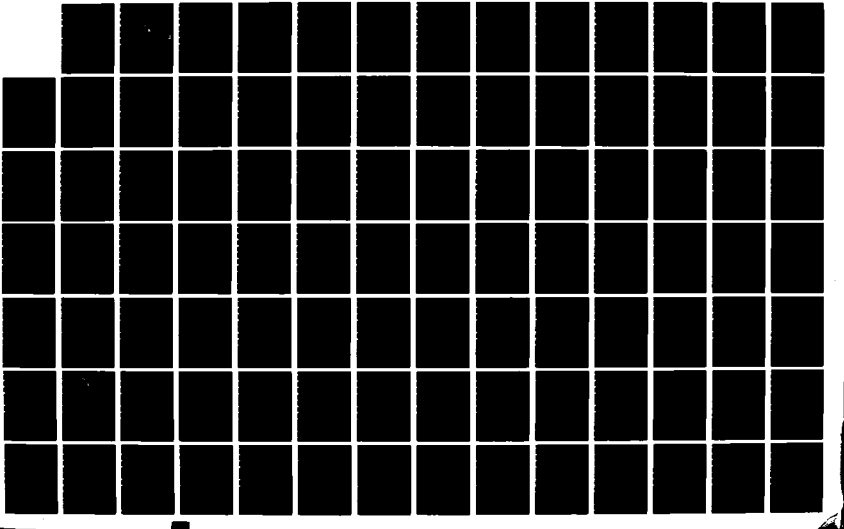
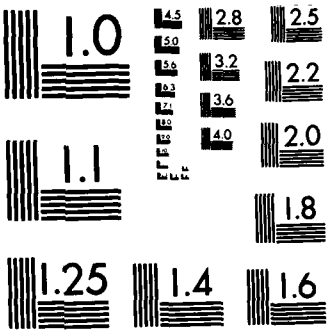
AD-A160 823 COMPUTER SIMULATION OF DIGITAL SIGNAL MODULATION TECHNIQUES IN SATELLITE COMMUNICATIONS(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA C D CARLSON SEP 85 1/4

UNCLASSIFIED F/G 9/2 NL





NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

COMPUTER SIMULATION OF
DIGITAL SIGNAL MODULATION TECHNIQUES IN
SATELLITE COMMUNICATIONS

by

Craig Dean Carlson

September 1985

Thesis Advisor: James L. Wayman

Approved for public release; distribution is unlimited

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A166 823	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Computer Simulation of Digital Signal Modulation Techniques in Satellite Communications		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis September 1985
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Craig D. Carlson		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943-5100		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943-5100		12. REPORT DATE September 1985
		13. NUMBER OF PAGES 291
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report)
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution is unlimited

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

18. SUPPLEMENTARY NOTES

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

Computer Simulation, Digital Signal, Modulation Techniques, Satellite Communications, Statistical Analysis

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

This thesis is a tutorial on digital signal modulation techniques used in satellite communications and includes computer simulation of those digital signal modulation techniques introduced. The purpose of the thesis is to introduce digital signal modulation techniques and through the use of computer simulation, generate statistics which represent the characteristics of the FFT for the respective signal type. Further, an analysis of the statistics

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

of the FFT's was conducted to determine if there is any relationship between the components of the FFT of the different signals. The statistic used to investigate this possible relationship was the F-distribution. The computer simulation was written and conducted in the FORTRAN programming language. A copy of the program, results of the simulation and the statistical analysis conducted are included in the appendices.

A

Approved for public release; distribution is unlimited.

Computer Simulation of Digital Signal Modulation Techniques in Satellite Communications

by

Craig D. Carlson
Lieutenant Commander, United States Navy
B.A., Concordia College, 1973

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS TECHNOLOGY
(SPACE SYSTEMS OPERATIONS)

from the

NAVAL POSTGRADUATE SCHOOL
September 1985

Author: Craig D. Carlson
Craig D. Carlson

Approved by: James L. Wayman
James L. Wayman, Thesis Advisor

Allen E. Fuhs
Allen E. Funs, Chairman,
Space Systems Academic Group

John N. Dyer
John N. Dyer,
Dean of Science and Engineering

Accession For	
NTIS GRA&I	X
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

ABSTRACT

This thesis is a tutorial on digital signal modulation techniques used in satellite communications and includes computer simulations of those digital signal modulation techniques introduced. The purpose of the thesis is to introduce digital signal modulation techniques and through the use of computer simulation, generate statistics which represent the characteristics of the FFT for the respective signal type. Further, an analysis of the statistics of the FFT's was conducted to determine if there is any relationship between the components of the FFT of the different signals. The statistic used to investigate this possible relationship was the F-distribution. The computer simulation was written and conducted in the FORTRAN programming language. A copy of the program, results of the simulations and the statistical analysis conducted are included in the appendices.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

A. BACKGROUND

Since the introduction of the sampling theorem and the matched filter, digital communications techniques have developed into a highly proficient discipline. The marriage of this discipline with the rapidly expanding space program has resulted in communication satellites employing a multitude of digital signal modulation techniques. A modulation technique is a method of transmitting the information contained in a message by varying or modulating the characteristics of a carrier waveform. These methods offer a range of advantages and disadvantages depending on the specific characteristics of the modulation technique employed. The applications of the various signal modulation techniques likewise vary. In order to understand this exciting new field it is necessary to look at some of the aspects of satellite communications systems in general. Then an investigation will be made into the general attributes of digital communications and their relationship to the satellite system.

The ability to understand these digital signal modulation techniques is the first step in being able to intercept, identify and demodulate unknown digital signals. These digital signals, transmitted from unknown sources, are believed to display frequency characteristics peculiar to the modulation technique employed in the encoding process. The digital computer offers unique opportunities in simulating these modulation techniques, in signal processing and in decoding of an intercepted signal.

B. SPECIFIC GOALS

It is the specific goal of this thesis to investigate the open literature on the topic of digital signal modulation techniques in satellite communications. This includes a basic understanding of the communications satellite system as well as the specific techniques employed in signal modulation. Additionally, once a basic understanding of these digital signal modulation techniques is achieved, computer code in the FORTRAN programming language will be developed which simulates these modulated signals. The statistics of the time-varying Fast Fourier Transforms (FFT) of these simulated signals will be investigated. The purpose of this analysis will be to lead to follow-on research in the area of signal analysis of intercepted digital signals whereby they can be classified by their FFT as employing a specific digital signal modulation technique.

C. SCOPE OF THE PROJECT

Although an indepth analysis of all the digital signal modulation techniques which will be introduced involves considerable higher level mathematics, it is not within the scope of this project to delve heavily into the mathematics. It would be advantageous, however, for the reader to have had integral and differential calculus and an introduction to statistics. Also a course in electrical engineering may prove helpful but is not essential since in actuality it is the electrical engineering aspects of satellite communications that this tutorial is attempting to present. The computer programming will be accomplished utilizing the techniques of software engineering and top down modular design. Existing blocks of code will be used as modules whenever appropriate and available. Again it is the

ultimate purpose of this project to examine a variety of digital signal modulation techniques and develop computer code that simulates common signals used in satellite communications that have been produced by one of the many digital signal modulation techniques to be investigated.

II. UNDERSTANDING SATELLITE COMMUNICATIONS

A. HISTORY AND APPLICATIONS

Forty years ago, in 1945, Arthur Clarke first envisioned the use of space stations placed in geosynchronous orbit for communicating to different points on the earth [Ref. 1]. Nine years later in 1954, J.R. Pierce of Bell Laboratories performed a system analysis on such a communications system [Ref. 2]. In 1957, the launch of Sputnik demonstrated the feasibility of using a satellite for just such an application. However, by 1961 the only satellite communications technologies which had been demonstrated were the Courier 1B satellite, a short-life active retransmission teletype communications satellite in a 1000 km orbit, and the Echo I balloon in a 1600 km orbit which demonstrated passive reflection of powerful microwave signals from one earth station to another. Active microwave communications demonstrated in Projects Telstar and Relay were years away [Ref. 3]. The first geostationary orbit was achieved by Project Syncom in 1963 [Ref. 4].

In 1964, communication organizations from several countries joined together to form the international organization of INTELSAT (International Telecommunications Satellite Organization). INTELSAT's purpose was to develop a satellite network which would provide truly global communications capabilities. This resulted in the launch in 1965 of the world's first commercial communications satellite, INTELSAT I, also known as "Early Bird". With "Early Bird", telecommunications utilizing satellite relay were established between the United States and Europe. [Ref. 3]

The reliability of these satellite systems has improved dramatically since INTELSAT I in 1965. Reliability of individual links in the system approach 99.99 percent or higher. Total system reliability exceeding 99.9 percent is common [Ref. 5], making satellite communications more reliable than many other modes of communications. In this sense, reliability is a measure of the probability that no failure will occur in a respective channel or in the system during the design life of the satellite.

In order for the operability, capability and reliability of these systems to have developed at this pace, it was necessary for the technologies associated with them to develop as rapidly or even more rapidly than the systems themselves. Engineering sciences and specifically the fields of aerospace and electrical engineering historically have required between 7 and 10 years to take a concept from operational requirement to full scale operation. This was not the case with the concept of communications satellites which has seen four generations of INTELSAT satellites within a decade's time. [Ref. 3]

The "Early Bird" satellite weighed approximately 38 kilograms and possessed limited power and bandwidth capacity enabling it to carry only 240 two-way telephone conversations. Today's communications satellites represent order-of-magnitude improvements in many important operating parameters such as power and bandwidth. Additionally, increased effective radiated power from the techniques of stabilized earth pointing antennas have greatly increased the capacity of later generation communications satellites. INTELSAT V, the current generation of communications satellites, is a three-axis stabilized platform using not only the 6/4 GHz frequency band (6 GHz receive/4 GHz transmit) as in earlier generation satellites but also a 500 MHz bandwidth available in the 14/11 GHz band. The

separation in the receive and transmit frequencies is necessary to prevent interference during simultaneous operation of the receiver and transmitter. Another factor increasing the capacity of present generation satellites is the increase in primary power available in the satellite. [Ref. 3]

The technologies which have contributed to the evolution of communications satellites come from two primary sources, namely technology from the space program of the 1960's supported by NASA and DoD and communications technology due largely to commercial and private sector contributions. [Ref. 3]

Electronic devices and components have contributed significantly to the rapid development of satellite systems. These electronic devices range from something which is now considered basic, i.e., the transistor, to devices such as Traveling Wave Tube amplifiers, lightweight antennas and antenna feed systems. There have been major improvements in satellite power sources including more efficient solar cells and high storage capacity/lightweight batteries. Additionally, two significant developments in electrical engineering have made modern day digital signal processing a reality. They are the sampling theorem and the matched filter. [Ref. 3]

Communications satellites of the future are likely to utilize onboard signal processing. Signal processing functions such as signal reshaping, switching and/or multiplexing and compression could soon take place on the satellite due to the reduction in size and weight of the necessary hardware. Operation at over 100 megabits per second are envisioned. Onboard signal processing will greatly reduce the expense and complexity of earth stations thereby making services of a satellite available to a wider range of small and geographically disperse users. [Ref. 3]

Satellite communications hold great promise to provide service to a multitude of users over a wide area. Applications lie not solely in retransmission but also in data collection from that same large geographic area. Military users have definite applications in these areas when warning, intelligence and surveillance systems provide digital inputs to a central source. Although most of the present applications for satellite communications are still provided by the government, commercial applications have seen tremendous increases in the last several years. [Ref. 3]

B. ORBITS AND LIMITATIONS

Satellite orbits are generally categorized as either equatorial (0 degrees inclination), polar (90 degrees inclination) or inclined at some angle other than 0 or 90 degrees relative to the spin axis of the earth as illustrated in Figure 2.1 [Ref. 6]. Each satellite orbit has a characteristic velocity which is dependent on the height of the orbit and the orbit's eccentricity. Eccentricity is a measure of the degree to which the orbit approximates a circle. A circle has eccentricity equal to zero since the focus is located at the center. See equation 2.1 and Figure 2.2.

A communications satellite in elliptical orbit about the earth obeys Kepler's second law of planetary motion. That is, a satellite's constant angular momentum about the earth means that its areal velocity also remains constant. See Figure 2.3. Of particular interest is the satellite velocity at apogee (Va) and perigee (Vp) given by equations 2.2 and 2.3.

Note that for a circle, e = 0 and Ra = Rp. Therefore a satellite in circular orbit about the earth has an orbital velocity as given in equation 2.4.

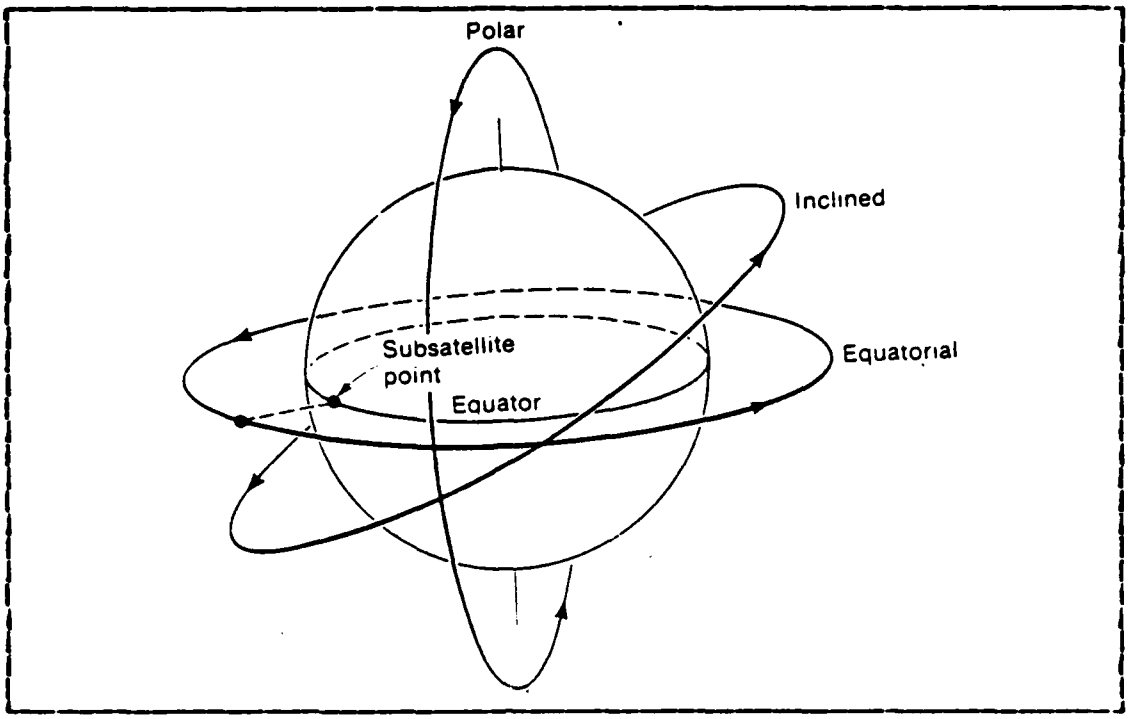


Figure 2.1 Satellite Orbits

Most communications satellites are placed in a circular equatorial orbit where the desire is to stabilize them over a fixed point on the surface of the earth called the subsatellite point. This type of orbit is called geostationary or stationary. Any satellite which has an orbital period equal to the period of rotation of the earth is called synchronous or geosynchronous. These terms are used almost interchangeably in most literature. As mentioned for a geosynchronous communications satellite, the orbital period of the satellite, T, must be equal to the period of rotation of the earth. Kepler's third law of planetary motion can be rewritten to yield equation 2.5.

The period of revolution of the earth for a sidereal day is 23 hr 56 min 4 sec. For that given period there is only one satellite altitude as expressed by Kepler's third law.

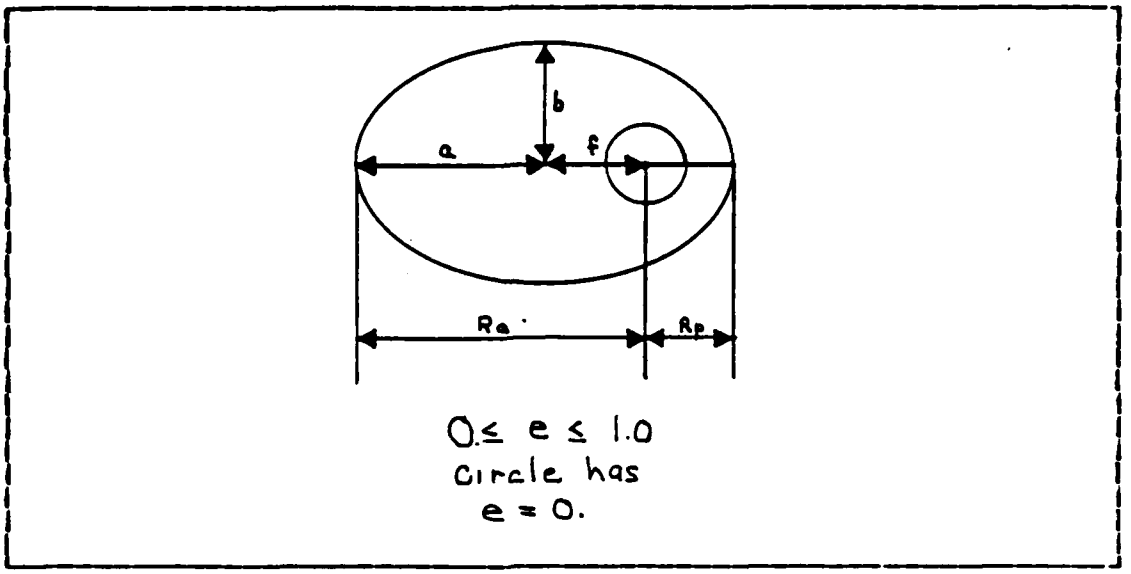


Figure 2.2 Eccentricity

eccentricity (e) = c/a (eqn 2.1)

c = distance from focus
a = semi-major axis

b = semi-minor axis
Ra = radius cf apogee
Rp = radius cf perigee

By rearranging terms that altitude is given in equation 2.6. Since the orbit is circular, a = Ra = Rp and the height of the orbit above the surface of the earth is h = a - Re; or 35,804 km.

One disadvantage of a geosynchronous communications satellite is the lack of global coverage. These satellites provide excellent coverage of the subsatellite point and

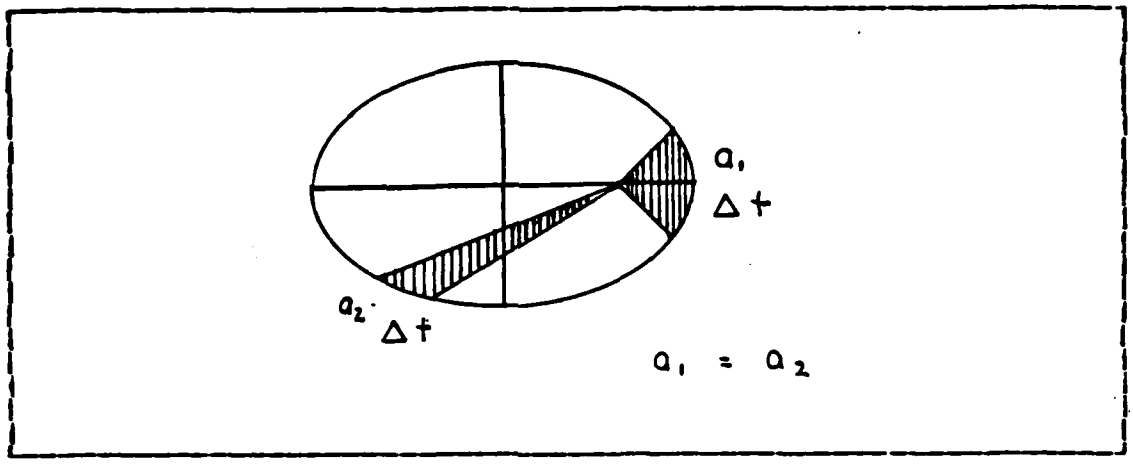


Figure 2.3 Kepler's Second Law of Planetary Motion

Va = (k/Ra(1-e))**.5 (eqn 2.2)

Vp = (k/Rp(1+e))**.5 (eqn 2.3)

k = GMe = 3.98866 x 10**-11 m^3/s^2
G = universal gravitational constant
G = 6.67 x 10**-11 N m^2/kg^2
N = newtons = m kg/s^2
Me = mass of the earth = 5.98 x 10**24 kg

Vc = (k/Rc)**.5 (eqn 2.4)

Rc = Re + h
Re = radius of the earth = 6.37 x 10**6 m
h = satellite orbital altitude

T = 2 pi a**1.5 / k**.5 (eqn 2.5)

T = period of rotation
a = semi-major axis

$$a = (T / 2\ pi)^{2/3}\ (k)^{1/3} \qquad \text{(eqn 2.6)}$$

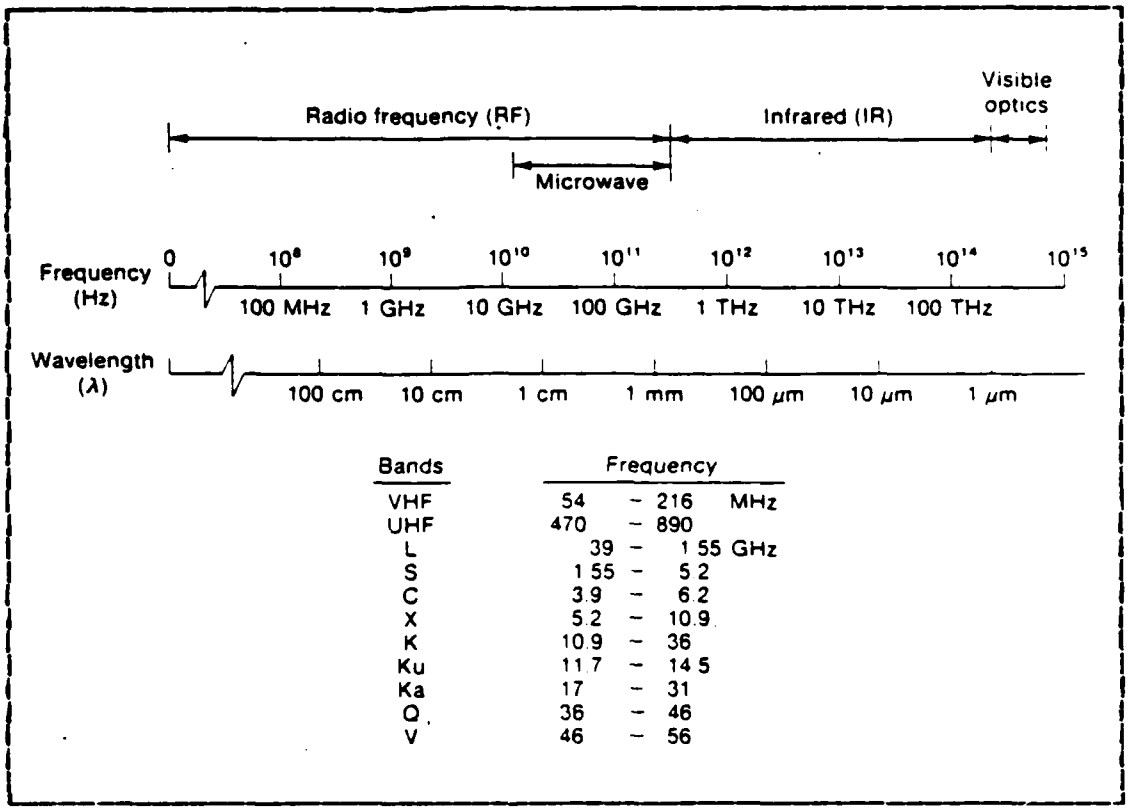
$$a = 42{,}173\ km$$

laterally to a latitude of about ±80° [Ref. 6]. This is generally no problem for commercial applications since the polar regions do not require a significant degree of access. The military, on the other hand, does have an interest in communications in the polar region and therefore has several communications satellites that have orbits inclined at various angles. This type of orbit generally has disadvantages of lack of continuous coverage and a much more complicated system of ground tracking and receiving stations.

C. FREQUENCY BAND CONSIDERATIONS

Although there appears to be an infinite number of frequencies available for communications, restrictions do exist as to those which are practicable. Limitations on available frequency bands for satellite communications are due to the need to select segments of the electromagnetic spectrum which reduce noise and interference and are most favorable in terms of power efficiency and propagation distortions. Trade offs must be made to arrive at the optimum frequency for a particular application since single frequencies seldom offer the best performance for all variables. The problems arise when consideration is given to the number of users requiring the same frequency bands including terrestrial communications networks. Since the problem of interference is global, a worldwide organization has been established to assign frequency bands for various applications. This organization is called WARC, World Administrative Radio Conference [Ref. 6]. Illustrations of

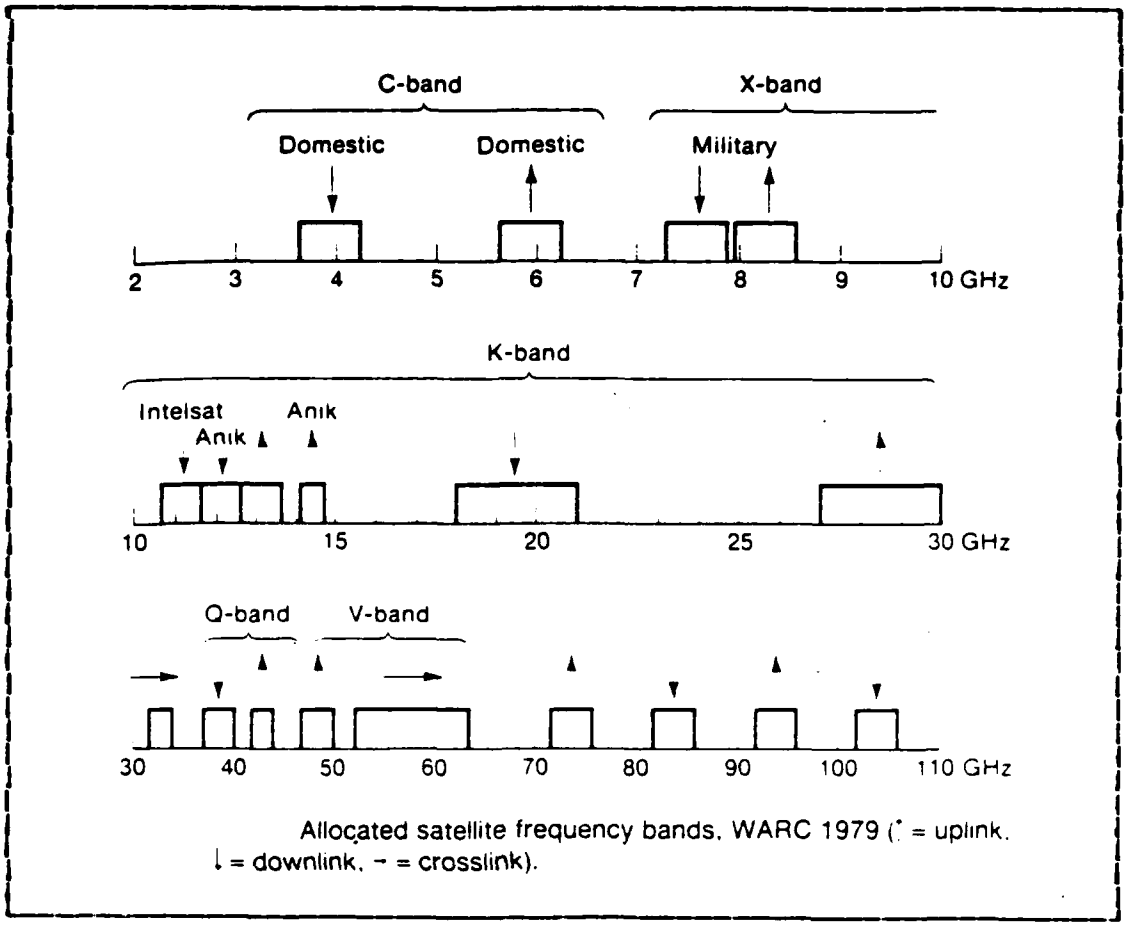
the portions of the electromagnetic spectrum in question and current allocations of satellite frequency bands are shown in Figures 2.4 and 2.5 [Ref. 6].



Bands	Frequency
VHF	54 – 216 MHz
UHF	470 – 890
L	39 – 1 55 GHz
S	1 55 – 5 2
C	3.9 – 6.2
X	5.2 – 10.9
K	10.9 – 36
Ku	11.7 – 14 5
Ka	17 – 31
O	36 – 46
V	46 – 56

Figure 2.4 Electromagnetic Spectrum

In general, as long as the hardware and technology will support it, higher frequencies are more desirable and more in demand because they offer higher theoretical capacity. This is due to the fact that only a percentage of the carrier frequency is capable of actually transmitting a signal of a given bandwidth. Higher frequencies would also experience less interference with existing land and satellite systems. A further discussion of the bandwidths available as a function of frequency will be included at a



Allocated satellite frequency bands. WARC 1979 (↑ = uplink, ↓ = downlink, → = crosslink).

Figure 2.5 Satellite Frequencies

later point under the topic of actual digital signal techniques.

As an additional note, in order to assist in the ability to handle more signals in an obviously limited electromagnetic spectrum, several multiple access schemes have been developed. The most common techniques involve frequency-division multiple access (FDMA) where the allocated satellite frequency band is divided among the users into specific uplink and downlink frequencies. Next is time-division multiple access (TDMA) where the satellite frequency band is shared by all users by carefully dividing

the time any one user has access. Finally there is code-division multiple access (CDMA) which involves modulating a specific address code which has been superimposed on the signal directly onto the carrier. In this manner all users share the satellite frequency band and only those receiving stations that can demodulate the address code can receive the specific signal. [Ref. 6]

D. WHY DIGITAL MODULATION

In general, digital signal modulation techniques are superior to analog signal modulation techniques used in satellite communications for the following reasons:

1. Compatibility with digital computers
2. Economic advantages
3. High degree of flexibility
4. Less susceptible to interference
5. Quality of signal independent of transmission distance and network makeup

1. Compatibility with Digital Computers

Clearly one of the most distinctive advantages of digital techniques over analog techniques involves computer applications. Properly formatted digital signals can be used to represent any analog signal (more on this under the discussion of the sampling theorem). Once in the digital format these signals can be easily manipulated within the digital computer. Arithmetic operations can be applied as well as logical operations. The signal can be stored without alteration or delayed and therefore can be used to "simulate" real physical situations.

The hardware associated with digital circuits is free from drift or aging and does not require calibration. Additionally digital circuitry is compatible with present day integrated circuit technology allowing a standardized

building block construction approach. The operating characteristics of these systems employing a digital computer can be changed by altering the software rather than the hardware as is the case for analog systems. Finally, the use of digital computer technology allows time multiplexing.

2. Flexibility and Economy

The flexibility and economy of digital satellite communications comes from the fact that more and more processing can be done onboard. This allows the uplink and downlink to be completely separated. This regenerative nature makes it possible for low error rates and high reliability through the use of digital techniques not available to analog systems. Because digital signal processing or multiplexing is less costly than for analog signals, simpler and cheaper interfaces between earth stations and terrestrial communications networks are possible. Additionally, there are reduced production costs and increased capacity associated with digital circuits. [Ref. 7]

3. Quality and Interference

The capability of digital systems to regenerate the signal and allow for multiple switching and signal processing without degradation in signal quality makes the digital signal basically independent of transmission distance. Multiple hops from satellite to earth station or satellite to satellite are possible without accumulation of the noise characteristic of analog systems. Additionally, digital systems are capable of operating at a signal to noise ratio of 20 dB to 30 dB as compared to analog systems requiring a much more powerful signal. [Ref. 8]

III. INTRODUCTION TO THE FOURIER TRANSFORM, SAMPLING THEOREM, AUTOCORRELATION FUNCTION AND THE MATCHED FILTER

Essential to the understanding of digital signal modulation techniques are a few basic tools of the electrical engineer and the mathematician. These tools will be developed and elaborated on to the extent necessary to understand their applicability to the subject of digital communications. The description is not meant to be a detailed investigation of the respective topics. Where relevant, the application of the concept being described will be mentioned.

A. FOURIER TRANSFORM

In mathematical terms, voltages can be expressed as functions of time or of frequency. It is more common to see voltages represented as a function of time as in equation 3.1.

$$v(t) = A \cos(wt) \qquad \text{(eqn 3.1)}$$

A = amplitude
w = angular frequency = 2 pi f
f = frequency = 1/T
T = period

It is important to note that both the time and frequency functions are representations of voltage and as such may be used interchangeably. The Fourier representation $[v(t)] = V(f)$ is a voltage descriptor in the frequency domain (a

function of frequency) while v(t) is a voltage descriptor in the time domain (a function of time). They are different but equivalent and either voltage descriptor may be used, depending on the concept being explored, to best represent the voltage within context.

The Fourier transform is of principle interest rather than the Fourier series since the latter is applicable only to periodic voltages. The Fourier transform is applicable to voltage pulses, random voltages and other non-periodic voltages.

DEFINITION:

$$\mathcal{F}[\,v(t)\,] = V(w) = \int_{-\infty}^{\infty} v(t)\,e{*}{*}{-jwt}\;dt \qquad \text{(eqn 3.2)}$$

v(t) <-------------> V(w)

time domain frequency domain

Remembering Euler's formula

$$e{*}{*}{-jwt} = \cos(wt) - j\,\sin(wt) \qquad \text{(eqn 3.3)}$$

the Fourier transform becomes

$$\mathcal{F}[\,v(t)\,] = V(w) = \int_{-\infty}^{\infty} v(t)[\cos(wt) - j\,\sin(wt)\,] \qquad \text{(eqn 3.4)}$$

$$V(w) = \int_{-\infty}^{\infty} v(t)\cos(wt) - j\int_{-\infty}^{\infty} v(t)\sin(wt) \qquad \text{(eqn 3.5)}$$

Since it will be seen that digital communications deals primarily with pulses of finite duration (expressed as period, T), it is worthwhile to examine the Fourier transform of a pulse of amplitude A and duration T.

$$\text{let } v(t) = \begin{cases} A; & -T/2 < t < T/2 \\ 0; & \text{elsewhere} \end{cases}$$

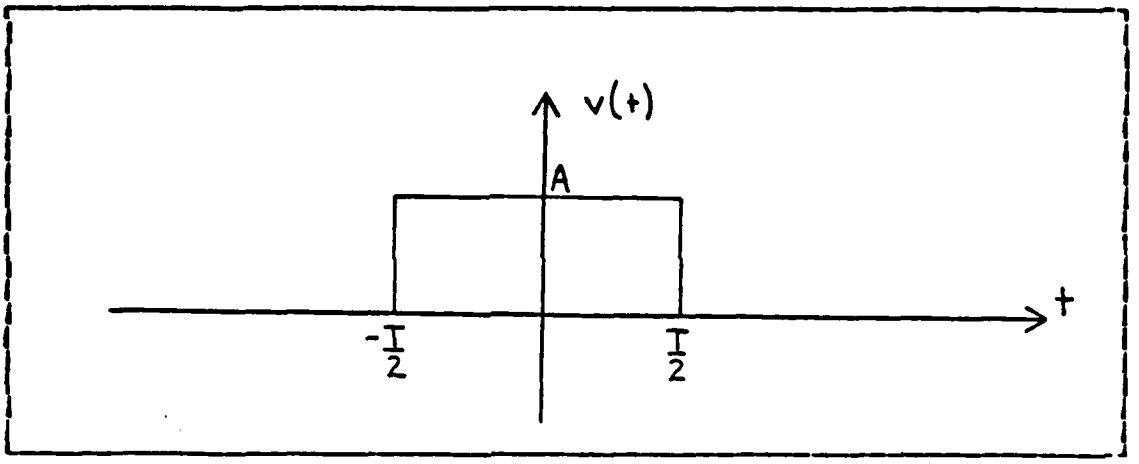


Figure 3.1 Square Pulse

Figure 3.1 is a representation of v(t) or the voltage expressed in the time domain. The position of v(t) on the t-axis was chosen for convenience of integration but could have been situated anywhere on the time line.

$$\mathcal{F}[v(t)] = V(w) = \int_{-\infty}^{\infty} v(t)\, e^{**-jwt}\, dt \qquad \text{(eqn 3.6)}$$

$$V(w) = \int_{-\infty}^{-T/2} e^{**-jwt}\, dt + \int_{-T/2}^{T/2} A\, e^{**-jwt}\, dt + \int_{T/2}^{\infty} 0 * e^{**-jwt}\, dt \qquad \text{(eqn 3.7)}$$

$$V(w) = \int_{-T/2}^{T/2} A\, e^{**-jwt}\, dt \qquad \text{(eqn 3.8)}$$

The integration above may be attacked either head on or by substituting cos(wt) - j sin(wt) for e**-jwt. The direct approach is illustrated due to the relative simplicity of the integrand.

$$V(w) = \int_{-T/2}^{T/2} A\, e^{**-jwt}\, dt \qquad \text{(eqn 3.9)}$$

$$V(w) = -A/jw[e^{**-jwt}]\Big|_{-T/2}^{T/2} \qquad \text{(eqn 3.10)}$$

$$V(w) = -A/jw[e^{-jwT/2} - e^{jwT/2}] \quad \text{(eqn 3.11)}$$

$$V(w) = A/jw[e^{jwT/2} - e^{-jwT/2}] \quad \text{(eqn 3.12)}$$

By substituting 2pi f = w, the following result is obtained:

$$V(f) = A/j2\ pi\ f[e^{j2\ pi\ fT/2} - e^{-j2\ pi\ fT/2}] \quad \text{(eqn 3.13)}$$

$$V(f) = A/j2\ pi\ f[e^{j\ pi\ fT} - e^{-j\ pi\ fT}]$$

Using Euler's formula, i.e., $\sin\theta = (e^{j\theta} - e^{-j\theta})/2j$:

$$V(f) = 2jA/j2\ pi\ f[(e^{j\ pi\ fT} - e^{-j\ pi\ fT})/2j] \quad \text{(eqn 3.14)}$$

$$V(f) = A/pi\ f[\sin(pi\ fT)]$$

Knowing that $\sin x / x = \text{sinc}\ x$

$$V(f) = AT/pi\ fT[\sin(pi\ fT)] \quad \text{(eqn 3.15)}$$

$$V(f) = AT\ \text{sinc}(pi\ fT)$$

The sinc function is common in digital electronics and plots as the product of sin(pi fT) and 1/(pi fT) as in Figure 3.2. In Figure 3.2, V(f) in the frequency domain is equivalent to v(t) in the time domain. Note that as the pulse, T, gets longer, 1/T gets smaller or the first zero crossing of the sinc function occurs at a lower and lower frequency.

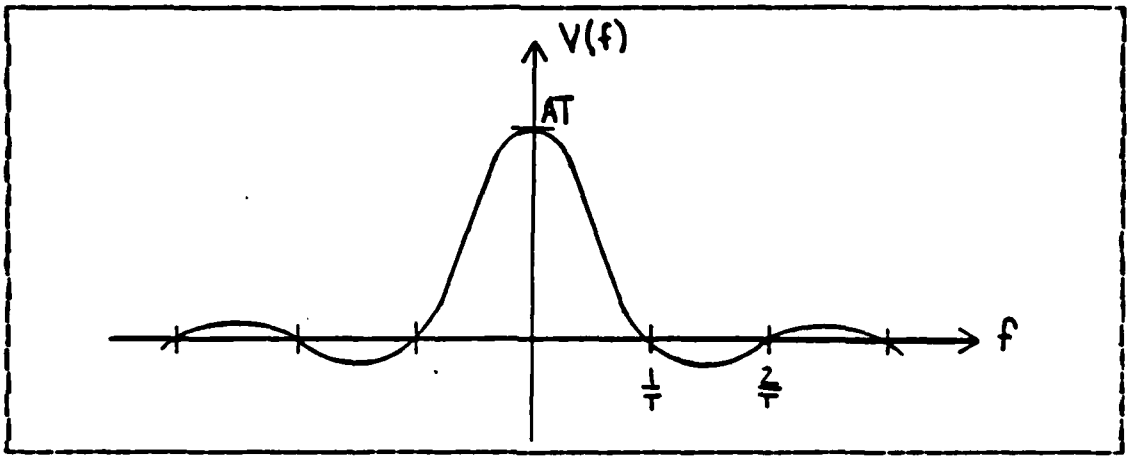


Figure 3.2 Plot of Sinc Function

1. Amplitude and Phase Spectrum

Recall that V(w) can be represented as in equation 3.16. The cosine term is the real part while the sine term is the imaginary part. By referencing Figure 3.3, a brief review of the complex plane is accomplished and its relationship to the amplitude and phase spectrum of a given voltage is represented. See equations 3.17 through 3.20.

$$V(w) = \underbrace{\int_{-\infty}^{\infty} v(t)\cos(wt)}_{\text{real}} - \underbrace{j\int_{-\infty}^{\infty} v(t)\sin(wt)}_{\text{imaginary}} \qquad \text{(eqn 3.16)}$$

|V(f)| is called the amplitude spectrum of the given voltage. The amplitude spectrum can also be calculated by using the complex conjugate of the Fourier transform and is always positive as shown in equation 3.21. The phase spectrum, θ, of the function in question is represented by the arctan[imaginary/real] and is illustrated in Figure 3.5.

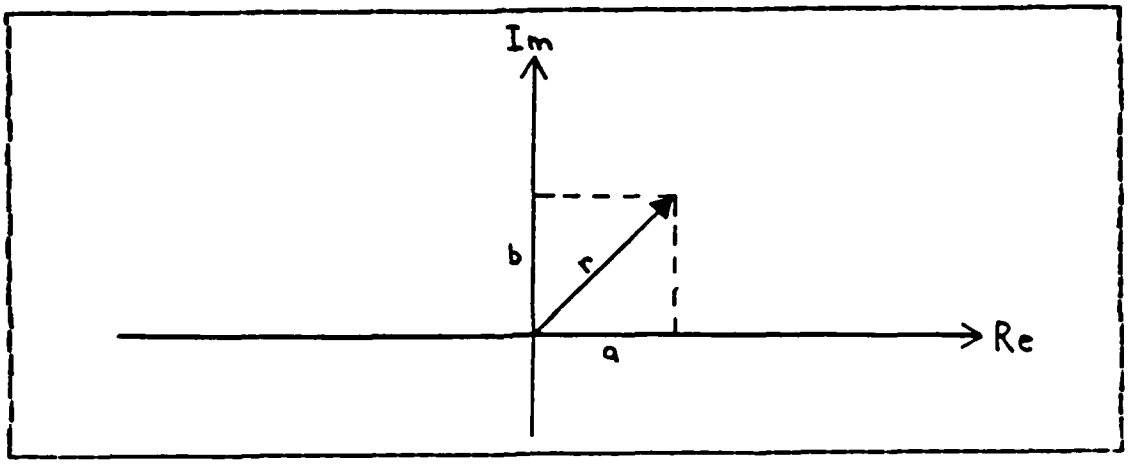


Figure 3.3 Phasor Diagram

$$r = [a^2 + b^2] \qquad \text{(eqn 3.17)}$$

$$a + jb = [a^2 + b^2]^{**}.5\ e^{**}j\theta \qquad \text{(eqn 3.18)}$$

$$V(f) = [\text{real}^2 + \text{imaginary}^2]^{**}.5\ e^{**}j\theta \qquad \text{(eqn 3.19)}$$

$$|V(f)| = [\text{real}^2 + \text{imaginary}^2]^{**}.5 \qquad \text{(eqn 3.20)}$$

since $|e^{**}j\theta| = 1$

$$|V(f)| = [V(f) \cdot V^*(f)]^{**}.5 \qquad \text{(eqn 3.21)}$$

2. Properties of the Fourier Transform

Several properties of the Fourier transform are useful in the study of digital signals. They are represented here without proof and without a great deal of detail.

a. Linearity

if $v1(t) \leftrightarrow V1(f)$ and

if $v2(t) \leftrightarrow V2(f)$ then

$\mathcal{F}[av1(t) + bv2(t)] = aV1(f) + bV2(f)$

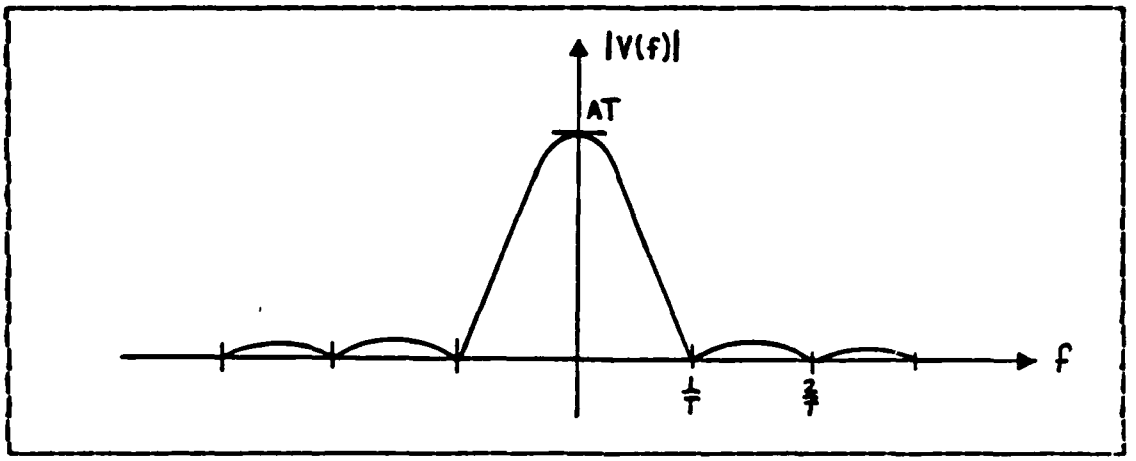


Figure 3.4 Amplitude Spectrum of Square Wave

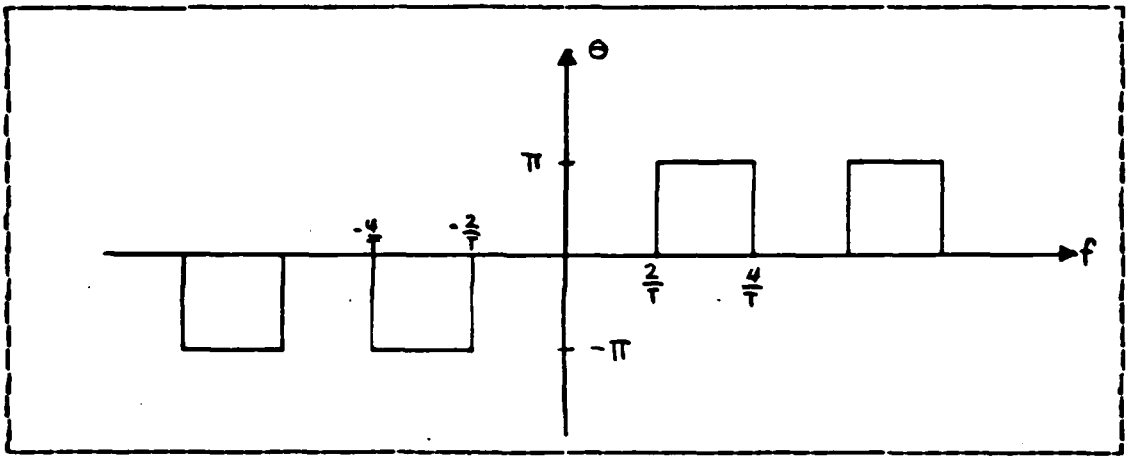


Figure 3.5 Phase Spectrum of Square Wave

b. Time-delay

$\mathcal{F}[v(t-to)] = V(f)\ e^{-jwto}$

Note that the amplitude spectrum of the delayed versicn is the same as the amplitude spectrum of the undelayed version. It is comforting to note that the Fourier transform of a pulse is the same tomorrow as it is today.

c. Scale change

$\mathcal{F}[v(at)] = 1/|a|\ V(f/a)$

d. Frequency translation

v(t)cos(2 pi fc t) <-> 1/2[V(f+fc) + V(f-fc)]

v(t) is any voltage

cos(2 pi fc t) is a carrier wave of frequency fc

Since understanding of this very important property of the Fourier transform is essential to the understanding of digital signal modulation it is expanded slightly here.

As we have seen, the Fourier representation of a voltage pulse of amplitude A and duration T is the sinc function of amplitude AT as illustrated in Figure 3.6.

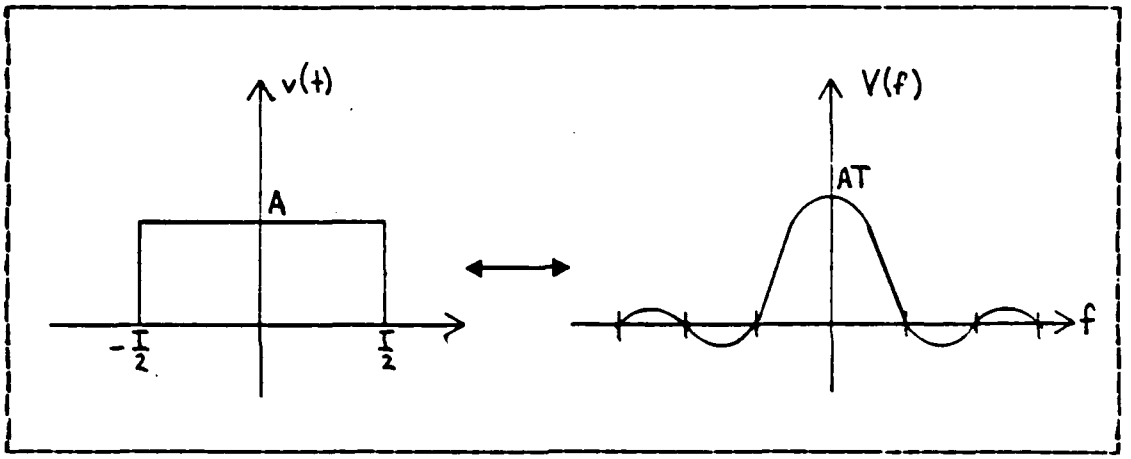


Figure 3.6 Square Wave in Time/Sinc Function in Frequency

The translation is the product of v(t) and in this case cos(2 pi fc t). In the time domain this product only exists in the interval between -T/2 and T/2 as in Figure 3.7. The significance of this translation and its relationship to the bandwidth of the voltage will be discussed further in the section dealing with bandwidth.

e. Differentiation

d v(t)/dt <-> jw V(f)

A differentiator could be used as a clock for timing but would never be used in the presence of noise since the

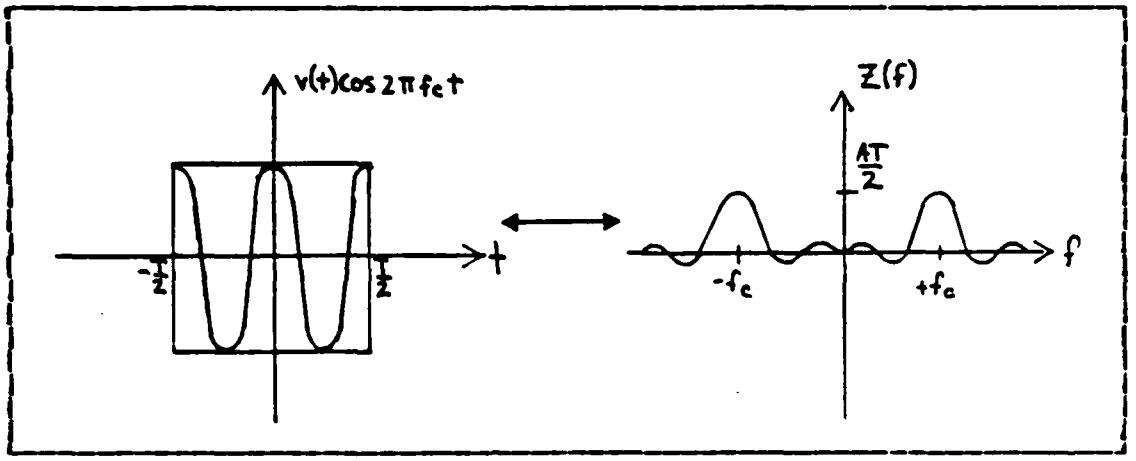


Figure 3.7 Multiplication and Translation Diagrams

result is exaggerated in the presence of high frequencies because w = 2 pi f.

f. Integration

v(t) dt <-> 1/w V(f)

An integrator could be used to reduce the effects of noise since w = 2 pi f is in the denominator tending to deemphasize the presence of high frequency noise.

B. THE SAMPLING THEOREM

Essential to the understanding of digital communications is the sampling theorem which was first introduced by Nyquist in 1928 [Ref. 9], and later by Shannon in 1948 [Ref. 10]. The sampling theorem states that any voltage can be uniquely represented by appropriately spaced sample values of the original voltage. More correctly stated, the sampling theorem places limits on the accuracy with which a signal can be represented.

The implication is that an analog signal can be represented digitally or by a set of numbers, i.e., samples. A description of the sampling theorem follows.

Given any analog voltage, v(t) as in Figure 3.8, the sampling theorem says that the entire analog signal is not required to accurately represent the voltage but only samples of it, call them vs(t). Figure 3.9 shows samples of a representative analog voltage. Samples can be taken of v(t) at every T seconds for a period of seconds. This can be accomplished by the use of a voltage clock, call it vc(t). The sampling can be viewed graphically as a block diagram representing an analog voltage multiplier as shown in Figure 3.10. To be of further use in the understanding of digital communications we are interested in a frequency description of the sample voltage, vs(t). Note that the system which describes the obtaining of vs(t) involves a voltage multiplication. It was demonstrated in the proceeding section that voltage multiplication amounted to frequency translation, a property of the Fourier transform.

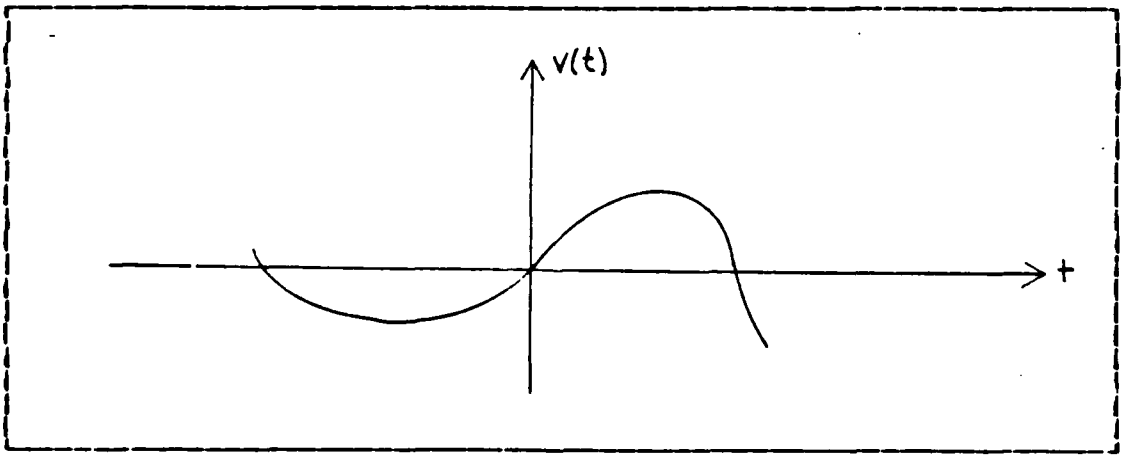


Figure 3.8 Representative Analog Voltage

First it is necessary to find the frequency description of the clock, vc(t), Let vc(t) be a periodic square wave of height 1 and duration d as in Figure 3.11. Since vc(t) is periodic, it can be shown that the Fourier series representing vc(t) is given by equation 3.22.

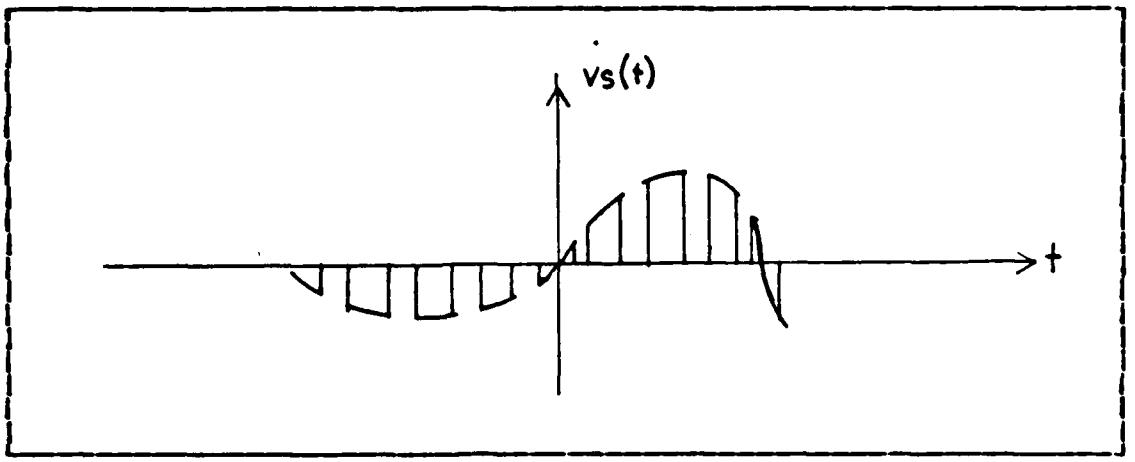


Figure 3.9 Samples of a Representative Voltage

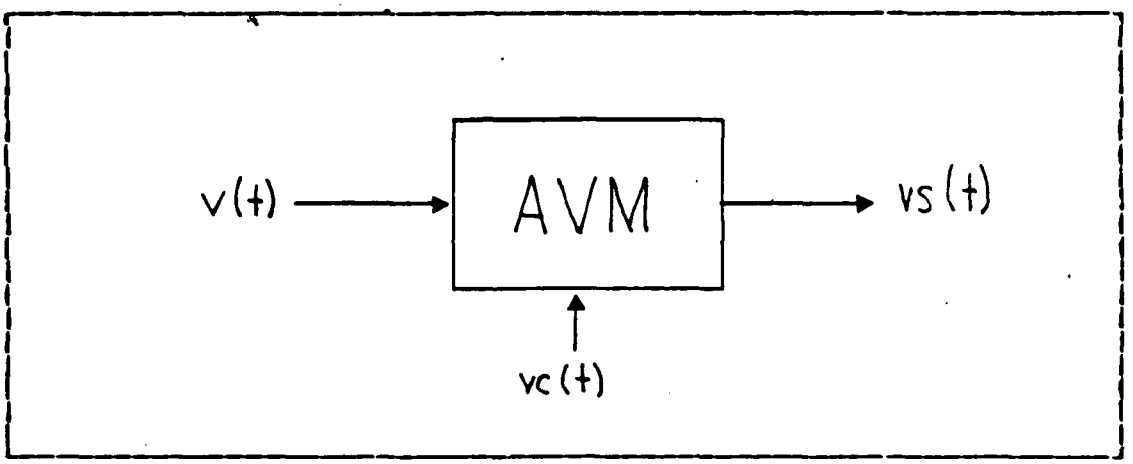


Figure 3.10 Analog Voltage Multiplier

Again it is noted that vs(t), the sample voltage, is the product of v(t), the original analog voltage, times vc(t), the clock voltage. In other words vs(t) = v(t) times a series cf cosine terms.

$$vc(t) = ao + \sum_{n=1}^{\infty} an \cos(2 \pi n fc t) \qquad \text{(eqn 3.22)}$$

ao and an are left unevaluated

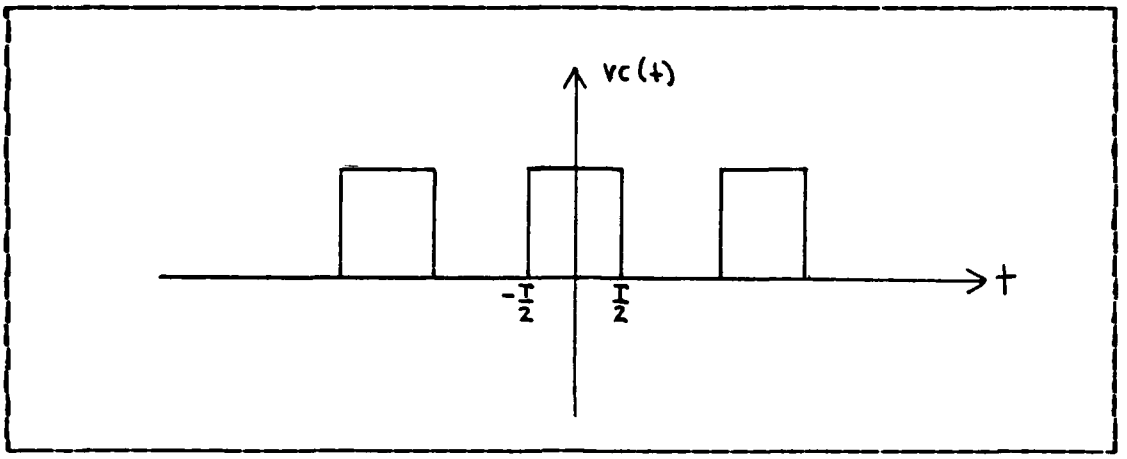


Figure 3.11 A Representative Voltage Clock

$$vs(t) = v(t)[ao + \sum_{n=1}^{\infty} an \cos(2 \text{ pi } n \text{ fc } t)] \qquad \text{(eqn 3.23)}$$

Now, assuming any form of the Fourier transform of v(t), as in Figure 3.12 and since vs(t) is a product and by utilizing the frequency translation property of the Fourier transform, the following is obtained as a representation of the frequency spectrum of vs(t). See Figure 3.13.

Remember that the curve highlighted in the box in Figure 3.13 is the Fourier represention of v(t), the original signal. This original signal can now be recovered by filtering with an appropriate low pass filter of bandwidth equal to or greater than B. This low pass filter would only permit the reception of that portion of the signal which represents the original voltage.

The only question left to resolve is how often to take a sample. Again referring to the diagram in Figure 3.13, it is noted that in order to prevent any overlap of successive translations (aliasing)

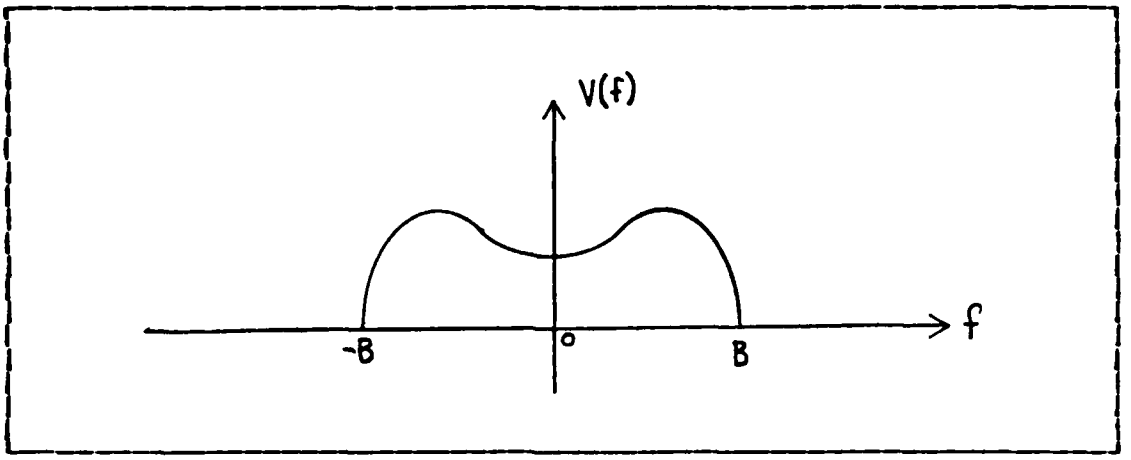


Figure 3.12 Fourier Transform of v(t)

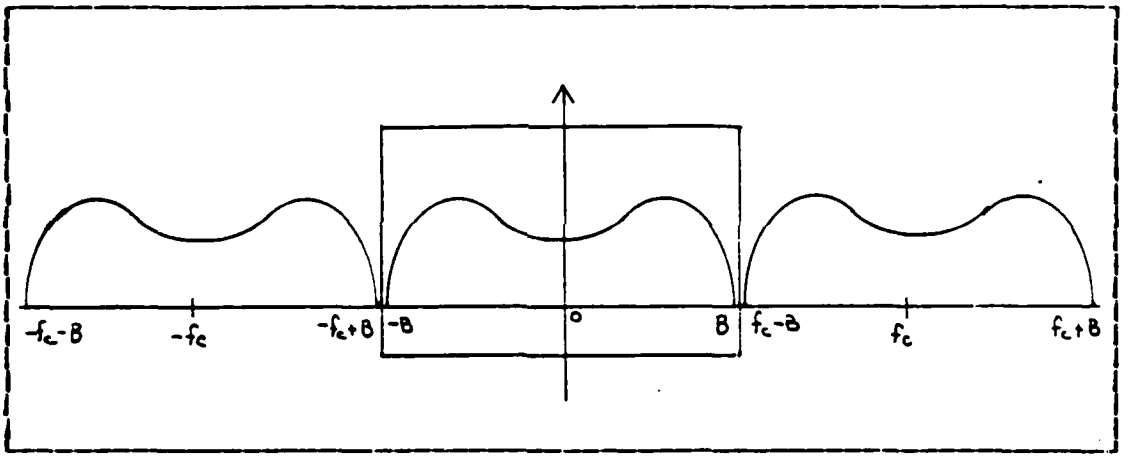


Figure 3.13 Fourier Transform of vs(t)

$f_o - B >> B$

$f_o >> 2B$

or the frequency of the clock, fc (the sampling rate = 1/T) must be strictly greater than 2 times the largest significant Fourier frequency component present. Since B will vary with different v(t), the sampling rate necessary to uniquely recover that v(t) also varies.

In summary on the sampling theorem, it can be said that any analog signal can be represented as certain sample values spaced the appropriate distance apart. The sampling theorem places limits on the accuracy of this representation. These sample values can then be transmitted from one position to another using analog to digital conversion and any one of a variety of modulation techniques. It has been shown that the Fourier frequency representation of the sample is not equivalent to the Fourier frequency representation of the source voltage; however, the source voltage can be recovered at the receiving end by filtering. A block representation of the system is shown in Figure 3.14.

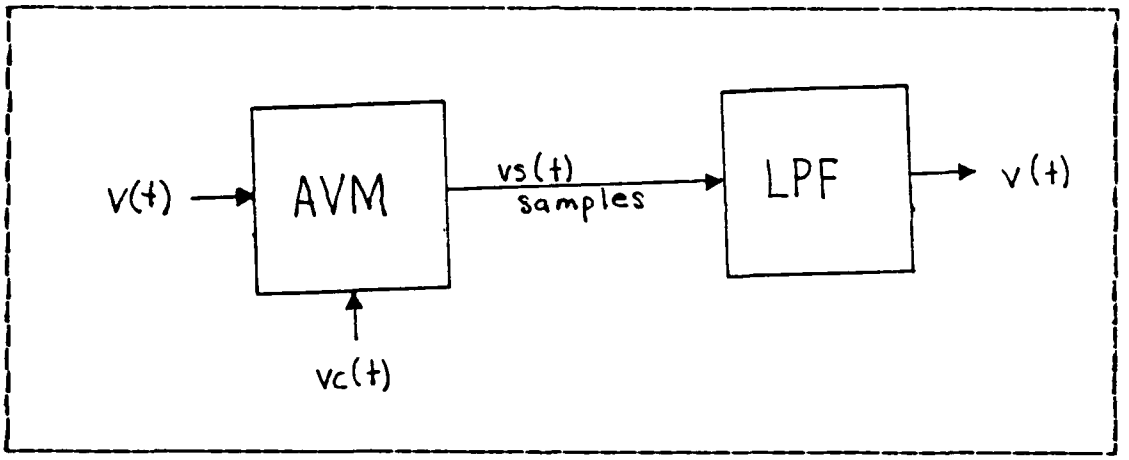


Figure 3.14 Analog Voltage Multiplier and Low Pass Filter

C. THE AUTOCORRELATION FUNCTION

The above concept certainly seems simple enough. If a signal is present and it is desired to transmit it from point A to point B, all that has to be done is to take appropriately spaced samples of the signal, encode them,

somehow transmit them to point B, filter the received signal and the uniqueness of the original voltage has been reproduced. Unfortunately it isn't quite that easy. The reason it is not, is due to the presence of noise. The sources of noise will not be discussed here, however, the effects of noise in general terms and how a faint signal can be recovered in the presence of noise will be discussed.

The time varying descriptor of voltage and the frequency varying descriptor of voltage have been introduced. Both are precise mathematical descriptions of something (voltage) that is deterministic. That is, it can be described in mathematical or graphical terms during any period of time. Such is not the case for noise corrupted voltages which are entirely random. Therefore other descriptors of voltages must be employed in the presence of noise. They are partial descriptors of voltage since a random signal cannot be described with precision. These partial descriptors are the

1. Autocorrelation function and the
2. Probability density function (p.d.f.)

It is important here to note the difference between the source voltage at the receiver and the sample voltage. The source voltage at the receiver, vsr(t), is the digitally converted sample voltage, vs(t), modulated onto a carrier by one of the digital modulation techniques yet to be discussed. Understanding the autocorrelation function is the basis for understanding how a decision is made that vsr(t) is present at the receiver in the presence of noise. It should be remembered that the exact form of the carrier is known at both the transmitter and receiver.

First of all, an additive noise model is assumed where the signal at the receiver, vr(t), is equal to the signal which is being transmitted, vsr(t), (i.e., the digital samples of v(t) modulated onto the carrier), plus random noise, vn(t)

$$vr(t) = vsr(t) + vn(t) \qquad \text{(eqn 3.24)}$$

vr(t) = voltage at the receiver
vsr(t) = signal voltage at the receiver
vn(t) = noise voltage at the receiver

If it is realized that most receivers operate on the principal of detection of DC voltage, the problem becomes one of rectification of the received signal and determination if the DC voltage, characteristic of the transmitted signal is present.

A common type of rectification of an analog voltage involves squaring the input waveform. If the given signal at the receiver is vr(t) as in equation 3.24 above, squaring the waveform introduces the square of not only the desired signal but also the square of the noise term. If instead of squaring vr(t) and introducing or at least not eliminating the noise, vr(t) is multiplied by vsr(t), the results in equation 3.25 are obtained.

$$vr(t) \cdot vsr(t) = vsr^{**}2(t) + vsr(t)\,vn(t) \qquad \text{(eqn 3.25)}$$

By averaging, all that remains is $\overline{vsr^{**}2(t)}$ since the average of vsr(t)vn(t) is zero because $\overline{vn(t)}$ is random and the average of any random voltage is zero. The DC component of vr(t)•vsr(t) is $\overline{vsr^{**}2(t)}$. Any remaining AC component can be removed by a low pass filter. Graphically in block diagram form this receiver looks like Figure 3.15 illustrated below. If vsr**2(t) > 0 then vsr(t) is present in the signal vr(t). If vr (t) is pure noise or some other signal is present exclusively, then the output of the receiver will be 0.

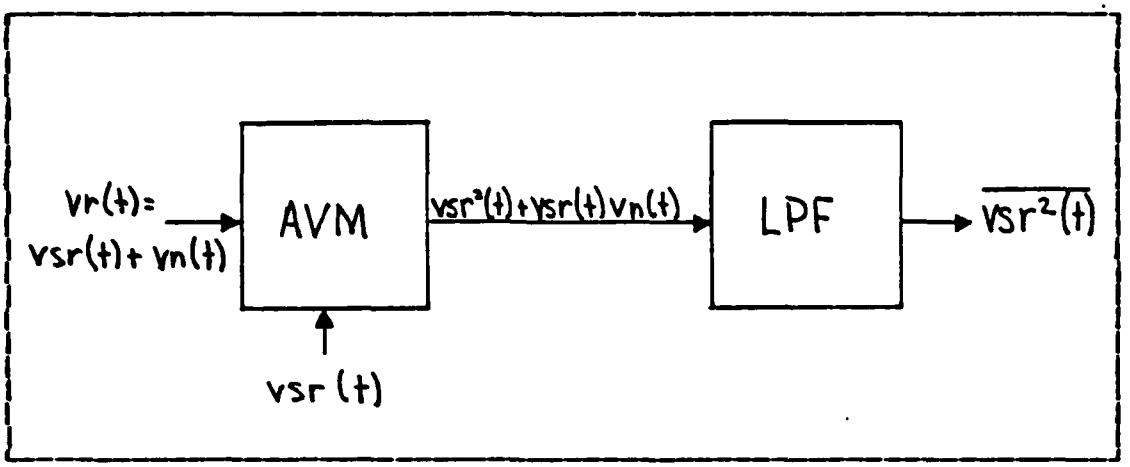


Figure 3.15 Rectification through an AVM and LPF

In practice, however, the exact waveform of vsr(t) may not be known or it may be a time delayed or distorted version of the original signal when it arrives at the receiver. If the distortion or delay is significant enough there will be little agreement or "correlation" in the receiver.

The solution is to multiply the received signal by a series of time delayed approximations of the transmitted signal. The signal is present whenever the output of the series of voltage multipliers or correlators exceeds a certain threshold approaching vsr**2(t). The concept is illustrated in Figure 3.16.

For the concept illustrated in Figure 3.16 to work, the average value of vr(t)•vsr(t-p) must have a DC component.

$$vr(t) \bullet vsr(t-p) = vsr(t)\,vsr(t-p) + vn(t)\,vsr(t-p) \qquad \text{(eqn 3.26)}$$

$$\overline{vr(t) \bullet vsr(t-p)} = \overline{vsr(t)\,vsr(t-p)} + \overline{vn(t)\,vsr(t-p)} \qquad \text{(eqn 3.27)}$$

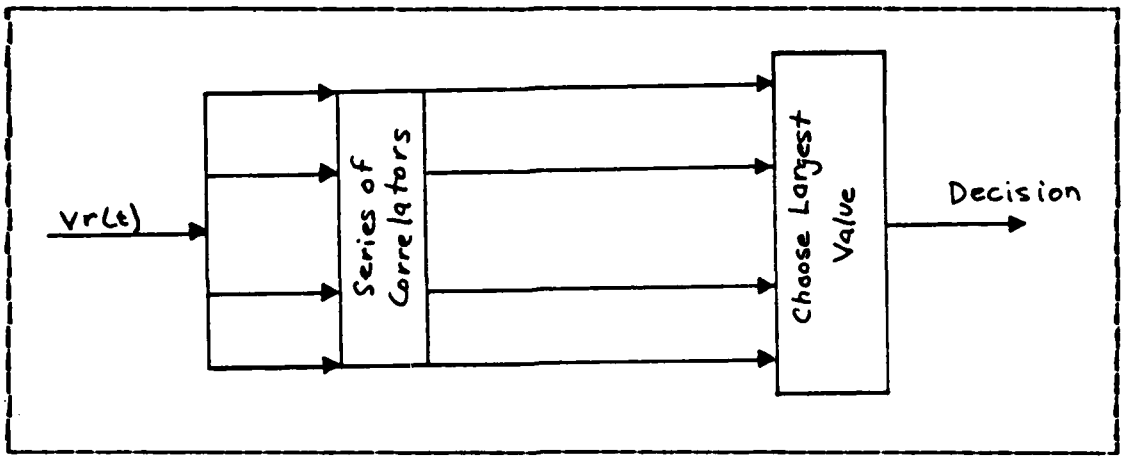


Figure 3.16 Correlation Device

But the average of $vn(t)\,vsr(t-p) = 0$ since $\overline{vn(t)}$ is random and the results of equation 3.28 are obtained.

$$\overline{vr(t) \cdot vsr(t-p)} = \overline{vsr(t)\,vsr(t-p)} \qquad \text{(eqn 3.28)}$$

What is of interest is the average value of $vsr(t)vsr(t-p)$. The autocorrelation function (ACF) will be defined as that average value of $vsr(t)vsr(t-p)$. The mathematical representation of the autocorrelation function for a continuous voltage is represented in equation 3.29 while the autocorrelation function for a pulse voltage is presented as equation 3.30.

$$ACF = Rvv(p) = 1/2T \int_{-T}^{T} v(t)\,v(t-p)\,dt \qquad \text{(eqn 3.29)}$$

$$ACF = Cvv(p) = \int_{-\infty}^{\infty} v(t)\,v(t-p)\,dt \qquad \text{(eqn 3.30)}$$

The autocorrelation function is a measure of the degree to which two identical signals which are corrupted in some manner are alike. The ACF of two signals which are identical, not distorted in any way and occurring at exactly

the same time, i.e., p = 0 has a maximum value. On the other hand there will be very little correlation between two identical time variant signals when the time difference between them is great. In this case the autocorrelation function is near zero.

A similar concept is used within the context of signal comparison. This concept is the crosscorrelation which is a measure of the degree to which 2 different signals are alike. When the crosscorrelation between voltage vsr and random noise voltage vn is 0 there is no relationship or correlation.

Again let us assume that the signal at the receiver is a noise disrupted version of the signal originally transmitted.

$$vr(t) = vsr(t) + vn(t) \qquad \text{(eqn 3.31)}$$

In the receiver , the time delayed version of the signal is applied to the incoming signal and the average is formed as before and shown in Figure 3.17.

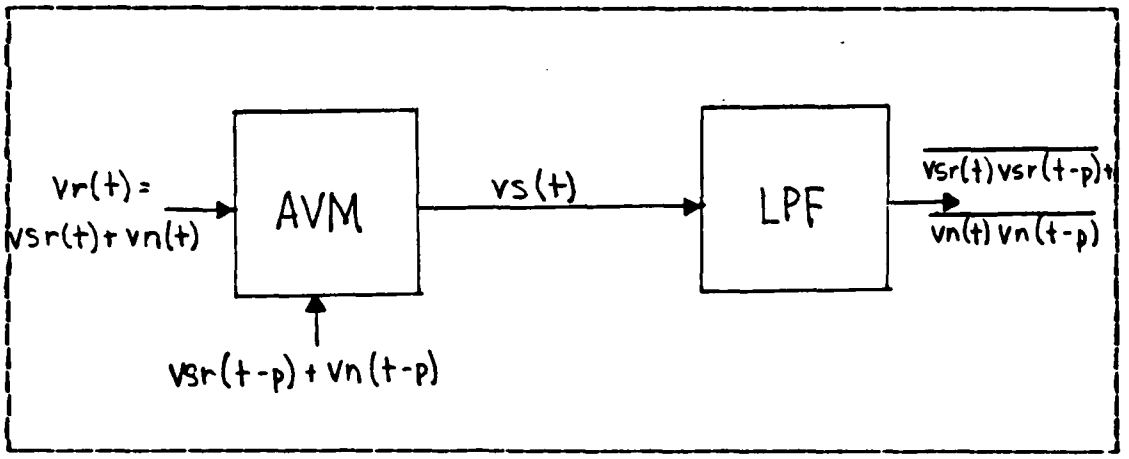


Figure 3.17 Voltage Correlator

$$\overline{[\,vsr(t) + vn(t)\,]} \cdot \overline{[\,vsr(t-p) + vn(t-p)\,]} = \qquad \text{(eqn 3.32)}$$

$$\overline{vsr(t)\,vsr(t-p)} + \overline{vn(t)\,vsr(t-p)} +$$

$$\overline{vsr(t)\,vn(t-p)} + \overline{vn(t)\,vn(t-p)} =$$

$$Rvsr\ vsr(p) + Rvn\ vsr(p) + Rvsr\ vn(p) + Rvn\ vn(p)$$

$$Rvn\ vsr(p) = 0$$

$Rvsr\ vn(p) = 0$; because average $vn = 0$

Therefore the result is simply $Rvsr = vsr(p) + Rvn\ vn(p)$ or the autocorrelation function of the desired signal plus the autocorrelation function of the noise. Assuming the shape of the autocorrelation function of both components (both the original signal and the noise which vary with p, time) is known, what is done in practice is to vary the value of p until the ratio of $Rvsr\ vsr(p)\ /\ Rvn\ vn(p)$ is a maximum or the autocorrelation function of the signal is a maximum while the autocorrelation of the noise is minimum and the signal is recovered.

D. THE MATCHED FILTER

The matched filter is a linear filter which has the characteristic response desired for optimum reception of the desired signal. In other words, we desire the response of the system to the linear filter to be in some way proportional to the autocorrelation function of the desired signal and in no way related to the autocorrelation function of the noise. This system could be illustrated as in Figure 3.18.

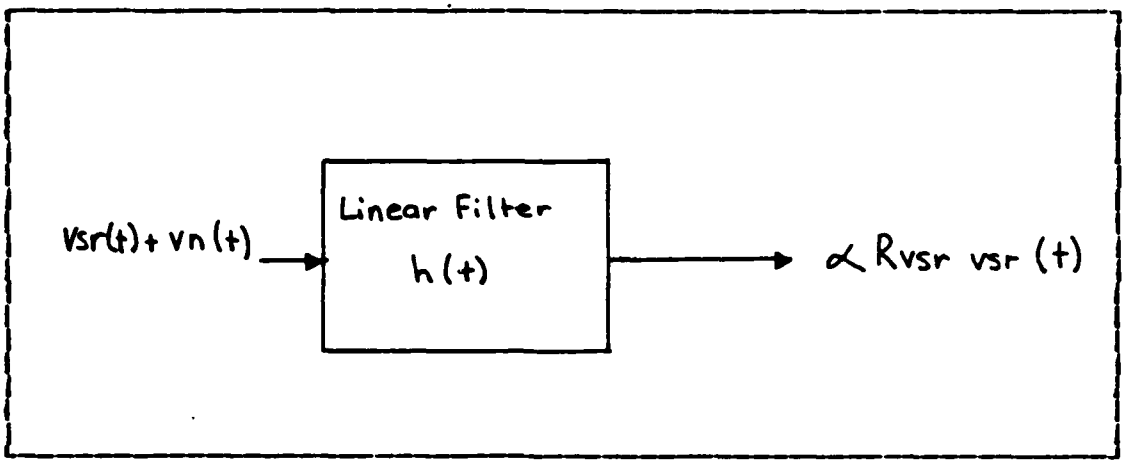


Figure 3.18 Matched Filter Block Representation

The question is what should the makeup or h(t), (i.e., the response of the linear filter) be? The output of a linear system is a convolution so the output of the linear filter, h(t) to an input vsr(t) would be vsr(t) * h(t), where * denotes convolution.

In the previous section it was stated that the autocorrelation function, Rvsr vsr(p), a function of p, is given by

$$R_{vsr\ vsr}(p) = \int_{-\infty}^{\infty} vsr(t)\,vsr(t-p)\,dt \qquad \text{(eqn 3.33)}$$

By simply changing variables, equation 3.33 can be rewritten as a function of time, t.

$$R_{vsr\ vsr}(t) = \int_{-\infty}^{\infty} vsr(p)\,vsr(p-t)\,dp \qquad \text{(eqn 3.34)}$$

From convolution theory it is known that the response of a linear filter as above, vsr(t) * h(t) can be expressed mathematically as in equation 3.35.

$$vsr(t) * h(t) = \int_{-\infty}^{\infty} vsr(p)\,h(t-p)\,dp \qquad \text{(eqn 3.35)}$$

As stated earlier, the desired output of the linear filter is the autocorrelation function Rvsr vsr(t). Note the similarities between the last two equations. If in equation 3.35 a simple substitution of h(t) = vsr(-t) is performed, the desired results are obtained.

$$vsr(t) * h(t) = \int_{-\infty}^{\infty} vsr(p)\, vsr(p-t)\, dp \qquad \text{(eqn 3.36)}$$

$$= Rvsr\ vsr(t)$$

Therefore, for every signal that is desired to be recovered, the matched filter will do the job very nicely if the response of that filter, h(t) is equal to the inverse of the signal that is desired to be detected.

IV. ANALOG TO DIGITAL CONVERSION

Key to the understanding of digital communications is how the analog information is converted to digital information. As illustrated earlier, each analog voltage or signal can be represented by appropriately spaced sample values. These sample values are still analog in that they can take on any value in the range of the original analog signal. What is desired is to change this infinite set of decimal numbers to a finite set of decimal numbers. This is called analog to digital conversion. Common types of A to D conversion include Pulse Code Modulation (PCM), Differential Pulse Code Modulation (DPCM), Delta Modulation (DM), Pulse Amplitude Modulation (PAM), Pulse Duration Modulation (PDM),and Pulse Position Modulation (PPM). PCM, DPCM, and DM will be examined in this chapter because of their widespread usage and easy application to digital technology.

A. PULSE CODE MODULATION (PCM)

PCM is the most widely used A to D conversion technique. It involves assigning the sample value obtained in sampling to one of several quantization levels within the range of the voltage sampled and then representing those quantization levels by various binary code words.

For example, the highest significant frequency component in human speech is 3300 Hz. In order to capture accurately the signal produced in speech, one would require a sampling rate greater than or equal to 2B or 2(3300) = 6600 samples/sec. The telephone company uses 8000 samples/sec.

B = 3300 Hz

$f \geq 2B$ = 6600 samples/sec

use f = 8000 samples/sec to ensure no aliasing

If a signal is sampled at the rate of 8000 samples/sec, then in the transmission of those samples, assuming the samples are transmitted immediately and not delayed, the transmission rate must be

Ts = 1/f

Ts = 1/8000 samples/sec

Ts = 125 microsec/sample

The number of quantization levels employed to represent the various analog sample values is arbitrary. The greater the number of levels, the more accurate the reconstruction of the original signal but the more rapid the data transmission rate must be(in all cases there will be some error present after recovery). Consider again the example of voice as illustrated in Figure 4.1.

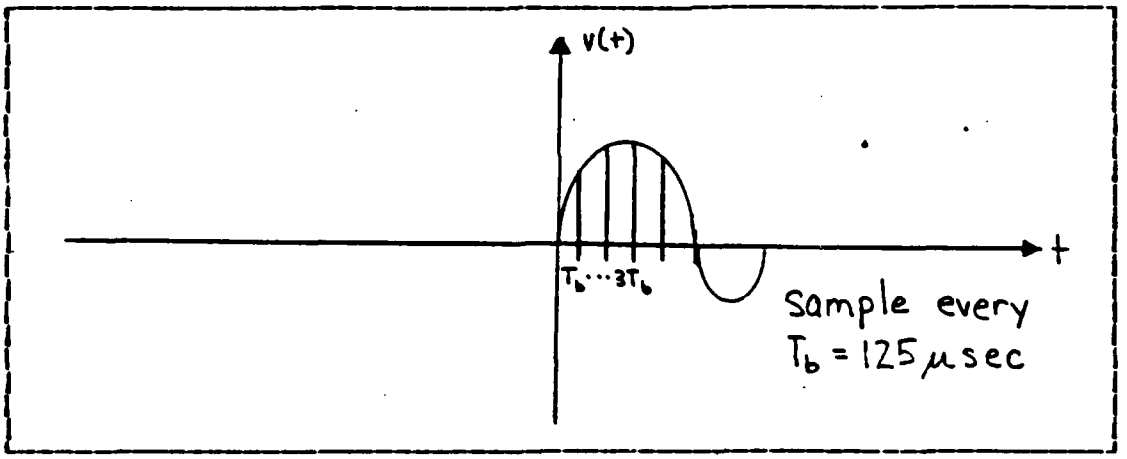


Figure 4.1 Samples of Voltage

Assume it is desired to represent each sample value (decimal number) with an 8 bit binary code word. Then the number of quantization levels is given by equation 4.1.

The spacing between quantization levels is the range of voltages to be represented divided by the number of quantization levels. The analog to digital conversion takes

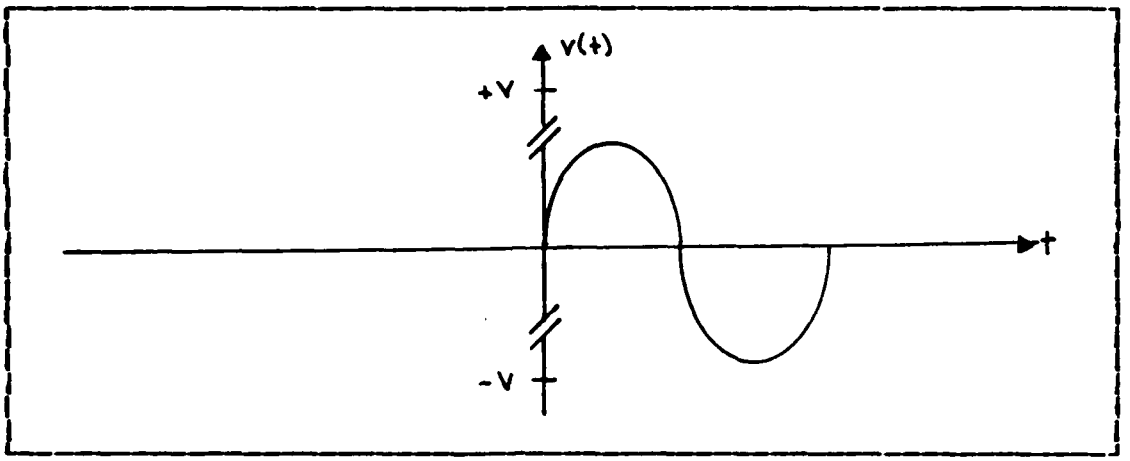


Figure 4.2 Analog Nature of Samples

#quantization levels = 2**n (eqn 4.1)

for n = 8 bits/binary code word
#quantization levels = 2**8
= 256

place by determining into which quantization level the respective sample values fall and assigning that sample value the binary number associated with that quantization level.

Now each sample value is assigned to an eight bit binary code word. For voice transmission it has been shown that the minimum transmission rate was Ts = 125 microsec/sample. The bit rate is then simply equal to the number of bits per sample times the inverse of the transmission rate as illustrated in equations 4.2 and 4.3.

If voice was to be time multiplexed onto a single channel, the bit rate for that channel would be the number of signals per channel times the bit rate for the most time intensive signal as illustrated in equation 4.4.

bit rate = (n bits/sample)(1 sample/T sec) (eqn 4.2)
bit rate = (8 bits/sample)(1 sample/125 microsec)
= 64 kbps

bit duration = 1/bit rate (eqn 4.3)
= 1/64 kbps = 15.6 microsec/bit

(#signals/channel)(bit rate/signal) = (eqn 4.4)
bit rate/channel

In other words, if 24 voice signals are to be multiplexed onto a single channel, the bit rate must be

bit rate/channel ≥ (eqn 4.5)
(24 signals/channel)(64 kbps/signal)
= 1536 Mbps

As can be seen, the data transmission rate increases significantly, necessitating better and better hardware and a wider and wider bandwidth. Advantages should be readily apparent for A to D conversion techniques or modulation techniques that reduce the bit rate required.

Recovery of the analog signal is by a process called Digital to Analog Conversion. All that will be said about D to A conversion is that it is the inverse process of A to D conversion and that a slight error is always introduced during the process due to the discrete nature of the quantization process during A to D and D to A conversion.

B. DIFFERENTIAL PULSE CODE MODULATION (DPCM)

In DPCM, what is converted to binary numbers is not the quantization level (decimal number) which each sample value

is represented by but the successive differences between quantization levels. The idea is that the range of maximum difference values will be smaller than the range of actual sample values. It is therefore possible to represent that range of delta values with fewer bits/binary word and therefore fewer quantization levels and ultimately a lower bit rate. [Ref. 11]

In actual practice, in a DPCM conversion technique, it is a statistical estimate of each successive sample value which is subtracted from the actual value that is converted to binary code words. The result is the same in that the range of amplitudes is reduced and therefore fewer bits/word are required to represent the sample thereby lowering the data transmission rate required to transmit the signal. [Ref. 11]

C. DELTA MODULATION (DM)

Delta modulation is a form of DPCM where successive quantization levels of the output differ by only 1 bit. That is to say that any successive quantization level can be represented by varying only one bit of successive output binary code words. In other words a type of gray code is employed. This A to D technique is implemented through the use of a DM coder or linear delta modulator. This DM coder approximates a given input signal with a series of linear segments of uniform slope. A comparison is made between the value of this approximation and the input signal at each sample increment. The sign of this difference value is what is encoded and is used to increment the DM coder in the direction of the input signal. By using the differential sign value of the input signal and the incremented approximation from the DM coder the linear approximation from the linear delta modulator is said to "track" the input

signal. [Ref. 8] Slope overload of this type of modulation technique occurs when the slope of the incoming signal exceeds the ability of the DM system to follow the source at the sampling rate being utilized. [Ref. 11].

V. DIGITAL SIGNAL MODULATION TECHNIQUES

Digital signal modulation techniques are the methods of encoding information for transmission utilizing digital technology. Factors affecting digital modulation include, but are not limited to, the physics of the method, hardware requirements, bandwidth considerations, power requirements, data transmission rates and error probabilities. It is these aspects which will be explored in the following sections.

