

AD-A160 823

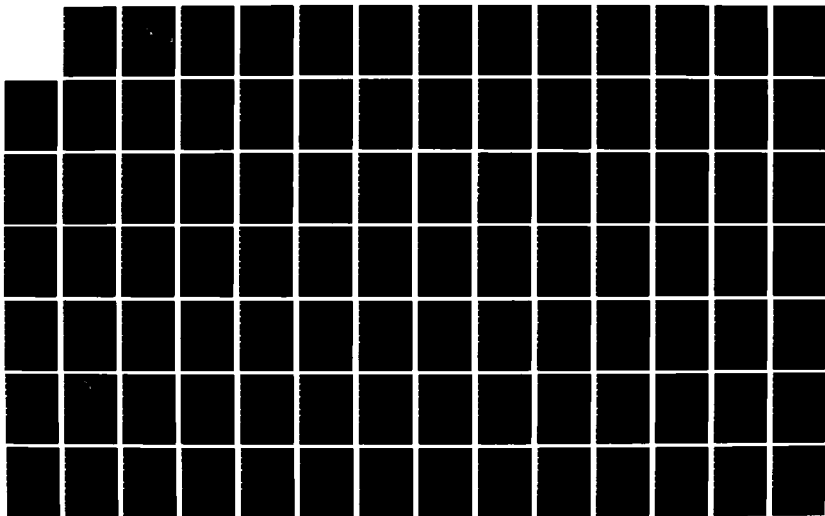
COMPUTER SIMULATION OF DIGITAL SIGNAL MODULATION
TECHNIQUES IN SATELLITE COMMUNICATIONS(U) NAVAL
POSTGRADUATE SCHOOL MONTEREY CA C D CARLSON SEP 85

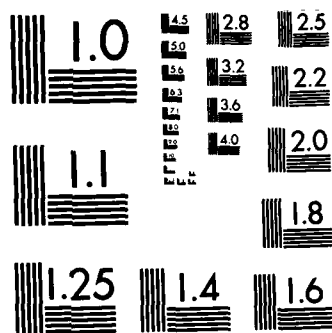
1/4

UNCLASSIFIED

F/G 9/2

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

2

NAVAL POSTGRADUATE SCHOOL

Monterey, California

AD-A160 823



DTIC
ELECTE
NOV 04 1985
S E D

THESIS

COMPUTER SIMULATION OF
DIGITAL SIGNAL MODULATION TECHNIQUES IN
SATELLITE COMMUNICATIONS

by

Craig Dean Carlson

September 1985

Thesis Advisor:

James L. Wayman

DTIC FILE COPY

Approved for public release; distribution is unlimited

85 11 04 00 8

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-7160 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
7. AUTHOR(s) Craig D. Carlson		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943-5100		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943-5100		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE September 1985
		13. NUMBER OF PAGES 291
		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		

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

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

3



Accession For	
NTIS GRA&I	☒
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

I.	INTRODUCTION	11
	A. BACKGROUND	11
	B. SPECIFIC GOALS	12
	C. SCOPE OF THE PROJECT	12
II.	UNDERSTANDING SATELLITE COMMUNICATIONS	14
	A. HISTORY AND APPLICATIONS	14
	B. ORBITS AND LIMITATIONS	17
	C. FREQUENCY BAND CONSIDERATIONS	21
	D. WHY DIGITAL MODULATION	24
	1. Compatibility with Digital Computers	24
	2. Flexibility and Economy	25
	3. Quality and Interference	25
III.	INTRODUCTION TO THE FOURIER TRANSFORM, SAMPLING THEOREM, AUTOCORRELATION FUNCTION AND THE MATCHED FILTER	26
	A. FOURIER TRANSFORM	26
	1. Amplitude and Phase Spectrum	30
	2. Properties of the Fourier Transform	31
	B. THE SAMPLING THEOREM	34
	C. THE AUTOCORRELATION FUNCTION	39
	D. THE MATCHED FILTER	45
IV.	ANALOG TO DIGITAL CONVERSION	48
	A. PULSE CODE MODULATION (PCM)	48
	B. DIFFERENTIAL PULSE CODE MODULATION (DPCM)	51
	C. DELTA MODULATION (DM)	52
V.	DIGITAL SIGNAL MODULATION TECHNIQUES	54

A.	DIGITAL MODULATION FORMATS	54
B.	HARDWARE	55
1.	Sampler	56
2.	Encoder	56
3.	Modulator	60
4.	Multiplexer	60
C.	BANDWIDTH	61
D.	SPECIFIC TECHNIQUES	63
1.	Phase Shift Keying (PSK)	63
2.	Amplitude Shift Keying (ASK)	74
3.	Frequency Shift Keying (FSK)	76
4.	Quadrature Partial Response Signalling (QPRS)	78

VI.	COMPUTER SIMULATION OF DIGITAL SIGNAL MODULATION TECHNIQUES	82
A.	GENERAL DESCRIPTION OF THE PROGRAM	82
B.	MAIN CONTROL MODULE	83
C.	SUBROUTINE BPSK	84
D.	SUBROUTINE DBPSK	86
E.	SUBROUTINE OBPSK	87
F.	SUBROUTINE QPSK	88
G.	SUBROUTINE OQPSK	88
H.	SUBROUTINE MPSK	89
I.	SUBROUTINE MASK	90
J.	SUBROUTINE QASK	91
K.	SUBROUTINE MSK	91
L.	SUBROUTINE MFSK	92
M.	SUBROUTINE QPRS	93
N.	SUBROUTINE STREAM	94
O.	SUBROUTINE CRTHO	94
P.	SUBROUTINE PLOT	94
Q.	SUBROUTINE STATS	94

VII.	STATISTICS OF THE FAST FOURIER TRANSFORMS	96
A.	DESCRIPTION OF THE PROBLEM	96
B.	ANALYSIS-OF-VARIANCE	96
1.	The F-Distribution	97
2.	Design of the Experiment	97
3.	Hypothesis and Hypothesis Testing	100
4.	Results	100
C.	CONCLUSION	100
APPENDIX A:	FORTRAN PROGRAM FCR DIGITAL SIGNAL MODULATION	104
APPENDIX B:	IMSL/NON-IMSL ROUTINES UTILIZED	180
APPENDIX C:	REPRESENTATIVE RESULTS OF TRIAL SIMULATIONS	198
APPENDIX D:	FORTRAN PROGRAM F-TEST	273
APPENDIX E:	RESULTS OF F-TEST	276
LIST OF REFERENCES		288
BIBLIOGRAPHY		290
INITIAL DISTRIBUTION LIST		291

LIST OF TABLES

1.	QPRS SYSTEM POLYNOMIALS	80
2.	ANALYSIS-OF-VARIANCE TABLE	99
3.	F-STATISTICS OF FFT COMPONENTS	102
4.	F-STATISTICS OF THE MEAN VARIANCES, MEAN SKEWNESS AND MEAN KURTOSIS	103

LIST OF FIGURES

2.1	Satellite Orbits	18
2.2	Eccentricity	19
2.3	Kepler's Second Law of Planetary Motion	20
2.4	Electromagnetic Spectrum	22
2.5	Satellite Frequencies	23
3.1	Square Pulse	28
3.2	Plot of Sinc Function	30
3.3	Phasor Diagram	31
3.4	Amplitude Spectrum of Square Wave	32
3.5	Phase Spectrum of Square Wave	32
3.6	Square Wave in Time/Sinc Function in Frequency	33
3.7	Multiplication and Translation Diagrams	34
3.8	Representative Analog Voltage	35
3.9	Samples of a Representative Voltage	36
3.10	Analog Voltage Multiplier	36
3.11	A Representative Voltage Clock	37
3.12	Fourier Transform of $v(t)$	38
3.13	Fourier Transform of $v_s(t)$	38
3.14	Analog Voltage Multiplier and Low Pass Filter	39
3.15	Rectification through an AVM and LPF	42
3.16	Correlation Device	43
3.17	Voltage Correlator	44
3.18	Matched Filter Block Representation	46
4.1	Samples of Voltage	49
4.2	Analog Nature of Samples	50
5.1	Basic Hardware Component Block Diagram	56
5.2	Bipolar Logic	57
5.3	Unipolar Positive Logic	58

5.4	Unipolar Negative Logic	59
5.5	Non Return to Zero Waveform	59
5.6	Manchester Waveform	60
5.7	Fourier Transform of Baseband Waveform	62
5.8	Phasor Diagram of BPSK	65
5.9	Differential Binary Phase Shift Keying	67
5.10	Orthogonal Binary Phase Shift Keying	68
5.11	Probability of Error for Orthogonal BPSK	69
5.12	Modulation and Phases of QPSK	71
5.13	Phasor Diagram of QPSK	72
5.14	Offset Quadrature Phase Shift Keying	72
5.15	Phasor Diagram of OQPSK	73
5.16	Phasor Diagram of FFSK or MSK	79
5.17	QPRS Linear Filter Impulse Responses	80
6.1	Representative Block Diagram	84

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 (V_a) and perigee (V_p) given by equations 2.2 and 2.3.

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

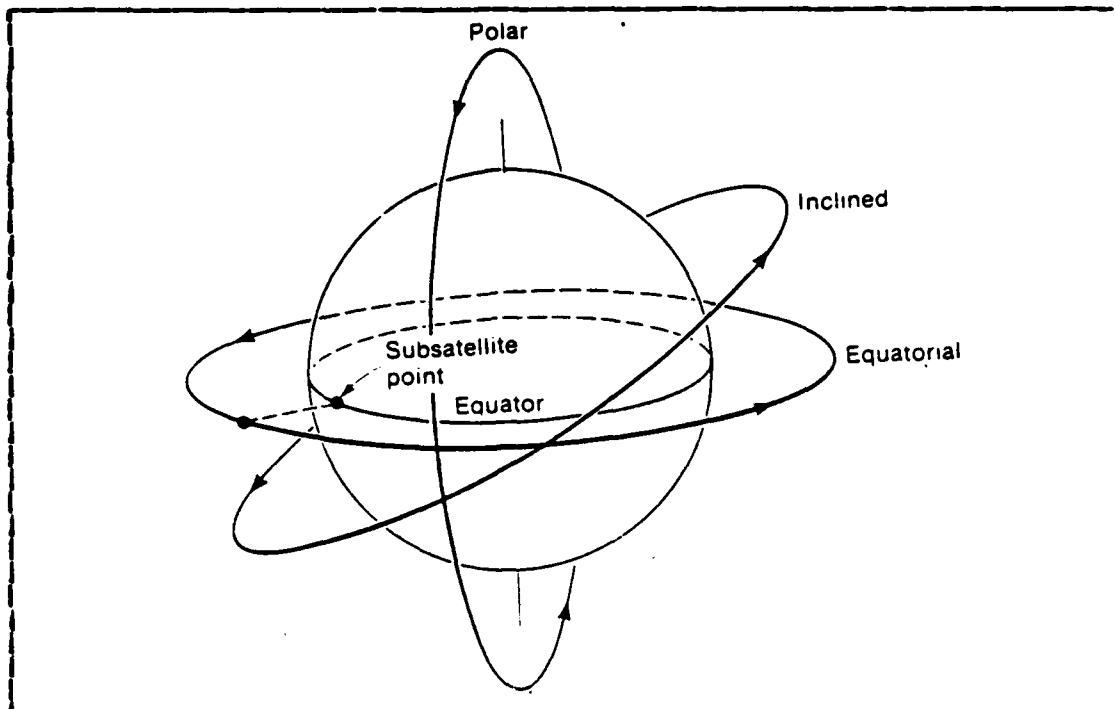


Figure 2.1 Satellite Orbits

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.

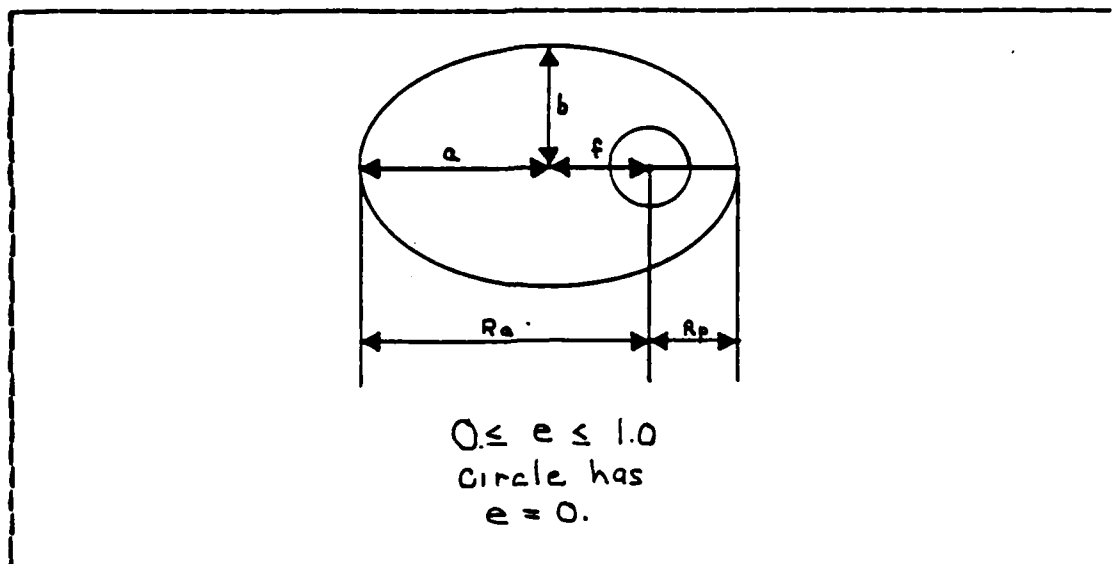


Figure 2.2 Eccentricity

$$\text{eccentricity } (e) = c/a$$

(eqn 2.1)

c = distance from focus

a = semi-major axis

b = semi-minor axis

R_a = radius of apogee

R_p = radius of perigee

By rearranging terms that altitude is given in equation 2.6. Since the orbit is circular, $a = R_a = R_p$ and the height of the orbit above the surface of the earth is $h = a - R_e$; 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

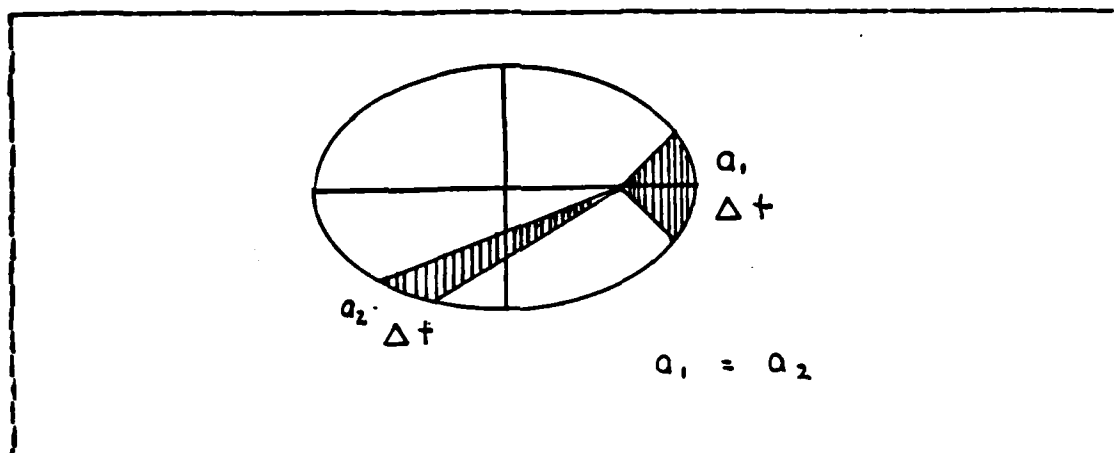


Figure 2.3 Kepler's Second Law of Planetary Motion

$$V_a = (k/R_a(1-e))^{.5} \quad (\text{eqn 2.2})$$

$$V_p = (k/R_p(1+e))^{.5} \quad (\text{eqn 2.3})$$

$$k = G M_e = 3.98866 \times 10^{-11} \text{ m}^3/\text{s}^2$$

G = universal gravitational constant

$$G = 6.67 \times 10^{-11} \text{ N m}^2/\text{kg}^2$$

N = newtons = $\text{m kg}/\text{s}^2$

$$M_e = \text{mass of the earth} = 5.98 \times 10^{24} \text{ kg}$$

$$V_c = (k/R_c)^{.5} \quad (\text{eqn 2.4})$$

$$R_c = R_e + h$$

$$R_e = \text{radius of the earth} = 6.37 \times 10^6 \text{ m}$$

h = satellite orbital altitude

$$T = 2 \pi a^{1.5} / k^{.5} \quad (\text{eqn 2.5})$$

T = period of rotation

a = semi-major axis

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

(eqn 2.6)

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

laterally to a latitude of about $\pm 80^\circ$ [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].

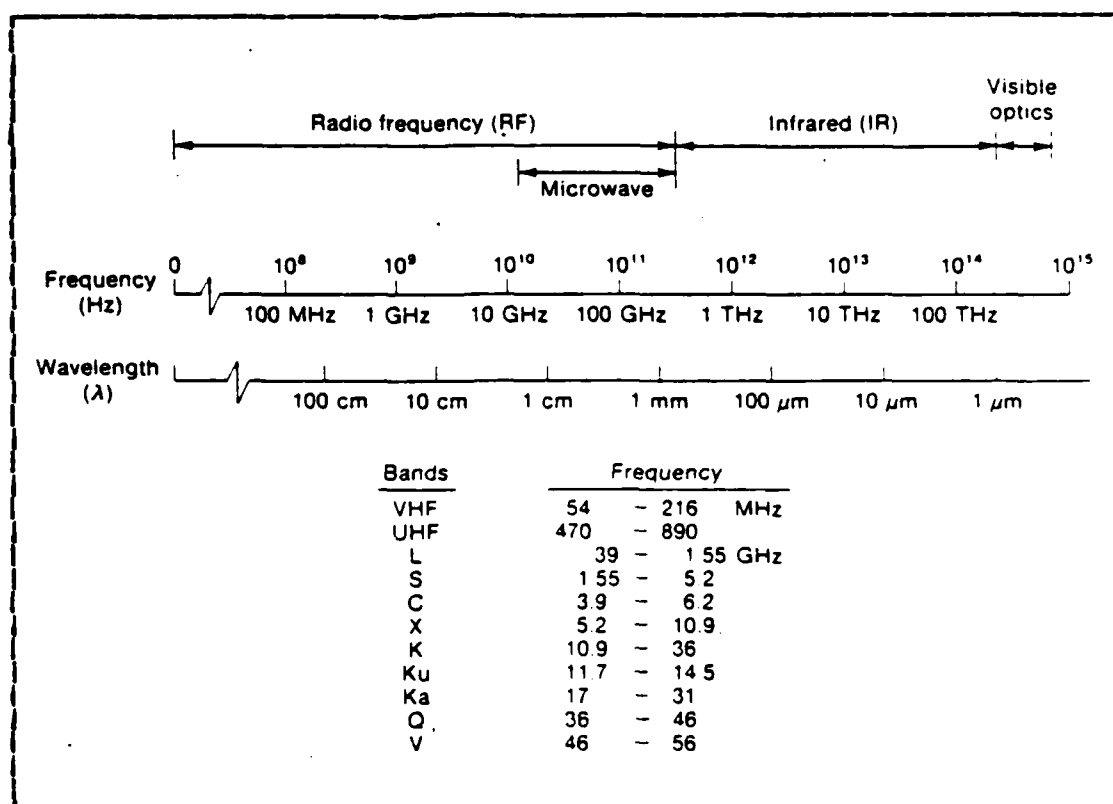


Figure 2.4 Electromagnetic Spectrum

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

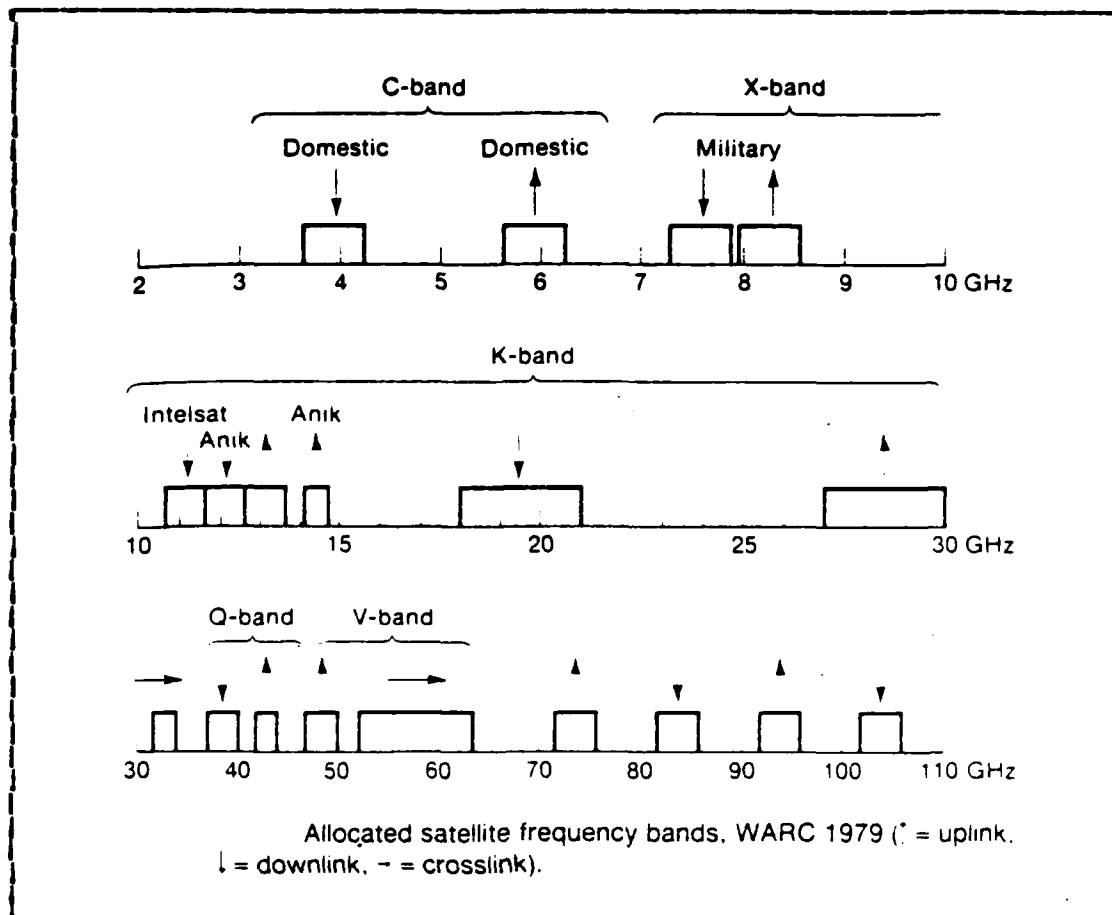


Figure 2.5 Satellite Frequencies

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.

$$v(t) = A \cos(\omega t) \quad (\text{eqn 3.1})$$

A = amplitude

ω = 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:

$$\mathcal{F}[v(t)] = V(\omega) = \int_{-\infty}^{\infty} v(t) e^{-j\omega t} dt \quad (\text{eqn 3.2})$$

$v(t)$ <-----> $V(\omega)$
 time domain frequency domain

Remembering Euler's formula

$$e^{-j\omega t} = \cos(\omega t) - j \sin(\omega t) \quad (\text{eqn 3.3})$$

the Fourier transform becomes

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

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

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 .

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

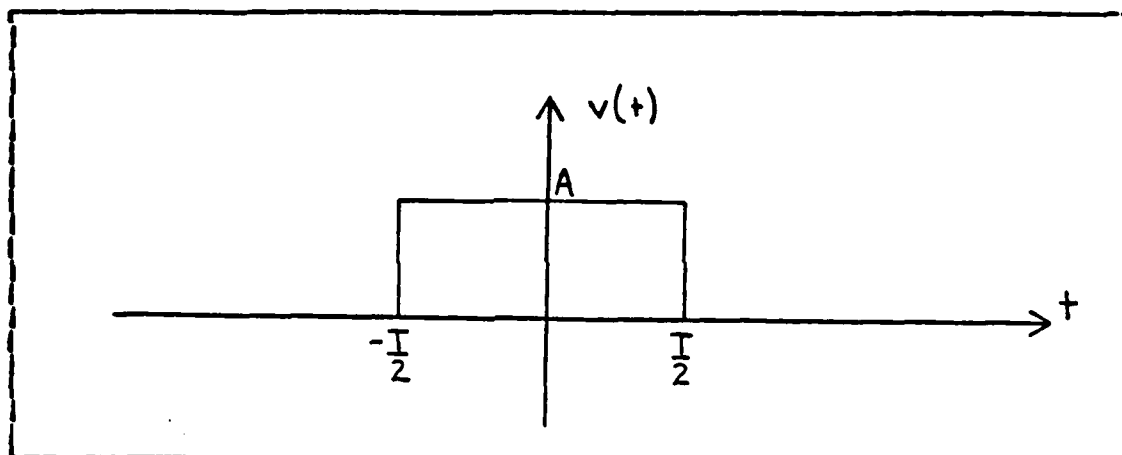


Figure 3.1 Square Pulse

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.

$$\mathcal{F}[v(t)] = V(\omega) = \int_{-\infty}^{\infty} v(t) e^{** - j\omega t} dt \quad (\text{eqn 3.6})$$

$$V(\omega) = \int_{-\infty}^{-T/2} 0 \cdot e^{** - j\omega t} dt + \int_{-T/2}^{T/2} A e^{** - j\omega t} dt + \int_{T/2}^{\infty} 0 \cdot e^{** - j\omega t} dt \quad (\text{eqn 3.7})$$

$$V(\omega) = \int_{-T/2}^{T/2} A e^{** - j\omega t} dt \quad (\text{eqn 3.8})$$

The integration above may be attacked either head on or by substituting $\cos(\omega t) - j \sin(\omega t)$ for $e^{** - j\omega t}$. The direct approach is illustrated due to the relative simplicity of the integrand.

$$V(\omega) = \int_{-T/2}^{T/2} A e^{** - j\omega t} dt \quad (\text{eqn 3.9})$$

$$V(\omega) = -A/j\omega [e^{** - j\omega t}] \Big|_{-T/2}^{T/2} \quad (\text{eqn 3.10})$$

$$V(\omega) = -A/j\omega[e^{**j\omega T/2} - e^{**j\omega T/2}] \quad (\text{eqn 3.11})$$

$$V(\omega) = A/j\omega[e^{**j\omega T/2} - e^{**j\omega T/2}] \quad (\text{eqn 3.12})$$

By substituting $2\pi f = \omega$, 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.

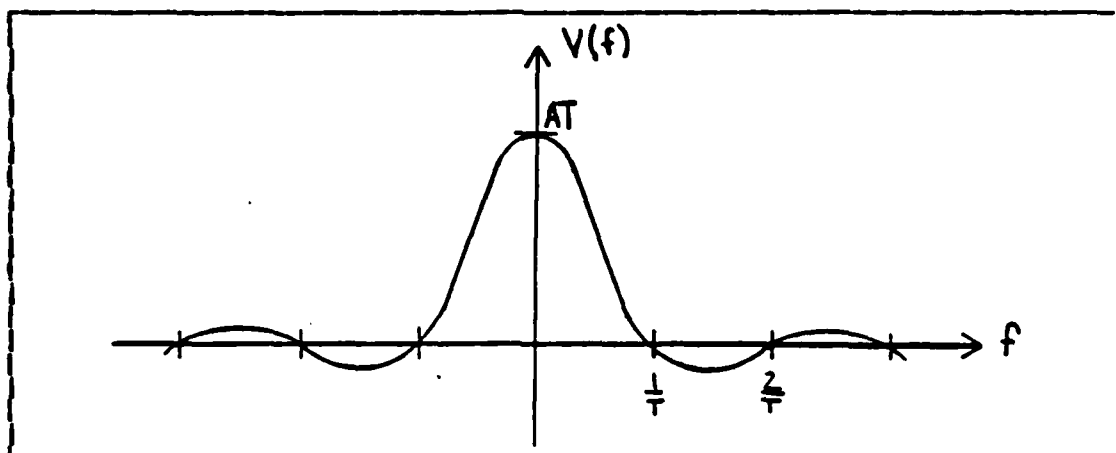


Figure 3.2 Plot of Sinc Function

1. Amplitude and Phase Spectrum

Recall that $V(\omega)$ 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.

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

$|V(f)|$ 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 $\arctan[\text{imaginary}/\text{real}]$ and is illustrated in Figure 3.5.

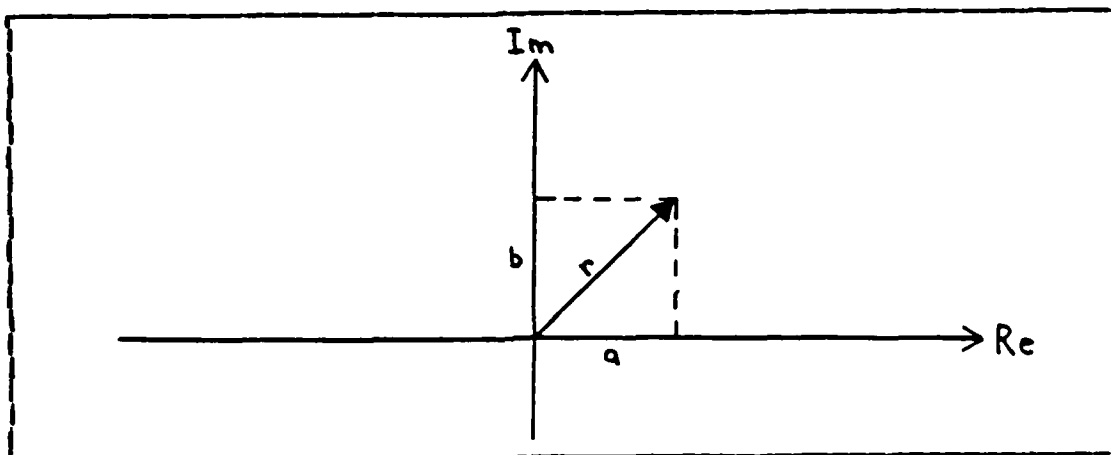


Figure 3.3 Phasor Diagram

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

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

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

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

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

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

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 $v_1(t) \leftrightarrow V_1(f)$ and

if $v_2(t) \leftrightarrow V_2(f)$ then

$$\mathcal{F}[av_1(t) + bv_2(t)] = aV_1(f) + bV_2(f)$$

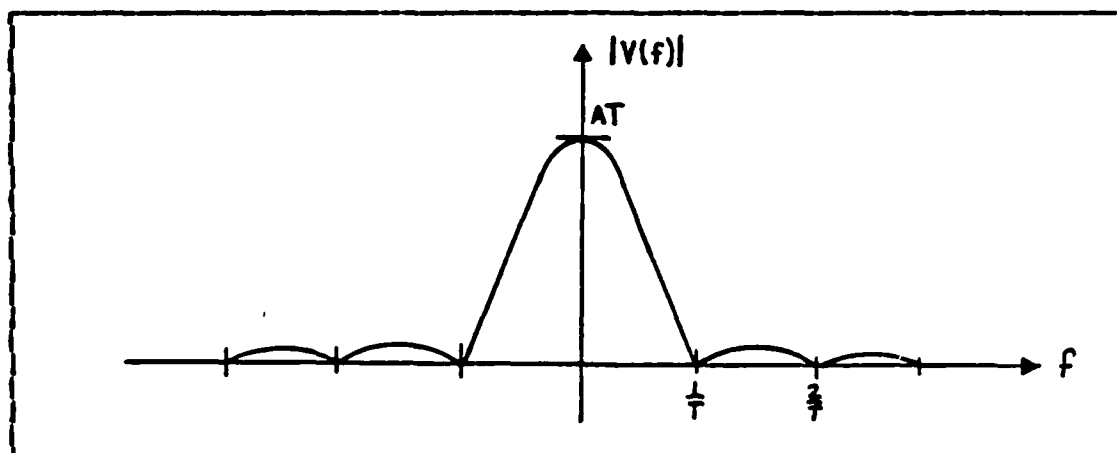


Figure 3.4 Amplitude Spectrum of Square Wave

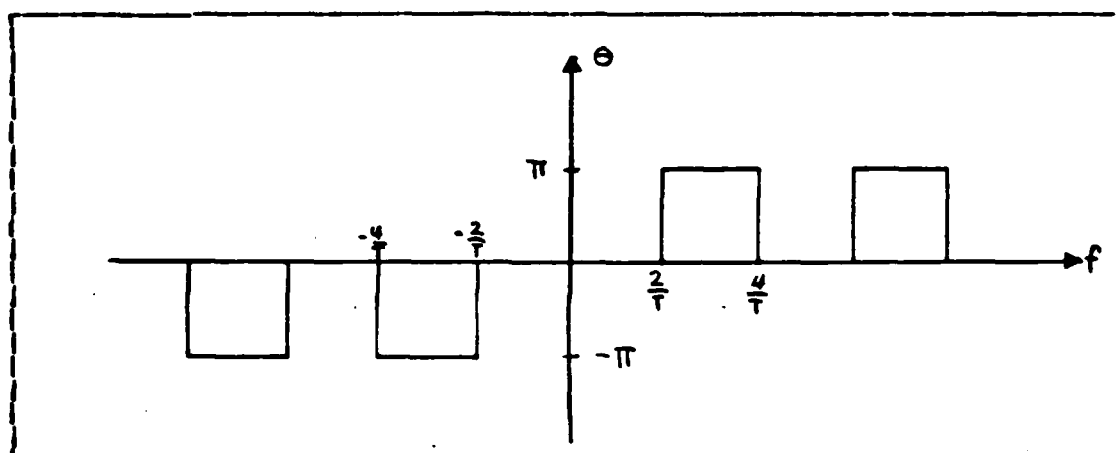


Figure 3.5 Phase Spectrum of Square Wave

b. Time-delay

$$\mathcal{F}[v(t-t_0)] = V(f) e^{-j\omega t_0}$$

Note that the amplitude spectrum of the delayed version 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

$$v(t)\cos(2\pi f_c t) \leftrightarrow \frac{1}{2}[V(f+f_c) + V(f-f_c)]$$

$v(t)$ is any voltage

$\cos(2\pi f_c t)$ is a carrier wave of frequency f_c

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.

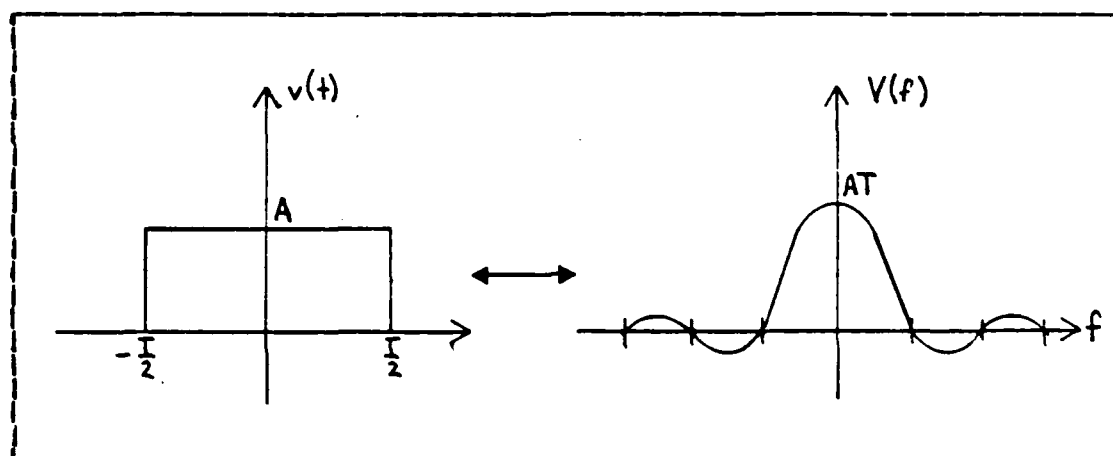


Figure 3.6 Square Wave in Time/Sinc Function in Frequency

The translation is the product of $v(t)$ and in this case $\cos(2\pi f_c t)$. 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

$$d v(t)/dt \leftrightarrow j\omega V(f)$$

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

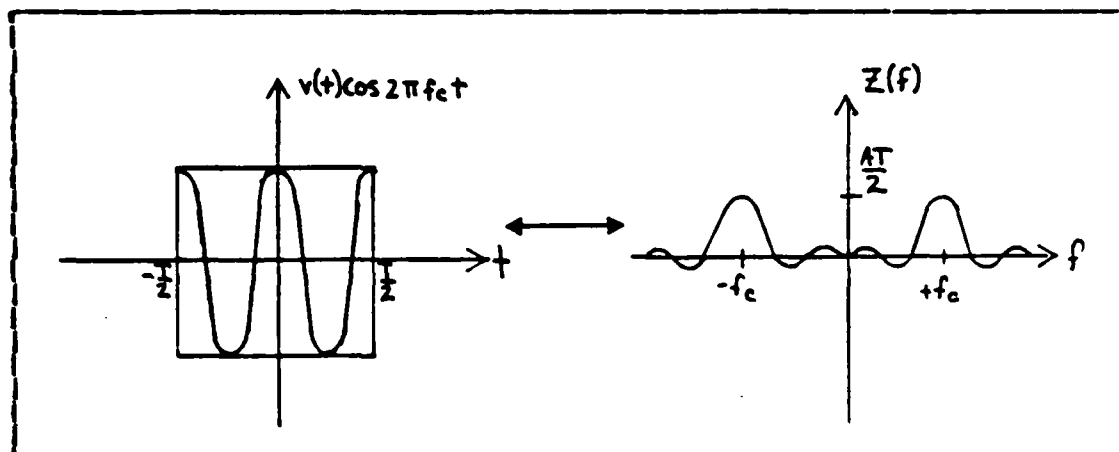


Figure 3.7 Multiplication and Translation Diagrams

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

f. Integration

$$v(t) dt \leftrightarrow 1/w V(f)$$

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 $v_s(t)$. 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 $v_c(t)$. 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, $v_s(t)$. Note that the system which describes the obtaining of $v_s(t)$ involves a voltage multiplication. It was demonstrated in the proceeding section that voltage multiplication amounted to frequency translation, a property of the Fourier transform.

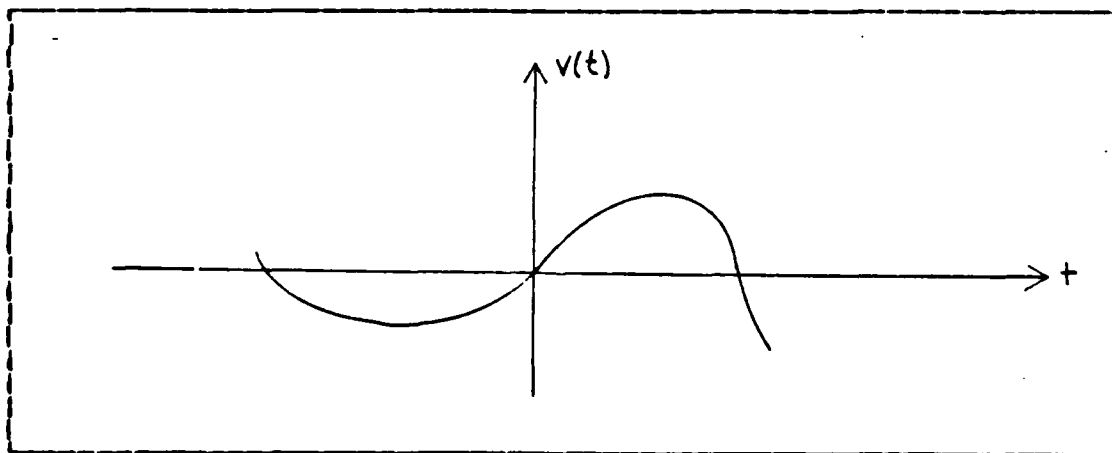


Figure 3.8 Representative Analog Voltage

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

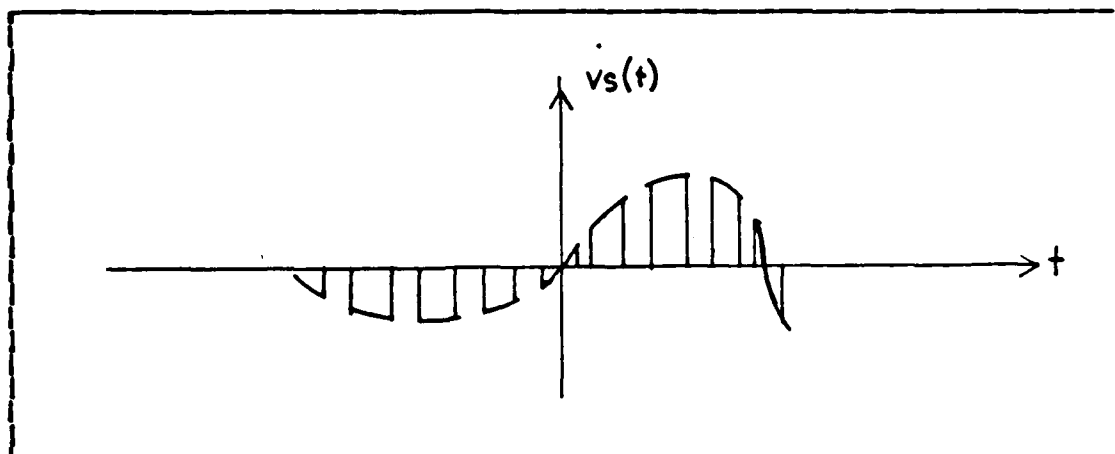


Figure 3.9 Samples of a Representative Voltage

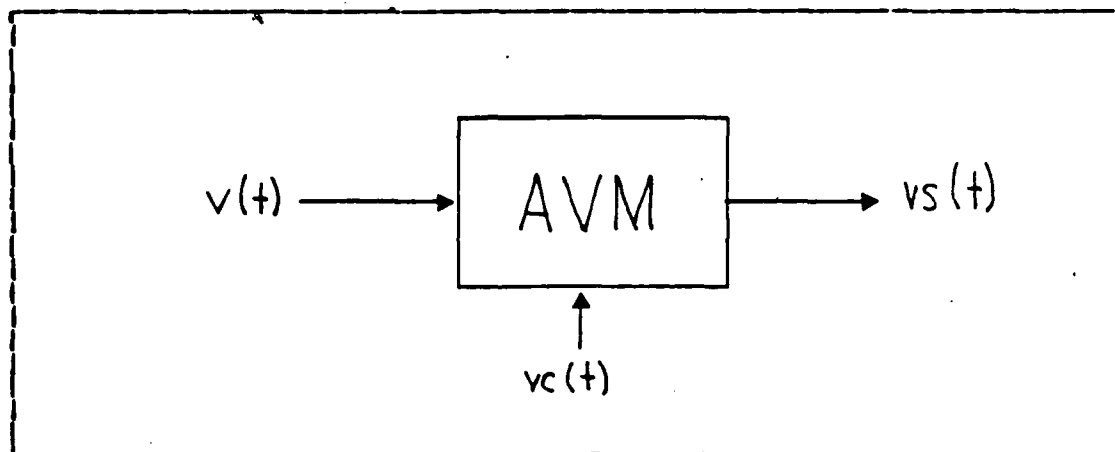


Figure 3.10 Analog Voltage Multiplier

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

$$v_c(t) = a_0 + \sum_{n=1}^{\infty} a_n \cos(2 \pi n f_c t) \quad (\text{eqn 3.22})$$

a_0 and a_n are left unevaluated

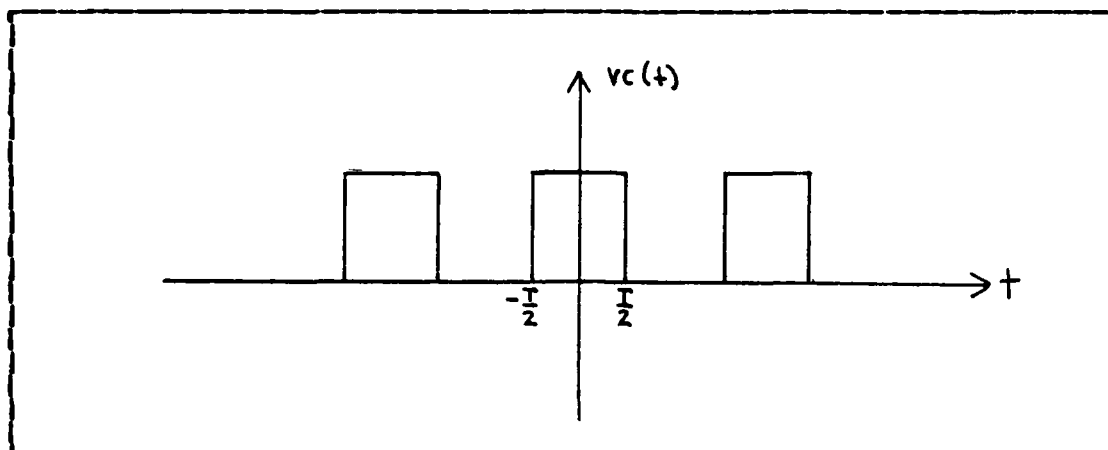


Figure 3.11 A Representative Voltage Clock

$$v_s(t) = v(t) \left[a_0 + \sum_{n=1}^{\infty} a_n \cos(2 \pi n f_c t) \right] \quad (\text{eqn 3.23})$$

Now, assuming any form of the Fourier transform of $v(t)$, as in Figure 3.12 and since $v_s(t)$ 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 $v_s(t)$. See Figure 3.13.

