NASA CONTRACTOR REPORT

NASA CR-17

C.1

NASA CR-1776

DOPPLER RADAR SIMULATION STUDIES

by William D. Stanley

Prepared by
OLD DOMINION UNIVERSITY
Norfolk, Virginia
for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • OCTOBER 1971

1. Report No.
NASA CR-1776

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
DOPPLER RADAR SIMULATION STUDIES

5. Report Date
October 1971

6. Performing Organization Code

7. Author(s)
William D. Stanley

8. Performing Organization Report No.
None

10. Work Unit No.

9. Performing Organization Name and Address
Old Dominion University
School of Engineering and Research Foundation
5201 Hampton Boulevard
Norfolk, VA 23508

11. Contract or Grant No.
NGR 47-003-015

13. Type of Report and Period Covered
Contractor Report

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Washington, DC 20546

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

A study of digital simulation and possible implementation techniques for Doppler radar altimeter signal processing is presented. A technique is described for generating simulated Doppler signals by employing a random number generator and a digital filter. Representative Doppler spectra are presented and are used in studying the measurement techniques considered.

The development of a digital phase-locked concept is given. The simulation of such system on a digital computer is developed, and suggestions for possible implementation are presented.

A study of processing Doppler signals with a Fast Fourier Transform spectrum processor is presented. Results of a computer simulation of a processor are discussed, and representative processed Doppler spectra are included.

17. Key Words (Suggested by Author(s))
Doppler radar
Doppler simulation
Digital techniques
Signal processing
Fast Fourier Transform
Frequency trackers
Digital filters

18. Distribution Statement
Unclassified - Unlimited

19. Security Classif. (of this report)
Unclassified

20. Security Classif. (of this page)
Unclassified

21. No. of Pages
120

22. Price*
$3.00

*For sale by the National Technical Information Service, Springfield, Virginia 22151

PREFACE

The work reported herein was performed by the School of Engineering of Old Dominion University for the Langley Research Center, NASA, Hampton, Virginia, as the final phase of NASA Grant NGR #47-003-015. The principal investigator was Dr. William D. Stanley of ODU, and Mr. Wayne R. Powell served as graduate assistant. This is the second and final comprehensive report on the grant. An earlier report entitled "Statistical Analysis of a Planetary Radar Altimeter Measuring Unit", dated September 1969, was submitted to NASA to describe the first phase of the grant.

Although this report should be interpreted as a "final report" for administrative purposes, it is likely that still another report may be delivered sometime after the expiration date of the grant. A separate problem dealing with the variation of radar return magnitude as a function of planetary surface specularity and altitude has been assigned as a master's thesis to the graduate assistant and will be supervised by the principal investigator. Inasmuch as this problem was generated during the grant, suitable copies of the thesis will be delivered to Langley Research Center after completion of this study.

The principal investigator would like to acknowledge the advice and assistance of Mr. W. T. Bundick, Mr. W. A. Southall, Mr. T. M. Walsh, and Mr. R. L. Kenimer, all of the Flight Instrumentation Division of LRC.

TABLE OF CONTENTS

TABLE OF CONTENTS (CONT.)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (CONT.)

INTRODUCTION

The purpose of this investigation was to explore the use of certain digital (or discrete) signal processing techniques for both the simulation and implementation of planetary Doppler radar altimeter systems. In order to predict accuracy requirements and the level of performance expected in remote planetary missions, it is desirable to employ a considerable amount of simulation in determining total system requirements. Suitable digital models of critical measurement processes are valuable in developing appropriate computer simulations.

With the increasing application of digital hardware and techniques, it is expected that improvements can be made on conventional analog systems by replacing some of the previous analog functions by digital processes. Throughout this investigation the concept of implementing actual systems with digital hardware was placed on an equal level of importance with the goal of developing digital simulation models.

The main body of this report is divided into three sections, chosen to correspond to three separate aspects of the investigation. The first section describes the simulation of Doppler return signals. The Doppler signal models were obtained by employing a Gaussian random number generation program and digital filters whose frequency response acts to provide the proper spectral shape to the simulated Doppler.

The next section describes the investigation of digital phase-locked loop frequency trackers. The first model considered was centered around a close correspondence with analog phase-locked loops, thus making it suitable for computer simulation. In addition, a model more suitable for direct digital implementation employing a zero-crossing error detector was investigated.

The last main section of the report describes the investigation of the Fast Fourier Transform concept for application in Doppler signal processing. This powerful concept could permit the actual real-time spectral analysis of certain Doppler signals, thus permitting maximum information regarding the spectrum to be available. The computer simulation of such a system will be described, and typical computer runs will be shown. Results obtained from performing certain statistical evaluations of the data will be discussed.

DOPPLER SIMULATION

Doppler Signal Assumptions

The Doppler shift results in a return signal (echo) having different spectral properties from the transmitted signal. If the relative velocity between the radar and the target were constant, and if the antenna beam were infinitesimally small, then the spectrum of the return signal would merely be a simple translated version of the transmitted signal, and measurement of the Doppler shift could be reasonably straightforward and accurate. In practice, however, the spectrum of a Doppler return signal undergoes a considerable modification, mainly in the form of spectral dispersion.

The most significant contributing factors to the spectral modification are as follows: (a) The nonzero beamwidth of the antenna results in a spreading of the radiated signal over some area of the planetary surface. The signal backscattered at each point on the illuminated area has a statistically random amplitude and phase with respect to the incident signal. Furthermore, the relative Doppler shift will be different at different points within the area because of different relative velocities between the surface and radar. The antenna beam "integrates" all the complex components to yield the actual return signal.

(b) Relative acceleration between radar and surface results in an additional frequency modulation effect which further modifies the spectrum.

The variation in backscattered signal as a function of incident angle varies considerably with the nature of the surface roughness. The spread of the beam on the surface depends heavily on the orientation of the beam with respect to the surface and the beamwidth. The result of these various effects is that the Doppler return is a rather complex phenomena, which can only be treated in a statistical sense.

The first unclassified treatment of Doppler airborne velocity measurements was presented by Berger (ref. 4). Further work by Berger (refs. 5 and 6) and Fried (refs. 22 and 23) established many basic concepts that have been widely employed in designing Doppler systems.

The basic assumptions that are widely used in evaluating Doppler errors were discussed by Berger (ref. 4). Most of these assumptions are concerned with the nature of the power spectrum. Notable among these assumptions are: (a) The Doppler spectrum is equivalent to the spectrum obtained by passing white noise through a bandpass filter. (b) The probability density function of the instantaneous frequency has approximately the same width as the power spectrum of the Doppler signal. Based on these and similar assumptions, Berger developed expressions for various error bounds. However, Berger noted that no rigorous general proof of these postulates had been found.

The earliest Doppler systems were of the continuous-wave (CW) variety. A significant development was the FM Bessel Sideband Doppler system described by Glegg (ref. 26). This system is capable of achieving greater immunity from noise than the CW system. The FM system was later extended in capability to the measurement of both velocity and range by Fried (ref. 24).

Regardless of the type of Doppler system employed, the fundamental operation that must be performed within the receiver is to measure in a statistical sense some pertinent characteristic of the power spectrum of the return signal. The appropriate characteristic could conceivably be the center of the spectrum, the frequency having peak power, or some otner suitable property.

As a starting point in this investigation, it was desirable to develop some simple computer programs for simulating Doppler returns. For the purpose of testing the phase-lock loop system to be described later, step and ramp function routines were first written.

Doppler Step and Ramp Simulations

Whenever a transient type waveform such as a step or ramp is used to excite a system for testing purposes, it is desirable that the system be in a steady-state condition initially. Since a step or ramp usually represents a deviation in frequency from some initial frequency, it is desirable to develop a routine that manually allows the operator to switch on the appropriate waveform at a proper reference time, t=o.

Let ω_c represent the reference radian center frequency, and let d_o represent the cyclic Doppler step input. The step function of Doppler may be implemented by the equations

$$\begin{aligned} x(t) &= \sin(\omega_c t+\phi) \qquad \text{for } t<o \\ &= \sin(\omega_c t+2\pi d_o t+\phi) \qquad \text{for } t>o \end{aligned} \tag{1}$$

where ϕ is an arbitrary phase, and the amplitude of the sinusoid is normalized to unity.

For a ramp input, let d_1 represent the slope of the deviation versus time (measured in Hz/sec). The ramp function of Doppler shift may be realized by the equations

$$\begin{aligned} x(t) &= \sin(\omega_c t+\phi) \qquad \text{for } t<o \\ &= \sin(\omega_c t+\pi d_1 t^2+\phi) \qquad \text{for } t>o \end{aligned} \tag{2}$$

Second-Order Doppler Simulation

The basic concept employed in developing a Doppler simulation computer program is that such a signal can be thought of as arising from passing white noise through a band-pass filter. On the digital computer "white noise" can be obtained from a random number generator program with Gaussian amplitude statistics. On the IBM 1130, a subroutine named GAUSS has the necessary characteristics to accomplish the desired purpose.

The spectrum of the random number generator was modified to represent approximate Doppler spectra by means of various digital filters. The use of digital filters in various signal processing applications has been receiving considerable attention in recent years (refs. 13, 17, 18, 20, 27, 28, 30, 31, 32, 33, 37, 38, 40, 42, 43, 44, 45, 47, 52, 53, 54, 58, 59, 60, 67, 69, 72, 75). The theoretical basis for digital filter design is the relationship between continuous and discrete linear systems. The z-transform was developed as a powerful analytical tool for describing discrete systems, and it has been used extensively in sampled-data system analysis and design (refs. 34, 35, 36, 41, 46, 55, 73). Many of the concepts developed in the literature on sampled-data systems are directly applicable to digital filter design.

The particular approach used in developing the Doppler simulation filters was the bilinear transformation method. This method can be made to yield good correlation between the behavior of the digital filter and a corresponding analog filter, thus permitting a large part of the design process to be performed in the analog domain.

The first digital filter employed was a second-order band-pass filter having a variable bandwidth (or Q) and a variable center frequency. If a flat noise spectrum excites such a filter, then the output spectrum must have the same characteristic as the filter bandpass shape. By varying the center frequency, the effect of a change in velocity may be represented, and by changing the Q, the effects of narrower or wider Doppler returns may be simulated.

For convenience, the center frequency of the analog transfer function is normalized to ω=1. In terms of a reference analog complex frequency p, the transfer function $H_d(p)$ is given by

$$H_d(p) = \frac{p/Q_d}{p^2+p/Q_d+1} \tag{3}$$

where Q_d is defined as

$$Q_d = \frac{\text{Center Frequency}}{\text{3dB Bandwidth}} \tag{4}$$

This response is identical with that of a simple series or parallel resonant circuit.

The analog filter may be mapped to the digital domain by means of the transformation

$$p = C_d\frac{(1-z^{-1})}{1+z^{-1}} \tag{5}$$

where C_d is a mapping constant and z is the z-transform variable. Let s represent the final Laplace transform variable of the digital filter. Assume that the samples of the time signal are spaced T seconds apart. Then z and s are related by

$$z = \varepsilon^{sT} = \varepsilon^{s/f_s} \tag{6}$$

where f_s is the sampling frequency (1/T).
Let $s=j\omega$ represent the imaginary axis of the s-plane and let $p=j\lambda$ represent the imaginary axis of the p-plane. These axes are related by

$$\lambda = C_d \tan\frac{\omega T}{2} = C_d \tan\pi\frac{f}{f_s} \tag{7}$$

Let f_d represent the center frequency of the Doppler signal. Since this frequency must correspond to $\lambda=1$, the constant C_d is

$$C_d = \cot\pi\frac{f_d}{f_s} \tag{8}$$

The z-domain transfer function may be represented as

$$H_d(z) = \frac{a_{do}(1-z^{-2})}{1+b_{d1}z^{-1}+b_{d2}z^{-2}} \tag{9}$$

where

$$a_{do} = \frac{C_d/Q_d}{1+C_d{}^2+C_d/Q_d} \tag{10}$$

$$b_{d_1} = \frac{2-2C_d{}^2}{1+C_d{}^2+C_d/Q_d} \tag{11}$$

$$b_{d_2} = \frac{1+C_d{}^2-C_d/Q_d}{1+C_d{}^2+C_d/Q_d} \tag{12}$$

Let r(n) represent the output of the random number generator, and let x(n) represent the output of the Doppler filter. The algorithm for generating x(n) is

$$x(n) = a_{do}[r(n)-r(n-2)] - b_{d_1}x(n-1)-b_{d_2}x(n-2) \tag{13}$$

The spectrum generated by this filter possesses geometric symmetry about the center frequency. On a linear scale, the rolloff above band-center is more gradual than the rolloff below band-center. This might be representative of certain types of skewed Doppler spectra. The result is that the mean frequency is somewhat higher than the peak-amplitude frequency. On the other hand, for a very narrow-band case, corresponding to a high Q, the response is very close to being arithmetically symmetrical.

Computer Simulation

A second-order Doppler simulation subroutine in FORTRAN IV is included in the Fast Fourier Transform Program shown in Appendix B. It is the portion of the program with the title SUBROUTINE KAY(NPPC,NTOT,Q). It is necessary (and often convenient) to modify the notation when writing such a program. Extensive computer runs were made in order to obtain data for use in other parts of this investigation. Many of these runs were plotted on an IBM 1627 XY Plotter. Some typical data obtained in this fashion are shown in Figures 13 through 22.

The parameter Q_d defined in (4) is of interest only as a means of determining the relative bandwidth of the Doppler signal. A more convenient parameter for characterizing the graphical displays is a parameter BW that will be defined as

$$\begin{aligned} BW &= \frac{\text{3dB Bandwidth}}{\text{Center Frequency}} \times 100\% \\ &= \frac{1}{Q_d} \times 100\% \end{aligned} \tag{14}$$

It will be observed that the figures are labeled with the appropriate value of BW in each case.

DIGITAL FREQUENCY TRACKER

General Discussion

An investigation of a digital phase-locked loop frequency tracker model and its digital computer simulation will be described. The reasons for this investigation were twofold: (a) In order to study the dynamic behavior of typical planetary altimeters under simulated conditions, it is desirable to have a suitable computer model whose characteristics can be programmed to correspond, within reasonable bounds, to the typical type of frequency trackers employed in actual altimeters.
(b) With the increasing availability of large-scale integrated digital circuits, it seems feasible that a digital phase-locked frequency tracker could be developed to have advantages over existing analog units. Although it may not be the best approach to merely attempt to "copy" the corresponding functions of the analog unit in constructing a digital hardware unit, it should add clarity and design assistance to work initially with the concept of employing a close similarity between the functions of each unit. Certainly for the simulation purpose, this approach would be the appropriate one.

In general, Doppler shifts may be either positive or negative representing, respectively, relative velocities either in the direction of or away from the radar system. In noncoherent systems it is necessary to employ two channels with local carriers phased 90° with respect to each other, in order to detect the sense of the Doppler shift. In order to make the digital simulation as simple as possible, only a single channel tracker was simulated, with the direction of the shift assumed to be known. Presumably, this assumption should not limit the accuracy of any measurement since the determination of the direction of the Doppler shift is a different problem from the measurement of the Doppler spectrum itself.

Analog Phase-Locked Loop

There have been several variations in the actual implementation and arrangement of the basic phase-locked loop frequency tracker. However, aside from the single channel assumption previously discussed, all analog units observed seem to essentially reduce to the basic phase-locked loop discriminator system widely used in communications systems. The references on phase-lock loops are quite extensive, and no attempt will be made here to document the literature on the subject. The primary source used in obtaining data in developing the model was the JPL report of Tausworthe (ref. 71).

The basic analog model considered in this investigation is shown in block form in Fig. 1a. The input signal x(t) is multiplied by the VCO output v(t), and the low-frequency portion of this signal is applied to the loop filter. The output of the loop filter $\hat{d}(t)$ represents an averaged estimate of the Doppler shift, which is simultaneously applied to the input of the VCO. By means of the feedback mechanism, the frequency of the VCO can be made to track very closely the statistical "average" of the input Doppler signal. This shift can be actually measured via the bias $\hat{d}(t)$ existing at the input to the VCO.

Although the analysis of this action is widely known, a brief review will be presented here in order to provide a basis for the discussion to follow. Assume that the input x(t) is the normalized function

$$x(t) = \sqrt{2}\,\sin[\omega_C t+\theta_1(t)] \qquad (15)$$

where $f_C=\omega_C/2\pi$ is the cyclic center frequency of the loop and $\theta_1(t)$ is the input phase variation. The phase variation is related to the input Doppler variation d(t) by

$$\theta_1(t) = 2\pi\int d(t)\,dt \qquad (16)$$

Assume that the feedback signal v(t) is of the normalized form

$$v(t) = \sqrt{2}\,\cos[\omega_C t+\theta_2(t)] \qquad (17)$$

where $\theta_2(t)$ is the feedback phase variation. Let d(t) represent the output Doppler estimate. Then $\theta_2(t)$ and $\hat{d}(t)$ are related by

$$\theta_2(t) = 2\pi\int \hat{d}(t)dt \tag{18}$$

The ideal output of the multiplier $e_1(t)$ would be the product of x(t) and v(t) as given by

$$\begin{aligned} e_1(t) = \; & \sin[\theta_1(t)-\theta_2(t)] \\ + \; & \sin[2\omega_C t+\theta_1(t)+\theta_2(t)] \end{aligned} \tag{19}$$

In all treatments of the phase-locked loop observed in the literature, the assumption is made that the second harmonic term in (19) can be eliminated due to either or both of the following two reasons: (a) Practical multipliers are frequency limited, and if the frequency is high enough, the multiplier will not respond to this second-harmonic term. (b) The bandwidth of the loop is so low compared with the second harmonic of the carrier frequency that this component will have negligible transmission through the filter.

While both of these assumptions are apparently valid in high frequency phase-locked loops designed for standard FM demodulation, in the case of low-frequency wide-band Doppler trackers, one or both assumptions may not be appropriate. This point will be discussed more fully in the digital simulation development later.

Continuing with the original development and considering only the low frequency term, this error e(t) is

$$e(t) = \sin[\theta_1(t)-\theta_2(t)] \qquad (20)$$

The loop filter then "smoothes" the error e(t) to yield an average estimate $\hat{d}(t)$ of the Doppler shift.

A widely employed linearized model of the phase-locked loop may be obtained by assuming that the error $\theta_1-\theta_2$ is sufficiently small so that the sine term in (20) may be approximated by its argument. With this assumption an s-plane model for the system can be developed in the form of Fig. 1b. The following terms are defined:

$d(t)$ = input Doppler cyclic frequency

$D(s) = L[d(t)]$ = Laplace transform of $d(t)$

$\hat{d}(t)$ = estimate of Doppler cyclic frequency

$\hat{D}(s) = L[\hat{d}(t)]$

$e(t)$ = Phase Error = $\theta_1-\theta_2$ in <u>linear</u> model

$E(s) = L[e(t)]$

$H_e(s)$ = transfer function of loop filter

It is sometimes convenient to replace the block diagram of Fig. 1b by the externally-equivalent unity feedback form of Fig. 1c. This model is very useful for determining input-output relationships, but it must be used with caution when computations inside the loop are made. For example, later in this report, an analysis will be made of the phase error. If the unit feedback form is employed, it is necessary to define the phase error at the midpoint between the $2\pi/s$ and $H_e(s)$ blocks.

Second-Order Analog Loop

The most widely used analog loop filter appears to be the first-order transfer function given by

$$H_e(s) = \frac{K_O{}^1(1+T_2s)}{1+T_1s} \tag{21}$$

However, in the analysis of such a loop, the approximation that $T_1 >> T_2$ is usually made so that the filter behaves approximately like a single lead coupled with a pure integration. In view of the eventual goal of digital implementation, it seems logical that the correspondence of pure integration (pure summation) should be considered, and hence the actual loop filter that will be assumed in this analysis is of the form

$$H_e(s) = \frac{K_O(1+s\tau)}{s} \tag{22}$$

The closed loop transfer function G(s) is given by

$$G(s) = \frac{2\pi K_O(1+s\tau)}{s^2+2\pi K_O\tau s+2\pi K_O} \tag{23}$$

Observe that the <u>effective</u> gain constant is actually 2π times the actual loop filter gain constant. For convenience, an effective gain constant K will be defined as

$$K = 2\pi K_o \tag{24}$$

With this definition the transfer function is

$$G(s) = \frac{K(1+s\tau)}{s^2+K\tau s+K} \tag{25}$$

Suppose it is desired that the quadratic in the denominator of G(s) have a damping ratio ζ and a natural radian frequency ω_o. It can be readily shown that the effective gain constant and numerator tine constant required are given by

$$K = \omega_o{}^2 \tag{26}$$

$$\tau = \frac{2\zeta}{\omega_o} \tag{27}$$

Although the linear model is not particularly accurate for relatively large errors (approaching $\pi/2$ radians), it is, nonetheless, interesting and meaningful for approximate purposes to determine the nature of the phase error predicted from the linear model as a function of simplified input signals. With this goal in mind the transfer function relating the phase error E(s) to the Doppler shift D(s) is given by

$$\frac{E(s)}{D(s)} = \frac{2\pi s}{s^2+K\tau s+\omega_o{}^2} = \frac{2\pi s}{s^2+2\zeta\omega_o s+\omega_o{}^2} \tag{28}$$

Assume that the input Doppler d(t) is a <u>step</u> frequency shift of magnitude d_o(in Hz). The time domain error $e_s(t)$

for an underdamped loop is given by

$$e_s(t) = \frac{d_o}{f_o} \frac{\varepsilon^{-\zeta\omega_o t}}{\sqrt{1-\zeta^2}} \sin\omega_o\sqrt{1-\zeta^2}t \tag{29}$$

where $f_o=\omega_o/2\pi$. The maximum phase error E_{sm} for the step input occurs at a time T_{sm}. The relationships can be shown to be

$$\omega_o T_{sm} = \frac{\cos^{-1}\zeta}{\sqrt{1-\zeta^2}} \tag{30}$$

$$E_{sm} = \frac{d_o}{f_o} \varepsilon^{\frac{-\zeta\cos^{-1}\zeta}{\sqrt{1-\zeta^2}}} \tag{31}$$

For the case of a critically damped loop ($\zeta=1$), the error is

$$e_s(t) = 2\pi d_o t\varepsilon^{-\omega_o t} \tag{32}$$

For this case, T_{sm} and E_{sm} are

$$\omega_o T_{sm} = 1 \tag{33}$$

$$E_{sm} = \frac{d_o\varepsilon^{-1}}{f_o} \tag{34}$$

The steady-state error for a step input is zero; i.e.

$$e_s(\infty) = 0 \tag{35}$$

Next assume that the Doppler input undergoes a ramp of input frequency shift. (This would correspond to a constant non-zero acceleration.) The input is given by

$$d(t) = d_1 t \tag{36}$$

where d_1 is measured in $H_z/\text{sec}=H_z^2$. The time domain error $e_r(t)$ for the underdamped loop is given by

$$e_r(t) = \frac{d_1}{2\pi f_o^2}\left[1-\frac{\varepsilon^{-\zeta\omega_o t}}{\sqrt{1-\zeta^2}}\sin(\omega_o\sqrt{1-\zeta^2}t+\cos^{-1}\zeta)\right] \tag{37}$$

The maximum phase error E_{rm} for the ramp input occurs at a time T_{rm}. For the underdamped loop, the relationships are

$$\omega_o T_{rm} = \frac{\pi}{\sqrt{1-\zeta^2}} \tag{38}$$

$$E_{rm} = \frac{d_1}{2\pi f_o^2}\left[1+\varepsilon^{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}}\right] \tag{39}$$

For the case of the critically damped loop and ramp input, the error in phase is

$$e_r(t) = \frac{d_1}{2\pi f_o^2}\left[1-\varepsilon^{-\omega_o t}(1+\omega_o t)\right] \tag{40}$$

In this case the maximum error occurs at $t=\infty$ and is identical with the steady-state error that will be discussed in the next paragraph.