A. DIGITAL MODULATION FORMATS

Modulation is the technique by which the characteristics of one waveform (called the carrier) are varied or modified by the characteristics of another (called the source). The carrier waveform of interest is the sinusoid. It should be obvious that the attributes of a particular sinusoid that differentiate it from every other sinusoid are its amplitude, phase and frequency. It follows that the characteristics of the carrier waveform to be varied by the source are its amplitude, phase, frequency or any combination of the three. This gives rise to the broad general formats of Amplitude Shift Keying (ASK), Phase Shift Keying (PSK) and Frequency Shift Keying (FSK). All other digital modulation techniques are variations of or combinations of these basic formats. Other factors involved in the description of the digital modulation technique being employed include the number of bits being encoded at one time, the employment of error correction techniques and the baseband (source) waveform.

When each bit of the baseband waveform is individually encoded by any of the techniques previously mentioned (ASK, PSK, FSK), the technique being utilized is referred to as binary encoding. When more than 1 bit of the source code is modulated onto the carrier at one time it is called block encoding. Block encoding allows for one of m = 2**k waveforms where k = number of bits.

This paper will not go deeply into signal detection or demodulation techniques, however, a basic understanding of what is involved is necessary and again further defines the digital modulation format being employed. In general terms, signal detection is referred to as either coherent or noncoherent detection of the transmitted signal. Coherent detection, perhaps the easiest to understand, is when all possible waveforms of the modulated carrier waveform are available at the receiver and the waveform at the receiver is in phase with the transmitted carrier. Noncoherent detection is involved when the receiver does not have knowledge of the phase of the transmitted information and one of a number of phase estimation techniques must be employed for signal recovery.

B. HARDWARE

Althougn significant advances have been made in recent years to improve the quality of hardware associated with satellite communications, most components are not what would be considered "off the shelf items". The hardware components most commonly referred to with regard to digital communications include the sampler, encoder, modulator, multiplexer and transmitter. There are variations of the above hardware requirements necessary for certain types of digital formatting, however, those exceptions will be addressed separately when the individual modulation techniques are examined.

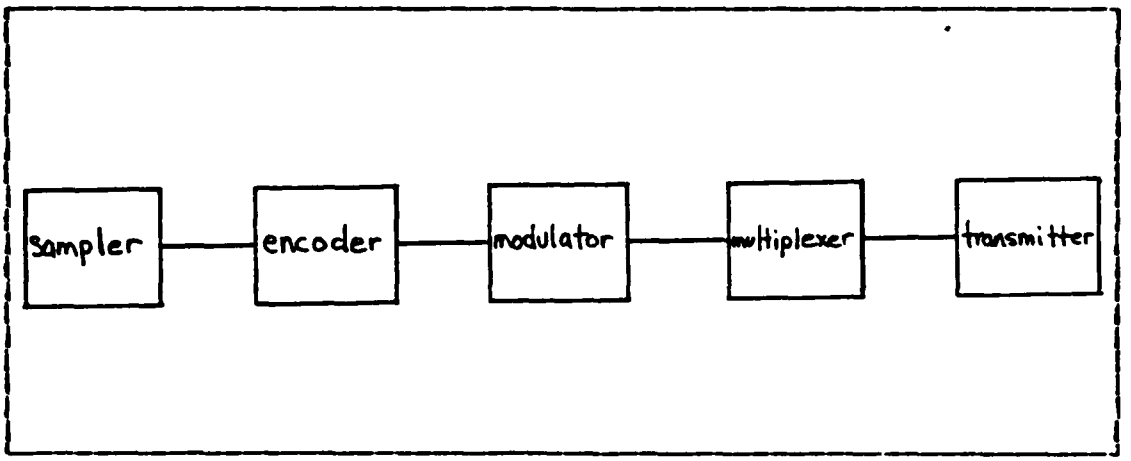


Figure 5.1 Basic Hardware Component Block Diagram

1. Sampler

The sampler is the device or component which samples or extracts those single characteristic values of a naturally occurring analog signal which are ultimately encoded into a digital word by the encoder. The frequency at which this sampler must operate was derived in the discussion of the Sampling Theorem. It was determined that the sampler must operate at a frequency greater than 2 times the highest significant Fourier frequency component of the source voltage. Modern solid state devices capable of taking thousands of samples/sec are available at modest costs.

2. Encoder

Often referred to as the A to D converter (analog to digital converter), the encoder transforms the samples of the analog signal derived from the sampler into a digital format through one of the analog to digital conversion techniques described in the chapter on A to D conversion. These digital bits can be stored for later use, coded,

delayed or used immediately either individually or in groups to modify one of the characteristic qualities of the carrier waveform. When sample values of the analog source signal are ccnverted into digital bits of information, they are simply that, digital bits of information. The carrier can only be modified to represent the source information by interaction with another voltage. Therefore the value of the digital bit (binary), either 0 or 1, is used to generate a baseband waveform or voltage which does the actual modulation of the carrier waveform.

a. Common Baseband Waveforms

It should be obvious that since it is desired to represent binary code words with a representative baseband waveform what is required is two levels of voltage. There are two basic logic schemes for representation of the baseband waveform. They are bipolar or unipolar logic. Bipolar logic involves representing the 0's or 1's of the binary codeword as either +V or -V as illustrated in Figure 5.2.

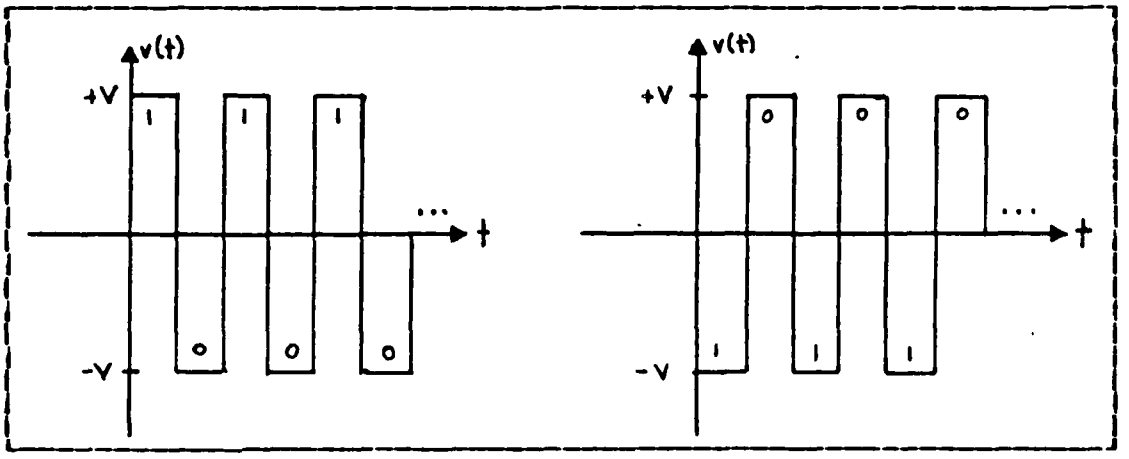


Figure 5.2 Bipolar Logic

The 0 or the 1 of the binary code word can be represented either as +V, -V or -V, +V respectively. This is a matter of convention but must be clearly understood in the various component designs in order to ensure compatibility between parts of the system.

Unipolar logic utilizes a voltage to represent either of the possible binary digits and the absence of voltage to represent the other. Two common conventions of unipolar logic are represented in Figures 5.3 and 5.4.

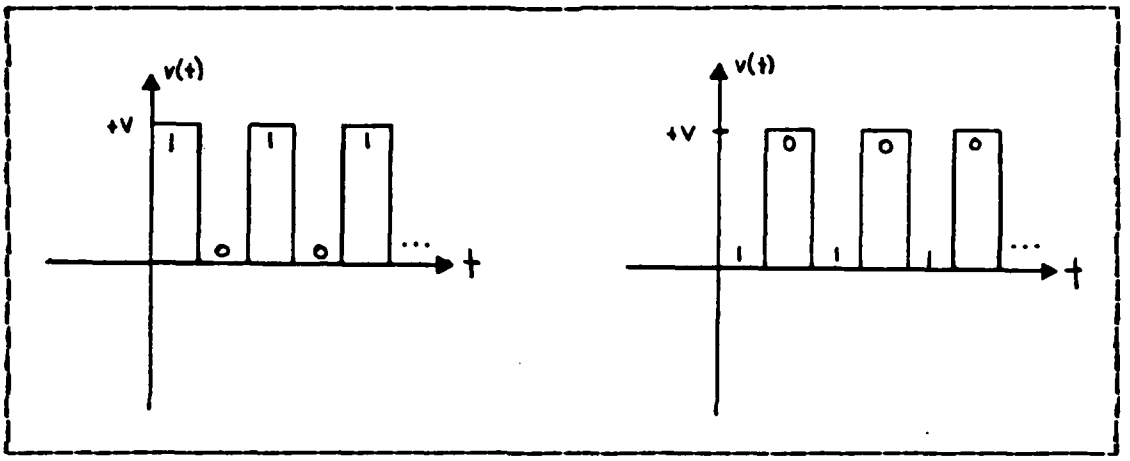


Figure 5.3 Unipolar Positive Logic

Again, which form of unipolar logic might be employed in the A to D converter is a matter of convention.

Variations of these two basic encoding formats have the advantages of ease of generation and improved decoding and clock recoverability. Two variations commonly used in satellite communications are the Non Return to Zero (NRZ) and the Manchester waveforms. The NRZ waveforms is simply the voltage representation which corresponds to the stream of bits represented in the logic scheme chosen. There is no transition as long as the same bit is present. Choosing bipolar logic as an example, a NRZ waveform can be described schematically as in Figure 5.5.

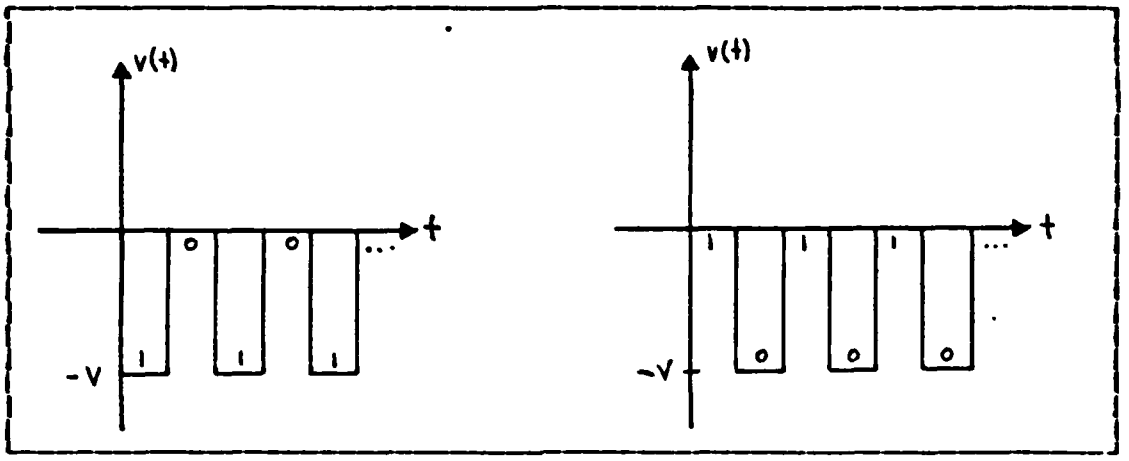


Figure 5.4 Unipolar Negative Logic

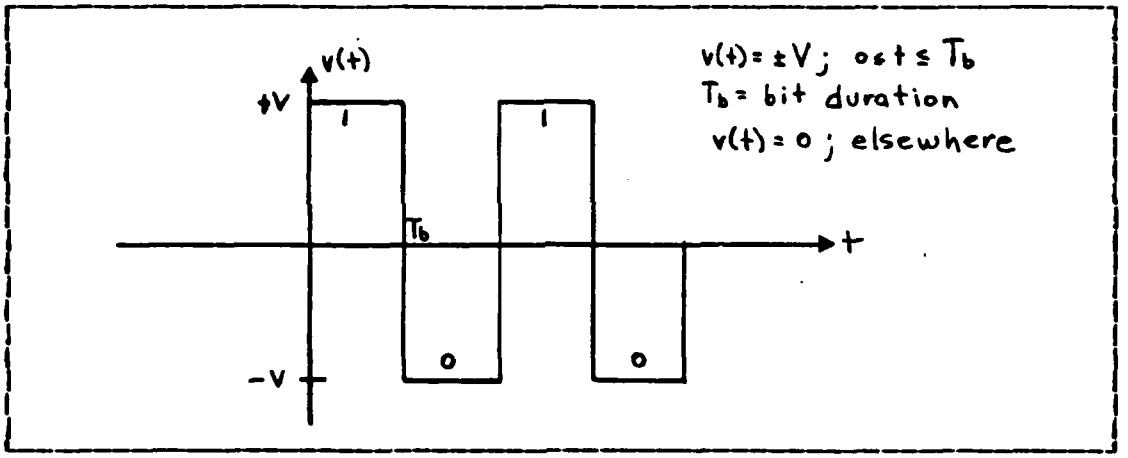


Figure 5.5 Non Return to Zero Waveform

The Manchester waveform of the same source voltage is represented with a transition at the midpoint of the bit duration from either +V to -V or -V to +V. This form of coding has the advantage of clock synchronization in all cases of digital encoding. Schematically, the Manchester code can be represented as in Figure 5.6.

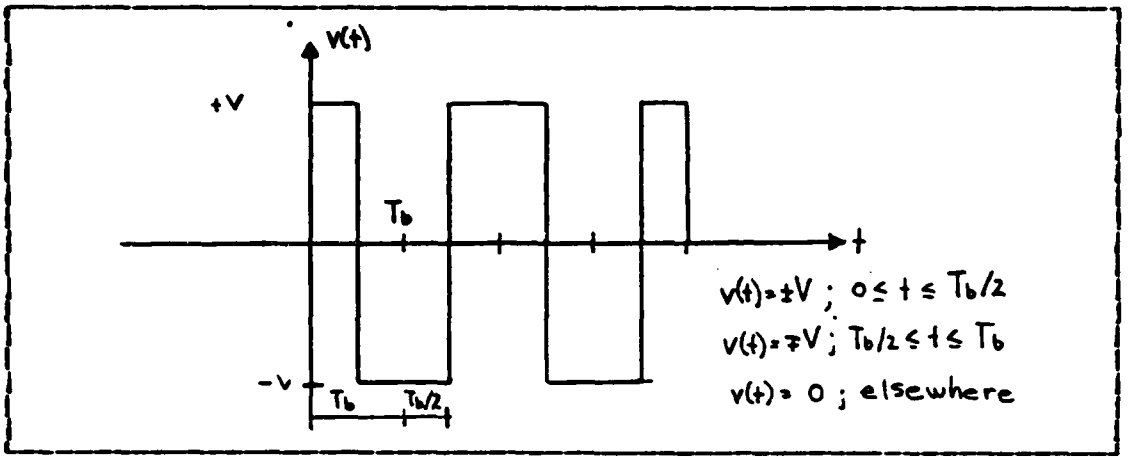


Figure 5.6 Manchester Waveform

3. Modulator

The modulator in a digital communications system is more commonly referred to as a modem (modulator/demodulator). This device does the actual transformation of the digital information into a waveform that can be transmitted from one point to another. The modulation as mentioned earlier can modify either the amplitude, phase or frequency of the carrier wave. In simple terms, the digital information derived from the analog source is mapped into the carrier wave. It is this mapping that determines the power and bandwidth characteristics of the modem.

4. Multiplexer

Multiplexing is a common method for increasing the utility of a given communications channel. Variations in multiplexing techniques give rise to the multiple access techniques employed in satellite communications.

a. Frequency-Division Multiplexing (FDM)

Frequency-Division Multiplexing is a technique whereby the capacity of a communications channel is increased by adding one signal to another. These signals occupy discrete nonoverlapping frequency bands. A specific signal is recovered by use of a band pass filter for the frequency band desired. [Ref. 7]

b. Time-Division Multiplexing (TDM)

Time-Division Multiplexing is a technique used to increase the capacity of a given communication channel by adding signals in time. That is, specific signals or portions of signals are allocated specific time slots or portions of the carrier wave. These time allocations cannot overlap and the signal is recovered through proper synchronization of the recovery hardware with the multiplexer. [Ref. 7]

c. Code-Division Multiplexing (CDM)

Code-Division Multiplexing is a technique for increasing the capacity of a given communications channel through the assignment of a characteristic code to the digital signal before it is multiplexed in the time or frequency domain. Since the code used in multiplexing is known at the demultiplexer, recovery of the digital source code could be accomplished through the use of a matched filter or correlator. [Ref. 7]

C. BANDWIDTH

For purposes of discussion and for uniformity, when examining the various digital signal modulation techniques of interest, the bandwidth will be defined as the highest significant Fourier frequency component of the signal in

question. For digital communications, it can be shown that by the above definition, the bandwidth of the baseband waveform is 1/Tb, where Tb is the bit duration. For example:

1. Voice sampled at the rate of 8000 samples/sec
2. Each sample represented by an 8 bit code word
3. Rb = bit rate = (8000 samples/sec)(8 bits/sample)
   Rb = 64 kbps
4. Tb = bit duration = 1/Rb
   Tb = 15.6 microsec

The Fourier transform of the baseband waveform of duration 15.6 microsec is represented in Figure 5.7.

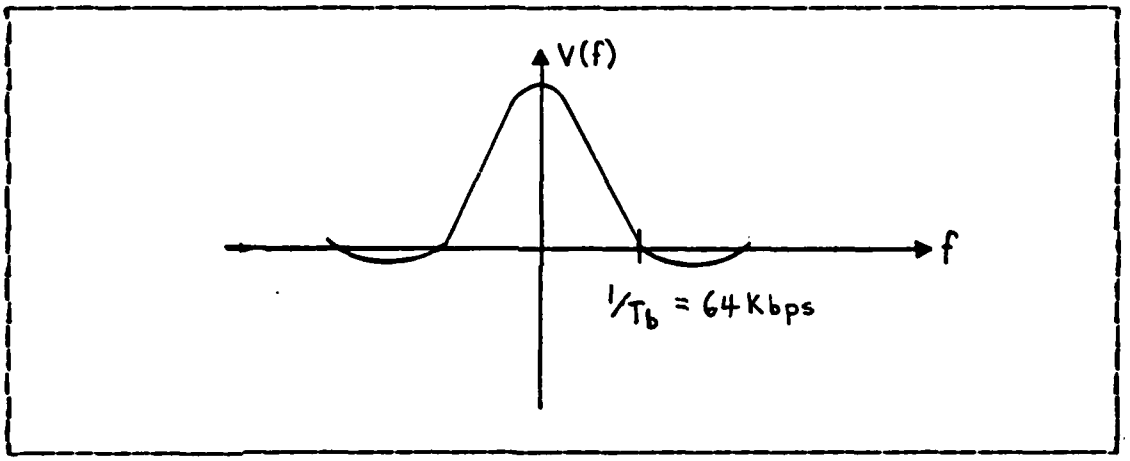


Figure 5.7 Fourier Transform of Baseband Waveform

It should be readily apparent that as the bit rate of the digital baseband waveform increases, so does the bandwidth. It is desireable to keep the bandwidth of the signal to a minimum due to limitations on availability of the electromagnetic spectrum. In addition, interference with signals which occupy adjacent frequency bands is reduced and propogation limitations of the medium and hardware limitations such as self-inductance and capacitance are minimized.

D. SPECIFIC TECHNIQUES

Now with some of the basics of digital signals already introduced it is possible to investigate some of the more significant modulation formats employed in digital satellite communications. The organization of this section will be around the three general types of modulation, i.e., phase, amplitude and frequency. Within each broad category, several techniques will be described under the subareas of binary encoding, where the information is modulated onto the carrier bit by bit, and block encoding, where groups of bits do the modulation. Variations of these areas will be pointed out. The characteristics of the modulation technique will be introduced including, where practical, an analytical description, phasor representation, error probability, spectral efficiency, advantages and disadvantages.

1. Phase Shift Keying (PSK)

Phase shift keying is a generic form of signal modulation where the baseband waveform is used to modify or change the phase of the carrier. Phase, as stated earlier is one of the characteristics which distinguish one sinusoid from another. In general, phase shift keying offers the advantages of power and bandwidth efficiency.

a. Binary Phase Shift Keying (BPSK)

As the name suggests, Binary Phase Shift Keying (BPSK) results in a modulated carrier waveform consisting of two phases of the same sinusoid. The baseband waveform is used to change the phase of the carrier bit by bit from one phase to the other. In the case of bipolar logic, the bit stream consists of ±1. The general waveform of the carrier can be represented as a cosine wave of amplitude A, angular

frequency w and initial phase angle delta as in equation 5.1.

$$vc(t) = A \cos(wt + delta) \qquad \text{(eqn 5.1)}$$

Since two phases are represented in vc(t), it is customary to let them differ by pi radians. This can be represented by the modification of equation 5.1 to account for a pi phase shift depending on whether the bit, v(t) to be modulated is either a +1 or -1 as shown in equation 5.2.

$$vc(t) = A \cos(wt + delta + pi/2(1-v(t))) \qquad \text{(eqn 5.2)}$$

Equation 5.2 can be expanded to result in a simplified form showing that the modulated carrier waveform is simply a product of the baseband waveform and the carrier. See equation 5.3 where v(t) represents the baseband waveform.

$$vc(t) = A\, v(t) \cos(wt + delta) \qquad \text{(eqn 5.3)}$$

A phasor diagram can be constructed to represent the modulated carrier waveform in BPSK and is shown in Figure 5.8. As can be seen by the phasor diagram in Figure 5.8, the phase of the modulated waveform depends on the value of the baseband waveform and differs by pi depending on whether the value of the modulated bit is ±1.

Remembering the section on the properties of the Fourier transform, it is possible to analyze the spectral and power efficiency of BPSK. Recall BPSK can be created by multiplication. Multiplication of two voltages is covered by the frequency translation property of the Fourier transform. Frequency translation results in double the bandwidth or spectral requirement of the translated waveform and half the power. In the case of a NRZ baseband waveform,

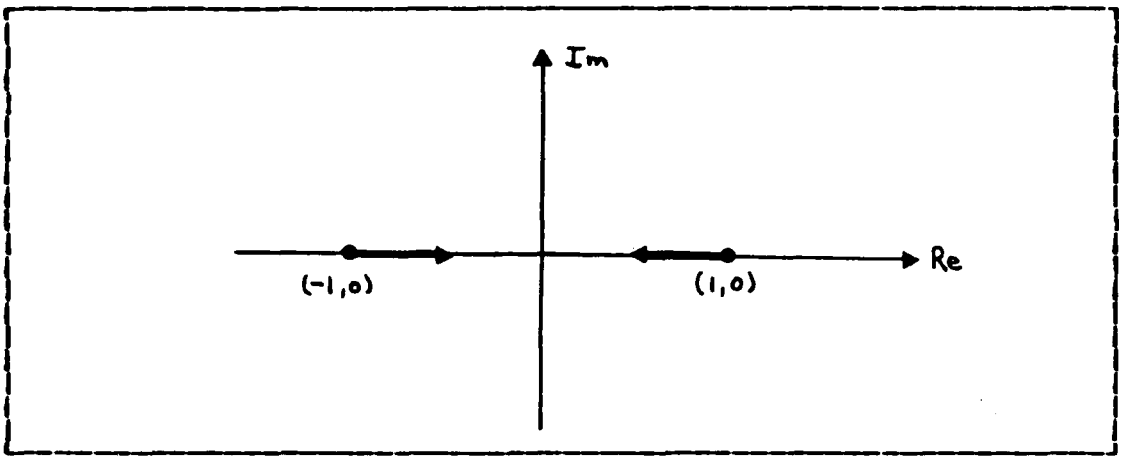


Figure 5.8 Phasor Diagram of BPSK

the bandwidth of the non-translated pulse, determined from the largest significant Fourier frequency component is 1/Tb, where Tb is the bit duration. The bandwidth of the translated voltage (BPSK carrier) is 2/Tb. The power in a sinuscid is $A^2/2$ and none of this power capacity is lost in BPSK since only the phase of the original sinusoid is changed. For a Manchester baseband waveform, the non-translated signal has bandwidth 2/Tb and the translated wave has bandwidth 4/Tb.

The advantage of BPSK is that it is relatively efficient in bandwidth utilization and power capacity. It does have the disadvantage of fairly large sidelobe components which contribute to interference with adjacent frequencies. This usually necessitates some type of filtering before the signal can be transmitted.

b. Differential Binary Phase Shift Keying (DBPSK)

The recovery of a Binary Phase Shift Keyed signal requires the use of a phase coherent reference signal as described in the previous sections on autocorrelation and matched filters. This may not always be practical or

possible so a technique called Differential Binary Phase Shift Keying has be developed. DBPSK uses the same carrier type as BPSK and offers many of the same advantages and disadvantages. As stated, the primary difference is the lack of necessity for a phase coherent reference.

The baseband waveform, v(t), for DBPSK is generated by comparing the phase of the next bit to be transmitted to that of the previously transmitted bit. For example, if a +1 is encoded onto the carrier as v(t), it represents a particular phase of the carrier. If the next bit to be sent is also a +1, there is no phase change, i.e., the bit remains the same over two successive bit durations. If, however, the second bit to be sent is a -1, this represents a pi phase change and is decoded as a bit change over successive bit durations. This can be expressed mathematically as in equation 5.4 where bk represents the bit which will be encoded on the carrier and bk-1 represents the previous bit already encoded and ak represents the present bit.

$$bk = bk\text{-}1 \bullet ak \qquad \text{(eqn 5.4)}$$

As can be seen when bk = -1, there is a phase change and the bit changes from one bit to the next depending on the logic type being employed. If bk = +1, this represents the phase remaining constant over two successive bit durations, i.e., no bit change. See Figure 5.9 [Ref. 8]. Recovery is effected by correlation of the received bit to the previously received bit.

Differential Binary Phase Shift Keying (DBPSK) differs from Differentially Encoded Binary Phase Shift Keying (DEBPSK) in the recovery step only. DPPSK takes advantage of the modulation of phase shift in the recovery process while DEBPSK still utilizes a phase coherent carrier

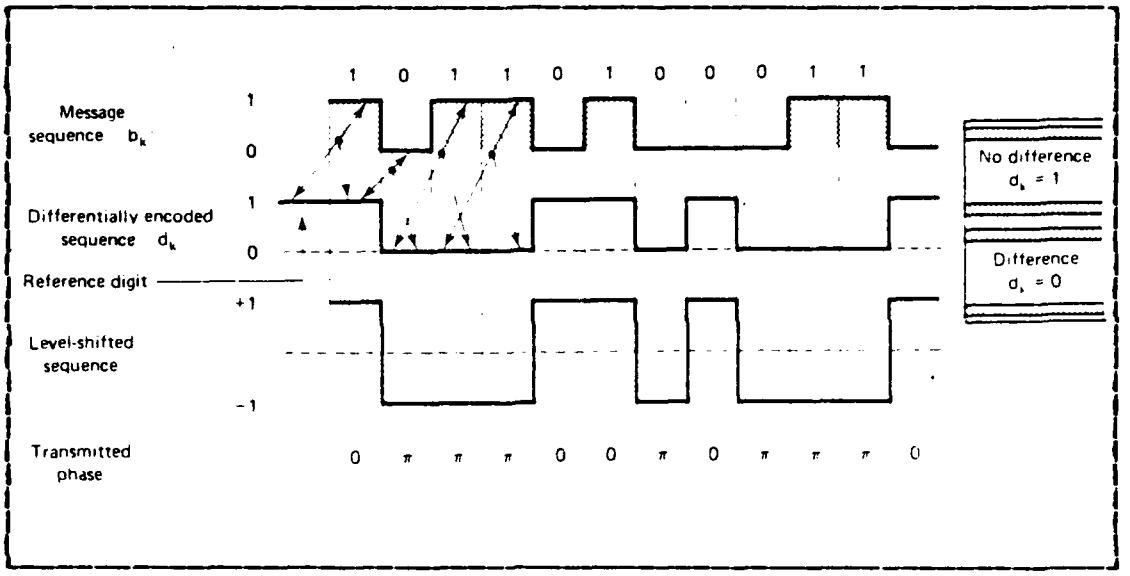


Figure 5.9 Differential Binary Phase Shift Keying

signal in demodulation. Both cases use the same scheme for differential encoding.

c. Orthogonal Binary Phase Shift Keying (Orthogonal BPSK)

In Binary Phase Shift Keying there is always a probability, with the interjection of noise on the transmitted signal that an individual bit will be decoded incorrectly from that which was actually transmitted. In some environments, this probability is so high that special encoding techniques must be employed in order to reduce the probability that a bit or a binary code word will be decoded in error. One such technique is Orthogonal BPSK. Orthogonal BPSK is a type of group encoding scheme where groups or blocks of bits from the original signal are assigned redundant bits according to a predetermined assignment routine. This new expanded sequence of bits is then transmitted as before in BPSK.

For example, assume that it has been determined that the respective values of the original signal will be represented by a k bit binary code word. It has been demonstrated that there exists 2**k binary code words, k bits long in this type of arrangement. With Orthogonal BPSK there also exists 2**k orthogonal sequences of bits (called channel symbols, bauds or chips) [Ref. 6], which represent the k binary code words. However, the number of bits or chips in the sequence is increased to n = 2**k, thereby providing the desired redundancy. Each of the 2**k sequences of n chips is constructed to be orthogonal to every other sequence in order to assure maximum separation. See Figure 5.10 [Ref. 6].

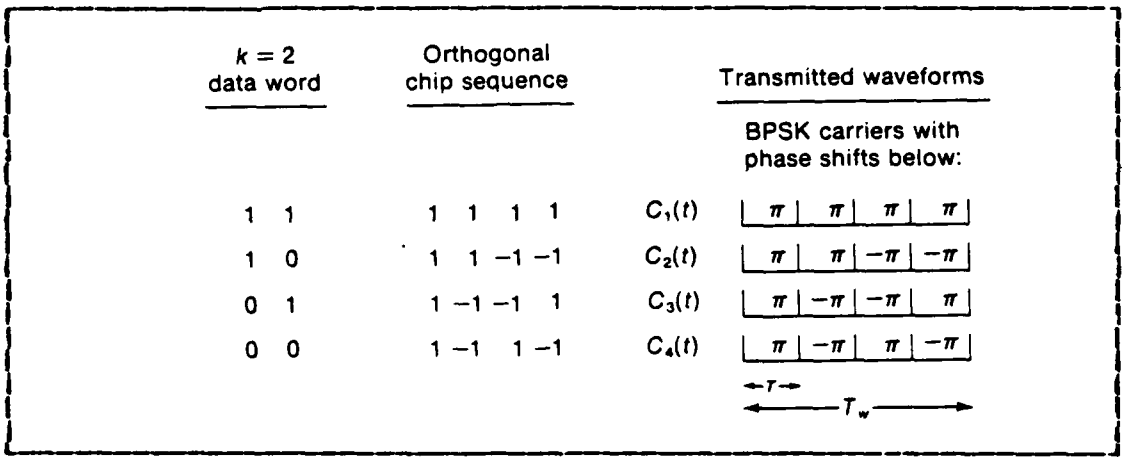


Figure 5.10 Orthogonal Binary Phase Shift Keying

Decoding is also done by blocks through a bank or 2**k correlators. Since the decoding is dore in blocks, the probability that the entire transmitted code word will be incorrectly decoded is reduced even though individual bits are incorrectly decoded. This is the major advantage of Orthogonal BPSK. Figure 5.11 [Ref. 6], shows the reduction in error probability with Orthogonal BPSK

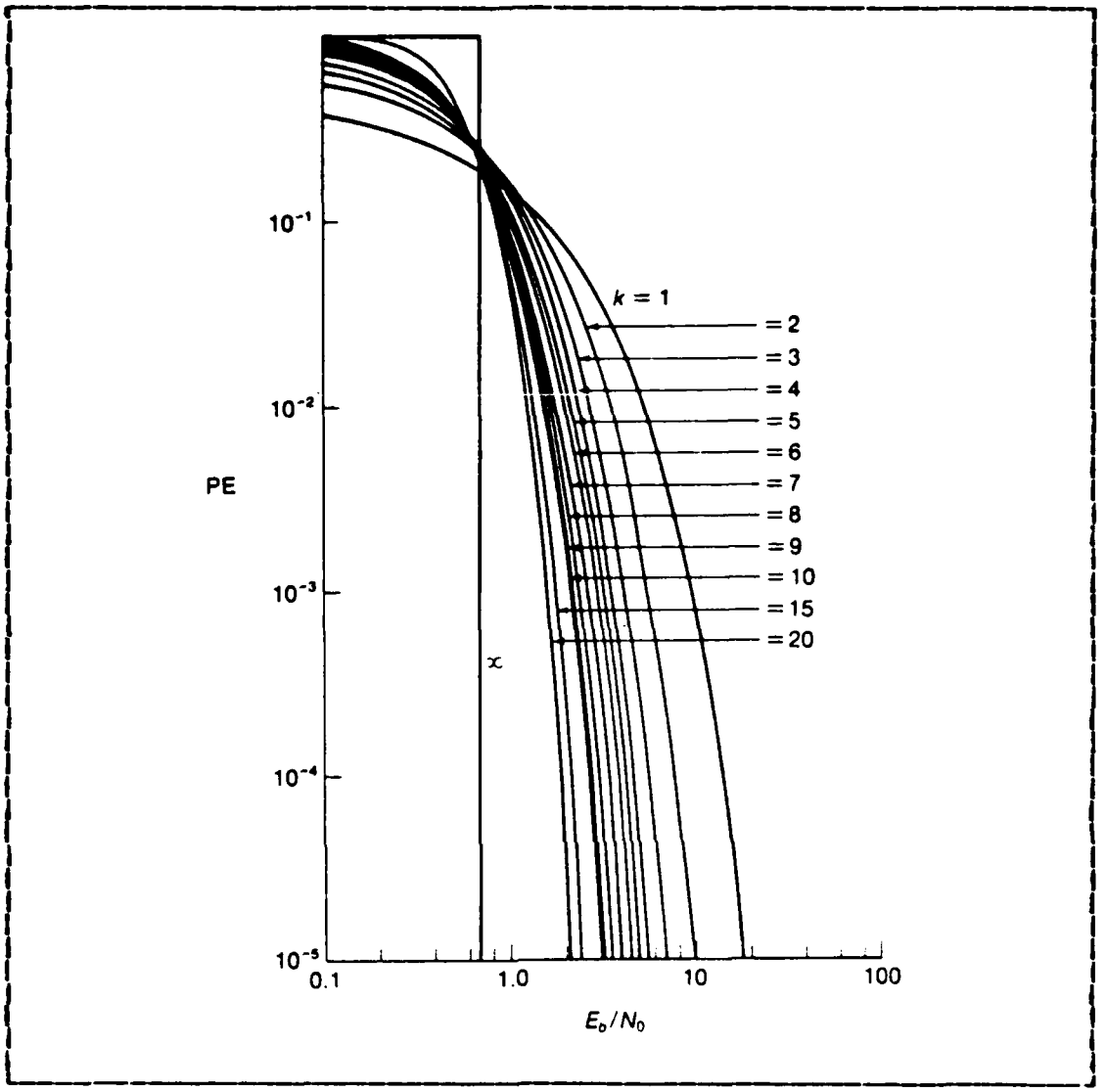


Figure 5.11 Probability of Error for Orthogonal BPSK

The major disadvantage to Orthogonal BPSK is that it results in a reduced data bit rate at a given transmission rate or the necessity for a higher transmission rate if the same bit rate is to be maintained. For example, suppose the bit rate of a PSK signal is R bits/sec. If each of the k bits in the PSK signal are represented by 2**k chips as is the case in Orthogonal BPSK modulation, then the

bit rate must be (2**k/k)R if the same data rate is to be maintained. Additionally, since the necessary bit rate increases, the bit duration decreases. It was shown in the section on bandwidth that as bit duration decreases as the result of a higher bit rate, an increase in bandwidth is the result. Therefore, the tradeoff in decreased transmission error comes at the expense of bandwidth and bit rate. [Ref. 6]

d. Quadrature Phase Shift Keying (QPSK)

Quadrature Phase Shift Keying derives its name from a four-phase modulation of the carrier waveform. These modulations are achieved by simultaneously modulating the inputs from two bit streams onto a single carrier. The two bit sequences could be from two separate sources or successive bits from a single source. To take advantage of the orthogonality of the sine and cosine functions, QPSK can be viewed as in equation 5.5 as the sum of a BPSK modulated sine and cosine.

$$vc(t) = A\ v1(t)\ \cos(wt + delta) + A\ v2(t)\ \sin(wt + delta) \qquad \text{(eqn 5.5)}$$

The signal can also be represented as in equation 5.6 where the phase angle, theta, of the modulated carrier is shown in equation 5.7.

$$vc(t) = 2^{**}.5\ A\ \cos(wt + theta + delta) \qquad \text{(eqn 5.6)}$$

$$theta = \arctan\ (v1(t)/v2(t)) \qquad \text{(eqn 5.7)}$$

The phases possible as a result of substitution of the possible bits in bipolar logic in equation 5.7 are ±45 degrees and ±135 degrees. See Figure 5.12.

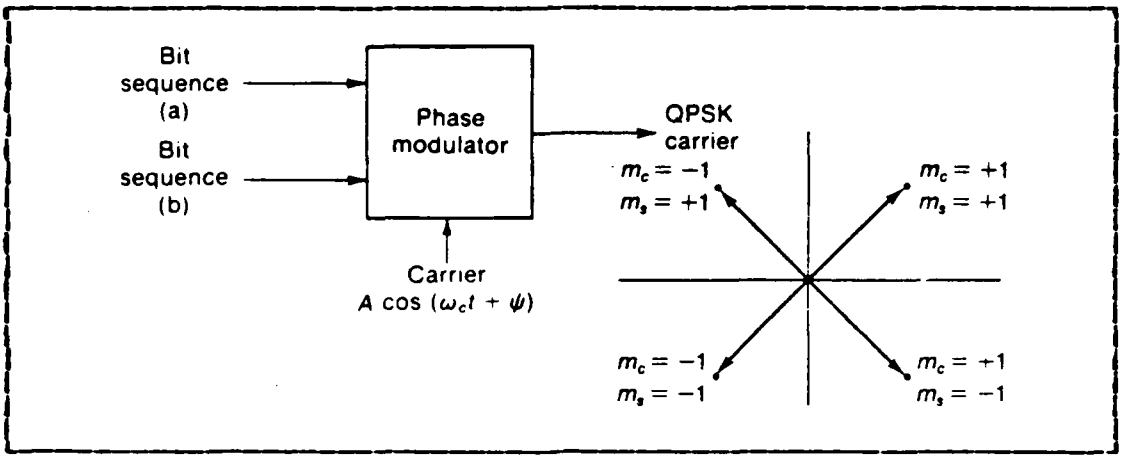


Figure 5.12 Modulation and Phases of QPSK

Since two bits are being encoded at one time for transmission, the bit rate in QPSK is twice that of BPSK. The BPSK bit rate is equal to the QPSK symbol rate, (symbol rate being the time the quadrature bits exist on the carrier). It is the symbol rate in QPSK which determines the bandwidth and since symbol rate is equal to bit rate in BPSK, the bandwidth of the two modulation techniques is identical. The great advantage of QPSK lies in the bandwidth utilization. Two bits of information are transmitted thereby effectively doubling the bit rate at no additional cost in bandwidth. A phasor representation of QPSK is illustrated in Figure 5.13. Each quadrature component of the modulated signal contains half the power of the total carrier or $(A^2/2)/2 = A^2/4$.

e. Offset Quadrature Phase Shift Keying (OQPSK)

In Quadrature Phase Shift Keying, both bits of the two baseband waveforms, which are to be modulated onto the carrier, change at the beginning of the respective symbol time. This allows the phase of the carrier to change up to 180 degrees for each symbol as illustrated in the

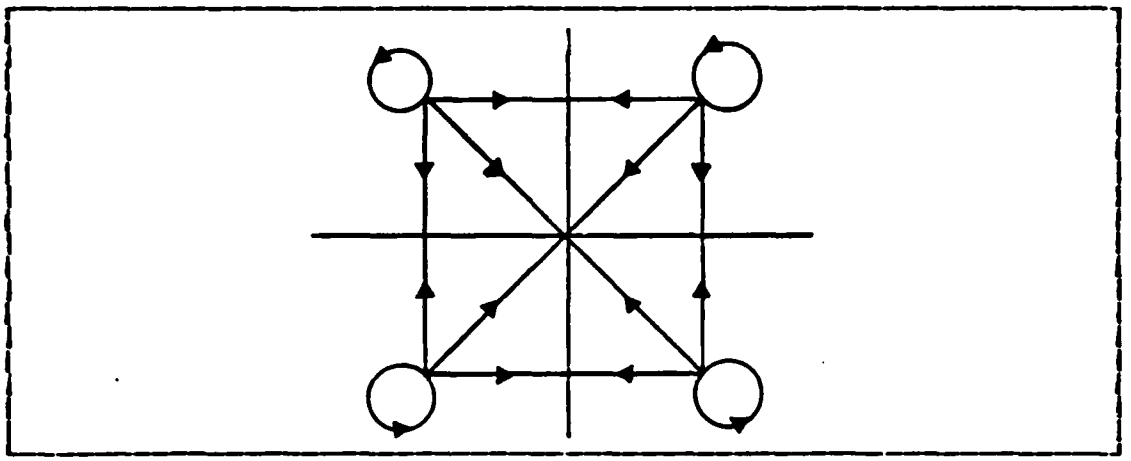


Figure 5.13 Phasor Diagram of QPSK

phasor diagram in Figure 5.13. The phase change of the modulated signal can be limited to 90 degrees through a modulation technique called Offset Quadrature Phase Shift Keying. This is accomplished by delaying the application of the second baseband channel for Ts/2 seconds, where Ts is symbol duration as illustrated in Figure 5.14. Modulation is then accomplished as before in QPSK.

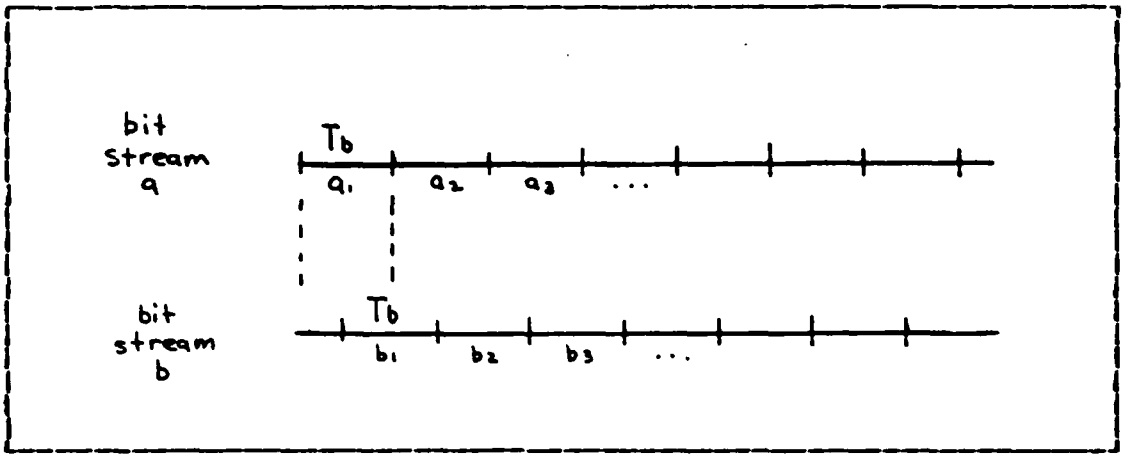


Figure 5.14 Offset Quadrature Phase Shift Keying

The result is a QPSK modulated signal in which the maximum phase shift between successive bits is 90 degrees. The bandwidth and power spectra of an OQPSK modulated signal are the same as that of a QPSK modulated signal. The advantage of OQPSK is in the limit which is placed on phase shift during encoding which simplifies the encoding hardware. Additionally OQPSK has spectral and interference advantages during decoding [Ref. 6].

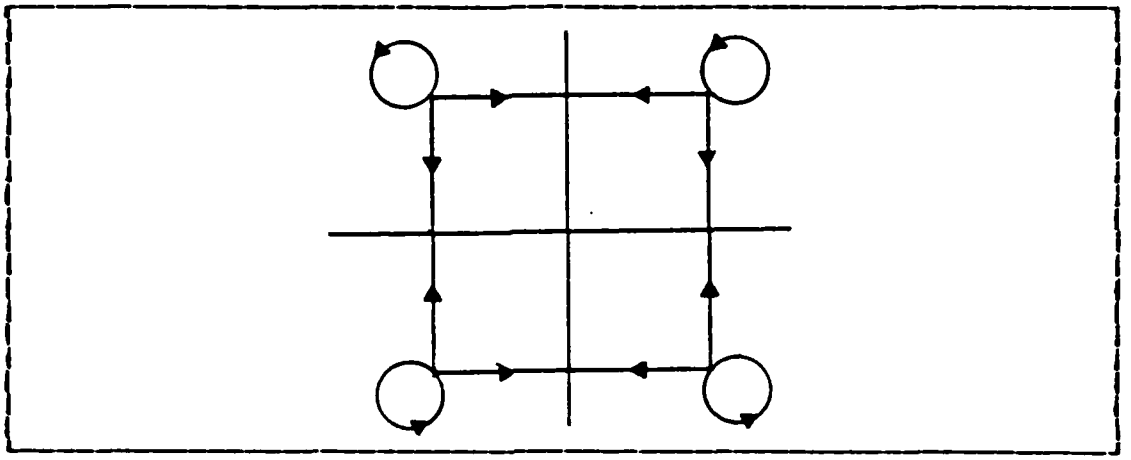


Figure 5.15 Phasor Diagram of OQPSK

f. Multiple Phase Shift Keying (MPSK)

Another form of digital modulation is known as Multiple or M-ary Phase Shift Keying (MPSK). Once again, as in Orthogonal Binary Phase Shift Keying, k data bits are transmitted. The k bits represent the word length and there are m = 2**k different binary words, k bits long, for this type of modulation. In MPSK, the binary code word is used to vary the phase of the carrier. There are therefore m = 2**k different phases in MPSK. These sequences of binary digits need not be orthogonal as in Orthogonal Binary Phase

Shift Keying. The length of the transmitted symbol is only k bits long as compared to 2**k bits for Orthogonal BPSK. The phases therefore are not all orthogonal leading to greater difficulty in decoding and a higher probability that an individual code word will be incorrectly decoded, especially as m becomes large.

As in other types of Phase Shift Keying, MPSK can be represented as in equation 5.8 where theta = phase = (2pi/m) i; i=0,1,...m-1.

$$vc(t) = A \cos (wt + theta + delta) \qquad \text{(eqn 5.8)}$$

An advantage of MPSK is that as long as the symbol transmission rate does not increase, the carrier bandwidth does not increase even as the size of the binary code word increases. The disadvantage, as stated earlier, has tc dc with error rates in decoding. Decoding requires a phase coherent reference of considerable stability especially as m increases and respective phases become closer together. Practical limits for this type of modulation is m = 8 or a 3 bit binary code word due to the complexity of the encoding and decoding hardware.

2. Amplitude Shift Keying (ASK)

Amplitude Shift Keying, as the name implies is modulation of the carrier through variation of its amplitude due tc variations in the baseband waveform. The information in the baseband waveform is thereby imparted to the carrier through this modulatich. Since the amplitude of the carrier varies between successive symbols or binary code words, the power also varies. Recovery is by comparison of the transmitted power during each symbol to the possible m = 2**k power levels.

a. Multiple Amplitude Shift Keying (MASK)

MASK is a type of group encoding where k bits are combined into a single waveform for transmission and subsequent recovery. Since each symbol or binary code word contains k bits, there are m = 2**k different symbols and consequently m different amplitudes of the carrier possible during the symbol duration time. An assignment scheme which maps the binary code words into the m different amplitudes would be to simply space them Ai = A/mi volts apart, where A is the maximum amplitude and mi represents the decimal equivalent of the binary code word from i = 1 to 2**k.

The analytic form of a MASK waveform is represented in equation 5.9 where all the variables are the same a presented earlier except for Ai which is equal to one of i different amplitudes depending on the bit sequence to be transmitted.

$$vc(t) = Ai \cos(wt + delta) \qquad \text{(eqn 5.9)}$$

MASK is not very popular in satellite communications since it depends on a carrier of very stable amplitude. This is not generally practical under actual conditions. However, MASK does have the spectral advantage of a constant bandwidth of 2/Ts, Ts equal to symbol duration, even as the binary code word increases in length.

b. Quadrature Amplitude Shift Keying (QASK)

Quadrature Amplitude Shift Keying (QASK) is a hybrid digital signal modulation technique. It represents separate amplitude modulation of the quadrature components of a common carrier. In that sense it is both ASK and PSK. Implementation of QASK involves simultaneous application of M-ary Amplitude Shift Keying to the quadratures. See

equation 5.10, where A1 and A2 are derived according to an assignment scheme as in MASK.

$$vc(t) = A1 \cos(wt + delta) + A2 \sin(wt + delta) \qquad \text{(eqn 5.10)}$$

By this method, the effective bit rate is doubled over that of ordinary MASK. In other words, QASK results in the same data rate as modulating 2k bits onto the carrier at the same time or during the same symbol duration with MASK. The advantage to QASK lies in the fact that this increase in bit rate comes at no further increase in bandwidth. The disadvantages associated with MASK are still present with an additional increase in the complexity of the decoding equipment. Decoding of QASK must be done in two steps. First the signal is recovered in phase through a phase-coherent correlator and then each amplitude modulated signal is recovered as before in MASK by comparison of the power levels of the received signal.

3. Frequency Shift Keying (FSK)

Frequency Shift Keying (FSK), is the generic term used to describe a number of digital signal modulation techniques which cause variations in the carrier frequency by interaction with the baseband waveform. Decoding is generally through measurement of the frequency of the received signal.

a. Mimimum Shift Keying or Fast Frequency Shift Keying

Power requirement for the transmission or retransmission of a satellite signal is extremely important. It is desireable to limit power required to accurately

transmit a piece of information to the minimum possible consistent with allowable error rates. For this reason, detection of FSK signals is limited to coherent methods as power required increases significantly for non-coherent methods.

MSK or FFSK are identical and represent a modulation technique known as continuous phase FSK. They get their names from the fact that "fast" indicates more bits per second can be transmitted in a given bandwidth than ordinary BPSK and "minimum" refers to the minimum modulation index for which orthogonal signalling occurs [Ref. 7].

Analytically, FFSK can be expressed as in equation 5.11 and equation 5.12.

$$vc(t) = A \cos((wc + deltaw)t) \qquad \text{(eqn 5.11)}$$

$$vc(t) = A \cos(\pm deltaw\ t) \cos(wc\ t) - A \sin(\pm deltaw\ t) \sin(wc\ t) \qquad \text{(eqn 5.12)}$$

As is noted, FFSK can be envisioned as the summation of separately modulated in-phase and quadrature components of the carrier by the baseband waveform. The technique is similar to that used in OQPSK. The frequency variation in the carrier waveform is between wc + deltaw and wc - deltaw. Maximum separation in phase of these two frequency components occurs at pi as noted in equation 5.13, where Tb represents the bit or symbol duration.

$$(w2-w1)Tb = pi \qquad \text{(eqn 5.13)}$$

This can be converted to the modulation index mentioned above simply by converting to frequency as in equation 5.14.

$$h = (f2-f1)Tb = .5 \qquad \text{(eqn 5.14)}$$

For FFSK, the frequency deviation or separation between frequencies of the transmitted carrier is exactly 1/2Tb and the deviation from the carrier frequency is 1/4Tb. FFSK can therefore be rewritten from equation 5.12 as in equation 5.15 by substitution for deltaw = 2pi(1/4Tb).

$$vc(t) = A \cos(\pm pi\ t/2Tb) \cos(2pi\ fc\ t) - A \sin(\pm pi\ t/2Tb) \sin(2pi\ fc\ t) \qquad \text{(eqn 5.15)}$$

The error rate in this modulation technique is identical to that of BPSK. Frequency Shift Keying can be implemented to incorporate any given modulation index by simple calculation of the frequency separation with equation 5.14 and substitution into equation 5.15. For M-ary Frequency Shift Keying, the separation between frequencies must be at least 1/Ts, where Ts is the symbol duration, in order to avoid carrier energy from one frequency being incorrectly interpreted as carrier energy from another frequency. This results in a modulation index of 1. Figure 5.16 represents the phasor diagram of an FFSK modulated signal.

4. Quadrature Partial Response Signalling (QPRS)

Quadrature Partial Response Signalling (QPRS) represents a specific type of the general class of digital signal modulation techniques called Partial Response Signalling. Partial Response Signalling is a method in which a controlled amount of intersymbol interference (ISI) is allowed during the encoding process. Previously any overlap in successive signals resulted in aliasing and an increased probability of error in decoding. The idea was

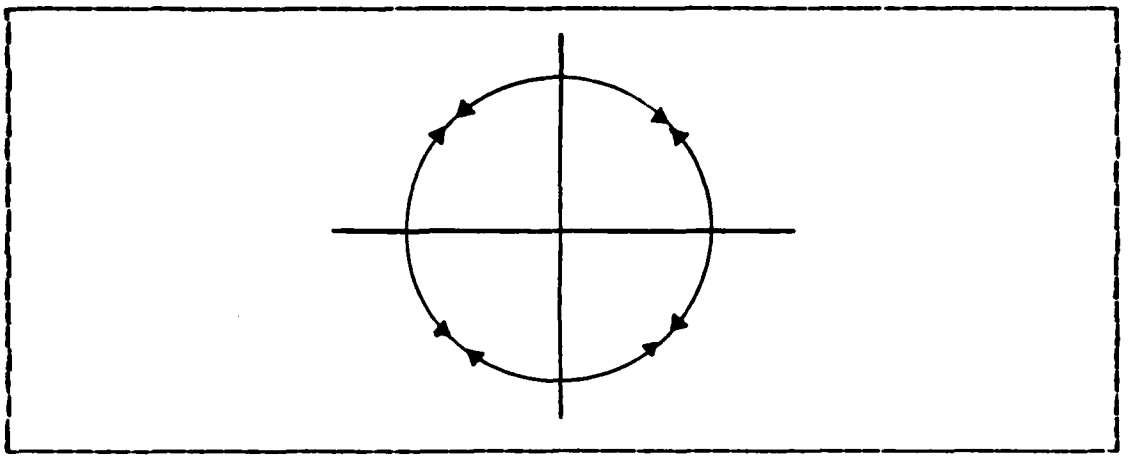


Figure 5.16 Phasor Diagram of FFSK or MSK

first introduced by Lender in 1963 [Ref. 12]. His concept was that in knowing and controlling the amount of interference allowed during the encoding process, compensations can then be made for the ISI at the receiver. Allowing for a limited amount of ISI makes it possible to transmit at rates equal to the Nyquist rate, something that is not possible with many other systems [Ref. 13].

QPRS is implemented, as with previous quadrature systems, through simultaneous modulation of the quadrature components of a common carrier. However, this time the modulation is accomplished with a PRS system. The modulation of the respective quadrature components represents the impulse response of one of a number of linear filters to the bits of the baseband waveform. The linear filter employed depends on the class of the Partial Response Signalling system being used. There are 5 classes of linear filters commonly used in PRS and they are illustrated analytically in Table 1 and graphically in Figure 5.17.

Output of a QPRS system can be represented as in equation 5.16, where an and bn represent the nth bit to be modulated and h(t - nTs) represents the contribution of the

TABLE 1

QPRS SYSTEM POLYNOMIALS

SYSTEM POLYNOMIAL P(D)	FREQUENCY RESPONSE H(ω) for $\|\omega\| \leq \pi/T$	IMPULSE RESPONSE h(t)
$1 + D$	$2T \cos \frac{\omega}{2} T$	$\frac{4T}{\pi} \frac{\cos(\pi t/T)}{T^2 - 4t^2}$
$1 + 2D + D^2$	$4T \cos^2 \frac{\omega}{2} T$	$\frac{2T^3}{\pi t} \frac{\sin(\pi t/T)}{T^2 - t^2}$
$2 + D - D^2$	$T + T \cos \omega T + jT \sin \omega T$	$\frac{T^2}{\pi t} \sin(\pi t/T) \left(\frac{3t-T}{t^2-T^2}\right)$
$1 - D^2$	$jT \sin \omega T$	$\frac{2T^2}{\pi} \frac{\sin(\pi t/T)}{t^2 - T^2}$
$1 - 2D^2 + D^4$	$-4T \sin^2 \omega T$	$\frac{8T^3}{\pi t} \frac{\sin(\pi t/T)}{t^2 - 4T^2}$

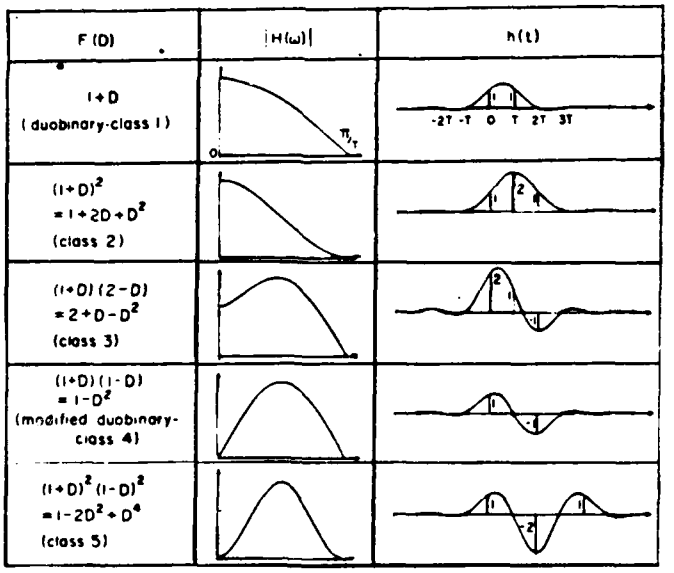


Figure 5.17 QPRS Linear Filter Impulse Responses

impulse response of the linear filter at time t for bit n. [Ref. 14]

$$v_C(t) = \sum_{n=-\infty}^{\infty} a_n\, h(t - n\,Ts)\, \cos(wt + delta) \qquad \text{(eqn 5.16)}$$
$$\sum_{n=-\infty}^{\infty} b_n\, h(t - n\,Ts)\, \sin(wt + delta)$$

QPRS has the advantage of rapid transmission rates with no increase in bandwidth and excellent error handling performance. The cost of such performance improvement comes in the form of a higher required signal to noise (SNR) ratio when compared to other binary systems.

VI. COMPUTER SIMULATION OF DIGITAL SIGNAL MODULATION TECHNIQUES

Due to the nature of digital signal modulation, it is possible to simulate systems through the use of the digital computer. This chapter is a description of the computer simulation included in the Appendix of this thesis.

A. GENERAL DESCRIPTION OF THE PROGRAM

The program was constructed using top-down design and modular programming. FORTRAN was selected as the programming language due to its capabilities in the area of numerical operations and the availability of mathematical routines to be used as modules in the program. Testing was accomplished on each module as the program was developed. The program was designed to be used interactively in order to allow for instructional use. Although the program was made user friendly in that error checking is accomplished on user inputs, care must be taken to insure instructions are followed correctly.

The program consists of twenty modules, sixteen of which were written by the author, three which were taken from the Double Precision International Mathematics and Statistics Library (IMSLDP) and one which was taken from a NON-IMSL library which resides on the IBM 3033 located at the Naval Postgraduate School. The IBM 3033 was used exclusively for development and testing.

The development was accomplished using the WATFIV compiler; however, the program was written to operate with the VS FORTRAN compiler as well. Testing and operation was done using the VS FORTRAN compiler. Figure 6.1 is a block

diagram of the relationship of the modules for a representative modulation subroutine.

B. MAIN CONTROL MODULE

MAIN operates as the control module for the entire computer simulation and accomplishes limited output. It introduces the program and allows the user to input various parameters of the digital signal modulation technique to be simulated and various other control functions. These inputs include:

1. Modulation technique
2. Class of QPRS system (when appropriate)
3. Digital logic scheme
4. Baud or symbol rate
5. Bits/binary code word (when appropriate)
6. Carrier frequency
7. Number of samples to be generated
8. Carrier max amplitude
9. Initial phase angle
10. Use of random number generator or no
11. Seed for RNG (when appropriate)
12. Number of repetitions of the simulation

Once the user has input the desired characteristics of the modulation to be simulated, MAIN calls the appropriate module to begin the actual calculation. The values necessary to calculate the time series of the signal are passed to the subroutines as parameters. MAIN calls the required subroutine the number of times the user specifies as repetitions. Each call to a subroutine produces the Discrete Fourier Transform and amplitude spectrum of the signal. It is these respective amplitude spectrums which produce the statistics in the final output after MAIN calls the statistics subroutine the final time.

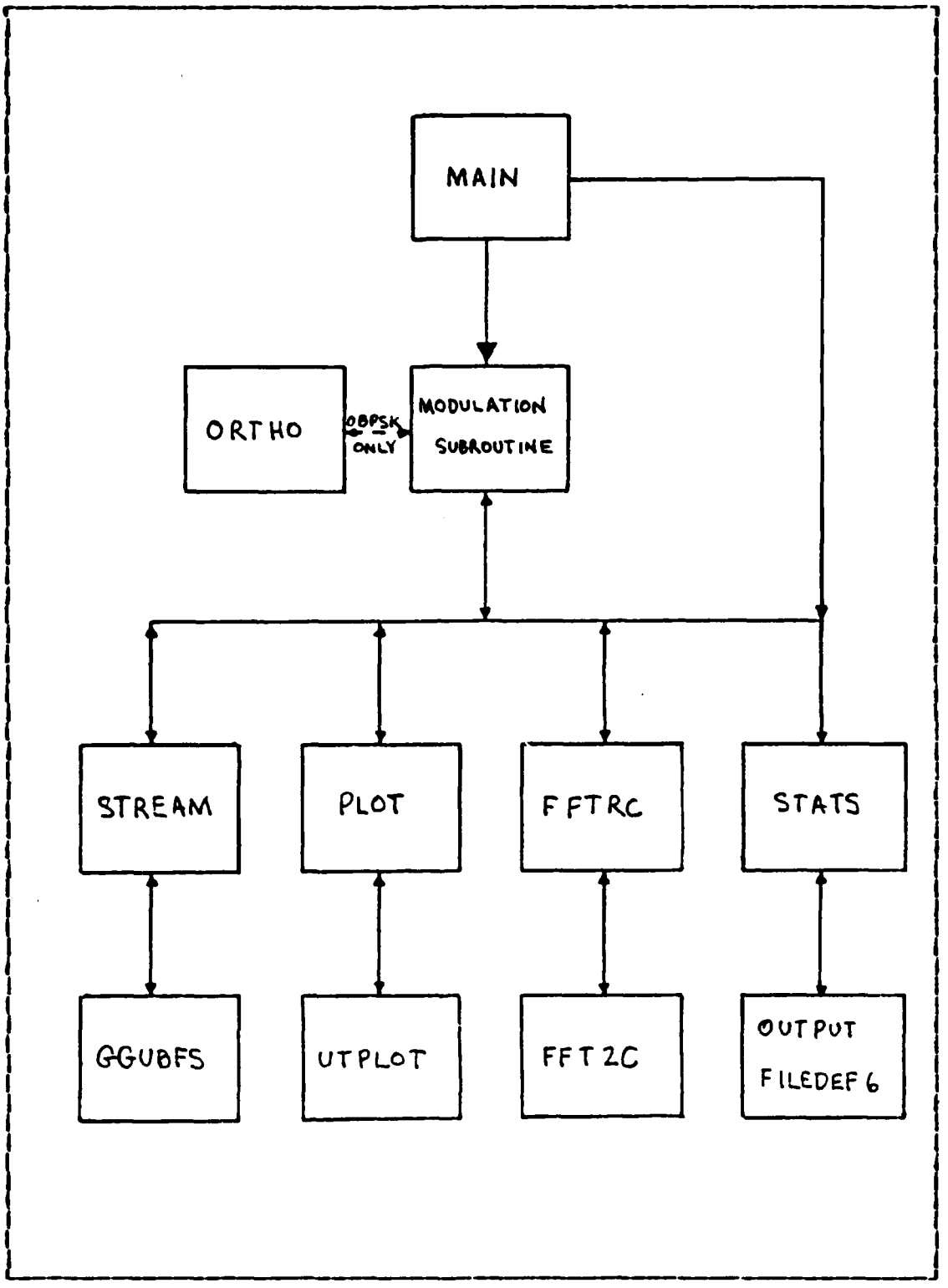


Figure 6.1 Representative Block Diagram

C. SUBROUTINE BPSK

This description of SUBROUTINE BPSK, a module that simulates Binary Phase Shift Keying, will serve as a guide on the construction and operation of subsequent modules involved in the actual signal modulation. A complete description including its interaction with other program subroutines will be included. Since other signal modulation subroutines interact in the same manner within the program, subsequent descriptions will concentrate on differences to this basic module.

SUBROUTINE BPSK begins with variable declarations as with all FORTRAN programs. All variables are declared to be either type integer or double precision. All calculations in the program are carried out in double precision accuracy.

Variable initialization is next. Significant variable initialized at this point are TIME, the time of the first sample which is set to 0. STEP, the delta time between samples is set to the normalized Nyquist sampling rate. OMEGA, the angular frequency and DELTA, the initial phase angle are computed in radians. NBAUD, a variable which keeps track of the number of symbols which have been modulated, is set to 1 or the first symbol is being modulated. NBAUD remains at its present value until total elapsed time exceed 1 symbol duration or BAUDD, the normalized baud rate.

Subsequently, an initial call is made to SUBROUTINE STREAM to receive the value of the first bit to be modulated. The modulation is then carried out according to equation 6.1.

The values of each sample are assigned to an array for storage along with the corresponding time increment. Modulation continues with the first drawn bit until 1 bit or symbol duration is exceeded when STREAM is called to draw

$$vc(t) = A\ v(t)\ \cos(wt + delta) \qquad \text{(eqn 6.1)}$$

A = amplitude
v(t) = bit to be modulated
w = angular frequency
t = time
delta = initial phase angle

another bit. This entire process is repeated until the number of samples specified by the user in MAIN is reached.

At this point, only on the first repetition of the simulation, SUBROUTINE PLOT is called which gives the user the opportunity to plot the time series of the simulated signal. Upon return from PLOT the subroutine calls the IMSL routine FFTRC to generate the Discrete Fourier Transform of the time series produced. This is accomplished on each repetition of the program.

Again on the first repetition only, information on the number of the principle harmonic of the FFT is displayed. Plot is called again to give the user the opportunity to plot the amplitude spectrum. The values of the FFT and the amplitude spectrum are computed on each successive repetition of the program and the amplitude spectrum is added to the statistics being accumulated by a call to SUBROUTINE STATS.

D. SUBROUTINE DBPSK

SUBROUTINE DBPSK simulated Differential Binary Phase Shift Keying and is essentially the same as SUBROUTINE BPSK. The difference lies in the fact that what is encoded during the modulation process is the product of the random bit drawn and the value of the bit previously modulated. A

reference bit equal to +1 is used to determine the first bit to be modulated. In this manner, modulation of +1 means no change occurs between successive bits and the phase of the carrier remains the same. Modulation of a -1 indicates a change of bits has occurred and is indicated by a change of phase of the carrier between successive bit durations. The rest of the module remains unchanged.

E. SUBROUTINE OBPSK

Orthogonal Binary Phase Shift Keying is accomplished in this module. Again SUBROUTINE OBPSK is constructed essentially the same as SUBROUTINE BPSK. In Orthogonal BPSK the only bit streams that are modulated are n bits long representing each binary code word, where n = 2**k (k = number of bits per word is limited to 6 bits in MAIN). Each sequence of n bits is orthogonal to every other allowed sequence.

The way this is accomplished in this module is first to draw a series of k random bits from SUBROUTINE STREAM. This series of k 1's and 0's is then changed to its decimal equivalent from 0 to 2**k-1 and saved for later use.

The program continues with a call to SUBROUTINE ORTHO which generates an n X n orthogonal matrix of +1's and -1's through the use of the Hadamard matrix and the Kronecker product [Ref. 15]. The Hadamard matrix is a 2 X 2 orthogonal matrix of 1's and -1's. The Kronecker product is the matrix which is formed when a matrix is expanded to twice its original size by replacing each element of the Hadamard matrix by the product of the Hadamard matrix and the original matrix. The value of the decimal equivalent to the k random bits is used to identify the row of the matrix and the respective columns represent the value of the binary digit which is modulated onto the carrier by BPSK. This

sequence continues until either one bit duration is exceeded, when another orthogonal bit is received from ORTHO or when all the orthogonal bits in a row are modulated. Then another sequence of k random bits is converted to its decimal equivalent and a new row and column of the orthogonal matrix is identified. Execution continues in this manner until all required samples have been produced. It should be noted that the bit rate for Orthogonal BPSK is n times the baud rate. This significantly increases the signal bandwidth requirement and decreases the time increment between successive samples in the simulation. The rest of the module is the same as those previously presented.

F. SUBRCUTINE QPSK

Quadrature Phase Shift Keying, simulated in SUBROUTINE QPSK, is accomplished through simultaneous BPSK modulation of the quadrature components of the carrier. Two random bits are drawn from successive calls of SUBROUTINE STREAM and each bit is in turn modulated onto the carrier components and the sum formed. This process is repeated at time intervals equal to the normalized Nyquist sampling rate until TIME exceeds one symbol duration, BAUDD. At this time two new random bits are drawn and the modulation continues until all required samples are produced. The rest of the modules is the same as those before. Equation 6.2 represents the analytical expression used to simulate

G. SUBROUTINE OQPSK

SUBROUTINE OQPSK is essentially the same as SUBROUTINE QPSK with these minor deviations. The time of the first sample is artificially initialized as TIME = .5(BAUDD) or 1/2 the symbol duration. In this manner, the two random

$$vc(t) = A\ v1(t)\ \cos(wt + delta) + A\ v2(t)\ \sin(wt + delta) \qquad \text{(eqn 6.2)}$$

v1(t) = in-phase bit
v2(t) = quadrature bit

bits first being modulated are both known to have existed on the carrier for 1/2 the symbol duration. The first of thest two bits only remains on the carrier until TIME = 1(BAUDD) when a third random bit is drawn and modulation begins with this bit. The second random bit is allowed to remain until TIME = 1.5(BAUDD) when a fourth random bit is drawn to replace it and so on. In this manner the two bits being modulated are offset in the time they change by 1/2 the symbol duration. This necessitates keeping track of the number of bits modulated on both the in-phase and quadrature component of the carrier. The remaining portion of the module is the same as SUBROUTINE QPSK.

H. SUBROUTINE MPSK

This module simulates M-ary Phase Shift Keying. This is accomplished by changing the phase of the carrier to any one of n =2**k phases where k is the number of bits in the binary code word. The max length of the binary code word is limited to 10 in the MAIN program. This modulation is accomplished by use of equation 6.3 to simulate the MPSK system.

The program first draws k random bits and converts them to the decimal equivalent. At this point modulation begins at TIME = 0 according to equation 6.3. Modulation continues until one symbol duration is exceeded when a new set of k random bits is drawn, conversion to decimal takes place and

$$vc(t) = A \cos[wt + (2\pi m)/n + delta] \qquad \text{(eqn 6.3)}$$

m = decimal equivalent of binary code word
n = 2**k; k = bits in binary code word

modulation continues or until all desired samples have been produced. The program continues as in previous subroutines.

I. SUBROUTINE MASK

The principles of operation of SUBROUTINE MASK is similar to SUBROUTINE MPSK which also involves a group encoding system. SUBROUTINE MASK allows for M-ary Amplitude Shift Keying. The analytic expression used in the simulation is provided in equation 6.4 Ai is one of 2**k equally spaced amplitudes.

$$vc(t) = Ai \cos(wt + delta) \qquad \text{(eqn 6.4)}$$

Once again a set of k random bits is drawn from SUBROUTINE STREAM. The decimal equivalent of the k bits is computed and used to adjust the amplitude Ai according to the assignment routine expressed in equation 6.5.

$$Ai = A/m \qquad \text{(eqn 6.5)}$$

A = max amplitude
m = 1 to n; n = 2**k

This process is repeated at each symbol duration until all desired samples have been produced. The rest of the module remains unchanged.

J. SUBROUTINE QASK

SUBROUTINE QASK combines the elements of the previously described amplitude modulation technique with the now familiar quadrature modulation technique. Two separate streams of random numbers are generated and converted to their decimal form. These numbers are used to produce two separate amplitudes with the same assignment scheme used in MASK. These amplitudes modulate the quadrature components of a common carrier and the output is formed by the sum of the quadratures. Modulation continues in the above manner until a symbol duration elapses when two new amplitudes are calculated. Processing terminates when all required samples are computed.

K. SUBROUTINE MSK

Minimum Shift Keying is frequency shift keying in which the modulation index is 1/2. The modulation index is represented in equation 6.6

$$h = .5 = (\text{deltaf})\,Tb \qquad \text{(eqn 6.6)}$$

deltaf = frequency separation
Tb = bit duration

The frequency deviation from the carrier is .5(deltaf) = 1/4Tb. Converting this frequency deviation to radians yields pi/2Tb. This angular frequency deviation from the central carrier is either ±pi/1Tb depending on the value of the bit to be modulated.

The simulation in SUBROUTINE MSK is accomplished by drawing a random bit and generating the signal according to equation 6.7.

$$vc(t) = A \cos[(w \pm pi/2Tb)t + delta] \qquad \text{(eqn 6.7)}$$

Once again modulation continues until one bit duration is elapsed when another bit is drawn. Modulation stops when all required samples are produced. The rest of the subroutine remains unchanged.

L. SUBROUTINE MFSK

This subroutine modulates a signal simulating M-ary Frequency Shift Keying. The separation between adjacent frequencies is established as 1/Ts, where Ts is the symbol duration. This amounts to a modulation index equal to 1. Initially the entire range of frequency deviation from the carrier is calculated as (n-1)Ts. The mean frequency deviation is then calculated. In other words the range of frequencies of the modulated signal is the frequency of the central carrier ±1/2 the magnitude of the entire range of frequency deviation. The minimum frequency is used as the base for computing the frequency to be used to modulate the signal during each respective symbol duration.

At this point in the subroutine a series of k random bits ($k \leq 10$) is drawn and converted to decimal. This decimal number is multiplied by the frequency separation and added to the minimum frequency and converted to radians. The analytical expression of the signal produced with this subroutine is illustrated in equation 6.8.

$$vc(t) = A \cos[(w + 2pi\ mf)t + delta] \qquad \text{(eqn 6.8)}$$

mf = modulation frequency

This modulation continues until a symbol duration elapses when a new frequency deviation or mf is calculated

or mcdulation stops do to completion of all required samples.

M. SUBROUTINE QPRS

This mcdule simulates 5 different classes of Quadrature Partial Response Signalling systems. The technique employed involves the summation, at each time increment, of all responses to the linear filter representing the class. This is accomplished for all of the bits being modulated during the duration of the simulation.

Initally, two arrays of random bits are generated representing those bits which would be modulated onto the quadrature components of the carrier during the length of time the signal is to be simulated. The the impulse response to each of these n bits is calculated for the time of the respective sample. These impulse responses represent the modulation for the bit in question onto the respective porticns of the carrier and the sum is formed according to equation 6.9.

$$vc(t) = \sum_{n=-\infty}^{\infty} A\, h(t - nTs)\, \cos(wt + delta) + \sum_{n=-\infty}^{\infty} A\, h(t - nTs)\, \sin(wt + delta) \qquad \text{(eqn 6.9)}$$

h(t - nTs) = impulse response of nth bit at time t

The time of the first sample is initialized at 6(BAUDD) in order to ensure that when the first sample is taken, contributions from bits modulated at time 0 are also summed do to the overlapping nature of the QPR signals. The value of the last sample is also formed by the sum of the quadrature components existing at 6(BAUDD) after it is produced. The rest of the program operation remains the same as previous modules.

N. SUBROUTINE STREAM

This subroutine interacts with the IMSL random number generator SUBROUTINE GGUBFS to produce a random number between 0 and 1 and make assignment on the basis of the value of this random number to random bits according to the digital logic scheme specified in the MAIN program. It also enables the user to manually insert binary digits if that option was specified previously in MAIN.

O. SUBROUTINE ORTHO

CRTHO produces an n X n orthogonal matrix from the Hadamard matrix by forming successive Kronecker products [Ref. 15]. It also selects the appropriate row and column of the orthogonal bit to be modulated and passes this bit back to SUBROUTINE OBPSK. Point of entry to SUBROUTINE ORTHO is controlled by a flag set in OBPSK.

P. SUBROUTINE PLOT

PLOT is an interactive module that allows the user to determine whether or not a graph of the output of program calculations is to be produced. The user is also able to selectively vary certain parameters associated with the plot. PLOT calls SUBROUTINE UTPLOT which actually produces the graph and performs the output. UTPLOT is a NON-IMSL library routine from the Naval Postgraduate School.

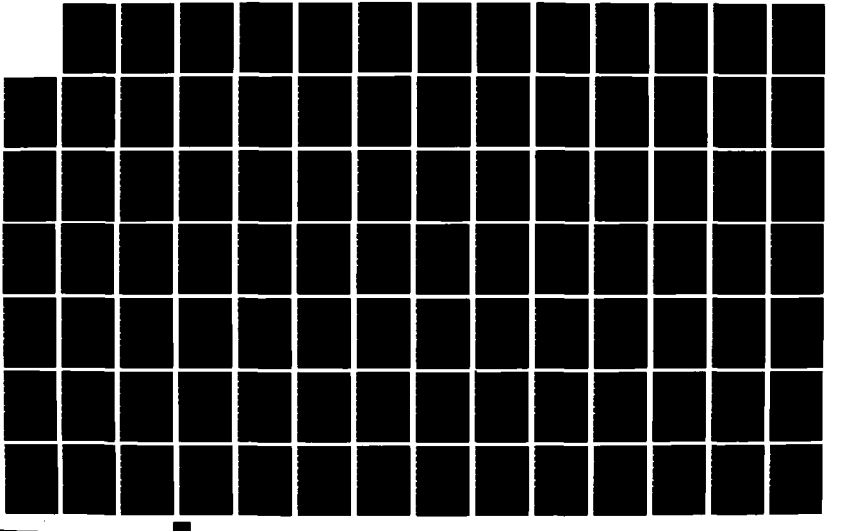
Q. SUBROUTINE STATS

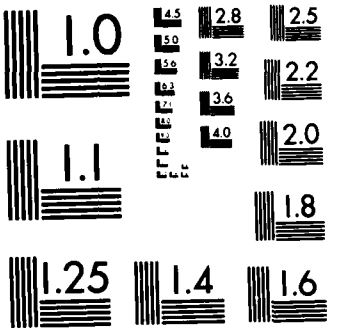
The statistical calculations associated with the amplitude spectrums of the various functions generated on successive repetitions of the program are computed in SUBROUTINE STATS. Upon completion of each repetition, each module calls STATS where the sum, sum of the squares, sum of

the cubes and sum of the quartics of each element of the amplitude spectrum of the signal produced are computed and stored. When the last repetition of the program is complete, MAIN instructs STATS to compute and output the final statistics, i.e., mean, variance, skewness and kurtosis of each component of the amplitude spectrum according to the estimations contained in [Ref. 16]. Additionally the mean variance and variance of the variance is calculated and output. Point of entrance to this subroutine is controlled by flags set in the individual modulation subroutines or in MAIN.

AD-A160 823 COMPUTER SIMULATION OF DIGITAL SIGNAL MODULATION TECHNIQUES IN SATELLITE COMMUNICATIONS(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA C D CARLSON SEP 85 2/4

UNCLASSIFIED F/G 9/2 NL





VII. STATISTICS OF THE FAST FOURIER TRANSFORMS

A. DESCRIPTION OF THE PROBLEM

As stated in Chapter II, if it can be shown that the statistics of the FFT can be somehow linked to a particular digital signal modulation technique, then the modulation technique can be identified on the basis of those statistics alone. Although it is beyond the scope of this thesis to actually derive those relationships if they exist, it seem prudent to attempt to determine if there is statistical differences between the components of the FFT's as they are derived in the enclosed computer simulations.

The statistic chosen to do the hypothesis testing is the F-test since the true mean and variance of the distribution need not be known [Ref. 17]. In addition, the information necessary to calculate the F-statistic is readily accumulated in SUBROUTINE STATS. Calculation of the F-statistic necessitated the writing of a small computer program to be used for that purpose once the output from the main program was generated and reformatted.

B. ANALYSIS-OF-VARIANCE

If there were no differences attributable to the modulation technique, then it would be reasonable to assume that for a certain set of characteristics, the components of successive FFT's would be the same. In other words, if a signal was modulated by BPSK and a statistically significant number of FFT's were generated of the signal, then the statistics of those FFT's would be assumed to be related. On the other hand, if it can be shown that the statistics are not related (i.e., not from the same distribution), then

significance can be placed in the variation among modulation techniques.

1. The F-Distribution

The assumptions necessary to use the F-distribution are:

1. Normal distribution
2. Random samples
3. Independent samples

These assumptions are not too difficult to intuitively accept in the model that will be proposed. Once again it would be expected that the FFT's of successive identically modulated signals to be related. For the simulations conducted as part of this test, they were all generated using bits from a random number generator for which it can be shown that each sequence of bits passes statistical tests for randomness and independence. The assumption of normality could also be argued on the basis of the central limit theorem.

The value of the F-statistic with which comparison will be made is 1.51 for 14 degrees of freedom in the numerator, an infinite number of degrees of freedom in the denominator and a 90% confidence region. How these values were obtained will be detailed under the design of the experiment.

2. Design of the Experiment

The experimental data used in the computation of the statistics was chosen to minimize the random contributions of variables other than the modulation technique. The computer simulations included in this thesis were used to generate the statistics and a separate program also included in the appendix was used to calculate the F-statistic. The variables in the experiment include:

1. Modulation technique
2. Logic type
3. Baud rate
4. Bits per binary code word
5. Carrier frequency
6. Carrier amplitude
7. Initial phase angle
8. Number of samples generated
9. Time between samples
10. Seed for random number generator
11. Number of repetitions or trials

Fifteen different modulation techniques were compared. In all cases bipolar logic was simulated, the baud rate was held constant at 1200 baud and the maximum carrier amplitude was established at 1 volt. Additionally the phase angle of all simulations was 0 degrees and the seed for the random number generator was 1 thereby ensuring the same random sequence of bits. The number of bits per binary code word did vary among the modulation techniques. When the modulation technique was M-ary, the bits per code word was always 3 so it is possible to infer that bits per code word is a function of the modulation technique. The time between samples was the normalized Nyquist sampling rate. This did vary between modulation techniques but was necessary to derive an accurate FFT. Carrier frequency also varied between simulations but was always twice the lowest carrier frequency recommended in the computer simulation. In most cases this was 2400 Hz, or twice the baud rate. Finally each simulation was repeated 100 times and the statistics of the FFT's based on those 100 repetitions. The number of samples produced was always 64 so there are 33 components of the FFT.

What this amounts to can be illustrated through an Analysis-of-Variance table as shown in Table 2 [Ref. 17].

TABLE 2

ANALYSIS-OF-VARIANCE TABLE

n Observations in Each of *r* Groups

		Observations					Sum	Mean
Group	1	Y_{11}	Y_{12}	Y_{13}	.....	Y_{1n}	$Y_{1.}$	$\overline{Y}_{1.}$
	2	Y_{21}	Y_{22}	Y_{23}	.....	Y_{2n}	$Y_{2.}$	$\overline{Y}_{2.}$
	3	Y_{31}	Y_{32}	Y_{33}	.....	Y_{3n}	$Y_{3.}$	$\overline{Y}_{3.}$
	⋮							
	r	Y_{r1}	Y_{r2}	Y_{r3}	.....	Y_{rn}	$Y_{r.}$	$\overline{Y}_{r.}$

In the case of the simulation described above, this amounts to 100 observations (repetitions or trials) from each of 15 groups (modulation techniques) for each of the 33 components of the FFT. The degrees of freedom in the numerator is equal to (15-1) or 14 while the degrees of freedom in the denominator is equal to 15(100-1) or 1485. Table values for the F-distribution use infinity as the degree of freedom for values greater than 120.

An F-test was also performed to determine if there is a relationship among the mean variance, mean skewness or mean kurtosis of the 15 different modulation techniques. Again the assumption is made that these statistics would be normally distributed from modulation techniques which had the same or statistically similar components of their FFT's. In this case there were again 15 modulation techniques to compare (14 degrees of freedom) but only 33 observations (the variance, skewness or kurtosis of the respective elements of the FFT). This number of observations yields 15(33-1) = 480 degrees of freedom in the denominator.

3. Hypothesis and Hypothesis Testing

The hypothesis posed for the model is that the means of the individual components of the FFT's from the 15 modulation techniques are from the same distribution. Therefore, the respective means must be equal. Also tested is that the mean variance, mean skewness and mean kurtosis of all components of the FFT's of each modulation technique are equal. These hypothesis are illustrated in equations 7.1 through equation 7.4. Each hypothesis is tested and compared to the F-distribution selecting a rejection region or level of significance of .9. This equates to an F-statistic of 1.51 for 14 degrees of freedom in the numerator and an infinite number of degrees of freedom in the denominator.

4. Results

The means of the 33 components of the FFT for each of the 15 modulation techniques were compared using the F-distribution. The results are shown in Table 3. Table 4 shows the F-statistics associated with the comparison of the mean of the variance, mean of the skewness and mean of the kurtosis for the respective groups. In addition a complete summation of the results of the F-test are included in the appendix.

C. CONCLUSION

The results of the F-test indicate that the null hypothesis must be rejected at the .9 significance level and the alternate hypothesis accepted that the means are not equal for FFT components 9-14. In addition the F-statistic for the comparison of mean variance, mean skewness and mean kurtosis all fall in the rejection area.

Ho: XBAR11 = XBAR21 = • • • = XBARij (eqn 7.1)

H1: XBAR11 ≠ XBAR21 ≠ • • • ≠ XBARij

i = ith modulation technique
j = jth component of the FFT

Ho: MVAR1 = MVAR2 = • • • = MVAR15 (eqn 7.2)

H1: MVAR1 ≠ MVAR2 ≠ • • • ≠ MVAR 15

H0: MSKEW1 = MSKEW2 = • • • = MSKEW15 (eqn 7.3)

H1: MSKEW1 ≠ MSKEW2 ≠ • • • ≠ MSKEW15

Ho: MKUR1 = MKUR2 = • • • = MKUR15 (eqn 7.4)

H1: MKUR1 ≠ MKUR2 ≠ • • • ≠ MKUR15

This indicates that there is a difference between the statistics of of the FFT's of the respective modulation technique. Since all the parameters used in the generation of these statistics were held essentially constant among the trials, it can be assumed that these differences are due to the modulation technique employed. It has not been established yet what those differences may be. This is an area where future research and study is warranted.

The results of this test would seem to point to those components of the FFT which offer the best opportunity to develcp those relationships or differences. FFT components 9-14 have obvious differences; however, FFT components 20, 24 and 25 also have large F-statistics but do not fall in the rejection area. These components may also be

TABLE 3

F-STATISTICS OF FFT COMPONENTS

FFT COMPONENT	AMONG GROUP SUM	WITHIN GROUP SUM	TOTAL OF SQUARES	F-STAT
1	0.121	17.69	17.81	0.727
2	0.112	18.48	18.59	0.642
3	0.108	20.65	20.76	0.557
4	0.135	25.40	25.54	0.562
5	0.172	32.74	32.91	0.558
6	0.254	41.72	41.98	0.647
7	0.422	50.10	50.52	0.893
8	0.962	82.83	83.79	1.231
9	7.591	390.2	397.8	2.063
10	49.09	1681.0	1730.1	3.097
11	53.01	1732.1	1785.1	3.246
12	61.56	2071.7	2133.3	3.152
13	45.73	1832.8	1878.6	2.646
14	3.864	250.5	254.4	1.636
15	0.539	98.13	98.67	0.582
16	0.366	110.1	110.5	0.353
17	0.504	151.0	151.5	0.354
18	0.943	140.3	141.2	0.713
19	1.299	158.5	159.8	0.870
20	2.671	200.7	203.4	1.412
21	1.853	226.3	228.2	0.868
22	2.277	232.1	234.3	1.041
23	1.712	197.6	199.3	0.919
24	2.512	181.9	184.4	1.476
25	2.144	171.3	173.5	1.327
26	1.257	143.7	145.0	0.928
27	1.233	113.9	115.1	1.149

TABLE 3
F-STATISTICS OF FFT COMPONENTS (con't)

28	0.858	115.8	116.7	0.785
29	0.547	63.72	64.27	0.911
30	0.356	55.77	56.13	0.677
31	0.183	41.41	51.59	0.469
32	0.051	13.79	13.84	0.389
33	0.031	10.92	10.96	0.304

TABLE 4

F-STATISTICS OF THE MEAN VARIANCES, MEAN SKEWNESS AND MEAN KURTOSIS

	AMONG GROUP SUM	WITHIN GROUP SUM	TOTAL OF SQUARES	F-STAT
VARIANCE	3.144 E5	3.755 E6	4.070 E6	2.870
SKEWNESS	5.307 E10	7.031 E11	7.562 E11	2.588
KURTOSIS	8.280 E11	1.262 E13	1.344 E13	2.250

interesting to examine. In addition, it should be noted that the statistics of the FFT associated with each respective modulation technique also display some striking differences. These statistics may prove in some way to be a fingerprint of the modulation techniques themselves.