Remember that the curve highlighted in the box in Figure 3.13 is the Fourier representation 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)

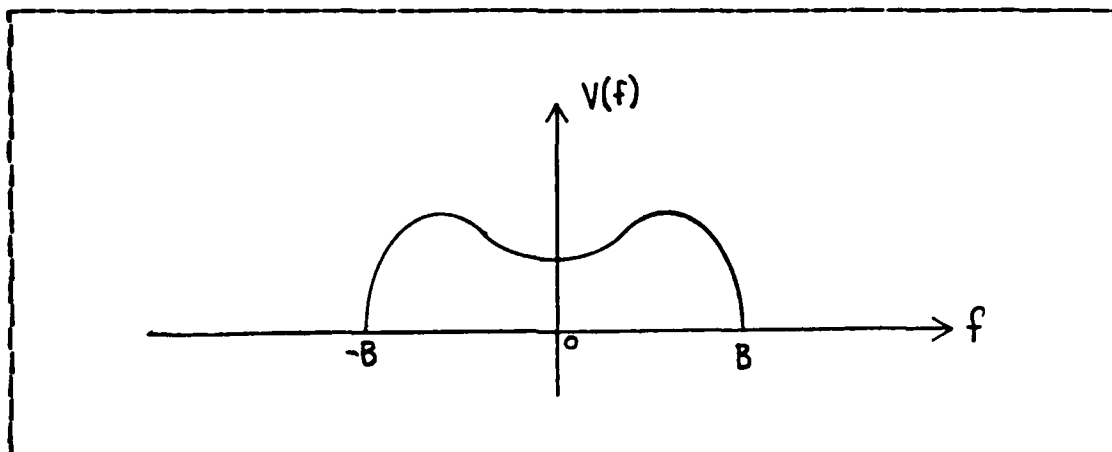


Figure 3.12 Fourier Transform of $v(t)$

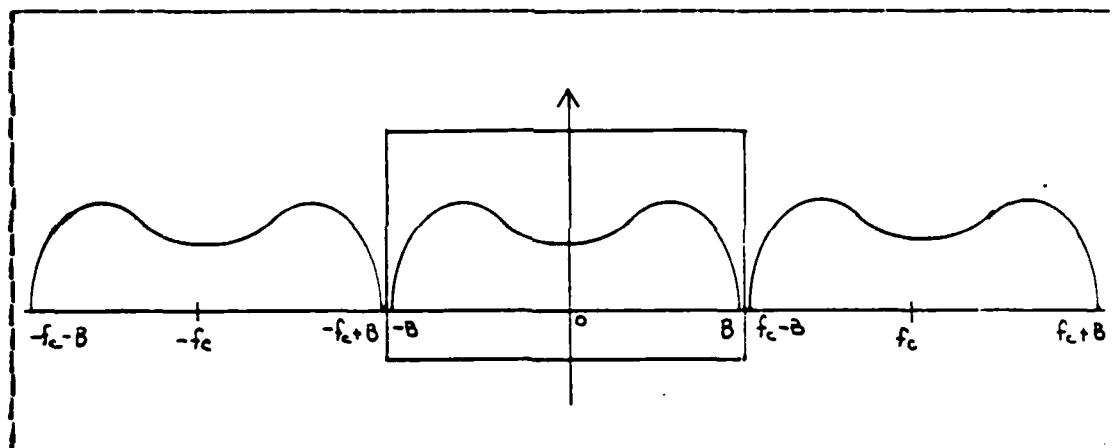


Figure 3.13 Fourier Transform of $v_s(t)$

$$f_0 - B \gg B$$

$$f_0 \gg 2B$$

or the frequency of the clock, f_c (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.

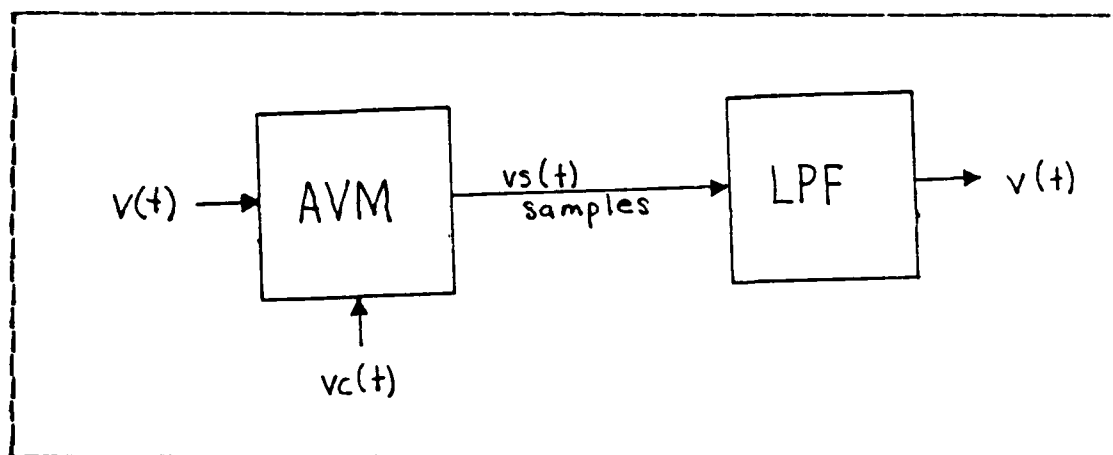


Figure 3.14 Analog Voltage Multiplier and Low Pass Filter

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, $v_{sr}(t)$, is the digitally converted sample voltage, $v_s(t)$, 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 $v_{sr}(t)$ 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, $v_r(t)$, is equal to the signal which is being transmitted, $v_{sr}(t)$, (i.e., the digital samples of $v(t)$ modulated onto the carrier), plus random noise, $v_n(t)$

$$v_r(t) = v_{sr}(t) + v_n(t)$$

(eqn 3.24)

$v_r(t)$ = voltage at the receiver

$v_{sr}(t)$ = signal voltage at the receiver

$v_n(t)$ = 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 $v_r(t)$ 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 $v_r(t)$ and introducing or at least not eliminating the noise, $v_r(t)$ is multiplied by $v_{sr}(t)$, the results in equation 3.25 are obtained.

$$v_r(t) \cdot v_{sr}(t) = v_{sr}^2(t) + v_{sr}(t) v_n(t)$$

(eqn 3.25)

By averaging, all that remains is $\overline{v_{sr}^2(t)}$ since the average of $v_{sr}(t) v_n(t)$ is zero because $\overline{v_n(t)}$ is random and the average of any random voltage is zero. The DC component of $v_r(t) \cdot v_{sr}(t)$ is $\overline{v_{sr}^2(t)}$. 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 $v_{sr}^2(t) > 0$ then $v_{sr}(t)$ is present in the signal $v_r(t)$. If $v_r(t)$ is pure noise or some other signal is present exclusively, then the output of the receiver will be 0.

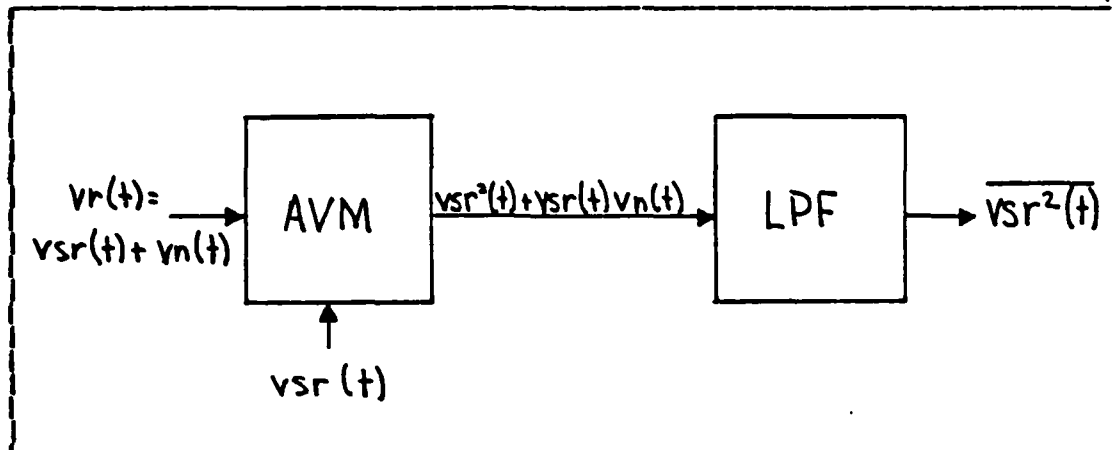


Figure 3.15 Rectification through an AVM and LPF

In practice, however, the exact waveform of $v_{sr}(t)$ 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 $v_{sr}^2(t)$. The concept is illustrated in Figure 3.16.

For the concept illustrated in Figure 3.16 to work, the average value of $v_r(t) \cdot v_{sr}(t-p)$ must have a DC component.

$$v_r(t) \cdot v_{sr}(t-p) = v_{sr}(t)v_{sr}(t-p) + v_n(t)v_{sr}(t-p) \quad (\text{eqn 3.26})$$

$$\overline{v_r(t) \cdot v_{sr}(t-p)} = \overline{v_{sr}(t)v_{sr}(t-p)} + \overline{v_n(t)v_{sr}(t-p)} \quad (\text{eqn 3.27})$$

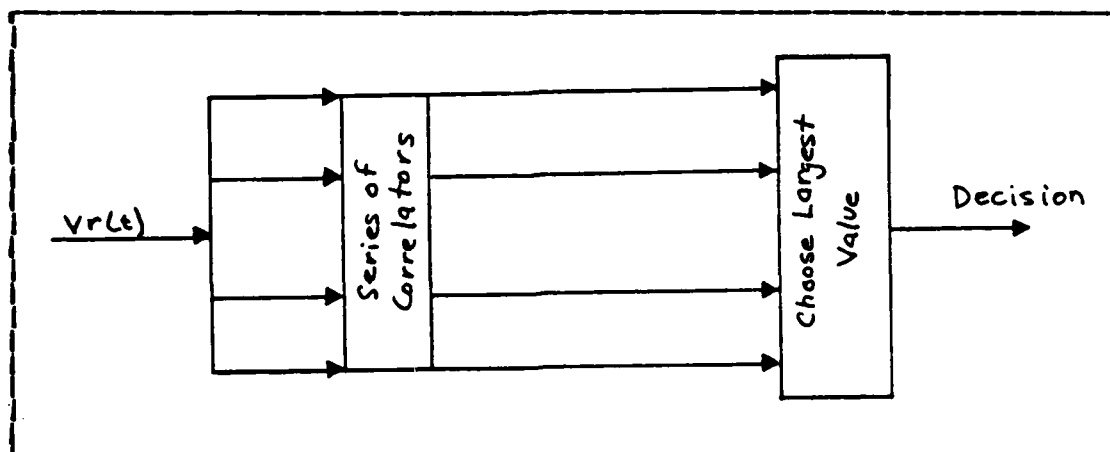


Figure 3.16 Correlation Device

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

$$\overline{v_r(t) \cdot v_{sr}(t-p)} = \overline{v_{sr}(t)v_{sr}(t-p)} \quad (\text{eqn 3.28})$$

What is of interest is the average value of $v_{sr}(t)v_{sr}(t-p)$. The autocorrelation function (ACF) will be defined as that average value of $v_{sr}(t)v_{sr}(t-p)$. 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.

$$\text{ACF} = R_{vv}(p) = \frac{1}{2T} \int_{-T}^T v(t)v(t-p) dt \quad (\text{eqn 3.29})$$

$$\text{ACF} = C_{vv}(p) = \int_{-\infty}^{\infty} v(t)v(t-p) dt \quad (\text{eqn 3.30})$$

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 v_{sr} and random noise voltage v_n 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.

$$v_r(t) = v_{sr}(t) + v_n(t) \quad (\text{eqn 3.31})$$

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.

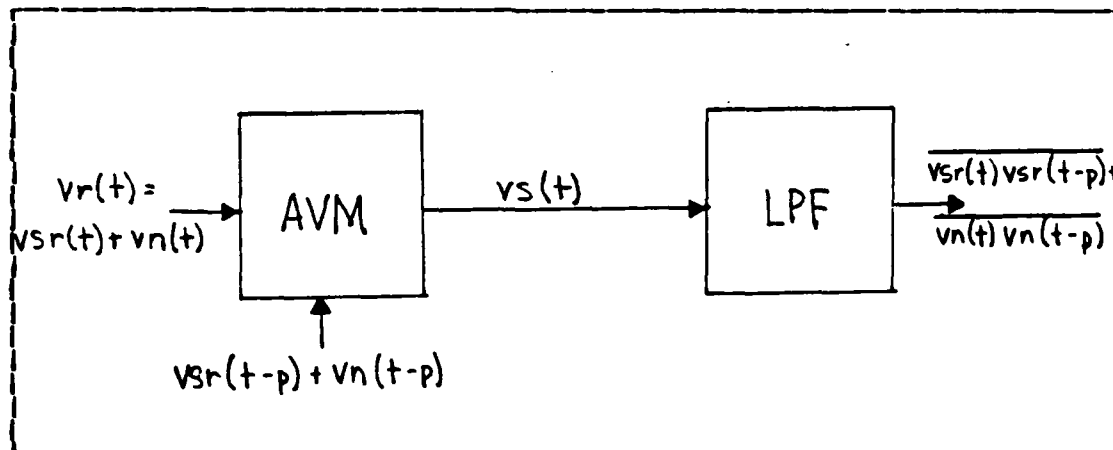


Figure 3.17 Voltage Correlator

$$\begin{aligned} & \overline{[v_{sr}(t) + v_n(t)] \cdot [v_{sr}(t-p) + v_n(t-p)]} = \quad \text{(eqn 3.32)} \\ & \overline{v_{sr}(t)v_{sr}(t-p)} + \overline{v_n(t)v_{sr}(t-p)} + \\ & \overline{v_{sr}(t)v_n(t-p)} + \overline{v_n(t)v_n(t-p)} = \end{aligned}$$

$$R_{v_{sr}} v_{sr}(p) + R_{v_n} v_{sr}(p) + R_{v_{sr}} v_n(p) + R_{v_n} v_n(p)$$

$$R_{v_n} v_{sr}(p) = 0$$

$$R_{v_{sr}} v_n(p) = 0; \text{ because average } v_n = 0$$

Therefore the result is simply $R_{v_{sr}} = v_{sr}(p) + R_{v_n} v_n(p)$ 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 $R_{v_{sr}} v_{sr}(p) / R_{v_n} v_n(p)$ 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.

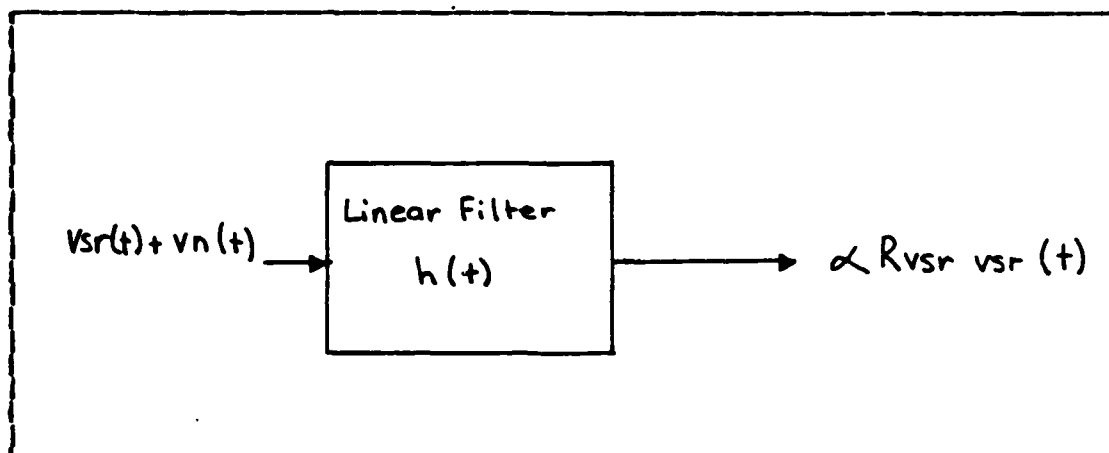


Figure 3.18 Matched Filter Block Representation

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 $v_{sr}(t)$ would be $v_{sr}(t) * h(t)$, where $*$ denotes convolution.

In the previous section it was stated that the autocorrelation function, $R_{vsr} v_{sr}(p)$, a function of p , is given by

$$R_{vsr} v_{sr}(p) = \int_{-\infty}^{\infty} v_{sr}(t) v_{sr}(t-p) dt \quad (\text{eqn 3.33})$$

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

$$R_{vsr} v_{sr}(t) = \int_{-\infty}^{\infty} v_{sr}(p) v_{sr}(p-t) dp \quad (\text{eqn 3.34})$$

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

$$v_{sr}(t) * h(t) = \int_{-\infty}^{\infty} v_{sr}(p) h(t-p) dp \quad (\text{eqn 3.35})$$

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

$$v_{sr}(t) * h(t) = \int_{-\infty}^{\infty} v_{sr}(p) v_{sr}(p-t) dp \quad (\text{eqn 3.36})$$

$$= R_{v_{sr}} v_{sr}(t)$$

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.

$$B = 3300 \text{ Hz}$$

$$f \geq 2B = 6600 \text{ samples/sec}$$

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

$$T_s = 1/f$$

$$T_s = 1/8000 \text{ samples/sec}$$

$$T_s = 125 \text{ microsec/sample}$$

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.

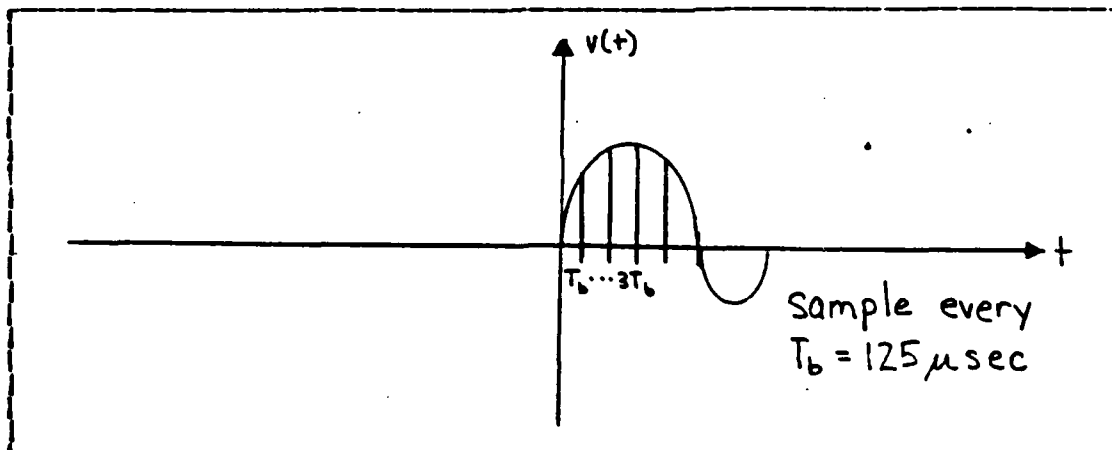


Figure 4.1 Samples of Voltage

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

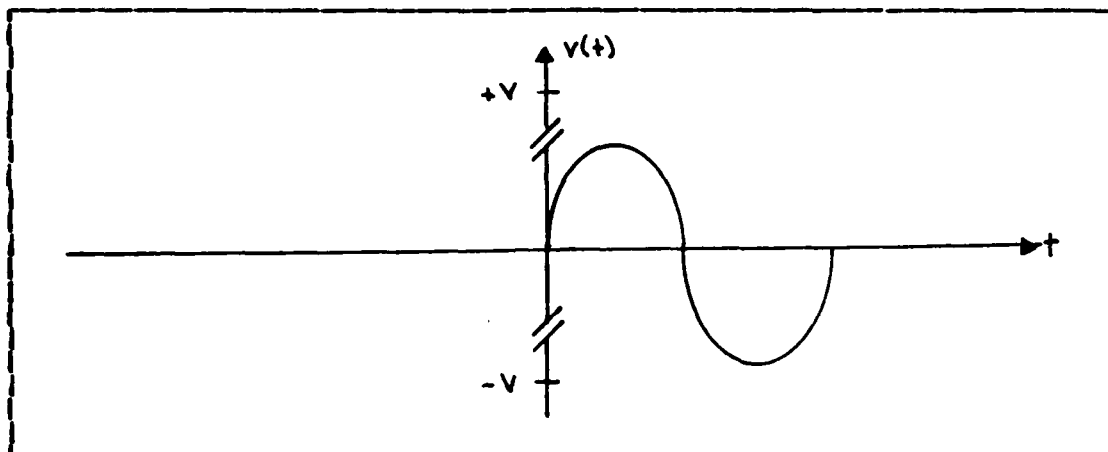


Figure 4.2 Analog Nature of Samples

$$\text{\#quantization levels} = 2^n \quad (\text{eqn 4.1})$$

for $n = 8$ bits/binary code word

$$\begin{aligned} \text{\#quantization levels} &= 2^8 \\ &= 256 \end{aligned}$$

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 $T_s = 125$ 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.

$$\begin{aligned}\text{bit rate} &= (n \text{ bits/sample}) (1 \text{ sample/T sec}) && (\text{eqn 4.2}) \\ \text{bit rate} &= (8 \text{ bits/sample}) (1 \text{ sample/125 microsec}) \\ &= 64 \text{ kbps}\end{aligned}$$

$$\begin{aligned}\text{bit duration} &= 1/\text{bit rate} && (\text{eqn 4.3}) \\ &= 1/64 \text{ kbps} = 15.6 \text{ microsec/bit}\end{aligned}$$

$$\begin{aligned}(\# \text{signals/channel}) (\text{bit rate/signal}) &= && (\text{eqn 4.4}) \\ \text{bit rate/channel}\end{aligned}$$

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

$$\begin{aligned}\text{bit rate/channel} &\geq && (\text{eqn 4.5}) \\ (24 \text{ signals/channel}) (64 \text{ kbps/signal}) \\ &= 1536 \text{ Mbps}\end{aligned}$$

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

Although 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.

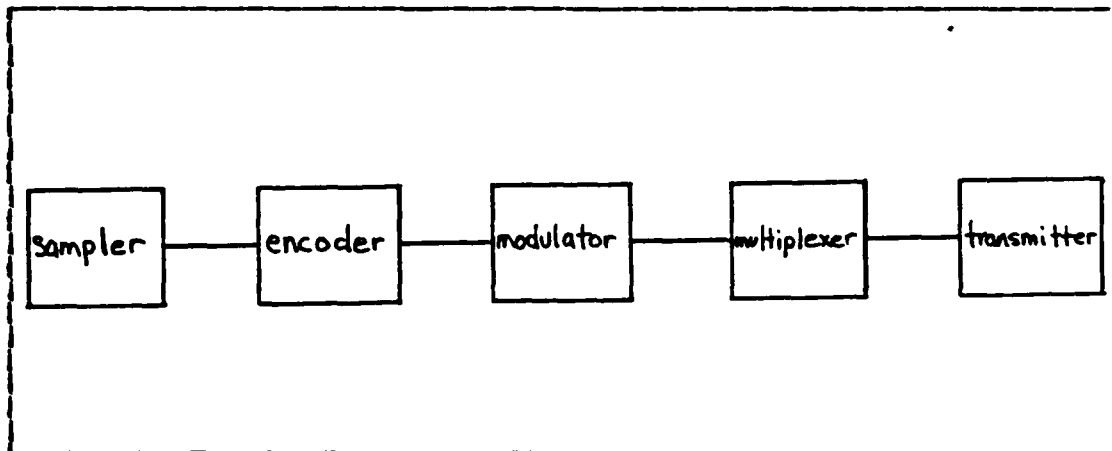


Figure 5.1 Basic Hardware Component Block Diagram

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 converted 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.

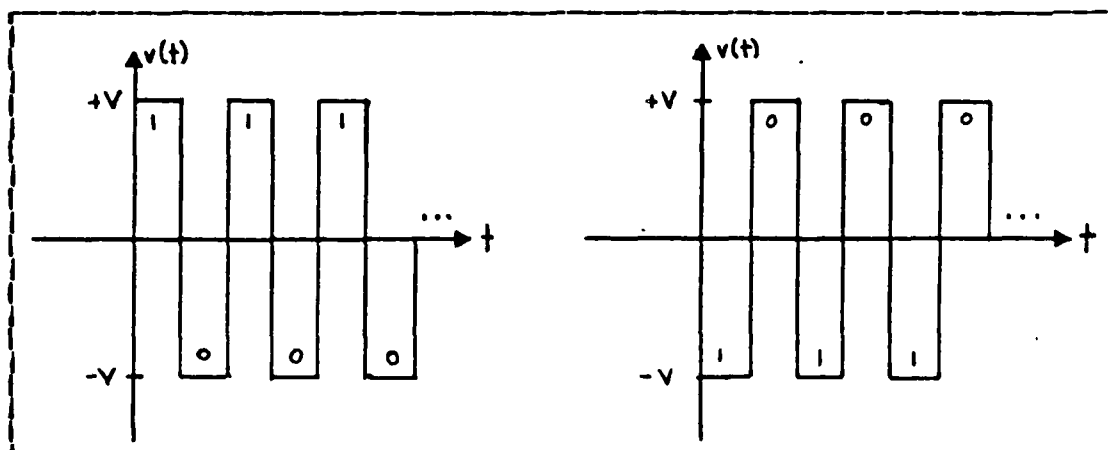


Figure 5.2 Bipolar Logic

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.

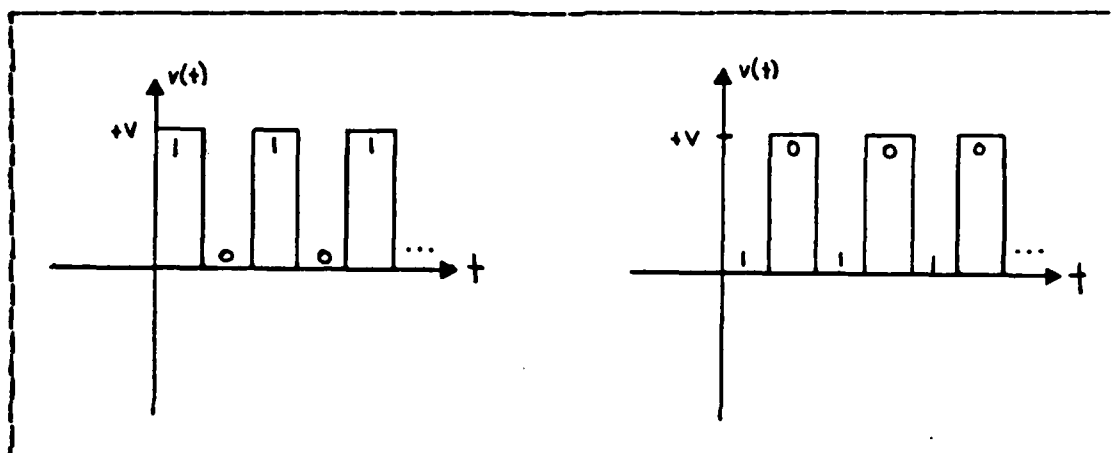


Figure 5.3 Unipolar Positive Logic

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 waveform 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.

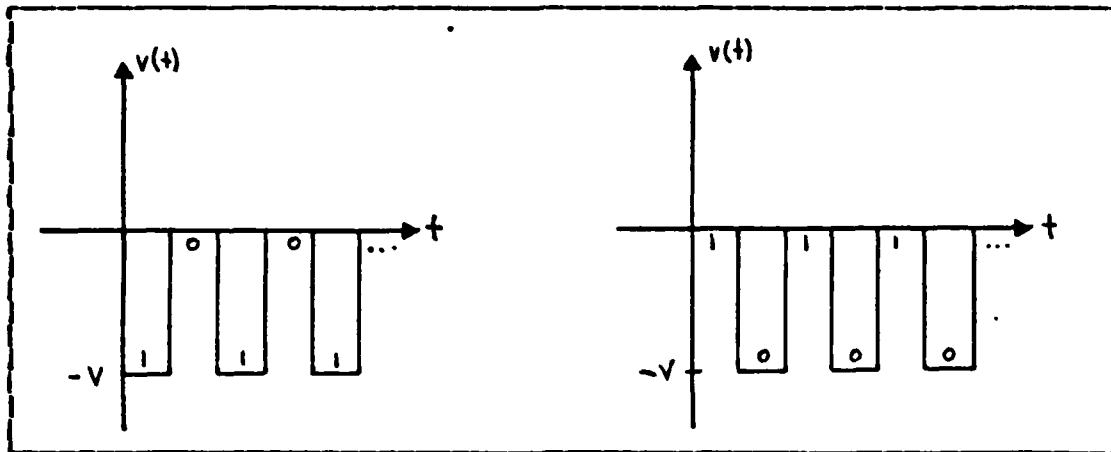


Figure 5.4 Unipolar Negative Logic

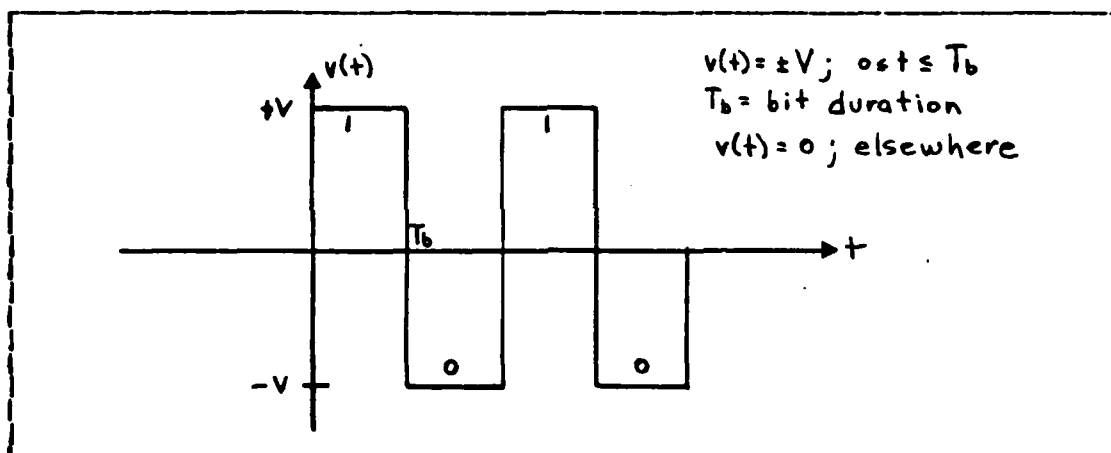


Figure 5.5 Non Return to Zero Waveform

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.

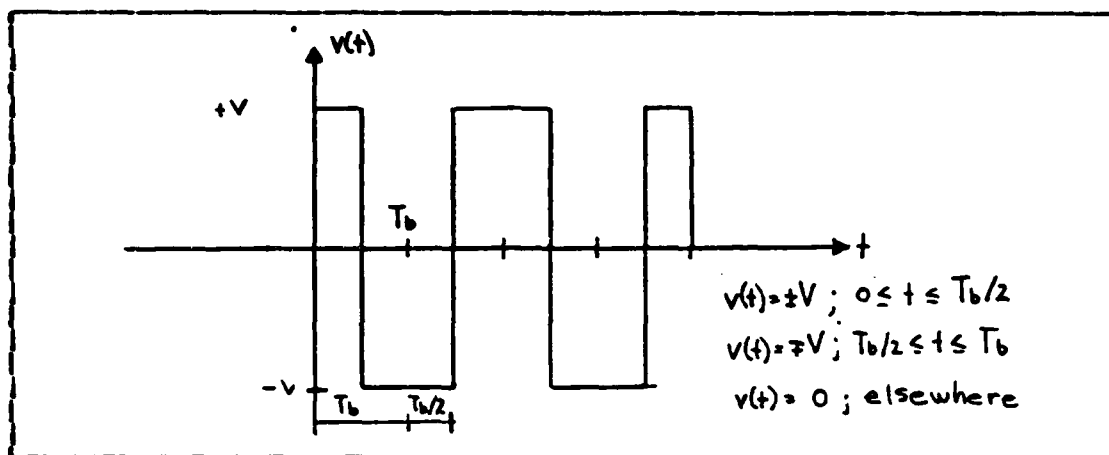


Figure 5.6 Manchester Waveform

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 $1/T_b$, where T_b 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. $R_b = \text{bit rate} = (8000 \text{ samples/sec}) (8 \text{ bits/sample})$
 $R_b = 64 \text{ kbps}$
4. $T_b = \text{bit duration} = 1/R_b$
 $T_b = 15.6 \text{ microsec}$

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

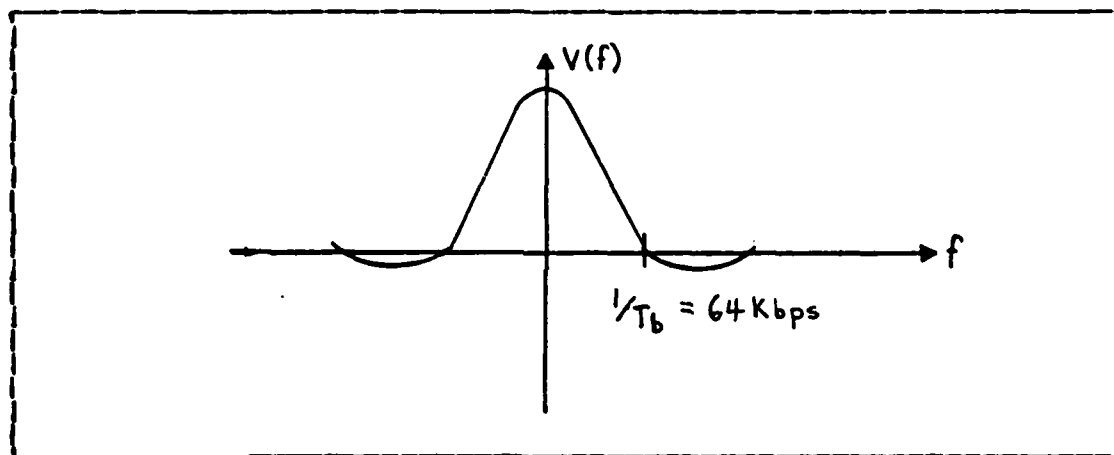


Figure 5.7 Fourier Transform of Baseband Waveform

It should be readily apparent that as the bit rate of the digital baseband waveform increases, so does the bandwidth. It is desirable 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 propagation 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 ω and initial phase angle δ as in equation 5.1.

$$v_c(t) = A \cos(\omega t + \delta) \quad (\text{eqn 5.1})$$

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

$$v_c(t) = A \cos(\omega t + \delta + \pi/2(1-v(t))) \quad (\text{eqn 5.2})$$

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.

$$v_c(t) = A v(t) \cos(\omega t + \delta) \quad (\text{eqn 5.3})$$

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 π 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,

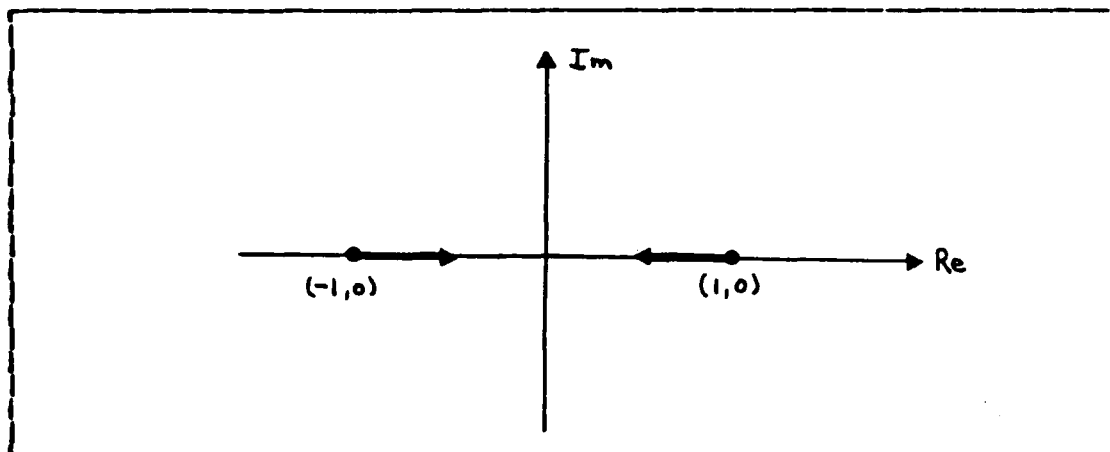


Figure 5.8 Phasor Diagram of BPSK

the bandwidth of the non-translated pulse, determined from the largest significant Fourier frequency component is $1/T_b$, where T_b is the bit duration. The bandwidth of the translated voltage (BPSK carrier) is $2/T_b$. The power in a sinusoid 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 $2/T_b$ and the translated wave has bandwidth $4/T_b$.

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 been 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 π phase change and is decoded as a bit change over successive bit durations. This can be expressed mathematically as in equation 5.4 where b_k represents the bit which will be encoded on the carrier and b_{k-1} represents the previous bit already encoded and a_k represents the present bit.

$$b_k = b_{k-1} \cdot a_k \quad (\text{eqn 5.4})$$

As can be seen when $b_k = -1$, there is a phase change and the bit changes from one bit to the next depending on the logic type being employed. If $b_k = +1$, 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. DBPSK takes advantage of the modulation of phase shift in the recovery process while DEBPSK still utilizes a phase coherent carrier

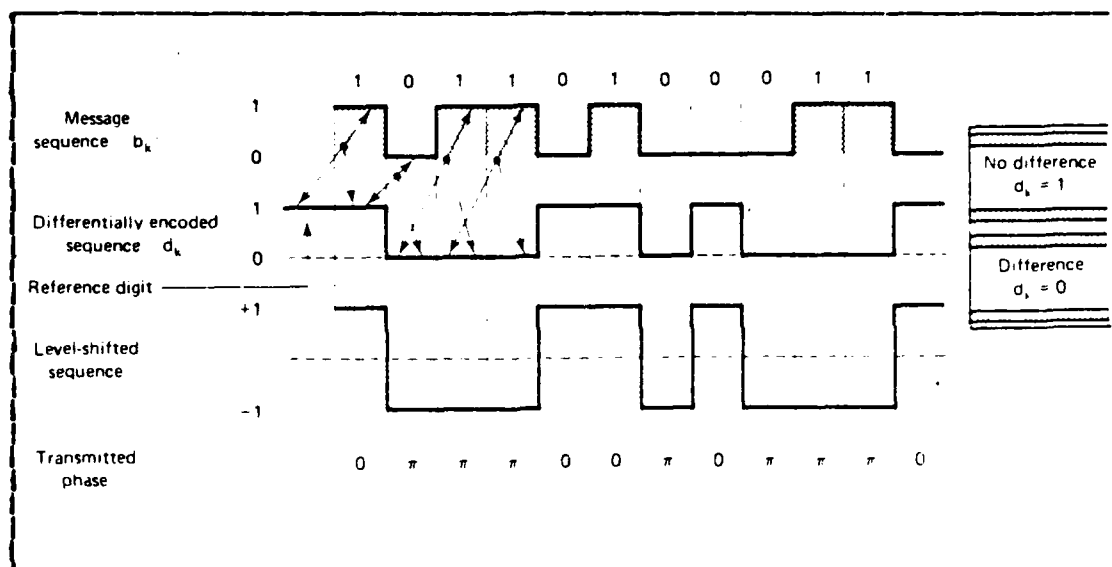


Figure 5.9 Differential Binary Phase Shift Keying

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].

$k = 2$ data word	Orthogonal chip sequence	Transmitted waveforms	
			BPSK carriers with phase shifts below:
1 1	1 1 1 1	$C_1(t)$	$\pi \quad \pi \quad \pi \quad \pi$
1 0	1 1 -1 -1	$C_2(t)$	$\pi \quad \pi \quad -\pi \quad -\pi$
0 1	1 -1 -1 1	$C_3(t)$	$\pi \quad -\pi \quad -\pi \quad \pi$
0 0	1 -1 1 -1	$C_4(t)$	$\pi \quad -\pi \quad \pi \quad -\pi$
			$\xleftarrow{T} \xrightarrow{T_w}$

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 done 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

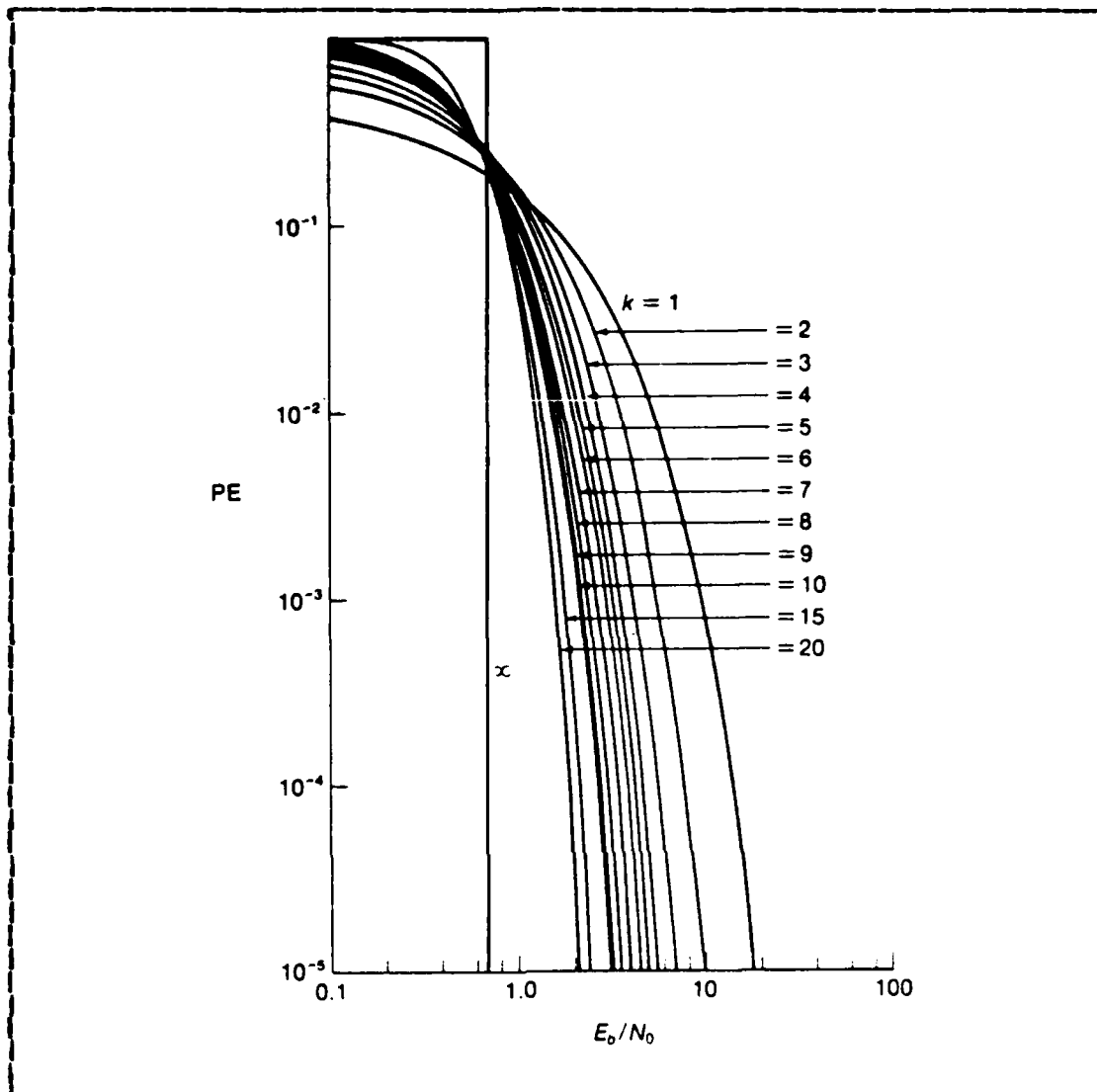


Figure 5.11 Probability of Error for Orthogonal BPSK

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 $(2 \cdot k/k)R$ 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.

$$v_c(t) = A v_1(t) \cos(\omega t + \delta) + A v_2(t) \sin(\omega t + \delta) \quad (\text{eqn 5.5})$$

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

$$v_c(t) = 2 \cdot .5 A \cos(\omega t + \theta + \delta) \quad (\text{eqn 5.6})$$

$$\theta = \arctan (v_1(t)/v_2(t)) \quad (\text{eqn 5.7})$$

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.

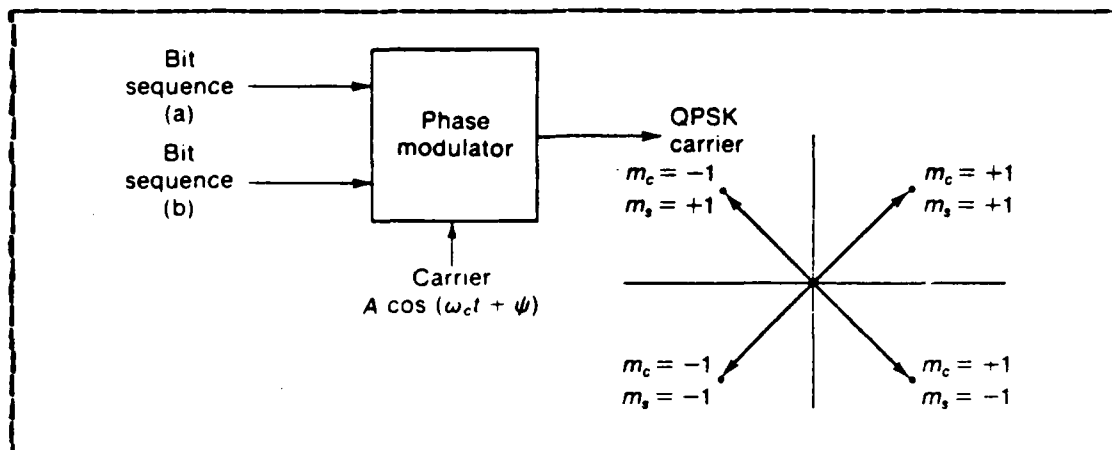


Figure 5.12 Modulation and Phases of QPSK

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

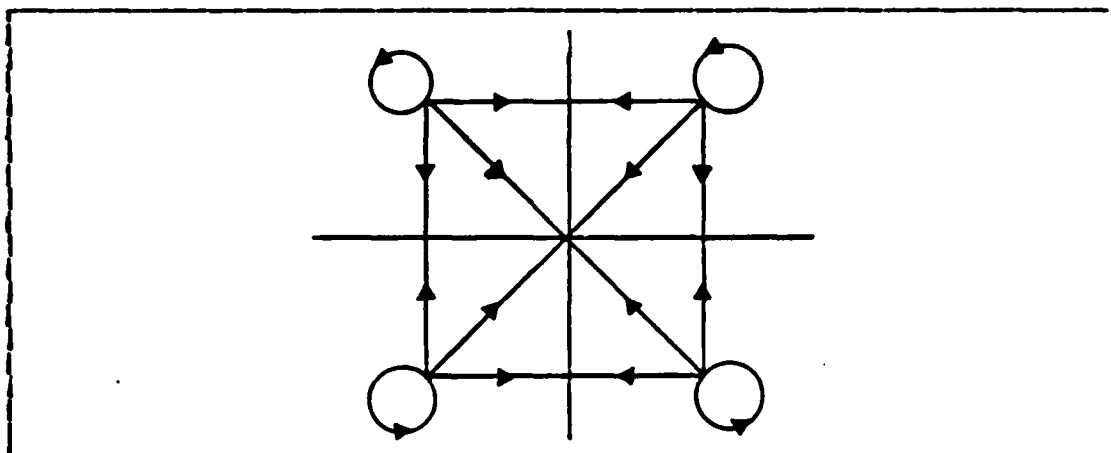


Figure 5.13 Phasor Diagram of QPSK

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 $T_s/2$ seconds, where T_s is symbol duration as illustrated in Figure 5.14. Modulation is then accomplished as before in QPSK.

