\, N_{pq}(k)\, k\, dk \qquad (2.6)$$

where

$$N_{pq}(k) = \int_0^{2\pi} \cos^p\beta \sin^q\beta \, F(\kappa,\beta) \, d\beta \, . \tag{2.7}$$

For an elementary wave train, the wave height and slope are

$$\begin{aligned} \zeta_1 &= \xi = R\, e^{i(k\cdot x-\omega t)} \\ \zeta_2 &= \partial_x\xi = R\, ik \cos\beta\, e^{i(k\cdot x-\omega t)} \\ \zeta_3 &= \partial_y\xi = R\, ik \sin\beta\, e^{i(k\cdot x-\omega t)} \, . \end{aligned} \tag{2.8}$$

The co-spectra C_{ij} and quadrature-spectra Q_{ij} of any pair of quantities ζ_i and ζ_j can be expressed in terms of the moments:

$$\begin{aligned} C_{11}(\omega) &= \int_0^{2\pi} F(\omega,\beta)\, d\beta &&= J^{-1} N_{00} \\ C_{22}(\omega) &= \int_0^{2\pi} k^2 \cos^2\beta\, F(\omega,\beta)\, d\beta &&= J^{-1} k^2 N_{20} \\ C_{33}(\omega) &= \int_0^{2\pi} k^2 \sin^2\beta\, F(\omega,\beta)\, d\beta &&= J^{-1} k^2 N_{02} \end{aligned} \tag{2.9}$$

$$\begin{aligned} C_{23} &= \int_0^{2\pi} k^2 \cos\beta \sin\beta\, F(\omega,\beta)\, d\beta &&= J^{-1} k^2 N_{11} \\ Q_{12} &= \int_0^{2\pi} k \cos\beta\, F(\omega,\beta)\, d\beta &&= J^{-1} k N_{10} \\ Q_{13} &= \int_0^{2\pi} k \sin\beta\, F(\omega,\beta)\, d\beta &&= J^{-1} k N_{01} \end{aligned}$$

and $C_{12} = 0, \; C_{13} = 0, \; Q_{23} = 0.$ (2.10)

Furthermore, a trigonometric identity gives

$$k^2 C_{11} = C_{22} + C_{33} \tag{2.11}$$

Equations (10) and (11) serve to estimate the accuracy of the buoy data.

The five moments determined by the buoy can serve to evaluate the first five Fourier terms of the directional distribution of ocean-wave energy (at each frequency)

$$F^{(5)}(k,\beta) = 1/2\, a_o + a_1\cos\beta + b_1\sin\beta + a_2\cos 2\beta + b_2\sin 2\beta$$

with

$$(a_o, a_1, b_1, a_2, b_2) = (N_{00}, N_{10}, N_{01}, N_{20} - N_{02}, 2N_{11}) \quad (2.12)$$

The terminated Fourier expansion can be expressed in the form (Longuett-Higgins, Cartwright, and Smith, 1963)

$$F^{(5)}(k,\beta) = \frac{1}{2\pi}\int_0^{2\pi} F(k,\beta)W(\beta'-\beta)\, d\beta' \quad (2.13)$$

where $W = 1 + 2\cos(\beta'-\beta) + 2\cos 2(\beta'-\beta)$ can be regarded as a weighting function associated with the buoy measurements. When $\beta'-\beta = 0, \pm 44°$, $W = 5, 5/2$ respectively. Thus the angular resolution of the tilt buoy can be taken at 88°. This is not very good, but it is the penalty one must pay for having a simple instrument.

3. DESIGN PARAMETERS

The wave buoy was to be used primarily in a trade-wind sea. To meet our requirements it must measure the ocean wave spectrum in the frequency band of 0.06 to 0.5 Hz (2-16 sec period) while wind speeds range up to 15 m/sec. Furthermore, it must be completely self-contained and operate for up to 24 hours unattended.

Typically, we expect to place it in operation and then conduct other experiments nearby. After three or four hours we would return and recover the buoy. The measured spectra should have an error of no more than 10% in amplitude or 10° in direction. These criteria determine the buoy size and transducer accuracy.

To estimate the accuracy required of the transducers using these criteria we assume an ocean wave spectrum proposed by Pierson and Moskowitz (1964)

$$S(\omega) = (\alpha g^2 \omega^{-5}) \exp[-\beta g^4 (v\omega)^{-4}] \qquad (3.1)$$

where

$$S(\omega) = \int_0^{2\pi} F(\omega,\beta)\, d\beta$$

and v is the mean wind velocity. We use $\alpha = 8.1 \times 10^{-3}$ and $\beta = 0.74$. The acceleration spectrum derived from (3.1) is:

$$S_A(\omega) = \alpha g^2 \omega^{-1} \exp[-\beta g^4 (v\omega)^{-4}] \qquad (3.2)$$

The root-mean-square acceleration is:

$$<\zeta_A>^{1/2} = (\int_0^{\infty} S_A(\omega)\, d\omega)^{1/2} \qquad (3.3)$$

This integral diverges because of the contributions at large ω. However, a buoy will respond only to waves whose frequency is less than some frequency, Ω; its acceleration is found by integrating (3.3) to this upper limit. This gives:

$$<\zeta_A>^{1/2} = (-\alpha g^2/4\ E_i[-\beta g^4 (v\Omega)^{-4}])^{1/2} \qquad (3.4)$$

Here E_i is the logarithmic integral defined by Jahnke and Emde (1945, p. 1). We expect the buoy to respond to waves whose wavelength is roughly twice the buoy diameter or greater (see the end of this section). Waves at the high frequency cut-off typically have a frequency of one Hertz. Letting $\Omega = 2\pi$/sec, $<\zeta_A>^{1/2} = 13\%$ g when v = 15 m/sec and 5% g when v = 2 m/sec. To measure these accelerations with an accuracy of 10% requires an accelerometer with a total error band of less than 0.5% g over a range of 0-2 g.

The directional spectrum (2.12) is calculated from the slope spectra and co-spectra. The accuracy is limited mainly by the coefficients a_2 and b_2 which depend on the two slope measurements. Each must be measured to an accuracy of $10\% \div \sqrt{2}$ if the two components, taken together, are to have a total error of less than 10% on average.

The root-mean-square slope is directly related to the acceleration spectrum through the dispersion relation:

$$<\zeta_2^2 + \zeta_3^2> \; = \; k^2<\zeta_1^2> \; = \; g^{-2}<\zeta_A^2> \tag{3.5}$$

The acceleration measured in units of g is the slope in radians. A 2 m/sec wind gives an RMS surface slope of 3°, a 15 m/sec wind gives 7°. A vertical gyro capable of measuring these angles to an accuracy of 7% must have an error of less than 0.2° over the band ±30°, the Stokes limit for a progressive wave of maximum amplitude.

The operation of the wave buoy requires a hull which accurately follows the water surface. One that is disc shaped and of shallow draft will perform well until the wavelength becomes less than about twice its own diameter. Since a 2 sec wave has a wavelength of 6.2 meters, the buoy should be somewhat smaller than three meters. On the other hand, it should be easily handled. As a compromise, we arbitrarily chose a hull 1.5 m (5 ft) in diameter, with a draft of about 8 cm. Such a buoy displaces 150 kgm. Hudson Laboratories' experience with smaller buoys indicated that a hull of this size will survive 15 m/sec winds with only a small chance of being upset (of course it would not be sunk).

We estimate the response of this size buoy hull from Kim (1966). From his figure 11, the half-power point in heave response occurs when $a = R\omega^2 g^{-1} = 1.5$, R is the radius of the buoy (0.75 m). This occurs at a frequency of 0.7 Hz. In a similar manner, the half-power point in the pitch response occurs when $a = 3.8$ (Kim's figure 12). This corresponds to a frequency of 1.0 Hz. The hull responds to wave slope slightly better than to wave height near the cut-off frequency. In either case, the response is within 10% of unity at frequencies of 0.5 Hz and lower, so the hull should meet our requirements.

4. BUOY CONSTRUCTION

The buoy consists of four main systems: 1) the transducers, 2) the signal conditioning and recording system, 3) safety and recovery system, and 4) the buoy hull that houses and supports this equipment. We will discuss each in order. The sources of supply and cost of equipment discussed below, together with detailed electrical schematics, are included in the appendix.

The transducers consist of a vertical gyro, an accelerometer mounted on the inner gimbal of the gyro, and a gyro stabilized magnetic compass.

The gyro-accelerometer assembly was made by Honeywell, and consists of a vertical gyro (part number JG7044A45) and a quartz-fiber accelerometer (GG326C1). The accelerometer has a range of 0 to 2 g, a total error band of 0.01% g, a drift in sensitivity of 0.02% g/°C, and in zero point of 4 x 10^{-6}g/°C. The gyro gimbal remains vertical to within ±0.25°, and has a full scale range of ±30°. The tilt of the buoy is measured with a resistance potentiometer having a 1% linearity. The entire assembly is 26 x 15 x 14 cm in size, weighs 4 kgm, and uses about 50 watts of power.

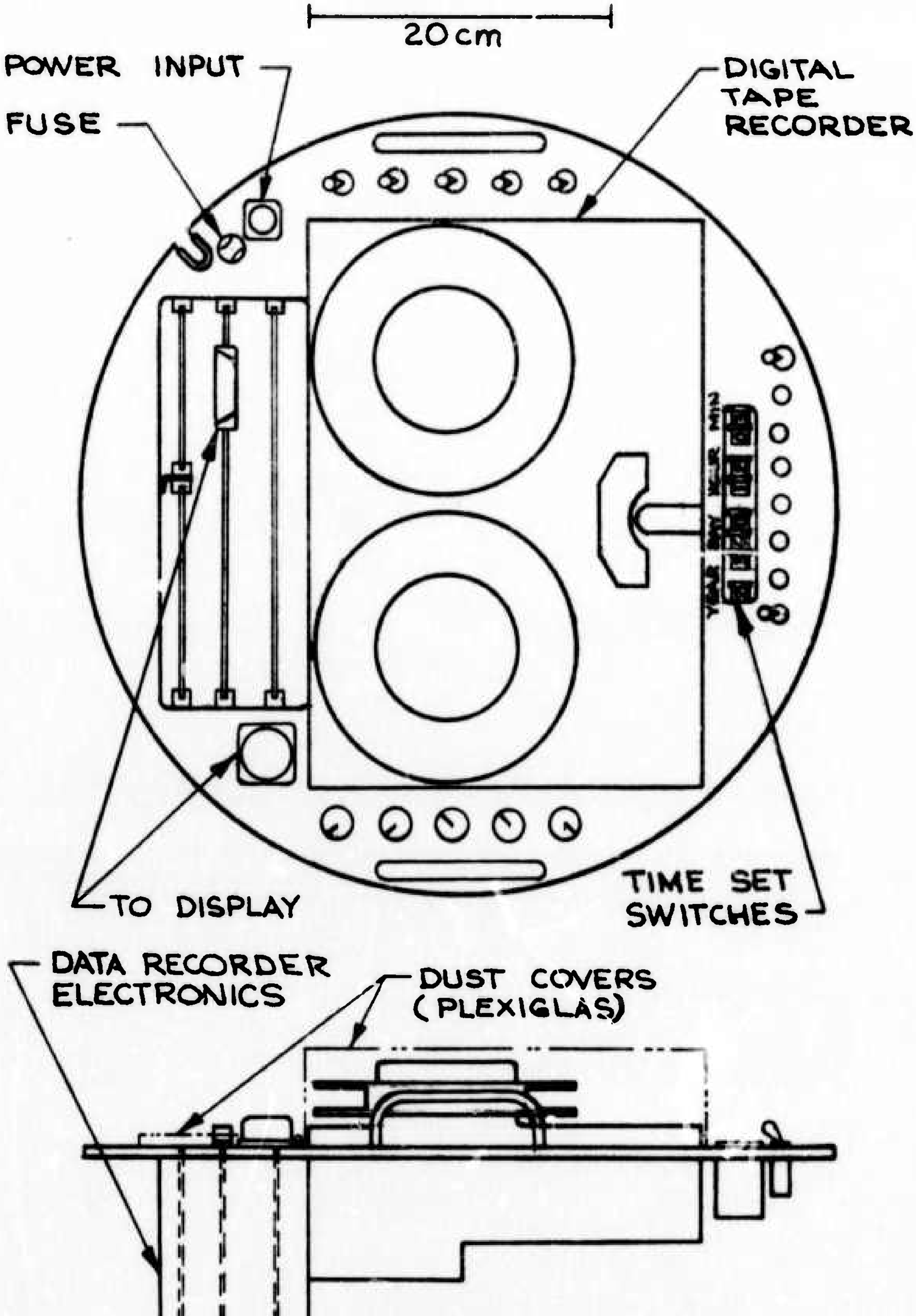
Buoy heading is measured by a Humphrey North Seeking Gyro (DG04-0122-1). This consists of a gyro stabilized and gimbaled magnet whose position is measured by a resistance potentiometer. The unit has an accuracy of ±1°, and is linear within ±1%. Physically, it is 21 x 8 x 8 cm, weighs 1 kgm, and uses 10 watts of power.

The recording system conditions the signals from the transducers, converts them to digital numbers, and records them on computer compatible magnetic tape. In addition, it controls the operation of the buoy, times its operation, and writes the time on the tape. The system was designed and built by Monitor Laboratories. The tape recorder is a standard unit manufactured by Precision Instruments (PI1387). The assembly is remarkably small (see figure 2), and its standby power requirement was less than two watts.

The signals from the transducers vary between ±5 volts, and contain some noise at higher frequencies, particularly at 400 Hz (the frequency of the gyro supplies). To reduce aliasing errors, the signals are sent through low-pass filters. The voltages from the potentiometers (tilts and heading) go through a simple RC filter with a cut-off frequency of 20 Hz. The accelerometer signal is more severly attenuated since it must be integrated twice to obtain sea surface heights. Even a small amount of aliased power could cause severe errors in the measurement of low-frequency wave heights. This signal goes through a two-pole filter with a cut-off frequency of 1.0 Hz. The complex filter response functions are:

$$L = (\omega^2 - 1)(\omega^4 + 1)^{-1} + j\sqrt{2}\,\omega(\omega^4 + 1)^{-1}$$

$$L = (\omega^2 + 1)^{-1} + j\omega(\omega^2 + 1)^{-1} \qquad (4.1)$$



• LOW POWER DATA RECORDER ASSY •

Figure 2.

where ω is frequency in Hz. Measured responses are very close to values calculated from these equations: within 4% in amplitude and 1/2° in phase at the half-power frequency.

The filtered signals are multiplexed, converted to 12 bit binary numbers (an accuracy of approximately ±0.01%), and recorded on 1/2" magnetic tape. The recording format is variable and controlled by switches. The correct time (year, day, hour, minute) is recorded at the beginning of each record. The time base was a quartz crystal having an accuracy of ±4 seconds/ day over a 50°C temperature range.

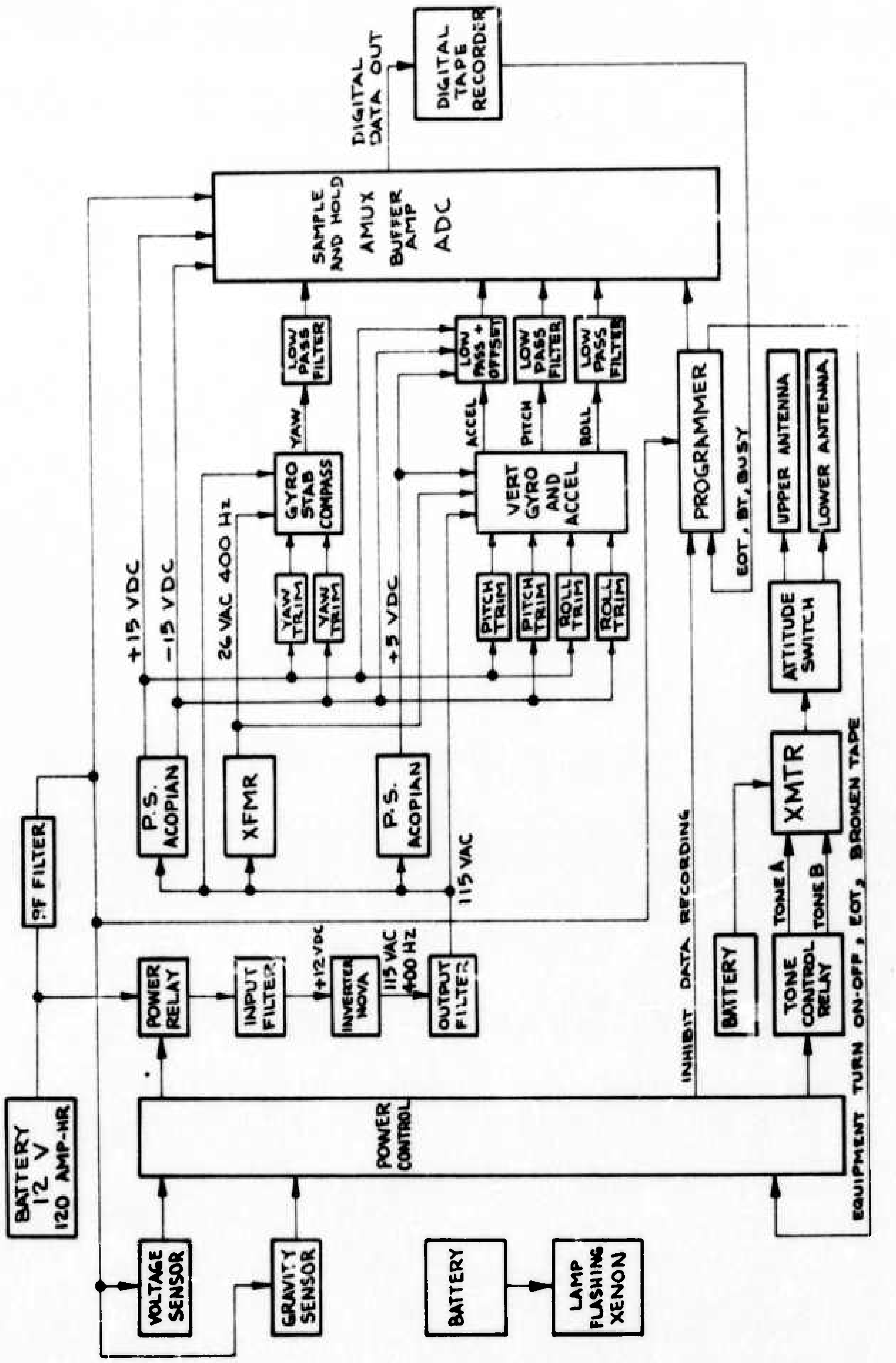
A 7 track, incremental tape recorder records the data at a density of 200 characters per inch and at a rate of up to 200 steps per second. It is rugged, can operate with 2 g RMS accelerations, uses one watt of power during standby and 40 watts while recording. The total amount of energy required to write 600 feet of tape is fixed at approximately 7 amp-hours from a 12 volt supply (72 joules). The rate of energy used is determined by the rate data is recorded. Typically, recording 144 bits/second uses 4 watts (144 bits/second = four 12-bit words, three times a second).

The recording system controls the buoy operation. The correct time is entered in the clock by switches. Once started, the system turns on power to the equipment 10 minutes before the next hour, and begins recording on the hour. Data are recorded in the format selected. After the selected amount has been written, recording stops and power is turned off. One, two, or four hours later the cycle repeats, thus allowing the buoy to operate unattended for a number of hours.

Typically, each transducer is sampled 3.125 times/sec; 2048 samples are written in each record; and 16 records comprise a file. The records contain 10 minutes of data, the file about 3 hrs. Each file uses about 9 meters (27 feet) of tape. Spectra calculated from this data have a resolution of 1.5×10^{-3} Hz and a Nyquist frequency of 1.6 Hz. This format ensures that low frequency waves are adequately resolved, aliasing errors are small, and the time series are easily handled in the computer.