APPENDIX A

FORTRAN PROGRAM FOR DIGITAL SIGNAL MODULATION

```
C     ************************************************************************MAI00020
C     *                                                                      *MAI00030
C     * TITLE:  DIGITAL SIGNAL MODULATION TECHNIQUES                         *MAI00040
C     *                                                                      *MAI00050
C     * PURPOSE:  THIS PROGRAM GENERATES A SIMULATED WAVEFORM WHICH IS       *MAI00060
C     * CHARACTERISTIC OF ONE OF A VARIETY OF DIGITAL SIGNAL MODULATION      *MAI00070
C     * TECHNIQUES.  THE PROGRAM IS INTERACTIVE AND DESIGNED TO BE AS        *MAI00080
C     * USER FRIENDLY AS POSSIBLE.  THE PROGRAM ALLOWS THE USER TO           *MAI00090
C     * SPECIFY ANY ONE OF A NUMBER OF PARAMETERS OR VARIABLES WHICH         *MAI00100
C     * ARE CHARACTERISTIC OF THE INDIVIDUAL MODULATION TECHNIQUE.           *MAI00110
C     * THE PROGRAM ALSO ALLOWS THE USER TO DEVELOP A PLOT OF THE            *MAI00120
C     * SIMULATED WAVEFORM, GENERATE THE FAST FOURIER TRANSFORM,             *MAI00130
C     * OR DO LIMITED STATISTICAL ANALYSIS.                                  *MAI00140
C     *                                                                      *MAI00150
C     * WRITTEN BY:  CRAIG CARLSON, LCDR, USN                                *MAI00160
C     * IN PARTIAL FULFILLMENT OF MASTER OF SCIENCE, SYSTEMS TECHNOLOGY      *MAI00170
C     *                                                                      *MAI00180
C     * THIS PROGRAM WAS WRITTEN FOR EXECUTION ON AN IBM 3033 COMPUTER       *MAI00190
C     * UTILIZING THE VMS/CMS OPERATING SYSTEM.  WITHIN THE CMS SYSTEM       *MAI00200
C     * THERE IS A ROUTINE TO CLEAR THE SCREEN WHEN DESIRED.  THIS           *MAI00210
C     * FUNCTION IS CALLED THROUGH USE OF THE COMMAND *                       MAI00220
C     * CALL FRTCMS('CLRSCRN ').  IF THE PROGRAM IS TO BEUSED ON THIS*       MAI00230
C     * OPERATING SYSTEM, REMOVAL OF THE COMMENT C BEFORE LINES OF           *MAI00240
C     * CODE CONTAINING THIS COMMAND WILL IMPROVE THE INTERACTIVE            *MAI00250
C     * PORTION OF THE PROGRAM.*                                              MAI00260
C     *                                                                      *MAI00270
C     ************************************************************************MAI00280
C                                                                             MAI00290
C     *** MAIN CONTROL MODULE ***                                            MAI00300
C                                                                             MAI00310
C     *** PURPOSE ***                                                        MAI00320
C                                                                             MAI00330
C     THIS PROGRAM IS THE MAIN BODY OF THE SIGNAL GENERATOR.  IT            MAI00340
C     CONTROLS INPUT AND OUTPUT AND CALLS VARIOUS SUBROUTINES DURING        MAI00350
C     ITS OPERATION.                                                        MAI00360
C                                                                           MAI00370
C     *** VARIABLE DEFINITIONS ***                                          MAI00380
C                                                                           MAI00390
C     ANS=          A INTEGER REPRESENTING THE RESPONSE TO AN INQUIRY      MAI00400
C     TYPE1=        AN INTEGER TO CONTROL INPUT                           MAI00410
C     TYPE2=        AN INTEGER TO CONTROL INPUT                           MAI00420
C     TYPE3=        AN INTEGER TO CONTROL INPUT                           MAI00430
C     ANS2=         AN INTEGER USED TO SPECIFY HOW BIT STREAM IS TO BE     MAI00440
```

```
C                 GENERATED                                                    MAI00450
C      IBAUD=     INTEGER VALUE OF BAUD RATE                                   MAI00460
C      BAUD=      SYMBOL RATE OR BAUD RATE                                     MAI00470
C      BAUDD=     BAUD DURATION                                                MAI00480
C      IBITS=     INTEGER VALUE OF BITS IN A BINARY CODE WORD                  MAI00490
C      BITS=      NUMBER OF BITS IN EACH BINARY CODE WORD                      MAI00500
C      BITR=      BIT RATE                                                     MAI00510
C      IFREQ=     INTEGER VALUE OF CARRIER FREQUENCY                           MAI00520
C      FREQ=      CARRIER FREQUENCY                                            MAI00530
C      IIPHAS=    INTEGER VALUE OF INITIAL PHASE ANGLE IN DEGREES              MAI00540
C      IPHAS=     INITIAL PHASE ANGLE IN DEGREES                               MAI00550
C      IAMP=      INTEGER VALUE OF IN-PHASE AMPLITUDE                          MAI00560
C      AMP=       AMPLITUDE                                                    MAI00570
C      ISEED=     INTEGER VALUE OF SEED FOR RANDOM NUMBER GENERATOR            MAI00580
C      DSEED=     SEED FOR RANDOM NUMBER GENERATOR                             MAI00590
C      ARRAY=     A TWO DIMENTIONAL ARRAY                                      MAI00600
C      R=         AN INTEGER REPRESENTING THE ROW OF ARRAY                     MAI00610
C      IN=        INTEGER OF NUMBER OF POSITIONS TO BE USED IN ARRAY           MAI00620
C      N=         REAL VALUE OF IN                                             MAI00630
C      REPS=      INTEGER VALUE OF NUMBER OF TIMES THE PROGRAM IS TO BE        MAI00640
C                 REPEATED                                                     MAI00650
C      REP=       INTEGER VALUE OF THE NUMBER OF THE REPETITION OF THE         MAI00660
C                 PROGRAM AND PASSED TO THE SUBPROGRAM FOR THE                 MAI00670
C                 MODULATION TECHNIQUE BEING EMPLOYED                          MAI00680
C      FLAG=      INTEGER TO CONTROL POINT OF ENTRANCE TO SUBROUTINE           MAI00690
C                 STATS                                                        MAI00700
C      FC=        RECOMMENDED MINIMUM CARRIER FREQUENCY                        MAI00710
C      STEP=      TIME INTERVAL BETWEEN SAMPLES                                MAI00720
C                                                                              MAI00730
C      *** VARIABLE DECLARATIONS ***                                           MAI00740
C                                                                              MAI00750
       INTEGER ANS,TYPE1,TYPE2,TYPE3,ANS2,IBAUD,IBITS,IFREQ,                  MAI00760
      *IIPHAS,IAMP,ISEED,R,IN,REPS,REP,FLAG                                   MAI00770
C                                                                              MAI00780
       DOUBLE PRECISION BAUD,BITS,BITR,FREQ,IPHAS,AMP,BAUDD,STEP,FC,          MAI00790
      *DSEED,N,PI                                                             MAI00800
C                                                                              MAI00810
       DOUBLE PRECISION ARRAY(1024,2),SUMX(513),SUMXSQ(513),SUMX3(513),       MAI00820
      *SUMX4(513)                                                             MAI00830
       COMMON ARRAY,SUMX,SUMXSQ,SUMX3,SUMX4                                   MAI00840
C                                                                              MAI00850
C      *** VARIABLE INITIALIZATIONS ***                                        MAI00860
C                                                                              MAI00870
       R=1                                                                    MAI00880
       PI=3.1415926535897930                                                  MAI00890
       TYPE3=0                                                                MAI00900
C                                                                              MAI00910
C      *** INTRODUCE THE PROGRAM ***                                           MAI00920
C                                                                              MAI00930
C      CALL FRTCMS('CLRSCRN ')                                                 MAI00940
```

```
C                                                                       MAI00950
10    WRITE(10,20)                                                      MAI00960
20    FORMAT(' HELLO!  YOU HAVE INITIATED A PROGRAM WHICH WILL ALLOW',/,MAI00970
     *' YOU TO PRODUCE ONE OF A NUMBER OF SIMULATED COMMUNICATIONS',/,  MAI00980
     *' SIGNALS UTILIZING VARIOUS DIGITAL SIGNAL MODULATION TECHNIQUES.'MAI00990
     *,/,' YOU WILL BE ASKED TO PROVIDE CERTAIN INFORMATION ABOUT THE', MAI01000
     */,' SIGNAL YOU WISH TO SIMULATE.  ADDITIONALLY YOU WILL BE',      MAI01010
     */,' GIVEN THE OPPORTUNITY TO PLOT THE RESULTS, GENERATE THE FFT' MAI01020
     *,/,' OF THE SIGNAL OR TO DO CERTAIN STATISTICAL ANALYSIS OF THE' MAI01030
     *,/,' FFT.  IF YOU ARE READY TO PROCEED, TYPE 1 <CR>.',//,' :')   MAI01040
C                                                                       MAI01050
      READ(5,21)ANS                                                     MAI01060
21    FORMAT(I1)                                                        MAI01070
C                                                                       MAI01080
      IF(ANS.NE.1)THEN                                                  MAI01090
C         CALL FRTCMS('CLRSCRN ')                                       MAI01100
          GO TO 10                                                      MAI01110
      END IF                                                            MAI01120
C                                                                       MAI01130
CC    *** DISPLAY MODULATION TECHNIQUES AVAILABLE AND INQUIRE WHICH     MAI01140
CC    TECHNIQUE IS TO BE SIMULATED ***                                  MAI01150
CC                                                                      MAI01160
C     CALL FRTCMS('CLRSCRN ')                                           MAI01170
C                                                                       MAI01180
29    WRITE(10,30)                                                      MAI01190
30    FORMAT(' THIS PROGRAM IS CAPABLE OF SIMULATING THE FOLLOWING',/,  MAI01200
     *' DIGITAL SIGNAL MODULATION TECHNIQUES:',//,                      MAI01210
     *' 1.  BINARY PHASE SHIFT KEYING (BPSK)',/,                        MAI01220
     *' 2.  DIFFERENTIAL BINARY PHASE SHIFT KEYING (DBPSK)',/,          MAI01230
     *' 3.  ORTHOGONAL BINARY PHASE SHIFT KEYING (ORTHOGONAL BPSK)',/,  MAI01240
     *' 4.  QUADRATURE PHASE SHIFT KEYING (QPSK)',/,                    MAI01250
     *' 5.  OFFSET QUADRATURE PHASE SHIFT KEYING (OQPSK)',/,            MAI01260
     *' 6.  MULTIPLE PHASE SHIFT KEYING (MPSK)',/,                      MAI01270
     *' 7.  MULTIPLE AMPLITUDE SHIFT KEYING (MASK)',/,                  MAI01280
     *' 8.  QUADRATURE AMPLITUDE SHIFT KEYING (QASK)',/,                MAI01290
     *' 9.  MINIMUM SHIFT KEYING (MSK) OR',/,                           MAI01300
     *'      FAST FREQUENCY SHIFT KEYING (FFSK)',/,                    MAI01310
     *' 10. MULTIPLE FREQUENCY SHIFT KEYING (MFSK)',/,                  MAI01320
     *' 11. QUADRATURE PARTIAL RESPONSE (QPRS)',//,                     MAI01330
     *' ENTER THE NUMBER (1-11) WHICH CORRESPONDS TO THE DIGITAL SIGNAL'MAI01340
     *,/,' SIGNAL MODULATION TECHNIQUE YOU WISH TO SIMULATE',//,' :')   MAI01350
C                                                                       MAI01360
      READ(5,*)TYPE1                                                    MAI01370
C                                                                       MAI01380
      IF(TYPE1.LT.1.OR.TYPE1.GT.11)THEN                                 MAI01390
C         CALL FRTCMS('CLRSCRN ')                                       MAI01400
          WRITE(10,40)                                                  MAI01410
40        FORMAT(' ERROR',/)                                            MAI01420
          GO TO 29                                                      MAI01430
      END IF                                                            MAI01440
```

```
C                                                                       MAI01450
CC     *** DETERMINE THE CLASS OF THE QUADRATURE PARTIAL RESPONE SYSTEM *MAI01460
C                                                                       MAI01470
       IF(TYPE1.EQ.11)THEN                                              MAI01480
C                                                                       MAI01490
CC         CALL FRTCMS('CLRSCRN ')                                      MAI01500
CC                                                                      MAI01510
41         WRITE(10,42)                                                 MAI01520
42         FORMAT(' ENTER THE CLASS OF THE QUADRATURE PARTIAL RESPONSE', MAI01530
     *     /,' SYSTEM TO BE SIMULATED FROM 1-5.',//,' :')               MAI01540
C                                                                       MAI01550
           READ(5,*)TYPE3                                               MAI01560
C                                                                       MAI01570
           IF(TYPE3.LT.1.OR.TYPE3.GT.5)THEN                             MAI01580
C              CALL FRTCMS('CLRSCRN ')                                  MAI01590
               WRITE(10,43)                                             MAI01600
43             FORMAT(' ERROR',/)                                       MAI01610
               GO TO 41                                                 MAI01620
           END IF                                                       MAI01630
C                                                                       MAI01640
       END IF                                                           MAI01650
C                                                                       MAI01660
CC     *** DETERMINE THE LOGIC SCHEME TO BE EMPLOYED DURING MODULATION **MAI01670
CC                                                                      MAI01680
CC     CALL FRTCMS('CLRSCRN ')                                          MAI01690
CC                                                                      MAI01700
49     WRITE(10,50)                                                     MAI01710
50     FORMAT(' ENTER THE NUMBER (1-3) WHICH CORRESPONDS TO THE',/,     MAI01720
     *' DESIRED DIGITAL LOGIC SCHEME:',//,                             MAI01730
     *' 1.  BIPOLAR',/,                                                MAI01740
     *' 2.  UNIPOLAR POSITIVE',/,                                      MAI01750
     *' 3.  UNIPOLAR NEGATIVE',//,' :')                                MAI01760
C                                                                       MAI01770
       READ(5,51)TYPE2                                                  MAI01780
51     FORMAT(I1)                                                       MAI01790
C                                                                       MAI01800
       IF(TYPE2.LT.1.OR.TYPE2.GT.3)THEN                                 MAI01810
C          CALL FRTCMS('CLRSCRN ')                                      MAI01820
           WRITE(10,60)                                                 MAI01830
60         FORMAT(' ERROR',/)                                           MAI01840
           GO TO 49                                                     MAI01850
       END IF                                                           MAI01860
C                                                                       MAI01870
CC     *** HAVE THE USER ENTER THE BAUD OR SYMBOL RATE ***              MAI01880
CC                                                                      MAI01890
CC     CALL FRTCMS('CLRSCRN ')                                          MAI01900
C                                                                       MAI01910
69     WRITE(10,70)                                                     MAI01920
70     FORMAT(' ENTER THE BAUD RATE OR SYMBOL RATE AT THIS TIME.',//,   MAI01930
     *' :')                                                            MAI01940
```

```
C
      READ(5,*)IBAUD                                                    MAI01950
C                                                                       MAI01960
      BAUD=IBAUD                                                        MAI01970
C                                                                       MAI01980
      BAUDD=1.D0/BAUD                                                   MAI01990
C                                                                       MAI02000
C     *** DETERMINE THE BITS EITHER BY THE TYPE OF DIGITAL              MAI02010
C     MODULATION SPECIFIED OR BY USER INPUT ***                         MAI02020
C                                                                       MAI02030
C     CALL FRTCMS('CLRSCRN ')                                           MAI02040
C                                                                       MAI02050
      IF(TYPE1.EQ.1.OR.TYPE1.EQ.2.OR.TYPE1.EQ.9)THEN                    MAI02060
          IBITS=1                                                       MAI02070
          BITS=IBITS                                                    MAI02080
C                                                                       MAI02090
      ELSE IF(TYPE1.EQ.3)THEN                                           MAI02100
79        WRITE(10,80)                                                  MAI02110
80        FORMAT(' ENTER THE NUMBER OF BITS IN EACH SYMBOL.',/,         MAI02120
     *    ' CAUTION!  THE NUMBER OF BITS PER SYMBOL MUST BE 6 OR LESS'  MAI02130
     *    ,//,' :')                                                     MAI02140
C                                                                       MAI02150
          READ(5,*)IBITS                                                MAI02160
C                                                                       MAI02170
          IF(IBITS.LT.1.OR.IBITS.GT.6)THEN                              MAI02180
C             CALL FRTCMS('CLRSCRN ')                                   MAI02190
              WRITE(10,90)                                              MAI02200
90            FORMAT(' ERROR',/)                                        MAI02210
              GO TO 79                                                  MAI02220
          ELSE                                                          MAI02230
              BITS=IBITS                                                MAI02240
          END IF                                                        MAI02250
C                                                                       MAI02260
      ELSE IF(TYPE1.EQ.4.OR.TYPE1.EQ.5.OR.TYPE1.EQ.11)THEN              MAI02270
          IBITS=2                                                       MAI02280
          BITS=IBITS                                                    MAI02290
C                                                                       MAI02300
      ELSE IF(TYPE1.GE.6.OR.TYPE1.LE.8.OR.TYPE1.EQ.10)THEN              MAI02310
99        WRITE(10,100)                                                 MAI02320
100       FORMAT(' ENTER THE NUMBER OF BITS IN EACH SYMBOL.',/,         MAI02330
     *    ' CAUTION!  THE NUMBER OF BITS PER SYMBOL MUST BE 10 OR LESS' MAI02340
     *    ,//,' :')                                                     MAI02350
C                                                                       MAI02360
          READ(5,*)IBITS                                                MAI02370
C                                                                       MAI02380
          IF(IBITS.LT.1.OR.IBITS.GT.11)THEN                             MAI02390
C             CALL FRTCMS('CLRSCRN ')                                   MAI02400
              WRITE(10,110)                                             MAI02410
110           FORMAT(' ERROR',/)                                        MAI02420
              GO TO 99                                                  MAI02430
                                                                        MAI02440
```

```
            ELSE                                                                MAI02450
                BITS=IBITS                                                      MAI02460
            END IF                                                              MAI02470
C                                                                               MAI02480
        END IF                                                                  MAI02490
C                                                                               MAI02500
CC      *** DETERMINE AND DISPLAY THE BIT RATE ***                              MAI02510
C                                                                               MAI02520
        IF(TYPE1.EQ.3)THEN                                                      MAI02530
            BITR=(2.D0**IBITS)*BAUD                                             MAI02540
        ELSE                                                                    MAI02550
            BITR=BITS*BAUD                                                      MAI02560
        END IF                                                                  MAI02570
C                                                                               MAI02580
C       CALL FRTCMS('CLRSCRN ')                                                 MAI02590
        WRITE(10,120)BITR                                                       MAI02600
120     FORMAT(' THE BIT RATE FOR THE SPECIFIED SIGNAL IS',F9.0,' BITS/SEC      MAI02610
       *')                                                                      MAI02620
C                                                                               MAI02630
CC      *** HAVE THE USER ENTER THE CARRIER FREQUENCY ***                       MAI02640
C                                                                               MAI02650
        IF(TYPE1.EQ.3)THEN                                                      MAI02660
            FC=BITR                                                             MAI02670
        ELSE IF(TYPE1.EQ.9)THEN                                                 MAI02680
            FC=1.25D0*BAUD                                                      MAI02690
        ELSE IF(TYPE1.EQ.10)THEN                                                MAI02700
            FC=BAUD+(((2.D0**IBITS)-1.D0)*(BAUD/2.D0))                          MAI02710
        ELSE                                                                    MAI02720
            FC=BAUD                                                             MAI02730
        END IF                                                                  MAI02740
C                                                                               MAI02750
129     WRITE(10,130)FC                                                         MAI02760
130     FORMAT(' ENTER THE CARRIER FREQUENCY AT THIS TIME.',//,                 MAI02770
       *' NOTE:  CARRIER FREQ SHOULD BE GREATER THAN',F9.0,' HZ.',//,           MAI02780
       *' :')                                                                   MAI02790
C                                                                               MAI02800
        READ(5,*)IFREQ                                                          MAI02810
C                                                                               MAI02820
        FREQ=IFREQ                                                              MAI02830
C                                                                               MAI02840
        IF(FREQ.LT.FC)THEN                                                      MAI02850
C           CALL FRTCMS('CLRSCRN ')                                             MAI02860
            WRITE(10,140)                                                       MAI02870
140         FORMAT(' THE CARRIER FREQUENCY IS LESS THAN RECOMMENDED.',/)        MAI02880
            GO TO 149                                                           MAI02890
        END IF                                                                  MAI02900
C                                                                               MAI02910
CC      *** HAVE THE USER ENTER THE NUMBER OF SAMPLES TO BE GENERATED ***       MAI02920
CC                                                                              MAI02930
C       CALL FRTCMS('CLRSCRN ')                                                 MAI02940
```

```
C                                                                       MAI02950
149   WRITE(10,150)                                                     MAI02960
150   FORMAT(' ENTER THE NUMBER OF SAMPLES OF THE MODULATED SIGNAL',    MAI02970
     */,' TO BE PRODUCED.  THIS NUMBER MUST BE EQUAL TO 2**N,',/,      MAI02980
     *' WHERE N IS A POSITIVE INTEGER EQUAL TO OR LESS THAN 10.',/,     MAI02990
     *' I.E., 2, 4, 8, ... , 1024',//,' :')                            MAI03000
C                                                                       MAI03010
      READ(5,*)IN                                                       MAI03020
      N=IN                                                              MAI03030
C                                                                       MAI03040
      IF(IN.EQ.2.OR.IN.EQ.4.OR.IN.EQ.8.OR.IN.EQ.16.OR.IN.EQ.32.OR.      MAI03050
     *IN.EQ.64.OR.IN.EQ.128.OR.IN.EQ.256.OR.IN.EQ.512.OR.IN.EQ.1024)THENMAI03060
          GO TO 170                                                     MAI03070
      ELSE                                                              MAI03080
C         CALL FRTCMS('CLRSCRN ')                                       MAI03090
          WRITE(10,160)                                                 MAI03100
160       FORMAT(' ERROR',/)                                            MAI03110
          GO TO 149                                                     MAI03120
      END IF                                                            MAI03130
C                                                                       MAI03140
C     *** DETERMINE AND DISPLAY INFO ABOUT THE RECORD LENGTH ***        MAI03150
C                                                                       MAI03160
170   IF(TYPE1.EQ.2)THEN                                                MAI03170
          STEP=1.D0/(2.D0*(FREQ+BITR))                                  MAI03180
      ELSE IF(TYPE1.EQ.9)THEN                                           MAI03190
          STEP=1.D0/(2.D0*(FREQ+(1.25D0*BAUD)))                         MAI03200
      ELSE IF(TYPE1.EQ.10)THEN                                          MAI03210
          STEP=1.D0/(2.D0*(FREQ+(((2.D0**IBITS)-1.)*(BAUD/2.D0))+BAUD)) MAI03220
      ELSE IF(TYPE1.EQ.11)THEN                                          MAI03230
          STEP=1.D0/(2.D0*(FREQ+((PI+1.D0)*BAUD)))                      MAI03240
      ELSE                                                              MAI03250
          STEP=1.D0/(2.D0*(FREQ+BAUD))                                  MAI03260
      END IF                                                            MAI03270
C                                                                       MAI03280
      IF(TYPE1.EQ.3.AND.((N-1.)*STEP).LE.1./BITR)THEN                   MAI03290
C         CALL FRTCMS('CLRSCRN ')                                       MAI03300
          WRITE(10,180)                                                 MAI03310
180       FORMAT(' ELAPSED TIME FOR SIGNAL GENERATION WILL BE LESS THAN'MAI03320
     *    ,/,' 1 BIT DURATION.  IF YOU WISH TO CHANGE THIS BY',/,       MAI03330
     *    ' INCREASING THE NUMBER OF SAMPLES TO BE PRODUCED, BY',/,     MAI03340
     *    ' DECREASING CARRIER FREQUENCY, BY DECREASING THE BAUD',/,    MAI03350
     *    ' RATE OR BY DECREASING THE SIZE OF THE BINARY CODE WORD.',/,/,MAI03360
     *    ' ENTER A 1.  ENTER ANY OTHER INTEGER VALUE TO CONTINUE',/    MAI03370
     *    ' WITH THE SIMULATION AS SPECIFIED.',//,' :')                 MAI03380
C                                                                       MAI03390
          READ(5,*)ANS                                                  MAI03400
C                                                                       MAI03410
          IF(ANS.EQ.1)THEN                                              MAI03420
              GO TO 29                                                  MAI03430
          END IF                                                        MAI03440
```

```
C                                                                       MAI03450
      ELSE IF(((N-1.D0)*STEP).LE.BAUDD)THEN                             MAI03460
C         CALL FRTCMS('CLRSCRN ')                                       MAI03470
          WRITE(10,190)                                                 MAI03480
190       FORMAT(' ELAPSED TIME FOR SIGNAL GENERATION WILL BE LESS THAN'MAI03490
     *    ,/,' 1 SYMBOL DURATION.  IF YOU WISH TO CHANGE THIS BY',/,    MAI03500
     *    ' INCREASING THE NUMBER OF SAMPLES TO BE PRODUCED, BY',/,     MAI03510
     *    ' DECREASING CARRIER FREQUENCY, BY DECREASING THE BAUD',/,    MAI03520
     *    ' RATE OR BY DECREASING THE SIZE OF THE BINARY CODE WORD.',/,/,MAI03530
     *    ' ENTER A 1.  ENTER ANY OTHER INTEGER VALUE TO CONTINUE',//,  MAI03540
     *    ' WITH THE SIMULATION AS SPECIFIED.',//,' :')                 MAI03550
C                                                                       MAI03560
          READ(5,*)ANS                                                  MAI03570
C                                                                       MAI03580
          IF(ANS.EQ.1)THEN                                              MAI03590
              GO TO 29                                                  MAI03600
          END IF                                                        MAI03610
C                                                                       MAI03620
      END IF                                                            MAI03630
C                                                                       MAI03640
CC    *** HAVE THE USER ENTER THE AMPLITUDE OF THE CARRIER WAVE ***     MAI03650
CC                                                                      MAI03660
CC    CALL FRTCMS('CLRSCRN ')                                           MAI03670
C                                                                       MAI03680
199   WRITE(10,200)                                                     MAI03690
200   FORMAT(' ENTER THE AMPLITUDE OF THE CARRIER.',//,' :')            MAI03700
C     READ(5,*)IAMP                                                     MAI03710
C                                                                       MAI03720
      AMP=IAMP                                                          MAI03730
C                                                                       MAI03740
CC    *** HAVE THE USER ENTER THE INITIAL PHASE ANGLE ***               MAI03750
CC                                                                      MAI03760
C     CALL FRTCMS('CLRSCRN ')                                           MAI03770
C                                                                       MAI03780
209   WRITE(10,210)                                                     MAI03790
210   FORMAT(' ENTER THE INITIAL PHASE ANGLE FROM 0 TO 360 DEGREES.',//,MAI03800
     *' :')                                                             MAI03810
C                                                                       MAI03820
      READ(5,*)IIPHAS                                                   MAI03830
C                                                                       MAI03840
      IF(IIPHAS.LT.0.OR.IIPHAS.GT.360)THEN                              MAI03850
C         CALL FRTCMS('CLRSCRN ')                                       MAI03860
          WRITE(10,220)                                                 MAI03870
220       FORMAT(' ERROR',/)                                            MAI03880
          GO TO 209                                                     MAI03890
      ELSE                                                              MAI03900
          IPHAS=IIPHAS                                                  MAI03910
      END IF                                                            MAI03920
C                                                                       MAI03930
                                                                        MAI03940
```

```
C     *** DETERMINE IF A RANDOM BIT STREAM IS TO BE GENERATED ***          MAI03950
C                                                                          MAI03960
C     CALL FRTCMS('CLRSCRN ')                                              MAI03970
C                                                                          MAI03980
229   WRITE(10,230)                                                        MAI03990
230   FORMAT(' ENTER 1 IF YOU DESIRE TO HAVE THE COMPUTER GENERATE A',/,   MAI04000
     *' RANDOM BIT STREAM OR ENTER 2 IF YOU WISH TO ENTER YOUR OWN BIT',   MAI04010
     */,' STREAM.',//,' :')                                               MAI04020
C                                                                          MAI04030
      READ(5,240) ANS2                                                     MAI04040
240   FORMAT(I1)                                                           MAI04050
C                                                                          MAI04060
      IF(ANS2.LT.1.OR.ANS2.GT.2) THEN                                      MAI04070
C         CALL FRTCMS('CLRSCRN ')                                          MAI04080
          WRITE(10,250)                                                    MAI04090
250       FORMAT(' ERROR',/)                                               MAI04100
          GO TO 229                                                        MAI04110
      END IF                                                               MAI04120
C                                                                          MAI04130
C     *** HAVE THE USER ENTER THE SEED FOR THE RANDOM NUMBER GENERATOR**   MAI04140
C                                                                          MAI04150
C     CALL FRTCMS('CLRSCRN ')                                              MAI04160
C                                                                          MAI04170
      IF(ANS2.EQ.1)THEN                                                    MAI04180
          WRITE(10,260)                                                    MAI04190
260       FORMAT(' ENTER THE SEED (A REAL NUMBER BETWEEN 1 AND 2**31)',/   MAI04200
     *     ,' FOR THE RANDOM NUMBER GENERATOR',//,' :')                    MAI04210
C                                                                          MAI04220
          READ(5,*)ISEED                                                   MAI04230
C                                                                          MAI04240
          DSEED=ISEED                                                      MAI04250
      END IF                                                               MAI04260
C                                                                          MAI04270
C     *** HAVE THE USER SPECIFY THE NUMBER OF TRIALS TO BE USED ***        MAI04280
C                                                                          MAI04290
C     CALL FRTCMS('CLRSCRN ')                                              MAI04300
C                                                                          MAI04310
      WRITE(10,270)                                                        MAI04320
270   FORMAT(' ENTER THE NUMBER OF TIMES YOU WOULD LIKE THE SIMULATION',   MAI04330
     */,' TO REPEAT.',//,' :')                                            MAI04340
C                                                                          MAI04350
      READ(5,*)REPS                                                        MAI04360
C                                                                          MAI04370
C     *** CALL THE APPROPRIATE SUBROUTINE FOR THE MODULATION TECHNIQUE     MAI04380
C     SPECIFIED, THE APPROPRIATE NUMBER OF TIMES ***                       MAI04390
C                                                                          MAI04400
      DO 280 REP=1,REPS                                                    MAI04410
C                                                                          MAI04420
      IF(TYPE1.EQ.1)THEN                                                   MAI04430
          CALL BPSK(TYPE2,BAUD,FREQ,IPHAS,AMP,ANS2,DSEED,IN,REP)           MAI04440
```

```
C
      ELSE IF(TYPE1.EQ.2)THEN                                          MAI04450
           CALL DBPSK(TYPE2,BAUD,FREQ,IPHAS,AMP,ANS2,DSEED,IN,REP)     MAI04460
C                                                                      MAI04470
      ELSE IF(TYPE1.EQ.3)THEN                                          MAI04480
           CALL OBPSK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,ANS2,DSEED,IN,REP)MAI04490
C                                                                      MAI04500
      ELSE IF(TYPE1.EQ.4)THEN                                          MAI04510
           CALL QPSK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,ANS2,DSEED,IN,REP) MAI04520
C                                                                      MAI04530
      ELSE IF(TYPE1.EQ.5)THEN                                          MAI04540
           CALL OQPSK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,ANS2,DSEED,IN,    MAI04550
     *     REP)                                                        MAI04560
C                                                                      MAI04570
      ELSE IF(TYPE1.EQ.6)THEN                                          MAI04580
           CALL MPSK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,ANS2,DSEED,IN,     MAI04590
     *     REP)                                                        MAI04600
C                                                                      MAI04610
      ELSE IF(TYPE1.EQ.7)THEN                                          MAI04620
           CALL MASK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,ANS2,DSEED,IN,     MAI04630
     *     REP)                                                        MAI04640
C                                                                      MAI04650
      ELSE IF(TYPE1.EQ.8)THEN                                          MAI04660
           CALL QASK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,ANS2,DSEED,IN,     MAI04670
     *     REP)                                                        MAI04680
C                                                                      MAI04690
      ELSE IF(TYPE1.EQ.9)THEN                                          MAI04700
           CALL MSK(TYPE2,BAUD,FREQ,IPHAS,AMP,ANS2,DSEED,IN,REP)       MAI04710
C                                                                      MAI04720
      ELSE IF(TYPE1.EQ.10)THEN                                         MAI04730
           CALL MFSK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,ANS2,DSEED,IN,REP) MAI04740
C                                                                      MAI04750
      ELSE IF(TYPE1.EQ.11)THEN                                         MAI04760
           CALL QPRS(TYPE2,TYPE3,BAUD,BITS,FREQ,IPHAS,AMP,ANS2,DSEED,  MAI04770
     *     IN,REP)                                                     MAI04780
C                                                                      MAI04790
      END IF                                                           MAI04800
C                                                                      MAI04810
280   CONTINUE                                                         MAI04820
C                                                                      MAI04830
C     *** OUTPUT A FINGERPRINT OF THE MODULATION ACCOMPLISHED ***      MAI04840
C                                                                      MAI04850
      IF(TYPE1.EQ.1)THEN                                               MAI04860
           WRITE(6,281)                                                MAI04870
281        FORMAT('1',' MODULATION TECHNIQUE = BPSK')                  MAI04880
      ELSE IF(TYPE1.EQ.2)THEN                                          MAI04890
           WRITE(6,282)                                                MAI04900
282        FORMAT('1',' MODULATION TECHNIQUE = DBPSK')                 MAI04910
      ELSE IF(TYPE1.EQ.3)THEN                                          MAI04920
           WRITE(6,283)                                                MAI04930
                                                                       MAI04940
```

```
283       FORMAT('1',' MODULATION TECHNIQUE = ORTHOGONAL BPSK')          MAI04950
      ELSE IF(TYPE1.EQ.4)THEN                                            MAI04960
          WRITE(6,284)                                                   MAI04970
284       FORMAT('1',' MODULATION TECHNIQUE = QPSK')                     MAI04980
      ELSE IF(TYPE1.EQ.5)THEN                                            MAI04990
          WRITE(6,285)                                                   MAI05000
285       FORMAT('1',' MODULATION TECHNIQUE = OQPSK')                    MAI05010
      ELSE IF(TYPE1.EQ.6)THEN                                            MAI05020
          WRITE(6,286)                                                   MAI05030
286       FORMAT('1',' MODULATION TECHNIQUE = MPSK')                     MAI05040
      ELSE IF(TYPE1.EQ.7)THEN                                            MAI05050
          WRITE(6,287)                                                   MAI05060
287       FORMAT('1',' MODULATION TECHNIQUE = MASK')                     MAI05070
      ELSE IF(TYPE1.EQ.8)THEN                                            MAI05080
          WRITE(6,288)                                                   MAI05090
288       FORMAT('1',' MODULATION TECHNIQUE = QASK')                     MAI05100
      ELSE IF(TYPE1.EQ.9)THEN                                            MAI05110
          WRITE(6,289)                                                   MAI05120
289       FORMAT('1',' MODULATION TECHNIQUE = MSK')                      MAI05130
      ELSE IF(TYPE1.EQ.10)THEN                                           MAI05140
          WRITE(6,290)                                                   MAI05150
290       FORMAT('1',' MODULATION TECHNIQUE = MFSK')                     MAI05160
      ELSE IF(TYPE1.EQ.11)THEN                                           MAI05170
          WRITE(6,291)                                                   MAI05180
291       FORMAT('1',' MODULATION TECHNIQUE = QPRS')                     MAI05190
      END IF                                                             MAI05200
C                                                                        MAI05210
      IF(TYPE1.EQ.11.AND.TYPE3.EQ.1)THEN                                 MAI05220
          WRITE(6,292)                                                   MAI05230
292       FORMAT(' QPRS CLASS 1 FILTER')                                 MAI05240
      ELSE IF(TYPE1.EQ.11.AND.TYPE3.EQ.2)THEN                            MAI05250
          WRITE(6,293)                                                   MAI05260
293       FORMAT(' QPRS CLASS 2 FILTER')                                 MAI05270
      ELSE IF(TYPE1.EQ.11.AND.TYPE3.EQ.3)THEN                            MAI05280
          WRITE(6,294)                                                   MAI05290
294       FORMAT(' QPRS CLASS 3 FILTER')                                 MAI05300
      ELSE IF(TYPE1.EQ.11.AND.TYPE3.EQ.4)THEN                            MAI05310
          WRITE(6,295)                                                   MAI05320
295       FORMAT(' QPRS CLASS 4 FILTER')                                 MAI05330
      ELSE IF(TYPE1.EQ.11.AND.TYPE3.EQ.5)THEN                            MAI05340
          WRITE(6,296)                                                   MAI05350
296       FORMAT(' QPRS CLASS 5 FILTER')                                 MAI05360
      END IF                                                             MAI05370
C                                                                        MAI05380
      IF(TYPE2.EQ.1)THEN                                                 MAI05390
          WRITE(6,297)                                                   MAI05400
297       FORMAT(' BIPOLAR LOGIC')                                       MAI05410
      ELSE IF(TYPE2.EQ.2)THEN                                            MAI05420
          WRITE(6,298)                                                   MAI05430
298       FORMAT(' UNIPOLAR POSITIVE LOGIC')                             MAI05440
```

```
      ELSE IF(TYPE2.EQ.3)THEN                                          MAI05450
           WRITE(6,299)                                                MAI05460
299        FORMAT(' UNIPOLAR NEGATIVE LOGIC')                          MAI05470
      END IF                                                           MAI05480
C                                                                      MAI05490
      WRITE(6,300)IBAUD                                                MAI05500
300   FORMAT(' BAUD OR SYMBOL RATE = ',I10,' HZ')                      MAI05510
C                                                                      MAI05520
      WRITE(6,301)IBITS                                                MAI05530
301   FORMAT(' BITS PER BINARY CODE WORD = ',I10)                      MAI05540
C                                                                      MAI05550
      WRITE(6,302)BITR                                                 MAI05560
302   FORMAT(' BIT RATE = ',E23.16)                                    MAI05570
C                                                                      MAI05580
      WRITE(6,303)IFREQ                                                MAI05590
303   FORMAT(' CARRIER FREQUENCY = ',I10,' HZ')                        MAI05600
C                                                                      MAI05610
      WRITE(6,304)IAMP                                                 MAI05620
304   FORMAT(' MAXIMUM CARRIER AMPLITUDE = ',I10,' VOLT(S)')           MAI05630
C                                                                      MAI05640
      WRITE(6,305)IIPHAS                                               MAI05650
305   FORMAT(' INITIAL PHASE ANGLE = ',I10,' DEGREES')                 MAI05660
C                                                                      MAI05670
      WRITE(6,306)STEP                                                 MAI05680
306   FORMAT(' TIME BETWEEN SAMPLES = ',E23.16,' SEC')                 MAI05690
C                                                                      MAI05700
      WRITE(6,307)IN                                                   MAI05710
307   FORMAT(' NUMBER OF SAMPLES GENERATED = ',I10)                    MAI05720
C                                                                      MAI05730
      IF(ANS2.EQ.1)THEN                                                MAI05740
           WRITE(6,308)ISEED                                           MAI05750
308        FORMAT(' SEED FOR RANDOM NUMBER GENERATOR = ',I10)          MAI05760
      END IF                                                           MAI05770
C                                                                      MAI05780
      WRITE(6,309)REPS                                                 MAI05790
309   FORMAT(' NUMBER OF TIMES SIMULATION REPEATS = ',I10)             MAI05800
C                                                                      MAI05810
C     *** CALL THE STATISTICS SUBROUTINE FOR FINAL CALCULATIONS ***    MAI05820
C                                                                      MAI05830
      FLAG=1                                                           MAI05840
      CALL STA1S(IN,REP,FLAG)                                          MAI05850
C                                                                      MAI05860
      IF(TYPE1.EQ.3.AND.((N-1.D0)*STEP).LE.1./BITR)THEN                MAI05870
           WRITE(6,310)                                                MAI05880
310        FORMAT(' LESS THAN 1 BIT WAS MODULATED DURING EACH REPETITION MAI05890
     *     ')                                                          MAI05900
      ELSE IF(((N-1.D0)*STEP).LE.BAUDD)THEN                            MAI05910
           WRITE(6,320)                                                MAI05920
320        FORMAT(' LESS THAN 1 BAUD WAS MODULATED DURING EACH REPETITIONMAI05930
     *     ')                                                          MAI05940
```

```
      END IF                                                                    MAI05950
C                                                                               MAI05960
CC    *** DETERMINE IF ANOTHER SIMULATION IS TO BE RUN ***                      MAI05970
C                                                                               MAI05980
      WRITE(10,330)                                                             MAI05990
330   FORMAT(' ENTER A 1 IF YOU DESIRE TO SIMULATE ANOTHER SIGNAL.',/,          MAI06000
     *' ENTER ANY CTHER INTEGER IF YOU ARE READY TO QUIT.',//,' :')              MAI06010
C                                                                               MAI06020
      READ(5,*)ANS                                                              MAI06030
C                                                                               MAI06040
      IF(ANS.EQ.1)THEN                                                          MAI06050
C         CALL FRTCMS('CLRSCRN ')                                               MAI06060
          GO TO 29                                                              MAI06070
      END IF                                                                    MAI06080
C                                                                               MAI06090
      STOP                                                                      MAI06100
      END                                                                       MAI06110
```

```
      SUBROUTINE BPSK(TYPE2,BAUD,FREQ,IPHAS,AMP,ANS2,                   BPS00110
     *DSEED,IN,REP)                                                     BPS00120
C                                                                       BPS00130
C     *** PURPOSE ***                                                   BPS00140
C                                                                       BPS00150
C     THIS SUBROUTINE MODULATES THE CARRIER USING BINARY PHASE SHIFT    BPS00160
C     KEYING AS THE MODULATION TECHNIQUE.                               BPS00170
C                                                                       BPS00180
C     *** PARAMETER DEFINITIONS ***                                     BPS00190
C                                                                       BPS00200
C     TYPE2=       INDICATES LOGIC TYPE TO BE EMPLOYED                  BPS00210
C     BAUD=        SYMBOL RATE OR BAUD RATE                             BPS00220
C     FREQ=        CARRIER FREQUENCY                                    BPS00230
C     IPHAS=       INITIAL PHASE ANGLE IN DEGREES                       BPS00240
C     AMP=         AMPLITUDE                                            BPS00250
C     ANS2=        INTEGER VARIABLE TO BE PASSED TO STREAM              BPS00260
C     DSEED=       DOUBLE PRECISION SEED FOR RANDOM NUMBER GENERATOR    BPS00270
C     IN=          NUMBER OF POSITIONS IN ARRAY TO BE UTILIZED          BPS00280
C     REP=         AN INTEGER EQUAL TO THE NUMBER OF THE REPETITION OF  BPS00290
C                  THE CALL THE THIS SUBROUTINE                         BPS00300
C                                                                       BPS00310
C     *** VARIABLE DEFINITIONS ***                                      BPS00320
C                                                                       BPS00330
C     ARRAY=       A DOUBLE PRECISION ARRAY FOR PASSING THE BINARY      BPS00340
C                  DIGITS AND STORING THE VALUE OF THE TIME FUNCTION    BPS00350
C                  AND THE FFT                                          BPS00360
C     R=           AN INTEGER USED TO REPRESENT THE ROW OF ARRAY        BPS00370
C     TIME=        TIME VARIABLE                                        BPS00380
C     MT=          VALUE OF THE BINARY DIGIT                            BPS00390
C     STEP=        INTERVAL AT WHICH THE SIGNAL IS REPRODUCED           BPS00400
C     PI=          NUMERICAL CONSTANT                                   BPS00410
C     OMEGA=       CARRIER ANGULAR FREQUENCY                            BPS00420
C     DELTA=       INITIAL PEASE OFFSET IN RADIANS                      BPS00430
C     NBAUD=       A COUNT OF THE NUMBER OF BAUDS MODULATED             BPS00440
C     BAUDD=       BAUD DURATION                                        BPS00450
C     X=           COMPLEX ARRAY TO RECEIVE THE FFT OF THE TIME SERIES  BPS00460
C     IWK=         INTEGER WORKING ARRAY USED BY FUNCTION FFTRC         BPS00470
C     MAXY=        MAXIMUM VALUE TO BE PLOTTED ON THE ABCISSAE          BPS00480
C     MINY=        MINIMUM VALUE TO BE PLOTTED ON THE ABCISSAE          BPS00490
C     INT=         INTERVAL BETWEEN POINTS ON THE ORDINATE              BPS00500
C     RMAX=        VALUE OF THE PRINCIPLE HARMONIC                      BPS00510
C     FLAG=        AN INTEGER WHICH CONTROL ENTRANCE AND EXIT FROM      BPS00520
C                  SUBPROGRAM STATS                                     BPS00530
C     IND2P1=      ARRAY LENGTH DIVIDED BY 2 PLUS 1                     BPS00540
C                                                                       BPS00550
C     *** VARIABLE DECLARATIONS ***                                     BPS00560
C                                                                       BPS00570
      INTEGER R,ANS2,TYPE2,IN,RMAX,REP,FLAG,IND2P1                      BPS00580
C                                                                       BPS00590
      DOUBLE PRECISION BAUD,FREQ,IPHAS,AMP,TIME,MT,                     BPS00600
```

```
     *STEP,PI,OMEGA,DELTA,NBAUD,BAUDD,                                          BPS00610
     *DSEED,MAXY,MINY,INT                                                       BPS00620
C                                                                               BPS00630
      DOUBLE PRECISION ARRAY(1024,2),SUMX(513),SUMXSQ(513),SUMX3(513),          BPS00640
     *SUMX4(513)                                                                BPS00650
      COMMON ARRAY,SUMX,SUMXSQ,SUMX3,SUMX4                                      BPS00660
C                                                                               BPS00670
      COMPLEX*16 X(513)                                                         BPS00680
C                                                                               BPS00690
      INTEGER IWK(10)                                                           BPS00700
C                                                                               BPS00710
CC    *** VARIABLE INITIALIZATION ***                                           BPS00720
C                                                                               BPS00730
      R=1                                                                       BPS00740
      TIME=0.D0                                                                 BPS00750
      STEP=1.D0/(2.D0*(FREQ*BAUD))                                              BPS00760
      PI=3.141592653589793D0                                                    BPS00770
      OMEGA=2.D0*PI*FREQ                                                        BPS00780
      DELTA=IPHAS*PI/180.D0                                                     BPS00790
      NBAUD=1.D0                                                                BPS00800
      BAUDD=1.D0/BAUD                                                           BPS00810
      MAXY=0.D0                                                                 BPS00820
      IND2P1=IN/2+1                                                             BPS00830
C                                                                               BPS00840
CC    *** DO THE MODULATION AND ASSIGN THE VALUES TO ARRAY ***                  BPS00850
C                                                                               BPS00860
      CALL STREAM(DSEED,ANS2,TYPE2,MT)                                          BPS00870
C                                                                               BPS00880
      DO 10 I=1,IN                                                              BPS00890
          ARRAY(R,1)=AMP*MT*DCOS((OMEGA*TIME)+DELTA)                            BPS00900
          ARRAY(R,2)=TIME                                                       BPS00910
          IF(DABS(ARRAY(R,1)).GT.MAXY)THEN                                      BPS00920
              MAXY=DABS(ARRAY(R,1))                                             BPS00930
          END IF                                                                BPS00940
          R=R+1                                                                 BPS00950
          TIME=TIME+STEP                                                        BPS00960
          IF(TIME.GT.NBAUD*BAUDD)THEN                                           BPS00970
              CALL STREAM(DSEED,ANS2,TYPE2,MT)                                  BPS00980
              NBAUD=NBAUD+1.D0                                                  BPS00990
          END IF                                                                BPS01000
10    CONTINUE                                                                  BPS01010
C                                                                               BPS01020
CC    *** PLOT THE TIME SERIES IF DESIRED ***                                   BPS01030
C                                                                               BPS01040
      IF(REP.EQ.1)THEN                                                          BPS01050
          MINY=-MAXY                                                            BPS01060
C                                                                               BPS01070
          CALL PLOT(MAXY,MINY,STEP,IN)                                          BPS01080
      END IF                                                                    BPS01090
C                                                                               BPS01100
```

```
C       *** GENERATE THE FFT ***                                                BPS01110
C                                                                               BPS01120
        CALL FFTRC(ARRAY(1,1),IN,X,IWK)                                         BPS01130
C                                                                               BPS01140
C       *** CALCULATE THE AMPLITUDE SPECTRUM OF THE FUNCTION ***                BPS01150
C       *** FIND THE NUMBER AND VALUE OF THE PRINCIPLE HARMONIC ***             BPS01160
C                                                                               BPS01170
        N=0.D0                                                                  BPS01180
        R=1                                                                     BPS01190
        MAXY=0.D0                                                               BPS01200
        DO 20 I=1,IND2P1                                                        BPS01210
            ARRAY(R,1)=CDABS(X(R))                                              BPS01220
            ARRAY(R,2)=N/STEP                                                   BPS01230
            IF(ARRAY(R,1).GT.MAXY)THEN                                          BPS01240
                MAXY=ARRAY(R,1)                                                 BPS01250
                RMAX=R-1                                                        BPS01260
            END IF                                                              BPS01270
            R=R+1                                                               BPS01280
            N=N+1.D0                                                            BPS01290
20      CONTINUE                                                                BPS01300
C                                                                               BPS01310
C       *** DISPLAY INFO IF THE FIRST TIME THROUGH SUBPROGRAM ***               BPS01320
C                                                                               BPS01330
        IF(REP.EQ.1)THEN                                                        BPS01340
C                                                                               BPS01350
29      WRITE(10,30)RMAX                                                        BPS01360
30      FORMAT(' THE FFT HAS BEEN GENERATED!  THE PRINCIPLE HARMONIC',/,        BPS01370
     *' IS THE',I5,' HARMONIC.  THE NEXT PLOT TO BE PRODUCED WILL',/,           BPS01380
     *' BE THE AMPLITUDE SPECTRUM.  ENTER A 1 IF YOU ARE READY TO',/,           BPS01390
     *' CONTINUE WITH THE PROGRAM.',//,' :')                                    BPS01400
C                                                                               BPS01410
        READ(5,*)L                                                              BPS01420
C                                                                               BPS01430
        IF(L.NE.1)THEN                                                          BPS01440
C           CALL FRTCMS('CLRSCRN ')                                             BPS01450
            WRITE(10,40)                                                        BPS01460
40          FORMAT(' ERROR',/)                                                  BPS01470
            GO TO 29                                                            BPS01480
        END IF                                                                  BPS01490
C                                                                               BPS01500
        END IF                                                                  BPS01510
C                                                                               BPS01520
C       *** ALLOW FOR THE PLOT OF THE AMPLITUDE SPECTRUM OF THE FFT ***         BPS01530
C                                                                               BPS01540
        IF(REP.EQ.1)THEN                                                        BPS01550
            MINY=0.D0                                                           BPS01560
            INT=1.D0/STEP                                                       BPS01570
C                                                                               BPS01580
            CALL PLOT(MAXY,MINY,INT,IND2P1)                                     BPS01590
        END IF                                                                  BPS01600
```

```
C                                                                       BPS01610
CC      *** ADD THE VALUES OF THE FFT TO THE ACCUMULATED STATISTICS ***  BPS01620
C                                                                       BPS01630
        FLAG=0                                                          BPS01640
        CALL STATS(IN,REP,FLAG)                                         BPS01650
C                                                                       BPS01660
        RETURN                                                          BPS01670
        END                                                             BPS01680
```

```
      SUBROUTINE DBPSK(TYPE2,BAUD,FREQ,IPHAS,AMP,ANS2,                  DBP00110
     *DSEED,IN,REP)                                                     DBP00120
C                                                                       DBP00130
C     *** PURPOSE ***                                                   DBP00140
C                                                                       DBP00150
C     THIS SUBROUTINE MODULATES THE CARRIER USING DIFFERENTIAL BINARY   DBP00160
C     SHIFT KEYING AS THE MODULATION TECHNIQUE.                         DBP00170
C                                                                       DBP00180
C     *** PARAMETER DEFINITIONS ***                                     DBP00190
C                                                                       DBP00200
C     TYPE2=       INDICATES LOGIC TYPE TO BE EMPLOYED                  DBP00210
C     BAUD=        SYMBOL RATE OR BAUD RATE                             DBP00220
C     FREQ=        CARRIER FREQUENCY                                    DBP00230
C     IPHAS=       INITIAL PHASE ANGLE IN DEGREES                       DBP00240
C     AMP=         AMPLITUDE                                            DBP00250
C     ANS2=        INTEGER VARIABLE TO BE PASSED TO STREAM              DBP00260
C     DSEED=       DOUBLE PRECISION SEED FOR RANDOM NUMBER GENERATOR    DBP00270
C     IN=          NUMBER OF POSITIONS OF ARRAY TO BE UTILIZED          DBP00280
C                  THE CALL THE THIS SUBROUTINE                         DBP00290
C     FLAG=        AN INTEGER WHICH CONTROL ENTRANCE AND EXIT FROM      DBP00300
C                                                                       DBP00310
C     *** VARIABLE DEFINITIONS ***                                      DBP00320
C                                                                       DBP00330
C     ARRAY=       A DOUBLE PRECISION ARRAY FOR PASSING THE BINARY      DBP00340
C                  DIGITS AND STORING THE VALUE OF THE TIME FUNCTION    DBP00350
C                  AND THE FFT                                          DBP00360
C     R=           AN INTEGER USED TO REPRESENT THE ROW OF ARRAY        DBP00370
C     TIME=        TIME VARIABLE                                        DBP00380
C     MT=          VALUE OF THE BINARY DIGIT                            DBP00390
C     STEP=        INTERVAL AT WHICH THE SIGNAL IS REPRODUCED           DBP00400
C     PI=          NUMERICAL CONSTANT                                   DBP00410
C     OMEGA=       CARRIER ANGULAR FREQUENCY                            DBP00420
C     DELTA=       INITIAL PHASE OFFSET IN RADIANS                      DBP00430
C     NBAUD=       A COUNT OF THE NUMBER OF BAUDS MODULATED             DBP00440
C     BAUDD=       BAUD DURATION                                        DBP00450
C     REF=         REFERENCE VOLTAGE                                    DBP00460
C     BK=          VALUE OF DIFFERENTIAL BIT TO BE MODULATED            DBP00470
C     X=           COMPLEX ARRAY TO RECEIVE THE FFT OF THE TIME SERIES  DBP00480
C     IWK=         INTEGER ARRAY USED IN SUBROUTINE FFTRC               DBP00490
C     MAXY=        MAXIMUM VALUE TO BE PLOTTED ON THE ABCISSAE          DBP00500
C     MINY=        MINIMUM VALUE TO BE PLOTTED ON THE ABCISSAE          DBP00510
C     KMAX=        PRINCIPLE HARMONIC                                   DBP00520
C     REP=         AN INTEGER EQUAL TO THE NUMBER OF THE REPETITION OF  DBP00530
C                  SUBPROGRAM STATS                                     DBP00540
C     IND2P1=      LENGTH OF ARRAY DIVIDED BY 2 + 1                     DBP00550
C                                                                       DBP00560
C     *** VARIABLE DECLARATIONS ***                                     DBP00570
C                                                                       DBP00580
      INTEGER R,ANS2,TYPE2,IN,KMAX,REP,FLAG,IND2P1                      DBP00590
C                                                                       DBP00600
```

```
      DOUBLE PRECISION BAUD,FREQ,IPHAS,AMP,TIME,MT,                     DBP00610
     *STEP,PI,OMEGA,DELTA,NBAUD,BAUDD,                                  DBP00620
     *DSEED,REF,BK,MAXY,MINY,INT                                        DBP00630
C                                                                       DBP00640
      DOUBLE PRECISION ARRAY(1024,2),SUMX(513),SUMXSQ(513),SUMX3(513),  DBP00650
     *SUMX4(513)                                                        DBP00660
      COMMON ARRAY,SUMX,SUMXSQ,SUMX3,SUMX4                              DBP00670
C                                                                       DBP00680
      COMPLEX*16 X(513)                                                 DBP00690
C                                                                       DBP00700
      INTEGER IWK(10)                                                   DBP00710
C                                                                       DBP00720
CC    *** VARIABLE INITIALIZATION ***                                   DBP00730
C                                                                       DBP00740
      R=1                                                               DBP00750
      TIME=0.D0                                                         DBP00760
      STEP=1.D0/(2.D0*(FREQ+BAUD))                                      DBP00770
      PI=3.141592653589793D0                                            DBP00780
      OMEGA=2.D0*PI*FREQ                                                DBP00790
      DELTA=IPHAS*PI/180.D0                                             DBP00800
      NBAUD=1.D0                                                        DBP00810
      BAUDD=1.D0/BAUD                                                   DBP00820
      REF=1.D0                                                          DBP00830
      MAXY=0.D0                                                         DBP00840
      IND2P1=IN/2+1                                                     DBP00850
C                                                                       DBP00860
CCC   *** DO THE MODULATION AND ASSIGN THE VALUES TO ARRAY ***          DBP00870
C                                                                       DBP00880
      CALL STREAM(DSEED,ANS2,TYPE2,MT)                                  DBP00890
C                                                                       DBP00900
      BK=REF*MT                                                         DBP00910
C                                                                       DBP00920
      DO 10 I=1,IN                                                      DBP00930
          ARRAY(R,1)=AMP*BK*DCOS((OMEGA*TIME)+DELTA)                    DBP00940
          ARRAY(R,2)=TIME                                               DBP00950
          IF(ARRAY(R,1).GT.MAXY)THEN                                    DBP00960
              MAXY=ARRAY(R,1)                                           DBP00970
          END IF                                                        DBP00980
          R=R+1                                                         DBP00990
          TIME=TIME+STEP                                                DBP01000
          IF(TIME.GT.NBAUD*BAUDD)THEN                                   DBP01010
              CALL STREAM(DSEED,ANS2,TYPE2,MT)                          DBP01020
              BK=BK*MT                                                  DBP01030
              NBAUD=NBAUD+1.D0                                          DBP01040
          END IF                                                        DBP01050
10    CONTINUE                                                          DBP01060
C                                                                       DBP01070
CCC   *** ALLOW FOR THE PLOT OF THE TIME SERIES ***                     DBP01080
C                                                                       DBP01090
      IF(REP.EQ.1)THEN                                                  DBP01100
```

```
            MINY=-MAXY                                                          DBP01110
C                                                                               DBP01120
            CALL PLOT(MAXY,MINY,STEP,IN)                                        DBP01130
        END IF                                                                  DBP01140
C                                                                               DBP01150
CC      *** GENERATE THE FFT ***                                                DBP01160
C                                                                               DBP01170
        CALL FFTRC(ARRAY(1,1),IN,X,IWK)                                         DBP01180
CC                                                                              DBP01190
CC      *** PRODUCE THE AMPLITUDE SPECTRUM OF THE FUNCTION ***                  DBP01200
CC      *** FIND THE NUMBER AND MAGNITUDE OF THE PRINCIPLE HARMONIC ***         DBP01210
C                                                                               DBP01220
        N=0.D0                                                                  DBP01230
        R=1                                                                     DBP01240
        MAXY=0.D0                                                               DBP01250
        DO 20 I=1,IND2P1                                                        DBP01260
            ARRAY(R,1)=CDABS(X(R))                                              DBP01270
            ARRAY(R,2)=N/STEP                                                   DBP01280
            IF(ARRAY(R,1).GT.MAXY)THEN                                          DBP01290
                MAXY=ARRAY(R,1)                                                 DBP01300
                RMAX=R-1                                                        DBP01310
            END IF                                                              DBP01320
            R=R+1                                                               DBP01330
            N=N+1.D0                                                            DBP01340
20      CONTINUE                                                                DBP01350
CC                                                                              DBP01360
CC      *** DISPLAY INFO IF THE FIRST TIME THROUGH THE SUBPROGRAM ***           DBP01370
C                                                                               DBP01380
        IF(REP.EQ.1)THEN                                                        DBP01390
C                                                                               DBP01400
29      WRITE(10,30)RMAX                                                        DBP01410
30      FORMAT(' THE FFT HAS BEEN GENERATED!  THE PRINCIPLE HARMONIC',/,        DBP01420
     *' IS THE',I5,' HARMONIC.  THE NEXT PLOT TO BE PRODUCED WILL',/,           DBP01430
     *' BE THE AMPLITUDE SPECTRUM.  ENTER A 1 IF YOU ARE READY TO',/,           DBP01440
     *' CONTINUE WITH THE PROGRAM.',//,' :')                                    DBP01450
C                                                                               DBP01460
        READ(5,*)L                                                              DBP01470
C                                                                               DBP01480
        IF(L.NE.1)THEN                                                          DBP01490
C           CALL FRTCMS('CLRSCRN ')                                             DBP01500
            WRITE(10,40)                                                        DBP01510
40          FORMAT(' ERROR',/)                                                  DBP01520
            GO TO 29                                                            DBP01530
        END IF                                                                  DBP01540
C                                                                               DBP01550
        END IF                                                                  DBP01560
CC                                                                              DBP01570
CC      *** ALLOW FOR THE PLOT OF THE AMPLITUDE SPECTRUM ***                    DBP01580
C                                                                               DBP01590
        IF(REP.EQ.1)THEN                                                        DBP01600
```

```
            MINY=0.D0                                                          DBP01610
            INT=1.D0/STEP                                                      DBP01620
C                                                                              DBP01630
            CALL PLOT(MAXY,MINY,INT,IND2P1)                                    DBP01640
         END IF                                                                DBP01650
C                                                                              DBP01660
CC       *** ADD THE VALUES OF THE FFT TO THE ACCUMULATED STATISTICS ***       DBP01670
C                                                                              DBP01680
         FLAG=0                                                                DBP01690
         CALL STATS(IN,REP,FLAG)                                               DBP01700
C                                                                              DBP01710
         RETURN                                                                DBP01720
         END                                                                   DBP01730
```

```
      SUBROUTINE OBPSK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,ANS2,                OBP00110
     *DSEED,IN,REP)                                                        OBP00120
C                                                                          OBP00130
C     *** PURPOSE ***                                                      OBP00140
C                                                                          OBP00150
C     THIS SUBROUTINE MODULATES THE CARRIER USING M-ARY PHASE SHIFT        OBP00160
C     KEYING AS THE MODULATION TECHNIQUE.                                  OBP00170
C                                                                          OBP00180
C     *** PARAMETER DEFINITIONS ***                                        OBP00190
C                                                                          OBP00200
C     TYPE2=       INDICATES LOGIC TYPE TO BE EMPLOYED                     OBP00210
C     BAUD=        SYMBOL RATE OR BAUD RATE                                OBP00220
C     BITS=        NUMBER OF BITS IN EACH BINARY CODE WORD                 OBP00230
C     FREQ=        CARRIER FREQUENCY                                       OBP00240
C     IPHAS=       INITIAL PHASE ANGLE IN DEGREES                          OBP00250
C     AMP=         AMPLITUDE                                               OBP00260
C     ANS2=        INTEGER VARIABLE TO BE PASSED TO STREAM                 OBP00270
C     DSEED=       DOUBLE PRECISION SEED FOR RANDOM NUMBER GENERATOR       OBP00280
C     IN=          NUMBER OF POSITIONS IN ARRAY TO BE UTILIZED             OBP00290
C     REP=         AN INTEGER EQUAL TO THE NUMBER OF THE REPETITION OF     OBP00300
C                  THE CALL THE THIS SUBROUTINE                            OBP00310
C                                                                          OBP00320
C     *** VARIABLE DEFINITIONS ***                                         OBP00330
C                                                                          OBP00340
C     ARRAY=       A DOUBLE PRECISION ARRAY FOR PASSING THE BINARY         OBP00350
C                  DIGITS AND STORING THE VALUE OF THE TIME FUNCTION       OBP00360
C                  AND THE FFT                                             OBP00370
C     R=           AN INTEGER USED TO REPRESENT THE ROW OF ARRAY           OBP00380
C     TIME=        TIME VARIABLE                                           OBP00390
C     MT=          VALUE OF THE BINARY DIGIT                               OBP00400
C     STEP=        INTERVAL AT WHICH THE SIGNAL IS REPRODUCED              OBP00410
C     PI=          NUMERICAL CONSTANT                                      OBP00420
C     OMEGA=       CARRIER ANGULAR FREQUENCY                               OBP00430
C     DELTA=       INITIAL PHASE OFFSET IN RADIANS                         OBP00440
C     NBIT=        A COUNT OF THE NUMBER OF BITS MODULATED                 OBP00450
C     BITR=        BIT RATE                                                OBP00460
C     BITD=        BIT DURATION                                            OBP00470
C     X=           COMPLEX ARRAY TO RECEIVE THE FFT OF THE TIME SERIES     OBP00480
C     IWK=         INTEGER WORKING ARRAY USED BY FUNCTION FFTRC            OBP00490
C     MAXY=        MAXIMUM VALUE TO BE PLOTTED ON THE ABCISSAE             OBP00500
C     MINY=        MINIMUM VALUE TO BE PLOTTED ON THE ABCISSAE             OBP00510
C     INT=         INTERVAL BETWEEN POINTS ON THE ORDINATE                 OBP00520
C     RMAX=        VALUE OF THE PRINCIPLE HARMONIC                         OBP00530
C     K=           INTEGER VALUE OF THE NUMBER OF BITS IN A BINARY WORD    OBP00540
C     KM1=         INTEGER VALUE OF K MINUS 1                              OBP00550
C     N=           NUMBER OF POSSIBLE BINARY CODE WORDS = 2**K             OBP00560
C     NDP=         DOUBLE PRECISION OF N                                   OBP00570
C     ND2=         NDP DIVIDED BY 2                                        OBP00580
C     SUM1=        VARIABLE USED TO HOLD THE DECIMAL EQUIVALENT TO A       OBP00590
C                  BINARY CODE WORD OF LENGTH K                            OBP00600
```

```
C      SUM2=        VARIABLE USED TO HOLD THE DECIMAL CONVERSION OF SUM1   OBP00610
CC     FLAG=        AN INTEGER WHICH CONTROL ENTRANCE AND EXIT FROM        OBP00620
CC                  SUBPROGRAM STATS AND ORTHO                            OBP00630
CC     IND2P1=      LENGTH OF ARRAY DIVIDED BY 2 + 1                      OBP00640
CCC                                                                       OBP00650
CC     *** VARIABLE DECLARATIONS ***                                      OBP00660
C                                                                         OBP00670
       INTEGER R,ANS2,TYPE2,IN,RMAX,K,KM1,REP,FLAG,IND2P1,N,C             OBP00680
C                                                                         OBP00690
       DOUBLE PRECISION BAUD,BITS,FREQ,IPHAS,AMP,TIME,MT,                 OBP00700
      *STEP,PI,OMEGA,DELTA,NBIT,BITD,BITR,                                OBP00710
      *DSEED,MAXY,MINY,INT,NDP,ND2,SUM1,SUM2                              OBP00720
C                                                                         OBP00730
       DOUBLE PRECISION ARRAY(1024,2),SUMX(513),SUMXSQ(513),SUMX3(513),   OBP00740
      *SUMX4(513)                                                         OBP00750
       COMMON ARRAY,SUMX,SUMXSQ,SUMX3,SUMX4                              OBP00760
C                                                                         OBP00770
       COMPLEX*16 X(513)                                                  OBP00780
C                                                                         OBP00790
       INTEGER IWK(10)                                                    OBP00800
CC                                                                        OBP00810
CC     *** VARIABLE INITIALIZATION ***                                    OBP00820
C                                                                         OBP00830
       R=1                                                                OBP00840
       K=BITS                                                             OBP00850
       N=2**K                                                             OBP00860
       NDP=N                                                              OBP00870
       NBIT=1.D0                                                          OBP00880
       BITR=N*BAUD                                                        OBP00890
       BITD=1.D0/BITR                                                     OBP00900
       TIME=0.D0                                                          OBP00910
       STEP=1.D0/(2.D0*(FREQ+BITR))                                       OBP00920
       PI=3.14159265358979D0                                              OBP00930
       OMEGA=2.D0*PI*FREQ                                                 OBP00940
       DELTA=IPHAS*PI/180.D0                                              OBP00950
       MAXY=0.D0                                                          OBP00960
       IND2P1=IN/2+1                                                      OBP00970
C                                                                         OBP00980
CCC    *** CONVERT A STREAM OF K BITS TO A SINGLE REPRESENTATIVE BINARY   OBP00990
CCC    VARIABLE IN DECIMAL FORM ***                                       OBP01000
C                                                                         OBP01010
       SUM1=0.D0                                                          OBP01020
       KM1=K-1                                                            OBP01030
       DO 10 I=1,K                                                        OBP01040
           CALL STREAM(DSEED,ANS2,TYPE2,MT)                               OBP01050
           IF(TYPE2.EQ.1.AND.MT.EQ.-1.D0)THEN                             OBP01060
               MT=0.D0                                                    OBP01070
           ELSE IF(TYPE2.EQ.3.AND.MT.EQ.-1.D0)THEN                        OBP01080
               MT=1.D0                                                    OBP01090
           END IF                                                         OBP01100
```

```
            SUM1=SUM1+(MT*(10.D0**KM1))                                    OBP01110
            KM1=KM1-1                                                      OBP01120
10      CONTINUE                                                           OBP01130
C                                                                          OBP01140
CC      *** CONVERT THE REPRESENTATIVE BINARY VARIABLE TO ITS DECIMAL      OBP01150
CC      EQUIVALENT ***                                                     OBP01160
C                                                                          OBP01170
        SUM2=0.D0                                                          OBP01180
        KM1=K-1                                                            OBP01190
        ND2=NDP/2.D0                                                       OBP01200
        DO 20 I=1,K                                                        OBP01210
            IF(SUM1/10.D0**KM1.GE.1.D0) THEN                               OBP01220
                SUM2=SUM2+ND2                                              OBP01230
                SUM1=SUM1-10.D0**KM1                                       OBP01240
                ND2=ND2/2.D0                                               OBP01250
                KM1=KM1-1                                                  OBP01260
            ELSE                                                           OBP01270
                ND2=ND2/2.D0                                               OBP01280
                KM1=KM1-1                                                  OBP01290
            END IF                                                         OBP01300
20      CONTINUE                                                           OBP01310
C                                                                          OBP01320
CC      *** DO THE MODULATION AND ASSIGN TO ARRAY ***                      OBP01330
C                                                                          OBP01340
        C=1                                                                OBP01350
        FLAG=0                                                             OBP01360
        CALL CRTHO(K,MT,SUM2,C,FLAG)                                       OBP01370
        FLAG=2                                                             OBP01380
        DO 30 I=1,IN                                                       OBP01390
            ARRAY(R,1)=AMP*MT*DCOS((OMEGA*TIME)+DELTA)                     OBP01400
            ARRAY(R,2)=TIME                                                OBP01410
            IF(ARRAY(R,1).GT.MAXY) THEN                                    OBP01420
                MAXY=ARRAY(R,1)                                            OBP01430
            END IF                                                         OBP01440
            R=R+1                                                          OBP01450
            TIME=TIME+STEP                                                 OBP01460
            IF(TIME.GT.NBIT*BITD) THEN                                     OBP01470
                NBIT=NBIT+1.D0                                             OBP01480
                IF(C.GT.N) THEN                                            OBP01490
C                                                                          OBP01500
CC      *** CONVERT A STREAM OF K BITS TO A SINGLE REPRESENTATIVE BINARY   OBP01510
CC      VARIABLE IN DECIMAL FORM ***                                       OBP01520
C                                                                          OBP01530
                    SUM1=0.D0                                              OBP01540
                    KM1=K-1                                                OBP01550
                    DO 40 J=1,K                                            OBP01560
                        CALL STREAM(DSEED,ANS2,TYPE2,MT)                   OBP01570
                        IF(TYPE2.EQ.1.AND.MT.EQ.-1.D0) THEN                OBP01580
                            MT=0.D0                                        OBP01590
                        ELSE IF(TYPE2.EQ.3.AND.MT.EQ.-1.D0) THEN           OBP01600
```

```
                        MT=1.D0                                                 OBP01610
                    END IF                                                      OBP01620
                    SUM1=SUM1+(MT*(10.D0**KM1))                                 OBP01630
                    KM1=KM1-1                                                   OBP01640
40              CONTINUE                                                        OBP01650
C                                                                               OBP01660
CC      *** CONVERT THE REPRESENTATIVE BINARY VARIABLE TO ITS DECIMAL           OBP01670
CC      EQUIVALENT ***                                                          OBP01680
C                                                                               OBP01690
                SUM2=0.D0                                                       OBP01700
                KM1=K-1                                                         OBP01710
                ND2=NDP/2.D0                                                    OBP01720
                DO 50 J=1,K                                                     OBP01730
                    IF(SUM1/10.D0**KM1.GE.1.D0) THEN                            OBP01740
                        SUM2=SUM2+ND2                                           OBP01750
                        SUM1=SUM1-10.D0**KM1                                    OBP01760
                        ND2=ND2/2.D0                                            OBP01770
                        KM1=KM1-1                                               OBP01780
                    ELSE                                                        OBP01790
                        ND2=ND2/2.D0                                            OBP01800
                        KM1=KM1-1                                               OBP01810
                    END IF                                                      OBP01820
50              CONTINUE                                                        OBP01830
                C=1                                                             OBP01840
                FLAG=1                                                          OBP01850
            END IF                                                              OBP01860
            CALL ORTHO(K,MT,SUM2,C,FLAG)                                        OBP01870
            FLAG=2                                                              OBP01880
        END IF                                                                  OBP01890
30      CONTINUE                                                                OBP01900
C                                                                               OBP01910
CC      *** PLOT THE TIME SERIES IF DESIRED ***                                 OBP01920
C                                                                               OBP01930
        IF(REP.EQ.1)THEN                                                        OBP01940
            MINY=-MAXY                                                          OBP01950
C                                                                               OBP01960
            CALL PLOT(MAXY,MINY,STEP,IN)                                        OBP01970
        END IF                                                                  OBP01980
C                                                                               OBP01990
CC      *** GENERATE THE FFT ***                                                OBP02000
C                                                                               OBP02010
        CALL FFTRC(ARRAY(1,1),IN,X,IWK)                                         OBP02020
C                                                                               OBP02030
CCC     *** CALCULATE THE AMPLITUDE SPECTRUM OF THE FUNCTION ***                OBP02040
CCC     *** FIND THE NUMBER AND VALUE OF THE PRINCIPLE HARMONIC ***             OBP02050
C                                                                               OBP02060
        N=0.D0                                                                  OBP02070
        R=1                                                                     OBP02080
        MAXY=0.D0                                                               OBP02090
        DO 60 I=1,IND2P1                                                        OBP02100
```

```
            ARRAY(R,1)=CDABS(X(R))                                          OBP02110
            ARRAY(R,2)=N/STEP                                               OBP02120
            IF(ARRAY(R,1).GT.MAXY)THEN                                      OBP02130
                MAXY=ARRAY(R,1)                                             OBP02140
                RMAX=R-1                                                    OBP02150
            END IF                                                          OBP02160
            R=R+1                                                           OBP02170
            N=N+1.D0                                                        OBP02180
   60     CONTINUE                                                         OBP02190
C                                                                          OBP02200
CC         *** DISPLAY INFO IF THE FIRST TIME THROUGH THE SUBPROGRAM ***   OBP02210
C                                                                          OBP02220
           IF(REP.EQ.1)THEN                                                OBP02230
C                                                                          OBP02240
   69      WRITE(10,70)RMAX                                                OBP02250
   70      FORMAT(' THE FFT HAS BEEN GENERATED!  THE PRINCIPLE HARMONIC',/,OBP02260
          *' IS THE',I5,' HARMONIC.  THE NEXT PLOT TO BE PRODUCED WILL',/,  OBP02270
          *' BE THE AMPLITUDE SPECTRUM.  ENTER A 1 IF YOU ARE READY TO',/,  OBP02280
          *' CONTINUE WITH THE PROGRAM.',//,' :')                           OBP02290
C                                                                          OBP02300
           READ(5,*)L                                                      OBP02310
C                                                                          OBP02320
           IF(L.NE.1)THEN                                                  OBP02330
C              CALL FRTCMS('CLRSCRN ')                                     OBP02340
               WRITE(10,80)                                                OBP02350
   80          FORMAT(' ERROR',/)                                          OBP02360
               GO TO 69                                                    OBP02370
           END IF                                                          OBP02380
C                                                                          OBP02390
           END IF                                                          OBP02400
C                                                                          OBP02410
CC         *** ALLOW FOR THE PLOT OF THE AMPLITUDE SPECTRUM OF THE FFT *** OBP02420
C                                                                          OBP02430
           IF(REP.EQ.1)THEN                                                OBP02440
               MINY=0.D0                                                   OBP02450
               INT=1.D0/STEP                                               OBP02460
C                                                                          OBP02470
               CALL PLOT(MAXY,MINY,INT,IND2P1)                             OBP02480
           END IF                                                          OBP02490
C                                                                          OBP02500
CC         *** ADD THE VALUES OF THE FFT TO THE ACCUMULATED STATISTICS *** OBP02510
C                                                                          OBP02520
           FLAG=0                                                          OBP02530
           CALL STATS(IN,REP,FLAG)                                         OBP02540
C                                                                          OBP02550
           RETURN                                                          OBP02560
           END                                                             OBP02570
```

```
      SUBROUTINE QPSK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,                   QPS00110
     *ANS2,DSEED,IN,REP)                                                QPS00120
C                                                                       QPS00130
C     *** PURPOSE ***                                                   QPS00140
C                                                                       QPS00150
C     THIS SUBROUTINE MODULATES THE CARRIER USING QUADRATURE PHASE SHIFTQPS00160
C     KEYING AS THE MODULATION TECHNIQUE.                               QPS00170
C                                                                       QPS00180
C     *** PARAMETER DEFINITIONS ***                                     QPS00190
C                                                                       QPS00200
C     TYPE2=       INDICATES LOGIC TYPE TO BE EMPLOYED                  QPS00210
C     BAUD=        SYMBOL RATE OR BAUD RATE                             QPS00220
C     BITS=        NUMBER OF BITS IN EACH BINARY CODE WORD              QPS00230
C     FREQ=        CARRIER FREQUENCY                                    QPS00240
C     IPHAS=       INITIAL PHASE ANGLE IN DEGREES                       QPS00250
C     AMP=         AMPLITUDE                                            QPS00260
C     ANS2=        INTEGER VARIABLE TO BE PASSED TO STREAM              QPS00270
C     DSEED=       DOUBLE PRECISION SEED FOR RANDOM NUMBER GENERATOR    QPS00280
C     IN=          NUMBER OF POSITIONS IN ARRAY TO BE UTILIZED          QPS00290
C     REP=         AN INTEGER EQUAL TO THE NUMBER OF THE REPETITION OF  QPS00300
C                  THE CALL THE THIS SUBROUTINE                         QPS00310
C                                                                       QPS00320
C     *** VARIABLE DEFINITIONS ***                                      QPS00330
C                                                                       QPS00340
C     ARRAY=       A DOUBLE PRECISION ARRAY FOR PASSING THE BINARY      QPS00350
C                  DIGITS AND STORING THE VALUE OF THE TIME FUNCTION    QPS00360
C                  AND THE FFT                                          QPS00370
C     R=           AN INTEGER USED TO REPRESENT THE ROW OF ARRAY        QPS00380
C     TIME=        TIME VARIABLE                                        QPS00390
C     MT=          VALUE OF BINARY DIGIT PASSED FROM SUBPROGRAM STREAM  QPS00400
C     MT1=         VALUE OF IN-PHASE BINARY DIGIT                       QPS00410
C     MT2=         VALUE OF QUADRATURE BINARY DIGIT                     QPS00420
C     STEP=        INTERVAL AT WHICH THE SIGNAL IS REPRODUCED           QPS00430
C     PI=          NUMERICAL CONSTANT                                   QPS00440
C     OMEGA=       CARRIER ANGULAR FREQUENCY                            QPS00450
C     DELTA=       INITIAL PHASE OFFSET IN RADIANS                      QPS00460
C     NBAUD=       A COUNT OF THE NUMBER OF BAUDS MODULATED             QPS00470
C     BAUDD=       BAUD DURATION                                        QPS00480
C     X=           COMPLEX ARRAY TO RECEIVE THE FFT OF THE TIME SERIES  QPS00490
C     IWK=         INTEGER WORKING ARRAY USED BY FUNCTION FFTRC         QPS00500
C     MAXY=        MAXIMUM VALUE TO BE PLOTTED ON THE ABCISSAE          QPS00510
C     MINY=        MINIMUM VALUE TO BE PLOTTED ON THE ABCISSAE          QPS00520
C     INT=         INTERVAL BETWEEN POINTS ON THE ORDINATE              QPS00530
C     RMAX=        VALUE OF THE PRINCIPLE HARMONIC                      QPS00540
C     FLAG=        AN INTEGER WHICH CONTROL ENTRANCE AND EXIT FROM       QPS00550
C                  SUBPROGRAM STATS                                     QPS00560
C     IND2P1=      LENGTH OF ARRAY DIVIDED BY 2 + 1                     QPS00570
C                                                                       QPS00580
C     *** VARIABLE DECLARATIONS ***                                     QPS00590
C                                                                       QPS00600
```

```
      INTEGER R,ANS2,TYPE2,IN,RMAX,REP,FLAG,IND2P1                      QPS00610
C                                                                       QPS00620
      DOUBLE PRECISION BAUD,BITS,FREQ,IPHAS,AMP,TIME,MT1,MT2,           QPS00630
     *STEP,PI,OMEGA,DELTA,NBAUD,BAUDD,                                  QPS00640
     *DSEED,MAXY,MINY,INT,MT                                            QPS00650
C                                                                       QPS00660
      DOUBLE PRECISION ARRAY(1024,2),SUMX(513),SUMXSQ(513),SUMX3(513),  QPS00670
     *SUMX4(513)                                                        QPS00680
      COMMON ARRAY,SUMX,SUMXSQ,SUMX3,SUMX4                              QPS00690
C                                                                       QPS00700
      COMPLEX*16 X(513)                                                 QPS00710
C                                                                       QPS00720
      INTEGER IWK(10)                                                   QPS00730
C                                                                       QPS00740
C     *** VARIABLE INITIALIZATION ***                                   QPS00750
C                                                                       QPS00760
      R=1                                                               QPS00770
      TIME=0.D0                                                         QPS00780
      STEP=1.D0/(2.D0*(FREQ+BAUD))                                      QPS00790
      PI=3.141592653589793D0                                            QPS00800
      OMEGA=2.D0*PI*FREQ                                                QPS00810
      DELTA=IPHAS*PI/180.D0                                             QPS00820
      NBAUD=1.D0                                                        QPS00830
      BAUDD=1.D0/BAUD                                                   QPS00840
      MAXY=0.D0                                                         QPS00850
      IND2P1=IN/2+1                                                     QPS00860
C                                                                       QPS00870
C     *** DO THE MODULATION AND ASSIGN THE VALUES TO ARRAY***           QPS00880
C                                                                       QPS00890
      CALL STREAM(DSEED,ANS2,TYPE2,MT)                                  QPS00900
      MT1=MT                                                            QPS00910
C                                                                       QPS00920
      CALL STREAM(DSEED,ANS2,TYPE2,MT)                                  QPS00930
      MT2=MT                                                            QPS00940
C                                                                       QPS00950
      DO 10 I=1,IN                                                      QPS00960
         ARRAY(R,1)=AMP*MT1*DCOS((OMEGA*TIME)+DELTA)+                   QPS00970
     *              AMP*MT2*DSIN((OMEGA*TIME)+DELTA)                    QPS00980
         ARRAY(R,2)=TIME                                                QPS00990
         IF(ARRAY(R,1).GT.MAXY) THEN                                    QPS01000
            MAXY=ARRAY(R,1)                                             QPS01010
         END IF                                                         QPS01020
         R=R+1                                                          QPS01030
         TIME=TIME+STEP                                                 QPS01040
         IF(TIME.GT.NBAUD*BAUDD)THEN                                    QPS01050
            CALL STREAM(DSEED,ANS2,TYPE2,MT)                            QPS01060
            MT1=MT                                                      QPS01070
            CALL STREAM(DSEED,ANS2,TYPE2,MT)                            QPS01080
            MT2=MT                                                      QPS01090
            NBAUD=NBAUD+1.D0                                            QPS01100
```

```
          END IF                                                                QPS01110
10        CONTINUE                                                              QPS01120
C                                                                               QPS01130
CC        *** PLOT THE TIME SERIES IF DESIRED ***                               QPS01140
C                                                                               QPS01150
          IF(REP.EQ.1)THEN                                                      QPS01160
               MINY=-MAXY                                                       QPS01170
C                                                                               QPS01180
               CALL PLOT(MAXY,MINY,STEP,IN)                                     QPS01190
          END IF                                                                QPS01200
C                                                                               QPS01210
CC        *** GENERATE THE FFT ***                                              QPS01220
C                                                                               QPS01230
          CALL FFTRC(ARRAY(1,1),IN,X,IWK)                                       QPS01240
C                                                                               QPS01250
CCC       *** CALCULATE THE AMPLITUDE SPECTRUM OF THE FUNCTION ***              QPS01260
CCC       *** FIND THE NUMBER AND VALUE OF THE PRINCIPLE HARMONIC ***           QPS01270
C                                                                               QPS01280
          N=0.D0                                                                QPS01290
          R=1                                                                   QPS01300
          MAXY=0.D0                                                             QPS01310
          DO 20 I=1,IND2P1                                                      QPS01320
               ARRAY(R,1)=CDABS(X(R))                                           QPS01330
               ARRAY(R,2)=N/STEP                                                QPS01340
               IF(ARRAY(R,1).GT.MAXY)THEN                                       QPS01350
                    MAXY=ARRAY(R,1)                                             QPS01360
                    RMAX=R-1                                                    QPS01370
               END IF                                                           QPS01380
               R=R+1                                                            QPS01390
               N=N+1.D0                                                         QPS01400
20        CONTINUE                                                              QPS01410
C                                                                               QPS01420
CC        *** DISPLAY INFO IF THE FIRST TIME THRCUGH THE SUBPROGRAM ***         QPS01430
C                                                                               QPS01440
          IF(REP.EQ.1)THEN                                                      QPS01450
C                                                                               QPS01460
29        WRITE(10,30)RMAX                                                      QPS01470
30        FORMAT(' THE FFT HAS BEEN GENERATED!  THE PRINCIPLE HARMONIC',/,      QPS01480
         *' IS THE',I5,' HARMONIC.  THE NEXT PLOT TO BE PRODUCED WILL',/,       QPS01490
         *' BE THE AMPLITUDE SPECTRUM.  ENTER A 1 IF YOU ARE READY TO',/,       QPS01500
         *' CONTINUE WITH THE PROGRAM.',//,' :')                                QPS01510
C                                                                               QPS01520
          READ(5,*)L                                                            QPS01530
C                                                                               QPS01540
          IF(L.NE.1)THEN                                                        QPS01550
C              CALL FRTCMS('CLRSCRN ')                                          QPS01560
               WRITE(10,40)                                                     QPS01570
40             FORMAT(' ERROR',/)                                               QPS01580
               GO TO 29                                                         QPS01590
          END IF                                                                QPS01600
```

```
C                                                                       QPS01610
      END IF                                                            QPS01620
C                                                                       QPS01630
CC    *** ALLOW FOR THE PLOT OF THE AMPLITUDE SPECTRUM OF THE FFT ***   QPS01640
C                                                                       QPS01650
      IF(REP.EQ.1)THEN                                                  QPS01660
          MINY=0.D0                                                     QPS01670
          INT=1.D0/STEP                                                 QPS01680
C                                                                       QPS01690
          CALL PLOT(MAXY,MINY,INT,IND2P1)                               QPS01700
      END IF                                                            QPS01710
C                                                                       QPS01720
CC    *** ADD THE VALUES OF THE FFT TO THE ACCUMULATED STATISTICS ***   QPS01730
C                                                                       QPS01740
      FLAG=0                                                            QPS01750
      CALL STATS(IN,REP,FLAG)                                           QPS01760
C                                                                       QPS01770
      RETURN                                                            QPS01780
      END                                                               QPS01790
```

```
      SUBROUTINE OQPSK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,                  OQP00110
     *ANS2,DSEED,IN,REP)                                                OQP00120
C                                                                        OQP00130
C     *** PURPOSE ***                                                    OQP00140
C                                                                        OQP00150
C     THIS SUBROUTINE MODULATES THE CARRIER USING OFFSET QUADRATURE      OQP00160
C     PHASE SHIFT KEYING AS THE MODULATION TECHNIQUE.                    OQP00170
C                                                                        OQP00180
C     *** PARAMETER DEFINITIONS ***                                      OQP00190
C                                                                        OQP00200
C     TYPE2=       INDICATES LOGIC TYPE TO BE EMPLOYED                   OQP00210
C     BAUD=        SYMBOL RATE OR BAUD RATE                              OQP00220
C     BITS=        NUMBER OF BITS IN EACH BINARY CODE WORD               OQP00230
C     FREQ=        CARRIER FREQUENCY                                     OQP00240
C     IPHAS=       INITIAL PHASE ANGLE IN DEGREES                        OQP00250
C     AMP=         AMPLITUDE                                             OQP00260
C     ANS2=        INTEGER VARIABLE TO BE PASSED TO STREAM               OQP00270
C     DSEED=       DOUBLE PRECISION SEED FOR RANDOM NUMBER GENERATOR     OQP00280
C     IN=          NUMBER OF POSITIONS IN ARRAY TO BE UTILIZED           OQP00290
C     REP=         AN INTEGER EQUAL TO THE NUMBER OF THE REPETITION OF   OQP00300
C                  THE CALL THE THIS SUBROUTINE                          OQP00310
C                                                                        OQP00320
C     *** VARIABLE DEFINITIONS ***                                       OQP00330
C                                                                        OQP00340
C     ARRAY=       A DOUBLE PRECISION ARRAY FOR PASSING THE BINARY       OQP00350
C                  DIGITS AND STORING THE VALUE OF THE TIME FUNCTION     OQP00360
C                  AND THE FFT                                           OQP00370
C     R=           AN INTEGER USED TO REPRESENT THE ROW OF ARRAY         OQP00380
C     TIME=        TIME VARIABLE                                         OQP00390
C     MT=          VALUE OF BINARY DIGIT PASSED FROM SUBPROGRAM STREAM   OQP00400
C     MT1=         VALUE OF IN-PHASE BINARY DIGIT                        OQP00410
C     MT2=         VALUE OF QUADRATURE BINARY DIGIT                      OQP00420
C     STEP=        INTERVAL AT WHICH THE SIGNAL IS REPRODUCED            OQP00430
C     PI=          NUMERICAL CONSTANT                                    OQP00440
C     OMEGA=       CARRIER ANGULAR FREQUENCY                             OQP00450
C     DELTA=       INITIAL PHASE OFFSET IN RADIANS                       OQP00460
C     NBAUD1=      A COUNT OF THE NUMBER OF BAUDS MODULATED IN-PHASE     OQP00470
C     NBAUD2=      A COUNT OF THE NUMBER OF BAUDS MODULATED QUADRATURE   OQP00480
C     X=           COMPLEX ARRAY TO RECEIVE THE FFT OF THE TIME SERIES   OQP00490
C     IWK=         INTEGER WORKING ARRAY USED BY FUNCTION FFTRC          OQP00500
C     MAXY=        MAXIMUM VALUE TO BE PLOTTED ON THE ABCISSAE           OQP00510
C     MINY=        MINIMUM VALUE TO BE PLOTTED ON THE ABCISSAE           OQP00520
C     INT=         INTERVAL BETWEEN POINTS ON THE ORDINATE               OQP00530
C     KMAX=        VALUE OF THE PRINCIPLE HARMONIC                       OQP00540
C     FLAG=        AN INTEGER WHICH CONTROL ENTRANCE AND EXIT FROM       OQP00550
C                  SUBPROGRAM STATS                                      OQP00560
C     IND2P1=      LENGTH OF ARRAY DIVIDED BY 2 + 1                      OQP00570
C                                                                        OQP00580
C     *** VARIABLE DECLARATIONS ***                                      OQP00590
C                                                                        OQP00600
```

```
      INTEGER R,ANS2,TYPE2,IN,RMAX,REP,FLAG,IND2P1                      OQP00610
C                                                                       OQP00620
      DOUBLE PRECISION BAUD,BITS,FREQ,IPHAS,AMP,TIME,MT1,MT2,           OQP00630
     *STEP,PI,OMEGA,DELTA,BAUDD,NBAUD1,NBAUD2,                          OQP00640
     *DSEED,MAXY,MINY,INT,MT                                            OQP00650
C                                                                       OQP00660
      DOUBLE PRECISION ARRAY(1024,2),SUMX(513),SUMXSQ(513),SUMX3(513),  OQP00670
     *SUMX4(513)                                                        OQP00680
      COMMON ARRAY,SUMX,SUMXSQ,SUMX3,SUMX4                              OQP00690
C                                                                       OQP00700
      COMPLEX*16 X(513)                                                 OQP00710
C                                                                       OQP00720
      INTEGER IWK(10)                                                   OQP00730
C                                                                       OQP00740
C     *** VARIABLE INITIALIZATION ***                                   OQP00750
C                                                                       OQP00760
      R=1                                                               OQP00770
      STEP=1.D0/(2.D0*(FREQ*BAUD))                                      OQP00780
      PI=3.14159265358979 3D0                                           OQP00790
      OMEGA=2.D0*PI*FREQ                                                OQP00800
      DELTA=IPHAS*PI/180.D0                                             OQP00810
      NBAUD1=1.D0                                                       OQP00820
      NBAUD2=1.D0                                                       OQP00830
      BAUDD=1.D0/BAUD                                                   OQP00840
      TIME=.5D0*BAUDD                                                   OQP00850
      MAXY=0.D0                                                         OQP00860
      IND2P1=IN/2+1                                                     OQP00870
C                                                                       OQP00880
C     *** DO THE MODULATION AND ASSIGN THE VALUES TO ARRAY ***          OQP00890
C                                                                       OQP00900
      CALL STREAM(DSEED,ANS2,TYPE2,MT)                                  OQP00910
      MT1=MT                                                            OQP00920
C                                                                       OQP00930
      CALL STREAM(DSEED,ANS2,TYPE2,MT)                                  OQP00940
      MT2=MT                                                            OQP00950
C                                                                       OQP00960
      DO 10 I=1,IN                                                      OQP00970
           ARRAY(R,1)=AMP*MT1*DCOS((OMEGA*TIME)+DELTA)+                 OQP00980
     *               AMP*MT2*DSIN((OMEGA*TIME)+DELTA)                   OQP00990
           ARRAY(R,2)=TIME                                              OQP01000
           IF(ARRAY(R,1).GT.MAXY) THEN                                  OQP01010
                MAXY=ARRAY(R,1)                                         OQP01020
           END IF                                                       OQP01030
           R=R+1                                                        OQP01040
           TIME=TIME+STEP                                               OQP01050
           IF(TIME.GT.NBAUD1*BAUDD) THEN                                OQP01060
                CALL STREAM(DSEED,ANS2,TYPE2,MT)                        OQP01070
                MT1=MT                                                  OQP01080
                NBAUD1=NBAUD1+1.D0                                      OQP01090
           END IF                                                       OQP01100
```

```
C                                                                       OQP01110
      IF(TIME.GT.((NBAUD2*BAUDD)+(BAUDD/2.D0)))THEN                     OQP01120
          CALL STREAM(DSEED,ANS2,TYPE2,MT)                              OQP01130
          MT2=MT                                                        OQP01140
          NBAUD2=NBAUD2+1.D0                                            OQP01150
      END IF                                                            OQP01160
10    CONTINUE                                                          OQP01170
C                                                                       OQP01180
CC    *** PLOT THE TIME SERIES IF DESIRED ***                           OQP01190
C                                                                       OQP01200
      IF(REP.EQ.1)THEN                                                  OQP01210
          MINY=-MAXY                                                    OQP01220
C                                                                       OQP01230
          CALL PLOT(MAXY,MINY,STEP,IN)                                  OQP01240
      END IF                                                            OQP01250
CC                                                                      OQP01260
CC    *** GENERATE THE FFT ***                                          OQP01270
C                                                                       OQP01280
      CALL FFTRC(ARRAY(1,1),IN,X,IWK)                                   OQP01290
C                                                                       OQP01300
CC    *** CALCULATE THE AMPLITUDE SPECTRUM OF THE FUNCTION ***          OQP01310
CC    *** FIND THE NUMBER AND VALUE OF THE PRINCIPLE HARMONIC ***       OQP01320
C                                                                       OQP01330
      N=0.D0                                                            OQP01340
      R=1                                                               OQP01350
      MAXY=0.D0                                                         OQP01360
      DO 20 I=1,IND2P1                                                  OQP01370
          ARRAY(R,1)=CDABS(X(R))                                        OQP01380
          ARRAY(R,2)=N/STEP                                             OQP01390
          IF(ARRAY(R,1).GT.MAXY)THEN                                    OQP01400
              MAXY=ARRAY(R,1)                                           OQP01410
              RMAX=R-1                                                  OQP01420
          END IF                                                        OQP01430
          R=R+1                                                         OQP01440
          N=N+1.D0                                                      OQP01450
20    CONTINUE                                                          OQP01460
C                                                                       OQP01470
CC    *** DISPLAY INFO IF THE FIRST TIME THROUGH THE SUBPROGRAM ***     OQP01480
C                                                                       OQP01490
      IF(REP.EQ.1)THEN                                                  OQP01500
C                                                                       OQP01510
29    WRITE(10,30)RMAX                                                  OQP01520
30    FORMAT(' THE FFT HAS BEEN GENERATED!  THE PRINCIPLE HARMONIC',/,  OQP01530
     *' IS THE',I5,' HARMONIC.  THE NEXT PLOT TO BE PRODUCED WILL',/,   OQP01540
     *' BE THE AMPLITUDE SPECTRUM.  ENTER A 1 IF YOU ARE READY TO',/,   OQP01550
     *' CONTINUE WITH THE PROGRAM.',//,' :')                            OQP01560
C                                                                       OQP01570
      READ(5,*)L                                                        OQP01580
C                                                                       OQP01590
      IF(L.NE.1)THEN                                                    OQP01600
```

```
C           CALL FRTCMS('CLRSCRN ')                                          OQP01610
            WRITE(10,40)                                                     OQP01620
40          FORMAT(' ERROR',/)                                               OQP01630
            GO TO 29                                                         OQP01640
        END IF                                                               OQP01650
C                                                                            OQP01660
        END IF                                                               OQP01670
C                                                                            OQP01680
C       *** ALLOW FOR THE PLOT OF THE AMPLITUDE SPECTRUM OF THE FFT ***      OQP01690
C                                                                            OQP01700
        IF(REP.EQ.1)THEN                                                     OQP01710
            MINY=0.D0                                                        OQP01720
            INT=1.D0/STEP                                                    OQP01730
C                                                                            OQP01740
            CALL PLOT(MAXY,MINY,INT,IND2P1)                                  OQP01750
        END IF                                                               OQP01760
C                                                                            OQP01770
C       *** ADD THE VALUES OF THE FFT TO THE ACCUMULATED STATISTICS ***      OQP01780
C                                                                            OQP01790
        FLAG=0                                                               OQP01800
        CALL STATS(IN,REP,FLAG)                                              OQP01810
C                                                                            OQP01820
        RETURN                                                               OQP01830
        END                                                                  OQP01840
```

```
      SUBROUTINE MPSK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,ANS2,                MPS00110
     *DSEED,IN,REP)                                                       MPS00120
C                                                                         MPS00130
C     *** PURPOSE ***                                                     MPS00140
C                                                                         MPS00150
C     THIS SUBROUTINE MODULATES THE CARRIER USING M-ARY PHASE SHIFT       MPS00160
C     KEYING AS THE MODULATION TECHNIQUE.                                 MPS00170
C                                                                         MPS00180
C     *** PARAMETER DEFINITIONS ***                                       MPS00190
C                                                                         MPS00200
C     TYPE2=       INDICATES LOGIC TYPE TO BE EMPLOYED                    MPS00210
C     BAUD=        SYMBOL RATE OR BAUD RATE                               MPS00220
C     BITS=        NUMBER OF BITS IN EACH BINARY CODE WORD                MPS00230
C     FREQ=        CARRIER FREQUENCY                                      MPS00240
C     IPHAS=       INITIAL PHASE ANGLE IN DEGREES                         MPS00250
C     AMP=         AMPLITUDE                                              MPS00260
C     ANS2=        INTEGER VARIABLE TO BE PASSED TO STREAM                MPS00270
C     DSEED=       DOUBLE PRECISION SEED FOR RANDOM NUMBER GENERATOR      MPS00280
C     IN=          NUMBER OF POSITIONS IN ARRAY TO BE UTILIZED            MPS00290
C     REP=         AN INTEGER EQUAL TO THE NUMBER OF THE REPETITION OF    MPS00300
C                  THE CALL THE THIS SUBROUTINE                           MPS00310
C                                                                         MPS00320
C     *** VARIABLE DEFINITIONS ***                                        MPS00330
C                                                                         MPS00340
C     ARRAY=       A DOUBLE PRECISION ARRAY FOR PASSING THE BINARY        MPS00350
C                  DIGITS AND STORING THE VALUE OF THE TIME FUNCTION      MPS00360
C                  AND THE FFT                                            MPS00370
C     R=           AN INTEGER USED TO REPRESENT THE ROW OF ARRAY           MPS00380
C     TIME=        TIME VARIABLE                                          MPS00390
C     MT=          VALUE OF THE BINARY DIGIT                              MPS00400
C     STEP=        INTERVAL AT WHICH THE SIGNAL IS REPRODUCED              MPS00410
C     PI=          NUMERICAL CONSTANT                                     MPS00420
C     OMEGA=       CARRIER ANGULAR FREQUENCY                              MPS00430
C     DELTA=       INITIAL PHASE OFFSET IN RADIANS                        MPS00440
C     NBAUD=       A COUNT OF THE NUMBER OF BAUDS MODULATED               MPS00450
C     BAUDD=       BAUD DURATION                                          MPS00460
C     X=           COMPLEX ARRAY TO RECEIVE THE FFT OF THE TIME SERIES    MPS00470
C     IWK=         INTEGER WORKING ARRAY USED BY FUNCTION FFTRC           MPS00480
C     MAXY=        MAXIMUM VALUE TO BE PLOTTED ON THE ABCISSAE            MPS00490
C     MINY=        MINIMUM VALUE TO BE PLOTTED ON THE ABCISSAE            MPS00500
C     INT=         INTERVAL BETWEEN POINTS ON THE ORDINATE                MPS00510
C     KMAX=        VALUE OF THE PRINCIPLE HARMONIC                        MPS00520
C     K=           INTEGER VALUE OF THE NUMBER OF BITS IN A BINARY WORD   MPS00530
C     KM1=         INTEGER VALUE OF K MINUS 1                             MPS00540
C     N=           NUMBER OF POSSIBLE BINARY CODE WORDS = 2**K            MPS00550
C     ND2=         N DIVIDED BY 2                                         MPS00560
C     SUM1=        VARIABLE USED TO HOLD THE DECIMAL EQUIVALENT TO A      MPS00570
C                  BINARY CODE WORD OF LENGTH K                           MPS00580
C     SUM2=        VARIABLE USED TO HOLD THE DECIMAL CONVERSION OF SUM1   MPS00590
C     J=           INTEGER USED TO INDEX A LOOP                           MPS00600
```

```
C      FLAG=        AN INTEGER WHICH CONTROL ENTRANCE AND EXIT FROM        MPS00610
C                   SUBPROGRAM STATS                                       MPS00620
C      IND2P1=      LENGTH OF ARRAY DIVIDED BY 2 + 1                       MPS00630
C                                                                          MPS00640
CC     *** VARIABLE DECLARATIONS ***                                       MPS00650
C                                                                          MPS00660
       INTEGER R,ANS2,TYPE2,IN,RMAX,K,KM1,J,REP,FLAG,IND2P1                MPS00670
C                                                                          MPS00680
       DOUBLE PRECISION BAUD,BITS,FREQ,IPHAS,AMP,TIME,MT,                  MPS00690
      *STEP,PI,OMEGA,DELTA,NBAUD,BAUDD,                                    MPS00700
      *DSEED,MAXY,MINY,INT,N,ND2,SUM1,SUM2                                 MPS00710
C                                                                          MPS00720
       DOUBLE PRECISION ARRAY(1024,2),SUMX(513),SUMXSQ(513),SUMX3(513),    MPS00730
      *SUMX4(513)                                                          MPS00740
       COMMON ARRAY,SUMX,SUMXSQ,SUMX3,SUMX4                               MPS00750
C                                                                          MPS00760
       COMPLEX*16 Y(513)                                                   MPS00770
C                                                                          MPS00780
       INTEGER IWK(10)                                                     MPS00790
CC                                                                         MPS00800
CC     *** VARIABLE INITIALIZATION ***                                     MPS00810
C                                                                          MPS00820
       R=1                                                                 MPS00830
       TIME=0.D0                                                           MPS00840
       STEP=1.D0/(2.D0*(FREQ+BAUD))                                        MPS00850
       PI=3.141592653589793D0                                              MPS00860
       OMEGA=2.D0*PI*FREQ                                                  MPS00870
       DELTA=IPHAS*PI/180.D0                                               MPS00880
       NBAUD=1.D0                                                          MPS00890
       BAUDD=1.D0/BAUD                                                     MPS00900
       K=BITS                                                              MPS00910
       N=2.D0**K                                                           MPS00920
       MAXY=0.D0                                                           MPS00930
       IND2P1=IN/2+1                                                       MPS00940
C                                                                          MPS00950
9      IF(R.LE.IN) THEN                                                    MPS00960
CC                                                                         MPS00970
CC     *** CONVERT A STREAM OF K BITS TO A SINGLE REPRESENTATIVE BINARY    MPS00980
CC     VARIABLE IN DECIMAL FORM ***                                        MPS00990
C                                                                          MPS01000
            SUM1=0.D0                                                      MPS01010
            KM1=K-1                                                        MPS01020
            DO 10 I=1,K                                                    MPS01030
               CALL STREAM(DSEED,ANS2,TYPE2,MT)                            MPS01040
               IF(TYPE2.EQ.1.AND.MT.EQ.-1.D0) THEN                         MPS01050
                  MT=0.D0                                                  MPS01060
               ELSE IF(TYPE2.EQ.3.AND.MT.EQ.-1.D0)THEN                     MPS01070
                  MT=1.D0                                                  MPS01080
               END IF                                                      MPS01090
               SUM1=SUM1+(MT*(10.D0**KM1))                                 MPS01100
```

```
                KM1=KM1-1                                                       MPS01110
10          CONTINUE                                                            MPS01120
C                                                                               MPS01130
C       *** CONVERT THE REPRESENTATIVE BINARY VARIABLE TO ITS DECIMAL           MPS01140
C       EQUIVALENT ***                                                          MPS01150
