DEDICATION

To AFAF my faithful wife, who I am not always able to convey my feelings for in words, and who devoted her time and energy giving me her full support throughout this work.

ACKNOWLEDGEMENTS

I would like to acknowledge my gratitude to Dr. R. L. Brewster of the Electrical and Electronic Engineering department at the UNIVERSITY OF ASTON IN BIRMINGHAM, for his guidance, encouragement and support throughout the period in which he supervised this piece of research. I would also like to add that it has been a real enjoyment to work with Dr. Brewster and make special mention of his helpfulness and good heartedness.

I am also grateful to Prof. J. E. Flood, head of the department of Electrical and Electronic Engineering.

A word of thankfulness should go the SYRIAN government for the generous financial support of this work.

Finally, my deep gratitude and thankfulness to my ever patient and beloved wife who has always been there to give her encouragement and love.

DIGITAL CODE DIVISION MULTIPLEX TECHNIQUES FOR DATA TRANSMISSION APPLICATIONS

Thesis

bY

IBRAHIM HAIDAR

for the degree of

DOCTOR OF PHILOSOPHY

Submitted to the department of
electrical and electronic engineering of
THE UNIVERSITY OF ASTON IN BIRMINGHAM

BIRMINGHAM

MAY 1985

DIGITAL CODE DIVISION MULTIPLEX TECHNIQUES FOR DATA TRANSMISSION APPLICATIONS

BY
IBRAHIM HAIDAR

Submitted for the degree of
DOCTOR OF PHILOSOPHY
at
THE UNIVERSITY OF ASTON IN BIRMINGHAM
1985

SUMMARY

Adaptive majority multiplexing techniques have been attracting research attention and investigation because of their adaptive error-correcting ability and their simplicity in hardware. A number of interesting research reports have recently appeared in the literature based on using Walsh functions as codes. One of the problems of using Walsh functions is the maximum number of channels which can be used.

This thesis is concerned with an investigation of the possibility of using different codes in order to increase the number of channels which can be used and the performance of the code division multiplexing system which uses these codes over channels impaired by White Gausian noise.

It has been found that there are a number of bridge function sets which are suitable for use as codes in such systems. These functions are three-valued functions which take the values +1, 0 and -1.

A formula which gives the necessary requirement for three-valued functions to be suitable as codes in a code division multiplexing system has been derived. Three groups of bridge functions have been investigated and found to be suitable as codes for such systems.

Three sets of results are provided. These are obtained from simulation of the systems which use the above three groups of bridge functions. These results give the performance of the system for every possible number of active channels when corrupted by White Gausian noise. All the results show the adaptive property in the relationship between the channel error-correcting properties and the number of active channels.

Comparisons are made between the results obtained and a T.D.M. system subject to the same conditions.

KEY WORDS

ORTHOGONAL, BRIDGE FUNCTIONS, CODE DIVISION MULTIPLEXING
DETERMINISTIC ERROR, ERROR CORRECTION MODE.

CONTENTS

CAPTION: PAGE NO.

CHAPTER THREE

CHAPTER FOUR

CAPTION: PAGE NO.

CHAPTER FIVE

CHAPTER SIX

LIST OF FIGURES

LIST OF TABLES

CHAPTER
ONE

1

INTRODUCTION

INTRODUCTION:

Reliable, economical, and efficient communication has become an important requirement of modern living. Communications systems are necessary to effect the transfer of information from one point to another over a suitable communication link.

A communications system will always be found, therefore, whenever information needs to be transmitted.

Efficient communication requires rapid transmission of data between widely separated points on the earths surface. The high cost of installing long distance communication links has led to the idea of multichannel systems where several signals are transmitted simultaneously over a single link.

Two techniques for combining separete channels into a single multiplexed link are in common use. These are frequency-division-multiplexing (F.D.M.) and time-division-multiplexing (T.D.M.).

In the first case (FDM) the available frequency spectrum is divided so that the signal at the transmitter is allocated to a different part of the spectrum and demultiplexing at the receiver is affected by frequency selective filters. Thus signal separation is achieved in the frequency domain. In the second case (TDM), each signal at the transmitter is allocated to a different time slot for transmission, and demultiplexing is affected by time dependant gating circuits. Thus the signal separation is in the time domain with all the signal sharing the same bandwidth.

Such systems are very inflexible in operation, in that they must be designed to cope adequately with busy hour channel loading, and take no account of the fact that in many applications the loading may be much less for a considerable proportion of the time.

An alternative approach to the problem is to use a technique in which all the signals share a common bandwidth and thus can all be transmitted simultaneously. Code- Division Multiplexing (CDM) is such a technique, where each signal is allocated a distinct waveform pattern by modulation with a unique code sequence. All the signals are then combined for transmission over the common communication channel. Suitable code sequences may be the row vectors of an orthogonal group code such as the set of Walsh functions [Gordon, J.A., Barrett, R., 1971], or discrete shifts of a psuedorandom code [Judge, W.J., 1962]. However any group of orthogonal functions can be used for this purpose. It will be shown in chapter 4 that a group of three-valued functions, called Bridge functions, has an excellent performance when used in this way.
Demultiplexing in code-division multiplexing systems is usually based on a pattern recognition process using correlation techniques.
Code-division multiplexing leads to a very flexible system with several advantages over conventional systems.

In the system described later, the code structure provides an adaptive error correcting facility such that, under condition of low traffic density, automatic error correction occurs. The performance gradually degrades with increasing traffic up to the limit of conventional Time-Division multiplex capacity.

In this thesis an introduction to orthogonal functions is given in chapter 2. This chapter has been included because most (CDM) systems are based on the orthogonality property and use sets of orthogonal functions as codes for the separate channels.
In chapter 3 a brief description of the various multiplexing techniques is given. The advantages and disadvantages of these techniques and their applications are also briefly discussed.
Details of Code-Division Multiplexing techniques are given in chapter 4. In Chapter 5 a code-division multiplexing system has been developed, which is based on Bridge functions as codes for the multiplexed channels, and the mathematical model for this system is presented.
Chapter 6 is devoted to the simulation results of the performance of the system when the transmitted signal is corrupted by additive white Gausian noise. The results are grouped into 3 sets according to the group codes used. The first set is the result from using the Bridge functions Bri(i,1,3,t) which is a group of 8 functions used as codes for multiplexing 8 channels. The second set is groups of 16 Bridge functions; Bri(i,1,4,t) and Bri(i,2,4,t); has been used for multiplexing 16

channels. Finally, the third set is groups of 32 Bridge functions; Bri(i,1,5,t), Bri(i,2,5,t) and Bri(i,3,5,t); used for multiplexing 32 channels. A discussion of the results and a comparision with the conventional TDM system is also presented. Conclusions and suggestions for further work follow in chapter 7.

Finally, Appendices, a list of references and a bibliography of useful reading are supplied.

CHAPTER
TWO

2

ORTHOGONAL FUNCTIONS

2.1 INTRODUCTION:

In communication systems, frequency-division multiplexing (FDM), time-division multiplexing (TDM) and code-division multiplexing are commonly used techniques. The common property of these techniques is the use of orthogonal functions, and the difference between them is the use of different sets of orthogonal functions. Sine and cosine functions are used as sub-carriers in frequency-division multiplexing, while block pulses are used in time-division multiplexing and Walsh or similar functions are generally used in code-division multiplexing. However, there are many orthogonal function sets which, in principle, can be used for code-division multiplexing.

This chapter gives a brief introduction to some sets of orthogonal functions. These are: Rademacher functions, Haar functions, block pulses functions, Walsh functions, and Bridge functions.

2.2 ORTHOGONALITY:

A set of functions $S_n(t)$ $(n=0,1,2,...)$ is said to be orthogonal with weight k over the interval $0 > t > T$ if

$$\int_0^T k.S_n(t).S_m(t).dt = \begin{cases} k & \text{if } n=m \\ 0 & \text{if } n \neq m \end{cases} \qquad (1)$$

where m and n has integer values and k is a non-negative constant which does not depend on the indices m and n. If the constant k is 1 then the set is normalised and called an orthonormal set of functions. It is always posible to convert a non-normalised set into an orthonormal set.

A set of functions is said to be complete or closed if no function exists which is orthogonal to every other function of the set unless the integral of the square of the function is itself zero. A necessary and sufficient condition for completeness is that Parsevals equation holds good for every function whose square is summable in the interval of orthogonality. In general it can be said that any set of functions which are capable of being integrated and of integrable square modulus may be used to form a closed set of linear combinations of the functions which will be normal and orthogonal [Beauchamp, K., 1975].

One of the important properties of orthogonal functions is that any time function can be represented by the superposition of members of the chosen orthogonal functions set. i.e orthogonal functions can be used to synthesise completely any time function to a required degree of accuracy. Another important property of orthogonal functions is that the identification of a particular member of the set contained in a given time function can be made by simple mathematical operation on the function.

2.3 RADEMACHER FUNCTIONS:

Rademacher functions are an incomplete but orthogonal function set. These functions represent a series of rectangular pulses or square-waves having unit mark-space ratio.

Figure (2.1a) shows some of the Rademacher functions and figure (2.1b) shows the matrix representation of the same Rademacher functions. This matrix is the sampling values of these functions at equally spaced intervals.

The first function, $R(0,t)$, is equal to unity for the entire interval $(0-T)$. The next and subsequent functions are square-waves having odd symmetry.

The incompleteness of these functions can easily be shown by considering the summation of a number of them. This summation will give a waveform also having odd symmetry about the centre and similar to the original functions, so that the even symmetry functions required for completeness can not be developed.

Rademacher functions have two arguments n and t such that $R(n,t)$ has 2^{n-1} periods of square wave over the unit time base $(0-1)$.

Rademacher functions can be derived from sinusoidal functions which have identical zero crossing positions.

Thus $R(n,t) = \text{sign}\ [\text{sine}(2^{n}\pi t)]$ (2)

The functions may be obtained from a sinusoidal waveform of appropriate frequency by amplification followed by a hard limiting process.

These functions are important principally because other functions, such as Walsh functions, can be derived from them.

2.4 HAAR FUNCTIONS:

Haar functions form a complete orthogonal function set of regtangular waveforms. Haar functions are three-valued functions where assume the values +1 , 0 , and -1 multiplied by powers of $\sqrt{2}$. The first few functions and their matrix representation are shown in figure (2.2).

The first two functions are identical to Rademacher functions R(0,t) and R(1,t). The next function, HAR(2,t), is simply HAR(1,t) squeezed into the left-half of the time base and modified in amplitude to $\pm\sqrt{2}$. The next function HAR(3,t) is identical but squeezed into the right-hand half of the time base. Subsequent pairs of function are similarly squeezed and shifted having amplitudes ± 1 multiplied by powers of $\sqrt{2}$.

In general all members of the same function subset (such as HAR(2,t) and HAR(3,t), or HAR(4,t), HAR(5,t), HAR(6,t), and HAR(7,t), etc.) are obtained by a lateral shift of the first member along the time axis by an amount proportional to its length.

A different definition for the set is as follows [Harmuth 1977].

$$HAR(0,t)=1 \qquad \text{for} \quad 0 \leqslant t < 1$$

$$HAR(i,j,t)=(2)^{i/2} \qquad \text{for} \quad \{\frac{j-1}{2^{i}} \leqslant t < \frac{j-.5}{2^{i}}\}$$

$$=-(2)^{i/2} \qquad \text{for} \quad \{\frac{j-.5}{2^{i}} \leqslant t < \frac{j}{2^{i}}\}$$

$$=0 \qquad \text{else where}$$

$$i=0,1,2,\ldots \quad . \quad j=1,2,3,\ldots 2^{i}$$

Then Haar functions can be re fered to by order j, and degree i, as well as time t. The degree i, then denotes a subset having the same number of zero crossings in a given width, $\frac{1}{2^i}$.

The order j, gives the position of the function within this subset. All members of the subset with the same degree are obtained by shifting the first member along the axis by an amount proportional to its order.

2.5 BLOCK PULSES FUNCTIONS:

Block pulses Blo(i,t) are an orthogonal but incomplete set of functions. The construction of a Block pulses set can be done by the sequence shift concept and may be explained as follows [Zhihua 1983]:

1. Decide the number of intervals N, in the unit interval (0,1)

   $N=2^p$ where p is an integer.

2. Taking the order number i as shift information, the sequence shift can be done as follows:
   a) The original value of the shift is always +1, and put at the position zero in the interval; and zero elsewhere.
   b) The original sequence +1 is shifted to the right, according to the value i. The value at position i in the interval is +1 ; and 0 else where.

Table (2.1) shows the process of forming all block pulses with order number i, i=0,1,2,...,7, and figure (2.3) shows the wave forms of these Block pulses.

2.6 WALSH FUNCTIONS:

Walsh functions WAL(i,t) are an orthogonal and complete set of functions, difined by J.L., WALSH [Walsh 1923]. These functions take the values +1 or -1 only in the unit interval (0,1).

Walsh functions can be derived by one of the following methods:

1. By means of a difference equation [Gibbs].
2. By Rademacher functions [Harmuth 1972].
3. From Hadamard matrices [Harmuth 1977].
4. From Boolean synthesis [Swick 1969].
5. By the concepts of "symmetric copy" [Zhihua 1983, L].

The last method will be explained here because the technique of symmetric copy will be used later for constructing Bridge functions. The method is based on certain features of symmetry, and is well-suited for the computer generation of Walsh functions. In this method, in order to construct the Walsh function WAL(n,t), the order number n is expressed in binary code. The function WAL(n,t) is then plotted according to the binary number n using the following steps:

1. The starting value is always +1 or + (for short).
2. The initial value is extended or "copied" as often as there are digits in the binary number n.
3. The digits of the binary number represent the mode of copying. A digit 0 means even symmetry copying, while a digit 1 means odd symmetry copying. The symmetry point moves to the right for each digit to extend the previously generated piece.

The construction of WAL(n,t) for n=0,1,2,...,7 is shown in table(2.2) and the waveforms of these functions are shown in figure (2.4).

2.7 BRIDGE FUNCTIONS:

Bridge functions are orthogonal functions which have a connection with both Walsh functions and Block pulses. These are a three-valued functions system taking the values +1, −1 and 0. The construction of these functions is obtained from a combination of the symmetric copying of Walsh functions with the shift of Block pulses.

The construction of Bridge functions by the concept of the shift and symmetric copy [Zhihua 1983, 2] is as follows:

1.Let i represent the order of the bridge fuction. Then i can be expressed in binary code $i_{p-1}\, i_{p-2} \dots i_{j-1} \dots i_1\, i_0$.

2.The binary number is thus divided into two parts ;

a) The j binary digits on the right side, $i_{j-1}\, i_{j-2} \dots i_1\, i_0$, these digits are used as shift information.

b) The (p−j) binary digits on the left side, $i_{p-1}\, i_{p-2} \dots i_j$, these are used as copying information.