The steady-state phase error for the ramp frequency input is a constant and is given by

$$e_r(\infty) = \frac{d_1}{2\pi f_o^2} \tag{41}$$

The steady-state frequency errors for both the step and ramp frequency inputs are zero in the ideal linear models. In effect the loop gain product in the system satisfies the type 2 criterion of control theory, thus providing zero steady-state errors between output frequency estimates and input Doppler shifts for both step and ramp frequency inputs.

Digital Phase Lock Loop

A discrete phase-lock concept for implementing a frequency tracker with digital techniques will be discussed in this section. The system has been successfully simulated on a digital computer and could be constructed with existing digital circuitry. It was designed around a close comparison between each analog operation and a corresponding digital operation, thus making it convenient for computer simulation of an analog unit. Whether or not this approach is the ideal one for actually designing a direct digital unit for implementation is an open question, as will be discussed later.

The system under consideration is shown in Fig. 2. As is true with any digital unit, the system operates with discrete samples which are obtained by sampling the input Doppler signal at intervals of T seconds apart. The sampling rate must be chosen to be somewhat higher than twice the highest frequency to be considered. In the noise-free case, the input signal is assumed to be of the discrete form

$$x(n) = \sqrt{2}\ \sin[n\omega_C T+\theta_1(n)] \tag{42}$$

Assume that the feedback signal from the VCO v(n) is of the form

$$v(n) = \sqrt{2}\ \cos[n\omega_C T+\theta_2(n)] \tag{43}$$

Each successive set of samples of x(n) and v(n) are multiplied together in a digital arithmetic unit: The output pulse train $e_1(n)$ can be determined as

$$\begin{aligned} e_1(n) = \ & \sin[\theta_1(n)-\theta_2(n)] \\ & + \sin[2n\omega_C T+\theta_1(n)+\theta_2(n)] \end{aligned} \tag{44}$$

For reasons that will be explained more fully later, it was found desirable to employ a digital band-rejection filter to remove the spurious second harmonic component before the signal is applied to the loop filter. It will be seen that the effect of the band-rejection filter on the amplitude of the error is negligible, but there is a small phase shift. However, for the moment, this additional phase term will be neglected, and the error signal e(n) at the input to the loop filter will be assumed to be

$$e(n) = \sin[\theta_1(n)-\theta_2(n)] \tag{45}$$

The digital loop filter is a first-order transfer function designed by means of the bilinear transformation applied to the analog prototype, thus making it behave very closely with the analog unit.

In order to develop a digital filter by the bilinear transformation technique, an analog prototype transfer function is first assumed. In this case it is

$$H_e(p) = \frac{K_o(1+p\tau)}{p} \tag{46}$$

where the complex variable p denotes a prototype reference variable. The bilinear transformation reads

$$p = C_e \frac{(1-z^{-1})}{1+z^{-1}} \tag{47}$$

where C_e is a mapping constant and z is the z-transform variable. The value z^{-1} denotes a one-unit (T) time delay.

Substitution of (47) into (46) yields for the z-transfer function

$$H_e(z) = \frac{A(1-a_1 z^{-1})}{1-z^{-1}} \tag{48}$$

where

$$A = \frac{K_o(1+C_e\tau)}{C_e} \tag{49}$$

$$a_1 = \frac{C_e\tau-1}{C_e\tau+1} \tag{50}$$

Let $\hat{d}(n)$ represent the sampled version of the output Doppler estimate, and let e(n) represent the input to the loop filter. The transfer function of (48) is generated by the algorithm

$$\hat{d}(n) = \hat{d}(n-1) + A[x(n)-a_1x(n-1)] \tag{51}$$

Because of the fact that a_1 is a number very close to unity, as observed from (50), it was discovered that accumulated error could be minimized in the process of (51) by rewriting in the form

$$\hat{d}(n) = \hat{d}(n-1) + A[x(n)-x(n-1)+bx(n-1)] \tag{52}$$

where

$$b = \frac{2}{1+C_e\tau} \tag{53}$$

The digital VCO is a numerical process in which the Doppler estimate $\hat{d}(n)$ is "integrated" by a zero-order summation and used to modulate the argument of a cosine function. The output of the VCO was expressed by equation (43). The angle $\theta_2(n)$ is given by

$$\theta_2(n) = \theta_2(n-1) + 2\pi T\hat{d}(n) \tag{54}$$

The transfer function of the VCO is then given by

$$\frac{\theta_2(z)}{D(z)} = \frac{2\pi T}{1-z^{-1}} \tag{55}$$

Let $p=j\lambda$ represent the imaginary axis of the prototype variable, and let $s=j\omega$ represent the imaginary axis of the final s-plane function. From the bilinear transformation of (47), it can be seen that

$$\lambda = C_e \tan\frac{\omega T}{2} = C_e \tan\frac{\pi f}{f_s} \tag{56}$$

where f_s is the sampling frequency given by

$$f_s = \frac{1}{T} \tag{57}$$

Assume that a particular prototype radian reference frequency λ_r is to correspond to a particular cyclic reference frequency f_r in the actual system. The constant C_e is then

$$C_e = \lambda_r \cot \frac{\pi f_r}{f_s} \tag{58}$$

If the sampling rate is chosen to be sufficiently high compared to the frequency range of main concern (order of ten times or so), the mapping constant is given by approximately

$$C_e \simeq \frac{\lambda_r \; f_s}{\pi \; f_r} \tag{59}$$

If one-to-one correspondence between analog and digital filters is desired ($\lambda_r = 2\pi f_r$), then

$$C_e \simeq 2f_s = 2\left(\frac{f_s}{f_c}\right) f_c = 2\alpha f_c \tag{60}$$

where

$$\alpha = \frac{f_s}{f_c} \tag{61}$$

The quantity α is a convenient parameter giving the ratio of sampling frequency to center frequency of the loop. It can also be interpreted as the number of points per cycle measured at the center frequency. Finally, since real-time operation is not sought in the simulation, f_c can be normalized to unity, yielding

$$C_e = 2\alpha \tag{62}$$

for $f_c = 1$. In subsequent discussions, all frequencies and time constants will often be selected with the assumption of the normalized center frequency of unity.

Notch Filter

In the last section, mention was made of employing a notch filter at the output of the multiplier in order to remove the component at the second-harmonic of the center frequency of the loop. It appears that this procedure is more desirable in a digital unit than in certain classical types of analog multipliers because of the "ideal" nature of the multiplication in the digital multiplier. Furthermore, a relatively low frequency wide-band Doppler frequency tracker would require a loop filter having a bandwidth too wide to adequately suppress the second harmonic term.

Let $H_n(p)$ represent the prototype transfer function of a second-order notch filter. If the notch frequency is normalized to unity, this function is

$$H_n(p) = \frac{p^2+1}{p^2+\frac{p}{Q_n}+1} \qquad (63)$$

where

$$Q_n = \frac{\text{Center Frequency}}{\text{3 dB Notch Bandwidth}} \qquad (64)$$

The transfer function (63) may be mapped to the discrete domain by the transformation

$$p = C_n \frac{(1-z^{-1})}{1+z^{-1}} \qquad (65)$$

The imaginary axes are related by

$$\lambda = C_n \tan \frac{\omega T}{2} = C_n \tan \frac{\pi f}{\alpha f_c} \qquad (66)$$

where α was defined in equation (61). Since the notch is required to be at twice the carrier frequency, $f=2f_c$ and $\lambda=1$, resulting in

$$C_n = \cot \frac{2\pi}{\alpha} \qquad (67)$$

Inserting (65) into (63) and subsequent manipulation yields

$$H_n(z) = \frac{a_{no} + a_{n1}z^{-1} + a_{n2}z^{-2}}{b_{no} + b_{n1}z^{-1} + b_{n2}z^{-2}} \qquad (68)$$

where

$$a_{no} = 1+C_n^2 \tag{69}$$

$$a_{n_1} = 2(1-C_n^2) \tag{70}$$

$$a_{n_2} = 1+C_n^2 \tag{71}$$

$$b_{no} = 1+C_n^2+C_n/Q_n \tag{72}$$

$$b_{n_1} = 2(1-C_n^2) \tag{73}$$

$$b_{n_2} = 1+C_n^2-C_n/Q_n \tag{74}$$

Let $e_1(n)$ represent the error signal entering the notch filter, and let $e(n)$ represent the output. The algorithm for generating $e(n)$ is given by

$$\begin{aligned} e(n) = {} & \frac{1}{b_{no}}[a_{no}e_1(n)+a_{n_1}e_1(n-1)+a_{n_2}e_1(n-2)] \\ & - \frac{1}{b_{no}}[b_{n_1}e(n-1)+b_{n_2}e(n-2)] \end{aligned} \tag{75}$$

Optimization of Notch Bandwidth

The purpose of the notch filter is to remove the high frequency component of the signal at the output of the multiplier as previously discussed. Ideally, the notch filter should have negligible effect on the low-frequency unsmoothed Doppler estimate, since it must also pass through the notch filter. This condition could be met if the undesirable high frequency component were a single frequency located at twice the carrier frequency. In this case the notch could be located at exactly this frequency, and the Q could be made to be as high as practical coefficient accuracy would permit, thus resulting in a very narrow band-rejection notch having virtually no effect away from the notch.

Unfortunately, the hiqh-frequency component does not exist solely at a single frequency as can be seen most easily from the original analog representation in equation (19). Letting $e_h(t)$ represent the high-frequency signal, it is

$$e_h(t) = \sin[2\omega_c t+\theta_1(t)+\theta_2(t)] \tag{76}$$

The instantaneous cyclic frequency $f_i(t)$ of this signal is

$$f_i(t) = 2f_c+f_1(t)+f_2(t) \tag{77}$$

where

$$f_1(t) = \text{instantaneous input frequency}$$
$$= \frac{1}{2\pi}\frac{d\theta_1(t)}{dt} \tag{78}$$

$$f_2(t) = \text{instantaneous feedback frequency}$$
$$= \frac{1}{2\pi}\frac{d\theta_2(t)}{dt} \tag{79}$$

Although there will certainly be a reasonable fraction of the time when f_1 and f_2 are not equal, e.g. during sudden changes of the Doppler shift while the loop is attempting to track the signal; from the point of view of steady-state behavior, the loop will always eventually reach zero frequency error for a sinusoidal input signal of a constant frequency. Considering this important limiting case, assume that the input Doppler is a single frequency f_d in which case

$$f_1 = f_2 = f_d \tag{80}$$

The instantaneous frequency of $e_h(t)$ is now

$$f_i = 2f_c+2f_d \tag{81}$$

Observe that since the Doppler shift may be either positive or negative, the range of the steady-state frequency signal is

$$f_c-2f_d \leq f \leq f_c+2f_d \tag{82}$$

The conclusions of the preceding development may now be stated. The notch cannot be made to be extremely sharp because the frequency of the undesired signal can vary over the range given in (82), and an exceptionally sharp notch would fail to provide adequate attenuation for a signal with any reasonable value of Doppler shift. In fact, as the Doppler shift increases, the width of the notch must be increased (lower Q) in order to accommodate the objective. However, as the width of the notch is increased, the effect on the low-frequency operation of the loop is more pronounced. Thus, there is a tradeoff between the attenuation effectiveness of the filter and the amount of undesired parasitic effect on the loop, which becomes more difficult as the Doppler shift increases.

It is possible to determine an optimum Q for the notch for a given range of Doppler shifts, if certain criteria are specified. An analysis of the steady-state response of the notch filter in the vicinity of the notch shows that for a given f_d, the magnitude transfer ratio (MTR) of the notch is given by

$$\mathrm{MTR}\Big]_{\substack{\text{notch}\\ \text{filter}}} \simeq \frac{2Q_n f_d}{f_c} \tag{83}$$

The magnitude transfer ratio of the loop filter itself in the vicinity of the notch is given by

$$\begin{aligned}\mathrm{MTR}\Big]_{\substack{\text{loop}\\ \text{filter}}} &= \frac{K_o\sqrt{1+\omega^2\tau^2}}{\omega}\\ &= \frac{K_o\sqrt{1+(2\pi)^2\,\tau^2 f^2}}{2\pi(2\pi f)} = \frac{K\sqrt{1+(2\pi)^2\left(\frac{2\zeta}{\omega_o^2}\right)^2(2f_c+2f_d)^2}}{(2\pi)^2(2f_c+2f_d)}\\ &\simeq \frac{K\zeta}{\pi\omega_o} = 2f_o\zeta\end{aligned} \tag{84}$$

The approximation in (84) is valid since the expression under the radical is normally much larger than one. The total magnitude transfer ratio between multiplier output and loop filter output is

$$\text{MTR}]_{\text{total}} = 2Q_n\frac{f_d}{f_c} \times 2f_o\zeta = 4\zeta Q_n\frac{f_o f_d}{f_c} \tag{85}$$

Let K_d represent the ratio of the peak high frequency component to the Doppler shift f_d. This value is

$$K_d = 4\zeta Q_n\frac{f_o}{f_c} \tag{86}$$

From this equation, the value of Q_n may be determined to yield a specified K_d with the other parameters known. It should be pointed out that Q_n cannot be decreased indefinitely, since to do so would cause some effect in the low-frequency loop passband, particularly in regard to additional phase shift for the data. The relationship of (86) was checked in several computer runs and was found to be quite accurate.

Computer Simulation

The digital frequency tracker previously described was simulated on an IBM 1130. Although several modifications on the program were considered, a representative

FORTRAN IV program of the essential loop function is shown in Appendix A. As usual it was necessary to modify the mathematical notation somewhat when constructing this program. Extensive runs were made with different types of inputs and different parameters for the loop.

Although it was obviously impossible to check all of the results with actual phase-locked loop frequency trackers, several representative runs were compared with the responses predicted from the linear model. One would not expect to achieve exact correlation between the simulated system and the linear model, since the simulated model is nonlinear. However, for relatively small frequency deviations, there should be close correspondence. This property was verified since the simulation did show close correlation for fairly small deviations.

Zero-Crossing Discriminator

A particular form of digital phase-locked loop that was considered early in this investigation was one employing zero-crossings detectors as a means for providing the error signal for the loop. The basic open-loop zero-crossing discriminator has been analyzed in the literature (refs. 21, 56).

The closed-loop system considered is shown in Fig. 3. The input zero-crossing detector provides a pulse each time the Doppler signal changes sign. This pulse train is applied to the positive input of an accumulator, which records the number of pulses received. At the same time a feedback pulse train is applied to the negative input of the accumulator, which subtracts from the total. The output of the accumulator at any time is equal to the difference between the number of zero-crossings obtained from the input Doppler signal and the number of zero-crossings of the feedback signal. The digital loop filter is chosen to smooth this error signal and provide an estimate of the Doppler.

A system of this type was actually simulated on the IBM 1130, and several runs were made. The results appeared to be promising, but the design was not finalized. It appeared that some further study was needed to determine the appropriate type of loop filter to employ. The loop filter considered for the classical type of tracker was not adequate. Because of a change in the direction of this investigation, this system was never fully developed. It is felt that this system could indeed warrant further study.

FAST FOURIER TRANSFORM PROCESSING

Introduction

The Fast-Fourier Transform (FFT) is a rapid computational method for evaluating the Fourier coefficients of a finite length record of a discrete signal. This technique appears to have considerable promise for processing signals in which the spectral content is of primary importance. The process may be applied to a normal computer by programming it to perform this function, or a special FFT computer may be built to specifically perform spectral analysis. The increasing advancement in micro-integrated digital circuits has resulted in the possibility of implementing certain special FFT processors for on-board spacecraft signal analysis within the normal size and weight constraints of such missions.

The purpose of this investigation was to assess the potential of the FFT technique for processing Doppler return signals in a planetary radar altimeter. Measurement of certain properties of the spectrum is required in order to estimate the velocity and, in some cases, the altitude of the spacecraft, depending on the exact type of radar system employed (refs. 2, 4, 5, 6, 22, 23, 24, 26, 31, 51, 61, 65, 70, 74).

Most Doppler altimeter systems in use today employ an analog phase-locked loop frequency tracker of the type considered in an earlier part of this report. Along with the concept of implementing the phase-locked loop with digital techniques investigated earlier, the use of the FFT concept as a different type of digital processing mechanism was investigated.

Discrete Fourier Transform

The Fast Fourier Transform (FFT) is a high-speed algorithm for evaluating the Discrete Fourier Transform (DFT). In this section the DFT will be developed as an approximation to the finite Fourier series. This approach is slightly different to most of the developments appearing in the literature, but it is felt that it offers certain advantages in achieving understanding of the process.

Consider a signal x(t) defined over the finite range $o \leq t \leq T_p$. This interval may represent one or more cycles of a periodic signal, or it may represent simply a segment of a random process having infinite limits. It is well-known in the theory of boundary value problems that such a function may be expanded in a Fourier series as long as it is reasonably "well-behaved." The evaluation of the Fourier series consists in determining a set of coefficients defining the amplitudes of the spectral components of the frequency representation of x(t). Let F represent the fundamental non-zero frequency contained in the spectrum. The quantities T_p and F are related by

$$F = \frac{1}{T_p} \tag{87}$$

Furthermore, the frequencies in the spectrum are spaced a distance F between components.

Let X(mF) represent the complex values of the spectral coefficients of x(t). The finite exponential Fourier serie pair can then be expressed as

$$X(mF) = \frac{1}{T_p} \int_0^{T_p} x(t)\varepsilon^{-j2\pi mFt}\,dt \tag{88}$$

$$x(t) = \sum_{-\infty}^{\infty} X(mF)\,\varepsilon^{j2\pi mFt} \tag{89}$$

An important property of the summation (or inverse transform) given by (89) is that the resulting time function is periodic with period T_p regardless of whether or not the original function is periodic. This can be seen by first noting that

$$\varepsilon^{j2\pi k} = 1 \text{ for k an integer} \tag{90}$$

Utilizing (90) it is readily shown that in (89)

$$x(t+T_p) = x(t) \tag{91}$$

The first step in achieving a complete digital implementation of (88) and (89) is to define a suitable numerical approximation to the integral of (88). Although many numerical procedures are theoretically possible, the technique that has received almost all the recent attention has been a zero-order approximation to the integration. However, instead of actually defining it as a numerical approximation, the operation under consideration has been defined as the Discrete Fourier Transform (DFT).

Assume that a finite length record of a signal x(t) is to be transformed. For convenience, let t=0 be the starting point, and assume that the record consists of N samples of x(t) defined at equally-spaced intervals separated by T seconds apart. Thus the input signal can be represented as x(nT), $0<n<N-1$. Let T_p represent the entire length of the record.

$$T_p = NT \tag{92}$$

The differential dt can be approximated by the sampling interval time T. Thus the integral of (88) may be approximated by the summation

$$X(mF) = \frac{1}{N} \sum_{0}^{N-1} x(nT) \varepsilon^{-j2\pi mnFT} \tag{93}$$

The summation expressed by (93) can be shown to yield a periodic signal in frequency with "period" F_p where

$$F_p = \frac{1}{T} \tag{94}$$

This can be seen by observing that

$$X(mF+F_p) = X(mF) \tag{95}$$

Observe in (93) that

$$FT = \frac{1}{N} \tag{96}$$

Substitution of (96) in (93) yields

$$X(mF) = \frac{1}{N} \sum_{0}^{N-1} x(nT)\varepsilon^{-j\frac{2\pi mn}{N}} \tag{97}$$

In view of the deductions of the past few paragraphs, it can be stated that the DFT of a finite length discrete time signal is a periodic spectrum defined only at discrete frequencies. The spacing between spectral components is the reciprocal of the record length as given by (87), and the "period" of the spectrum is the reciprocal of the spacing between time samples as shown by (94). The latter relationship is related to the minimum sampling-rate concept inherent in sampled signal theory. The frequency F_p is equivalent to the sampling rate. The maximum unambiguous frequency is called the Nyquist frequency F_n and is given by

$$F_n = \frac{F_p}{2} = \frac{1}{2T} \tag{98}$$

It can be shown that the spectrum between F_n and F_p is ambiguous and is dependent on the spectrum between dc and F_n.

As a matter of convenience the symbols F and T in the arguments of X(mF) and x(nT) will be omitted and understood. The complex value W will be defined as

$$W = \varepsilon^{-j\frac{2\pi}{N}} \tag{99}$$

The Discrete Fourier Transform may now be stated as

$$X(m) = \frac{1}{N}\sum_{0}^{N-1} x(n)W^{mn} \tag{100}$$

It should now be observed that the inverse transform given by (89) appears to require summation over both positive and negative frequencies; whereas, the transform has been computed only for positive frequencies. However, it can be shown that summation of the ambiguous portion of the spectrum from N/2 to N is mathematically equivalent to summing over the negative frequency range. The inverse time function is then given by

$$x(n) = \sum_{0}^{N-1} X(m)W^{-nm} \tag{101}$$

In many applications, only the transformation from time to frequency is desired. It is immaterial whether the 1/N factor is placed with the time-to-frequency transformation of (100) or the frequency-to-time transformation of (101). For convenience, the quantities will be redefined so that this factor is grouped with the latter function. Thus, the final definition of the DFT pair will be stated as

$$X(m) = \sum_{0}^{N-1} x(n)W^{mn} \tag{102}$$

$$x(n) = \frac{1}{N} \sum_{0}^{N-1} X(m)W^{-nm} \tag{103}$$

The remainder of the report will concentrate on computation and interpretation of (102), since for the Doppler application under consideration, this is the function of primary concern.

Fast Fourier Transform

The Fast Fourier Transform (FFT) is a high-speed algorithm for evaluating the Discrete Fourier Transform (DFT) or its inverse. There is an extensive volume on recent literature appearing on the subject. Many of the sources encountered in this investigation are listed at the end of this report (refs. 1, 7, 8, 9, 11, 12, 14, 15, 16, 19, 25, 27, 29, 52, 57, 62, 63, 64, 66, 68). Most of the current interest on the subject is based on the computational algorithm developed by Cooley and Tukey (ref. 19).

A brief discussion of the computational technique will be discussed. Observing equation (102), it can be seen that the direct computation of the spectrum from m=0 to m=N can be represented by the matrix equation

$$\begin{bmatrix} X(0) \\ X(1) \\ \vdots \\ X(N-1) \end{bmatrix} = \begin{bmatrix} W^0 W^0 & \ldots W^0 \\ W^0 W^1 & \ldots W^{N-1} \\ \vdots \; \vdots & \vdots \\ W^0 W^{N-1} & \ldots W^{(N-1)^2} \end{bmatrix} \begin{bmatrix} X(0) \\ X(1) \\ \vdots \\ X(N-1) \end{bmatrix} \tag{104}$$

Let $\overline{X}$ represent the vector defining the N spectral components, $\overline{x}$ the vector defining the N time samples, and [W] the N x N matrix. The array of (104) can be expressed as

$$\overline{X} = [W]\overline{x} \tag{105}$$

If a direct application of (104) is employed to compute the spectrum, it is necessary only to compute the spectrum from m=0 to m=N/2-1. This is true since the spectrum in the range from m=N/2 to N-1 is not arbitrary and can be deduced from the spectrum in the former range. However, the FFT technique to be considered shortly requires that the fully expanded form of (104) be used in expressing the algorithm.

With the preceding points in mind, it can be seen that a total of $N^2/2$ complex multiplication and $N^2/2$ complex additions are required to compute the spectrum directly from (104) in the range of interest. For the sake of discussion the value N^2 will be assumed as a reference when specifying the approximate number of complex arithmetic computations (including both multiplication and addition) for computing the spectrum of an N point signal <u>without</u> the use of the FFT algorithm.

The approach of Cooley and Tukey can be thought of as equivalent to factoring the matrix [W] into a number of separate matrices, each having a large number of zero entries (sparse matrices), and chosen in a manner to reduce the number of computations by a considerable amount. The most efficient factorization results when N is selected as an integer multiple of 2. From here on, this restriction will be made. Let

$$N = 2^L \tag{106}$$

$$\text{or } L = \log_2 N$$

With this assumption the number of matrix factors is L.