C                                                                               MPS01160
            SUM2=0.D0                                                           MPS01170
            KM1=K-1                                                             MPS01180
            ND2=N/2.D0                                                          MPS01190
            DO 20 I=1,K                                                         MPS01200
                IF(SUM1/10.D0**KM1.GE.1.D0)THEN                                 MPS01210
                    SUM2=SUM2+ND2                                               MPS01220
                    SUM1=SUM1-10.D0**KM1                                        MPS01230
                    ND2=ND2/2.D0                                                MPS01240
                    KM1=KM1-1                                                   MPS01250
                ELSE                                                            MPS01260
                    ND2=ND2/2.D0                                                MPS01270
                    KM1=KM1-1                                                   MPS01280
                END IF                                                          MPS01290
20          CONTINUE                                                            MPS01300
C                                                                               MPS01310
C       *** DO THE MODULATION AND ASSIGN TO ARRAY ***                           MPS01320
C                                                                               MPS01330
            J=R                                                                 MPS01340
            DO 30 I=J,IN                                                        MPS01350
                ARRAY(R,1)=AMP*DCOS((OMEGA*TIME)+                               MPS01360
        *                   ((SUM2*2.D0*PI)/N)+DELTA)                           MPS01370
                ARRAY(R,2)=TIME                                                 MPS01380
                IF(ARRAY(R,1).GT.MAXY)THEN                                      MPS01390
                    MAXY=ARRAY(R,1)                                             MPS01400
                END IF                                                          MPS01410
                R=R+1                                                           MPS01420
                TIME=TIME+STEP                                                  MPS01430
                IF(TIME.GT.NBAUD*BAUDD)THEN                                     MPS01440
                    NBAUD=NBAUD+1.D0                                            MPS01450
                    GO TO 9                                                     MPS01460
                END IF                                                          MPS01470
30          CONTINUE                                                            MPS01480
        END IF                                                                  MPS01490
C                                                                               MPS01500
C       *** PLOT THE TIME SERIES IF DESIRED ***                                 MPS01510
C                                                                               MPS01520
        IF(REP.EQ.1)THEN                                                        MPS01530
            MINY=-MAXY                                                          MPS01540
C                                                                               MPS01550
            CALL PLOT(MAXY,MINY,STEP,IN)                                        MPS01560
        END IF                                                                  MPS01570
C                                                                               MPS01580
C       *** GENERATE THE FFT ***                                                MPS01590
C                                                                               MPS01600
```

```
      CALL FFTRC(ARRAY(1,1),IN,X,IWK)                                   MPS01610
C                                                                       MPS01620
C     *** CALCULATE THE AMPLITUDE SPECTRUM OF THE FUNCTION ***          MPS01630
C     *** FIND THE NUMBER AND VALUE OF THE PRINCIPLE HARMONIC ***       MPS01640
C                                                                       MPS01650
      N=0.D0                                                            MPS01660
      R=1                                                               MPS01670
      MAXY=0.D0                                                         MPS01680
      DO 40 I=1,IND2P1                                                  MPS01690
         ARRAY(R,1)=CDABS(X(R))                                         MPS01700
         ARRAY(R,2)=N/STEP                                              MPS01710
         IF(ARRAY(R,1).GT.MAXY)THEN                                     MPS01720
            MAXY=ARRAY(R,1)                                             MPS01730
            RMAX=R-1                                                    MPS01740
         END IF                                                         MPS01750
         R=R+1                                                          MPS01760
         N=N+1.D0                                                       MPS01770
40    CONTINUE                                                          MPS01780
C                                                                       MPS01790
C     *** DISPLAY INFO IF THE FIRST TIME THROUGH THE SUBPROGRAM ***     MPS01800
C                                                                       MPS01810
      IF(REP.EQ.1)THEN                                                  MPS01820
C                                                                       MPS01830
49    WRITE(10,50)RMAX                                                  MPS01840
50    FORMAT(' THE FFT HAS BEEN GENERATED!  THE PRINCIPLE HARMONIC',/,  MPS01850
     *' IS THE',I5,' HARMONIC.  THE NEXT PLOT TO BE PRODUCED WILL',/,   MPS01860
     *' BE THE AMPLITUDE SPECTRUM.  ENTER A 1 IF YOU ARE READY TO',/,   MPS01870
     *' CONTINUE WITH THE PROGRAM.',//,' :')                            MPS01880
C                                                                       MPS01890
      READ(5,*)L                                                        MPS01900
C                                                                       MPS01910
      IF(L.NE.1)THEN                                                    MPS01920
C        CALL FRTCMS('CLRSCRN ')                                        MPS01930
         WRITE(10,60)                                                   MPS01940
60       FORMAT(' ERROR',/)                                             MPS01950
         GO TO 49                                                       MPS01960
      END IF                                                            MPS01970
C                                                                       MPS01980
      END IF                                                            MPS01990
C                                                                       MPS02000
C     *** ALLOW FOR THE PLOT OF THE AMPLITUDE SPECTRUM OF THE FFT ***   MPS02010
C                                                                       MPS02020
      IF(REP.EQ.1)THEN                                                  MPS02030
         MINY=0.D0                                                      MPS02040
         INT=1.D0/STEP                                                  MPS02050
C                                                                       MPS02060
         CALL PLOT(MAXY,MINY,INT,IND2P1)                                MPS02070
      END IF                                                            MPS02080
C                                                                       MPS02090
C     *** ADD THE VALUES OF THE FFT TO THE ACCUMULATED STATISTICS ***   MPS02100
```

```
C
      FLAG=0                                                          MPS02110
      CALL STATS(IN,REP,FLAG)                                         MPS02120
C                                                                     MPS02130
      RETURN                                                          MPS02140
      END                                                             MPS02150
                                                                      MPS02160
```

```
      SUBROUTINE MASK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,ANS2,                MAS00110
     *DSEED,IN,REP)                                                       MAS00120
C                                                                         MAS00130
CC    *** PURPOSE ***                                                     MAS00140
CC                                                                        MAS00150
CC    THIS SUBROUTINE MODULATES THE CARRIER USING M-ARY AMPLITUDE SHIFT    MAS00160
CC    KEYING AS THE MODULATION TECHNIQUE.                                 MAS00170
CC                                                                        MAS00180
CC    *** PARAMETER DEFINITIONS ***                                       MAS00190
CC                                                                        MAS00200
CC    TYPE2=       INDICATES LOGIC TYPE TO BE EMPLOYED                    MAS00210
CC    BAUD=        SYMBOL RATE OR BAUD RATE                               MAS00220
CC    BITS=        NUMBER OF BITS IN EACH BINARY CODE WORD                MAS00230
CC    FREQ=        CARRIER FREQUENCY                                      MAS00240
CC    IPHAS=       INITIAL PHASE ANGLE IN DEGREES                         MAS00250
CC    AMP=         AMPLITUDE                                              MAS00260
CC    ANS2=        INTEGER VARIABLE TO BE PASSED TO STREAM                MAS00270
CC    DSEED=       DOUBLE PRECISION SEED FOR RANDOM NUMBER GENERATOR      MAS00280
CC    IN=          NUMBER OF POSITIONS IN ARRAY TO BE UTILIZED            MAS00290
CC                 THE CALL THE THIS SUBROUTINE                           MAS00300
CC    FLAG=        AN INTEGER WHICH CONTROL ENTRANCE AND EXIT FROM        MAS00310
CC                                                                        MAS00320
CC    *** VARIABLE DEFINITIONS ***                                        MAS00330
CC                                                                        MAS00340
CC    ARRAY=       A DOUBLE PRECISION ARRAY FOR PASSING THE BINARY        MAS00350
CC                 DIGITS AND STORING THE VALUE OF THE TIME FUNCTION      MAS00360
CC                 AND THE FFT                                            MAS00370
CC    R=           AN INTEGER USED TO REPRESENT THE ROW OF ARRAY          MAS00380
CC    TIME=        TIME VARIABLE                                          MAS00390
CC    MT=          VALUE OF THE BINARY DIGIT                              MAS00400
CC    STEP=        INTERVAL AT WHICH THE SIGNAL IS REPRODUCED             MAS00410
CC    PI=          NUMERICAL CONSTANT                                     MAS00420
CC    OMEGA=       CARRIER ANGULAR FREQUENCY                              MAS00430
CC    DELTA=       INITIAL PHASE OFFSET IN RADIANS                        MAS00440
CC    NBAUD=       A COUNT OF THE NUMBER OF BAUDS MODULATED               MAS00450
CC    BAUDD=       BAUD DURATION                                          MAS00460
CC    X=           COMPLEX ARRAY TO RECEIVE THE FFT OF THE TIME SERIES    MAS00470
CC    IWK=         INTEGER WORKING ARRAY USED BY FUNCTION FFTRC           MAS00480
CC    MAXY=        MAXIMUM VALUE TO BE PLOTTED ON THE ABCISSAE            MAS00490
CC    MINY=        MINIMUM VALUE TO BE PLOTTED ON THE ABCISSAE            MAS00500
CC    INT=         INTERVAL BETWEEN POINTS ON THE ORDINATE                MAS00510
CC    RMAX=        VALUE OF THE PRINCIPLE HARMONIC                        MAS00520
CC    K=           INTEGER VALUE OF THE NUMBER OF BITS IN A BINARY WORD   MAS00530
CC    KM1=         INTEGER VALUE OF K MINUS 1                             MAS00540
CC    N=           NUMBER OF POSSIBLE BINARY CODE WORDS = 2**K            MAS00550
CC    ND2=         N DIVIDED BY 2                                         MAS00560
CC    SUM1=        VARIABLE USED TO HOLD THE DECIMAL EQUIVALENT TO A      MAS00570
CC                 BINARY CODE WORD OF LENGTH K                           MAS00580
CC    SUM2=        VARIABLE USED TO HOLD THE DECIMAL CONVERSION OF SUM1   MAS00590
C     J=           INTEGER USED TO INDEX A LOOP                           MAS00600
```

```
C      AMP1=        PORTION OF AMP USED TO MODULATE THE CARRIER DURING     MAS00610
C                   ANY RESPECTIVE BAUD DURATION                           MAS00620
C      REP=         AN INTEGER EQUAL TO THE NUMBER OF THE REPETITION OF    MAS00630
C                   SUBPROGRAM STATS                                       MAS00640
C      IND2P1=      LENGTH OF ARRAY DIVIDED BY 2 + 1                       MAS00650
C                                                                          MAS00660
C      *** VARIABLE DECLARATIONS ***                                       MAS00670
C                                                                          MAS00680
       INTEGER R,ANS2,TYPE2,IN,RMAX,K,KM1,J,REP,FLAG,IND2P1               MAS00690
C                                                                          MAS00700
       DOUBLE PRECISION BAUD,BITS,FREQ,IPHAS,AMP,TIME,MT,                 MAS00710
      *STEP,PI,OMEGA,DELTA,NBAUD,BAUDD,                                   MAS00720
      *DSEED,MAXY,MINY,INT,N,ND2,SUM1,SUM2,AMP1                           MAS00730
C                                                                          MAS00740
       DOUBLE PRECISION ARRAY(1024,2),SUMX(513),SUMXSQ(513),SUMX3(513),   MAS00750
      *SUMX4(513)                                                         MAS00760
       COMMON ARRAY,SUMX,SUMXSQ,SUMX3,SUMX4                               MAS00770
C                                                                          MAS00780
       COMPLEX*16 X(513)                                                  MAS00790
C                                                                          MAS00800
       INTEGER IWK(10)                                                    MAS00810
C                                                                          MAS00820
C      *** VARIABLE INITIALIZATION ***                                     MAS00830
C                                                                          MAS00840
       R=1                                                                MAS00850
       TIME=0.D0                                                          MAS00860
       STEP=1.D0/(2.D0*(FREQ+BAUD))                                       MAS00870
       PI=3.14159265358979 3D0                                            MAS00880
       OMEGA=2.D0*PI*FREQ                                                 MAS00890
       DELTA=IPHAS*PI/180.D0                                              MAS00900
       NBAUD=1.D0                                                         MAS00910
       BAUDD=1.D0/BAUD                                                    MAS00920
       K=BITS                                                             MAS00930
       N=2.D0**K                                                          MAS00940
       MAXY=0.D0                                                          MAS00950
       IND2P1=IN/2+1                                                      MAS00960
C                                                                         MAS00970
9      IF(R.LE.IN) THEN                                                   MAS00980
C                                                                          MAS00990
C      *** CONVERT A STREAM OF K BITS TO A SINGLE REPRESENTATIVE BINARY    MAS01000
C      VARIABLE IN DECIMAL FORM ***                                        MAS01010
C                                                                          MAS01020
           SUM1=0.D0                                                      MAS01030
           KM1=K-1                                                        MAS01040
           DO 10 I=1,K                                                    MAS01050
               CALL STREAM(DSEED,ANS2,TYPE2,MT)                           MAS01060
               IF(TYPE2.EQ.1.AND.MT.EQ.-1.D0) THEN                        MAS01070
                   MT=0.D0                                                MAS01080
               ELSE IF(TYPE2.EQ.3.AND.MT.EQ.-1.D0) THEN                   MAS01090
                   MT=1.D0                                                MAS01100
```

```
                END IF                                                          MAS01110
                SUM1=SUM1+(MT*(10.D0**KM1))                                     MAS01120
                KM1=KM1-1                                                       MAS01130
10          CONTINUE                                                            MAS01140
C                                                                               MAS01150
CC      *** CONVERT THE REPRESENTATIVE BINARY VARIABLE TO ITS DECIMAL           MAS01160
CC      EQUIVALENT ***                                                          MAS01170
C                                                                               MAS01180
            SUM2=0.D0                                                           MAS01190
            KM1=K-1                                                             MAS01200
            ND2=N/2.D0                                                          MAS01210
            DO 20 I=1,K                                                         MAS01220
                IF(SUM1/10.D0**KM1.GE.1.D0)THEN                                 MAS01230
                    SUM2=SUM2+ND2                                               MAS01240
                    SUM1=SUM1-10.D0**KM1                                        MAS01250
                    ND2=ND2/2.D0                                                MAS01260
                    KM1=KM1-1                                                   MAS01270
                ELSE                                                            MAS01280
                    ND2=ND2/2.D0                                                MAS01290
                    KM1=KM1-1                                                   MAS01300
                END IF                                                          MAS01310
20          CONTINUE                                                            MAS01320
C                                                                               MAS01330
CC      *** DO THE MODULATION AND ASSIGN TO ARRAY ***                           MAS01340
C                                                                               MAS01350
            SUM2=SUM2+1.D0                                                      MAS01360
            AMP1=AMP/SUM2                                                       MAS01370
            J=R                                                                 MAS01380
            DO 30 I=J,IN                                                        MAS01390
                ARRAY(R,1)=AMP1*DCOS((OMEGA*TIME)+DELTA)                        MAS01400
                ARRAY(R,2)=TIME                                                 MAS01410
                IF(ARRAY(R,1).GT.MAXY)THEN                                      MAS01420
                    MAXY=ARRAY(R,1)                                             MAS01430
                END IF                                                          MAS01440
                R=R+1                                                           MAS01450
                TIME=TIME+STEP                                                  MAS01460
                IF(TIME.GT.NBAUD*BAUDD)THEN                                     MAS01470
                    NBAUD=NBAUD+1.D0                                            MAS01480
                    GO TO 9                                                     MAS01490
                END IF                                                          MAS01500
30          CONTINUE                                                            MAS01510
        END IF                                                                  MAS01520
C                                                                               MAS01530
CC      *** PLOT THE TIME SERIES IF DESIRED ***                                 MAS01540
C                                                                               MAS01550
        IF(REP.EQ.1)THEN                                                        MAS01560
            MINY=-MAXY                                                          MAS01570
C                                                                               MAS01580
            CALL PLOT(MAXY,MINY,STEP,IN)                                        MAS01590
        END IF                                                                  MAS01600
```

```
C                                                                       MAS01610
C       *** GENERATE THE FFT ***                                        MAS01620
C                                                                       MAS01630
        CALL FFTRC(ARRAY(1,1),IN,X,IWK)                                 MAS01640
C                                                                       MAS01650
C       *** CALCULATE THE AMPLITUDE SPECTRUM OF THE FUNCTION ***        MAS01660
C       *** FIND THE NUMBER AND VALUE OF THE PRINCIPLE HARMONIC ***     MAS01670
C                                                                       MAS01680
        N=0.D0                                                          MAS01690
        R=1                                                             MAS01700
        MAXY=0.D0                                                       MAS01710
        DO 40 I=1,IND2P1                                                MAS01720
            ARRAY(R,1)=CDABS(X(R))                                      MAS01730
            ARRAY(R,2)=N/STEP                                           MAS01740
            IF(ARRAY(R,1).GT.MAXY)THEN                                  MAS01750
                MAXY=ARRAY(R,1)                                         MAS01760
                RMAX=R-1                                                MAS01770
            END IF                                                      MAS01780
            R=R+1                                                       MAS01790
            N=N+1.D0                                                    MAS01800
40      CONTINUE                                                        MAS01810
C                                                                       MAS01820
C       *** DISPLAY INFO IF THE FIRST TIME THROUGH THE SUBPROGRAM ***   MAS01830
C                                                                       MAS01840
        IF(REP.EQ.1)THEN                                                MAS01850
C                                                                       MAS01860
49      WRITE(10,50)RMAX                                                MAS01870
50      FORMAT(' THE FFT HAS BEEN GENERATED!  THE PRINCIPLE HARMONIC',/, MAS01880
     *' IS THE',I5,' HARMONIC.  THE NEXT PLOT TO BE PRODUCED WILL',/,    MAS01890
     *' BE THE AMPLITUDE SPECTRUM.  ENTER A 1 IF YOU ARE READY TO',/,    MAS01900
     *' CONTINUE WITH THE PROGRAM.',//,' :')                             MAS01910
C                                                                       MAS01920
        READ(5,*)L                                                      MAS01930
C                                                                       MAS01940
        IF(L.NE.1)THEN                                                  MAS01950
C           CALL FRTCMS('CLRSCRN ')                                     MAS01960
            WRITE(10,60)                                                MAS01970
60          FORMAT(' ERROR',/)                                          MAS01980
            GO TO 49                                                    MAS01990
        END IF                                                          MAS02000
C                                                                       MAS02010
        END IF                                                          MAS02020
C                                                                       MAS02030
C       *** ALLOW FOR THE PLOT OF THE AMPLITUDE SPECTRUM OF THE FFT ***  MAS02040
C                                                                       MAS02050
        IF(REP.EQ.1)THEN                                                MAS02060
            MINY=0.D0                                                   MAS02070
            INT=1.D0/STEP                                               MAS02080
C                                                                       MAS02090
            CALL PLCT(MAXY,MINY,INT,IND2P1)                             MAS02100
```

```
      END IF                                                                    MAS02110
C                                                                               MAS02120
C     *** ADD THE VALUES OF THE FFT TO THE ACCUMULATED STATISTICS ***           MAS02130
C                                                                               MAS02140
      FLAG=0                                                                    MAS02150
      CALL STATS(IN,REP,FLAG)                                                   MAS02160
C                                                                               MAS02170
      RETURN                                                                    MAS02180
      END                                                                       MAS02190
```

```
      SUBROUTINE QASK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,ANS2,                QAS00110
     *DSEED,IN,REP)                                                       QAS00120
C                                                                         QAS00130
C     *** PURPOSE ***                                                     QAS00140
C                                                                         QAS00150
C     THIS SUBROUTINE MODULATES THE CARRIER USING QUADRATURE AMPLITUDE    QAS00160
C     SHIFT KEYING AS THE MODULATION TECHNIQUE.                           QAS00170
C                                                                         QAS00180
C     *** PARAMETER DEFINITIONS ***                                       QAS00190
C                                                                         QAS00200
C     TYPE2=       INDICATES LOGIC TYPE TO BE EMPLOYED                    QAS00210
C     BAUD=        SYMBOL RATE OR BAUD RATE                               QAS00220
C     BITS=        NUMBER OF BITS IN EACH BINARY CODE WORD                QAS00230
C     FREQ=        CARRIER FREQUENCY                                      QAS00240
C     IPHAS=       INITIAL PHASE ANGLE IN DEGREES                         QAS00250
C     AMP=         AMPLITUDE                                              QAS00260
C     ANS2=        INTEGER VARIABLE TO BE PASSED TO STREAM                QAS00270
C     DSEED=       DOUBLE PRECISION SEED FOR RANDOM NUMBER GENERATOR      QAS00280
C     IN=          NUMBER OF POSITIONS IN ARRAY TO BE UTILIZED            QAS00290
C                  THE CALL THE THIS SUBROUTINE                           QAS00300
C     FLAG=        AN INTEGER WHICH CONTROL ENTRANCE AND EXIT FROM        QAS00310
C                                                                         QAS00320
C     *** VARIABLE DEFINITIONS ***                                        QAS00330
C                                                                         QAS00340
C     ARRAY=       A DOUBLE PRECISION ARRAY FOR PASSING THE BINARY        QAS00350
C                  DIGITS AND STORING THE VALUE OF THE TIME FUNCTION      QAS00360
C                  AND THE FFT                                            QAS00370
C     R=           AN INTEGER USED TO REPRESENT THE ROW OF ARRAY          QAS00380
C     TIME=        TIME VARIABLE                                          QAS00390
C     MT=          VALUE OF THE BINARY DIGIT                              QAS00400
C     STEP=        INTERVAL AT WHICH THE SIGNAL IS REPRODUCED             QAS00410
C     PI=          NUMERICAL CONSTANT                                     QAS00420
C     OMEGA=       CARRIER ANGULAR FREQUENCY                              QAS00430
C     DELTA=       INITIAL PHASE OFFSET IN RADIANS                        QAS00440
C     NBAUD=       A COUNT OF THE NUMBER OF BAUDS MODULATED               QAS00450
C     BAUDD=       BAUD DURATION                                          QAS00460
C     X=           COMPLEX ARRAY TO RECEIVE THE FFT OF THE TIME SERIES    QAS00470
C     IWK=         INTEGER WORKING ARRAY USED BY FUNCTION FFTRC           QAS00480
C     MAXY=        MAXIMUM VALUE TO BE PLOTTED ON THE ABCISSAE            QAS00490
C     MINY=        MINIMUM VALUE TO BE PLOTTED ON THE ABCISSAE            QAS00500
C     IN1=         INTERVAL BETWEEN POINTS ON THE ORDINATE                QAS00510
C     RMAX=        VALUE OF THE PRINCIPLE HARMONIC                        QAS00520
C     K=           INTEGER VALUE OF THE NUMBER OF BITS IN A BINARY WORD   QAS00530
C     KM1=         INTEGER VALUE OF K MINUS 1                             QAS00540
C     N=           NUMBER OF POSSIBLE BINARY CODE WORDS = 2**K            QAS00550
C     ND2=         N DIVIDED BY 2                                         QAS00560
C     SUM1=        VARIABLE USED TO HOLD THE DECIMAL EQUIVALENT TO A      QAS00570
C                  BINARY CODE WORD OF LENGTH K                           QAS00580
C     SUM2=        VARIABLE USED TO HOLD THE DECIMAL CONVERSION OF SUM1   QAS00590
C     J=           INTEGER USED TO INDEX A LOOP                           QAS00600
```

```
C      AMPA=        PORTION OF AMP USED TO MODULATE THE IN-PHASE          QAS00610
C                   COMPONENT OF THE CARRIER DURING ANY RESPECTIVE BAUD   QAS00620
C                   DURATION                                              QAS00630
C      AMPB=        PORTION OF AMP USED TO MODULATE THE QUADRATURE        QAS00640
C                   COMPONENT OF THE CARRIER DURING ANY RESPECTIVE BAUD   QAS00650
C                   DURATION                                              QAS00660
C      SUM2A=       THE VALUE OF THE IN-PHASE CODE WORD                   QAS00670
C      SUM2B=       THE VALUE OF THE QUADRATURE CODE WORD                 QAS00680
C      REP=         AN INTEGER EQUAL TO THE NUMBER OF THE REPETITION OF   QAS00690
C                   SUBPROGRAM STATS                                      QAS00700
C      IND2P1=      LENGTH OF ARRAY DIVIDED BY 2 + 1                      QAS00710
C                                                                         QAS00720
C      *** VARIABLE DECLARATIONS ***                                      QAS00730
C                                                                         QAS00740
       INTEGER R,ANS2,TYPE2,IN,RMAX,K,KM1,J,M,REP,FLAG,IND2P1             QAS00750
C                                                                         QAS00760
       DOUBLE PRECISION BAUD,BITS,FREQ,IPHAS,AMP,TIME,MT,                 QAS00770
      *STEP,PI,OMEGA,DELTA,NBAUD,BAUDD,                                   QAS00780
      *DSEED,MAXY,MINY,INT,N,ND2,SUM1,SUM2,AMPA,AMPB,SUM2A,SUM2B          QAS00790
C                                                                         QAS00800
       DOUBLE PRECISION ARRAY(1024,2),SUMX(513),SUMXSQ(513),SUMX3(513),   QAS00810
      *SUMX4(513)                                                         QAS00820
       COMMON ARRAY,SUMX,SUMXSQ,SUMX3,SUMX4                              QAS00830
C                                                                         QAS00840
       COMPLEX*16 X(513)                                                 QAS00850
C                                                                         QAS00860
       INTEGER IWK(10)                                                   QAS00870
C                                                                         QAS00880
C      *** VARIABLE INITIALIZATION ***                                    QAS00890
C                                                                         QAS00900
       R=1                                                               QAS00910
       TIME=0.D0                                                         QAS00920
       STEP=1.D0/(2.D0*(FREQ+BAUD))                                      QAS00930
       PI=3.14159265358979 3D0                                           QAS00940
       OMEGA=2.D0*PI*FREQ                                                QAS00950
       DELTA=IPHAS*PI/180.D0                                             QAS00960
       NBAUD=1.D0                                                        QAS00970
       BAUDD=1.D0/BAUD                                                   QAS00980
       K=BITS                                                            QAS00990
       N=2.D0**K                                                         QAS01000
       MAXY=0.D0                                                         QAS01010
       IND2P1=IN/2+1                                                     QAS01020
C                                                                        QAS01030
9      IF(R.LE.IN) THEN                                                  QAS01040
C                                                                        QAS01050
C      *** CONVERT A STREAM OF K BITS TO A SINGLE REPRESENTATIVE BINARY   QAS01060
C      VARIABLE IN DECIMAL FORM-DO THIS TWICE, ONCE FOR THE IN-PHASE     QAS01070
C      CODE WORD AND ONCE FOR THE QUADRATURE CODE WORD ***               QAS01080
C                                                                        QAS01090
           DO 16 M=1,2                                                   QAS01100
```

```
C                                                                       QAS01110
                SUM1=0.D0                                               QAS01120
                KM1=K-1                                                 QAS01130
                DO 10 I=1,K                                             QAS01140
                    CALL STREAM(DSEED,ANS2,TYPE2,MT)                    QAS01150
                    IF(TYPE2.EQ.1.AND.MT.EQ.-1.D0)THEN                  QAS01160
                        MT=0.D0                                         QAS01170
                    ELSE IF(TYPE2.EQ.3.AND.MT.EQ.-1.D0)THEN             QAS01180
                        MT=1.D0                                         QAS01190
                    END IF                                              QAS01200
                    SUM1=SUM1+(MT*(10.D0**KM1))                         QAS01210
                    KM1=KM1-1                                           QAS01220
10              CONTINUE                                                QAS01230
C                                                                       QAS01240
CCC     *** CONVERT THE REPRESENTATIVE BINARY VARIABLE TO ITS DECIMAL   QAS01250
CCC     EQUIVALENT ***                                                  QAS01260
C                                                                       QAS01270
                SUM2=0.D0                                               QAS01280
                KM1=K-1                                                 QAS01290
                ND2=N/2.D0                                              QAS01300
                DO 20 I=1,K                                             QAS01310
                    IF(SUM1/10.D0**KM1.GE.1.D0)THEN                     QAS01320
                        SUM2=SUM2+ND2                                   QAS01330
                        SUM1=SUM1-10.D0**KM1                            QAS01340
                        ND2=ND2/2.D0                                    QAS01350
                        KM1=KM1-1                                       QAS01360
                    ELSE                                                QAS01370
                        ND2=ND2/2.D0                                    QAS01380
                        KM1=KM1-1                                       QAS01390
                    END IF                                              QAS01400
20              CONTINUE                                                QAS01410
C                                                                       QAS01420
                IF(M.EQ.1)THEN                                          QAS01430
                    SUM2A=SUM2+1.D0                                     QAS01440
                ELSE IF(M.EQ.2)THEN                                     QAS01450
                    SUM2B=SUM2+1.D0                                     QAS01460
                END IF                                                  QAS01470
C                                                                       QAS01480
16          CONTINUE                                                    QAS01490
C                                                                       QAS01500
CCC     *** DO THE MODULATION AND ASSIGN TO ARRAY ***                   QAS01510
C                                                                       QAS01520
            AMPA=AMP/SUM2A                                              QAS01530
            AMPB=AMP/SUM2B                                              QAS01540
            J=R                                                         QAS01550
            DO 30 I=J,IN                                                QAS01560
                ARRAY(R,1)=AMPA*DCOS((OMEGA*TIME)+DELTA)+               QAS01570
     *                     AMPB*DSIN((OMEGA*TIME)+DELTA)                QAS01580
                ARRAY(R,2)=TIME                                         QAS01590
                IF(ARRAY(R,1).GT.MAXY)THEN                              QAS01600
```

```
                    MAXY=ARRAY(R,1)                                          QAS01610
               END IF                                                        QAS01620
               R=R+1                                                         QAS01630
               TIME=TIME+STEP                                                QAS01640
               IF(TIME.GT.NBAUD*BAUDD) THEN                                  QAS01650
                    NBAUD=NBAUD+1.D0                                         QAS01660
                    GO TO 9                                                  QAS01670
               END IF                                                        QAS01680
30        CONTINUE                                                         QAS01690
     END IF                                                               QAS01700
C                                                                         QAS01710
CC   *** PLOT THE TIME SERIES IF DESIRED ***                              QAS01720
C                                                                         QAS01730
     IF(REP.EQ.1) THEN                                                    QAS01740
          MINY=-MAXY                                                      QAS01750
C                                                                         QAS01760
          CALL PLOT(MAXY,MINY,STEP,IN)                                    QAS01770
     END IF                                                               QAS01780
C                                                                         QAS01790
CC   *** GENERATE THE FFT ***                                             QAS01800
C                                                                         QAS01810
     CALL FFTRC(ARRAY(1,1),IN,X,IWK)                                      QAS01820
C                                                                         QAS01830
CCC  *** CALCULATE THE AMPLITUDE SPECTRUM OF THE FUNCTION ***             QAS01840
CC   *** FIND THE NUMBER AND VALUE OF THE PRINCIPLE HARMONIC ***          QAS01850
C                                                                         QAS01860
     N=0.D0                                                               QAS01870
     R=1                                                                  QAS01880
     MAXY=0.D0                                                            QAS01890
     DO 40 I=1,IND2P1                                                     QAS01900
          ARRAY(R,1)=CDABS(X(R))                                          QAS01910
          ARRAY(R,2)=N/STEP                                               QAS01920
          IF(ARRAY(R,1).GT.MAXY) THEN                                     QAS01930
               MAXY=ARRAY(R,1)                                            QAS01940
               RMAX=R-1                                                   QAS01950
          END IF                                                          QAS01960
          R=R+1                                                           QAS01970
          N=N+1.D0                                                        QAS01980
40   CONTINUE                                                             QAS01990
C                                                                         QAS02000
CC   *** DISPLAY INFO IF THE FIRST TIME THROUGH THE SUBPROGRAM ***        QAS02010
C                                                                         QAS02020
     IF(REP.EQ.1) THEN                                                    QAS02030
C                                                                         QAS02040
49   WRITE(10,50) RMAX                                                    QAS02050
50   FORMAT(' THE FFT HAS BEEN GENERATED!  THE PRINCIPLE HARMONIC',/,     QAS02060
    *' IS THE',I5,' HARMONIC.  THE NEXT PLOT TO BE PRODUCED WILL',/,      QAS02070
    *' BE THE AMPLITUDE SPECTRUM.  ENTER A 1 IF YOU ARE READY TO',/,      QAS02080
    *' CONTINUE WITH THE PROGRAM.',//,' :')                               QAS02090
C                                                                         QAS02100
```

```
      READ(5,*)L                                                        QAS02110
C                                                                       QAS02120
      IF(L.NE.1)THEN                                                    QAS02130
C         CALL FRTCMS('CLRSCRN ')                                       QAS02140
          WRITE(10,60)                                                  QAS02150
60        FORMAT(' ERROR',/)                                            QAS02160
          GO TO 49                                                      QAS02170
      END IF                                                            QAS02180
C                                                                       QAS02190
      END IF                                                            QAS02200
C                                                                       QAS02210
CC    *** ALLOW FOR THE PLOT OF THE AMPLITUDE SPECTRUM OF THE FFT ***   QAS02220
C                                                                       QAS02230
      IF(REP.EQ.1)THEN                                                  QAS02240
          MINY=0.D0                                                     QAS02250
          INT=1.D0/STEP                                                 QAS02260
C                                                                       QAS02270
          CALL PLOT(MAXY,MINY,INT,IND2P1)                               QAS02280
      END IF                                                            QAS02290
C                                                                       QAS02300
CC    *** ADD THE VALUES OF THE FFT TO THE ACCUMULATED STATISTICS ***   QAS02310
C                                                                       QAS02320
      FLAG=0                                                            QAS02330
      CALL STATS(IN,REP,FLAG)                                           QAS02340
C                                                                       QAS02350
      RETURN                                                            QAS02360
      END                                                               QAS02370
```

```
      SUBROUTINE MSK(TYPE2,BAUD,FREQ,IPHAS,AMP,ANS2,                    MSK00110
     *DSEED,IN,REP)                                                     MSK00120
C                                                                       MSK00130
C     *** PURPOSE ***                                                   MSK00140
C                                                                       MSK00150
C     THIS SUBROUTINE MODULATES THE CARRIER USING MINIMUM SHIFT KEYING  MSK00160
C     OR FAST FREQUENCY SHIFT KEYING AS THE MODULATION TECHNIQUE.       MSK00170
C                                                                       MSK00180
C     *** PARAMETER DEFINITIONS ***                                     MSK00190
C                                                                       MSK00200
C     TYPE2=       INDICATES LOGIC TYPE TO BE EMPLOYED                  MSK00210
C     BAUD=        SYMBOL RATE OR BAUD RATE                             MSK00220
C     FREQ=        CARRIER FREQUENCY                                    MSK00230
C     IPHAS=       INITIAL PHASE ANGLE IN DEGREES                       MSK00240
C     AMP=         AMPLITUDE                                            MSK00250
C     ANS2=        INTEGER VARIABLE TO BE PASSED TO STREAM              MSK00260
C     DSEED=       DOUBLE PRECISION SEED FOR RANDOM NUMBER GENERATOR    MSK00270
C     IN=          NUMBER OF POSITIONS IN ARRAY TO BE UTILIZED          MSK00280
C     REP=         AN INTEGER EQUAL TO THE NUMBER OF THE REPETITION OF  MSK00290
C                  THE CALL THE THIS SUBROUTINE                         MSK00300
C                                                                       MSK00310
C     *** VARIABLE DEFINITIONS ***                                      MSK00320
C                                                                       MSK00330
C     ARRAY=       A DOUBLE PRECISION ARRAY FOR PASSING THE BINARY      MSK00340
C                  DIGITS AND STORING THE VALUE OF THE TIME FUNCTION    MSK00350
C                  AND THE FFT                                          MSK00360
C     R=           AN INTEGER USED TO REPRESENT THE ROW OF ARRAY        MSK00370
C     TIME=        TIME VARIABLE                                        MSK00380
C     MT=          VALUE OF THE BINARY DIGIT                            MSK00390
C     STEP=        INTERVAL AT WHICH THE SIGNAL IS REPRODUCED           MSK00400
C     PI=          NUMERICAL CONSTANT                                   MSK00410
C     OMEGA=       CARRIER ANGULAR FREQUENCY                            MSK00420
C     DELTA=       INITIAL PHASE OFFSET IN RADIANS                      MSK00430
C     NBAUD=       A COUNT OF THE NUMBER OF BAUDS MODULATED             MSK00440
C     BAUDD=       BAUD DURATION                                        MSK00450
C     X=           COMPLEX ARRAY TO RECEIVE THE FFT OF THE TIME SERIES  MSK00460
C     IWK=         INTEGER WORKING ARRAY USED BY FUNCTION FFTRC         MSK00470
C     MAXY=        MAXIMUM VALUE TO BE PLOTTED ON THE ABCISSAE          MSK00480
C     MINY=        MINIMUM VALUE IO BE PLOTTED ON THE ABCISSAE          MSK00490
C     INT=         INTERVAL BETWEEN POINTS ON THE ORDINATE              MSK00500
C     RMAX=        VALUE OF THE PRINCIPLE HARMONIC                      MSK00510
C     FLAG=        AN INTEGER WHICH CONTROL ENTRANCE AND EXIT FROM      MSK00520
C                  SUBPROGRAM STATS                                     MSK00530
C     IND2P1=      LENGTH OF ARRAY DIVIDED BY 2 + 1                     MSK00540
C                                                                       MSK00550
C     *** VARIABLE DECLARATIONS ***                                     MSK00560
C                                                                       MSK00570
      INTEGER R,ANS2,TYPE2,IN,RMAX,REP,FLAG,IND2P1                      MSK00580
C                                                                       MSK00590
      DOUBLE PRECISION BAUD,FREQ,IPHAS,AMP,TIME,MT,                     MSK00600
```

```
     *STEP,PI,OMEGA,DELTA,NBAUD,BAUDD,                                          MSK00610
     *DSEED,MAXY,MINY,INT                                                       MSK00620
C                                                                               MSK00630
      DOUBLE PRECISION ARRAY(1024,2),SUMX(513),SUMXSQ(513),SUMX3(513),          MSK00640
     *SUMX4(513)                                                                MSK00650
      COMMON ARRAY,SUMX,SUMXSQ,SUMX3,SUMX4                                      MSK00660
C                                                                               MSK00670
      COMPLEX*16 X(513)                                                         MSK00680
C                                                                               MSK00690
      INTEGER IWK(10)                                                           MSK00700
C                                                                               MSK00710
CC    *** VARIABLE INITIALIZATION ***                                           MSK00720
C                                                                               MSK00730
      R=1                                                                       MSK00740
      TIME=0.D0                                                                 MSK00750
      STEP=1.D0/(2.D0*(FREQ+(1.25D0*BAUD)))                                     MSK00760
      PI=3.14159265358979 3D0                                                   MSK00770
      OMEGA=2.D0*PI*FREQ                                                        MSK00780
      DELTA=IPHAS*PI/180.D0                                                     MSK00790
      NBAUD=1.D0                                                                MSK00800
      BAUDD=1.D0/BAUD                                                           MSK00810
      MAXY=0.D0                                                                 MSK00820
      IND2P1=IN/2+1                                                             MSK00830
C                                                                               MSK00840
      CALL STREAM(DSEED,ANS2,TYPE2,MT)                                          MSK00850
C                                                                               MSK00860
      DO 10 I=1,IN                                                              MSK00870
        ARRAY(R,1)=AMP*DCOS(((OMEGA+((MT*PI)/(2.D0*BAUDD)))*TIME)               MSK00880
     *    +DELTA)                                                              MSK00890
        ARRAY(R,2)=TIME                                                         MSK00900
        IF(DABS(ARRAY(R,1)).GT.MAXY) THEN                                       MSK00910
          MAXY=DABS(ARRAY(R,1))                                                 MSK00920
        END IF                                                                  MSK00930
        R=R+1                                                                   MSK00940
        TIME=TIME+STEP                                                          MSK00950
        IF(TIME.GT.NBAUD*BAUDD) THEN                                            MSK00960
          CALL STREAM(DSEED,ANS2,TYPE2,MT)                                      MSK00970
          NBAUD=NBAUD+1.D0                                                      MSK00980
        END IF                                                                  MSK00990
10    CONTINUE                                                                  MSK01000
C                                                                               MSK01010
CC    *** PLOT THE TIME SERIES IF DESIRED ***                                   MSK01020
C                                                                               MSK01030
      IF(REP.EQ.1) THEN                                                         MSK01040
        MINY=-MAXY                                                              MSK01050
C                                                                               MSK01060
        CALL PLOT(MAXY,MINY,STEP,IN)                                            MSK01070
      END IF                                                                    MSK01080
C                                                                               MSK01090
CC    *** GENERATE THE FFT ***                                                  MSK01100
```

```
C                                                                       MSK01110
      CALL FFTRC(ARRAY(1,1),IN,X,IWK)                                   MSK01120
C                                                                       MSK01130
CC    *** CALCULATE THE AMPLITUDE SPECTRUM OF THE FUNCTION ***          MSK01140
CC    *** FIND THE NUMBER AND VALUE OF THE PRINCIPLE HARMONIC ***       MSK01150
C                                                                       MSK01160
      N=0.D0                                                            MSK01170
      R=1                                                               MSK01180
      MAXY=0.D0                                                         MSK01190
      DO 20 I=1,IND2P1                                                  MSK01200
          ARRAY(R,1)=CDABS(X(R))                                        MSK01210
          ARRAY(R,2)=N/STEP                                             MSK01220
          IF(ARRAY(R,1).GT.MAXY) THEN                                   MSK01230
              MAXY=ARRAY(R,1)                                           MSK01240
              RMAX=R-1                                                  MSK01250
          END IF                                                        MSK01260
          R=R+1                                                         MSK01270
          N=N+1.D0                                                      MSK01280
20    CONTINUE                                                          MSK01290
C                                                                       MSK01300
CC    *** DISPLAY INFO IF THE FIRST TIME THROUGH SUBPROGRAM ***         MSK01310
C                                                                       MSK01320
      IF(REP.EQ.1)THEN                                                  MSK01330
C                                                                       MSK01340
29    WRITE(10,30)RMAX                                                  MSK01350
30    FORMAT(' THE FFT HAS BEEN GENERATED!  THE PRINCIPLE HARMONIC',/,  MSK01360
     *' IS THE',I5,' HARMONIC.  THE NEXT PLOT TO BE PRODUCED WILL',/,   MSK01370
     *' BE THE AMPLITUDE SPECTRUM.  ENTER A 1 IF YOU ARE READY TO',/,   MSK01380
     *' CONTINUE WITH THE PROGRAM.',//,' :')                            MSK01390
C                                                                       MSK01400
      READ(5,*)L                                                        MSK01410
C                                                                       MSK01420
      IF(L.NE.1)THEN                                                    MSK01430
C         CALL FRTCMS('CLRSCRN ')                                       MSK01440
          WRITE(10,40)                                                  MSK01450
40        FORMAT(' ERROR',/)                                            MSK01460
          GO TO 29                                                      MSK01470
      END IF                                                            MSK01480
C                                                                       MSK01490
      END IF                                                            MSK01500
C                                                                       MSK01510
CC    *** ALLOW FOR THE PLOT OF THE AMPLITUDE SPECTRUM OF THE FFT ***   MSK01520
C                                                                       MSK01530
      IF(REP.EQ.1)THEN                                                  MSK01540
          MINY=0.D0                                                     MSK01550
          INT=1.D0/STEP                                                 MSK01560
C                                                                       MSK01570
          CALL PLOT(MAXY,MINY,INT,IND2P1)                               MSK01580
      END IF                                                            MSK01590
C                                                                       MSK01600
```

```
C       *** ADD THE VALUES OF THE FFT TO THE ACCUMULATED STATISTICS ***    MSK01610
C                                                                          MSK01620
        FLAG=0                                                             MSK01630
        CALL STATS(IN,REP,FLAG)                                            MSK01640
C                                                                          MSK01650
        RETURN                                                             MSK01660
        END                                                                MSK01670
```

```
      SUBROUTINE MFSK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,ANS2,                MFS00110
     *DSEED,IN,REP)                                                       MFS00120
C                                                                         MFS00130
C     *** PURPOSE ***                                                     MFS00140
C                                                                         MFS00150
C     THIS SUBROUTINE MODULATES THE CARRIER USING M-ARY FREQUENCY SHIFT    MFS00160
C     KEYING AS THE MODULATION TECHNIQUE.                                 MFS00170
C                                                                         MFS00180
C     *** PARAMETER DEFINITIONS ***                                       MFS00190
C                                                                         MFS00200
C     TYPE2=       INDICATES LOGIC TYPE TO BE EMPLOYED                    MFS00210
C     BAUD=        SYMBOL RATE OR BAUD RATE                               MFS00220
C     BITS=        NUMBER OF BITS IN EACH BINARY CODE WORD                MFS00230
C     FREQ=        CARRIER FREQUENCY                                      MFS00240
C     IPHAS=       INITIAL PHASE ANGLE IN DEGREES                         MFS00250
C     AMP=         AMPLITUDE                                              MFS00260
C     ANS2=        INTEGER VARIABLE TO BE PASSED TO STREAM                MFS00270
C     DSEED=       DOUBLE PRECISION SEED FOR RANDOM NUMBER GENERATOR      MFS00280
C     IN=          NUMBER OF POSITIONS IN ARRAY TO BE UTILIZED            MFS00290
C     REP=         AN INTEGER EQUAL TO THE NUMBER OF THE REPETITION OF    MFS00300
C                  THE CALL THE THIS SUBROUTINE                           MFS00310
C                                                                         MFS00320
C     *** VARIABLE DEFINITIONS ***                                        MFS00330
C                                                                         MFS00340
C     ARRAY=       A DOUBLE PRECISION ARRAY FOR PASSING THE BINARY        MFS00350
C                  DIGITS AND STORING THE VALUE OF THE TIME FUNCTION      MFS00360
C                  AND THE FFT                                            MFS00370
C     R=           AN INTEGER USED TO REPRESENT THE ROW OF ARRAY          MFS00380
C     TIME=        TIME VARIABLE                                          MFS00390
C     MT=          VALUE OF THE BINARY DIGIT                              MFS00400
C     STEP=        INTERVAL AT WHICH THE SIGNAL IS REPRODUCED             MFS00410
C     PI=          NUMERICAL CONSTANT                                     MFS00420
C     OMEGA=       CARRIER ANGULAR FREQUENCY                              MFS00430
C     DELTA=       INITIAL PHASE OFFSET IN RADIANS                        MFS00440
C     NBAUD=       A COUNT OF THE NUMBER OF BAUDS MODULATED               MFS00450
C     BAUDD=       BAUD DURATION                                          MFS00460
C     X=           COMPLEX ARRAY TO RECEIVE THE FFT OF THE TIME SERIES    MFS00470
C     IWK=         INTEGER WORKING ARRAY USED BY FUNCTION FFTRC           MFS00480
C     MAXY=        MAXIMUM VALUE TO BE PLOTTED ON THE ABCISSAE            MFS00490
C     MINY=        MINIMUM VALUE TO BE PLOTTED ON THE ABCISSAE            MFS00500
C     INT=         INTERVAL BETWEEN POINTS ON THE ORDINATE                MFS00510
C     RMAX=        VALUE OF THE PRINCIPLE HARMONIC                        MFS00520
C     K=           INTEGER VALUE OF THE NUMBER OF BITS IN A BINARY WORD   MFS00530
C     KM1=         INTEGER VALUE OF K MINUS 1                             MFS00540
C     N=           NUMBER OF POSSIBLE BINARY CODE WORDS = 2**K            MFS00550
C     ND2=         N DIVIDED BY 2                                         MFS00560
C     SUM1=        VARIABLE USED TO HOLD THE DECIMAL EQUIVALENT TO A      MFS00570
C                  BINARY CODE WORD OF LENGTH K                           MFS00580
C     SUM2=        VARIABLE USED TO HOLD THE DECIMAL CONVERSION OF SUM1   MFS00590
C     J=           INTEGER USED TO INDEX A LOOP                           MFS00600
```

```
C     FLAG=        AN INTEGER WHICH CONTROL ENTRANCE AND EXIT FROM        MFS00610
C                  SUBPROGRAM STATS                                       MFS00620
C     DELF=        DELTA FREQUENCY BETWEEN SUCCESSIVE CODE WORDS          MFS00630
C     FRNG=        FREQUENCY RANGE AROUND THE CENTRAL CARRIER FREQUENCY   MFS00640
C     FRNGD2=      FREQUENCY RANGE DIVIDED BY 2.                          MFS00650
C     MFREQ=       FREQUENCY OF THE CODE WORD TO BE MODULATED ONTO THE    MFS00660
C                  CARRIER                                                MFS00670
C     IND2P1=      LENGTH OF ARRAY DIVIDED BY 2 + 1                       MFS00680
C                                                                         MFS00690
C     *** VARIABLE DECLARATIONS ***                                       MFS00700
C                                                                         MFS00710
      INTEGER R,ANS2,TYPE2,IN,RMAX,K,KM1,J,REP,FLAG,IND2P1                MFS00720
C                                                                         MFS00730
      DOUBLE PRECISION BAUD,BITS,FREQ,IPHAS,AMP,TIME,MT,                  MFS00740
     *STEP,PI,OMEGA,DELTA,NBAUD,BAUDD,                                    MFS00750
     *DSEED,MAXY,MINY,INT,N,ND2,SUM1,SUM2,DELF,FRNG,FRNGD2,MFREQ          MFS00760
C                                                                         MFS00770
      DOUBLE PRECISION ARRAY(1024,2),SUMX(513),SUMXSQ(513),SUMX3(513),    MFS00780
     *SUMX4(513)                                                          MFS00790
      COMMON ARRAY,SUMX,SUMXSQ,SUMX3,SUMX4                               MFS00800
C                                                                         MFS00810
      COMPLEX*16 X(513)                                                   MFS00820
C                                                                         MFS00830
      INTEGER IWK(10)                                                     MFS00840
C                                                                         MFS00850
C     *** VARIABLE INITIALIZATION ***                                     MFS00860
C                                                                         MFS00870
      R=1                                                                 MFS00880
      TIME=0.D0                                                           MFS00890
      BAUDD=1.D0/BAUD                                                     MFS00900
      DELF=1.D0/BAUDD                                                     MFS00910
      K=BITS                                                              MFS00920
      N=2.D0**K                                                           MFS00930
      FRNG=(N-1.D0)*DELF                                                  MFS00940
      FRNGD2=FRNG/2.D0                                                    MFS00950
      STEP=1.D0/(2.D0*(FREQ+FRNGD2+BAUD))                                 MFS00960
      PI=3.14159265358979 3D0                                             MFS00970
      OMEGA=2.D0*PI*FREQ                                                  MFS00980
      DELTA=IPHAS*PI/180.D0                                               MFS00990
      NBAUD=1.D0                                                          MFS01000
      MAXY=0.D0                                                           MFS01010
      IND2P1=IN/2+1                                                       MFS01020
C                                                                         MFS01030
9     IF(R.LE.IN) THEN                                                    MFS01040
C                                                                         MFS01050
C     *** CONVERT A STREAM OF K BITS TO A SINGLE REPRESENTATIVE BINARY    MFS01060
C     VARIABLE IN DECIMAL FORM ***                                        MFS01070
C                                                                         MFS01080
          SUM1=0.D0                                                       MFS01090
          KM1=K-1                                                         MFS01100
```

```
      DO 10 I=1,K                                                       MFS01110
           CALL STREAM(DSEED,ANS2,TYPE2,MT)                             MFS01120
           IF(TYPE2.EQ.1.AND.MT.EQ.-1.D0) THEN                          MFS01130
                MT=0.D0                                                 MFS01140
           ELSE IF(TYPE2.EQ.3.AND.MT.EQ.-1.D0) THEN                     MFS01150
                MT=1.D0                                                 MFS01160
           END IF                                                       MFS01170
           SUM1=SUM1+(MT*(10.D0**KM1))                                  MFS01180
           KM1=KM1-1                                                    MFS01190
10    CONTINUE                                                          MFS01200
C                                                                       MFS01210
CCC   *** CONVERT THE REPRESENTATIVE BINARY VARIABLE TO ITS DECIMAL     MFS01220
CCC   EQUIVALENT ***                                                    MFS01230
C                                                                       MFS01240
      SUM2=0.D0                                                         MFS01250
      KM1=K-1                                                           MFS01260
      ND2=N/2.D0                                                        MFS01270
      DO 20 I=1,K                                                       MFS01280
           IF(SUM1/10.D0**KM1.GE.1.D0) THEN                             MFS01290
                SUM2=SUM2+ND2                                           MFS01300
                SUM1=SUM1-10.D0**KM1                                    MFS01310
                ND2=ND2/2.D0                                            MFS01320
                KM1=KM1-1                                               MFS01330
           ELSE                                                         MFS01340
                ND2=ND2/2.D0                                            MFS01350
                KM1=KM1-1                                               MFS01360
           END IF                                                       MFS01370
20    CONTINUE                                                          MFS01380
CCC                                                                     MFS01390
CCC   *** DO THE MODULATION AND ASSIGN TO ARRAY ***                     MFS01400
C                                                                       MFS01410
      J=R                                                               MFS01420
      MFREQ=-FRNGD2+(SUM2*DELF)                                         MFS01430
      DO 30 I=J,IN                                                      MFS01440
           ARRAY(R,1)=AMP*DCOS(((OMEGA+(2.D0*PI*MFREQ))*TIME)+          MFS01450
     *                     DELTA)                                       MFS01460
           ARRAY(R,2)=TIME                                              MFS01470
           IF(ARRAY(R,1).GT.MAXY) THEN                                  MFS01480
                MAXY=ARRAY(R,1)                                         MFS01490
           END IF                                                       MFS01500
           R=R+1                                                        MFS01510
           TIME=TIME+STEP                                               MFS01520
           IF(TIME.GT.NBAUD*BAUDD) THEN                                 MFS01530
                NBAUD=NBAUD+1.D0                                        MFS01540
                GO TO 9                                                 MFS01550
           END IF                                                       MFS01560
30    CONTINUE                                                          MFS01570
      END IF                                                            MFS01580
C                                                                       MFS01590
C     *** PLOT THE TIME SERIES IF DESIRED ***                           MFS01600
```

```
C                                                                       MFS01610
      IF(REP.EQ.1)THEN                                                  MFS01620
          MINY=-MAXY                                                    MFS01630
C                                                                       MFS01640
          CALL PLOT(MAXY,MINY,STEP,IN)                                  MFS01650
      END IF                                                            MFS01660
C                                                                       MFS01670
CC    *** GENERATE THE FFT ***                                          MFS01680
C                                                                       MFS01690
      CALL FFTRC(ARRAY(1,1),IN,X,IWK)                                   MFS01700
C                                                                       MFS01710
CCC   *** CALCULATE THE AMPLITUDE SPECTRUM OF THE FUNCTION ***          MFS01720
CCC   *** FIND THE NUMBER AND VALUE OF THE PRINCIPLE HARMONIC ***       MFS01730
C                                                                       MFS01740
      N=0.D0                                                            MFS01750
      R=1                                                               MFS01760
      MAXY=0.D0                                                         MFS01770
      DO 40 I=1,IND2P1                                                  MFS01780
          ARRAY(R,1)=CDABS(X(R))                                        MFS01790
          ARRAY(R,2)=N/STEP                                             MFS01800
          IF(ARRAY(R,1).GT.MAXY)THEN                                    MFS01810
              MAXY=ARRAY(R,1)                                           MFS01820
              RMAX=R-1                                                  MFS01830
          END IF                                                        MFS01840
          R=R+1                                                         MFS01850
          N=N+1.D0                                                      MFS01860
40    CONTINUE                                                          MFS01870
C                                                                       MFS01880
CC    *** DISPLAY INFO IF THE FIRST TIME THROUGH THE SUBPROGRAM ***     MFS01890
C                                                                       MFS01900
      IF(REP.EQ.1)THEN                                                  MFS01910
C                                                                       MFS01920
49    WRITE(10,50)RMAX                                                  MFS01930
50    FORMAT(' THE FFT HAS BEEN GENERATED!  THE PRINCIPLE HARMONIC',/,  MFS01940
     *' IS THE',I5,' HARMONIC.  THE NEXT PLOT TO BE PRODUCED WILL',/,   MFS01950
     *' BE THE AMPLITUDE SPECTRUM.  ENTER A 1 IF YOU ARE READY TO',/,   MFS01960
     *' CONTINUE WITH THE PROGRAM.',//,' :')                            MFS01970
C                                                                       MFS01980
      READ(5,*)L                                                        MFS01990
C                                                                       MFS02000
      IF(L.NE.1)THEN                                                    MFS02010
C         CALL FRTCMS('CLRSCRN ')                                       MFS02020
          WRITE(10,60)                                                  MFS02030
60        FORMAT(' ERROR',/)                                            MFS02040
          GO TO 49                                                      MFS02050
      END IF                                                            MFS02060
C                                                                       MFS02070
      END IF                                                            MFS02080
C                                                                       MFS02090
C     *** ALLOW FOR THE PLOT OF THE AMPLITUDE SPECTRUM OF THE FFT ***   MFS02100
```

```
C
      IF(REP.EQ.1)THEN                                                  MFS02110
          MINY=0.D0                                                     MFS02120
          INT=1.D0/STEP                                                 MFS02130
C                                                                       MFS02140
          CALL PLOT(MAXY,MINY,INT,IND2P1)                               MFS02150
      END IF                                                            MFS02160
C                                                                       MFS02170
      FLAG=0                                                            MFS02180
      CALL STATS(IN,REP,FLAG)                                           MFS02190
C                                                                       MFS02200
      RETURN                                                            MFS02210
      END                                                               MFS02220
                                                                        MFS02230
```

```
      SUBROUTINE QPRS(TYPE2,TYPE3,BAUD,BITS,FREQ,IPHAS,AMP,                QPR00110
     *ANS2,DSEED,IN,REP)                                                   QPR00120
C                                                                          QPR00130
CC    *** PURPOSE ***                                                      QPR00140
CC                                                                         QPR00150
CC    THIS SUBROUTINE MODULATES THE CARRIER USING QUADRATURE PHASE SHIFT   QPR00160
CC    KEYING AS THE MODULATION TECHNIQUE.                                  QPR00170
CC                                                                         QPR00180
CC    *** PARAMETER DEFINITIONS ***                                        QPR00190
CC                                                                         QPR00200
CC    TYPE2=       INDICATES LOGIC TYPE TO BE EMPLOYED                     QPR00210
CC    TYPE3=       INDICATES THE CLASS OF QPRS                             QPR00220
CC    BAUD=        SYMBOL RATE OR BAUD RATE                                QPR00230
CC    BITS=        NUMBER OF BITS IN EACH BINARY CODE WORD                 QPR00240
CC    FREQ=        CARRIER FREQUENCY                                       QPR00250
CC    IPHAS=       INITIAL PHASE ANGLE IN DEGREES                          QPR00260
CC    AMP=         AMPLITUDE                                               QPR00270
CC    ANS2=        INTEGER VARIABLE TO BE PASSED TO STREAM                 QPR00280
CC    DSEED=       DOUBLE PRECISION SEED FOR RANDOM NUMBER GENERATOR       QPR00290
CC    IN=          NUMBER OF POSITIONS IN ARRAY TO BE UTILIZED             QPR00300
CC    REP=         AN INTEGER EQUAL TO THE NUMBER OF THE REPETITION OF     QPR00310
CC                 THE CALL THE THIS SUBROUTINE                            QPR00320
CC                                                                         QPR00330
CC    *** VARIABLE DEFINITIONS ***                                         QPR00340
CC                                                                         QPR00350
CC    ARRAY=       A DOUBLE PRECISION ARRAY FOR PASSING THE BINARY         QPR00360
CC                 DIGITS AND STORING THE VALUE OF THE TIME FUNCTION       QPR00370
CC                 AND THE FFT                                             QPR00380
CC    ASUBN=       A DOUBLE PRECISION ARRAY CONTAINING THE VALUES OF       QPR00390
CC                 THE IN PHASE BINARY DIGIT TO BE MODULATED               QPR00400
CC    BSUBN=       A DOUBLE PRECISION ARRAY CONTAINING THE VALUES OF THE   QPR00410
CC                 QUADRATURE BINARY DIGIT TO BE MODULATED                 QPR00420
CC    HOFT=        THE IMPULSE RESPONSE FOR THE SPECIFIED CLASS FILTER     QPR00430
CC    T=           VARIABLE WHICH DETERMINES THE MAGNITUDE OF THE IMPULSE  QPR00440
CC                 RESPONSE AT A GIVEN TIME FOR A SPECIFIED BINARY DIGIT   QPR00450
CC    NDP=         VARIABLE USED IN THE COMPUTATION OF T                   QPR00460
CC    R=           AN INTEGER USED TO REPRESENT THE ROW OF ARRAY            QPR00470
CC    TIME=        TIME VARIABLE                                           QPR00480
CC    MT=          VALUE OF BINARY DIGIT PASSED FROM SUBPROGRAM STREAM      QPR00490
CC    STEP=        INTERVAL AT WHICH THE SIGNAL IS REPRODUCED               QPR00500
CC    PI=          NUMERICAL CONSTANT                                      QPR00510
CC    OMEGA=       CARRIER ANGULAR FREQUENCY                               QPR00520
CC    DELTA=       INITIAL PHASE OFFSET IN RADIANS                         QPR00530
CC    BAUDD=       BAUD DURATION AND QPRS BIT DURATION                     QPR00540
CC    X=           COMPLEX ARRAY TO RECEIVE THE FFT OF THE TIME SERIES      QPR00550
CC    IWK=         INTEGER WORKING ARRAY USED BY FUNCTION FFTRC            QPR00560
CC    MAXY=        MAXIMUM VALUE TO BE PLOTTED ON THE ABCISSAE              QPR00570
CC    MINY=        MINIMUM VALUE TO BE PLOTTED ON THE ABCISSAE              QPR00580
CC    INT=         INTERVAL BETWEEN POINTS ON THE ORDINATE                  QPR00590
CC    RMAX=        VALUE OF THE PRINCIPLE HARMONIC                         QPR00600
```

```
C      FLAG=         AN INTEGER WHICH CONTROL ENTRANCE AND EXIT FROM      QPR00610
CC                   SUBPROGRAM STATS                                    QPR00620
CC     IND2P1=       LENGTH OF ARRAY DIVIDED BY 2 + 1                    QPR00630
CC     MAX=          MAX NUMBER OF INCREMENTS IN 6 BAUD DURATIONS        QPR00640
CC     INDEX=        INDEX OF A LOOP                                     QPR00650
CC                                                                       QPR00660
CC     *** VARIABLE DECLARATIONS ***                                     QPR00670
C                                                                        QPR00680
       INTEGER R,ANS2,TYPE2,TYPE3,IN,RMAX,REP,FLAG,IND2P1,MAX,INDEX      QPR00690
C                                                                        QPR00700
       DOUBLE PRECISION BAUD,BITS,FREQ,IPHAS,AMP,TIME,MT1,MT2,           QPR00710
      *STEP,PI,OMEGA,DELTA,NBAUD,BAUDD,HOFT,T,NDP,                       QPR00720
      *DSEED,MAXY,MINY,INT,MT                                            QPR00730
C                                                                        QPR00740
       DOUBLE PRECISION ARRAY(1024,2),SUMX(513),SUMXSQ(513),SUMX3(513),  QPR00750
      *SUMX4(513),ASUBN(2000),BSUBN(2000)                                QPR00760
       COMMON ARRAY,SUMX,SUMXSQ,SUMX3,SUMX4                             QPR00770
C                                                                        QPR00780
       COMPLEX*16 X(513)                                                 QPR00790
C                                                                        QPR00800
       INTEGER IWK(10)                                                   QPR00810
CC                                                                       QPR00820
CC     *** VARIABLE INITIALIZATION ***                                   QPR00830
C                                                                        QPR00840
       R=1                                                               QPR00850
       PI=3.1415926535897 93D0                                           QPR00860
       BAUDD=1.D0/BAUD                                                   QPR00870
       TIME=6.D0/BAUD                                                    QPR00880
       STEP=1.D0/(2.D0*(FREQ+((PI+1.D0)*BAUD)))                          QPR00890
       MAX=TIME/STEP+1.D0                                                QPR00900
       INDEX=IN+MAX                                                      QPR00910
       OMEGA=2.D0*PI*FREQ                                                QPR00920
       DELTA=IPHAS*PI/180.D0                                             QPR00930
       MAXY=0.D0                                                         QPR00940
       IND2P1=IN/2+1                                                     QPR00950
C                                                                        QPR00960
       DO 10 I=1,INDEX                                                   QPR00970
            CALL STREAM(DSEED,ANS2,TYPE2,MT)                             QPR00980
            ASUBN(I)=MT                                                  QPR00990
            CALL STREAM(DSEED,ANS2,TYPE2,MT)                             QPR01000
            BSUBN(I)=MT                                                  QPR01010
10     CONTINUE                                                          QPR01020
C                                                                        QPR01030
       DO 20 I=1,1024                                                    QPR01040
            ARRAY(I,1)=0.D0                                              QPR01050
20     CONTINUE                                                          QPR01060
C                                                                        QPR01070
CC     *** DO THE MODULATION AND ASSIGN THE VALUES TO ARRAY***           QPR01080
C                                                                        QPR01090
       DO 30 I=1,IN                                                      QPR01100
```

```
C                                                                       QPR01110
      DO 40 J=1,INDEX                                                   QPR01120
        NDP=J-1                                                         QPR01130
        T=TIME-(NDP/BAUD)                                               QPR01140
        IF(TYPE3.EQ.1)THEN                                              QPR01150
          IF(T.EQ.-BAUDD/2.D0.OR.T.EQ.BAUDD/2.D0)THEN                   QPR01160
            HOFT=1.D0                                                   QPR01170
          ELSE                                                          QPR01180
            HOFT=(4.D0/(BAUD**2*PI))*(DCOS(PI*T*BAUD)/                  QPR01190
     *           ((1.D0/BAUD**2)-(4.D0*T**2)))                          QPR01200
          END IF                                                        QPR01210
        ELSE IF(TYPE3.EQ.2)THEN                                         QPR01220
          IF(T.EQ.0.D0)THEN                                             QPR01230
            HOFT=2.D0                                                   QPR01240
          ELSE IF(T.EQ.-BAUDD.OR.T.EQ.BAUDD)THEN                        QPR01250
            HOFT=1.D0                                                   QPR01260
          ELSE                                                          QPR01270
            HOFT=(2.D0/(BAUD**3*PI*T))*(DSIN(PI*T*BAUD)/                QPR01280
     *           ((1.D0/BAUD**2)-T**2))                                 QPR01290
          END IF                                                        QPR01300
        ELSE IF(TYPE3.EQ.3)THEN                                         QPR01310
          IF(T.EQ.0.D0)THEN                                             QPR01320
            HOFT=1.D0                                                   QPR01330
          ELSE IF(T.EQ.-BAUDD)THEN                                      QPR01340
            HOFT=2.D0                                                   QPR01350
          ELSE IF(T.EQ.BAUDD)THEN                                       QPR01360
            HOFT=-1.D0                                                  QPR01370
          ELSE                                                          QPR01380
            HOFT=(1.D0/(BAUD**2*PI*T))*(DSIN(PI*T*BAUD))*               QPR01390
     *           ((3.D0*T-BAUDD)/(T**2-(1.D0/BAUD**2)))                 QPR01400
          END IF                                                        QPR01410
        ELSE IF(TYPE3.EQ.4)THEN                                         QPR01420
          IF(T.EQ.-BAUDD)THEN                                           QPR01430
            HOFT=1.D0                                                   QPR01440
          ELSE IF(T.EQ.BAUDD)THEN                                       QPR01450
            HOFT=-1.D0                                                  QPR01460
          ELSE                                                          QPR01470
            HOFT=(2.D0/(BAUD**2*PI))*(DSIN(PI*T*BAUD)/                  QPR01480
     *           (T**2-(1.D0/BAUD**2)))                                 QPR01490
          END IF                                                        QPR01500
        ELSE IF(TYPE3.EQ.5)THEN                                         QPR01510
          IF(T.EQ.0.D0)THEN                                             QPR01520
            HOFT=-2.D0                                                  QPR01530
          ELSE IF(T.EQ.-2.D0*BAUDD.OR.T.EQ.2.D0*BAUDD)THEN              QPR01540
            HOFT=1.D0                                                   QPR01550
          ELSE                                                          QPR01560
            HOFT=(8.D0/(BAUD**3*PI*T))*(DSIN(PI*T*BAUD)/                QPR01570
     *           (T**2-(4.D0/BAUD**2)))                                 QPR01580
          END IF                                                        QPR01590
C                                                                       QPR01600
```

```
                END IF                                                          QPR01610
C                                                                               QPR01620
                ARRAY(I,1)=ARRAY(I,1)+((ASUBN(J)*HOFT*AMP*DCOS((OMEGA*          QPR01630
     *                    TIME)+DELTA))+(BSUBN(J)*HOFT*AMP*DSIN((OMEGA*        QPR01640
     *                    TIME)+DELTA)))                                        QPR01650
C                                                                               QPR01660
                IF(DABS(ARRAY(I,1)).GT.MAXY)THEN                                QPR01670
                    MAXY=DABS(ARRAY(I,1))                                       QPR01680
                END IF                                                          QPR01690
C                                                                               QPR01700
40          CONTINUE                                                            QPR01710
C                                                                               QPR01720
            ARRAY(I,2)=TIME                                                     QPR01730
            TIME=TIME+STEP                                                      QPR01740
C                                                                               QPR01750
30      CONTINUE                                                                QPR01760
CC                                                                              QPR01770
CC      *** PLOT THE TIME SERIES IF DESIRED ***                                 QPR01780
C                                                                               QPR01790
        IF(REP.EQ.1)THEN                                                        QPR01800
            MINY=-MAXY                                                          QPR01810
C                                                                               QPR01820
            CALL PLOT(MAXY,MINY,STEP,IN)                                        QPR01830
        END IF                                                                  QPR01840
CC                                                                              QPR01850
CC      *** GENERATE THE FFT ***                                                QPR01860
CC                                                                              QPR01870
        CALL FFTRC(ARRAY(1,1),IN,X,IWK)                                         QPR01880
C                                                                               QPR01890
CC      *** CALCULATE THE AMPLITUDE SPECTRUM OF THE FUNCTION ***                QPR01900
CC      *** FIND THE NUMBER AND VALUE OF THE PRINCIPLE HARMONIC ***             QPR01910
C                                                                               QPR01920
        N=0.D0                                                                  QPR01930
        R=1                                                                     QPR01940
        MAXY=0.D0                                                               QPR01950
        DO 50 I=1,IND2P1                                                        QPR01960
            ARRAY(R,1)=CDABS(X(R))                                              QPR01970
            ARRAY(R,2)=N/STEP                                                   QPR01980
            IF(ARRAY(R,1).GT.MAXY)THEN                                          QPR01990
                MAXY=ARRAY(R,1)                                                 QPR02000
                RMAX=R-1                                                        QPR02010
            END IF                                                              QPR02020
            R=R+1                                                               QPR02030
            N=N+1.D0                                                            QPR02040
50      CONTINUE                                                                QPR02050
CC                                                                              QPR02060
CC      *** DISPLAY INFO IF THE FIRST TIME THROUGH THE SUBPROGRAM ***           QPR02070
C                                                                               QPR02080
        IF(REP.EQ.1)THEN                                                        QPR02090
C                                                                               QPR02100
```

```
59     WRITE(10,60)RMAX                                                    QPR02110
60     FORMAT(' THE FFT HAS BEEN GENERATED!  THE PRINCIPLE HARMONIC',/,    QPR02120
     *' IS THE',I5,' HARMONIC.  THE NEXT PLOT TO BE PRODUCED WILL',/,      QPR02130
     *' BE THE AMPLITUDE SPECTRUM.  ENTER A 1 IF YOU ARE READY TO',/,      QPR02140
     *' CONTINUE WITH THE PROGRAM.',//,' :')                               QPR02150
C                                                                          QPR02160
       READ(5,*)L                                                          QPR02170
C                                                                          QPR02180
       IF(L.NE.1)THEN                                                      QPR02190
C          CALL FRTCMS('CLRSCRN ')                                         QPR02200
           WRITE(10,70)                                                    QPR02210
70         FORMAT(' ERROR',/)                                              QPR02220
           GO TO 59                                                        QPR02230
       END IF                                                              QPR02240
C                                                                          QPR02250
       END IF                                                              QPR02260
C                                                                          QPR02270
CC     *** ALLOW FOR THE PLOT OF THE AMPLITUDE SPECTRUM OF THE FFT ***     QPR02280
C                                                                          QPR02290
       IF(REP.EQ.1)THEN                                                    QPR02300
           MINY=0.D0                                                       QPR02310
           INT=1.D0/STEP                                                   QPR02320
C                                                                          QPR02330
           CALL PLOT(MAXY,MINY,INT,IND2P1)                                 QPR02340
       END IF                                                              QPR02350
C                                                                          QPR02360
CC     *** ADD THE VALUES OF THE FFT TO THE ACCUMULATED STATISTICS ***     QPR02370
C                                                                          QPR02380
       FLAG=0                                                              QPR02390
       CALL STATS(IN,REP,FLAG)                                             QPR02400
C                                                                          QPR02410
       RETURN                                                              QPR02420
       END                                                                 QPR02430
```

```
      SUBROUTINE STREAM(DSEED,ANS2,TYPE2,MT)                            STR00110
C                                                                       STR00120
C     *** PURPOSE ***                                                   STR00130
C                                                                       STR00140
C     THIS SUBPROGRAM ALLOWS THE INPUT OF SUCCESSIVE BITS DURING        STR00150
C     MODULATION                                                        STR00160
C                                                                       STR00170
C     *** PARAMETER DEFINITIONS ***                                     STR00180
C                                                                       STR00190
C     DSEED=       DOUBLE PRECISION SEED FOR RANDOM NUMBER GENERATOR    STR00200
C     ANS2=        WHETHER OR NOT THE BIT STREAM IS TO BE GENERATED     STR00210
C                  BY THE RANDOM NUMBER GENERATOR                       STR00220
C     TYPE2=       THE TYPE OF BINARY LOGIC TO BE EMPLOYED              STR00230
C     MT=          VALUE OF THE BINARY DIGIT TO BE PASSED               STR00240
C                                                                       STR00250
C     *** VARIABLE DEFINITIONS ***                                      STR00260
C                                                                       STR00270
C     GGUBFS=      RANDOM NUMBER GENERATOR                              STR00280
C     IMT=         INTEGER VALUE OF BINARY DIGIT TO BE MODULATED        STR00290
C                                                                       STR00300
C     *** VARIABLE DECLARATIONS ***                                     STR00310
C                                                                       STR00320
      INTEGER ANS2,TYPE2,IMT                                            STR00330
C                                                                       STR00340
      DOUBLE PRECISION DSEED,MT                                         STR00350
C                                                                       STR00360
      IF(ANS2.EQ.1)THEN                                                 STR00370
          IF(TYPE2.EQ.1)THEN                                            STR00380
              IF(GGUBFS(DSEED).LE..5D0)THEN                             STR00390
                  MT=1.D0                                               STR00400
              ELSE                                                      STR00410
                  MT=-1.D0                                              STR00420
              END IF                                                    STR00430
          ELSE IF(TYPE2.EQ.2)THEN                                       STR00440
              IF(GGUBFS(DSEED).LE..5D0)THEN                             STR00450
                  MT=1.D0                                               STR00460
              ELSE                                                      STR00470
                  MT=0.D0                                               STR00480
              END IF                                                    STR00490
          ELSE IF(TYPE2.EQ.3)THEN                                       STR00500
              IF(GGUBFS(DSEED).LE..5D0)THEN                             STR00510
                  MT=-1.D0                                              STR00520
              ELSE                                                      STR00530
                  MT=0.D0                                               STR00540
              END IF                                                    STR00550
          END IF                                                        STR00560
      ELSE IF(ANS2.EQ.2)THEN                                            STR00570
          WRITE(10,10)                                                  STR00580
10        FORMAT(' ENTER THE NEXT BIT IN THE DESIRED BIT STREAM.',/)    STR00590
          READ(5,*)IMT                                                  STR00600
```

```
      MT=IMT                                                    STR00610
    END IF                                                      STR00620
    RETURN                                                      STR00630
    END                                                         STR00640
```

```
      SUBROUTINE ORTHO(K,MT,SUM2,C,FLAG)                                ORT00110
C                                                                       ORT00120
C     *** PURPOSE ***                                                   ORT00130
C                                                                       ORT00140
C     THIS SUBROUTINE GENERATES AN ORTHOGONAL SET OF N VECTORS OF LENGTHORT00150
C     N, WHERE N IS 2**K, WHERE K IS THE LENGTH OF THE BINARY CODE WORD.ORT00160
C     AND DETERMINES THE VALUE, MT, OF A BINARY DIGIT TO BE RETURNED    ORT00170
C                                                                       ORT00180
C     *** PARAMETER DEFINITIONS ***                                     ORT00190
C                                                                       ORT00200
C     K=            NUMBER OF BITS IN THE BINARY CODE WORD              ORT00210
C     MT=           THE VALUE OF THE BINARY DIGIT TO BE MODULATED       ORT00220
C     SUM2=         THE DECIMAL EQUIVALENT OF A SET OF RANDOMLY DRAWN   ORT00230
C                   BINARY DIGITS                                       ORT00240
C     C=            THE COLUMN OF THE VECTOR WHERE THE BINARY DIGIT TO  ORT00250
C                   BE MODULATED IS LOCATED                             ORT00260
C     FLAG=         INTEGER WHICH CONTROLS POINT OF ENTRY TO SUBROUTINE  ORT00270
C                   CRTHO                                               ORT00280
C                                                                       ORT00290
C     *** VARIABLE DEFINITIONS ***                                      ORT00300
C                                                                       ORT00310
C     ROW=          ROW OF ARRAY H2N WHICH MATCHES A BINARY CODE WORD   ORT00320
C     KM1=          INTEGER VALUE OF K MINUS 1                          ORT00330
C     N=            NUMBER OF POSSIBLE BINARY CODE WORDS = 2**K         ORT00340
C     H2N=          MATRIX OF N ORTHOGONAL VECTORS OF LENGTH N          ORT00350
C     RH2N=         ROW OF MATRIX H2N                                   ORT00360
C     CH2N=         COLUMN OF MATRIX H2N                                ORT00370
C     H2NR=         RANK OF MATRIX H2N                                  ORT00380
C     RH2NP1=       ROW OF MATRIX H2N EQUAL TO HALF THE RANK PLUS 1     ORT00390
C     CH2NP1=       COLUMN OF MATRIX H2N EQUAL TO HALF THE RANK PLUS 1  ORT00400
C     X=            DOUBLE PRECISION EQUIVALENT ROW REPRESENTATION OF H2N ORT00410
C                                                                       ORT00420
C     *** VARIABLE DECLARATIONS ***                                     ORT00430
C                                                                       ORT00440
      INTEGER K,RH2N,CH2N,RH2NP1,CH2NP1,H2NR,N,KM1,H2N(65,65),          ORT00450
     *FLAG,ROW,C                                                        ORT00460
C                                                                       ORT00470
      DOUBLE PRECISION X,MT,SUM2                                        ORT00480
C                                                                       ORT00490
C     *** VARIABLE INITIALIZATION ***                                   ORT00500
C                                                                       ORT00510
      KM1=K-1                                                           ORT00520
      N=2**K                                                            ORT00530
      RH2N=1                                                            ORT00540
      CH2N=1                                                            ORT00550
      H2NR=2                                                            ORT00560
      RH2NP1=H2NR+1                                                     ORT00570
      CH2NP1=H2NR+1                                                     ORT00580
      H2N(1,1)=1                                                        ORT00590
      H2N(1,2)=1                                                        ORT00600
```

```
      H2N(2,1)=1                                                        ORT00610
      H2N(2,2)=-1                                                       ORT00620
C                                                                       ORT00630
C     *** GENERATE A MATRIX OF N ORTHOGONAL VECTORS OF LENGTH N ***     ORT00640
C                                                                       ORT00650
      IF(FLAG.EQ.0)THEN                                                 ORT00660
      DO 10 I=1,KM1                                                     ORT00670
          DO 20 L=1,H2NR                                                ORT00680
              DO 30 M=1,H2NR                                            ORT00690
                  H2N(RH2N,CH2N)=H2N(RH2N,CH2N)                         ORT00700
                  H2N(RH2N,CH2NP1)=H2N(RH2N,CH2N)                       ORT00710
                  H2N(RH2NP1,CH2N)=H2N(RH2N,CH2N)                       ORT00720
                  H2N(RH2NP1,CH2NP1)=-H2N(RH2N,CH2N)                    ORT00730
                  CH2N=CH2N+1                                           ORT00740
                  CH2NP1=CH2NP1+1                                       ORT00750
30            CONTINUE                                                  ORT00760
              RH2N=RH2N+1                                               ORT00770
              RH2NP1=RH2NP1+1                                           ORT00780
              CH2N=1                                                    ORT00790
              CH2NP1=H2NR+1                                             ORT00800
20        CONTINUE                                                      ORT00810
          H2NR=2*H2NR                                                   ORT00820
          RH2N=1                                                        ORT00830
          CH2N=1                                                        ORT00840
          RH2NP1=H2NR+1                                                 ORT00850
          CH2NP1=H2NR+1                                                 ORT00860
10    CONTINUE                                                          ORT00870
C                                                                       ORT00880
C     *** ASSIGN EACH ROW OF H2N A REPRESENTATIVE DECIMAL NUMBER ***     ORT00890
C                                                                       ORT00900
      DO 40 I=1,N                                                       ORT00910
          H2N(I,65)=I-1                                                 ORT00920
40    CONTINUE                                                          ORT00930
C                                                                       ORT00940
C     *** DETERMINE WHICH ROW OR H2N MATCHES THE VALUE OF THE           ORT00950
C     REPRESENTATIVE DECIMAL EQUIVALENT TO THE BINARY CODE WORD ***     ORT00960
C                                                                       ORT00970
      DO 50 I=1,N                                                       ORT00980
          X=H2N(I,65)                                                   ORT00990
          IF(SUM2.EQ.X)THEN                                             ORT01000
              ROW=I                                                     ORT01010
          END IF                                                        ORT01020
50    CONTINUE                                                          ORT01030
C                                                                       ORT01040
      END IF                                                            ORT01050
C                                                                       ORT01060
C     *** DETERMINE THE NEW ROW OF H2N ***                              ORT01070
C                                                                       ORT01080
      IF(FLAG.EQ.1)THEN                                                 ORT01090
          DO 60 I=1,N                                                   ORT01100
```

```
                X=H2N(I,65)                                                    ORT01110
                IF(SUM2.EQ.X) THEN                                             ORT01120
                   ROW=I                                                       ORT01130
                END IF                                                         ORT01140
60          CONTINUE                                                           ORT01150
         END IF                                                                ORT01160
C                                                                              ORT01170
CC       *** DETERMINE THE VALUE OF MT TO BE RETURNED ***                      ORT01180
C                                                                              ORT01190
         MT=H2N(ROW,C)                                                         ORT01200
         C=C+1                                                                 ORT01210
C                                                                              ORT01220
         RETURN                                                                ORT01230
         END                                                                   ORT01240
```

```
      SUBROUTINE PLOT(MAXY,MINY,INT,IN)                                 PLO00110
C                                                                       PLO00120
CCC   *** PURPOSE ***                                                   PLO00130
C                                                                       PLO00140
CCC   THIS SUBPROGRAM CONTROL THE USERS ABILITY TO MAKE A PLOT OF THE   PLO00150
CCC   TIME FUNCTION OR THE FFT                                          PLO00160
C                                                                       PLO00170
CCC   *** PARAMETER DEFINITIONS ***                                     PLO00180
C                                                                       PLO00190
CCC   MAXY=        MAXIMUM VALUE TO BE PLOTTED ON THE ABCISSAE          PLO00200
CCC   MINY=        MINIMUM VALUE TO BE PLOTTED ON THE ABCISSAE          PLO00210
CCC   INT=         DELTA TIME AT WHICH THE FUNCTION IS REPRODUCED       PLO00220
CCC   IN=          MAX LENGTH OF VECTOR TO BE PLOTTED                   PLO00230
C                                                                       PLO00240
CCC   *** VARIABLE DEFINITIONS ***                                      PLO00250
C                                                                       PLO00260
CCC   ANS=         RESPONSE TO INQUIRY                                  PLO00270
CCC   ARRAY=       ARRAY CONTAINING VALUES TO BE PLOTTED                PLO00280
CCC   RANGE=       ARRAY TO SPECIFY MIN AND MAX VALUES FOR THE ORDINATE PLO00290
CCC                AND ABCISSA OF THE PLOT                              PLO00300
CCC   NI=          INTEGER NUMBER OF X,Y PAIRS TO BE PLOTTED            PLO00310
CCC   ND=          DOUBLE PRECISION OF NI                               PLO00320
CCC   R=           INTEGER VALUE FOR ROW OF ARRAY                       PLO00330
C                                                                       PLO00340
CCC   *** VARIABLE DECLARATIONS ***                                     PLO00350
C                                                                       PLO00360
      DOUBLE PRECISION MAXY,MINY,INT,ND                                 PLO00370
C                                                                       PLO00380
      DOUBLE PRECISION ARRAY(1024,2),SUMX(513),SUMXSQ(513),SUMX3(513),  PLO00390
     *SUMX4(513)                                                        PLO00400
      COMMON ARRAY,SUMX,SUMXSQ,SUMX3,SUMX4                              PLO00410
C                                                                       PLO00420
      REAL RANGE(4)                                                     PLO00430
C                                                                       PLO00440
      INTEGER ANS,NI,R,IN                                               PLO00450
C                                                                       PLO00460
      CALL FRTCMS('CLRSCRN ')                                           PLO00470
C                                                                       PLO00480
      WRITE(10,10)                                                      PLO00490
10    FORMAT(' ENTER A 1 IF YOU DESIRE TO GENERATE A PLOT.',/,          PLO00500
     *' ENTER ANY OTHER INTEGER TO CONTINUE THE PROGRAM.',//,' :')       PLO00510
C                                                                       PLO00520
      READ(5,*)ANS                                                      PLO00530
C                                                                       PLO00540
      IF(ANS.EQ.1)THEN                                                  PLO00550
C          CALL FRTCMS('CLRSCRN ')                                      PLO00560
19         WRITE(10,20)IN,IN                                            PLO00570
20         FORMAT(' YOU MAY PLOT A MAXIMUM OF',I5,' X,Y PAIRS ON YOUR'  PLO00580
     *     ,/,' GRAPH.  IT IS RECOMMENDED THAT YOU PLOT NO MORE THAN'   PLO00590
     *     ,/,' 129 CONSECUTIVE PAIRS OF COORDINATE POINTS SO YOU',/,   PLO00600
```

```
     *     ' WILL HAVE THE OPPORTUNITY TO PLOT MORE THAN ONE SECTION',/,  PLO00610
     *     ' OF THE TOTAL RECORD OF',I5,' POINTS.  ENTER THE NUMBER',/,   PLO00620
     *     ' OF POINTS TO BE PLOTTED.',//,' :')                           PLO00630
C                                                                         PLO00640
      READ(5,*)NI                                                         PLO00650
C                                                                         PLO00660
      IF(NI.LE.IN)THEN                                                    PLO00670
          CONTINUE                                                        PLO00680
      ELSE                                                                PLO00690
C         CALL FRTCMS('CLRSCRN ')                                         PLO00700
          WRITE(10,30)                                                    PLO00710
30        FORMAT(' ERROR',/)                                              PLO00720
          GO TO 19                                                        PLO00730
      END IF                                                              PLO00740
C                                                                         PLO00750
C     CALL FRTCMS('CLRSCRN ')                                             PLO00760
39    WRITE(10,40)IN                                                      PLO00770
40    FORMAT(' AT WHICH NUMBER OF THE',I5,' X,Y PAIRS DO YOU WISH',/      PLO00780
     *     ' TO START THE PLOT.',//,                                      PLO00790
     *     ' CAUTION!  THE POINT YOU SPECIFY TO START THE PLOT MUST BE',  PLO00800
     *     /,' SMALL ENOUGH TO ALLOW THE ENTIRE RANGE OF POINTS YOU',/,   PLO00810
     *     ' DESIRED TO HAVE PLOTTED AS SPECIFIED BY YOUR PREVIOUS INPUT' PLO00820
     *     )                                                              PLO00830
C                                                                         PLO00840
      READ(5,*)R                                                          PLO00850
C                                                                         PLO00860
      IF(R.GT.((IN+1)-NI))THEN                                            PLO00870
C         CALL FRTCMS('CLRSCRN ')                                         PLO00880
          WRITE(10,50)                                                    PLO00890
50        FORMAT(' ERROR',/)                                              PLO00900
          GO TO 39                                                        PLO00910
      END IF                                                              PLO00920
C                                                                         PLO00930
      ND=NI                                                               PLO00940
      RANGE(1)=(ARRAY(R,2)+(ND*INT))                                      PLO00950
      RANGE(2)=ARRAY(R,2)                                                 PLO00960
      RANGE(3)=1.5D0*MAXY                                                 PLO00970
      RANGE(4)=1.5D0*MINY                                                 PLO00980
C                                                                         PLO00990
      WRITE(10,60)                                                        PLO01000
60    FORMAT('1')                                                         PLO01010
      CALL UTPLOT(ARRAY(R,2),ARRAY(R,1),NI,RANGE,2,0)                     PLO01020
C                                                                         PLO01030
      WRITE(10,70)                                                        PLO01040
70    FORMAT(' IF YOU WOULD LIKE ANOTHER PLOT OF THE SAME GRAPH OR',      PLO01050
     *     /,' A PLOT OVER A DIFFERENT RANGE OR FROM ANOTHER STARTING',/, PLO01060
     *     ' POINT, ENTER A 1.  ENTER ANY OTHER INTEGER TO CONTINUE WITH' PLO01070
     *     ,/,' THE PROGRAM.',//,' :')                                    PLO01080
C                                                                         PLO01090
      READ(5,*)ANS                                                        PLO01100
```

```
C
      IF(ANS.EQ.1)THEN                                          PLO01110
          GO TO 19                                              PLO01120
      END IF                                                    PLO01130
  END IF                                                        PLO01140
C                                                               PLO01150
  RETURN                                                        PLO01160
  END                                                           PLO01170
                                                                PLO01180
```

```
      SUBROUTINE STATS(IN,REP,FLAG)                                     STA00110
C                                                                       STA00120
C     *** PURPOSE ***                                                   STA00130
C                                                                       STA00140
C     THIS SUBPROGRAM COMPILES THE ACCUMULATED STATISTICS OF THE        STA00150
C     ELEMENTS OF THE AMPLITUDE SPECTRUM OF THE FFTS.                   STA00160
C                                                                       STA00170
C     *** PARAMETER DEFINITIONS ***                                     STA00180
C                                                                       STA00190
C     IN=         INTEGER OF THE NUMBER OF POSITIONS USED IN ARRAY      STA00200
C     REP=        INTEGER OF THE NUMBER OF THE REPETITION OF THE        STA00210
C                 GENERATION OF THE AMPLITUDE SPECTRUM OF THE FFT       STA00220
C     FLAG=       AN INTEGER THAT CONTROLS THE POINT OF ENTRANCE OR      STA00230
C                 EXIT FROM THE SUBROUTINE                              STA00240
C                                                                       STA00250
C     *** VARIABLE DEFINITIONS ***                                      STA00260
C                                                                       STA00270
C     IND2P1=     POSITIONS USED IN ARRAYS SUMX AND SUMXSQ              STA00280
C     ARRAY=      AN ARRAY CONTAINING THE VALUES OF THE AMPLITUDE       STA00290
C                 SPECTRUM OF THE FFT                                   STA00300
C     SUMX=       AN ARRAY CONTAINING THE SUM OF THE VALUES OF THE      STA00310
C                 AMPLITUDE SPECTRUM OF THE FFT                         STA00320
C     SUMXSQ=     AN ARRAY CONTAINING THE SUM OF THE SQUARES OF THE     STA00330
C                 VALUES OF THE AMPLITUDE SPECTRUM OF THE FFT           STA00340
C     SUMX3=      AN ARRAY CONTAINING THE SUM OF THE SQUARES OF THE     STA00350
C                 VALUES OF THE AMPLITUDE SPECTRUM OF THE FFT           STA00360
C     SUMX4=      AN ARRAY CONTAINING THE SUM OF THE QUARTICS OF THE    STA00370
C                 VALUES OF THE AMPLITUDE SPECTRUM OF THE FFT           STA00380
C     XBAR=       AN ARRAY CONTAINING THE AVERAGE VALUE OF THE          STA00390
C                 ELEMENTS OF THE AMPLITUDE SPECTRUM                    STA00400
C     VAR=        AN ARRAY CONTAINING THE VARIANCE OF THE VALUES OF     STA00410
C                 THE ELEMENTS OF THE AMPLITUDE SPECTRUM                STA00420
C     SKEW=       AN ARRAY CONTAINING THE SKEWNESS OF THE DISTRIBUTION  STA00430
C                 OF THE ELEMENTS OF THE AMPLITUDE SPECTRUM             STA00440
C     KUR=        AN ARRAY CONTAINING THE KURTOSIS OF THE DISTRIBUTION  STA00450
C                 OF THE ELEMENTS OF THE AMPLITUDE SPECTRUM             STA00460
C     SUMVAR=     SUM OF THE VARIANCES OF AN ELEMENT OF THE AMPLITUDE   STA00470
C                 SPECTRUM                                              STA00480
C     AVAR=       AVERAGE VARIANCE OF AN ELEMENT OF THE AMPLITUDE       STA00490
C                 SPECTRUM                                              STA00500
C     VARVAR=     VARIANCE OF THE VARIANCES OF AN ELEMENT OF THE        STA00510
C                 AMPLITUDE SPECTRUM                                    STA00520
C     SUMVSQ=     SUM OF THE SQUARE OF THE VARIANCES OF AN ELEMENT OF   STA00530
C                 AMPLITUDE SPECTRUM                                    STA00540
C     SUMSKW=     SUM OF THE SKEWNESS OF THE ELEMENT OF THE AMPLITUDE   STA00550
C                 SPECTRUM                                              STA00560
C     ASKEW=      AVERAGE SKEWNESS OF THE ELEMENTS OF THE AMPLITUDE     STA00570
C                 SPECTRUM                                              STA00580
C     VARSKW=     VARIANCE OF THE SKEWNESS OF THE ELEMENTS OF THE       STA00590
C                 AMPLITUDE SPECTRUM                                    STA00600
```

```
C      SUMSSQ=      SUM OF THE SQUARE OF THE SKEWNESS OF THE ELEMENTS OF   STA00610
CC                  THE AMPLITUDE SPECTRUM                                 STA00620
C      SUMKUR=      SUM OF THE KURTOSIS OF THE ELEMENTS OF THE AMPLITUDE   STA00630
CC                  SPECTRUM                                               STA00640
C      AKUR=        AVERAGE KURTOSIS OF AN ELEMENT OF THE AMPLITUDE        STA00650
CC                  SPECTRUM                                               STA00660
C      VARKUR=      VARIANCE OF THE KURTOSIS OF AN ELEMENT OF THE          STA00670
CC                  AMPLITUDE SPECTRUM                                     STA00680
C      SUMKSQ=      SUM OF THE SQUARE OF THE KURTOSIS OF THE ELEMENTS OF   STA00690
CC                  THE AMPLITUDE SPECTRUM                                 STA00700
C                                                                          STA00710
C      REPDP=       DOUBLE PRECISION VALUE OF REP                          STA00720
C                                                                          STA00730
C      *** VARIABLE DECLARATIONS ***                                       STA00740
C                                                                          STA00750
       INTEGER IN,REP,FLAG,IND2P1                                         STA00760
C                                                                          STA00770
       DOUBLE PRECISION REPDP,SUMVAR,SUMVSQ,AVAR,VARVAR,SUMSKW,SUMSSQ,    STA00780
      *ASKEW,VARSKW,SUMKUR,SUMKSQ,AKUR,VARKUR,NDP                          STA00790
C                                                                          STA00800
       DOUBLE PRECISION ARRAY(1024,2),SUMX(513),SUMXSQ(513),SUMX3(513),    STA00810
      *SUMX4(513),XBAR(513),VAR(513),SKEW(513),KUR(513)                    STA00820
       COMMON ARRAY,SUMX,SUMXSQ,SUMX3,SUMX4                               STA00830
C                                                                          STA00840
C      *** VARIABLE INITIALIZATION ***                                     STA00850
C                                                                          STA00860
       IND2P1=IN/2+1                                                      STA00870
       IF(REP.EQ.1)THEN                                                   STA00880
           DO 10 I=1,IND2P1                                               STA00890
               SUMX(I)=0.D0                                               STA00900
               SUMXSQ(I)=0.D0                                             STA00910
               SUMX3(I)=0.D0                                              STA00920
               SUMX4(I)=0.D0                                              STA00930
10         CONTINUE                                                       STA00940
       END IF                                                             STA00950
C                                                                          STA00960
C      *** IF FINAL TIME THROUGH STATS, COMPUTE THE OUTPUT STATISTICS ***  STA00970
C                                                                          STA00980
       IF(FLAG.EQ.1)THEN                                                  STA00990
           GO TO 30                                                       STA01000
       END IF                                                             STA01010
C                                                                          STA01020
C      *** ADD THE ELEMENTS OF ARRAY TO THE ELEMENTS OF SUMX, SUMXSQ,      STA01030
C      SUMX3, AND SUMX4 ***                                                STA01040
C                                                                          STA01050
       DO 20 I=1,IND2P1                                                   STA01060
           SUMX(I)=SUMX(I)+ARRAY(I,1)                                     STA01070
           SUMXSQ(I)=SUMXSQ(I)+ARRAY(I,1)**2                              STA01080
           SUMX3(I)=SUMX3(I)+ARRAY(I,1)**3                                STA01090
           SUMX4(I)=SUMX4(I)+ARRAY(I,1)**4                                STA01100
```

```
20    CONTINUE                                                                  STA01110
C                                                                               STA01120
C     *** COMPUTE THE FINAL STATISTICS IF FLAG EQUALS 1 ***                     STA01130
C                                                                               STA01140
30    IF(FLAG.EQ.1)THEN                                                         STA01150
C                                                                               STA01160
C     *** COMPUTE STATISTICS ASSOCIATED WITH EACH ELEMENT OF FFT ***            STA01170
C                                                                               STA01180
      REP=REP-1                                                                 STA01190
      REPDP=REP                                                                 STA01200
      DO 40 I=1,IND2P1                                                          STA01210
        XBAR(I)=SUMX(I)/REP                                                     STA01220
        IF(REP.EQ.1)THEN                                                        STA01230
        VAR(I)=((REPDP*SUMXSQ(I))-SUMX(I)**2)/REPDP**2                          STA01240
        ELSE                                                                    STA01250
        VAR(I)=((REPDP*SUMXSQ(I))-SUMX(I)**2)/(REPDP*(REPDP-1.D0))              STA01260
        END IF                                                                  STA01270
        SKEW(I)=(SUMX3(I)/REPDP)-(3.D0*(SUMXSQ(I)/REPDP)*                       STA01280
     *                (SUMX(I)/REPDP))+(2.D0*(SUMX(I)/REPDP)**2)                STA01290
        KUR(I)=(SUMX4(I)/REPDP)-(4.D0*(SUMX3(I)/REPDP)*                         STA01300
     *               (SUMX(I)/REPDP))+(6.D0*(SUMX(I)/REPDP)**2*                 STA01310
     *               (SUMXSQ(I)/REPDP))-(3.D0*(SUMX(I)/REPDP)**4)               STA01320
40    CONTINUE                                                                  STA01330
C                                                                               STA01340
C     *** OUTPUT SOME DATA ***                                                  STA01350
C                                                                               STA01360
      WRITE(6,50)                                                               STA01370
50    FORMAT('1',2X,' N',11X,' MEAN',17X,' VARIANCE',17X,' SKEWNESS'            STA01380
     *  ,16X,' KURTOSIS')                                                       STA01390
C                                                                               STA01400
      DO 60 I=1,IND2P1                                                          STA01410
        WRITE(6,70)I,XBAR(I),VAR(I),SKEW(I),KUR(I)                              STA01420
70      FORMAT(1X,I5,2X,E23.16,2X,E23.16,2X,E23.16,2X,E23.16)                   STA01430
60    CONTINUE                                                                  STA01440
C                                                                               STA01450
      WRITE(6,61)                                                               STA01460
61    FORMAT('1',2X,' N',10X,' SUM X',18X,' SUM X**2',16X,                      STA01470
     *  ' SUM X**3',16X,' SUM X**4')                                            STA01480
C                                                                               STA01490
      DO 62 I=1,IND2P1                                                          STA01500
        WRITE(6,63)I,SUMX(I),SUMXSQ(I),SUMX3(I),SUMX4(I)                        STA01510
63      FORMAT(1X,I5,2X,E23.16,2X,E23.16,2X,E23.16,2X,E23.16)                   STA01520
62    CONTINUE                                                                  STA01530
C                                                                               STA01540
C     *** COMPUTE STATISTICS ASSOCIATED WITH ALL ELEMENTS OF FFT ***            STA01550
C                                                                               STA01560
      SUMVAR=0.D0                                                               STA01570
      SUMVSQ=0.D0                                                               STA01580
      VARVAR=0.D0                                                               STA01590
      SUMSKW=0.D0                                                               STA01600
```

```
      SUMSSQ=0.D0                                                       STA01610
      VARSKW=0.D0                                                       STA01620
      SUMKUR=0.D0                                                       STA01630
      SUMKSQ=0.D0                                                       STA01640
      VARKUR=0.D0                                                       STA01650
      NDP=IND2P1                                                        STA01660
C                                                                       STA01670
      DO 80 I=1,IND2P1                                                  STA01680
          SUMVAR=SUMVAR+VAR(I)                                          STA01690
          SUMVSQ=SUMVSQ+(VAR(I)**2)                                     STA01700
          SUMSKW=SUMSKW+SKEW(I)                                         STA01710
          SUMSSQ=SUMSSQ+(SKEW(I)**2)                                    STA01720
          SUMKUR=SUMKUR+KUR(I)                                          STA01730
          SUMKSQ=SUMKSQ+(KUR(I)**2)                                     STA01740
80    CONTINUE                                                          STA01750
C                                                                       STA01760
      AVAR=SUMVAR/NDP                                                   STA01770
      ASKEW=SUMSKW/NDP                                                  STA01780
      AKUR=SUMKUR/NDP                                                   STA01790
C                                                                       STA01800
      DO 90 I=1,IND2P1                                                  STA01810
          VARVAR=VARVAR+(VAR(I)-AVAR)**2                                STA01820
          VARSKW=VARSKW+(SKEW(I)-ASKEW)**2                              STA01830
          VARKUR=VARKUR+(KUR(I)-AKUR)**2                                STA01840
90    CONTINUE                                                          STA01850
C                                                                       STA01860
      VARVAR=VARVAR/(NDP-1.D0)                                          STA01870
      VARSKW=VARSKW/(NDP-1.D0)                                          STA01880
      VARKUR=VARKUR/(NDP-1.D0)                                          STA01890
C                                                                       STA01900
      WRITE(6,91)                                                       STA01910
91    FORMAT('1',2X,' SUM OF VARIANCES',7X,' SUM OF VARIANCES**2')      STA01920
C                                                                       STA01930
      WRITE(6,92)SUMVAR,SUMVSQ                                          STA01940
92    FORMAT(1X,E23.16,2X,E23.16)                                       STA01950
C                                                                       STA01960
      WRITE(6,93)                                                       STA01970
93    FORMAT('0',2X,' SUM OF SKEWNESS',9X,' SUM OF SKEWNESS**2')        STA01980
C                                                                       STA01990
      WRITE(6,94)SUMSKW,SUMSSQ                                          STA02000
94    FORMAT(1X,E23.16,2X,E23.16)                                       STA02010
C                                                                       STA02020
      WRITE(6,95)                                                       STA02030
95    FORMAT('0',2X,' SUM OF KURTOSIS',9X,' SUM OF KURTOSIS**2')        STA02040
C                                                                       STA02050
      WRITE(6,96)SUMKUR,SUMKSQ                                          STA02060
96    FORMAT(1X,E23.16,2X,E23.16)                                       STA02070
C                                                                       STA02080
      WRITE(6,100)                                                      STA02090
100   FORMAT('0',3X,' MEAN VARIANCE',7X,' VARIANCE OF THE VARIANCES'    STA02100
```

```
     *    )                                                                  STA02110
C                                                                           STA02120
          WRITE(6,101)AVAR,VARVAR                                           STA02130
101       FORMAT(1X,E23.16,2X,E23.16)                                       STA02140
C                                                                           STA02150
          WRITE(6,102)                                                      STA02160
102       FORMAT('0',3X,' MEAN SKEWNESS',7X,' VARIANCE OF THE SKEWNESS')    STA02170
C                                                                           STA02180
          WRITE(6,103)ASKEW,VARSKW                                          STA02190
103       FORMAT(1X,E23.16,2X,E23.16)                                       STA02200
C                                                                           STA02210
          WRITE(6,104)                                                      STA02220
104       FORMAT('0',3X,' MEAN KURTOSIS',7X,' VARIANCE OF THE KURTOSIS')    STA02230
C                                                                           STA02240
          WRITE(6,105)AKUR,VARKUR                                           STA02250
105       FORMAT(1X,E23.16,2X,E23.16)                                       STA02260
C                                                                           STA02270
          WRITE(6,106)                                                      STA02280
106       FORMAT('1')                                                       STA02290
C                                                                           STA02300
      END IF                                                                STA02310
C                                                                           STA02320
      RETURN                                                                STA02330
      END                                                                   STA02340
```