3.Sequence shift is done first, followed by sequence copying.

a)The value of the original sequence is always +1 (+for short) in the interval 0, and zero in the other L−1 intervals where $L=2^j$.

b)According to the shift information $i_{j-1}\, i_{j-2} \dots i_0$, the original sequence "+" is shifted to the right side.

c)According to the copying information $i_{p-1} i_{p-2} \dots i_j$, the copying mode is used sequentially, The first time i_{p-1} is taken as the information, then i_{p-2} is taken and so on, until p-j digits are used.

A Bridge function of order i is represented by the expression:

Bri(i,j,p,t) (4)

It can be seen from the expression there are four independent variables for a Bridge fuction representation:

1.The order of the Bridge function i.

2.The independent variable, time t.

3.The number of digits representing the order of the function in binary code p, which follows from $N=2^p$ where N represents the total number of sub-intervals in the unit interval.

4.The number of shift information digits j which follows from $L=2^j \leqslant N$, where L represents the number of sub-intervals occupied by the shift.

The construction of Bridge functions Bri(i,1,4,t) is shown in table(2.3), and the wave-form of Bridge functions bri(i,1,3,t) are shown in figure (2.5).

2.8 OTHER ORTHOGONAL FUNCTIONS:

In addition to the above mentioned orthogonal functions there are many other orthogonal functions. One class of these can be generated from certain polynomials which can be made orthogonal by multiplications by a weighting factor.

This class consists of a series $f_n(x)$ $(n=0,1,2,\dots)$, where n is the degree of the polynomial. Among these functions which belong to this class are: Gegenbur, Laguerre, Jacobi, Chebychev, Hermite, and Legendre polynomial .

None of these functions has the essential simplicity of the previously mentioned orthogonal functions where members of these functions assume a single value having either a positive or negative sign, this has the effect of reducing multiplicative operations and therefor this gives rise to simplicity in calculations and in hardware implementation.

Legendre polynomials will be taken as an example of this class. One of the posible definitions of these polynomials can be given by the general expression [Sansone 1959].

$$P_n(x)=\frac{1*3*5*\dots*(2n-1)}{n!}\left[x^n-\frac{n(n-1)}{2(2n-1)}x^{n-2}+\frac{n(n-1)(n-2)(n-3)}{2*4*(2n-1)(2n-3)}x^{n-4}\dots\right]$$

And $P_0(x)=1$

From that the first few orthogonal polynomials of these functions are given by the following equations:

$$P_1(x)=x$$

$$P_2(x)=\frac{1}{2}(5x^2-1)$$

$$P_3(x)=\frac{1}{2}(5x^2-3x)$$

$$P_4(x)=\frac{1}{8}(35x^4-30x^2+3)$$

$$P_5(x)=\frac{1}{8}(63x^5-70x^3+15x)$$

These polynomials are shown in figure(2.6).

2.9 CONCLUSION:

Although it is possible, in theory, to use any orthogonal functions set for multiplexing, there are some practical factors which should be considered in choosing one set rather than the other. Among these factors are ease of generation, detectability , and flexibility in engineering.

Table(2.1): The process of forming Blo(i,t),i=0,1,...7

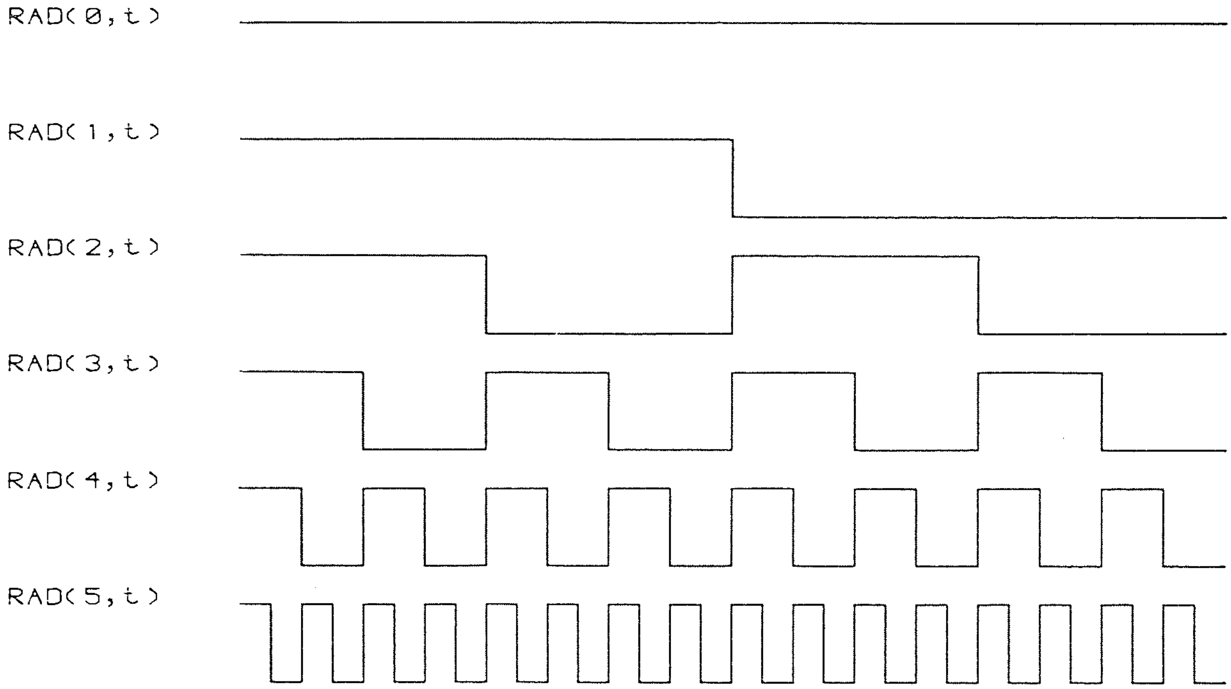
Order number	Original sequence								Shifted sequence							
	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
0	+	0	0	0	0	0	0	0	+	0	0	0	0	0	0	0
1	+	0	0	0	0	0	0	0	0	+	0	0	0	0	0	0
2	+	0	0	0	0	0	0	0	0	0	+	0	0	0	0	0
3	+	0	0	0	0	0	0	0	0	0	0	+	0	0	0	0
4	+	0	0	0	0	0	0	0	0	0	0	0	+	0	0	0
5	+	0	0	0	0	0	0	0	0	0	0	0	0	+	0	0
6	+	0	0	0	0	0	0	0	0	0	0	0	0	0	+	0
7	+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	+

Table(2.2): The process of forming wal(i,t),i=0,1,.. 7.

Order number	Binary code	Original sequence	First copy	Second copy	Third copy
i	i_2 i_1 i_0		i_2	i_1	i_0
0	0 0 0	+	+	+ +	+ + + +
1	0 0 1	+	+	+ +	- - - -
2	0 1 0	+	+	- -	- - + +
3	0 1 1	+	+	- -	+ + - -
4	1 0 0	+	-	- +	+ - - +
5	1 0 1	+	-	- +	- + + -
6	1 1 0	+	-	+ -	- + - +
7	1 1 1	+	-	+ -	+ - + -

Table(2.3): The process of forming Bri(i,1,4,t),i=0,1,.. .15.

Order number	Binary code	copy information	Shift info	Original sequence	Sequence after shift	First copy sequence	2nd copy sequence	3rd copy sequence
i	i_3 i_2 i_1 i_0	i_3 i_2 i_1	i_0			i_3	i_2	i_1
0	0 0 0 0	0 0 0	0	+ 0	+ 0	0 +	+ 0 0 +	+ 0 0 + + 0 0 +
1	0 0 0 1	0 0 0	1	+ 0	0 +	+ 0	0 + + 0	0 + + 0 0 + + 0
2	0 0 1 0	0 0 1	0	+ 0	+ 0	0 +	+ 0 0 +	- 0 0 - - 0 0 -
3	0 0 1 1	0 0 1	1	+ 0	0 +	+ 0	0 + + 0	0 - - 0 0 - - 0
4	0 1 0 0	0 1 0	0	+ 0	+ 0	0 +	- 0 0 -	- 0 0 - + 0 0 +
5	0 1 0 1	0 1 0	1	+ 0	0 +	+ 0	0 - - 0	0 - - 0 0 + + 0
6	0 1 1 0	0 1 1	0	+ 0	+ 0	0 +	- 0 0 -	+ 0 0 + - 0 0 -
7	0 1 1 1	0 1 1	1	+ 0	0 +	+ 0	0 - - 0	0 + + 0 0 - - 0
8	1 0 0 0	1 0 0	0	+ 0	+ 0	0 -	- 0 0 +	+ 0 0 - - 0 0 +
9	1 0 0 1	1 0 0	1	+ 0	0 +	- 0	0 - + 0	0 + - 0 0 - + 0
10	1 0 1 0	1 0 1	0	+ 0	+ 0	0 -	- 0 0 +	- 0 0 + + 0 0 -
11	1 0 1 1	1 0 1	1	+ 0	0 +	- 0	0 - + 0	0 - + 0 0 + - 0
12	1 1 0 0	1 1 0	0	+ 0	+ 0	0 -	+ 0 0 -	- 0 0 + - 0 0 +
13	1 1 0 1	1 1 0	1	+ 0	0 +	- 0	0 + - 0	0 - + 0 0 - + 0
14	1 1 1 0	1 1 1	0	+ 0	+ 0	0 -	+ 0 0 -	+ 0 0 - + 0 0 -
15	1 1 1 1	1 1 1	1	+ 0	0 +	- 0	0 + - 0	0 + - 0 0 + - 0



(a)

RAD(0,t) +

RAD(1,t) + + + + + + + + + + + + + + + + - - - - - - - - - - - - - - - -

RAD(2,t) + + + + + + + + - - - - - - - - + + + + + + + + - - - - - - - -

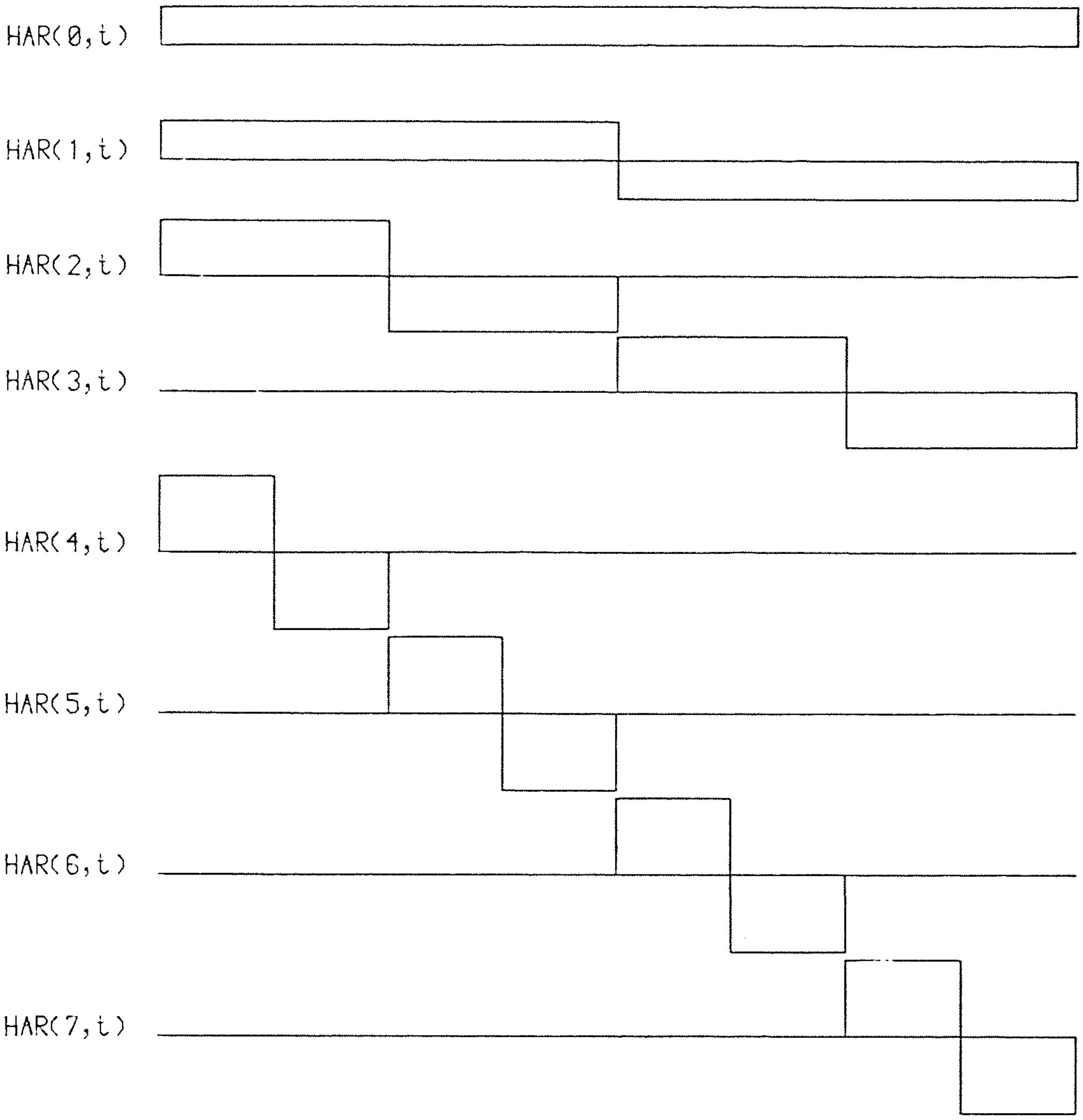
RAD(3,t) + + + + - - - - + + + + - - - - + + + + - - - - + + + + - - - -

RAD(4,t) + + - - + + - - + + - - + + - - + + - - + + - - + + - - + + - -

RAD(5,t) + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + -

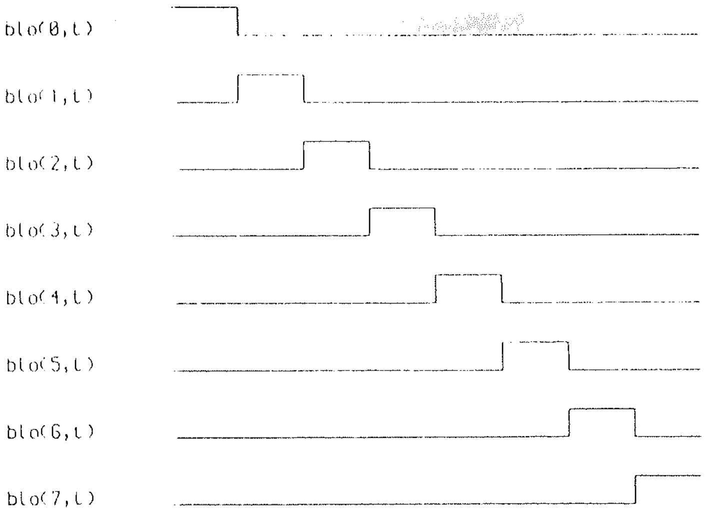
(b)