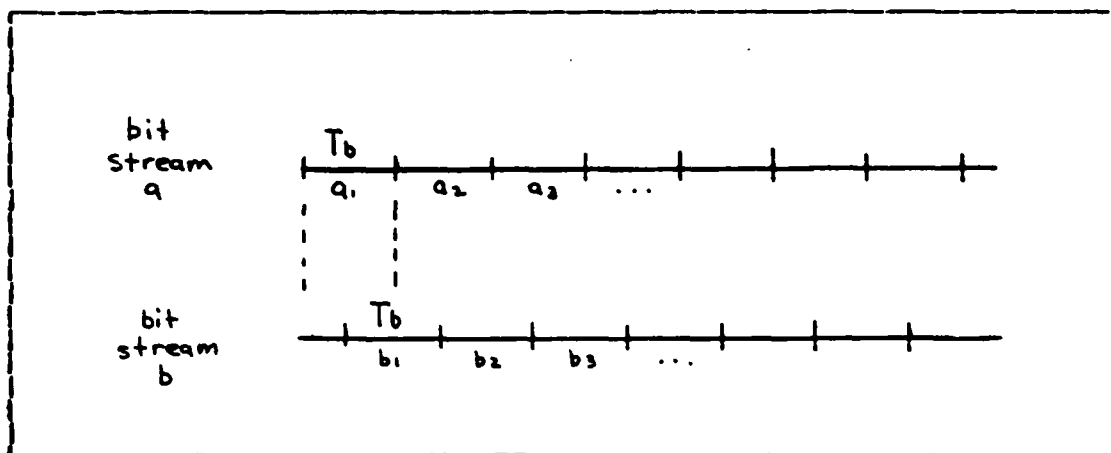


Figure 5.14 Offset Quadrature Phase Shift Keying

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].

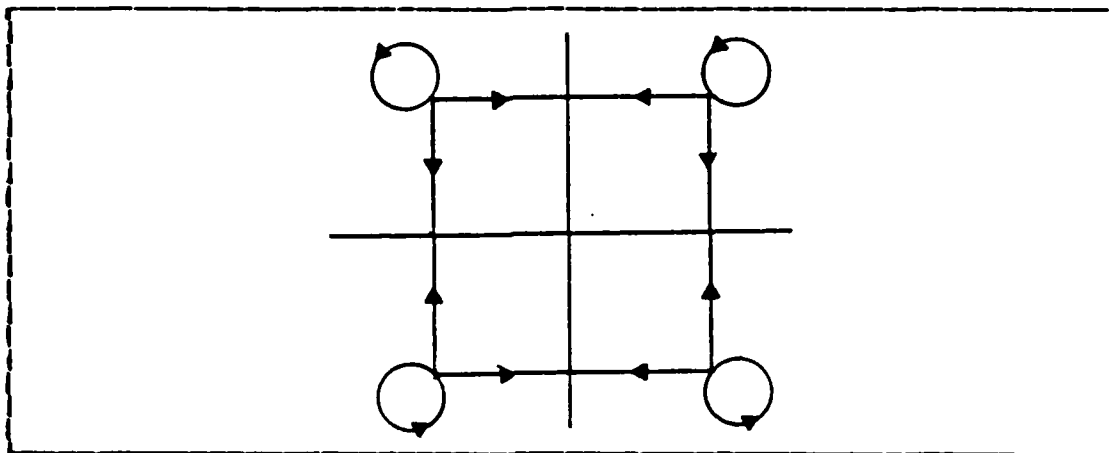


Figure 5.15 Phasor Diagram of OQPSK

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 $\theta_i = \text{phase} = (2\pi/m) i$; $i=0, 1, \dots, m-1$.

$$v_c(t) = A \cos(\omega t + \theta_i + \delta) \quad (\text{eqn 5.8})$$

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 to do 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 to variations in the baseband waveform. The information in the baseband waveform is thereby imparted to the carrier through this modulation. 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 $A_i = A/m_i$ volts apart, where A is the maximum amplitude and m_i 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 as presented earlier except for A_i which is equal to one of i different amplitudes depending on the bit sequence to be transmitted.

$$v_c(t) = A_i \cos(\omega t + \delta) \quad (\text{eqn 5.9})$$

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 $2/T_s$, T_s 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.

$$v_c(t) = A_1 \cos(\omega t + \delta) + A_2 \sin(\omega t + \delta) \quad (\text{eqn 5.10})$$

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. Minimum Shift Keying or Fast Frequency Shift Keying

Power requirement for the transmission or retransmission of a satellite signal is extremely important. It is desirable 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.

$$v_c(t) = A \cos((\omega_c + \Delta\omega)t) \quad (\text{eqn 5.11})$$

$$v_c(t) = A \cos(\pm\Delta\omega t) \cos(\omega_c t) - A \sin(\pm\Delta\omega t) \sin(\omega_c t) \quad (\text{eqn 5.12})$$

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 QPSK. The frequency variation in the carrier waveform is between $\omega_c + \Delta\omega$ and $\omega_c - \Delta\omega$. Maximum separation in phase of these two frequency components occurs at π as noted in equation 5.13, where T_b represents the bit or symbol duration.

$$(\omega_2 - \omega_1)T_b = \pi \quad (\text{eqn 5.13})$$

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

$$h = (f_2 - f_1)T_b = .5 \quad (\text{eqn 5.14})$$

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

$$v_c(t) = A \cos(\pm\pi t/2T_b) \cos(2\pi f_c t) - A \sin(\pm\pi t/2T_b) \sin(2\pi f_c t) \quad (\text{eqn 5.15})$$

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 $1/T_s$, where T_s 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

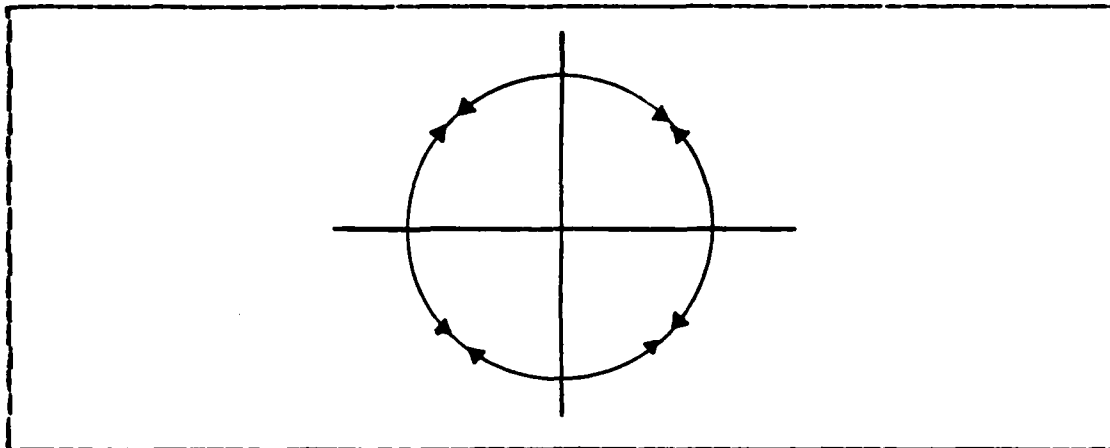


Figure 5.16 Phasor Diagram of FFSK or MSK

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 a_n and b_n represent the n th bit to be modulated and $h(t - nTs)$ represents the contribution of the

TABLE 1
QPRS SYSTEM POLYNOMIALS

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

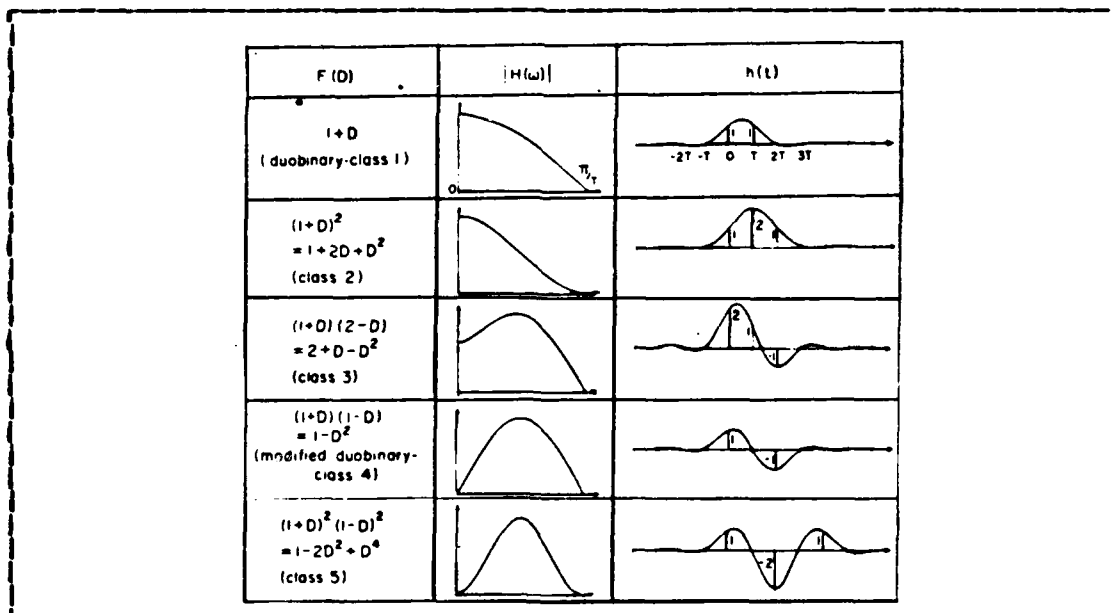


Figure 5.17 QPRS Linear Filter Impulse Responses

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

$$v_C(t) = \sum_{n=-\infty}^{\infty} a_n h(t - n T_s) \cos(\omega t + \delta) + \sum_{n=-\infty}^{\infty} b_n h(t - n T_s) \sin(\omega t + \delta) \quad (\text{eqn 5.16})$$

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.

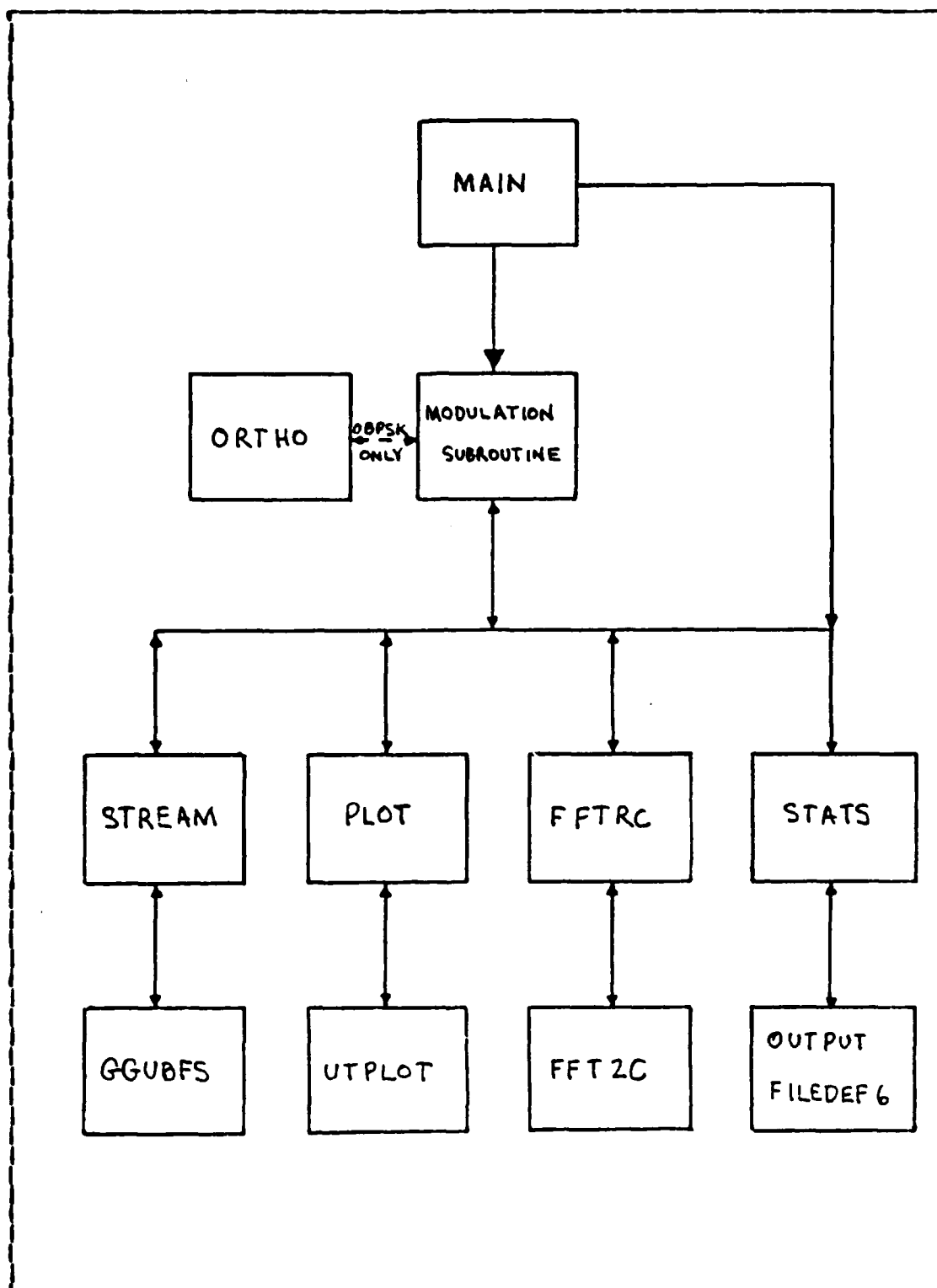


Figure 6.1 Representative Block Diagram

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 BAUD, 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

$$v_c(t) = A v(t) \cos(\omega t + \delta)$$

(eqn 6.1)

A = amplitude

v(t) = bit to be modulated

ω = angular frequency

t = time

δ = 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 $n \times n$ 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 2×2 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. SUBROUTINE 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

$$v_c(t) = A v_1(t) \cos(\omega t + \delta) + A v_2(t) \sin(\omega t + \delta) \quad (\text{eqn 6.2})$$

$v_1(t)$ = in-phase bit

$v_2(t)$ = quadrature bit

bits first being modulated are both known to have existed on the carrier for 1/2 the symbol duration. The first of these two bits only remains on the carrier until $\text{TIME} = 1(\text{BAUDD})$ when a third random bit is drawn and modulation begins with this bit. The second random bit is allowed to remain until $\text{TIME} = 1.5(\text{BAUDD})$ 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 $\text{TIME} = 0$ 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

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

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. A_i is one of $2^{**}k$ equally spaced amplitudes.

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

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 A_i according to the assignment routine expressed in equation 6.5.

$$A_i = A/m \quad (\text{eqn 6.5})$$

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

$$h = .5 = (\Delta f) T_b \quad (\text{eqn 6.6})$$

Δf = frequency separation

T_b = bit duration

The frequency deviation from the carrier is $.5(\Delta f) = 1/4T_b$. Converting this frequency deviation to radians yields $\pi/2T_b$. This angular frequency deviation from the central carrier is either $\pm\pi/1T_b$ 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.

$$v_c(t) = A \cos[(\omega \pm \pi/2T_b)t + \delta] \quad (\text{eqn 6.7})$$

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 $1/T_s$, where T_s 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 $(n-1)T_s$. 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 $\pm 1/2$ 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.

$$v_c(t) = A \cos[(\omega + 2\pi mf)t + \delta] \quad (\text{eqn 6.8})$$

mf = modulation frequency

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

or modulation stops do to completion of all required samples.

M. SUBROUTINE QPRS

This module 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.

Initially, 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 portions of the carrier and the sum is formed according to equation 6.9.

$$v_c(t) = \sum_{n=-\infty}^{\infty} A h(t - nTs) \cos(\omega t + \delta) + \sum_{n=-\infty}^{\infty} A h(t - nTs) \sin(\omega t + \delta) \quad (\text{eqn 6.9})$$

$h(t - nTs)$ = 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

ORTHO produces an $n \times n$ 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.

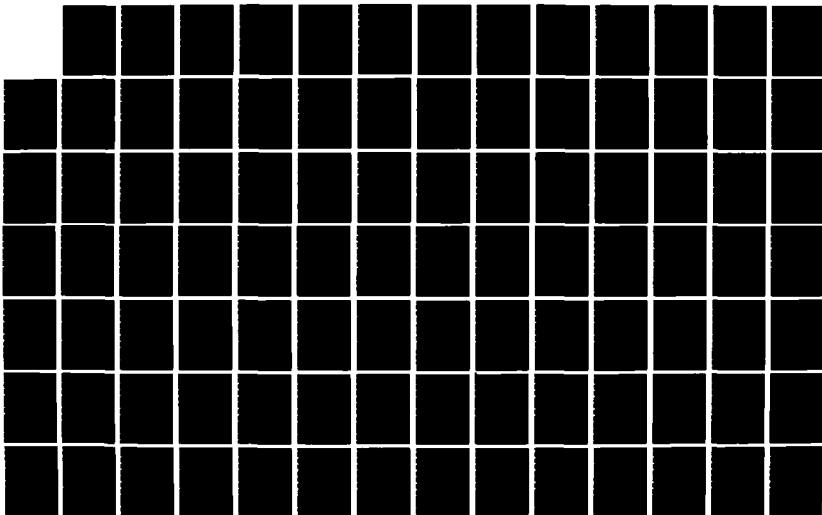
AD-A160 823

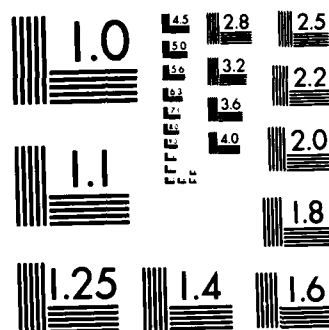
COMPUTER SIMULATION OF DIGITAL SIGNAL MODULATION
TECHNIQUES IN SATELLITE COMMUNICATIONS(U) NAVAL
POSTGRADUATE SCHOOL MONTEREY CA C D CARLSON SEP 85
F/G 9/2

2/4

UNCLASSIFIED

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

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 seems 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}	$Y_{1.}$	$\bar{Y}_{1.}$
	2	Y_{21}	Y_{22}	Y_{23}		Y_{2n}	$Y_{2.}$	$\bar{Y}_{2.}$
	3	Y_{31}	Y_{32}	Y_{33}		Y_{3n}	$Y_{3.}$	$\bar{Y}_{3.}$
	⋮							
	r	Y_{r1}	Y_{r2}	Y_{r3}		Y_{rn}	$Y_{r.}$	$\bar{Y}_{r.}$

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.

$$H_0: \bar{X}_{B11} = \bar{X}_{B21} = \dots = \bar{X}_{Bij} \quad (\text{eqn 7.1})$$

$$H_1: \bar{X}_{B11} \neq \bar{X}_{B21} \neq \dots \neq \bar{X}_{Bij}$$

i = i th modulation technique

j = j th component of the FFT

$$H_0: MVAR1 = MVAR2 = \dots = MVAR15 \quad (\text{eqn 7.2})$$

$$H_1: MVAR1 \neq MVAR2 \neq \dots \neq MVAR15$$

$$H_0: MSKEW1 = MSKEW2 = \dots = MSKEW15 \quad (\text{eqn 7.3})$$

$$H_1: MSKEW1 \neq MSKEW2 \neq \dots \neq MSKEW15$$

$$H_0: MKUR1 = MKUR2 = \dots = MKUR15 \quad (\text{eqn 7.4})$$

$$H_1: MKUR1 \neq MKUR2 \neq \dots \neq MKUR15$$

This indicates that there is a difference between the statistics 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 develop 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 CCMPONENT	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**

```

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

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

IBAUD=
BAUD=
BAUDD=
IBITS=
BITS=
BITR=
IFREQ=
IFREQ=
IPHAS=
IPHAS=
IAMP=
AMP=
ISEED=
DSEED=
ARRAY=
K=
IN=
N=
REPS=
REP=
FLAG=
FC=
STEP=
*** VARIABLE DECLARATIONS ***
INTEGER ANS, TYPE1, TYPE2, TYPE3, ANS2, IBAUD, IBITS, IFREQ,
*IIPHAS, IAMP, ISEED, R, IN, REPS, REP, FLAG
DOUBLE PRECISION BAUD, BITS, BITR, FREQ, IPHAS, AMP, BAUDD, STEP, FC,
*DSEED, N, PI
DOUBLE PRECISION ARRAY(1024, 2), SUMX(513), SUMXSQ(513), SUMX3(513),
*SUMX4(513)
COMMON ARRAY, SUMX, SUMXSQ, SUMX3, SUMX4
*** VARIABLE INITIALIZATIONS ***
R=1
PI=3.141592653589793D0
TYPE3=0
*** INTRODUCE THE PROGRAM ***
CALL FRCKMS('CLRSCRN ')

```

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
CCCC
CCCC

```



```

C      READ (5,*) IBAUD
C      BAUD=IBAUD
C      BAUD=1. DO/BAUD
C      *** DETERMINE THE BITS EITHER BY THE TYPE OF DIGITAL
C      MODULATION SPECIFIED OR BY USER INPUT ***
C      CALL FRTCMS ('CLRSCRN ')
C      IF (TYPE1.EQ. 1.OR.TYPE1.EQ. 2.OR.TYPE1.EQ. 9) THEN
C          IBITS=1
C          BITS=IBITS
C      ELSE IF (TYPE1.EQ. 3) THEN
C          WRITE (10,80)
C          FORMAT (' ENTER THE NUMBER OF BITS IN EACH SYMBOL. ', /)
C          * CAUTION! THE NUMBER OF BITS PER SYMBOL MUST BE 6 OR LESS
C          * //, ' : '
C      READ (5,*) IBITS
C      IF (IBITS.LT. 1.OR. IBITS.GT. 6) THEN
C          CALL FRTCMS ('CLRSCRN ')
C          WRITE (10,90)
C          FORMAT (' ERROR ', /)
C          GO TO 79
C      ELSE
C          BITS=IBITS
C      END IF
C      ELSE IF (TYPE1.EQ. 4.OR.TYPE1.EQ. 5.OR.TYPE1.EQ. 11) THEN
C          IBITS=2
C          BITS=IBITS
C      ELSE IF (TYPE1.EQ. 6.OR.TYPE1.LE. 8.OR.TYPE1.EQ. 10) THEN
C          WRITE (10,100)
C          FORMAT (' ENTER THE NUMBER OF BITS IN EACH SYMBOL. ', /)
C          * CAUTION! THE NUMBER OF BITS PER SYMBOL MUST BE 10 OR LESS
C          * //, ' : '
C      READ (5,*) IBITS
C      IF (IBITS.LT. 1.OR. IBITS.GT. 11) THEN
C          CALL FRTCMS ('CLRSCRN ')
C          WRITE (10,110)
C          FORMAT (' ERROR ', /)
C          GO TO 99
C      110

```

MAI 01950
MAI 01960
MAI 01970
MAI 01980
MAI 01990
MAI 02000
MAI 02010
MAI 02020
MAI 02030
MAI 02040
MAI 02050
MAI 02060
MAI 02070
MAI 02080
MAI 02090
MAI 02100
MAI 02110
MAI 02120
MAI 02130
MAI 02140
MAI 02150
MAI 02160
MAI 02170
MAI 02180
MAI 02190
MAI 02200
MAI 02210
MAI 02220
MAI 02230
MAI 02240
MAI 02250
MAI 02260
MAI 02270
MAI 02280
MAI 02290
MAI 02300
MAI 02310
MAI 02320
MAI 02330
MAI 02340
MAI 02350
MAI 02360
MAI 02370
MAI 02380
MAI 02390
MAI 02400
MAI 02410
MAI 02420
MAI 02430
MAI 02440

```

C      ELSE      BITS=IBITS
C      END IF
C
C      END IF
C
C      *** DETERMINE AND DISPLAY THE BIT RATE ***
C      IF (TYPE1.EQ.3) THEN
C      BITR=(2.D0**IBITS)*BAUD
C      ELSE
C      BITR=BITS*BAUD
C      END IF
C
C      CALL FRTCMS('CLRSCRN ')
C      WRITE(10,120)BITR
120    FORMAT(' THE BIT RATE FOR THE SPECIFIED SIGNAL IS',F9.0,' BITS/SEC
C      *)
C
C      *** HAVE THE USER ENTER THE CARRIER FREQUENCY ***
C      IF (TYPE1.EQ.3) THEN
C      FC=BITR
C      ELSE IF (TYPE1.EQ.9) THEN
C      FC=1.25D0*BAUD
C      ELSE IF (TYPE1.EQ.10) THEN
C      FC=BAUD+((2.D0**IBITS)-1.D0)*(BAUD/2.D0))
C      ELSE
C      FC=BAUD
C      END IF
C
C      WRITE(10,130)FC
129    FORMAT(' ENTER THE CARRIER FREQUENCY AT THIS TIME.',//H2.'',//)
130    *' NOTE: CARRIER FREQ SHOULD BE GREATER THAN',F9.0,' HZ.'',//)
C
C      READ(5,*)IFREQ
C
C      FREQ=IFREQ
C
C      IF (FREQ.LT.FC) THEN
C      CALL FRTCMS('CLRSCRN ')
C      WRITE(10,140)
140    FORMAT(' THE CARRIER FREQUENCY IS LESS THAN RECOMMENDED.',//)
C      GO TO 149
C      END IF
C
C      *** HAVE THE USER ENTER THE NUMBER OF SAMPLES TO BE GENERATED ***
C      CALL FRTCMS('CLRSCRN ')

```

```

C149 WRITE(10,150)
C150 FORMAT('ENTER THE NUMBER OF SAMPLES OF THE MODULATED SIGNAL'.
*//, 'TO BE PRODUCED. THIS NUMBER MUST BE EQUAL TO 2**N'.,//,
*//, 'WHERE N IS A POSITIVE INTEGER EQUAL TO OR LESS THAN 10.'.,//,
*//, 'I.E., 2, 4, 8, ... , 1024'.,//, '.:')
C READ(5,*)IN
C N=IN
C IF(IN.EQ.2.OR.IN.EQ.4.OR.IN.EQ.8.OR.IN.EQ.16.OR.IN.EQ.32.OR.
*//, 'IN.EQ.64.OR.IN.EQ.128.OR.IN.EQ.256.OR.IN.EQ.512.OR.IN.EQ.1024') THEN
C GO TO 170
C ELSE
C CALL FRTCMS('CLRSCRN ')
C WRITE(10,160)
C FORMAT('ERROR',/)
C GO TO 149
C END IF
C *** DETERMINE AND DISPLAY INFO ABOUT THE RECORD LENGTH ***
C IF(TYPE1.EQ.2) THEN
C STEP=1.D0/(2.D0*(FREQ+BITR))
C ELSE IF(TYPE1.EQ.9) THEN
C STEP=1.D0/(2.D0*(FREQ+(1.25D0*BAUD)))
C ELSE IF(TYPE1.EQ.10) THEN
C STEP=1.D0/(2.D0*(FREQ+((2.D0**IBITS)-1.)*(BAUD/2.D0))+BAUD))
C ELSE IF(TYPE1.EQ.11) THEN
C STEP=1.D0/(2.D0*(FREQ+(PI+1.D0)*BAUD)))
C ELSE
C STEP=1.D0/(2.D0*(FREQ+BAUD))
C END IF
C IF(TYPE1.EQ.3.AND.((N-1.)*STEP).LE.1./BITR) THEN
C CALL FRTCMS('CLRSCRN ')
C WRITE(10,180)
C FORMAT('1 BIT DURATION. IF YOU WISH TO CHANGE THIS BY'.,//,
*//, 'INCREASING THE NUMBER OF SAMPLES TO BE PRODUCED, BY'.,//,
*//, 'DECREASING CARRIER FREQUENCY, BY DECREASING THE BAUD'.,//,
*//, 'RATE OR BY DECREASING THE SIZE OF THE BINARY CODE WORD'.,//,
*//, 'ENTER A 1. ENTER ANY OTHER INTEGER VALUE TO CONTINUE'.,//,
*//, ':.')
C READ(5,*)ANS
C IF(ANS.EQ.1) THEN
C GO TO 29
C END IF

```



```

C      ELSE IF (TYPE1.EQ.2) THEN
C          CALL DBPSK(TYPE2,BAUD,FREQ,IPHAS,AMP,ANS2,DSEED,IN,REP)
C
C      ELSE IF (TYPE1.EQ.3) THEN
C          CALL OBPSK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,ANS2,DSEED,IN,REP)
C
C      ELSE IF (TYPE1.EQ.4) THEN
C          CALL QPSK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,ANS2,DSEED,IN,REP)
C
C      ELSE IF (TYPE1.EQ.5) THEN
C          CALL OQPSK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,ANS2,DSEED,IN,
C              *      REP)
C
C      ELSE IF (TYPE1.EQ.6) THEN
C          CALL MPSK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,ANS2,DSEED,IN,
C              *      REP)
C
C      ELSE IF (TYPE1.EQ.7) THEN
C          CALL MASK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,ANS2,DSEED,IN,
C              *      REP)
C
C      ELSE IF (TYPE1.EQ.8) THEN
C          CALL QASK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,ANS2,DSEED,IN,
C              *      REP)
C
C      ELSE IF (TYPE1.EQ.9) THEN
C          CALL MSK(TYPE2,BAUD,FREQ,IPHAS,AMP,ANS2,DSEED,IN,REP)
C
C      ELSE IF (TYPE1.EQ.10) THEN
C          CALL MFSK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,ANS2,DSEED,IN,REP)
C
C      ELSE IF (TYPE1.EQ.11) THEN
C          CALL QPRS(TYPE2,TYPE3,BAUD,BITS,FREQ,IPHAS,AMP,ANS2,DSEED,
C              *      IN,REP)
C
C      END IF
C
C      CONTINUE
C
C      ** OUTPUT A FINGERPRINT OF THE MODULATION ACCOMPLISHED ***
C
C      IF (TYPE1.EQ.1) THEN
C          WRITE(6,281)
C          FORMAT(1,MODULATION TECHNIQUE = BPSK')
C      ELSE IF (TYPE1.EQ.2) THEN
C          WRITE(6,282)
C          FORMAT(1,MODULATION TECHNIQUE = DBPSK')
C      ELSE IF (TYPE1.EQ.3) THEN
C          WRITE(6,283)

```

```

MAI 04450
MAI 04460
MAI 04470
MAI 04480
MAI 04490
MAI 04500
MAI 04510
MAI 04520
MAI 04530
MAI 04540
MAI 04550
MAI 04560
MAI 04570
MAI 04580
MAI 04590
MAI 04600
MAI 04610
MAI 04620
MAI 04630
MAI 04640
MAI 04650
MAI 04660
MAI 04670
MAI 04680
MAI 04690
MAI 04700
MAI 04710
MAI 04720
MAI 04730
MAI 04740
MAI 04750
MAI 04760
MAI 04770
MAI 04780
MAI 04790
MAI 04800
MAI 04810
MAI 04820
MAI 04830
MAI 04840
MAI 04850
MAI 04860
MAI 04870
MAI 04880
MAI 04890
MAI 04900
MAI 04910
MAI 04920
MAI 04930
MAI 04940

```

MAI 04950
MAI 04960
MAI 04970
MAI 04980
MAI 04990
MAI 05000
MAI 05010
MAI 05020
MAI 05030
MAI 05040
MAI 05050
MAI 05060
MAI 05070
MAI 05080
MAI 05090
MAI 05100
MAI 05110
MAI 05120
MAI 05130
MAI 05140
MAI 05150
MAI 05160
MAI 05170
MAI 05180
MAI 05190
MAI 05200
MAI 05210
MAI 05220
MAI 05230
MAI 05240
MAI 05250
MAI 05260
MAI 05270
MAI 05280
MAI 05290
MAI 05300
MAI 05310
MAI 05320
MAI 05330
MAI 05340
MAI 05350
MAI 05360
MAI 05370
MAI 05380
MAI 05390
MAI 05400
MAI 05410
MAI 05420
MAI 05430
MAI 05440

```

283      FORMAT('1', MODULATION TECHNIQUE = ORTHOGONAL BPSK')
      ELSE IF (TYPE1.EQ.4) THEN
284      WRITE(6,284)
      FORMAT('1', MODULATION TECHNIQUE = QPSK')
      ELSE IF (TYPE1.EQ.5) THEN
285      WRITE(6,285)
      FORMAT('1', MODULATION TECHNIQUE = OQPSK')
      ELSE IF (TYPE1.EQ.6) THEN
286      WRITE(6,286)
      FORMAT('1', MODULATION TECHNIQUE = MPSK')
      ELSE IF (TYPE1.EQ.7) THEN
287      WRITE(6,287)
      FORMAT('1', MODULATION TECHNIQUE = MASK')
      ELSE IF (TYPE1.EQ.8) THEN
288      WRITE(6,288)
      FORMAT('1', MODULATION TECHNIQUE = QASK')
      ELSE IF (TYPE1.EQ.9) THEN
289      WRITE(6,289)
      FORMAT('1', MODULATION TECHNIQUE = MSK')
      ELSE IF (TYPE1.EQ.10) THEN
290      WRITE(6,290)
      FORMAT('1', MODULATION TECHNIQUE = MFSK')
      ELSE IF (TYPE1.EQ.11) THEN
291      WRITE(6,291)
      FORMAT('1', MODULATION TECHNIQUE = QPRS')
      END IF
      C
      IF (TYPE1.EQ.11.AND.TYPE3.EQ.1) THEN
292      WRITE(6,292)
      FORMAT('1', QPRS CLASS 1 FILTER')
      ELSE IF (TYPE1.EQ.11.AND.TYPE3.EQ.2) THEN
293      WRITE(6,293)
      FORMAT('1', QPRS CLASS 2 FILTER')
      ELSE IF (TYPE1.EQ.11.AND.TYPE3.EQ.3) THEN
294      WRITE(6,294)
      FORMAT('1', QPRS CLASS 3 FILTER')
      ELSE IF (TYPE1.EQ.11.AND.TYPE3.EQ.4) THEN
295      WRITE(6,295)
      FORMAT('1', QPRS CLASS 4 FILTER')
      ELSE IF (TYPE1.EQ.11.AND.TYPE3.EQ.5) THEN
296      WRITE(6,296)
      FORMAT('1', QPRS CLASS 5 FILTER')
      END IF
      C
      IF (TYPE2.EQ.1) THEN
297      WRITE(6,297)
      FORMAT('1', BIPOLAR LOGIC')
      ELSE IF (TYPE2.EQ.2) THEN
298      WRITE(6,298)
      FORMAT('1', UNIPOLAR POSITIVE LOGIC')

```

```

299      ELSE IF (TYPE2.EQ.3) THEN
300          WRITE(6,299)
301          FORMAT(' UNIPOLAR NEGATIVE LOGIC')
302          END IF
303          WRITE(6,300) IBAUD
304          FORMAT(' BAUD OR SYMBOL RATE = ',I10,' HZ')
305          WRITE(6,301) IBITS
306          FORMAT(' BITS PER BINARY CODE WORD = ',I10)
307          WRITE(6,302) BITR
308          FORMAT(' BIT RATE = ',E23.16)
309          WRITE(6,303) IFREQ
310          FORMAT(' CARRIER FREQUENCY = ',I10,' HZ')
311          WRITE(6,304) IAMP
312          FORMAT(' MAXIMUM CARRIER AMPLITUDE = ',I10,' VOLT(S)')
313          WRITE(6,305) IIPHAS
314          FORMAT(' INITIAL PHASE ANGLE = ',I10,' DEGREES')
315          WRITE(6,306) STEP
316          FORMAT(' TIME BETWEEN SAMPLES = ',E23.16,' SEC')
317          WRITE(6,307) IN
318          FORMAT(' NUMBER OF SAMPLES GENERATED = ',I10)
319          IF (ANS2.EQ.1) THEN
320              WRITE(6,308) ISEED
321              FORMAT(' SEED FOR RANDOM NUMBER GENERATOR = ',I10)
322          END IF
323          WRITE(6,309) REPS
324          FORMAT(' NUMBER OF TIMES SIMULATION REPEATS = ',I10)
325          *** CALL THE STATISTICS SUBROUTINE FOR FINAL CALCULATIONS ***
326          FLAG=1
327          CALL STATS(IN,REP,FLAG)
328          IF (TYPE1.EQ.3.AND. ((N-1.00)*STEP).LE.1./BITR) THEN
329              WRITE(6,310)
330              FORMAT(' LESS THAN 1 BIT WAS MODULATED DURING EACH REPETITION
331              *
332              ELSE IF ((N-1.00)*STEP).LE.BAUDD) THEN
333              WRITE(6,320)
334              FORMAT(' LESS THAN 1 BAUD WAS MODULATED DURING EACH REPETITION
335              *

```

```

MAI 05450
MAI 05460
MAI 05470
MAI 05480
MAI 05490
MAI 05500
MAI 05510
MAI 05520
MAI 05530
MAI 05540
MAI 05550
MAI 05560
MAI 05570
MAI 05580
MAI 05590
MAI 05600
MAI 05610
MAI 05620
MAI 05630
MAI 05640
MAI 05650
MAI 05660
MAI 05670
MAI 05680
MAI 05690
MAI 05700
MAI 05710
MAI 05720
MAI 05730
MAI 05740
MAI 05750
MAI 05760
MAI 05770
MAI 05780
MAI 05790
MAI 05800
MAI 05810
MAI 05820
MAI 05830
MAI 05840
MAI 05850
MAI 05860
MAI 05870
MAI 05880
MAI 05890
MAI 05900
MAI 05910
MAI 05920
MAI 05930
MAI 05940

```

```
C C C  
C 330  
END IF  
*** DETERMINE IF ANOTHER SIMULATION IS TO BE RUN ***  
  
WRITE(10,330)  
FORMAT(' ENTER A 1 IF YOU DESIRE TO SIMULATE ANOTHER SIGNAL.',//,  
*' ENTER ANY OTHER INTEGER IF YOU ARE READY TO QUIT.',//,' :')  
  
READ(5,*)ANS  
  
IF (ANS.EQ.1) THEN  
    CALL FRCHMS('CLRSCRN ')  
    GO TO 29  
END IF  
  
STOP  
END
```

```

SUBROUTINE BPSK(TYPE2,BAUD,FREQ,IPHAS,AMP,ANS2,
*DSEED,IN,REP)
*** PURPOSE ***
THIS SUBROUTINE MODULATES THE CARRIER USING BINARY PHASE SHIFT
KEYING AS THE MODULATION TECHNIQUE.
*** PARAMETER DEFINITIONS ***
TYPE2= INDICATES LOGIC TYPE TO BE EMPLOYED
BAUD= SYMBOL RATE OR BAUD RATE
FREQ= CARRIER FREQUENCY
IPHAS= INITIAL PHASE ANGLE IN DEGREES
AMP= AMPLITUDE
ANS2= INTEGER VARIABLE TO BE PASSED TO STREAM
DSEED= DOUBLE PRECISION SEED FOR RANDOM NUMBER GENERATOR
IN= NUMBER OF POSITIONS IN ARRAY TO BE UTILIZED
REP= AN INTEGER EQUAL TO THE NUMBER OF THE REPETITION OF
THE CALL THE THIS SUBROUTINE
*** VARIABLE DEFINITIONS ***
ARRAY= A DOUBLE PRECISION ARRAY FOR PASSING THE BINARY
DIGITS AND STORING THE VALUE OF THE TIME FUNCTION
AND THE FFT
R= AN INTEGER USED TO REPRESENT THE ROW OF ARRAY
TIME= TIME VARIABLE
MT= VALUE OF THE BINARY DIGIT
STEP= INTERVAL AT WHICH THE SIGNAL IS REPRODUCED
PI= NUMERICAL CONSTANT
OMEGA= CARRIER ANGULAR FREQUENCY
DELTA= INITIAL PHASE OFFSET IN RADIAN
NBAUD= A COUNT OF THE NUMBER OF BAUDS MODULATED
BAUDD= BAUD DURATION
X= COMPLEX ARRAY TO RECEIVE THE FFT OF THE TIME SERIES
IWK= INTEGER WORKING ARRAY USED BY FUNCTION FFTRC
MAXY= MAXIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
MINY= MINIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
INT= INTERVAL BETWEEN POINTS ON THE ORDNATE
RMAX= VALUE OF THE PRINCIPLE HARMONIC
FLAG= AN INTEGER WHICH CONTROL ENTRANCE AND EXIT FROM
SUBPROGRAM STATS
IND2P1= ARRAY LENGTH DIVIDED BY 2 PLUS 1
*** VARIABLE DECLARATIONS ***
INTEGER R,ANS2,TYPE2,IN,KMAX,REP,FLAG,IND2P1
DOUBLE PRECISION BAUD,FREQ,IPHAS,AMP,TIME,MT,

```

BPS00110
BPS00120
BPS00130
BPS00140
BPS00150
BPS00160
BPS00170
BPS00180
BPS00190
BPS00200
BPS00210
BPS00220
BPS00230
BPS00240
BPS00250
BPS00260
BPS00270
BPS00280
BPS00290
BPS00300
BPS00310
BPS00320
BPS00330
BPS00340
BPS00350
BPS00360
BPS00370
BPS00380
BPS00390
BPS00400
BPS00410
BPS00420
BPS00430
BPS00440
BPS00450
BPS00460
BPS00470
BPS00480
BPS00490
BPS00500
BPS00510
BPS00520
BPS00530
BPS00540
BPS00550
BPS00560
BPS00570
BPS00580
BPS00590
BPS00600


```

C C C C C
*** GENERATE THE FFT ***
CALL FFTFC(ARRAY(1,1),IN,X,INK)

*** CALCULATE THE AMPLITUDE SPECTRUM OF THE FUNCTION ***
*** FIND THE NUMBER AND VALUE OF THE PRINCIPLE HARMONIC ***

N=0.DO
R=1
MAXY=0.DO
DO 20 I=1,IND2P1
  ARRAY(R,1)=CDABS(X(R))
  ARRAY(R,2)=N/STEP
  IF(ARRAY(R,1).GT.MAXY) THEN
    MAXY=ARRAY(R,1)
    RMAX=R-1
  END IF
  R=R+1
  N=N+1.DO
CONTINUE

*** DISPLAY INFO IF THE FIRST TIME THROUGH SUBPROGRAM ***

IF(REP.EQ.1) THEN
  WRITE(10,30)RMAX
  FORMAT(' THE FFT HAS BEEN GENERATED! THE PRINCIPLE HARMONIC',//
    ' IS THE',I5,' HARMONIC. THE NEXT PLOT TO BE PRODUCED WILL',//
    ' BE THE AMPLITUDE SPECTRUM. ENTER A 1 IF YOU ARE READY TO',//
    ' CONTINUE WITH THE PROGRAM.',//,' :')
  READ(5,*)L
  IF(L.NE.1) THEN
    CALL FRTCMS('CLRSCRN ')
    WRITE(10,40)
    FORMAT(' ERROR',//)
    GO TO 29
  END IF
END IF

*** ALLOW FOR THE PLOT OF THE AMPLITUDE SPECTRUM OF THE FFT ***

IF(REP.EQ.1) THEN
  MINY=0.DO
  INT=1.DO/STEP
  CALL PLOT(MAXY,MINY,INT,IND2P1)
END IF

```

BPS01110
 BPS01120
 BPS01130
 BPS01140
 BPS01150
 BPS01160
 BPS01170
 BPS01180
 BPS01190
 BPS01200
 BPS01210
 BPS01220
 BPS01230
 BPS01240
 BPS01250
 BPS01260
 BPS01270
 BPS01280
 BPS01290
 BPS01300
 BPS01310
 BPS01320
 BPS01330
 BPS01340
 BPS01350
 BPS01360
 BPS01370
 BPS01380
 BPS01390
 BPS01400
 BPS01410
 BPS01420
 BPS01430
 BPS01440
 BPS01450
 BPS01460
 BPS01470
 BPS01480
 BPS01490
 BPS01500
 BPS01510
 BPS01520
 BPS01530
 BPS01540
 BPS01550
 BPS01560
 BPS01570
 BPS01580
 BPS01590
 BPS01600

BPS01610
BPS01620
BPS01630
BPS01640
BPS01650
BPS01660
BPS01670
BPS01680

*** ADD THE VALUES OF THE FFT TO THE ACCUMULATED STATISTICS ***

FLAG=0

CALL STATS(IN,FEF,FLAG)

RETURN

END

C
C
C

C

```

SUBROUTINE DEPSK (TYPE2,BAUD,FREQ,IPHAS,AMP,ANS2,
*DSEED,IN,REP)
*** PURPOSE ***
THIS SUBROUTINE MODULATES THE CARRIER USING DIFFERENTIAL BINARY
SHIFT KEYING AS THE MODULATION TECHNIQUE.
*** PARAMETER DEFINITIONS ***
TYPE2= INDICATES LOGIC TYPE TO BE EMPLOYED
BAUD= SYMBOL RATE OR BAUD RATE
FREQ= CARRIER FREQUENCY
IPHAS= INITIAL PHASE ANGLE IN DEGREES
AMP= AMPLITUDE
ANS2= INTEGER VARIABLE TO BE PASSED TO STREAM
DSEED= DOUBLE PRECISION SEED FOR RANDOM NUMBER GENERATOR
IN= NUMBER OF POSITIONS OF ARRAY TO BE UTILIZED
FLAG= THE CALL THE THIS SUBROUTINE
      AN INTEGER WHICH CONTROL ENTRANCE AND EXIT FROM
*** VARIABLE DEFINITIONS ***
ARRAY= A DOUBLE PRECISION ARRAY FOR PASSING THE BINARY
      DIGITS AND STORING THE VALUE OF THE TIME FUNCTION
      AND THE FFT
R= AN INTEGER USED TO REPRESENT THE ROW OF ARRAY
TIME= TIME VARIABLE
MT= VALUE OF THE BINARY DIGIT
STEP= INTERVAL AT WHICH THE SIGNAL IS REPRODUCED
PI= NUMERICAL CONSTANT
OMEGA= CARRIER ANGULAR FREQUENCY
DELTA= INITIAL PHASE OFFSET IN RADIAN
NBAUD= A COUNT OF THE NUMBER OF BAUDS MODULATED
BAUDD= BAUD DURATION
REF= REFERENCE VOLTAGE
BK= VALUE OF DIFFERENTIAL BIT TO BE MODULATED
X= COMPLEX ARRAY TO RECEIVE THE FFT OF THE TIME SERIES
IWK= INTEGER ARRAY USED IN SUBROUTINE FFTRC
MAXY= MAXIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
MINY= MINIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
KMAX= PRINCIPLE HARMONIC
REP= AN INTEGER EQUAL TO THE NUMBER OF THE REPETITION OF
      SUBPROGRAM STATS
IND2P1= LENGTH OF ARRAY DIVIDED BY 2 + 1
*** VARIABLE DECLARATIONS ***
INTEGER R,ANS2,TYPE2,IN,EMAX,REP,FLAG,IND2P1

```

```

DBP00110
DBP00120
DBP00130
DBP00140
DBP00150
DBP00160
DBP00170
DBP00180
DBP00190
DBP00200
DBP00210
DBP00220
DBP00230
DBP00240
DBP00250
DBP00260
DBP00270
DBP00280
DBP00290
DBP00300
DBP00310
DBP00320
DBP00330
DBP00340
DBP00350
DBP00360
DBP00370
DBP00380
DBP00390
DBP00400
DBP00410
DBP00420
DBP00430
DBP00440
DBP00450
DBP00460
DBP00470
DBP00480
DBP00490
DBP00500
DBP00510
DBP00520
DBP00530
DBP00540
DBP00550
DBP00560
DBP00570
DBP00580
DBP00590
DBP00600

```