APPENDIX B

IMSL/NON-IMSL ROUTINES UTILIZED

IMSL ROUTINES

```
C     IMSL ROUTINE NAME   - GGUBFS                                          GGU00110
C                                                                           GGU00120
C-----------------------------------------------------------------------    GGU00130
C                                                                           GGU00140
C     COMPUTER            - IBM/SINGLE                                      GGU00150
C                                                                           GGU00160
C     LATEST REVISION     - JUNE 1, 1980                                    GGU00170
C                                                                           GGU00180
C     PURPOSE             - BASIC UNIFORM (0,1) RANDOM NUMBER GENERATOR -   GGU00190
C                           FUNCTION FORM OF GGUBS                          GGU00200
C                                                                           GGU00210
C     USAGE               - FUNCTION GGUBFS (DSEED)                         GGU00220
C                                                                           GGU00230
C     ARGUMENTS    GGUBFS - RESULTANT DEVIATE.                              GGU00240
C                  DSEED  - INPUT/OUTPUT DOUBLE PRECISION VARIABLE          GGU00250
C                           ASSIGNED AN INTEGER VALUE IN THE                GGU00260
C                           EXCLUSIVE RANGE (1.D0, 2147483647.D0).          GGU00270
C                           DSEED IS REPLACED BY A NEW VALUE TO BE          GGU00280
C                           USED IN A SUBSEQUENT CALL.                      GGU00290
C                                                                           GGU00300
C     PRECISION/HARDWARE  - SINGLE/ALL                                      GGU00310
C                                                                           GGU00320
C     REQD. IMSL ROUTINES - NONE REQUIRED                                   GGU00330
C                                                                           GGU00340
C     NOTATION            - INFORMATION ON SPECIAL NOTATION AND             GGU00350
C                           CONVENTIONS IS AVAILABLE IN THE MANUAL          GGU00360
C                           INTRODUCTION OR THROUGH IMSL ROUTINE UHELP      GGU00370
C                                                                           GGU00380
C     COPYRIGHT           - 1978 BY IMSL, INC. ALL RIGHTS RESERVED.         GGU00390
C                                                                           GGU00400
C     WARRANTY            - IMSL WARRANTS ONLY THAT IMSL TESTING HAS BEEN   GGU00410
C                           APPLIED TO THIS CODE. NO OTHER WARRANTY,        GGU00420
C                           EXPRESSED OR IMPLIED, IS APPLICABLE.            GGU00430
C                                                                           GGU00440
C-----------------------------------------------------------------------    GGU00450
C                                                                           GGU00460
      REAL FUNCTION GGUBFS (DSEED)                                          GGU00470
C                                                                           GGU00480
      DOUBLE PRECISION   DSEED                                              GGU00490
C                                SPECIFICATIONS FOR ARGUMENTS               GGU00500
C                                SPECIFICATIONS FOR LOCAL VARIABLES
```

```
      DOUBLE PRECISION   D2P31M,D2P31                                        GGU00510
C                                 D2P31M=(2**31) - 1                       GGU00520
C                                 D2P31 =(2**31) (OR AN ADJUSTED VALUE)    GGU00530
      DATA               D2P31M/2147483647.D0/                             GGU00540
      DATA               D2P31 /2147483648.D0/                             GGU00550
C                                 FIRST EXECUTABLE STATEMENT               GGU00560
      DSEED = DMOD(16807.D0*DSEED,D2P31M)                                  GGU00570
      GGUBFS = DSEED / D2P31                                               GGU00580
      RETURN                                                               GGU00590
      END                                                                  GGU00600
```

```
C   IMSL ROUTINE NAME   - FFTRC                                          FFT00110
C                                                                        FFT00120
C-----------------------------------------------------------------------FFT00130
C                                                                        FFT00140
C   COMPUTER            - IBM/DOUBLE                                     FFT00150
C                                                                        FFT00160
C   LATEST REVISION     - JANUARY 1, 1978                                FFT00170
C                                                                        FFT00180
C   PURPOSE             - COMPUTE THE FAST FOURIER TRANSFORM OF A        FFT00190
C                           REAL VALUED SEQUENCE                         FFT00200
C                                                                        FFT00210
C   USAGE               - CALL FFTRC (A,N,X,IWK,WK)                      FFT00220
C                                                                        FFT00230
C   ARGUMENTS     A     - INPUT REAL VECTOR OF LENGTH N WHICH            FFT00240
C                           CONTAINS THE DATA TO BE TRANSFORMED.         FFT00250
C                 N     - INPUT NUMBER OF DATA POINTS TO BE TRANSFORMED. FFT00260
C                           N MUST BE A POSITIVE EVEN INTEGER.           FFT00270
C                 X     - OUTPUT COMPLEX VECTOR OF LENGTH N/2+1          FFT00280
C                           CONTAINING THE FIRST N/2+1 COEFFICIENTS OF   FFT00290
C                           THE FOURIER TRANSFORM. THE REMAINING         FFT00300
C                           COEFFICIENTS MAY BE DETERMINED BY            FFT00310
C                             X(N+2-I) = CONJG(X(I)), FOR I=2,...,N/2.   FFT00320
C                 IWK   - INTEGER WORK VECTOR.                           FFT00330
C                           IF N IS A POWER OF 2, THEN IWK SHOULD BE OF  FFT00340
C                           LENGTH M WHERE N=2**M.                       FFT00350
C   PRECISION/HARDWARE  - SINGLE AND DOUBLE/H32                          FFT00430
C                       - SINGLE/H36,H48,H60                             FFT00440
C                                                                        FFT00450
C   REQD. IMSL ROUTINES - FFTCC,FFT2C                                    FFT00460
C                                                                        FFT00470
C   NOTATION            - INFORMATION ON SPECIAL NOTATION AND            FFT00480
C                           CONVENTIONS IS AVAILABLE IN THE MANUAL       FFT00490
C                           INTRODUCTION OR THROUGH IMSL ROUTINE UHELP    FFT00500
C                                                                        FFT00510
C   REMARKS  1. FFTRC COMPUTES THE FOURIER TRANSFORM, X, ACCORDING       FFT00520
C               TO THE FOLLOWING FORMULA;                                FFT00530
C                                                                        FFT00540
C                 X(K+1) = SUM FROM J = 0 TO N-1 OF                      FFT00550
C                          A(J+1)*CEXP((0.0,(2.0*PI*J*K)/N))             FFT00560
C                 FOR K=0,1,...,N/2 AND PI=3.1415...                     FFT00570
C                                                                        FFT00580
C               THE USER CAN COMPUTE THE REMAINING X VALUES BY           FFT00590
C               PERFORMING THE FOLLOWING STEPS;                          FFT00600
C                                                                        FFT00610
C                    ND2 = N/2                                           FFT00620
C                    DO 10 I=2,ND2                                       FFT00630
C                       X(N+2-I) = CONJG(X(I))                           FFT00640
C                 10 CONTINUE                                            FFT00650
C            2. FFTRC CAN BE USED TO COMPUTE                             FFT00660
C                                                                        FFT00670
```

```
C                  X(K+1) = (1/N)*SUM FROM J = 0 TO N-1 OF                      FFT00680
C                           A(J+1)*CEXP((0.0,(-2.0*PI*J*K)/N))                  FFT00690
C                  FOR K=0,1,...,N/2 AND PI=3.1415...                           FFT00700
C                                                                               FFT00710
C                BY PERFORMING THE FOLLOWING STEPS;                             FFT00720
C                                                                               FFT00730
C                     CALL FFTRC (A,N,X,IWK,WK)                                 FFT00740
C                     ND2P1 = N/2+1                                             FFT00750
C                     DO 10 I=1,ND2P1                                           FFT00760
C                        X(I) = CONJG(X(I))/N                                   FFT00770
C                  10 CONTINUE                                                  FFT00780
C                                                                               FFT00790
C  COPYRIGHT            - 1978 BY IMSL, INC. ALL RIGHTS RESERVED.               FFT00800
C                                                                               FFT00810
C  WARRANTY             - IMSL WARRANTS ONLY THAT IMSL TESTING HAS BEEN         FFT00820
C                            APPLIED TO THIS CODE. NO OTHER WARRANTY,           FFT00830
C                            EXPRESSED OR IMPLIED, IS APPLICABLE.               FFT00840
C                                                                               FFT00850
C  MODIFIED 5 AUGUST 1985 TO ALLOW THE GENERATION ONLY OF FFT OF VECTOR         FFT00860
C  OF LENGTH EQUAL TO AN EVEN POWER OF 2 AND THE ENTIRE FFT.                    FFT00870
C-------------------------------------------------------------------------------FFT00880
C                                                                               FFT00890
      SUBROUTINE FFTRC  (A,N,X,IWK)                                             FFT00900
C                                     SPECIFICATIONS FOR ARGUMENTS              FFT00910
      INTEGER             N,IWK(1)                                              FFT00920
      DOUBLE PRECISION    A(N)                                                  FFT00930
      COMPLEX*16          X(1)                                                  FFT00940
C                                     SPECIFICATIONS FOR LOCAL VARIABLES        FFT00950
      INTEGER             ND2P1,ND2,I,MTWO,M,IMAX,ND4,NP2,K,NMK,J               FFT00960
      DOUBLE PRECISION    RPI,ZERO,ONE,HALF,THETA,TP,G(2),B(2),Z(2),AI,         FFT00970
     1                    AR                                                    FFT00980
      COMPLEX*16          XIMAG,ALPH,BETA,GAM,S1,ZD                             FFT00990
      EQUIVALENCE         (GAM,G(1)),(ALPH,B(1)),(Z(1),AR),(Z(2),AI),           FFT01000
     1                    (ZD,Z(1))                                             FFT01010
      DATA                ZERO/0.0D0/,HALF/0.5D0/,ONE/1.0D0/,IMAX/24/           FFT01020
      DATA                RPI/3.14159265358979 3D0/                             FFT01030
C                                     FIRST EXECUTABLE STATEMENT                FFT01040
      IF (N .NE. 2) GO TO 5                                                     FFT01050
C                                     N EQUAL TO 2                              FFT01060
      ZD = DCMPLX(A(1),A(2))                                                    FFT01070
      THETA = AR                                                                FFT01080
      TP = AI                                                                   FFT01090
      X(2) = DCMPLX(THETA-TP,ZERO)                                              FFT01100
      X(1) = DCMPLX(THETA+TP,ZERO)                                              FFT01110
      GO TO 9005                                                                FFT01120
    5 CONTINUE                                                                  FFT01130
C                                     N GREATER THAN 2                          FFT01140
      ND2 = N/2                                                                 FFT01150
      ND2P1 = ND2+1                                                             FFT01160
C                                     MOVE A TO X                               FFT01170
```

```
      J = 1                                                              FFT01180
      DO 6 I=1,ND2                                                       FFT01190
         X(I) = DCMPLX(A(J),A(J+1))                                      FFT01200
         J = J+2                                                         FFT01210
    6 CONTINUE                                                           FFT01220
C                                     COMPUTE THE CENTER COEFFICIENT     FFT01230
      GAM = DCMPLX(ZERO,ZERO)                                            FFT01240
      DO 10 I=1,ND2                                                      FFT01250
         GAM = GAM + X(I)                                                FFT01260
   10 CONTINUE                                                           FFT01270
      TP = G(1)-G(2)                                                     FFT01280
      GAM = DCMPLX(TP,ZERO)                                              FFT01290
C                                     DETERMINE THE SMALLEST M SUCH THAT FFT01300
C                                     N IS LESS THAN OR EQUAL TO 2**M    FFT01310
      MTWO = 2                                                           FFT01320
      M = 1                                                              FFT01330
      DO 15 I=1,IMAX                                                     FFT01340
         IF (ND2 .LE. MTWO) GO TO 20                                     FFT01350
         MTWO = MTWO+MTWO                                                FFT01360
         M = M+1                                                         FFT01370
   15 CONTINUE                                                           FFT01380
   20 IF (ND2 .EQ. MTWO) GO TO 25                                        FFT01390
C                                     N IS NOT A POWER OF TWO, CALL FFTCC FFT01400
C                                     N IS A POWER OF TWO, CALL FFT2C    FFT01410
   25 CALL FFT2C (X,M,IWK)                                               FFT01420
   30 ALPH = X(1)                                                        FFT01430
      X(1) = B(1) + B(2)                                                 FFT01440
      ND4 = (ND2+1)/2                                                    FFT01450
      IF (ND4 .LT. 2) GO TO 40                                           FFT01460
      NP2 = ND2 + 2                                                      FFT01470
      THETA = RPI/ND2                                                    FFT01480
      TP = THETA                                                         FFT01490
      XIMAG = DCMPLX(ZERO,ONE)                                           FFT01500
C                                     DECOMPOSE THE COMPLEX VECTOR X     FFT01510
C                                     INTO THE COMPONENTS OF THE TRANSFORM FFT01520
C                                     OF THE INPUT DATA.                 FFT01530
      DO 35 K = 2,ND4                                                    FFT01540
         NMK = NP2 - K                                                   FFT01550
         S1 = DCONJG(X(NMK))                                             FFT01560
         ALPH = X(K) + S1                                                FFT01570
         BETA = XIMAG*(S1-X(K))                                          FFT01580
         S1 = DCMPLX(DCOS(THETA),DSIN(THETA))                            FFT01590
         X(K) = (ALPH+BETA*S1)*HALF                                      FFT01600
         X(NMK) = DCONJG(ALPH-BETA*S1)*HALF                              FFT01610
         THETA = THETA + TP                                              FFT01620
   35 CONTINUE                                                           FFT01630
   40 CONTINUE                                                           FFT01640
      X(ND2P1) = GAM                                                     FFT01650
C     ND2=N/2                                                            FFT01660
C     DO 90 I=2,ND2                                                      FFT01670
```

```
C           X(N+2-I)=DCONJG(X(I))                                   FFT01680
C90       CONTINUE                                                  FFT01690
 9005   RETURN                                                      FFT01700
        END                                                         FFT01710
```

```
C    IMSL ROUTINE NAME   - FFT2C                                          FFT00110
C                                                                         FFT00120
C-----------------------------------------------------------------------FFT00130
C                                                                         FFT00140
C    COMPUTER            - IBM/DOUBLE                                     FFT00150
C                                                                         FFT00160
C    LATEST REVISION     - NOVEMBER 1, 1984                               FFT00170
C                                                                         FFT00180
C    PURPOSE             - COMPUTE THE FAST FOURIER TRANSFORM OF A        FFT00190
C                            COMPLEX VALUED SEQUENCE OF LENGTH EQUAL TO   FFT00200
C                            A POWER OF TWO                               FFT00210
C                                                                         FFT00220
C    USAGE               - CALL FFT2C (A,M,IWK)                           FFT00230
C                                                                         FFT00240
C    ARGUMENTS     A     - COMPLEX VECTOR OF LENGTH N, WHERE N=2**M.      FFT00250
C                            ON INPUT A CONTAINS THE COMPLEX VALUED       FFT00260
C                            SEQUENCE TO BE TRANSFORMED.                  FFT00270
C                            ON OUTPUT A IS REPLACED BY THE               FFT00280
C                            FOURIER TRANSFORM.                           FFT00290
C                  M     - INPUT EXPONENT TO WHICH 2 IS RAISED TO         FFT00300
C                            PRODUCE THE NUMBER OF DATA POINTS, N         FFT00310
C                            (I.E. N = 2**M).                             FFT00320
C                  IWK   - WORK VECTOR OF LENGTH M+1.                     FFT00330
C                                                                         FFT00340
C    PRECISION/HARDWARE  - SINGLE AND DOUBLE/H32                          FFT00350
C                        - SINGLE/H36,H48,H60                             FFT00360
C                                                                         FFT00370
C    REQD. IMSL ROUTINES - NONE REQUIRED                                  FFT00380
C                                                                         FFT00390
C    NOTATION            - INFORMATION ON SPECIAL NOTATION AND            FFT00400
C                            CONVENTIONS IS AVAILABLE IN THE MANUAL       FFT00410
C                            INTRODUCTION OR THROUGH IMSL ROUTINE UHELP   FFT00420
C                                                                         FFT00430
C    REMARKS  1.  FFT2C COMPUTES THE FOURIER TRANSFORM, X, ACCORDING      FFT00440
C                 TO THE FOLLOWING FORMULA;                               FFT00450
C                                                                         FFT00460
C                   X(K+1) = SUM FROM J = 0 TO N-1 OF                     FFT00470
C                            A(J+1)*CEXP((0.0,(2.0*PI*J*K)/N))            FFT00480
C                   FOR K=0,1,...,N-1 AND PI=3.1415...                    FFT00490
C                                                                         FFT00500
C                 NOTE THAT X OVERWRITES A ON OUTPUT.                     FFT00510
C             2.  FFT2C CAN BE USED TO COMPUTE THE INVERSE FOURIER        FFT00520
C                 TRANSFORM, X, ACCORDING TO THE FOLLOWING FORMULA;       FFT00530
C                                                                         FFT00540
C                   X(K+1) = (1/N)*SUM FROM J = 0 TO N-1 OF               FFT00550
C                            A(J+1)*CEXP((0.0,(-2.0*PI*J*K)/N))           FFT00560
C                   FOR K=0,1,...,N-1 AND PI=3.1415...                    FFT00570
C                                                                         FFT00580
C                 BY PERFORMING THE FOLLOWING STEPS;                      FFT00590
C                                                                         FFT00600
```

```
C                     DO 10 I=1,N                                          FFT00610
C                        A(I) = CONJG(A(I))                                FFT00620
C                  10 CONTINUE                                            FFT00630
C                     CALL FFT2C (A,M,IWK)                                 FFT00640
C                     DO 20 I=1,N                                          FFT00650
C                        A(I) = CONJG(A(I))/N                              FFT00660
C                  20 CONTINUE                                            FFT00670
C                                                                         FFT00680
C     COPYRIGHT            - 1978 BY IMSL, INC. ALL RIGHTS RESERVED.      FFT00690
C                                                                         FFT00700
C     WARRANTY             - IMSL WARRANTS ONLY THAT IMSL TESTING HAS BEEN FFT00710
C                              APPLIED TO THIS CODE. NO OTHER WARRANTY,   FFT00720
C                              EXPRESSED OR IMPLIED, IS APPLICABLE.       FFT00730
C                                                                         FFT00740
C-------------------------------------------------------------------------FFT00750
C                                                                         FFT00760
      SUBROUTINE  FFT2C (A,M,IWK)                                         FFT00770
C                                SPECIFICATIONS FOR ARGUMENTS             FFT00780
      INTEGER                  M,IWK(1)                                   FFT00790
      COMPLEX*16               A(1)                                       FFT00800
C                                SPECIFICATIONS FOR LOCAL VARIABLES       FFT00810
      INTEGER                  I,ISP,J,JJ,JSP,K,K0,K1,K2,K3,KB,KN,MK,MM,MP,N,FFT00820
     1                         N4,N8,N2,LM,NN,JK                          FFT00830
      DOUBLE PRECISION         RAD,C1,C2,C3,S1,S2,S3,CK,SK,SQ,A0,A1,A2,A3,FFT00840
     1                         B0,B1,B2,B3,TWOPI,TEMP,                    FFT00850
     2                         ZERO,ONE,Z0(2),Z1(2),Z2(2),Z3(2)           FFT00860
      COMPLEX*16               ZA0,ZA1,ZA2,ZA3,AK2                        FFT00870
      EQUIVALENCE              (ZA0,Z0(1)),(ZA1,Z1(1)),(ZA2,Z2(1)),       FFT00880
     1                         (ZA3,Z3(1)),(A0,Z0(1)),(B0,Z0(2)),(A1,Z1(1)),FFT00890
     2                         (B1,Z1(2)),(A2,Z2(1)),(B2,Z2(2)),(A3,Z3(1)),FFT00900
     3                         (B3,Z3(2))                                 FFT00910
      DATA                     SQ/.7071067811865475D0/,                   FFT00920
     1                         SK/.3826834323650898D0/,                   FFT00930
     2                         CK/.9238795325112868D0/,                   FFT00940
     3                         TWOPI/6.2831853071795860/                  FFT00950
      DATA                     ZERO/0.0D0/,ONE/1.0D0/                     FFT00960
C                                SQ=SQRT2/2,SK=SIN(PI/8),CK=COS(PI/8)     FFT00970
C                                TWOPI=2*PI                               FFT00980
C                                FIRST EXECUTABLE STATEMENT               FFT00990
      MP = M+1                                                            FFT01000
      N = 2**M                                                            FFT01010
      IWK(1) = 1                                                          FFT01020
      MM = (M/2)*2                                                        FFT01030
      KN = N+1                                                            FFT01040
C                                INITIALIZE WORK VECTOR                   FFT01050
      DO 5  I=2,MP                                                        FFT01060
         IWK(I) = IWK(I-1)+IWK(I-1)                                       FFT01070
    5 CONTINUE                                                            FFT01080
      RAD = TWOPI/N                                                       FFT01090
      MK = M - 4                                                          FFT01100
```

```
      KB = 1                                                          FFT01110
      IF (MM .EQ. M) GO TO 15                                         FFT01120
      K2 = KN                                                         FFT01130
      K0 = IWK(MM+1) + KB                                             FFT01140
   10 K2 = K2 - 1                                                     FFT01150
      K0 = K0 - 1                                                     FFT01160
      AK2 = A(K2)                                                     FFT01170
      A(K2) = A(K0) - AK2                                             FFT01180
      A(K0) = A(K0) + AK2                                             FFT01190
      IF (K0 .GT. KB) GO TO 10                                        FFT01200
   15 C1 = ONE                                                        FFT01210
      S1 = ZERO                                                       FFT01220
      JJ = 0                                                          FFT01230
      K = MM - 1                                                      FFT01240
      J = 4                                                           FFT01250
      IF (K .GE. 1) GO TO 30                                          FFT01260
      GO TO 70                                                        FFT01270
   20 IF (IWK(J) .GT. JJ) GO TO 25                                    FFT01280
      JJ = JJ - IWK(J)                                                FFT01290
      J = J-1                                                         FFT01300
      IF (IWK(J) .GT. JJ) GO TO 25                                    FFT01310
      JJ = JJ - IWK(J)                                                FFT01320
      J = J - 1                                                       FFT01330
      K = K + 2                                                       FFT01340
      GO TO 20                                                        FFT01350
   25 JJ = IWK(J) + JJ                                                FFT01360
      J = 4                                                           FFT01370
   30 ISP = IWK(K)                                                    FFT01380
      IF (JJ .EQ. 0) GO TO 40                                         FFT01390
C                                  RESET TRIGONOMETRIC PARAMETERS     FFT01400
      C2 = JJ * ISP * RAD                                             FFT01410
      C1 = DCOS(C2)                                                   FFT01420
      S1 = DSIN(C2)                                                   FFT01430
   35 C2 = C1 * C1 - S1 * S1                                          FFT01440
      S2 = C1 * (S1 + S1)                                             FFT01450
      C3 = C2 * C1 - S2 * S1                                          FFT01460
      S3 = C2 * S1 + S2 * C1                                          FFT01470
   40 JSP = ISP + KB                                                  FFT01480
C                                  DETERMINE FOURIER COEFFICIENTS     FFT01490
C                                    IN GROUPS OF 4                   FFT01500
      DO 50 I=1,ISP                                                   FFT01510
      K0 = JSP - I                                                    FFT01520
      K1 = K0 + ISP                                                   FFT01530
      K2 = K1 + ISP                                                   FFT01540
      K3 = K2 + ISP                                                   FFT01550
      ZA0 = A(K0)                                                     FFT01560
      ZA1 = A(K1)                                                     FFT01570
      ZA2 = A(K2)                                                     FFT01580
      ZA3 = A(K3)                                                     FFT01590
      IF (S1 .EQ. ZERO) GO TO 45                                      FFT01600
```

```
         TEMP  = A1                                                     FFT01610
         A1 =  A1 * C1 - B1 * S1                                        FFT01620
         B1 =  TEMP * S1 + B1 * C1                                      FFT01630
         TEMP  = A2                                                     FFT01640
         A2 =  A2 * C2 - B2 * S2                                        FFT01650
         B2 =  TEMP * S2 + B2 * C2                                      FFT01660
         TEMP  = A3                                                     FFT01670
         A3 =  A3 * C3 - B3 * S3                                        FFT01680
         B3 =  TEMP * S3 + B3 * C3                                      FFT01690
   45    TEMP  = A0 + A2                                                FFT01700
         A2 =  A0 - A2                                                  FFT01710
         A0 =  TEMP                                                     FFT01720
         TEMP  = A1 + A3                                                FFT01730
         A3 =  A1 - A3                                                  FFT01740
         A1 =  TEMP                                                     FFT01750
         TEMP  = B0 + B2                                                FFT01760
         B2 =  B0 - B2                                                  FFT01770
         B0 =  TEMP                                                     FFT01780
         TEMP  = B1 + B3                                                FFT01790
         B3 =  B1 - B3                                                  FFT01800
         B1 =  TEMP                                                     FFT01810
         A(K0)  = DCMPLX(A0+A1,B0+B1)                                   FFT01820
         A(K1)  = DCMPLX(A0-A1,B0-B1)                                   FFT01830
         A(K2)  = DCMPLX(A2-B3,B2+A3)                                   FFT01840
         A(K3)  = DCMPLX(A2+B3,B2-A3)                                   FFT01850
   50 CONTINUE                                                          FFT01860
      IF (K .LE. 1) GO TO 55                                            FFT01870
      K = K - 2                                                         FFT01880
      GO TO 30                                                          FFT01890
   55 KB = K3 + ISP                                                     FFT01900
C                                       CHECK FOR COMPLETION OF FINAL   FFT01910
C                                         ITERATION                     FFT01920
      IF (KN .LE. KB) GO TO 70                                          FFT01930
      IF (J .NE. 1) GO TO 60                                            FFT01940
      K = 3                                                             FFT01950
      J = MK                                                            FFT01960
      GO TO 20                                                          FFT01970
   60 J = J - 1                                                         FFT01980
      C2 = C1                                                           FFT01990
      IF (J .NE. 2) GO TO 65                                            FFT02000
      C1 = C1 * CK + S1 * SK                                            FFT02010
      S1 = S1 * CK - C2 * SK                                            FFT02020
      GO TO 35                                                          FFT02030
   65 C1 = (C1 - S1) * SQ                                               FFT02040
      S1 = (C2 + S1) * SQ                                               FFT02050
      GO TO 35                                                          FFT02060
   70 CONTINUE                                                          FFT02070
C                                       PERMUTE THE COMPLEX VECTOR IN   FFT02080
C                                         REVERSE BINARY ORDER TO NORMAL FFT02090
C                                         ORDER                         FFT02100
```

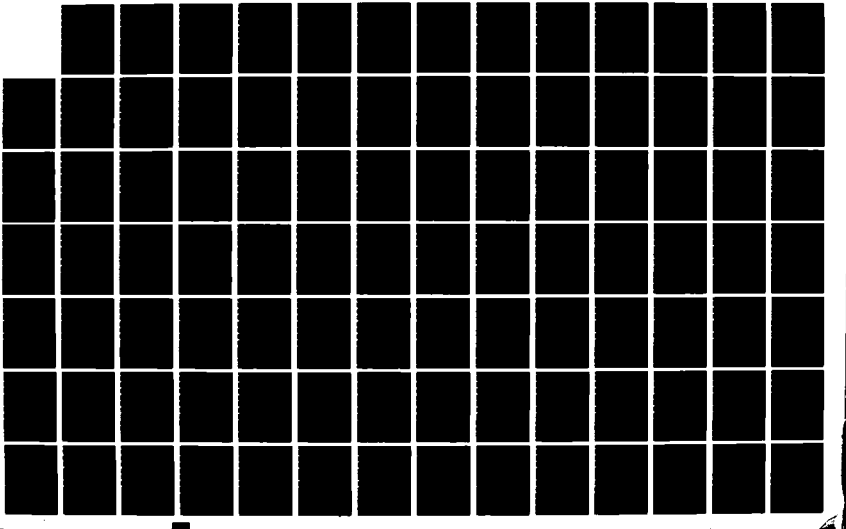
```
      IF(M .LE. 1) GO TO 9005                                          FFT02110
      MP = M+1                                                         FFT02120
      JJ = 1                                                           FFT02130
C                                     INITIALIZE WORK VECTOR           FFT02140
      IWK(1) = 1                                                       FFT02150
      DO 75  I = 2,MP                                                  FFT02160
         IWK(I) = IWK(I-1) * 2                                         FFT02170
   75 CONTINUE                                                         FFT02180
      N4 = IWK(MP-2)                                                   FFT02190
      IF (M .GT. 2) N8 = IWK(MP-3)                                     FFT02200
      N2 = IWK(MP-1)                                                   FFT02210
      LM = N2                                                          FFT02220
      NN = IWK(MP)+1                                                   FFT02230
      MP = MP-4                                                        FFT02240
C                                     DETERMINE INDICES AND SWITCH A   FFT02250
      J = 2                                                            FFT02260
   80 JK = JJ + N2                                                     FFT02270
      AK2 = A(J)                                                       FFT02280
      A(J) = A(JK)                                                     FFT02290
      A(JK) = AK2                                                      FFT02300
      J = J+1                                                          FFT02310
      IF (JJ .GT. N4) GO TO 85                                         FFT02320
      JJ = JJ + N4                                                     FFT02330
      GO TO 105                                                        FFT02340
   85 JJ = JJ - N4                                                     FFT02350
      IF (JJ .GT. N8) GO TO 90                                         FFT02360
      JJ = JJ + N8                                                     FFT02370
      GO TO 105                                                        FFT02380
   90 JJ = JJ - N8                                                     FFT02390
      K = MP                                                           FFT02400
   95 IF (IWK(K) .GE. JJ) GO TO 100                                    FFT02410
      JJ = JJ - IWK(K)                                                 FFT02420
      K = K - 1                                                        FFT02430
      GO TO 95                                                         FFT02440
  100 JJ = IWK(K) + JJ                                                 FFT02450
  105 IF (JJ .LE. J) GO TO 110                                         FFT02460
      K = NN - J                                                       FFT02470
      JK = NN - JJ                                                     FFT02480
      AK2 = A(J)                                                       FFT02490
      A(J) = A(JJ)                                                     FFT02500
      A(JJ) = AK2                                                      FFT02510
      AK2 = A(K)                                                       FFT02520
      A(K) = A(JK)                                                     FFT02530
      A(JK) = AK2                                                      FFT02540
  110 J = J + 1                                                        FFT02550
C                                     CYCLE REPEATED UNTIL LIMITING NUMBER FFT02560
C                                       OF CHANGES IS ACHIEVED         FFT02570
      IF (J .LE. LM) GO TO 80                                          FFT02580
C                                                                      FFT02590
 9005 RETURN                                                           FFT02600
```

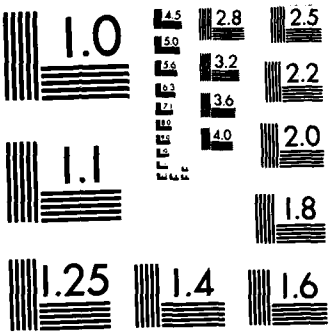
END

FFT02610

AD-A160 823 COMPUTER SIMULATION OF DIGITAL SIGNAL MODULATION TECHNIQUES IN SATELLITE COMMUNICATIONS(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA C D CARLSON SEP 85 3/4

UNCLASSIFIED F/G 9/2 NL





```
C.....                                                                  UTP00110
C                                                                       UTP00120
C     SUBROUTINE UTPLOT                                                 UTP00130
C                                                                       UTP00140
C     PURPOSE                                                           UTP00150
C                                                                       UTP00160
C         PRINTS GRAPHS ON THE STANDARD OUTPUT PRINTER                  UTP00170
C                                                                       UTP00180
C     FEATURES                                                          UTP00190
C                                                                       UTP00200
C     1) FULL CONTROL OVER SCALING                                      UTP00210
C     2) ABILITY TO PLOT SINGLE OR DOUBLE PRECISION VECTORS             UTP00220
C                                                                       UTP00230
C     CALLING SEQUENCE                                                  UTP00240
C                                                                       UTP00250
C     CALL UTPLOT(X,Y,N,RANGE,K,MODCUR)                                 UTP00260
C                                                                       UTP00270
C     DESCRIPTION OF ARGUMENTS                                          UTP00280
C                                                                       UTP00290
C     X       VECTOR OF ABSCISSAE                                       UTP00300
C                                                                       UTP00310
C     Y       VECTOR OF ASSOCIATED ORDINATES                            UTP00320
C                                                                       UTP00330
C     N       NUMBER OF (X,Y) PAIRS                                     UTP00340
C                                                                       UTP00350
C     RANGE   4 WORD SCALING VECTOR WHERE                               UTP00360
C             RANGE(1) = MAXIMUM X TO BE PLOTTED                        UTP00370
C             RANGE(2) = MINIMUM X TO BE PLOTTED                        UTP00380
C             RANGE(3) = MAXIMUM Y TO BE PLOTTED                        UTP00390
C             RANGE(4) = MINIMUM Y TO BE PLOTTED                        UTP00400
C                                                                       UTP00410
C             ALL (X,Y) POINTS OUTSIDE THE ABOVE RANGE WILL BE PLOTTED  UTP00420
C             IN THE BORDER OF THE GRAPH.                               UTP00430
C                                                                       UTP00440
C     K       EVERY KTH ELEMENT OF X & Y WILL BE PLOTTED, E.G.,         UTP00450
C             FOR REAL*4 DATA (SINGLE PRECISION) K=1                    UTP00460
C             FOR REAL*8 DATA (DOUBLE PRECISION) K=2.                   UTP00470
C                                                                       UTP00480
C     MODCUR  CONTROLS THE NUMBER OF CURVES ON ONE GRAPH                UTP00490
C             =0 THERE IS ONLY 1 CURVE ON THIS GRAPH                    UTP00500
C             =1 THIS IS THE FIRST OF TWO OR MORE CURVES ON THIS GRAPH  UTP00510
C             =2 THIS IS AN INTERMEDIATE CURVE ON THIS GRAPH            UTP00520
C             =3 THIS IS THE LAST CURVE ON THIS GRAPH                   UTP00530
C                                                                       UTP00540
C     SCALING                                                           UTP00550
C                                                                       UTP00560
C             SCALING IS PERFORMED ONLY ON THE FIRST SET OF POINTS (WHEN UTP00570
```

```
C       MODCUR = 0 OR 1.) ARRAY RANGE IS USED TO SET UP THE SCALE FAC-  UTP00580
C       TORS AND NEED ONLY BE DEFINED FOR THE FIRST CALL TO UTPLOT.     UTP00590
C                                                                       UTP00600
C   GRID LABELLING                                                      UTP00610
C                                                                       UTP00620
C       THE DATA TO BE GRAPHED WILL BE FIT INTO AN 80 COLUMN BY         UTP00630
C   BY 60 ROW GRID. THE GRID WILL BE LABELLED THUSLY:                   UTP00640
C                                                                       UTP00650
C   IN THE X DIRECTION (COLUMN-WISE), THERE WILL BE 5 VALUES:           UTP00660
C   THE MAXIMUM, THE MINIMUM, AND 3 INTERMEDIATE AT INCREMENTS OF       UTP00670
C   (RANGE(2)-RANGE(1))/4. FROM THE MINIMUM.                            UTP00680
C                                                                       UTP00690
C   IN THE Y DIRECTION (ROW-WISE) THERE WILL BE 7 VALUES: THE MAXI-     UTP00700
C   MUM, THE MINIMUM, AND 5 INTERMEDIATE AT INCREMENTS OF               UTP00710
C   (RANGE(4)-RANGE(3))/6. FROM THE MINIMUM.                            UTP00720
C                                                                       UTP00730
C   IF THE LABELS HAVE A VALUE BETWEEN 1. AND 10**8, THEY WILL BE       UTP00740
C   PRINTED IN AN F11.2 FORMAT, OTHERWISE THEY WILL BE PRINTED IN A     UTP00750
C   1PE10.3 FORMAT.                                                     UTP00760
C                                                                       UTP00770
C   PLOTTING                                                            UTP00780
C                                                                       UTP00790
C       FOUR CHARACTERS ARE USED FOR PLOTTING CURVES, " ","+","*",      UTP00800
C   AND "X". WHEN MORE THAN 4 CURVES ARE PLOTTED THE CHARACTERS         UTP00810
C   ARE USED REPEATEDLY.  IF A NEW CURVE IS TO BE PLACED IN THE         UTP00820
C   PLOTTING GRID WHERE AN OLD CURVE EXISTS, THE NEW CURVE REPLACES     UTP00830
C   THE OLD ONE. THUS, IF 3 IDENTICAL CURVES ARE PLOTTED, THEY WILL     UTP00840
C   APPEAR AS ONE CURVE COMPOSED OF "*"'S.                              UTP00850
C                                                                       UTP00860
C   MESSAGES                                                            UTP00870
C                                                                       UTP00880
C       UNDER CERTAIN CIRCUMSTANCES, A PLOT WILL NOT BE OUTPUT AND      UTP00890
C   ONE OF THE FOLLOWING MESSAGES WILL BE PRINTED ON THE STANDARD       UTP00900
C   OUTPUT IN PLACE OF THE PLOT.                                        UTP00910
C                                                                       UTP00920
C   "ALL Y-VALUES=0. CANNOT SETUP PLOT GRID. CHECK MAX & MIN Y          UTP00930
C    WHEN MODCUR=0 OR 1."                                               UTP00940
C                                                                       UTP00950
C   "ALL X VALUES=0. CANNOT SETUP PLOT GRID. CHECK MAX AND MIN X        UTP00960
C    WHEN MODCUR=0 OR 1."                                               UTP00970
C                                                                       UTP00980
C   "GRID NOT SETUP WHEN MODCUR LAST 0 OR 1. NO PLOT UNTIL GRID         UTP00990
C    PROPERLY SETUP."                                                   UTP01000
C                                                                       UTP01010
C   NOTE                                                                UTP01020
C                                                                       UTP01030
C       THE USER IS EXPECTED TO PROVIDE THE NECESSARY CARRIAGE          UTP01040
C   CONTROLS TO PLACE THE GRAPH PROPERLY ON THE PAGE.  BEFORE           UTP01050
C   CALLING UTPLOT THE USER SHOULD ISSUE A PRINT STATEMENT WHICH        UTP01060
C   EJECTS A PAGE SO THAT THE GRAPH WILL BE PLOTTED AT THE TOP          UTP01070
```

```
C        OF THE NEXT PAGE. A TITLE CAN BE PRINTED AT THE BOTTOM OF      UTP01080
CC       THE GRAPH BY ISSUING A PRINT STATEMENT RIGHT AFTER CALLING     UTP01090
CC       THE SUBROUTINE.                                                UTP01100
CC                                                                      UTP01110
CC     MAINTENANCE                                                      UTP01120
CC        ALTERED FOR WATFIV COMPILATION BY J. FOUST 5/81               UTP01130
CC                                                                      UTP01140
C.....................................................................UTP01150
C                                                                       UTP01160
      SUBROUTINE UTPLOT (X   ,Y   ,NDATA,RANGE,KKZ,MODCUR)              UTP01170
      DIMENSION GRID(61,81),XSCALE(5),YSCALE(7)                         UTP01180
      DIMENSION X   (1),Y   (1),RANGE(4)                                UTP01190
      INTEGER*2 GRID,BLANK,DOT,XCHAR(4)/1H.,1H*,1H*,1HX/                UTP01200
      DATA DOT,BLANK/Z4B40,Z4040/                                       UTP01210
      KDATA=NDATA*KKZ                                                   UTP01220
      IF(MODCUR.GT.1) GO TO 444                                         UTP01230
C                                                                       UTP01240
CC    GRID IS THE MATRIX USED TO PLOT THE POINTS                        UTP01250
C                                                                       UTP01260
      IERR=0                                                            UTP01270
      XMAX=RANGE(1)                                                     UTP01280
      XMIN=RANGE(2)                                                     UTP01290
      YMAX=RANGE(3)                                                     UTP01300
      YMIN=RANGE(4)                                                     UTP01310
C                                                                       UTP01320
CC    CHECKING X AND Y POINTS AND PLOTTING THOSE OUT OF RANGE           UTP01330
CC    AT THE MARGIN                                                     UTP01340
C                                                                       UTP01350
      DO 30 I=1,KDATA,KKZ                                               UTP01360
      IF(X(I).GT.XMAX.OR.X(I).LT.XMIN.OR.Y(I).GT.YMAX.OR.Y(I).LT.YMIN)  UTP01370
     1  IERR=IERR+1                                                     UTP01380
      IF(X(I).LE.XMAX) GO TO 205                                        UTP01390
      X   (I)=XMAX                                                     UTP01400
      GOTO 210                                                          UTP01410
  205 IF(X(I).GE.XMIN) GO TO 210                                        UTP01420
      X   (I)=XMIN                                                     UTP01430
  210 IF(Y(I).LE.YMAX) GO TO 215                                        UTP01440
      Y   (I)=YMAX                                                     UTP01450
      GOTO 30                                                           UTP01460
  215 IF(Y(I).GE.YMIN) GO TO 30                                         UTP01470
      Y   (I)=YMIN                                                     UTP01480
C                                                                       UTP01490
   30 CONTINUE                                                          UTP01500
C                                                                       UTP01510
CC    PLOTTING X AND Y AXIS , IF NECESSARY                              UTP01520
C                                                                       UTP01530
      JERR=0                                                            UTP01540
      XRANGE=XMAX-XMIN                                                  UTP01550
      YRANGE=YMAX-YMIN                                                  UTP01560
      IF (YRANGE.NE.0.)  GO TO 298                                      UTP01570
```

```
      IF(YMIN.EQ.0.) GO TO 889                                          UTP01580
      YMIN=0.                                                           UTP01590
      YRANGE=YMAX                                                       UTP01600
      GO TO 299                                                         UTP01610
  298 IF (XRANGE.NE.0.)  GO TO 299                                      UTP01620
      IF(XMIN.EQ.0.) GO TO 887                                          UTP01630
      XMIN=0.                                                           UTP01640
      XRANGE=XMAX                                                       UTP01650
C                                                                       UTP01660
CC    BLANKING OUT MATRIX-(GRID)                                        UTP01670
C                                                                       UTP01680
  299 DO 300 I=1,61                                                     UTP01690
      DO 301 JJ=1,81                                                    UTP01700
301   GRID(I,JJ)=BLANK                                                  UTP01710
  300 CONTINUE                                                          UTP01720
      IF(XMAX*XMIN.GE.0.) GO TO 222                                     UTP01730
      IYAXIS=80.*(-XMIN)/XRANGE+1.5                                     UTP01740
      DO 40 I=1,61                                                      UTP01750
   40 GRID(I,IYAXIS)=DOT                                                UTP01760
  222 IF(YMAX*YMIN.GE.0.) GO TO 333                                     UTP01770
      IXAXIS=60.*YMAX/YRANGE+1.5                                        UTP01780
      DO 60 I=1,81                                                      UTP01790
   60 GRID(IXAXIS,I)=DOT                                                UTP01800
C                                                                       UTP01810
CCC   COMPUTE PROPER SCALE NUMBERS                                      UTP01820
C                                                                       UTP01830
  333 XINCR=XRANGE/4.                                                   UTP01840
      YINCR=YRANGE/6.                                                   UTP01850
      XSCALE(1)=XMAX                                                    UTP01860
      XSCALE(5)=XMIN                                                    UTP01870
      DO 80 I=2,4                                                       UTP01880
      XSCALE(I)=XSCALE(I-1)-XINCR                                       UTP01890
      IF(ABS(XSCALE(I)).LT.1.E-4) XSCALE(I)=0.                          UTP01900
   80 CONTINUE                                                          UTP01910
      YSCALE(1)=YMAX                                                    UTP01920
      YSCALE(7)=YMIN                                                    UTP01930
      DO 81 I=2,6                                                       UTP01940
      YSCALE(I)=YSCALE(I-1)-YINCR                                       UTP01950
      IF(ABS(YSCALE(I)).LT.1.E-4) YSCALE(I)=0.                          UTP01960
   81 CONTINUE                                                          UTP01970
      DO 85 II=1,2                                                      UTP01980
      JJ=6-II                                                           UTP01990
      XT=XSCALE(JJ)                                                     UTP02000
      XSCALE(JJ)=XSCALE(II)                                             UTP02010
   85 XSCALE(II)=XT                                                     UTP02020
C                                                                       UTP02030
CCC                                                                     UTP02040
CCC   PLACING POINTS IN THEIR PROPER GRID POSITIONS                     UTP02050
C                                                                       UTP02060
  444 IF(MODCUR.LT.2) JSET=0                                            UTP02070
```

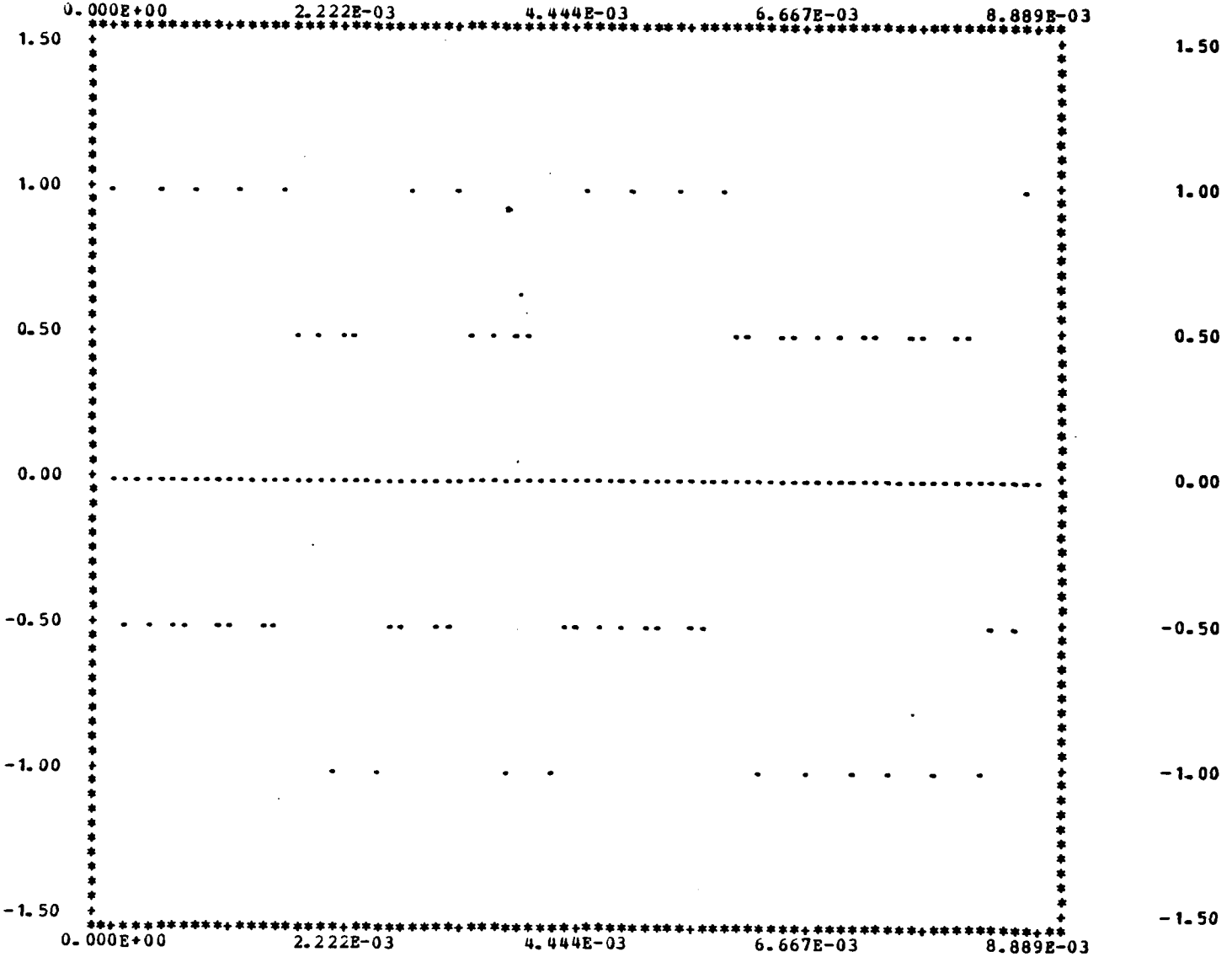
```
      IF(JERR.GT.0) GO TO 885                                                  UTP02080
      JSET=JSET+1                                                              UTP02090
      IF(JSET.GT.4) JSET=1                                                     UTP02100
      DO 700 I=1,KDATA,KKZ                                                     UTP02110
      IPTX=60.*(YMAX-Y(I))/YRANGE+1.5                                          UTP02120
      IPTY=80.*(X(I)-XMIN)/XRANGE+1.5                                          UTP02130
      IF(IPTX.GT.61.OR.IPTY.GT.81) GO TO 70                                    UTP02140
      IF(IPTX.LE.0.OR.IPTY.LE.0) GO TO 70                                      UTP02150
      GRID(IPTX,IPTY) = XCHAR(JSET)                                            UTP02160
      GO TO 700                                                                UTP02170
   70 IERR=IERR+1                                                              UTP02180
  700 CONTINUE                                                                 UTP02190
C     OUTPUT SECTION WITH GRAPH                                                UTP02200
C                                                                              UTP02210
      IF(MODCUR.EQ.1.OR.MODCUR.EQ.2) RETURN                                    UTP02220
      AXR=ABS(XRANGE)                                                          UTP02230
      AYR=ABS(YRANGE)                                                          UTP02240
      IF(AXR.LT.1.E+8.AND.AXR.GE..95) GO TO 400                                UTP02250
      WRITE(6,17) XSCALE                                                       UTP02260
   17 FORMAT(12X,1PE10.3,4(10X,E10.3)/15X,'**',8('+'),'+**')                   UTP02270
      GO TO 401                                                                UTP02280
  400 WRITE(6,117) XSCALE                                                      UTP02290
  117 FORMAT( 8X,F11.2,4( 9X,F11.2)/15X,'**',8('+'),'+**')                     UTP02300
  401 II=1                                                                     UTP02310
      DO 101 IK=1,61                                                           UTP02320
      IF(MOD(IK-1,10).NE.0) GO TO 92                                           UTP02330
      IF(AYR.LT.1.E+8.AND.AYR.GE..95) GO TO 404                                UTP02340
      WRITE(6,18) YSCALE(II),(GRID(IK,IX),IX=1,81),YSCALE(II)                  UTP02350
   18 FORMAT(3X,1PE10.3,2X,1H+,1X,81A1,1X,1H+,2X,E10.3)                        UTP02360
      GO TO 405                                                                UTP02370
  404 WRITE(6,118) YSCALE(II),(GRID(IK,IX),IX=1,81),YSCALE(II)                 UTP02380
  118 FORMAT(2X,F11.2,'   + ',81A1,' +   ',F11.2)                              UTP02390
  405 II=II+1                                                                  UTP02400
      GO TO 101                                                                UTP02410
   92 WRITE(6,19) (GRID(IK,IX),IX=1,81)                                        UTP02420
   19 FORMAT(15X,'* ',81A1,' *')                                               UTP02430
  101 CONTINUE                                                                 UTP02440
      IF(AXR.LT.1.E+8.AND.AXR.GE..95) GO TO 402                                UTP02450
      WRITE(6,22) XSCALE                                                       UTP02460
   22 FORMAT(15X,'**',8('+'),'+**'/12X,1PE10.3,4(10X,E10.3),//)                UTP02470
      GO TO 403                                                                UTP02480
  402 WRITE(6,217) XSCALE                                                      UTP02490
  217 FORMAT(15X,'**',8('+'),'+**'/ 8X,F11.2,4( 9X,F11.2),//)                  UTP02500
  403 IF(IERR.GT.0) WRITE(6,20) IERR                                           UTP02510
   20   FORMAT(10X,'NUMBER OF POINTS OUT OF RANGE =', I4)                      UTP02520
 1000 RETURN                                                                   UTP02530
C                                                                              UTP02540
  889 WRITE(6,888)                                                             UTP02550
  888 FORMAT(' ALL Y VALUES=0. CANNOT SETUP PLOT GRID. CHECK MAX & MIN YUTP02560
     1 WHEN MODCUR=0 OR 1.')                                                   UTP02570
```

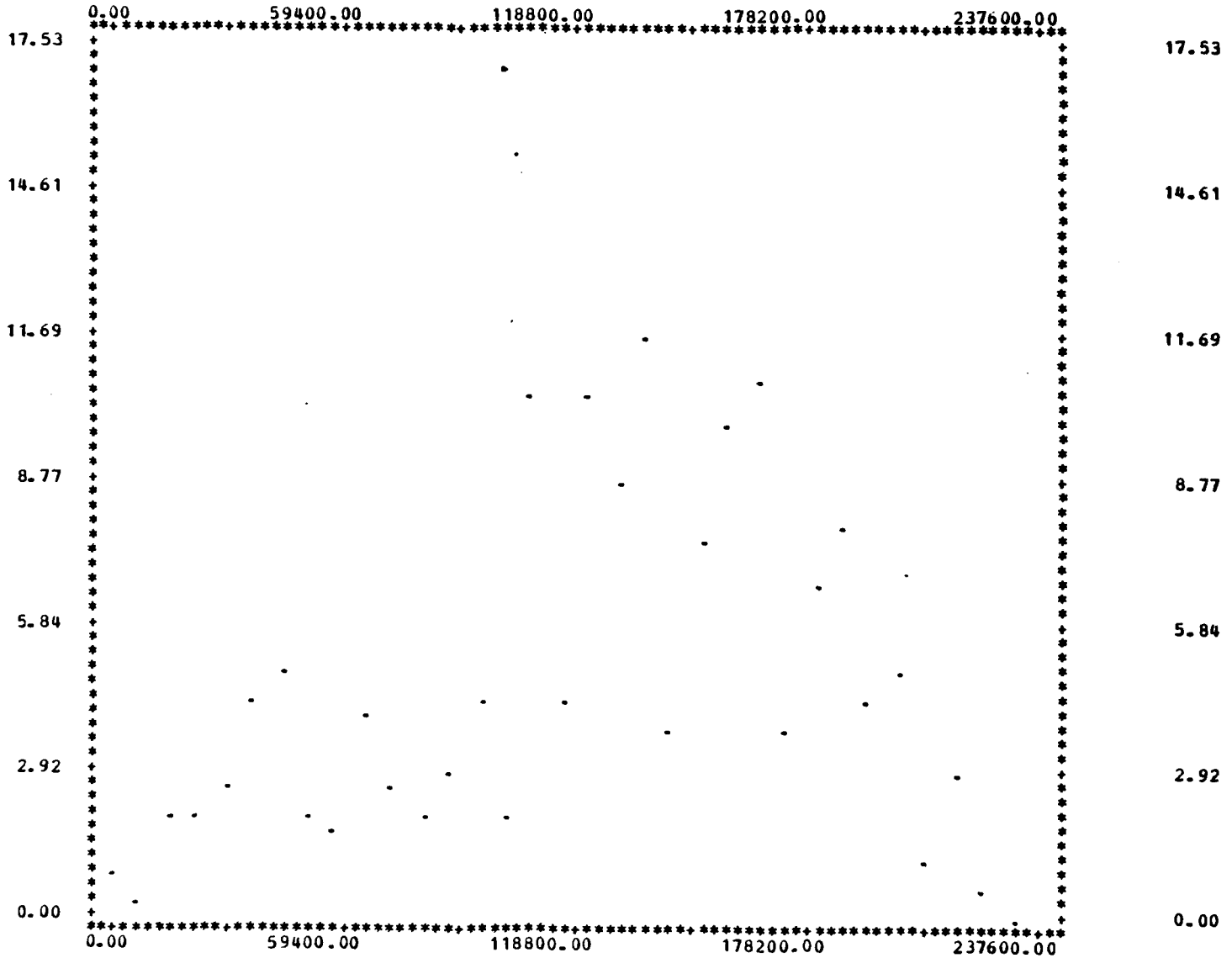
```
      JERR=10                                                                   UTP02580
      RETURN                                                                    UTP02590
  887 WRITE(6,886)                                                              UTP02600
  886 FORMAT(' ALL X VALUES=0. CANNOT SETUP PLOT GRID. CHECK MAX & MIN XUTP02610
     1 WHEN MODCUR=0 OR 1.')                                                    UTP02620
      JERR=10                                                                   UTP02630
      RETURN                                                                    UTP02640
  885 WRITE(6,884)                                                              UTP02650
  884 FORMAT(' GRID NOT SETUP WHEN MODCUR LAST 0 OR 1. NO PLOT UNTIL GRIUTP02660
     1D PROPERLY SETUP')                                                        UTP02670
      RETURN                                                                    UTP02680
      END                                                                       UTP02690
```

APPENDIX C

REPRESENTATIVE RESULTS OF TRIAL SIMULATIONS

```
MODULATION TECHNIQUE = BPSK
BIPOLAR LOGIC
BAUD OR SYMBOL RATE =        1200 HZ
BITS PER BINARY CODE WORD =           1
BIT RATE =  0.1200000000000000E+04
CARRIER FREQUENCY =        2400 HZ
MAXIMUM CARRIER AMPLITUDE =           1 VOLT(S)
INITIAL PHASE ANGLE =           0 DEGREES
TIME BETWEEN SAMPLES =  0.1388888888888889E-03 SEC
NUMBER OF SAMPLES GENERATED =          64
SEED FOR RANDOM NUMBER GENERATOR =           1
NUMBER OF TIMES SIMULATION REPEATS =         100
   SUM OF VARIANCES         SUM OF VARIANCES**2
 0.2015734723265312E+03   0.2783281622639908E+04
   SUM OF SKEWNESS           SUM OF SKEWNESS**2
-0.1039295264723739E+05   0.1054665065832765E+08
   SUM OF KURTOSIS           SUM OF KURTOSIS**2
 0.7097901826525830E+04   0.5645466161770898E+07
    MEAN VARIANCE         VARIANCE OF THE VARIANCES
 0.6108287040197914E+01   0.4850040606187702E+02
    MEAN SKEWNESS         VARIANCE OF THE SKEWNESS
-0.3149379590071935E+03   0.2272973551108844E+06
    MEAN KURTOSIS         VARIANCE OF THE KURTOSIS
 0.2150879341371464E+03   0.1287122850373493E+06
```

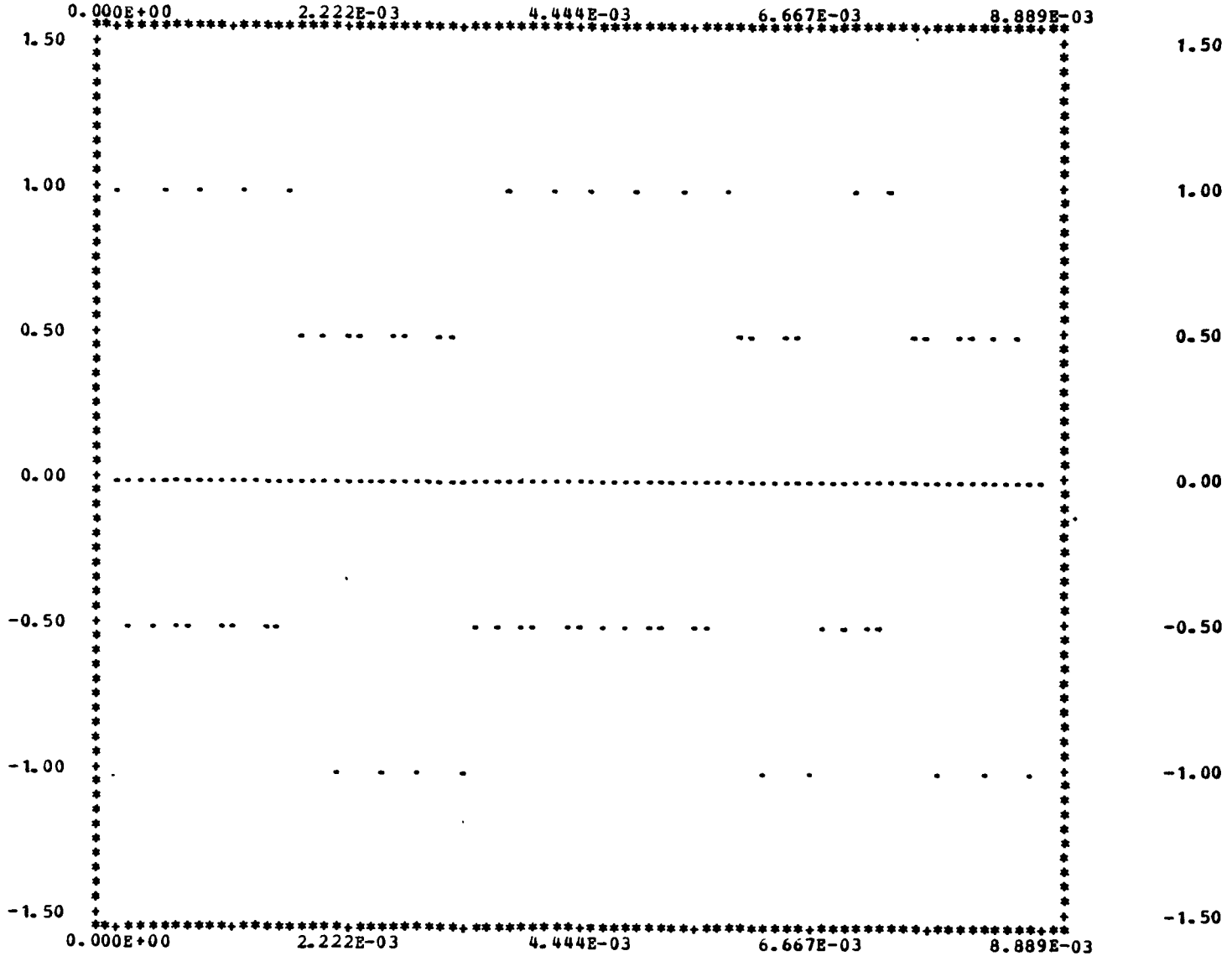

0.000E+00
2.222E-03
4.444E-03
6.667E-03
8.889E-03
1.50
1.00
0.50
0.00
-0.50
-1.00
-1.50

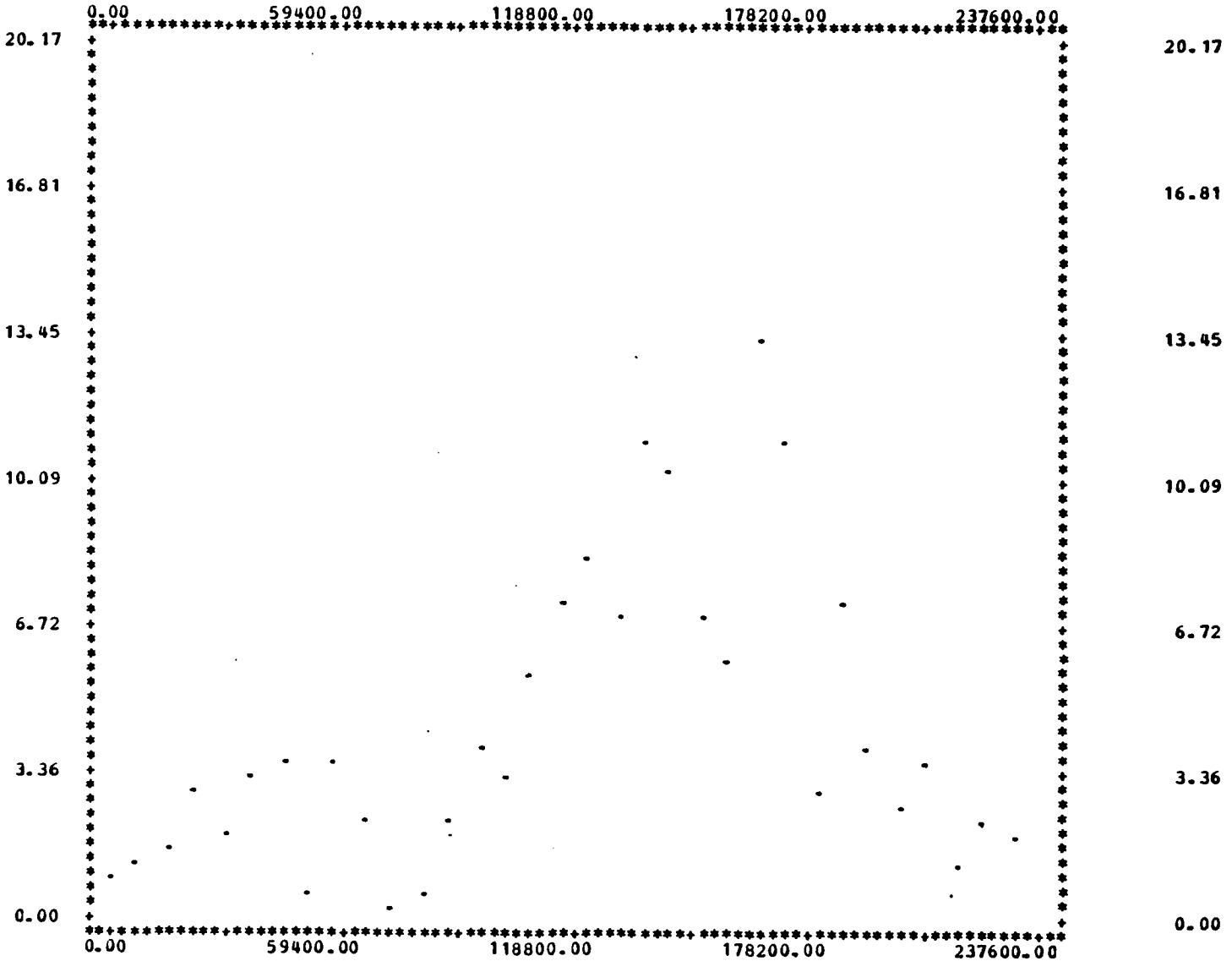

0.00
59400.00
118800.00
178200.00
237600.00
17.53
14.61
11.69
8.77
5.84
2.92
0.00

N	MEAN	VARIANCE	SKEWNESS	KURTOSIS
1	0.10000000000000006E+01	0.15617585788019998E-14	-0.14654943925052 07E-13	0.22204460492503 13E-14
2	0.10474936087386 69E+01	0.27098402684988 55E+00	-0.49861649582595 42E-01	0.18595903369388 43E+00
3	0.17819901495025 63E+01	0.60525719912713 20E+00	-0.48513774115861 08E+01	0.94028016725030 74E+00
4	0.23741333194221 79E+01	0.10612173115251 98E+01	-0.15591587286545 02E+02	0.25556983059345 59E+01
5	0.26415836902075 20E+01	0.13271393624012 61E+01	-0.22386360806945 28E+02	0.46598156369656 93E+01
6	0.26346264721782 04E+01	0.18020091515639 38E+01	-0.21052443723499 55E+02	0.10437292574859 51E+02
7	0.29301785934097 05E+01	0.17904649954522 98E+01	-0.31807661873270 60E+02	0.92303317811799 99E+01
8	0.27425848599746 33E+01	0.14449923792228 51E+01	-0.25561073454448 32E+02	0.54070863070965 91E+01
9	0.28524005695562 03E+01	0.19319844176050 75E+01	-0.29334168093063 37E+02	0.84046398122746 88E+01
10	0.20568528754678 49E+01	0.94609876708812 28E+00	-0.85729910548165 28E+01	0.21937280301634 35E+01
11	0.14568091840093 50E+01	0.10989438470468 64E+01	-0.15226169278735 01E+01	0.20011094089895 01E+01
12	0.14456889200577 78E+01	0.55734802879192 84E+00	-0.17017803278642 10E+01	0.50795784811824 98E+00
13	0.19209456258229 21E+01	0.89272819157995 00E+00	-0.65894275337994 27E+01	0.23481723496082 27E+01
14	0.32245662863639 47E+01	0.20996063102607 72E+01	-0.46087997723253 35E+02	0.95352381587707 67E+01
15	0.40837557642087 47E+01	0.39452430481489 41E+01	-0.10147726958977 62E+03	0.31455354392598 85E+02
16	0.53910879959936 49E+01	0.55201393401574 94E+01	-0.24787088036278 54E+03	0.94244220920778 16E+02
17	0.51773157290582 15E+01	0.11369092769336 18E+02	-0.19386826947235 99E+03	0.33472842213472 90E+03
18	0.74377475418305 37E+01	0.12966762172702 20E+02	-0.69630609306674 82E+03	0.46502188405090 96E+03
19	0.82810714753 16468E+01	0.14667817335687 76E+02	-0.99295108121632 69E+03	0.42780827174800 65E+03
20	0.91680336724564 26E+01	0.20271973042382 61E+02	-0.13672707353972 45E+04	0.93774487023967 78E+03
21	0.87700447108335 03E+01	0.19383464543620 91E+02	-0.11686364753125 38E+04	0.99212912012353 63E+03
22	0.95241068645935 13E+01	0.19615335860496 62E+02	-0.15109928800293 81E+04	0.13786926713289 36E+04
23	0.89622196209415 68E+01	0.18421757591839 58E+02	-0.12620486843368 93E+04	0.76407639590783 40E+03
24	0.79911369765868 05E+01	0.15913040918808 72E+02	-0.86978728090315 44E+03	0.58317432335979 81E+03
25	0.81328130490123 53E+01	0.16735856896146 68E+02	-0.91912747261762 90E+03	0.61368771642773 39E+03
26	0.e3618585268674 42E+01	0.85667618675737 08E+01	-0.42689217702064 31E+03	0.19037850808209 35E+03
27	0.52942988176409 66E+01	0.70061925238428 90E+01	-0.22995560051053 39E+03	0.13301921366538 15E+03
28	0.40579300558895 89E+01	0.49309411269138 54E+01	-0.95194194930045 50E+02	0.67818822617122 92E+02
29	0.34168288806861 42E+01	0.25013140427852 88E+01	-0.55019999133514 84E+02	0.17266423705627 35E+02
30	0.28675334283015 56E+01	0.14126488502520 24E+01	-0.30773934811793 99E+02	0.41506464439664 73E+01
31	0.20203039088375 64E+01	0.10607448827420 50E+01	-0.80509054704337 77E+01	0.26145324036069 47E+01
32	0.13655717437594 67E+01	0.45197516094207 14E+00	-0.12737651890382 58E+01	0.47044243859126 96E+00
33	0.10800000000007 54E+01	0.10036363636363 23E+01	-0.34560000000201 34E+00	0.10126771199999 34E+01