FIG.(2.1): Rademacher functions(a) & their matrix representation(b)



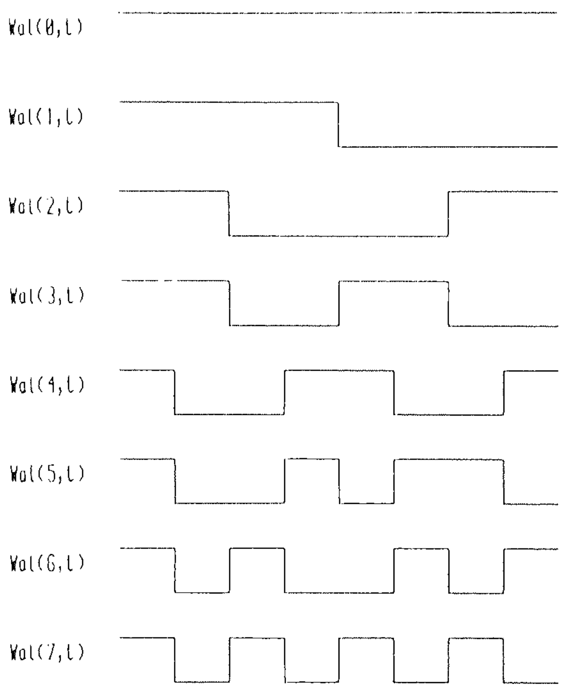
$$\begin{matrix}
1.00 & 1.00 & 1.00 & 1.00 & 1.00 & 1.00 & 1.00 & 1.00 \\
1.00 & 1.00 & 1.00 & 1.00 & -1.00 & -1.00 & -1.00 & -1.00 \\
1.41 & 1.41 & -1.41 & -1.41 & .00 & .00 & .00 & .00 \\
.00 & .00 & .00 & .00 & 1.41 & 1.41 & -1.41 & -1.41 \\
2.00 & -2.00 & .00 & .00 & .00 & .00 & .00 & .00 \\
.00 & .00 & 2.00 & -2.00 & .00 & .00 & .00 & .00 \\
.00 & .00 & .00 & .00 & 2.00 & -2.00 & .00 & .00 \\
.00 & .00 & .00 & .00 & .00 & .00 & 2.00 & -2.00
\end{matrix}$$

FIG.(2.2):Some of Haar functions & their matrix representation.



$$\begin{matrix} + & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & + & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & + & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & + & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & + & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & + & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & + & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0. & + \end{matrix}$$

FIG.(2.3): The block functions blo(i,t)And their matrix representation.

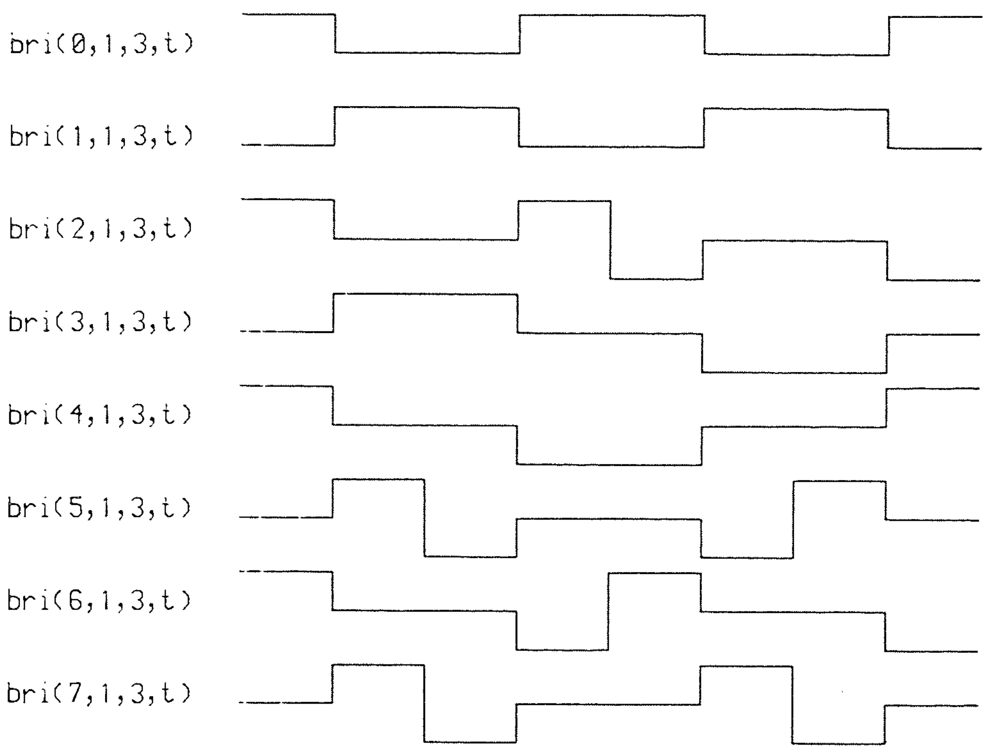


(A)

$$\begin{matrix} + & + & + & + & + & + & + & + \\ + & + & + & + & - & - & - & - \\ + & + & - & - & - & - & + & + \\ + & + & - & - & + & + & - & - \\ + & - & - & + & + & - & - & + \\ + & - & - & + & - & + & + & - \\ + & - & + & - & - & + & - & + \\ + & - & + & - & + & - & + & - \end{matrix}$$

(B)

FIG. (2.4) :Some of Walsh Function waveforms (A) and their matrix representation (B).



$$\begin{matrix}
+ & 0 & 0 & + & + & 0 & 0 & + \\
0 & + & + & 0 & 0 & + & + & 0 \\
+ & 0 & 0 & + & - & 0 & 0 & - \\
0 & + & + & 0 & 0 & - & - & 0 \\
+ & 0 & 0 & - & - & 0 & 0 & + \\
0 & + & - & 0 & 0 & - & + & 0 \\
+ & 0 & 0 & - & + & 0 & 0 & - \\
0 & + & - & 0 & 0 & + & - & 0
\end{matrix}$$

FIG.(2.5): The Bridge Functions bri(i,1,3,t)

And their matrix representation.

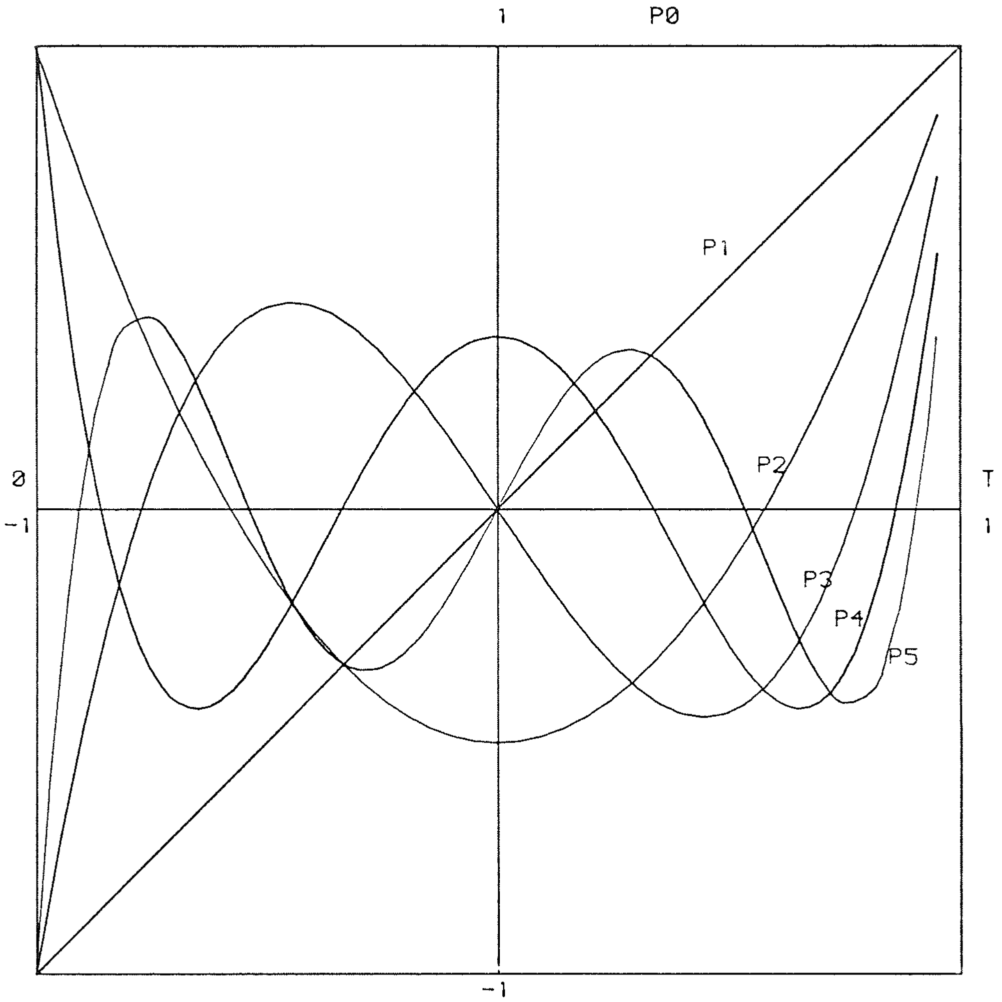


FIG.(2.6):The First 6 Lagendre Functions.

CHAPTER
THREE

3

MULTIPLEXING TECHNIQUES

3.1 INTRODUCTION:

Multiplexing is the technique where by a number of independent signals can be combined into a composite signal suitable for transmission over a common communication channel.

A general block diagram of a multiplexing system is shown in figure(3.1) which consists of multiplexer, demultiplexer and a common communication channel.

This chapter introduces the idea of multiplexing based on the channel capacity, and from there the conventional multiplexing systems of FDM and TDM are explained. The CDM technique is then briefly discused.

3.2 CHANNEL CAPACITY AND MULTIPLEXING:

The idea of multiplexing is based on the capacity of a channel for transmission of information. When the capacity of a channel exceeds the information rate of each signal to be transmitted, it is possible in theory to transmit more than one signal at a time.

The maximum rate at which a channel can transmit information is given by [Shannon 1948]:

$$C=W.\mathrm{LOG}_2\left(1+\frac{S}{N}\right) \qquad (3.1)$$

Where C is the channel capacity in bits/sec.

W is the bandwidth in Hz.

$\frac{S}{N}$ is the signal-to-noise ratio (SNR).

This equation shows that there are two posible ways of multiplexing:

The first way is by allowing the bandwidth of the common channel (W) to be more than that of each signal to be transmitted, so that the bandwidth of the channel is shared by the transmitted signals. This way may be called multiplexing by bandwidth sharing.

The other way of multiplexing is possible when the signal-to-noise ratio (SNR) exceeds that required for adequate transmission of the signals. This method may be called level-division multiplexing [Flood 1980].

3.3 BASIC MULTIPLEX SYSTEM:

The basic multiplex problem can be demonstrated by an example of multiplexing two signals. Figure (3.2) illustrates a basic multiplexing system for two signals, in which modulation inputs S1 and S2 are multiplied by carriers C1 and C2. The sum of these two multiplications is transmitted over the common medium.

In order to individually extract the two signals S1 and S2 from the received signal Rx at the receiving terminal, it is necessary for the carriers to be orthogonal. Suitable filters at the receiving terminals, followed by demodulators, enable the modulating inputs to be recovered. The separation process can be explained mathematically by refering to figure(3.2) as follows:

$$Rx = S1.C1 + S2.C2$$

$$A = Rx.C1 = S1.C1^2 + S2.C1.C2$$

Now if C1 and C2 are orthonormal then $C1.C2 = 0$, and $C1^2 = 1$

Therefore $A = S1$

In a simillar way $B = S2$

The carrier which can be used in such a system could be sine waves as the case in (F.D.M.), pulses displaced in time as the case in (T.D.M.), or any other carriers which could be used in a simillar manner.

3.4 FREQUENCY DIVISION MULTIPLEXING:

The first commercial use of frequency division multiplexing for telephone traffic is believed to have been in 1918, on a line between Baltimore and Pittesburgh in U.S.A. [Pearce 1977]. This enabled up to four telephone channels to be handled simultaneously by transmitting each channel in a different frequency band. Nowadays the same technique has been developed to the point where up to 10,000 channels can be carried by a single coaxial cable.

The essentials of a frequency division multiplexing system are shown in the block diagram of figure (3.3a). At the transmission end, each channel input is first filtered to its required baseband frequency and then shifted to a separate frequency band by means of a single-side-band ballanced modulator (SSB B.M.), and then the sum of all the channel inputs is transmitted.

The spectrum of the transmitted signal is shown in figure (3.3b).

At the receiving end (Demultiplexer) the individual signals (channels) are separated by the use of band pass filters and then the frequency of each signal shifted back to its original base band by a demodulator and low pass filter.

For the individual channels to be recovered without cross-talk certain requirements must be met:

1. The use of high quality filters is essential to avoid interference because poor filtering of either the unwanted side-bands which are generated during the mixing process or any out-of-band signal present at a channel input, will generally lead to interference between the neighbouring channels when the signals are combined at the multiplexer output. Also, at the demultiplexer, poor band-pass filtering will allow neighbouring channels to be passed into the same outlet via the demodulator, and poor base-band filtering will allow out-of-band signals generated by the demodulator to appear at the output.

2. An adequate guard band between channels is therefore necessary to allow for practical filter cut-off characteristics.

Thus, if the bandwidth of the channel is f_m and the guard band is B_g then each channel has an overall bandwidth [Stein & Jones 1967]

$B_o = f_m + B_g$

Then, for an FDM system utilizing SSB consisting of a group of N such channels, a total transmission bandwidth B_t is required

where $B_t = N.B_o = N.f_m + N.B_g$

This results in a transmission bandwidth greater than the sum of the indivisual channels by $N.B_g$.

3. Another requirement for good recovery of the indivisual channels is a high degree of amplitude linearity in those parts of the system carrying the complete multiplex signal.
Any part of the system carrying the multiplex signal with amplitude non-linearity may lead to the generation of spurious components in one channel due to mixing between signals in other channels [Pearce 1977].

3.5 TIME DIVISION MULTIPLEXING:

Time-division multiplexing permits a number of different signal sources to use the same transmission medium by giving each source access to the whole transmission path for a period of time. To understand how this principle can be realised, a time interval of one digit time-slot is defined as the time allowed in the multiplex for the transmission of one bit.
A synchronous digital multiplex equipment could be devised in which n independent binary signal streams, operating at the same rate T bits/sec., form the input tributaries to the multiplex equipment. Each of the n parallel input tributaries, therefore, contains T digit time slots per unit time interval, and the multiplex output must generate nT digit time-slots in the same time interval, that is the multiplex output would be exactly n times the input rate, and the duration of an output

digit time slot would be 1/n of the duration of an input digit time slot [Brigham, Snaith and Wilcox 1976].