```

C      DOUBLE PRECISION BAUD,FREQ,IPHAS,AMP,TIME,MT,
C      *STEP,PI,OMEGA,DELTA,NBAUD,BAUDD,
C      *DSEED,REF,BK,MAXY,MINY,INT
C
C      DOUBLE PRECISION ARRAY(1024,2),SUMX(513),SUMXSQ(513),SUMX3(513),
C      *SUMX4(513)
C      COMMON ARRAY,SUMX,SUMXSQ,SUMX3,SUMX4
C
C      COMPLEX*16 X(513)
C
C      INTEGER IWK(10)
C
C      *** VARIABLE INITIALIZATION ***
C
C      R=1
C      TIME=0.D0
C      STEP=1.D0/(2.D0*(FREQ+BAUD))
C      PI=3.141592653589793D0
C      OMEGA=2.D0*PI*FREQ
C      DELTA=IPHAS*PI/180.D0
C      NBAUD=1.D0
C      BAUDD=1.D0/BAUD
C      REF=1.D0
C      MAXY=0.D0
C      IND2P1=IN/2+1
C
C      *** DO THE MODULATION AND ASSIGN THE VALUES TO ARRAY ***
C
C      CALL STREAM(DSEED,ANS2,TYPE2,MT)
C
C      BK=REF*MT
C
C      DO 10 I=1,IN
C        ARRAY(R,1)=AMP*BK*DCOS((OMEGA*TIME)+DELTA)
C        ARRAY(R,2)=TIME
C        IF(ARRAY(R,1).GT.MAXY) THEN
C          MAXY=ARRAY(R,1)
C        END IF
C        R=R+1
C        TIME=TIME+STEP
C        IF(TIME.GT.NBAUD*BAUDD) THEN
C          CALL STREAM(DSEED,ANS2,TYPE2,MT)
C          BK=BK*MT
C          NBAUD=NBAUD+1.D0
C        END IF
C      CONTINUE
C      10
C
C      *** ALLOW FOR THE PLOT OF THE TIME SERIES ***
C
C      IF (REP.EQ.1) THEN

```



```

DBP01610
DBP01620
DBP01630
DBP01640
DBP01650
DBP01660
DBP01670
DBP01680
DBP01690
DBP01700
DBP01710
DBP01720
DBP01730

```

```

MINY=0. DO
INT=1. DO/STEP
CALL PLOT(MAXY, MINY, INT, IND2P1)
END IF
*** ADD THE VALUES OF THE FFT TO THE ACCUMULATED STATISTICS ***
FLAG=0
CALL STATS(IN, REP, FLAG)
RETURN
END

```

```

C
C
C
C

```

```

SUBROUTINE OBPSK (TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,ANS2,
*DSEED,IN,REP)
*** PURPOSE ***
THIS SUBROUTINE MODULATES THE CARRIER USING M-ARY PHASE SHIFT
KEYING AS THE MODULATION TECHNIQUE.
*** PARAMETER DEFINITIONS ***
TYPE2= INDICATES LOGIC TYPE TO BE EMPLOYED
BAUD= SYMBOL RATE OR BAUD RATE
BITS= NUMBER OF BITS IN EACH BINARY CODE WORD
FREQ= CARRIER FREQUENCY
IPHAS= INITIAL PHASE ANGLE IN DEGREES
AMP= AMPLITUDE
ANS2= INTEGER VARIABLE TO BE PASSED TO STREAM
DSEED= DOUBLE PRECISION SEED FOR RANDOM NUMBER GENERATOR
IN= NUMBER OF POSITIONS IN ARRAY TO BE UTILIZED
REP= AN INTEGER EQUAL TO THE NUMBER OF THE REPETITION OF
THE CALL THE THIS SUBROUTINE
*** VARIABLE DEFINITIONS ***
ARRAY= A DOUBLE PRECISION ARRAY FOR PASSING THE BINARY
DIGITS AND STORING THE VALUE OF THE TIME FUNCTION
AND THE FFT
AN INTEGER USED TO REPRESENT THE ROW OF ARRAY
TIME VARIABLE
VALUE OF THE BINARY DIGIT
INTERVAL AT WHICH THE SIGNAL IS REPRODUCED
NUMERICAL CONSTANT
CARRIER ANGULAR FREQUENCY
INITIAL PHASE OFFSET IN RADIANS
A COUNT OF THE NUMBER OF BITS MODULATED
BIT RATE
BIT DURATION
COMPLEX ARRAY TO RECEIVE THE FFT OF THE TIME SERIES
INTEGER WORKING ARRAY USED BY FUNCTION FFTC
MAXIMUM VALUE TO BE PLOTTED ON THE ABSCISSAE
MINIMUM VALUE TO BE PLOTTED ON THE ABSCISSAE
INTERVAL BETWEEN POINTS ON THE ORDNATE
VALUE OF THE PRINCIPLE HARMONIC
INTEGER VALUE OF THE NUMBER OF BITS IN A BINARY WORD
INTEGER VALUE OF K MINUS 1
NUMBER OF POSSIBLE BINARY CODE WORDS = 2**K
DOUBLE PRECISION OF N
NDP DIVIDED BY 2
VARIABLE USED TO HOLD THE DECIMAL EQUIVALENT TO A
BINARY CODE WORD OF LENGTH K

```

```

OBP001110
OBP001120
OBP001130
OBP001140
OBP001150
OBP001160
OBP001170
OBP001180
OBP001190
OBP002000
OBP002010
OBP002020
OBP002030
OBP002040
OBP002050
OBP002060
OBP002070
OBP002080
OBP002090
OBP002100
OBP002110
OBP002120
OBP002130
OBP002140
OBP002150
OBP002160
OBP002170
OBP002180
OBP002190
OBP002200
OBP002210
OBP002220
OBP002230
OBP002240
OBP002250
OBP002260
OBP002270
OBP002280
OBP002290
OBP002300
OBP002310
OBP002320
OBP002330
OBP002340
OBP002350
OBP002360
OBP002370
OBP002380
OBP002390
OBP002400
OBP002410
OBP002420
OBP002430
OBP002440
OBP002450
OBP002460
OBP002470
OBP002480
OBP002490
OBP002500
OBP002510
OBP002520
OBP002530
OBP002540
OBP002550
OBP002560
OBP002570
OBP002580
OBP002590
OBP002600

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

C C C C C C C
SUM2=      VARIABLE USED TO HOLD THE DECIMAL CONVERSION OF SUM1
FLAG=      AN INTEGER WHICH CONTROL ENTRANCE AND EXIT FROM
            SUBPROGRAM STATS AND ORTHO
IND2P1=    LENGTH OF ARRAY DIVIDED BY 2 + 1

*** VARIABLE DECLARATIONS ***
INTEGER R,ANS2,TYPE2,IN,RMAX,K,KM1,REP,FLAG,IND2P1,N,C
DOUBLE PRECISION BAUD,BITS,FREQ,IPHAS,AMP,TIME,MT,
*STEP,PI,OMEGA,DELTA,NBIT,BITD,BITR
*DSEED,MXY,MINY,INT,NDP,ND2,SUM1,SUM2
DOUBLE PRECISION ARRAY(1024,2),SUMX(513),SUMXSQ(513),SUMX3(513),
*SUMX4(513)
COMMON ARRAY,SUMX,SUMXSQ,SUMX3,SUMX4
COMPLEX*16 X(513)
INTEGER INK(10)

*** VARIABLE INITIALIZATION ***
R=1
K=BITS
N=2**K
NDP=N
NBIT=1.DO
BITR=N*BAUD
BITD=1.DO/BITR
TIME=0.DO
STEP=1.DO/(2.DO*(FREQ+BITR))
PI=3.141592653589793D0
OMEGA=2.DO*PI*FREQ
DELTA=IPHAS*PI/180.DO
MXY=0.DO
IND2P1=IN/2+1

*** CONVERT A STREAM OF K BITS TO A SINGLE REPRESENTATIVE BINARY
VARIABLE IN DECIMAL FORM ***
SUM1=0.DO
KM1=K-1
DO 10 I=1,K
  CALL STREAM(DSEED,ANS2,TYPE2,MT)
  IF(TYPE2.EQ.1.AND.MT.EQ.-1.DO) THEN
    MT=0.DO
  ELSE IF(TYPE2.EQ.3.AND.MT.EQ.-1.DO) THEN
    MT=1.DO
  END IF
10 CONTINUE
C C C C

```

```

10      SUM1=SUM1+(MT*(10.DO**KM1))
C      KM1=KM1-1
C      CONTINUE
C      *** CONVERT THE REPRESENTATIVE BINARY VARIABLE TO ITS DECIMAL
C      EQUIVALENT ***
C      SUM2=0.DO
C      KM1=K-1
C      ND2=NDP/2.DO
C      DO 20 I=1,K
C          IF(SUM1/10.DO**KM1.GE.1.DO) THEN
C              SUM2=SUM2+ND2
C              SUM1=SUM1-10.DO**KM1
C              ND2=ND2/2.DO
C              KM1=KM1-1
C          ELSE
C              ND2=ND2/2.DO
C              KM1=KM1-1
C          END IF
C      CONTINUE
C      *** DO THE MODULATION AND ASSIGN TO ARRAY ***
C      C=1
C      FLAG=0
C      CALL ORTHO(K,MT,SUM2,C,FLAG)
C      FLAG=2
C      DO 30 I=1,IIN
C          ARRAY(R,1)=AMP*MT*DCOS((OMEGA*TIME)+DELTA)
C          ARRAY(R,2)=TIME
C          IF(ARRAY(R,1).GT.MAXY) THEN
C              MAXY=ARRAY(R,1)
C          END IF
C          R=R+1
C          TIME=TIME+STEP
C          IF(TIME.GT.NBIT*BITD) THEN
C              NBIT=NBIT+1.DO
C              IF(C.GT.N) THEN
C                  *** CONVERT A STREAM OF K BITS TO A SINGLE REPRESENTATIVE BINARY
C                  VARIABLE IN DECIMAL FORM ***
C                  SUM1=0.DO
C                  KM1=K-1
C                  DO 40 J=1,K
C                      CALL STREAM(DSEED,ANS2,TYPE2,MT)
C                      IF (TYPE2.EQ.1.AND.MT.EQ.-1.DO) THEN
C                          MT=0.DO
C                      ELSE IF (TYPE2.EQ.3.AND.MT.EQ.-1.DO) THEN

```



```

40      MT=1.D0
      END IF
      SUM1=SUM1+(MT*(10.D0**KM1))
      KM1=KM1-1
      CONTINUE

      *** CONVERT THE REPRESENTATIVE BINARY VARIABLE TO ITS DECIMAL
      EQUIVALENT ***

      SUM2=0.D0
      KM1=K-1
      ND2=NDP/2.D0
      DO 50 J=1,K
        IF (SUM1/10.D0**KM1.GE.1.D0) THEN
          SUM2=SUM2+ND2
          SUM1=SUM1-10.D0**KM1
          ND2=ND2/2.D0
          KM1=KM1-1
        ELSE
          ND2=ND2/2.D0
          KM1=KM1-1
        END IF
      CONTINUE
      C=1
      FLAG=1
      END IF
      CALL ORTHO(K,MT,SUM2,C,FLAG)
      FLAG=2
      END IF
      CONTINUE

30      *** PLOT THE TIME SERIES IF DESIRED ***

      IF (REP.EQ.1) THEN
        MINY=-MAXY
        CALL PLOT(MAXY,MINY,STEP,IN)
      END IF

      *** GENERATE THE FFT ***

      CALL FFTC(ARRAY(1,1),IN,X,IWK)

      *** CALCULATE THE AMPLITUDE SPECTRUM OF THE FUNCTION ***
      *** FIND THE NUMBER AND VALUE OF THE PRINCIPLE HARMONIC ***

      N=0.D0
      R=1
      MAXY=0.D0
      DO 60 I=1,IND2P1

```

```

OBP01610
OBP01620
OBP01630
OBP01640
OBP01650
OBP01660
OBP01670
OBP01680
OBP01690
OBP01700
OBP01710
OBP01720
OBP01730
OBP01740
OBP01750
OBP01760
OBP01770
OBP01780
OBP01790
OBP01800
OBP01810
OBP01820
OBP01830
OBP01840
OBP01850
OBP01860
OBP01870
OBP01880
OBP01890
OBP01900
OBP01910
OBP01920
OBP01930
OBP01940
OBP01950
OBP01960
OBP01970
OBP01980
OBP01990
OBP02000
OBP02010
OBP02020
OBP02030
OBP02040
OBP02050
OBP02060
OBP02070
OBP02080
OBP02090
OBP02100

```

```

60      ARRAY(R,1)=CDABS(X(R))
C      ARRAY(R,2)=N/STEP
C      IF(ARRAY(R,1).GT.MAXY) THEN
C          MAXY=ARRAY(R,1)
C          RMAX=R-1
C      END IF
C      R=R+1
C      N=N+1.DO
C      CONTINUE
C
C      *** DISPLAY INFO IF THE FIRST TIME THROUGH THE SUBPROGRAM ***
C
C      IF(REP.EQ.1) THEN
C          WRITE(10,70)RMAX
C          FORMAT(10,70)RMAX
C          * IS THE 15. HARMONIC. THE NEXT PLOT TO BE PRODUCED WILL.
C          * BE THE AMPLITUDE SPECTRUM. ENTER A 1 IF YOU ARE READY TO.
C          ** CONTINUE WITH THE PROGRAM.
C      READ(5,*)L
C
C      IF(L.NE.1) THEN
C          CALL FRTCMS('CLRSCRN ')
C          WRITE(10,80)
C          FORMAT(10,80)
C          GO TO 69
C      END IF
C
C      END IF
C
C      *** ALLOW FOR THE PLOT OF THE AMPLITUDE SPECTRUM OF THE FFT ***
C
C      IF(REP.EQ.1) THEN
C          MINY=0.DO
C          INT=1.DO/STEP
C          CALL PLOT(MAXY,MINY,INT,IND2P1)
C      END IF
C
C      *** ADD THE VALUES OF THE FFT TO THE ACCUMULATED STATISTICS ***
C
C      FLAG=0
C      CALL STATS(IN,REP,FLAG)
C
C      RETURN
C      END

```

```

SUBROUTINE QPSK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,
*ANS2,DSEED,IN,REP)
** PURPOSE ***
THIS SUBROUTINE MODULATES THE CARRIER USING QUADRATURE PHASE SHIFT
KEYING AS THE MODULATION TECHNIQUE.
** PARAMETER DEFINITIONS ***
TYPE2= INDICATES LOGIC TYPE TO BE EMPLOYED
BAUD= SYMBOL RATE OR BAUD RATE
BITS= NUMBER OF BITS IN EACH BINARY CODE WORD
FREQ= CARRIER FREQUENCY
IPHAS= INITIAL PHASE ANGLE IN DEGREES
AMP= AMPLITUDE
ANS2= INTEGER VARIABLE TO BE PASSED TO STREAM GENERATOR
DSEED= DOUBLE PRECISION SEED FOR RANDOM NUMBER GENERATOR
IN= NUMBER OF POSITIONS IN ARRAY TO BE UTILIZED
REP= AN INTEGER EQUAL TO THE NUMBER OF THE REPETITION OF
THE CALL THE THIS SUBROUTINE
** VARIABLE DEFINITIONS **
ARRAY= A DOUBLE PRECISION ARRAY FOR PASSING THE BINARY
DIGITS AND STORING THE VALUE OF THE TIME FUNCTION
AND THE FFT USED TO REPRESENT THE ROW OF ARRAY
R= AN INTEGER USED TO REPRESENT THE ROW OF ARRAY
TIME= TIME VARIABLE
MT= VALUE OF BINARY DIGIT PASSED FROM SUBPROGRAM STREAM
MT1= VALUE OF IN-PHASE BINARY DIGIT
MT2= VALUE OF QUADRATURE BINARY DIGIT
STEP= INTERVAL AT WHICH THE SIGNAL IS REPRODUCED
PI= NUMERICAL CONSTANT
OMEGA= CARRIER ANGULAR FREQUENCY
DELTA= INITIAL PHASE OFFSET IN RADIAN
NBAUD= A COUNT OF THE NUMBER OF BAUDS MODULATED
BAUDD= BAUD DURATION
X= COMPLEX ARRAY TO RECEIVE THE FFT OF THE TIME SERIES
IWK= INTEGER WORKING TO ARRAY USED BY FUNCTION FFTRC
MAXY= MAXIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
MINY= MINIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
INT= INTERVAL BETWEEN POINTS ON THE ORDINATE
RMAX= VALUE OF THE PRINCIPLE HARMONIC
FLAG= AN INTEGER WHICH CONTROL ENTRANCE AND EXIT FROM
SUBPROGRAM STATS
IND2P1= LENGTH OF ARRAY DIVIDED BY 2 + 1
** VARIABLE DECLARATIONS ***

```

QPS00110
QPS00120
QPS00130
QPS00140
QPS00150
QPS00160
QPS00170
QPS00180
QPS00190
QPS00200
QPS00210
QPS00220
QPS00230
QPS00240
QPS00250
QPS00260
QPS00270
QPS00280
QPS00290
QPS00300
QPS00310
QPS00320
QPS00330
QPS00340
QPS00350
QPS00360
QPS00370
QPS00380
QPS00390
QPS00400
QPS00410
QPS00420
QPS00430
QPS00440
QPS00450
QPS00460
QPS00470
QPS00480
QPS00490
QPS00500
QPS00510
QPS00520
QPS00530
QPS00540
QPS00550
QPS00560
QPS00570
QPS00580
QPS00590
QPS00600

QPS01610
QPS01620
QPS01630
QPS01640
QPS01650
QPS01660
QPS01670
QPS01680
QPS01690
QPS01700
QPS01710
QPS01720
QPS01730
QPS01740
QPS01750
QPS01760
QPS01770
QPS01780
QPS01790

```

SUBROUTINE OQPSK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,
*ANS2,DSEED,IN,REP)
** PURPOSE **
THIS SUBROUTINE MODULATES THE CARRIER USING OFFSET QUADRATURE
PHASE SHIFT KEYING AS THE MODULATION TECHNIQUE.
** PARAMETER DEFINITIONS **
TYPE2= INDICATES LOGIC TYPE TO BE EMPLOYED
BAUD= SYMBOL RATE OR BAUD RATE
BITS= NUMBER OF BITS IN EACH BINARY CODE WORD
FREQ= CARRIER FREQUENCY
IPHAS= INITIAL PHASE ANGLE IN DEGREES
AMP= AMPLITUDE
ANS2= INTEGER VARIABLE TO BE PASSED TO STREAM GENERATOR
DSEED= DOUBLE PRECISION SEED FOR RANDOM NUMBER GENERATOR
IN= NUMBER OF POSITIONS IN ARRAY TO BE UTILIZED
REP= AN INTEGER EQUAL TO THE NUMBER OF THE REPETITION OF
THE CALL THE THIS SUBROUTINE
** VARIABLE DEFINITIONS **
AKRAY= A DOUBLE PRECISION ARRAY FOR PASSING THE BINARY
DIGITS AND STORING THE VALUE OF THE TIME FUNCTION
AND THE FFT
AN INTEGER USED TO REPRESENT THE ROW OF ARRAY
TIME VARIABLE
VALUE OF BINARY DIGIT PASSED FROM SUBPROGRAM STREAM
VALUE OF IN-PHASE BINARY DIGIT
VALUE OF QUADRATURE BINARY DIGIT
INTERVAL AT WHICH THE SIGNAL IS REPRODUCED
NUMERICAL CONSTANT
CARRIER ANGULAR FREQUENCY
INITIAL PHASE OFFSET IN RADIAN
A COUNT OF THE NUMBER OF BAUDS MODULATED IN-PHASE
A COUNT OF THE NUMBER OF BAUDS MODULATED QUADRATURE
COMPLEX ARRAY TO RECEIVE THE FFT OF THE TIME SERIES
INTEGER WORKING ARRAY USED BY FUNCTION FFTRC
MAXIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
MINIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
INTERVAL BETWEEN POINTS ON THE ORDNATE
VALUE OF THE PRINCIPLE HARMONIC
AN INTEGER WHICH CONTROL ENTRANCE AND EXIT FROM
SUBPROGRAM STAYS
LENGTH OF ARRAY DIVIDED BY 2 + 1
** VARIABLE DECLARATIONS **

```

```

OQP00110
OQP00120
OQP00130
OQP00140
OQP00150
OQP00160
OQP00170
OQP00180
OQP00190
OQP00200
OQP00210
OQP00220
OQP00230
OQP00240
OQP00250
OQP00260
OQP00270
OQP00280
OQP00290
OQP00300
OQP00310
OQP00320
OQP00330
OQP00340
OQP00350
OQP00360
OQP00370
OQP00380
OQP00390
OQP00400
OQP00410
OQP00420
OQP00430
OQP00440
OQP00450
OQP00460
OQP00470
OQP00480
OQP00490
OQP00500
OQP00510
OQP00520
OQP00530
OQP00540
OQP00550
OQP00560
OQP00570
OQP00580
OQP00590
OQP00600

```

```

C      INTEGER R,ANS2,TYPE2,IN,EMAX,REP,FLAG,IND2P1
C      DOUBLE PRECISION BAUD,BITS,FREQ,IPHAS,AMP,TIME,MT1,MT2,
C      *STEP,PI,OMEGA,DELTA,BAUDD,NBAUD1,NBAUD2,
C      *DSEED,MAY,MINY,INT,MT
C      DOUBLE PRECISION ARRAY(1024,2),SUMX(513),SUMXSQ(513),SUMX3(513),
C      *SUMX4(513)
C      COMMON ARRAY,SUMX,SUMXSQ,SUMX3,SUMX4
C      COMPLEX*16 X(513)
C      INTEGER INK(10)
C      *** VARIABLE INITIALIZATION ***
C      R=1
C      STEP=1.D0/(2.D0*(FREQ+BAUD))
C      PI=3.141592653589793D0
C      OMEGA=2.D0*PI*FREQ
C      DELTA=IPHAS*PI/180.D0
C      NBAUD1=1.D0
C      NBAUD2=1.D0
C      BAUDD=1.D0/BAUD
C      TIME=.5D0*BAUDD
C      MAY=0.D0
C      IND2P1=IN/2+1
C      *** DO THE MODULATION AND ASSIGN THE VALUES TO ARRAY ***
C      CALL STREAM(DSEED,ANS2,TYPE2,MT)
C      MT1=MT
C      CALL STREAM(DSEED,ANS2,TYPE2,MT)
C      MT2=MT
C      DO 10 I=1,IN
C      *      ARRAY(R,1)=AMP*MT1*DCOS((OMEGA*TIME)+DELTA)+
C      *      AMP*MT2*DSIN((OMEGA*TIME)+DELTA)
C      *      ARRAY(R,2)=TIME
C      *      IF(ARRAY(R,1).GT.MAY) THEN
C      *      *      END IF
C      *      R=R+1
C      *      TIME=TIME+STEP
C      *      IF(TIME.GT.NBAUD1*BAUDD) THEN
C      *      *      CALL STREAM(DSEED,ANS2,TYPE2,MT)
C      *      *      MT1=MT
C      *      *      NBAUD1=NBAUD1+1.D0
C      *      *      END IF

```



```

C      CALL FRICMS('CLRSCRN ')
C      WRITE(10,40)
C      FORMAT(' ERROR',/)
C      GO TO 29
C      END IF
C      END IF
C      *** ALLOW FOR THE PLOT OF THE AMPLITUDE SPECTRUM OF THE FFT ***
C      IF (REP.EQ.1) THEN
C      MINY=0.00
C      INT=1.00/STEP
C      CALL PLOT(MAXY,MINY,INT,IND2P1)
C      END IF
C      *** ADD THE VALUES OF THE FFT TO THE ACCUMULATED STATISTICS ***
C      FLAG=0
C      CALL STATS(IN,REP,FLAG)
C      RETURN
C      END
OQP 01610
OQP 01620
OQP 01630
OQP 01640
OQP 01650
OQP 01660
OQP 01670
OQP 01680
OQP 01690
OQP 01700
OQP 01710
OQP 01720
OQP 01730
OQP 01740
OQP 01750
OQP 01760
OQP 01770
OQP 01780
OQP 01790
OQP 01800
OQP 01810
OQP 01820
OQP 01830
OQP 01840

```

```

SUBROUTINE MPSK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,ANS2,
*DSEED,IN,REP)
*** PURPOSE ***
THIS SUBROUTINE MODULATES THE CARRIER USING M-ARY PHASE SHIFT
KEYING AS THE MODULATION TECHNIQUE.
*** PARAMETER DEFINITIONS ***
TYPE2= INDICATES LOGIC TYPE TO BE EMPLOYED
BAUD= SYMBOL RATE OR BAUD RATE
BITS= NUMBER OF BITS IN EACH BINARY CODE WORD
FREQ= CARRIER FREQUENCY
IPHAS= INITIAL PHASE ANGLE IN DEGREES
AMP= AMPLITUDE
ANS2= INTEGER VARIABLE TO BE PASSED TO STREAM GENERATOR
DSEED= DOUBLE PRECISION SEED FOR RANDOM NUMBER
IN= NUMBER OF POSITIONS IN ARRAY TO BE UTILIZED
REP= AN INTEGER EQUAL TO THE NUMBER OF THE REPETITION OF
THE CALL THE THIS SUBROUTINE
*** VARIABLE DEFINITIONS ***
ARRAY= A DOUBLE PRECISION ARRAY FOR PASSING THE BINARY
DIGITS AND STORING THE VALUE OF THE TIME FUNCTION
AND THE FFT USED TO REPRESENT THE ROW OF ARRAY
R= TIME VARIABLE
TIME= VALUE OF THE BINARY DIGIT
MT= INTERVAL AT WHICH THE SIGNAL IS REPRODUCED
STEP= NUMERICAL ANGULAR FREQUENCY
PI= CARRIER ANGLE OFFSET IN RADIANS
OMEGA= INITIAL PHASE OFSET IN RADIANS
DELTA= A COUNT OF THE NUMBER OF BAUDS MODULATED
NBAUD= BAUD DURATION
NBAUDD= COMPLEX ARRAY TO RECEIVE THE FFT OF THE TIME SERIES
X= INTEGER WORKING ARRAY USED BY FUNCTION FFTC
INWK= MAXIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
MAY= MINIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
NINT= INTERVAL BETWEEN POINTS ON THE ORDINATE
KMAX= VALUE OF THE PRINCIPLE HARMONIC
K= INTEGER VALUE OF THE NUMBER OF BITS IN A BINARY WORD
KM1= INTEGER VALUE OF K MINUS 1
N= NUMBER OF POSSIBLE BINARY CODE WORDS = 2**K
ND2= N DIVIDED BY 2
SUM1= VARIABLE USED TO HOLD THE DECIMAL EQUIVALENT TO A
BINARY CODE WORD OF LENGTH K
SUM2= VARIABLE USED TO HOLD THE DECIMAL CONVERSION OF SUM1
J= INTEGER USED TO INDEX A LOOP

```

```

MPS00110
MPS00120
MPS00130
MPS00140
MPS00150
MPS00160
MPS00170
MPS00180
MPS00190
MPS00200
MPS00210
MPS00220
MPS00230
MPS00240
MPS00250
MPS00260
MPS00270
MPS00280
MPS00290
MPS00300
MPS00310
MPS00320
MPS00330
MPS00340
MPS00350
MPS00360
MPS00370
MPS00380
MPS00390
MPS00400
MPS00410
MPS00420
MPS00430
MPS00440
MPS00450
MPS00460
MPS00470
MPS00480
MPS00490
MPS00500
MPS00510
MPS00520
MPS00530
MPS00540
MPS00550
MPS00560
MPS00570
MPS00580
MPS00590
MPS00600

```


MPS01110
MPS01120
MPS01130
MPS01140
MPS01150
MPS01160
MPS01170
MPS01180
MPS01190
MPS01200
MPS01210
MPS01220
MPS01230
MPS01240
MPS01250
MPS01260
MPS01270
MPS01280
MPS01290
MPS01300
MPS01310
MPS01320
MPS01330
MPS01340
MPS01350
MPS01360
MPS01370
MPS01380
MPS01390
MPS01400
MPS01410
MPS01420
MPS01430
MPS01440
MPS01450
MPS01460
MPS01470
MPS01480
MPS01490
MPS01500
MPS01510
MPS01520
MPS01530
MPS01540
MPS01550
MPS01560
MPS01570
MPS01580
MPS01590
MPS01600

```

10      KM1=KM1-1
      CONTINUE
      *** CONVERT THE REPRESENTATIVE BINARY VARIABLE TO ITS DECIMAL
      EQUIVALENT ***
      SUM2=0; DO
      KM1=K-1
      ND2=N/2; DO
      DO 20 I=1, K
      IF (SUM1/10. DO**KM1.GE. 1. DO) THEN
      SUM2=SUM2+ND2
      SUM1=SUM1-10. DO**KM1
      ND2=ND2/2. DO
      KM1=KM1-1
      ELSE
      ND2=ND2/2. DO
      KM1=KM1-1
      END IF
      CONTINUE
      *** DO THE MODULATION AND ASSIGN TO ARRAY ***
      J=R
      DO 30 I=J, IN
      ARRAY(R, 1) =AMP*DCOS((OMEGA*TIME)+
      ((SUM2*2. DO*PI)/N)+DELTA)
      ARRAY(R, 2) =TIME
      IF (ARRAY(R, 1) .GT. MAXY) THEN
      MAXY=ARRAY(R, 1)
      END IF
      R=R+1
      TIME=TIME+STEP
      IF (TIME.GT. NBAUD*NBAUD) THEN
      NBAUD=NBAUD+1. DO
      GO TO 9
      END IF
      CONTINUE
      END IF
      *** PLOT THE TIME SERIES IF DESIRED ***
      IF (REP.EQ.1) THEN
      MINY=-MAXY
      CALL PLOT(MAXY, MINY, STEP, IN)
      END IF
      *** GENERATE THE FFT ***

```

10
C
C
C

20
C
C
C

30
C
C
C
C
C
C

MPS02110
MPS02120
MPS02130
MPS02140
MPS02150
MPS02160

FLAG=0
CALL STATS(IN,REP,FLAG)
RETURN
END

C

C

```

SUBROUTINE MASK (TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,ANS2,
*DSEED,IN,REP)
*** PURPOSE ***
THIS SUBROUTINE MODULATES THE CARRIER USING M-ARY AMPLITUDE SHIFT
KEYING AS THE MODULATION TECHNIQUE.
*** PARAMETER DEFINITIONS ***
TYPE2= INDICATES LOGIC TYPE TO BE EMPLOYED
BAUD= SYMBOL RATE OR BAUD RATE
BITS= NUMBER OF BITS IN EACH BINARY CODE WORD
FREQ= CARRIER FREQUENCY
IPHAS= INITIAL PHASE ANGLE IN DEGREES
AMP= AMPLITUDE
ANS2= INTEGER VARIABLE TO BE PASSED TO STREAM GENERATOR
DSEED= DOUBLE PRECISION SEED FOR RANDOM NUMBER GENERATOR
IN= NUMBER OF POSITIONS IN ARRAY TO BE UTILIZED
FLAG= THE CALL THE THIS SUBROUTINE
      AN INTEGER WHICH CONTROL ENTRANCE AND EXIT FROM
*** VARIABLE DEFINITIONS ***
ARRAY= A DOUBLE PRECISION ARRAY FOR PASSING THE BINARY
      DIGITS AND STORING THE VALUE OF THE TIME FUNCTION
      AND THE FFT
R= AN INTEGER USED TO REPRESENT THE ROW OF ARRAY
TIME= TIME VARIABLE
NT= VALUE OF THE BINARY DIGIT
STEP= INTERVAL AT WHICH THE SIGNAL IS REPRODUCED
PI= NUMERICAL CONSTANT
OMEGA= CARRIER ANGULAR FREQUENCY
DELTA= INITIAL PHASE OFFSET IN RADIAN
NBAUD= A COUNT OF THE NUMBER OF BAUDS MODULATED
BAUDD= BAUD DURATION
X= COMPLEX ARRAY TO RECEIVE THE FFT OF THE TIME SERIES
IWK= INTEGER WORKING ARRAY USED BY FUNCTION FFTRC
MAXY= MAXIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
MINY= MINIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
INT= INTERVAL BETWEEN POINTS ON THE ORDINATE
RMAX= VALUE OF THE PRINCIPLE HARMONIC
K= INTEGER VALUE OF THE NUMBER OF BITS IN A BINARY WORD
KM1= INTEGER VALUE OF K MINUS 1
N= NUMBER OF POSSIBLE BINARY CODE WORDS = 2**K
ND2= N DIVIDED BY 2
SUM1= VARIABLE USED TO HOLD THE DECIMAL EQUIVALENT TO A
      BINARY CODE WORD OF LENGTH K
SUM2= BINARY CODE WORD OF LENGTH K
      VARIABLE USED TO HOLD THE DECIMAL CONVERSION OF SUM1
J= INTEGER USED TO INDEX A LOOP

```

```

MAS00110
MAS00120
MAS00130
MAS00140
MAS00150
MAS00160
MAS00170
MAS00180
MAS00190
MAS00200
MAS00210
MAS00220
MAS00230
MAS00240
MAS00250
MAS00260
MAS00270
MAS00280
MAS00290
MAS00300
MAS00310
MAS00320
MAS00330
MAS00340
MAS00350
MAS00360
MAS00370
MAS00380
MAS00390
MAS00400
MAS00410
MAS00420
MAS00430
MAS00440
MAS00450
MAS00460
MAS00470
MAS00480
MAS00490
MAS00500
MAS00510
MAS00520
MAS00530
MAS00540
MAS00550
MAS00560
MAS00570
MAS00580
MAS00590
MAS00600

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


```

C C C C C
*** GENERATE THE PFT ***
CALL FFTRC (ARRAY(1,1),IN,X,IWK)

*** CALCULATE THE AMPLITUDE SPECTRUM OF THE FUNCTION ***
*** FIND THE NUMBER AND VALUE OF THE PRINCIPLE HARMONIC ***
N=0.DO
R=1
MAXY=0.DO
DO 40 I=1,IND2P1
  ARRAY(R,1)=CDABS(X(R))
  ARRAY(R,2)=N/STEP
  IF (ARRAY(R,1)-GT.MAXY) THEN
    MAXY=ARRAY(R,1)
    RMAX=R-1
  END IF
  R=R+1
  N=N+1.DO
CONTINUE

40
C C C
*** DISPLAY INFO IF THE FIRST TIME THROUGH THE SUBPROGRAM ***
IF (REP.EQ.1) THEN
  WRITE(10,50) RMAX
  FORMAT(10,50)
  * IS THE 15 HARMONIC. THE NEXT PLOT TO BE PRODUCED WILL.
  * BE THE AMPLITUDE SPECTRUM. ENTER A 1 IF YOU ARE READY TO.
  * CONTINUE WITH THE PROGRAM.
  READ(5,*) L
  IF (L.NE.1) THEN
    CALL FRTCMS('CLRSCRN ')
    WRITE(10,60)
    FORMAT(10,60)
    GO TO 49
  END IF
  END IF

60
C C C
*** ALLOW FOR THE PLOT OF THE AMPLITUDE SPECTRUM OF THE FFT ***
IF (REP.EQ.1) THEN
  MINY=0.DO
  INT=1.DO/STEP
  CALL PLCT(MAXY,MINY,INT,IND2P1)

```

```

MAS01610
MAS01620
MAS01630
MAS01640
MAS01650
MAS01660
MAS01670
MAS01680
MAS01690
MAS01700
MAS01710
MAS01720
MAS01730
MAS01740
MAS01750
MAS01760
MAS01770
MAS01780
MAS01790
MAS01800
MAS01810
MAS01820
MAS01830
MAS01840
MAS01850
MAS01860
MAS01870
MAS01880
MAS01890
MAS01900
MAS01910
MAS01920
MAS01930
MAS01940
MAS01950
MAS01960
MAS01970
MAS01980
MAS01990
MAS02000
MAS02010
MAS02020
MAS02030
MAS02040
MAS02050
MAS02060
MAS02070
MAS02080
MAS02090
MAS02100

```

MAS02110
MAS02120
MAS02130
MAS02140
MAS02150
MAS02160
MAS02170
MAS02180
MAS02190

END IF
*** ADD THE VALUES OF THE FFT TO THE ACCUMULATED STATISTICS ***
FLAG=0
CALL STATS(IN,REP,FLAG)
RETURN
END

C
C
C
C

```

SUBROUTINE QASK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,ANS2,
*DSEED,IN,REP)
*** PURPOSE ***
THIS SUBROUTINE MODULATES THE CARRIER USING QUADRATURE AMPLITUDE
SHIFT KEYING AS THE MODULATION TECHNIQUE.
*** PARAMETER DEFINITIONS ***
TYPE2= INDICATES LOGIC TYPE TO BE EMPLOYED
BAUD= SYMBOL RATE OR BAUD RATE
BITS= NUMBER OF BITS IN EACH BINARY CODE WORD
FREQ= CARRIER FREQUENCY
IPHAS= INITIAL PHASE ANGLE IN DEGREES
AMP= AMPLITUDE
ANS2= INTEGER VARIABLE TO BE PASSED TO STREAM
DSEED= DOUBLE PRECISION SEED FOR RANDOM NUMBER GENERATOR
IN= NUMBER OF POSITIONS IN ARRAY TO BE UTILIZED
FLAG= THE CALL THE THIS SUBROUTINE
      AN INTEGER WHICH CONTROL ENTRANCE AND EXIT FROM
*** VARIABLE DEFINITIONS ***
ARRAY= A DOUBLE PRECISION ARRAY FOR PASSING THE BINARY
      DIGITS AND STORING THE VALUE OF THE TIME FUNCTION
      AND THE FFT
      AN INTEGER USED TO REPRESENT THE ROW OF ARRAY
      TIME VARIABLE
      VALUE OF THE BINARY DIGIT
      INTERVAL AT WHICH THE SIGNAL IS REPRODUCED
      NUMERICAL ANGLE CONSTANT
      CARRIER ANGULAR FREQUENCY
      INITIAL PHASE OFFSET IN RADIAN
      A COUNT OF THE NUMBER OF BAUDS MODULATED
      BAUD DURATION
      COMPLEX ARRAY TO RECEIVE THE FFT OF THE TIME SERIES
      INTEGER WORKING ARRAY USED BY FUNCTION FFTRC
      MAXIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
      MINIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
      INTERVAL BETWEEN POINTS ON THE ORDINATE
      VALUE OF THE PRINCIPLE HARMONIC
      INTEGER VALUE OF THE NUMBER OF BITS IN A BINARY WORD
      INTEGER VALUE OF K MINUS 1
      NUMBER OF POSSIBLE BINARY CODE WORDS = 2**K
      N DIVIDED BY 2
      VARIABLE USED TO HOLD THE DECIMAL EQUIVALENT TO A
      BINARY CODE WORD OF LENGTH K
      VARIABLE USED TO HOLD THE DECIMAL CONVERSION OF SUM1
      INTEGER USED TO INDEX A LOOP

```

QAS00110
QAS00120
QAS00130
QAS00140
QAS00150
QAS00160
QAS00170
QAS00180
QAS00190
QAS00200
QAS00210
QAS00220
QAS00230
QAS00240
QAS00250
QAS00260
QAS00270
QAS00280
QAS00290
QAS00300
QAS00310
QAS00320
QAS00330
QAS00340
QAS00350
QAS00360
QAS00370
QAS00380
QAS00390
QAS00400
QAS00410
QAS00420
QAS00430
QAS00440
QAS00450
QAS00460
QAS00470
QAS00480
QAS00490
QAS00500
QAS00510
QAS00520
QAS00530
QAS00540
QAS00550
QAS00560
QAS00570
QAS00580
QAS00590
QAS00600

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

QAS01110
QAS01120
QAS01130
QAS01140
QAS01150
QAS01160
QAS01170
QAS01180
QAS01190
QAS01200
QAS01210
QAS01220
QAS01230
QAS01240
QAS01250
QAS01260
QAS01270
QAS01280
QAS01290
QAS01300
QAS01310
QAS01320
QAS01330
QAS01340
QAS01350
QAS01360
QAS01370
QAS01380
QAS01390
QAS01400
QAS01410
QAS01420
QAS01430
QAS01440
QAS01450
QAS01460
QAS01470
QAS01480
QAS01490
QAS01500
QAS01510
QAS01520
QAS01530
QAS01540
QAS01550
QAS01560
QAS01570
QAS01580
QAS01590
QAS01600

```

SUM1=0.D0
KM1=K-1
DO 10 I=1,K
  CALL STREAM(DSEED,ANS2,TYPE2,MT)
  IF(TYPE2.EQ.1.AND.MT.EQ.-1.D0) THEN
    MT=0.D0
  ELSE IF(TYPE2.EQ.3.AND.MT.EQ.-1.D0) THEN
    MT=1.D0
  END IF
  SUM1=SUM1+(MT*(10.D0**KM1))
  KM1=KM1-1
CONTINUE

```

*** CONVERT THE REPRESENTATIVE BINARY VARIABLE TO ITS DECIMAL EQUIVALENT ***

```

SUM2=0.D0
KM1=K-1
ND2=N/2.D0
DO 20 I=1,K
  IF(SUM1/10.D0**KM1.GE.1.D0) THEN
    SUM2=SUM2+ND2
    SUM1=SUM1-10.D0**KM1
    ND2=ND2/2.D0
    KM1=KM1-1
  ELSE
    ND2=ND2/2.D0
    KM1=KM1-1
  END IF
CONTINUE

```

```

IF(M.EQ.1) THEN
  SUM2A=SUM2+1.D0
ELSE IF(M.EQ.2) THEN
  SUM2B=SUM2+1.D0
END IF

```

CONTINUE

*** DO THE MODULATION AND ASSIGN TO ARRAY ***

```

AMPA=AMP/SUM2A
AMPB=AMP/SUM2B
J=R
DO 30 I=J,IN
  ARRAY(R,1)=AMPA*DCOS((OMEGA*TIME)+DELTA)+
    AMPB*DSIN((OMEGA*TIME)+DELTA)
  ARRAY(R,2)=TIME
  IF(ARRAY(R,1).GT.MAXY) THEN

```

*


```

C      READ(5,*)L
C      IF(L.NE.1)THEN
C          CALL FRTCHS('CLRSCRN ')
C          WRITE(10,60)
C          FORMAT(' ERROR',/)
C          GO TO 49
C      END IF
C      END IF
C      *** ALLOW FOR THE PLOT OF THE AMPLITUDE SPECTRUM OF THE FFT ***
C      IF (REP.EQ.1) THEN
C          MINY=0.D0
C          INT=1.D0/STEP
C          CALL PLOT(MAXY,MINY,INT,IND2P1)
C      END IF
C      *** ADD THE VALUES OF THE FFT TO THE ACCUMULATED STATISTICS ***
C      FLAG=0
C      CALL STATS(IN,REP,FLAG)
C      RETURN
C      END

```

```

OAS02110
OAS02120
OAS02130
OAS02140
OAS02150
OAS02160
OAS02170
OAS02180
OAS02190
OAS02200
OAS02210
OAS02220
OAS02230
OAS02240
OAS02250
OAS02260
OAS02270
OAS02280
OAS02290
OAS02300
OAS02310
OAS02320
OAS02330
OAS02340
OAS02350
OAS02360
OAS02370

```

```

SUBROUTINE MSK(TYPE2,BAUD,FREQ,IPHAS,ANP,ANS2,
*DSEED,IN,REP)
*** PURPOSE ***
THIS SUBROUTINE MODULATES THE CARRIER USING MINIMUM SHIFT KEYING
OR FAST FREQUENCY SHIFT KEYING AS THE MODULATION TECHNIQUE.
*** PARAMETER DEFINITIONS ***
TYPE2= INDICATES LOGIC TYPE TO BE EMPLOYED
BAUD= SYMBOL RATE OR BAUD RATE
FREQ= CARRIER FREQUENCY
IPHAS= INITIAL PHASE ANGLE IN DEGREES
ANP= AMPLITUDE
ANS2= INTEGER VARIABLE TO BE PASSED TO STREAM
DSEED= DOUBLE PRECISION SEED FOR RANDOM NUMBER GENERATOR
IN= NUMBER OF POSITIONS IN ARRAY TO BE UTILIZED
REP= AN INTEGER EQUAL TO THE NUMBER OF THE REPETITION OF
THE CALL THE THIS SUBROUTINE
*** VARIABLE DEFINITIONS ***
ARRAY= A DOUBLE PRECISION ARRAY FOR PASSING THE BINARY
DIGITS AND STORING THE VALUE OF THE TIME FUNCTION
AND THE FFT
K= AN INTEGER USED TO REPRESENT THE ROW OF ARRAY
TIME= TIME VARIABLE
MT= VALUE OF THE BINARY DIGIT
STEP= INTERVAL AT WHICH THE SIGNAL IS REPRODUCED
PI= NUMERICAL CONSTANT
OMEGA= CARRIER ANGULAR FREQUENCY
DELTA= INITIAL PHASE OFFSET IN RADIANS
NBAUD= A COUNT OF THE NUMBER OF BAUDS MODULATED
BAUDD= BAUD DURATION
X= COMPLEX ARRAY TO RECEIVE THE FFT OF THE TIME SERIES
INWK= INTEGER WORKING ARRAY USED BY FUNCTION FFTC
MAXY= MAXIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
MINY= MINIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
INT= INTERVAL BETWEEN POINTS ON THE ORDINATE
KMAX= VALUE OF THE PRINCIPLE HARMONIC
FLAG= AN INTEGER WHICH CONTROL ENTRANCE AND EXIT FROM
SUBPROGRAM STATS
IND2P1= LENGTH OF ARRAY DIVIDED BY 2 + 1
*** VARIABLE DECLARATIONS ***
INTEGER R,ANS2,TYPE2,IN,KMAX,REP,FLAG,IND2P1
DOUBLE PRECISION BAUD,FREQ,IPHAS,ANP,TIME,MT,

```

```

MSK000110
MSK000120
MSK000130
MSK000140
MSK000150
MSK000160
MSK000170
MSK000180
MSK000190
MSK000200
MSK000210
MSK000220
MSK000230
MSK000240
MSK000250
MSK000260
MSK000270
MSK000280
MSK000290
MSK000300
MSK000310
MSK000320
MSK000330
MSK000340
MSK000350
MSK000360
MSK000370
MSK000380
MSK000390
MSK000400
MSK000410
MSK000420
MSK000430
MSK000440
MSK000450
MSK000460
MSK000470
MSK000480
MSK000490
MSK000500
MSK000510
MSK000520
MSK000530
MSK000540
MSK000550
MSK000560
MSK000570
MSK000580
MSK000590
MSK000600