Power for the operation of the gyros and recording system comes from sealed, gel-cell, lead-acid batteries. In normal operation the buoy uses 7 amperes at 12 volts. Twenty-four hours of continuous operation requires 168 amp-hours of energy, but intermittant operation reduces this considerably. The buoy has six 20 amp-hour batteries. They operate in any position, and need not be protected from sea water. Each battery is isolated by diodes: failure of one does not affect the operation of the others. The batteries weigh 48 kgm and are a major part of the buoy weight, but are small in size ($0.02\ m^3$).

The 12 VDC power is converted into other voltages required by the systems (see figure 3). The gyros operate from 115 and 26 VAC, 400 Hz. The first voltage is produced by a switched inverter (Nova). Switching transients occur on the 12 VDC lines and are reduced by the input and RF filters. The 26 VAC is derived from a transformer. A small amount of ±15 and +5



SIGNAL AND POWER FLOW CHART

Figure 3.

VDC is produced by two Acopian power supplies. This serves the potentiometers, accelerometers, and data recorder.

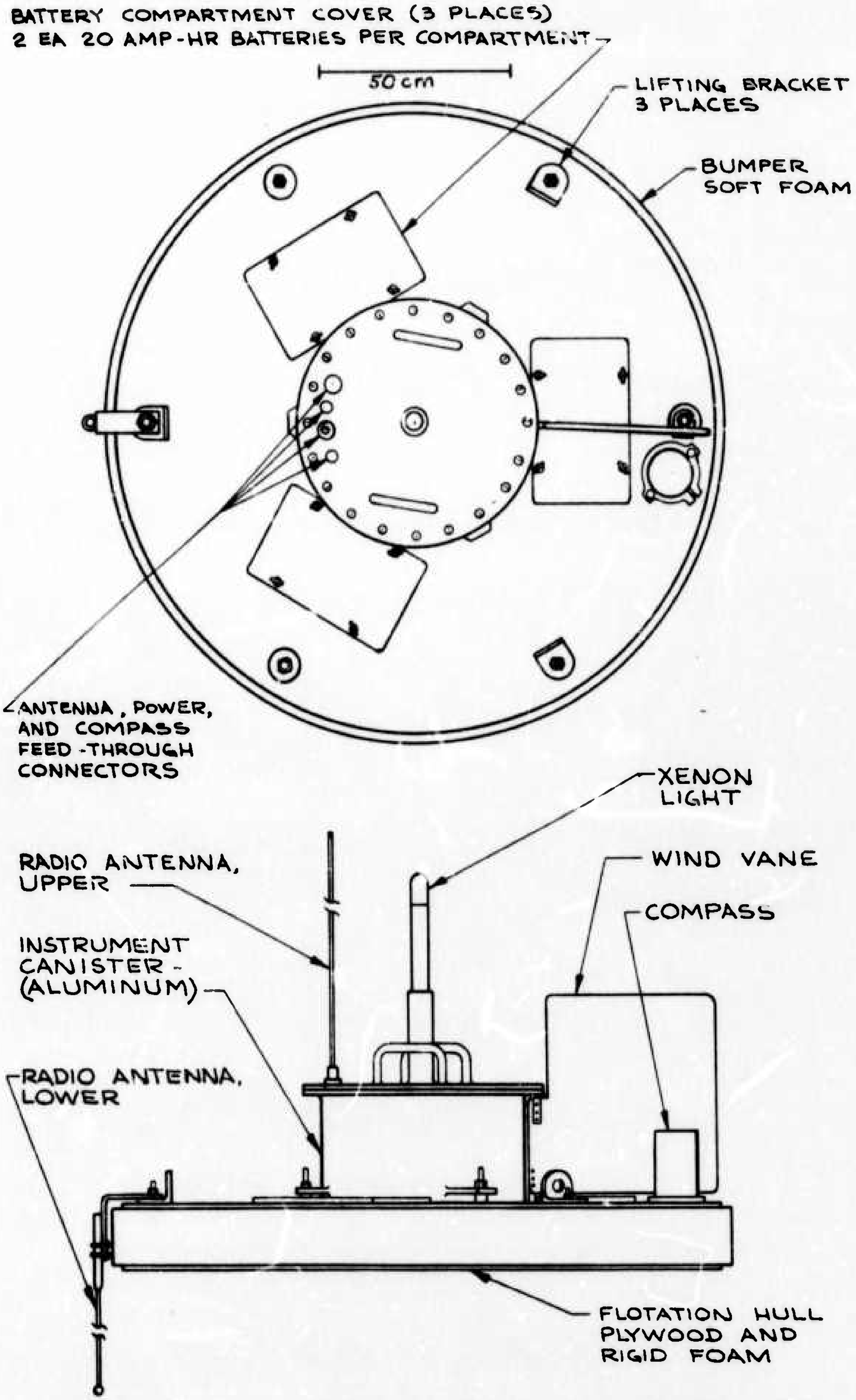
The buoy is designed to operate unattended, consequently it has systems that help in finding it once it is out of sight. It also recognizes and responds to certain dangerous conditions such as overturning in large seas. In this event, the gyros tumble and no data can be obtained. Other mishaps include loss of power (batteries discharged), broken magnetic tape, or end of tape. The occurrence of any of these events turns off the power to the gyros, terminates the recording of data, and causes the buoy to call for help on its radio.

To aid in finding the buoy, it is painted bright yellow, has a flashing zenon light, and a citizen band (27 MHz) radio transmitter. The buoy can be seen for several hundred meters in 2 meter seas. The flashing light can be seen for one kilometer at night. The radio can be heard from 10 km away. The transmitter sends out 500 m watt pulses and has two antennas, one on top and one on the bottom of the hull. In the event of a mishap the pulse rate doubles. If the buoy overturns, the bottom antenna is activated. The antennas are small (40 cm long) and are not likely to be damaged while the buoy is handled. Both the light and the radio have their own batteries; both can operate for two days.

In typical operation, the buoy is started, and placed in the water. Some time later it starts and data is recorded. It is difficult to stand around hoping all is well. To aleviate distress in the operators, the buoy has a small (100 m watt) citizen-band radio transmitter that transmitts everything it hears. The operators, several hundred yards away can hear the gyros turn on and spin up, and can hear the tape recorder step each time data is recorded. This verifies that all is well, and the operators are relieved.

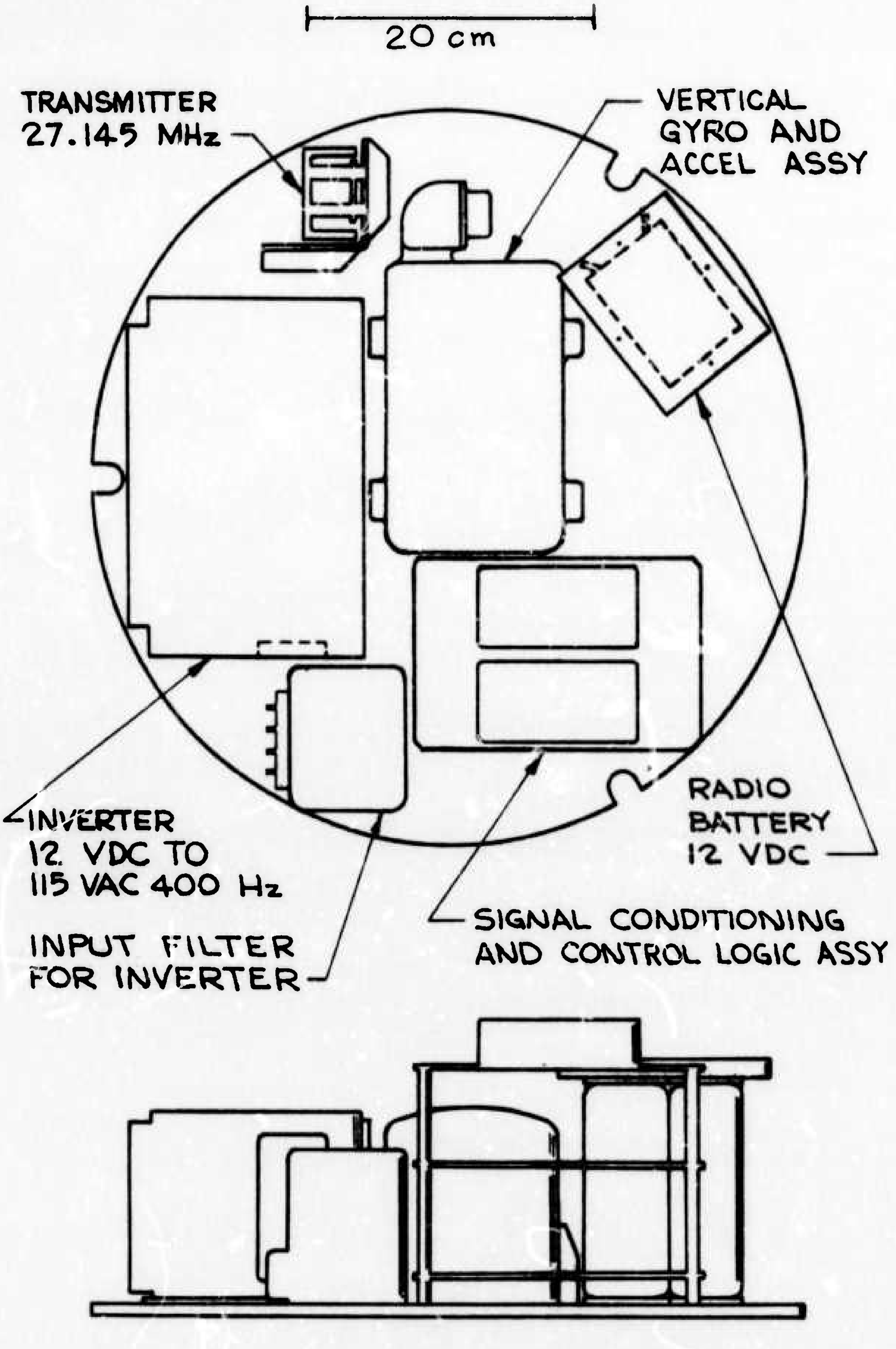
The general layout of the buoy hull is shown in the cut-away view (frontispiece) and figure 4. The control logic, vertical gyro, inverter, and transmitter are mounted together on a frame (figure 5). The data recorder (figure 2) mounts directly on top of this. The entire assembly is housed in a watertight aluminum can. The light and transmitting antenna are mounted on top of this. The north-seeking gyro (in a waterproof polyvinylchloride container) and the batteries mount on the buoy hull. Underwater type connectors connect them to the lid of the can. The container is 52 cm in diameter, 40 cm high, and weighs 60 kgm.

The hull is made of two, 1.5 meter (5 feet) diameter sheets of 1" marine plywood. Ten centimeters (4 in) of closed-cell styrofoam is sandwiched between them for floatation, and the three bolt together to form the hull. The aluminum can and gyro bolt into holes in the top piece of wood. The batteries are held below three hatch covers. All wiring is routed between the plywood sheets and protected. A small wind vane provides



● EXTERNAL BUOY CONFIGURATION ●

Figure 4.



• EQUIPMENT MOUNTING PLATE ASSY •

Figure 5.

further protection for the compass gyro. The edge of the hull has a soft neoprene bumper glued to the styrofoam.

The entire buoy assembly bolts together. The aluminum can is easily removed and carried into a protected laboratory for repairs, calibration, and setting switches for turning on the electronics. The electronic rack may be unplugged and removed. All electronic components, gyros, etc., plug in and can be removed for repair. Connections to the data recorder are by screws on a terminal strip. All internal wiring is laced together. Thus the entire assembly may be quickly and easily disassembled for repairs. In normal operation, the unit can be switched on, the lid to the aluminum can bolted on, the can bolted into the hull, and the buoy launched, all in 20 minutes, by three people.

A three-point bridle of spliced 3/4" nylon rope is attached to the buoy for launching. Usually, 30 meters of 3/4" polypropylene rope is attached to the bridle. When the buoy is in the water, this line floats on the surface upwind of the buoy. The heavy line is necessary to handle the buoy in rough seas. A smaller line (1/2") parted during one recovery.

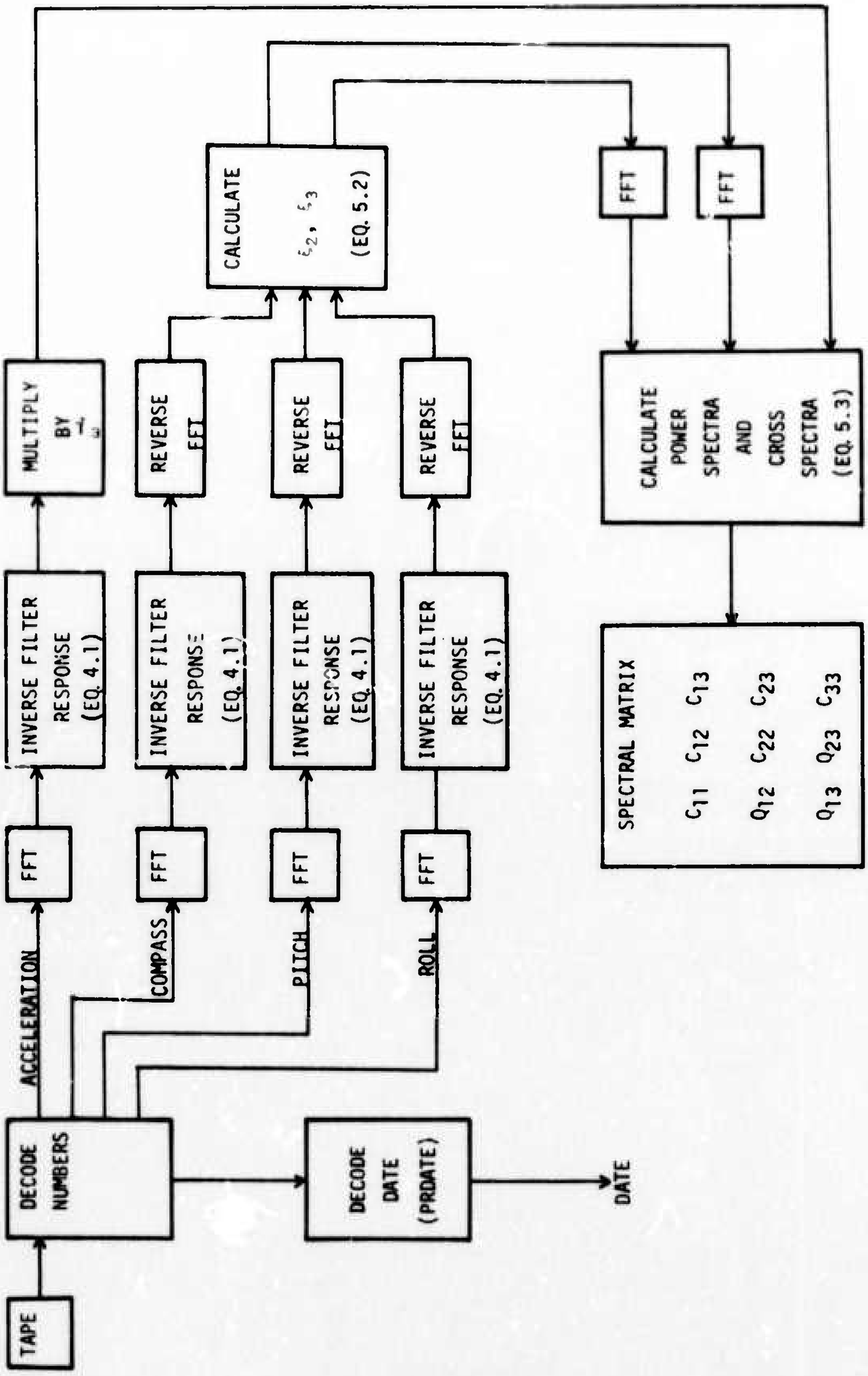
The aluminum can and wind vane are painted bright yellow (epoxy coating). The hull is bright red. English and Japanese lettering on the vane identifies the buoy. External fittings are stainless steel, external wiring is neoprene jacketed. This greatly reduces corrosion and electrical shorts to sea water.

5. DATA ANALYSIS

The buoy data are analysed on a Burroughs B6700 computer. A complete listing of the program is produced in the appendix. Although it appears long and complicated, it is, in fact, straightforward. Processing of data is quick, typically one hour of buoy data is analysed in 11 minutes at a cost of about $25. The following paragraphs outline the procedure, shown graphically in figure 6.

Data from the four sensors (acceleration, two tilts, heading) are written on magnetic tape as six-bit characters. The first 24 bits of each record is a time word. The procedure TAPESTART positions the tape at the record to be read, and notes any read errors in records passed over. The procedure READTAPE transfers into the computer the bits written in a particular record, converts them into time series with appropriate units, removes the mean, and notes any apparent errors. Procedure PRDATE decodes the first 24 bits into time and date. The four time series are transformed into the frequency domain (using the Fast Fourier Transform) by the computer's library routines FFT and BITRV2.

The frequency data are multiplied by the appropriate inverse filter function L^{-1} calculated from (4.1). Some data were recorded with sample-and-hold modules installed in the data recorder. In this case, all channels were sampled simultaneously. Other data were recorded without the modules and the chan-



DATA ANALYSIS FLOW CHART

Figure 6.

nels were sampled sequentially. This is equivalent to applying the linear filter:

$$L = \cos n\omega\tau + j\sin n\omega\tau \qquad (5.1)$$

where $n = 0,1,2,3$ is the order of sampling, and τ is the time delay between samples. When appropriate, the inverse of this filter was applied. Finally, the tilt and heading were transformed back into the time domain.

At this point, the slopes are relative to the buoy, and must be transformed into a geographic coordinate system. Let r be the rotation about the outer gimbal axis, p the rotation about the inner gimbal axis, and °T the angle measured clockwise from North to the inner gimbal axis in the coordinate system shown in figure 1a. The coordinate transformation is

$$\begin{aligned} \zeta_2 &= (-\sin{}^\circ T/\cos p)\tan r + \cos{}^\circ T \tan p \\ \zeta_3 &= (-\cos{}^\circ T/\cos p)\tan r - \sin{}^\circ T \tan p \end{aligned} \qquad (5.2)$$

These equations follow from Saenger's (1969) equations (9.6) for $\psi = 270°$ and $\psi = 0°$, and noting $m = -°T + 90°$.

The slope series are transformed back into the frequency domain and the appropriate Co- and Quadrature-spectra are computed from the Fourier coefficients of the data using:

$$\begin{aligned} 2C_{ij} &= a_i a_j - b_i b_j \\ 2Q_{ij} &= a_i b_j + a_j b_i \end{aligned} \qquad (5.3)$$

where a_i, b_i are the cosine and sine transformation of the first series, a_j, b_j are those of the second series. Note that the sign of Q_{ij} is arbitrary. Our definition agrees with (2.9), but is in disagreement with the convention used by some workers.

The spectral quantities (2.9) are used to calculate various parameters of the ocean-wave directional spectrum. The mean wave direction (at each frequency) is:

$$\tan \beta_o = N_{01}/N_{10} \tag{5.4}$$

The root-mean-square beamwidth can be calculated several ways; we use

$$\tan 2\gamma = \left(\frac{1 - R}{1 + R}\right)^{1/2} \tag{5.5}$$

where

$$R^2 = [(N_{20} - N_{02})^2 + 4N_{11}^2]/N_{00} \tag{5.6}$$

This agrees with equation (22) in Longuet-Higgins, Cartwright and Smith (1963).

6. BUOY RESPONSE TO WAVES

We have estimated the response of the buoy to waves by observing the internal consistency of data recorded at sea and by calibration in a wave tank. In general, the buoy responds as a damped harmonic oscillator with a 1/2 power point near

0.7 Hz. The accuracy of the data meets the design criteria specified in section 3.

Before describing the calibration of the buoy, we first give a gross overview of some typical data. They were obtained off Monterey, California in a 10 m/sec wind when the significant wave height ($4<\zeta^2>^{1/2}$) was 2.5 m. A short section of the digitized data from the four transducers (figure 7) indicate the general range of these variables. The root-mean-square acceleration was 10.3% g, and the slope was 6.8°. This is in very good agreement with the values of 10.7% g and 6.1° predicted by eqs. (3.4) and (3.5) using $\Omega = 0.7$ and $v = 10$ m/sec.