Figure (3.4a) shows the basic block diagram of a T.D.M system. In this system each channel is sampled and connected to the transmission line in turn by means of a rotating switch or commutator. Demultiplexing is effected by an identical and synchronised commutator which directs each sample from the line to its correct outlet.

Synchronisation is achieved by inserting a synchronising signal at the multiplex end which can be recognised at the demultiplex, this signal can be used to set the commutator into its correct phase. This kind of synchronisation is called block sync.

Synchronisation can be achieved by another method, that is by using one of the data channels to carry the synchronising signal. At the demultiplex, synchronisation is achieved by slowly advancing the phase of the commutator until this signal is recognised in the correct channel outlet. Although this kind of synchronisation is simpler than the block synchronising , since no storage is involved, this can lead to very long synchronisation times and is therefore not often used [Pearce 1977].

In principle, the arrangment shown in figure (3.4a) can be used to multiplex either analogue or digital signals. In the case of analogue signals, the use of low pass filters with cut-off frequencies at half the sampling rate are necessary at both input and output. In the digital case, these are replaced by clocked bistable memories. However, in practice the performance of an analogue system is usually limited by inter-symbol interference introduced during transmission and by cross-talk resulting from poor isolation in the commutators. Therefore, the technique is better suited to the multiplexing of digital signals where a degree of cross-talk can be tolerated and subsequently removed by signal regeneration. This process is not possible with analogue signals.

Figure (3.4b) shows the basic form of a digital T.D.M signal in which a single data bit is transmitted from each channel in turn, to give a bit interleaved structure. The synchronising signal is inserted between the data bits at regular intervals in the form of a unique sequence of bits. When this sequence is first recognised at the demultiplex, this sets the commutator into its correct phase. The recieved data is then continually examined for the presence of the sync. word in its expected position and a further synchronisation initiated in the event of failure, which may be caused by false synchronisation on an immitation of the sync. word present in the data [Pearce 1977].

As indicated before, separation or demultiplexing of the pulse streams is accomplished by the output commutator, which must be accurately synchronised with the input commutator process. If a loss of synchronism occurs between the input and output commutator, the pulse samples will be gated into the wrong output channel filters, resulting in confusion of the message waveforms appearing in all channels. Perhaps the most significant problem in T.D.M systems is the acquisition and tracking of synchronisation information. A common device used in

T.D.M systems to help ease the problems associated with channel separation and synchronisation is to include guard time slots or vacant time intervals between pulses. Guard time slots have the effect of further decreasing pulse widths and causing an additional widening of the transmission bandwidth.

3.6 LIMITATIONS OF CONVENTIONAL SYSTEMS:

There are three noticable limitations associated with conventional systems [Leakey 1973]:

3.6.1 Overload characteristic:

Conventional systems suffer from very sudden overload characteristics. Their performance is adequate up to their designed carrying capacity, but any added traffic meets absolute blocking. The equipment must therefore be designed so that the chance of requiring more circuits than the designed maximum capacity is very small. As a result of this requirment even the average busy hour loading often does not exceed 50% of the maximum capacity. Thus on average, a high percentage of the available capacity remains unused, a situation which is more significant when it is realised that the traffic loading outside the busy hours is probably less than 10%. This limitation is further compounded in that the performance of the individual circuit does not improve significantly under conditions of light overall loading. A better multiplex system would provide optimum performance at the designed loading, improved performance below this loading and a degraded performance above. A gradual overload characteristic would reduce the extent to which it was necessary to provide normally unused capacity for the rareheavily loaded conditions, and would enable the available capacity to be usefully employed at all times.

3.6.2 INTERFERENCE:

Conventional multiplex systems which use sine waves or pulses are prone to disruption by interference, since unpredictable interference noise normally occurs in forms similar to sine waves or pulses. The design of systems which are susceptible to such interference is therefore enhanced by employing a high signal-to-noise ratio and by including special additional subsystems, such as error-correcting units. A better solution could result from using carriers which are not readily simulated by interference , where many of the precautions necessary with conventional systems would be either automatically inbuilt or unnecessary. Also, in the conventional multiplexing systems the trend is to send discrete messages over discrete physical paths with the result that, if a break down occurs, some messages suffer disruption while others are not. An alternative aproach would be to send all messages down all circuits so that on the physical disruption of one circuit all messages will be affected equally. This would enable the effect of faults and deterioration to be distributed and the effects of interference to be automatically minimised and distributed.

3.6.3 INSTANTANEOUS LOADING REQUIRMENTS:

The third limitation concerns system adaptability to instantaneous loading requirements. Although the transmission medium is normally capable of supporting many types of signal, most conventional systems involved strict allocation of capacity in terms of the various message signals which they are required to transmit. Although the transmission media can be made general purpose, the multiplex equipment has to be rigidly defined in accordance with the signals that it is required to handle.

For example, a system capable of transmitting television can not normally be automatically adjusted to carry an equivalent number of speech circuits without substantial changes to multiplex equipment at each end. This is a serious limitation, since it is necessary to know, not only the total communication requirements, but also the manner in which these are divided between the various message signal requirements. A more convenient and desirable system would be adaptable to the instantaeous requirements in which the multiplex arrangement automatically adjusts to the various message signals applied to it.

3.7 CODE DIVISION MULTIPLEXING:

Code-division-multiplexing is a technique by which two or more information sources communicate over a common communication channel (or transmission medium), yet share the same carrier frequency and have transmissions occuring simultaneously.

Figure (3.5) shows a basic block diagram of a code-division-multiplexing system. In this system, each information bit from the channel inputs is multiplied by a higher rate digital code, which is unique to that channel, then the output of the multipliers are added and the result of this addition is transmitted.

At the demultiplex end, information is recovered by multiplying the received signal by the same code, which must be correctly phased so as to obtain a correlation with the wanted channel, while rejecting the others. This rejection depends heavily upon the cross-correlation properties of the codes.

The cross-correlation product over each information bit can be reduced to zero if simple sequences such as Walsh functions are used as codes. A code-division-multiplex system based on such a code set will occupy a bandwidth similar to that of an equivalent T.D.M system, but it has the advantage of resistance to interference which improves as the channel loading is reduced [Pearce 1977].

In this section only a brief introduction to code division multiplexing techniques is given.A more detailed discussion of this technique is given in the next chapter.

3.8 CONCLUSION:

Conventional multiplex systems (F.D.M and T.D.M) have certain limitations: they suffer from very sudden overload characteristics, they are prone to disruption by interference and they are not flexible to instantaneous changes in loading.

Considerable scope therefore exists for an investigation into multiplex systems other than those based on sinewave and block pulses.

Code-division multiplexing is a possible alternative technique which can eliminate, or at least rectify some of the deficiencies of existing systems.

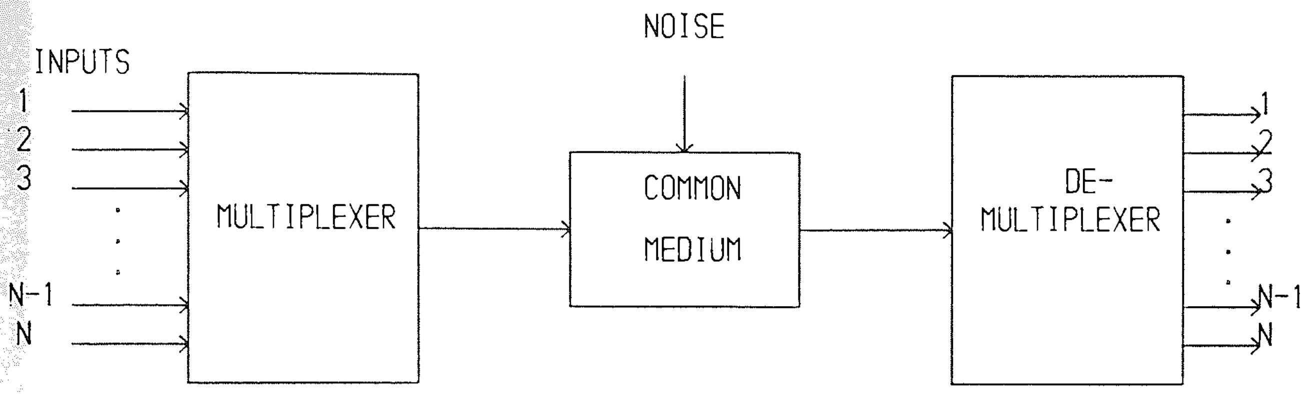


FIG.(3.1):Block diagram of general multiplexing system.

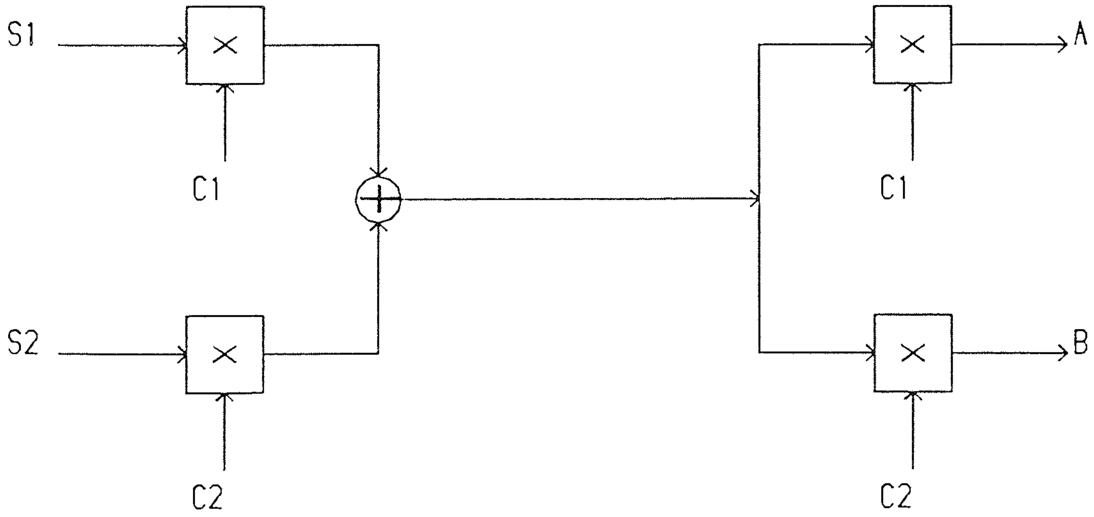


FIG.(3.2):Basic multiplex system of two signals.

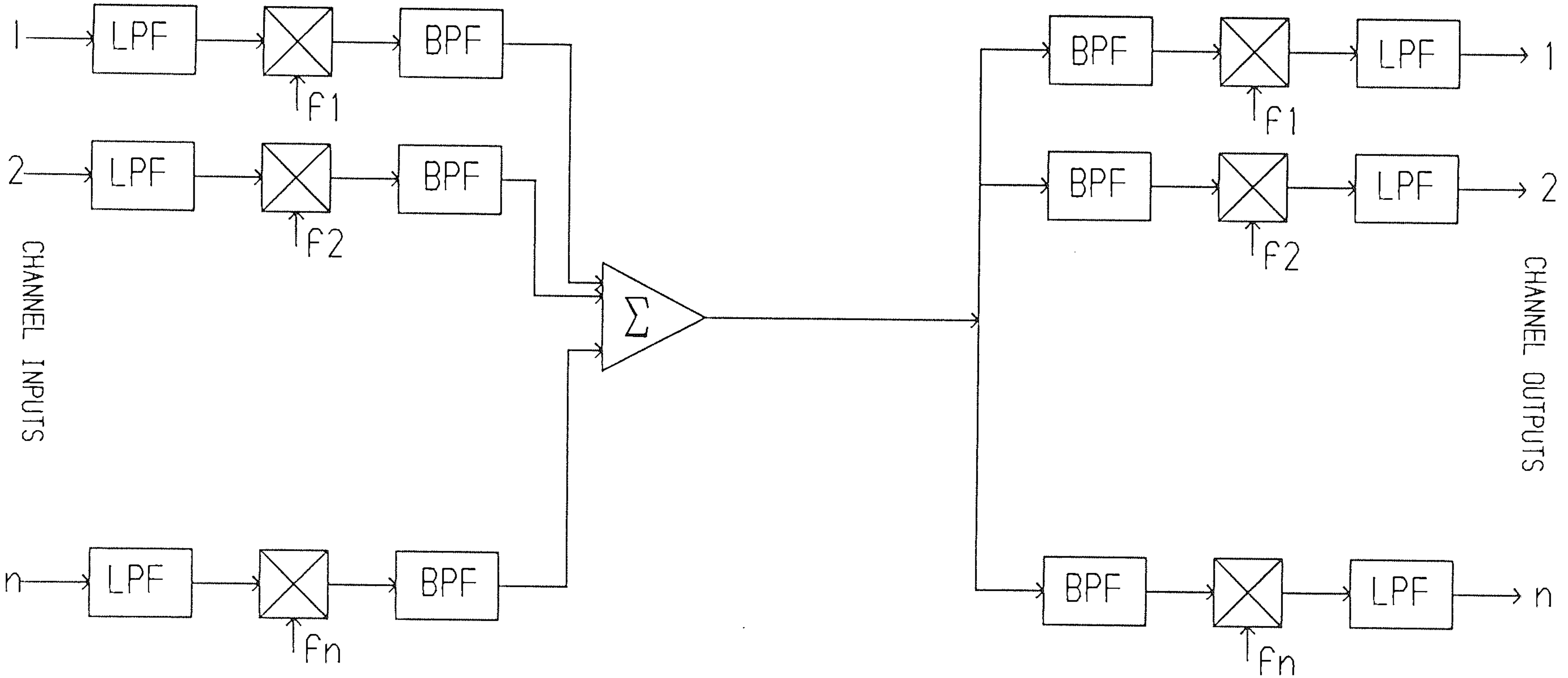


FIG.(3.3a):Block diagram for basic frequency division multiplexing system.

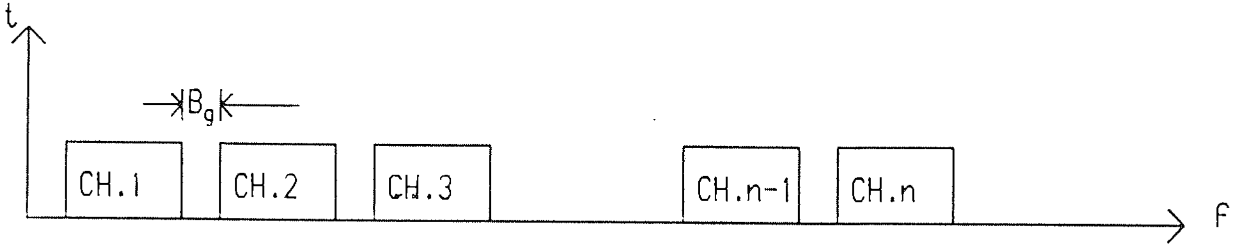


FIG.(3.3b): The spectrum of F.D.M. signal.