```

```

C
*STEP,PI,OMEGA,DELTA,NBAUD,BAUDD,
*DSEED,MXXY,MINY,INT
C
DOUBLE PRECISION ARRAY(1024,2),SUMX(513),SUMXSQ(513),SUMX3(513),
SUMX4(513)
COMMON ARRAY,SUMX,SUMXSQ,SUMX3,SUMX4
C
COMPLEX*16 X(513)
C
INTEGER IWK(10)
C
*** VARIABLE INITIALIZATION ***
C
R=1
TIME=0.D0
STEP=1.D0/(2.D0*(FREQ+(1.25D0*BAUD)))
PI=3.141592653589793D0
OMEGA=2.D0*PI*FREQ
DELTA=IPHAS*PI/180.D0
NBAUD=1.D0
BAUDD=1.D0/BAUD
MXXY=0.D0
IND2P1=IN/2+1
C
CALL STREAM(DSEED,ANS2,TYPE2,MT)
C
DO 10 I=1,IN
  ARRAY(R,1)=AMP*DCOS(((OMEGA+((MT*PI)/(2.D0*BAUDD)))*TIME)
  +DELTA)
  ARRAY(R,2)=TIME
  IF(DABS(ARRAY(R,1)).GT.MXXY) THEN
    MXXY=DABS(ARRAY(R,1))
  END IF
  R=R+1
  TIME=TIME+STEP
  IF(TIME.GT.NBAUD*BAUDD) THEN
    CALL STREAM(DSEED,ANS2,TYPE2,MT)
    NBAUD=NBAUD+1.D0
  END IF
CONTINUE
C
*** PLOT THE TIME SERIES IF DESIRED ***
C
IF(REP.EQ.1) THEN
  MINY=-MXXY
  CALL PLOT(MXXY,MINY,STEP,IN)
END IF
C
*** GENERATE THE FFT ***

```

```

MSK00610
MSK00620
MSK00630
MSK00640
MSK00650
MSK00660
MSK00670
MSK00680
MSK00690
MSK00700
MSK00710
MSK00720
MSK00730
MSK00740
MSK00750
MSK00760
MSK00770
MSK00780
MSK00790
MSK00800
MSK00810
MSK00820
MSK00830
MSK00840
MSK00850
MSK00860
MSK00870
MSK00880
MSK00890
MSK00900
MSK00910
MSK00920
MSK00930
MSK00940
MSK00950
MSK00960
MSK00970
MSK00980
MSK00990
MSK01000
MSK01010
MSK01020
MSK01030
MSK01040
MSK01050
MSK01060
MSK01070
MSK01080
MSK01090
MSK01100

```


MSK01610
MSK01620
MSK01630
MSK01640
MSK01650
MSK01660
MSK01670

*** ADD THE VALUES OF THE PFT TO THE ACCUMULATED STATISTICS ***

FLAG=0

CALL STATS(IN,REP,FLAG)

RETURN

END

C

C

C

```

SUBROUTINE MFSK(TYPE2,BAUD,BITS,FREQ,IPHAS,AMP,ANS2,
*DSEED,IN,REP)
*** PURPOSE ***
THIS SUBROUTINE MODULATES THE CARRIER USING M-ARY FREQUENCY SHIFT
KEYING AS THE MODULATION TECHNIQUE.
*** PARAMETER DEFINITIONS ***
TYPE2= INDICATES LOGIC TYPE TO BE EMPLOYED
BAUD= SYMBOL RATE OR BAUD RATE
BITS= NUMBER OF BITS IN EACH BINARY CODE WORD
FREQ= CARRIER FREQUENCY
IPHAS= INITIAL PHASE ANGLE IN DEGREES
AMP= AMPLITUDE
ANS2= INTEGER VARIABLE TO BE PASSED TO STREAM
DSEED= DOUBLE PRECISION SEED FOR RANDOM NUMBER GENERATOR
IN= NUMBER OF POSITIONS IN ARRAY TO BE UTILIZED
REP= AN INTEGER EQUAL TO THE NUMBER OF THE REPETITION OF
THE CALL THE THIS SUBROUTINE
*** VARIABLE DEFINITIONS ***
ARRAY= A DOUBLE PRECISION ARRAY FOR PASSING THE BINARY
DIGITS AND STORING THE VALUE OF THE TIME FUNCTION
AND THE FFT
R= AN INTEGER USED TO REPRESENT THE ROW OF ARRAY
TIME= TIME VARIABLE
MT= VALUE OF THE BINARY DIGIT
STEP= INTERVAL AT WHICH THE SIGNAL IS REPRODUCED
PI= NUMERICAL CONSTANT
OMEGA= CARRIER ANGULAR FREQUENCY
DELTA= INITIAL PHASE OFFSET IN RADIAN
NBAUD= A COUNT OF THE NUMBER OF BAUDS MODULATED
BAUDD= BAUD DURATION
X= COMPLEX ARRAY TO RECEIVE THE FFT OF THE TIME SERIES
IWK= INTEGER WORKING ARRAY USED BY FUNCTION FFTC
MAXY= MAXIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
MINY= MINIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
INT= INTERVAL BETWEEN POINTS ON THE ORDINATE
RMAX= VALUE OF THE PRINCIPLE HARMONIC
K= INTEGER VALUE OF THE NUMBER OF BITS IN A BINARY WORD
KM1= INTEGER VALUE OF K MINUS 1
N= NUMBER OF POSSIBLE BINARY CODE WORDS = 2**K
ND2= N DIVIDED BY 2
SUM1= VARIABLE USED TO HOLD THE DECIMAL EQUIVALENT TO A
BINARY CODE WORD OF LENGTH K
SUM2= VARIABLE USED TO HOLD THE DECIMAL CONVERSION OF SUM1
J= INTEGER USED TO INDEX A LOOP

```

MFS00110
MFS00120
MFS00130
MFS00140
MFS00150
MFS00160
MFS00170
MFS00180
MFS00190
MFS00200
MFS00210
MFS00220
MFS00230
MFS00240
MFS00250
MFS00260
MFS00270
MFS00280
MFS00290
MFS00300
MFS00310
MFS00320
MFS00330
MFS00340
MFS00350
MFS00360
MFS00370
MFS00380
MFS00390
MFS00400
MFS00410
MFS00420
MFS00430
MFS00440
MFS00450
MFS00460
MFS00470
MFS00480
MFS00490
MFS00500
MFS00510
MFS00520
MFS00530
MFS00540
MFS00550
MFS00560
MFS00570
MFS00580
MFS00590
MFS00600

MFS01110
MFS01120
MFS01130
MFS01140
MFS01150
MFS01160
MFS01170
MFS01180
MFS01190
MFS01200
MFS01210
MFS01220
MFS01230
MFS01240
MFS01250
MFS01260
MFS01270
MFS01280
MFS01290
MFS01300
MFS01310
MFS01320
MFS01330
MFS01340
MFS01350
MFS01360
MFS01370
MFS01380
MFS01390
MFS01400
MFS01410
MFS01420
MFS01430
MFS01440
MFS01450
MFS01460
MFS01470
MFS01480
MFS01490
MFS01500
MFS01510
MFS01520
MFS01530
MFS01540
MFS01550
MFS01560
MFS01570
MFS01580
MFS01600

```

DO 10 I=1,K
CALL STREAM(DSEED,ANS2,TYPE2,MT)
IF (TYPE2.EQ.1.AND.MT.EQ.-1.D0) THEN
    MT=0.D0
ELSE IF (TYPE2.EQ.3.AND.MT.EQ.-1.D0) THEN
    MT=1.D0
END IF
SUM1=SUM1+(MT*(10.D0**KM1))
KM1=KM1-1
CONTINUE

*** CONVERT THE REPRESENTATIVE BINARY VARIABLE TO ITS DECIMAL
EQUIVALENT ***
SUM2=0.D0
KH1=K-1
ND2=N/2.D0
DO 20 I=1,K
    IF (SUM1/10.D0**KM1.GE.1.D0) THEN
        SUM2=SUM2+ND2
        SUM1=SUM1-10.D0**KM1
        NL2=ND2/2.D0
        KM1=KM1-1
    ELSE
        ND2=ND2/2.D0
        KM1=KM1-1
    END IF
END IF
CONTINUE

*** DO THE MODULATION AND ASSIGN TO ARRAY ***
J=R
MFREQ=-FRNGD2+(SUM2*DELF)
DO 30 I=J,IN
    ARRAY(R,1)=AMP*DCOS(((OMEGA+(2.D0*PI*MFREQ))*TIME)+
        DELTA)
    ARRAY(R,2)=TIME
    IF (ARRAY(R,1).GT.MAXY) THEN
        MAXY=ARRAY(R,1)
    END IF
    R=R+1
    TIME=TIME+STEP
    IF (TIME.GT.NBAUD*NBAUD) THEN
        NBAUD=NBAUD+1.D0
        GO TO 9
    END IF
CONTINUE
END IF

*** PLOT THE TIME SERIES IF DESIRED ***

```

10
C
C
C
C

20
C
C
C

30
C
C

MFS02110
MFS02120
MFS02130
MFS02140
MFS02150
MFS02160
MFS02170
MFS02180
MFS02190
MFS02200
MFS02210
MFS02220
MFS02230

```

C
C
C
C
      IF (REP.EQ.1) THEN
        MINY=0.D0
        INT=1.D0/STEP
        CALL PLOT(MAXY,MINY,INT,IND2P1)
      END IF
      FLAG=0
      CALL STATS(IN,REP,FLAG)
      RETURN
      END

```

```

SUBROUTINE QPRS(TYPE2,TYPE3,BAUD,BITS,FREQ,IPHAS,AMP,
*ANS2,DSSEED,IN,REP)
*** PURPOSE ***
THIS SUBROUTINE MODULATES THE CARRIER USING QUADRATURE PHASE SHIFT
KEYING AS THE MODULATION TECHNIQUE.
*** PARAMETER DEFINITIONS ***
TYPE2= INDICATES LOGIC TYPE TO BE EMPLOYED
TYPE3= INDICATES THE CLASS OF QPRS
BAUD= SYMBOL RATE OR BAUD RATE
BITS= NUMBER OF BITS IN EACH BINARY CODE WORD
FREQ= CARRIER FREQUENCY
IPHAS= INITIAL PHASE ANGLE IN DEGREES
AMP= AMPLITUDE
ANS2= INTEGER VARIABLE TO BE PASSED TO STREAM
DSSEED= DOUBLE PRECISION SEED FOR RANDOM NUMBER GENERATOR
IN= NUMBER OF POSITIONS IN ARRAY TO BE UTILIZED
REP= AN INTEGER EQUAL TO THE NUMBER OF THE REPETITION OF
THE CALL THE THIS SUBROUTINE
*** VARIABLE DEFINITIONS ***
ARFAY= A DOUBLE PRECISION ARRAY FOR PASSING THE BINARY
DIGITS AND STORING THE VALUE OF THE TIME FUNCTION
AND THE FFT
ASUBN= A DOUBLE PRECISION ARRAY CONTAINING THE VALUES OF
THE IN PHASE BINARY DIGIT TO BE MODULATED
BSUBN= A DOUBLE PRECISION ARRAY CONTAINING THE VALUES OF THE
QUADRATURE BINARY DIGIT TO BE MODULATED
HOFT= THE IMPULSE RESPONSE FOR THE SPECIFIED CLASS FILTER
T= VARIABLE WHICH DETERMINES THE MAGNITUDE OF THE IMPULSE
RESPONSE AT A GIVEN TIME FOR A SPECIFIED BINARY DIGIT
NDP= VARIABLE USED IN THE COMPUTATION OF T
R= AN INTEGER USED TO REPRESENT THE ROW OF ARRAY
TIME= TIME VARIABLE
MT= VALUE OF BINARY DIGIT PASSED FROM SUBPROGRAM STREAM
STEP= INTERVAL AT WHICH THE SIGNAL IS REPRODUCED
PI= NUMERICAL ANGLE OF CONSTANT FREQUENCY
OMEGA= CARRIER ANGULAR FREQUENCY
DELTA= INITIAL PHASE OFFSET IN RADIAN
BAUDD= BAUD DURATION AND QPRS BIT DURATION
X= COMPLEX WORKING ARRAY USED BY FUNCTION FFTRC
IPWK= INTEGER WORKING ARRAY PLOTTED ON THE ABCISSAE
MAXY= MAXIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
MINY= MINIMUM VALUE TO BE PLOTTED ON THE ABCISSAE
INT= INTERVAL BETWEEN POINTS ON THE ORDINATE
KMAX= VALUE OF THE PRINCIPLE HARMONIC

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

CCCCCCCCC C
FLAG=      AN INTEGER WHICH CONTROL ENTRANCE AND EXIT FROM
SUBPROGRAM STATS
IND2P1=    LENGTH OF ARRAY DIVIDED BY 2 + 1
MAX=       MAX NUMBER OF INCREMENTS IN 6 BAUD DURATIONS
INDEX=     INDEX OF A LOOP
*** VARIABLE DECLARATIONS ***
INTEGER R,ANS2,TYPE2,TYPE3,IN,RMAX,REP,FLAG,IND2P1,MAX,INDEX
DOUBLE PRECISION BAUD,BITS,FREQ,IPHAS,AMP,TIME,MT1,MT2,
*STEP,PI,OMEGA,DELTA,NBAUD,BAUDD,HOFT,T,NDP,
*DSEED,MXY,MINY,INT,MT
DOUBLE PRECISION ARRAY(1024,2),SUMX(513),SUMXSQ(513),SUMX3(513),
*SUMX4(513),ASUBN(2000),BSUBN(2000)
COMMON ARRAY,SUMX,SUMXSQ,SUMX3,SUMX4
COMPLEX*16 X(513)
INTEGER IWK(10)
*** VARIABLE INITIALIZATION ***
R=1
PI=3.141592653589793D0
BAUDD=1.D0/BAUD
TIME=6.D0/BAUD
STEP=1.D0/(2.D0*(FREQ+(PI+1.D0)*BAUD))
MAX=TIME/STEP+1.D0
INDEX=IN+MAX
OMEGA=2.D0*PI*FREQ
DELTA=IPHAS*PI/180.D0
MXY=0.D0
IND2P1=IN/2+1
DO 10 I=1,INDEX
  CALL STREAM(DSEED,ANS2,TYPE2,MT)
  ASUBN(I)=MT
  CALL STREAM(DSEED,ANS2,TYPE2,MT)
  BSUBN(I)=MT
CONTINUE
DO 20 I=1,1024
  ARRAY(I,1)=0.D0
CONTINUE
*** DO THE MODULATION AND ASSIGN THE VALUES TO ARRAY***
DO 30 I=1,IN

```

```

QPR00610
QPR00620
QPR00630
QPR00640
QPR00650
QPR00660
QPR00670
QPR00680
QPR00690
QPR00700
QPR00710
QPR00720
QPR00730
QPR00740
QPR00750
QPR00760
QPR00770
QPR00780
QPR00790
QPR00800
QPR00810
QPR00820
QPR00830
QPR00840
QPR00850
QPR00860
QPR00870
QPR00880
QPR00890
QPR00900
QPR00910
QPR00920
QPR00930
QPR00940
QPR00950
QPR00960
QPR00970
QPR00980
QPR00990
QPR01000
QPR01010
QPR01020
QPR01030
QPR01040
QPR01050
QPR01060
QPR01070
QPR01080
QPR01090
QPR01100

```

```

DO 40 J=1,INDEX
NDP=J-1
T=TIME-(NDP/BAUD)
IF(TYPE3.EQ.1) THEN
  IF(T.EQ.-BAUDD/2.D0.OR.T.EQ.BAUDD/2.D0) THEN
    HOFT=1.D0
  ELSE
    HOFT= {4.D0/(BAUD**2*PI)} * {DCOS(PI*T*BAUD) /
      {(1.D0/BAUD**2) - (4.D0*T**2)}}
  END IF
  ELSE IF (TYPE3.EQ.2) THEN
    IF(T.EQ.0.D0) THEN
      HOFT=2.D0
    ELSE IF (T.EQ.-BAUDD.OR.T.EQ.BAUDD) THEN
      HOFT=1.D0
    ELSE
      HOFT= {2.D0/(BAUD**3*PI*T)} * {DSIN(PI*T*BAUD) /
        {(1.D0/BAUD**2) - T**2}}
    END IF
  ELSE IF (TYPE3.EQ.3) THEN
    IF(T.EQ.0.D0) THEN
      HOFT=1.D0
    ELSE IF (T.EQ.-BAUDD) THEN
      HOFT=2.D0
    ELSE IF (T.EQ.BAUDD) THEN
      HOFT=-1.D0
    ELSE
      HOFT= {1.D0/(BAUD**2*PI*T)} * {DSIN(PI*T*BAUD) *
        {(3.D0*T-BAUDD) / (T**2 - {1.D0/BAUD**2})}}
    END IF
  ELSE IF (TYPE3.EQ.4) THEN
    IF(T.EQ.-BAUDD) THEN
      HOFT=1.D0
    ELSE IF (T.EQ.BAUDD) THEN
      HOFT=-1.D0
    ELSE
      HOFT= {2.D0/(BAUD**2*PI)} * {DSIN(PI*T*BAUD) /
        {T**2 - {1.D0/BAUD**2}}
    END IF
  ELSE IF (TYPE3.EQ.5) THEN
    IF(T.EQ.0.D0) THEN
      HOFT=-2.D0
    ELSE IF (T.EQ.-2.D0*BAUDD.OR.T.EQ.2.D0*BAUDD) THEN
      HOFT=1.D0
    ELSE
      HOFT= {8.D0/(BAUD**3*PI*T)} * {DSIN(PI*T*BAUD) /
        {T**2 - {4.D0/BAUD**2}}}
    END IF

```

```
C
END IF
**
ARRAY(I,1)=ARRAY(I,1)+((ASUBN(J)*HOPT*AMP*DCOS((OMEGA*
TIME)+DELTA))+ (BSUBN(J)*HOPT*AMP*DSIN((OMEGA*
TIME)+DELTA)))
C
IF(DABS(ARRAY(I,1)).GT.MAXY) THEN
MAXY=DABS(ARRAY(I,1))
END IF
CONTINUE
C
ARRAY(I,2)=TIME
TIME=TIME+STEP
C
CONTINUE
*** PLOT THE TIME SERIES IF DESIRED ***
IF(REP.EQ.1) THEN
MINY=-MAXY
CALL PLOT(MAXY,MINY,STEP,IN)
END IF
C
*** GENERATE THE FFT ***
CALL FFTRC(ARRAY(1,1),IN,X,INX)
C
*** CALCULATE THE AMPLITUDE SPECTRUM OF THE FUNCTION ***
*** FIND THE NUMBER AND VALUE OF THE PRINCIPLE HARMONIC ***
N=0.DO
R=1
MAXY=0.DO
DO 50 I=1,IND2P1
ARRAY(R,1)=CDABS(X(R))
ARRAY(R,2)=N/STEP
IF(ARRAY(R,1).GT.MAXY) THEN
MAXY=ARRAY(R,1)
RMAX=R-1
END IF
R=R+1
N=N+1.DO
CONTINUE
50 C
*** DISPLAY INFO IF THE FIRST TIME THROUGH THE SUBPROGRAM ***
IF(REP.EQ.1) THEN
```

```

59      WRITE(10,60)RMAX
60      FORMAT(' THE FFT HAS BEEN GENERATED! THE PRINCIPLE HARMONIC',//
*      ' IS THE',15,' HARMONIC. THE NEXT PLOT TO BE PRODUCED WILL',//
*      ' BE THE AMPLITUDE SPECTRUM. ENTER A 1 IF YOU ARE READY TO',//
*      ' CONTINUE WITH THE PROGRAM.',//,' :')
C      READ(5,*)L
C      IF(L.NE. 1) THEN
C          CALL FKTCMS('CLRSRN ')
70      WRITE(10,70)
C          FORMAT(' ERROR',//)
C          GO TO 59
C      END IF
C      END IF
C      *** ALLOW FOR THE PLOT OF THE AMPLITUDE SPECTRUM OF THE FFT ***
C      IF(REP.EQ.1) THEN
C          MINY=0.00
C          INT=1.00/STEP
C          CALL PLOT(MAXY,MINY,INT,IND2P1)
C      END IF
C      *** ADD THE VALUES OF THE FFT TO THE ACCUMULATED STATISTICS ***
C      FLAG=0
C      CALL STATS(IN,REP,FLAG)
C      RETURN
C      END

```

```

QPR02110
QPR02120
QPR02130
QPR02140
QPR02150
QPR02160
QPR02170
QPR02180
QPR02190
QPR02200
QPR02210
QPR02220
QPR02230
QPR02240
QPR02250
QPR02260
QPR02270
QPR02280
QPR02290
QPR02300
QPR02310
QPR02320
QPR02330
QPR02340
QPR02350
QPR02360
QPR02370
QPR02380
QPR02390
QPR02400
QPR02410
QPR02420
QPR02430

```

UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU U U

STR00610
STR00620
STR00630
STR00640

MT=INT
END IF
RETURN
END

```

SUBROUTINE ORTHO(K,MT,SUM2,C,FLAG)
*** PURPOSE ***
THIS SUBROUTINE GENERATES AN ORTHOGONAL SET OF N VECTORS OF LENGTH
N, WHERE N IS 2**K, WHERE K IS THE LENGTH OF THE BINARY CODE WORD.
AND DETERMINES THE VALUE, MT, OF A BINARY DIGIT TO BE RETURNED
*** PARAMETER DEFINITIONS ***
K=      NUMBER OF BITS IN THE BINARY CODE WORD
MT=     THE VALUE OF THE BINARY DIGIT TO BE MODULATED
SUM2=   THE DECIMAL EQUIVALENT OF A SET OF RANDOMLY DRAWN
        BINARY DIGITS
C=      THE COLUMN OF THE VECTOR WHERE THE BINARY DIGIT TO
        BE MODULATED IS LOCATED
FLAG=   INTEGER WHICH CONTROLS POINT OF ENTRY TO SUBROUTINE
        CRTHO
*** VARIABLE DEFINITIONS ***
ROW=    RCN OF ARRAY H2N WHICH MATCHES A BINARY CODE WORD
KM1=    INTEGER VALUE OF K MINUS 1
N=      NUMBER OF POSSIBLE BINARY CODE WORDS = 2**K
H2N=    MATRIX OF N ORTHOGONAL VECTORS OF LENGTH N
RH2N=   ROW OF MATRIX H2N
CH2N=   COLUMN OF MATRIX H2N
H2NRP1= ROW OF MATRIX H2N EQUAL TO HALF THE RANK PLUS 1
CH2NRP1= COLUMN OF MATRIX H2N EQUAL TO HALF THE RANK PLUS 1
X=      DOUBLE PRECISION EQUIVALENT ROW REPRESENTATION OF H2N
*** VARIABLE DECLARATIONS ***
INTEGER K,RH2N,CH2N,RH2NP1,CH2NP1,H2NR,N,KM1,H2N(65,65),
*FLAG,ROW,C
DOUBLE PRECISION X,MT,SUM2
*** VARIABLE INITIALIZATION ***
KM1=K-1
N=2**K
RH2N=1
CH2N=1
H2NR=2
EH2NRP1=H2NR+1
CH2NRP1=H2NR+1
H2N(1,1)=1
H2N(1,2)=1

```

```

ORT00110
ORT00120
ORT00130
ORT00140
ORT00150
ORT00160
ORT00170
ORT00180
ORT00190
ORT00200
ORT00210
ORT00220
ORT00230
ORT00240
ORT00250
ORT00260
ORT00270
ORT00280
ORT00290
ORT00300
ORT00310
ORT00320
ORT00330
ORT00340
ORT00350
ORT00360
ORT00370
ORT00380
ORT00390
ORT00400
ORT00410
ORT00420
ORT00430
ORT00440
ORT00450
ORT00460
ORT00470
ORT00480
ORT00490
ORT00500
ORT00510
ORT00520
ORT00530
ORT00540
ORT00550
ORT00560
ORT00570
ORT00580
ORT00590
ORT00600

```

```

C C C
H2N(2,1)=1
H2N(2,2)=-1
*** GENERATE A MATRIX OF N ORTHOGONAL VECTORS OF LENGTH N ***
IF (FLAG.EQ.0) THEN
DO 10 I=1,KM1
DO 20 L=1,H2NR
DO 30 M=1,H2NR
H2N(RH2N,CH2NP1)=H2N(RH2N,CH2N)
H2N(RH2N,CH2NP1)=H2N(RH2N,CH2N)
H2N(RH2NP1,CH2N)=H2N(RH2N,CH2N)
H2N(RH2NP1,CH2NP1)=-H2N(RH2N,CH2N)
CH2N=CH2NP1+1
CH2NP1=CH2NP1+1
CONTINUE
RH2N=RH2N+1
RH2NP1=RH2NP1+1
CH2N=1
CH2NP1=H2NR+1
CONTINUE
H2NR=2*H2NR
RH2N=1
CH2N=1
RH2NP1=H2NR+1
CH2NP1=H2NR+1
CONTINUE
*** ASSIGN EACH ROW OF H2N A REPRESENTATIVE DECIMAL NUMBER ***
DO 40 I=1,N
H2N(I,65)=I-1
CONTINUE
*** DETERMINE WHICH ROW OR H2N MATCHES THE VALUE OF THE
REPRESENTATIVE DECIMAL EQUIVALENT TO THE BINARY CODE WORD ***
DO 50 I=1,N
X=H2N(I,65)
IF (SUM2.EQ.X) THEN
ROW=I
END IF
CONTINUE
END IF
*** DETERMINE THE NEW ROW OF H2N ***
IF (FLAG.EQ.1) THEN
DO 60 I=1,N

```

```

ORT 00610
ORT 00620
ORT 00630
ORT 00640
ORT 00650
ORT 00660
ORT 00670
ORT 00680
ORT 00690
ORT 00700
ORT 00710
ORT 00720
ORT 00730
ORT 00740
ORT 00750
ORT 00760
ORT 00770
ORT 00780
ORT 00790
ORT 00800
ORT 00810
ORT 00820
ORT 00830
ORT 00840
ORT 00850
ORT 00860
ORT 00870
ORT 00880
ORT 00890
ORT 00900
ORT 00910
ORT 00920
ORT 00930
ORT 00940
ORT 00950
ORT 00960
ORT 00970
ORT 00980
ORT 00990
ORT 01000
ORT 01010
ORT 01020
ORT 01030
ORT 01040
ORT 01050
ORT 01060
ORT 01070
ORT 01080
ORT 01090
ORT 01100

```

```

ORT01110
ORT01120
ORT01130
ORT01140
ORT01150
ORT01160
ORT01170
ORT01180
ORT01190
ORT01200
ORT01210
ORT01220
ORT01230
ORT01240

```

```

X=H2N(I,65)
IF (SUM2.EQ.X) THEN
  ROW=I
  END IF
  CONTINUE
END IF
*** DETERMINE THE VALUE OF MT TO BE RETURNED ***
MT=H2N(ROW,C)
C=C+1
RETURN
END

```

```

60
C
C
C
C

```



```

C      IF (ANS-EG-1) THEN
          GO TO 19
        END IF
      END IF
      RETURN
    END
  C

```

```

P1001110
P1001120
P1001130
P1001140
P1001150
P1001160
P1001170
P1001180

```

STA00110
STA00120
STA00130
STA00140
STA00150
STA00160
STA00170
STA00180
STA00190
STA00200
STA00210
STA00220
STA00230
STA00240
STA00250
STA00260
STA00270
STA00280
STA00290
STA00300
STA00310
STA00320
STA00330
STA00340
STA00350
STA00360
STA00370
STA00380
STA00390
STA00400
STA00410
STA00420
STA00430
STA00440
STA00450
STA00460
STA00470
STA00480
STA00490
STA00500
STA00510
STA00520
STA00530
STA00540
STA00550
STA00560
STA00570
STA00580
STA00590
STA00600

SUBROUTINE STATS(IN,REP,FLAG)

*** PURPOSE ***

THIS SUBPROGRAM COMPILES THE ACCUMULATED STATISTICS OF THE ELEMENTS OF THE AMPLITUDE SPECTRUM OF THE FFTS.

*** PARAMETER DEFINITIONS ***

IN= INTEGER OF THE NUMBER OF POSITIONS USED IN ARRAY
REP= INTEGER OF THE NUMBER OF THE REPETITION OF THE
GENERATION OF THE AMPLITUDE SPECTRUM OF THE FFT
FLAG= AN INTEGER THAT CONTROLS THE POINT OF ENTRANCE OR
EXIT FROM THE SUBROUTINE

*** VARIABLE DEFINITIONS ***

IND2P1= POSITIONS USED IN ARRAYS SUMX AND SUMXSQ
ARRAY= AN ARRAY CONTAINING THE VALUES OF THE AMPLITUDE
SPECTRUM OF THE FFT
SUMX= AN ARRAY CONTAINING THE SUM OF THE VALUES OF THE
AMPLITUDE SPECTRUM OF THE FFT
SUMXSQ= AN ARRAY CONTAINING THE SUM OF THE SQUARES OF THE
VALUES OF THE AMPLITUDE SPECTRUM OF THE FFT
SUMX3= AN ARRAY CONTAINING THE SUM OF THE SQUARES OF THE
VALUES OF THE AMPLITUDE SPECTRUM OF THE FFT
SUMX4= AN ARRAY CONTAINING THE SUM OF THE QUARTICS OF THE
VALUES OF THE AMPLITUDE SPECTRUM OF THE FFT
XBAR= AN ARRAY CONTAINING THE AVERAGE VALUE OF THE
ELEMENTS OF THE AMPLITUDE SPECTRUM
VAR= AN ARRAY CONTAINING THE VARIANCE OF THE VALUES OF
THE ELEMENTS OF THE AMPLITUDE SPECTRUM
SKEW= AN ARRAY CONTAINING THE SKEWNESS OF THE DISTRIBUTION
OF THE ELEMENTS OF THE AMPLITUDE SPECTRUM
KUR= AN ARRAY CONTAINING THE KURTOSIS OF THE DISTRIBUTION
OF THE ELEMENTS OF THE AMPLITUDE SPECTRUM
SUMVAR= SUM OF THE VARIANCES OF AN ELEMENT OF THE AMPLITUDE
SPECTRUM
AVAR= AVERAGE VARIANCE OF AN ELEMENT OF THE AMPLITUDE
SPECTRUM
VARVAR= VARIANCE OF THE VARIANCES OF AN ELEMENT OF THE
AMPLITUDE SPECTRUM
SUMVSQ= SUM OF THE SQUARE OF THE VARIANCES OF AN ELEMENT OF
AMPLITUDE SPECTRUM
SUMSKW= SUM OF THE SKEWNESS OF THE ELEMENT OF THE AMPLITUDE
SPECTRUM
ASKEW= AVERAGE SKEWNESS OF THE ELEMENTS OF THE AMPLITUDE
SPECTRUM
VARSKW= VARIANCE OF THE SKEWNESS OF THE ELEMENTS OF THE
AMPLITUDE SPECTRUM

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


```

SUMSSQ=0.D0
VARSIW=0.D0
SUMK1=0.D0
SUMK2=0.D0
VARKUR=0.D0
NDP=IND2P1
C
DO 80 I=1,IND2P1
    SUMVAR=SUMVAR+VAR(I)**2
    SUMVSO=SUMVSO+(VAR(I)**2)
    SUMSKW=SUMSKW+SKW(I)**2
    SUMSSQ=SUMSSQ+(SKW(I)**2)
    SUMKUR=SUMKUR+KUR(I)**2
    SUMKSQ=SUMKSQ+(KUR(I)**2)
CONTINUE
80 C
AVAR=SUMVAR/NDP
ASKEW=SUMSKW/NDP
AKUR=SUMKUR/NDP
C
DO 90 I=1,IND2P1
    VARVAR=VARVAR+(VAR(I)-AVAR)**2
    VARSKW=VARSKW+(SKW(I)-ASKEW)**2
    VARKUR=VARKUR+(KUR(I)-AKUR)**2
CONTINUE
90 C
VARVAR=VARVAR/(NDP-1.D0)
VARSKW=VARSKW/(NDP-1.D0)
VARKUR=VARKUR/(NDP-1.D0)
C
WRITE(6,91)
FORMAT(1,2X,' SUM OF VARIANCES',7X,' SUM OF VARIANCES**2')
91 C
WRITE(6,92)SUMVAR,SUMVSO
FORMAT(1X,E23.16,2X,E23.16)
92 C
WRITE(6,93)
FORMAT(1,2X,' SUM OF SKEWNESS',9X,' SUM OF SKEWNESS**2')
93 C
WRITE(6,94)SUMSKW,SUMSSQ
FORMAT(1X,E23.16,2X,E23.16)
94 C
WRITE(6,95)
FORMAT(1,2X,' SUM OF KURTOSIS',9X,' SUM OF KURTOSIS**2')
95 C
WRITE(6,96)SUMKUR,SUMKSQ
FORMAT(1X,E23.16,2X,E23.16)
96 C
WRITE(6,100)
FORMAT(1,3X,' MEAN VARIANCE',7X,' VARIANCE OF THE VARIANCES**2')
100

```

```

C      *      )
101    WRITE(6,101)AVAR,VARVAR
C      FORMAT(1X,E23.16,2X,E23.16)
102    WRITE(6,102)
C      FORMAT(10,3X,' MEAN SKEWNESS',7X,' VARIANCE OF THE SKEWNESS')
103    WRITE(6,103)ASKED,VARSKW
C      FORMAT(1X,E23.16,2X,E23.16)
104    WRITE(6,104)
C      FORMAT(10,3X,' MEAN KURTOSIS',7X,' VARIANCE OF THE KURTOSIS')
105    WRITE(6,105)AKUR,VARKUR
C      FORMAT(1X,E23.16,2X,E23.16)
106    WRITE(6,106)
C      FORMAT(10,3X,' MEAN KURTOSIS',7X,' VARIANCE OF THE KURTOSIS')
C      END IF
C      RETURN
C      END
STA02110
STA02120
STA02130
STA02140
STA02150
STA02160
STA02170
STA02180
STA02190
STA02200
STA02210
STA02220
STA02230
STA02240
STA02250
STA02260
STA02270
STA02280
STA02290
STA02300
STA02310
STA02320
STA02330
STA02340

```

APPENDIX B IMSL/NON-IMSL ROUTINES UTILIZED

IMSL ROUTINES

IMSL ROUTINE NAME	-	GGUBFS	GGU00110
COMPUTER	-	IEM/SINGLE	GGU00120
LATEST REVISION	-	JUNE 1, 1980	GGU00130
PURPOSE	-	BASIC UNIFORM (0,1) RANDOM NUMBER GENERATOR - FUNCTION FORM OF GGUBS	GGU00140
USAGE	-	FUNCTION GGUBFS (DSEED)	GGU00150
ARGUMENTS	GGUBFS	RESULTANT DEVIATE.	GGU00160
	DSEED	INPUT/OUTPUT DOUBLE PRECISION VARIABLE ASSIGNED AN INTEGER VALUE IN THE EXCLUSIVE RANGE (1.D0, 2147483647.D0). DSEED IS REPLACED BY A NEW VALUE TO BE USED IN A SUBSEQUENT CALL.	GGU00170
PRECISION/HARDWARE	-	SINGLE/ALL	GGU00180
REQD. IMSL ROUTINES	-	NCNE REQUIRED	GGU00190
NOTATION	-	INFORMATION ON SPECIAL NOTATION AND CONVENTIONS IS AVAILABLE IN THE MANUAL INTRODUCTION OR THROUGH IMSL ROUTINE UHELP	GGU00200
COPYRIGHT	-	1978 BY IMSL, INC. ALL RIGHTS RESERVED.	GGU00210
WARRANTY	-	IMSL WARRANTS ONLY THAT IMSL TESTING HAS BEEN APPLIED TO THIS CODE. NO OTHER WARRANTY, EXPRESSED OR IMPLIED, IS APPLICABLE.	GGU00220
REAL FUNCTION GGUBFS (DSEED)			GGU00230
DOUBLE PRECISION DSEED			GGU00240
SPECIFICATIONS FOR ARGUMENTS			GGU00250
SPECIFICATIONS FOR LOCAL VARIABLES			GGU00260
			GGU00270
			GGU00280
			GGU00290
			GGU00300
			GGU00310
			GGU00320
			GGU00330
			GGU00340
			GGU00350
			GGU00360
			GGU00370
			GGU00380
			GGU00390
			GGU00400
			GGU00410
			GGU00420
			GGU00430
			GGU00440
			GGU00450
			GGU00460
			GGU00470
			GGU00480
			GGU00490
			GGU00500

```

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

```

IMSL ROUTINE NAME - FFTRC

COMPUTER - IBM/DOUBLE

LATEST REVISION - JANUARY 1, 1978

PURPOSE - COMPUTE THE FAST FOURIER TRANSFORM OF A REAL VALUED SEQUENCE

USAGE - CALL FFTRC (A,N,X,INWK,WK)

ARGUMENTS A - INPUT REAL VECTOR OF LENGTH N WHICH CONTAINS THE DATA TO BE TRANSFORMED.
N - INPUT NUMBER OF DATA POINTS TO BE TRANSFORMED.
X - OUTPUT COMPLEX VECTOR OF LENGTH $N/2+1$. N MUST BE A POSITIVE EVEN INTEGER.
INWK - CONTAINING THE FIRST $N/2+1$ COEFFICIENTS OF THE FOURIER TRANSFORM. THE REMAINING COEFFICIENTS MAY BE DETERMINED BY $X(N/2+1) = \text{CONJG}(X(I))$, FOR $I=2, \dots, N/2$.
IF N IS A POWER OF 2, THEN INWK SHOULD BE OF LENGTH N WHERE $N=2**I$.
PRECISION/HARDWARE - SINGLE AND DOUBLE/H32
- SINGLE/H36, H48, H60

REQD. IMSL ROUTINES - FFTC, FFT2C

NOTATION - INFORMATION ON SPECIAL NOTATION AND CONVENTIONS IS AVAILABLE IN THE MANUAL INTRODUCTION OR THROUGH IMSL ROUTINE UHELP

REMARKS 1. FFTRC COMPUTES THE FOURIER TRANSFORM, X, ACCORDING TO THE FOLLOWING FORMULA;

$$X(K+1) = \text{SUM FROM } J = 0 \text{ TO } N-1 \text{ OF } A(J+1) * \text{CEXP}((0.0(2.0*PI)*J*K)/N)$$
FOR $K=0, 1, \dots, N/2$ AND $PI=3.1415\dots$
THE USER CAN COMPUTE THE REMAINING X VALUES BY PERFORMING THE FOLLOWING STEPS;
ND2 = N/2
DO 10 I=2, ND2
X(N+2-I) = CONJG(X(I))
10 CONTINUE
2. FFTRC CAN BE USED TO COMPUTE

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
X(K+1) = (1/N)*SUM FROM J = 0 TO N-1 OF
A(J+1)*CEXP(0.0*(-2.0*PI*J*K)/N))
FOR K=0,1,....,N/2 AND PI=3.1415...
BY PERFORMING THE FOLLOWING STEPS:

CALL FFTRC (A,N,X,IWK,WK)
ND2P1 = N/2+1
DO 10 I=1,ND2P1
X(I) = CONJG(X(I))/N
10 CONTINUE

COPYRIGHT - 1978 BY IMSL, INC. ALL RIGHTS RESERVED.
WARRANTY - IMSL WARRANTS ONLY THAT IMSL TESTING HAS BEEN
APPLIED TO THIS CODE. NO OTHER WARRANTY,
EXPRESSED OR IMPLIED, IS APPLICABLE.

MODIFIED 5 AUGUST 1985 TO ALLOW THE GENERATION ONLY OF FFT CF VECTOR
OF LENGTH EQUAL TO AN EVEN POWER OF 2 AND THE ENTIRE FFT.
-----
SUBROUTINE FFTRC (A,N,X,IWK) SPECIFICATIONS FOR ARGUMENTS
INTEGER N,IWK(1)
DOUBLE PRECISION A(N)
COMPLEX*16 X(1)

INTEGER ND2P1,ND2,I,M TWO,H,IMAX,ND4,NP2,K,NMK,J
DOUBLE PRECISION RPI,ZERO,ONE,HALF,THETA,TP,G(2),B(2),Z(2),AI,
COMPLEX*16 XIMAG,ALPH,BETA,GAM,S1,ZD
EQUIVALENCE (GAM,G(1)),(ALPH,B(1)),(Z(1),AR),(Z(2),AI),
(ZD,Z(1))
DATA ZERO/0.0D0/,HALF/0.5D0/,ONE/1.0D0/,IMAX/24/
DATA RPI/3.141592653589793D0/
IF (N.NE.2) GO TO 5
N EQUAL TO 2
ZD = DCMPLX(A(1),A(2))
THETA = AR
TP = AI
X(2) = DCMPLX(THETA-TP,ZERO)
X(1) = DCMPLX(THETA+TP,ZERO)
GO TO 9005
5 CONTINUE
ND2 = N/2
ND2P1 = ND2+1
N GREATER THAN 2
MOVE A TO X
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```



```

C
J = 1
DO 6 I=1,ND2
  X(I) = DCMPLX(A(J),A(J+1))
  J = J+2
6 CONTINUE

C
  GAM = DCMPLX(ZERO,ZERO)
  DO 10 I=1,ND2
    GAM = GAM + X(I)
  10 CONTINUE
  TP = G(1)-G(2)
  GAM = DCMPLX(TP,ZERO)

C
  MTWO = 2
  M = 1
  DO 15 I=1,IMAX
    IF (ND2 .LE. MTWO) GO TO 20
    MTWO = MTWO+MTWO
    M = M+1
  15 CONTINUE
  20 IF (ND2 .EQ. MTWO) GO TO 25

C
  25 CALL FFT2C(X,M,IMK)
  30 ALPHA = X(1)
  X(1) = B(1) + B(2)
  ND4 = (ND2+1)/2
  IF (ND4 .LT. 2) GO TO 40
  NP2 = ND2 + 2
  THETA = RPI/ND2
  TP = THETA
  XIMAG = DCMPLX(ZERO,CNE)

C
  DO 35 K = 2,ND4
    NEK = NP2 - K
    S1 = DCONJG(X(NMK))
    ALPHA = X(K) + S1
    BETA = XIMAG*(S1-X(K))
    S1 = DCMPLX(DCOS(THETA),DSIN(THETA))
    X(K) = (ALPHA+BETA*S1)*HALF
    X(NMK) = DCONJG(ALPHA-BETA*S1)*HALF
    THETA = THETA + TP
  35 CONTINUE
  40 X(ND2P1) = GAM
  ND2=N/2
  DO 90 I=2,ND2
    FFT01180
    FFT01190
    FFT01200
    FFT01210
    FFT01220
    FFT01230
    FFT01240
    FFT01250
    FFT01260
    FFT01270
    FFT01280
    FFT01290
    FFT01300
    FFT01310
    FFT01320
    FFT01330
    FFT01340
    FFT01350
    FFT01360
    FFT01370
    FFT01380
    FFT01390
    FFT01400
    FFT01410
    FFT01420
    FFT01430
    FFT01440
    FFT01450
    FFT01460
    FFT01470
    FFT01480
    FFT01490
    FFT01500
    FFT01510
    FFT01520
    FFT01530
    FFT01540
    FFT01550
    FFT01560
    FFT01570
    FFT01580
    FFT01590
    FFT01600
    FFT01610
    FFT01620
    FFT01630
    FFT01640
    FFT01650
    FFT01660
    FFT01670
  
```

C
C90 X(N+2-I)=DCONJG(X(I))
9005 CONTINUE
RETURN
END

FFT01680
FFT01690
FFT01700
FFT01710

IMSL ROUTINE NAME	- FFT2C	
COMPUTER	- IBM/DOUBLE	
LATEST REVISION	- NOVEMBER 1, 1984	
PURPOSE	- COMPUTE THE FAST FOURIER TRANSFORM OF A COMPLEX VALUED SEQUENCE OF LENGTH EQUAL TO A POWER OF TWO	
USAGE	- CALL FFT2C (A,M,IWK)	
ARGUMENTS	A - COMPLEX VECTOR OF LENGTH N, WHERE N=2**M. ON INPUT A CONTAINS THE COMPLEX VALUED SEQUENCE TO BE TRANSFORMED. ON OUTPUT A IS REPLACED BY THE FOURIER TRANSFORM.	
M	- INPUT EXPONENT TO WHICH 2 IS RAISED TO PRODUCE THE NUMBER OF DATA POINTS, N (I.E. N = 2**M).	
IWK	- WORK VECTOR OF LENGTH M+1.	
PRECISION/HARDWARE	- SINGLE AND DOUBLE/H32 - SINGLE/H36,H48,H60	
REQD. IMSL ROUTINES	- NCNE REQUIRED	
NOTATION	- INFORMATION ON SPECIAL NOTATION AND CONVENTIONS IS AVAILABLE IN THE MANUAL INTRODUCTION OR THROUGH IMSL ROUTINE UHELP	
REMARKS	1. FFT2C COMPUTES THE FOURIER TRANSFORM, X, ACCORDING TO THE FOLLOWING FORMULA: $X(K+1) = \sum_{J=0}^{N-1} A(J+1) * \exp\{i(2.0 * \pi * J * K) / N\}$ FOR K=0,1,...,N-1 AND $\pi = 3.1415...$ 2. NOTE THAT X OVERWRITES A ON OUTPUT. FFT2C CAN BE USED TO COMPUTE THE INVERSE FOURIER TRANSFORM, X, ACCORDING TO THE FOLLOWING FORMULA: $X(K+1) = (1/N) * \sum_{J=0}^{N-1} A(J+1) * \exp\{i(2.0 * \pi * J * K) / N\}$ FOR K=0,1,...,N-1 AND $\pi = 3.1415...$ BY PERFORMING THE FOLLOWING STEPS;	

```

CCCCCCCCCCCCCCCCCCCC
DO 10 I=1,N
  A(I) = CONJG (A(I))
10 CONTINUE
  CALL FFT2C (A,M,IWK)
DO 20 I=1,N
  A(I) = CONJG (A(I))/N
20 CONTINUE

  - 1978 BY IMSL, INC. ALL RIGHTS RESERVED.
  - IMSL WARRANTIES ONLY THAT IMSL TESTING HAS BEEN
    APPLIED TO THIS CODE. NO OTHER WARRANTY,
    EXPRESSED OR IMPLIED, IS APPLICABLE.
-----
SUBROUTINE FFT2C (A,M,IWK)
  INTEGER
  COMPLEX*16
  INTEGER
1 DOUBLE PRECISION
1
2 COMPLEX*16
EQUIVALENCE
1
2
3 DATA
1
2
3 DATA
MP = M+1
N = 2**M
IWK(1) = 1
MM = (M/2)*2
KN = N+1
DC 5 I=2,MP
  IWK(I) = IWK(I-1)+IWK(I-1)
5 CONTINUE
RAD = TWOPI/N
MK = M - 4

  SPECIFICATIONS FOR ARGUMENTS
  M, IWK(1)
  A(1)
  I, ISP, J, JJ, JSP, K, KO, K1, K2, K3, KB, KN, MK, MM, MP, N,
  N4, N8, W2, LM, MN, JK
  RAD, C1, C2, C3, S1, S2, S3, CK, SK, SQ, AO, A1, A2, A3,
  BO, B1, B2, B3, TWOPI, TEMP
  ZERO, ONE, Z0(2), Z1(2), Z2(2), Z3(2)
  ZA0, ZA1, ZA2, ZA3, A2
  {ZA0, Z0(1)}, {ZA1, Z1(1)}, {ZA2, Z2(1)}, {ZA3, Z3(1)},
  {B1, Z1(2)}, {B2, Z2(2)}, {B3, Z3(2)},
  {B3, Z3(2)},
  SQ/.7071067811865475D0/,
  SK/.3826834323650898D0/,
  CK/.9238795325112868D0/,
  TWOPI/6.283185307179586D0/
  ZERO/0.0D0/, ONE/1.0D0/
  SQ=SQRT2/2, SK=SIN (PI/8), CK=COS (PI/8)
  TWOPI=2*PI
  FIRST EXECUTABLE STATEMENT

  INITIALIZE WORK VECTOR