Thus, [W] can be represented as

$$[W] = [W_1][W_2]\ldots[W_L] \tag{107}$$

Each of the individual matrices is an N x N matrix having only two non-zero elements on each row. One of the elements is unity, and the other is W raised to some power in the range between 0 and N.

There is one minor problem in the natural operation of the basic algorithm. If the input vector $\overline{x}$ contains its elements in their natural order, the output spectral components are "scrambled" or shuffled out of natural order upon computation. Let $\overline{X}_s$ represent this scrambled spectrum vector. The operation can be written as

$$\overline{X}_s = [W_1][W_2]\ldots[W_L]\overline{x} \tag{108}$$

The order in which (108) is performed is as follows: First, the vector $\overline{x}$ is multiplied by $[W_L]$ to yield a new vector. Next, this vector is multiplied by $[W_{L-1}]$ to yield another vector. This process is repeated until the L_{th} matrix multiplication, in which the scrambled vector $\overline{X}_s$ is produced. As previously pointed out, each row of a given matrix contains the constant unity and W raised to an integral power, while all other entries on that row are zero. The values of W^k may be stored in memory, so there is not necessarily any computation involved in obtaining these quantities. (This point will be discussed more fully later.)

The multiplication of a particular row of a matrix by a given vector involves only one complex multiplication and one complex addition. Since there are N rows, N complex multiplications and N complex additions are required to transform any vector to the next vector. However, there are $L=\log_2 N$ such operations involved. Thus, there are a total of $N \log_2 N$ complex multiplications and $N \log_2 N$ complex additions. By taking advantage of certain symmetries, the number of computations may be reduced by one-half or more if the basic algorithm is further modified. For the sake of discussion, the value $N \log_2 N$ will be assumed as a reference for specifying the approximate number of complex arithmetic computations (including both multiplication and addition) for computing the spectrum of an N point signal with the FFT algorithm.

A comparison of the direct definition of the DFT with the FFT for a signal with 1024 points shows that the number of computations required with the DFT would be of the order of 10^6, while for the FFT the number of computations would be of the order of 10^4. The computation time in this case would be about 1% for the FFT as compared with the straight DFT definition.

The vector $\overline{X}_s$ as computed from (108) must be "unscrambled" in order to yield the spectral components in their natural order. This can be achieved by an algorithm in which the storage addresses of all the components are listed in binary. The component of the unscrambled vector $\overline{X}$ at a particular storage location address can be found by going to the address corresponding to the bil-reversed value of the reference location and reading the value there.

The algorithm given by (108) is only one of several different techniques. An alternate approach having essentially identical computation time is to first sort the input to yield a scrambled time signal $\overline{x}_s$. In this case the corresponding matrix factorization yields an output in natural order according to the equation

$$\overline{X} = [U_1][U_2]\ldots[U_L]\overline{x}_s \tag{109}$$

Although (108) and (109) both require scrambling they have the advantage of "computation in place." This term refers to the fact that as the progressive sequence of operations to yield the components of an intermediate vector take place, a given new component may be stored in the same location as the previous corresponding component. On the other hand it is possible to organize the computations so that both a natural input and a natural output order are utilized. The only disadvantage to this approach is that "computations in place" are not possible. Thus, the amount of internal storage required is increased, but the logic is still straightforward.

Previously, it was shown that the complex values W^k are required in the computation. Using the definition of W given in (99), the value of W^k is given by

$$W^k = \varepsilon^{-j\frac{2\pi k}{N}} = \cos\frac{2\pi k}{N} - j\sin\frac{2\pi k}{N} \tag{110}$$

Note that W is periodic with period N. Furthermore, the values of W^k in any range of k could be deduced from the values of either sine or cosine in the first quadrant by means of trigonometric identities. Thus, in theory it would be possible to deduce all values of W^k required from the N/4 sine or cosine values in the first quadrant (or even less if desired). However, in practice it may be desirable to employ more values than this in order to avoid more complex logic in calling the appropriate values. The values of W^k required may be stored in the memory of the FFT processor for recall when needed. If memory is extremely limited, but processing speed is not critical, it may even be desirable to compute some of the trigonometric values as they are required.

The exact organization of the logic and memory is a problem that must be considered in the light of the overall system requirements and hardware limitations in terms of size, weight, and the present state of the art. Obviously, if the processing for a given system can be performed on the ground, it is not necessary to be so particular about optimizing the logical design of the FFT processor. In fact, for ground processing a general purpose computer may be readily programmed to provide FFT data analysis.

Practical Considerations

Successful employment of the FFT requires an understanding of some of the properties and limitations of the technique. Unless some of the important fine points of this approach are understood, there is a strong possibility that erroneous interpretations and invalid answers can result.

First of all, it is impossible to obtain a true Fourier representation in the strictest sense unless the nature of the time signal is known for all time. Obviously, this is completely impossible for any meaningful signal in which measurement is desired, and the best that can be done is to determine a Fourier representation that is valid over a reasonable segment of time. This suggests the concept that repeated Fourier analyses can be made on successive records of the signal with the intent that each analysis should represent the spectrum of that segment. The longer that a given record is made, the finer is the resolution between successive frequencies as suggested by equation (87). However, as the record length is increased, either the number of sample points must be increased for equal spacing between time samples, or the spacing between time sample must be increased if the number of points is fixed. If the latter possibility is chosen, the highest frequency at which the spectrum can be calculated is reduced as the record length is increased as can be seen by equation (98). Thus, there is a tradeoff between the resolution between successive frequencies and the maximum frequency at which the spectrum is desired.

Actually, if it is expected that the spectrum is going to change quite rapidly, it is desirable to keep the record length relatively short if the resolution requirements can be met. Otherwise sudden changes in the information being sought will go undetected, which could be disastrous in some situations, e.g. Doppler radar velocity measurements with rapid acceleration. In general, most of the classical established concepts of spectral analysis hold for the FFT method as long as the appropriate special conditions are considered. A classic source of information on spectral analysis is the work by Bıackman and Tukey (ref. 10).

The determination of the optimum record length, sampling rate, etc., is a very complex problem which must be carefully studied in view of the nature of the spectrum and the desired accuracy requirements. As a starting point the relationship between the frequency resolution F, the Nyquist frequency F_n, and the number of samples N should be observed.

$$FN = 2F_n \tag{111}$$

From this equation, any one of the three quantities may be determined if the other two are known. Once the three quantities are specified, the record length T_p and sample time T are determined from the equations

$$T_p = \frac{1}{F} \tag{112}$$

$$T = \frac{1}{2F_n} \tag{113}$$

The discussion of the preceding paragraph seems deceptively simple. However, there are subtle difficulties that may arise that could obscure the validity of the measurement. In order to best explain the phenomena involved, refer to Fig. 4. The time signal is assumed to be a pure sinusoid running from $-\infty$ to $+\infty$ as suggested in (a). The ideal spectrum is a pair of lines as shown in (b). The waveform that is actually transformed is the gated sinusoid shown in (c). The spectrum is the convolution of the lines of (b) and the spectrum of a pulse-type waveform, resulting in the modified spectrum shown in (d). Furthermore, since the time signal is sampled as shown in (e), the spectrum must be periodic as shown in (f). Finally, since the representation is in fact a periodic time domain representation with N terms or a period T_p in the time domain as shown in (g), the spectrum is defined only at discrete frequencies as shown in (h). Although the distortions arising with the sinusoid are perhaps exaggerated as compared with many waveforms, the fact remains that care must be taken in processing a signal with this approach.

Bergland (ref. 8) considered several difficulties that may arise from incorrect use and misunderstanding of the FFT. The two most significant difficulties pertinent to this investigation are aliasing and leakage.

The aliasing effect results when the sampling rate is too low. Considering Fig. 4f, if the sampling rate is less than twice the highest frequency in the spectrum, some portion of the translated spectrum will overlap the original spectrum, thus making it impossible to recover or measure the actual spectrum. Before attempting to use the FFT concept, it is necessary to carefully study the qualitative characteristics of the given signal to make sure that the sampling rate is sufficiently high. It is often desirable to pass the signal through a bandlimiting filter before transforming it in order to minimize aliasing effects.

The leakage effect results from the fact that the signal in the time domain can be looked at only for a finite time. The term "window" is used to denote the short duration pulse-type signal which can be thought of as turning on and off the observed record. Since the resulting spectrum is the convolution of the desired spectrum and the spectrum of the window function, there is a subsequent spreading or "leakage" of the spectrum as can be observed in Fig. 4d. Although the square-pulse is the simplest type of window, it can result in significant leakage if the record length is quite short. The leakage problem can be reduced by choosing an optimum window function in the time domain so that the sidelobe energy is minimized. This problem was considered in detail by Blackman and Tukey (ref. 10) and more recently by others in conjunction with the FFT method.

FFT Simulation Programs

Two digital computer programs have been written for computing the FFT. These may be thought of as simulating the approximate logic organization in typical FFT signal processors, although no attempt was made to "optimize" this organization. The programs were written in FORTRAN IV for use on the IBM 1130 in the School of Engineering at Old Dominion University. Both programs require that the input signal have a number of points that is an integer power of two. However, the particular power of two is arbitrary and can be selected in a given case. Unfortunately, storage limitations of the 1130 computer resulted in a maximum array size of 256 for the data presented in this report. It is felt that by very careful attention to storage and computation, this value could probably be doubled on the given machine. For the purpose of this investigation, it was not deemed necessary to concentrate any more effort in this direction since the study of spectral analysis of a 256 point array might lead to some interesting data regarding the effectiveness of analysis with a relatively small array.

The first program was designed around an "in-place" algorithm with the input arranged in natural order. This means that the output data must be "unscrambled" by finding the bit-reversed addresses of the output components.

Apparently, this is straightforward with logic circuits operating in binary. However, it was necessary to develop a subroutine to perform this operation in FORTRAN. The subroutine is entitled INREV(K,NLOG). The trigonometric functions were computed and stored at the beginning of the program.

The complete first program, along with the Doppler subroutine, the bit-reversal subroutine, and various plotting subroutines, is shown in Appendix B. Note that a considerable amount of this program is devoted to various scaling and plotting instructions to provide suitable output displays. The FFT computational algorithm occurs between the statement IN-PLACE FFT ALGORITHM REQUIRING BIT-REVERSAL and the statement END OF IN-PLACE FFT ALGORITHM.

The second program was designed around an algorithm permitting natural order for both input and output data, thus eliminating the need for either "scrambling" or "unscrambling" of the data. However, it was necessary to increase the amount of internal storaqe since "computation in place" is not possible with this algorithm. In order to keep the storage for this program to be comparable with that of the first program, it was decided to compute the trigonometric values as they are needed rather than store them. By paying careful attention to the logic, it can be determined that there are rather long segments of the program in which the same trigonometric functions are repeatedly used. Consequently, the actual computation time required to determine the trigonometric functions as they are needed is approximately twice the time required to compute them once and store them.

The FFT algorithm portion of the second program is shown in Appendix C. The remainder of the program is essentially the same as the program of Appendix B with the exceptions that the subroutine INREV(K,NLOG) is omitted and the DIMENSION card will replace WR and WI with XR and XI. The instructions of Appendix C replace the FFT algorithm over the limits described in a preceding paragraph. Since both programs produce identical results, it will not be necessary in subsequent discussions to refer to either program specifically.

Computation of the FFT yields the complex spectrum defined by (104). A more pertinent result for application in Doppler measurement is the power spectrum denoted by S(m). The power spectrum definition used is given by

$$S(m) = |X(m)|^2 \tag{114}$$

The operation of (114) is easily implemented by taking the sum of the squares of the real and imaginary parts of X(m).

Statistical Evaluation of the Spectrum

A discussion of some of the important properties of Doppler spectra was presented in and earlier portion of this report. From the discussion there, it is evident that for any realizable system, there is a certain amount of statistical averaging inherent in the measurement of any Doppler spectrum. Traditional frequency trackers accomplish this goal, as best as can be done, by locking in on some pertinent characteristic of the spectrum and tracking it.

With the application of the FFT, there is a complete "picture" of the spectrum available, and subsequent processing of the data can center around this fact. Although there is no unique strategy that can be proposed at this point, one can make a rather strong argument to support the assumption that some sort of statistical evaluation of the spectrum would be most appropriate. Unless some particular correlation between the shape of the return spectrum and the surface-to-spacecraft geometry can be programmed, the statistical mean of the spectrum seems to be the most pertinent characteristic to identify. Thus, all the components of the return signal can be weighted, and the adverse effect of some single strong erroneous component can be minimized.

The assumption will be made that the power spectrum

S(m) satisfies, in some sense, a form of statistical representation of the return signal as far as frequency is concerned. This assumption certainly appears to be compatible with that of Berger (ref. 4). It is first necessary to normalize S(m) so that it qualifies as a legitimate probablity density function. The normalizing quantity S_T is computed from the summation

$$S_T = \sum_{0}^{N/2-1} S(m) \tag{115}$$

A normalized power spectrum is then computed from

$$S_o(m) = \frac{S(m)}{S_T} \tag{116}$$

Let $\overline{m}$ represent the location of the mean component of the spectrum. From basic statistical theory it is

$$\overline{m} = \sum_{0}^{N/2-1} mS_o(m) \tag{117}$$

The mean frequency in the spectrum $\overline{f}$ is then given by

$$\overline{f} = \overline{m}F \tag{118}$$

Another quantity of interest might be the standard integer deviation σ_m. It is determined from the equation

$$\sigma_m{}^2 = \sum_{0}^{N/2-1} (m-\overline{m})^2 S_o(m) \tag{119}$$

The frequency standard deviation σ_f is then given by

$$\sigma_f = \sigma_m F \qquad (120)$$

Possible FFT Processors

Two schemes will be proposed for utilizing the FFT processor in the measurement of Doppler return signals. The first scheme is an open-loop measurement whose essential features are shown in Fig. 5. It is assumed that the RF signal is first translated through one or more appropriate IF channels so that the signal appearing at the input to the processor has a frequency range sufficiently low to permit real-time digital processing.

A record of the signal is first obtained by sampling and storing until the required array size is obtained. From this stage the data is fed to the FFT processor. If storage space is at a premium, it may be necessary to alternately perform recording and computation. During a recording period the output display would simply reflect the results of the previous computation period, and no new results could be permitted during this period. During the computation period, the array obtained during the recording period would be processed to yield updated values. With this approach, careful attention would have to be paid to the overall system behavior expected in order to avoid possible loss of information due to the "blind" periods and lag. The appropriate estimates are obtained from a statistical computation of the spectrum as previously discussed.

If more storage space is permitted, it is possible to simultaneously record and compute. In this manner, all of the potentially available samples are utilized; whereas, if separate record and compute periods are used, half of the available samples will never be used. However, with either approach, there is always a delay involved before a given array can be processed. This is true regardless of the type of system employed, since any bandlimited measurement cannot be made instantaneously.

The second scheme that will be proposed is a closed-loop measurement whose basic form is shown in Fig. 6. This figure should be interpreted merely as an explanation guide, since an actual system employing such an approach might require quadrature channels or more complex interaction with the remainder of the system. This system has the advantage that it is always performing a spectrum analysis of a low-pass signal, since the desirable part of the error spectrum at the output of the multiplier is low-pass in nature. (It may be necessary to employ a band-rejection filter here as was considered in an earlier section in conjunction with the digital phase-locked loop.) Consequently, a more efficient FFT algorithm can probably be implemented.

The estimates at the output of the statistical analyzer are used to bias a VCO which mixes with the input samples. Thus, the superior closed-loop operation of traditional analog frequency trackers could conceivably be applied to the FFT processor.

Results of Simulation

The FFT program previously described was used to study certain aspects of the open-loop measurement scheme suggested in the previous section. A basic statistical analysis routine was used in the FFT program to determine the mean and standard deviation. The time and frequency functions for certain cases were plotted with the IBM 1627 Plotter as shown in Figures 7 through 22. For plotting purposes, the power spectrum in each case was normalized so that the maximum value is unity. The percentage bandwidth (BW) as defined in equation (14) was printed on the time plot, and the computed mean value was also printed on the frequency plot.

It will be observed that on some of the plots, values for BW and MEAN have been typed over the values originally printed. The reason for this is that the first instructions to the plotter were made to print the results in fixed-point arithmetic. Due to roundoff in printing these quantities on the time plots and the subsequent desire to print more accurate results on the frequency plots, it was later decided to take values from the printouts and type more accurate values on the plots. In case there is a misunderstanding of these points, the actual data plotted is correct; only the original labels in some cases were misleading due to roundoff not taken into consideration in programming only the lettering to the plotter. This discrepancy was amended during the plotting so that some of the curves were labeled by the plotter as desired.

In view of the fact that the simulation was not performed in any real-time sense, but simply with a set of relative parameters that could represent any appropriate scale, it seemed desirable to employ a normalized scale that was not associated with any particular time or frequency units. The terms Time Integer and Frequency Integer seemed to be most logical. A given unit on the scale represents a different sample point. The data taken was based on N=256, and, thus, the time integer scale represents the 256 points obtained in a given record. To enhance the presentation, the plotter was instructed to extrapolate lines between successive points.

On the frequency integer scale, a given unit corresponds to a particular multiple of the fundamental frequency F. Since the spectrum is ambiguous above N/2, only 128 samples are actually shown in each plot. If any given values for the time scale were specified, it would be a simple matter to compute the appropriate frequency values. As long as the normalized scales are employed, the correspondence between a given period and a given frequency may be determined in integer form. Let N_t=number of time sample points corresponding to some reference period, and let N_f=number of frequency integer location corresponding to the time integer period. It can be shown that, in general, N_f is given by

$$N_f = \frac{N}{N_t} \tag{121}$$

For the 256 point system, this is

$$N_f = \frac{256}{N_t} \tag{122}$$

The results of assuming pure sinusoidal returns are shown in Figures 7 through 12. (Notice that higher frequency time plots do not always look like sinusoids due to the type of extrapolation employed on the plotter.) Letting N_t=period integer as previously discussed, these curves represent the following values for N_t: 16, 8, 4, 20, 10, 5. The corresponding expected values for N_f are as follows: 16, 32, 64, 12.8, 25.6, 51.2. Observe that the expected values for the first three cases occur exactly at integer frequency values. Also, since a square window function was employed, the zero crossings of the spectrum of the window are at integer displacements from the main lobe. The resulting spectra thus appear to be single line spectra as observed in Figures 7, 8, and 9. Looking at it from another point of view, these three sinusoids in the time domains each have been chosen to have an integer number of cycles. Whenever this situation exists, the spectrum will appear to be a single line, even though there is actually some leakage as previously noted.

Neither of the three sinusoids of Figures 10, 11, and 12 have an integer number of cycles in the time domain interval, and their corresponding frequency values do not occur at the proper integer values as previously calculated. Thus, the leakage effect can be readily observed from the figures. This leakage could be reduced by choosing an appropriate window function. The effect of the leakage on the mean frequency measurements is to introduce slight errors as can be observed from the data.

Typical signals obtained from the Doppler simulation programs and their corresponding spectra are shown in Figures 13 through 22. The signals of Figures 13 through 17 were generated with 8 points per cycle measured at the center of the Doppler spectra. This is equivalent to sampling at 4 times the Nyquist rate at band-center. The "correct" mean frequency integer for this case is 32. The actual computed means vary from 35.52 at 20% bandwidth to

32.41 at 1% bandwidth. The higher trend in these values is expected due to the skewed nature of the band-pass filter used to generate the Doppler. Thus, it is probably not strictly correct to refer to the mean frequency integer as being 32. Rather, the program is apparently computing a more correct mean frequency integer. The value of 32 simply corresponds to the so-called center frequency of the resonant curve.

The signals of Figures 18 through 22 were generated with 4 points per cycle measured at the center of the spectra. This is equivalent to sampling at twice the Nyquist rate at band-center. In this case, the center frequency integer is 64. The actual computed means vary from 64.18 at 20% bandwidth to 64.39 at 2% bandwidth. Some of the trends here are in the opposite sense from what was originally expected. In fact all of the runs yield very close correspondence between predicted and computed values.

The reason for this phenomena is due to the warping behavior of the bilinear transformation at half the Nyquist frequency. The tangent curve, providing a relationship between analog and digital frequency curves, is sufficiently curved in this frequency range to almost cancel the effects of the original skewed spectra. Thus, the actual spectra are very close to being arithmetically symmetrical, and the additional area at larger bandwidths tends to improve the symmetry in some cases.

SUMMARY AND SUGGESTED STUDIES

Summary

Investigations have been made concerning both the digital simulation of Doppler analog altimeter measurement systems and the possible digital implementation of on-board systems. Simulated Doppler spectra were generated on the computer and used in subsequent measurement studies. The simulation and possible implementation of digital phase-locked loop frequency trackers was first investigated. This was followed by a study of the possible use of the Fast Fourier Transform method for real-time altimeter signal processing. Various conclusions and comments pertinent to these individual investigations have been made at appropriate places in this report.

As is true with many investigations, there are often more new questions raised than there are old questions answered. In the remainder of this report, an attempt will be made to pose certain questions that were generated, but were not pursued in depth due to the limited scope of the overall investigation. It is felt that each topic is probably worthy of further study.

Further Doppler Simulation

All the programs used in the Doppler simulation were based on the second-order analog filter function described in the report. It is suggested that more study be made to develop programs that would simulate a more general situation than was possible with the functions used.

Early in this study, some special band-pass transfer functions having approximate Gaussian frequency response were considered for use in representing certain types of spectra. These functions were previously developed by the principal investigator in conjunction with an industrial investigation. Due to lack of time, these functions were never applied to the present study. An entirely different approach to the generation of Gaussian shaped spectra was developed by Matthews (ref. 45).

The development of a more general Doppler simulation program should be aimed at establishing a better correlation of the spectral components with the geometric relationship of the spacecraft to the planetary surface. More will be said about this point later.

Digital Frequency Tracker

Initial investigations of the digital frequency tracker containing the zero-crossing detector showed definite promise, but it was not pursued in depth. The largest single difficulty was due to the abruptness of the discrete error signal coupled with the fact that only a first-order loop filter was employed. Further studies could be made relative to determining a higher-order loop filter to optimally process the error signal. It is also quite possible that some of the newer estimation techniques, such as Kalman filtering, might have potential application to this problem.

FFT Processing

The FFT method has been initially explored for potential application in Doppler systems, but there is much yet to be answered before a practical implementation can be achieved. There are questions regarding such things as the minimum size of an array needed, the optimum record length for a given type of measurement and the optimum sampling rate. The performance of the system in the presence of background noise should be studied. A study of the particular statistical method described in the report, and possible alternate methods could be made. Comparison of open-loop and closed-loop FFT methods would be desirable.

Before any specific FFT altimeter system is implemented, it is extremely desirable that an overall comparison be made between the frequency tracker concept and the FFT concept. This study should take into consideration such factors as the relative accuracy, speed of response, behavior under worst-case conditions, cost, and amount of hardware required. Although there are certain clear advantages to the FFT concept in terms of the total information on the spectrum provided, there may be some difficulties that are not clear at this point.

Walsh Transform Processing

There has been increasing interest recently in the application of Walsh Functions in communications signal analysis (ref. 39). The Walsh Functions are a set of orthogonal functions composed only of square-wave segments. They are particularly convenient for digital implementation since they may be thought of as alternate segments of 1's and 0's in a binary sense. A given function possessing "reasonable" behavior can be expanded in a Walsh series in much the same way that a function is expanded in a Fourier series. It appears that a Walsh expansion can be readily implemented with a special-purpose digital processor.

In order for the Walsh functions to show promise in Doppler radar altimeter signal processing, it is necessary that certain suitable properties of the Walsh spectrum be directly related to the appropriate properties of an equivalent Fourier spectrum in which the Doppler information is imbedded. There are definite computational relationships between a Walsh spectrum and a Fourier spectrum, but it is not clear whether or not the additional effort would reduce the overall effort or not. It is definitely felt that such an investigation is needed.

Correlation of Surface Geometry with Spectrum

One of the most significant sources of error in classical Doppler measurements is the uncertainty of the velocity components in the case of a broad spectrum. On the other hand, when the FFT method is employed, a total picture of the spectrum is obtained. It would seem that the most optimum approach for an FFT system would be to use this total information as much as possible.