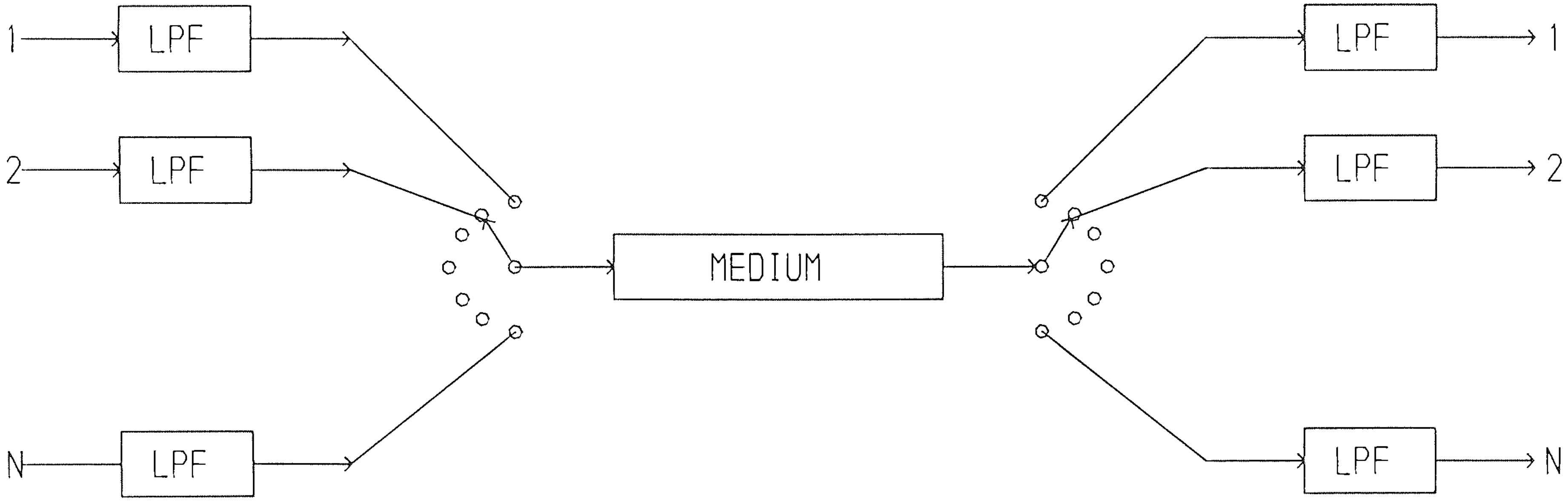


FIG.(3.4a): Basic block diagram of T.D.M.

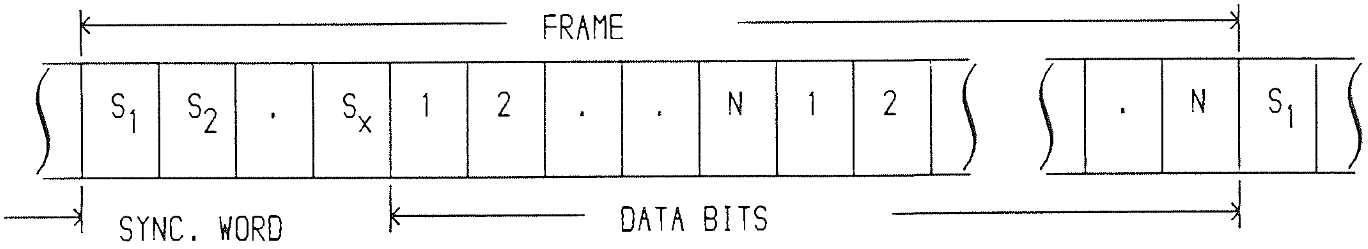


FIG.(3.4b): Digital T.D.M. signal.

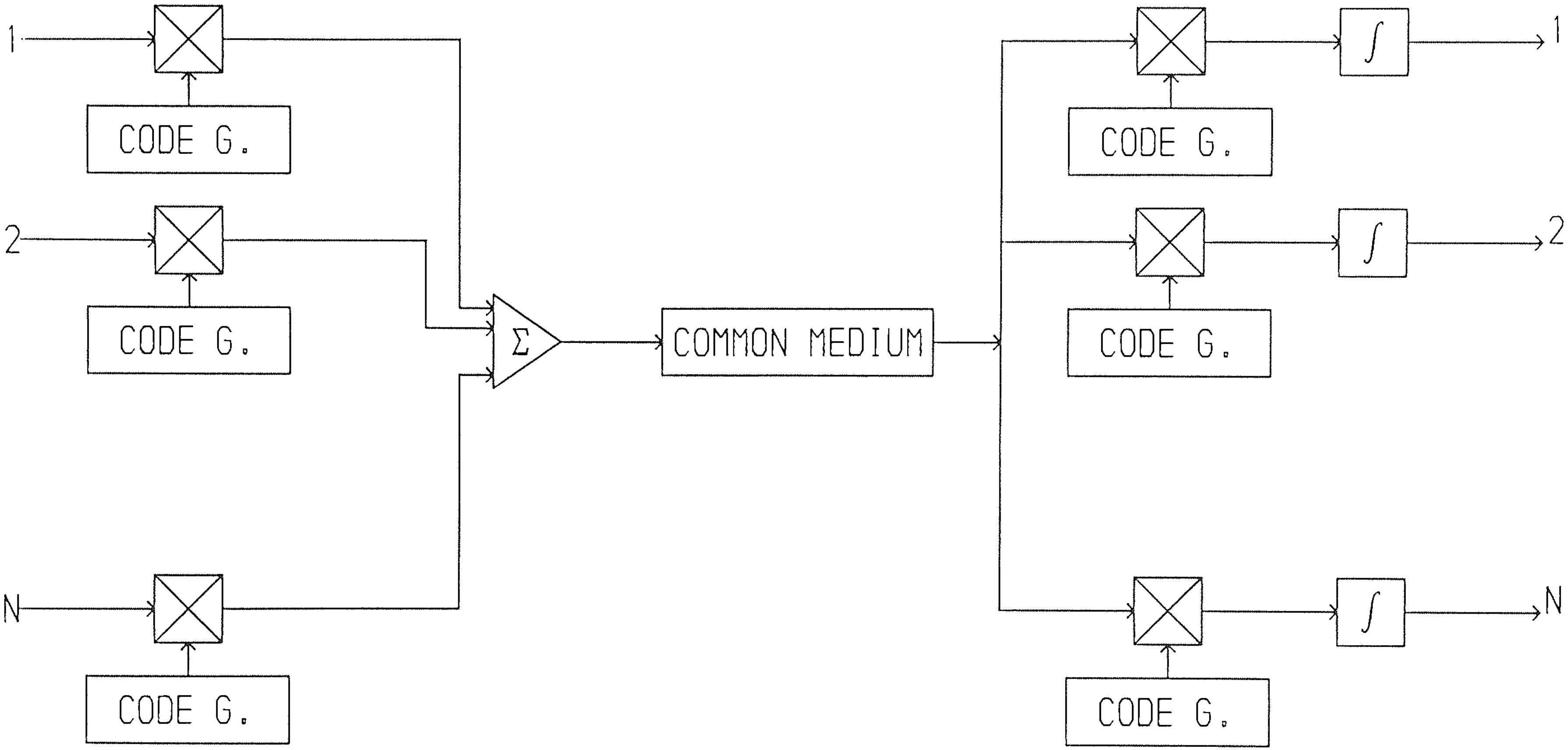


FIG.(3.5): Basic block diagram of code-division multiplexing.

CHAPTER FOUR

4

CODE–DIVISION MULTIPLEXING

4.1 INTRODUCTION:

Prior to 1960 there was a long held belief by communications and telemetry engineers that basic frequency-division and time division were the only two multiplexing methods available to them. However, the theory of orthogonal signal has upset this belief and it is now apparent that the sine wave and the non-overlapping-pulse subcarriers of these methods are merely two of many signal types suitable for multiplexing.

The name ORTHOMUX (meaning orthogonal waveform multiplexing) was adopted to describe the new concept [Ballard 1962] and different names were adopted to describe different approaches to the new concept. Among these names are:
A Boolean-function-multiplexed telemetry system [Titsworth 1963], correlation-recovered adaptive majority multiplexing [Gordon and Barrett 1971], Sequency-division multiplexing [Schreiber 1975 and Qishan 1981] and code-division multiplexing [Schneider 1979 and Goldberg 1981].

In this chapter the development of code-division multiplexing technique is described.

4.2 EARLY CODE-DIVISION MULTIPLEXING SYSTEMS:

Perhaps the earliest mention of multiplexing systems other than the conventional F.D.M. and T.D.M. was given by Ballard in 1962, where he describes a new method of multiplex transmission and reception conceived at Bernard Electronic company in Washington. The new method was referred to by Ballard as ORTHOMUX.

The main feature of ORTHOMUX is the use of orthogonal polynomial waveforms as carrier signals. The set of orthogonal polynomials which was used in the system were the Legendre polynomials. Being orthogonal, the carrier signals may be transmitted simultaneously without interaction over a single wire or radio link.

Information is conveyed in the case of analogue transmission by double side-band amplitude modulation of the carrier waveforms. In the case of digital transmission, information is conveyed by simple polarity reversal of the carrier waveforms . Spectral occupancy of the transmitted signals is at least as efficient as F.D.M. and T.D.M., and in many cases more efficient.

At the receiver , separation of the carriers and recovery of the information in each channel is effected by a matched filter correlation process.

Automatic synchronisation was achieved by sending an unmodulated reference signal over one channel.

The correlation process assures maximum rejection of extraneous noise and interference, while the orthogonality of the carriers assures minimum crosstalk between channels.

One advantage of this system over F.D.M. is that the carriers can be derived from a single oscillator, so eliminating the need for bulky iron-core oscillators, and permits a reduction of unused guard bands. Figure (4.1) shows the polynomial synthesizer for Legendre functions.

In contrast to T.D.M, the system uses a simplified form of automatic synchronisation while eliminates any danger of frame slippage and gives complete flexibility in the number of channels provided.

4.3 CORRELATION-RECOVERED ADAPTIVE MAJORITY MULTIPLEXING:

The mathematical principles of majority multiplexing were derived by R.C. Titsworth in 1963. In his paper, Titsworth described a telemetry system in which information from several data sources is send simultaneously over a single channel by inserting the encoded data inputs (the data inputs multiplied by code sequences) simultaneously into a Boolean function.

As the data source is 0 or 1, either a code word $a^{(i)}$, L symbols long, or its complement, is fed into the logic network. Titsworth proved that the optimum encoding logic f(x) occurs when f(x) is a strict majority logic, provided that the data sources are independant of each other.

The analysis of this system is based on the fact that it is possible to write the output sequence as a sum of terms involving the input codes, weighted by the coefficient determined by the Boolean logic. The crosstalk terms in the system are then evident, and the receiver must minimise channel crosstalk.

The receiver is a correlating device in which a replica of the codes in the transmitter are generated in the receiver. In order to recover the data of the ith channel the received signal is multiplied by the ith code word and integrated, then passed through a decision circuit to decide whether a 0 or +1 has been sent.

Titsworth proved that, for n channels with code words of length L binary characters, the crosstalk is zero provided that:

a) The code words form an algebraic group and $L=2^n$, or

b) The code words are the first n phase shifts of a pseudorandom binary signal of length $L = 2^n -1$.

Thus, for seven channels, the code must be at least 127 character in length. The data could have been conveyed using only seven binary characters, one per channel, in a conventional T.D.M. system. The advantages offered by Titsworth's proposal is the trade-off between redundancy of encoding and the number of channels in use at the same time. Also, by comparison with other pattern-recognition systems, this one uses only a binary signal, which makes the design easy.

In 1968, a system which employs correlation detection of pseudorandom noise carriers was developed by Barrett and Karran. In this scheme, assuming there are n channels, the carriers are the first n phase shifts of a linear maximal-length shift register sequence with cycle length ($2^m -1$) as generated by an n stage shift register with modulo-2 feedback over the first m stages. See figure (4.2).

The multiplexer contains a majority logic device consisting of a summer and threshold detector which generates a function $f(x_t)$, the output of this device at time t being the value +1 if more than n/2 input channels are 1, and the value −1 if less than n/2 channels are 1.

The received signal is then fed to n correlation detectors, each of which is also fed with the relevent phase of the sequence. The component of the rth channel is therefore extracted by correlation against the sequence $x^{(r)}$ (the r-phase shift sequence). The contribution of the rth channel has been maximised in its own channel by the choice of the majority logic, the crosstalk depend ing on the orthogonality of the code sequences. For a quasiorthogonal set of sequences (as in the case of maximal-length shift-register sequences) and for reasonable m, the correlator output depends only on the rth channel signal, with very little crosstalk.

The 2-level autocorrelation characteristic of the maximal-length sequence leads to straight forward recovery of the synchronising information, which controls the phase of the local maximal-length shift register.

In 1971 J.A. Gordon and R. Barrett puplished a paper in which they describe a new method of multiplexing based on a pattern recognition technique, in which binary data are combined for transmission over a common medium by modulation onto binary codewords. A majority logic gate forms the transmitted signal from the modulated codewords. Demodulation is effected by a correlation process using digital matched filters. Truncated Walsh functions were used as codewords in this method.

The prototype of the system is shown in figure (4.3).

The data inputs and the truncated Walsh functions are fed to a series of modulo-2 gates, the outputs of these gates (the modulated data) is then fed to the majority logic function whose output is 1 when more than n/2 of his inputs are 1 and the output is 0 when less than n/2 of the inputs are 1, where n is the number of channels in use at the same time

The multiplexer Walsh function generator also provides a synchronising pulse which is inserted at the end of each frame. This pulse is denoted as the second clock and is derived from another clock, called the first clock, by dividing by 8.

The demultiplexer input is a slicer set at the midpotential of the logic. The signal is then fed into a shift register timed at the second clock rate. The shift register provides the normal and complimentary outputs.

The correlation on the ith channel is performed by a digital matched filter consisting of a resistive star with its jth branch connected to the normal or complimentary output from the register, depending on whether the jth character of the ith code word was a 0 or 1, respectively. The signals are then sliced and taken to a set of flip-flops used to sample and hold the signal at the moment each complete frame just fills the shift register.

The recovery of the clock is performed by the following method: The sliced input to the demultiplexer is differentiated and fullwave rectified, and then fed to a very narrow band pass active filter centred on the clock rate of 2400 Hz. This provides an averaging function on the pulse dither, thus reducing the effect of error. The output from the filter is used to synchronise an astable multivibrator which constitutes that first recovered block proper. This was used to clock the incoming signal stream into the register.

The second clock is recovered by seeking the normal position in the register (when each complete frame just filled the register). In this position the end of frame pulses should appear in the first and last positions in the register (the matched filter connecting with other positions). When a 1 appeared at both ends of the register, the second clock recovery circuit counted 8 and looked again; if a 1 appeared again at both ends it would repeat the count; if not, it would wait until a 1 did appear at both ends. Thus within a few frames, synchronisation must occur.