N	SUM X	SUM X**2	SUM X**3	SUM X**4
1	0.1000000000000007E+03	0.1000000000000015E+03	0.1000000000000022E+03	0.1000000000000030E+03
2	0.1047493608738669E+03	0.1365517046929747E+03	0.2046763767568517E+03	0.3383849595387993E+03
3	0.1781990149502563E+03	0.3774693520060027E+03	0.8977044812979923E+03	0.2326050692513711E+04
4	0.2374133319422179E+03	0.6687114156800523E+03	0.2076369626799344E+04	0.6889765711506957E+04
5	0.2641583690207520E+03	0.8291832361147630E+03	0.2936841778974143E+04	0.1138916448013025E+05
6	0.2634626472178204E+03	0.8725245707950468E+03	0.3402833293597278E+04	0.1502031925104808E+05
7	0.2930178593409705E+03	0.1035850693477425E+04	0.4207727078806093E+04	0.1899359170922933E+05
8	0.2742584859974633E+03	0.8952314169592701E+03	0.3305282702701148E+04	0.1337153864187625E+05
9	0.2852400569556203E+03	0.1004885358263358E+04	0.4038352093600026E+04	0.1772010346221309E+05
10	0.2056852875467849E+03	0.5167281530737599E+03	0.1485073506709137E+04	0.4690631226816507E+04
11	0.1456809184009350E+03	0.3210247407190382E+03	0.8262950792109978E+03	0.2278514895214681E+04
12	0.1445688920057778E+03	0.2641791002081834E+03	0.5575810707434266E+03	0.1272769524951599E+04
13	0.1920945625822921E+03	0.4573833007032363E+03	0.1238876179577409E+04	0.3712419048853903E+04
14	0.3224566286363947E+03	0.1247643798231314E+04	0.5380965068146941E+04	0.2495637275319738E+05
15	0.4083755764208747E+03	0.2058285175937562E+04	0.1173346266754204E+05	0.7229234279025983E+05
16	0.5391087995993649E+03	0.3452876772730274E+04	0.2524443357094650E+05	0.2050931970507939E+06
17	0.5177315729058215E+03	0.3805999999999641E+04	0.3436684441547441E+05	0.3486209999999403E+06
18	0.7437747541830537E+03	0.6815718304698137E+04	0.7138614919385783E+05	0.8261276298378326E+06
19	0.8281071475316468E+03	0.8309728394163094E+04	0.9343002724010123E+05	0.1129295226467696E+07
20	0.9168033672456426E+03	0.1041220947312536E+05	0.1328408193392659E+06	0.1833740201322952E+07
21	0.8770044710833503E+03	0.9610331412820340E+04	0.1206027241518287E+06	0.1669696517013331E+07
22	0.9524106864593513E+03	0.1101277940700889E+05	0.1454196535290649E+06	0.2152530635514160E+07
23	0.8962219620941568E+03	0.9855892054991122E+04	0.1227228629308833E+06	0.1661508753740510E+07
24	0.7991136976758o806E+03	0.7961218068819355E+04	0.9110717013921001E+05	0.1143542851587153E+07
25	0.8132813049012353E+03	0.8271114641737083E+04	0.9666101038478942E+05	0.1235886896537965E+07
26	0.6361858526867442E+03	0.4895433816477395E+04	0.4264830561897630E+05	0.4069483181946355E+06
27	0.5294298817640966E+03	0.3496573056907900E+04	0.2693422775779996E+05	0.2313456000907510E+06
28	0.4057930055689589E+03	0.2134842805413680E+04	0.1317634959336049E+05	0.9107918677629661E+05
29	0.3416828880886142E+03	0.1415102050361508E+04	0.6668540831626178E+04	0.3463189945762253E+05
30	0.2867533428301556E+03	0.9621270324176382E+03	0.3554851209525683E+04	0.1400578999184000E+05
31	0.2020303908637565E+03	0.5131765317978972E+03	0.1488901535488743E+04	0.4723919613838425E+04
32	0.1365571743759467E+03	0.2312241596686921E+03	0.4469257803795999E+03	0.9444076554027791E+03
33	0.1080000000000754E+03	0.2160000000001588E+03	0.4320000000004758E+03	0.8640000000012697E+03

```
MODULATION TECHNIQUE = DBPSK
BIPOLAR LOGIC
BAUD OR SYMBOL RATE =        1200 HZ
BITS PER BINARY CODE WORD =           1
BIT RATE =  0.1200000000000000E+04
CARRIER FREQUENCY =        2400 HZ
MAXIMUM CARRIER AMPLITUDE =           1 VOLT(S)
INITIAL PHASE ANGLE =           0 DEGREES
TIME BETWEEN SAMPLES =  0.1388888888888889E-03 SEC
NUMBER OF SAMPLES GENERATED =          64
SEED FOR RANDOM NUMBER GENERATOR =           1
NUMBER OF TIMES SIMULATION REPEATS =         100
   SUM OF VARIANCES         SUM OF VARIANCES**2
 0.2149282745632018E+03   0.3048987837599078E+04
   SUM OF SKEWNESS           SUM OF SKEWNESS**2
-0.9719603900974997E+04   0.8598103883304453E+07
   SUM OF KURTOSIS           SUM OF KURTOSIS**2
 0.8665643426712650E+04   0.8293411900734219E+07
    MEAN VARIANCE         VARIANCE OF THE VARIANCES
 0.6512978017066721E+01   0.5153639719133948E+02
    MEAN SKEWNESS         VARIANCE OF THE SKEWNESS
-0.2945334515446968E+03   0.1792298502094682E+06
    MEAN KURTOSIS         VARIANCE OF THE KURTOSIS
 0.2625952553549290E+03   0.1880579703838077E+06
```

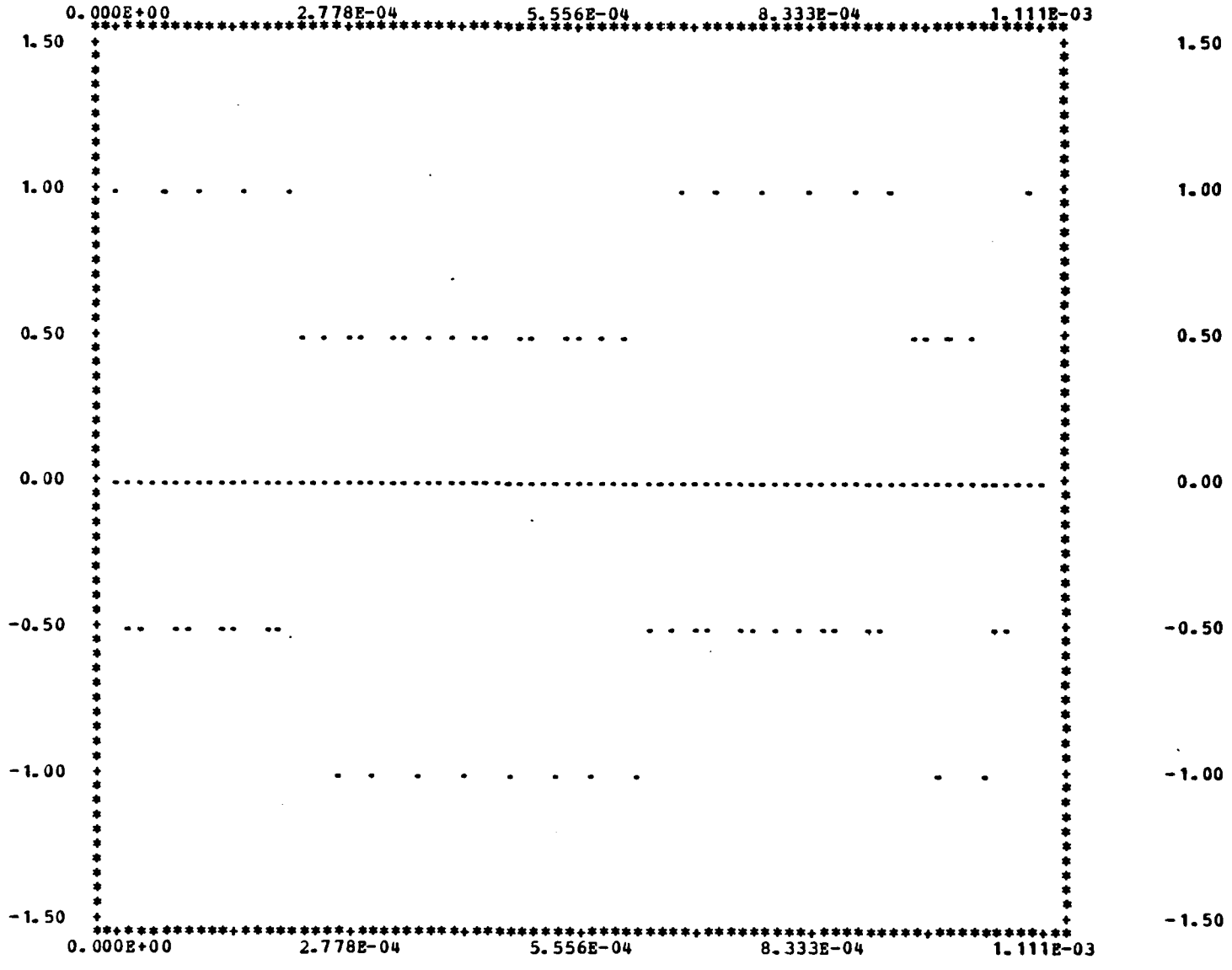

0.000E+00
2.222E-03
4.444E-03
6.667E-03
8.889E-03
1.50
1.00
0.50
0.00
-0.50
-1.00
-1.50

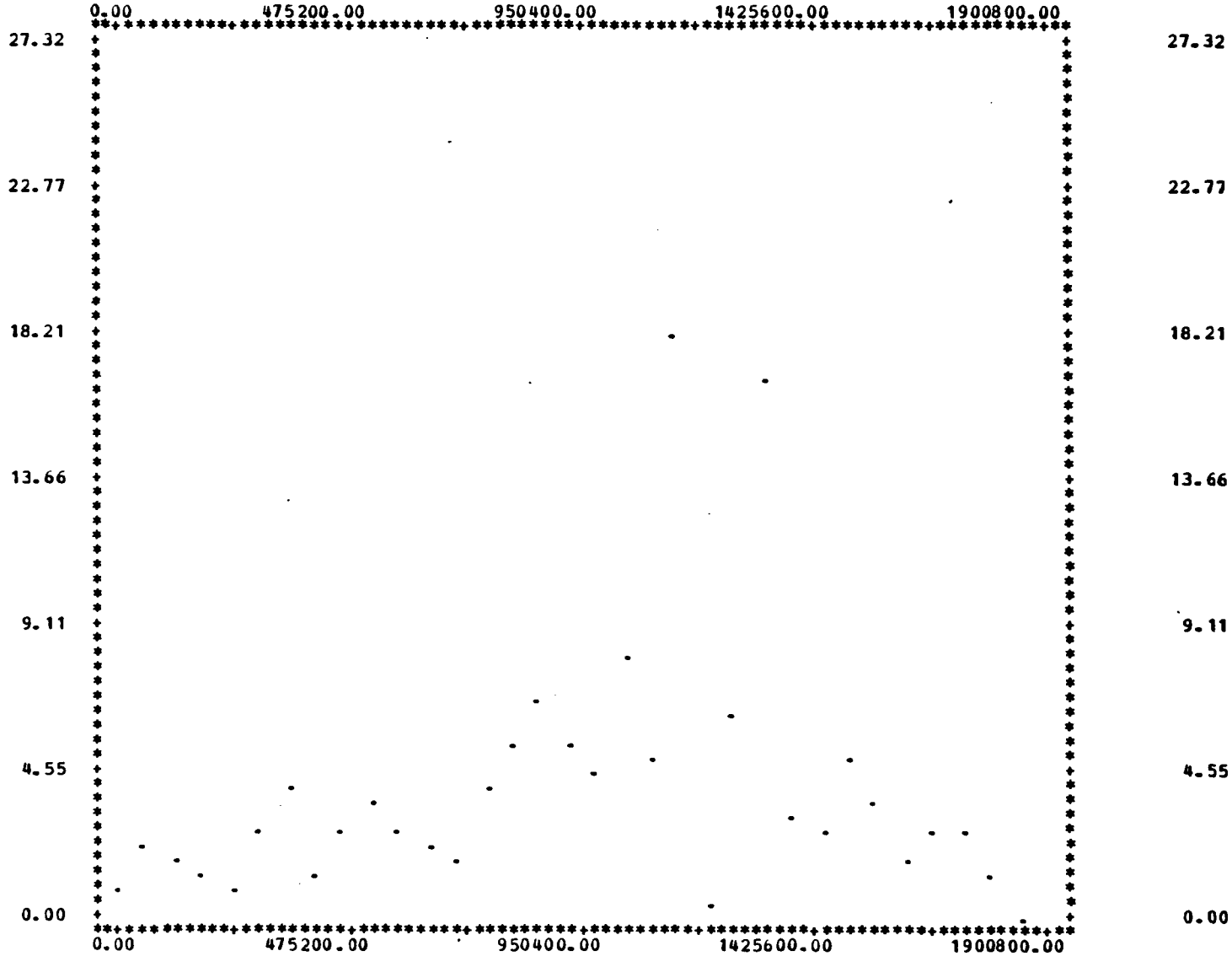

0.00
59400.00
118800.00
178200.00
237600.00
20.17
16.81
13.45
10.09
6.72
3.36
0.00

N	MEAN	VARIANCE	SKEWNESS	KURTOSIS
1	0.99999999999999964E+00	0.14698904271077 63E-14	0.57731597280508 14E-14	0.22204460492503 13E-14
2	0.12069912150192 78E+01	0.33945791151908 50E+00	-0.58253780559475 65E+00	0.21218495177473 92E+00
3	0.16141938143604 49E+01	0.65941389229334 82E+00	-0.30025376157503 80E+01	0.11427438413920 47E+01
4	0.22276699861064 46E+01	0.10286887610266 41E+01	-0.11534444895283 57E+02	0.36029774006613 02E+01
5	0.26068303843591 21E+01	0.18607719627260 79E+01	-0.20564679000786 79E+02	0.83932330990096 28E+01
6	0.28698275599932 43E+01	0.27736070565238 04E+01	-0.27064588260368 14E+02	0.24840784135861 56E+02
7	0.31938929229353 19E+01	0.19196376588110 91E+01	-0.43899370255183 55E+02	0.10490693988386 91E+02
8	0.28870295628730 31E+01	0.21880922538864 60E+01	-0.29962734620759 21E+02	0.12406237189273 21E+02
9	0.26451285513611 1E+01	0.15644235193311 396E+01	-0.22827821273769 95E+02	0.47839088038363 40E+01
10	0.21655326383634 91E+01	0.10129070975130 55E+01	-0.10237894519585 66E+02	0.30653609317834 77E+01
11	0.16912773580745 38E+01	0.10049601535915 63E+01	-0.39114848048149 08E+01	0.14991847015663 39E+01
12	0.15890186092188 27E+01	0.64095170413685 64E+00	-0.29572435844628 53E+01	0.60089478702435 88E+00
13	0.20222066216960 48E+01	0.86748550683966 50E+00	-0.81249204565791 11E+01	0.17595843754149 42E+01
14	0.28931808794861 28E+01	0.20565271747771 21E+01	-0.30299724712234 51E+02	0.10992954594464 58E+02
15	0.38004400080631 25E+01	0.38503692838933 50E+01	-0.77698741231129 18E+02	0.37378698768705 87E+02
16	0.55700475617526 25E+01	0.74940365824540 18E+01	-0.27692260815755 63E+03	0.13048514891267 16E+03
17	0.63093960474347 41E+01	0.19870223754151 37E+02	-0.33593327528536 99E+03	0.12323192928915 67E+04
18	0.77259945589885 40E+01	0.15315495115317 86E+02	-0.78772680971775 33E+03	0.53898867300557 89E+03
19	0.78092158033807 67E+01	0.15446493139540 77E+02	-0.80250802973050 87E+03	0.73113685787785 02E+03
20	0.84146549852681 34E+01	0.16534862795720 41E+02	-0.10105170113283 95E+04	0.83423659798145 13E+03
21	0.91763817069658 88E+01	0.22994617440779 66E+02	-0.13177510571634 12E+04	0.14830702635955 37E+04
22	0.87374509799827 24E+01	0.20482243074439 57E+02	-0.11279353516427 33E+04	0.14501981620980 86E+04
23	0.87097958164724 41E+01	0.16664530810753 22E+02	-0.11603473101887 29E+04	0.60123820671004 71E+03
24	0.83594466185652 24E+01	0.14167328326133 74E+02	-0.10234225949616 01E+04	0.51247369841548 24E+03
25	0.73658729512430 31E+01	0.12624677555343 49E+02	-0.66638650464996 41E+03	0.44047159928280 96E+03
26	0.62654201852471 74E+01	0.10293411825290 51E+02	-0.39427277606144 18E+03	0.27356911084126 66E+03
27	0.56610862079935 09E+01	0.74785652443779 20E+01	-0.28915081501075 28E+03	0.14831814538218 55E+03
28	0.46370406462588 31E+01	0.63848871297703 13E+01	-0.14505761530041 11E+03	0.12435055892766 38E+03
29	0.35805360362747 82E+01	0.35790404295954 00E+01	-0.63161349984625 17E+02	0.34540422372828 12E+02
30	0.25683035701481 02E+01	0.13673769581639 26E+01	-0.20087041879639 02E+02	0.53643780406499 46E+01
31	0.17919067909411 65E+01	0.84112207887716 88E+00	-0.48412942697617 84E+01	0.17860126113311 70E+01
32	0.13904753445732 20E+01	0.61843200198660 56E+00	-0.12081326060450 36E+01	0.91417907609846 58E+00
33	0.92000000000077 91E+00	0.10036363636363 74E+01	0.29439999999891 06E+00	0.10126771200000 26E+01

N	SUM X	SUM X**2	SUM X**3	SUM X**4
1	0.99999999999999964E+02	0.99999999999999943E+02	0.99999999999999921E+02	0.99999999999999902E+02
2	0.12069912150192 78E+03	0.17928911255376 07E+03	0.29958181221675 70E+03	0.53713294988466 72E+03
3	0.16141938143604 49E+03	0.32584414236899 51E+03	0.75654870153390 66E+03	0.19417592568846 56E+04
4	0.22276699861064 46E+03	0.59809154404158 69E+03	0.18511045418882 58E+04	0.64346357176172 18E+04
5	0.26068303843591 21E+03	0.86377288959167 40E+03	0.33395474106774 92E+04	0.14296820536307 66E+05
6	0.28698275599932 44E+03	0.10981781210055 34E+04	0.51011046414736 61E+04	0.27123327837429 79E+05
7	0.31938929229353 19E+03	0.12101393285399 29E+04	0.51650388854145 19E+04	0.24185781255864 38E+05
8	0.28870295628730 32E+03	0.10501151028250 44E+04	0.44318786373695 78E+04	0.20745971883713 01E+05
9	0.26451286551361 11E+03	0.85454848863602 52E+03	0.30990488356617 87E+04	0.12079943994273 54E+05
10	0.21655326383634 91E+03	0.56923096343554 66E+03	0.17363689167382 38E+04	0.59281427780637 67E+04
11	0.16912773580745 38E+03	0.38553296539912 37E+03	0.99289722464400 80E+03	0.27048609295927 14E+04
12	0.15890186092188 27E+03	0.31595223275392 22E+03	0.70544154587562 92E+03	0.16699471898478 40E+04
13	0.20222066218960 49E+03	0.49481302734114 97E+03	0.13714865714831 53E+04	0.41457441042037 95E+04
14	0.28931808794861 28E+03	0.10406457504453 47E+04	0.43282575710126 39E+04	0.19944258566803 07E+05
15	0.38004400080631 25E+03	0.18255209845941 26E+04	0.10154805982140 63E+05	0.62492083420928 74E+05
16	0.55700475617526 26E+03	0.38444526056815 85E+04	0.30344004803857 81E+05	0.26223713625894 36E+06
17	0.63093960474347 41E+03	0.59479999999994 99E+04	0.71029839845202 02E+05	0.97058299999983 00E+06
18	0.77259945589885 40E+03	0.74853332089685 21E+04	0.82784051577241 63E+05	0.10003179658253 23E+07
19	0.78092158033807 67E+03	0.76275879671917 28E+04	0.86248868219406 61E+05	0.10920086767767 39E+07
20	0.84146549858681 35E+03	0.87175932698958 67E+04	0.10485363418082 07E+06	0.14131476080288 59E+07
21	0.91763817069658 88E+03	0.10697065249831 01E+05	0.14586475966758 02E+06	0.22250028041633 20E+07
22	0.87374509799827 25E+03	0.96620470271296 25E+04	0.12520284170770 52E+06	0.18435321378667 26E+07
23	0.87097958164724 41E+03	0.92358428667286 53E+04	0.11012007703488 90E+06	0.14192476297938 49E+07
24	0.83594466185652 24E+03	0.83906002811513 95E+04	0.94103996394121 92E+05	0.11448272151155 74E+07
25	0.73658729512430 31E+03	0.66754515113643 77E+04	0.70021715842912 69E+05	0.81714505600587 80E+06
26	0.62654201852471 74E+03	0.49445967804740 33E+04	0.45061553803185 85E+05	0.46939482576211 25E+06
27	0.56610862079935 09E+03	0.39451676646268 47E+04	0.31677141851380 88E+05	0.28165428652550 81E+06
28	0.46370406462588 31E+03	0.27823184213529 12E+04	0.19898980110892 06E+05	0.16127235773150 59E+06
29	0.35805360362747 82E+03	0.16363488332361 77E+04	0.86968352360799 2E+04	0.51448631196131 08E+05
30	0.25683035701481 02E+03	0.79498864170177 75E+03	0.27973756664785 96E+04	0.10864022932902 85E+05
31	0.17919067909411 65E+03	0.40436408055094 61E+03	0.10474328093953 70E+04	0.29889220407328 39E+04
32	0.13904753445732 20E+03	0.25456693658327 54E+03	0.55440954921003 02E+03	0.13433135546801 40E+04
33	0.92000000000077 91E+02	0.18400000000014 44E+03	0.36800000000043 29E+03	0.73600000000115 46E+03

```
MODULATION TECHNIQUE = ORTHOGONAL BPSK
BIPOLAR LOGIC
BAUD OR SYMBOL RATE =        1200 HZ
BITS PER BINARY CODE WORD =           3
BIT RATE =  0.9600000000000000E+04
CARRIER FREQUENCY =       19200 HZ
MAXIMUM CARRIER AMPLITUDE =           1 VOLT(S)
INITIAL PHASE ANGLE =           0 DEGREES
TIME BETWEEN SAMPLES =  0.2450980392156863E-04 SEC
NUMBER OF SAMPLES GENERATED =          64
SEED FOR RANDOM NUMBER GENERATOR =           1
NUMBER OF TIMES SIMULATION REPEATS =         100
   SUM OF VARIANCES         SUM OF VARIANCES**2
 0.3565282677437312E+03   0.1043543362358570E+05
   SUM OF SKEWNESS           SUM OF SKEWNESS**2
-0.5381895541377615E+04   0.3582631135045827E+07
   SUM OF KURTOSIS           SUM OF KURTOSIS**2
 0.3658999158877697E+05   0.3330606974397919E+09
    MEAN VARIANCE         VARIANCE OF THE VARIANCES
 0.1080388690132519E+02   0.2057357044299072E+03
    MEAN SKEWNESS         VARIANCE OF THE SKEWNESS
-0.1630877436781095E+03   0.8452843545285224E+05
    MEAN KURTOSIS         VARIANCE OF THE KURTOSIS
 0.1108787623902332E+04   0.9140317737733299E+07
```

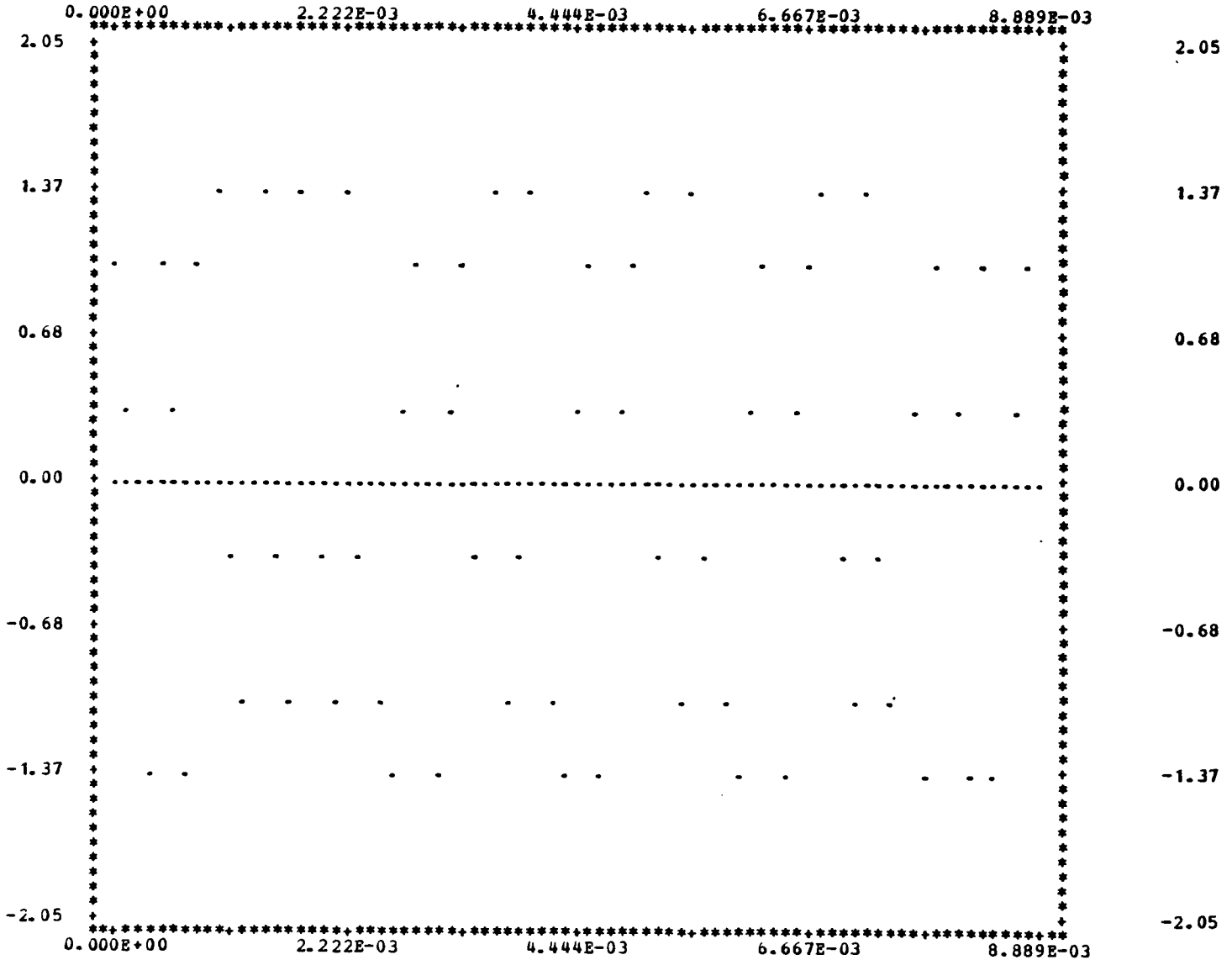

0.000E+00
2.778E-04
5.556E-04
8.333E-04
1.111E-03
1.50
1.00
0.50
0.00
-0.50
-1.00
-1.50

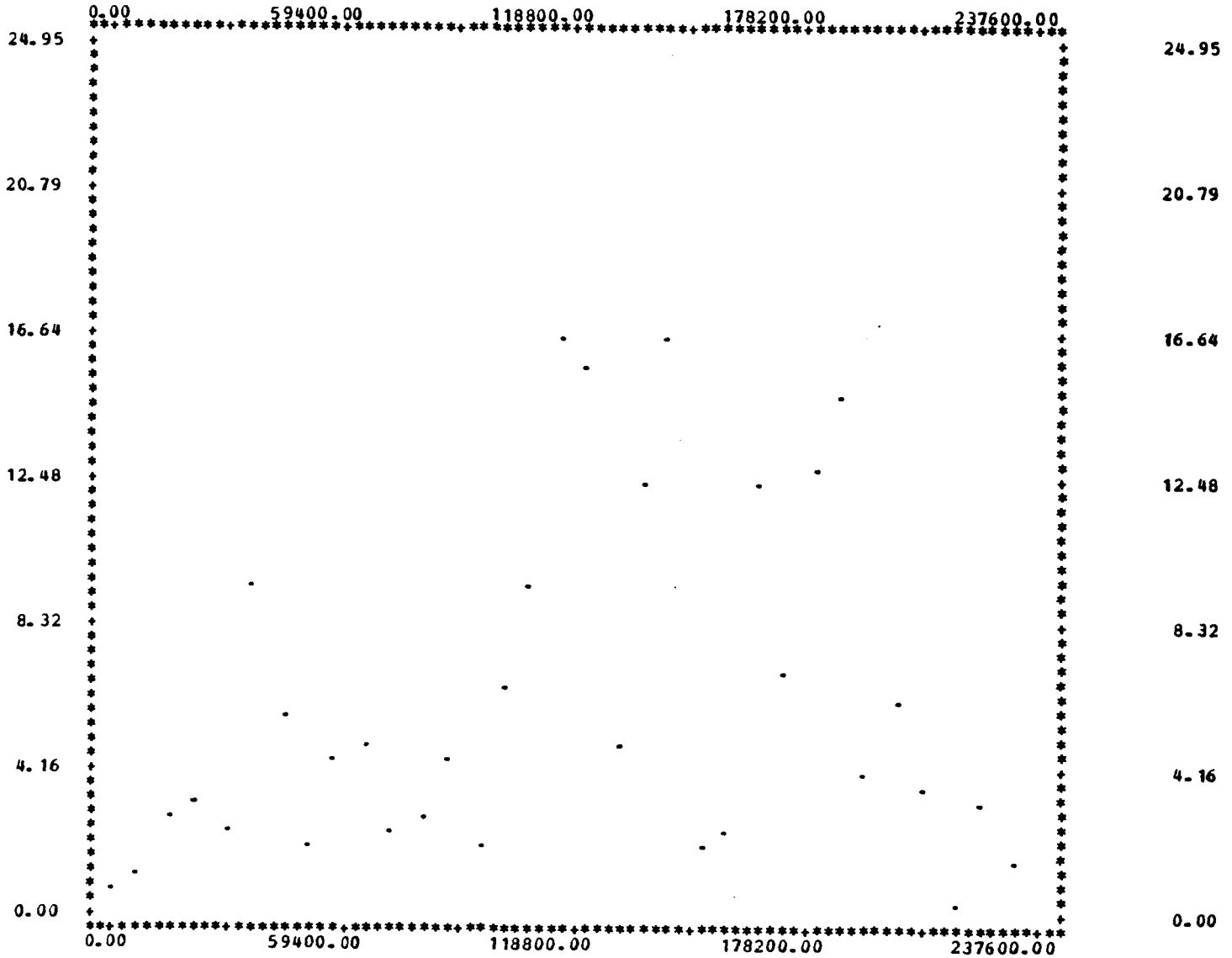

0.00
475200.00
950400.00
1425600.00
1900800.00
27.32
22.77
18.21
13.66
9.11
4.55
0.00

N	MEAN	VARIANCE	SKEWNESS	KURTOSIS
1	0.99999999999999837E+00	0.11942859720250 57E-14	0.319744231092045 1E-13	0.111022302462515 7E-14
2	0.14115550671393 62E+01	0.20757125400375 67E+00	-0.155510411298318 6E+01	0.147782816380499 3E+00
3	0.15816625629712 17E+01	0.69134860021066 69E+00	-0.251203364473896 4E+01	0.132665463525768 6E+01
4	0.17460116688284 25E+01	0.16607289375603 71E+01	-0.273572081753999 09E+01	0.718115321839041 3E+01
5	0.24873768324019 61E+01	0.31871325627890 80E+01	-0.134586653914579 2E+02	0.231310702783005 8E+02
6	0.23923171331983 17E+01	0.44039996748596 14E+01	-0.157914006872995 8E+00	0.952389220872759 3E+02
7	0.28739419937808 31E+01	0.33176112123616 97E+01	-0.280895170722359 6E+02	0.222468447684044 3E+02
8	0.29360293929176 52E+01	0.36999965875187 91E+01	-0.298012664716013 4E+02	0.298469796360983 6E+02
9	0.24782640419162 65E+01	0.29717233152333 70E+01	-0.121100809958272 0E+02	0.273905034383821 7E+02
10	0.17758208993509 69E+01	0.10410756050531 06E+01	-0.459567009697062 8E+01	0.240820598439626 1E+01
11	0.15188612171052 41E+01	0.83012599577176 36E+00	-0.227749027919878 6E+01	0.119237989450032 2E+01
12	0.14343177442044 84E+01	0.59444798798643 32E+00	-0.179691409872974 5E+01	0.429320063605368 7E+00
13	0.19450151832503 01E+01	0.49162501576493 99E+00	-0.673503499572502 1E+01	0.104112915619569 8E+01
14	0.28046587383390 67E+01	0.28043589636811 70E+01	-0.244377333227744 9E+02	0.228968639169829 9E+02
15	0.39230508526105 31E+01	0.30748102138433 98E+01	-0.886641093916191 0E+02	0.211540618701768 6E+02
16	0.56440120862555 05E+01	0.11199686174935 84E+02	-0.281664370788183 1E+03	0.227678126436764 1E+03
17	0.50903457139400 15E+01	0.32432707588453 00E+02	0.180395884209526 0E+03	0.654766409625028 6E+04
18	0.69794808972423 55E+01	0.19898410394620 72E+02	-0.529308324613837 0E+03	0.834025077403497 6E+03
19	0.72196392565455 43E+01	0.14058942163505 26E+02	-0.645066911261434 1E+03	0.441513211637960 0E+03
20	0.74263256419639 08E+01	0.31430133433205 32E+02	-0.504597598167530 9E+03	0.287350451721581 2E+04
21	0.91810749194215 93E+01	0.31475599882440 39E+02	-0.122106603480384 0E+04	0.264531957537546 8E+04
22	0.78111783512003 67E+01	0.65016379852617 42E+02	-0.323425752864646 5E+02	0.162095799860924 1E+05
23	0.71551738811157 02E+01	0.24370366214467 84E+02	-0.567759857528937 4E+03	0.110056260655464 4E+04
24	0.72960608865612 13E+01	0.17285293427269 21E+02	-0.619632113288268 0E+03	0.898941356634658 4E+03
25	0.66868813797712 13E+01	0.29577594015490 51E+02	-0.283552899845750 2E+03	0.287890743383348 6E+04
26	0.66398691556305 94E+01	0.18178287642315 62E+02	-0.454928587173738 1E+03	0.661620789858950 0E+03
27	0.47674896564540 35E+01	0.10170334556986 02E+02	-0.146772825866940 5E+03	0.273925832492854 0E+03
28	0.34421485311873 25E+01	0.11073125746266 55E+02	0.660961993312539 8E+01	0.632376262735424 2E+03
29	0.33393901755805 69E+01	0.64074698215913 61E+01	-0.426785144032657 1E+02	0.899625322047325 0E+02
30	0.23452635417767 11E+01	0.22969217427485 96E+01	-0.117827490046260 0E+02	0.141638352122049 4E+02
31	0.19367214978664 93E+01	0.84199792763246 13E+00	-0.665332423107759 5E+01	0.242136381367239 7E+01
32	0.15074940217659 24E+01	0.82997578406240 45E+00	-0.232070455810058 5E+01	0.118992093980538 8E+01
33	0.96000000000005 325E+00	0.10084848484849 35E+01	0.153599999999107 8E+00	0.100319232000018 0E+01

N	SUM X	SUM X**2	SUM X**3	SUM X**4
1	0.99999999999999837E+02	0.99999999999999686E+02	0.99999999999999536E+02	0.99999999999999387E+02
2	0.14111550671393 62E+03	0.21979832490305 28E+03	0.37676436498526 34E+03	0.70540229205082 21E+03
3	0.15816625629712 17E+03	0.31860915773132 39E+03	0.76026187391544 71E+03	0.20377580055919 90E+04
4	0.17460116688284 25E+03	0.46926783958697 91E+03	0.15747579398833 18E+04	0.59208699163949 98E+04
5	0.24873768324019 61E+03	0.93423047435312 01E+03	0.43880744736697 41E+04	0.22775390832590 13E+05
6	0.23923171331983 17E+03	0.10083140943905 23E+04	0.60761935971212 37E+04	0.42870475526848 54E+05
7	0.28739419937808 32E+03	0.11543977683855 02E+04	0.54921564483229 34E+04	0.28618468638538 25E+05
8	0.29360293929176 52E+03	0.12283265217720 00E+04	0.61150279496932 74E+04	0.33561987111256 26E+05
9	0.24782640419162 65E+03	0.90837987435361 79E+03	0.43142489051592 08E+04	0.23348442114148 64E+05
10	0.17758208993509 69E+03	0.41842047155742 60E+03	0.11388444713124 94E+04	0.33967661302484 46E+04
11	0.15188612171052 41E+03	0.31287641326404 59E+03	0.73651064227609 20E+03	0.18597370612132 69E+04
12	0.14343177442044 84E+03	0.26457708994466 08E+03	0.54731795631180 97E+03	0.11869132502084 75E+04
13	0.19450151832503 01E+03	0.42697928286814 93E+03	0.10613232521482 84E+04	0.29629986614772 88E+04
14	0.28046587383390 67E+03	0.10642426012586 05E+04	0.49375164740121 45E+04	0.26015889976659 44E+05
15	0.39230508526105 31E+03	0.18434390103873 18E+04	0.97512384067108 93E+04	0.55965695460282 66E+05
16	0.56440120862555 05E+03	0.42942561742984 70E+04	0.38173089682875 71E+05	0.36822730437984 86E+06
17	0.50903457139400 15E+03	0.58019999999995 95E+04	0.10145982202030 08E+06	0.20200169999997 56E+07
18	0.69794808972423 56E+03	0.68412579885645 47E+04	0.80571825152500 85E+05	0.10451364586432 42E+07
19	0.72196392565455 43E+03	0.66041543736523 68E+04	0.68107507191848 61E+05	0.76066836469789 52E+06
20	0.74263256419639 08E+03	0.86266144639363 93E+04	0.13070232226575 15E+06	0.22278066698267 87E+07
21	0.91810749194215 93E+03	0.11545298055964 82E+05	0.17902970844100 59E+06	0.31317548629636 41E+07
22	0.78111783512003 67E+03	0.12538072328835 25E+05	0.27837419844684 82E+06	0.68454852474158 00E+07
23	0.71551738811157 03E+03	0.75323175821323 51E+04	0.94669837677132 12E+05	0.12921273691940 78E+07
24	0.72960608885612 13E+03	0.70344944982589 15E+04	0.81362588311892 37E+05	0.10677270296683 44E+07
25	0.66868137977112 13E+03	0.73996 20125646657E+04	0.11114297960469 96E+06	0.18752865578860 83E+07
26	0.66398691556305 94E+03	0.62084367169787 00E+04	0.69359191187101 30E+05	0.84912755695153 16E+06
27	0.47674896564540 35E+03	0.32797588835812 37E+04	0.27685575547837 06E+05	0.26306401378699 49E+06
28	0.34421485311873 25E+03	0.22810780999558 95E+04	0.21846711358502 179E+05	0.24398886752084 97E+06
29	0.33393901755805 69E+03	0.17494921868139 47E+04	0.11028554273425 01E+05	0.76560747542291 15E+05
30	0.23452635417767 11E+03	0.77742136057081 54E+03	0.31914468040955 30E+04	0.14775257731951 63E+05
31	0.19367214978664 93E+03	0.45844681086543 69E+03	0.12481409275268 72E+04	0.38145931552155 03E+04
32	0.15074940217659 24E+03	0.30942142518817 80E+03	0.71277474509035 28E+03	0.17473036589940 75E+04
33	0.96000000000053 25E+02	0.19200000000011 09E+03	0.38400000000033 22E+03	0.76800000000088 64E+03

```
MODULATION TECHNIQUE = QPSK
BIPOLAR LOGIC
BAUD OR SYMBOL RATE =        1200 HZ
BITS PER BINARY CODE WORD =           2
BIT RATE =  0.2400000000000000E+04
CARRIER FREQUENCY =        2400 HZ
MAXIMUM CARRIER AMPLITUDE =           1 VOLT(S)
INITIAL PHASE ANGLE =           0 DEGREES
TIME BETWEEN SAMPLES =  0.1388888888888889E-03 SEC
NUMBER OF SAMPLES GENERATED =          64
SEED FOR RANDOM NUMBER GENERATOR =           1
NUMBER OF TIMES SIMULATION REPEATS =         100
   SUM OF VARIANCES         SUM OF VARIANCES**2
 0.4090708290354776E+03   0.1062919801427033E+05
   SUM OF SKEWNESS           SUM OF SKEWNESS**2
-0.2867982085342969E+05   0.7352270919634266E+08
   SUM OF KURTOSIS           SUM OF KURTOSIS**2
 0.3172130538617988E+05   0.9751577105721891E+08
    MEAN VARIANCE          VARIANCE OF THE VARIANCES
 0.1239608572834781E+02   0.1736975296431322E+03
    MEAN SKEWNESS          VARIANCE OF THE SKEWNESS
-0.8690854804069604E+03   0.1518671665998567E+07
    MEAN KURTOSIS          VARIANCE OF THE KURTOSIS
 0.9612516783690871E+03   0.2094487906709223E+07
```

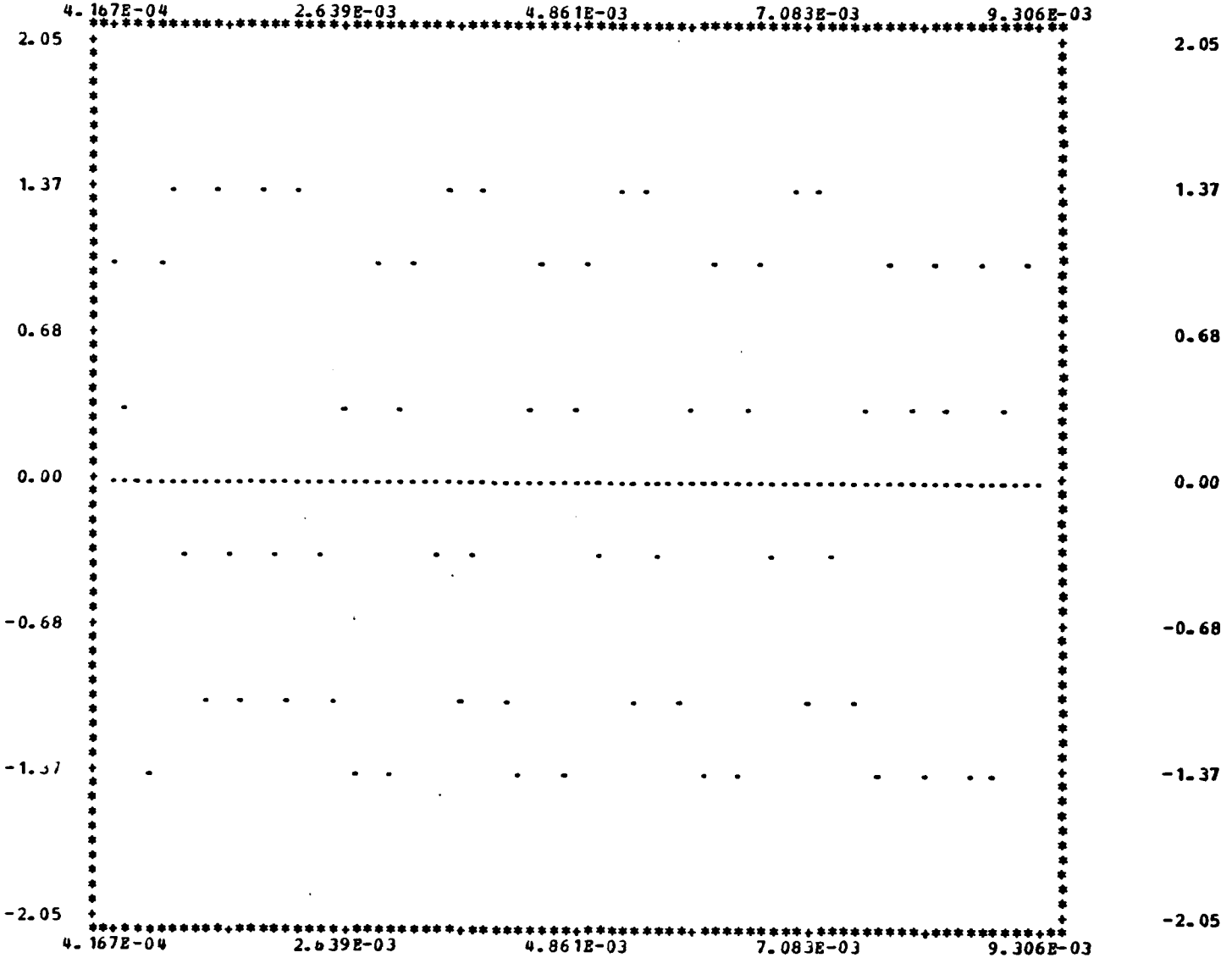

0.000E+00
2.222E-03
4.444E-03
6.667E-03
8.889E-03
2.05
1.37
0.68
0.00
-0.68
-1.37
-2.05

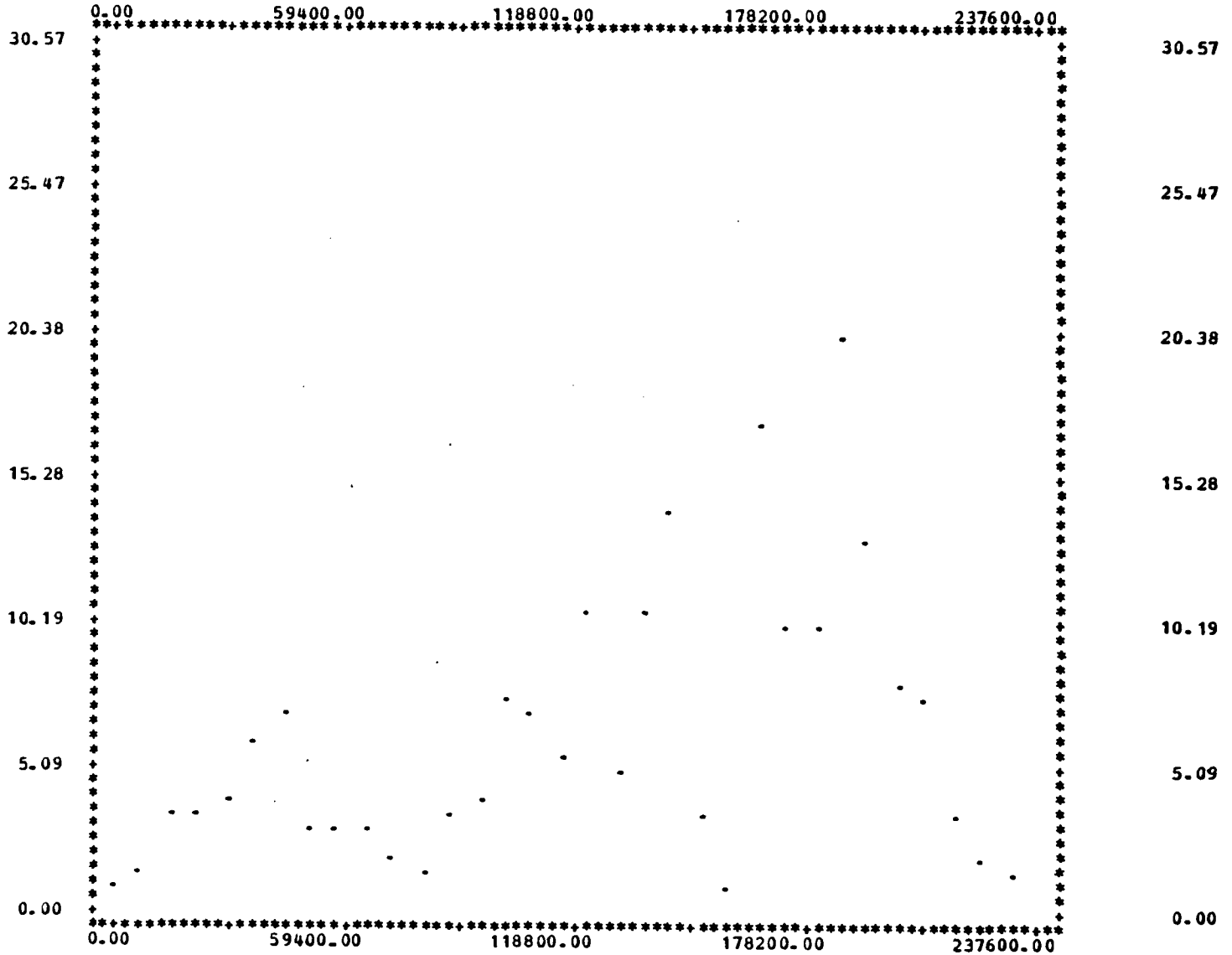

0.00
59400.00
118800.00
178200.00
237600.00
24.95
20.79
16.64
12.48
8.32
4.16
0.00

N	MEAN	VARIANCE	SKEWNESS	KURTOSIS
1	0.99999999999999883E+00	0.14698904271077 63E-14	0.21982415887578 10E-13	0.24424906541753 44E-14
2	0.14285950977732444E+01	0.43938047543711 42E+00	-0.16994320274213 95E+01	0.43868836060089 58E+00
3	0.19041231243113 41E+01	0.82840874959538 19E+00	-0.63968248883996 20E+01	0.13493183492171 00E+01
4	0.25532157885240 60E+01	0.18793029634204 28E+01	-0.18625446220193 88E+02	0.10269231662956 30E+02
5	0.28952891683411 45E+01	0.19124220142824 12E+01	-0.30393762386033 14E+02	0.10845705046518 63E+02
6	0.35933196779491 41E+01	0.31843097312263 40E+01	-0.64026918309457 79E+02	0.33669258987618 04E+02
7	0.39304643411724 16E+01	0.32830457871474 81E+01	-0.89400379345453 91E+02	0.22832477281932 20E+02
8	0.31763748833679 92E+01	0.31454364539734 28E+01	-0.40145076890684 21E+02	0.24708095503393 91E+02
9	0.29091093956305 61E+01	0.21513049992525 31E+01	-0.30954635700674 97E+02	0.11351746154766 98E+02
10	0.23101136641512 72E+01	0.14489362371016 54E+01	-0.13019851079449 30E+02	0.50722640667945 46E+01
11	0.19296351101333 16E+01	0.97445621395398 73E+00	-0.67131887575622 22E+01	0.19686960110138 79E+01
12	0.17363337975928 57E+01	0.68027116027733 72E+00	-0.44833541756800 31E+01	0.98589415894865 34E+00
13	0.26360688597277 89E+01	0.17943581241550 69E+01	-0.21864589074253 96E+02	0.72416985640946 48E+01
14	0.38034813443652 38E+01	0.36965405559311 17E+01	-0.79094430267618 86E+02	0.30199467892593 99E+02
15	0.49864744388906 47E+01	0.66431867068596 01E+01	-0.18208133008381 55E+03	0.18513169841657 33E+03
16	0.66627884012896 27E+01	0.11979992634857 65E+02	-0.49281164821165 19E+03	0.30537802662866 40E+03
17	0.85029524680167 74E+01	0.18287726286800 51E+02	-0.10321924119327 95E+04	0.10402298456203 65E+04
18	0.96073745410405 77E+01	0.22598461106929 97E+02	-0.15461300758967 06E+04	0.12977252658812 22E+04
19	0.10507497805331 53E+02	0.27601550758456 78E+02	-0.20422636304766 43E+04	0.18677165406394 82E+04
20	0.10676957770340 67E+02	0.31483954149527 13E+02	-0.21382689862347 39E+04	0.27450051358579 46E+04
21	0.11599989389333 32E+02	0.34403837978335 90E+02	-0.27792515531935 42E+04	0.30521096441953 52E+04
22	0.12017300384636 90E+02	0.38981940484898 74E+02	-0.30367606708139 72E+04	0.41575438440771 50E+04
23	0.11504690259682 73E+02	0.35143215833449 66E+02	-0.26562791304194 67E+04	0.38872249925263 28E+04
24	0.13867577055810 11E+02	0.32912823902601 02E+02	-0.48045320684816 80E+04	0.43731776035549 12E+04
25	0.11823453377535 49E+02	0.31940359779053 71E+02	-0.28653922956925 25E+04	0.39802262719695 17E+04
26	0.98986431640089 99E+01	0.27865653053825 17E+02	-0.16494603734559 86E+04	0.21434504993970 09E+04
27	0.93382360970524 30E+01	0.20660927747021 54E+02	-0.14125041350190 07E+04	0.11367534227811 73E+04
28	0.80391498289266 04E+01	0.19181358663676 53E+02	-0.86791921071218 75E+03	0.97985613348114 20E+03
29	0.64843031301271 03E+01	0.10118336376384 92E+02	-0.45155079755541 96E+03	0.22415528613476 00E+03
30	0.52298418531470 19E+01	0.67187950906403 44E+01	-0.22452241979610 23E+03	0.13849752227993 74E+03
31	0.36898936916172 48E+01	0.33261506654947 82E+01	-0.71261532176310 72E+02	0.26466776319711 76E+02
32	0.24450944032305 28E+01	0.24531815993344 63E+01	-0.14597894164896 32E+02	0.15585680627972 97E+02
33	0.18499484522387 45E+01	0.13512027515752 51E+01	-0.52227999893714 43E+01	0.41386537502108 96E+01

N	SUM X	SUM X**2	SUM X**3	SUM X**4
1	0.99999999999999883E+02	0.99999999999999780E+02	0.99999999999999675E+02	0.99999999999999573E+02
2	0.14285950977324444E+03	0.24758706239479915E+03	0.48298499740235537E+03	0.10216122514723944E+04
3	0.19041231243113411E+03	0.44458095346366610E+03	0.11747911590082225E+04	0.33549344962734094E+04
4	0.25532157885240600E+03	0.83794207965547643E+03	0.32520140483620047E+04	0.14213338915810056E+05
5	0.28952891683411145E+03	0.10275997162453150E+04	0.42096788712404560E+04	0.19234058713404750E+05
6	0.35933196779491411E+03	0.16064412941850590E+04	0.83322902490626900E+04	0.48691060769618080E+05
7	0.39304643411724160E+03	0.18698765266503920E+04	0.10018701109193120E+05	0.58071860704839720E+05
8	0.31763748833679920E+03	0.13203339489124720E+04	0.65492476099443330E+04	0.36292885239549590E+05
9	0.29091093956305610E+03	0.10592709425006010E+04	0.44565580888241180E+04	0.20692790884319620E+05
10	0.23101136641512720E+03	0.67710720160290540E+03	0.23232736593496920E+04	0.88383977878370770E+04
11	0.19296351101333160E+03	0.46882033100736620E+03	0.12979393057603450E+04	0.39004921103385550E+04
12	0.17363337975928570E+03	0.36883235053377970E+03	0.86994179873160320E+03	0.21955756456003350E+04
13	0.26360688597277890E+03	0.87252735761400850E+03	0.33238958759298710E+04	0.13879818397410760E+05
14	0.38034813443652380E+03	0.18126045487306200E+04	0.98798856632766860E+04	0.58783479590194760E+05
15	0.49864744388906470E+03	0.31441682169500800E+04	0.23853824861858350E+05	0.21070141958782380E+06
16	0.66627884012896270E+03	0.56252941988868660E+04	0.54280720149318590E+05	0.57006281976496360E+06
17	0.85029524680167740E+03	0.90405049697285060E+04	0.11293367080546270E+06	0.15915152883752900E+07
18	0.96073745410405770E+03	0.11467412206769530E+05	0.15744183555676920E+06	0.23852952868476710E+07
19	0.10507497805331530E+04	0.13773304537991910E+05	0.20786103654186460E+06	0.34560571213102050E+07
20	0.10676957770340670E+04	0.14516654183766970E+05	0.22835472699075590E+06	0.39964922714398940E+07
21	0.11599989389333320E+04	0.16861955343119830E+05	0.28195840310492360E+06	0.52063237462285110E+07
22	0.12017300384636900E+04	0.18300762961464390E+05	0.32721812853954820E+06	0.65441363459275880E+07
23	0.11504690259682730E+04	0.16714968164635380E+05	0.28480210166756840E+06	0.54763980002663830E+07
24	0.13867577055810110E+04	0.22489338906240600E+05	0.41670277512162490E+06	0.86973473675682030E+07
25	0.11823453377535490E+04	0.17141500595201870E+05	0.29351715980169490E+06	0.57646026261708820E+07
26	0.98986431640090000E+03	0.12557013301166900E+05	0.18834951697862240E+06	0.31699189064613290E+07
27	0.93382360970524300E+03	0.10765697187384430E+05	0.14290692207274810E+06	0.21001763204541310E+07
28	0.80391498289266040E+03	0.83617475048970420E+04	0.10194651600495660E+06	0.13868550867520770E+07
29	0.64843031301271030E+03	0.52063340095997220E+04	0.47714026572586810E+05	0.47690996779388310E+06
30	0.52298418531470190E+03	0.34002852948662190E+04	0.25426371901799220E+05	0.21216832867874190E+06
31	0.36898936916172480E+03	0.16908204614276590E+04	0.68676269541194260E+04	0.51015801382700730E+05
32	0.24450944032305290E+03	0.84071364240503730E+03	0.35113859206608430E+04	0.16466767386817020E+05
33	0.18499484522387450E+03	0.47600000000000270E+03	0.14349845356716930E+04	0.47720000000009890E+04

```
MODULATION TECHNIQUE = OQPSK
BIPOLAR LOGIC
BAUD OR SYMBOL RATE =        1200 HZ
BITS PER BINARY CODE WORD =           2
BIT RATE =  0.2400000000000000E+04
CARRIER FREQUENCY =        2400 HZ
MAXIMUM CARRIER AMPLITUDE =           1 VOLT(S)
INITIAL PHASE ANGLE =           0 DEGREES
TIME BETWEEN SAMPLES =  0.1388888888888889E-03 SEC
NUMBER OF SAMPLES GENERATED =          64
SEED FOR RANDOM NUMBER GENERATOR =           1
NUMBER OF TIMES SIMULATION REPEATS =         100
   SUM OF VARIANCES         SUM OF VARIANCES**2
 0.4302799744926702E+03   0.1192673508267923E+05
   SUM OF SKEWNESS           SUM OF SKEWNESS**2
-0.2766466990808353E+05   0.6316161942013623E+08
   SUM OF KURTOSIS           SUM OF KURTOSIS**2
 0.3332128031283272E+05   0.9992700833249297E+08
    MEAN VARIANCE         VARIANCE OF THE VARIANCES
 0.1303878710583849E+02   0.1973876906051148E+03
    MEAN SKEWNESS         VARIANCE OF THE SKEWNESS
-0.8383233305479857E+03   0.1249052537633781E+07
    MEAN KURTOSIS         VARIANCE OF THE KURTOSIS
 0.1009735767055537E+04   0.2071291243641945E+07
```

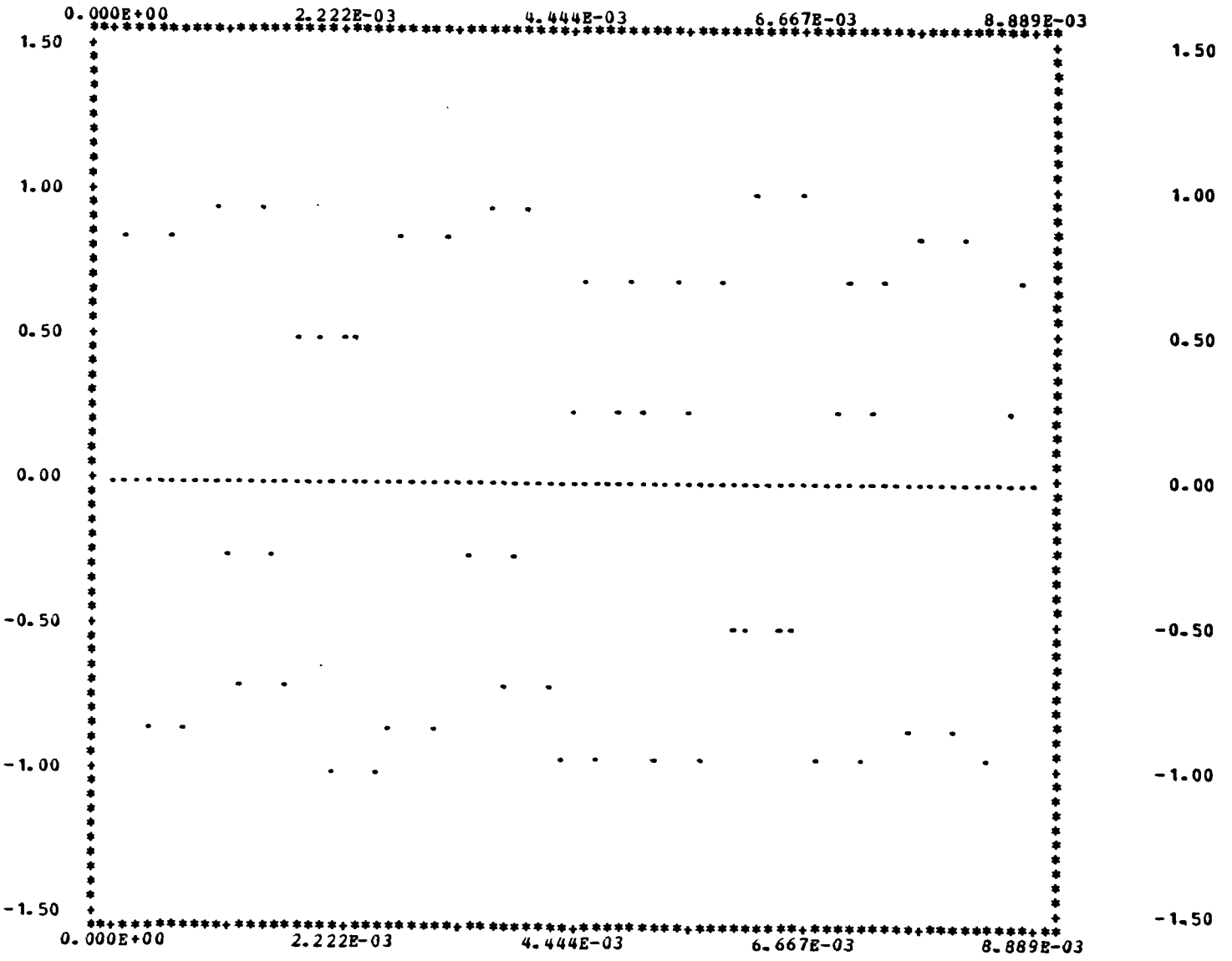

4.167E-04
2.639E-03
4.861E-03
7.083E-03
9.306E-03
2.05
1.37
0.68
0.00
-0.68
-1.37
-2.05

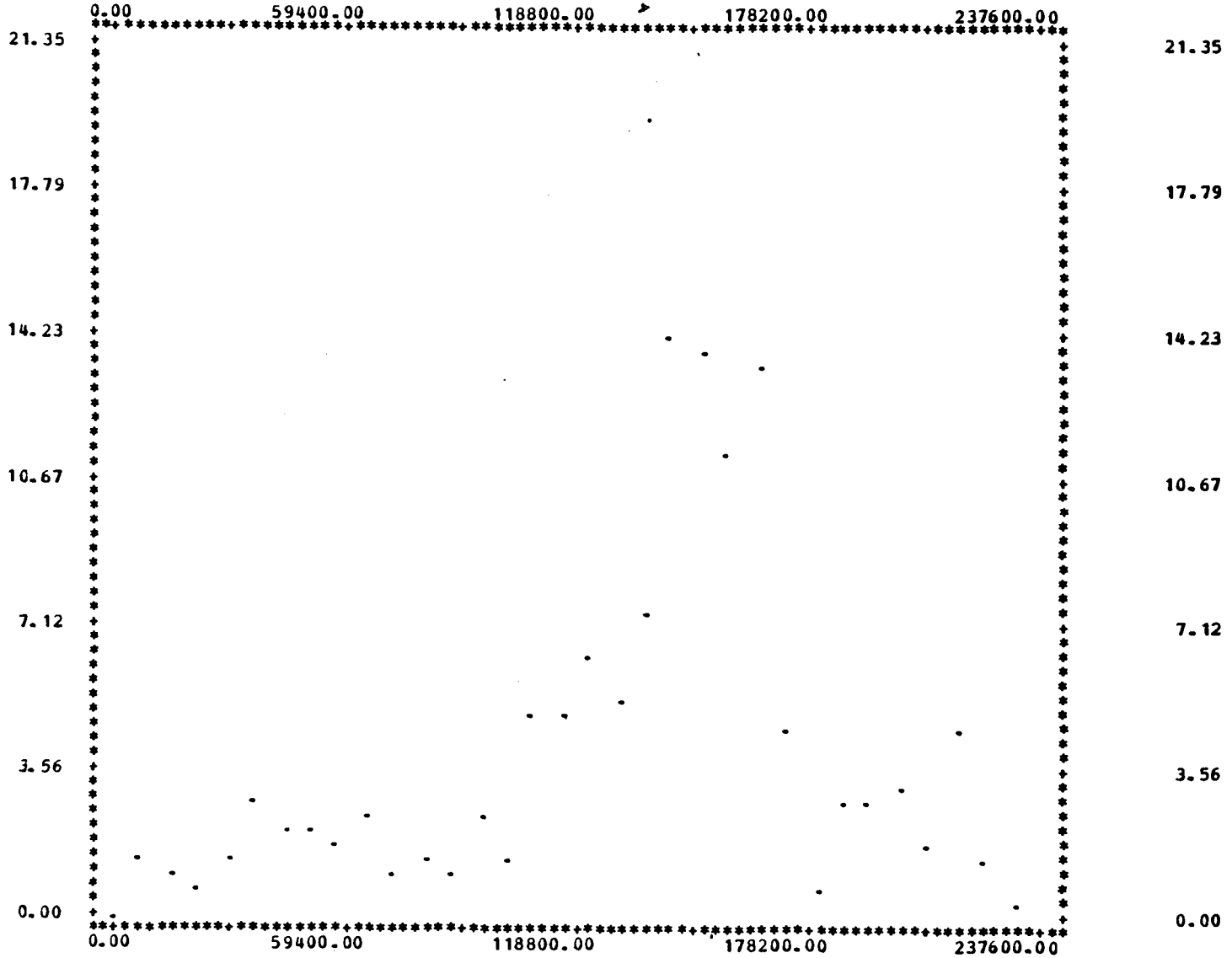

0.00
59400.00
118800.00
178200.00
237600.00
30.57
25.47
20.38
15.28
10.19
5.09
0.00

N	MEAN	VARIANCE	SKEWNESS	KURTOSIS
1	0.9999999999999913E+00	0.1561758578801998E-14	0.1598721155460225E-13	0.2664535259100376E-14
2	0.1372323826180889E+01	0.4068678643542643E+00	-0.1386376962675256E+01	0.4041815975386704E+00
3	0.2001396615374287E+01	0.9269424591664253E+00	-0.7733658389670000E+01	0.1862322872670187E+01
4	0.2772862445069488E+01	0.1678068152390965E+01	-0.2623023671424697E+02	0.8004509154649167E+01
5	0.3251532646964970E+01	0.2932577358584998E+01	-0.4366853012949963E+02	0.2588931213699834E+02
6	0.3777641084824826E+01	0.4020832039397192E+01	-0.7125587489602146E+02	0.7882832999429900E+02
7	0.3829098409125102E+01	0.3882607022946678E+01	-0.7927245839330759E+02	0.3918838665086150E+02
8	0.2961659942016187E+01	0.3191184577366257E+01	-0.3117942587408732E+02	0.2612505891404616E+02
9	0.2968246427181287E+01	0.2173213541787000E+01	-0.3417545686971072E+02	0.1113594383145398E+02
10	0.2148359327285216E+01	0.1060831343738275E+01	-0.1029303883092354E+02	0.2522974975096634E+01
11	0.1345271169743835E+01	0.4197276401107519E+00	-0.1125305770333078E+01	0.4717607434503110E+00
12	0.1227933909092941E+01	0.5054447847122679E+00	-0.7207547009699091E+00	0.3665403064081190E+00
13	0.2439525645946195E+01	0.1781420668748019E+01	-0.1588289360955499E+02	0.9490600068457177E+01
14	0.3624460779686187E+01	0.2958165706320683E+01	-0.6557803973417343E+02	0.2992394022408189E+02
15	0.5344290929937282E+01	0.8707443216668699E+01	-0.2327526112313629E+03	0.2336896609052334E+03
16	0.6817466947394810E+01	0.8177726358497095E+01	-0.5351265563927940E+03	0.1713281150266957E+03
17	0.7712888151201321E+01	0.2474902892394821E+02	-0.6993895599099603E+03	0.1906132600582511E+04
18	0.9428086215184937E+01	0.1931615055428528E+02	-0.1470236620891471E+04	0.8926570202434632E+03
19	0.1072214978861974E+02	0.2805072869446679E+02	-0.2185062282026585E+04	0.1838937522132344E+04
20	0.1081933826880679E+02	0.3440851415469506E+02	-0.2210060342212764E+04	0.2931776835225231E+04
21	0.1199506255571578E+02	0.3580001811338520E+02	-0.3041862605032027E+04	0.3576329272194000E+04
22	0.1128875410898178E+02	0.3443719159620903E+02	-0.2585353070858207E+04	0.2812818126869951E+04
23	0.1122041374376493E+02	0.3976866880087194E+02	-0.2372557155779416E+04	0.5281536943398933E+04
24	0.1239679867770296E+02	0.3228274309630087E+02	-0.3399416891619934E+04	0.3179717696454303E+04
25	0.1194093966023447E+02	0.3512070429981194E+02	-0.3060399165727626E+04	0.3009353203186677E+04
26	0.9561857571851002E+01	0.2543911177410117E+02	-0.1477814600152428E+04	0.1874123783999903E+04
27	0.1060742458462544E+02	0.3139753080593636E+02	-0.2076204826664873E+04	0.2806095691376629E+04
28	0.8728674900290654E+01	0.2466868494522429E+02	-0.1082343896000136E+04	0.2016793419812381E+04
29	0.6834317564181064E+01	0.1211209931277713E+02	-0.5263413260618458E+03	0.4487766787298260E+03
30	0.5256141872574309E+01	0.5747778732450851E+01	-0.2337931678447483E+03	0.7781670547209961E+02
31	0.3639693642400281E+01	0.2758452726478166E+01	-0.6782334107449190E+02	0.2464070669964832E+02
32	0.2386927083876014E+01	0.1399515226938646E+01	-0.1523753288228533E+02	0.4542469052879014E+01
33	0.1732050807568335E+01	0.0000000000000000E+00	-0.4392304845407262E+01	0.3552713678800501E-13

N	SUM X	SUM X**2	SUM X**3	SUM X**4
1	0.999999999999913E+02	0.999999999999842E+02	0.999999999999767E+02	0.999999999999698E+02
2	0.137232382618088 9E+03	0.228607186961447 7E+03	0.425877035461874 7E+03	0.859019748194300 2E+03
3	0.200139661537428 7E+03	0.492326144660641 2E+03	0.138153611737890 6E+04	0.422732528264830 0E+04
4	0.277286244506948 8E+03	0.935005361014379 7E+03	0.361714685520619 2E+04	0.155207656723637 6E+05
5	0.325153264696497 0E+03	0.134757161392781 7E+04	0.666367336662294 8E+04	0.373077205046611 4E+05
6	0.377764108482482 6E+03	0.182511958847597 1E+04	0.107042383036827 8E+05	0.744516787809469 5E+05
7	0.382909840912510 2E+03	0.185057755794816 0E+04	0.103984859846221 1E+05	0.648794198934911 3E+05
8	0.296165994201618 7E+03	0.119307023437359 2E+04	0.572817645363099 3E+04	0.307639525988092 8E+05
9	0.296824642718128 7E+03	0.109619682588436 1E+04	0.458170387829019 5E+04	0.208513362919126 6E+05
10	0.214835932728521 6E+03	0.566567082943427 6E+03	0.169917558860383 7E+04	0.555499913516082 8E+04
11	0.134527116974383 5E+03	0.222528488385359 0E+03	0.423601998552400 9E+03	0.892846099395904 3E+03
12	0.122793390902941 E+03	0.200821202196541 6E+03	0.366145684408784 4E+03	0.700307326073644 3E+03
13	0.243952564594619 5E+03	0.771489183928973 8E+03	0.286765651289315 9E+04	0.120091719330403 9E+05
14	0.362446077968618 7E+03	0.160652999927408 8E+04	0.828326775916060 8E+04	0.482267327083166 0E+05
15	0.534429092993728 2E+03	0.371818143283119 1E+04	0.306255796901245 5E+05	0.285604029509266 3E+06
16	0.681746694739481 0E+03	0.545738046737328 4E+04	0.488083061119826 7E+05	0.474302136763948 5E+06
17	0.771288815120132 1E+03	0.839901822676504 6E+04	0.112505379771235 8E+06	0.172537130582862 2E+07
18	0.942808621518493 7E+03	0.108011798729702 7E+05	0.140701941178912 7E+06	0.200520784393878 0E+07
19	0.107221497886197 4E+04	0.142734717497120 5E+05	0.217627778891556 3E+06	0.363704001138477 4E+07
20	0.108193382668806 79E+04	0.151122509588015 1E+05	0.246096015042869 8E+06	0.444027354288283 4E+07
21	0.119950625557157 8E+04	0.179323543647786 2E+05	0.312336571484037 6E+06	0.607338077938786 9E+07
22	0.112887541088178 8E+04	0.161528789013299 9E+05	0.263015133255390 4E+06	0.467896567982571 4E+07
23	0.112204137437649 3E+04	0.165268666694132 1E+05	0.293879593262367 8E+06	0.598885881585495 1E+07
24	0.123967986777029 6E+04	0.185640533312083 8E+05	0.319726662003042 0E+06	0.614002960595630 7E+07
25	0.119409396602344 7E+04	0.177355537226174 4E+05	0.300780405961230 7E+06	0.559351744098619 7E+07
26	0.956185757185100 2E+03	0.116613840880724 5E+05	0.168446197162279 5E+06	0.274069124241772 8E+07
27	0.106074245846254 4E+04	0.143601011816393 1E+05	0.226847097005298 3E+06	0.400914451234662 9E+07
28	0.872867490029065 4E+03	0.100611763610736 1E+05	0.139989870007896 2E+06	0.223149570475091 4E+07
29	0.683431756418106 4E+03	0.586988748872316 E+04	0.583743135730506 1E+05	0.650139748463551 7E+06
30	0.525614187257431 0E+03	0.333173283297553 1E+04	0.236314590934993 2E+05	0.181323424496701 7E+06
31	0.363969364240028 1E+03	0.159782380097424 1E+04	0.801495932069039 9E+04	0.447981219720690 6E+05
32	0.238692708387601 4E+03	0.708294097841011 4E+03	0.240871C16274820 10E+04	0.897731140384116 0E+04
33	0.173205080756833 5E+03	0.299999999999812 1E+03	0.519615242270174 8E+03	0.899999999998874 2E+03

```
MODULATION TECHNIQUE = MPSK
BIPOLAR LOGIC
BAUD OR SYMBOL RATE =        1200 HZ
BITS PER BINARY CODE WORD =           3
BIT RATE =  0.3600C00000000000E+04
CARRIER FREQUENCY =        2400 HZ
MAXIMUM CARRIER AMPLITUDE =            1 VOLT(S)
INITIAL PHASE ANGLE =            0 DEGREES
TIME BETWEEN SAMPLES =  0.1388888888888889E-03 SEC
NUMBER OF SAMPLES GENERATED =           64
SEED FOR RANDOM NUMBER GENERATOR =            1
NUMBER OF TIMES SIMULATION REPEATS =          100
   SUM OF VARIANCES          SUM OF VARIANCES**2
 0.2100020931141337E+03    0.2899383925937582E+04
   SUM OF SKEWNESS            SUM OF SKEWNESS**2
-0.9600995280888690E+04    0.8204154937490084E+07
   SUM OF KURTOSIS            SUM OF KURTOSIS**2
 0.8272550682771584E+04    0.7152001918873632E+07
    MEAN VARIANCE          VARIANCE OF THE VARIANCES
 0.6363699791337385E+01    0.4884355155646109E+02
    MEAN SKEWNESS          VARIANCE OF THE SKEWNESS
-0.2909392509360209E+03    0.1690890175696266E+06
    MEAN KURTOSIS          VARIANCE OF THE KURTOSIS
 0.2506833540233813E+03    0.1586940989808761E+06
```

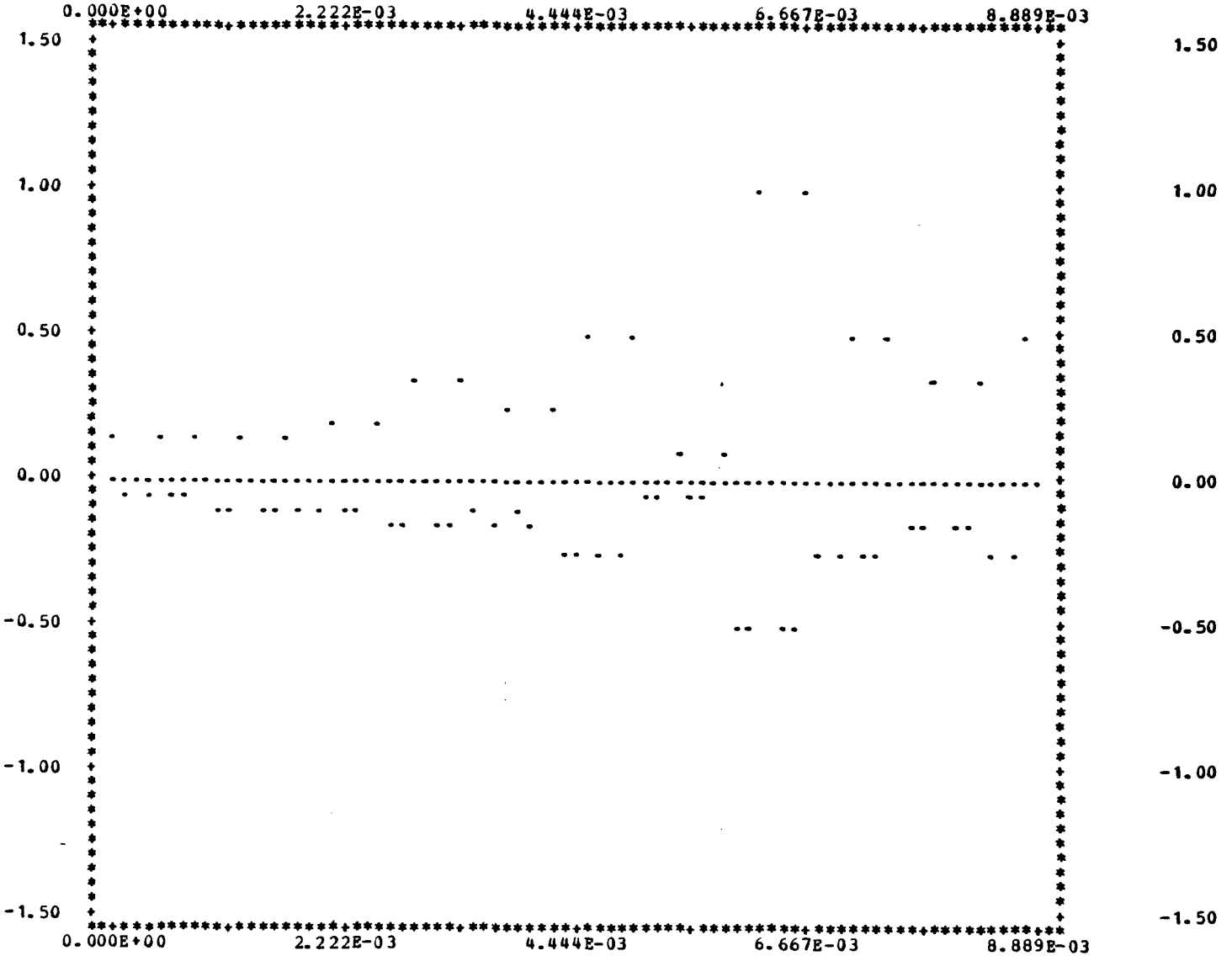

0.000E+00
2.222E-03
4.444E-03
6.667E-03
8.889E-03
1.50
1.00
0.50
0.00
-0.50
-1.00
-1.50

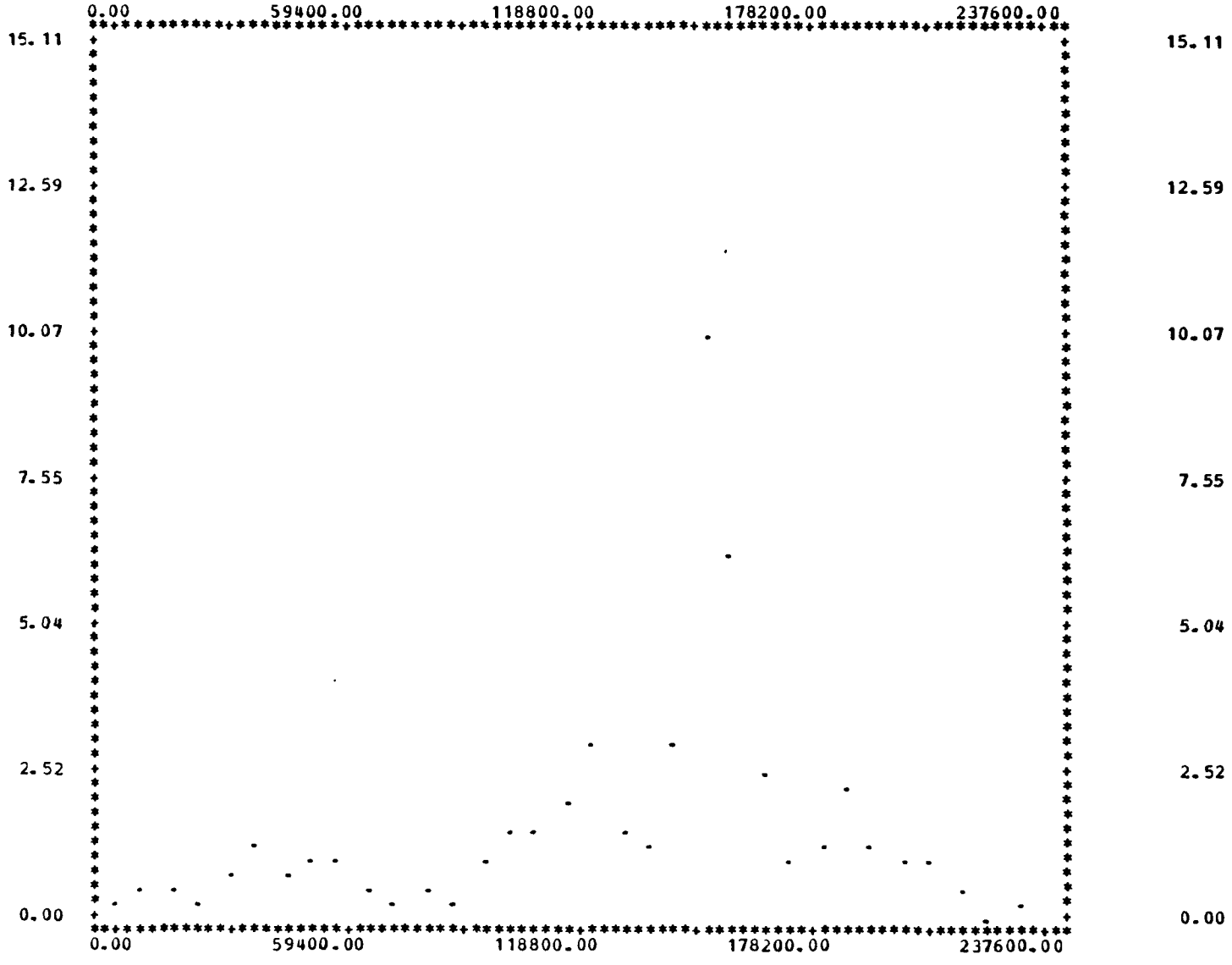

0.00
59400.00
118800.00
178200.00
237600.00
21.35
17.79
14.23
10.67
7.12
3.56
0.00

N	MEAN	VARIANCE	SKEWNESS	KURTOSIS
1	0.58526911934583 17E+00	0.14894955347491 13E+00	0.24736883835421 96E+00	0.40646767757362 67E-01
2	0.89913368041470 69E+00	0.29591761160675 64E+00	0.27896957897845 79E+00	0.25586962625549 49E+00
3	0.12795920767328 64E+01	0.39298090535834 19E+00	-0.77969039778050 32E+00	0.56742713160670 37E+00
4	0.17373563021597 23E+01	0.69367429702186 30E+00	-0.43455934694826 51E+01	0.10759332271332 94E+01
5	0.21219751234316 36E+01	0.15238309490253 44E+01	-0.89493560797972 02E+01	0.64714266781946 21E+01
6	0.24242378006752 34E+01	0.14778479469437 27E+01	-0.15743796060 14421E+02	0.66723493983452 01E+01
7	0.24934079778801 65E+01	0.12469250635507 47E+01	-0.18132011916003 08E+02	0.43412977770742 92E+01
8	0.24788045460556 75E+01	0.20981453969373 12E+01	-0.16949304951768 07E+02	0.93597541436377 26E+01
9	0.20715933292319 43E+01	0.10380709141716 32E+01	-0.89595183974069 67E+01	0.26306866558446 20E+01
10	0.17922658540827 27E+01	0.81720589746594 58E+00	-0.48584090335329 32E+01	0.17084122038265 55E+01
11	0.13384219659154 24E+01	0.46024103194660 23E+00	-0.10765733541483 09E+01	0.49301092317833 48E+00
12	0.12067941094599 43E+01	0.34651460257074 62E+00	-0.58249249614945 71E+00	0.27843834783265 26E+00
13	0.17010792147250 41E+01	0.82400341025200 97E+00	-0.36762387523147 77E+01	0.18452164454106 00E+01
14	0.28341506105514 14E+01	0.18238550473152 13E+01	-0.28413760758618 41E+02	0.10256882334266 90E+02
15	0.37438075536188 62E+01	0.33670911282480 24E+01	-0.75362189787637 67E+02	0.27644321216362 7E+02
16	0.46951950210295 40E+01	0.58785722815421 08E+01	-0.16042688459579 75E+03	0.77338630124472 96E+02
17	0.56106482695289 47E+01	0.78322475116346 61E+01	-0.28703891968471 28E+03	0.15769594260406 05E+03
18	0.67068908951556 74 7E+01	0.13368435093586 19E+02	-0.48833806901795 77E+03	0.47171878081352 41E+03
19	0.72335281979043 89E+01	0.11790796859933 31E+02	-0.63917234541857 12E+03	0.38943719816628 16E+03
20	0.78384759817572 54E+01	0.15505649018109 32E+02	-0.81483087308853 05E+03	0.83691448207366 28E+03
21	0.83456234815321 00E+01	0.21432643379950 21E+02	-0.96004189625797 80E+03	0.14239312466642 69E+04
22	0.89441334160904 30E+01	0.18495434863231 11E+02	-0.12170759952864 70E+04	0.12632906908733 13E+04
23	0.88147839780466 04E+01	0.20371750203055 52E+02	-0.11969201889774 85E+04	0.94637139331380 24E+03
24	0.87495097537322 53E+01	0.18386598267241 27E+02	-0.11680603814407 85E+04	0.84023284931043 33E+03
25	0.82859821750526 35E+01	0.16664039238108 72E+02	-0.97234276668023 74E+03	0.63723868432786 55E+03
26	0.72484188015960 32E+01	0.13769326557321 41E+02	-0.63403855678204 26E+03	0.51863325154605 35E+03
27	0.60820180007236 16E+01	0.81797635421921 11E+01	-0.36743325944301 92E+03	0.16110044467803 80E+03
28	0.55744269420614 57E+01	0.85311450869948 27E+01	-0.26676893074706 44E+03	0.32777377781942 465E+03
29	0.45409600616799 76E+01	0.62929888963135 68E+01	-0.13619456004358 81E+03	0.10561825021269 08E+03
30	0.36008233673474 76E+01	0.28389801508144 18E+01	-0.65103625432273 04E+02	0.22285527375175 83E+02
31	0.29796329311117 14E+01	0.24943504991859 18E+01	-0.33373095247351 76E+02	0.16658436662459 58E+02
32	0.17888772688513 79E+01	0.89799177669100 76E+00	-0.47842118374383 46E+01	0.17439157932192 37E+01
33	0.14388925829815 57E+01	0.71612613233885 49E+00	-0.17481238699412 82E+01	0.92539625601841 92E+00