Assume that an ensemble of spacecraft-to-surface geometrical configurations expected in a given case is available. If a suitable backscatter model of the planet is employed, certain approximate spectral return shapes can be predicted. Certain key properties of the various spectra can be stored in memory. When a given actual spectrum is measured it can then be related to the "standard" spectra available. By finding the one most closely approximating the measured spectrum, certain additional information regarding the measurement could be deduced. A more accurate measurement might be possible with this approach. A study of this concept should be coupled with the study seeking more general simulation programs for Doppler signals previously discussed.

REFERENCES

1. Andrews, H.: A High-Speed Algorithm for the Computer Generation of Fourier Transforms. IEEE Trans. on Computers, vol. C-17, April 1968, pp. 373-375.

2. Barton, D. K.: *Radar System Analysis*, Prentice-Hall Inc., 1964.

3. Bendat, J. S.: *Principles and Applications of Random Noise Theory*. John Wiley and Sons, Inc., 1958.

4. Berger, F. B.: The Nature of Doppler Velocity Measurement. IRE Trans. on Aeronautical and Navigational Electronics, vol. ANE-4, Sept. 1957, pp. 103-112.

5. Berger, F. B.: The Design of Airborne Doppler Velocity Measuring Systems. IRE Trans. on Aeronautical and Navigational Electronics, vol. ANE-4, Dec. 1957, pp. 157-175.

6. Berger, F. B.: Application of Doppler Techniques to Space Navigation. Navigation, vol. 6, no. 7, Autumn 1959, pp. 460-464.

7. Bergland, G. D.; and Hale, H. W.: Digital Real-Time Spectral Analysis, IEEE Trans. on Electronic Computers, vol. C-16, April 1967, pp. 180-185.

8. Bergland, G. D.: A Guided Tour of the Fast Fourier Transform. IEEE Spectrum, July 1969, pp. 41-52.

9. Bice, P. K.: Speed up the Fast Fourier Transform. Electronic Design, vol. 18, no. 9, April 26, 1970, pp. 66-69.

10. Blackman, R. B.; and Tukey, J. W.: The Measurement of Power Spectra. Dover Publications, Inc., 1959, reprinted from Bell System Technical Journal, vol. 37, Jan. and March 1958.

11. Brigham, E. P.; and Morrow, R. E.: The Fast Fourier Transform. IEEE Spectrum, vol. 4, no. 12, Dec. 1967, pp. 63-70.

12. Bruce, J. D.: Discrete Fourier Transforms, Linear Filters, and Spectrum Weighting. IEEE Trans. on Audio and Electroacoustics, vol. AU-16, no. 4, Dec. 1968, pp. 495-499.

13. Carney, R.: Design of a Digital Notch Filter with Tracking Requirements. IEEE Trans. on Space Electronics and Telemetry, Dec. 1963, pp. 109-114.

14. Chadwick, H. D.: Time Synchronization in an MFSK Receiver. JPL Space Programs Summary 37-48, vol. III, pp. 252-263.

15. Chadwick, H. D.: Frequency Acquisition in an MFSK Receiver. JPL Space Programs Summary 37-52, vol. III, pp. 239-247.

16. Chadwick, H. D.: Frequency Tracking in an MFSK Receiver. JPL Space Programs Summary 37-57, vol. III, pp. 47-54.

17. Clapp, W. A.: Digital Filter Building Blocks for LSI Technologies. IEEE International Convention Record, March 1970, pp. 182-183.

18. Cooke, C. H.: An Enlarged Class of Fixed Weight Recursive Polynomial Filters. BTL Technical Memorandum no. 68-6414-10, Sept. 23,1968.

19. Cooley, J. W.; and Tukey, J. W.: An Algorithm for the Machine Calculation of Complex Fourier Series. Mathematics of Computation, vol. 19, no. 90, April 1965, pp. 297-301.

20. Davis, A. C.: Digital Generation of Low-Frequency Sine Waves. IEEE Trans. on Instrumentation and Measurement, vol. IM-18, no. 2, June 1969, pp. 97-105.

21. Ehrman, L.: Analysis of a Zero-Crossing Frequency Discriminator with Random Inputs. IEEE Trans. on Aerospace and Navigational Electronics, June 1965, pp. 113-119.

22. Fried, W. R.: Principles and Performance Analysis of Doppler Navigation Systems. IRE Trans. on Aeronautical and Navigational Electronics, vol. ANE-4, Dec. 1957, pp. 176-196.

23. Fried, W. R.: Performance Profiles and Future Outlook of Doppler Navigation Systems. IRE Trans. on Aeronautical and Navigational Electronics, vol. ANE-5, no. 4, Dec. 1958, pp. 194-199.

24. Fried, W. R.: An FM-CW Radar for Simultaneous Three-Dimentional Velocity and Altitude Measurement. IEEE Trans. on Aerospace and Navigational Electronics, March 1964, pp. 45-57.

25. Glassman, J. A.: A Generalization of the Fast Fourier Transform. IEEE Trans. on Computers, vol. C-19, no. 2, Feb. 1970, pp. 105-116.

26. Glegg, K.C.M.: A Low Noise CW Doppler Technique. Proceedings of 1958 National Conference on Aeronautical Electronics, Dayton, Ohio, pp. 133-144.

27. Gold, B.; and Rader, C. M.: Digital Processing of Signals. McGraw-Hill Book Co., Inc., 1969.

28. Gold, B.; and Jordan, K. L.: A Direct Search Procedure for Designing Finite Duration Impulse Response Filters. IEEE Trans. on Audio and Electroacoustics, vol. AU-17, no. 1, March 1969, pp. 33-36.

29. Helms, H. D.: Application of Fast Fourier Transforms to Digital Filtering-NEREM Convention Record, 1968, pp. 220-221.

30. Helms, H. D.: Nonrecursive Digital Filters: Design Methods for Achieving Specifications on Frequency Response. IEEE Trans. on Audio and Electroacoustics, vol. AU-16, no. 3, Sept. 1968, pp. 336-342.

31. Helms, H. D.: Applications of Digital Waveform Processing in Radars. IEEE International Convention Record, March 1970, pp. 180-181.

32. Hsia, J. C.: A Technique for Synthesizing Digital Filters. IEEE Trans. on Instrumentation and Measurement, vol. IM-18, no. 2, June 1969, pp. 93-96.

33. Jones, J. W.; and Caroll, C. C.: A Timed-Shared Digital Filter Realization. IEEE Trans. on Computers, vol. C-18, no. 11, Nov. 1969, pp. 1027-1030.

34. Jury, E. I.: Sampled-Data Control Systems. John Wiley and Sons, Inc., 1958.

35. Jury, E. I.: Theory and Application of the Z Transform Method. John Wiley and Sons, Inc., 1964.

36. Kuo, B. C.: Analysis and Synthesis of Sampled-Data Control Systems. Prentice-Hall, Inc., 1963.

37. Kuo, F. F.; and Kaiser, J. F. (editors): System Analysis by Digital Computer. John Wiley & Sons, Inc., 1966. (Chapt. 7 by Kaiser is entitled "Digital Filters." It contains an extensive bibliography at the end.)

38. Langenthal, I. M.: Coefficient Sensitivity and Generalized Digital Filter Synthesis. EASCON Convention Record, Sept. 1968, pp. 386-392.

39. Lee, J. D.: Review of Recent Work on Applications of Walsh Functions in Communications. Symposium on Walsh Functions at Naval Research Laboratories, March 1970. (Contains extensive bibliography on Walsh functions.)

40. Lewis, P. M.: Synthesis of Sampled-Signal Networks. IRE Trans. on Circuit Theory, vol. CT-5, March 1958, pp. 74-77.

41. Lendorff, D. P.: Theory of Sampled-Data Control Systems. John Wiley and Sons, Inc., 1965.

42. Manske, R. A.: Computer Simulation of Narrowband Systems. IEEE Trans. on Computers, vol. C-17, no. 4, April 1968, pp. 301-308.

43. Markel, J. D.: Z Transform Applications Using Digital Computers. Electro-Technology, vol. 82, no. 6, Dec. 1968, pp. 21-36.

44. Martin, M. A.: Frequency Domain Applications to Data Processing. IRE Trans. on Space Electronics and Telemetry, March 1959, pp. 33-41.

45. Matthews, S. B.: Generation of Pseudorandom Noise Having a Gaussian Spectral Density. IEEE Trans. on Computers, vol. C-17, April 1968, pp. 382-385.

46. Monroe, A. J.: Digital Processes for Sampled Data Systems. John Wiley and Sons, Inc., 1962.

47. Ormsby, J. F. A.: Design of Numerical Filters with Applications to Missile Data Processing. Journal ACM, vol. 8, no. 3, July 1961, pp. 440-466.

48. Papoulis, A.: Probability Random Variables, and Stochastic Processes, McGraw-Hill Book Co., Inc., 1965.

49. Parzen E.: Statistical Spectral Analysis (Single Channel Case) in 1968. Stanford University Dept. of Statistics Technical Report No. 11, June 10, 1968.

50. Pawula, R. F.: Analysis of an Estimator of the Center Frequency of a Power Spectrum. IEEE Trans. on Information Theory, vol. IT-14, no. 5, Sept. 1968, pp. 669-676.

51. Price, C. F.: Signal Processing in a Satellite Radar Altimeter. MIT Experimental Astronomy Laboratory Report no. RE-48, Aug. 1968.

52. Proceedings of the Symposium on Computer Processing in Communications. Polytechnic Press of the Polytechnic Institute of Brooklyn, April 1969, (Contains numerous papers by various authors dealing with digital signal processing.)

53. Rabiner, L. R.: A Design Technique for Nonrecursive Digital Filters. IEEE International Convention Record, March 1970, pp. 176-177.

54. Rabiner, L. R.; Schafer, R. W.; and Rader, C. M.: The Chirp Z-Transform Algorithm and its Application. Bell System Technical Journal, vol. 48, no. 5, May-June 1969, pp. 1249-1292.

55. Ragazzini, J. R.; and Franklin, G. F: *Sampled-Data Control Systems*. McGraw-Hill Book Co., Inc., 1958.

56. Rainal, A. J.: Zero-Crossing Principle for Detecting Narrow-Band Signals. IEEE Trans. on Instrumentation and Measurement, vol. IM-15, no. 1-2, March-June 1966, pp. 38-43.

57. Randall, C. M.: Fast Fourier Transform for Unequal Number of Input and Output Points. Applied Optics, vol. 6, no. 8, Aug. 1967, pp. 1432-1433.

58. Rhoads, R. L. and Ekstrom, M. D.: Removal of Intervening System Distortion by Deconvolution. IEEE Trans. on Instrumentation and Measurement, vol. IM-17, no. 4, Dec. 1968, pp. 333-337.

59. Sage, A. P.; and Smith, S. L.: Real-Time Digital Simulation for Systems Control. Proc. IEEE, vol. 54, no. 12, Dec. 1966, pp. 1802-1812.

60. Salzer, J. M.: Frequency Analysis of Digital Computers Operating in Real Time. Proc. IRE, vol. 42, Feb. 1954, pp. 457-466.

61. Schetzen, M.: The Power Density Spectrum of the Echo from an Airborne Doppler Radar. MIT Instrumentation Laboratory, Report no. R-541, March 1966.

62. Shanks, J. L.: Computation of the Fast Walsh-Fourier Transform. IEEE Trans. on Computers, vol. C-18, May 1969, pp. 457-459.

63. Shively, R. R.: A Digital Processor to Generate Spectra in Real Time. IEEE Trans. on Computers, vol. C-17, no. 5, May 1968, pp. 485-491.

64. Singleton, R. C.: On Computing the Fast Fourier Transform. Communications of the ACM, vol. 10, no. 10, Oct. 1967, pp. 647-654.

65. Skolnik, M. I.: Introduction to Radar Systems. McGraw-Hill Book Co., Inc., 1962.

66. Special Issue on Fast Fourier Transform, IEEE Trans. on Audio and Electroacoustics, vol. AU-15, no. 2, June 1967 (Contains 11 papers).

67. Special Issue on Digital Filters, IEEE Trans. on Audio and Electroacoustics, vol. AU-16, no. 3, Sept. 1968. (Contains 13 papers).

68. Special Issue on Fast Fourier Transform, IEEE Trans. on Audio and Electroacoustics, vol. AU-17, no. 2, June 1969. (Contains 17 papers including a comprehensive bibliography).

69. Steiglitz K.: The General Theory of Digital Filters with Applications to Spectral Analysis. AFOSR Report no. 64-1664, May 1963.

70. Swerling, P.: Estimation of Doppler Shifts in Noise Spectra. IRE International Convention Record, part 4, March 1960, pp. 148-153.

71. Tausworthe, R. C.: Theory and Practical Design of Phase-Locked Receivers. JPL Technical Report #32-819, Feb. 15, 1966.

72. Tierney, J.: A Method of Digital Frequency Synthesis. IEEE International Convention Record, March 1970, pp. 174-175.

73. Tou, J. T.: Digital and Sampled Data Control Systems. McGraw-Hill Book Co., Inc., 1959.

74. Wallace, N. D.: Performance Prediction Method for a Class of FM-CW Radars. IEEE Trans. on Aerospace and Navigational Electronics, March 1964, pp. 36-44.

75. Weinstein, C.: Quantization Effects in Frequency Sampling Filters. NEREM Convention Record, 1968, pp. 222-223.

Appendix A

Digital Frequency Tracker Program

```
PAGE   1

// JOB T

LOG DRIVE   CART SPEC   CART AVAIL  PHY DRIVE
  0000        0009        0009        0000

V2 M06   ACTUAL  8K  CONFIG  8K

// FOR
**SINGLE CHANNEL DOPPLER FREQUENCY TRACKER SIMULATION
*LIST SOURCE PROGRAM
*IOCS(CARD, 1132 PRINTER,KEYBOARD)
*IOCS(TYPEWRITER)
*EXTENDED PRECISION
```

```
   77 WRITE(1,78)
   78 FORMAT(' TURN SWITCH 3 OFF AND PROCEED WITH INPUTS')
      WRITE(1,112)
  112 FORMAT(' SW!TCH 1 ON TO SHIFT AND SWITCH 2 ON TO DELETE PRINT OUT'
     2)
      WRITE(1,201)
  201 FORMAT(' SWITCH 3 ON TO ENTER NEW DATA, SWITCH 4 ON FOR RAMP')
      WRITE(1,202)
  202 FORMAT(' SWITCH 5 ON TO PAUSE FOR SWITCH SETTING')
      WRITE(1,120)
C       RCON=SLOPE OF RAMP DOPPLER SHIFT
  120 FORMAT (' INPUT RCON IN F12.0 FORMAT')
      READ(6,11)RCON
C       G1=GAIN CONSTANT OF LOOP FILTER
      WRITE(1,113)
  113 FORMAT(' INPUT G1 IN F12.0 FORMAT')
      READ(6,11)G1
C       TCON=TIME CONSTANT OF LOOP FILTER
      WRITE(1,114)
  114 FORMAT(' INPUT TCON IN F12.0 FORMAT')
      READ(6,11)TCON
C       ALPHA=NUMBER OF POINTS PER CYCLE AT CENTER FREQUENCY
      WRITE(1,115)
  115 FORMAT(' INPUT ALPHA IN F12.0 FORMAT')
      READ(6,11)ALPHA
      WRITE(1,116)
C       SHIFT=FREQUENCY SHIFT FOR DOPPLER STEP
  116 FORMAT(' INPUT SHIFT IN F12.0 FORMAT')
      READ(6,11)SHIFT
      WRITE(1,118)
C       OBR=Q OF NOTCH FILTER
  118 FORMAT(' INPUT OBR IN F12.0 FORMAT')
      READ(6,11)OBR
   11 FORMAT(F12.0)
      WRITE(3,111)G1,TCON,ALPHA,SHIFT,OBR,RCON
  111 FORMAT(' G1=',F12.5,5X,'TCON=',F12.5,5X,'ALPHA=',F12.5,5X,'SHIFT='
     1,F12.5,5X,' OBR =',F12.5,2X,'RCON=',F7.4)
C       ASSUME CENTER FREQUENCY OF LOOP IS F=1.
C       TIME=TIME BASED ON PRECEDING ASSUMPTION
C       FREQ=FILTER ESTIMATE OF DOPPLER SHIFT
C       DOPPH=NORMALIZED PHASE OF INPUT=ACTUAL PHASE/2*PI
C       DOPLR=INPUT DOPPLER SIGNAL
```

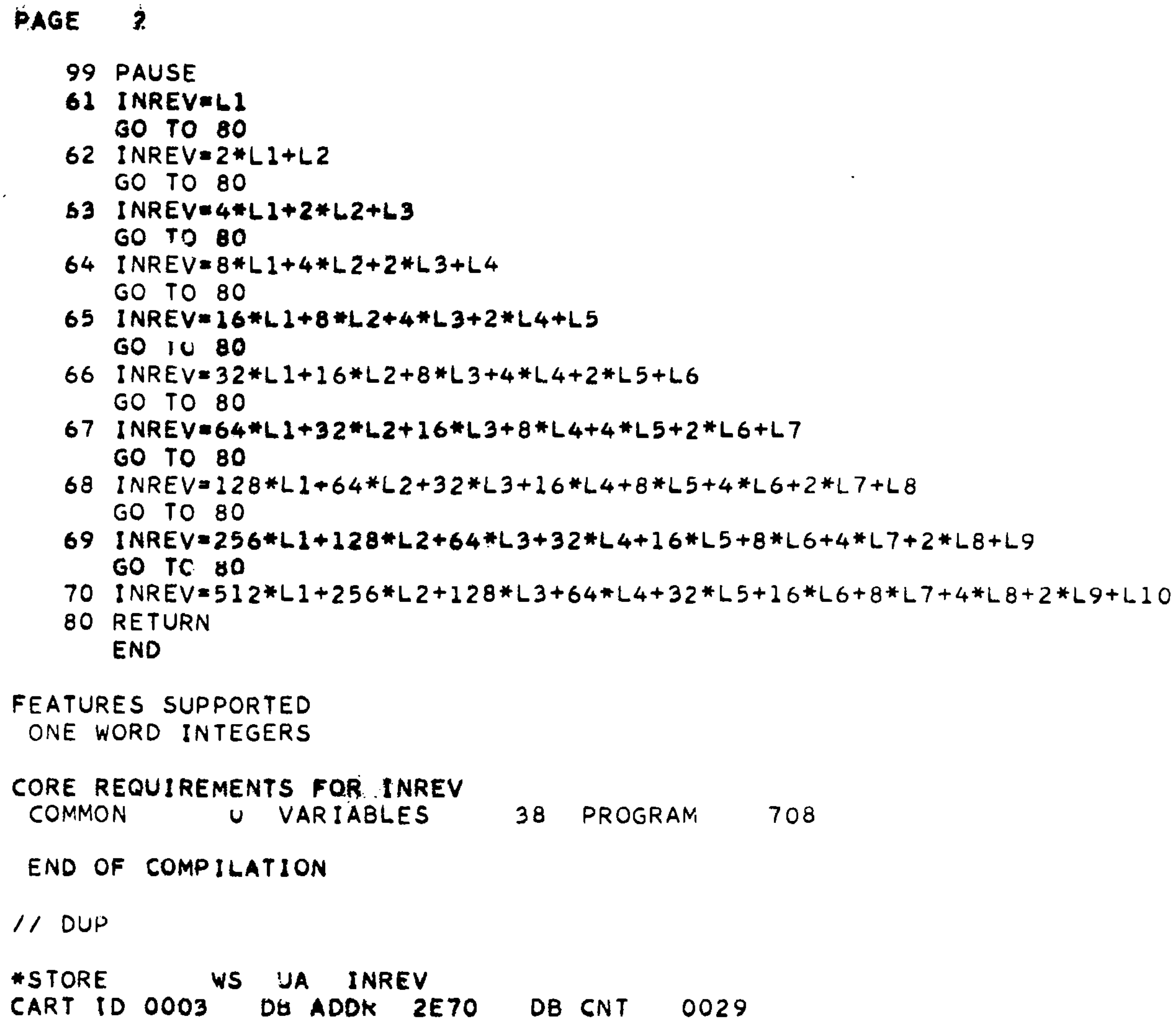
```
C        VCOPH=NORMALIZED INTEGRATED PHASE OF ESTIMATE
C        ERROR=OUTPUT OF PHASE DETECTOR
C        ERBRJ=ERROR SIGNAL AFTER PASSING THROUGH NOTCH FILTER
C        PHDIFF=NORMALIZED PHASE DIFFERENCE BETWEEN INPUT AND FEEDBACK
      WRITE(3,10)
   10 FORMAT('    TIME           FREQ           DOPPH          DOPLR
     1 VCOPH         VCOUT         ERROR         ERBRJ         PHDIFF')
      PI=3.141593
      H=G1*(1.+2.*ALPHA*TCON)/(2.*ALPHA)
      C=2./(1.+2.*ALPHA*TCON)
      DELT=1./ALPHA
      B=COS(2.*PI/ALPHA)/SIN(2.*PI/ALPHA)
      BNZ=1.+B**2.
      BN1=2.*(1.-B**2.)
      BN2=BNZ
      BDZ=BNZ+B/OBR
      BD1=BN1
      BD2=BNZ-B/OBR
      T=0.
      THETN=0.
      DOPPH=0.0
      VCOUT=1.
      Z=0.0
      FREQ=0.0
      VCOPH=0.
      ERR1=0.0
      ERR2=0.0
      ERB1=0.0
      ERB2=0.0
      F1=0.
      SHIFF=1.
   90 CALL DATSW(5,M)
      GO TO (82,2),M
   82 PAUSE
    2 CALL DATSW(3,L)
      GO TO (77,12),L
   12 CALL DATSW(1,J)
      GO TO (4,5),J
    4 CALL DATSW(4,N)
      GO TO (13,3),N
   13 SHIFT=(RCON/2.)*Z
    3 SHIFF=1.+SHIFT
```

```
PAGE   4                SINGLE CHANNEL DOPPLER FREQUENCY TRACKER SIMULATION
    5 DOPPH=DOPPH+SHIFF*DELT
      DOPLR=SIN(2.*PI*(DOPPH-IFIX(DOPPH)))
      ERROR=2.*DOPLR*VCOUT
      EN=BNZ*ERROR+BN1*ERR1+BN2*ERR2
      ED=BD1*ERB1+BD2*ERB2
      ERBRJ=(EN-ED)/BDZ
      FREQ=H*(ERBRJ-ERB1+C*ERB1)+F1
      CALL DATSW(2,K)
      TPQ=DOPPH-VCOPH
      GO TO (100,200),K
  200 WRITE(3,6)Z,FREQ,DOPPH,DOPLR,VCOPH,VCOUT,ERROR,ERBRJ,TPQ
    6 FORMAT(1H 8(F11.5,2X),4X,F11.5)
  100 ERR2=ERR1
      ERR1=ERROR
      ERB2=ERB1
      ERB1=ERBRJ
      F1=FREQ
      THETN=THETN+FREQ*DELT
      VCOPH=T+THETN
      VCOUT=COS(2.*PI*(VCOPH-IFIX(VCOPH)))
      T=T+DELT
      GO TO (300,400),J
  300 Z=Z+DELT
  400 CONTINUE
      GO TO 90
      END

FEATURES SUPPORTED
 EXTENDED PRECISION
 IOCS

CORE REQUIREMENTS FOR
 COMMON       0  VARIABLES     132  PROGRAM     888

 END OF COMPILATION
```