The necessary requirement for a set of code words to be suitable for the system was found to be:

$$D'_c = S^* \{ A(S^* D_c^T A)^T \}$$

Where A is the code word matrix, D_c is a column matrix whose elements are the data to be transmitted, D'_c is a column matrix whose elements are the estimate of the received data and $S^*(x)$ is defined as the sign of the variable x, such that if x is positive then $S^*(x) = +$ and if x is negative then $S^*(x) = -$.

Several sets of code words were tested to determine their suitability for the system and it was found that for n = L (i.e. number of channels = code length) there are two important sets which are suitable. These are:

1. All cyclic shifts of an m-sequence of length 7.
2. The seven Walsh function sal(1,T), cal(1,t), sal(2,t), cal(2,t), sal(3,t), cal(3,t) and sal(4,t) with the first character omitted in each case (this was the set adopted in the prototype).

Other successful codes were found for n=7, L=8 and n=L=3.

Many other codes were tried and found unsuitable. Among these are:

1. All cyclic shifts of an m-sequence of length-15.
2. A set of truncated Walsh functions of length 15.

The performance of the system depends on the trade-off between the number of channels in use and the redundancy.

One limitation of the system is that the number of channels n must be an odd number to facilitate making a majority decision.

Onother limitation is that the maximum number of channels which can be used when n=L was 7 channels.

However, 6 years later a scheme was suggusted by Mukherjee and Mukhopadhyah (1978), in which the number of channels that may be multiplexed is increased whilst still preserving the advantages of the previous system. This scheme works by using a two stage transformation process. In this process the data vector A of length L is arranged in a matrix form of (mXn) elements, the ith row of this matrix being:

$$A_i = \{ a_{ni-(n-1)}, a_{ni-(n-2)}, \ldots\ldots, a_{ni-1}, a_{ni} \}$$

The matrix A can therefore be written as column matrix:

$$A = \begin{bmatrix} A_1 \\ A_2 \\ . \\ . \\ . \\ A_m \end{bmatrix}$$

Each vector A_i is transformed in the same way as before. The operation is therefore eqivalent to the processing of successive blocks of n data elements serially and then arranging them in a matrix form.

The transformation process can be carried out in two stages. In the first stage the data matrix A(mXn) is first transformed by postmultiplying with the matrix Q(nXn). A majority operation is then carried out on A.Q to obtain a matrix U. In the second stage, U is transformed by premultiplying with the matrix P(mXm). The resultant signal obtained by carrying out a majority operation on the matrix V=P.U

In these two stages Q and P are both formed by code words taken from the seven truncated Walsh functions for which the majority multiplexing is valid.

In the receiver, the estimate of A is also obtained in two steps. In the first step, the received signal is processed with P^T (the transpose of P) and then the majority operation is applied to get a matrix W. In the second step, W is processed with Q^T to obtain the estimate of A.

The estimate of A is error free because both P and Q are formed by code words that satisfy the requirement stated by Gordon.

Since the available code sets that satisfy the requirement form either a (3X3) or (7X7) matrix, then 9, 21, or 49 channels can be multiplexed by using the two stage transform with properly chosen P and Q.

The even active channels (EAC) situation has been studied by Changxin (1980), where he found that, by modification of the criteria of majority operation, it is posible to multiplex an even number of channels. Two criteria of majority operation have been proved to be valid and equivalent for the EAC situation of the 7 channel system using truncated Walsh functions as carrier code words. These two criteria are:

1. $$S^*(x) = \begin{cases} +1 & x \geq 0 \\ -1 & x < 0 \end{cases}$$

2. $$S^*(x) = \begin{cases} +1 & x > 0 \\ -1 & x \leq 0 \end{cases}$$

Changxin proved mathimatically that when the number of active channels is 2 or 6, any code word set taken from the 7 truncated Walsh functions can be used as carriers. But it is not true for the case of 4 active channels. In this case he proved that the code word sets should be selected in such a way that the unused 3 Walsh functions form a semi-group. Using this condition he found 7 sets which can be used as carrier code words for the 4 active channel situation.

4.4 SEQUENCY DIVISION MULTIPLEXING:

The word sequency has been adopted as being consistent with the well known term, frequency. The frequency of a sinusoidal function is defined as one half the number of zero crossing per unit of time. Similarly the sequency of a Walsh function is defined as one half the average number of zero crossing per unit time [Harmuth 1969 and 1971].

Sequency-division is the term adopted for multiplexing systems which use Walsh functions as subcarriers [Qishan 1980, 1981 and 1982]. A block diagram of a sequency-division multiplexing system is shown in figure (4.4), which is suitable for transmission of 16 signals..

At the transmitter the signals first pass through sequency low pass filters TP. The outputs of these filters are fed to the multipliers M, at the same time Walsh functions are fed to the multipliers M from Walsh functions generator WG.The Walsh functions generator is triggered by the pulse generator and controller CPC. After the modulation process is performed in the multipliers M, The amplitude modulated signals are added in a SUM stage, from which a combined signal is obtained.

At the receiver the channel separation is achieved by correlation. The received signal is fed to all the demodulators and to the synchronization circuit S. The Walsh function generator is triggered by the synchronisation signal which is selected in S so that synchronism is maintained between the Walsh functions generators at the transmitter and the receiver. The received signal is multiplied by the different Walsh functions which are fed into the multipliers M. The desired signal is separated by its property of orthogonality.
The crosstalk between the channels is not identically zero, a proper choice of the subcarriers minimises the crosstalk.

4.5 CONCLUSION:

The adaptive majority multiplexing method using truncated Walsh functions as carrier codes suggested by Gordon and Barrett in 1971 offered a trade-off between redundancy of encoding and the number of channels in use at the same time. This advantage has attracted the research attention , so that, the necessity of the number of channels being odd is now no longer applicable if the majority decision criteria are changed accordingly. There has also been an attempt to increase number of channels which can be handled by this method but still using the same 7 truncated Walsh functions.

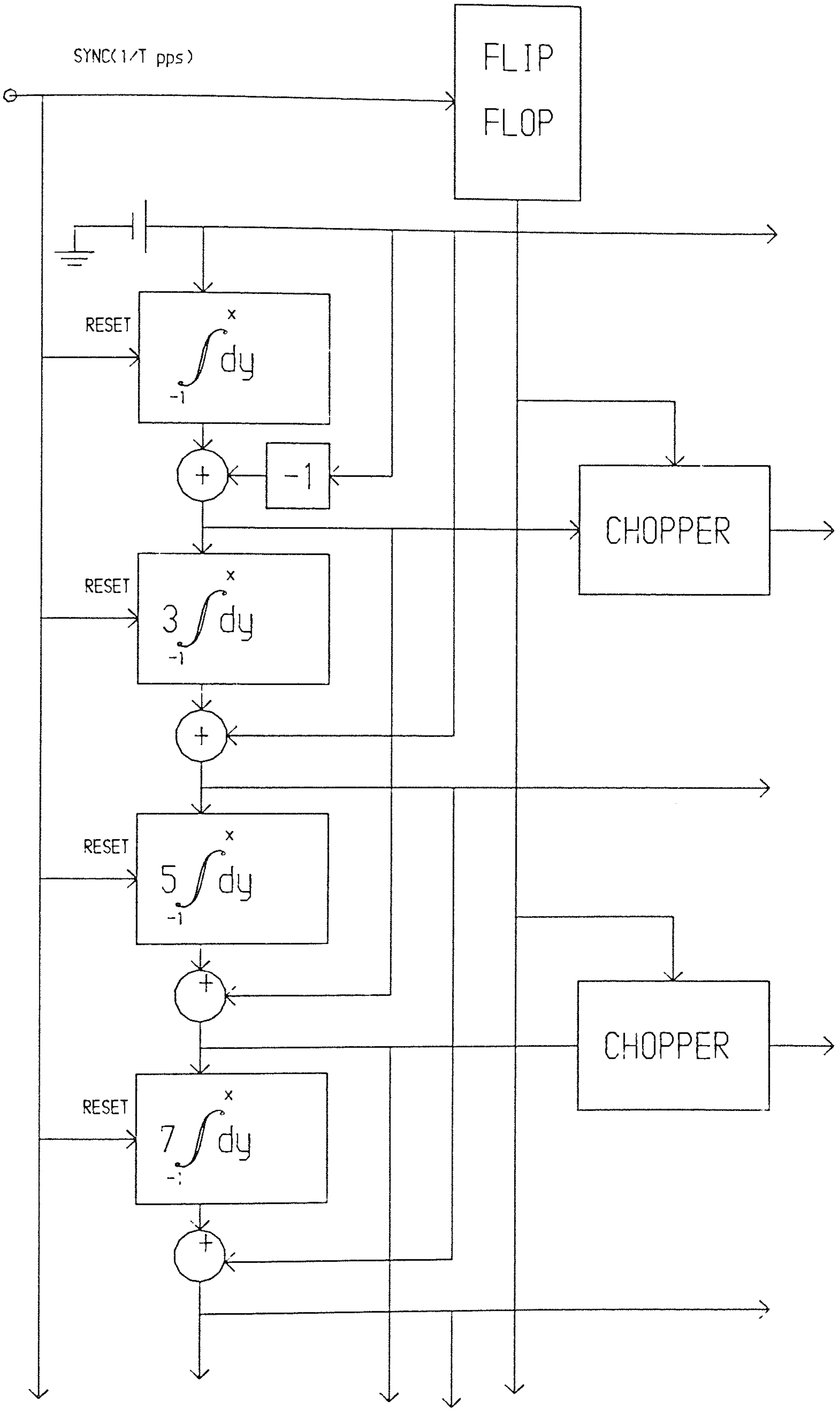


FIG.(4.1):Legendre polynomial waveform synthesizer.

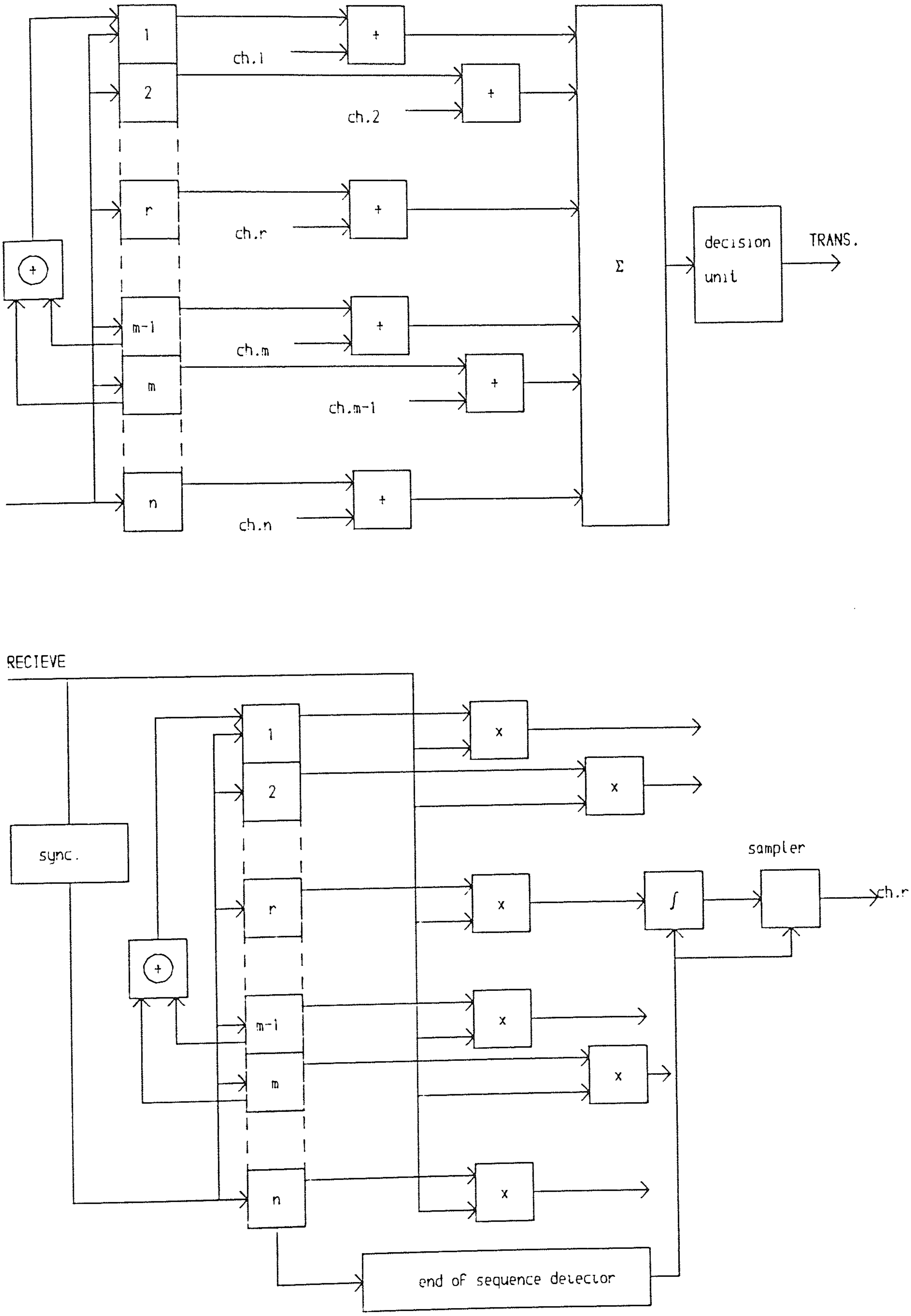


FIG.(4.2): Correlation-multiplex data transmition system.