The first time the buoy was operated at sea we sampled the transducers at the maximum possible rate to determine the optimum rate for later operations. The Nyquist frequency (maximum frequency of the digital spectrum) was 6.25 Hz. The acceleration spectrum recorded on this occasion is shown in figure 8. The noise at low frequencies has a spectral power of around 10 $\mathrm{cm^2 sec^{-4} Hz^{-1}}$. The high-frequency noise is considerably lower, and the Nyquist frequency could be reduced considerably. Consequently, all later data were recorded with a Nyquist frequency at 1.5625 Hz (each signal was sampled 3.125 times per second).

The spectra derived from the Monterey data are plotted in figure 9. The Nyquist frequency is 1.5625 Hz, the resolution is 0.0015 Hz. The figure gives a good indication of the signal/noise ratio of the data. The acceleration values are approxi-

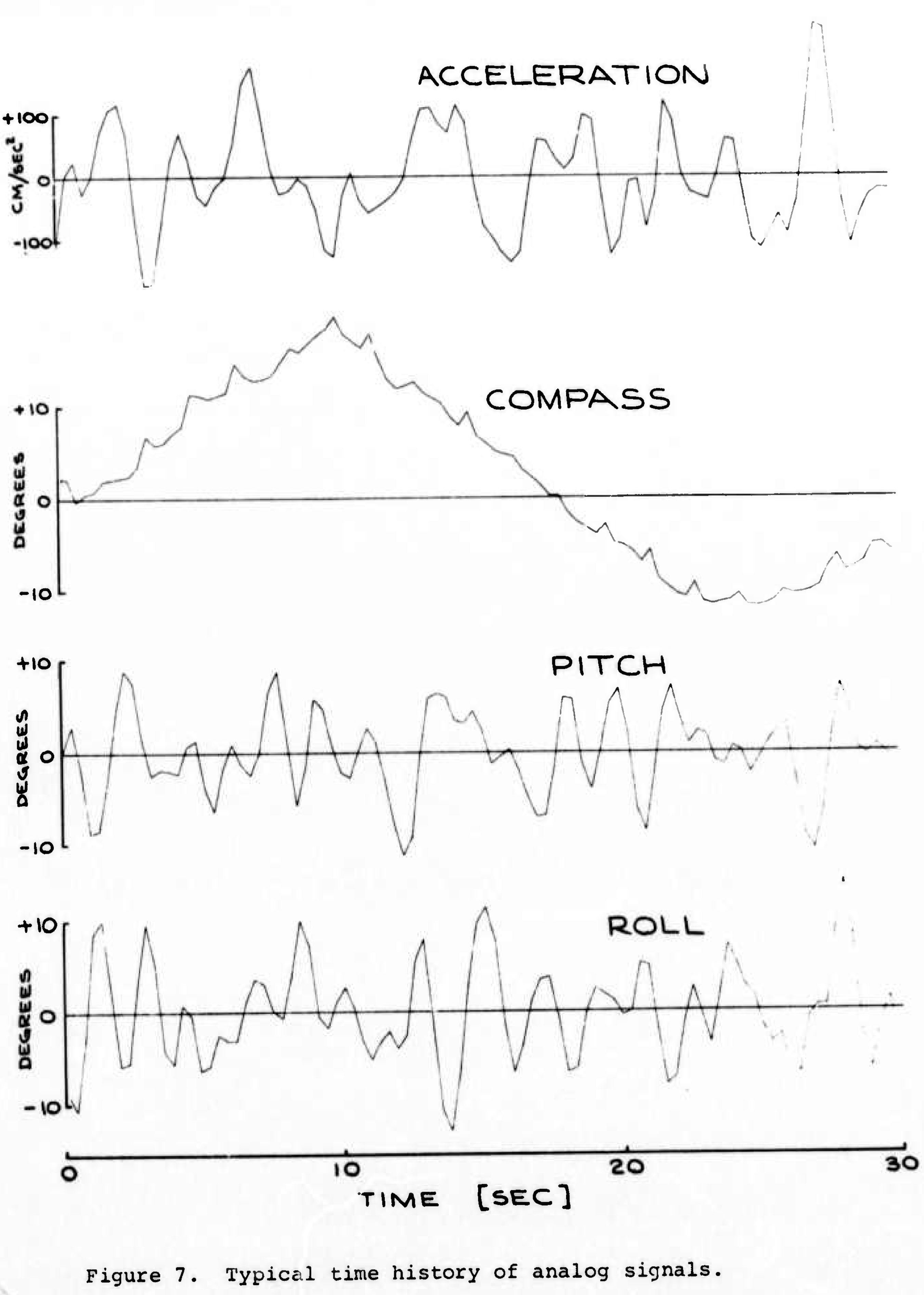


Figure 7. Typical time history of analog signals.

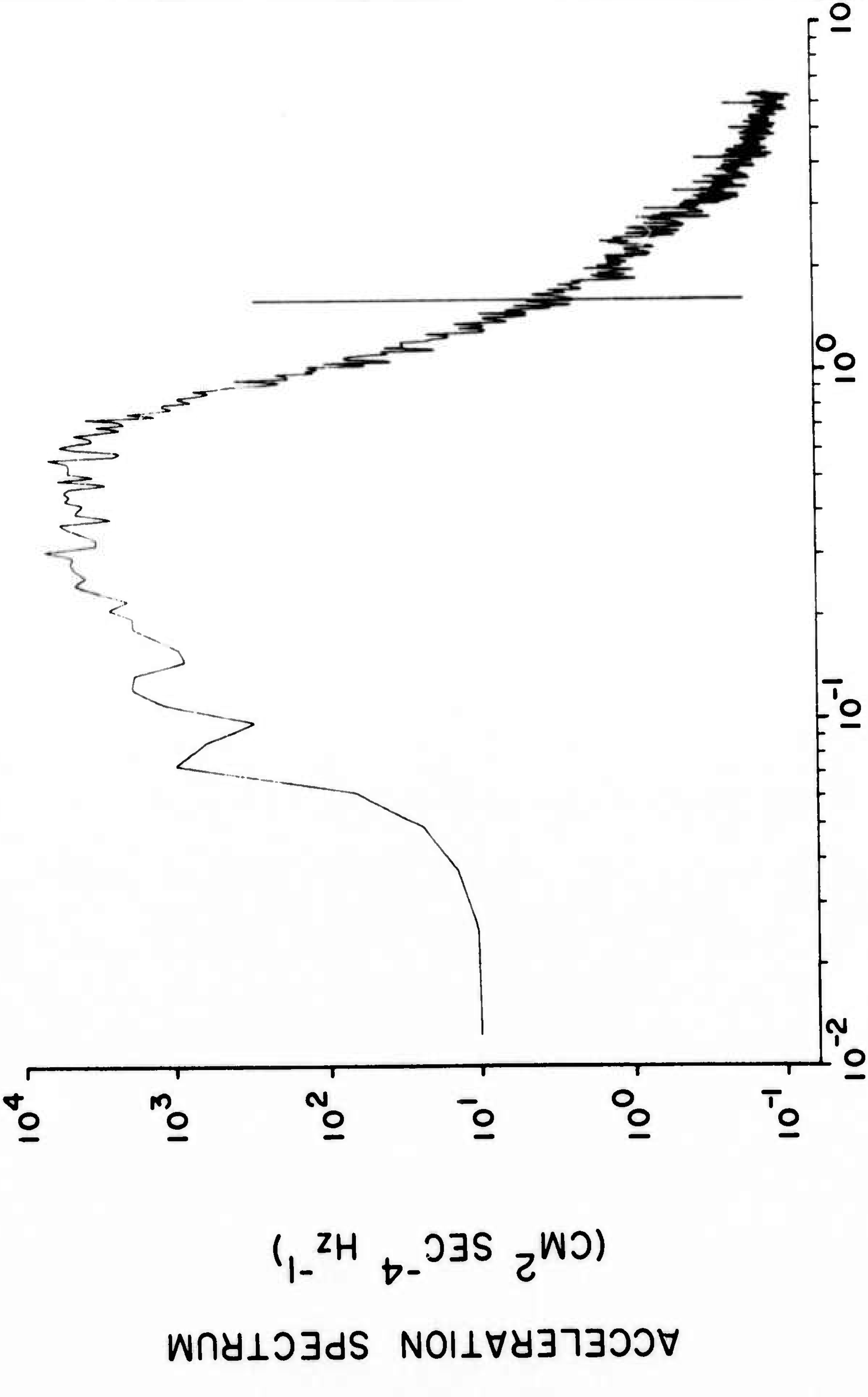


Figure 8. Acceleration spectrum with 6.25 Hz Nyquist frequency. Vertical line is at 1.5625, the Nyquist frequency used for subsequent recordings.

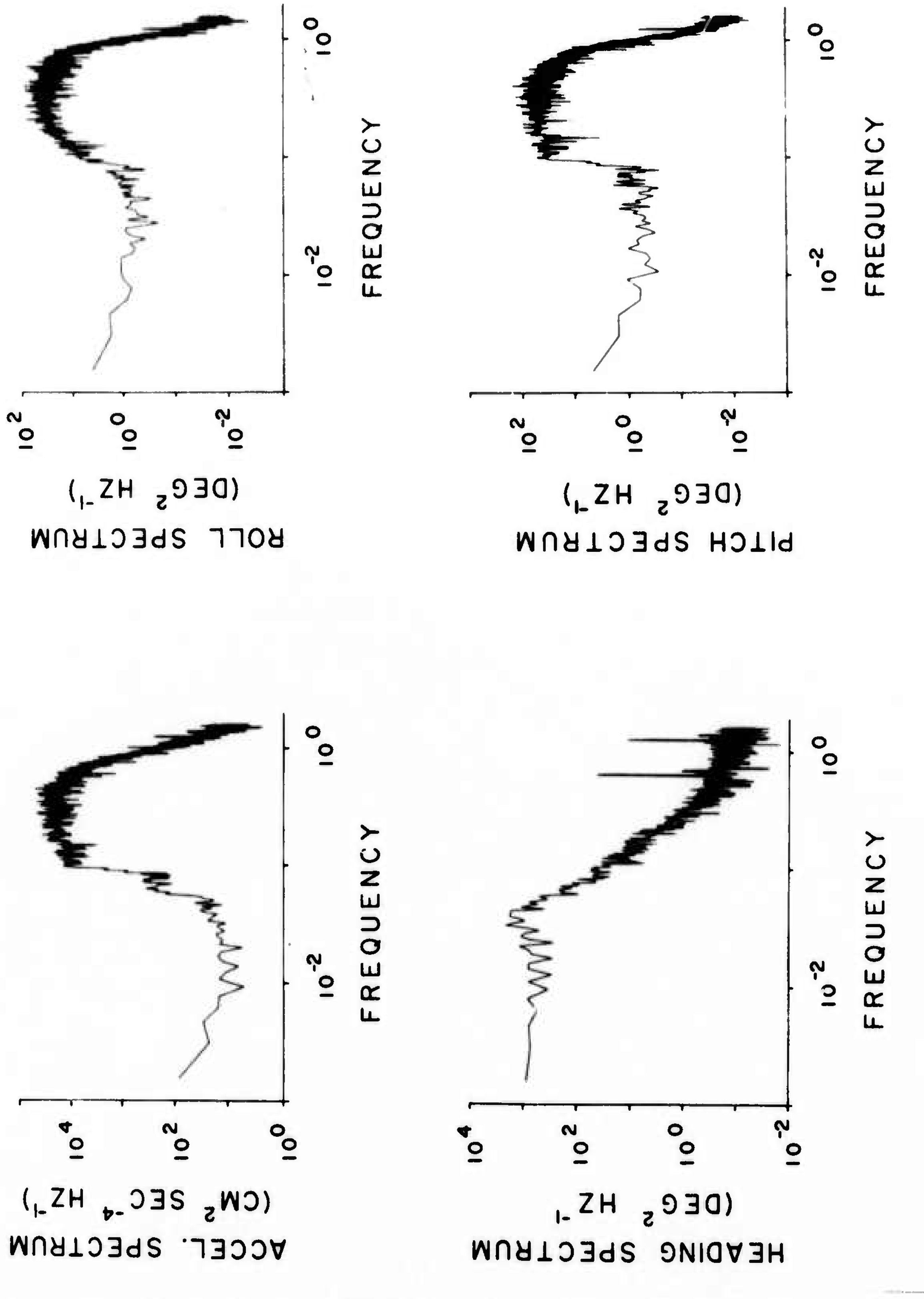


Figure 9. Typical spectra of acceleration, pitch, roll, and compass heading, as a function of frequency in Hz. Nyquist frequency is at 1.5625 Hz. Each spectrum has two degrees of freedom.

mately 1000 times the low-frequency noise, and the low and high frequency ends of the spectrum are identical to those in figure 8. The slope spectra are approximately 100 times larger than the low-frequency noise. The slight rise in energy near zero frequency is due to drift in the sample-and-hold modules, and is the reason for not using them on subsequent days. The buoy heading swings slowly with time, so its spectrum is large at low frequencies. The small spikes in the spectrum are due to aliased 400 Hz noise.

The acceleration noise values require some comment. The noise is 30 times that expected from least count noise (which itself is about equal to the noise from the transducer). It is also about 30 times larger than the noise measured with the instrument on land (no motion). It is probably due to the slightly non-linear way the buoy responds to the motion of the sea, and is not significant for our work. For example, a sinusoidal wave with an amplitude of 0.8 cm at the lowest frequency of interest (0.06 Hz) is detectable with a signal/noise ratio of one. The contribution to this same frequency band by a fully developed sea is 100 times larger.

The buoy was calibrated by observing its response to sinusoidal waves in a tank 2.6 meters wide, 2 meters deep, and 30 meters long. This facility is just barely adequate; fortunately, the best data was obtained near the buoy cut-off frequency. Lower frequency waves were too long for the shallow tank, and shorter waves were not two dimensional. The observed response is plotted in figure 10, together with the response of a 1/8 ellipsoidal hull calculated by Kim (1966). Although this hull is not very similar to the buoy in plan view, it has nearly the same cross-section along one axis, and it is of shallow draft. The wavenumber k of the incident waves was calculated from the wave frequency ω using $\omega^2 = gk \tanh kh$, where h is the depth of the water. The figure indicates the model accurately predicts the 1/2 power point in the buoy's response, and the ratio of pitch to heave response at low frequencies. This ratio diverges from the observed values at high frequencies.

We estimate the internal consistency of the data by observing how well eqs. (2.10) and (2.11) are met. In general, C_{12}, C_{13}, and Q_{23} are zero within the statistical fluctuations of the data. Six hours of data recorded in a trade-wind sea, when averaged to give spectral estimates with over 2000 degrees of freedom, yields values for these quantities that are less than 5% of the leading terms (C_{11}). To evaluate (2.11) we have averaged together 17.23 hours of data and plotted it in figure 11. The peak near 0.9 Hz is predicted by Kim (cf. figure 10), but the slight rise above unity (near 0.5 Hz) is unexplained. Perhaps it is due to the additional moment of inertial of the high instrument can.

7. CONCLUSION

The wave buoy has worked well. Typically we work near shore. The buoy is taken 20-30 km offshore, data is collected

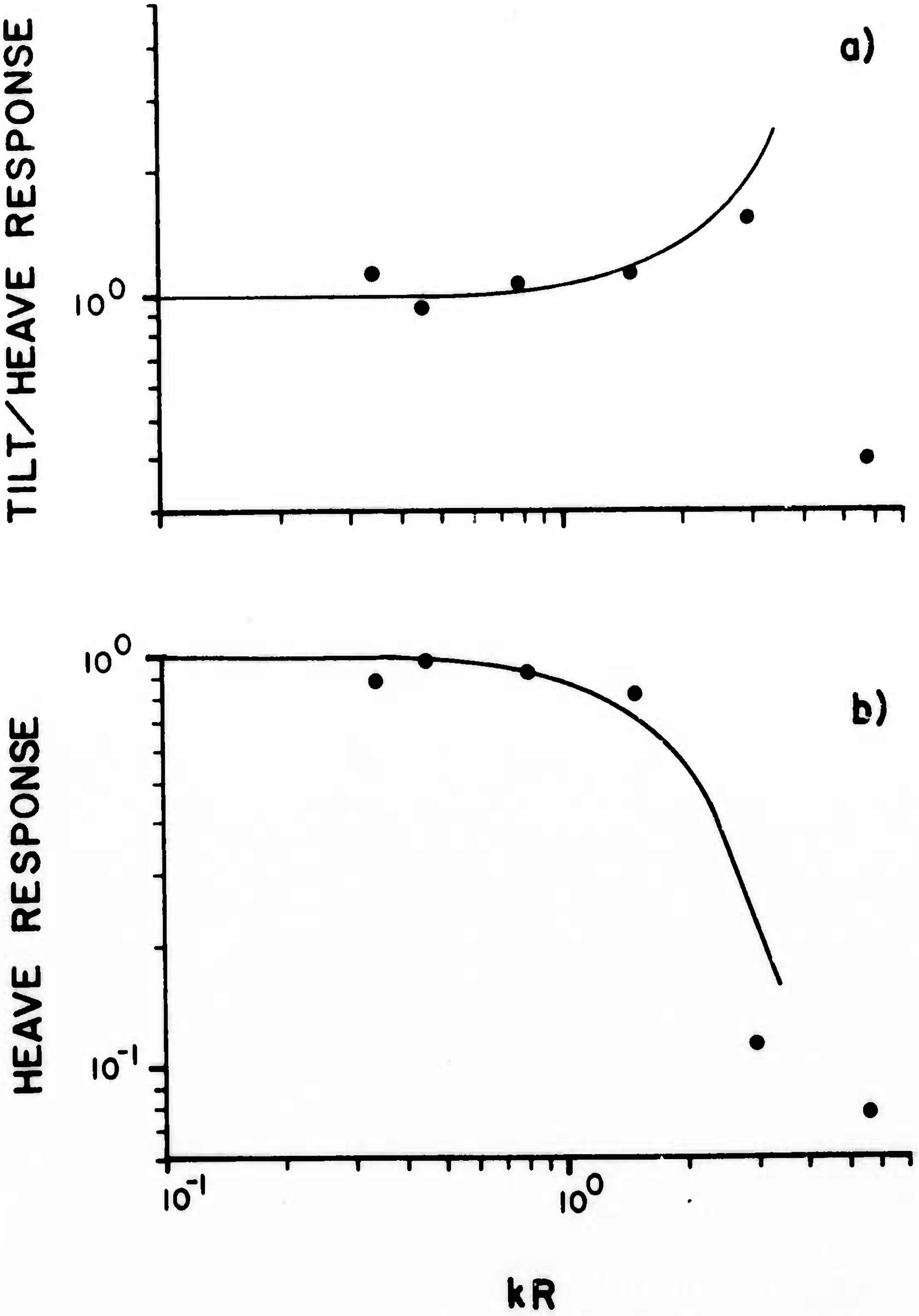


Figure 10. Measured response of buoy to wave tunnel waves (solid points) compared with Kim's (1966) theory (solid line); k is the wavenumber of the waves, R is the buoy radius. a) measured slope/(k·measured amplitude). b) measured amplitude/amplitude of waves.