Appendix B

Fast Fourier Transform Program with In-Place Algorithm

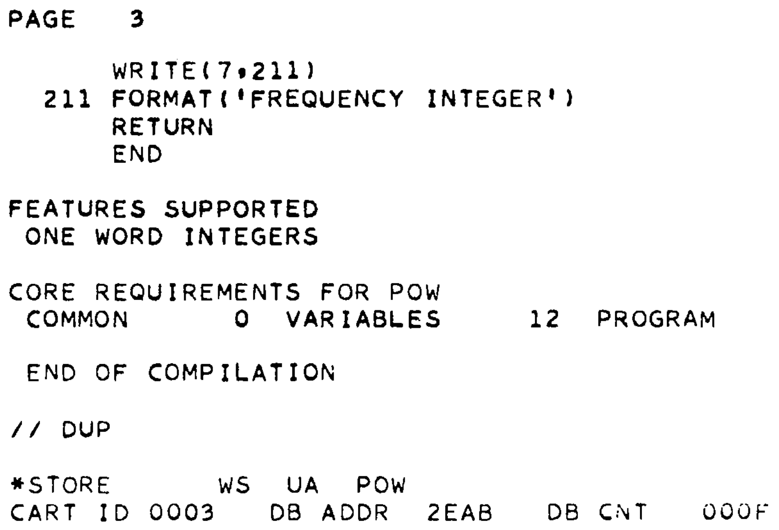
```
PAGE   1

// JOB

LOG DRIVE   CART SPEC   CART AVAIL  PHY DRIVE
  0000        0003        0003        0000

V2 M07   ACTUAL  8K  CONFIG  8K

// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
      FUNCTION INREV(K,NLOG)
C THIS FUNCTION IS USED TO FIND A BIT-REVERSED NUMBER
      L1A=K/2
      L1=K-2*L1A
      IF(NLOG-1)61,61,52
   52 L2A=L1A/2
      L2=L1A-2*L2A
      IF(NLOG-2)62,62,53
   53 L3A=L2A/2
      L3=L2A-2*L3A
      IF(NLOG-3)63,63,54
   54 L4A=L3A/2
      L4=L3A-2*L4A
      IF(NLOG-4)64,64,55
   55 L5A=L4A/2
      L5=L4A-2*L5A
      IF(NLOG-5)65,65,56
   56 L6A=L5A/2
      L6=L5A-2*L6A
      IF(NLOG-6)66,66,57
   57 L7A=L6A/2
      L7=L6A-2*L7A
      IF(NLOG-7)67,67,58
   58 L8A=L7A/2
      L8=L7A-2*L8A
      IF(NLOG-8)68,68,59
   59 L9A=L8A/2
      L9=L8A-2*L9A
      IF(NLOG-9)69,69,60
   60 L10A=L9A/2
      L10=L9A-2*L10A
      IF(NLOG-10)70,70,99
```



```
PAGE    2

    99 PAUSE
    61 INREV=L1
       GO TO 80
    62 INREV=2*L1+L2
       GO TO 80
    63 INREV=4*L1+2*L2+L3
       GO TO 80
    64 INREV=8*L1+4*L2+2*L3+L4
       GO TO 80
    65 INREV=16*L1+8*L2+4*L3+2*L4+L5
       GO IU 80
    66 INREV=32*L1+16*L2+8*L3+4*L4+2*L5+L6
       GO TO 80
    67 INREV=64*L1+32*L2+16*L3+8*L4+4*L5+2*L6+L7
       GO TO 80
    68 INREV=128*L1+64*L2+32*L3+16*L4+8*L5+4*L6+2*L7+L8
       GO TO 80
    69 INREV=256*L1+128*L2+64*L3+32*L4+16*L5+8*L6+4*L7+2*L8+L9
       GO TC 80
    70 INREV=512*L1+256*L2+128*L3+64*L4+32*L5+16*L6+8*L7+4*L8+2*L9+L10
    80 RETURN
       END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR INREV
 COMMON       0  VARIABLES     38   PROGRAM    708

 END OF COMPILATION

// DUP

*STORE       WS   UA   INREV
CART ID 0003   DB ADDR  2E70   DB CNT   0029
```

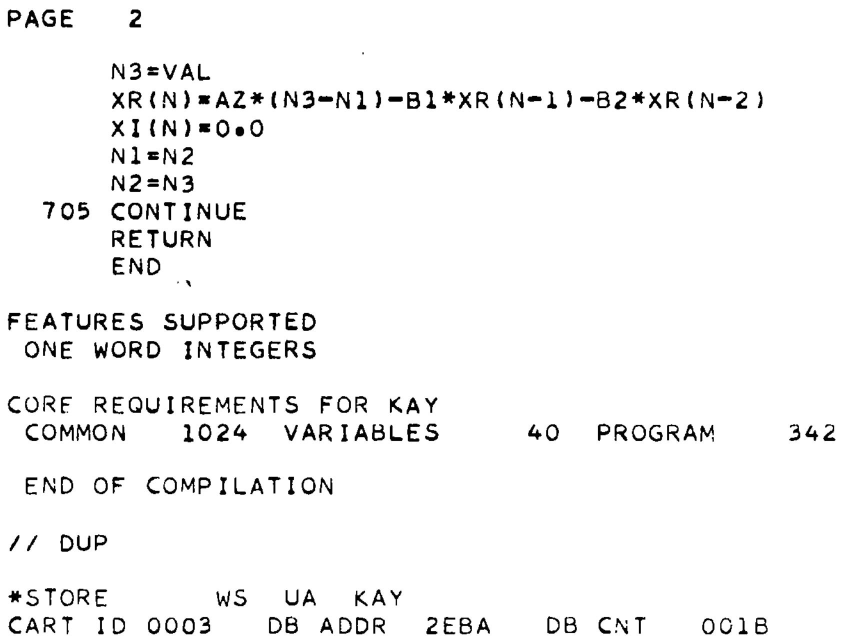
```
PAGE    1

// JOB

LOG DRIVE    CART SPEC    CART AVAIL   PHY DRIVE
  0000         0003         0003         0000

V2 M07   ACTUAL  8K  CONFIG  8K

// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
      SUBROUTINE DOP(BW)
      CALL SCALF(.0234,1.,0.,0.)
      CALL FGRID(0,0.,6.50,16.,16)
      CALL FGRID(0,0.,7.75,16.,16)
      CALL FGRID(3,0.,7.75,.25,5)
      CALL FGRID(1,0.,7.75,.25,5)
      A=32.
      DO 200 I=1,8
      IA=A
      CALL FCHAR(A-8.,6.3,.1,.1,0.)
      WRITE(7,201)IA
  201 FORMAT(I3)
      A=A+32.
  200 CONTINUE
      C=-1.0
      B=6.75
      DO 202 J=1,5
      CALL FCHAR(-25.,B,.1,.1,0.)
      WRITE(7,203)C
  203 FORMAT(F4.1)
      C=C+.5
  202 B=B+.5
      CALL FCHAR(64.,9.75,.2,.2,0.)
      WRITE(7,204)
  204 FORMAT('DOPPLER RETURN')
      CALL FCHAR(128.,5.9,.2,.2,0.)
      WRITE(7,210)
  210 FORMAT('TIME INTEGER')
      CALL FCHAR(94.,9.25,.15,.15,0.0)
      WRITE(7,337)BW
  337 FORMAT('(BW=',F7.2,2X,'PERCENT)')
      CALL FPLOT(3,0.0,8.25)
```

```
PAGE   2

      RETURN
      END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR DOP
 COMMON       0  VARIABLES     12  PROGRAM    266

 END OF COMPILATION

// DUP

*STORE      WS  UA  DOP
CART ID 0003   DB ADDR  2E99   DB CNT   0012

// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
      SUBROUTINE POW
      CALL FPLOT(+1,0.0,+2.00)
      CALL SCALF(.0468,2.,0.,0.)
      CALL FGRID(1,0.,0.,.20,6)
      CALL FGRID(0,0.,0.,8.,16)
      AA=16.
      DO 205 I=1,8
      IAA=AA
      CALL FCHAR(AA-4.,-.1,.1,.1,0.)
      WRITE(7,206)IAA
  206 FORMAT(I3)
  205 AA=AA+16.
      CC=0.
      DO 207 I=1,6
      CALL FCHAR(-10.,CC,.1,.1,0.)
      DD=CC+.01
      WRITE(7,208)DD
  208 FORMAT(F3.1)
  207 CC=CC+.2
      CALL FCHAR(32.,1.5,.2,.2,0.)
      WRITE(7,209)
  209 FORMAT('POWER SPECTRUM')
      CALL FCHAR(56.,-.3,.2,.2,0.)
```

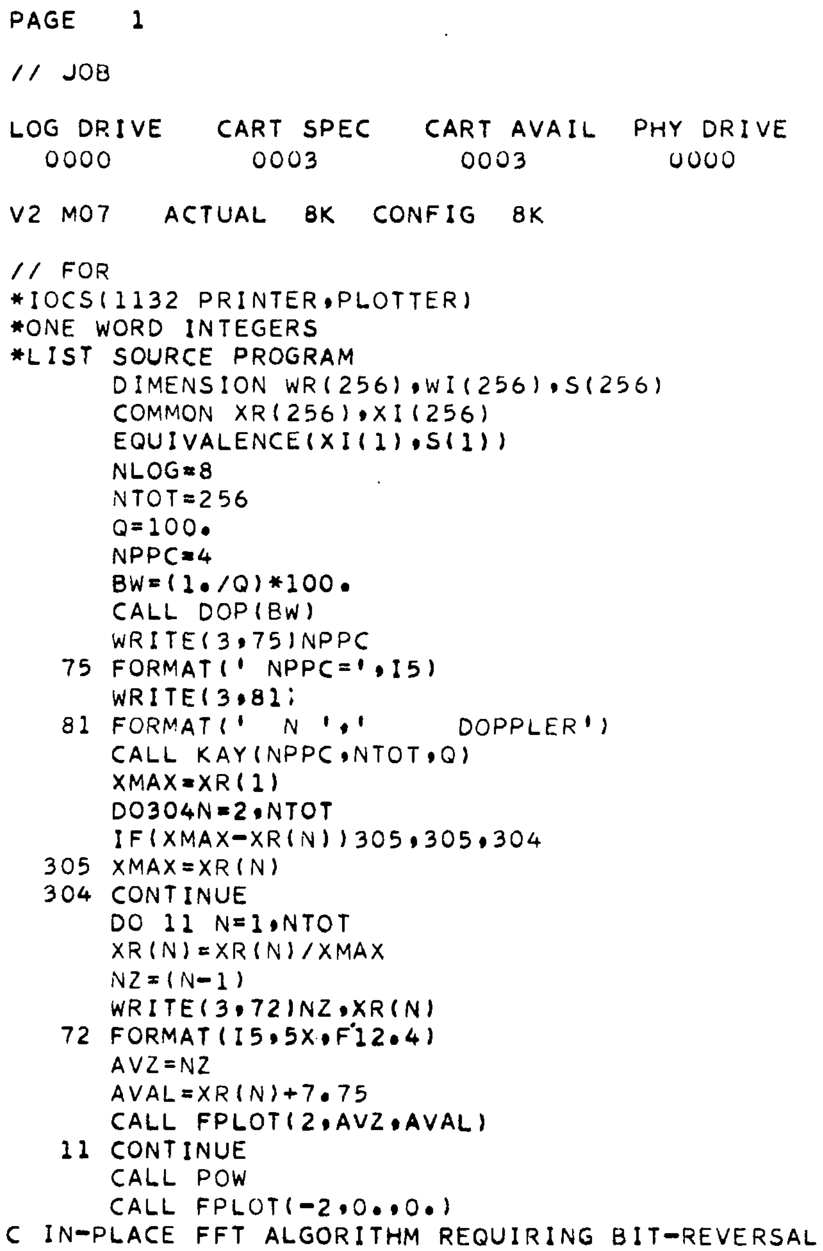


```
PAGE   3

      WRITE(7,211)
  211 FORMAT('FREQUENCY INTEGER')
      RETURN
      END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR POW
 COMMON       0  VARIABLES      12  PROGRAM     212

 END OF COMPILATION

// DUP

*STORE       WS  UA  POW
CART ID 0003   DB ADDR  2EAB   DB CNT    000F
```

```
PAGE   1

// JOB

LOG DRIVE   CART SPEC   CART AVAIL  PHY DRIVE
  0000        0003        0003        0000

V2 M07   ACTUAL  8K  CONFIG  8K

// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
      SUBROUTINE KAY(NPPC,NTOT,Q)
C GENERATES DOPPLER SIGNAL
      REAL N1,N2,N3
      COMMON XR(256),XI(256)
      N1=0.0
      N2=0.0
      N3=0.0
      XR1=0.0
      XR2=0.0
      AAA=3.14159/NPPC
      A=COS(AAA)/SIN(AAA)
      AZ=(A/Q)/(1.+(A/Q)+A**2)
      B1=(2.-2.*A**2)/(1.+(A/Q)+A**2)
      B2=(1.-(A/Q)+A**2)/(1.+A/Q+A**2)
      IY=33
      M=0.0
 703  CALL GAUSS(IY,1.0,0.,VAL)
      N3=VAL
      XR3=AZ*(N3-N1)-B1*XR2-B2*XR1
      N1=N2
      N2=N3
      XR1=XR2
      XR2=XR3
      M=M+1
      IF(M-15.*Q)703,703,704
  704 XR(1)=XR1
      XR(2)=XR2
      XI(2)=0.0
      XI(1)=0.0
      IX=31
      DO705N=3,NTOT
      CALL GAUSS(IX,1.0,0.0,VAL)
```



```
PAGE    2

      N3=VAL
      XR(N)=AZ*(N3-N1)-B1*XR(N-1)-B2*XR(N-2)
      XI(N)=0.0
      N1=N2
      N2=N3
  705 CONTINUE
      RETURN
      END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR KAY
 COMMON   1024  VARIABLES     40  PROGRAM    342

 END OF COMPILATION

// DUP

*STORE      WS  UA  KAY
CART ID 0003   DB ADDR  2EBA   DB CNT   001B
```



```
PAGE   1

// JOB

LOG DRIVE   CART SPEC   CART AVAIL  PHY DRIVE
  0000        0003        0003        0000

V2 M07   ACTUAL  8K  CONFIG  8K

// FOR
*IOCS(1132 PRINTER,PLOTTER)
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
      DIMENSION WR(256),WI(256),S(256)
      COMMON XR(256),XI(256)
      EQUIVALENCE(XI(1),S(1))
      NLOG=8
      NTOT=256
      Q=100.
      NPPC=4
      BW=(1./Q)*100.
      CALL DOP(BW)
      WRITE(3,75)NPPC
   75 FORMAT(' NPPC=',I5)
      WRITE(3,81;
   81 FORMAT('  N ','      DOPPLER')
      CALL KAY(NPPC,NTOT,Q)
      XMAX=XR(1)
      DO304N=2,NTOT
      IF(XMAX-XR(N))305,305,304
  305 XMAX=XR(N)
  304 CONTINUE
      DO 11 N=1,NTOT
      XR(N)=XR(N)/XMAX
      NZ=(N-1)
      WRITE(3,72)NZ,XR(N)
   72 FORMAT(I5,5X,F12.4)
      AVZ=NZ
      AVAL=XR(N)+7.75
      CALL FPLOT(2,AVZ,AVAL)
   11 CONTINUE
      CALL POW
      CALL FPLOT(-2,0.,0.)
C IN-PLACE FFT ALGORITHM REQUIRING BIT-REVERSAL
```

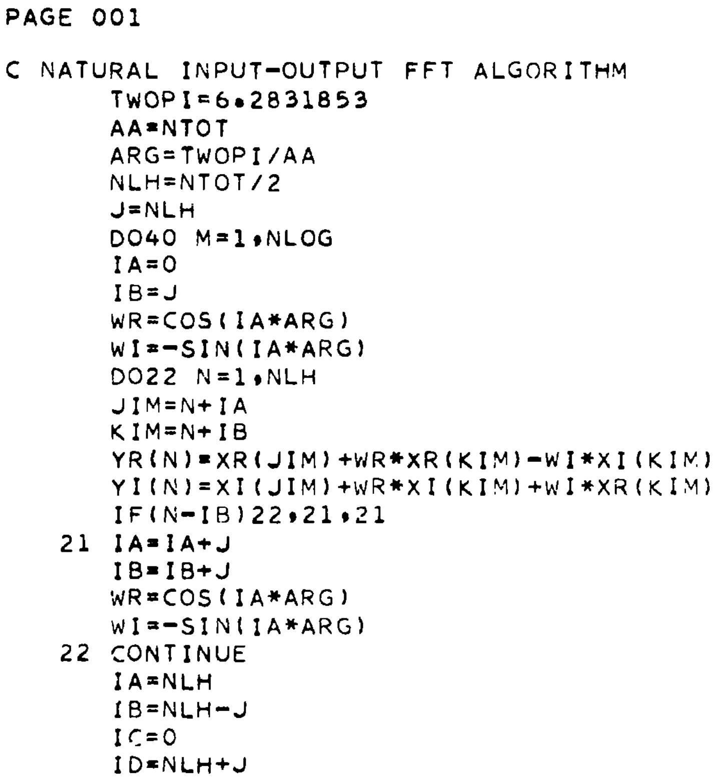
PAGE   2

```
      TWOPI=6.2831853
      AA=NTOT
      AB=TWOPI/AA
      DO22 N=1,NTOT
      AC=N-1.
      AD=AB*AC
      WR(N)=COS(AD)
   22 WI(N)=-SIN(AD)
      J=NTOT/2
      DO 35 M=1,NLOG
      JA=J
      IA=NLOG-M
      IDIV=2**IA
      DO 34 N=1,NTOT
      IARG=(N-1)/IDIV
      I=INREV(IARG,NLOG)+1
      IF(N-JA)31,31,32
   31 NJ1=N+J
      XR(N)=XR(N)+WR(I)*XR(NJ1)-WI(I)*XI(NJ1)
      XI(N)=XI(N)+WR(I)*XI(NJ1)+WI(I)*XR(NJ1)
      GO TO 34
   32 NJ2=N-J
      XAN=XR(N)
      XR(N)=XR(NJ2)+2.*(WR(I)*XR(N)-WI(I)*XI(N))
      XI(N)=XI(NJ2)+2.*(WR(I)*XI(N)+WI(I)*XAN)
      IF(N-J-JA)34,33,33
   33 JA=JA+2*J
   34 CONTINUE
      J=J/2
   35 CONTINUE
      WRITE(3,100)
  100 FORMAT('  N       SPECTRUM')
      DO 40 N=1,NTOT
      XR(N)=XR(N)**2+XI(N)**2
   40 CONTINUE
      DO 73 N=1,NTOT
      NA=N-1
      NB=INREV(NA,NLOG)+1
      S(N)=XR(NB)
C END OF IN-PLACE FFT ALGORITHM
      NCB=N-1
      WRITE(3,101)NCB,S(N)
```

```
PAGE   3

  101 FORMAT(2X,I5,5X,F12.4)
   73 CONTINUE
      NN=NTOT/2
      AMAX=S(1)
      DO 155 N=2,NN
      IF(AMAX-S(N))154,154,155
  154 AMAX=S(N)
  155 CONTINUE
      WRITE(3,156)AMAX
  156 FORMAT(3X,'AMAX=',F12.4)
      DO 157 N=1,NN
      S(N)=S(N)/AMAX
      CUN=N-1
      CALL FPLOT(1,CUN,0.)
      CALL FPLOT(+2,CUN,S(N))
      CALL FPLOT(0,CUN,0.)
  157 CONTINUE
      TTT=0.0
      DO500N=1,128
  500 TTT= S(N)+TTT
      DO501N=1,128
  501 S(N)=S(N)/TTT
      XMEAN=0.0
      DO502N=1,128
  502 XMEAN=N*S(N)+XMEAN
      VARX=0.0
      DO503N=1,128
  503 VARX=VARX+((N-XMEAN)**2)* S(N)
      VARX=VARX**.5
      XMEAN=XMEAN-1.
      CALL ABLE(XMEAN)
      CALL FPLOT(1,200.,-1.0)
      WRITE(3,504)NPPC,Q,XMEAN,VARX
  504 FORMAT(3X,'NPPC=',I5,10X,'Q=',F12.4,10X,'MEAN=',F12.4,10X,'SIGMA',
     1F12.4)
      CALL EXIT
      END

FEATURES SUPPORTED
 ONE WORD INTEGERS
 IOCS

CORE REQUIREMENTS FOR
 COMMON    1024  VARIABLES   1080  PROGRAM    944
```

Appendix C

Fast Fourier Transform Algorithm with Natural Input-Output for Data



```
PAGE 001

C NATURAL INPUT-OUTPUT FFT ALGORITHM
      TWOPI=6.2831853
      AA=NTOT
      ARG=TWOPI/AA
      NLH=NTOT/2
      J=NLH
      DO40 M=1,NLOG
      IA=0
      IB=J
      WR=COS(IA*ARG)
      WI=-SIN(IA*ARG)
      DO22 N=1,NLH
      JIM=N+IA
      KIM=N+IB
      YR(N)=XR(JIM)+WR*XR(KIM)-WI*XI(KIM)
      YI(N)=XI(JIM)+WR*XI(KIM)+WI*XR(KIM)
      IF(N-IB)22,21,21
   21 IA=IA+J
      IB=IB+J
      WR=COS(IA*ARG)
      WI=-SIN(IA*ARG)
   22 CONTINUE
      IA=NLH
      IB=NLH-J
      IC=0
      ID=NLH+J
```

PAGE 002

```
      WR=COS(IC*ARG)
      WI=-SIN(IC*ARG)
      NL1=NLH+1
      DO32N=NL1,NTOT
      LIM=N-IA
      NIM=N-IB
      YR(N)=XR(LIM)-WR*XR(NIM)+WI*XI(NIM)
      YI(N)=XI(LIM)-WR*XI(NIM)-WI*XR(NIM)
      IF(N-ID)32,31,31
   31 IA=IA-J
      IB=IB-J
      IC=IC+J
      ID=ID+J
      WR=COS(IC*ARG)
      WI=-SIN(IC*ARG)
   32 CONTINUE
      DO38N=1,NTOT
      XR(N)=YR(N)
   38 XI(N)=YI(N)
      J=J/2
   40 CONTINUE
      DO 73 N=1,NLH
      S(N)=XR(N)**2+XI(N)**2
   73 CONTINUE
C END OF NATURAL INPUT-OUTPUT FFT ALGORITHM
```

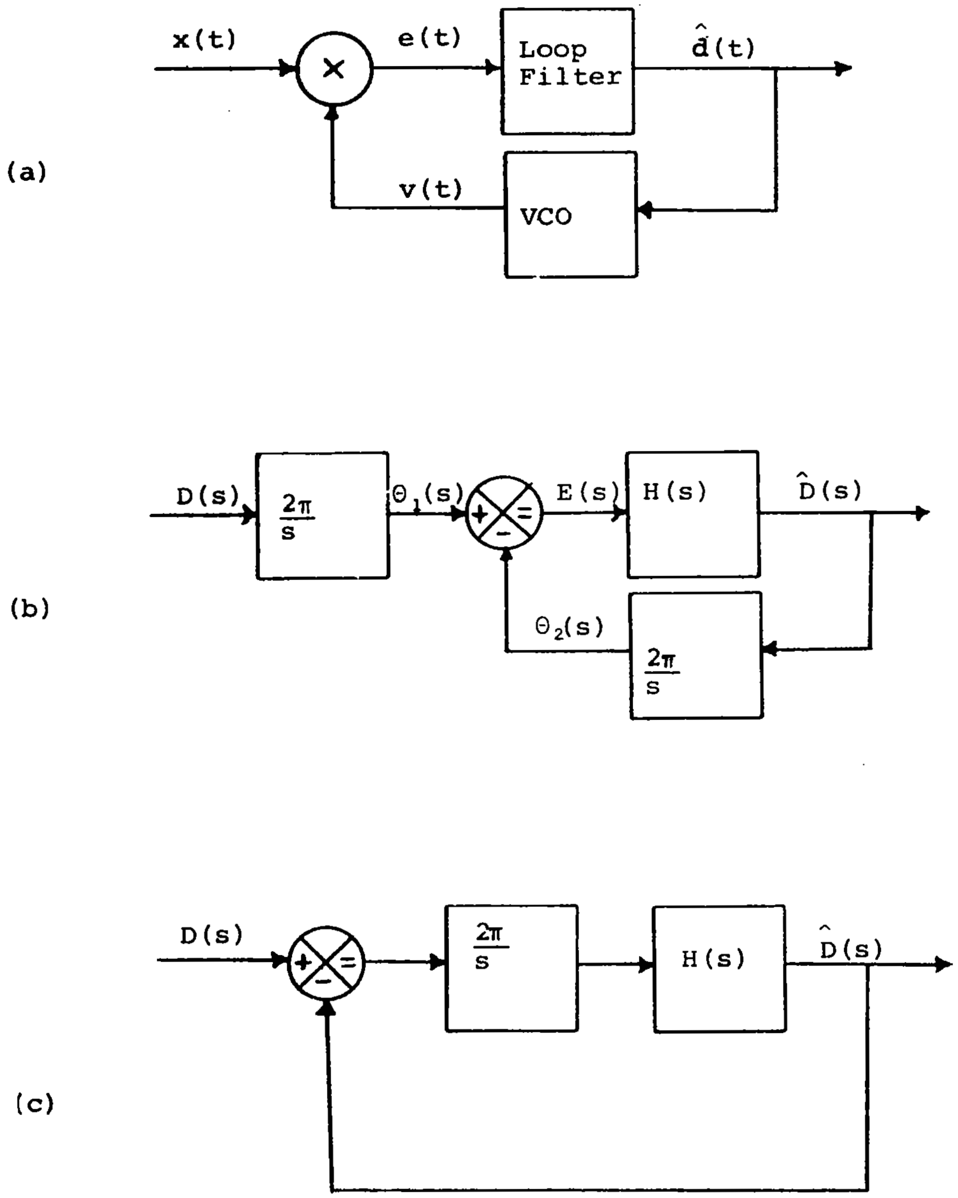


Fig. 1. Simplified block diagrams of analog phase-locked loop.