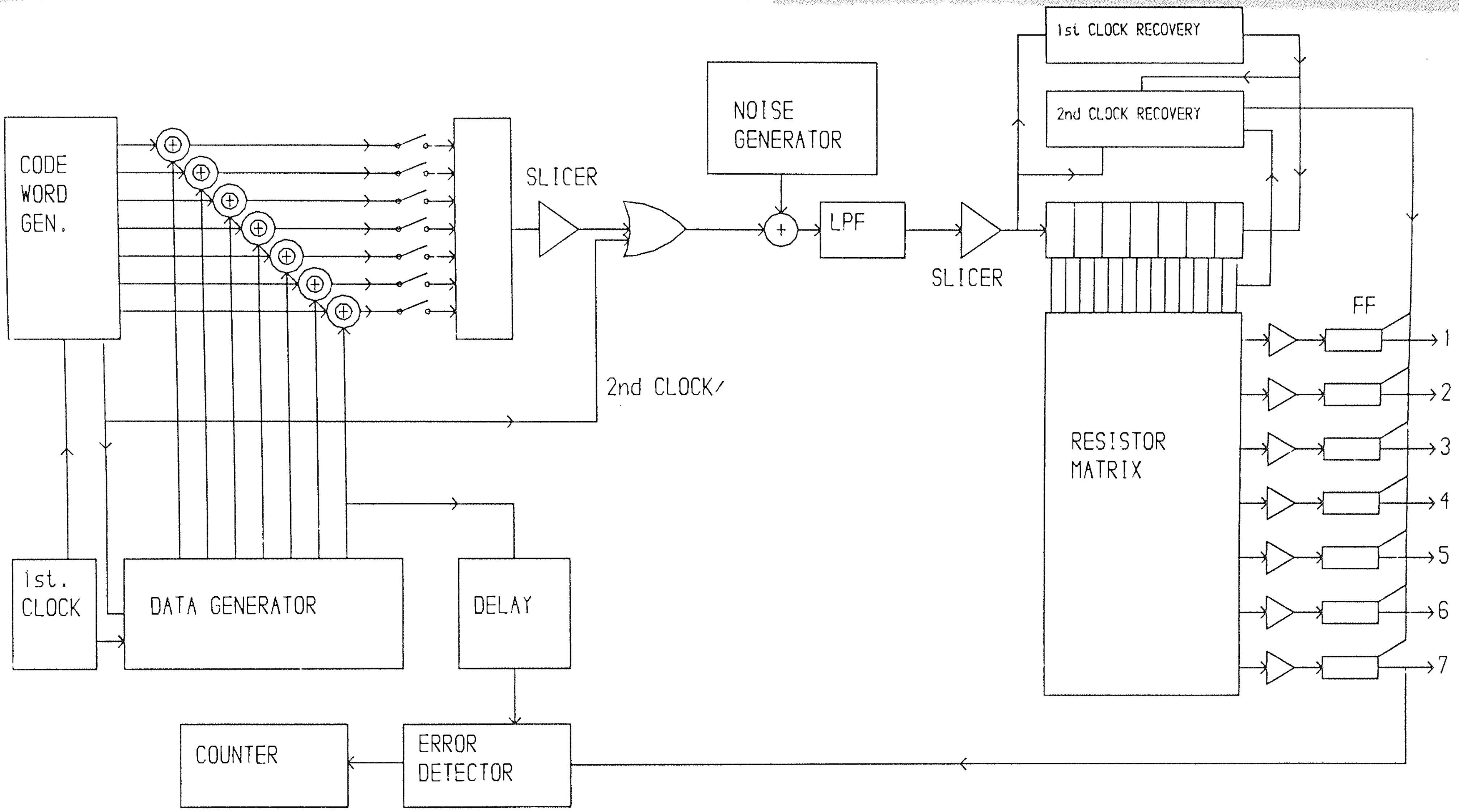


FIGURE (4.3): ADAPTIVE MAJORITY LOGIC MULTIPLEXING PROTOTYPE.

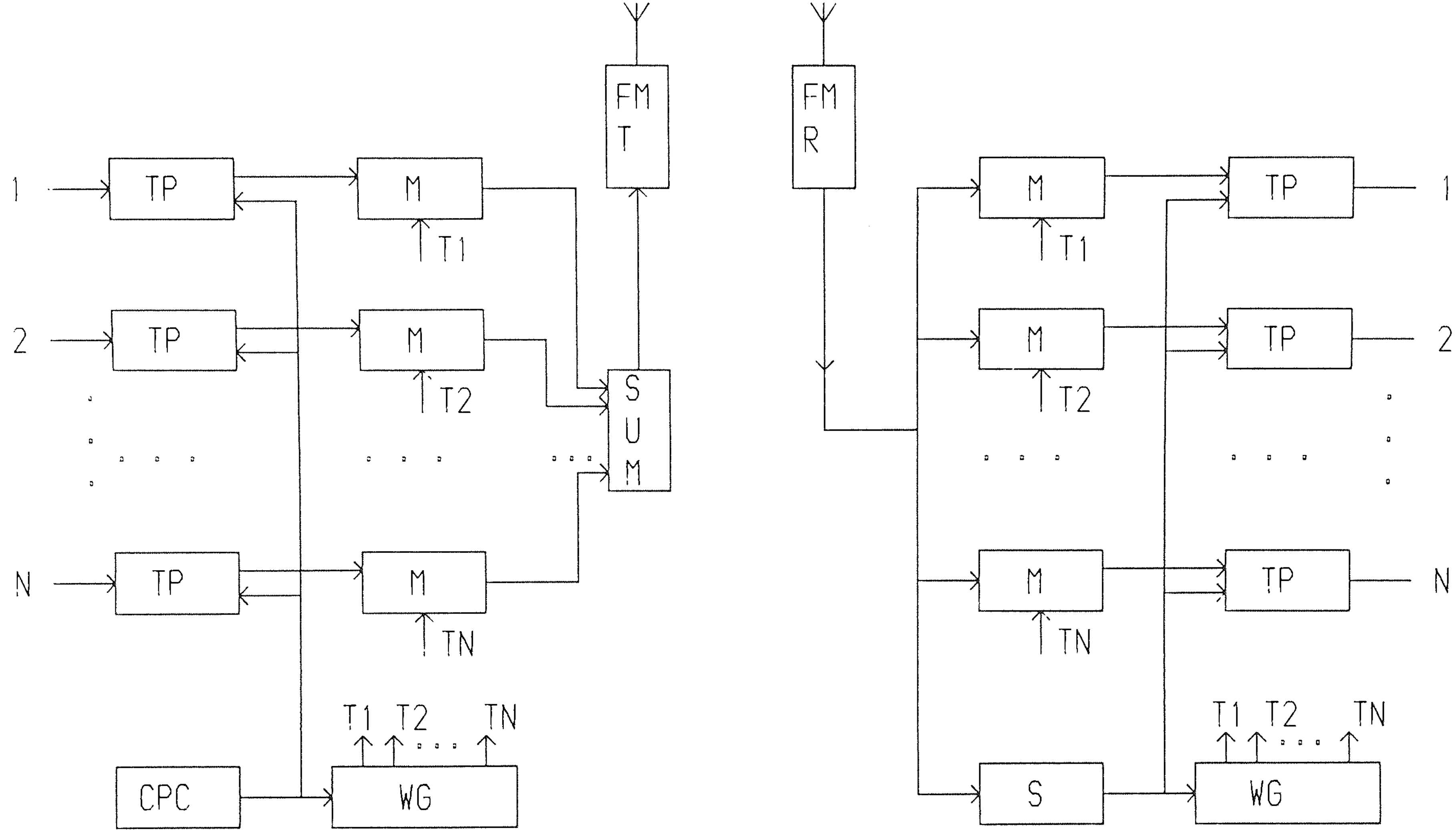


FIG.(4.4): BLOCK DIAGRAM OF SDM.

CHAPTER FIVE

5

SYSTEM DEVELOPMENT

5.1 INTRODUCTION:

In this chapter a new system of code division multiplexing will be described, this system uses as codes for its channels a set of orthogonal functions called bridge functions.

The derivation of a new formula which gives the mathematical model for the proccess of the above system is presented. This formula gives the necessary requirement for a set of functions to be suitable as codes for this system.

A flow chart of a computer program based on the mentioned formula is presented. Also some results which show the suitability of bridge functions as code sets for the system are briefly mentioned.

5.2 DESCRIBTION OF THE SYSTEM:

The system can be divided into three parts for clarity, these are:

1. The multiplexer.
2. The demultiplexer.
3. The channel simulator.

Figure (5.1) shows the basic block diagram of the system for multiplexing 8 channels.

5.2.1 The multiplexer:

The multiplexer basically consists of data sources, bridge functions generator, multipliers, adder and timer.

The information bit from each data source is fed into one input of a multiplier while the other input of the multiplier is connected to the corresponding bridge function. Each data bit is multiplied by one bit from the bridge function and then the outputs of the multipliers are added in the adder. The result of the addition is compared with a threshold level which gives a three valued output 0, +1 or -1. One of these values is sent to the line and transmitted.

The multiplexing of 8 bits (one bit from each data source) is completed by sending 8 bits to the line. Hence the bit rate of the bridge function generator is 8 times the data rate. i.e the timer will send a clock signal to the the bridge functions generator at a rate equal to 8 times the one which it send to the data sources.

The function of the multiplexer is better explained by an example.

suppose that the data sources have the following information bits [+ + + + - + - -]. i.e. ch 1 has +1 bit, ch2 has +1 bit, etc. The proccess of multiplexing is as follows:

1. The first bit of each bridge function is multiplied by the corresponding data bit. If the first bit of each code is put under the corresponding data bit then the multiplication is carried out as follows:

```
+ + + + - + - -
+ 0 + 0 + 0 + 0   x
---------------
+ 0 + 0 - 0 - 0
```

2. The multiplication results are then added.

   In this example the addition of the resultant multiplication is A = +2-2 = 0 .

3. The result A is then compared with a threshold level.

If $A = 0$ then a 0 bit is fed to the line.

If $A > 0$ then a +1 bit is fed to the line.

If $A < 0$ then a -1 bit is fed to the line.

So that, in this example, a 0 bit is fed to the line, which is the first bit of 8 to be sent.

4. To get the 2nd, 3rd, ..., 8th bit, procedure 1, 2, and 3 are repeated but instead of taking the 1st bit of each code in 1. the 2nd, 3rd, ..., 8th bit is to be taken in turn.

Figure (5.2) shows the complete process of multiplexing the data bits [+ + + + - + - -] to get the transmitted line bits; [0 + + + 0 - + 0].

5.2.2 The de-multiplexer:

The de-multiplexer basically consists of a bridge functions generator, multipliers, integrators and timer.

The received signal is first passed through a timing extraction circuit (timer) which controls the timing of the bridge functions generator and the integrators.

The signal then is correlated with each of the code words, this correlation is done by multiplying each bit from the received signal with each bit from the corresponding code words.

A negative correlation will be taken as indicating that the corresponding channel was sent a "-1" data bit, otherwise a "+1" is assumed to have been transmitted.

To clarify the proccess of correlation, the previous example of multiplexing the data bits [+ + + + - + - -] is continued here.

The transmitted signal was [0 + + + 0 - + 0].

1. correlation of this signal with the code word of the first channel (Bo = + 0 0 + + 0 0 +) gives:

$$\begin{array}{l} 0 + + + 0 - + 0 \\ \underline{+ 0 0 + + 0 0 +} \quad x \\ 0 0 0 + 0 0 0 0 \end{array}$$

2. The integration of the results of the correlation is positive, so that a "+1" bit is assumed to be transmitted.

3. By carrying on the correlation of the transmitted signal with each of the code word and integrating the results of each correlation we get the vector [+ + + + - + - -] which is exactly the multiplexed data vctor.

This example is shown in figure(5.3).

5.2.3 The channel simulator:

The multiplexer output is fed to a channel simulator which consists of a mixer in which the output of a white Gausian noise generator is added to the signal.

5.3 MATHEMATICAL MODEL:

In this section a mathematical model for the code-division multiplexing system using the 3-valued functions as code words is derived.

Let us consider a system with n channels, this needs to use n code words, one for each channel. When Bridge functions are used as code words then the number of characters in each code word is n. Each character is either +1, -1 or 0. Let B be the matrix representation of the set of code words (Bridge functions) whose elements are given as follows:

$$B = \begin{bmatrix} b_{11} & b_{12} \ldots b_{1j} \ldots b_{1n} \\ b_{21} & b_{22} \ldots b_{2j} \ldots b_{2n} \\ \ldots & \ldots\ldots\ldots\ldots \\ b_{i1} & b_{i2} \ldots b_{ij} \ldots b_{in} \\ \ldots & \ldots\ldots\ldots\ldots \\ b_{n1} & b_{n2} \ldots b_{nj} \ldots b_{nn} \end{bmatrix}$$

where the element b_{ij} in B is the jth character in the ith code word.

The ith row of the matrix B is B_i and represents the code word for channel i.

$$B_i = [\, b_{i1} \;\; b_{i2} \;\; \ldots\ldots \; b_{in} \,] \qquad (2)$$

The jth column of the matrix B is B_j and is a vector consisting of all the jth characters of all the code words, the jth time slot vector;

$$B_j = \begin{bmatrix} b_{1j} \\ b_{2j} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ b_{nj} \end{bmatrix} \qquad (3)$$

So that B can be expresed as:

$$B = \begin{bmatrix} B_1 \\ B_2 \\ \cdot \\ \cdot \\ \cdot \\ B_n \end{bmatrix}$$

Or $$B = [\, B_1 \; B_2 \ldots B_j \ldots B_n \,]^T \qquad (5)$$

The suffix i will always indicate the channel number and may take the values $1 \leq i \leq n$. The suffix j will always indicate the time slot number and may take the values 1 to n.

A row vector corresponds to consecutive time slots, and a column vector to a simultaneous presentation of a number of components.

Let the input data to the system be:

$d_1, d_2, \ldots, d_i, \ldots d_n$ where $d = \pm 1$

Then the matrix D whose diagonal elements is the input data will be called the data matrix;

$$D = \begin{bmatrix} d_1 & 0 & 0 & \cdots & 0 \\ 0 & d_2 & 0 & & 0 \\ 0 & 0 & d_3 & & \vdots \\ \vdots & & & & \vdots \\ 0 & 0 & 0 & \cdots & d_n \end{bmatrix} \qquad (6)$$

A column matrix whose elements are the data input will be called data column matrix D_c ;

$$D_c = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix} \qquad (7)$$

The row matrix, which is the transpose of the data column matrix D_c is given by ;

$$D_c^T = [\, d_1 , d_2 , \ldots\ldots , d_n \,] \qquad (8)$$

The i-th code word is modulated by the data from the i-th channel.

Thus M_i , the ith code word modulated by d_i is given by ;

$$M_i = d_i B_i \qquad (9)$$

and the matrix of modulated code words M is given by;

$$M = D B \qquad (10)$$

and a time slot vector of M is given by;

$$M_j = D B_j \qquad (11)$$

The adder at the multiplexer will add the elements of the time-slot vector M_j and transmit to the line the sum of these elements, so that if the sum of the elements of a column vector M_j is denoted by c_j then;

$$c_j = \sum_{\text{all } i} m_{ij} \qquad (12)$$

The row matrix $C = [\, c_1 , c_2 , \ldots , c_n \,]$ which is the row vector of the transmitted signal (line signal) is given by:

$$C = D_c^T B \qquad (13)$$

According to the structure of the code words (Bridge functions) the amplitude of the transmitted signal is given by the inequality;

$$-\frac{n}{2} \leq c \leq \frac{n}{2} \qquad (14)$$

ie. $c = -\frac{n}{2} , \ldots\ldots , 0 , \ldots\ldots , \frac{n}{2}$

In fact this signal is sliced and sent as a ternary signal by taking only the values +1, -1, and 0 ; so that if the value of c_j is positive then +1 is transmitted, -1 is transmitted when the value is negative and 0 is transmitted when the value is 0.

The transmitted signal for n channels consists of n digits. At the demultiplexer this signal is correlated with each of the code words. This correlation is done by multiplying each digit from the transmitted signal with each digit from the corresponding code words.

A negative correlation will be taken to indicate that the corresponding channel sent a "-1", otherwise a "+1" is assumed to have been transmitted.