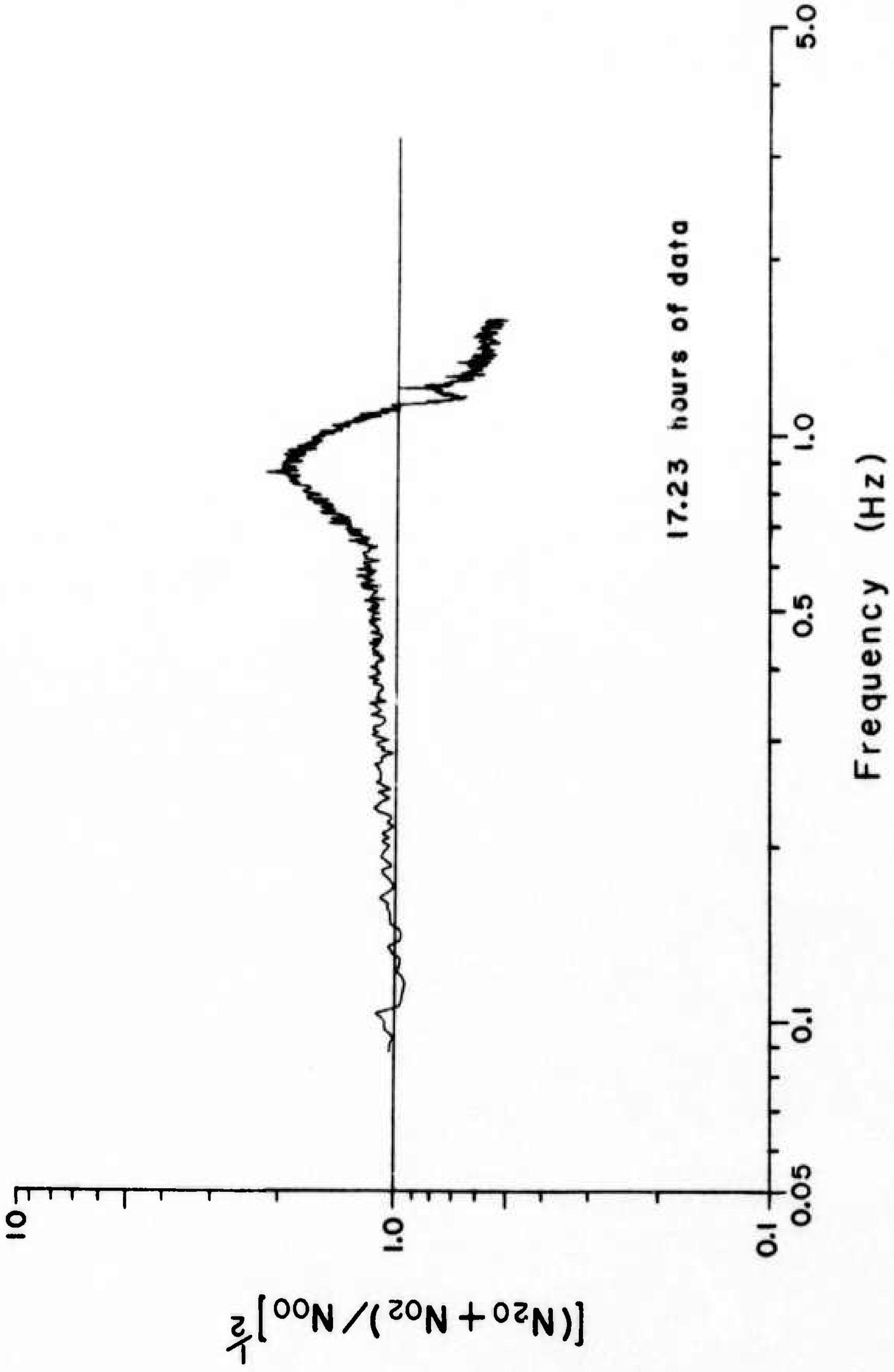


Figure 11. Ratio of wave slope to wave amplitude (times k) calculated from 17.23 hours of data recorded at sea.

for three hours, and we return, all in one day. In one year, it has operated at sea on nine different days in four different parts of the world and recorded more than 30 hrs of data. During this time there was only one major failure. On the last day of operation the unit stopped because of low battery voltage. Inspection revealed sea water had corroded away most of the terminals and discharged the batteries.

The data recorder has also worked well. We usually record data in 10 minute segments (records) of 2048 scans. Out of over 200 records recorded, 5% (11 records) have errors. All but two are due to the tape recorder adding an extra 6 bit character in the middle of a record. This is not serious because the data can be recovered with a little extra computer work. We prefer to ignore these records. Once the clock failed to act properly. No data was lost but the time word was incorrect. Once the record was too long. We suspect these errors were due to noise introduced into the logic circuits by strong radar signals from ships used to deploy the buoy. Certainly, the errors were less frequent when the ship did not have a radar or was not close to the buoy.

The buoy has withstood rough handling. It has been shipped from San Diego to Monterey and back, to San Clemente Island and back, to Hawaii and Wake Island. The waterproof connectors have been damaged by inexperienced crane operators on the ships, and the radio antennas have been bent. These items are easily repaired. The internal components, which are harder to repair, have not been damaged.

The buoy has been deployed by helicopter and from a variety of very small ships with little difficulty. Two men holding lines attached to the bridle mounting points can steady its motion on a rolling ship. It can be lifted by small hydraulic hoists or cranes found on many small workboats, and an inexpensive two-man helicopter can carry it to sea.

Our experience with the buoy shows ways it can be improved. The instrument can could be considerably smaller. It was originally designed to hold the compass, but iron in the inverter caused interference. If only three hours of data are recorded at one time, the much cheaper, low-power, cassette type data recorders could be used. This would make changing tapes easier, and would further reduce the size of the instrument can. Thus the buoy would be more disc shaped, and should have a better response to waves.

In conclusion, we can say the buoy has met the design criteria specified before it was built, and has provided the ocean wave spectra we need in our research program.

REFERENCES

Goldberg, H. D. and Goldberg, Milton J. 1969 Transducer instrumentation for surface wave measurements. Hudson Laboratories of Columbia University Technical Report 180, Dobbs Ferry, New York.

Jahnke, E. and Emde, F. 1945 Tables of Functions with formulae and curves. Dover, New York.

Jordan, W. N. 1969 Mechanical design, construction, calibration and field deployment of surface wave floats. Hudson Laboratories of Columbia University Technical Report 171, Dobbs Ferry, New York.

Kim, W. D. 1966 On a free-floating ship in waves. Journal of Ship Research, <u>10</u>, p. 182-191.

Longuett-Higgins, M. S., Cartwright, D. E., and Smith N. D. 1963 Observations of the directional spectrum of sea waves using the motions of a floating buoy, in Ocean Wave Spectra. Prentice-Hall, Englewood, New Jersey.

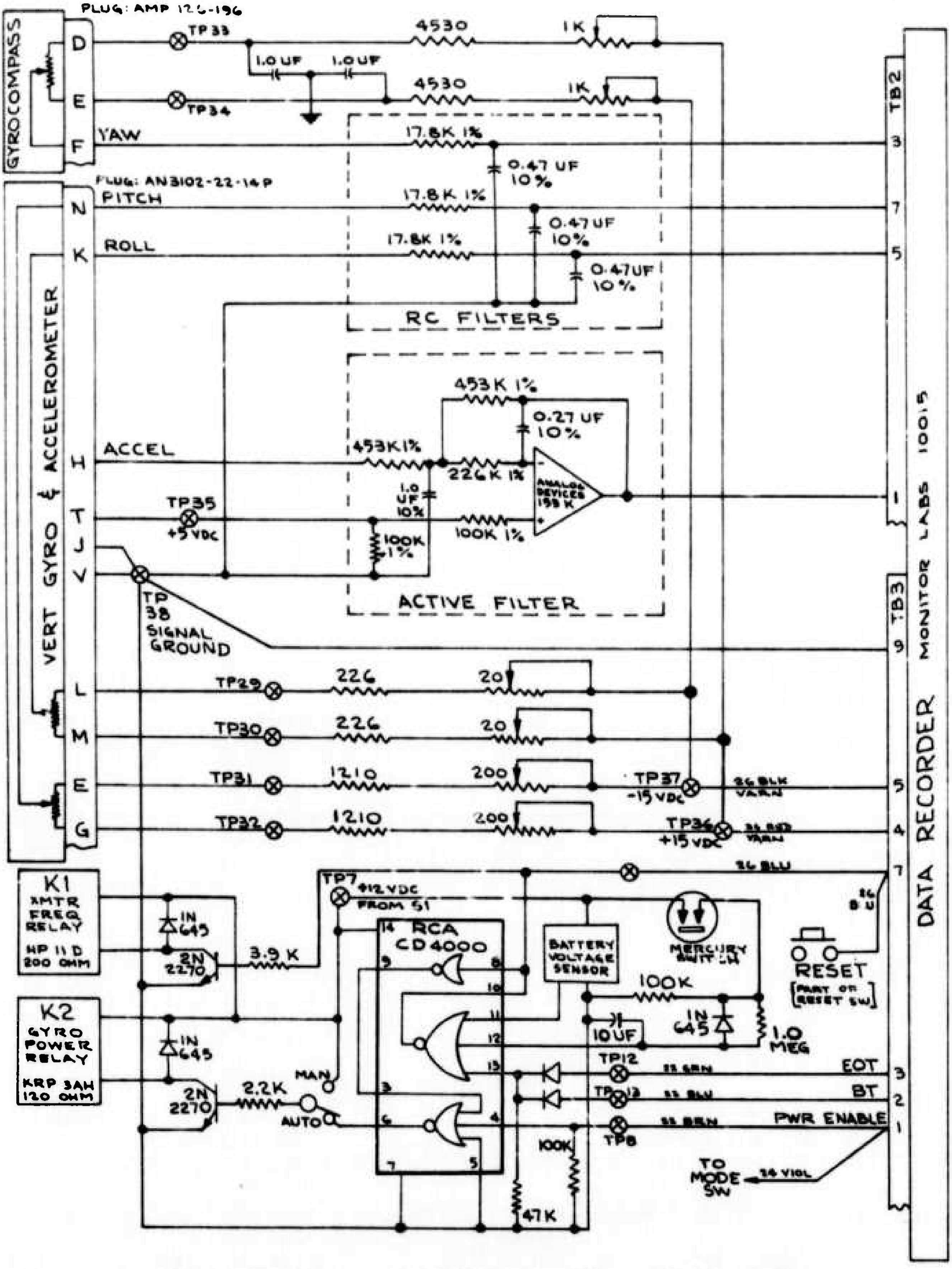
Pierson, W. J. and Moskowitz, L. 1964 A proposed spectral form for fully developed wind seas based on the similarity theory of S. A. Kitaigorodskii. J. of Geophy. Res., <u>69</u>, p. 5181-5190.

Saenger, R. A. 1969a Measurement of the statistical properties of the ocean surface with instrumented surface floats. Part 1: Theory of measurement. Hudson Laboratories of Columbia University Technical Report 183. Dobbs Ferry, New York.

Saenger, R. A. 1969b Measurements of the statistical properties of the ocean surface with instrumented surface floats. Part 2: Engineering and data processing. New York University School of Engineering and Science Technical Report TR-69-10. University Heights, New York.

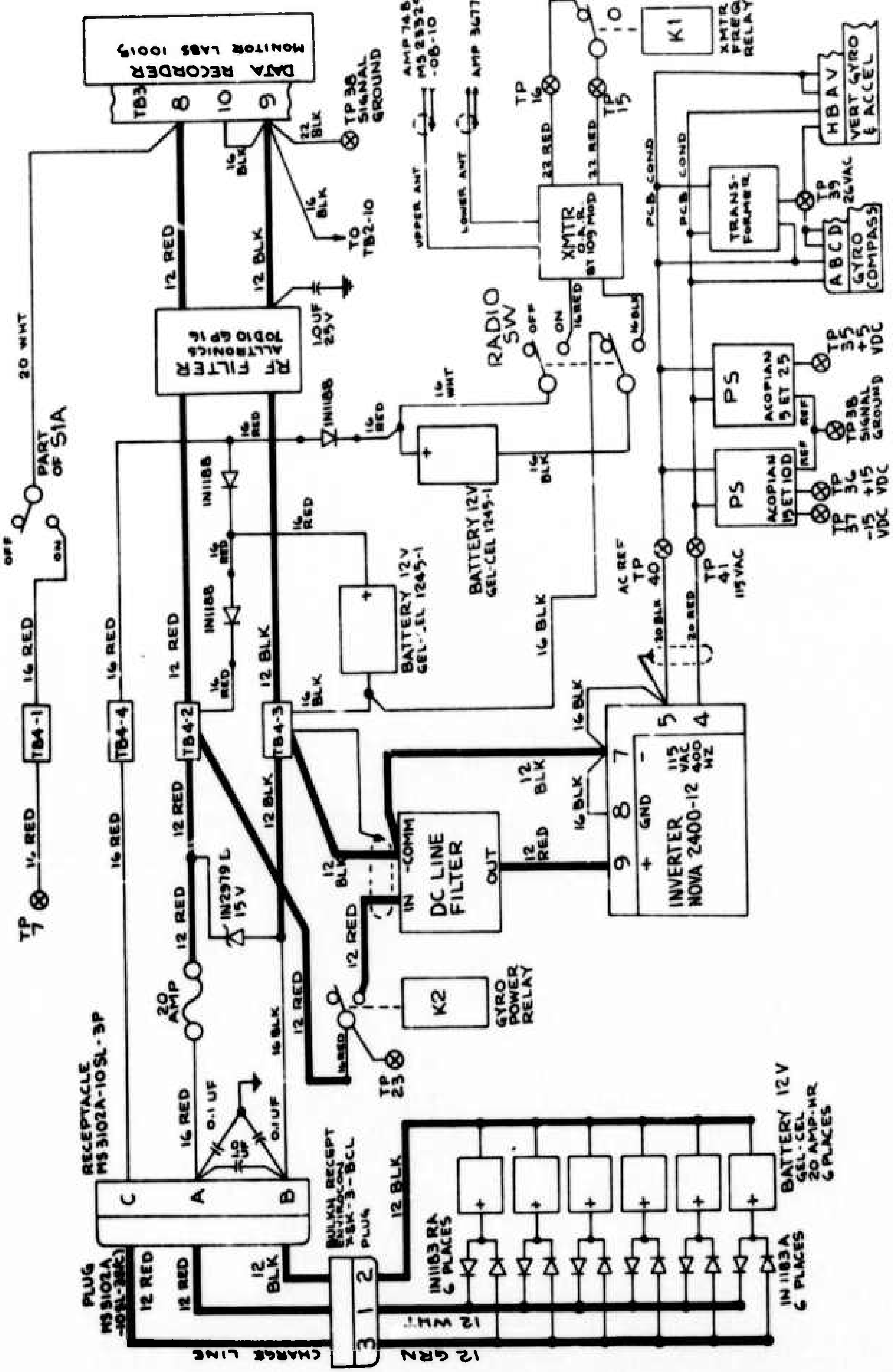
APPENDIX

This section contains many of the details necessary to operate the buoy and to analyse the data from it. We have included detailed circuit schematics (figs. A1, A2, A3), program listings, and a list of major components and their source of supply.



● SIGNAL WIRING DIAGRAM ●
WAVE BUOY

Figure A1.



POWER WIRING DIAGRAM

WAVE BUOY

Figure A2.

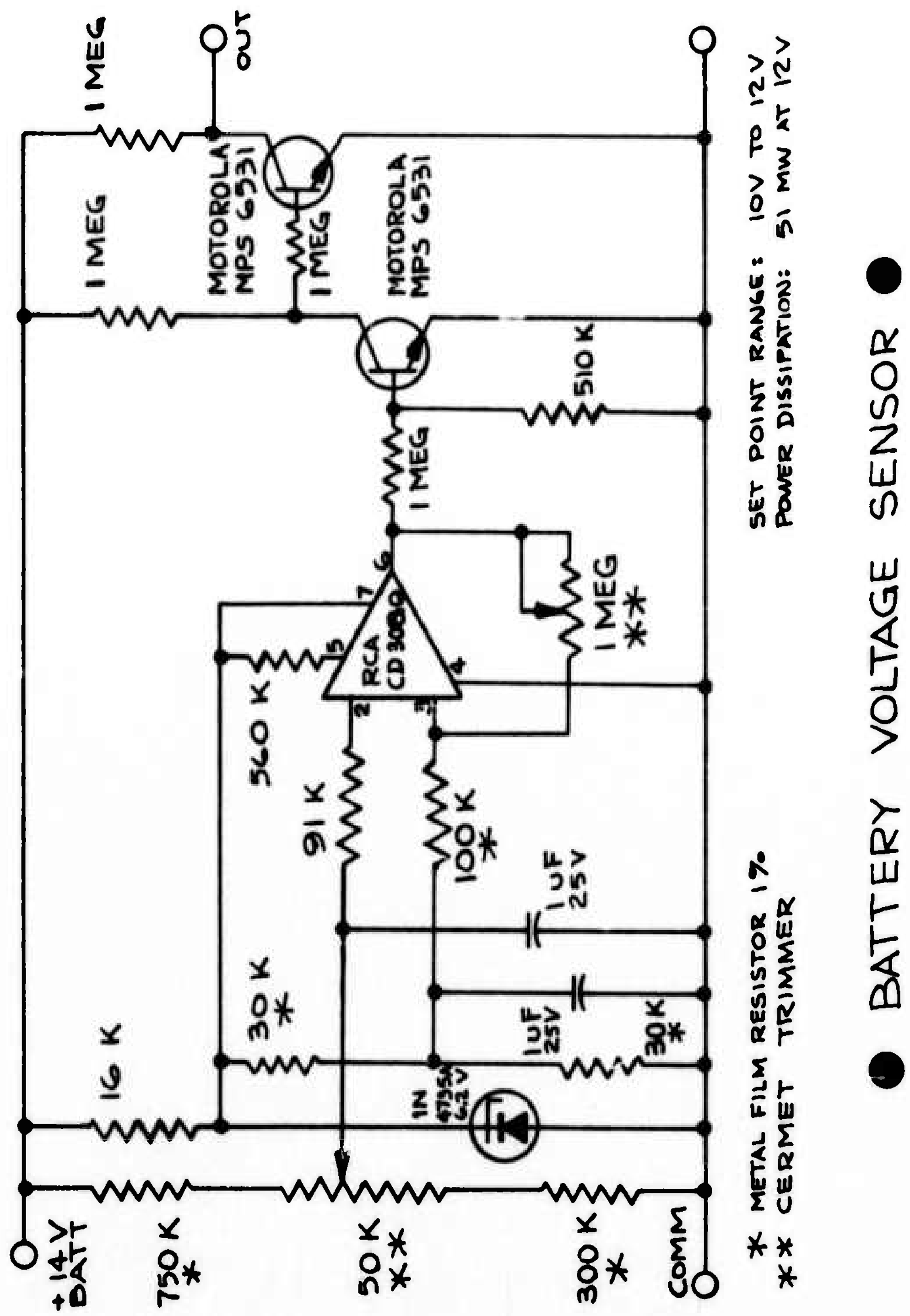

+14V BATT
750 K *
50 K **
300 K *
COMM
16 K
30 K *
1N 4735A 6.2V
1UF 25V
30K *
91 K
100 K *
1UF 25V
560 K
RCA CD 3080
2
3
4
5
6
7
1 MEG **
1 MEG
510 K
MOTOROLA MPS 6531
1 MEG
1 MEG
MOTOROLA MPS 6531
1 MEG
OUT
* METAL FILM RESISTOR 1%
** CERMET TRIMMER
SET POINT RANGE: 10V TO 12V
POWER DISSIPATION: 51 MW AT 12V
BATTERY VOLTAGE SENSOR

Figure A3.