```

```

KB = 1
IF (MM - EQ - M) GO TO 15
K2 = KN
K0 = IWK (MM+1) + KB
10 K2 = K2 - 1
K0 = K0 - 1
AK2 = A (K2)
A (K2) = A (K0) - AK2
A (K0) = A (K0) + AK2
IF (K0 - GI - KB) GO TO 10
15 C1 = ONE
S1 = ZERO
JJ = 0
K = MM - 1
J = 4
IF (K - GE - 1) GO TO 30
GO TO 70
20 IF (IWK (J) - GT - JJ) GO TO 25
JJ = JJ - IWK (J)
J = J - 1
IF (IWK (J) - GT - JJ) GO TO 25
JJ = JJ - IWK (J)
J = J - 1
K = K + 2
GO TO 20
25 JJ = IWK (J) + JJ
J = 4
30 ISP = IWK (K)
IF (JJ - EQ - 0) GO TO 40
C
C2 = JJ * ISP * RAD
C1 = DCOS (C2)
35 C2 = C1 * C1 - S1 * S1
S2 = C1 * (S1 + S1)
C3 = C2 * C1 - S2 * S1
S3 = C2 * S1 + S2 * C1
40 JSP = ISP + KB
C
DO 50 I = 1, ISP - 1
K0 = JSP - I
K1 = K0 + ISP
K2 = K1 + ISP
K3 = K2 + ISP
ZA0 = A (K0)
ZA1 = A (K1)
ZA2 = A (K2)
ZA3 = A (K3)
IF (S1 - EQ - ZERO) GO TO 45

```

RESET TRIGONOMETRIC PARAMETERS

DETERMINE FOURIER COEFFICIENTS
IN GROUPS OF 4

```

FFT01110
FFT01120
FFT01130
FFT01140
FFT01150
FFT01160
FFT01170
FFT01180
FFT01190
FFT01200
FFT01210
FFT01220
FFT01230
FFT01240
FFT01250
FFT01260
FFT01270
FFT01280
FFT01290
FFT01300
FFT01310
FFT01320
FFT01330
FFT01340
FFT01350
FFT01360
FFT01370
FFT01380
FFT01390
FFT01400
FFT01410
FFT01420
FFT01430
FFT01440
FFT01450
FFT01460
FFT01470
FFT01480
FFT01490
FFT01500
FFT01510
FFT01520
FFT01530
FFT01540
FFT01550
FFT01560
FFT01570
FFT01580
FFT01590
FFT01600

```

```

TEMP = A1 * C1 - B1 * S1
A1 = TEMP * S1 + B1 * C1
B1 = TEMP * A2
TEMP = A2 * C2 - B2 * S2
A2 = TEMP * S2 + B2 * C2
B2 = TEMP * A3
TEMP = A3 * C3 - B3 * S3
A3 = TEMP * S3 + B3 * C3
B3 = TEMP * A0 - A2
TEMP = A0 - A2
A0 = TEMP * A1 + A3
TEMP = A1 - A3
A3 = TEMP * B0 + B2
TEMP = B0 - B2
B0 = TEMP * B1 + B3
TEMP = B1 - B3
B1 = TEMP
A(K0) = DCMPLEX(A0+A1, B0+B1)
A(K1) = DCMPLEX(A0-A1, B0-B1)
A(K2) = DCMPLEX(A2-B3, B2+A3)
A(K3) = DCMPLEX(A2+B3, B2-A3)
50 CONTINUE
IF (K .LE. 1) GO TO 55
K = K - 2
GO TO 30
55 KB = K3 + 1
IF (KN .LE. KB) GO TO 70
IF (J .NE. 1) GO TO 60
K = 3
J = MK
GO TO 20
60 J = J - 1
C2 = C1
IF (J .NE. 2) GO TO 65
C1 = C1 * CK + S1 * SK
S1 = S1 * CK - C2 * SK
GO TO 35
65 C1 = (C1 - S1) * SQ
S1 = (C2 + S1) * SQ
GO TO 35
70 CONTINUE

```

C
C
CHECK FOR COMPLETION OF FINAL
ITERATION

C
C
C
PERMUTE THE COMPLEX VECTOR IN
REVERSE BINARY ORDER TO NORMAL
ORDER

FFT01610
FFT01620
FFT01630
FFT01640
FFT01650
FFT01660
FFT01670
FFT01680
FFT01690
FFT01700
FFT01710
FFT01720
FFT01730
FFT01740
FFT01750
FFT01760
FFT01770
FFT01780
FFT01790
FFT01800
FFT01810
FFT01820
FFT01830
FFT01840
FFT01850
FFT01860
FFT01870
FFT01880
FFT01890
FFT01900
FFT01910
FFT01920
FFT01930
FFT01940
FFT01950
FFT01960
FFT01970
FFT01980
FFT01990
FFT02000
FFT02010
FFT02020
FFT02030
FFT02040
FFT02050
FFT02060
FFT02070
FFT02080
FFT02090
FFT02100

PFT02110
PFT02120
PFT02130
PFT02140
PFT02150
PFT02160
PFT02170
PFT02180
PFT02190
PFT02200
PFT02210
PFT02220
PFT02230
PFT02240
PFT02250
PFT02260
PFT02270
PFT02280
PFT02290
PFT02300
PFT02310
PFT02320
PFT02330
PFT02340
PFT02350
PFT02360
PFT02370
PFT02380
PFT02390
PFT02400
PFT02410
PFT02420
PFT02430
PFT02440
PFT02450
PFT02460
PFT02470
PFT02480
PFT02490
PFT02500
PFT02510
PFT02520
PFT02530
PFT02540
PFT02550
PFT02560
PFT02570
PFT02580
PFT02590
PFT02600

INITIALIZE WORK VECTOR

```

IF (M .LE. 1) GO TO 9005
MP = M+1
JJ = 1
IWK(J) = 1
DO 75 I = 2, MP
  IWK(I) = IWK(I-1) * 2
75 CONTINUE
N4 = IWK(MP-2)
IF (M .GT. 2) N8 = IWK(MP-3)
N2 = IWK(MP-1)
LM = N2
NN = IWK(MP) + 1
MP = MP-4
J = 2
JK = JJ + N2
AK2 = A(J)
A(J) = A(JK)
A(JK) = AK2
J = J+1
IF (JJ .GT. N4) GO TO 85
JJ = JJ + N4
GO TO 105
85 JJ = JJ - N4
IF (JJ .GT. N8) GO TO 90
JJ = JJ + N8
GO TO 105
90 JJ = JJ - N8
K = MP
95 IF (IWK(K) .GE. JJ) GO TO 100
JJ = JJ - IWK(K)
K = K - 1
GO TO 95
100 JJ = IWK(K) + JJ
105 IF (JJ .LE. J) GO TO 110
K = NN - J
JK = NN - JJ
AK2 = A(J)
A(J) = A(JJ)
A(JJ) = AK2
AK2 = A(K)
A(K) = A(JK)
A(JK) = AK2
J = J + 1
110 J = J + 1

```

DETERMINE INDICES AND SWITCH A

CYCLE REPEATED UNTIL LIMITING NUMBER OF CHANGES IS ACHIEVED

```

C
C IF (J .LE. LM) GO TO 80
C
C 9005 RETURN

```

FFT02610

END

AD-A160 823

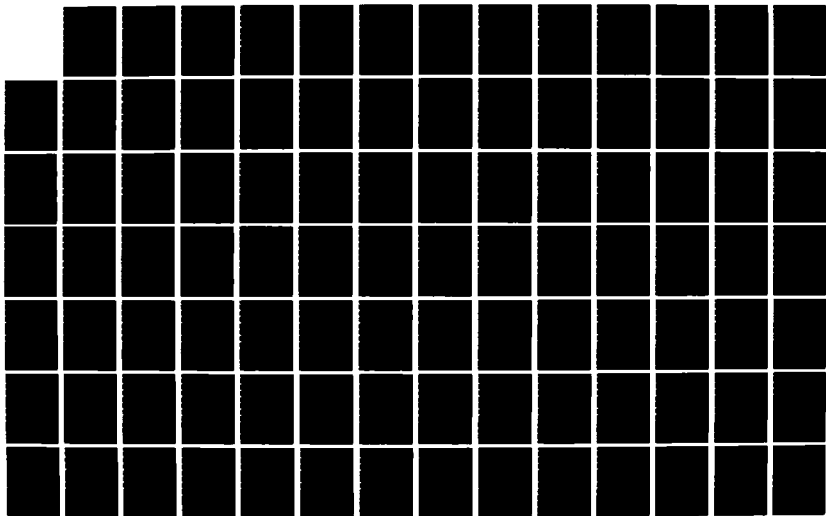
COMPUTER SIMULATION OF DIGITAL SIGNAL MODULATION
TECHNIQUES IN SATELLITE COMMUNICATIONS(U) NAVAL
POSTGRADUATE SCHOOL MONTEREY CA C D CARLSON SEP 85

3/4

UNCLASSIFIED

F/G 9/2

NL



NON-INSL ROUTINE

```

.....
SUBROUTINE UTPLLOT
PURPOSE
    PRINTS GRAPHS ON THE STANDARD OUTPUT PRINTER
FEATURES
    1) FULL CONTROL OVER SCALING
    2) ABILITY TO PLOT SINGLE OR DOUBLE PRECISION VECTORS
CALLING SEQUENCE
CALL UTPLLOT(X,Y,N,RANGE,K,MODCUR)
DESCRIPTION OF ARGUMENTS
X      VECTOR OF ABSCISSAE
Y      VECTOR OF ASSOCIATED ORDINATES
N      NUMBER OF (X,Y) PAIRS
RANGE  4 WORD SCALING VECTOR WHERE
        RANGE(1) = MAXIMUM X TO BE PLOTTED
        RANGE(2) = MINIMUM X TO BE PLOTTED
        RANGE(3) = MAXIMUM Y TO BE PLOTTED
        RANGE(4) = MINIMUM Y TO BE PLOTTED
        ALL (X,Y) POINTS OUTSIDE THE ABOVE RANGE WILL BE PLOTTED
        IN THE BORDER OF THE GRAPH.
K      EVERY KTH ELEMENT OF X & Y WILL BE PLOTTED, E.G.,
        FOR REAL*4 DATA (SINGLE PRECISION) K=1
        FOR REAL*8 DATA (DOUBLE PRECISION) K=2.
MODCUR  CONTROLS THE NUMBER OF CURVES ON ONE GRAPH
        =0 THERE IS ONLY 1 CURVE ON THIS GRAPH
        =1 THIS IS THE FIRST OF TWO OR MORE CURVES ON THIS GRAPH
        =2 THIS IS AN INTERMEDIATE CURVE ON THIS GRAPH
        =3 THIS IS THE LAST CURVE ON THIS GRAPH
SCALING
    SCALING IS PERFORMED ONLY ON THE FIRST SET OF POINTS (WHEN
UTP000110
UTP000120
UTP000130
UTP000140
UTP000150
UTP000160
UTP000170
UTP000180
UTP000190
UTP000200
UTP000210
UTP000220
UTP000230
UTP000240
UTP000250
UTP000260
UTP000270
UTP000280
UTP000290
UTP000300
UTP000310
UTP000320
UTP000330
UTP000340
UTP000350
UTP000360
UTP000370
UTP000380
UTP000390
UTP000400
UTP000410
UTP000420
UTP000430
UTP000440
UTP000450
UTP000460
UTP000470
UTP000480
UTP000490
UTP000500
UTP000510
UTP000520
UTP000530
UTP000540
UTP000550
UTP000560
UTP000570

```

MODCUR = 0 OR 1.) ARRAY RANGE IS USED TO SET UP THE SCALE FACTORS AND NEED ONLY BE DEFINED FOR THE FIRST CALL TO UTPLOT.

GRID LABELLING

THE DATA TO BE GRAPHED WILL BE FIT INTO AN 80 COLUMN BY BY 60 ROW GRID. THE GRID WILL BE LABELLED THUSLY:

IN THE X DIRECTION (COLUMN-WISE), THERE WILL BE 5 VALUES: THE MAXIMUM, THE MINIMUM, AND 3 INTERMEDIATE AT INCREMENTS OF (RANGE(2)-RANGE(1))/4. FROM THE MINIMUM.

IN THE Y DIRECTION (ROW-WISE), THERE WILL BE 7 VALUES: THE MAXIMUM, THE MINIMUM, AND 5 INTERMEDIATE AT INCREMENTS OF (RANGE(4)-RANGE(3))/6. FROM THE MINIMUM.

IF THE LABELS HAVE A VALUE BETWEEN 1. AND 10**8, THEY WILL BE PRINTED IN AN F11.2 FORMAT, OTHERWISE THEY WILL BE PRINTED IN A 1PE10.3 FORMAT.

PLOTTING

FOUR CHARACTERS ARE USED FOR PLOTTING CURVES, "X", "Y", "I", AND "S". WHEN MORE THAN 4 CURVES ARE PLOTTED THE CHARACTERS ARE USED REPEATEDLY. IF A NEW CURVE IS TO BE PLACED IN THE PLOTTING GRID WHERE AN OLD CURVE EXISTS, THE NEW CURVE REPLACES THE OLD ONE. THUS, IF 3 IDENTICAL CURVES ARE PLOTTED, THEY WILL APPEAR AS ONE CURVE COMPOSED OF "X" 'S.

MESSAGES

UNDER CERTAIN CIRCUMSTANCES, A PLOT WILL NOT BE OUTPUT AND ONE OF THE FOLLOWING MESSAGES WILL BE PRINTED ON THE STANDARD OUTPUT IN PLACE OF THE PLOT.

"ALL Y-VALUES=0. CANNOT SETUP PLOT GRID. CHECK MAX & MIN Y WHEN MODCUR=0 OR 1."

"ALL X VALUES=0. CANNOT SETUP PLOT GRID. CHECK MAX AND MIN X WHEN MODCUR=0 OR 1."

"GRID NOT SETUP WHEN MODCUR LAST 0 OR 1. NO PLOT UNTIL GRID PROPERLY SETUP."

NOTE

THE USER IS EXPECTED TO PROVIDE THE NECESSARY CARRIAGE CONTROLS TO PLACE THE GRAPH PROPERLY ON THE PAGE. BEFORE CALLING UTPLOT THE USER SHOULD ISSUE A PRINT STATEMENT WHICH EJECTS A PAGE SO THAT THE GRAPH WILL BE PLOTTED AT THE TOP

UTP00580
UTP00590
UTP00600
UTP00610
UTP00620
UTP00630
UTP00640
UTP00650
UTP00660
UTP00670
UTP00680
UTP00690
UTP00700
UTP00710
UTP00720
UTP00730
UTP00740
UTP00750
UTP00760
UTP00770
UTP00780
UTP00790
UTP00800
UTP00810
UTP00820
UTP00830
UTP00840
UTP00850
UTP00860
UTP00870
UTP00880
UTP00890
UTP00900
UTP00910
UTP00920
UTP00930
UTP00940
UTP00950
UTP00960
UTP00970
UTP00980
UTP00990
UTP01000
UTP01010
UTP01020
UTP01030
UTP01040
UTP01050
UTP01060
UTP01070


```

UTP01580
UTP01590
UTP01600
UTP01610
UTP01620
UTP01630
UTP01640
UTP01650
UTP01660
UTP01670
UTP01680
UTP01690
UTP01700
UTP01710
UTP01720
UTP01730
UTP01740
UTP01750
UTP01760
UTP01770
UTP01780
UTP01790
UTP01800
UTP01810
UTP01820
UTP01830
UTP01840
UTP01850
UTP01860
UTP01870
UTP01880
UTP01890
UTP01900
UTP01910
UTP01920
UTP01930
UTP01940
UTP01950
UTP01960
UTP01970
UTP01980
UTP01990
UTP02000
UTP02010
UTP02020
UTP02030
UTP02040
UTP02050
UTP02060
UTP02070

```

```

IF(YMIN.EQ.0.) GO TO 889
YMIN=0.
YRANGE=YMAX
GO TO 299
298 IF (XRANGE.NE.0.) GO TO 299
IF(XMIN.EQ.0.) GO TO 887
XMIN=0.
XRANGE=XMAX

C BLANKING OUT MATRIX-(GRID)
C
C 299 DO 300 I=1,61
DO 301 JJ=1,81
301 GRID(I,JJ)=BLANK
300 CONTINUE
IF(XMAX*XMIN-GE.0.) GO TO 222
IYAXIS=80.*(-XMIN)/XRANGE+1.5
DO 40 I=1,61
40 GRID(I,IYAXIS)=DOT
222 IF(YMAX*YMIN-GE.0.) GO TO 333
IYAXIS=60.*YMAX/YRANGE+1.5
DO 60 I=1,81
60 GRID(IYAXIS,I)=DOT

C COMPUTE PROPER SCALE NUMBERS
C
C 333 XINCR=XRANGE/4.
YINCR=YRANGE/6.
XSCALE(1)=XMAX
XSCALE(5)=XMIN
DO 80 I=2,4
XSCALE(I)=XSCALE(I-1)-XINCR
IF(ABS(XSCALE(I)).LT.1.E-4) XSCALE(I)=0.
80 CONTINUE
YSCALE(1)=YMAX
YSCALE(7)=YMIN
DO 81 I=2,6
YSCALE(I)=YSCALE(I-1)-YINCR
IF(ABS(YSCALE(I)).LT.1.E-4) YSCALE(I)=0.
81 CONTINUE
DO 85 II=1,2
JJ=6-II
XT=XSCALE(JJ)
XSCALE(JJ)=XSCALE(II)
85 XSCALE(II)=XT

C PLACING POINTS IN THEIR PROPER GRID POSITIONS
C
C 444 IF(MODCUR.LT.2) JSET=0

```

```

IF(JERR.GT.0) GO TO 885
JSET=JSET+1
IF(JSET.GT.4) JSET=1
DO 700 I=1,KLATA,KKZ
IPTX=60.*(YMAX-Y(I))/YRANGE+1.5
IPTV=80.*(X(I)-XMIN)/XRANGE+1.5
IF(IPTX.GT.61.OR.IPTY.GT.81) GO TO 70
IF(IPTX.LE.0.OR.IPTY.LE.0) GO TO 70
GRID(IPTX,IPTY)=XCHAR(JSET)
GO TO 700
70 IERR=IERR+1
700 CONTINUE
C
OUTPUT SECTION WITH GRAPH
IF(MODCUR.EQ.1.OR.MODCUR.EQ.2) RETURN
AYR=ABS(XRANGE)
AYR=ABS(YRANGE)
IF(AYR.LT.1.E+8.AND.AYR.GE..95) GO TO 400
WRITE(6,17) XSCALE
17 FORMAT(12X,1PE10.3,4(10X,E10.3)/15X,'**',8('*****'),'+****')
GO TO 401
400 WRITE(6,117) XSCALE
117 FORMAT(8X,F11.2,4(9X,F11.2)/15X,'**',8('*****'),'+****')
II=1
DO 101 IK=1,61
IF(MOD(1K-1,10).NE.0) GO TO 92
IF(AYR.LT.1.E+8.AND.AYR.GE..95) GO TO 404
WRITE(6,18) YSCALE(II),GRID(1K,IX),IX=1,81,YSCALE(II)
18 FORMAT(3X,1PE10.3,2X,1H+,1X,81A1,1X,1H+,2X,E10.3)
GO TO 405
404 WRITE(6,118) YSCALE(II),GRID(1K,IX),IX=1,81,YSCALE(II)
118 FORMAT(2X,F11.2,1X,81A1,1X,1H+,2X,E10.3)
II=II+1
GO TO 101
92 WRITE(6,19) (GRID(1K,IX),IX=1,81)
19 FORMAT(15X,'*',81A1,*)
101 CONTINUE
IF(AYR.LT.1.E+8.AND.AYR.GE..95) GO TO 402
WRITE(6,22) XSCALE
22 FORMAT(15X,'**',8('*****'),'+****'/12X,1PE10.3,4(10X,E10.3),//)
GO TO 403
402 WRITE(6,217) XSCALE
217 FORMAT(15X,'**',8('*****'),'+****'/8X,F11.2,4(9X,F11.2),//)
403 IF(IERR.GT.0) WRITE(6,20) IERR
20 FORMAT(10X,'NUMBER OF POINTS OUT OF RANGE =',I4)
1000 RETURN
C
889 WRITE(6,888)
888 FORMAT(' ALL Y VALUES=0. CANNOT SETUP PLOT GRID. CHECK MAX & MIN Y')
1 WHEN MODCUR=0 OR 1.'

```

```

JERR=10
RETURN
887 WRITE(6,886)
886 FORMAT(' ALL X VALUES=0. CANNOT SETUP PLOT GRID. CHECK MAX & MIN
1 WHEN MODCUR=0 OR 1.')
JERR=10
RETURN
885 WRITE(6,884)
884 FORMAT(' GRID NOT SETUP WHEN MODCUR LAST 0 OR 1. NO PLOT UNTIL GRID
1D PROPERLY SETUP')
RETURN
END
UTP02580
UTP02590
UTP02600
UTP02610
UTP02620
UTP02630
UTP02640
UTP02650
UTP02660
UTP02670
UTP02680
UTP02690

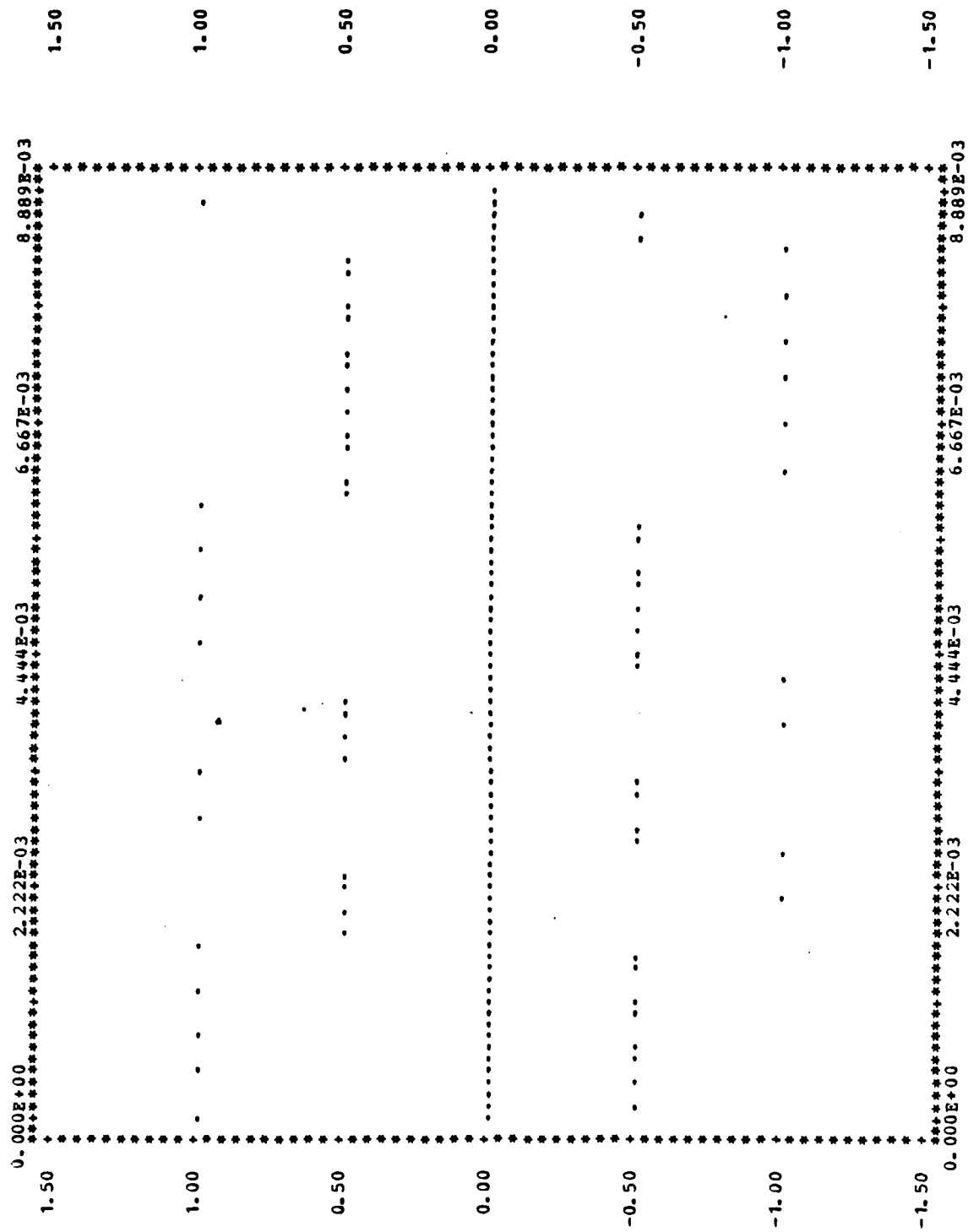
```

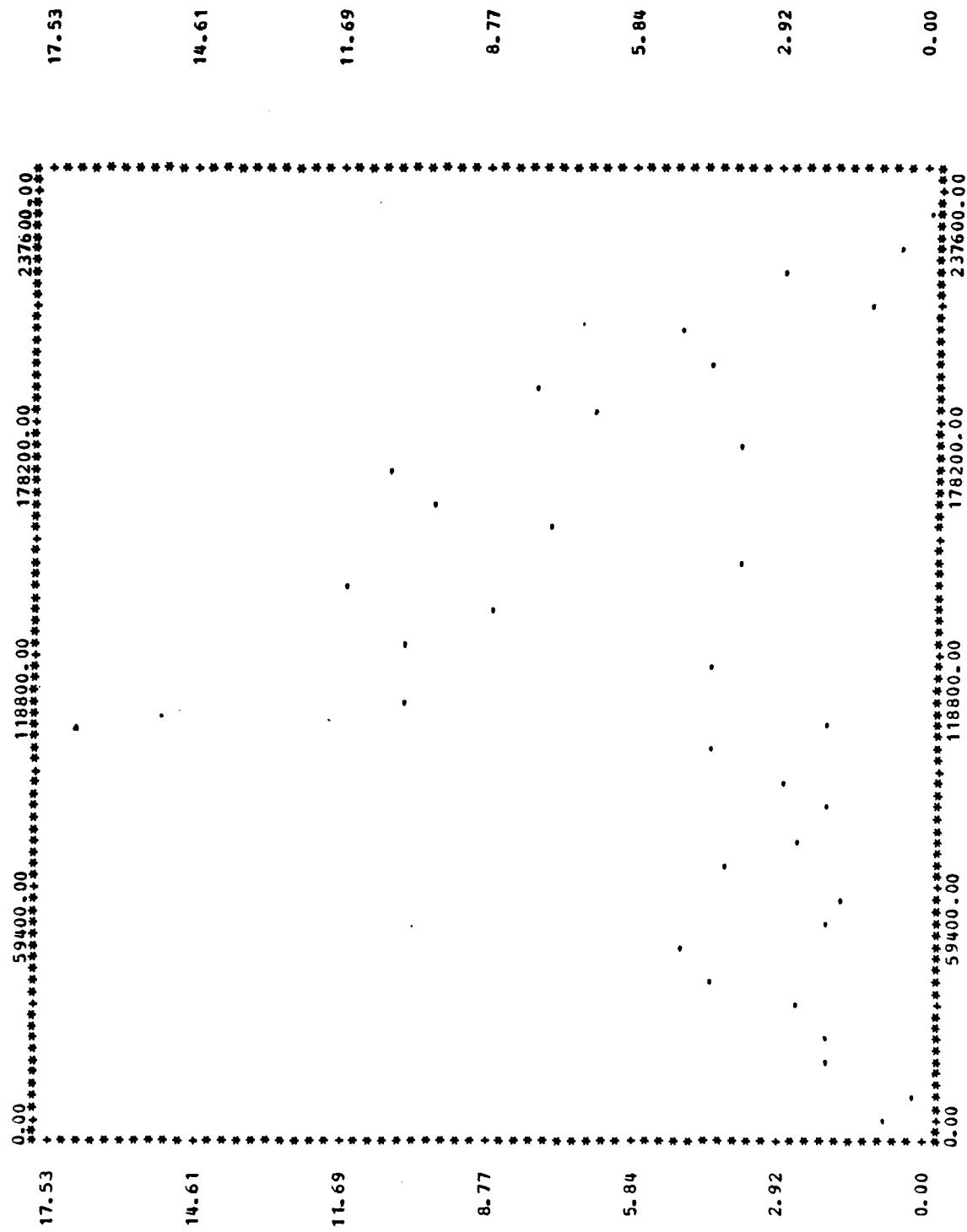

APPENDIX C
REPRESENTATIVE RESULTS OF TRIAL SIMULATIONS

```

MODULATION TECHNIQUE = BPSK
BIFOLAR 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 = 0.138888888888889E-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

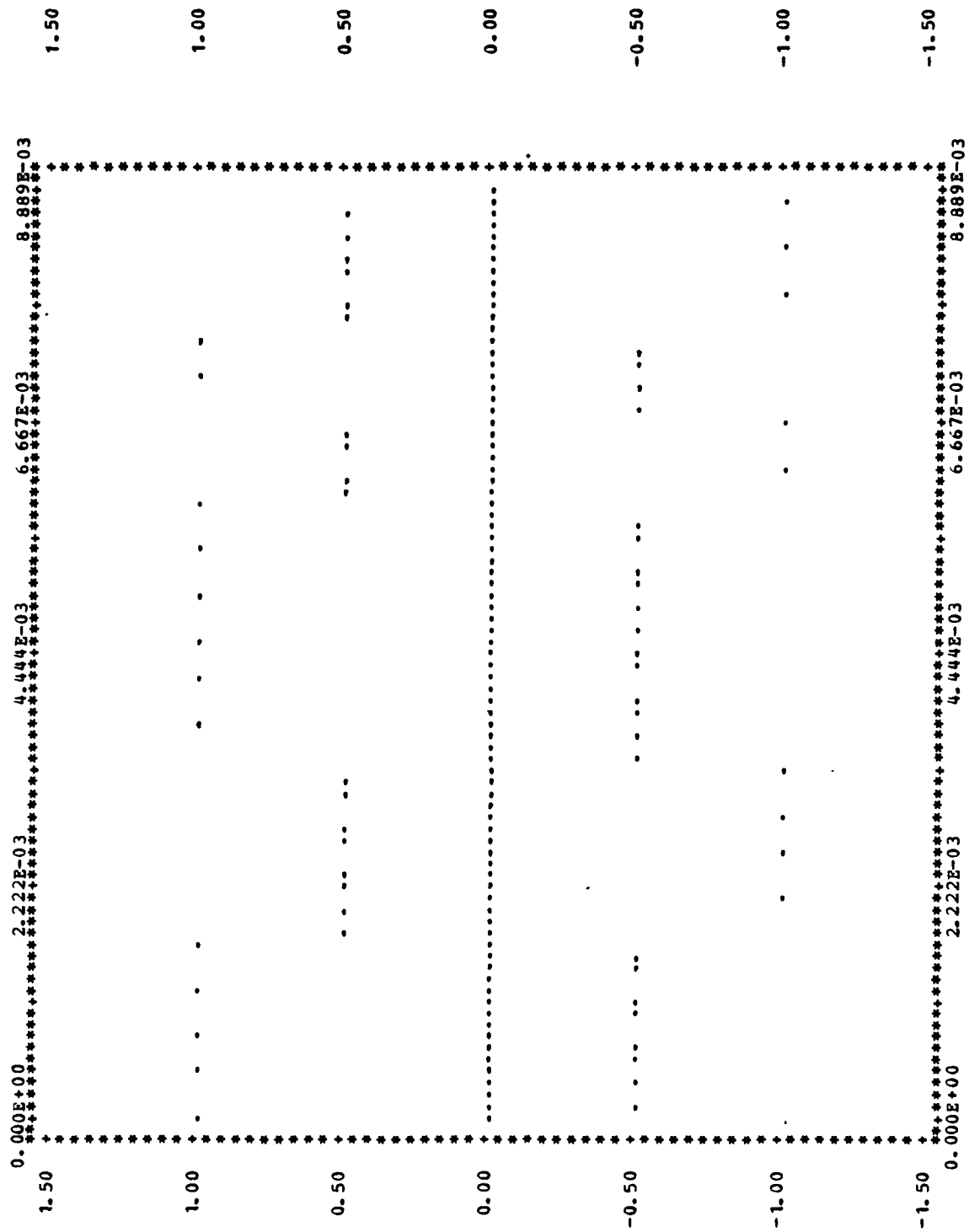
```

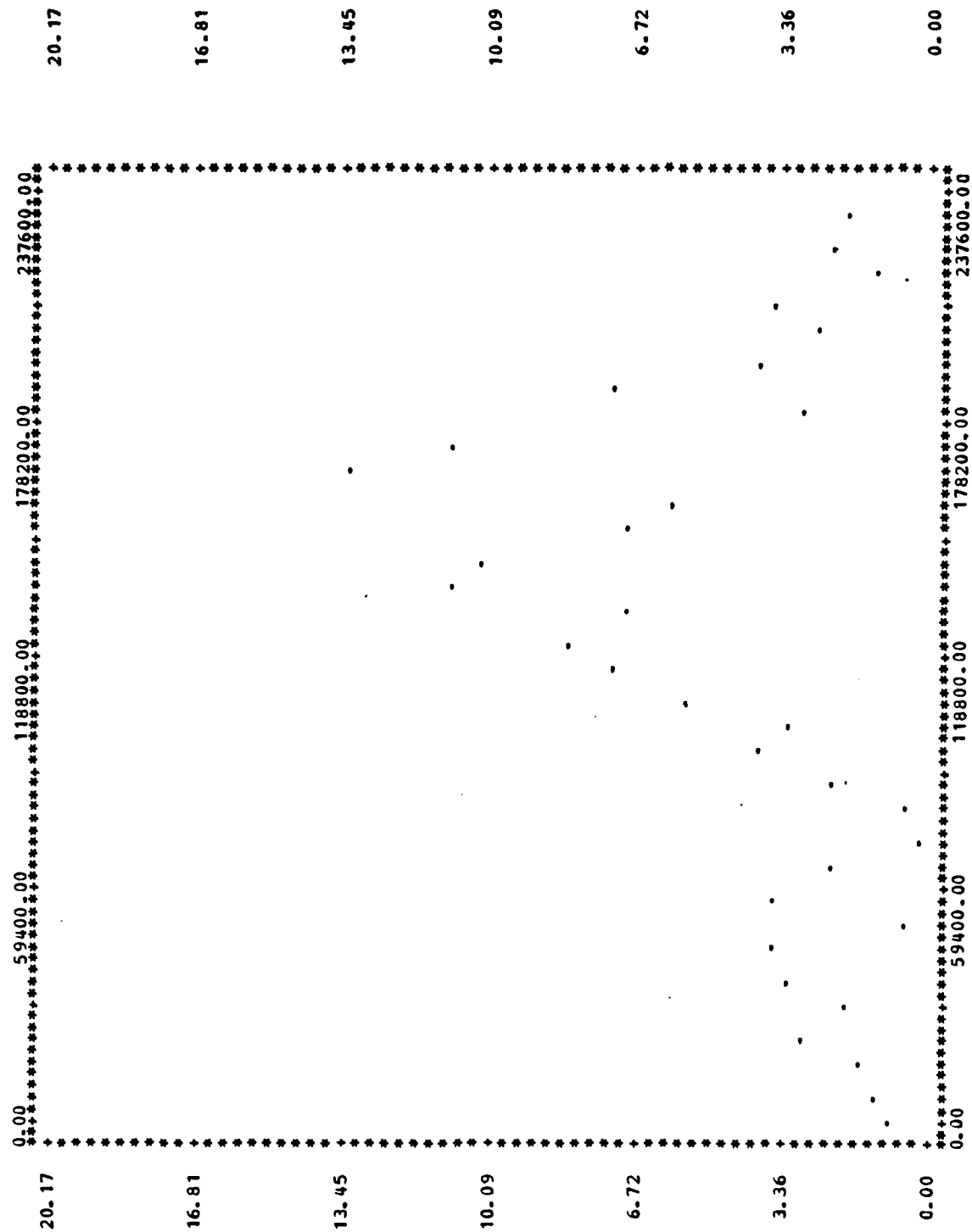




[illegible]

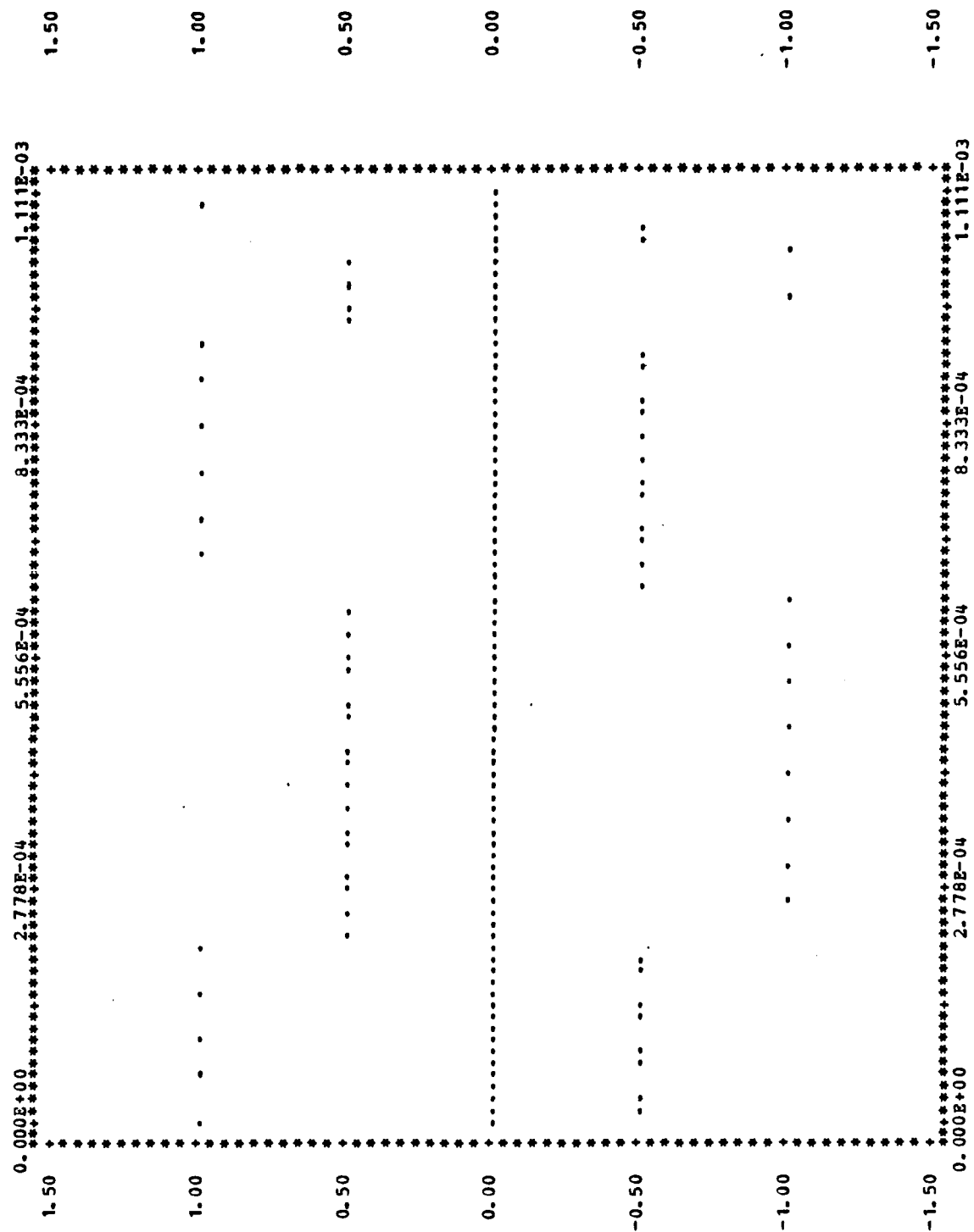
MODULATION TECHNIQUE = DBPSK
 BIFOIAR LOGIC
 BAUD OR SYMBOL RATE = 1200 HZ
 BITS PER BINARY CODE WORD = 1
 BIT RATE = 0.120000000000000E+04
 CARRIER FREQUENCY = 2400 HZ
 MAXIMUM CARRIER AMPLITUDE = 1 VOLT(S)
 INITIAL PHASE ANGLE = 0 DEGREES
 TIME BETWEEN SAMPLES = 0.138888888888889E-03 SEC
 NUMBER OF 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
 0.2025952553549290E+03 0.1880579703838077E+06

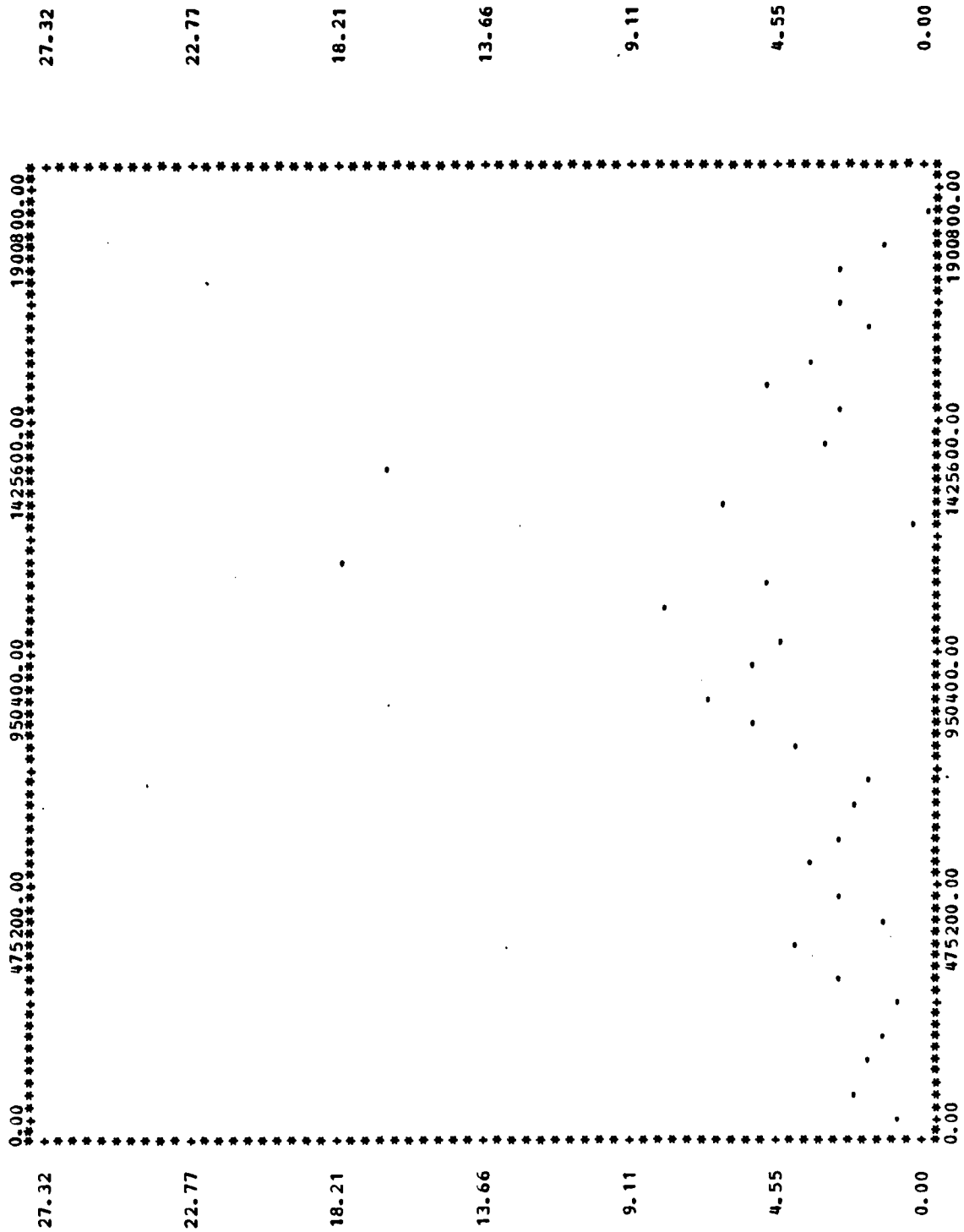




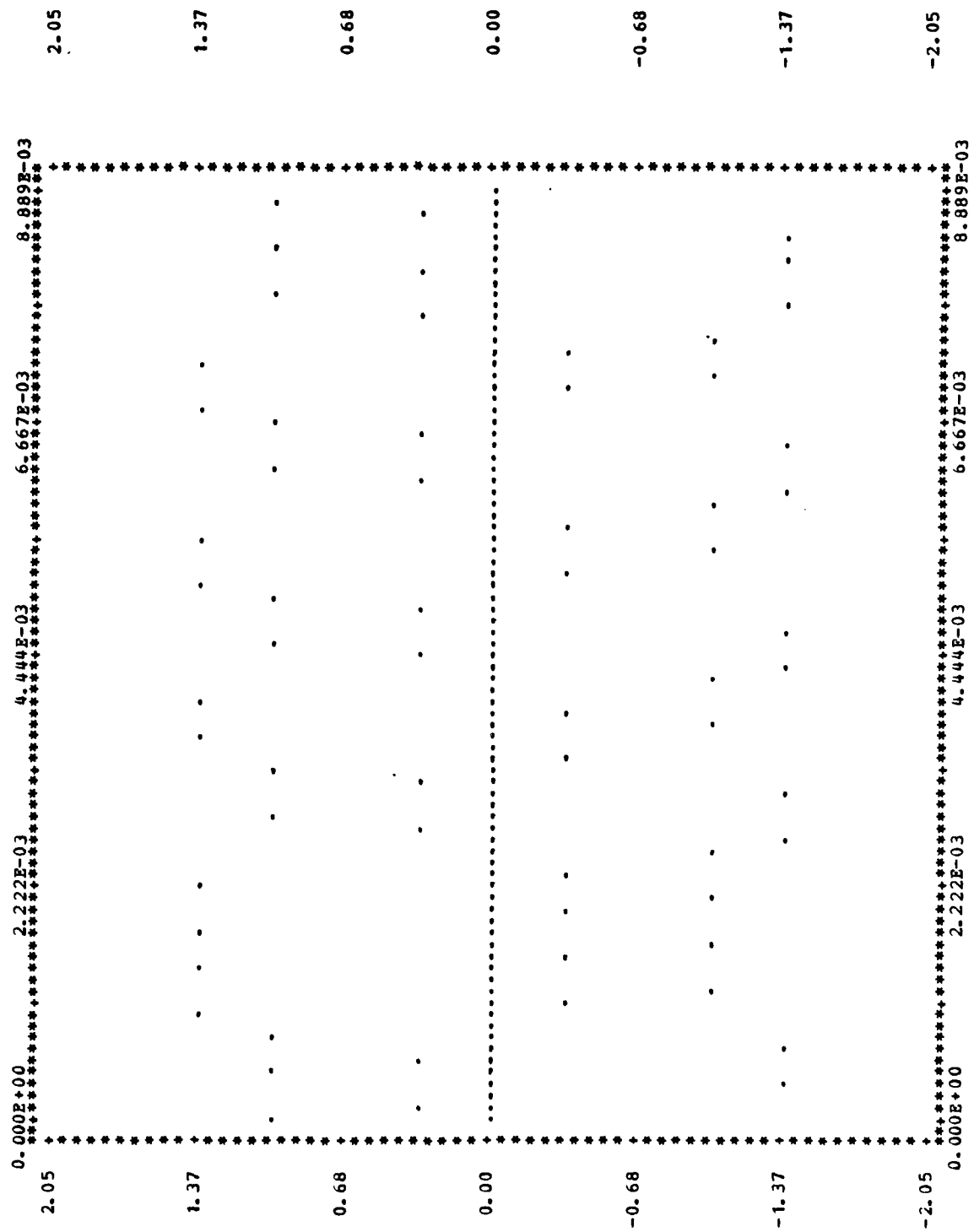
MODULATION TECHNIQUE = ORTHOGONAL BPSK
 BIFOLAR LOGIC
 BAUD OR SYMBOL RATE = 1200 HZ
 BITS PER BINARY CODE WORD = 3
 BIT RATE = 0.960000000000000E+04
 CARRIER FREQUENCY = 19200 HZ
 MAXIMUM CARRIER AMPLITUDE = 1 VOLT(S)
 INITIAL PHASE ANGLE = 0 DEGREES
 TIME BETWEEN SAMPLES = 0.2450980392156863E-04 SEC
 NUMBER OF SAMPLES GENERATED = 64
 SEED FOR RANDOM NUMBER GENERATOR = 1
 NUMBER OF TIMES SIMULATION REPEATS = 100