N	SUM X	SUM X**2	SUM X**3	SUM X**4
1	0.585269119345831 7E+02	0.490000000000007 6E+02	0.422634559672914 7E+02	0.375000000000009 0E+02
2	0.899133680414706 9E+02	0.110139981074678 5E+03	0.163300382480072 1E+03	0.247252388818925 1E+03
3	0.127959207673286 4E+03	0.202640697914228 3E+03	0.372452078078437 8E+03	0.776600278573921 6E+03
4	0.173735630215972 4E+03	0.370514447470575 3E+03	0.892906099983608 2E+03	0.233583134979753 7E+04
5	0.212197512343163 6E+03	0.601137106399779 9E+03	0.203130266378376 6E+04	0.773041845555456 5E+04
6	0.242423780067523 4E+03	0.733999838169698 9E+03	0.258840507128250 5E+04	0.102463714158869 4E+05
7	0.249340797788016 5E+03	0.745153915707169 2E+03	0.251730029408709 8E+04	0.934028992509511 0E+04
8	0.247880454605567 5E+03	0.822163592051422 3E+03	0.319012365804952 3E+04	0.135825413695183 6E+05
9	0.207159332923194 3E+03	0.531918912674820 1E+03	0.155150738948404 3E+04	0.494815035624074 2E+04
10	0.179226585408272 7E+03	0.402125073020217 4E+03	0.103386083063850 6E+04	0.292787857389238 4E+04
11	0.133842196591542 4E+03	0.224701198047204 3E+03	0.436303050517858 1E+03	0.932694552693879 9E+03
12	0.120679410945994 3E+03	0.179940147917225 7E+03	0.301932477545187 2E+03	0.549276350124541 9E+03
13	0.170107921472504 1E+03	0.370943387091905 1E+03	0.946654282679844 1E+03	0.269752841267316 0E+04
14	0.283415061055141 5E+03	0.983802618013101 7E+03	0.391687635899210 4E+04	0.173718058744116 6E+05
15	0.374380755361886 3E+03	0.173495152154992 0E+04	0.914653585615288 5E+04	0.527677269620183 4E+05
16	0.469519502102954 0E+03	0.278646428442272 7E+04	0.187973199868148 2E+05	0.137991674281452 3E+06
17	0.561064826952894 7E+03	0.392332990408864 9E+04	0.310375056421535 3E+05	0.268593183499076 9E+06
18	0.670690895155674 8E+03	0.582173784271223 6E+04	0.593072645140199 1E+05	0.674014214015041 9E+06
19	0.723352319790438 9E+03	0.639968190812119 0E+04	0.644948180402068 7E+05	0.717243104233642 7E+06
20	0.783847598175725 5E+03	0.767922982445135 9E+04	0.868089471598977 3E+05	0.110706592385058 7E+07
21	0.834562348153210 0E+03	0.908677482416506 8E+04	0.117570328146942 3E+06	0.172516468098043 9E+07
22	0.894413341609043 0E+03	0.983080030794242 7E+04	0.126076864581927 7E+06	0.183816532726326 4E+07
23	0.881478397804660 4E+03	0.978684492806520 8E+04	0.123574669388933 6E+06	0.170032805614830 7E+07
24	0.874950975373225 3E+03	0.947566532152246 9E+04	0.116605456131080 2E+06	0.157074042818218 3E+07
25	0.828598217505263 6E+03	0.851548994510176 5E+04	0.100711816901777 8E+06	0.130795132754259 5E+07
26	0.724841880159603 2E+03	0.661712084150790 5E+04	0.699792186571861 4E+05	0.822975843024257 4E+06
27	0.608201800072361 7E+03	0.450889088678962 9E+04	0.381279520737324 3E+05	0.353459812519402 3E+06
28	0.557442694206145 7E+03	0.395200693685055 3E+04	0.331987816107979 1E+05	0.325882975314092 7E+06
29	0.454096006187997 6E+03	0.268503772909394 3E+04	0.188344276162928 8E+05	0.148028621767871 9E+06
30	0.360082336734747 6E+03	0.157765192721418 9E+04	0.793898944736646 8E+04	0.442764049285311 0E+05
31	0.297963293111117 14E+03	0.113476193983594 4E+04	0.503057012915391 6E+04	0.248217048105069 2E+05
32	0.178887726885137 9E+03	0.408909374193726 6E+03	0.107602849319973 2E+04	0.309481956429468 0E+04
33	0.143889258298155 7E+03	0.277937673637480 3E+03	0.610872611318361 3E+03	0.144176874758186 1E+04

```
MODULATION TECHNIQUE = MASK
BIPOLAR LOGIC
BAUD OR SYMBOL RATE =        1200 HZ
BITS PER BINARY CODE WORD =           3
BIT RATE =  0.3600000000000000E+04
CARRIER FREQUENCY =        2400 HZ
MAXIMUM CARRIER AMPLITUDE =           1 VOLT(S)
INITIAL PHASE ANGLE =           0 DEGREES
TIME BETWEEN SAMPLES =  0.1388888888888889E-03 SEC
NUMBER OF SAMPLES GENERATED =          64
SEED FOR RANDOM NUMBER GENERATOR =           1
NUMBER OF TIMES SIMULATION REPEATS =         100
   SUM OF VARIANCES          SUM OF VARIANCES**2
 0.267201139977162 5E+02    0.623646461209102 6E+02
   SUM OF SKEWNESS            SUM OF SKEWNESS**2
-0.1657760855944905E+04     0.1880354503890174E+07
   SUM OF KURTOSIS            SUM OF KURTOSIS**2
 0.1906467874763258E+03     0.7184370776848387E+04
    MEAN VARIANCE          VARIANCE OF THE VARIANCES
 0.8097004241732196E+00     0.1272792452593831E+01
    MEAN SKEWNESS          VARIANCE OF THE SKEWNESS
-0.5023517745287592E+02     0.5615864353491720E+05
    MEAN KURTOSIS          VARIANCE OF THE KURTOSIS
 0.5777175378070478E+01     0.1900928390728725E+03
```

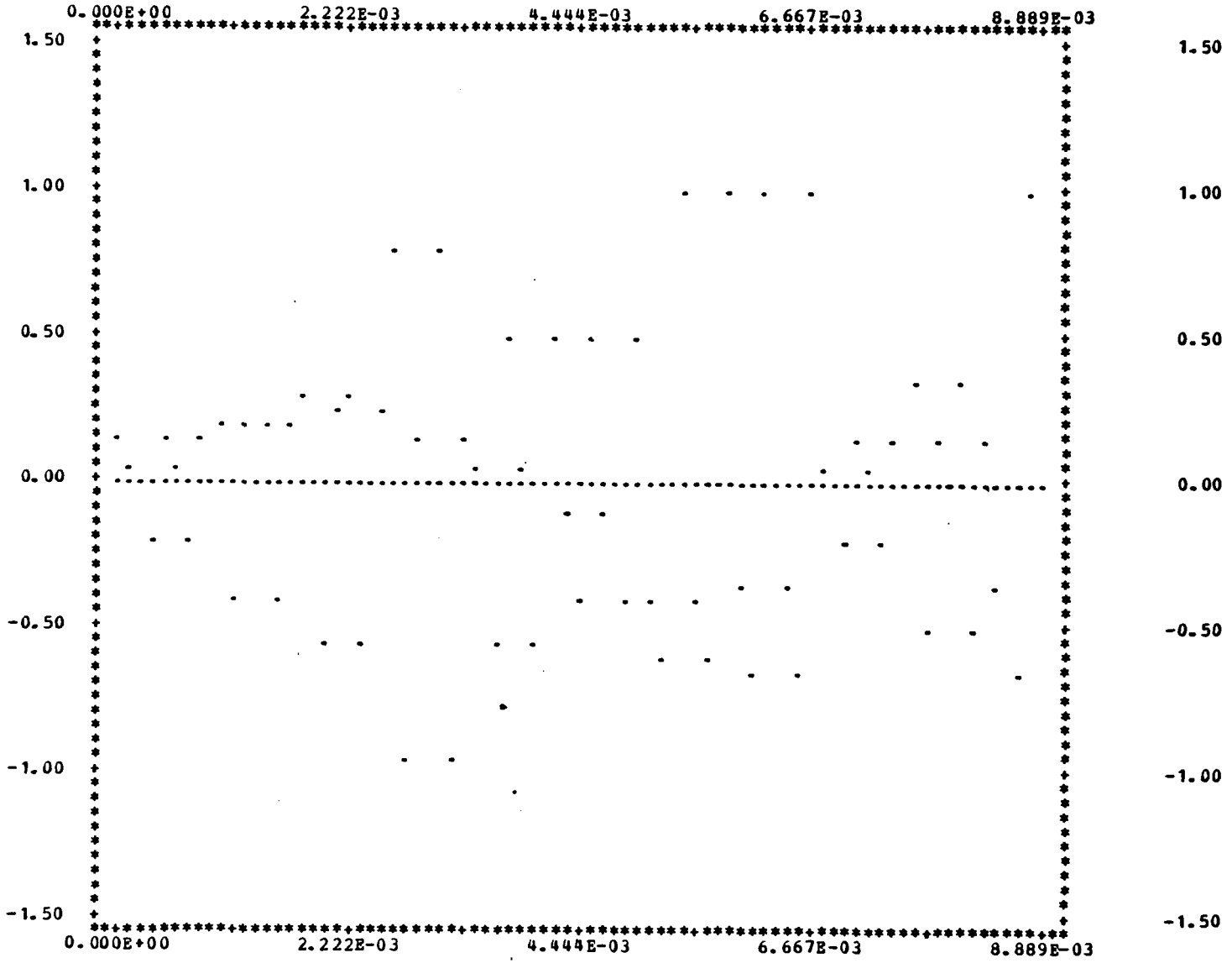

0.000E+00
2.222E-03
4.444E-03
6.667E-03
8.889E-03
1.50
1.00
0.50
0.00
-0.50
-1.00
-1.50

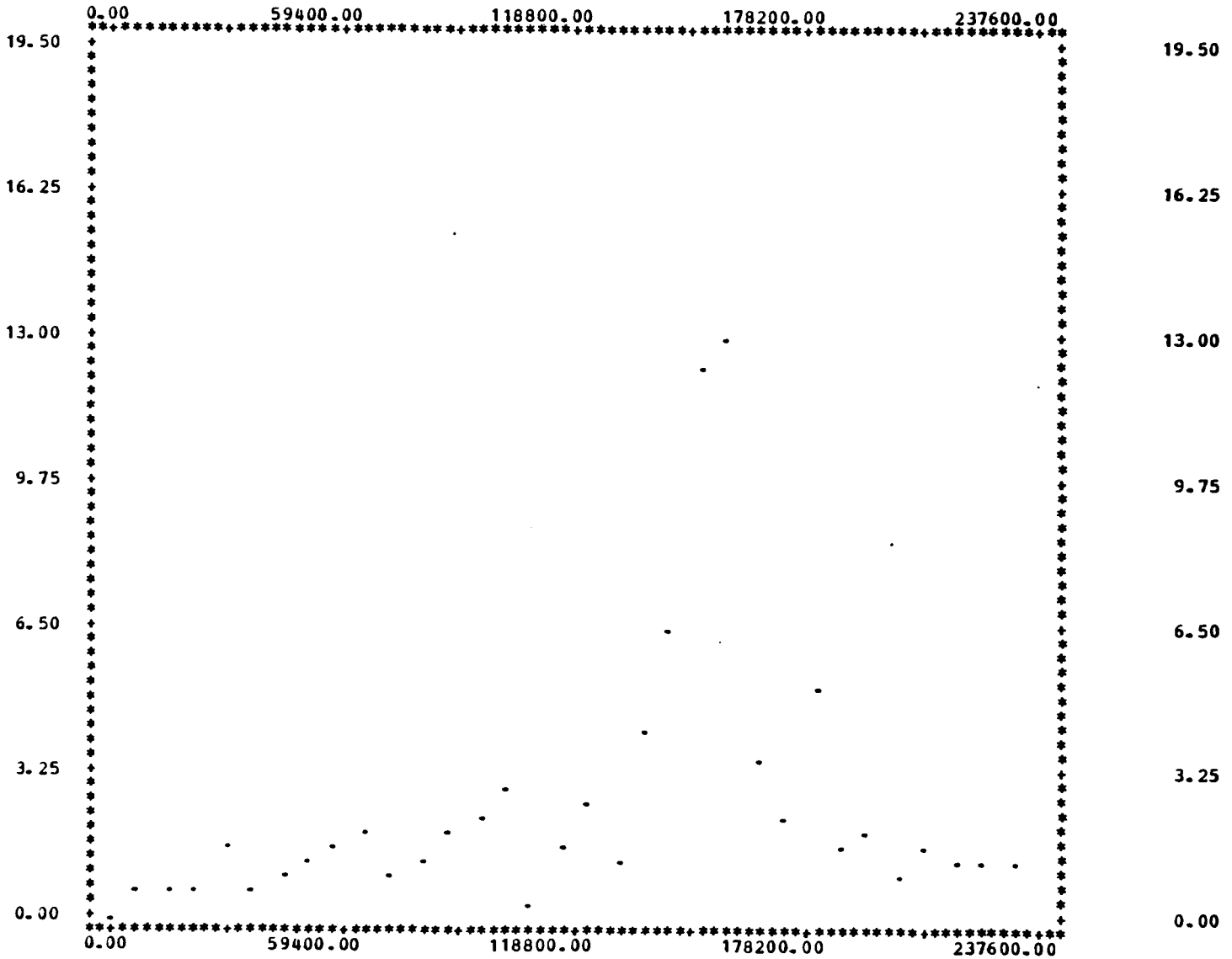

0.00
59400.00
118800.00
178200.00
237600.00
15.11
12.59
10.07
7.55
5.04
2.52
0.00

N	MEAN	VARIANCE	SKEWNESS	KURTOSIS
1	0.33935714285709 85E+00	0.83844648869647 83E-01	0.18933827235784 15E+00	0.27427360318874 71E-01
2	0.40040126106894 17E+00	0.76681901148222 22E-01	0.22152588931749 14E+00	0.28914998702946 14E-01
3	0.56796133698667 36E+00	0.77847615050694 71E-01	0.28834427315017 78E+00	0.15564754462154 72E-01
4	0.65693557725572 77E+00	0.14570172339343 83E+00	0.33098024323252 56E+00	0.61048568178966 56E-01
5	0.71610050214346 23E+00	0.15862389975804 87E+00	0.34833306874744 83E+00	0.10905889687572 79E+00
6	0.84135045336752 61E+00	0.25063688167973 36E+00	0.33415000811506 26E+00	0.21199298275628 88E+00
7	0.87505194145758 49E+00	0.15821414030876 46E+00	0.21531149739899 20E+00	0.85531778826533 51E-01
8	0.88106725680796 62E+00	0.24755484199039 26E+00	0.29670832926595 43E+00	0.23693403442060 32E+00
9	0.71932200638079 60E+00	0.14551121032059 57E+00	0.31805833661194 36E+00	0.55360739921602 95E-01
10	0.64006957405931 27E+00	0.18574087528978 42E+00	0.40254410932714 51E+00	0.18524651582146 89E+00
11	0.50962013152453 69E+00	0.16264023820728 94E+00	0.33660010135720 22E+00	0.97611985036641 66E-01
12	0.46822894217285 21E+00	0.15746343690964 20E+00	0.31721257277034 93E+00	0.98591665373933 66E-01
13	0.66488695354793 73E+00	0.15805941532395 95E+00	0.36272254214980 26E+00	0.10485373655261 8E+00
14	0.83762509124442 08E+00	0.26464978319131 86E+00	0.33750162417854 87E+00	0.22589431551391 70E+00
15	0.12101256088888 72E+01	0.42299145309433 88E+00	-0.50960432856921 29E+00	0.46255656901608 02E+00
16	0.15175800383207 35E+01	0.70628788597780 58E+00	-0.18660155645931 58E+01	0.16817195719382 55E+01
17	0.18091984270295 23E+01	0.11140077246853 03E+01	-0.43052839607341 15E+01	0.40616047444655 62E+01
18	0.21843659834292 02E+01	0.13575042088944 86E+01	-0.97444914835835 19E+01	0.74779738810210 72E+01
19	0.21877980640347 54E+01	0.12424718783170 58E+01	-0.10560720774637 51E+02	0.45068699982497 94E+01
20	0.27548177828815 66E+01	0.23043394347172 48E+01	-0.23310728748453 57E+02	0.18406765773411 28E+02
21	0.32844822831451 03E+01	0.35665536843678 36E+01	-0.43414074815221 82E+02	0.44003061983378 02E+02
22	0.91476946734243 42E+01	0.49314576177330 18E+01	-0.13595558090564 29E+04	0.65079449264572 76E+02
23	0.47702824097327 31E+01	0.23786168240976 82E+01	-0.16925167567609 91E+03	0.17542237057265 94E+02
24	0.26827853739686 61E+01	0.22793308049254 13E+01	-0.22249171613742 69E+02	0.15202166750067 41E+02
25	0.22643387411059 86E+01	0.13176003600274 86E+01	-0.11966878071532 50E+02	0.55621616977919 15E+01
26	0.17300423065581 64E+01	0.84979065801793 76E+00	-0.39195609529084 56E+01	0.18793821107694 63E+01
27	0.15205480024683 91E+01	0.76651044754714 58E+00	-0.19335186081753 82E+01	0.19728539040937 68E+01
28	0.12681845573813 21E+01	0.51518659144198 95E+00	-0.66049730576000 12E+00	0.78697559544041 78E+00
29	0.93741323887142 57E+00	0.28065028233862 08E+00	0.27139843445703 48E+00	0.34350980707651 37E+00
30	0.71999883428307 23E+00	0.17630862874400 97E+00	0.32501954339392 74E+00	0.78461168891302 17E-01
31	0.53159974818724 89E+00	0.84314791107430 91E-01	0.27074806444697 04E+00	0.16693168800371 37E-01
32	0.36819073295986 34E+00	0.63962531046508 91E-01	0.19205439542968 89E+00	0.17788177932436 78E-01
33	0.26033333333342 80E+00	0.89057579193350 87E-01	0.12862370982625 13E+00	0.20523922278518 67E-01

N	SUM X	SUM X**2	SUM X**3	SUM X**4
1	0.33935714285709 85E+02	0.19816947278908 45E+02	0.16076240980317 89E+02	0.14850751766200 11E+02
2	0.40040126106894 17E+02	0.23623625201835 49E+02	0.18465142922348 47E+02	0.17452025797922 61E+02
3	0.56796133696667 36E+02	0.39964921921187 76E+02	0.32414002713457 26E+02	0.29062122845360 81E+02
4	0.65693557725572 77E+02	0.57580905882382 02E+02	0.60265990724640 39E+02	0.71243489349656 93E+02
5	0.71610050214346 23E+02	0.66983758993058 72E+02	0.76174631391879 06E+02	0.10189418585277 45E+03
6	0.84135045336752 61E+02	0.95600109824473 75E+02	0.13314047096357 67E+03	0.21355967617326 65E+03
7	0.87505194145758 49E+02	0.92234789915436 55E+02	0.11051866564646 72E+03	0.14753385044963 94E+03
8	0.88106725680796 62E+02	0.10213588045896 03E+03	0.14438067077570 01E+03	0.23759674531511 29E+03
9	0.71932200638079 60E+02	0.66148024708108 37E+02	0.71066193441944 51E+02	0.84972794048731 48E+02
10	0.64006957405931 27E+02	0.59357252617335 65E+02	0.72294913205747 64E+02	0.10806518442251 91E+03
11	0.50962013152453 69E+02	0.42072651428030 29E+02	0.46040684907719 32E+02	0.58288634748879 66E+02
12	0.46822894217285 21E+02	0.37512714482885 37E+02	0.40567204680434 24E+02	0.50912407066133 79E+02
13	0.66488695354793 73E+02	0.59855348216897 69E+02	0.67248442403781 06E+02	0.89201626505812 95E+02
14	0.83762509124442 08E+02	0.96361907884162 97E+02	0.13557245937328 54E+03	0.21885020389196 60E+03
15	0.12101256080888 72E+03	0.18831655259158 68E+03	0.33981881782686 85E+03	0.67986411960733 20E+03
16	0.15175800383207 35E+03	0.30022741798275 92E+03	0.71964601846040 44E+03	0.19792332941997 76E+04
17	0.18091984270295 23E+03	0.43760665958045 51E+03	0.12899836547651 79E+04	0.43614135298811 82E+04
18	0.21843659834292 02E+03	0.61153839163681 65E+03	0.20787308824865 24E+04	0.82331010499762 78E+04
19	0.21877980640347 54E+03	0.60165075285281 05E+03	0.19355069056866 99E+04	0.69830759967266 11E+04
20	0.27546177828815 66E+03	0.98703170572505 84E+03	0.43084000407376 89E+04	0.21650454653347 77E+05
21	0.32844822831451 03E+03	0.14318712015818 23E+04	0.76098945248425 68E+04	0.46611058367647 93E+05
22	0.31476946734243 42E+03	0.88562460879751 77E+04	0.90351061023234 97E+05	0.96668137664858 90E+06
23	0.47702824097327 31E+03	0.25110424924462 21E+04	0.14458859074091 91E+05	0.90149124388719 81E+05
24	0.26827853739686 61E+03	0.94538748596563 29E+03	0.39444305263145 05E+04	0.18563322257364 84E+05
25	0.22643387411059 86E+03	0.64316542909006 55E+03	0.21468994000381 32E+04	0.81020499040272 80E+04
26	0.17300423065581 64E+03	0.38343391339188 52E+03	0.99950530402429 69E+03	0.29063691273582 44E+04
27	0.15205480024683 91E+03	0.30709115708822 89E+03	0.74507543007897 90E+03	0.20725817315457 04E+04
28	0.12681845573813 21E+03	0.21183267971080 26E+03	0.41822065458173 84E+03	0.93206956298732 51E+03
29	0.93741323887142 57E+02	0.11565873599266 51E+03	0.17665121829539 82E+03	0.31858259688110 39E+03
30	0.71999883428307 23E+02	0.69294386382555 27E+02	0.78497922318997 91E+02	0.99008251535808 44E+02
31	0.53159974816724 89E+02	0.36606993546910 31E+02	0.28935953644437 05E+02	0.25086794832863 66E+02
32	0.36819073295986 34E+02	0.19888732157356 53E+02	0.14061096987443 12E+02	0.11823567329794 05E+02
33	0.26033333333342 80E+02	0.15594044784591 11E+02	0.11486631070496 48E+02	0.90506032135872 12E+01

```
MODULATION TECHNIQUE = QASK
BIPOLAR LOGIC
BAUD OR SYMBOL RATE =        1200 HZ
BITS PER BINARY CODE WORD =           3
BIT RATE =  0.3600000000000000E+04
CARRIER FREQUENCY =        2400 HZ
MAXIMUM CARRIER AMPLITUDE =           1 VOLT(S)
INITIAL PHASE ANGLE =           0 DEGREES
TIME BETWEEN SAMPLES =  0.1388888888888889E-03 SEC
NUMBER OF SAMPLES GENERATED =          64
SEED FOR RANDOM NUMBER GENERATOR =           1
NUMBER OF TIMES SIMULATION REPEATS =         100
   SUM OF VARIANCES         SUM OF VARIANCES**2
 0.4885105807865026E+02   0.1949104911522201E+03
   SUM OF SKEWNESS           SUM OF SKEWNESS**2
-0.543833334859747 2E+04   0.1951490781083542E+08
   SUM OF KURTOSIS           SUM OF KURTOSIS**2
 0.6866043785868234E+03   0.9002560164061202E+05
    MEAN VARIANCE         VARIANCE OF THE VARIANCES
 0.1480335093292432E+01   0.3831079860435228E+01
    MEAN SKEWNESS         VARIANCE OF THE SKEWNESS
-0.1647979802605294E+03   0.5818337955938460E+06
    MEAN KURTOSIS         VARIANCE OF THE KURTOSIS
 0.2080619329050980E+02   0.2366874319550756E+04
```

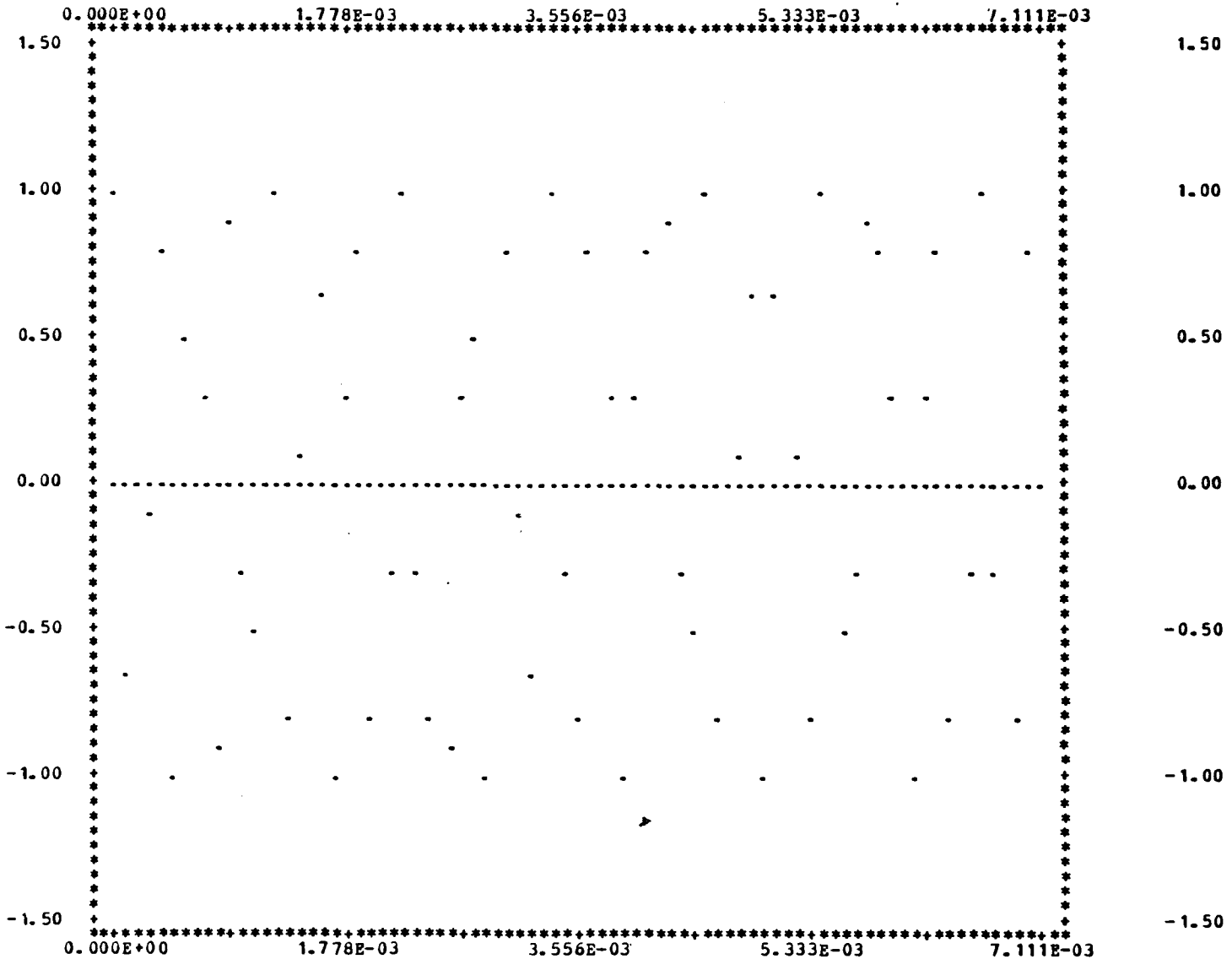

0.000E+00
2.222E-03
4.444E-03
6.667E-03
8.889E-03
1.50
1.00
0.50
0.00
-0.50
-1.00
-1.50

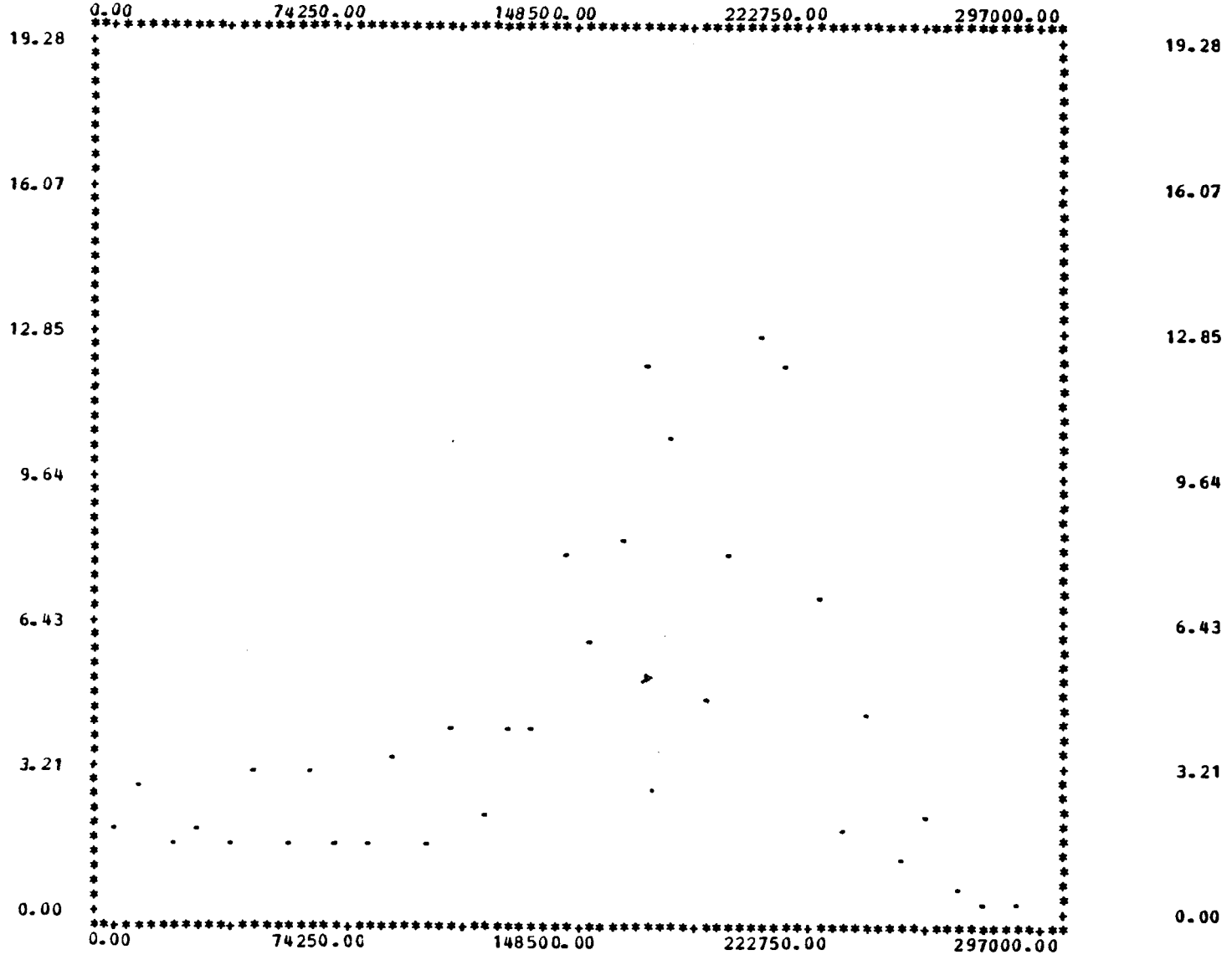

0.00
59400.00
118800.00
178200.00
237600.00
19.50
16.25
13.00
9.75
6.50
3.25
0.00

N	MEAN	VARIANCE	SKEWNESS	KURTOSIS
1	0.36310714285701 45E+00	0.91000037077118 14E-01	0.20487644481739 50E+00	0.27365093464212 54E-01
2	0.43487565783459 95E+00	0.10774029603760 41E+00	0.25199936714612 99E+00	0.38497481041346 76E-01
3	0.67508992683565 37E+00	0.10447298990198 33E+00	0.31213948104751 25E+00	0.31876680665553 82E-01
4	0.74807862586347 35E+00	0.13986694669834 51E+00	0.28812110107875 42E+00	0.45286947045051 94E-01
5	0.92212664660535 53E+00	0.22370622958898 53E+00	0.17778048236438 14E+00	0.12591988485762 82E+00
6	0.99411972420660 54E+00	0.24484280850973 90E+00	0.88743889100388 66E-01	0.14078863710807 98E+00
7	0.10261035230875 46E+01	0.28219210144611 65E+00	-0.12999446491491 54E-01	0.16986377242380 30E+00
8	0.10202814171077 85E+01	0.23298021916932 31E+00	0.15759972992987 81E-01	0.14687812521876 03E+00
9	0.90396039101291 96E+00	0.25731603908712 51E+00	0.27612386848169 87E+00	0.25389510364035 36E+00
10	0.80958739742068 75E+00	0.25721442594662 62E+00	0.35326470864065 04E+00	0.17978845444724 88E+00
11	0.69211235681450 22E+00	0.21922131289559 97E+00	0.38626274607500 74E+00	0.14480633319928 47E+00
12	0.64862610535272 44E+00	0.21327308347655 32E+00	0.38325010537155 90E+00	0.13149552528114 68E+00
13	0.87540973569654 94E+00	0.22735818936912 60E+00	0.25415727235493 17E+00	0.12197936843272 22E+00
14	0.12097219593326 01E+01	0.34946198096526 73E+00	-0.45505293767074 17E+00	0.41756234290980 05E+00
15	0.15881095221480 07E+01	0.67561642391920 16E+00	-0.26960421725427 27E+01	0.12541943606412 40E+01
16	0.19069746090849 11E+01	0.10937116400601 20E+01	-0.58257116245767 04E+01	0.39568904539646 88E+01
17	0.23574650802336 30E+01	0.15545817853243 49E+01	-0.13724807659482 16E+02	0.67223003947526 44E+01
18	0.27528799917041 21E+01	0.18105791048821 14E+01	-0.25265904731173 68E+02	0.83089109267699 93E+01
19	0.31012768519179 39E+01	0.22839953683341 45E+01	-0.39112146978394 33E+02	0.13111139702170 36E+02
20	0.36986348914452 04E+01	0.41655815873868 05E+01	-0.67136299160998 45E+02	0.63909206356103 88E+02
21	0.41769672189937 68E+01	0.49472655055982 36E+01	-0.10258141102069 36E+03	0.83200538588762 70E+02
22	0.13336459618244 57E+02	0.72087451916746 09E+01	-0.43768441508857 77E+04	0.17229999308178 96E+03
23	0.69895735301714 78E+01	0.73255994874333 67E+01	-0.57108004113670 75E+03	0.21223607338359 23E+03
24	0.40428553842466 94E+01	0.36862587330680 45E+01	-0.93970035526120 41E+02	0.55058426569002 05E+02
25	0.33115450459191 14E+01	0.21697708468067 09E+01	-0.48700335135406 95E+02	0.16387131476477 58E+02
26	0.33730867096045 16E+01	0.25580593277139 10E+01	-0.51783061128924 27E+02	0.19908426631765 74E+02
27	0.26776251047416 95E+01	0.18309462880860 45E+01	-0.22605328089151 64E+02	0.98907947855528 60E+01
28	0.23278443633693 72E+01	0.19895313640174 29E+01	-0.11763716875613 31E+02	0.13261997003934 31E+02
29	0.19697481671840 20E+01	0.12468667930029 16E+01	-0.68388085413894 47E+01	0.35895049780370 20E+01
30	0.13667607305066 31E+01	0.46286434489657 60E+00	-0.12327080155533 22E+01	0.66353942563268 72E+00
31	0.11849962679072 73E+01	0.37742610541422 14E+00	-0.38823339693996 98E+00	0.51024560056214 98E+00
32	0.87009769470919 32E+00	0.27539766660237 84E+00	0.27688358907621 74E+00	0.18318240421472 03E+00
33	0.65062288222721 12E+00	0.23761385425959 25E+00	0.41408283758330 57E+00	0.17587871297180 80E+00

N	SUM X	SUM X**2	SUM X**3	SUM X**4
1	0.36310714285701 45E+02	0.22193683390013 13E+02	0.18294339938645 18E+02	0.16965808310367 88E+02
2	0.43487566578345995E+02	0.29577973085430 37E+02	0.25964790668019 69E+02	0.26182966865913 56E+02
3	0.67508992683565 37E+02	0.55917466931793 16E+02	0.53312622221215 57E+02	0.56557253253631 17E+02
4	0.74807862586347 35E+02	0.69808990770514 44E+02	0.73555625678685 89E+02	0.84183443601355 90E+02
5	0.92212664660535 53E+02	0.10717867196727 33E+03	0.14421146586690 20E+03	0.21461347605171 65E+03
6	0.99411972420660 54E+02	0.12306684064812 59E+03	0.17824910475099 49E+03	0.28614575981643 88E+03
7	0.10261035230875 46E+03	0.13322586205243 29E+03	0.19823294658744 64E+03	0.32155687650990 48E+03
8	0.10202814171077 85E+03	0.12716245870730 99E+03	0.18260564399861 78E+03	0.29077555076597 10E+03
9	0.90396039101291 96E+02	0.10718872672164 84E+03	0.15486659910255 87E+03	0.26014808247008 48E+03
10	0.80958739742068 75E+02	0.91007403574956 21E+02	0.12527546107037 35E+03	0.19464696572249 05E+03
11	0.69211235681450 22E+02	0.69604861422196 85E+02	0.87345525770426 20E+02	0.12507769114817 37E+03
12	0.64862610535272 44E+02	0.63185617718683 12E+02	0.77133369033673 77E+02	0.10687382548366 24E+03
13	0.87540973569654 94E+02	0.99142681282773 73E+02	0.13251869141903 32E+03	0.19655133396180 76E+03
14	0.12097219599326 02E+03	0.18093945800471 23E+03	0.31846856942874 27E+03	0.63652532207605 96E+03
15	0.15881095221480 07E+03	0.31909521140171 82E+03	0.74625184297499 29E+03	0.19454976629690 14E+04
16	0.19069746090849 11E+03	0.47193266833540 67E+03	0.13900092527433 53E+04	0.46686453483556 75E+04
17	0.23574650802336 30E+03	0.70966775719920 62E+03	0.25350417816421 34E+04	0.10179059692599 94E+05
18	0.27528799917041 21E+03	0.93708215625581 77E+03	0.36967640327564 42E+04	0.16158081199033 36E+05
19	0.31012768519179 39E+03	0.11879073526892 84E+04	0.52172904050673 48E+04	0.25232292607656 24E+05
20	0.36986348914452 04E+03	0.17803825831728 81E+04	0.10305345498590 66E+05	0.68862927210275 52E+05
21	0.41769672189937 68E+03	0.22344847999090 78E+04	0.14252557149901 13E+05	0.10385875267807 08E+06
22	0.13336459616244 57E+04	0.18499781288882 60E+05	0.26690811219822 64E+06	0.40036893727759 44E+07
23	0.69895735301714 79E+03	0.56106481626232 82E+04	0.50769281913325 28E+05	0.51204388506494 50E+06
24	0.40428553842466 94E+03	0.19994075803669 85E+04	0.11584007620572 41E+05	0.76902220472531 89E+05
25	0.33115450459151 14E+03	0.13114403729490 07E+04	0.59653819784018 17E+04	0.30445227606291 25E+05
26	0.33730867096045 16E+03	0.13910192684947 39E+04	0.66222369190956 53E+04	0.35216356424195 34E+05
27	0.26776251047416 95E+03	0.89823130267481 55E+03	0.35209120084970 06E+04	0.15480920505587 73E+05
28	0.23278443633693 72E+03	0.73884954304478 14E+03	0.28996366688890 50E+04	0.13112691635922 74E+05
29	0.19697481671840 20E+03	0.51143059671976 94E+03	0.15623060190278 80E+04	0.52786321295674 80E+04
30	0.13667607305066 31E+03	0.23262705959026 29E+03	0.45695880925725 59E+03	0.10041008881868 20E+04
31	0.11849962675074 74E+03	0.17778679993142 46E+03	0.31236351252091 54E+03	0.62526326966939 65E+03
32	0.87009769470919 32E+02	0.10297136882746 07E+03	0.14505981115344 21E+03	0.22739036009232 06E+03
33	0.65062288222721 12E+02	0.65654785059464 02E+02	0.85286146974327 75E+02	0.12603979974302 25E+03

```
MODULATION TECHNIQUE = MSK
BIPOLAR LOGIC
BAUD OR SYMBOL RATE =        1200 HZ
BITS PER BINARY CODE WORD =           1
BIT RATE =  0.1200000000000000E+04
CARRIER FREQUENCY =        3000 HZ
MAXIMUM CARRIER AMPLITUDE =           1 VOLT(S)
INITIAL PHASE ANGLE =           0 DEGREES.
TIME BETWEEN SAMPLES =  0.1111111111111111E-03 SEC
NUMBER OF SAMPLES GENERATED =          64
SEED FOR RANDOM NUMBER GENERATOR =           1
NUMBER OF TIMES SIMULATION REPEATS =         100
   SUM OF VARIANCES          SUM OF VARIANCES**2
 0.1525890150844788E+03    0.2084564025759793E+04
   SUM OF SKEWNESS            SUM OF SKEWNESS**2
-0.1665578258850635E+05    0.9704853819233804E+08
   SUM OF KURTOSIS            SUM OF KURTOSIS**2
 0.5423666237610907E+04    0.6118691548293858E+07
    MEAN VARIANCE          VARIANCE OF THE VARIANCES
 0.4623909548014508E+01    0.4309394443714190E+02
    MEAN SKEWNESS          VARIANCE OF THE SKEWNESS
-0.5047206845001924E+03    0.2770063131355647E+07
    MEAN KURTOSIS          VARIANCE OF THE KURTOSIS
 0.1643535223518457E+03    0.1633529030650538E+06
```


0.000E+00
1.778E-03
3.556E-03
5.333E-03
7.111E-03
1.50
1.00
0.50
0.00
-0.50
-1.00
-1.50


0.00
74250.00
148500.00
222750.00
297000.00
19.28
16.07
12.85
9.64
6.43
3.21
0.00

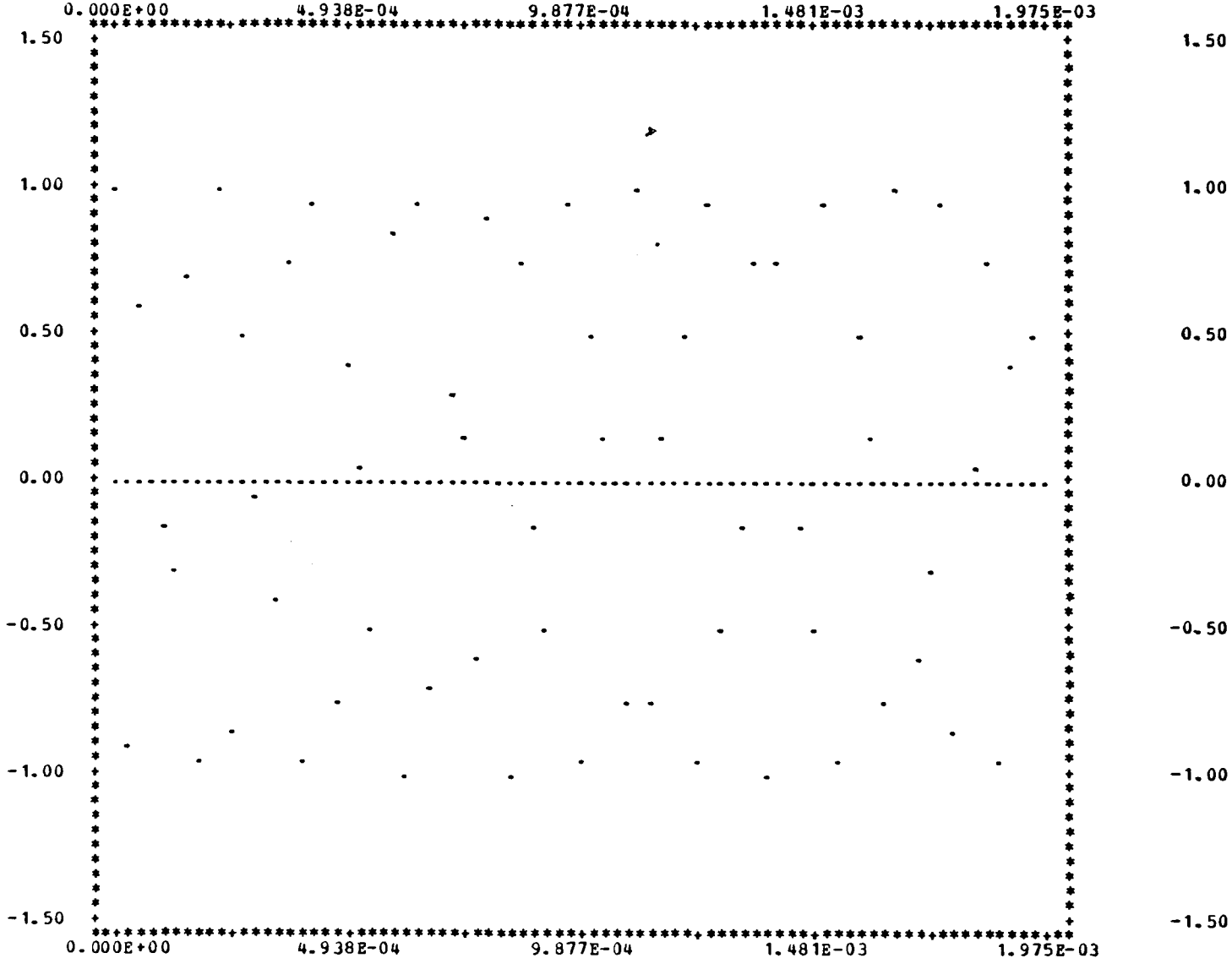
240

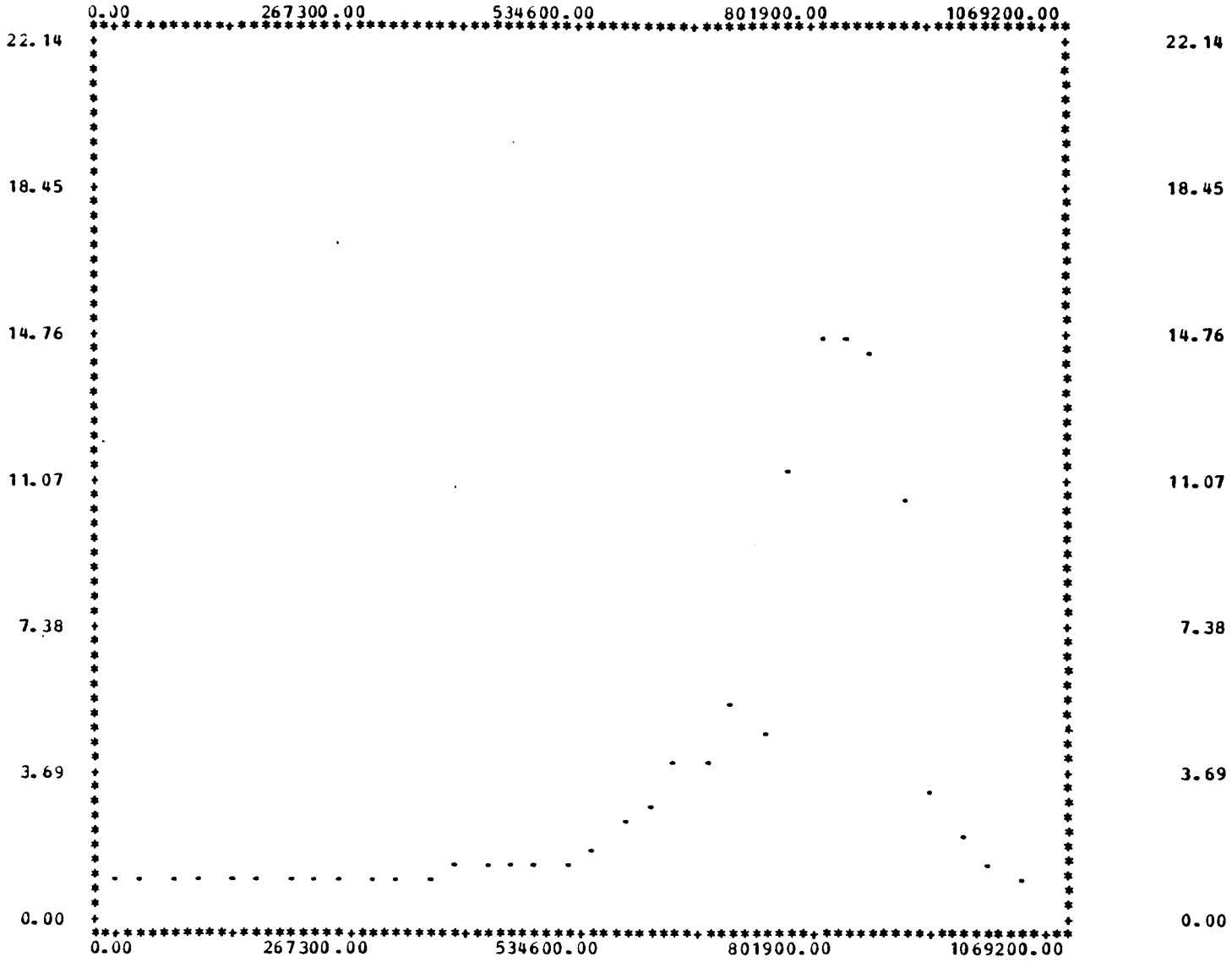
N	MEAN	VARIANCE	SKEWNESS	KURTOSIS
1	0.1745381057261094E+01	0.1355578995944516E+01	-0.3424362342353158E+01	0.5473118008703764E+01
2	0.1670893117553454E+01	0.5901911630020479E+00	-0.3495315053453071E+01	0.1090755818972799E+01
3	0.1549301951756876E+01	0.6168607541593016E+00	-0.2050204688587283E+01	0.2072568134087788E+01
4	0.1661199892949759E+01	0.6791172245377799E+00	-0.3416316593482210E+01	0.1031766180903137E+01
5	0.1789165204967162E+01	0.1187649235767541E+01	-0.4253824550585361E+01	0.4425773685632588E+01
6	0.1793980989469851E+01	0.6705838354600710E+00	-0.5135274827787229E+01	0.9492102864198628E+00
7	0.1744198082222983E+01	0.8410759608141099E+00	-0.4231013557271023E+01	0.1621868551827440E+01
8	0.1974720124125793E+01	0.9695494611582597E+00	-0.7310166411774525E+01	0.2277449619040244E+01
9	0.2053488465398217E+01	0.9611806657981106E+00	-0.8487669792860372E+01	0.2604980667152322E+01
10	0.2088844746733196E+01	0.8767072712879614E+00	-0.9051519777937444E+01	0.2339329653500119E+01
11	0.2078335081298121E+01	0.8586827233682362E+00	-0.9240745527458049E+01	0.1528701296758548E+01
12	0.1846376751847116E+01	0.1102239221695370E+01	-0.5290548653695978E+01	0.3019964469265048E+01
13	0.2245946499482377E+01	0.1194857291064416E+01	-0.1223253783690999E+02	0.3015304601135153E+01
14	0.2963675540203763E+01	0.2445326920526502E+01	-0.3299670208754627E+02	0.1525288363748558E+02
15	0.3173784304747806E+01	0.2467783740162910E+01	-0.4212913869181091E+02	0.1518478689342305E+02
16	0.3465632704745810E+01	0.5120393955377951E+01	-0.4209380992274822E+02	0.1417059143970117E+03
17	0.4614191970595573E+01	0.4884566208791112E+01	-0.1448217452612039E+03	0.8434809081291280E+02
18	0.6056467485580745E+01	0.1025151080329688E+02	-0.3488271204826219E+03	0.2873969971148171E+03
19	0.6406000917972383E+01	0.7829202751917509E+01	-0.4368456489015806E+03	0.1481115108052472E+03
20	0.1698791308733019E+02	0.2854008167165727E+02	-0.9245547474814263E+04	0.2008740923582969E+04
21	0.6683213299026370E+01	0.1119947335159933E+02	-0.4907458525202289E+03	0.3031919538307138E+03
22	0.6486578159166406E+01	0.1017117379541286E+02	-0.4465802292353152E+03	0.2680409342301182E+03
23	0.6465666718341273E+01	0.7377630480256458E+01	-0.4512761578680025E+03	0.1522694237588385E+03
24	0.1116454812650914E+02	0.2162335800890371E+02	-0.2501564247125116E+04	0.1165597750187054E+04
25	0.1044373719208894E+02	0.1462960849421151E+02	-0.2068981176065553E+04	0.6248220438872377E+03
26	0.4981051119955246E+01	0.6452035401677616E+01	-0.1905233821159606E+03	0.1076197803024863E+03
27	0.4123269596834735E+01	0.4439736504910233E+01	-0.1001918569207450E+03	0.6134152643541944E+02
28	0.2864935711027876E+01	0.1615487880606123E+01	-0.3017501863379686E+02	0.5746676318150122E+01
29	0.1801795405750350E+01	0.9106026395844705E+00	-0.4741745708792403E+01	0.2331085528965168E+01
30	0.1309968810623288E+01	0.3731036902683101E+00	-0.9703918339010844E+00	0.3451406134067727E+00
31	0.8612683866837359E+00	0.2253789614502914E+00	0.2871099110118729E+00	0.1541083993062977E+00
32	0.6043391628396291E+00	0.7921584690864393E-01	0.2862143793099431E+00	0.1130792746883191E-01
33	0.5680265238925039E+00	0.4907017290140923E-01	0.2752850066625604E+00	0.2607974479555311E-02

N	SUM X	SUM X**2	SUM X**3	SUM X**4
1	0.174538105726109 4E+03	0.438837824107 0927E+03	0.134611043495315 5E+04	0.470816322717992 1E+04
2	0.167089311755345 4E+03	0.337617306161 9528E+03	0.784459032316075 7E+03	0.203491919039359 9E+04
3	0.154930195175687 6E+03	0.301102868435 3374E+03	0.714410008828758 7E+03	0.202659944555598 9E+04
4	0.166119989294975 9E+03	0.343191113662869 3E+03	0.816778447618721 9E+03	0.213270817834439 0E+04
5	0.178916520496716 2E+03	0.437688487407505 2E+03	0.128368615566107 5E+04	0.429707988676225 2E+04
6	0.179398098546985 1E+03	0.388224578768469 5E+03	0.932201500972156 6E+03	0.239499860475278 9E+04
7	0.174419808222298 3E+03	0.387489215123630 1E+03	0.996027091969004 9E+03	0.281482367587664 4E+04
8	0.197472012412579 4E+03	0.485937353517406 7E+03	0.136785025826282 8E+04	0.422457323856694 E+04
9	0.205348848539821 7E+03	0.516838381880319 2E+03	0.149183502679059 5E+04	0.477234986379371 8E+04
10	0.208884474673319 6E+03	0.523121257452995 1E+03	0.150035081862073 9E+04	0.478625322479581 8E+04
11	0.207833508129812 1E+03	0.516957260628902 3E+03	0.143526133611377 3E+04	0.428414070265940 4E+04
12	0.184637675184711 6E+03	0.450032393923999 22E+03	0.128191176183618 8E+04	0.405091633007366 1E+04
13	0.224594649948237 7E+03	0.622718439669091 4E+03	0.196366797981534 2E+04	0.672913761928913 2E+04
14	0.296367554020376 3E+03	0.112042463589233 0E+04	0.490538051383237 8E+04	0.237747611699 6190E+05
15	0.317378430474780 7E+03	0.125160127158248 0E+04	0.568944218285881 9E+04	0.285421855450776 7E+05
16	0.346563270474581 0E+03	0.170798000600279 3E+04	0.111461911024501 0E+05	0.888781443094928 1E+05
17	0.461419197059557 3E+03	0.261264880882118 6E+04	0.174254614325242 8E+05	0.132289749999496 2E+06
18	0.605646748558074 5E+03	0.468297940991606 7E+04	0.428680658663607 5E+05	0.440247637544683 0E+06
19	0.640600091797238 3E+03	0.487877584854613 5E+04	0.418683932507334 1E+05	0.391596385179998 8E+06
20	0.169879130873301 9E+04	0.316843871917625 1E+05	0.632482261822974 8E+06	0.133015752104038 0E+08
21	0.668321329902637 0E+03	0.557528186183662 8E+04	0.537747404024616 8E+05	0.572238055684963 5E+06
22	0.648657815916640 6E+03	0.521451582724333 7E+04	0.483999312603460 7E+05	0.497286801957919 1E+06
23	0.646566671834127 4E+03	0.491087002881199 1E+04	0.417675617008345 7E+05	0.387952029305662 6E+06
24	0.111645481265091 4E+04	0.146054259279535 E+05	0.214103090417754 8E+06	0.341594229820323 1E+07
25	0.104437371920889 4E+04	0.123554958946691 2E+05	0.158400209091534 5E+06	0.216282436149220 7E+07
26	0.498105111995524 6E+03	0.311983853072682 5E+04	0.226057133591515 8E+05	0.181401164221202 4E+06
27	0.412326959683473 5E+03	0.213966913080428 1E+04	0.130478418972824 6E+05	0.897834140078428 5E+05
28	0.286493571102787 6E+03	0.980718963012286 4E+03	0.377001714980413 2E+04	0.156911639692977 6E+05
29	0.180179540575035 0E+03	0.414796329737169 4E+03	0.111866645601176 3E+04	0.337766986510697 2E+04
30	0.130996881062328 8E+03	0.208539093871741 8E+03	0.379296285737031 8E+03	0.758253516157826 9E+03
31	0.861268386837359 E+02	0.964908405736593 7E+02	0.129667876092925 8E+03	0.197747192973357 2E+03
32	0.604339162839629 1E+02	0.443649512181261 3E+02	0.360107056184035 5E+02	0.309791270685826 1E+02
33	0.568026523892503 9E+02	0.371233603017796 4E+02	0.262588342194623 8E+02	0.192873278435029 1E+02

MODULATION TECHNIQUE = MFSK
BIPOLAR LOGIC
BAUD OR SYMBOL RATE = 1200 HZ
BITS PER BINARY CODE WORD = 3
BIT RATE = 0.360000000000000E+04
CARRIER FREQUENCY = 10800 HZ
MAXIMUM CARRIER AMPLITUDE = 1 VOLT(S)
INITIAL PHASE ANGLE = 0 DEGREES
TIME BETWEEN SAMPLES = 0.308641975308641 9E-04 SEC
NUMBER OF SAMPLES GENERATED = 64
SEED FOR RANDOM NUMBER GENERATOR = 1
NUMBER OF TIMES SIMULATION REPEATS = 100

SUM OF VARIANCES	SUM OF VARIANCES**2
0.480618114192526 1E+03	0.1352273679574422E+05
SUM OF SKEWNESS	**SUM OF SKEWNESS**2**
-0.1654247621695240E+04	0.5890267815807785E+06
SUM OF KURTOSIS	**SUM OF KURTOSIS**2**
0.5341287673934422E+05	0.2790612658473240E+09
MEAN VARIANCE	**VARIANCE OF THE VARIANCES**
0.1456418527856140E+02	0.2038414228878589E+03
MEAN SKEWNESS	**VARIANCE OF THE SKEWNESS**
-0.5012871580894667E+02	0.1581567102109974E+05
MEAN KURTOSIS	**VARIANCE OF THE KURTOSIS**
0.1618572022404370E+04	0.6019021185027759E+07

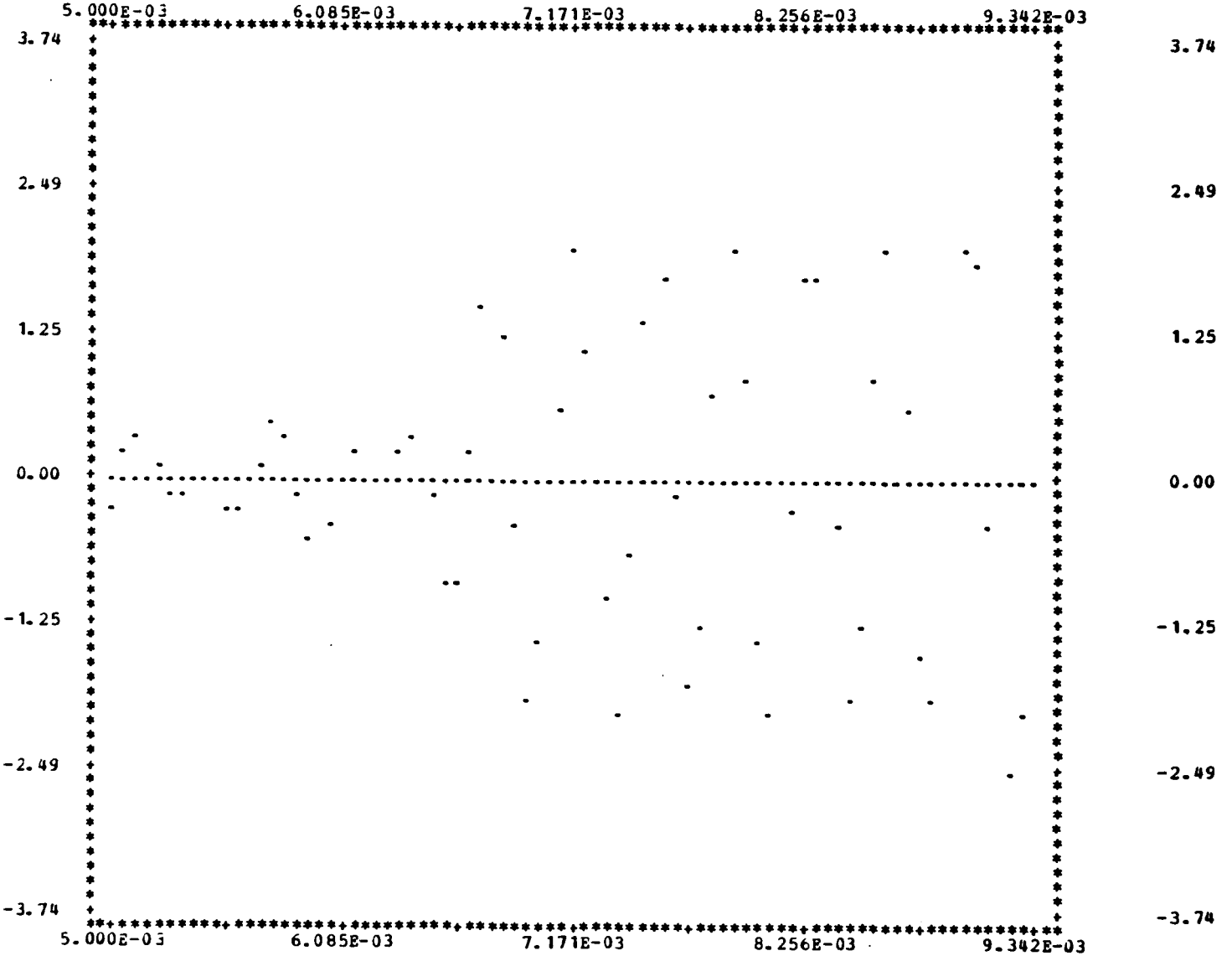

0.000E+00
4.938E-04
9.877E-04
1.481E-03
1.975E-03
1.50
1.00
0.50
0.00
-0.50
-1.00
-1.50

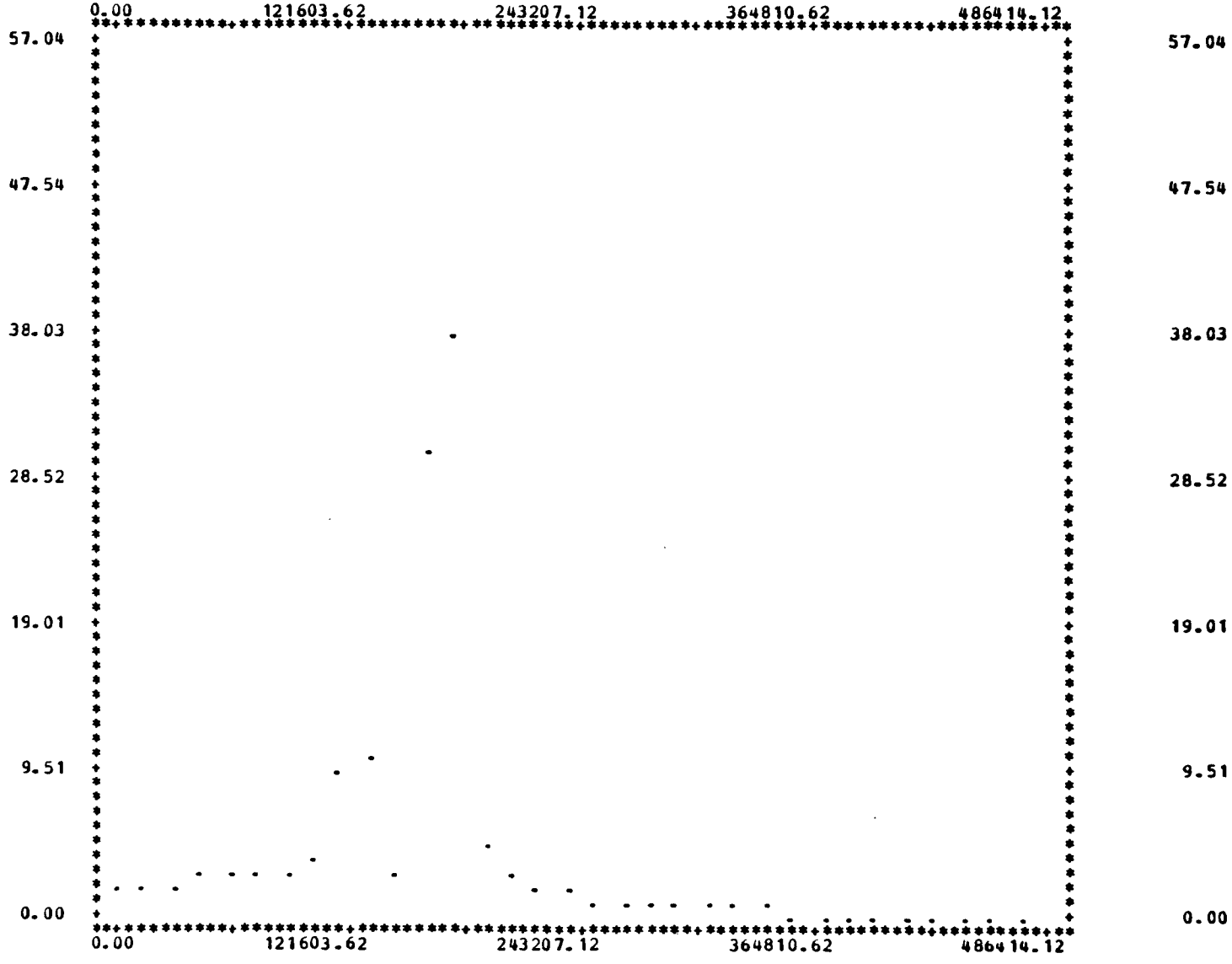

0.00
267300.00
534600.00
801900.00
1069200.00
22.14
18.45
14.76
11.07
7.38
3.69
0.00

N	MEAN	VARIANCE	SKEWNESS	KURTOSIS
1	0.5772242127037718E+00	0.1701870281944493E+00	0.2881221581889301E+00	0.5202216579936257E-01
2	0.5802219943279051E+00	C.1692179389704175E+00	0.2881870091118721E+00	0.4962377142977341E-01
3	0.6028101560497944E+00	0.1630844574986009E+00	0.2975209809667937E+00	0.4759923908949447E-01
4	0.6236957775C89775E+00	0.1517433865402484E+00	0.3011297117696356E+00	0.4060584454262756E-01
5	0.6603884574738126E+00	0.1541310275696224E+00	0.3222793291193875E+00	0.5372574080695565E-01
6	0.6914911473735869E+00	0.1625277259401340E+00	0.3303730158038245E+00	0.6338732183899803E-01
7	0.6712915135223571E+00	0.1989459204647430E+00	0.3457384652309939E+00	0.9079440007191118E-01
8	0.8082796112917950E+00	0.2158539044684256E+00	0.2513665218179688E+00	0.8711759471530689E-01
9	0.8347057408743954E+00	0.2063633947272559E+00	0.2416258755581653E+00	0.7897411259783627E-01
10	0.1155203641180799E+01	0.3253381023466752E+00	-0.3616711703060194E+00	0.3009444434532169E+00
11	0.1405578654040239E+01	0.5901481998178570E+00	-0.1545347282296657E+01	0.6485308087822852E+00
12	0.1653469595833664E+01	0.1006408909069761E+01	-0.3264525934540579E+01	0.1839454693919151E+01
13	0.3282094270411689E+01	0.8604653034726285E+01	-0.2794322254332445E+02	0.1742698305978736E+03
14	0.5652678805043730E+01	0.4459844156194543E+02	0.1192554149182590E+03	0.8446317926807378E+04
15	0.5302792986411789E+01	0.1999384407499615E+02	-0.1915066362357811E+03	0.8114244563340110E+03
16	0.4839236365113163E+01	0.3116228371110405E+02	0.1153475974583148E+03	0.5030156057294299E+04
17	0.4799926056893745E+01	0.1627827150809915E+02	-0.9300212955009531E+02	0.1027015195900844E+04
18	0.4973004323275781E+01	0.1660039482645575E+02	-0.1210599846314087E+03	0.9267637549193778E+03
19	0.4949076398807320E+01	0.3195721379055708E+02	0.1586666023965808E+03	0.6953906279136898E+04
20	0.5240481051222312E+01	0.1710861900951835E+02	-0.1294472248356874E+03	0.1518760678679011E+04
21	0.5676550996298115E+01	0.4183417025969290E+02	0.1323276886342935E+03	0.8939299980725979E+04
22	0.6101993574152731E+01	0.1635330658894298E+02	-0.3215862356984157E+03	0.7932740986113413E+03
23	0.6259114577537281E+01	0.2770207472842645E+02	-0.2779436555304097E+03	0.1949698968797135E+04
24	0.6284446589130466E+01	0.2662812884562886E+02	-0.2772962242103021E+03	0.2129199108385613E+04
25	0.5742277487325188E+01	0.1282122464430269E+02	-0.2785797147037739E+03	0.4954058112039018E+03
26	0.5170868203869401E+01	0.2976675343581255E+02	-0.3721000245806551E+02	0.2486653253210702E+04
27	0.5917383281247872E+01	0.2149629222581870E+02	-0.2452121102148178E+03	0.1728196939758512E+04
28	0.5541030766931423E+01	0.3540159389584892E+02	-0.2604124431555488E+02	0.4059766140333669E+04
29	0.5288745936690563E+01	0.1787438125039946E+02	-0.2028946595998615E+03	0.6603169983834629E+03
30	0.5074754282284268E+01	0.3131561713839394E+02	0.1173219615183484E+01	0.2905905006347744E+04
31	0.4253003494334969E+01	0.2577288744924743E+02	0.5235457234353824E+02	0.2345083299105233E+04
32	0.1937545429384470E+01	0.2626549296496931E+01	-0.3005281031163669E+01	0.2094124943834442E+02
33	0.1137815044472511E+01	0.1207462920504103E+01	0.1860809816827437E+01	0.7168925235852774E+01

N	SUM X	SUM X**2	SUM X**3	SUM X**4
1	0.57722421270377 18E+02	0.50167294964399 40E+02	0.49047989490505 17E+02	0.51462415063064 05E+02
2	0.58022199432790 51E+02	0.50418332228256 49E+02	0.49248664199314 51E+02	0.51422227078386 69E+02
3	0.60281015604979 44E+02	0.52483369716039 24E+02	0.51988606114957 87E+02	0.55302020804138 26E+02
4	0.62369577750897 75E+02	0.53922237555737 40E+02	0.53206902232506 21E+02	0.56342385447849 42E+02
5	0.66038845747381 26E+02	0.58870263205856 78E+02	0.61637076887789 74E+02	0.71204080446531 05E+02
6	0.69149114737358 69E+02	0.63906245557677 24E+02	0.69977109396223 78E+02	0.85139571608119 46E+02
7	0.67129151352235 71E+02	0.64758875738723 25E+02	0.74863638423633 33E+02	0.95926935269284 23E+02
8	0.80827961129179 50E+02	0.86701129545375 67E+02	0.10470973203828 13E+03	0.13543786234076 61E+03
9	0.83470574087439 54E+02	0.90103343462865 68E+02	0.11044518696737 61E+03	0.14561748029505 61E+03
10	0.11552036411807 99E+03	0.16565801739205 85E+03	0.27104002709621 73E+03	0.49036435769942 01E+03
11	0.14055786540402 39E+03	0.25598980705132 48E+03	0.52977642656128 51E+03	0.11799026559909 48E+04
12	0.16534695958336 64E+03	0.37303065243254 03E+03	0.97713959201027 51E+03	0.27698827649514 51E+04
13	0.32820942704116 89E+03	0.19290749304248 26E+04	0.14045466514720 15E+05	0.11195125816980 45E+06
14	0.56526788050437 30E+03	0.76105234819316 59E+04	0.13459452030203 59E+06	0.27351399729725 73E+07
15	0.53027929864117 89E+03	0.47913519090984 25E+04	0.51448055582067 87E+05	0.60124401542645 41E+06
16	0.48392363651131 63E+03	0.54268869471426 67E+04	0.85637084018257 00E+05	0.15629838952596 45E+07
17	0.47999260566937 45E+03	0.39154778944665 69E+04	0.42473942127185 59E+05	0.53617232526086 21E+06
18	0.49730043232757 82E+03	0.41165162877510 81E+04	0.44362207024459 09E+05	0.54778573542425 32E+06
19	0.49490763988073 20E+03	0.56130998853883 15E+04	0.94306969300181 90E+05	0.19173956130420 84E+07
20	0.52404810512223 12E+03	0.44400174467643 27E+04	0.51366231077382 03E+05	0.72326246797837 23E+06
21	0.56765509962981 15E+03	0.73639059770669 10E+04	0.13219288605300 92E+06	0.27832965535323 04E+07
22	0.61019935741527 31E+03	0.53424099102054 77E+04	0.58192564142049 24E+05	0.72208234025263 33E+06
23	0.62591145775372 81E+03	0.66601569275881 88E+04	0.89430387330470 49E+05	0.13288994276970 67E+07
24	0.62844465891304 66E+03	0.65856116488806 13E+04	0.88532297785079 97E+05	0.13458015765039 97E+07
25	0.57422774873251 88E+03	0.45666763139301 33E+04	0.44216646149480 79E+05	0.48785538055049 06E+06
26	0.51708682038694 01E+03	0.56206963883241 99E+04	0.78123064871803 46E+05	0.11772828843273 80E+07
27	0.59173832812478 72E+03	0.56296754200752 35E+04	0.68414545627895 71E+05	0.97723118096256 75E+06
28	0.55410307669314 23E+03	0.65750599916971 07E+04	0.10055310030166 90E+06	0.17062053033217 39E+07
29	0.52887459366905 63E+03	0.45666471020756 41E+04	0.46571876239648 49E+05	0.51957168739749 23E+06
30	0.50747542822842 68E+03	0.56755591992572 52E+04	0.81372900808771 66E+05	0.12643670170746 87E+07
31	0.42530034943349 69E+03	0.43603197297580 42E+04	0.57251214631024 71E+05	0.83340214419015 11E+06
32	0.19375454293844 70E+03	0.63543660944606 13E+03	0.26422173335852 45E+04	0.12486841635292 50E+05
33	0.11378150444725 11E+03	0.24900113667270 44E+03	0.77710808478802 42E+03	0.28223525719609 14E+04

```
MODULATION TECHNIQUE = QPRS
QPRS CLASS 1 FILTER
BIPOLAR LOGIC
BAUD OR SYMBOL RATE =        1200 HZ
BITS PER BINARY CODE WORD =          2
BIT RATE =  0.2400000000000000E+04
CARRIER FREQUENCY =       2400 HZ
MAXIMUM CARRIER AMPLITUDE =          1 VOLT(S)
INITIAL PHASE ANGLE =           0 DEGREES
TIME BETWEEN SAMPLES =  0.6784342273548912E-04 SEC
NUMBER OF SAMPLES GENERATED =         64
SEED FOR RANDOM NUMBER GENERATOR =          1
NUMBER OF TIMES SIMULATION REPEATS =        100
   SUM OF VARIANCES         SUM OF VARIANCES**2
 0.7925891201989905E+03   0.1512135267846902E+06
   SUM OF SKEWNESS           SUM OF SKEWNESS**2
-0.1732359062524069E+06   0.9850778236273860E+10
   SUM OF KURTOSIS           SUM OF KURTOSIS**2
 0.3533192079165041E+06   0.4536060891356758E+11
    MEAN VARIANCE         VARIANCE OF THE VARIANCES
 0.2401785212724214E+02   0.4130538703065308E+04
    MEAN SKEWNESS         VARIANCE OF THE SKEWNESS
-0.5249572916739602E+04   0.2794176160813870E+09
    MEAN KURTOSIS         VARIANCE OF THE KURTOSIS
 0.1070664266413649E+05   0.1299304575250932E+10
```

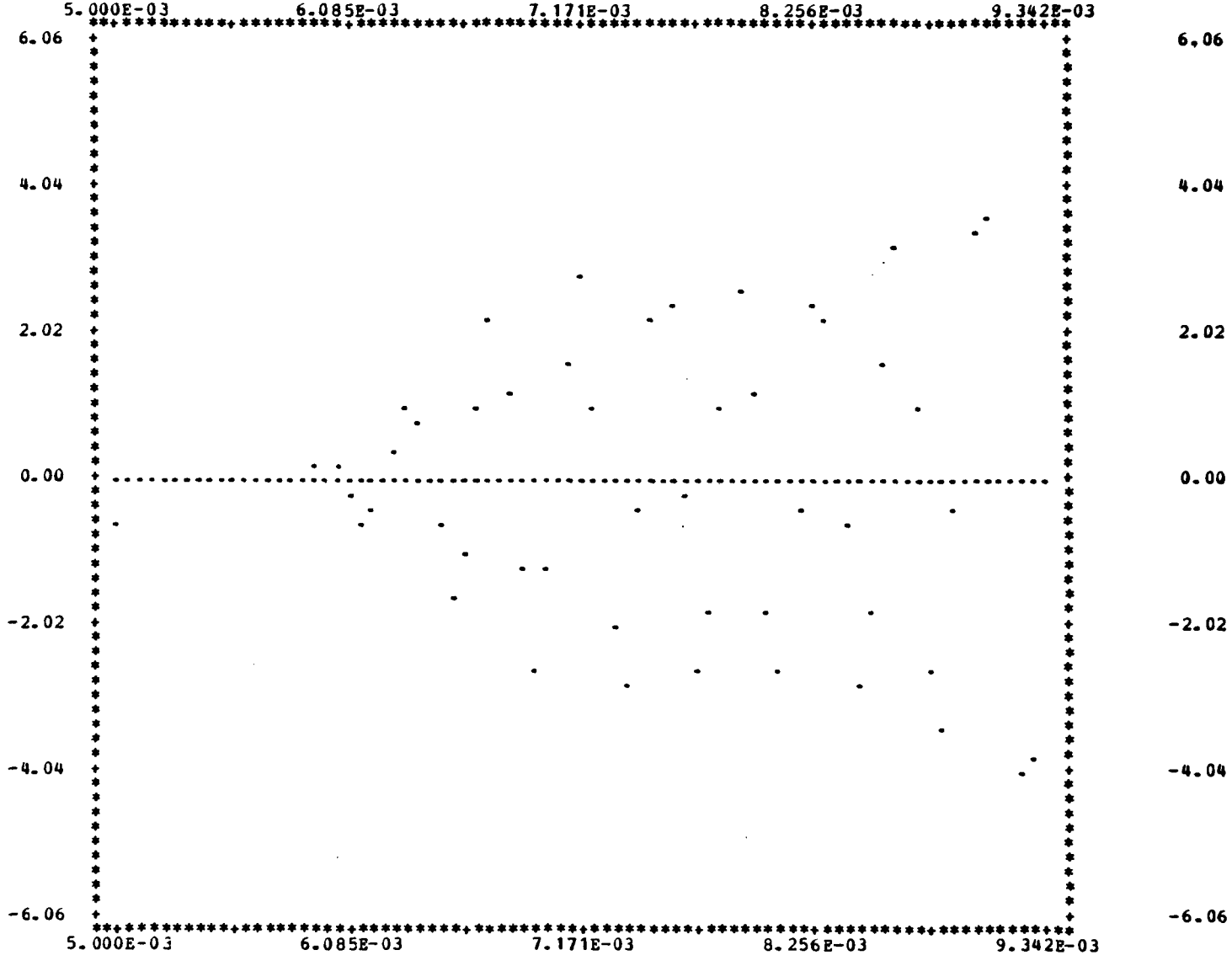

5.000E-03
6.085E-03
7.171E-03
8.256E-03
9.342E-03
3.74
2.49
1.25
0.00
-1.25
-2.49
-3.74

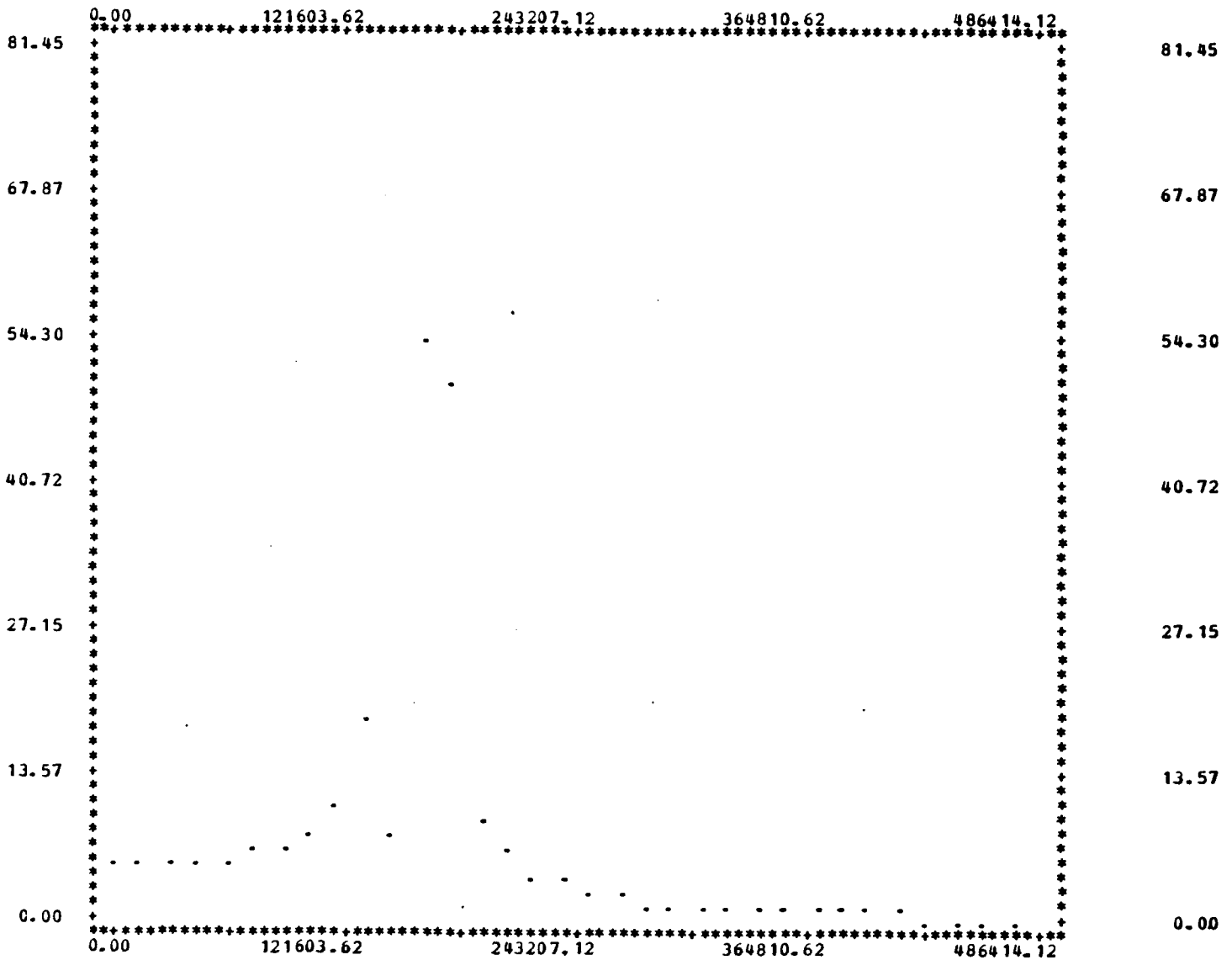

0.00
121603.62
243207.12
364810.62
486414.12
57.04
47.54
38.03
28.52
19.01
9.51
0.00

N	MEAN	VARIANCE	SKEWNESS	KURTOSIS
1	0.1746811380402326E+01	0.1388348922506595E+01	-0.3602282705486656E+01	0.5140926308007252E+01
2	0.1783934079105657E+01	0.1373900592274223E+01	-0.4000383957156492E+01	0.5076096043644750E+01
3	0.1881639570810958E+01	0.1391153714557888E+01	-0.5219656911680200E+01	0.5267763272060726E+01
4	0.2045686771930169E+01	0.1468678508208007E+01	-0.7699351745328097E+01	0.5884490676168223E+01
5	0.2297388544402002E+01	0.1644047467555237E+01	-0.1261209753570515E+02	0.7239435166189224E+01
6	0.2683505678955374E+01	0.1999006273090073E+01	-0.2311823186803285E+02	0.1024292414275401E+02
7	0.3312924180891911E+01	0.2756479367453149E+01	-0.4950242176250788E+02	0.1834851180285642E+02
8	0.4514755520309362E+01	0.4811195957225893E+01	-0.1408993176802396E+03	0.5414450274312225E+02
9	0.9536555141797188E+01	0.2327656842676565E+02	-0.1505196860526031E+04	0.1409137635928702E+04
10	0.2195585557239151E+02	0.1129481209797817E+03	-0.1990155013949577E+05	0.2735320445369462E+05
11	0.3232856430932140E+02	0.2402926487517075E+03	-0.6527459761041918E+05	0.1283904731804624E+06
12	0.3329064024908647E+02	0.2623808601021426E+03	-0.7047843429734775E+05	0.1648935745345857E+06
13	0.2012368604390696E+02	0.1051549869380090E+03	-0.1496394469756025E+05	0.3059724135122297E+05
14	0.7502438388978744E+01	0.1396155254214945E+02	-0.7105602094481494E+03	0.4730046115964915E+03
15	0.4016348508681620E+01	0.4194625326863587E+01	-0.9514782451504106E+02	0.4603296277462294E+02
16	0.2970080794310893E+01	0.2376936951023241E+01	-0.3349528176027253E+02	0.1564937982990288E+02
17	0.2390874625765719E+01	0.1616678403564954E+01	-0.1494345472058580E+02	0.7763585773794798E+01
18	0.2016812253485666E+01	0.1211509773096788E+01	-0.7479041281848517E+01	0.4629945683853585E+01
19	0.1753574112083879E+01	0.9704493989317589E+00	-0.3959409356771873E+01	0.3114448386156138E+01
20	0.1558493513598559E+01	0.8148158249946477E+00	-0.2125930435216646E+01	0.2271002271430383E+01
21	0.1408735774632585E+01	0.7084187564785975E+00	-0.1102582833912087E+01	0.1754746256637649E+01
22	0.1290826693315262E+01	0.6327165569438174E+00	-0.5020678660303746E+00	0.1417160155580148E+01
23	0.1196276693709804E+01	0.5773557484689719E+00	-0.1363393565939444E+00	0.1185743805976271E+01
24	0.1194726161491 26E+01	0.5361642063822957E+00	0.9264377890596 9C9E-01	0.1021682879331106E+01
25	0.1056567176528774E+01	0.5052688758882325E+00	0.2389420417312620E+00	0.9026889684670851E+00
26	0.1004854132567686E+01	0.4821360846597600E+00	0.3337377691836187E+00	0.8152494762041542E+00
27	0.9623989485715112E+00	0.4650482001544234E+00	0.3956861449378448E+00	0.7508171701876838E+00
28	0.9278127756762294E+00	0.4528007126313904E+00	0.4362793202635347E+00	0.7038029740335077E+00
29	0.9001115711570695E+00	0.4445161706018519E+00	0.4627541658592416E+00	0.6704545675391549E+00
30	0.8786288878768683E+00	0.4395214050389312E+00	0.4797259791699269E+00	0.6481934210539040E+00
31	0.8629071538189785E+00	0.4372540540716741E+00	0.4901293234968129E+00	0.6351869105735284E+00
32	0.8529998029052858E+00	0.4371419567298135E+00	0.4957619571123395E+00	0.6299589957549181E+00
33	0.8490527574548856E+00	0.4382132490394413E+00	0.4975782018316388E+00	0.6304885575454902E+00

N	SUM X	SUM X**2	SUM X**3	SUM X**4
1	0.1746811380402326E+03	0.4425815431984608E+03	0.1348821158956003E+04	0.4629031129955732E+04
2	0.1783934079105657E+03	0.4542582384946031E+03	0.1394577701660524E+04	0.4823443801265942E+04
3	0.1881639570810958E+03	0.4917809651853957E+03	0.1545984386736991E+04	0.5476301191979648E+04
4	0.2045686771930169E+03	0.5638826091976001E+03	0.1853679535368173E+04	0.6851959557828961E+04
5	0.2297388544402002E+03	0.6905601116829235E+03	0.2442646091043520E+04	0.9659305428541077E+04
6	0.2683505678955374E+03	0.9180218939344913E+03	0.3638487164735292E+04	0.1597192190773616E+05
7	0.3312924180891911E+03	0.1370438120211696E+04	0.6475137258677694E+04	0.3353247956116903E+05
8	0.4514755520309362E+03	0.2514610140581750E+04	0.1589201479119533E+05	0.1095168622809752E+06
9	0.9536555141797188E+03	0.1139896867150364E+05	0.1574118170391285E+06	0.2406790106482331E+07
10	0.2195585557239151E+04	0.5938782336856992E+05	0.1825164485184334E+07	0.6097112730194072E+08
11	0.3232856430932141E+04	0.1283025792566120E+06	0.5707027578545261E+07	0.2739699803410351E+09
12	0.3329064024908647E+04	0.1368023779695217E+06	0.6393219475235099E+07	0.3266222349853693E+09
13	0.2012368604390696E+04	0.5090661770623646E+05	0.1495899359077720E+07	0.4897850625075236E+08
14	0.7502438388978744E+03	0.7010851879714993E+04	0.7548211554455331E+05	0.8952526070200140E+06
15	0.4016348508681620E+03	0.2028373441678403E+04	0.1169897042246145E+05	0.7429632439420937E+05
16	0.2970080794310893E+03	0.1117454750624743E+04	0.4842988519051916E+04	0.2330127244219820E+05
17	0.2390874625765719E+03	0.7316793095659672E+03	0.2610458719032500E+04	0.1044932950102081E+05
18	0.2016812253485666E+03	0.5266926341175751E+03	0.1625310013580066E+04	0.5684185523967754E+04
19	0.1753574112083879E+03	0.4035767071513206E+03	0.1112159628710498E+04	0.3503146647320578E+04
20	0.1558493513598559E+03	0.3235569698673484E+03	0.8144108665461795E+03	0.2358656873687845E+04
21	0.1408735774632585E+03	0.2685871051643480E+03	0.6279392110129216E+03	0.1697266410506734E+04
22	0.1290826693315262E+03	0.2292622943549593E+03	0.5043601709341667E+03	0.1286755414806408E+04
23	0.1196276693709804E+03	0.2002660118897543E+03	0.4188711664557241E+03	0.1017735231758019E+04
24	0.1119472616149126E+03	0.1784021502626242E+03	0.3577695558724314E+03	0.8339267628334791E+03
25	0.1056567176528774E+03	0.1616550385647335E+03	0.3130255874735007E+03	0.7042934667581610E+03
26	0.1004854132567686E+03	0.1487046551551519E+03	0.2797068731648117E+03	0.6107399469756730E+03
27	0.9623989485715112E+02	0.1386609454364429E+03	0.2546677115393676E+03	0.5422335522854365E+03
28	0.9278127756762294E+02	0.1309109252213106E+03	0.2358431093725300E+03	0.4918077124087745E+03
29	0.9001115711570695E+02	0.1250271849426682E+03	0.2218504961080261E+03	0.4549510898295330E+03
30	0.8786288878768683E+02	0.1207114913600284E+03	0.2117566636175042E+03	0.4287039457588381E+03
31	0.8629671538189785E+02	0.1177593822101386E+03	0.2049379073396893E+03	0.4111351799552002E+03
32	0.8529980290528580E+02	0.1160379200918972E+03	0.2009954318637255E+03	0.4010352890071484E+03
33	0.8490527574548856E+02	0.1154721701490792E+03	0.1997055966179411E+03	0.3977393746428140E+03

```
MODULATION TECHNIQUE = QPRS
QPRS CLASS 2 FILTER
BIPOLAR LOGIC
BAUD OR SYMBOL RATE =        1200 HZ
BITS PER BINARY CODE WORD =           2
BIT RATE =  0.2400000000000000E+04
CARRIER FREQUENCY =        2400 HZ
MAXIMUM CARRIER AMPLITUDE =           1 VOLT(S)
INITIAL PHASE ANGLE =           0 DEGREES
TIME BETWEEN SAMPLES =  0.6784342273548912E-04 SEC
NUMBER OF SAMPLES GENERATED =          64
SEED FOR RANDOM NUMBER GENERATOR =           1
NUMBER OF TIMES SIMULATION REPEATS =         100
   SUM OF VARIANCES         SUM OF VARIANCES**2
 0.2406301297164022E+04   0.1712404031156036E+07
   SUM OF SKEWNESS           SUM OF SKEWNESS**2
-0.1014071770004887E+07   0.4097280851254902E+12
   SUM OF KURTOSIS           SUM OF KURTOSIS**2
 0.4062278349001294E+07   0.7316406994491162E+13
    MEAN VARIANCE         VARIANCE OF THE VARIANCES
 0.7291822112618248E+02   0.4802940065853782E+05
    MEAN SKEWNESS         VARIANCE OF THE SKEWNESS
-0.3072944757590566E+05   0.1183019436971622E+11
    MEAN KURTOSIS         VARIANCE OF THE KURTOSIS
 0.1230993439091301E+06   0.2130107248422762E+12
```