Listing of Computer Program used to Reduce Buoy Data

```
BEGIN                                                                                     000:0000:0
                                                                             BLOCK IS SEGMENT 00002
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %         1            002:0000:0
%                                                                                         002:0000:0
%      PROGRAM WAVEBUOY PROVIDES A MEANS OF COMPUTING ESTIMATES OF                        002:0000:0
%      THE DIRECTIONAL SPECTRUM OF OCEAN WAVES FROM DATA TAPES                            002:0000:0
%      GENERATED BY THE S. I. O. SURFACE-FOLLOWING WAVEBUOY.                              002:0000:0
%                                                                                         002:0000:0
%      THE CARD INPUT PARAMETERS TO BE SUPPLIED ARE: CARD 1, THE TAPE                     002:0000:0
%      REEL NAME (6 CHARACTERS); CARD 2, THE TIME-SERIES LENGTH, 2**MEX,                  002:0000:0
%      FOLLOWED BY THE NYQUIST FREQUENCY IN HERTZ. CARD 3 CONTAINS FIVE                   002:0000:0
%      INTEGER CONSTANTS VIZ.,THE FILE NO. OF THE DATA, THE FIRST RECORD                  002:0000:0
%      OF THE DATA TO BE USED, THE NUMBER OF RECORDS TO BE AVERAGED,                      002:0000:0
%      AN INTEGER VARABLE,"CHANNELS",WHICH SELECTS WHICH OF THE FOUR                      002:0000:0
%      DATA CHANNELS ARE TO BE USED. IF ALL CHANNELS ARE REQUIRED,AS IN                   002:0000:0
%      COMPUTING DIRECTIONAL SPECTRA,THEN SET CHANNELS==1111 (BINARY) =                   002:0000:0
%      15 (DECIMAL). FOR INSTANCE,IF ONLY CHANNEL 1 POWER SPECTRUM (VERT.                 002:0000:0
%      ACCELERATION CONVERTED TO SURFACE ELEVATION) THEN SET =1000                        002:0000:0
%      (BINARY) = 8 (DECIMAL INPUT VALUE)."AVEFQ"IS THE NUMBER OF FREQ.                   002:0000:0
%      BANDS AVERAGED OVER AFTER THE AVERAGING OVER SPECTRA FROM                          002:0000:0
%      INDIVIDUAL RECORDS HAS BEEN DONE. THE NEXT CARD READ INPUTS:                       002:0000:0
%      A BOOLEAN VARIABLE "LAV" WHICH,IF SET TRUE,DOES A LOGRITHMIC TYPE                  002:0000:0
%      AVERAGING OVER SPECTRAL BANDS (CAUTION:IF LAV=TRUE,SET AVEFQ=1),                   002:0000:0
%      A BOOLEAN VARIABLE "SAHIN", WHICH IF SET TRUE MEANS THAT THE                       002:0000:0
%      SAMPLE AND HOLD MODULES OF THE WAVEBUOY WERE OPERATIONAL AT THE                    002:0000:0
%      TIME OF DATA COLLECTION; THE MAGNETIC COMPASS DEVIATION FROM TRUE                  002:0000:0
%      NORTH AND EITHER "E" OR "W" CHARACTER FOLLOWING; "TLIM" IS                         002:0000:0
%      THE GREATEST WAVE PERIOD TO BE INCLUDED IN FINDING THE VARIANCE:IF                 002:0000:0
%      BLANK THIS IS AUTOMATICALLY SET TO 2**(MEX-1)/NYQUIST, THE LONGEST                 002:0000:0
%      PERIOD  FOR THE TIME SERIES LENGTH.  FINALLY, THERE IS A CARD                      002:0000:0
%      WHICH CONTAINS SIX BOOLEAN VARIABLES SPECIFYING WHICH TABLES ARE                   002:0000:0
%      TO BE PRINTED OUT AND WHICH TYPE(S) OF SPECTRA ARE TO BE PUNCHED.                  002:0000:0
%                                                                                         002:0000:0
%      THIS PROGRAM UTILIZES THE "$SET INSALLATION" CONTROLL CARD                         002:0000:0
%      IN ORDER TO ACCESS THE SYSTEM INTRINSICS: FFT,BITRV2,FFTR,SINCOS.                  002:0000:0
%                                                                                         002:0000:0
%                                                 JOE JOY , PAUL HANSEN                   002:0000:0
%                                                 BOB STEWART                             002:0000:0
%                                                 APRIL, 1973.                            002:0000:0
%                                                                                         002:0000:0
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %                      002:0000:0
```

```
FILE PRINTER(KIND = 6, MAXRECSIZE = 22);                                          002:0000:0
                                                                              DATA IS 0005 LONG
FILE CARD (KIND = 9, MAXRECSIZE = 14);                                            002:0000:0
                                                                              DATA IS 0005 LONG
FILE PUNCHER (KIND = PUNCH, MAXRECSIZE = 14);                                     002:0000:0
                                                                              DATA IS 0005 LONG
BOOLEAN ERR;                                                                      002:0000:0
INTEGER CHANNELS, FIRSTREC, J, K, LASTREC, MEX, N, NM1, NCHANNEL, NN,             002:0000:0
        NFILE, REC, NWORDS, NREC, TAPENAME, AVEFQ;                                002:0000:0
REAL FSTEP, OMEG, SCALE, VARN, X, XNYQUIST;                                       002:0000:0
REAL ARRAY CAL,MEAN[1:4];                                                         002:0000:0
DEFINE PI = 3.14159265358979324#;                                                 002:0002:1
%                                                                                 002:0002:1
%  NOTE: 3777 OCTAL = 2047 DECIMAL = +5.0 VOLTS.       4000 OCTAL=2048            002:0002:1
   FILL CAL[*] WITH  .478515625  %  (980 CM/SEC ** 2)/2048.                       002:0002:1
                   ,.001533980768 %  PI/2048.                                     002:0003:5
                   ,.000383495197 %  (PI/4)/2048.                                 002:0003:5
                   ,.000383495197;%  (PI/4)/2048.                                 002:0003:5
READ(CARD,<A6>,TAPENAME);                                                         002:0006:3
   READ(CARD,< I4, F10.4>, MEX, XNYQUIST);                                        002:0014:5
   READ(CARD, <5I3>, NFILE, FIRSTREC, NREC, CHANNELS, AVEFQ);                     002:0025:2
IF AVEFQ LEQ 0. THEN AVEFQ:=1;                                                    002:0038:5
WRITE(PRINTER,<"TAPE REEL=", A6>,TAPENAME);                                       002:003D:4
WRITE(PRINTER,<"MEX=",I3,X5,"NYQUIST=",F8.4," HERTZ.">,MEX,XNYQUIST);             002:004B:4
   NWORDS := 0 & 1[MEX:0:1];  %  2 ** MEX                                         002:005B:4
   N := NWORDS DIV 2 + 1;                                                         002:005D:1
   NM1 := N -1;                 SCALE:=2./NWORDS**2;                              002:005F:0
%                                                                                 002:0063:2
   WRITE(PRINTER,<"FILE NO.=",I3,"  BEGINNING WITH RECORD NO. ",I3,               002:0063:2
                 " TOTAL NO. OF RECORDS READ=",I3,                                002:0065:1
    "   CHANNELS=",I3>,NFILE,FIRSTREC,NREC,CHANNELS);                             002:0065:1
%                                                                                 002:0077:1
%                                                                                 002:0077:1
%                                                                                 002:0077:1
%                                                                                 002:0077:1
%                                                                                 002:0077:1
BEGIN    % <<<<<<<<<<<<<% MAIN PROGRAM BLOCK BEGINS HERE.                         002:0077:1
%                                                                       2         002:0077:1
%                                                                                 002:0077:1
%                                                                                 002:0077:1
BOOLEAN SAHIN, TABLE1, TABLE2, TABLE3, PUNCHXS, PUNCHPS, PUNCHDS, LAV;            002:0077:1
                                                                          BLOCK IS SEGMENT 00005
INTEGER EAST,EORW,ENDPT,FREEDOM,VALAVEFQ,WEST;                                    005:0000:0
REAL C, C11, C33, C44, C34, COSAZ, COSPITCH, K2, K1, MAGDEV, Q13, Q14,            005:0000:0
     RR, SINAZ, TLIM, TANPITCH, TANROLL, Y, ZETA, Z;                              005:0000:0
REAL NZERO, DENOM1, DENOM2, JACOBK, K11, K22, N01, N10, N20, N02, N11,            005:0000:0
     N12, N21, N22, C14, C13, Q34;                                                005:0000:0
REAL ARRAY R, I [1:4, 0:NWORDS],  PSA, A, B [1:4, 0:NM1],                         005:0000:0
           CSN, CSW, CNW, CO, FREQ, OMEG4, PER, PS, QSN, QSW, QNW,                005:0008:3
           SI[0:NM1], AVEPER, A0, A1, A2, B1, B2, BETA, CREST[0:1 +               005:0008:3
           ENTIER(NM1/AVEFQ)];                                                    005:0011:0
REAL ARRAY NA6,NA7,NA8,RLD,NA1,NA2,NA3,NA4,NA5,ANG1,ANG2,RNEW,NNN,                005:0019:4
           XALPH[0: 1 + ENTIER(NM1/AVEFQ)], JACOB,KONE,KTWO[0:NM1];               005:0019:4
DIRECT FILE BUOYTAPE (KIND = 13, LABELTYPE = OMITTEDEOF, DENSITY = LOW,           005:0029:4
                      PARITY = 0, MAXRECSIZE = NWORDS + 1);                       005:0029:4
                                                                              DATA IS 0025 LONG
                                                                              DATA IS 0003 LONG
DIRECT ARRAY TAPEBUF[0:NWORDS];                                                   005:002C:4
LABEL ADDIR,AGAIN,EXIT,FINI,LOOPER,NEXT,PUN1,PUN2,PUN3,TAB2,TAB3;                 005:0030:4
%                                                                                 005:0030:4
%< < < < < < < < < < < < < < < < < < <                                            005:0030:4
FORMAT F5("  N ",X5,"PERIOD",X3,"A0=",X12, "A1=",X12,"A2=",X12,"B1=",             005:0030:4
         X12, "B2=",X6,"DEG.  TRUE",X5,"RMS DEGREES",/);                          005:0030:4
FORMAT F8(I4,X2,F5.3,X1,5(F12.4,X3),2(F7.2,X7));                                  005:0030:4
FORMAT F9("  ANG.",X4,"PERIOD",X3,"ACCEL.",X4,"SURFACE",X3,"N-SLOPE",             005:0030:4
         X3,"E-SLOPE",X3,"CO-SN",X4,"QUAD-SN",X4,"CO-SE",X4,"QUAD-SE",            005:0030:4
         X4,"CO-NE",X4,"QUAD-NE",X2,"WAVES    BEAM");                             005:0030:4
FORMAT F10(X113,"FROM",X5,"WDTH");                                                005:0030:4
FORMAT F11(" FREQ.",X5,"SEC.",X3,"CM2/SEC4",X3,"(M**2",X4,2("PER HERTZ "          005:0030:4
         ),4(" CM./HZ.  "),2("PER HERTZ ")," DEG. (T)");                          005:0030:4
FORMAT F12(E8.2,F8.2,10E10.2,2(X2,F6.1));                                         005:0030:4
FORMAT F13(" AVERAGED DIRECTIONAL COEFFICIENTS FOR .989 SEC. < T < ",             005:0030:4
          F7.1," SEC.",//);                                                       005:0030:4
FORMAT F14(X32,"WAVES FROM",X5,"BEAM WIDTH");                                     005:0030:4
FORMAT F15(X18,"PER HERTZ",X2,"PER HZ.",/);                                       005:0030:4
FORMAT F16("ANG. FREQ.",X5,"N01/N00",X5,"N10/N00",X5,"N20/N00",X5,                005:0030:4
          "N02/N00",X5,"N11/N00",X5,"ALPHA1",X3,"ALPHA2',X3,"RNEW",               005:0030:4
          X6,"ROLL",X5,"NEXP",X3,"ALPHA",/);                                      005:0030:4
FORMAT F17(E10.4,X5,5(E9.2,X3),2(F6.1,X2),2(E9.2,X1),F6.2,X2,E9.2);               005:0030:4
FORMAT F18("ANG. FREQ.",X5,"N12/N00",X5,"N21/N00",X5,"N22/N00",X6,                005:0030:4
          "NCHECK",/);                                                            005:0030:4
FORMAT F19(E10.4,X5,4(E9.2,X3));                                                  005:0030:4
%< < < < < < < < < < < < < < < < < < <                                            005:0030:4
```

```
PROCEDURE TAPESTART (DIRECTFILE, DB, RS, SF, SR, ERR);                              005:0030:4
%                                                                                   005:0030:4
VALUE RS, SF, SR;         INTEGER RS, SF, SR;        BOOLEAN ERR;                   005:0030:4
DIRECT FILE DIRECTFILE;               DIRECT ARRAY DB[0];                           005:0030:4
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %               005:0030:4
%                                                                                   005:0030:4
%     THIS PROCEDURE IS FOR READING A FIELD DATE LOGGER TAPE BY MEANS               005:0030:4
%     OF DIRECT I/O. THE TAPE IS ADVANCED  TO  A SPECIFIED FILE GIVEN               005:0030:4
%     BY "SF" AND THEN TO A RECORD GIVEN BY "SR".THE EXPECTED NUMBER                005:0030:4
%     OF WORDS EACH RECORD CONTAINS IS "RS". THE ACTUAL NUMBER OF WORDS             005:0030:4
%     TRANSFERED FROM EACH RECORD AND THE READ ERROR TYPE ARE PRINTED               005:0030:4
%     OUT FOR EACH FILE AS WELL AS THE NUMBER OF RECORDS PER FILE.                  005:0030:4
%     PROCEDURE TAPESTART REQUIRES FOREWARD DECLARATION OF DIRECT FILE              005:0030:4
%     AND DIRECT ARRAY DB.                                                          005:0030:4
%                                                                                   005:0030:4
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %               005:0030:4
BEGIN                                                                               005:0030:4
   INTEGER EOFCT, RC;                                                     3         005:0030:4
                                                                          TAPESTART IS SEGMENT 0007
FILE PRINT (KIND = PRINTER, MAXRECSIZE = 22);                                       007:0000:0
                                                                                 DATA IS 009A LONG
                                                                                 DATA IS 0005 LONG
FORMAT SKRC ("FILE", I3, " RECORD", I3, "  CONTAINS", I5, " WORDS,  ",              007:0000:0
  "ERROR NO.", I2);                                                                 007:0000:0
   WRITE(PRINT[SPACE 2],<"SPACE TO FILE ", I2, "  RECORD ", I2>, SF,SR);            007:0000:0
   ERR := FALSE; %  NO ERROR INDICATION.                                            007:0010:5
   RC := 0;                                                                         007:0011:4
   EOFCT := 1;                                                                      007:0012:2
   WHILE EOFCT LSS SF DO                                                            007:0013:0
   BEGIN                                                                            007:0013:5
      READ (DIRECTFILE, RS, DB[*]);                                       4         007:0014:2
      WAIT(DB[*]);                                                                  007:0018:2
      IF DB[*].IOEOF THEN                                                           007:001A:0
      BEGIN                                                                         007:001C:0
        CLOSE(DIRECTFILE,*);                                              5         007:001C:3
         WRITE (PRINT[SPACE 2], <"EOF ENCOUNTERED FOR FILE ", I2, " OF",            007:001E:0
           I3, " RECORD(S)">, EOFCT, RC);                                           007:0020:3
         EOFCT := * + 1;                                                            007:002E:5
         DIRECTFILE.OPEN := TRUE;                                                   007:0030:1
         RC := 0;                                                                   007:0031:5
      END ELSE                                                                      007:0032:3
      BEGIN                                                               5         007:0032:3
         RC := * + 1;                                                     5         007:0033:0
         WRITE(PRINT, SKRC, EOFCT, RC, DB[*].IOWORDS,DB[*].IOERRORTYPE);            007:0034:2
      END                                                                           007:004B:1
   END;                                                                   5         007:004B:1
   RC := 1;                                                               4         007:004B:4
   WHILE RC LSS SR DO                                                               007:004C:2
   BEGIN                                                                            007:004D:1
      READ (DIRECTFILE, RS, DB[*]);                                       4         007:004D:4
      WAIT(DB[*]);                                                                  007:0051:4
      WRITE(PRINT,SKRC, EOFCT, RC, DB[*].IOWORDS, DB[*].IOERRORTYPE);               007:0053:2
      RC := * + 1;                                                                  007:006A:1
   END;                                                                             007:006B:3
END   TAPESTART;                                                          4         007:006C:0
                                                                          TAPESTART(007) IS 0078 LONG
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%     3         005:0030:4
```

```
%                                                                                          005:0030:4
PROCEDURE PRDATE(DAT);                                                                     005:0030:4
%                                                                                          005:0030:4
%                                                                                          005:0030:4
VALUE DAT;                  INTEGER DAT;                                                   005:0030:4
%                                                                                          005:0030:4
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %                      005:0030:4
%                                                                                          005:0030:4
%     THIS PROCEDURE DECODES THE DATE AND TIME INFORMATION CONTAINED                       005:0030:4
%     IN THE LEADING 24 BITS OF EACH TAPE RECORD AND PRINTS THIS OUT.                      005:0030:4
%                                                                                          005:0030:4
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %                      005:0030:4
 BEGIN                                                                                     005:0030:4
   INTEGER ARRAY MONTHS[1:12];                                                3            005:0030:4
                                                                                 PRDATE IS SEGMENT 00009
   VALUE ARRAY MONTH( "JAN", "FEB", "MAR", "APR", "MAY", "JUN", "JUL",                     009:0010:0
                                                                                       DATA IS 0011 LONG
         "AUG", "SEP", "OCT", "NOV", "DEC");                                               009:0010:0
                                                                                       DATA IS 000C LONG
   INTEGER YEAR, DAYS, HOUR, MIN10, MIN1, NM;                                              009:0010:0
   LABEL END, EXIT;                                                                        009:0000:0
   FILL MONTHS[*] WITH 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31;                     009:0010:0
   COMMENT DAT (DATE AND TIME) IS IN THE FORM:                                             009:0014:2
         Y DD DDDD DDDD HH HHHH MMM MMMM    Y-YEAR, D-DAY, H-HOUR, M-MIN                   009:0004:2
           IN BINARY CODED DECIMAL DIGITS;                                                 009:0014:2
   YEAR := IF DAT.[23:1] EQL 1 THEN 73 ELSE 72;                                            009:0004:2
   DAYS := DAT.[22:2] * 100 + DAT.[20:4] * 10 + DAT.[16:4];                                009:0017:5
   HOUR := DAT.[12:2] * 10 + DAT.[10:4];                                                   009:000C:2
   MIN10 := DAT.[6:3];                                                                     009:000F:2
   MIN1 := DAT.[3:4];                                                                      009:0011:5
   IF YEAR MOD 4 EQL 0 THEN MONTHS[2] := 29;                                               009:0012:2
   FOR NM := 1 STEP 1 UNTIL 12 DO                                                          009:0015:1
      IF DAYS LEQ MONTHS[NM] THEN  GO  END ELSE DAYS := DAYS-MONTHS[NM];                   009:0015:5
       WRITE(PRINTER,<"ERROR IN PRDATE: DAYS TOO LARGE">);                                 009:001C:2
    GO EXIT;                                                                               009:0021:5
  END: WRITE(PRINTER, <"DATE:  ", I2, X1, A3, X1, I2, ";   TIME:  ", I2,                   009:0022:2
     ":", I1,I1>, DAYS, MONTH[NM-1], YEAR, HOUR, MIN10, MIN1);                             009:0024:1
     WRITE(PRINTER,</>);                                                                   009:0038:1
 EXIT: END PRDATE;                                                                         009:0040:4
                                                                              PRDATE(009) IS 0049 LONG
%< < < < < < < < < < < < < < < < < < < < < < < < < < < < < < < < < < <       3            005:0030:4
```

```
%                                                                                      005:0030:4
PROCEDURE READTAPE(NFILE, REC, CHANNELS);                                              005:0030:4
%                                                                                      005:0030:4
%                                                                                      005:0030:4
VALUE NFILE, REC, CHANNELS;                                                            005:0030:4
INTEGER NFILE, REC, CHANNELS;                                                          005:0030:4
%                                                                                      005:0030:4
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %                005:0030:4
%                                                                                      005:0030:4
%      THIS PROCEDURE READS ONE RECORD/CALL BY MEANS OF DIRECT I/O.                    005:0030:4
%      THE HIGH ORDER 24 BITS OF THE FIRST WORD, WHICH CONTAIN THE DATE                005:0030:4
%      AND TIME INFORMATION,ARE ISOLATED INTO THE VARIABLE "DATE".                     005:0030:4
%      THE FOUR CHANNELS OF DATA ARE SEPARATED AND CONVERTED INTO THEIR                005:0030:4
%      APPROPRIATE UNITS. THE MEAN VALUE OF EACH TIME-SERIES IS FOUND                  005:0030:4
%      AND SUBTRACTED OFF. VARIOUS ERROR DIAGNOSTICS ARE PRINTED OUT.                  005:0030:4
%      READTAPE REQUIRES FOREWARD DECLARATION OF TAPEFILE & DIRECT ARRAY.              005:0030:4
%                                                                                      005:0030:4
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %                  005:0030:4
BEGIN                                                                                  005:0030:4
   INTEGER DATE, ERRCT, IOERR, J, L, LIMIT;                                   3        005:0030:4
                                                                               READTAPE IS SEGMENT 000C
   REAL CF, XMEAN;                                                                     00C:0000:0
   BOOLEAN ARRAY SWCH[1:4];                                                            00C:0000:0
   LABEL CONT, ERROR, LEAVE;                                                           00C:0000:0
   DEFINE GETFIELD(SUBC, LBF) =  DO R[L,J] := TAPEBUF[SUBC].[LBF:12]                   00C:0000:0
   UNTIL J := * + 1 GEQ NWORDS#;                                                       00C:0000:0
%                                                                                      00C:0000:0
   ERR:=FALSE;                                                                         00C:0000:0
  COMMENT  READ THE NEXT RECORD USING DIRECT I/O;                                      00C:0000:4
   READ(BUOYTAPE, NWORDS+1, TAPEBUF[*]);                                               00C:0000:4
   WAIT( TAPEBUF[*]);                                                                  00C:0004:5
   IOERR:= TAPEBUF[*].IOERRORTYPE;                                                     00C:0006:3
     WRITE(PRINTER[SPACE 2]);                                                          00C:0009:1
     WRITE(PRINTER,<"FILE NO.", I3, " RECORD", I3, " CONTAINS", I5,                    00C:000E:3
   " WORDS  ERROR NO.", I2>, NFILE, REC, TAPEBUF[*].IOWORDS, IOERR);                   00C:0010:2
     CASE IOERR OF BEGIN                                                               00C:0024:1
     ; % 0 NO ERROR;                                                          4        00C:0026:3
     BEGIN WRITE(PRINTER,<"NOT READY">); GO TO ERROR     ; END;                        00C:0026:3
 IF TAPEBUF[*].IOWORDS NEQ NWORDS + 1 THEN GO ERROR ELSE WRITE(PRINTER,       5        00C:002D:0
       <"PARITY ERROR, RECORD WILL BE PROCESSED.">);    % 2 PARITY.                    00C:0031:3
     BEGIN WRITE(PRINTER,<"REWINDING">); GO TO ERROR;  END;                            00C:0036:2
     BEGIN WRITE(PRINTER,<"DESC. ERROR">);  GO TO ERROR;  END;                5        00C:003C:5
     BEGIN WRITE(PRINTER,<"END/BEG. TAPE">); GO EXIT; END;                    5        00C:0043:2
     BEGIN WRITE(PRINTER,<"END OF FILE">);  GO EXIT;  END;                    5        00C:004B:0
     BEGIN WRITE(PRINTER,<"ERROR #7">);  GO EXIT;  END;                       5        00C:0052:4
     BEGIN WRITE(PRINTER,<"ERROR # 8">); GO ERROR; END ;                      5        00C:005A:2
     BEGIN  WRITE(PRINTER,<"SHORT RECORD ",F10.3," CHARACTERS">,              5        00C:0060:5
             TAPEBUF[*].IOCHARACTERS);   GO TO ERROR;      END;               5        00C:0063:3
  END;                                                                        5        00C:0071:1
  L := 1;                                                                     4        00C:0076:3
    DO  SWCH[L] := BOOLEAN(CHANNELS.[4-L:1]) UNTIL L := * + 1 GTR 4;                   00C:0077:1
  COMMENT  DUMP HEADER INFORMATION ON THE LINE PRINTER;                                00C:007C:3
   CONT:                                                                               00C:007C:3
  DATE := 0 & TAPEBUF[0] [23:47:24];                                                   00C:007C:3
  PRDATE(DATE);                                                                        00C:007E:3
     FOR L := 1 STEP 1 UNTIL 4 DO                                                      00C:007F:4
      IF SWCH[L] THEN BEGIN                                                            00C:0080:2
```

```
   J := 0;                                                                      4              00C:0081:5
   CASE L-1 OF BEGIN                                                                           00C:0082:3
     GETFIELD(J,23);                                                            5              00C:0085:3
     GETFIELD(J,11);                                                                           00C:0086:2
     GETFIELD(J+1, 47);                                                                        00C:0091:1
     GETFIELD(J+1, 35);                                                                        00C:0097:2
   END OF CASES;                                                                               00C:009E:3