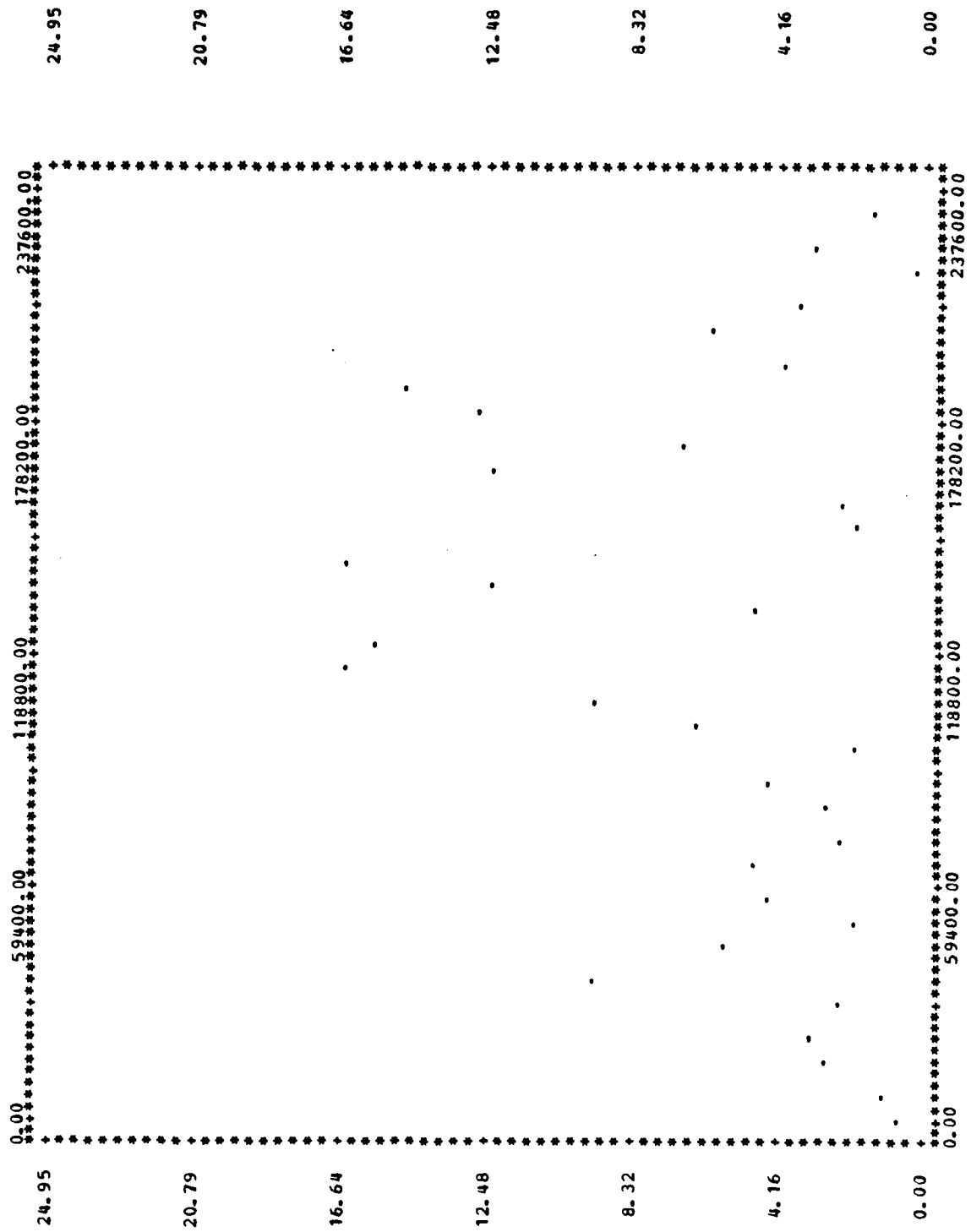
SUM OF VARIANCES	SUM OF VARIANCES**2
0.3565282677437312E+03	0.1043543362358570E+05
SUM OF SKEWNESS	SUM OF SKEWNESS**2
-0.5381895541377615E+04	0.3582631135045827E+07
SUM OF KURTOSIS	SUM OF KURTOSIS**2
0.3658999158877697E+05	0.3330606974397919E+09
MEAN VARIANCE	VARIANCE OF THE VARIANCES
0.1080388690132519E+02	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





MODULATION TECHNIQUE = QPSK
 BIFOLAR 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 = 0.1388888888888888E-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



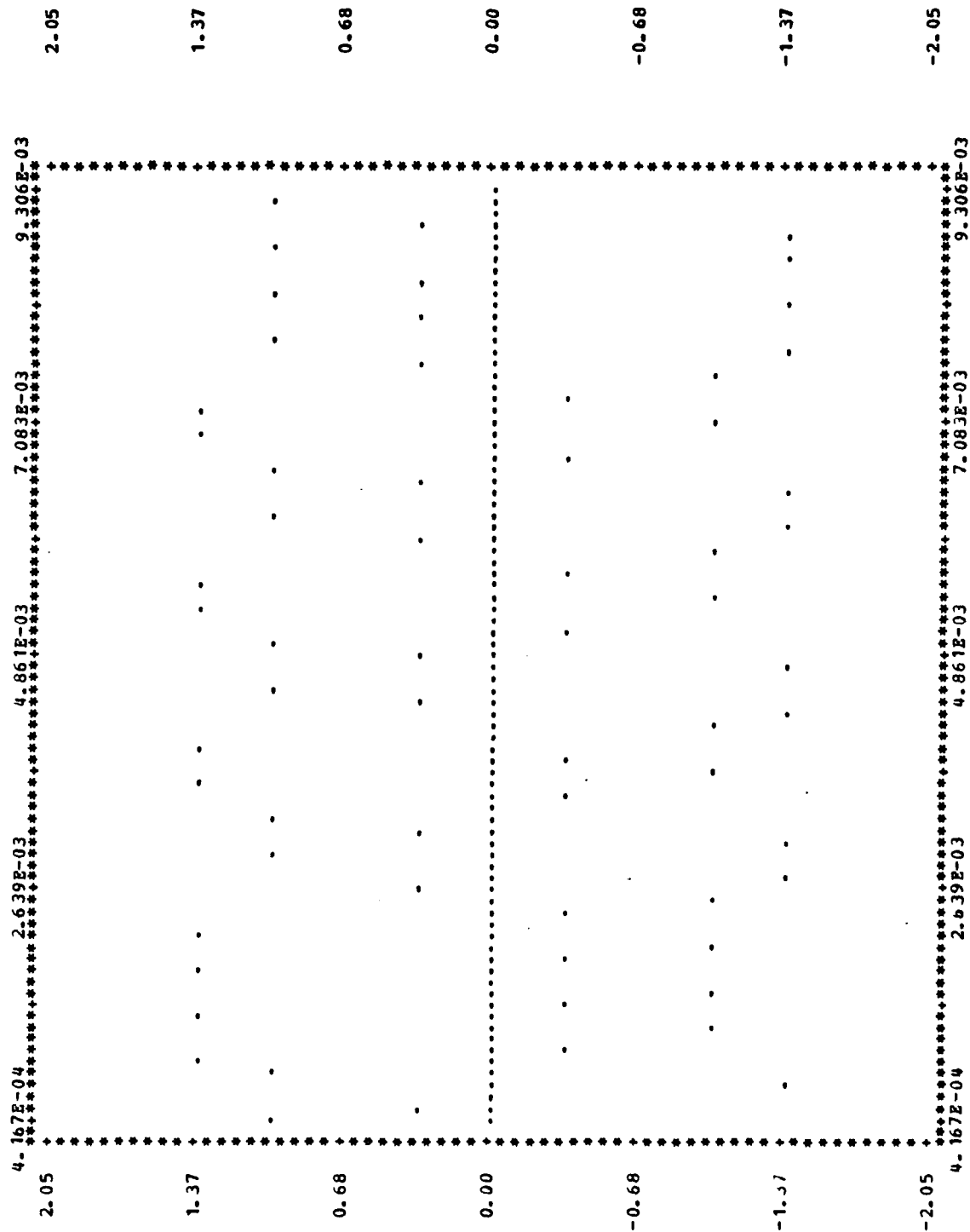


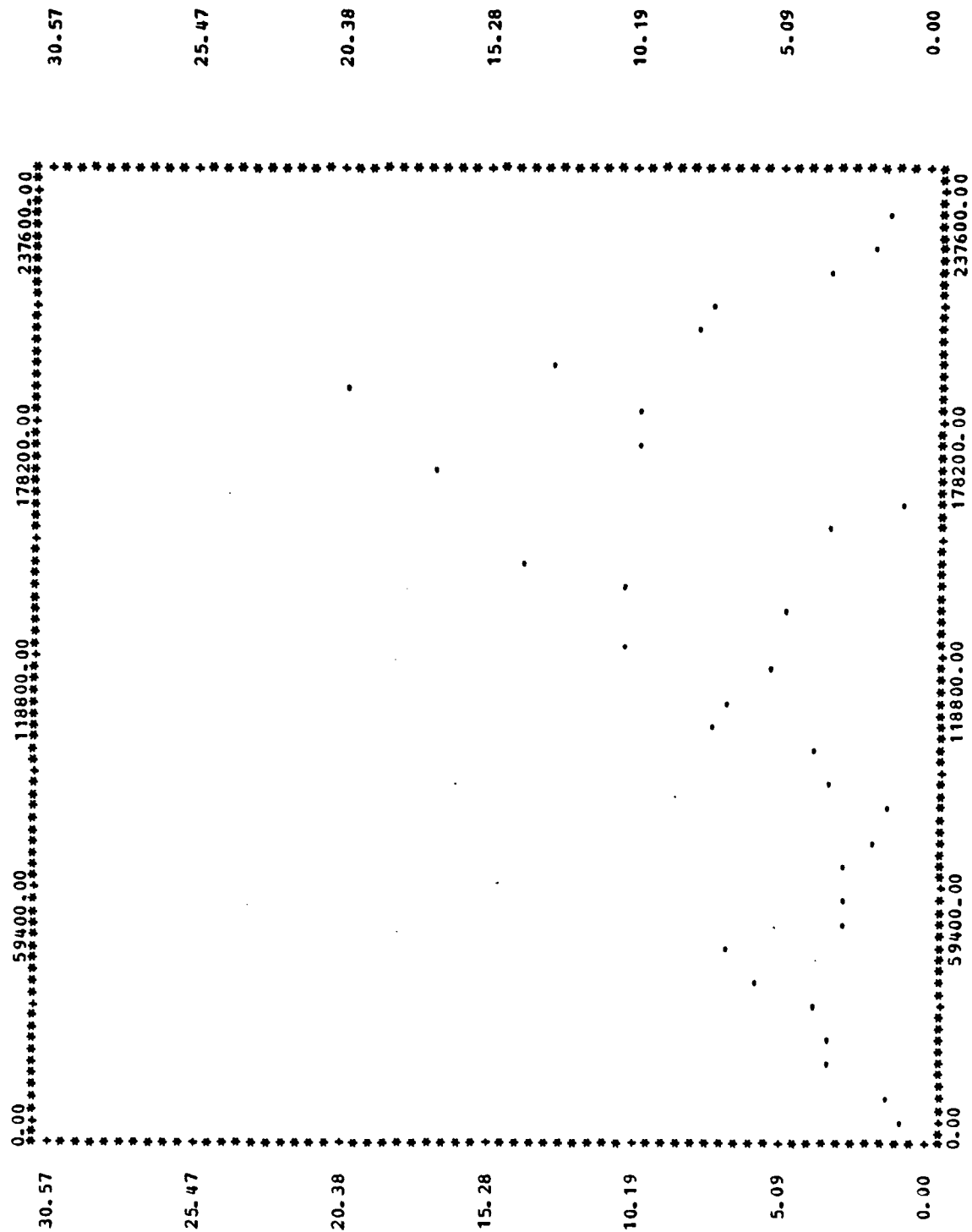
217

```

MODULATION TECHNIQUE = OQPSK
BIFOLAR LOGIC
BAUD OR SYMBOL RATE = 1200 HZ
BITS PER BINARY CODE WORD = 2
BIT RATE = 0.240000000000000E+04
CARRIER FREQUENCY = 2400 HZ
MAXIMUM CARRIER AMPLITUDE = 1 VOLT(S)
INITIAL PHASE ANGLE = 0 DEGREES
TIME BETWEEN SAMPLES = 0.138888888888888E-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

```

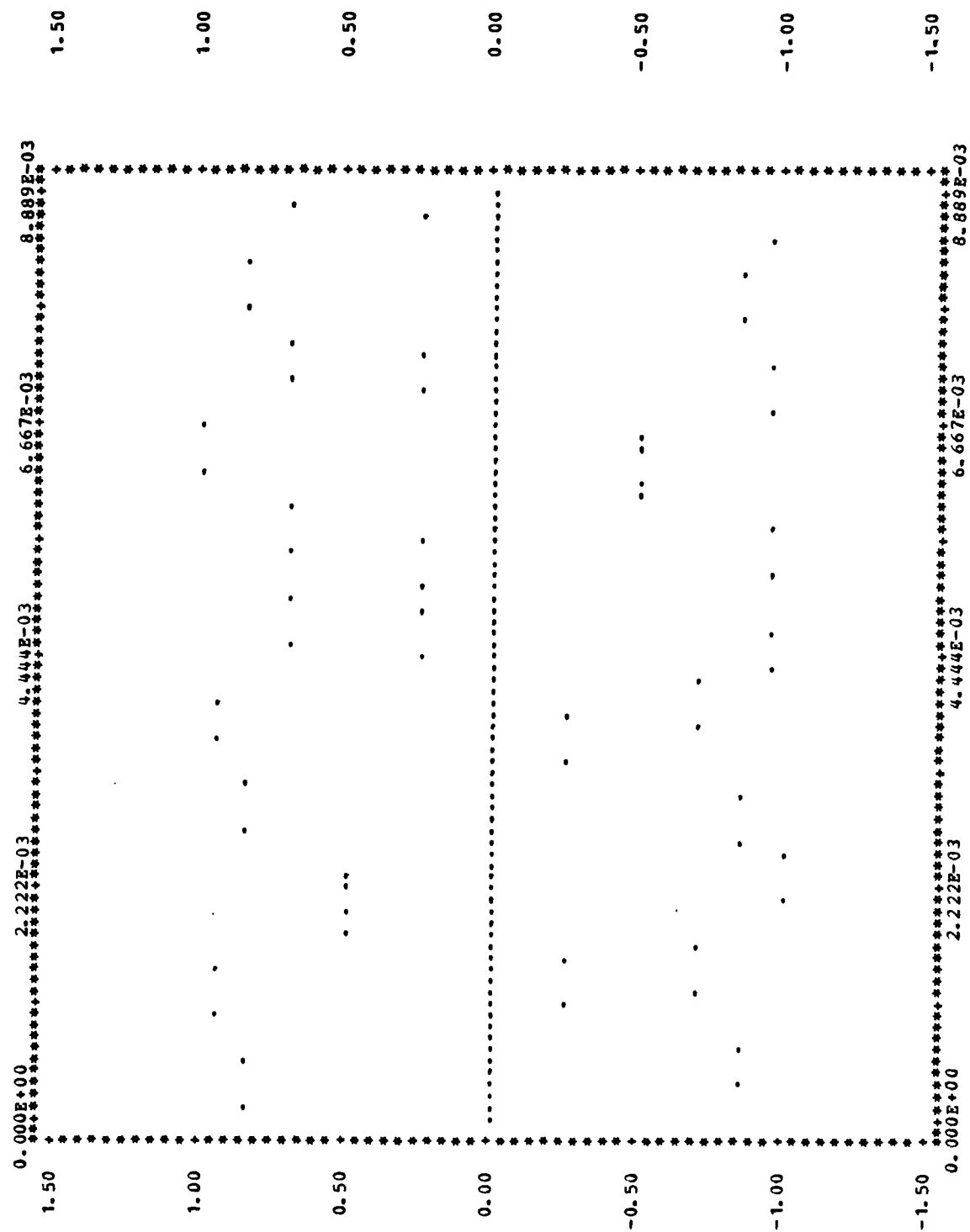


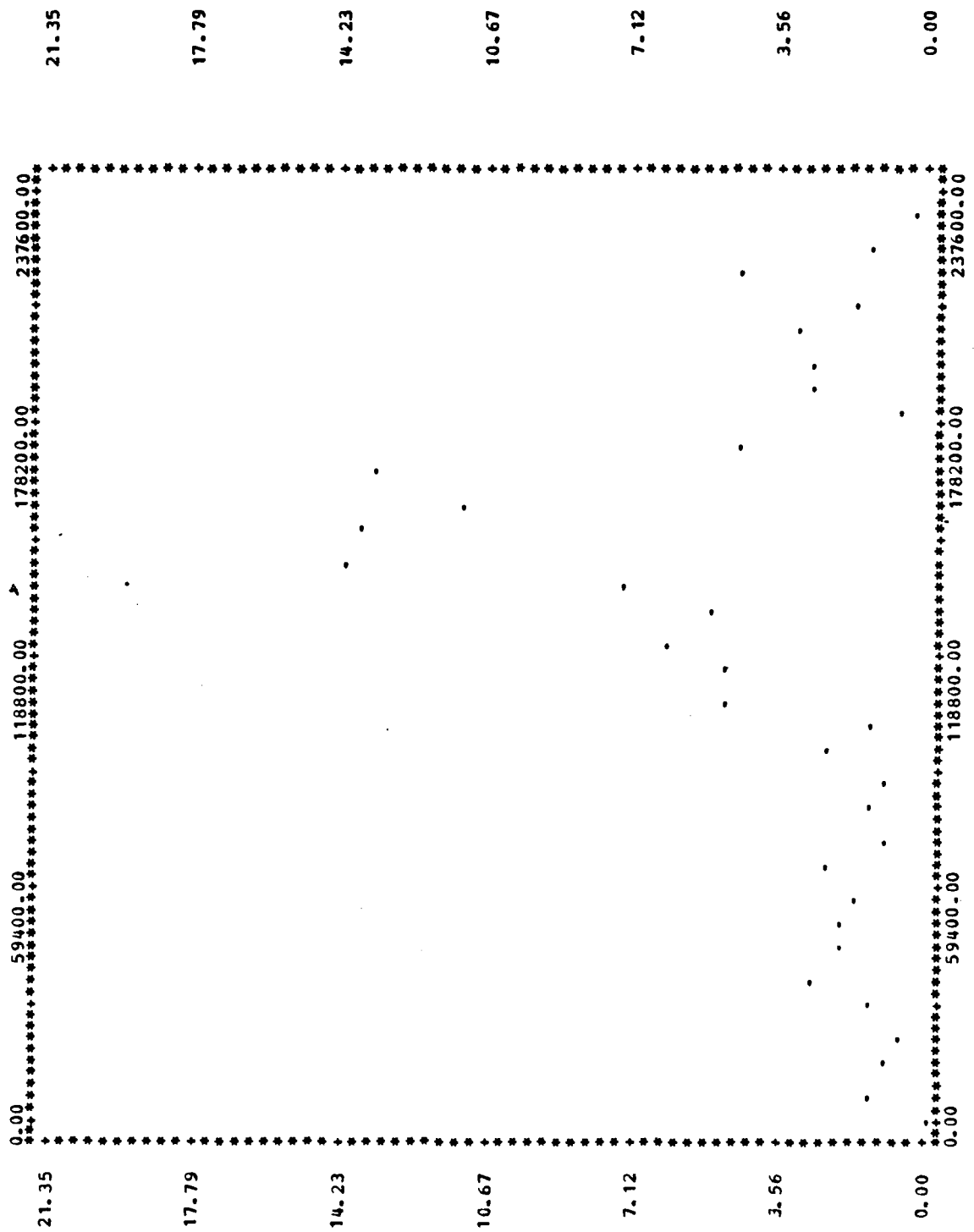


1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000	1001	1002	1003	1004	1005	1006	1007	1008	1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023	1024	1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035	1036	1037	1038	1039	1040	1041	1042	1043	1044	1045	1046	1047	1048	1049	1050	1051	1052	1053	1054	1055	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066	1067	1068	1069	1070	1071	1072	1073	1074	1075	1076	1077	1078	1079	1080	1081	1082	1083	1084	1085	1086	1087	1088	1089	1090	1091	1092	1093	1094	1095	1096	1097	1098	1099	1100	1101	1102	1103	1104	1105	1106	1107	1108	1109	1110	1111	1112	1113	1114	1115	1116	1117	1118	1119	1120	1121	1122	1123	1124	1125	1126	1127	1128	1129	1130	1131	1132	1133	1134	1135	1136	1137	1138	1139	1140	1141	1142	1143	1144	1145	1146	1147	1148	1149	1150	1151	1152	1153	1154	1155	1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167	1168	1169	1170	1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181	1182	1183	1184	1185	1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199	1200	1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211	1212	1213	1214	1215	1216	1217	1218	1219	1220	1221	1222	1223	1224	1225	1226	1227	1228	1229	1230	1231	1232	1233	1234	1235	1236	1237	1238	1239	1240	1241	1242	1243	1244	1245	1246	1247	1248	1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260	1261	1262	1263	1264	1265	1266	1267	1268	1269	1270	1271	1272	1273	1274	1275	1276	1277	1278	1279	1280	1281	1282	1283	1284	1285	1286	1287	1288	1289	1290	1291	1292	1293	1294	1295	1296	1297	1298	1299	1300	1301	1302	1303	1304	1305	1306	1307	1308	1309	1310	1311	1312	1313	1314	1315	1316	1317	1318	1319	1320	1321	1322	1323	1324	1325	1326	1327	1328	1329	1330	1331	1332	1333	1334	1335	1336	1337	1338	1339	1340	1341	1342	1343	1344	1345	1346	1347	1348	1349	1350	1351	1352	1353	1354	1355	1356	1357	1358	1359	1360	1361	1362	1363	1364	1365	1366	1367	1368	1369	1370	1371	1372	1373	1374	1375	1376	1377	1378	1379	1380	1381	1382	1383	1384	1385	1386	1387	1388	1389	1390	1391	1392	1393	1394	1395	1396	1397	1398	1399	1400	1401	1402	1403	1404	1405	1406	1407	1408	1409	1410	1411	1412	1413	1414	1415	1416	1417	1418	1419	1420	1421	1422	1423	1424	1425	1426	1427	1428	1429	1430	1431	1432	1433	1434	1435	1436	1437	1438	1439	1440	1441	1442	1443	1444	1445	1446	1447	1448	1449	1450	1451	1452	1453	1454	1455	1456	1457	1458	1459	1460	1461	1462	1463	1464	1465	1466	1467	1468	1469	1470	1471	1472	1473	1474	1475	1476	1477	1478	1479	1480	1481	1482	1483	1484	1485	1486	1487	1488	1489	1490	1491	1492	1493	1494	1495

N 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33

MODULATION TECHNIQUE = MPSK
 BIFOLAR 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 = 0.138888888888889E-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
 -0.960095280888690E+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

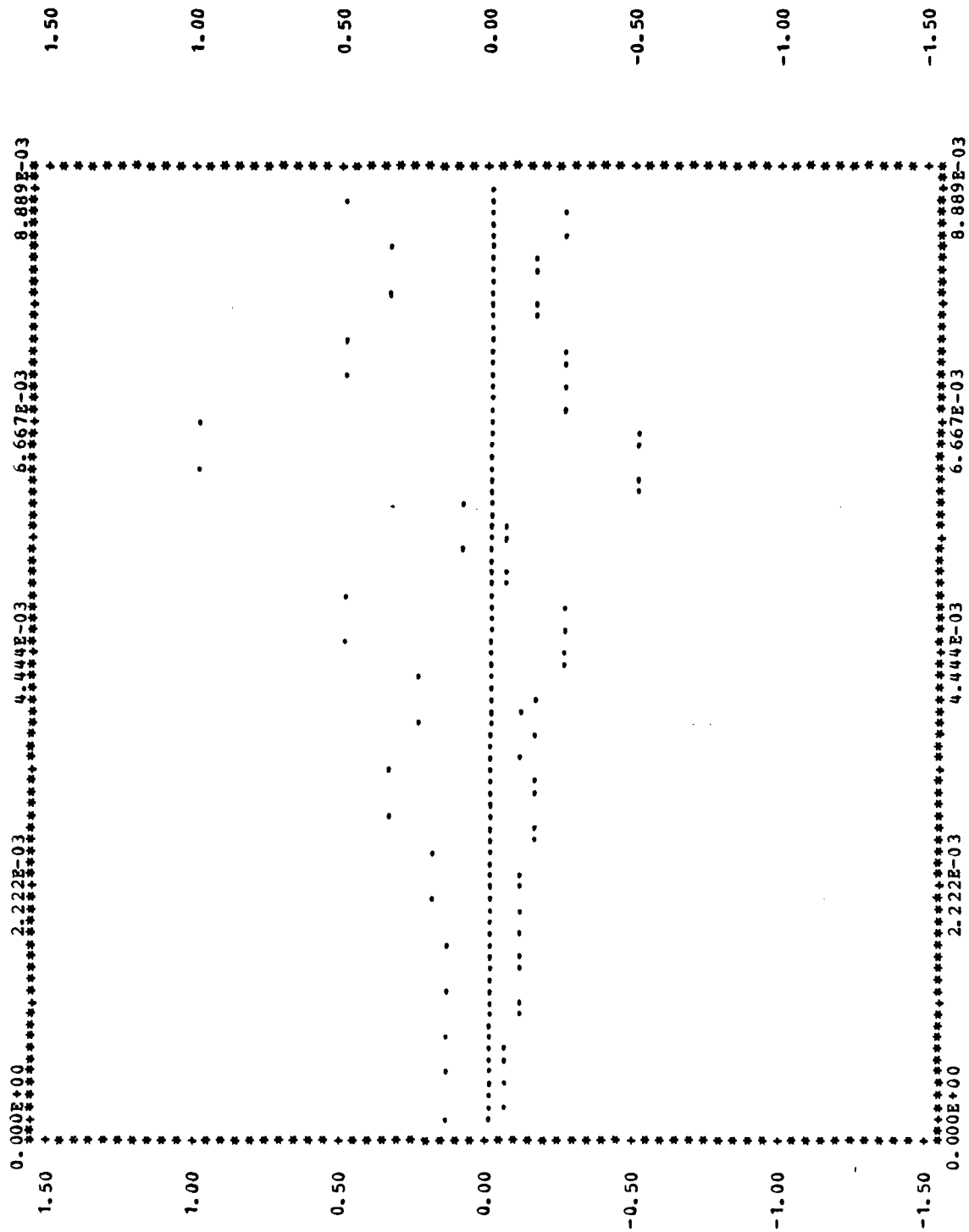


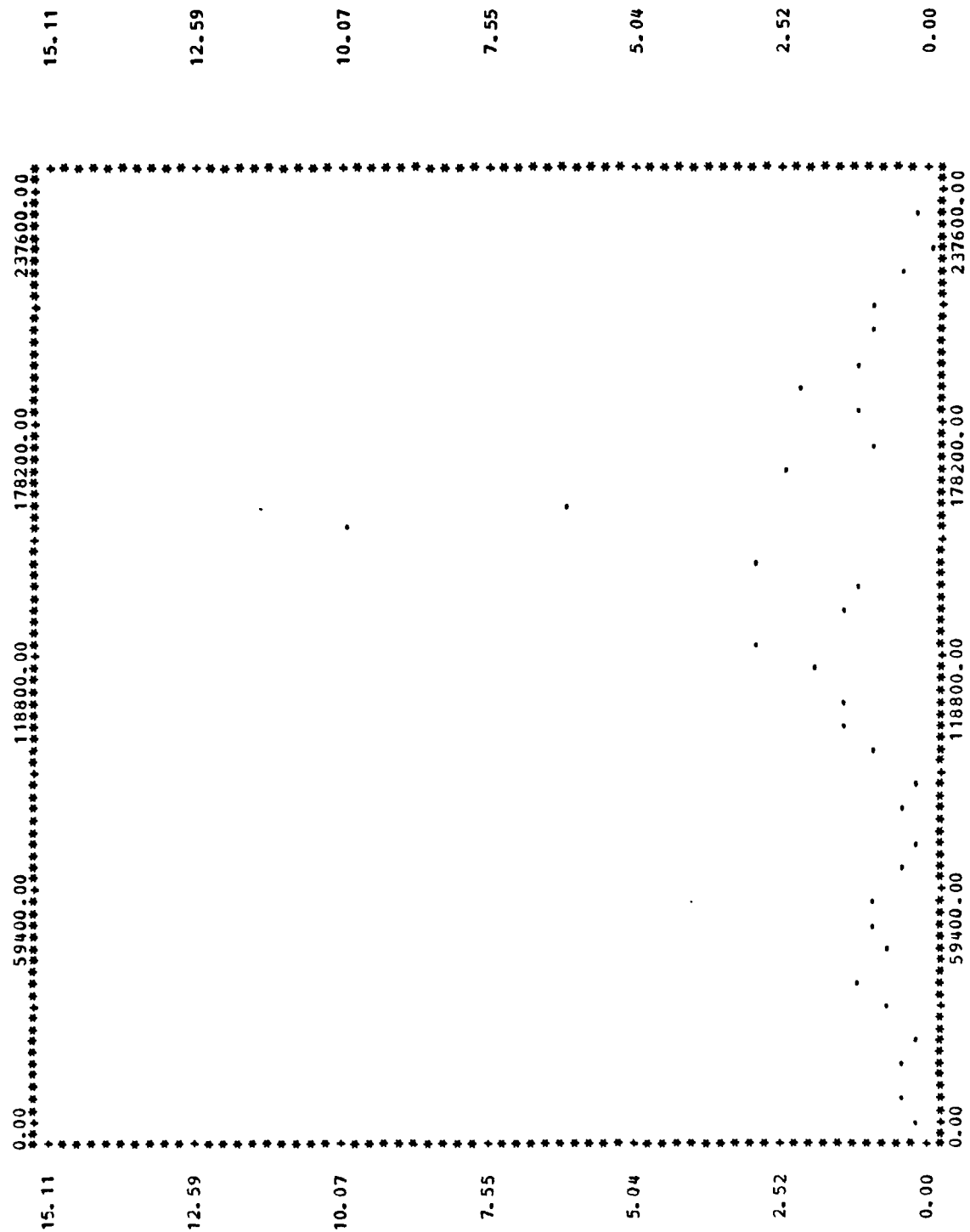


N 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33

MODULATION TECHNIQUE = MASK
 BIFOLAR LOGIC
 BAUD OR SYMBOL RATE = 1200 HZ
 BITS PER BINARY CODE WORD = 3
 BIT RATE = 0.360000000000000E+04
 CARRIER FREQUENCY = 2400 HZ
 MAXIMUM CARRIER AMPLITUDE = 1 VOLT(S)
 INITIAL PHASE ANGLE = 0 DEGREES
 TIME BETWEEN SAMPLES = 0.138888888888889E-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
0.2672011399771625E+02	0.6236464612091026E+02
SUM OF SKEWNESS	SUM OF SKEWNESS**2
-0.165776085594905E+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

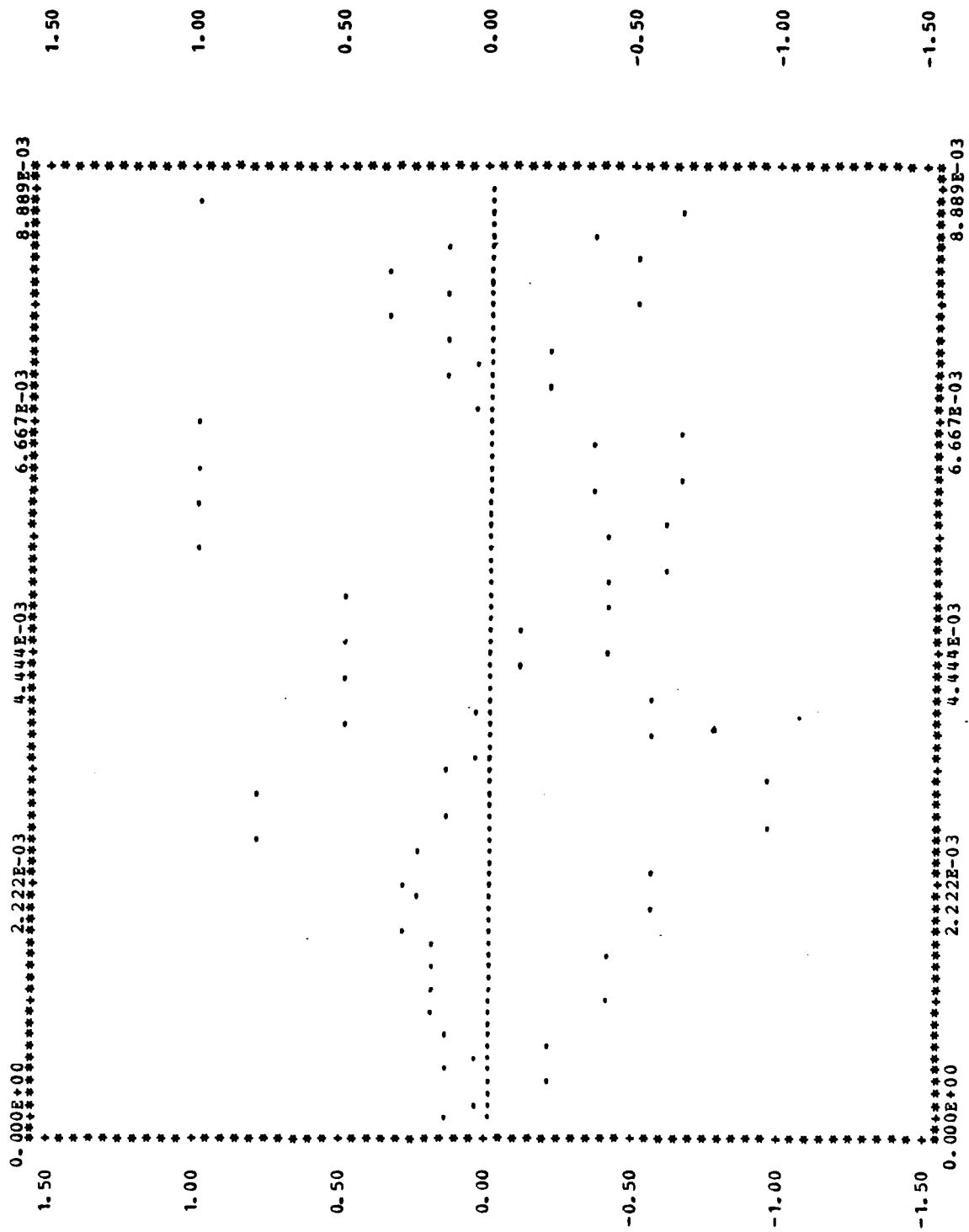


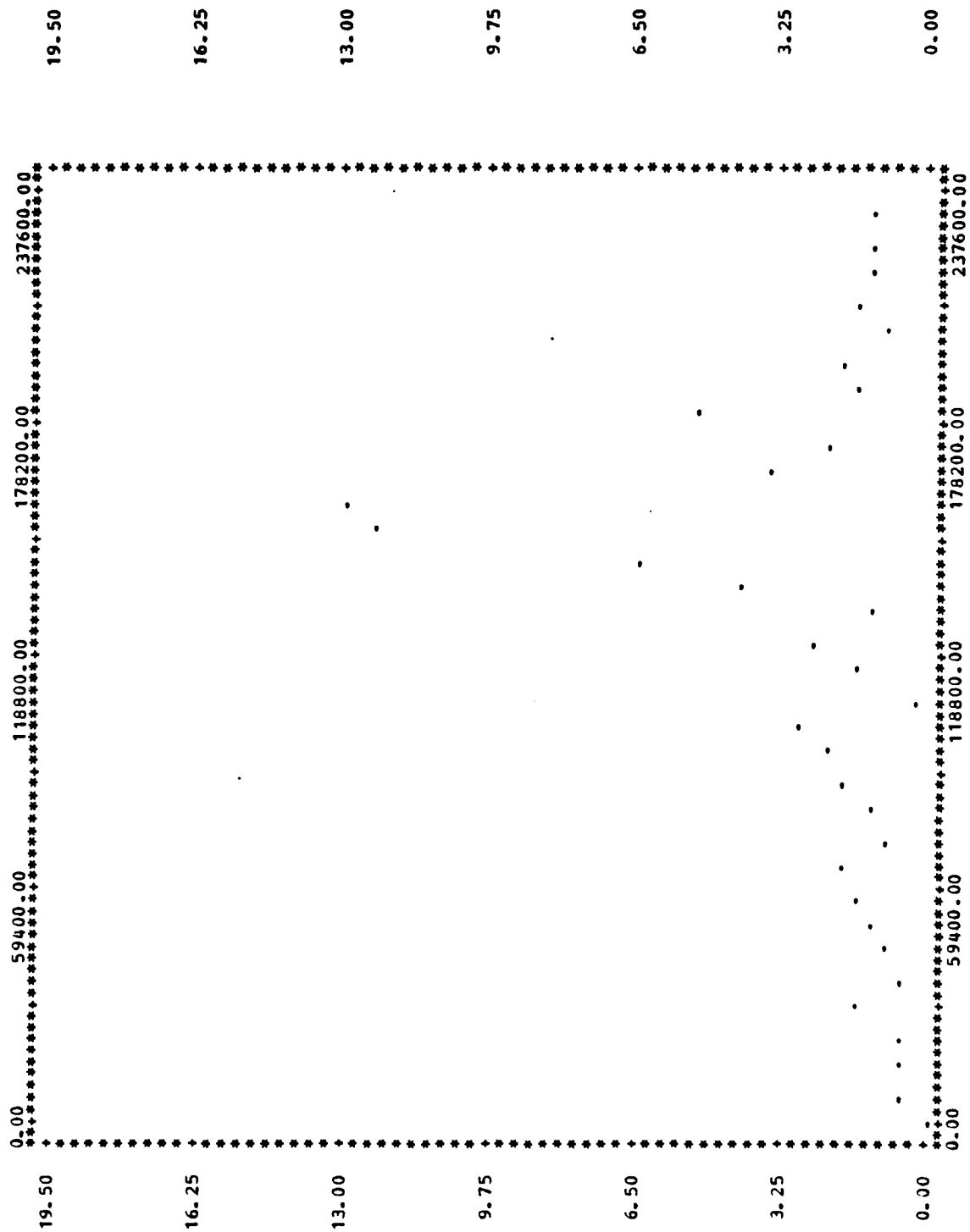


[illegible]

[illegible]

MODULATION TECHNIQUE = QASK
 BIFOLAR 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 = 0.138888888888889E-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
 -0.5438333348597472E+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

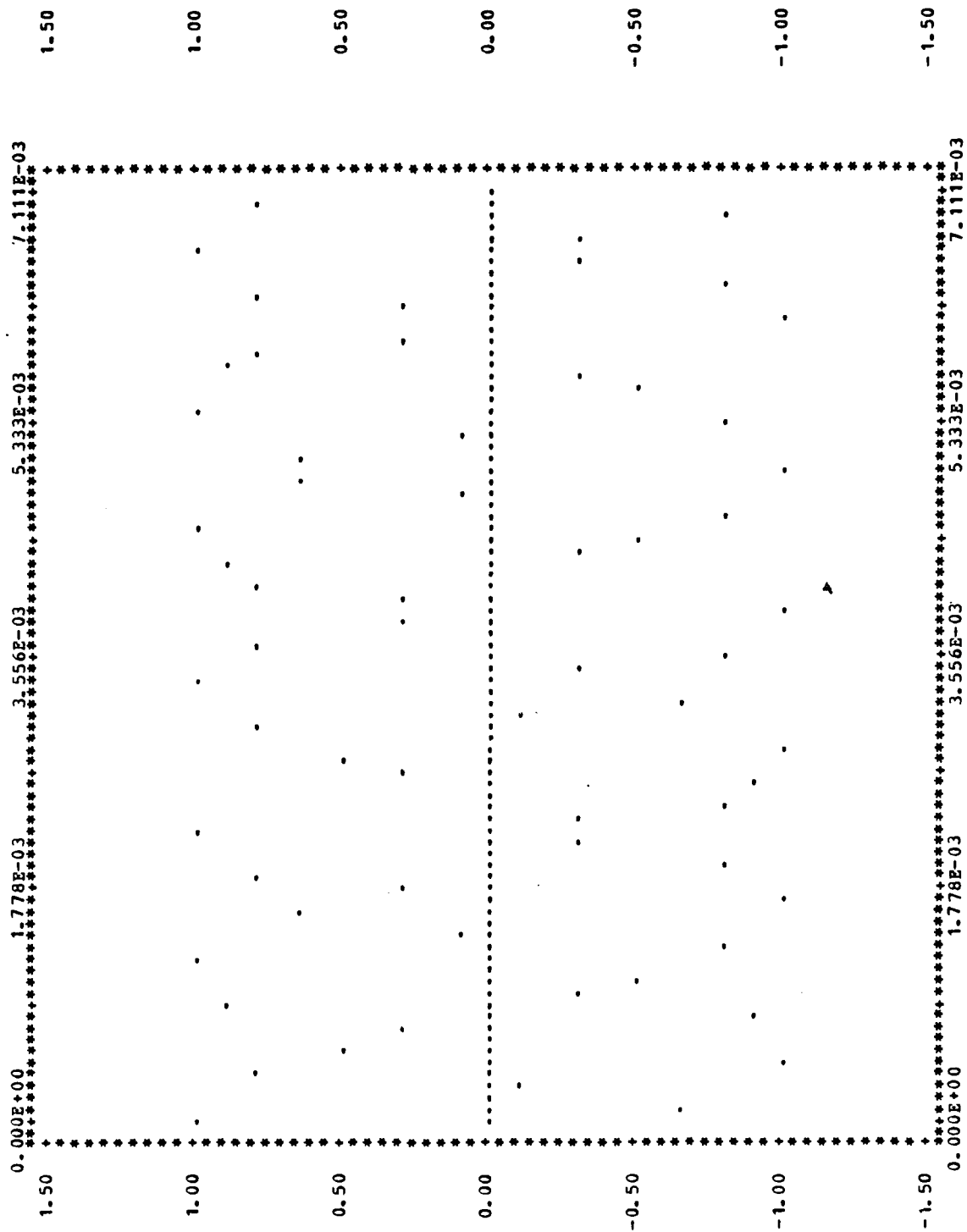


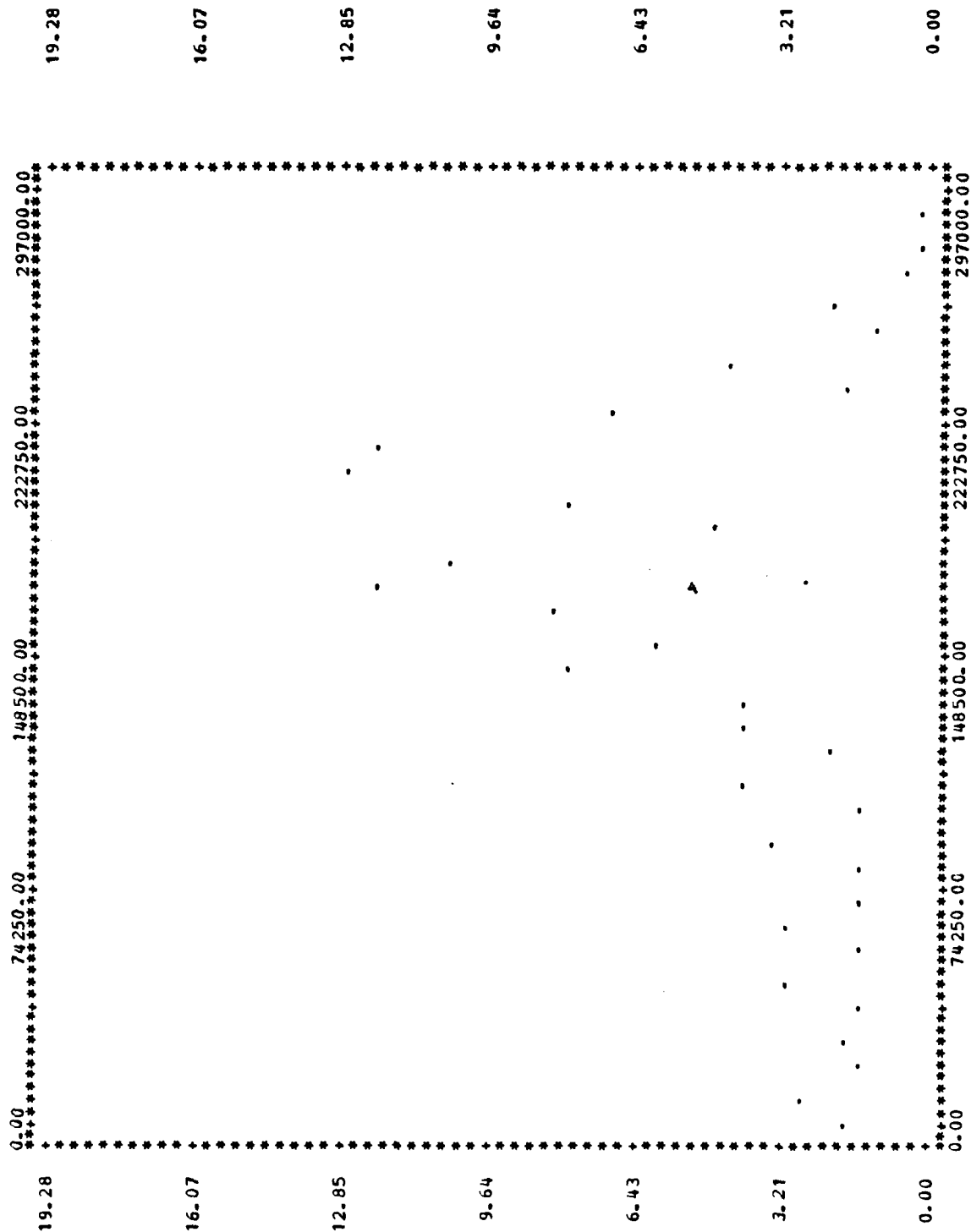


237

MODULATION TECHNIQUE = MSK
 BIFOLAR LOGIC
 BAUD OR SYMBOL RATE = 1200 HZ
 BITS PER BINARY CODE WORD = 1
 BIT RATE = 0.120000000000000E+04
 CARRIER FREQUENCY = 3000 HZ
 MAXIMUM CARRIER AMPLITUDE = 1 VOLT(S)
 INITIAL PHASE ANGLE = 0 DEGREES
 TIME BETWEEN SAMPLES = 0.111111111111111E-03 SEC
 NUMBER OF SAMPLES GENERATED = 64
 SEED FOR RANDOM NUMBER GENERATOR = 1
 NUMBER OF TIMES SIMULATION REPEATS = 100