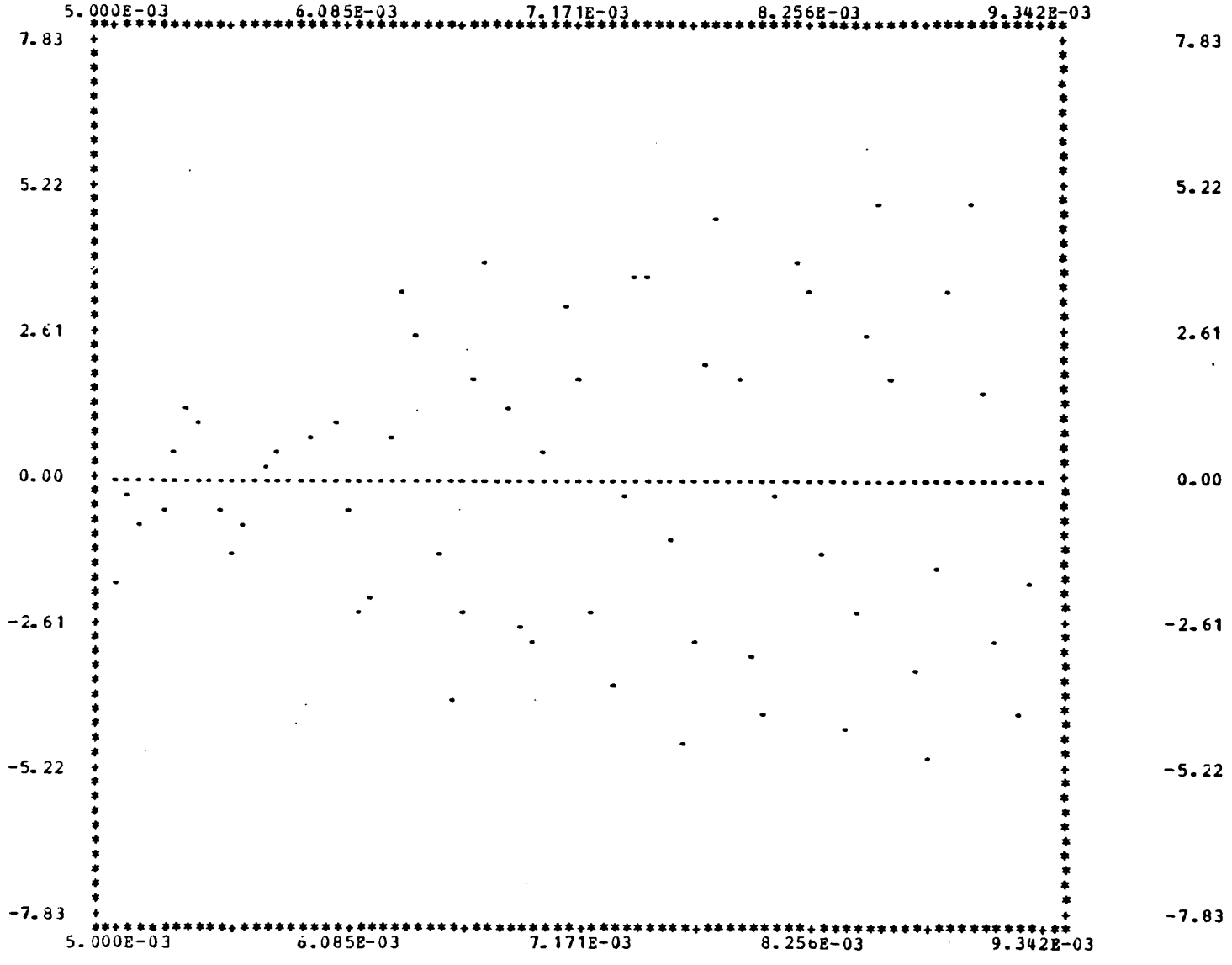

5.000E-03
6.085E-03
7.171E-03
8.256E-03
9.342E-03
6.06
4.04
2.02
0.00
-2.02
-4.04
-6.06

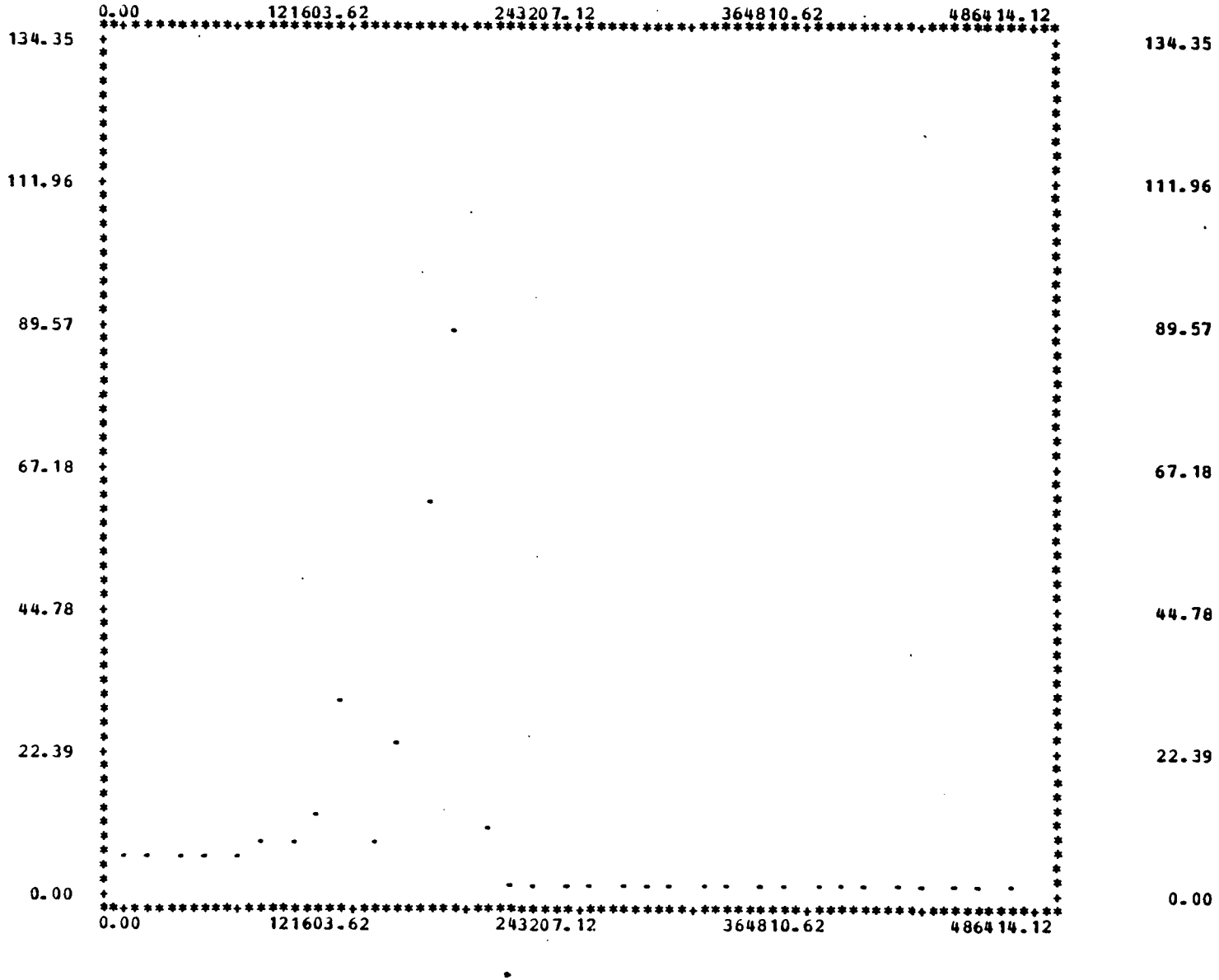

0.00
121603.62
243207.12
364810.62
486414.12
81.45
67.87
54.30
40.72
27.15
13.57
0.00

N	MEAN	VARIANCE	SKEWNESS	KURTOSIS
1	0.28837442762367 32E+01	0.39922014919269 21E+01	-0.25325188418264 97E+02	0.46932930259362 65E+02
2	0.29455756972216 25E+01	0.39630520088938 56E+01	-0.27624243758649 02E+02	0.46714175246901 46E+02
3	0.31127875615883 49E+01	0.40136649624060 92E+01	-0.34612334615240 47E+02	0.48509143970690 39E+02
4	0.33921056680051 65E+01	0.42312399339429 08E+01	-0.48433880909389 81E+02	0.53788833124454 81E+02
5	0.38143309945226 09E+01	0.47288919366053 61E+01	-0.74794103968501 61E+02	0.65241206302471 99E+02
6	0.44483047691195 60E+01	0.57296521728521 98E+01	-0.12845251513800 06E+03	0.89964076680085 33E+02
7	0.54464961635941 99E+01	0.77839571503510 74E+01	-0.25382113930464 06E+03	0.15165175601980 72E+03
8	0.72115061885285 57E+01	0.12687351317955 87E+02	-0.63076399836311 89E+03	0.36768546279382 01E+03
9	0.11829987918910 19E+02	0.33648745924618 98E+02	-0.29758914226305 22E+04	0.26495568224763 61E+04
10	0.32742654676887 94E+02	0.25566009160182 81E+03	-0.67218407062825 54E+05	0.15185923505206 92E+06
11	0.61562464061397 04E+02	0.86894151739415 95E+03	-0.45827815857742 53E+06	0.16812194437770 66E+07
12	0.61002515288187 68E+02	0.92107146042344 81E+03	-0.43981086066978 74E+06	0.21105011108813 29E+07
13	0.28127650337355 17E+02	0.20280990386209 81E+03	-0.41765134397807 50E+05	0.11234049789321 70E+06
14	0.10158124242558 07E+02	0.27377352403025 46E+02	-0.18302857250102 82E+04	0.20592054315090 25E+04
15	0.66284037695201 18E+01	0.11634265811207 65E+02	-0.47780873993306 68E+03	0.39214298482534 29E+03
16	0.50785439640818 08E+01	0.68215262754577 62E+01	-0.20100928903444 62E+03	0.14163951506376 68E+03
17	0.41572300143811 44E+01	0.46473323769456 44E+01	-0.10296397278892 50E+03	0.69001082615744 51E+02
18	0.35427509158549 95E+01	0.34612629379769 36E+01	-0.59354160506287 37E+02	0.39928584768750 06E+02
19	0.31035945769524 11E+01	0.27411221585140 52E+01	-0.37057582806240 42E+02	0.25930287369086 22E+02
20	0.27747264893960 73E+01	0.22727009552983 11E+01	-0.24508388135268 47E+02	0.18315318911844 02E+02
21	0.25200303167113 11E+01	0.19535429564347 40E+01	-0.16926105347203 81E+02	0.13799485119789 22E+02
22	0.23177464770118 71E+01	0.17294356817387 51E+01	-0.12086781154948 04E+02	0.10949478537792 63E+02
23	0.21539899676599 38E+01	0.15693438714006 43E+01	-0.88607372422015 25E+01	0.90679226952571 38E+01
24	0.20196134990689 20E+01	0.14539795899148 99E+01	-0.66343050105690 30E+01	0.77823038049316 97E+01
25	0.19086892643830 96E+01	0.13697021750902 21E+01	-0.50571618123988 73E+01	0.68719119711198 96E+01
26	0.18164806207258 51E+01	0.13097789199216 57E+01	-0.39140781234615 68E+01	0.62276842381102 11E+01
27	0.17395794479585 21E+01	0.12695828302443 30E+01	-0.30713457826835 35E+01	0.57801591606209 86E+01
28	0.16758066626378 84E+01	0.12448882989082 47E+01	-0.24449778566563 03E+01	0.54772470491242 02E+01
29	0.16237158779427 06E+01	0.12323937283791 81E+01	-0.19803977423719 57E+01	0.52837993388090 50E+01
30	0.15824392240570 63E+01	0.12293754050352 80E+01	-0.16420435862787 24E+01	0.51753833782608 68E+01
31	0.15516457934304 60E+01	0.12332611984086 28E+01	-0.14075641731574 25E+01	0.51332720821498 76E+01
32	0.15316056638585 21E+01	0.12410583081934 79E+01	-0.12648361219195 16E+01	0.51395535546350 57E+01
33	0.15236481471158 39E+01	0.12476610808397 41E+01	-0.12122777665684 93E+01	0.51655847469633 01E+01

N	SUM X	SUM X**2	SUM X**3	SUM X**4
1	0.288374427623673 2E+03	0.122682605277357 6E+04	0.641784277089959 1E+04	0.382560514578051 2E+05
2	0.294557569722162 5E+03	0.125998376768675 8E+04	0.663642508149812 2E+04	0.398549852792831 0E+05
3	0.311278756198834 9E+03	0.136629747188514 0E+04	0.735985858664265 0E+04	0.452232341883989 5E+05
4	0.339210566805165 5E+03	0.156953083975162 5E+04	0.882737910936964 3E+04	0.565138166136426 2E+05
5	0.381433099452260 9E+03	0.192307239530151 4E+04	0.116164693423228 7E+05	0.793893557838085 3E+05
6	0.444830476911956 0E+03	0.254597709700955 0E+04	0.171731116104958 3E+05	0.129754049013308 0E+06
7	0.544649616399419 9E+03	0.373704380432510 9E+04	0.297464062114602 2E+05	0.262070307473672 4E+06
8	0.721150618852855 8E+03	0.645662993119619 9E+04	0.662085159796312 8E+05	0.743308615919487 3E+06
9	0.118299879189101 9E+04	0.173260872626933 8E+05	0.289323343903574 4E+06	0.528283530858473 5E+07
10	0.327426546768879 4E+04	0.132518492597574 1E+06	0.608076473093175 9E+07	0.303971329219601 9E+09
11	0.615624640613970 4E+04	0.465018908353102 0E+06	0.392973262460690 7E+08	0.357981543334316 5E+10
12	0.610025152881876 8E+04	0.463316761730478 5E+06	0.400651160809355 0E+08	0.379691589100654 4E+10
13	0.281276503373551 7E+04	0.991946518343793 6E+05	0.403559106391847 2E+07	0.182187458561662 3E+09
14	0.101581242425580 7E+04	0.130291067006241 2E+05	0.193387783776979 6E+06	0.319140163547648 6E+07
15	0.662840376952011 8E+03	0.554536596848840 8E+04	0.537027527670270 4E+05	0.580334668374913 4E+06
16	0.507854396408180 8E+03	0.325449198078149 5E+04	0.243249911529845 4E+05	0.204236720792866 5E+06
17	0.415723001438114 4E+03	0.218834204456476 4E+04	0.135394141308037 1E+05	0.947310875562892 8E+05
18	0.354275091566549 5E+03	0.159777343603885 7E+04	0.853590705057930 9E+04	0.518914096751270 7E+05
19	0.310359457695241 1E+03	0.123460102350173 3E+04	0.586288498347789 8E+04	0.318591783947960 3E+05
20	0.277472648936073 E+03	0.994908103670158 2E+03	0.429113337758710 8E+04	0.212818684849780 9E+05
21	0.252003031671131 1E+03	0.828456032401450 2E+03	0.330048185899286 3E+04	0.151811394951766 2E+05
22	0.231774647701187 1E+03	0.708409005662230 5E+03	0.264266956963636 1E+04	0.114192289893544 2E+05
23	0.215398996765993 8E+03	0.619332321346629 8E+03	0.218809854010846 0E+04	0.897633465626368 4E+04
24	0.201961349906892 0E+03	0.551827847963715 7E+03	0.186423867454783 1E+04	0.732457361343662 9E+04
25	0.190868926438309 6E+03	0.499909986131060 5E+03	0.162818334822462 9E+04	0.617230067168480 9E+04
26	0.181648062072585 1E+03	0.459628297619501 3E+03	0.145338950464844 2E+04	0.534965332932544 1E+04
27	0.173957944795852 1E+03	0.428302365770156 1E+03	0.132283606959704 2E+04	0.475337165978483 4E+04
28	0.167580666263788 4E+03	0.404076738646068 6E+03	0.122530009274627 1E+04	0.431852069426054 4E+04
29	0.162371587794270 6E+03	0.385652304337864 3E+03	0.115323888506190 3E+04	0.400324918213464 7E+04
30	0.158243922405706 3E+03	0.372119554881924 8E+03	0.110154260085670 2E+04	0.378023264187160 5E+04
31	0.155164579343046 0E+03	0.362853325469498 3E+03	0.106678215732117 1E+04	0.363172330568762 8E+04
32	0.153160563858521 E+03	0.357446363467504 6E+03	0.104675383033273 3E+04	0.354664722331 6597E+04
33	0.152364814711583 9E+03	0.355668814624087 4E+03	0.104021387926787 7E+04	0.351893330531793 1E+04

```
MODULATION TECHNIQUE = QPRS
QPRS CLASS 3 FILTER
BIPOLAR LOGIC
BAUD OR SYMBOL RATE =        1200 HZ
BITS PER BINARY CODE WORD =           2
BIT RATE =  0.2400000000000000E+04
CARRIER FREQUENCY =        2400 HZ
MAXIMUM CARRIER AMPLITUDE =           1 VOLT(S)
INITIAL PHASE ANGLE =           0 DEGREES
TIME BETWEEN SAMPLES =  0.6784342273548912E-04 SEC
NUMBER OF SAMPLES GENERATED =          64
SEED FOR RANDOM NUMBER GENERATOR =           1
NUMBER OF TIMES SIMULATION REPEATS =         100
   SUM OF VARIANCES         SUM OF VARIANCES**2
 0.2212313780321216E+04   0.9349388767869366E+06
   SUM OF SKEWNESS           SUM OF SKEWNESS**2
-0.7876831322230185E+06   0.1444178694786092E+12
   SUM OF KURTOSIS           SUM OF KURTOSIS**2
 0.2375351460755018E+07   0.1504008831530188E+13
    MEAN VARIANCE         VARIANCE OF THE VARIANCES
 0.6703981152488534E+02   0.2458205556000923E+05
    MEAN SKEWNESS         VARIANCE OF THE SKEWNESS
-0.2386918582493996E+05   0.3925516075762725E+10
    MEAN KURTOSIS         VARIANCE OF THE KURTOSIS
 0.7198034729560660E+05   0.4165719401362236E+11
```

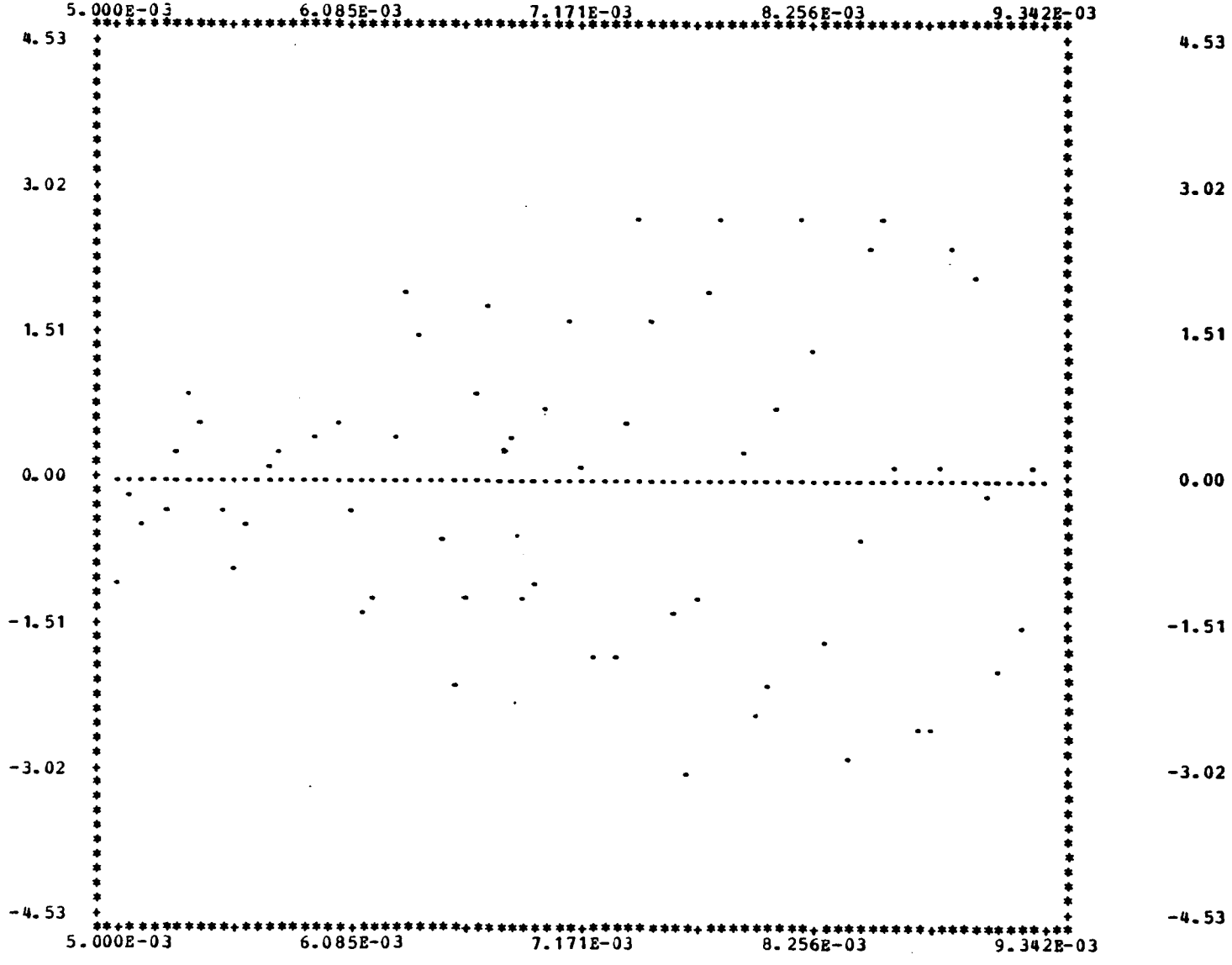

5.000E-03
6.085E-03
7.171E-03
8.256E-03
9.342E-03
7.83
5.22
2.61
0.00
-2.61
-5.22
-7.83

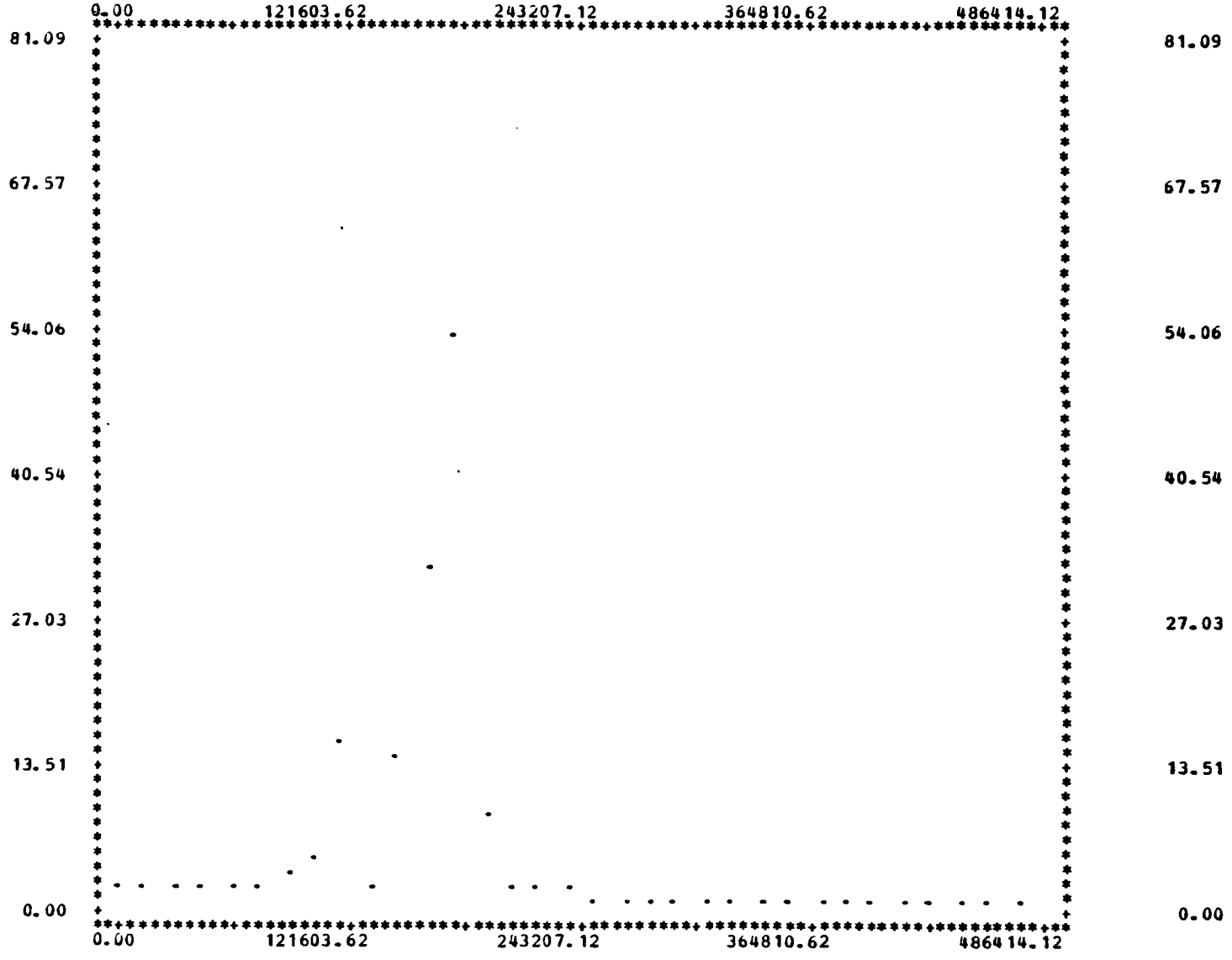

0.00
121603.62
243207.12
364810.62
486414.12
134.35
111.96
89.57
67.18
44.78
22.39
0.00

N	MEAN	VARIANCE	SKEWNESS	KURTOSIS
1	0.2859193411467364E+01	0.4053038110163085E+01	-0.2388872277111327E+02	0.5315861712866549E+02
2	0.2941193920455125E+01	0.3960147196244054E+01	-0.2682550486427233E+02	0.5214126306947926E+02
3	0.3144223362768181E+01	0.3965144755612565E+01	-0.3529891766507135E+02	0.5435317557178752E+02
4	0.3472939022748247E+01	0.4207631276200674E+01	-0.5205508235858120E+02	0.6262547090742004E+02
5	0.3969306386409815E+01	0.4860943786245437E+01	-0.8500175932943047E+02	0.8273400379153577E+02
6	0.4735423482398556E+01	0.6300931752245403E+01	-0.1566149435795907E+03	0.1321350345761201E+03
7	0.6018537756096500E+01	0.9736414158625507E+01	-0.3457034404393740E+03	0.2903044306984410E+03
8	0.8654258686250665E+01	0.2089270222273523E+02	-0.1095988817483587E+04	0.1229022976292699E+04
9	0.2281137743621147E+02	0.1439177760640681E+03	-0.2193236345074323E+05	0.5403321656593276E+05
10	0.4754546111151572E+02	0.4646395551694064E+03	-0.2077441296070791E+06	0.5503560665874106E+06
11	0.4244750014125263E+02	0.3748277455711410E+03	-0.1463406628119970E+06	0.4240689354081044E+06
12	0.4734966449442010E+02	0.4555200840149120E+03	-0.2070845754790160E+06	0.4079376089497462E+06
13	0.4636502495407004E+02	0.5858023419797556E+03	-0.1907034976555182E+06	0.9229169440607270E+06
14	0.1775571715328273E+02	0.7836469157643520E+02	-0.1046518127146919E+05	0.1312130526653664E+05
15	0.8010362291212992E+01	0.1596765263482768E+02	-0.8825082674162869E+03	0.6106026606095356E+03
16	0.5748360868725290E+01	0.7781191394513882E+01	-0.3051632597123039E+03	0.1547481205147622E+03
17	0.4587361831012248E+01	0.4764166354630095E+01	-0.1458247303643428E+03	0.6122038538085117E+02
18	0.3859990888844249E+01	0.3291449429627744E+01	-0.8182688020467373E+02	0.3123619875679458E+02
19	0.3356671073359412E+01	0.2458185419647806E+01	-0.5067489115592908E+02	0.1885213610174446E+02
20	0.2985704246868344E+01	0.1947285393959427E+01	-0.3354116060432313E+02	0.1288351792221858E+02
21	0.2700289329982362E+01	0.1617844173346278E+01	-0.2327974567069668E+02	0.9657495559635674E+01
22	0.2474887178278520E+01	0.1393588316418582E+01	-0.1674673133820691E+02	0.7685551297040409E+01
23	0.2293436153883331E+01	0.1235603072617695E+01	-0.1238328622708189E+02	0.6375226655986307E+01
24	0.2143603457021659E+01	0.1129741939228789E+01	-0.9340535778002867E+01	0.5530298158106696E+01
25	0.2017352208785796E+01	0.1065020962976242E+01	-0.7145984024478282E+01	0.5008270937596276E+01
26	0.1909802109242861E+01	0.1031153159062995E+01	-0.5524790323883738E+01	0.4698582156733970E+01
27	0.1817803178471110E+01	0.1020234834778769E+01	-0.4307541772447900E+01	0.4531591295706342E+01
28	0.1739297395622849E+01	0.1026428800839872E+01	-0.3385787401612433E+01	0.4465390207474467E+01
29	0.1673014887490447E+01	0.1045320201357048E+01	-0.2688643953287969E+01	0.4474563821840981E+01
30	0.1618309679485713E+01	0.1073390686558981E+01	-0.2169671349142156E+01	0.4543413084630725E+01
31	0.1575127269274742E+01	0.1107432254242980E+01	-0.1799539780561719E+01	0.4660898333893368E+01
32	0.1544328394719743E+01	0.1143084440129354E+01	-0.1563241471185823E+01	0.4811604062719301E+01
33	0.1530498278895295E+01	0.1165859218662538E+01	-0.1470070156508873E+01	0.4923039669179051E+01

N	SUM X	SUM X**2	SUM X**3	SUM X**4
1	0.2859193411467364E+03	0.1218749469323984E+04	0.6430051688814439E+04	0.3912446482248652E+05
2	0.2941193920455125E+03	0.1257116740200380E+04	0.6679597518767498E+04	0.4099902143818628E+05
3	0.3144223362768182E+03	0.1381163386303369E+04	0.7520938683542618E+04	0.4741984235567450E+05
4	0.3472939022742847E+03	0.1622686041912871E+04	0.9288699702861901E+04	0.6151115249264073E+05
5	0.3969306386409816E+03	0.2056772753757673E+04	0.1284062910987031E+05	0.9218510865065588E+05
6	0.4735423482398556E+03	0.2866215799237462E+04	0.2057189533450305E+05	0.1680977610919490E+06
7	0.6018537756096500E+03	0.4586184673859834E+04	0.4099147345992035E+05	0.4127460555429674E+06
8	0.8654258686250665E+03	0.9557996860905298E+04	0.1235740116398711E+06	0.1788354599317327E+07
9	0.2281137743621147E+04	0.6628375388407253E+05	0.2238763050033797E+07	0.8396436177827680E+08
10	0.4754546111151572E+04	0.2720564031924366E+06	0.1757861427920527E+08	0.1241206409612254E+10
11	0.4244750014125263E+04	0.2172869736357072E+06	0.1267544219743408E+08	0.8194713555490152E+09
12	0.4734966449442010E+04	0.2692955610908911E+06	0.1709630770902248E+08	0.1164219318416618E+10
13	0.4636502495407004E+04	0.2729659857551496E+06	0.1846793135009938E+08	0.1382943828406290E+10
14	0.1775571715328273E+04	0.3928465362880496E+05	0.9830103694208338E+06	0.2663547206451179E+08
15	0.8010362291212992E+03	0.7997388014496647E+04	0.9110191858964195E+05	0.1136318940332887E+07
16	0.5748360868725290E+03	0.4074703215766091E+04	0.3314353704488787E+05	0.2972652853806565E+06
17	0.4587361831012248E+03	0.2576041325971184E+04	0.1666045021145127E+05	0.1194261062431947E+06
18	0.3859990888844249E+03	0.1815806459729208E+04	0.9864395218518330E+04	0.5970034454704064E+05
19	0.3356671073359412E+03	0.1370084426017916E+04	0.6475831028085240E+04	0.3429694022355070E+05
20	0.2985704246868344E+03	0.1084224238978750E+04	0.4574516714243508E+04	0.2176964006173594E+05
21	0.2700289329582362E+03	0.8893228197229408E+03	0.3417999702830066E+04	0.1492684967005119E+05
22	0.2474887178278520E+03	0.7504718978461813E+03	0.2672313390051410E+04	0.1089804187849496E+05
23	0.2293436153883331E+03	0.6483096433830683E+03	0.2170271824041791E+04	0.8386780312190165E+04
24	0.2143603457021659E+03	0.5713480300791711E+03	0.1821170103329349E+04	0.6750614232021021E+04
25	0.2017352208785796E+03	0.5124080687639409E+03	0.1572582258655386E+04	0.5647284703365001E+04
26	0.1909802109224862E+03	0.4668185723940848E+03	0.1392645430893823E+04	0.4883614843619811E+04
27	0.1817803178471110E+03	0.4314440882090652E+03	0.1261205448260302E+04	0.4345381869358503E+04
28	0.1739297395622849E+03	0.4041319943251899E+03	0.1165107349407684E+04	0.3962508764554495E+04
29	0.1673014887490447E+03	0.3833845813108151E+03	0.1095564178420125E+04	0.3690809901269517E+04
30	0.1618309679485713E+03	0.3681582998410544E+03	0.1046630041989700E+04	0.3501983236805778E+04
31	0.1575127269274742E+03	0.3577383846113457E+03	0.1014291293664126E+04	0.3377938995533378E+04
32	0.1544328394719743E+03	0.3516603786465719E+03	0.9959231388592782E+03	0.3307531236388883E+04
33	0.1530498278895295E+03	0.3496625608177372E+03	0.9899818305858252E+03	0.3284702451033463E+04

```
MODULATION TECHNIQUE = QPRS
QPRS CLASS 4 FILTER
BIPOLAR LOGIC
BAUD OR SYMBOL RATE =        1200 HZ
BITS PER BINARY CODE WORD =          2
BIT RATE =  0.2400000000000000E+04
CARRIER FREQUENCY =       2400 HZ
MAXIMUM CARRIER AMPLITUDE =          1 VOLT(S)
INITIAL PHASE ANGLE =           0 DEGREES
TIME BETWEEN SAMPLES =  0.6784342273548912E-04 SEC
NUMBER OF SAMPLES GENERATED =         64
SEED FOR RANDOM NUMBER GENERATOR =          1
NUMBER OF TIMES SIMULATION REPEATS =        100
   SUM OF VARIANCES         SUM OF VARIANCES**2
 0.7516311188359808E+03   0.1147433777377834E+06
   SUM OF SKEWNESS           SUM OF SKEWNESS**2
-0.1502958760239533E+06   0.5926297375617011E+10
   SUM OF KURTOSIS           SUM OF KURTOSIS**2
 0.3178921864476514E+06   0.3285296126177793E+11
    MEAN VARIANCE         VARIANCE OF THE VARIANCES
 0.2277670057078729E+02   0.3050740650136574E+04
    MEAN SKEWNESS         VARIANCE OF THE SKEWNESS
-0.4554420485574342E+04   0.1638058362173805E+09
    MEAN KURTOSIS         VARIANCE OF THE KURTOSIS
 0.9633096559019739E+04   0.9309585979490561E+09
```

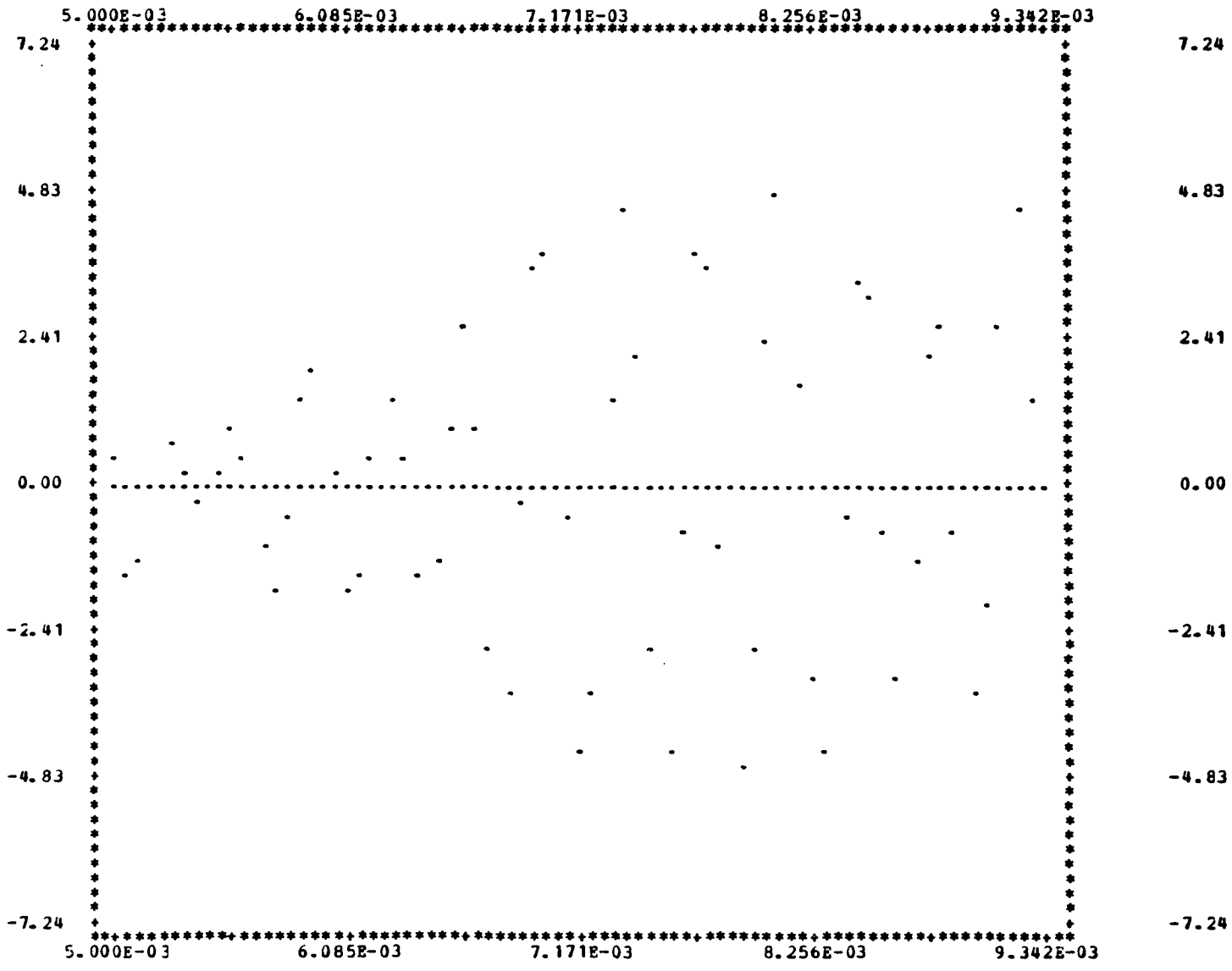

5.000E-03
6.085E-03
7.171E-03
8.256E-03
9.342E-03
4.53
3.02
1.51
0.00
-1.51
-3.02
-4.53

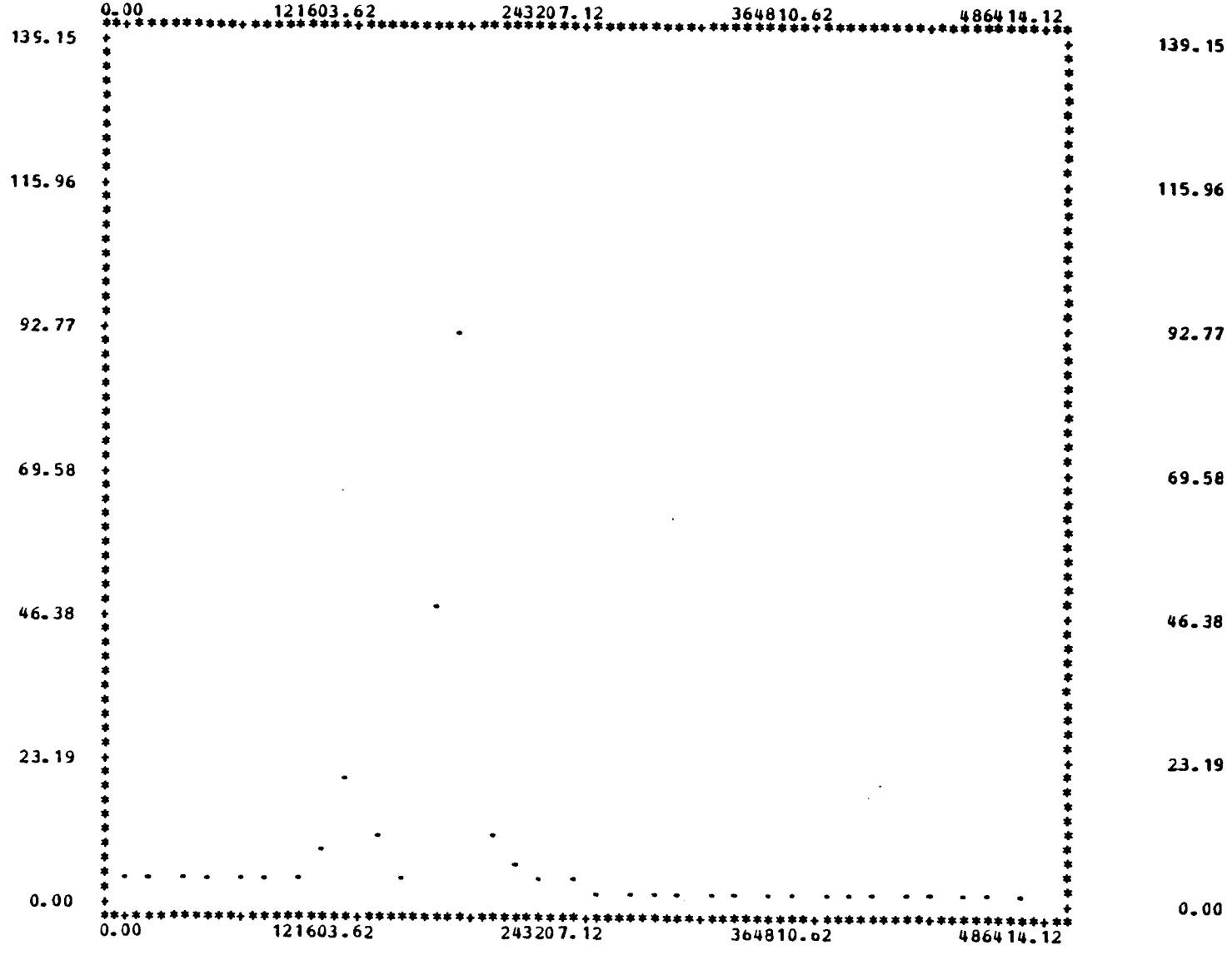

0.00
121603.62
243207.12
364810.62
486414.12
81.09
67.57
54.06
40.54
27.03
13.51
0.00

N	MEAN	VARIANCE	SKEWNESS	KURTOSIS
1	0.16397831098410 63E+01	0.17092487201512 34E+01	-0.11340990022260 32E+01	0.10623286480977 34E+02
2	0.16941190104725 07E+01	0.16597993980228 93E+01	-0.16318413912203 86E+01	0.10361622169832 15E+02
3	0.18220802679058 23E+01	0.16379753665594 96E+01	-0.30336020189085 17E+01	0.10518110114254 01E+02
4	0.20249444832126 05E+01	0.16960031798444 38E+01	-0.58527175820524 59E+01	0.11497264153385 65E+02
5	0.23295753015734 74E+01	0.18894100560861 23E+01	-0.11625566610544 61E+02	0.13968808358537 68E+02
6	0.27996184591863 39E+01	0.23510812025625 22E+01	-0.24856977287701 99E+02	0.20038859065387 44E+02
7	0.35936066907015 01E+01	0.35175604218943 56E+01	-0.62215322651401 16E+02	0.39230657396282 20E+02
8	0.52502686788252 91E+01	0.76798728266727 29E+01	-0.22401126306389 12E+03	0.16155965534640 01E+03
9	0.14687324457287 72E+02	0.56984323516529 23E+02	-0.57663317115178 94E+04	0.77001446268325 53E+04
10	0.29715319338226 15E+02	0.19770415254140 51E+03	-0.49800466002737 65E+05	0.10158875606208 18E+06
11	0.19461962250725 21E+02	0.85207332020767 75E+02	-0.13381966657306 89E+05	0.30821673177680 66E+05
12	0.24933195580196 01E+02	0.12031577580155 37E+03	-0.29539353488181 28E+05	0.32292162842539 84E+05
13	0.29556818665421 02E+02	0.22296067968986 62E+03	-0.48521163984030 28E+05	0.14309627976542 48E+06
14	0.11348435941841 10E+02	0.28833615505904 56E+02	-0.26286156640347 68E+04	0.20140173640653 28E+04
15	0.49700904848030 92E+01	0.50075216970749 69E+01	-0.19269576724989 94E+03	0.61639591976775 31E+02
16	0.35327571031563 59E+01	0.24366134523799 79E+01	-0.61953680804611 59E+02	0.15303045811772 05E+02
17	0.28124367953956 93E+01	0.15432564662609 07E+01	-0.28039943881138 45E+02	0.66346409876264 15E+01
18	0.23692464252239 95E+01	0.11161290682811 24E+01	-0.14993456151495 28E+02	0.37677658898976 74E+01
19	0.20677526811990 91E+01	0.87306319861659 04E+00	-0.88630858447893 77E+01	0.24839632403687 16E+01
20	0.18526645630485 47E+01	0.70861804910376 25E+00	-0.56115312494912 62E+01	0.17217964438680 16E+01
21	0.16910492579182 67E+01	0.59699803144483 85E+00	-0.37166801549164 90E+01	0.12751281938696 37E+01
22	0.15644851526709 45E+01	0.52280659056889 73E+00	-0.25329824475210 70E+01	0.10155778852697 64E+01
23	0.14627245375011 72E+01	0.47377584567206 25E+00	-0.17552622515532 10E+01	0.85929488262670 97E+00
24	0.13793611988785 35E+01	0.44206321198644 55E+00	-0.12241658447485 08E+01	0.76279035994580 43E+00
25	0.13102129714549 39E+01	0.42251518822749 44E+00	-0.85062679791525 97E+00	0.70263012374654 74E+00
26	0.12524666060837 59E+01	0.41165128574789 18E+00	-0.58211876344032 18E+00	0.66567613861537 49E+00
27	0.12041969573910 62E+01	0.40705684292119 03E+00	-0.38631995632335 70E+00	0.64429161199148 03E+00
28	0.11640703648477 72E+01	0.40704063612023 67E+00	-0.24267234205472 43E+00	0.63399711664060 47E+00
29	0.11311639354023 71E+01	0.41041097657061 29E+00	-0.13779795668928 86E+00	0.63220347881891 01E+00
30	0.11049050189849 23E+01	0.41620985542632 31E+00	-0.62925045690357 84E-01	0.63719234254611 06E+00
31	0.10850949578715 04E+01	0.42341437651769 91E+00	-0.12399720923725 65E-01	0.64724650333316 67E+00
32	0.10720177994822 93E+01	0.43054942851207 94E+00	0.17154493763123 83E-01	0.65965799074073 05E+00
33	0.10668961457450 31E+01	0.43459438672808 97E+00	0.27135432677034 11E-01	0.66785496309 5C203E+00

N	SUM X	SUM X**2	SUM X**3	SUM X**4
1	0.1639783109E41063E+03	0.4381044880269750E+03	0.1504001389749991E+04	0.6028224494587657E+04
2	0.1694119010472507E+03	0.4513240625687009E+03	0.1556598039393050E+04	0.6283642510452042E+04
3	0.1822080267905823E+03	0.4941572115585657E+03	0.1733826810843475E+04	0.7151628698908221E+04
4	0.2024944483212605E+03	0.5779443308139159E+03	0.2105563762633040E+04	0.9029491598424876E+04
5	0.2329575301573474E+03	0.7297437041226407E+03	0.2852037850613845E+04	0.1304690135038802E+05
6	0.2799618459186339E+03	0.1016543390755379E+04	0.4484530491794313E+04	0.2284824450720208E+05
7	0.3593606690701501E+03	0.1639639386513001E+04	0.8872323134501433E+04	0.5444320652383947E+05
8	0.5250268678825291E+03	0.3516839529825987E+04	0.2747886749340246+05	0.2395389115966948E+06
9	0.1468732445728772E+04	0.2721319799950061E+05	0.5792905345227176E+06	0.1354096031979063E+08
10	0.2971531933822615E+04	0.1078727314388748E+06	0.4459771346830303E+07	0.2026499395458332E+09
11	0.1946196225072521E+04	0.4631232333492131E+05	0.1290035804802437E+07	0.4129848397914642E+08
12	0.2493319558C19601E+04	0.7407768598838439E+05	0.2462712101444010E+07	0.8847358410767536E+08
13	0.2955681806542102E+04	0.1094336602513548E+06	0.4676695051500612E+07	0.2225674204211198E+09
14	0.1134843594184110E+04	0.1573322776769165E+05	0.2470236163714422E+06	0.4233144694722940E+07
15	0.4970090484803092E+03	0.2965924590723445E+04	0.2001280415057830E+05	0.1474973714749340E+06
16	0.3532757103156359E+03	0.1489262006775789E+04	0.7092160168252687E+04	0.3695846281762211E+05
17	0.2812436795395693E+03	0.9437624629693893E+03	0.3576862297172163E+04	0.1488190681227194E+05
18	0.2369246425223995E+03	0.6718296401034992E+03	0.2153176579687351E+04	0.7608053377373551E+04
19	0.2067752681199091E+03	0.5139933717236453E+03	0.1447004702900242E+04	0.4515033391001683E+04
20	0.1852664563048547E+03	0.4133897851788587E+03	0.1049991495597069E+04	0.2974218440064030E+04
21	0.1691049257918267E+03	0.3450675643836312E+03	0.8069812120149764E+03	0.2118738265791955E+04
22	0.1564485152670945E+03	0.2965192317591039E+03	0.6488788033678661E+03	0.1604859418202128E+04
23	0.1462724537501172E+03	0.2608601159823359E+03	0.5412606378313685E+03	0.1277349367688637E+04
24	0.1379361198878536E+03	0.2340279896838212E+03	0.4654833373950505E+03	0.1058943962953624E+04
25	0.1310212971454939E+03	0.2134948066914001E+03	0.4107767092907294E+03	0.9081718957376042E+03
26	0.1252466606083759E+03	0.1976207372245383E+03	0.3705937259151374E+03	0.8013993822923571E+03
27	0.1204196957391062E+03	0.1853076586681869E+03	0.3407906981781624E+03	0.7244971790085427E+03
28	0.1164070364847772E+03	0.1758030044075860E+03	0.3186620053773817E+03	0.6686943101747503E+03
29	0.1131163935402371E+03	0.1685838715559886E+03	0.3024018213639951E+03	0.6283944137293207E+03
30	0.1104905018984923E+03	0.1632862857850134E+03	0.2907919853211530E+03	0.5999720029213764E+03
31	0.1085094957871504E+03	0.1596611300350684E+03	0.2830162758953530E+03	0.5810861131486764E+03
32	0.1072017799482293E+03	0.1575466096633816E+03	0.2785493263166431E+03	0.5702822834014495E+03
33	0.1066896145745031E+03	0.1568515828666411E+03	0.2770931137498631E+03	0.5667653419404973E+03

```
MODULATION TECHNIQUE = QPRS
QPRS CLASS 5 FILTER
BIPOLAR LOGIC
BAUD OR SYMBOL RATE =        1200 HZ
BITS PER BINARY CODE WORD =           2
BIT RATE =  0.2400000000000000E+04
CARRIER FREQUENCY =        2400 HZ
MAXIMUM CARRIER AMPLITUDE =           1 VOLT(S)
INITIAL PHASE ANGLE =           0 DEGREES
TIME BETWEEN SAMPLES =  0.6784342273548912E-04 SEC
NUMBER OF SAMPLES GENERATED =          64
SEED FOR RANDOM NUMBER GENERATOR =           1
NUMBER OF TIMES SIMULATION REPEATS =         100
   SUM OF VARIANCES         SUM OF VARIANCES**2
 0.2422177865073176E+04   0.1348538369218457E+07
   SUM OF SKEWNESS          SUM OF SKEWNESS**2
-0.8202160755049249E+06   0.2049138949944578E+12
   SUM OF KURTOSIS          SUM OF KURTOSIS**2
 0.3582703927263425E+07   0.4782889442290025E+13
    MEAN VARIANCE         VARIANCE OF THE VARIANCES
 0.7339932924464169E+02   0.3658600433158962E+05
    MEAN SKEWNESS         VARIANCE OF THE SKEWNESS
-0.2485503259105833E+05   0.5766481178314780E+10
    MEAN KURTOSIS         VARIANCE OF THE KURTOSIS
 0.1085667856746492E+06   0.1373102122775966E+12
```


5.000E-03
6.085E-03
7.171E-03
8.256E-03
9.342E-03
7.24
4.83
2.41
0.00
-2.41
-4.83
-7.24


0.00
121603.62
243207.12
364810.62
486414.12
139.15
115.96
92.77
69.58
46.38
23.19
0.00

N	MEAN	VARIANCE	SKEWNESS	KURTOSIS
1	0.31750079699776 90E+01	0.51141571862255 71E+01	-0.36473772457226 82E+02	0.63901779029936 34E+02
2	0.32406199447650 44E+01	0.50861027017887 36E+01	-0.39457221747185 13E+02	0.63845816063397 65E+02
3	0.34184928799213 29E+01	0.51721927982439 49E+01	-0.48490172658212 98E+02	0.67412486419946 79E+02
4	0.37200509866916 12E+01	0.54893828206908 38E+01	-0.66512919252116 35E+02	0.77343837904010 80E+02
5	0.41880736115071 33E+01	0.62150595541914 26E+01	-0.10169997970303 95E+03	0.99598328699899 236E+02
6	0.49212143565879 52E+01	0.77604981673463 90E+01	-0.17692559279860 75E+03	0.15261834673861 06E+03
7	0.61646708833203001E+01	0.11348962854489 64E+02	-0.37198888183603 94E+03	0.31365035936168 17E+03
8	0.87349387424476 45E+01	0.22397947820700 82E+02	-0.11268945804188 45E+04	0.11636669610309 95E+04
9	0.20014588959166 63E+02	0.13312459966829 23E+03	-0.14517371311886 66E+05	0.43149970439749 16E+05
10	0.54092321561075 56E+02	0.71864078190467 12E+03	-0.30609429614864 85E+06	0.12458837176534 42E+07
11	0.30710917233369 24E+02	0.23598668646713 04E+03	-0.54168095418989 84E+05	0.18543169047370 86E+06
12	0.41250661180587 45E+02	0.40076722195726 11E+03	-0.13503875600752 82E+06	0.34657582384665 81E+06
13	0.54051984835919 09E+02	0.77113661400960 73E+03	-0.29964351155120 78E+06	0.17533590999453 96E+07
14	0.15664446964869 08E+02	0.48485736561961 7E+02	-0.71517240185086 88E+04	0.56374262654509 37E+04
15	0.80274263391579 71E+01	0.11877591493647 16E+02	-0.89622000924433 05E+03	0.37184185368134 51E+03
16	0.57741[illegible]249598E+01	0.59666047586072 88E+01	-0.31159224733163 41E+03	0.10101497600397 74E+03
17	0.45963104764288 44E+01	0.38603128286199 21E+01	-0.14723288359085 98E+03	0.46849051544965 22E+02
18	0.38634179600178 63E+01	0.28487110222750 91E+01	-0.81982247525097 45E+02	0.27960503529527 78E+02
19	0.33631558255347 46E+01	0.22783439994121 12E+01	-0.50722450284644 29E+02	0.19157314979243 38E+02
20	0.30032610818379 05E+01	0.19115273211942 57E+01	-0.33853131233793 64E+02	0.14068668378631 03E+02
21	0.27319042809914 12E+01	0.16718311191240 26E+01	-0.23890829145344 72E+02	0.11056736562283 76E+02
22	0.25212344418456 05E+01	0.15093307722747 32E+01	-0.17615028511108 26E+02	0.91368220960885 69E+01
23	0.23542052829864 97E+01	0.13965221778123 53E+01	-0.13459828333860 97E+02	0.78439450201224 83E+01
24	0.22198518574234 73E+01	0.13170212711774 17E+01	-0.10599865563867 77E+02	0.69356587921546 44E+01
25	0.21106035540026 56E+01	0.12615456122290 99E+01	-0.85694586944234 58E+01	0.62866918886975 18E+01
26	0.20210755111127 47E+01	0.12245948435706 20E+01	-0.70926190248019 98E+01	0.58265853473606 73E+01
27	0.19475703391225 54E+01	0.12019593964553 56E+01	-0.60006603480076 21E+01	0.55061108187042 12E+01
28	0.18875714465760 71E+01	0.11902433997523 15E+01	-0.51872789992996 26E+01	0.52902008505725 00E+01
29	0.18393966496484 18E+01	0.11866604689067 29E+01	-0.45840060840009 71E+01	0.51539590267050 35E+01
30	0.18020071983129 46E+01	0.11887596069116 89E+01	-0.41465982169596 48E+01	0.50787525976854 12E+01
31	0.17749073284154 59E+01	0.11941481357197 85E+01	-0.38472890246275 49E+01	0.50489518074506 10E+01
32	0.17581123519771 82E+01	0.12001732295859 94E+01	-0.36704634278019 61E+01	0.50486518208808 67E+01
33	0.17522633447758 10E+01	0.12032020488648 05E+01	-0.36110326996010 22E+01	0.50552887180970 53E+01

N	SUM X	SUM X**2	SUM X**3	SUM X**4
1	0.31750079769577690E+03	0.15143691223785 17E+04	0.87608897315126 76E+04	0.56544579758569 09E+05
2	0.32406199447650 44E+03	0.15536859301179 85E+04	0.90586711391231 97E+04	0.58995321605987 35E+05
3	0.34184928799213 29E+03	0.16806564440334 33E+04	0.10049700282730 70E+05	0.67288148183253 67E+05
4	0.37200509866916 12E+03	0.19273268336069 17E+04	0.12090214473181 21E+05	0.85061680388075 75E+05
5	0.41880736115071 33E+03	0.23692869534051 92E+04	0.16090254417548 81E+05	0.12246037915293 54E+06
6	0.49212143565879 52E+03	0.31901243929160 30E+04	0.24561628456601 15E+05	0.21115523404348 20E+06
7	0.61646768332030 01E+03	0.49238713683774 53E+04	0.46262690987754 24E+05	0.48267867344497 81E+06
8	0.87349387424476 45E+03	0.98473123176806 66E+04	0.13009772060932 52E+06	0.19003678841479 010E+07
9	0.20014588959166 63E+04	0.53237712487600 41E+05	0.16647389122672 98E+07	0.57774482751646 25E+08
10	0.54092321561075 56E+04	0.36374336259524 24E+06	0.27832543340388 06E+08	0.23292690916475 67E+10
11	0.30710917233369 24E+04	0.11767872569173 15E+06	0.52366231851794 29E+07	0.26275514677529 67E+09
12	0.41250661180587 45E+04	0.20983765975733 13E+06	0.12123627606411 24E+08	0.76135833959085 20E+09
13	0.54051984835919 09E+04	0.36850423125719 39E+06	0.29206480791596 32E+08	0.25909933165179 71E+10
14	0.15664446964869 08E+04	0.29333898663483 03E+05	0.61425051806701 35E+06	0.13927159075165 43E+08
15	0.80274263391579 71E+03	0.76198389209317 83E+04	0.80993171311532 05E+05	0.93747361979647 26E+06
16	0.57741088032495 99E+03	0.39247271182765 73E+04	0.30158112984497 88E+05	0.25500948918533 01E+06
17	0.45963104764288 44E+03	0.24947779696261 27E+04	0.15451819999623 744E+05	0.10643447964747 76E+06
18	0.38634179600178 63E+03	0.17746222245840 93E+04	0.93848977048476 50E+04	0.55734757410907 85E+05
19	0.33631558255347 47E+03	0.13566377666246 29E+04	0.63533441740621 20E+04	0.33697162516616 12E+05
20	0.30032610818379 05E+03	0.10911989173664 50E+04	0.46422371746951 44E+04	0.22527182485508 63E+05
21	0.27319042809911 412E+03	0.91184138084319 90E+03	0.35914470010976 54E+04	0.16229809812792 17E+05
22	0.25212344418456 05E+03	0.78508605753005 07E+03	0.29053305509121 02E+04	0.12392780157531 17E+05
23	0.23542052829864 97E+03	0.69248394704757 64E+03	0.24363086763293 834E+04	0.99140969615886 93E+04
24	0.22198518574234 73E+03	0.62315933273720 87E+03	0.21044291965743 28E+04	0.82398317780164 03E+04
25	0.21106035540026 56E+03	0.57035775182754 51E+03	0.18635219523044 70E+04	0.70700024192096 24E+04
26	0.20210755111127 47E+03	0.52970951167545 64E+03	0.16855376193481 39E+04	0.62322257071082 18E+04
27	0.19475703391225 54E+03	0.49829700283207 44E+03	0.15527333084014 96E+04	0.56226303550346 76E+04
28	0.18875714465760 71E+03	0.47412669316840 75E+03	0.14535309311368 41E+04	0.51762566086018 79E+04
29	0.18393966496484 18E+03	0.45581738989554 88E+03	0.13802103241284 15E+04	0.48513702858036 80E+04
30	0.18020071983129 46E+03	0.44241019536142 47E+03	0.13275732596946 73E+04	0.46207664309782 44E+04
31	0.17749073284154 59E+03	0.43325026788254 89E+03	0.12921491191527 55E+04	0.44667848781810 24E+04
32	0.17581123519771 82E+03	0.42791305394648 74E+03	0.12717195259316 49E+04	0.43784064092096 04E+04
33	0.17522633447758 10E+03	0.42615968578210 65E+03	0.12650433534024 02E+04	0.43495887400602 86E+04

APPENDIX D

FORTRAN PROGRAM F-TEST

```
C     ****************************************************************FTE00020
CC    *                                                               *FTE00030
CC    * THIS PROGRAM COMPUTES THE F STATISTIC FOR A GIVEN SET OF INPUT *FTE00040
CC    * DATA.  THE DATA MUST BE FORMATTED AS E23.16 IN TWO COLUMNS     *FTE00050
CC    * STARTING IN THE SECOND COLUMN WITH TWO BLANK COLUMNS BETWEEN   *FTE00060
CC    * THE DATA.  THIS PROGRAM IS NOT MEANT TO BE EXTREMELY VERSATILE *FTE00070
CC    * AND WAS DESIGNED TO DO THE ANALYSIS FOR SPECIFIC SETS OF DATA. *FTE00080
CC    * THE INPUT DATA MUST BE THE SUM OF THE OBSERVATIONS FOR A SET   *FTE00090
CC    * OF N OBSERVATIONS AND THE SUM OF THE SQUARE OF THE OBSERVATIONS*FTE00100
CC    * FOR THE SET OF N OBSERVATIONS.  THE INDEX FOR THE DO LOOP MUST *FTE00110
CC    * BE CHANGED TO ACCOUNT FOR THE NUMBER OR GROUPS FOR WHICH THE   *FTE00120
CC    * DATA IS REPRESENTATIVE.  THE DATA IS READ IN FROM THE FILE     *FTE00130
CC    * WHICH CONTAINS THE PROGRAM FORMATTED AS DESCRIBED ABOVE.       *FTE00140
CC    * THE INITIALIZATION MUST BE CHANGED FOR THE NUMBER OF GROUPS    *FTE00150
CC    * AND THE NUMBER OF OBSERVATIONS PER GROUP TO REFLECT THE DATA.  *FTE00160
CC    *                                                               *FTE00170
CC    * WRITTEN BY LCDR CRAIG D. CARLSON IN PARTIAL FULFILLMENT OF THE *FTE00180
CC    * REQUIREMENTS FOR A MASTER OF SCIENCE DEGREE IN SYSTEMS         *FTE00190
CC    * TECHNOLOGY (SPACE SYSTEMS OPERATIONS)*                         FTE00200
CC    *                                                               *FTE00210
CC    * 27 SEPTEMBER 1985                                             *FTE00220
CC    *                                                               *FTE00230
CC    ****************************************************************FTE00240
CC    *** VARIABLE DECLARATIONS ***                                    FTE00250
      DOUBLE PRECISION SUMX(15),SUMXSQ(15),XA,XE,TOTAL,R,N,X,Y,Z,FSTAT, FTE00260
     *DF1,DF2                                                          FTE00270
C                                                                      FTE00280
CC    *** INITIALIZE THE NUMBER OF OBSERVATIONS AND THE NUMBER OF GROUPSFTE00290
C                                                                      FTE00300
      R=15.D0                                                          FTE00310
      N=33.D0                                                          FTE00320
C                                                                      FTE00330
CC    *** DO THE F-TEST FOR EACH OF THE SETS OF DATA PRESENT ***       FTE00340
C                                                                      FTE00350
      DO 100 J=1,3                                                     FTE00360
C                                                                      FTE00370
C                                                                      FTE00380
      DO 10 I=1,15                                                     FTE00390
         READ(5,20)SUMX(I),SUMXSQ(I)                                   FTE00400
20       FORMAT(1X,E23.16,2X,E23.16)                                   FTE00410
10    CONTINUE                                                         FTE00420
                                                                       FTE00430
                                                                       FTE00440
```

```
C                                                                       FTE00450
      X=0.D0                                                            FTE00460
      Y=0.D0                                                            FTE00470
      Z=0.D0                                                            FTE00480
C                                                                       FTE00490
      DO 30 I=1,15                                                      FTE00500
           X=X+SUMX(I)                                                  FTE00510
           Y=Y+SUMXSQ(I)                                                FTE00520
           Z=Z+SUMX(I)**2                                               FTE00530
30    CONTINUE                                                          FTE00540
C                                                                       FTE00550
C                                                                       FTE00560
      XA=(Z/N)-(X**2/(R*N))                                             FTE00570
      XE=Y-(Z/N)                                                        FTE00580
      TOTAL=Y-(X**2/(R*N))                                              FTE00590
C                                                                       FTE00600
      DF1=R-1.D0                                                        FTE00610
      DF2=R*(N-1.D0)                                                    FTE00620
C                                                                       FTE00630
      FSTAT=(XA/DF1)/(XE/DF2)                                           FTE00640
C                                                                       FTE00650
C     *** OUTPUT THE RESULTS-THE FORMATS REPRESENT VARIABLE DEFINITIONS  FTE00660
C                                                                       FTE00670
      WRITE(6,31) J                                                     FTE00680
31    FORMAT('1',' F-STATISTICS FOR THE',I4,' COMPONENT')               FTE00690
C                                                                       FTE00700
      WRITE(6,40) X                                                     FTE00710
40    FORMAT('0',' SUM OF SUMS =',E23.16)                               FTE00720
C                                                                       FTE00730
      WRITE(6,41) Y                                                     FTE00740
41    FORMAT('0',' SUM OF SUMS**2 =',E23.16)                            FTE00750
C                                                                       FTE00760
      WRITE(6,42) Z                                                     FTE00770
42    FORMAT('0',' SUM OF SQUARE OF SUMS =',E23.16)                     FTE00780
C                                                                       FTE00790
      WRITE(6,50) XA                                                    FTE00800
50    FORMAT('0',' AMONG GROUP SUM OF SQUARES =',E23.16)                FTE00810
C                                                                       FTE00820
      WRITE(6,51) XE                                                    FTE00830
51    FORMAT('0',' WITHIN GROUP SUM OF SQUARES =',E23.16)               FTE00840
C                                                                       FTE00850
      WRITE(6,60) TOTAL                                                 FTE00860
60    FORMAT('0',' TOTAL SUM OF SQUARES =',E23.16)                      FTE00870
C                                                                       FTE00880
      WRITE(6,70) DF1                                                   FTE00890
70    FORMAT('0',' DEGREES OF FREEDOM IN NUMERATOR =',E23.16)           FTE00900
C                                                                       FTE00910
      WRITE(6,71) DF2                                                   FTE00920
71    FORMAT('0',' DEGREES OF FREEDOM IN DENOMINATOR =',E23.16)         FTE00930
C                                                                       FTE00940
```

```
      WRITE(6,80) FSTAT                                                  FTE00950
80    FORMAT('0',' F-STATISTIC =',E23.16)                                FTE00960
C                                                                        FTE00970
100   CONTINUE                                                           FTE00980
C                                                                        FTE00990
      WRITE(6,81)                                                        FTE01000
81    FORMAT('1')                                                        FTE01010
      STOP                                                               FTE01020
      END                                                                FTE01030
```

APPENDIX E

RESULTS OF F-TEST

F-STATISTICS FOR THE 1 COMPONENT
SUM OF SUMS = 0.2091487882294994D 02
SUM OF SUMS**2 = 0.18106554694534O4D 02
SUM OF SQUARE OF SUMS = 0.41280975106790 48D 02
AMCNG GROUP SUM OF SQUARES = 0.1211883136154515D 00
WITHIN GROUP SUM OF SQUARES = 0.1769374494346614D 02
TOTAL SUM OF SQUARES = 0.1781493325708159D 02
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.7265063214387600D 00

F-STATISTICS FOR THE 2 COMPONENT
SUM OF SUMS = 0.2305792717805020D 02
SUM OF SUMS**2 = 0.1894701234015291D 02
SUM OF SQUARE OF SUMS = 0.4663825126655468D 02
AMCNG GROUP SUM OF SQUARES = 0.1119371755000360D 00
WITHIN GROUP SUM OF SQUARES = 0.1848062982748736D 02
TOTAL SUM OF SQUARES = 0.1859256700298739D 02
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.6424746465014737D 00

F-STATISTICS FOR THE 3 COMPONENT
SUM OF SUMS = 0.2693734535827635D 02
SUM OF SUMS**2 = 0.2124674921974186D 02
SUM OF SQUARE OF SUMS = 0.5921566381358867D 02
AMCNG GROUP SUM OF SQUARES = 0.1084095881685183D 00
WITHIN GROUP SUM OF SQUARES = 0.2065459258160598D 02
TOTAL SUM OF SQUARES = 0.2076300216977449D 02
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.5567362242775778D 00

```
F-STATISTICS FOR THE   4 COMPONENT
SUM OF SUMS = 0.3175688631627065D 02
SUM OF SUMS**2 = 0.26211045423002l4D 02
SUM OF SQUARE OF SUMS = 0.8070513093726266D 02
AMONG GROUP SUM OF SQUARES = 0.1347180903696015D 00
WITHIN GROUP SUM OF SQUARES = 0.2540399411362951D 02
TOTAL SUM OF SQUARES = 0.2553871220399911D 02
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04

F-STATISTIC = 0.5624997485041869D 00
F-STATISTICS FOR THE   5 COMPONENT
SUM OF SUMS = 0.3669104349531116D 02
SUM CF SUMS**2 = 0.3380633740317695D 02
SUM OF SQUARE OF SUMS = 0.1069682473356158D 03
AMCNG GROUP SUM OF SQUARES = 0.1721940248396148D 00
WITHIN GROUP SUM OF SQUARES = 0.3273665492982079D 02
TOTAL SUM OF SQUARES = 0.3290884895466041D 02
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.5579331866788867D 00

F-STATISTICS FOR THE   6 COMPONENT
SUM OF SUMS = 0.4160097878948431D 02
SUM OF SUMS**2 = 0.4313236642020094D 02
SUM CF SQUARE OF SUMS = 0.1408215358007360D 03
AMCNG GROUP SUM OF SQUARES = 0.2544544005119440D 00
WITHIN GROUP SUM OF SQUARES = 0.4172415106219358D 02
TCTAL SUM OF SQUARES = 0.4197860546270552D 02
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.6468757561623442D 00
```

F-STATISTICS FOR THE 7 COMPONENT
SUM CF SUMS = 0.4810387092348111D 02
SUM OF SUMS**2 = 0.5206409381611744D 02
SUM OF SQUARE OF SUMS = 0.1964653659903027D 03
AMONG GROUP SUM OF SQUARES = 0.4219987280210724D 00
WITHIN GROUP SUM OF SQUARES = 0.5009944015621441D 02
TOTAL SUM OF SQUARES = 0.5052143888423548D 02
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.8934632362547178D 00

F-STATISTICS FOR THE 8 COMPONENT
SUM OF SUMS = 0.5623255941290002D 02
SUM CF SUMS**2 = 0.8590285622098203D 02
SUM OF SQUARE OF SUMS = 0.3070018365343552D 03
AMCNG GROUP SUM OF SQUARES = 0.9619512065933320D 00
WITHIN GROUP SUM OF SQUARES = 0.8283283785563848D 02
TOTAL SUM OF SQUARES = 0.8379478906223181D 02
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.1231824736914027D 01

F-STATISTICS FOR THE 9 COMPONENT
SUM OF SUMS = 0.9931605295569187D 02
SUM OF SUMS**2 = 0.4043531056175881D 03
SUM OF SQUARE OF SUMS = 0.1416642417283150D 04
AMCNG GROUP SUM OF SQUARES = 0.7590638589699638D 01
WITHIN GROUP SUM OF SQUARES = 0.3901866814447566D 03
TOTAL SUM OF SQUARES = 0.3977773200344562D 03
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.2063499133280512D 01

F-STATISTICS FOR THE 10 COMPONENT
SUM OF SUMS = 0.20299426287819240 03
SUM OF SUMS**2 = 0.17575647578199240 04
SUM OF SQUARE OF SUMS = 0.76556922504519600 04
AMONG GROUP SUM OF SQUARES = 0.49085808663545810 02
WITHIN GROUP SUM OF SQUARES = 0.16810078353154040 04
TOTAL SUM OF SQUARES = 0.17300936439789500 04
DEGREES OF FREEDOM IN NUMERATOR = 0.14000000000000000 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.14850000000000000 04
F-STATISTIC = 0.30973096841926410 01

F-STATISTICS FOR THE 11 COMPONENT
SUM OF SUMS = 0.20047733022472460 03
SUM OF SUMS**2 = 0.18118750775616170 04
SUM OF SQUARE OF SUMS = 0.79804398631082660 04
AMONG GROUP SUM OF SQUARES = 0.53010292008393790 02
WITHIN GROUP SUM OF SQUARES = 0.17320706789305340 04
TOTAL SUM OF SQUARES = 0.17850809709389280 04
DEGREES OF FREEDOM IN NUMERATOR = 0.14000000000000000 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.14850000000000000 04
F-STATISTIC = 0.32463325375329120 01

F-STATISTICS FOR THE 12 COMPONENT
SUM OF SUMS = 0.22108346527731090 03
SUM OF SUMS**2 = 0.21658597652189440 04
SUM OF SQUARE OF SUMS = 0.94142799328204380 04
AMONG GROUP SUM OF SQUARES = 0.61557533582188420 02
WITHIN GROUP SUM OF SQUARES = 0.20717169658907400 04
TOTAL SUM OF SQUARES = 0.21332744994729280 04
DEGREES OF FREEDOM IN NUMERATOR = 0.14000000000000000 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.14850000000000000 04
F-STATISTIC = 0.31517314545855650 01

F-STATISTICS FOR THE 13 COMPONENT
SUM OF SUMS = 0.19795834344717910 03
SUM OF SUMS**2 = 0.1904701075347160D 04
SUM OF SQUARE OF SUMS = 0.7185446672840862D 04
AMONG GROUP SUM OF SQUARES = 0.4572946290150773D 02
WITHIN GROUP SUM OF SQUARES = 0.1832846608618751D 04
TOTAL SUM OF SQUARES = 0.18785760715202590 04
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.26464786714598640 01

F-STATISTICS FOR THE 14 COMPONENT
SUM OF SUMS = 0.9227736272614619D 02
SUM OF SUMS**2 = 0.2600827196886254D 03
SUM OF SQUARE OF SUMS = 0.9540726819081072D 03
AMONG GROUP SUM OF SQUARES = 0.3863985704619236D 01
WITHIN GROUP SUM OF SQUARES = 0.2505419928695444D 03
TOTAL SUM OF SQUARES = 0.2544059785741636D 03
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.1635887377498235D 01

F-STATISTICS FOR THE 15 COMPONENT
SUM OF SUMS = 0.6880926336210145D 02
SUM OF SUMS**2 = 0.1018300362534556D 03
SUM OF SQUARE OF SUMS = 0.3695062863415098D 03
AMONG GROUP SUM OF SQUARES = 0.5385863804584070D 00
WITHIN GROUP SUM OF SQUARES = 0.9813497339004056D 02
TOTAL SUM OF SQUARES = 0.9867355977049896D 02
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.5821433971075596D 00

F-STATISTICS FOR THE 16 COMPONENT
SUM OF SUMS = 0.6961387326450431D 02
SUM OF SUMS**2 = 0.1137157033969463D 03
SUM OF SQUARE OF SUMS = 0.3597053597878584D 03
AMONG GROUP SUM OF SQUARES = 0.3663260306209388D 00
WITHIN GROUP SUM OF SQUARES = 0.1101186497990677D 03
TOTAL SUM OF SQUARES = 0.1104849758296886D 03
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.3528623485827816D 00

F-STATISTICS FOR THE 17 COMPONENT
SUM OF SUMS = 0.7052854165691613D 02
SUM OF SUMS**2 = 0.1548042004914454D 03
SUM OF SQUARE OF SUMS = 0.3820324148469269D 03
AMONG GROUP SUM OF SQUARES = 0.5041406896350342D 00
WITHIN GROUP SUM OF SQUARES = 0.1509838763429761D 03
TOTAL SUM OF SQUARES = 0.1514880170326111D 03
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.3541763825767676D 00

F-STATISTICS FOR THE 18 COMPONENT
SUM OF SUMS = 0.7950452893326030D 02
SUM OF SUMS**2 = 0.1454127656122291D 03
SUM OF SQUARE OF SUMS = 0.5157161626920556D 03
AMONG GROUP SUM OF SQUARES = 0.9431815463208067D 00
WITHIN GROUP SUM OF SQUARES = 0.1402556039853086D 03
TOTAL SUM OF SQUARES = 0.1411987855316294D 03
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.7133020797581566D 00

```
F-STATISTICS FOR THE  19 COMPONENT
SUM OF SUMS = 0.8206200282896035D 02
SUM OF SUMS**2 = 0.16425037691583 88D 03
SUM OF SQUARE OF SUMS = 0.57887468750660 06D 03
AMONG GROUP SUM OF SQUARES = 0.1299298669532475D 01
WITHIN GROUP SUM OF SQUARES = 0.15846163004077 28D 03
TOTAL SUM OF SQUARES = 0.1597609287103052D 03
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.8697276809711277D 00

F-STATISTICS FOR THE  20 COMPONENT
SUM OF SUMS = 0.9520048302882186D 02
SUM OF SUMS**2 = 0.2094086558414699D 03
SUM OF SQUARE OF SUMS = 0.8713219218815827D 03
AMONG GROUP SUM OF SQUARES = 0.2671131239535162D 01
WITHIN GROUP SUM OF SQUARES = 0.2006954366226541D 03
TOTAL SUM OF SQUARES = 0.2033665678621893D 03
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.1411744637781583D 01
F-STATISTICS FOR THE  21 COMPONENT

SUM OF SUMS = 0.8994139952150146D 02
SUM OF SUMS**2 = 0.2335862791765991D 03
SUM OF SQUARE OF SUMS = 0.7245755635510632D 03
AMONG GROUP SUM OF SQUARES = 0.1852785403586404D 01
WITHIN GROUP SUM OF SQUARES = 0.2263405235410884D 03
TOTAL SUM OF SQUARES = 0.2281933089446748D 03
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.8682828488687504D 00
```

F-STATISTICS FOR THE 22 COMPONENT
SUM OF SUMS = 0.1035648300735959D 03
SUM OF SUMS**2 = 0.2414810868436007D 03
SUM OF SQUARE OF SUMS = 0.9427341493765320D 03
AMCNG GROUP SUM OF SQUARES = 0.2276892141650119D 01
WITHIN GROUP SUM OF SQUARES = 0.23205374534983540 03
TOTAL SUM OF SQUARES = 0.2343306374914855D 03
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.1040764077320930D 01

F-STATISTICS FOR THE 23 COMPONENT
SUM OF SUMS = 0.9031234717154746D 02
SUM OF SUMS**2 = 0.2047768116906234D 03
SUM OF SQUARE OF SUMS = 0.7149379755971323D 03
AMCNG GROUP SUM OF SQUARES = 0.1711833054881912D 01
WITHIN GROUP SUM OF SQUARES = 0.1976274319346521D 03
TOTAL SUM OF SQUARES = 0.1993392649895340D 03
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.9187822552243536D 00

F-STATISTICS FOR THE 24 COMPONENT
SUM OF SUMS = 0.9171706807335522D 02
SUM OF SUMS**2 = 0.1900438745495707D 03
SUM OF SQUARE OF SUMS = 0.8139562480340389D 03
AMCNG GROUP SUM OF SQUARES = 0.2531548763025405D 01
WITHIN GROUP SUM OF SQUARES = 0.1819043120692303D 03
TOTAL SUM OF SQUARES = 0.1844358608322557D 03
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.1476188171339999D 01

F-STATISTICS FOR THE 25 COMPONENT
SUM OF SUMS = 0.84401266234443660 02
SUM OF SUMS**2 = 0.17822548954371470 03
SUM OF SQUARE OF SUMS = 0.68929010739458400 03
AMONG GROUP SUM OF SQUARES = 0.21438519126275460 01
WITHIN GROUP SUM OF SQUARES = 0.17133258846976890 03
TOTAL SUM OF SQUARES = 0.17347644038239640 03
DEGREES OF FREEDOM IN NUMERATOR = 0.14000000000000000 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.14850000000000000 04
F-STATISTIC = 0.13272514998401340 01

F-STATISTICS FOR THE 26 COMPONENT
SUM OF SUMS = 0.69235794724921470 02
SUM OF SUMS**2 = 0.14819850583661250 03
SUM OF SQUARE OF SUMS = 0.44529803341859130 03
AMONG GROUP SUM OF SQUARES = 0.12572501533916050 01
WITHIN GROUP SUM OF SQUARES = 0.14374552550242660 03
TOTAL SUM OF SQUARES = 0.14500277565581820 03
DEGREES OF FREEDOM IN NUMERATOR = 0.14000000000000000 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.14850000000000000 04
F-STATISTIC = 0.92773892874769100 00

F-STATISTICS FOR THE 27 COMPONENT
SUM OF SUMS = 0.63660928221297440 02
SUM OF SUMS**2 = 0.11779068199127300 03
SUM OF SQUARE OF SUMS = 0.39351590415351810 03
AMONG GROUP SUM OF SQUARES = 0.12333498535370580 01
WITHIN GROUP SUM OF SQUARES = 0.11385552294973780 03
TOTAL SUM OF SQUARES = 0.11508887280327490 03
DEGREES OF FREEDOM IN NUMERATOR = 0.14000000000000000 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.14850000000000000 04
F-STATISTIC = 0.11490279742581360 01

```
F-STATISTICS FOR THE  28 COMPONENT
SUM OF SUMS = 0.5387592494868525D 02
SUM OF SUMS**2 = 0.1186133442790129D 03
SUM OF SQUARE OF SUMS = 0.2792595957915495D 03
AMCNG GROUP SUM OF SQUARES = 0.8575190985312517D 00
WITHIN GROUP SUM OF SQUARES = 0.1158207483210974D 03
TOTAL SUM OF SQUARES = 0.1166782674196286D 03
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.7853366268738306D 00

F-STATISTICS FOR THE  29 COMPONENT
SUM OF SUMS = 0.45361441519067000 02
SUM OF SUMS**2 = 0.6564305139058856D 02
SUM OF SQUARE OF SUMS = 0.1919053830677748D 03
AMONG GROUP SUM OF SQUARES = 0.5472802462192574D 00
WITHIN GROUP SUM OF SQUARES = 0.6372399755991081D 02
TOTAL SUM OF SQUARES = 0.6427127780613007D 02
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.9 109723144851782D 00

F-STATISTICS FOR THE  30 COMPONENT
SUM OF SUMS = 0.3732568029970994D 02
SUM OF SUMS**2 = 0.5705765228634419D 02
SUM OF SQUARE OF SUMS = 0.1284820507967475D 03
AMCNG GROUP SUM OF SQUARES = 0.3560162347433721D 00
WITHIN GROUP SUM OF SQUARES = 0.5577283177837672D 02
TOTAL SUM OF SQUARES = 0.5612884801312009D 02
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.6770886363437515D 00
```

F-STATISTICS FOR THE 31 COMPONENT
SUM OF SUMS = 0.2973876286269883D 02
SUM OF SUMS**2 = 0.4217833610659068D 02
SUM OF SQUARE OF SUMS = 0.7726205204918280D 02
AMONG GROUP SUM OF SQUARES = 0.1830245094226048D 00
WITHIN GROUP SUM OF SQUARES = 0.4140571558609885D 02
TOTAL SUM OF SQUARES = 0.4158874009552145D 02
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.4688645251806333D 00

F-STATISTICS FOR THE 32 COMPONENT
SUM OF SUMS = 0.2142367689889270D 02
SUM OF SUMS**2 = 0.1414820425416038D 02
SUM OF SQUARE OF SUMS = 0.3565723402182890D 02
AMONG GROUP SUM OF SQUARES = 0.5058971897285643D-01
WITHIN GROUP SUM OF SQUARES = 0.1379163191394209D 02
TOTAL SUM OF SQUARES = 0.1384222163291495D 02
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.3890854828465455D 00

F-STATISTICS FOR THE 33 COMPONENT
SUM OF SUMS = 0.1732004830070321D 02
SUM OF SUMS**2 = 0.1115582097066481D 02
SUM OF SQUARE OF SUMS = 0.2312899538600082D 02
AMONG GROUP SUM OF SQUARES = 0.3130057176754676D-01
WITHIN GROUP SUM OF SQUARES = 0.1092453101680480D 02
TOTAL SUM OF SQUARES = 0.1095583158857235D 02
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04
F-STATISTIC = 0.3039120267386334D 00

F-STATISTICS OF MEAN VARIANCE, SKEWNESS AND KURTOSIS

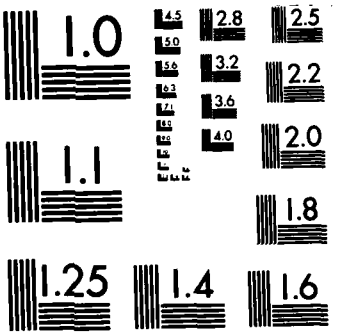
F-STATISTICS FOR THE 1 COMPONENT
SUM OF SUMS = 0.1111617439422250D 05
SUM OF SUMS**2 = 0.43194257777749392D 07
SUM OF SQUARE OF SUMS = 0.1861260733960752D 08
AMONG GROUP SUM OF SQUARES = 0.3143833877400904D 06
WITHIN GROUP SUM OF SQUARES = 0.3755407373518861D 07
TOTAL SUM OF SQUARES = 0.4069790761258952D 07
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.4800000000000000D 03
F-STATISTIC = 0.2870223636519026D 01

F-STATISTICS FOR THE 2 COMPONENT
SUM OF SUMS =-0.3062348822555926D 07
SUM OF SUMS**2 = 0.7751235739069674D 12
SUM OF SQUARE OF SUMS = 0.2376364181020636D 13
AMONG GROUP SUM OF SQUARES = 0.5306562101878761D 11
WITHIN GROUP SUM OF SQUARES = 0.7031125381184634D 12
TOTAL SUM OF SQUARES = 0.7561781591372509D 12
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.4800000000000000D 03
F-STATISTIC = 0.2587626620217669D 01

F-STATISTICS FOR THE 3 COMPONENT
SUM OF SUMS = 0.1087692759375071D 08
SUM OF SUMS**2 = 0.1368235571001090D 14
SUM OF SQUARE OF SUMS = 0.3521258942662316D 14
AMONG GROUP SUM OF SQUARES = 0.8280430048704694D 12
WITHIN GROUP SUM OF SQUARES = 0.1261530754556778D 14
TOTAL SUM OF SQUARES = 0.1344335055043825D 14
DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02
DEGREES OF FREEDOM IN DENOMINATOR = 0.4800000000000000D 03
F-STATISTIC = 0.2250444214596077D 01

AD-A160 823 COMPUTER SIMULATION OF DIGITAL SIGNAL MODULATION TECHNIQUES IN SATELLITE COMMUNICATIONS(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA C D CARLSON SEP 85 4/4

UNCLASSIFIED F/G 9/2 NL



LIST OF REFERENCES

1. CLARKE, A.C., "Extra-Terrestrial Relays", Wireless World, vol. 51, no. 10, p. 305, Oct. 1945.

2. Pierce, J.R., "Orbital Radio Relays", Jet Propulsion, vol. 25, p. 153, Apr. 1955.

3. "Communication Satellites", Research Report, Encyclopedia Britannica, Inc., Chicago, Ill., 1978.

4. Jaffe, L., "NASA Communications Satellite Developments", Astronautics and Aerospace Engineering, vol. 1, no. 8, p. 48, Sep. 1963.

5. Edelson, B.I., Strauss, R. and Bargellini, P.L., "INTELSAT System Reliability", Acta Astronautica, vol. 2, p. 691, 1975.

6. Gagliardi, R.M., Satellite Communications, Lifetime Learning Publications, Belmont, Ca., 1984.

7. Bhargava, V.K., et al, Digital Communications by Satellite, John Wiley and Sons, Inc., 1981.

8. Feher, K., Digital Communications-Satellite/Earth Station Engineering, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1984.

9. Nyquist, "Certain Topics in Telegraph Transmission Theory", Transactions of the AIEE, Apr., 1928.

10. Shannon, et al, "The Philosophy of PCM", Proceedings of the IERE, Nov. 1948.

11. "Digital Signal Modulation Techniques in Satellite Communications", Research Report, Encyclopedia Britannica, Inc., Chicago, Ill., 1984.

12. Lender, A., "The Duobinary Technique for High Speed Data Transmission", IEEE Transactions on Communications and Electronics, vol. 82, p. 214, May, 1963.

13. Kabal, P. and Pasupathy, S., "Partial Response Signalling", IEEE Transactions on Communications, vol. COM-23, no. 9, Sep. 1975.

14. Oshita, S. and Feher, K., "Combined Effect of the Carrier Recovery and Symbol Timing Error on the Pe Performance of QPR and Offset QPR Systems", IEEE Transactions on Communications, vol. COM-30, no. 12, Dec. 1982.

15. Harmuth, H.F., Sequency Theory, Foundations and Applications, vol. 9, Advances in Electronics and Electron Physics, Academic Press, 1977.

16. Hahn, G.J. and Shapiro, S.S., Statistical Models in Engineering, John Wiley and Sons, Inc., 1967.

17. Hickman, E.P. and Hilton, J.G., Probability and Statistical Analysis, Intex Educational Publishers, 1971.

BIBLIOGRAPHY

Ahmed, N. and Rao, K.R., Orthogonal Transforms for Digital Signal Processing, Springer-Verlag, New York, 1975.

Boutin, N. and Morisette, S., "Discrete Finite Duration Partial Response Signaling Pulse with Minimum Out of Band Power", IEEE Globecom, vol. 2, 1983.

Brigham, E.O., The Fast Fourier Transform, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1977.

Feher, K., Digital Communications Microwave Applications, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1981.

Herman, R., et al, "A Modular Program for Simulation of Digital Systems and Its Application to Satellite System Optimization", Proceedings on the Joint Conference on Digital Processing of Signals in Communications, IERE, Apr. 1972.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Superintendent Attn: Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5100	2
3.	Commander Naval Space Command Dahlgren, Virginia 22448	1
4.	Professor Allen E. Fuhs, Code 72 Space Systems Academic Group Naval Postgraduate School Monterey, California 93943-5100	1
5.	Professor James L. Wayman, Code 53Ww Department of Mathematics Naval Postgraduate School Monterey, California 93943-5100	7
6.	Space Systems Academic Group, Code 75 Naval Postgraduate School Monterey, California 93943-5100	1
7.	LTCOL John T. Malokos, Code 39 Naval Postgraduate School Monterey, California 93943-5100	1
8.	ICDR Craig D. Carlson Attack Squadron 42 Oceana, Virginia 23460	6

END

FILMED

11-85

DTIC