     REPLACE POINTER(I[L,0]) BY 4"000000000000" FOR NWORDS WORDS;               5              00C:009F:3
     ERRCT := 0;                                                                               00C:00A2:0
     MEAN[L] := R[L,0];                                                                        00C:00A2:4
    IF L EQL 2 THEN LIMIT:=256 ELSE LIMIT:=1024;                                               00C:00A5:1
%               LIMIT=256 CORRESPONDS TO 22.5 DEG. AZIMUTH                                     00C:00A9:0
%               LIMIT=1024 CORRESPONDS TO 22 DEG. TILT OR .5*G ACCEL.                          00C:00A9:0
    J := 1;                                                                                    00C:00A9:0
    DO BEGIN                                                                                   00C:00A9:4
          COMMENT  ERROR IN DATA FIRST OCTAL DIGIT DIFFERENCE;                  5              00C:00A9:4
      IF ABS(R[L,J] - R[L,J-1]) GTR LIMIT THEN                                                 00C:00A9:4
       IF ERRCT := *+1 GTR 4 THEN BEGIN                                                        00C:00AD:4
        WRITE(PRINTER,<"ERROR IN READTAPE: TOO MANY BAD DATA POINTS.",          6              00C:00B0:3
           "  LAST ONE AT ", 14, ".">, J);                                                     00C:00B2:2
      GO ERROR;                                                                                00C:00BE:1
     END ELSE BEGIN                                                                            00C:00BF:1
        WRITE(PRINTER,<"ERROR IN READTAPE:  BAD DATA POINT AT: ROW (",          6              00C:00BF:4
           I1, "), COLUMN(", 14, "). IN REFRENCE TO ", F7.2, ", POINT ",                       00C:00C1:3
          F7.2, " OUT OF LINE">, L,J,R[L,J-1], R[L,J]);                                        00C:00C1:3
         R[L,J] := R[L, J-1];                                                                  00C:00D6:4
      END;                                                                                     00C:00DA:1
          MEAN[L] := MEAN[L] + R[L,J];                                          6              00C:00DA:1
       END UNTIL J := * + 1 GEQ NWORDS;                                                        00C:00DE:0
       XMEAN := MEAN[L] / NWORDS;                                               5              00C:00E0:2
      CF := CAL[L];                                                                            00C:00E2:2
      MEAN[L] := (XMEAN - 2048) * CF;                                                          00C:00E3:5
      IF L EQL 1 THEN                                                                          00C:00E6:1
      WRITE(PRINTER,<"MEAN VALUE OF CHANNEL ",I1," IS ",F11.5,                                 00C:00E7:2
                 " CM./SEC.**2">,L,MEAN[L])                   ELSE                             00C:00E9:4
      WRITE(PRINTER,<"MEAN VALUE OF CHANNEL ",I1," IS ",F11.5,                                 00C:00F8:4
                 " RADIANS.">,L,MEAN[L]);                                                      00C:00FB:0
      IF L EQL 4 THEN WRITE(PRINTER,</>);                                                      00C:010A:1
        J := 0;                                                                                00C:0111:0
      DO R[L,J] := (R[L,J]-XMEAN) * CF UNTIL J:= * + 1 GEQ NWORDS;                             00C:0111:4
     END;                                                                                      00C:0116:1
   GO LEAVE;                                                                    4              00C:011A:3
  ERROR: ERR := TRUE;                                                                          00C:011B:0
LEAVE:END READTAPE;                                                                            00C:011B:4
                                                                                READTAPE(00C) IS 012E LONG
%< < < < < < < < < < < < < < < < < < < < < < < < < < < < < < < < < < < < <      3              005:0030:4
```

```
%                                                                              005:0030:4
PROCEDURE ADJPHASE(R,I,SI,CO,MEX,CHANNELS,NYQ,SAHIN);                          005:0030:4
%                                                                              005:0030:4
%                                                                              005:0030:4
REAL ARRAY R,I[1,0],SI,CO[0];                                                  005:0030:4
INTEGER MEX,CHANNELS;          REAL NYQ;           BOOLEAN SAHIN;             005:0030:4
%                                                                              005:0030:4
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %          005:0030:4
%                                                                              005:0030:4
%   FOR EACH CHANNEL, J, THIS PROCEDURE CORRECTS THE TIME-SERIES BY           005:0030:4
%   MULTIPLYING ITS FOURIER TRANSFORM BY THE APPROPRIATE INVERSE FREQ.        005:0030:4
%   RESPONSE OF ITS OUTPUT FILTER AND THEN APPLYING THE REVERSE               005:0030:4
%   TRANSFORM TO THE PRODUCT. THE RESULTING REAL TIME-SERIES IS THEN          005:0030:4
%   SCALED AND RETURNED THROUGH ARRAY R.                                      005:0030:4
%   THE CHANNEL #1 TIME HISTORY IS CONVERTED TO SURFACE ELEVATION.            005:0030:4
%   WHEN THE SAMPLE & HOLD MODULES ARE OPERATING, SET SAHIN=TRUE.             005:0030:4
%                                                                              005:0030:4
%   NOTE: PROCEDURE SINCOS MUST BE CALLED PRIOR TO CALLING ADJPHASE.          005:0030:4
%   ADJPHASE REQUIRES FOREWARD DECLARATION OF ARRAYS.                         005:0030:4
%                                                                              005:0030:4
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %          005:0030:4
BEGIN                                                                          005:0030:4
INTEGER HFPTS,J,K,N,NPTS;                                         3            005:0030:4
                                                                   ADJPHASE IS SEGMENT 000D
REAL A,B,CS,CDS,DS,F1,F2,PI,SCALE,SDS,SN,SQ2,TAU,TEMPI,TEMPR,VARN,             00D:0000:0
     W,WI1,WI2,WO,WO2,ZC,ZS;                                                   00D:0000:0
%                                                                              00D:0000:0
NPTS:=0 & 1[MEX:0:1];          % =2**MEX                                       00D:0000:0
SCALE:=2./NPTS**2;                                                             00D:0001:3
PI:=3.1415926536; HFPTS:=NPTS DIV 2; SQ2:=SQRT(2.); W:=2.*PI*NYQ/HFPTS;        00D:0003:5
TAU:=0.01;     %(SECONDS)  SCAN INTERVAL BETWEEN CHANNEL SAMPLES.              00D:000D:1
F1:=1.0;       %(HERTZ) CUT-OFF FREQ. OF CHANNEL #1 FILTER.                    00D:000F:3
F2:=20.0:      %(HERTZ)  CUT-OFF FREQ. FOR CHANNELS 2,3,4 FILTERS.             00D:0011:0
WI1:=1./(2.*PI*F1);           WI2:=1./(2.*PI*F2);                              00D:0012:3
%                                                                              00D:001A:4
FOR J:=1 STEP 1 UNTIL 4 DO IF CHANNELS.[4-J:1] NEQ 0 THEN BEGIN               00D:001A:4
FFT(R[J,*],I[J,*].SI,CO,MEX);              BITRV2(R[J,*],I[J,*],MEX):          00D:001D:5
VARN:=0.;                                                                      00D:0026:3
N:=0;   DO VARN:=*+(R[J,N]**2 + I[J,N]**2)*SCALE UNTIL N:=*+1 GEQ HFPTS,       00D:0027:1
IF J EQL 1 THEN                                                                00D:002F:1
WRITE(PRINTER,<"VARIANCE OF INPUT TIME HISTORY CHAN. #",I1," IS ",E12.5,       00D:002F:5
      " CM.**2/SEC.**4">,J,VARN)                   ELSE                        00D:0032:1
WRITE(PRINTER,<"VARIANCE OF INPUT TIME HISTORY CHAN. #",I1," IS ",E12.5,       00D:0040:1
      " RADIANS**2">,J,VARN);                                                  00D:0042:3
%                                                                              00D:0050:4
DS:=(J-1)*TAU*W;          SDS:=SIN(DS);        CDS:=COS(DS);                   00D:0050:4
N:=0;                                                                          00D:0055:5
THRU NPTS DO BEGIN                                                             00D:0056:3
        IF N LEQ HFPTS THEN K:=-N ELSE K:=NPTS - N;                   5        00D:005A:4
                                          % WHERE S=(I-1)*TAU*W*K; K=1,2..     00D:005F:1
        IF N NEQ 0 AND J NEQ 1 THEN BEGIN ZC:=CS;     ZS:=SN;                  00D:005F:1
                IF N EQL HFPTS+1 THEN ZS:=-SN;                         6        00D:0062:5
                CS:=ZC*CDS+ZS*SDS;       SN:=ZS*CDS-ZC*SDS;                    00D:0065:3
        END ELSE BEGIN                                                         00D:006A:1
                CS:=1.;     SN:=0.;      END;                          6        00D:006A:4
%               SN=SIN(S)=SIN(S+DS) ,    CS=COS(S)=COS(S+DS).          6        00D:006D:1
        TEMPR:=R[J,N];                   TEMPI:=I[J,N];                         00D:006D:1
```

```
% FUNCTIONS A AND B ARE REAL AND IMAGINARY PARTS OF INVERSE FILTER RESP.                000:0070:5
        IF J EQL 1 THEN BEGIN                                                            00D:0070:5
                WO2:=(WO:=W*K*WI1)**2 - 1.;                                   6          00D:0072:0
                IF SAHIN THEN BEGIN A:=WO2;  B:=SQ2*WO;  END                             00D:0076:4
                ELSE BEGIN                                                    7          00D:0079:4
                A:=WO2*CS - SQ2*WO*SN;                                        7          00D:007A:1
                B:=WO2*SN + SQ2*WO*CS;        END;                                       00D:007D:0
                IF N EQL HFPTS THEN B:=0.;                                    7          00D:007F:5
                R[J,N]:=A*TEMPR + B*TEMPI;                                               00D:0081:5
                I[J,N]:=A*TEMPI - B*TEMPR;                                               00D:0085:3
        END ELSE BEGIN                                                                   00D:0089:1
                WO:=W*K*WI2;                                                  6          00D:0089:4
                IF SAHIN THEN BEGIN  A:=1.;  B:=WO;   END ELSE BEGIN                     00D:008B:3
                A:=CS + WO*SN;       B:=WO*CS - SN;      END;                 7          00D:008F:5
                IF N EQL HFPTS THEN B:=0.;                                    7          00D:0093:3
                R[J,N]:=A*TEMPR - B*TEMPI;                                               00D:0095:3
                I[J,N]:=A*TEMPI + B*TEMPR;                                               00D:0099:1
        END;    N:=*+1;                                                                  00D:009C:5
END;                                                                          6          00D:009E:1
%                                                                             5          00D:009F:2
N:=1;          % CONVERT ACCELERATION TRANSFORM TO SURFACE ELEVATION.                    00D:009F:2
IF J EQL 1 THEN BEGIN                                                                    00D:00A0:0
        THRU (NPTS-1) DO BEGIN                                                5          00D:00A1:1
              IF N LEQ HFPTS THEN                                             6          00D:00A5:4
                 WO:=-1./(N*W)**2 ELSE WO:=-1./((NPTS-N)*W)**2;                          00D:00A6:3
             R[1,N]:=WO*R[1,N];              I[1,N]:=WO*I[1,N];                          00D:00AE:0
             N:=*+1;                                                                     00D:00B3:2
        END;                                                                             00D:00B4:4
END ELSE BEGIN                                                                6          00D:00B5:5
%                                                                             5          00D:00B6:2
BITRV2(R[J,*],I[J,*],MEX);             FFTR(R[J,*],I[J,*],SI,CO,MEX);                    00D:00B6:2
%                                                                                        00D:00BB:0
N:=0;    DO R[J,N]:=R[J,N]/NPTS UNTIL N:=*+1 GEQ NPTS;                                   00D:00BE:0
REPLACE POINTER(I[J,*]) BY 0. FOR NPTS WORDS;                                            00D:00C5:4
END;                                                                                     00D:00C8:1
END;                                                                          5          00D:00C8:1
END ADJPHASE;                                                                 4          00D:00CA:3
                                                                              ADJPHASE(00D) IS 00D4 LONG
% % % % % % % % % % % % % % % % % % % % % . % % % % % % % % % % % % % % % %   3          005:0030:4
```

```
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -                 005:0030:4
   EAST:= 4"C5";                                                                          005:0030:4
   WEST:= 4"E5";                                                                          005:0031:3
   TAPEBUF[*].IOCW := 0 & 1[44:1] & 6[33:3] & 1[30:1]; % 7-TRK, 200 BPI                   005:0032:2
   TAPEBUF[*].IOMASK := 0 & 1[7:1]; %  SUPPRESS PARITY ERROR RETRY.                       005:0035:1
   FSTEP := XNYQUIST * 2./NWORDS;                                                         005:0037:0
   OMEG := 1./(2. * PI * FSTEP);                                                          005:003A:1
   SINCOS(SI, CO, MEX); % SET UP SIN-COS TABLE FOR FFTF                                   005:003F:1
   J := 1;                                                                                005:0041:1
   DO BEGIN                                                                               005:0041:5
      FREQ[J] := FSTEP * J;                                                      3        005:0041:5
      PER[J] := 1./FREQ[J];                                                               005:0043:4
      OMEG4[J] := (OMEG/J) ** 4;                                                          005:0046:3
      KTWO[J]:=(KONE[J]:=.0402840996*(FSTEP*J)**2)**2;                                    005:0049:0
      JACOB[J]:=480200./(2.*PI*FSTEP*J)**3;                                               005:004E:0
   END UNTIL J := * + 1 GEQ N;                                                            005:0054:0
            PS[0] := OMEG4[0] := FREQ[0] := PER[0] := 0.;                        3        005:0056:2
   NN := 0;                                                                               005:0059:5
   READ(CARD,<2(I5,X5),F6.1,X1,A1,X5,F6.1>,IAV,SAHIN,MAGDEV,EORW,TLIM);                   005:005A:3
   READ(CARD,<6(L5,X5)>, TABLE1,TABLE2,TABLE3,PUNCHXS,PUNCHPS,PUNCHDS);                   005:0070:5
   IF TLIM LEQ 0. THEN TLIM:=NM1/XNYQUIST;                                                005:0089:2
   IF LAV THEN WRITE(PRINTER,<"SPECTRAL BANDS AVERAGED ON LOGRITHMIC ",                   005:008B:5
                             "BASIS.">)          ELSE BEGIN                               005:008E:3
   FREEDOM:=2*NREC*AVEFQ;                                                        3        005:0092:4
   WRITE(PRINTER,<I2," ADJACENT FREQS. AVERAGED YIELDING ",I3,                            005:0094:4
                  " DEGREES OF FREEDOM IN SPECTRAL ESTIMATES.">,                          005:0096:3
                  AVEFQ,FREEDOM);                END;                                     005:0097:2
   IF MEX LEQ 8 AND LAV THEN BEGIN                                               3        005:009A:4
         WRITE(PRINTER,<"TIME HISTORY TOO SHORT TO IMPLIMENT LOGRITHMIC"         3        005:00A6:3
                ," AVERAGING.">);        GO TO FINI;    END;                              005:00AB:2
   WRITE(PRINTER,<"SAMPLE & HOLD MODULES IN=",L5>,SAHIN);                        3        005:00AC:3
   WRITE(PRINTER,<"MAGNETIC COMPASS DEVIATION=",F6.1,X1,A1,".",//>,                       005:00BA:4
                  MAGDEV,EORW);                                                           005:00BD:2
   MAGDEV:= MAGDEV*PI/180.;                                                               005:00CA:1
WRITE(PRINTER,<6(L5,X5)>,TABLE1,TABLE2,TABLE3,PUNCHXS,PUNCHPS,PUNCHDS);                   005:00CF:4
%                                                                                         005:00E7:4
   TAPESTART(BUOYTAPE, TAPEBUF, NWORDS+1, NFILE, FIRSTREC, ERR);                          005:00E7:4
%                                                                                         005:00EB:4
REC:=1;                                                                                   005:00EB:4
NEXT: BEGIN READTAPE(NFILE, REC+FIRSTREC-1, CHANNELS);                                    005:00EC:2
      IF NOT ERR THEN ADJPHASE(R,I,SI,CO,MEX,CHANNELS,XNYQUIST,SAHIN)            3        005:00EF:2
                      ELSE GO TO AGAIN;                                                   005:00F4:1
      NN:= NN + 1;                                                                        005:00F4:5
%                                                                                         005:00F6:1
      K:=0;                                                                               005:00F6:1
      IF EORW EQL EAST THEN DO R[2,K]:=* + MEAN[2] + MAGDEV                               005:00F6:5
                               UNTIL K:=*+1 GEQ NWORDS ELSE                               005:00FA:2
      IF EORW EQL WEST THEN DO R[2,K]:=* + MEAN[2] - MAGDEV                               005:00FD:2
                               UNTIL K:=*+1 GEQ NWORDS ELSE GO TO EXIT;                   005:0101:2
      K:=0;                                                                               005:0104:2
%     CONVERT ANGLE TIME HISTORIES TO SLOPE TIME HISTORIES CHANS. 3,4.                    005:0105:0
      THRU NWORDS DO BEGIN                                                                005:0105:0
           SINAZ:=SIN(R[2,K]);           COSAZ:=COS(R[2,K]);                     4        005:0109:1
           TANROLL:=TAN(R[3,K]);         TANPITCH:=TAN(R[4,K]);                           005:010D:1
           COSPITCH:=COS(R[4,K]);                                                         005:0111:3
           R[3,K]:= -COSAZ*TANROLL/COSPITCH - SINAZ*TANPITCH;                             005:0113:4
           R[4,K]:= +SINAZ*TANROLL/COSPITCH - COSAZ*TANPITCH;                             005:0117:4
```

```
        IF EORW EQL EAST THEN R[2,K]:=* - MEAN[2] - MAGDEV ELSE                        005:0116:3
                              R[2,K]:=* - MEAN[2] + MAGDEV;                            005:011D:4
        K:=*+1;                                                                        005:0123:0
     END;                                                                              005:0124:2
%                                                                          4           005:0125:3
%                                                                                      005:0125:3
     FOR NCHANNEL:=1 STEP 1 UNTIL 4 DO IF CHANNELS.[4-NCHANNEL:1] NEQ 0                005:0125:3
     THEN BEGIN IF NCHANNEL GEQ 2 THEN BEGIN                                           005:0127:5
         FFT(R[NCHANNEL,*], I[NCHANNEL,*], SI, CO, MEX);                   5           005:012A:0
         BITRV2(R[NCHANNEL, *], I[NCHANNEL, *], MEX);                                  005:012E:2
         END;                                                                          005:0131:4
%                                                                          5           005:0131:4
%    CALCULATE POWER SPECTRA;  CALCULATE VARIANCES.                                    005:0131:4
     VARN:=0.;                 K:=0;                                                   005:0131:4
       DO BEGIN                                                                        005:0133:0
          X := (R[NCHANNEL, K] ** 2 + I[NCHANNEL, K] ** 2)*SCALE;          5           005:0133:0
          IF NCHANNEL EQL 1 AND K LSS INTEGER(1./(TLIM*FSTEP)) THEN                    005:0137:3
          X:=0.;                                                                       005:0138:0
          A[NCHANNEL,K]:=R[NCHANNEL,K]; B[NCHANNEL,K]:=I[NCHANNEL,K];                  005:013C:1
          VARN := * + X;                                                               005:0142:3
          PSA[NCHANNEL, K] := IF NN EQL 1 THEN X ELSE                                  005:0143:5
          PSA[NCHANNEL,K] + X; END UNTIL K:=*+1 GEQ N;                                 005:0147:3
%                                                                          5           005:014B:5
   CASE NCHANNEL-1 OF BEGIN                                                            005:014B:5
   WRITE(PRINTER,</,"SURFACE ELEVATION VARIANCE=",E12.5," CM**2">,VARN);   5           005:014E:3
   WRITE(PRINTER,<  "HEADING VARIANCE=",E12.5," RADIANS**2">,VARN);                    005:015D:1
   WRITE(PRINTER,<  "N-S SLOPE VARIANCE=",E12.5," SLOPE**2 (NO UNITS)">,               005:016B:4
        VARN);                                                                         005:016D:3
   WRITE(PRINTER,<  "E-W SLOPE VARIANCE=",E12.5," SLOPE**2 (NO UNITS)",                005:017A:1
        ////>,VARN);                                                                   005:017C:0
   END OF CASES;                                                                       005:0186:4
%                                                                          5           005:0188:0
     END;     % END OF FFT LOOP.   ..........                                          005:018B:0
%                                                                          4           005:018D:2
%    CALCULATE CROSS SPECTRA                                                           005:018D:2
     K:=1;  DO CSN[K]:=*+(A[1,K]*A[3,K] + B[3,K]*B[1,K])*SCALE                         005:018D:2
            UNTIL K:=*+1 GEQ N - 1;                                                    005:0193:2
     K:=1;  DO QSN[K]:=*+(A[3,K]*B[1,K] - A[1,K]*B[3,K])*SCALE                         005:0196:5
            UNTIL K:=*+1 GEQ N - 1;                                                    005:019C:5
     K:=1;  DO CSW[K]:=*+(A[1,K]*A[4,K] + B[4,K]*B[1,K])*SCALE                         005:01A0:2
            UNTIL K:=*+1 GEQ N - 1;                                                    005:01A6:2
     K:=1;  DO QSW[K]:=*+(A[4,K]*B[1,K] - A[1,K]*B[4,K])*SCALE                         005:01A9:5
            UNTIL K:=*+1 GEQ N - 1;                                                    005:01AF:5
     K:=1;  DO CNW[K]:=*+(A[3,K]*A[4,K] + B[4,K]*B[3,K])*SCALE                         005:01B3:2
            UNTIL K:=*+1 GEQ N - 1;                                                    005:01B9:4
     K:=1;  DO QNW[K]:=*+(A[4,K]*B[3,K] - A[3,K]*B[4,K])*SCALE                         005:01BD:1
            UNTIL K:=*+1 GEQ N - 1;                                                    005:01C3:3
     K:=1;  DO PS[K]:=*+(A[1,K]**2 + B[1,K]**2)*SCALE/OMEG4[K]                         005:01C7:0
            UNTIL K:=*+1 GEQ N - 1;                                                    005:01CC:1
%                                                                                      005:01CF:4
     END;                                                                              005:01CF:4
%                                                                          3           005:01CF:4
AGAIN:REC:=*+1;                                                                        005:01CF:4
IF REC LEQ NREC THEN GO TO NEXT;                                                       005:01D1:0
%                                                                                      005:01D2:2
EXIT: IF NN GTR 0 THEN BEGIN                                                           005:01D2:2
     NCHANNEL:=1;              Z:=1./NN;                                   3           005:01D3:3
```

```
                                                                                     005:01D7:0
   DO BEGIN                                                                4         005:01E7:0
     K:=1;                                                                           005:01D7:4
     IF BOOLEAN(CHANNELS.[4-NCHANNEL:1]) THEN                                        005:01E9:2
         IF NCHANNEL EQL 1 THEN DO BEGIN                                             005:01D8:0
                                  PS[K]:=* * Z;                            5         005:01DC:5
                                  PSA[1,K]:=* * Z;       END                         005:01DF:0
                                  UNTIL K:=*+1 GEQ N-1 ELSE                5         005:01E1:4
                                  DO PSA[NCHANNEL,K]:=* * Z                          005:01E4:1
                                  UNTIL K:=*+1 GEQ N-1 ELSE                          005:01E7:3
   END UNTIL NCHANNEL:=*+1 GTR 4;                                                    005:01E9:5
   END ELSE BEGIN WRITE(PRINTER,<"UNSUCCESSFUL TAPE READ">);               4
                                                                               DATA IS 00FC LONG
                                                                           3         005:01EF:4
                      GO TO FINI;        END;                              3         005:01F0:1
%                                                                                    005:01F0:1
   K:=0;                                                                             005:01F0:5
   IF NN GTR 1 THEN DO BEGIN                                               3         005:01F2:0
      CSN[K]:=* * Z;        QSN[K]:=* * Z;        CSW[K]:=* * Z;                     005:01F7:3
      QSW[K]:=* * Z;        CNW[K]:=* * Z;        QNW[K]:=* * Z;                     005:01FE:0
   END UNTIL K:=*+1 GEQ N;                                                 3         005:01FF:2
   WRITE(PRINTER[SKIP 1]);                                                           005:0204:3
   FOR NCHANNEL:=1,3,4 DO BEGIN                                            3         005:020B:2
      VARN:=0.;             K:=0;                                                    005:020C:4
      DO VARN:=* + PSA[NCHANNEL,K] UNTIL K:=*+1 GEQ N;                               005:0211:2
      IF NCHANNEL EQL 1 THEN                                                         005:0212:0
      WRITE(PRINTER,<"AVERAGE VARIANCE OF ",I2," RECORDS, CHAN.#",                   005:0214:2
                    I1," = ",E12.5,"  CM.**2",/>,NREC,NCHANNEL,VARN)                 005:021F:0

      ELSE                                                                           005:0224:4
      WRITE(PRINTER,<"AVERAGE VARIANCE OF ",I2," RECORDS, CHAN.#",                   005:0227:0
      I1," = ",E12.5," SLOPE**2 (NO UNITS)",/>,NREC,NCHANNEL,VARN);                  005:0237:1
      END;                                                                 3         005:023A:3
%                                                                                    005:023B:3
IF TABLE1 THEN BEGIN                                                       3         005:023B:2
WRITE(PRINTER,<//,"UNNORMALIZED POWER SPECTRA AVERAGED OVER ",I2,                    005:023D:1
                " RECORDS.">,NREC);                                                  005:0249:1