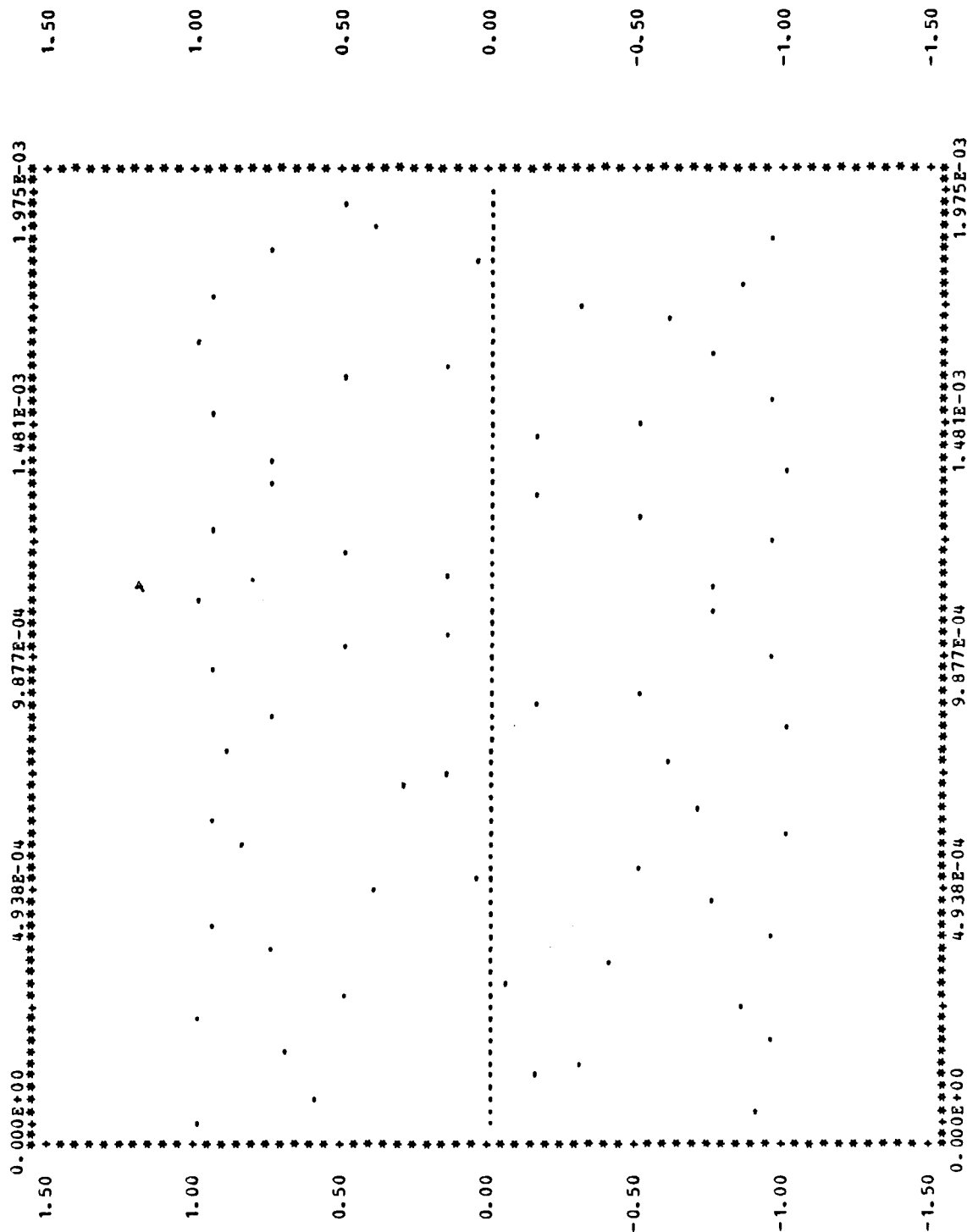
SUM OF VARIANCES	SUM OF VARIANCES**2
0.1525890150844788E+03	0.2084564025759793E+04
SUM OF SKEWNESS	SUM OF SKEWNESS**2
-0.1665578258850635E+05	0.9704853819233804E+08
SUM OF KURTOSIS	SUM OF KURTOSIS**2
0.5423666237610907E+04	0.6118691548293858E+07
MEAN VARIANCE	VARIANCE OF THE VARIANCES
0.4623909548014508E+01	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

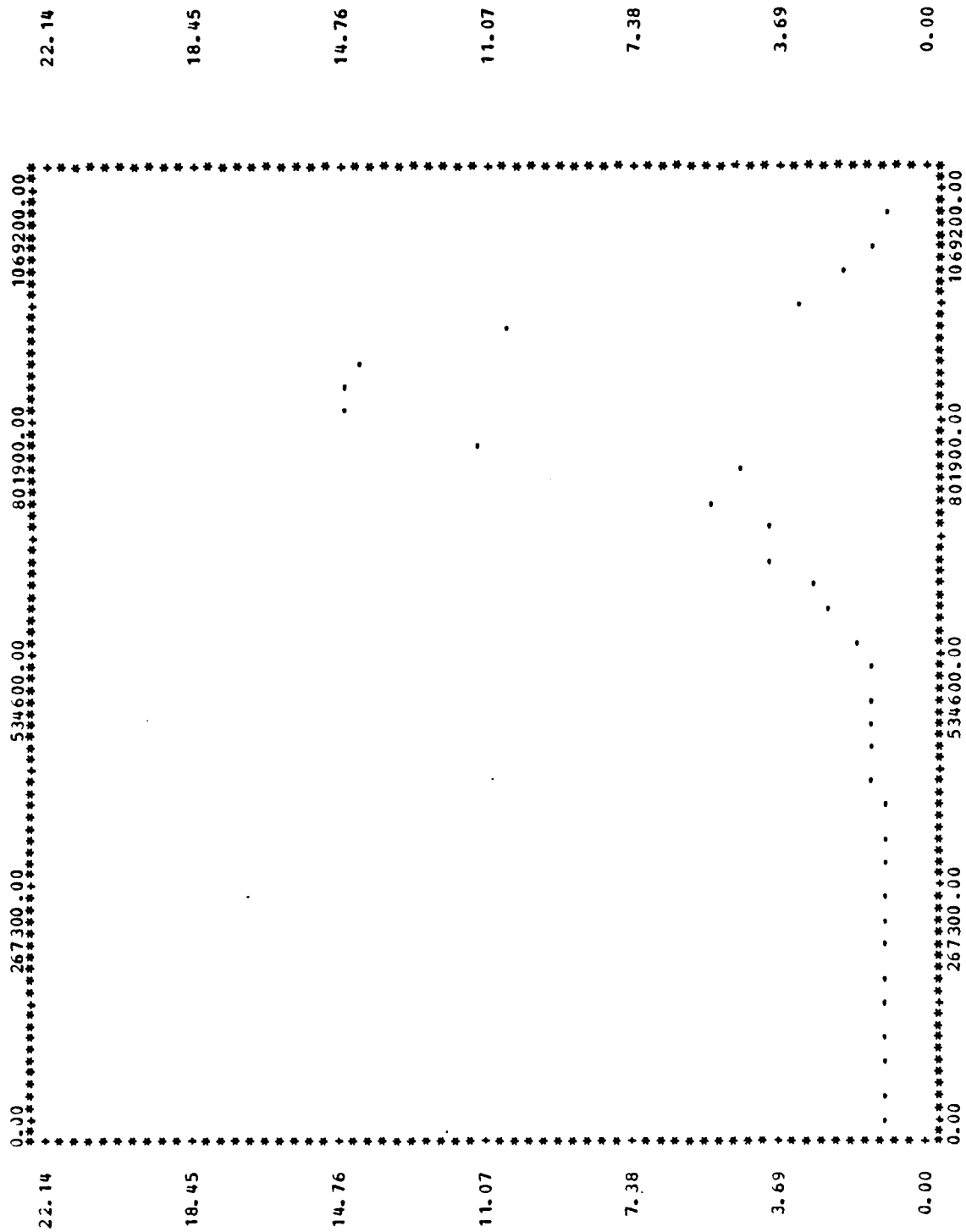




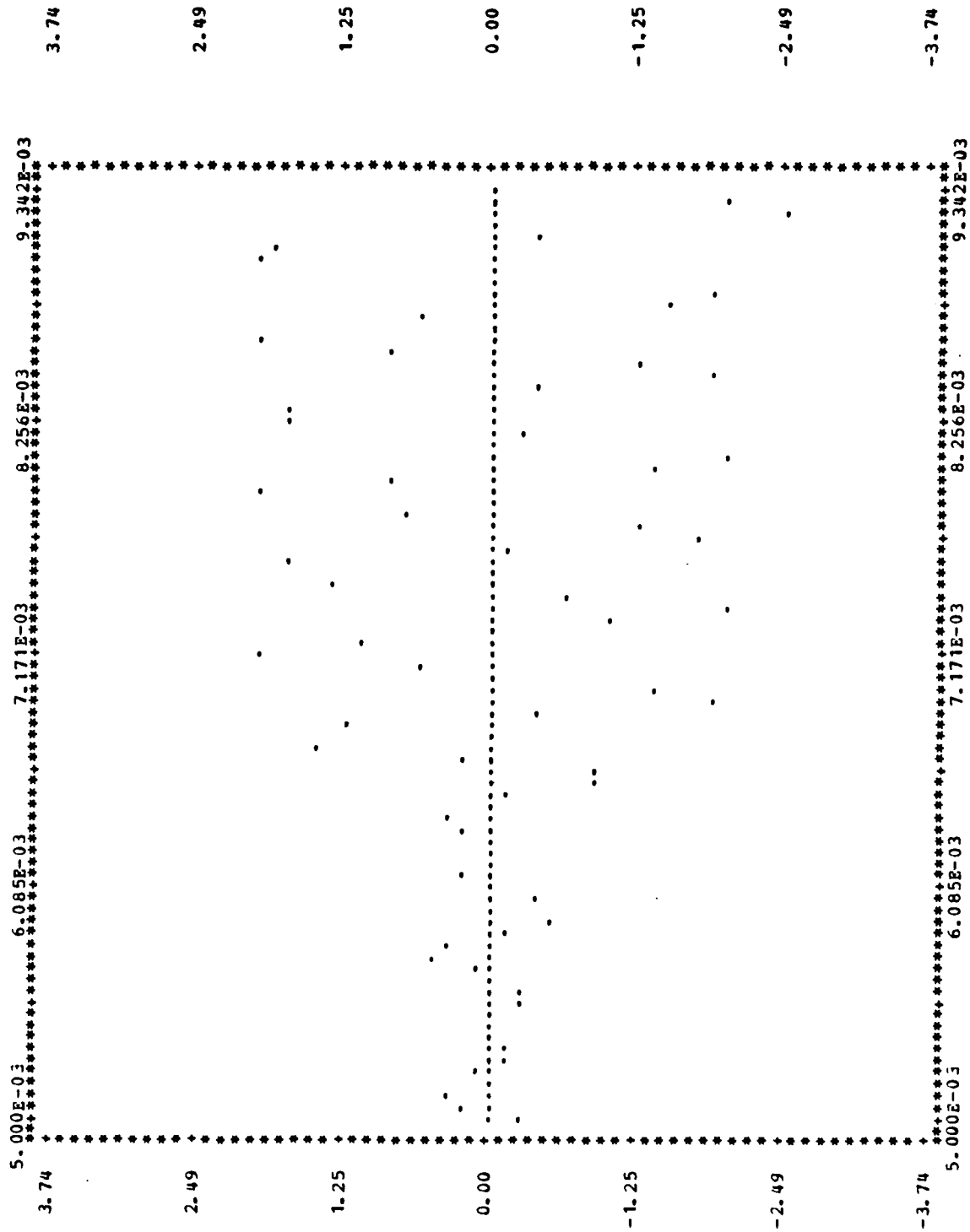
MODULATION TECHNIQUE = MFSK
 BIPOLAR LOGIC
 BAUD OR SYMBOL RATE = 1200 HZ
 BITS PER BINARY CODE WORD = 3
 BIT RATE = 0.3600000000000000E+04
 CARRIER FREQUENCY = 10800 HZ
 MAXIMUM CARRIER AMPLITUDE = 1 VOLT(S)
 INITIAL PHASE ANGLE = 0 DEGREES
 TIME BETWEEN SAMPLES = 0.3086419753086419E-04 SEC
 NUMBER OF SAMPLES GENERATED = 64
 SEED FOR RANDOM NUMBER GENERATOR = 1
 NUMBER OF TIMES SIMULATION REPEATS = 100

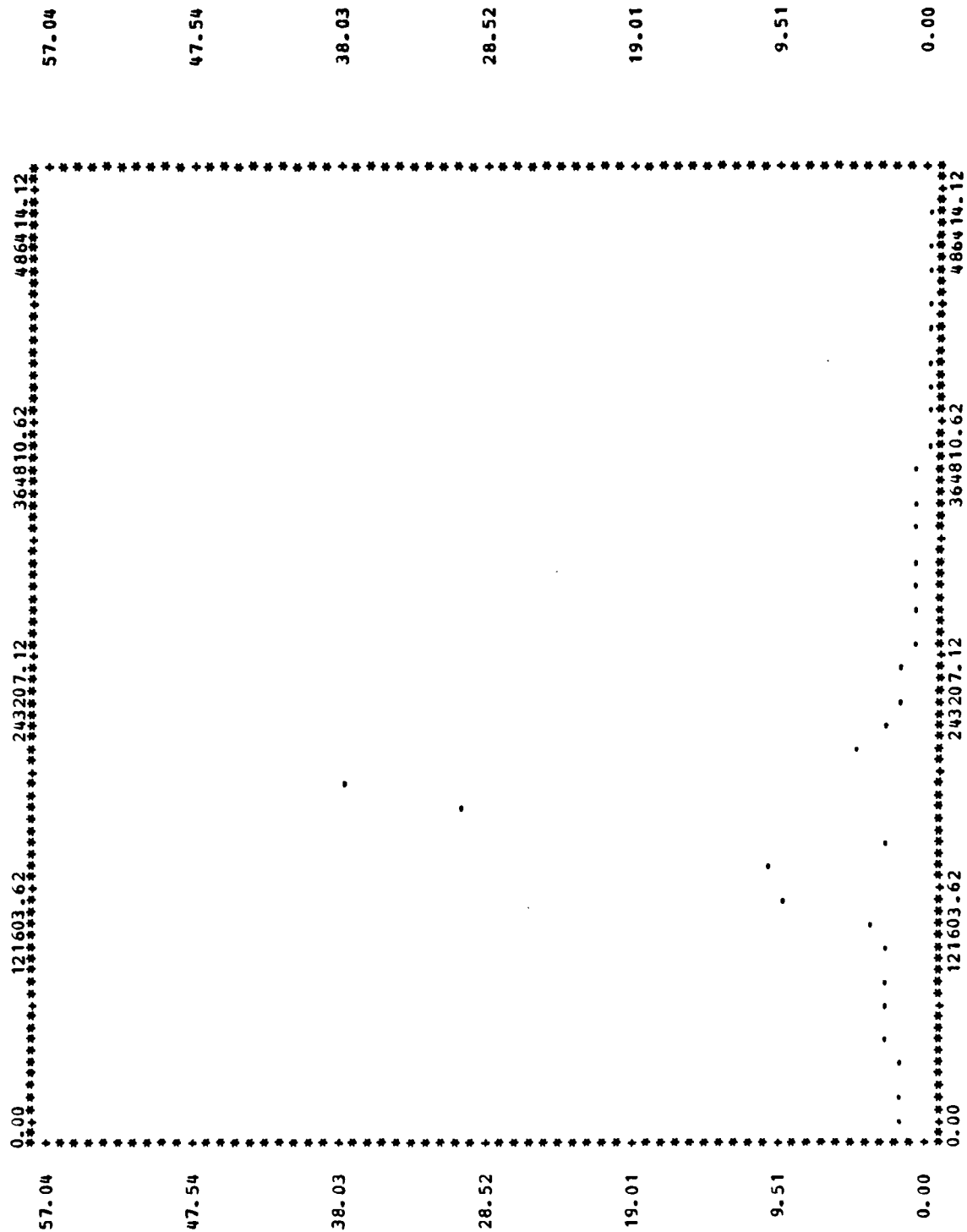
SUM OF VARIANCES	SUM OF VARIANCES**2
0.4806181141925261E+03	0.1352273679574422E+05
SUM OF SKEWNESS	SUM OF SKEWNESS**2
-0.1654247621695240E+04	0.5890267815807785E+06
SUM OF KURTOSIS	SUM OF KURTOSIS**2
0.5341287673934422E+05	0.2790612658473240E+09
MEAN VARIANCE	VARIANCE OF THE VARIANCES
0.1456418527856140E+02	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





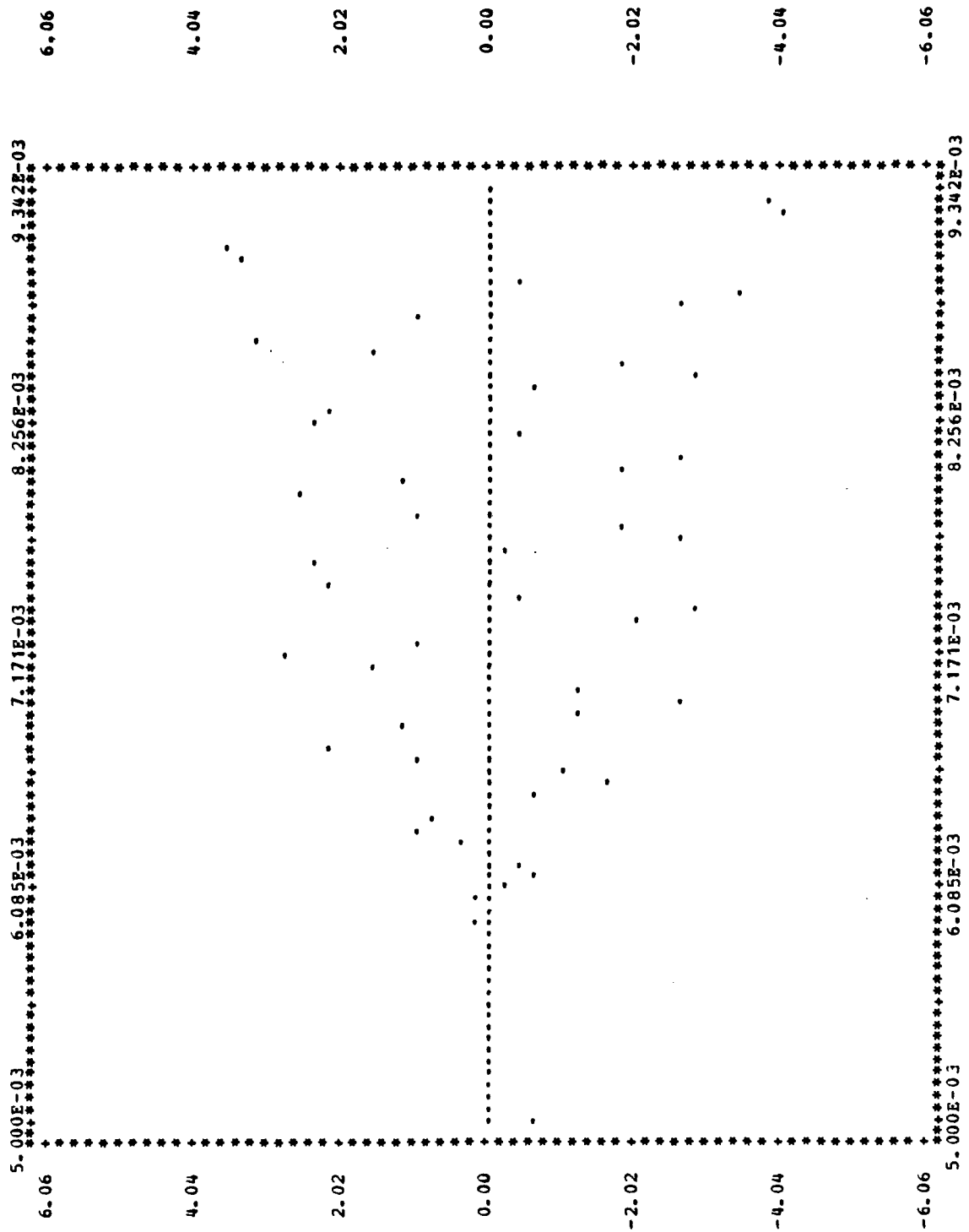
MODULATION TECHNIQUE = QPRS
 QPRS CLASS 1 FILTER
 BIFOLAR LOGIC
 BAUD OR SYMBOL RATE = 1200 HZ
 BITS PER BINARY CODE WORD = 2
 BIT RATE = 0.2400000000000000E+04
 CARRIER FREQUENCY = 2400 HZ
 MAXIMUM CARRIER AMPLITUDE = 1 VOLT(S)
 INITIAL PHASE ANGLE = 0 DEGREES
 TIME BETWEEN SAMPLES = 0.6784342273548912E-04 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

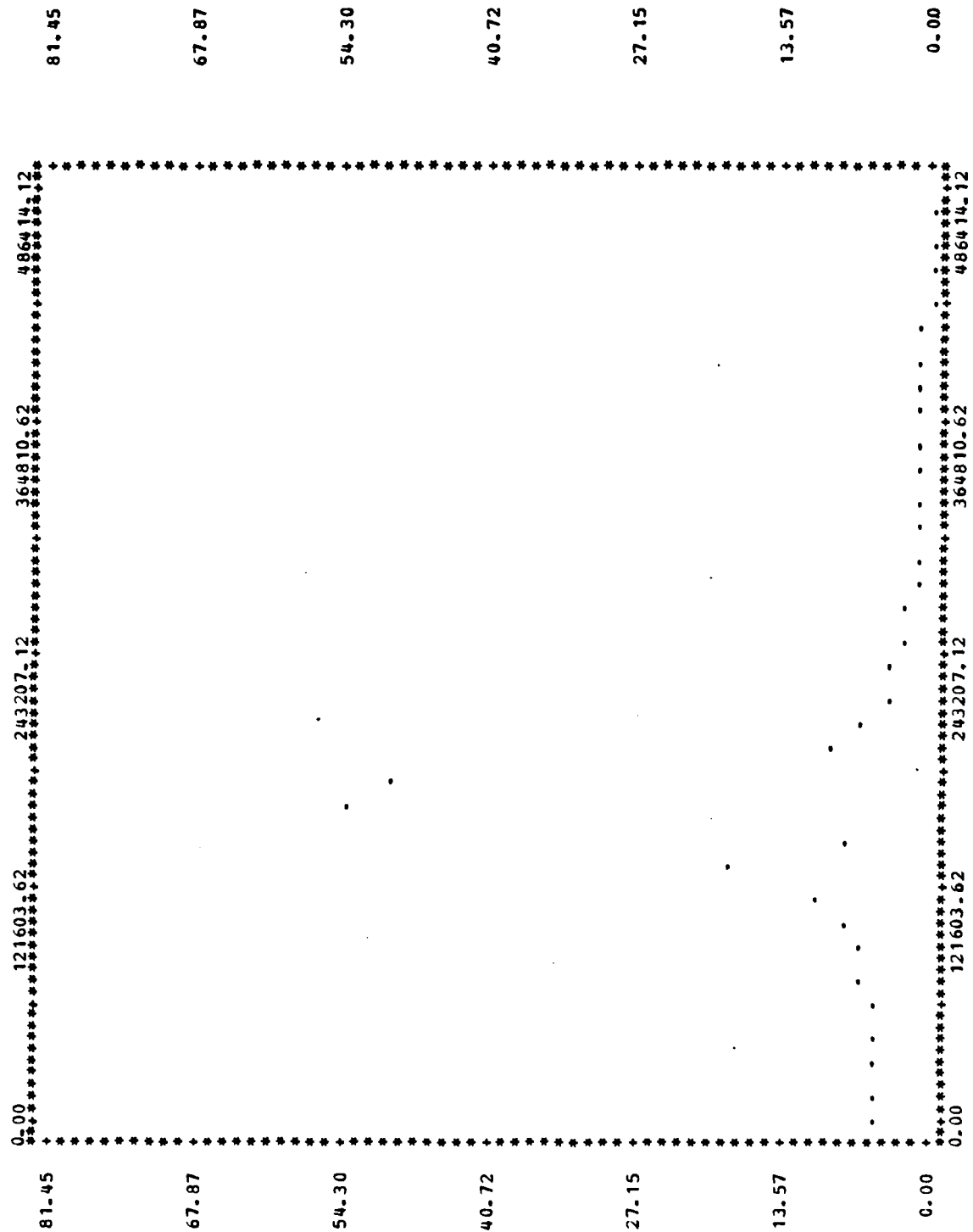




[illegible]

MODULATION TECHNIQUE = QPRS
 QPRS CLASS 2 FILTER
 BIFOLAR 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

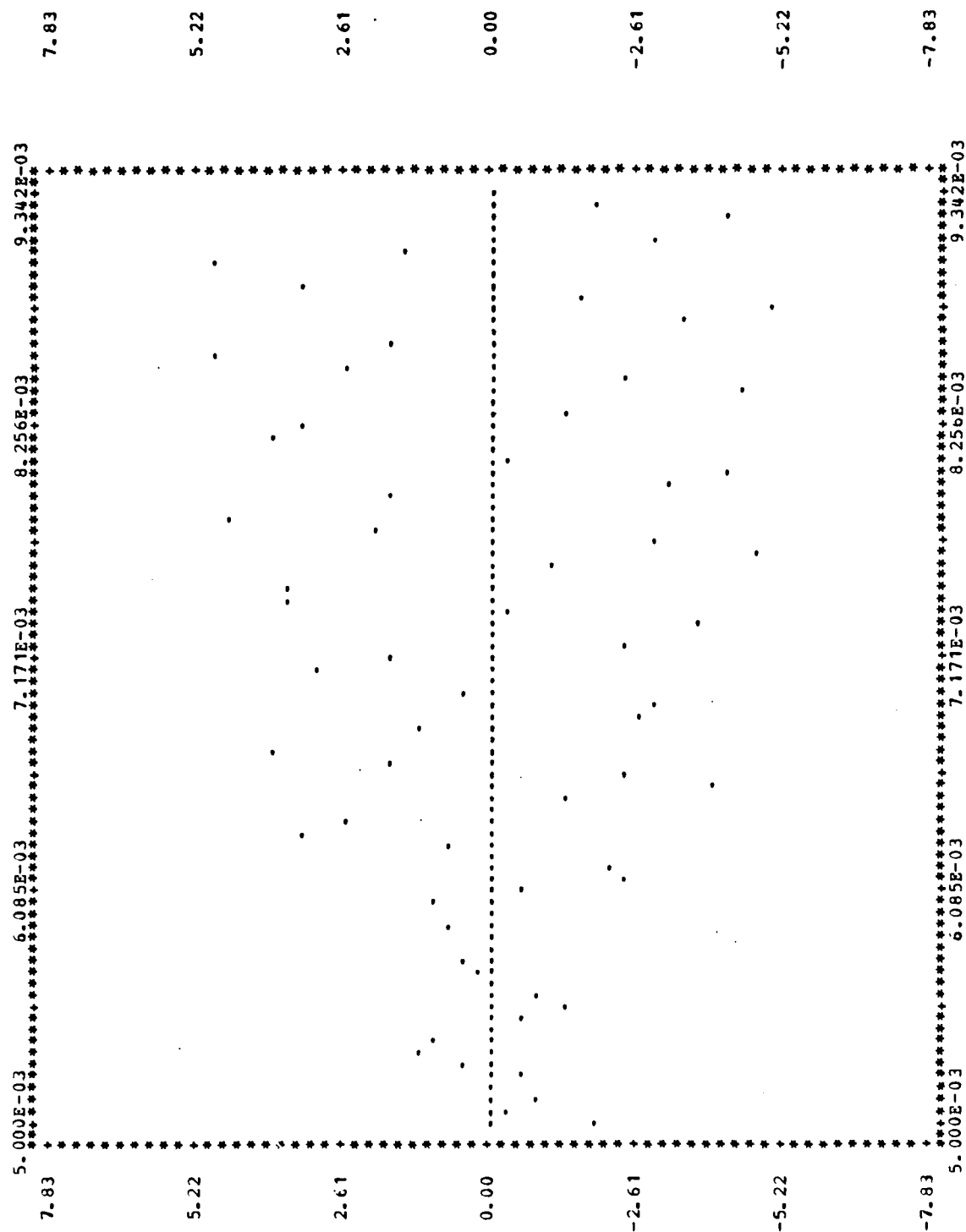




[illegible]

257

MODULATION TECHNIQUE = QPRS
 QPRS CLASS 3 FILTER
 BIFOLAR 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

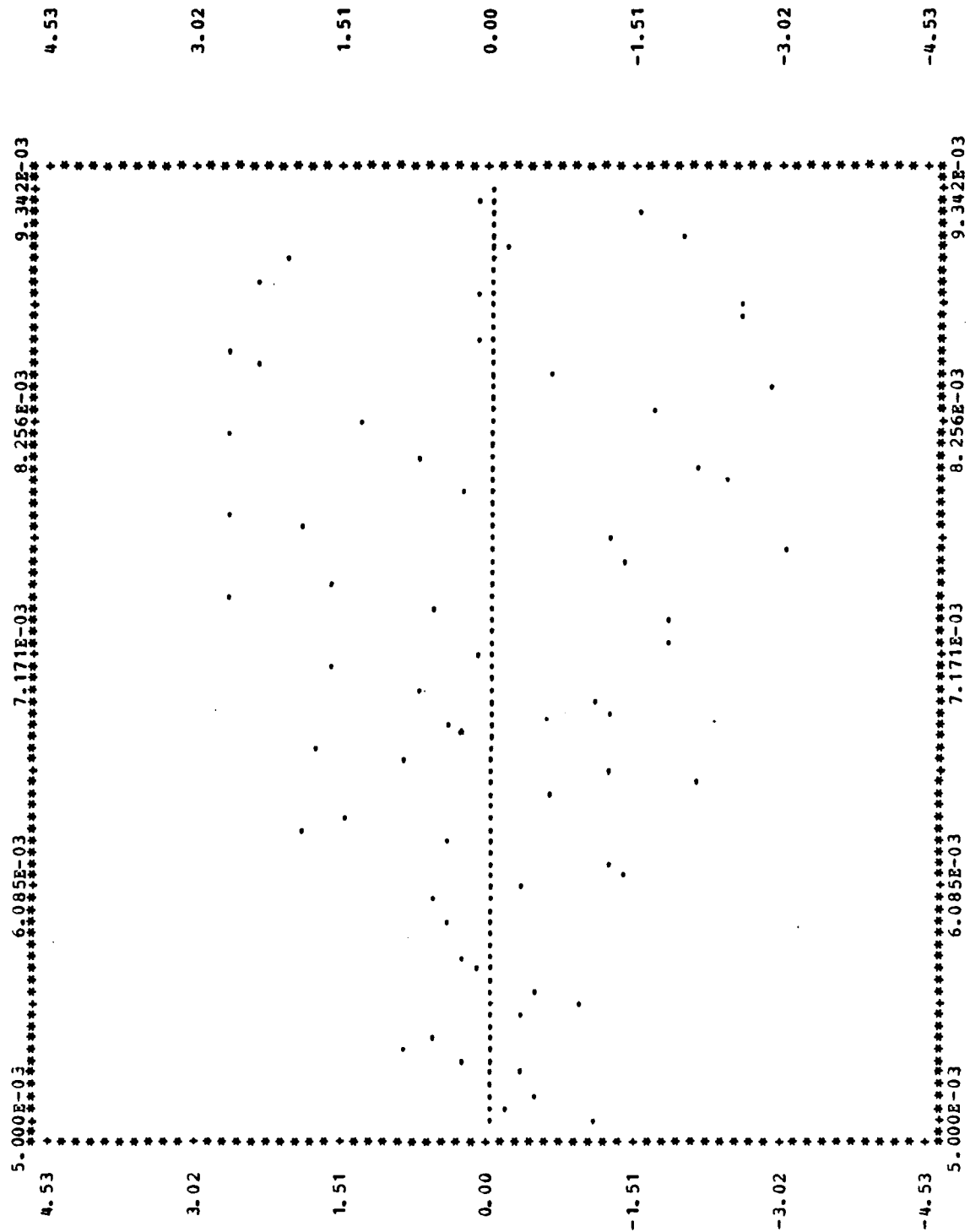


0.00	*****	121603.62	*****	243207.12	*****	364810.62	*****	486414.12	*****	134.35
134.35	*****	121603.62	*****	243207.12	*****	364810.62	*****	486414.12	*****	111.96
111.96	*****	121603.62	*****	243207.12	*****	364810.62	*****	486414.12	*****	89.57
89.57	*****	121603.62	*****	243207.12	*****	364810.62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

261

[illegible]

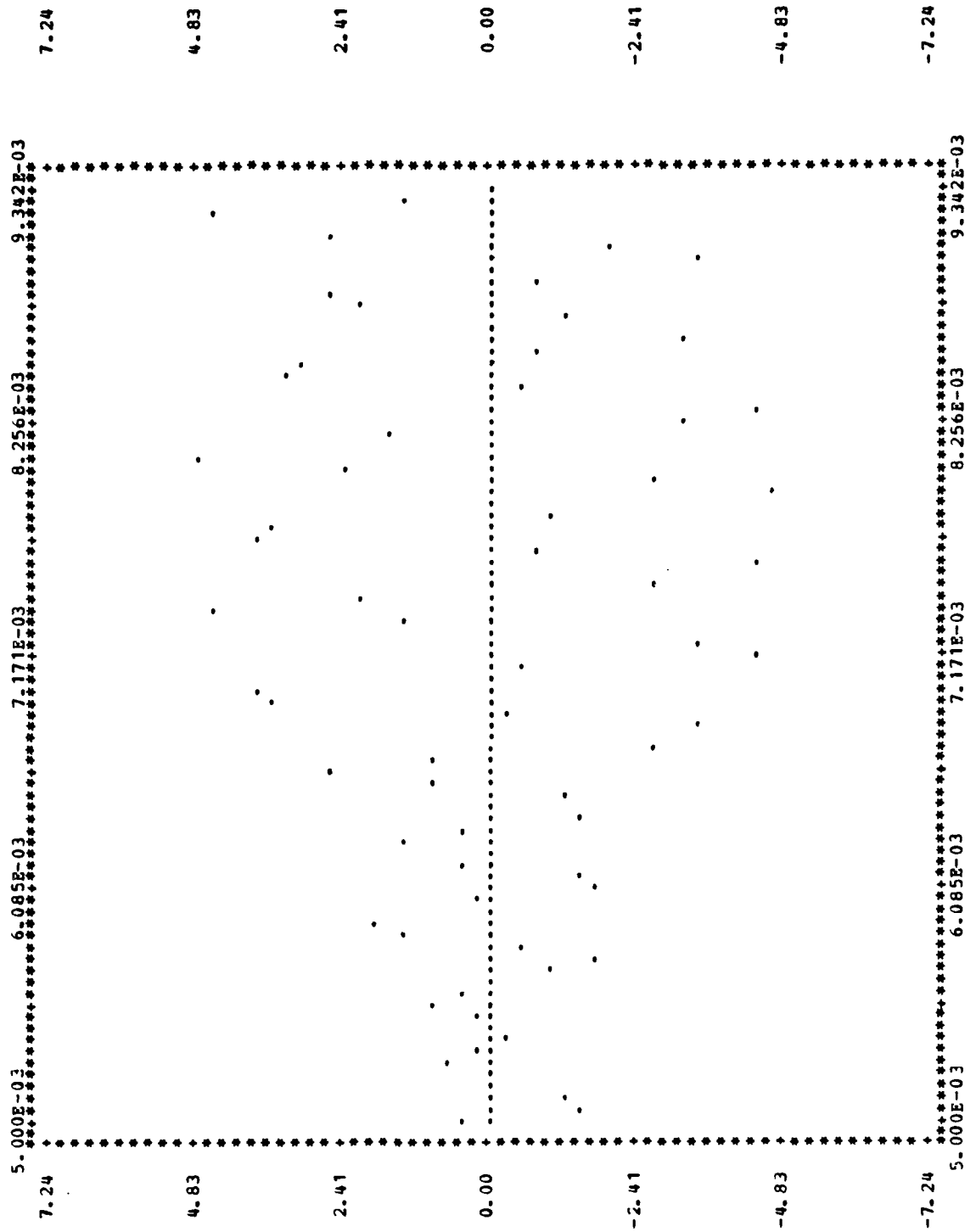
MODULATION TECHNIQUE = QPRS
 QPS CLASS 4 FILTER
 BIFOLAR LOGIC
 BAUD OR SYMBOL RATE = 1200 HZ
 BITS PER BINARY CODE WORD = 2
 BIT RATE = 0.2400000000000000E+04
 CARRIER FREQUENCY = 2400 HZ
 MAXIMUM CARRIER AMPLITUDE = 1 VOLT (S)
 INITIAL PHASE ANGLE = 0 DEGREES
 TIME BETWEEN SAMPLES = 0.6784342273548912E-04 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



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

[illegible]

MODULATION TECHNIQUE = QPRS
 QPS CLASS 5 FILTER
 BIPOLAR LOGIC
 BAUD OR SYMBOL RATE = 1200 HZ
 BITS PER BINARY CODE WORD = 2
 BIT RATE = 0.240000000000000E+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 RANDON 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



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

APPENDIX D

FORTRAN PROGRAM P-TEST

[illegible]

```

C      X=0.D0
      Y=0.D0
      Z=0.D0
C
C      DO 30 I=1,15
      X=X+SUM X(I)
      Y=Y+SUM Y(I)
      Z=Z+SUM Z(I)**2
      CONTINUE
C
C      XA=(Z/N)-(X**2/(K*N))
      XE=Y-(Z/N)
      TOTAL=Y-(X**2/(R*N))
C
C      DF1=R-1.D0
      DF2=R*(N-1.D0)
C
C      FSTAT=(XA/DF1)/(XE/DF2)
C      ** OUTPUT THE RESULTS-THE FORMATS REPRESENT VARIABLE DEFINITIONS
C
31      WRITE(6,31) J
      FORMAT(11,' F-STATISTICS FOR THE',I4,' COMPONENT')
C
40      WRITE(6,40) X
      FORMAT(10,' SUM OF SUMS =',E23.16)
C
41      WRITE(6,41) Y
      FORMAT(10,' SUM OF SUMS**2 =',E23.16)
C
42      WRITE(6,42) Z
      FORMAT(10,' SUM OF SQUARE OF SUMS =',E23.16)
C
50      WRITE(6,50) XA
      FORMAT(10,' AMONG GROUP SUM OF SQUARES =',E23.16)
C
51      WRITE(6,51) XE
      FORMAT(10,' WITHIN GROUP SUM OF SQUARES =',E23.16)
C
60      WRITE(6,60) TOTAL
      FORMAT(10,' TOTAL SUM OF SQUARES =',E23.16)
C
70      WRITE(6,70) DF1
      FORMAT(10,' DEGREES OF FREEDOM IN NUMERATOR =',E23.16)
C
71      WRITE(6,71) DF2
      FORMAT(10,' DEGREES OF FREEDOM IN DENOMINATOR =',E23.16)
C

```

```

80      WRITE(6,80) FSTAT
      C      FORMAT('0', ' F-STATISTIC =', E23.16)
100     CONTINUE
      C
81      WRITE(6,81)
      C      FORMAT('1')
      C      STOP
      C      END

```

```

PTE00950
PTE00960
PTE00970
PTE00980
PTE00990
PTE01000
PTE01010
PTE01020
PTE01030

```

APPENDIX E

RESULTS OF F-TEST

F-STATISTICS FOR THE 1 COMPONENT

SUM OF SUMS = 0.2091487882294994D 02
SUM OF SUMS**2 = 0.1810655469453404D 02
SUM OF SQUARE OF SUMS = 0.4128097510679048D 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 = 0.2621104542300214D 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 OF SUMS**2 = 0.3380633740317695D 02
SUM OF SQUARE OF SUMS = 0.1069682473356158D 03
AMONG 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 OF SQUARE OF SUMS = 0.1408215358007360D 03
AMONG GROUP SUM OF SQUARES = 0.2544544005119440D 00
WITHIN GROUP SUM OF SQUARES = 0.4172415106219358D 02
TOTAL 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 OF SUMS = 0.481038709234811D 02
SUM OF SUMS**2 = 0.520640938161174D 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 OF SUMS**2 = 0.8590285622098203D 02
SUM OF SQUARE OF SUMS = 0.3070018365343552D 03
AMONG 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
AMONG 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 = 0.2029942628781924D 03

SUM OF SUMS**2 = 0.1757564757819924D 04

SUM OF SQUARE OF SUMS = 0.7655692250451960D 04

AMCNG GROUP SUM OF SQUARES = 0.4908580866354581D 02

WITHIN GROUP SUM OF SQUARES = 0.1681007835315404D 04

TOTAL SUM OF SQUARES = 0.1730093643978950D 04

DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02

DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04

F-STATISTIC = 0.3097309684192641D 01

F-STATISTICS FOR THE 11 COMPONENT

SUM OF SUMS = 0.2004773302247246D 03

SUM OF SUMS**2 = 0.1811875077561617D 04

SUM OF SQUARE OF SUMS = 0.7980439863108266D 04

AMCNG GROUP SUM OF SQUARES = 0.5301029200839379D 02

WITHIN GROUP SUM OF SQUARES = 0.1732070678930534D 04

TOTAL SUM OF SQUARES = 0.1785080970938928D 04

DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02

DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04

F-STATISTIC = 0.3246332537532912D 01

F-STATISTICS FOR THE 12 COMPONENT

SUM OF SUMS = 0.2210834652773109D 03

SUM OF SUMS**2 = 0.2165859765218944D 04

SUM OF SQUARE OF SUMS = 0.9414279932820438D 04

AMCNG GROUP SUM OF SQUARES = 0.6155753358218842D 02

WITHIN GROUP SUM OF SQUARES = 0.2071716965890740D 04

TOTAL SUM OF SQUARES = 0.2133274499472928D 04

DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02

DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04

F-STATISTIC = 0.3151731454585565D 01

F-STATISTICS FOR THE 13 COMPONENT

SUM OF SUMS = 0.1979583434471791D 03

SUM OF SUMS**2 = 0.1904701075347160D 04

SUM OF SQUARE OF SUMS = 0.7185446672840862D 04

AMCNG GROUP SUM OF SQUARES = 0.4572946290150773D 02

WITHIN GROUP SUM OF SQUARES = 0.1832846608618751D 04

TOTAL SUM OF SQUARES = 0.1878576071520259D 04

DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02

DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04

F-STATISTIC = 0.2646478671459864D 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

AMCNG 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

AMCNG 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 = 0.1642503769158388D 03

SUM OF SQUARE OF SUMS = 0.5788746875066006D 03

AMCNG GROUP SUM OF SQUARES = 0.1299298669532475D 01

WITHIN GROUP SUM OF SQUARES = 0.1584616300407728D 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

AMCNG 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

AMCNG 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 = 0.2320537453498354D 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 = 0.8440126623444366D 02

SUM OF SUMS**2 = 0.1782254895437147D 03

SUM OF SQUARE OF SUMS = 0.6892901073945840D 03

AMONG GROUP SUM OF SQUARES = 0.2143851912627546D 01

WITHIN GROUP SUM OF SQUARES = 0.1713325884697689D 03

TOTAL SUM OF SQUARES = 0.1734764403823964D 03

DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02

DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04

F-STATISTIC = 0.1327251499840134D 01

F-STATISTICS FOR THE 26 COMPONENT

SUM OF SUMS = 0.6923579472492147D 02

SUM OF SUMS**2 = 0.1481985058366125D 03

SUM OF SQUARE OF SUMS = 0.4452980334185913D 03

AMONG GROUP SUM OF SQUARES = 0.1257250153391605D 01

WITHIN GROUP SUM OF SQUARES = 0.1437455255024266D 03

TOTAL SUM OF SQUARES = 0.1450027756558182D 03

DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02

DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04

F-STATISTIC = 0.9277389287476910D 00

F-STATISTICS FOR THE 27 COMPONENT

SUM OF SUMS = 0.6366092822129744D 02

SUM OF SUMS**2 = 0.1177906819912730D 03

SUM OF SQUARE OF SUMS = 0.3935159041535181D 03

AMONG GROUP SUM OF SQUARES = 0.1233349853537058D 01

WITHIN GROUP SUM OF SQUARES = 0.1138555229497378D 03

TOTAL SUM OF SQUARES = 0.1150888728032749D 03

DEGREES OF FREEDOM IN NUMERATOR = 0.1400000000000000D 02

DEGREES OF FREEDOM IN DENOMINATOR = 0.1485000000000000D 04

F-STATISTIC = 0.1149027974258136D 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 = 0.4536144151906700D 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 = 0.9109723144851782D 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 = 0.4319425777749392D 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

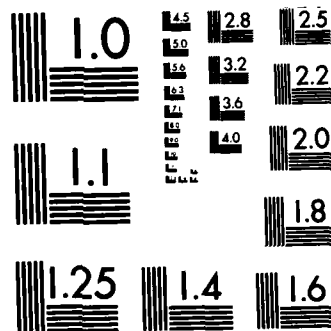
F/G 9/2

NL

END

FILED

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

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 Performance of QPR and Offset QPR Systems", IEEE Transactions on Communications, vol. COM-30, no. 12, Dec. 1982.
15. Harnuth, 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 Morissette, 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, R., 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. ICDa Craig D. Carlson Attack Squadron 42 Oceana, Virginia 23460	6	

END

FILMED

11-85

DTIC