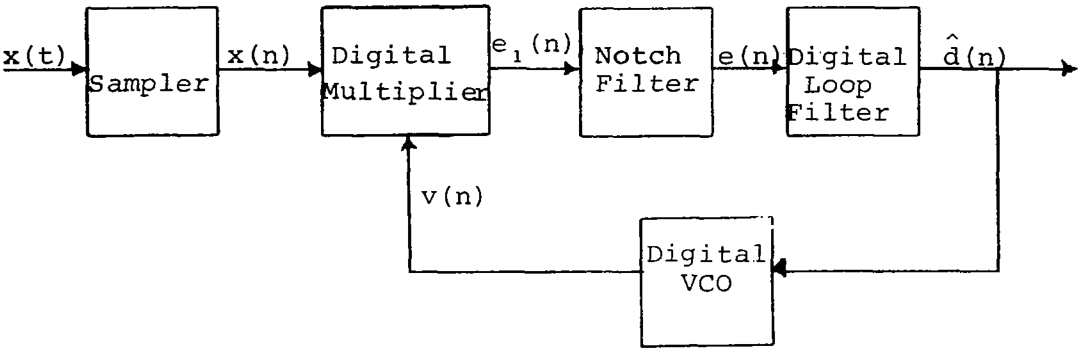


Fig. 2. Block diagram of digital phase-locked loop.

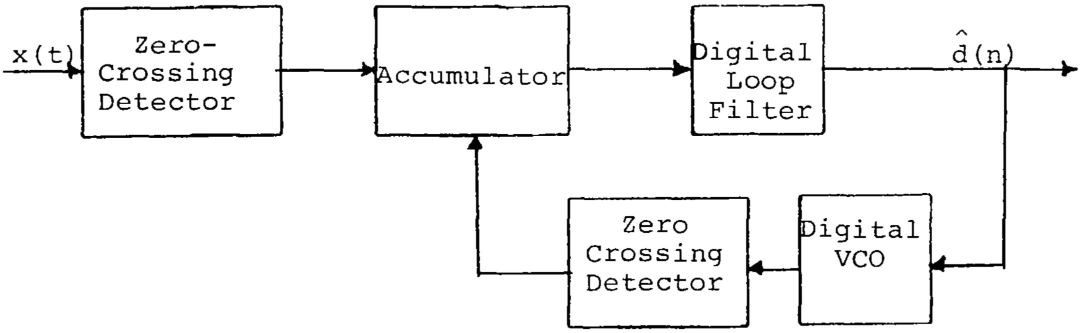


Fig. 3. Digital phase-locked loop with zero-crossing detector.

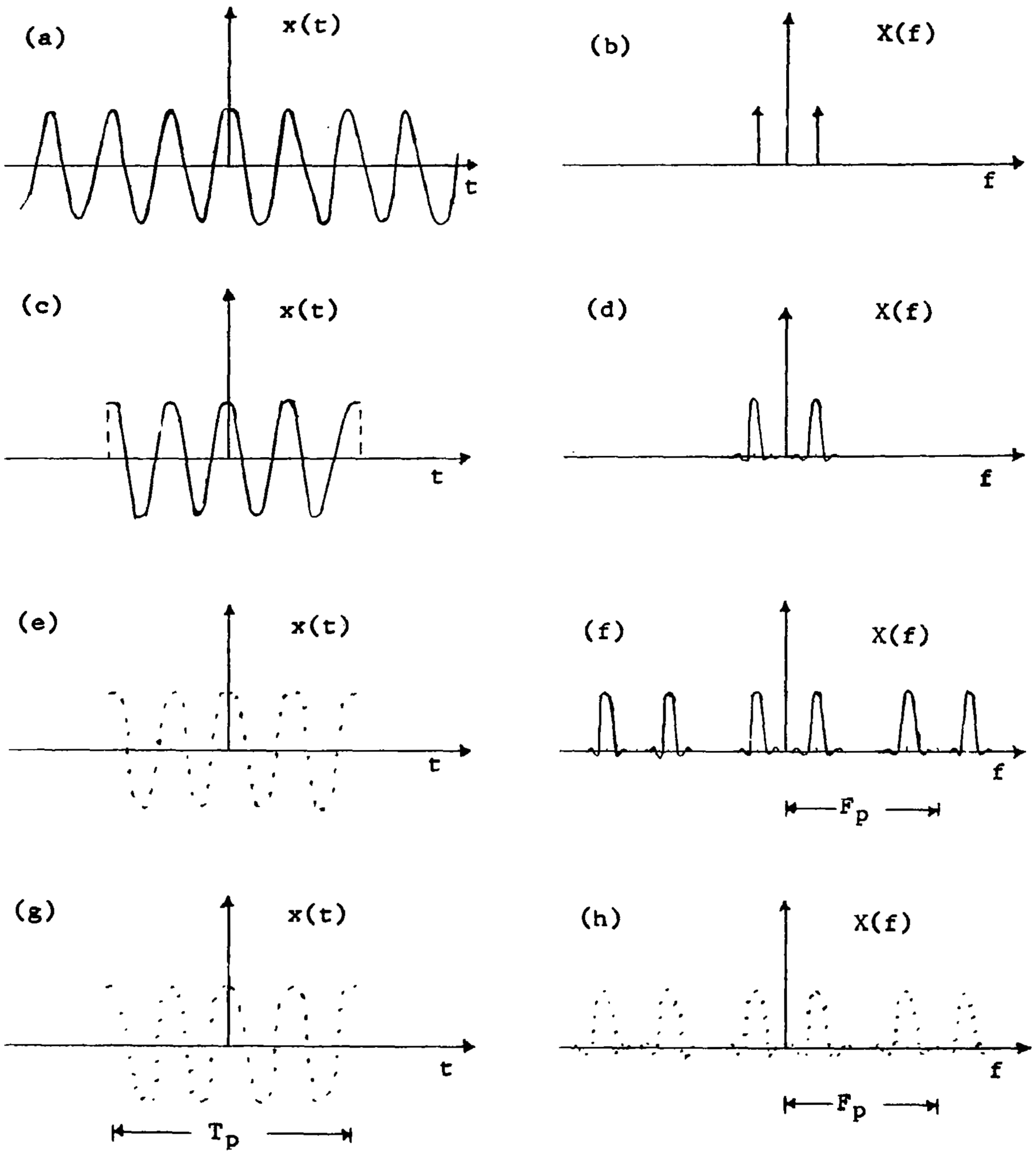


Fig. 4. Illustration of properties of the discrete Fourier transform.

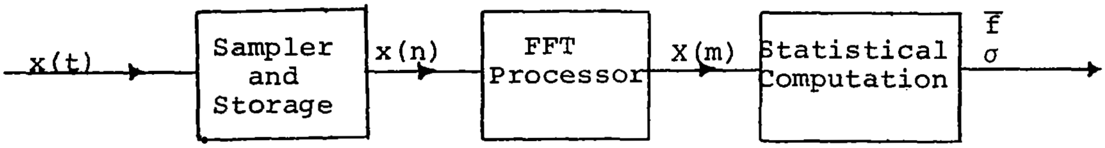


Fig. 5. Block diagram of open-loop FFT processor.

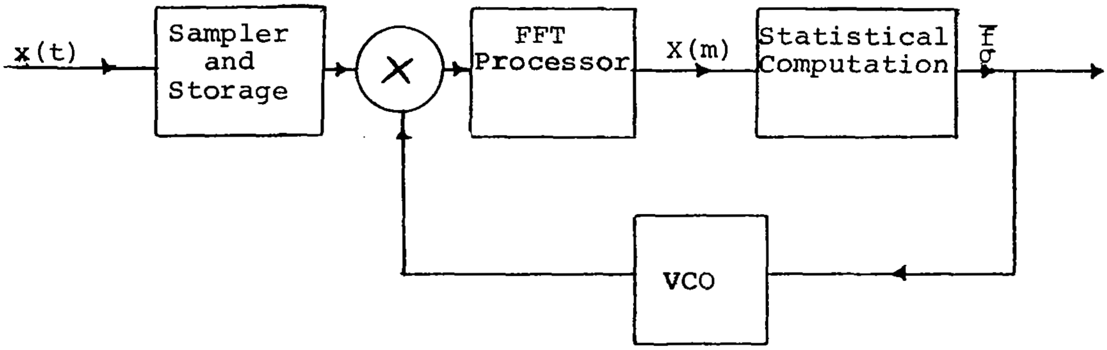


Fig. 6. Block diagram of close-loop FFT processor.

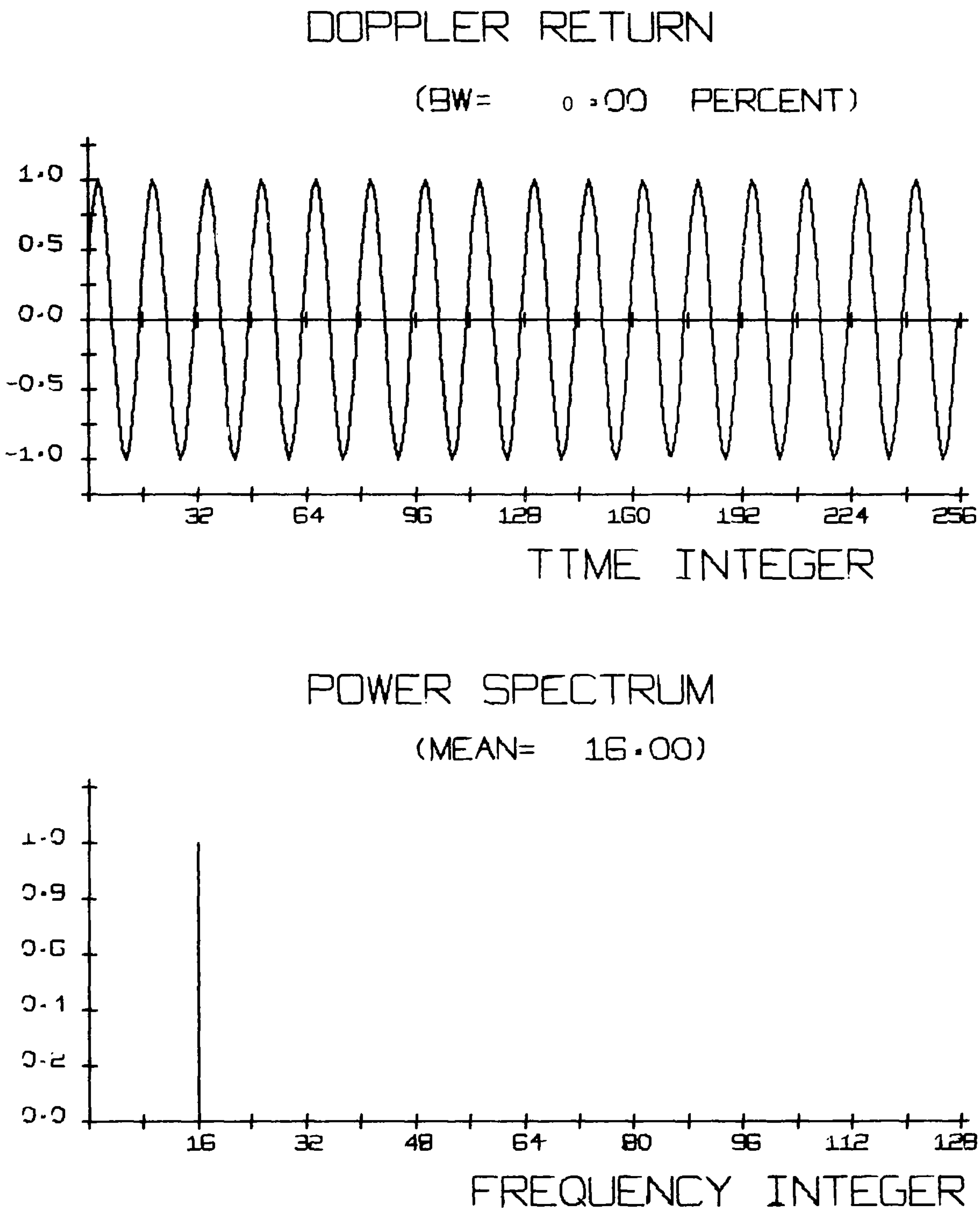


Fig. 7. Spectral analysis of sinusoid with 16 points per cycle.

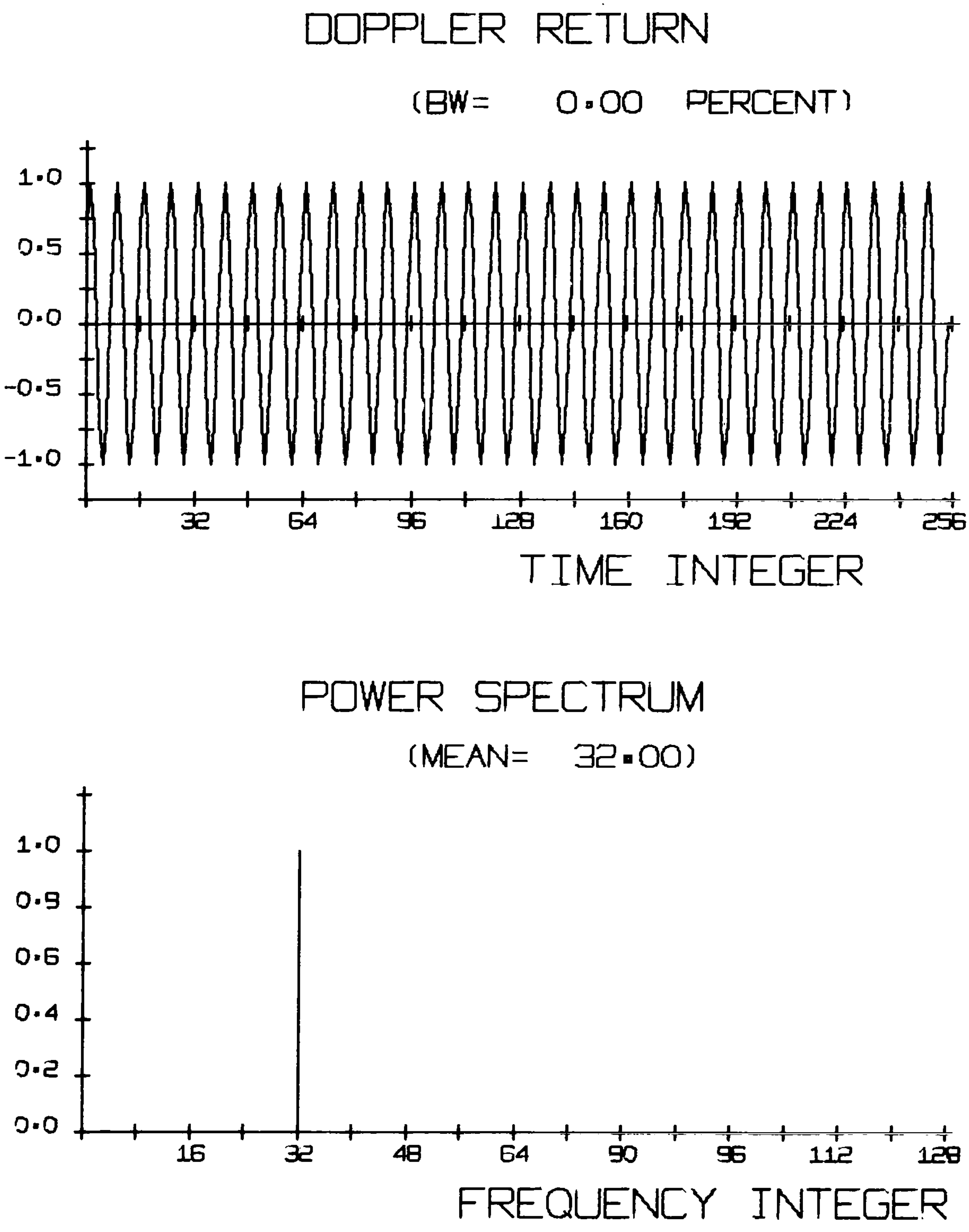


Fig. 8. Spectral analysis of sinusoid with 8 points per cycle.

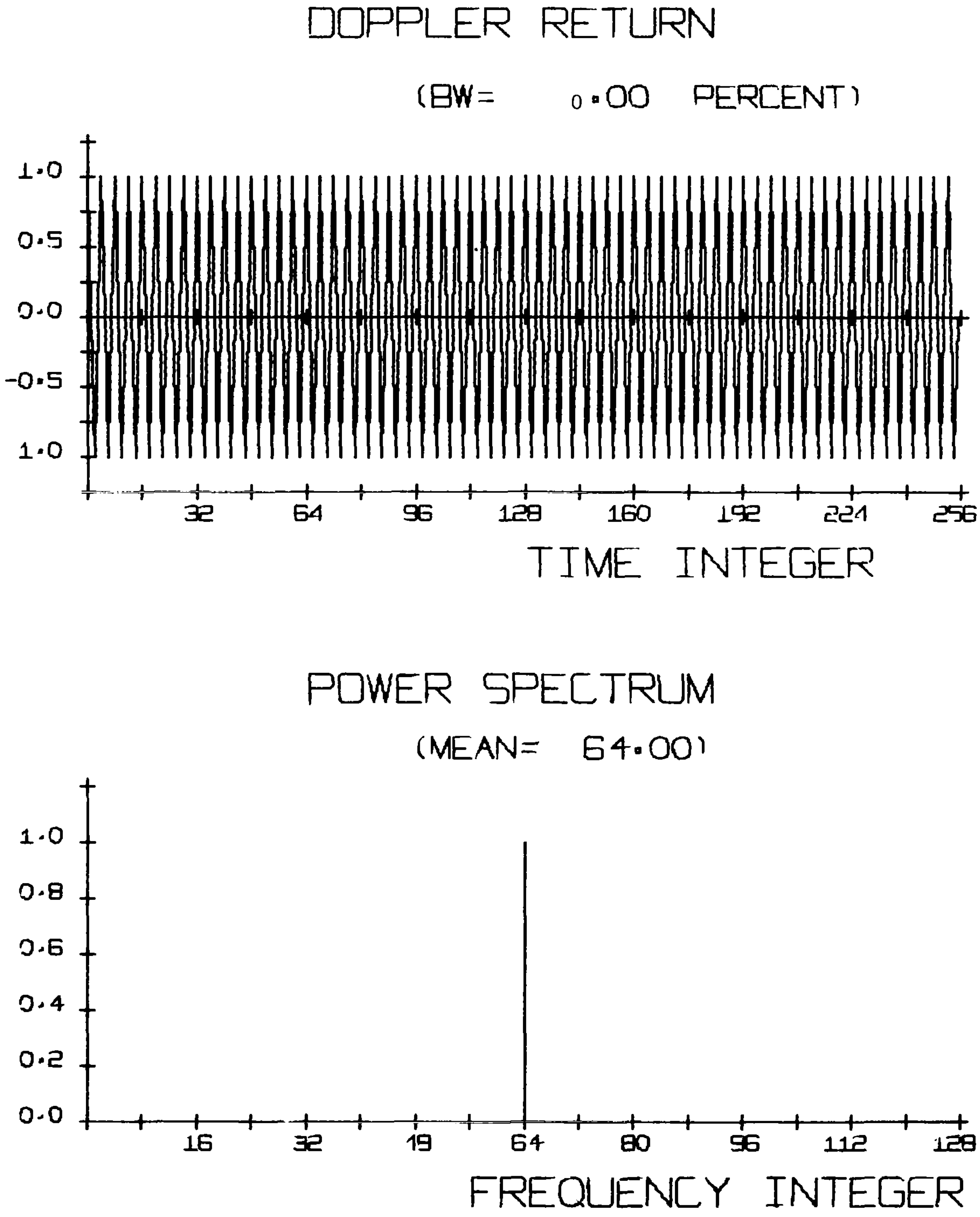


Fig. 9. Spectral analysis of sinusoid with 4 points per cycle.

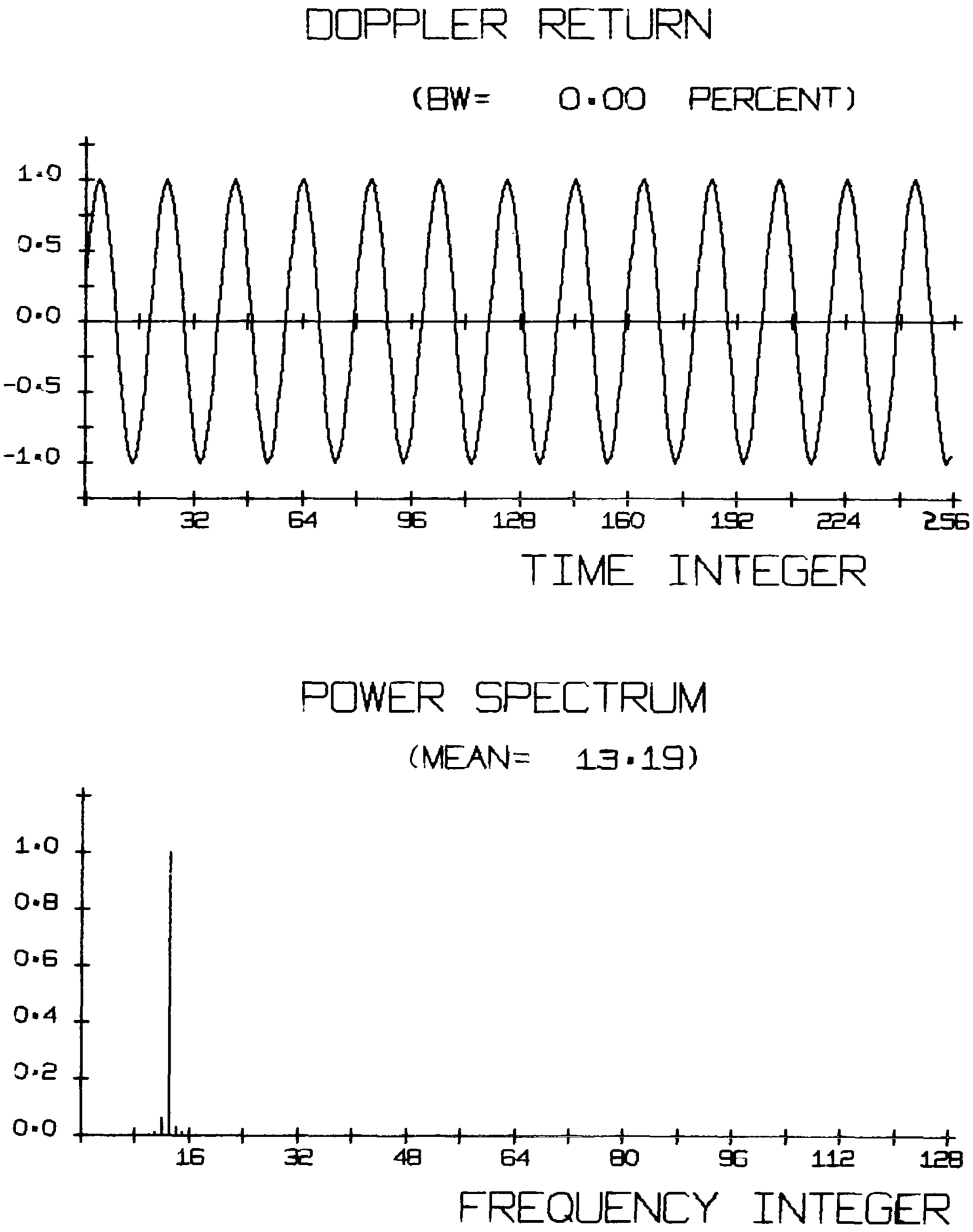


Fig. 10. Spectral analysis of sinusoid with 20 points per cycle.