  WRITE(PRINTER,<X55,"ACCEL.",X3,"SURFACE",X4,"COMP.",                               005:024B:0
     X3,"N-SLOPE",X3,"E-SLOPE",/>);                                                  005:024E:4
  WRITE(PRINTER, <" STEP", X2, "FREQUENCY", X2, "PERIOD", X3, "OMEGA-4"              005:0250:3
    , X3, "CHANNEL(S):",      5I10/>, FOR NCHANNEL  := 1,1,2,3,4 DO                  005:0262:2
    IF CHANNELS.[4-NCHANNEL:1] EQL 1 THEN NCHANNEL);                                 005:026E:1
     J := 0;                                                                         005:026E:5
DO WRITE(PRINTER,<I4,3E10.2,X18,5E10.2>, J,FREQ[J],PER[J],OMEG4[J],                  005:0280:2
          IF CHANNELS.[3:1] EQL 1 THEN PS[J], FOR NCHANNEL:=1,2,3,4 DO               005:028D:5
          IF CHANNELS.[4-NCHANNEL:1] EQL 1 THEN PSA[NCHANNEL,J])                     005:0292:4
          UNTIL J:=*+1 GEQ N;                                                        005:029C:3
END;                                                                       3         005:029C:3
%                                                                                    005:029C:3
IF CHANNELS LSS 15 THEN GO TO FINI;                                                  005:029D:5
%                                                                                    005:029D:5
%AVERAGE TOGETHER ADJACENT FREQUENCY BANDS.                                          005:029D:5
J:=1;                                                                                005:029E:3
IF LAV THEN BEGIN                                                          3         005:029F:2
           X:=1./FSTEP;            Y:=-X;                                            005:02A3:0
   DO BEGIN AVEPER[J]:=1./FREQ[J]; PS[J]:=* * X;  PSA[1,J]:=* * X;         4         005:02AA:0
           PSA[2,J]:=* * X;   PSA[3,J]:=* * X;   PSA[4,J]:=* * X;                    005:02B0:5
           CSN[J]:=* * X;   QSN[J]:=* * X;   CSW[J]:=* * Y;                          005:02B6:2
           QSW[J]:=* * Y;   CNW[J]:=* * Y;   QNW[J]:=* * Y;                          005:02BB:5
   END UNTIL J:=*+1 GEQ 65;
```

```
   VALAVEFQ:=1;                ENDPT:=128;         J:=65;                        4             005:02BE:1
%                                                                                              005:02C0:3
LOOPER:VALAVEFQ:=* * 2;                                                                        005:02C0:3
   DO BEGIN                                                                                    005:02C2:0
         K:=1;                                                                   4             005:02C2:3
         DO BEGIN                                                                              005:02C2:4
         FREQ[J]:=* + FREQ[J+K];     OMEG4[J]:=* + OMEG4[J+K];                   5             005:02C2:4
         PS[J]:=* + PS[J+K];                                                                   005:02C8:0
         PSA[1,J]:=* + PSA[1,J+K];       PSA[2,J]:=* + PSA[2,J+K];                             005:02CA:4
         PSA[3,J]:=* + PSA[3,J+K];       PSA[4,J]:=* + PSA[4,J+K];                             005:02D1:3
         CSN[J]:=* + CSN[J+K];           QSN[J]:=* + QSN[J+K];                                 005:02D6:0
         CSW[J]:=* + CSW[J+K];           QSW[J]:=* + QSW[J+K];                                 005:02DD:2
         CNW[J]:=* + CNW[J+K];           QNW[J]:=* + QNW[J+K];                                 005:02E2:4
         JACOB[J]:=* + JACOB[J+K];       KONE[J]:=* + KONE[J+K];                               005:02E8:0
         KTWO[J]:=* + KTWO[J+K];                                                               005:02ED:2
         END UNTIL K:=*+1 GEQ VALAVEFQ;                                                        005:02F0:0
%                                                                                5             005:02F2:2
   Z:=1./VALAVEFQ;             X:=1./(VALAVEFQ*FSTEP);         Y:=-X;                          005:02F2:2
   BEGIN                                                                                       005:02F8:3
%  NOTE: THE "NATURAL" RIGHT-HANDED BUOY COORDINATES ARE NORTH & WEST.           5             005:02F8:3
%  BUT EAST AND NORTH ARE MORE CONVENTIONAL; SO, SINCE E. SLOPE=-W.SLOPE                       005:02F8:3
%  FOR NORMALIZATION MULTIPLY ALL CROSS SPECTRA INVOLVING W. SLOPE BY                          005:02F8:3
%  Y = -X  OR BY 2 IF ONLY AVERAGING IS DONE.                                                  005:02F8:3
          FREQ[J]:=* * Z;      OMEG4[J]:=* * Z;                PS[J]:=* * X;                   005:02F8:3
          PSA[1,J]:=* * X;     PSA[2,J]:=* * X;     PSA[3,J]:=* * X;                           005:02FE:0
          PSA[4,J]:=* * X;     CSN[J]:=* * X;       QSN[J]:=* * X;                             005:0304:4
          CSW[J]:=* * Y;       QSW[J]:=* * Y;  CNW[J]:=* * Y;QNW[J]:=**Y;                      005:030A:4
          JACOB[J]:=* * Z;     KONE[J]:=* * Z;      KTWO[J]:=* * Z;                            005:0312:0
   END;                                                                                        005:0317:3
   AVEPER[J]:=1./FREQ[J];                                                        5             005:0317:3
   END UNTIL J:=* + VALAVEFQ GEQ ENDPT;                                                        005:031B:0
%                                                                                4             005:031D:3
   ENDPT:=* * 2;                                                                               005:031D:3
   IF ENDPT LEQ NM1 THEN GO TO LOOPER;                                                         005:031F:0
%                                                                                              005:0320:2
END ELSE DO BEGIN                                                                              005:0320:2
     K:=1;                                                                       3             005:0320:5
     DO BEGIN                                                                                  005:0321:3
          FREQ[J]:=* + FREQ[J+K];     OMEG4[J]:=* + OMEG4[J+K];                  4             005:0321:3
          PS[J]:=* + PS[J+K];                                                                  005:0326:5
          PSA[1,J]:=* + PSA[1,J+K];       PSA[2,J]:=* + PSA[2,J+K];                            005:0329:3
          PSA[3,J]:=* + PSA[3,J+K];       PSA[4,J]:=* + PSA[4,J+K];                            005:032F:5
          CSN[J]:=* + CSN[J+K];           QSN[J]:=* + QSN[J+K];                                005:0336:5
          CSW[J]:=* + CSW[J+K];           QSW[J]:=* + QSW[J+K];                                005:033C:1
          CNW[J]:=* + CNW[J+K];           QNW[J]:=* + QNW[J+K];                                005:0341:3
          JACOB[J]:=* + JACOB[J+K];       KONE[J]:=* + KONE[J+K];                              005:0346:5
          KTWO[J]:=* + KTWO[J+K];                                                              005:034C:1
     END UNTIL K:=*+1 GEQ AVEFQ;                                                               005:034E:5
%                                                                                4             005:0351:1
       Z:=1./AVEFQ;            X:=1./(AVEFQ*FSTEP);             Y:= -X;                        005:0351:1
     BEGIN                                                                                     005:0357:3
          FREQ[J]:=* * Z;      OMEG4[J]:=* * Z;     PS[J]:=* * X;                4             005:0357:3
             PSA[1,J]:=* * X;   PSA[3,J]:=* * X;     PSA[4,J]:=* * X;                          005:035D:0
             CSN[J]:=* * Y;     QSN[J]:=* * X;    CSW[J]:=* * Y;                               005:0363:5
             QSW[J]:=* * Y;     CNW[J]:=* * Y;    QNW[J]:=* * Y;                               005:0369:2
          JACOB[J]:=* * Z;      KONE[J]:=* * Z;     KTWO[J]:=* * Z,                            005:036E:5
   END;                                                                                        005:0374:2
```

```
    AVEPER[1 +(J DIV AVEFQ)]:=1./FREQ[J];                              4        005:0374:2
END UNTIL J:=*+AVEFQ GEQ N - 1;                                                 005:0376:0
%                                                                      3        005:037A:5
%     CALCULATE WAVE DIRECTION AND RMS BEAM WIDTH.                              005:037A:5
IF LAV THEN BEGIN VALAVEFQ:=1;    ENDPT:=65;     END                            005:037A:5
       ELSE BEGIN VALAVEFQ:=AVEFQ;     ENDPT:=NM1;         END;        3        005:037D:1
%                                                                      3        005:037F:4
     K:=1;                                                                      005:037F:4
NDDIR:DO BEGIN                                                                  005:0380:2
       IF LAV THEN J:=K ELSE J:=1 + K DIV AVEFQ;                       3        005:0380:2
       C11:=PSA[1,K];       C33:=PSA[3,K];       C44:=PSA[4,K];                 005:0384:3
       C34:=CNW[K];         Q13:=QSN[K];         Q14:=QSW[K];                   005:0386:5
       C14:=CSW[K];         C13:=CSN[K];         Q34:=QNW[K];                   005:038C:2
       JACOBK:=JACOB[K];                 K11:=KONE[K];        K22:=KTWO[K];     005:038F:5
       IF Q13 NEQ 0. THEN ZETA:=ARCTAN2( Q14 , -Q13)                            005:0393:2
                          ELSE ZETA:=SIGN(-Q14)*PI/2;                           005:0395:2
       ZETA:=* * 180./PI;                                                       005:039C:1
%      ZETA IS THE DIRECTION OF PROPAGATION WRT N-W COORDINATES.                005:039F:4
%      BETA IS THE DIRECTION FROM WHICH THE WAVES ARE COMMING DEG. T.           005:039F:4
       BETA[J]:=180. - ZETA;                                                    005:039F:4
       K2:=(K1:=.04023840996*(FSTEP*K)**2)**2;                                  005:03A2:1
%      4 * PI**2/GEE = .04023840996.                                            005:03A6:4
       IF C11 NEQ 0. THEN                                                       005:03A6:4
       RR:=SQRT((C33 - C44)**2  +  4.*C34**2)/(K2*C11) ELSE RR:=0.;             005:03A7:2
       RLL[J]:=RR;                                                              005:03AE:1
       C:=(1. - RR)/(1. + RR);        IF C LSS 0. THEN C:=0.;                   005:03AF:3
       C:=2.*ARCTAN(SQRT(C));                                                   005:03B5:0
       CREST[J]:=180.*C/PI;                                                     005:03B8:2
%    CALCULATE FOURIER COEFFICIENTS OF DIRECTION FOR EA. FREQ.                  005:03BD:2
%      DIRECTIONAL COEFFICIENTS ARE WRT ANGLE ZETA NOT BETA.                    005:03BD:2
       A0[J]:=C11/PI;                                                           005:03BD:2
       A1[J]:=C13/(PI*K1);              B1[J]:=C14/(PI*K1);                     005:03C0:2
       A2[J]:=(C33 - C44)/(PI*K2);          B2[J]:=2.*C34/(PI*K2);              005:03C7:5
%                                                                               005:03D0:5
%      CALCULATE SPECTRAL MOMENTS AND RATIOS OF MOMENTS ETC.                    005:03D0:5
       NZERO:=C11*JACOBK;                % THIS IS N00.                         005:03D0:5
       N01   :=Q13*JACOBK/K11;           % THIS IS N01 (QSN).                   005:03D2:1
       N10   :=Q14*JACOBK/K11;           % THIS IS N10 (QSE).                   005:03D4:3
       N20   :=C44*JACOBK/K22;           % THIS IS N20 (CO-EE).                 005:03D5:5
       N02   :=C33*JACOBK/K22;           % THIS IS N02 (CO-NN).                 005:03D7:4
       N11   :=C34*JACOBK/K22;           % THIS IS N11 (CO-NE).                 005:03D9:3
%                                                                               005:03DB:2
       N12:=C14*JACOBK/K11;              % (CO-SE) & NA6[J].                    005:03DB:2
       N21:=C13*JACOBK/K11;              % (CO-SN) & NA7[J].                    005:03DD:1
       N22:=Q34*JACOBK/K22;              % (QNE) & NA8[J].                      005:03DF:0
%                                                                               005:03E0:5
       IF NZERO NEQ 0. THEN ANG1[J]:=ARCTAN2( N01 , N10 )                       005:03E0:5
       ELSE ANG1[J]:=0.;                                                        005:03E3:4
       ANG1[J]:=* * 180./PI;                                                    005:03E6:0
       IF NZERO NEQ 0. THEN                                                     005:03E8:2
            ANG2[J]:=.5*ARCTAN2( 2.*N11 , (N20 - N02)) ELSE                     005:03EC:0
            ANG2[J]:=0.;                                                        005:03F1:5
       ANG2[J]:=* * 180./PI;                                                    005:03F3:3
%                                                                               005:03F6:2
       DENOM1:=N10**2 + N01**2;                                                 005:03F8:2
       IF DENOM1 NEQ 0. THEN                                                    005:03FA:2
            RNEW[J]:=(N10*(N20 - N02) + 2.*N01*N11)/DENOM1 ELSE                 005:03FB:0
```

```
          RNEW[J]:=0.;                                                                    005:0400:5
%                                                                                         005:0402:3
       NNN[J]:=(2. + 4.*RNEW[J]/PI)/(1. - RNEW[J]/PI);                                    005:0402:3
%                                                                                         005:040C:4
       DENOM2:=SQRT(DENOM1);                                                              005:040C:4
       IF DENOM2 GTR 0. AND NNN[J] NEQ 0. THEN                                            005:040E:1
            XALPH[J]:=NZERO/(2.*PI*DENOM2) - (NNN[J]+2.)/NNN[J] ELSE                      005:0410:0
            XALPH[J]:=0.;                                                                 005:0417:2
%                                                                                         005:0419:0
       NA1[J]:=N01;  NA2[J]:=N10;  NA3[J]:=N20;  NA4[J]:=N02;                             005:0419:0
       NA5[J]:=N11;  NA6[J]:=N12;  NA7[J]:=N21;  NA8[J]:=N22;                             005:041E:2
%                                                                                         005:0423:4
       IF NZERO EQL 0. THEN BEGIN                                                         005:0423:4
       NA1[J]:=0.; NA2[J]:=0.; NA3[J]:=0.; NA4[J]:=0.; NA5[J]:=0.;              4         005:0424:5
       NA6[J]:=0.;   NA7[J]:=0.;   NA8[J]:=0.;                                            005:042A:4
       END ELSE BEGIN                                                                     005:042E:1
%      NOTE: THE RATIO OF THE AVERAGE MOMENTS ARE COMPUTED, NOT THE             4         005:042E:4
%           AVERAGE RATIO.                                                                005:042E:4
       NA1[J]:=*/NZERO;    NA2[J]:=*/NZERO;    NA3[J]:=*/NZERO;                           005:042E:4
       NA4[J]:=*/NZERO;    NA5[J]:=*/NZERO;                                               005:0434:1
       NA6[J]:=*/NZERO;    NA7[J]:=*/NZERO;    NA8[J]:=*/NZERO;                           005:0437:5
       END;                                                                               005:043C:2
     END UNTIL K:=* + VALAVEFQ GEQ ENDPT;                                       4         005:043D:2
%                                                                               3         005:043F:5
     IF (LAV AND ENDPT EQL 65) THEN BEGIN                                                 005:043F:5
        K:=ENDPT;        VALAVEFQ:=* * 2;    ENDPT:=128;     END ELSE           3         005:0441:4
     IF LAV THEN BEGIN K:=ENDPT+1;  VALAVEFQ:=* * 2;  ENDPT:=* * 2;END;         3         005:0445:0
     IF (LAV AND ENDPT LEQ NM1) THEN GO TO ADDIR;                               3         005:044A:4
%                                                                                         005:044C:3
     K:=1;          WEST:=1;                                                              005:044C:3
     WHILE AVEPER[K] GTR .985 DO BEGIN WEST:=*+AVEFQ;    K:=*+1;   END;                   005:044D:5
%    A .985 SEC. PERIOD WAVE HAS A WAVELENGTH EQL BUOY DIA.=5 FT.               3         005:0454:0
     WEST:=NM1;   % TO PRINT OUT ALL VALUE UP TO NYQUIST.                                 005:0454:0
%                                                                                         005:0455:0
     IF TABLE1 THEN WRITE(PRINTER[SKIP 1]);                                               005:0455:0
     IF TABLE2 THEN BEGIN                                                                 005:0458:0
     WRITE(PRINTER,<"NORMALIZED AND AVERAGED CROSS SPECTRA",/>);                3         005:045B:5
     WRITE(PRINTER,F9);      WRITE(PRINTER,F10);   WRITE(PRINTER,F11);                    005:0461:2
     WRITE(PRINTER,F15);                                                                  005:0471:5
     K:=1;                                                                                005:0477:2
     IF LAV THEN BEGIN       VALAVEFQ:=1;         ENDPT:=65;                              005:0478:0
 TAB2:WRITE(PRINTER,F12,DO [2.*PI*FREQ[K],AVEPER[K],                            4         005:047A:2
         PS[K],PSA[1,K],PSA[3,K],PSA[4,K],CSN[K],QSN[K],CSW[K],QSW[K],                    005:0488:5
         CNW[K],QNW[K],BETA[K],CREST[K]] UNTIL K:=*+VALAVEFQ GEQ                          005:049E:2
         ENDPT);                                                                          005:04AA:0
     IF ENDPT EQL 65 THEN BEGIN K:=65;  VALAVEFQ:=* * 2; ENDPT:=128;                      005:04AF:1
     END ELSE BEGIN K:=ENDPT+1;  VALAVEFQ:=* * 2;  ENDPT:=* * 2;   END;         5         005:04B3:4
     IF ENDPT LEQ NM1 THEN GO TO TAB2;                                          5         005:04B8:3
%                                                                                         005:04B9:5
     END ELSE                                                                             005:04B9:5
     WRITE(PRINTER,F12,DO [2.*PI*FREQ[K], AVEPER[1 + (K DIV AVEFQ)],            4         005:04B9:5
         PS[K],PSA[1,K],PSA[3,K],PSA[4,K],CSN[K],QSN[K],CSW[K],QSW[K],                    005:04C9:4
         CNW[K],QNW[K],BETA[1 + (K DIV AVEFQ)],CREST[1+(K DIV AVEFQ)]]                    005:04DF:2
     UNTIL K:=* + AVEFQ GEQ WEST);                                                        005:04E8:1
     END;                                                                                 005:04F2:1
     IF ((TABLE2 AND TABLE3) OR (TABLE1 AND TABLE3)) THEN                       3         005:04F2:1
     WRITE(PRINTER[SKIP 1]);                                                              005:04F4:0
```

```
      J:=1;          K:=1;                                                                005:04F9:4
      WHILE AVEPER[J] GTR TLIM DO J:=*+1;                                                 005:04FB:0
      WHILE AVEPER[K] GTR .989 DO K:=*+1;                                                 005:04FE:3
      IF TABLE3 THEN BEGIN                                                                005:0503:3
      WRITE(PRINTER,F13,TLIM);          WRITE(PRINTER,F14);                  3            005:0514:2
      WRITE(PRINTER,F5);                                                                  005:0517:4
      IF (NOT LAV) THEN                                                                   005:051D:0
      WRITE(PRINTER,F8, DO [J,AVEPER[J],A0[J],A1[J],A2[J],                                005:051D:2
          B1[J],B2[J], BETA[J],CREST[J]]  UNTIL J:=*+1 GEQ K)                             005:0531:5
      ELSE BEGIN                                                                          005:053D:2
         VALAVEFQ:=1;                ENDPT:=65;                              4            005:0543:0
    TAB3:WRITE(PRINTER,F6,DO [J,AVEPER[J],A0[J],A1[J],A2[J],B1[J],B2[J],                  005:0544:3
                         BETA[J],CREST[J]] UNTIL J:=*+VALAVEFQ GEQ ENDPT);                005:055D:5
      IF ENDPT EQL 65 THEN BEGIN J:=65;  VALAVEFQ:=* * 2; ENDPT:=128;                     005:0569:4
      END ELSE BEGIN                                                         5            005:056E:1
         J:=ENDPT+1;     VALAVEFQ:=* * 2;          ENDPT:=* * 2;    END;     5            005:055E:4
         IF ENDPT LEQ NM1 THEN GO TO TAB3;                 END;              5            005:0573:0
      END;                                                                   4            005:0574:2
%FOR TABLE 4 THERE IS AN UNCONDITIONAL PRINT.                                3            005:0574:2
WRITE(PRINTER[SKIP 1]);                                                                   005:0574:2
WRITE(PRINTER,F16);             K:=1;                                                     005:0573:3
WRITE(PRINTER,F17,DO [2.*PI*FREQ[K],NA1[1+K DIV AVEFQ],                                   005:057F:4
         NA2[1+K DIV AVEFQ],                                                              005:058E:4
         NA3[1+K DIV AVEFQ],NA4[1+K DIV AVEFQ],NA5[1+K DIV AVEFQ],                        005:0592:1
         ANG1[1+K DIV AVEFQ],ANG2[1+K DIV AVEFQ], RNEW[1+K DIV AVEFQ],                    005:059C:4
         RLD[1 + K DIV AVEFQ],                                                            005:05A7:1
         NNN[1+K DIV AVEFQ],XALPH[1+K DIV AVEFQ]] UNTIL K:=*+AVEFQ                        005:05AA:4
         GEQ NM1);                                                                        005:05B2:2
%                                                                                         005:05B8:4
%        TABLE 5 IS ALSO AN UNCONDITIONAL PRINT.                                          005:05B8:4
         WRITE(PRINTER[SKIP 1]);                                                          005:05B8:4
         WRITE(PRINTER,<"THESE MOMENT RATIOS SHOULD BE SMALL; NCHECK=",                   005:05BD:5
                   "N01 + N10 (SHOULD EQL 1.0).",/>);                                     005:05BF:4
         WRITE(PRINTER,F18);             K:=1;                                            005:05C3:2
         WRITE(PRINTER,F19,DO [2.*PI*FREQ[K],NA6[1+K DIV AVEFQ],                          005:05C9:3
                   NA7[1+K DIV AVEFQ],NA8[1+K DIV AVEFQ],                                 005:05D8:4
                   NA1[1+K DIV AVEFQ] + NA2[1+K DIV AVEFQ]]                               005:05DF:4
         UNTIL K:=*+AVEFQ GEQ NM1);                                                       005:05F4:4
%                                                                                         005:05FB:4
       IF PUNCHKS THEN BEGIN                                                              005:05E6:4
       J:=1;                                                                 3            005:05EC:3
       IF (NOT LAV) THEN                                                                  005:05ED:1
       WRITE(PUNCHER,<8E10.2>, DO [CSN[J],QSN[J],CSW[J],QSW[J],CNW[J],                    005:05ED:3
               QNW[J],PS[J],FREQ[J]] UNTIL J:=*+AVEFQ GEQ NM1)                            005:0602:2
       ELSE BEGIN                                                                         005:0608:3
          VALAVEFQ:=1;               ENDPT:=65;                              4            005:0611:1
   PUN1:WRITE(PUNCHER,<8E10.2>, DO [CSN[J],QSN[J],CSW[J],QSW[J],CNW[J],                   005:0612:4
               QNW[J],PS[J],FREQ[J]] UNTIL J:=*+VALAVEFQ GEQ ENDPT);                      005:0627:2
       IF ENDPT EQL 65 THEN BEGIN J:=65;  VALAVEFQ:=* * 2; ENDPT:=128;                    005:0635:4
       END ELSE BEGIN                                                        5            005:063A:1
          J:=ENDPT+1;    VALAVEFQ:=* * 2;          ENDPT:=* * 2;    END;     5            005:063A:4
     .    IF ENDPT LEQ NM1 THEN GO TO PUN1;                 END;             5            005:063F:0
       END;          IF PUNCHPS THEN BEGIN                                   4            005:0640:2
       IF (NOT LAV) THEN  BEGIN                                              3            005:0641:1
       J:=1;WRITE(PUNCHER,<8E10.2>,DO PSA[1,J] UNTIL J:=*+AVEFQ GEQ NM1);    4            005:0642:0
       J:=1;WRITE(PUNCHER,<8E10.2>,DO PSA[3,J] UNTIL J:=*+AVEFQ GEQ NM1);                 005:0654:1
       J:=1;WRITE(PUNCHER,<8E10.2>,DO PSA[4,J] UNTIL J:=*+AVEFQ GEQ NM1);                 005:0666:4

       END ELSE BEGIN          FOR K:=1, 3, 4   DO BEGIN                                  005:0679:1
          VALAVEFQ:=1;              ENDPT:=65;              J:=1;            5            005:0680:2
  PUN2:WRITE(PUNCHER,<8E10.2>,DO PSA[K,J] UNTIL J:=*+VALAVEFQ GEQ ENDPT);                 005:0662:3
       IF ENDPT EQL 65 THEN BEGIN J:=65;  VALAVEFQ:=* * 2; ENDPT:=128;                    005:0694:4
       END ELSE BEGIN                                                        6            005:0699:1
          J:=ENDPT+1;    VALAVEFQ:=* * 2;          ENDPT:=* * 2;    END;     6            005:0699:4
       IF ENDPT LEQ NM1 THEN GO TO PUN2;                  END;      END;     6            005:069E:0
       END;          IF PUNCHDS THEN BEGIN                                   4            005:06A2:3
       IF (NOT LAV) THEN BEGIN                                               3            005:06A3:2
       K:=NM1 DIV AVEFQ;                                                     4            005:06A4:1
       J:=1;WRITE(PUNCHER,<8E10.2>,DO [FREQ[1+(J-1)*AVEFQ],BETA[J],                       005:06A5:4
       CREST[J],A0[J],A1[J],A2[J],B1[J],B2[J]] UNTIL J:=*+1 GEQ K);                       005:06B4:2
       END ELSE BEGIN                                                                     005:06CA:0
          VALAVEFQ:=1;              ENDPT:=65;          J:=1;                4            005:06CA:3
   PUN3:WRITE(PUNCHER,<8E10.2>, DO [FREQ[J],BETA[J],                                      005:06CC:4
       CREST[J],A0[J],A1[J],A2[J],B1[J],B2[J]] UNTIL J:=*+VALAVEFQ GEQ                    005:06D9:5
       ENDPT);                                                                            005:06EA:3
       IF ENDPT EQL 65 THEN BEGIN J:=65;  VALAVEFQ:=* * 2; ENDPT:=128;                    005:06EF:4
       END ELSE BEGIN                                                        5            005:06F4:1
          J:=ENDPT+1;    VALAVEFQ:=* * 2;          ENDPT:=* * 2;    END;     5            005:06F4:4
       IF ENDPT LEQ NM1 THEN GO TO PUN3;          END;                       5            005:06F9:0
       END;                                                                  4            005:06FA:2
  FINI:END;          % END OF MAIN BLOCK.                                    3            005:06FA:2
                                                                          BLOCK(005) IS 06FC LONG
                                                                       STACKCODE IS SEGMENT 00019
                                                                      STACKCODE(019) IS 005F LONG
  D.                                                                   2            002:0377:5
                                                                          BLOCK(002) IS 0094 LONG
                                                                                DATA IS 005C LONG
```