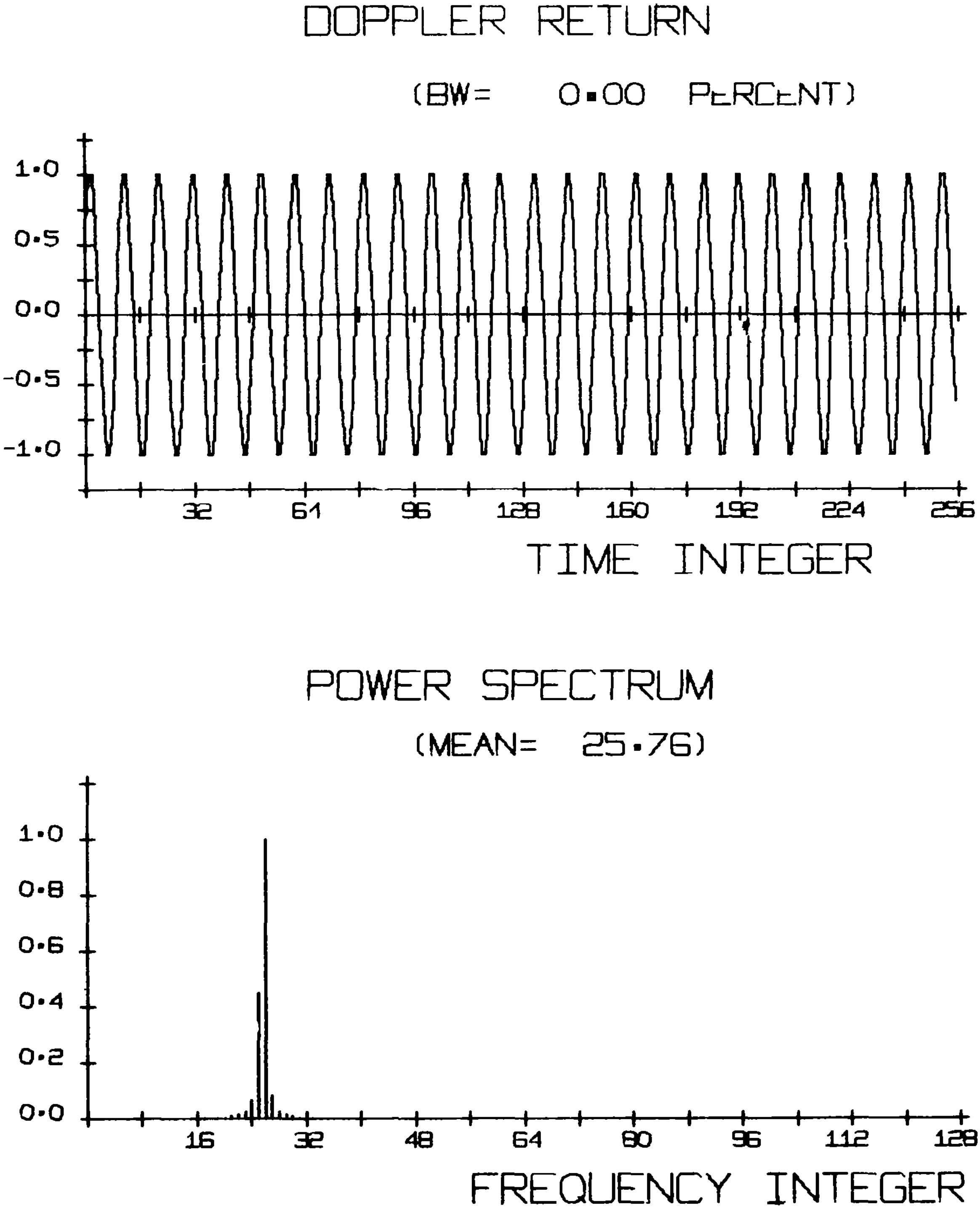


Fig. 11. Spectral analysis of sinusoid with 10 points per cycle.

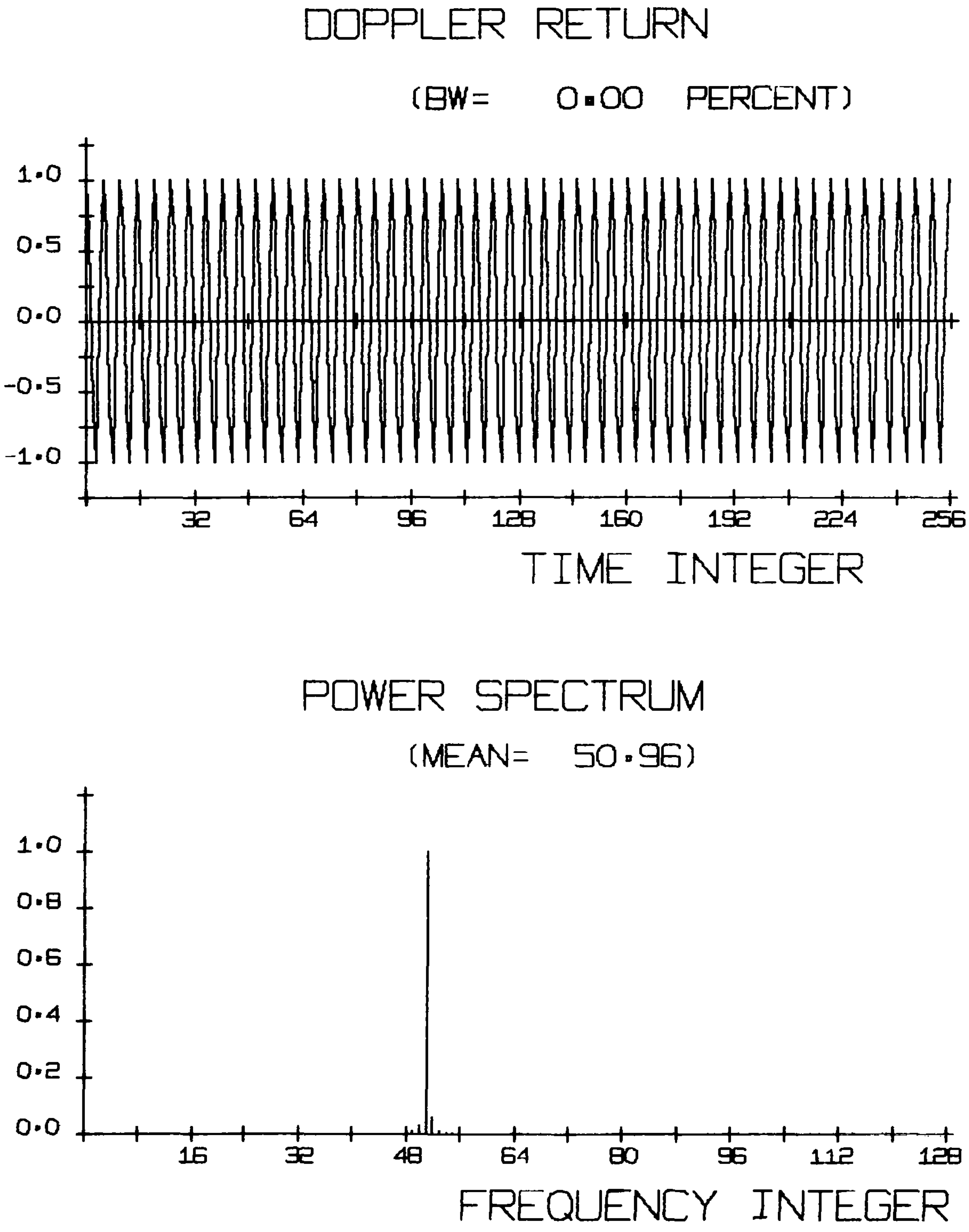
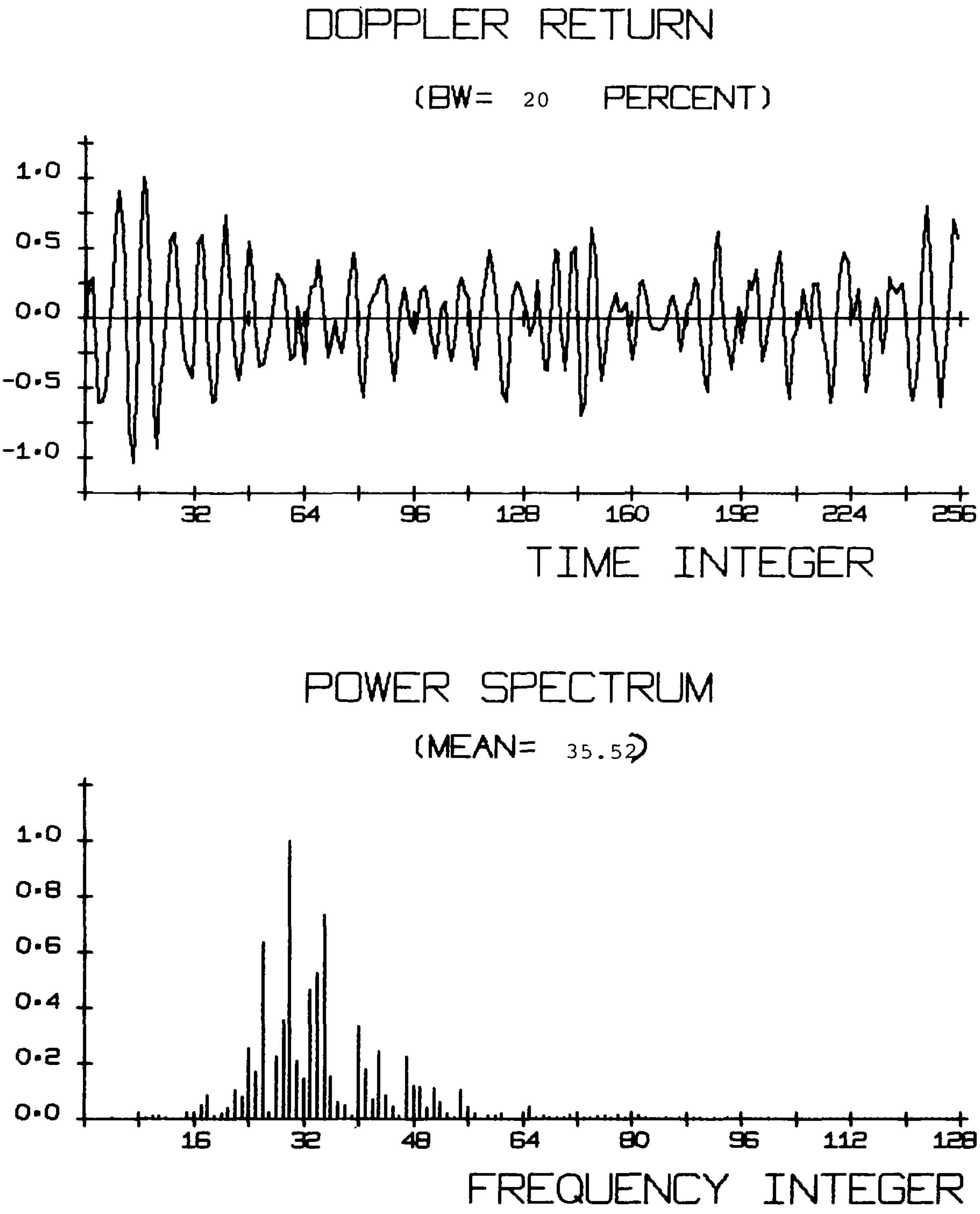


Fig. 12. Spectral analysis of sinusoid with 5 points per cycle.



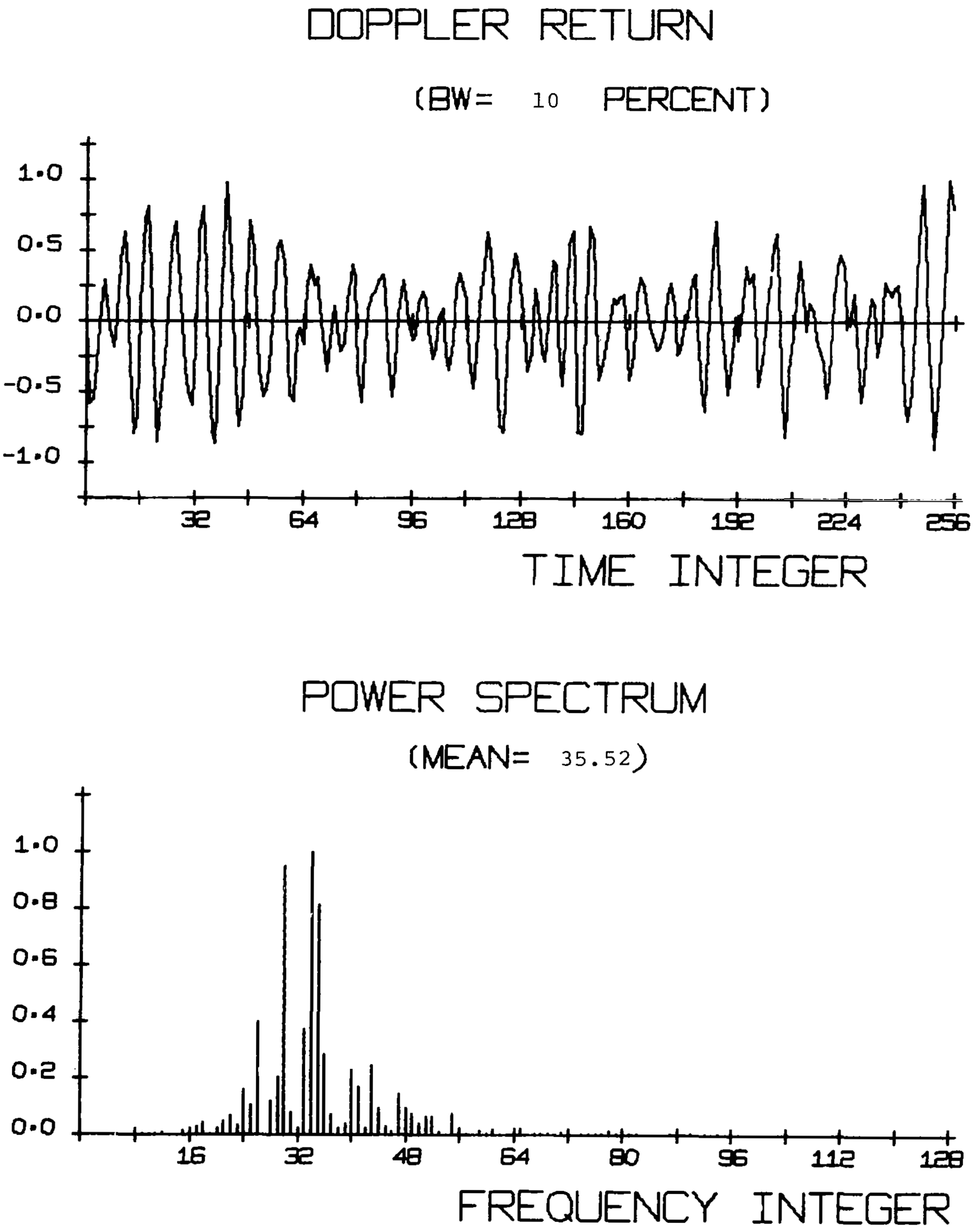
DOPPLER RETURN

(BW= 20 PERCENT)

POWER SPECTRUM

(MEAN= 35.52)

Fig. 13. Spectral analysis of Doppler return with BW=20% and 8 points per cycle at band-center.


DOPPLER RETURN

(BW= 10 PERCENT)

POWER SPECTRUM

(MEAN= 35.52)

Fig. 14. Spectral analysis of Doppler return with BW=10% and 8 points per cycle at band-center.

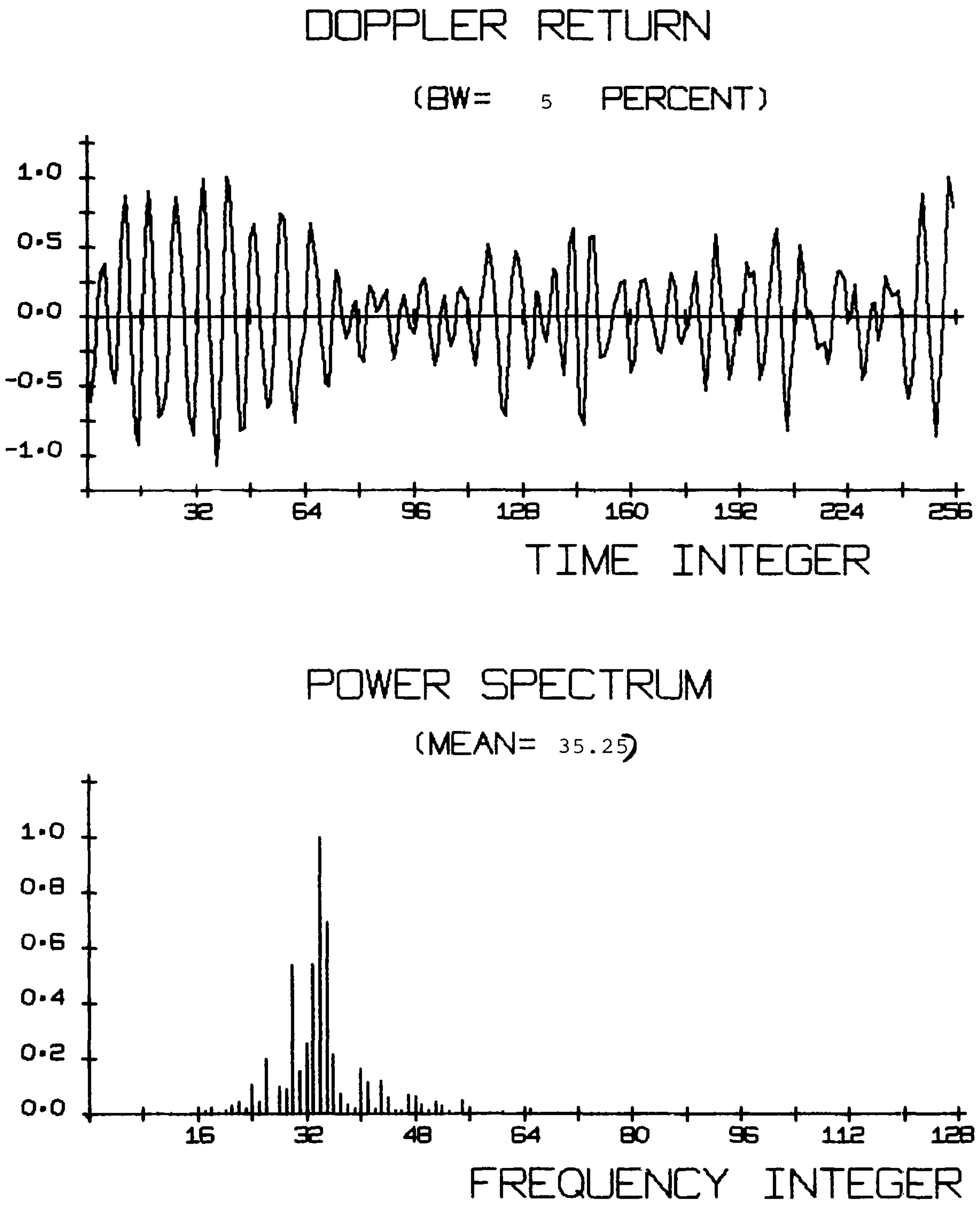


Fig. 15. Spectral analysis of Doppler return with BW=5% and 8 points per cycle at band-center.

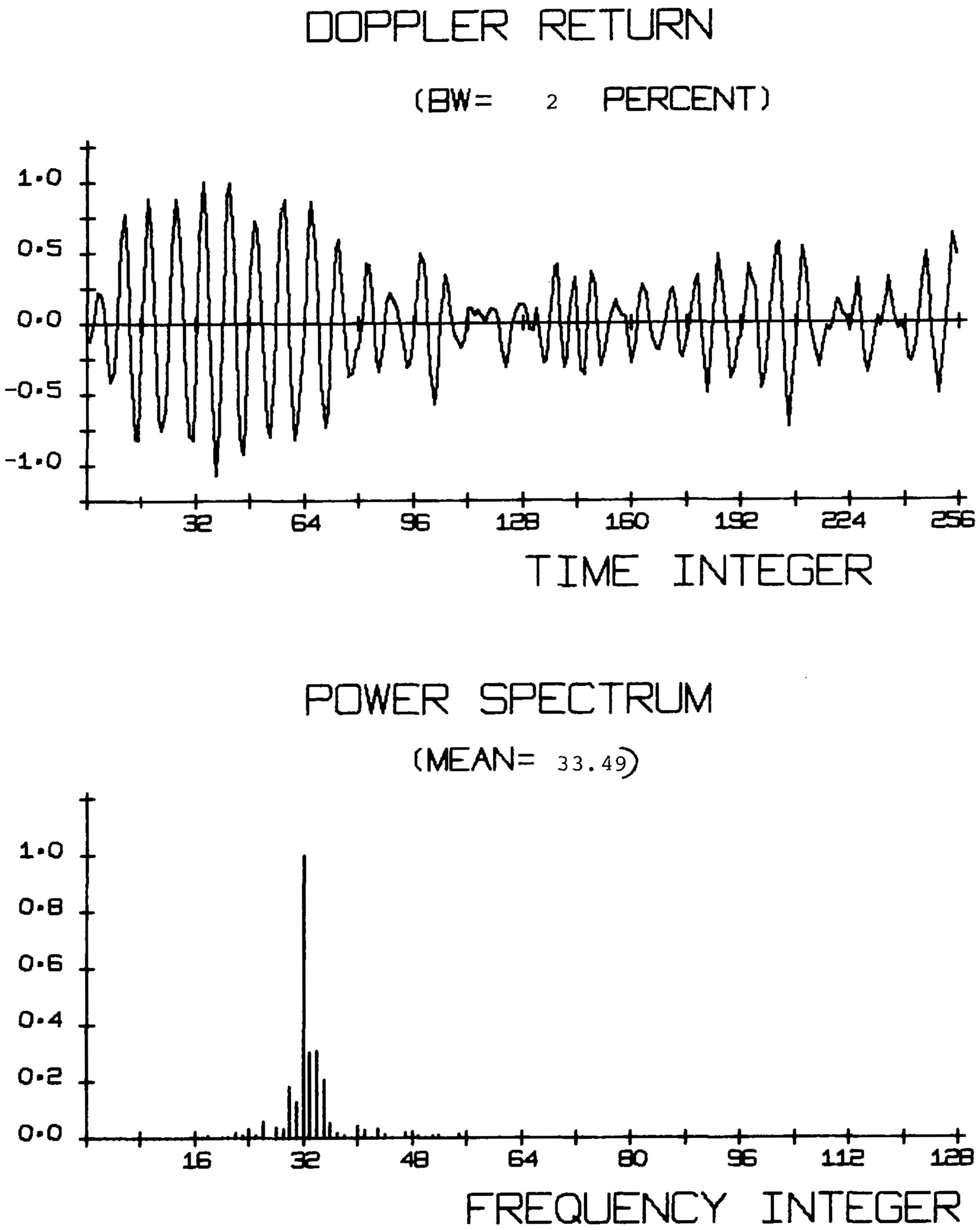


Fig. 16. Spectral analysis of Doppler return with BW=2% and 8 points per cycle at band-center.

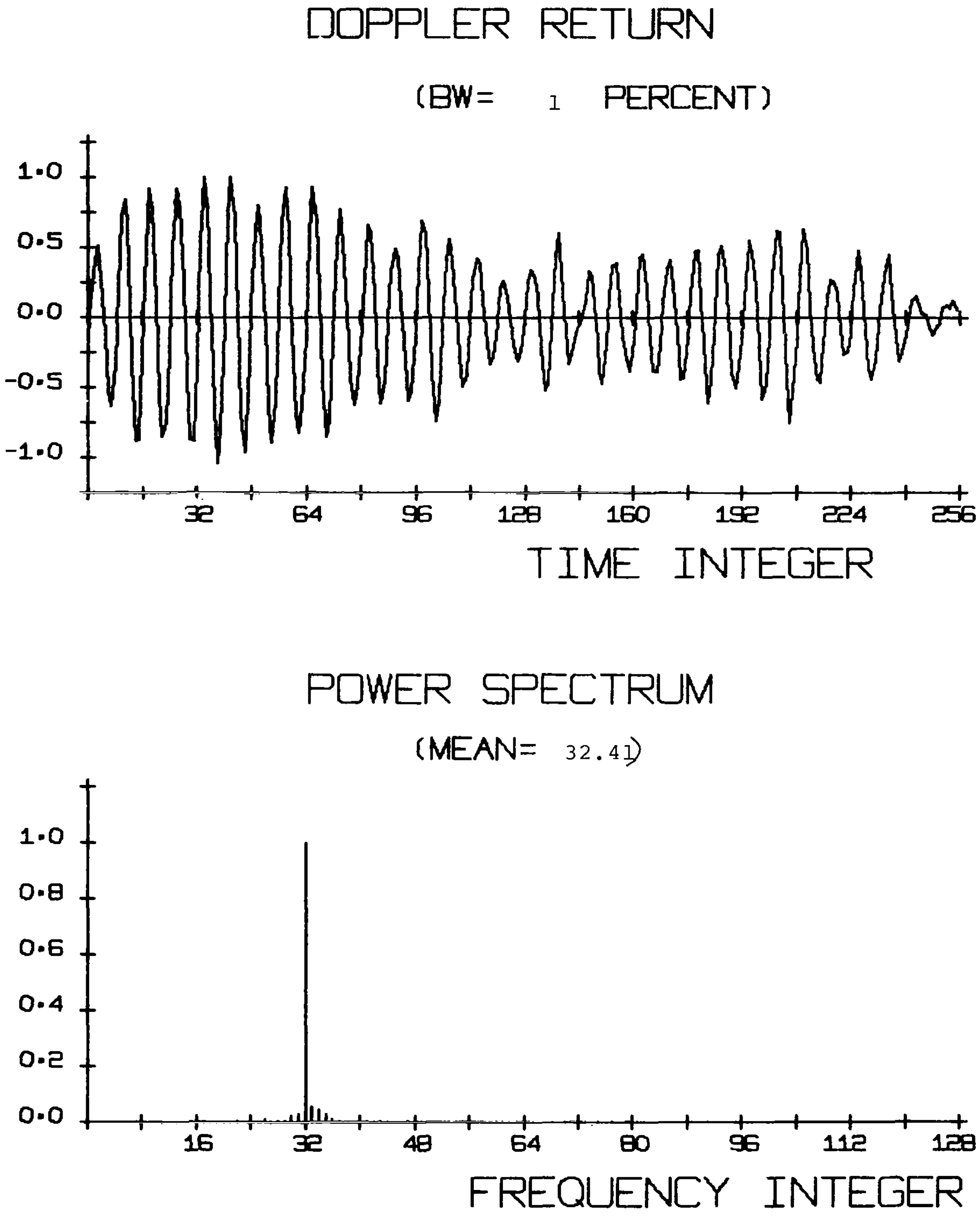


Fig. 17. Spectral analysis of Doppler return with BW=1% and 8 points per cycle at band-center.

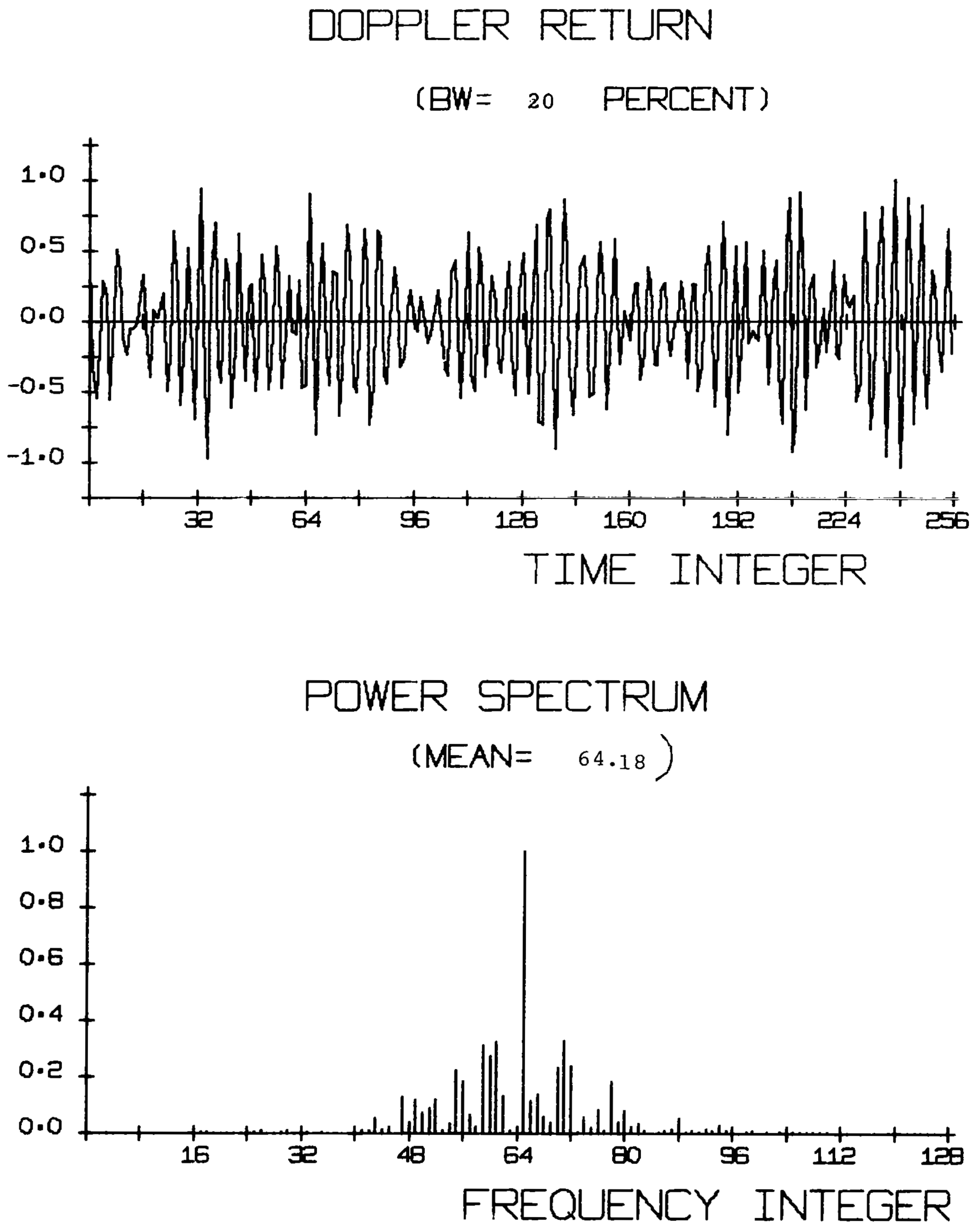


Fig. 18. Spectral analysis of Doppler return with BW=20% and 4 points per cycle at band-center.

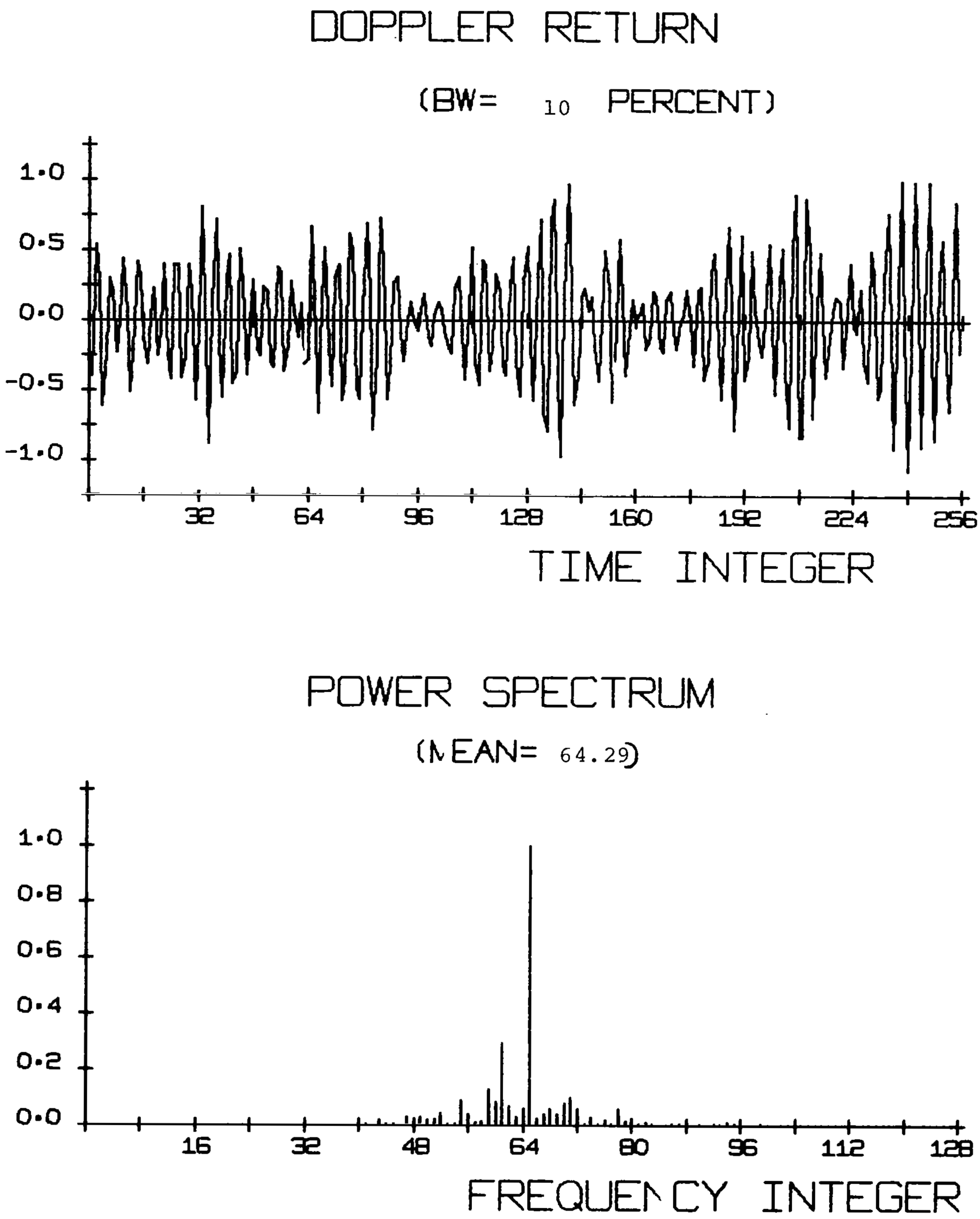


Fig. 19. Spectral analysis of Doppler return with BW=10% and 4 points per cycle at band-center.

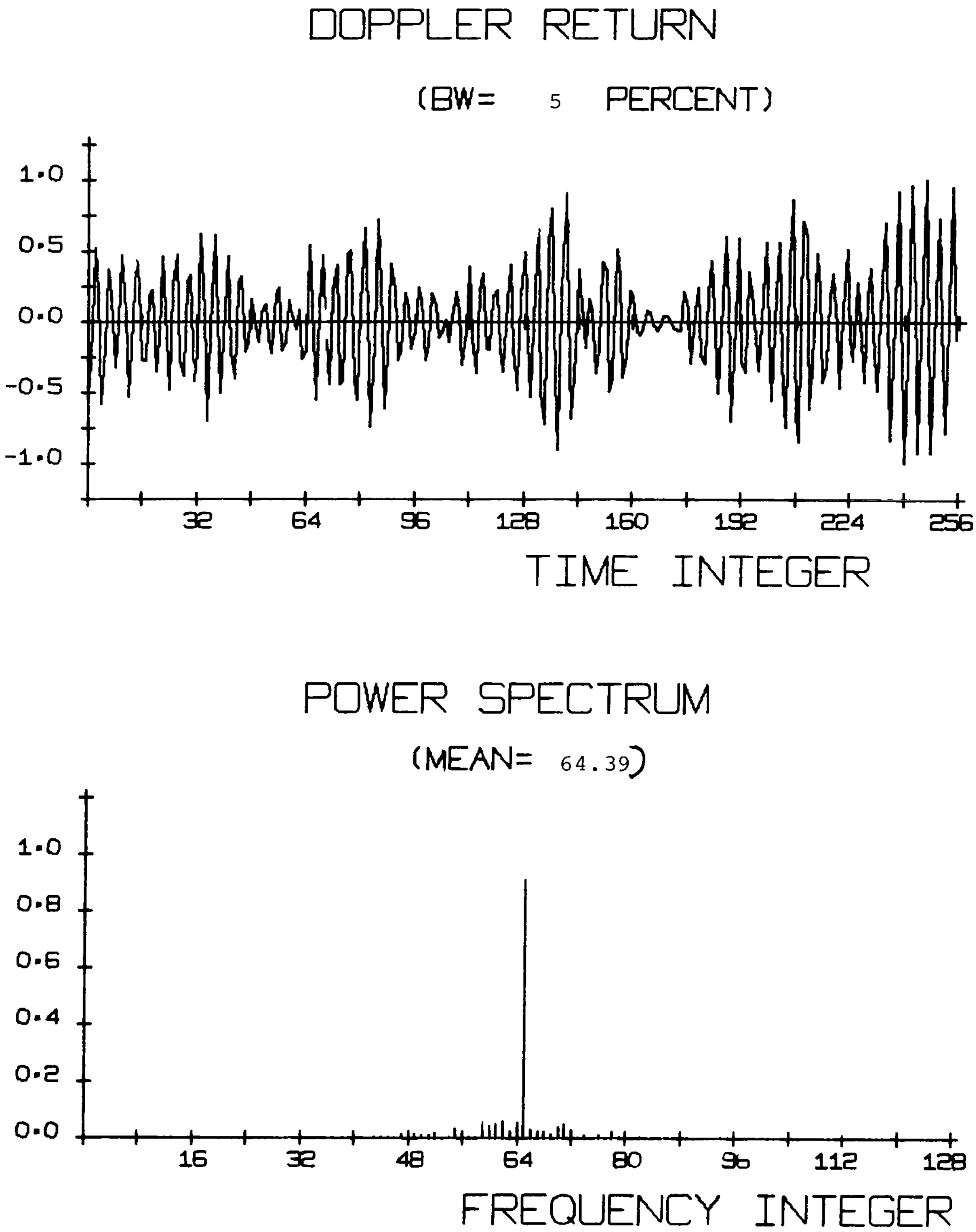


Fig. 20. Spectral analysis of Doppler return with BW=5% and 4 points per cycle at band-center.

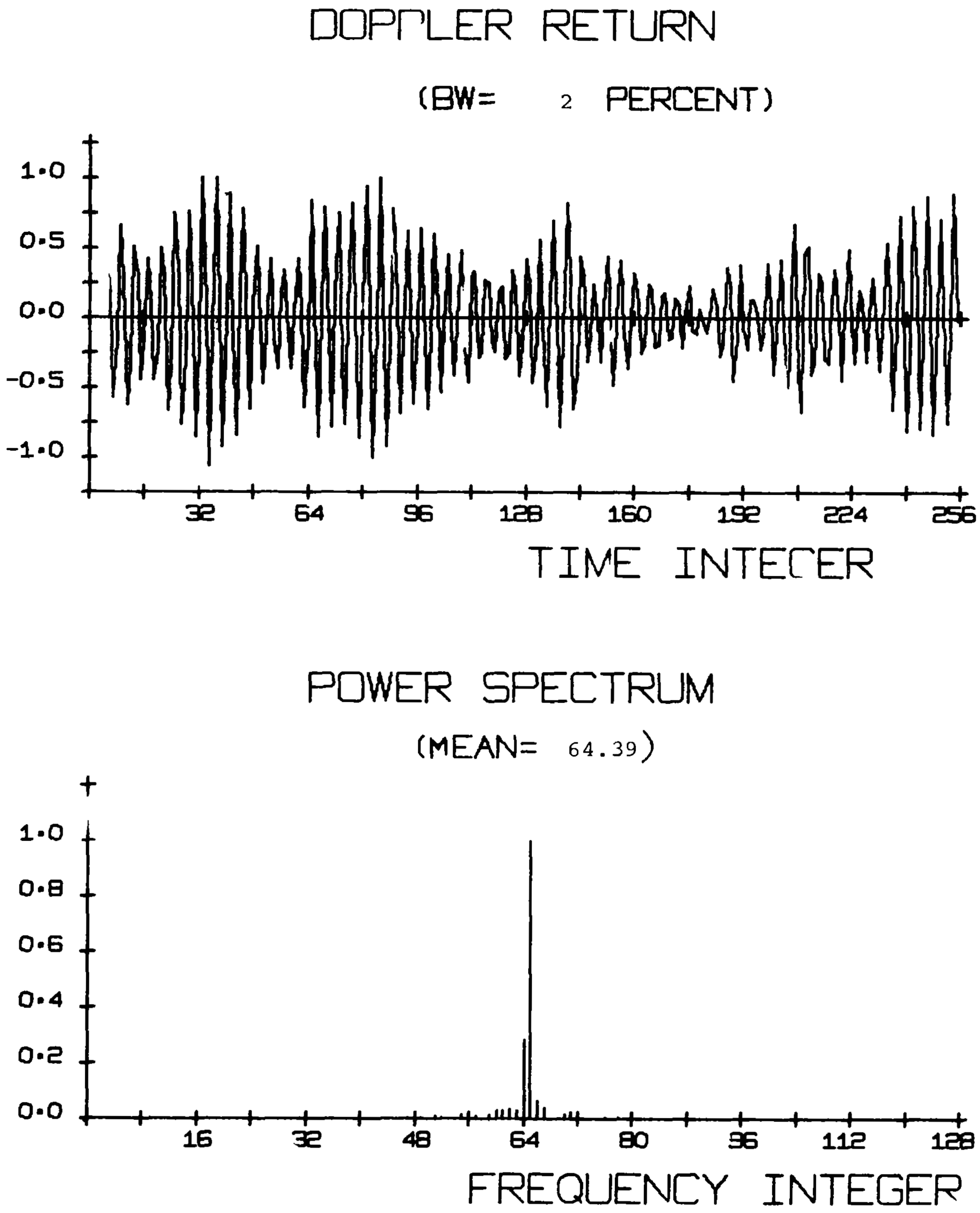


Fig. 21. Spectral analysis of Doppler return with BW=2% and 4 points per cycle at band-center.

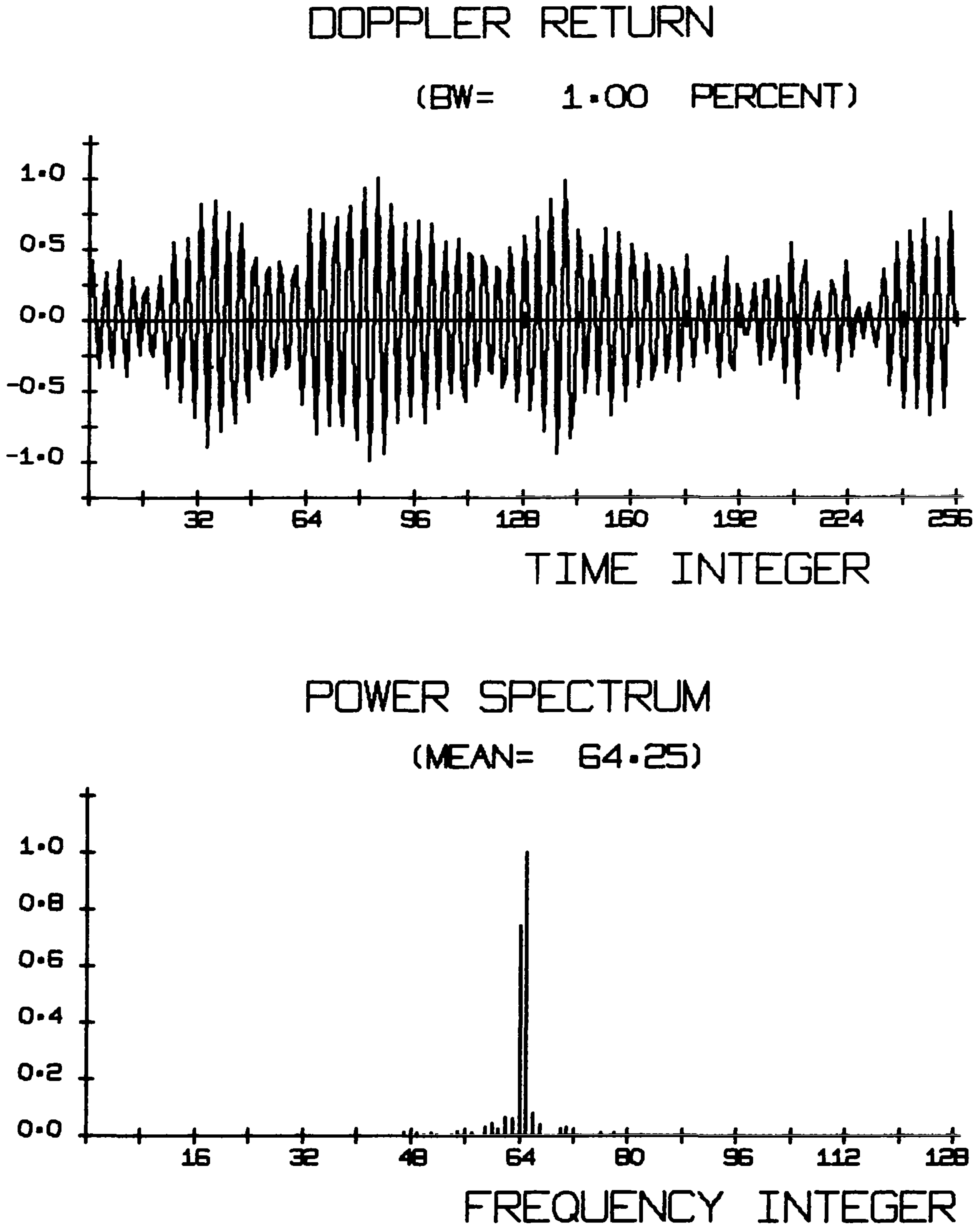


Fig. 22. Spectral analysis of Doppler return with BW=1% and 4 points per cycle at band-center.

NASA-Langley, 1971 —— 7 CR-1776