COMPONENTS

The following is a list of the major components of the buoy, procurement source, and approximate cost in 1972.

1. Battery: Lead-Acid 12 volts, 20 ampere hour capacity (6 required) — $40 ea
   Model GC12200
   Globe Battery Division
   P.O. Box 591
   Milwaukee, Wisconsin 53201

2. Inverter: 12 VDC to 115 VAC 400 Hz 100 watt capacity — $250
   Model 12400-12(w/input-output filters)
   Nova Electric Manufacturing Co.
   263 Hillside Ave.
   Nutley, New Jersey 07100

3. Radio Transmitter: Citizen band (Marine data channels) 500 mw, modulated — $150
   Model BT-109 (modified)
   Ocean Applied Research Corp.
   10475 Roselle Street
   San Diego, California 92121

4. Visual Beacon: Flashing Xenon Lamp — $200
   0.5 flashes/sec, self contained batteries
   Model
   Ocean Applied Research Corp.
   10475 Roselle Street
   San Diego, California 92121

5.	Vertical Gyro/Accelerometer Assy: Includes Honeywell Model GG326C1 Quartz fiber accelerometer mounted on inner gimbal of Honeywell Model JG7044A45 vertical accelerometer Model GG1134AA01 Honeywell Inc. Aerospace Division Dept 691 2600 Ridgeway Rd. Minneapolis, Minnesota 55413	$6000
6.	Directional Gyroscope: North seeking Model DG04-0122-1 Humphrey, Inc. 9212 Balboa St. San Diego, California	$700
7.	Data Recorder System	$4000
	A. Data Recorder/Logic Assembly Model 10015 Monitor Labs, Inc. 10451 Roselle St. San Diego, California 92121	
	B. Incremental Digital Tape Recorder Model 1387 Precision Instrument Co. 3170 Porter Drive Palo Alto, California	$2500

C. Low-Power Analog-to-Digital Converter $900
Model ADC-12QL
Analog Devices, Inc.
P.O. Box 280
Norwood, Mass. 02062

D. Low-drift, low-power operational amplifier $50
Model 153J
Analog Devices, Inc.
P.O. Box 280
Norwood, Mass. 02062

8. Buoy Hull Assy $10,000
No Part Number
Honeywell Marine Systems Center
5303 Shilshole Ave., N.W.
Seattle, Washington 98107

9. Radio Transmitter Antenna: 3 needed $75 ea
Model AW257
Ocean Applied Research Corp.
10475 Roselle St.
San Diego, Ca. 92121