The non-normalised correlation coefficient of the signal with the i-th code word is given by;

$$p_i = \sum_{\text{all } j} c_j b_{ij} \qquad (15)$$

The correlation coefficient of all channels is a column vector P and is given by;

$$P = B C^T \qquad (16)$$

where C^T is the transpose of C.

If the received data on the ith channel is denoted by d'_i, where it has been assumed that $d'_i = \text{sign } p_i$, then the column vector of the received data is given by;

$$\begin{aligned} D'_c &= \text{sign } P \\ &= \text{sign } (B C^T) \\ &= \text{sign } \{ B (D_c B^T)^T \} \end{aligned}$$

$$= \text{sign} \{ B B^T D_c \} \qquad (17)$$

Thus the necessary requirement for a set of code words to be suitable for this system is;

$$D = \text{sign} \{ B B^T D_c \} \qquad (18)$$

If the code words are orthogonal then;

$$B B^T = I \qquad (19)$$

Then $D'_c = \text{sign} (I D_c)$

$$= \text{sign} (D_c) \qquad (20)$$

5.4 COMPUTER PROGRAM:

A computer program has been written to determine whether a given set of code words fulfils equation 18.

A set of code words B is fed into the computer, along with the parameter n (the number of channels in use). Also the program uses , as data for the channels, a random function generator,IRAN, which gives all the 2^n posible forms of data vectors.

The program perform the multiplexing and the de-multiplexing proccess as explained earlier, the output of the de-multiplexing proccess is compared with the data vector, and, in the event of any discrepancy on any one of the channels, an error is printed. A flow chart of the program is shown in figure (5.4).

Using this program, several sets of suitable code words have been found. These sets are divided into three groups:

Group A : which contains two sets:

1. The set of bridge functions Bri(i,1,3,t).
2. The set of bridge functions Bri(i,2,3,t).

Group B : which contains three sets:

1. The set of bridge functions Bri(i,1,4,t).
2. The set of bridge functions Bri(i,2,4,t).
3. The set of bridge functions Bri(i,3,4,t).

Group c : which contains four sets:

1. The set of bridge functions Bri(i,1,5,t).
2. The set of bridge functions Bri(i,2,5,t).
3. The set of bridge functions Bri(i,3,5,t).
4. The set of bridge functions Bri(i,4,5,t).

The waveforms and the matrix representation of these sets are shown in the appendix.

The most important sets of all these are the first set of each group i.e;

Bri(i,1,3,t), Bri(i,1,4,t) and Bri(i,1,5,t).

It has been found that the above sets of bridge functions are suitable as codes for the system if all the channels are in use i.e. the system is fully loaded. However, they are also suitable for any number of active channels if a suitable choice of the bridge functions has been made for the codes for the active channels. An unsuitable choice of the codes leads to a disasterous error such that some channels are always in error.

This point will be explained later in this chapter and the suitable choice of the code words for any number of active channels will be discussed.

5.4 GROUP STRUCTURE OF BRIDGE FUNCTIONS:

The structure of the above mentioned bridge function sets have some interesting properties which have been found during the research. The important properties which are related to the work are listed below:

1. The set of bridge functions Bri(i,1,3,t) contains two algebraic groups, each one of these groups consists of three bridge functions, where the term-by-term product of two bridge functions gives the third bridge function in the group. Thus if B(i), B(j) and B(k) are three bridge functions in the set and have the property that:

B(i)B(j) = B(k) then

B(j)b(k) = B(i) and

B(i)b(k) = B(j)

Hence these three bridge functions form an algebraic group.

The two algebraic groups in the set Bri(i,1,3,t) are:

a) Bri(i,1,3,t) for i=2,4,6

b) Bri(i,1,3,t) for i=3,5,7

2. The set of bridge functions Bri(i,1,4,t) has a number of three element algebraic groups each of which has the same property as the one mentioned in 1. for the set Bri(i,1,3,t). These groups are:

Bri(i,1,4,t) for i=2,4,6 i=3,5,7

i=2,8,10 i=3,9,11

i=2,12,14 i=3,13,15

i=4,8,12 i=5,9,13

i=4,10,14 i=5,11,15

i=6,8,14 i=7,9,15

i=6,10,12 i=7,11,13

Also this set, Bri(i,1,4,t), has the property that:

If B(i), B(j), B(k) and B(n) are elements in the set where;

B(i)B(j)B(k) = B(n) then

B(i)B(j)B(n) = B(k)

B(i)B(n)B(k) = B(j)

B(j)B(k)B(n) = B(i)

There are 14 quadruple group of this type these are:

Bri(i,1,4,t) for i=2,4,10,12 i=3,5,11,13

i=2,4,8,14 i=3,5,9,15

i=2,6,8,12 i=3,7,9,13

i=2,6,10,14 i=3,7,11,15

i=4,6,8,10 i=5,7,9,11

i=4,6,12,14 i=5,7,13,15

i=8,10,12,14 i=9,11,13,15

3. The set of bridge functions Bri(i,1,5,t) has a number of triple element groups and a number of quadruple element groups similar to those mentioned in 1 and 2. This set also has a number of quintrøuple element groups in which if B(i), B(j), B(k), B(n) and B(m) are elements in the set Bri(i,1,5,t) and;

$B(i)B(j)B(k)B(n) = B(m)$ then

$B(i)B(j)B(k)B(m) = B(n)$

$B(i)B(j)B(n)B(m) = B(k)$

$B\ i)B(n)B(k)B(m) = B(j)$

$B(j)B(k)B(n)B(m) = B(i)$

5.6 DETERMINISTIC ERROR:

In the system of multiplexing described here, a deterministic error occurs if only two channels are turned off, the error then occuring in the third of the triples. In order to avoid this kind of error when two channels should be turned off, the two channels should not be from the same triplet group. Also, if three channels should be turned off then they should be from the same triplet.

5.7 SELECTION OF BRIDGE FUNCTIONS SUBCARRIERS:

The bridge functions are used as subcarriers in the system.

A proper choice of the subcarriers minimises the cross-talk between the channels. The principles for selecting bridge functions as subcarriers are as follow:

1. If three channels are turned off, they should be from the same triplet. This rule applies for all the sets.
2. When four channels are turned off and the set of bridge functions has quadruple groups, it is preferred that the four channels be a quadruple.

3. When five channels are turned off and the set of bridge functions has quintet groups, it is also prefered that the five channels be a quintet.

4. Where it is possible, the number of bridge functions with odd order should be equal to the number of bridge functions with even order, otherwise the error will not be equally distributed over the channels. i.e. some channels will have more error than the others.

As an example of selection of subcarriers table (5.1) shows the recommended selection of subcarriers for the set Bri(i,1,3,t).

no. of ch.	recommended order numbers
2	any two (one with odd order)
3	2,4,6 or 3,5,7
4	0-3 or 1-4
5	1,2,3,4,6
6	2-7
7	1-7

Table (5.1): the recommended bridge functions from the set Bri(i,1,3,t).

5.8 CONCLUSION:

A code division multiplexing system based on bridge functions as codes for its channels has been described. A mathematical model for the system which gives the necessary requirement for a set of codes to be suitable for this system is also presented.

A number of bridge function sets have been tested and found to fullfil the requirement and hence are suitable as codes for the system provided that the codes have been chosen properly according to the number of channels in use. Some rules for the proper choice of bridge functions were briefly mentioned.

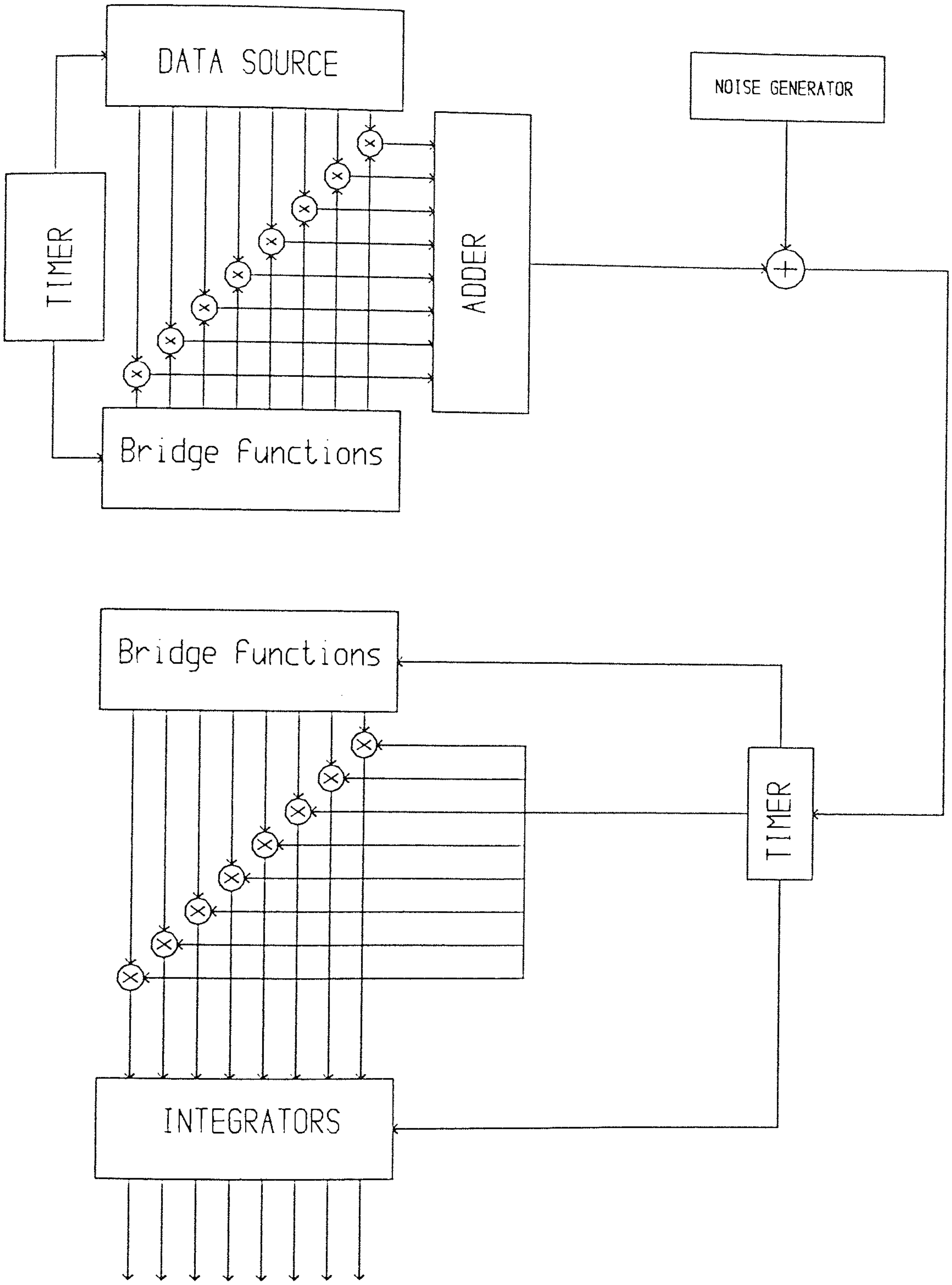


FIG.(5.1) :The basic block diagram of code division multiplexing system

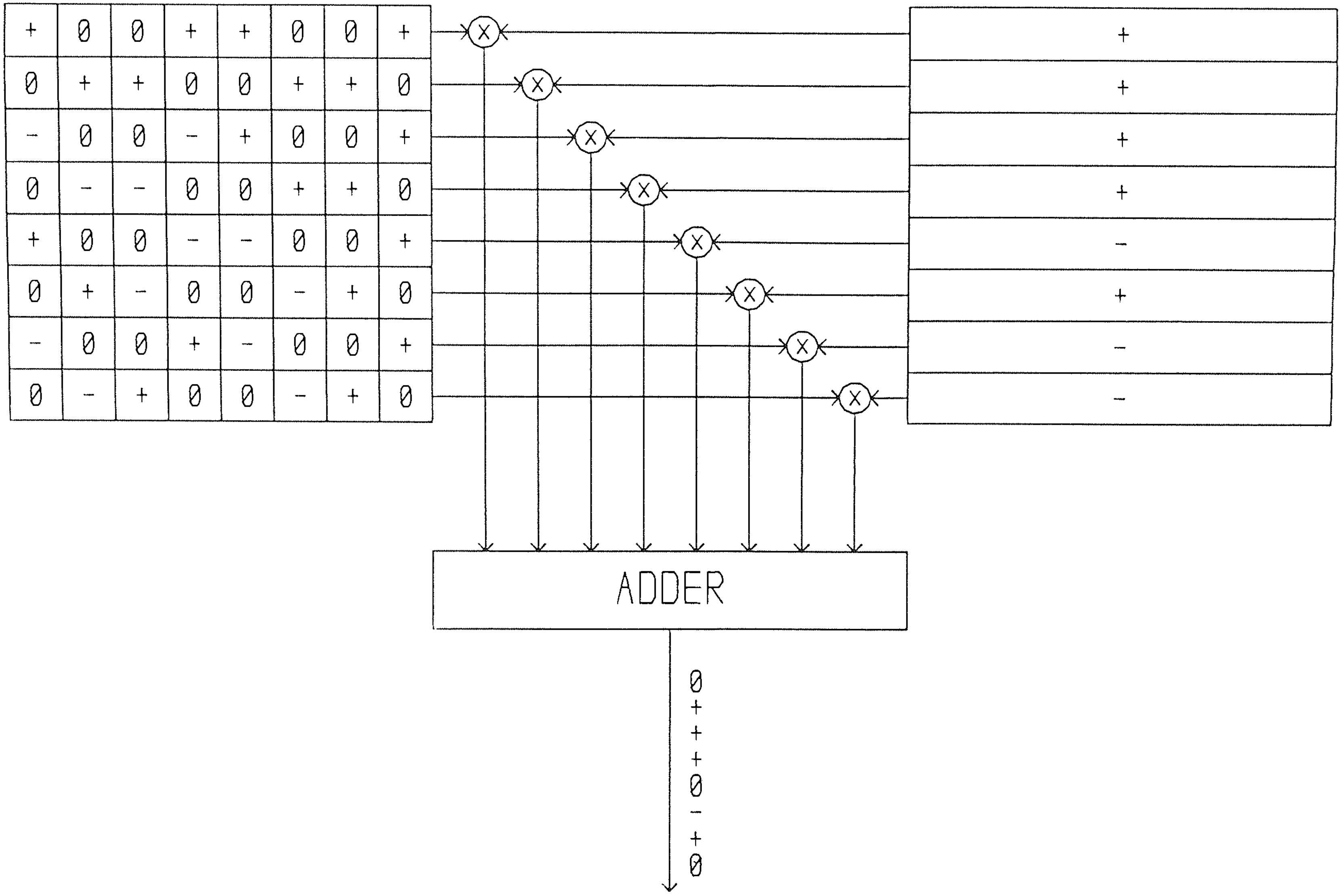
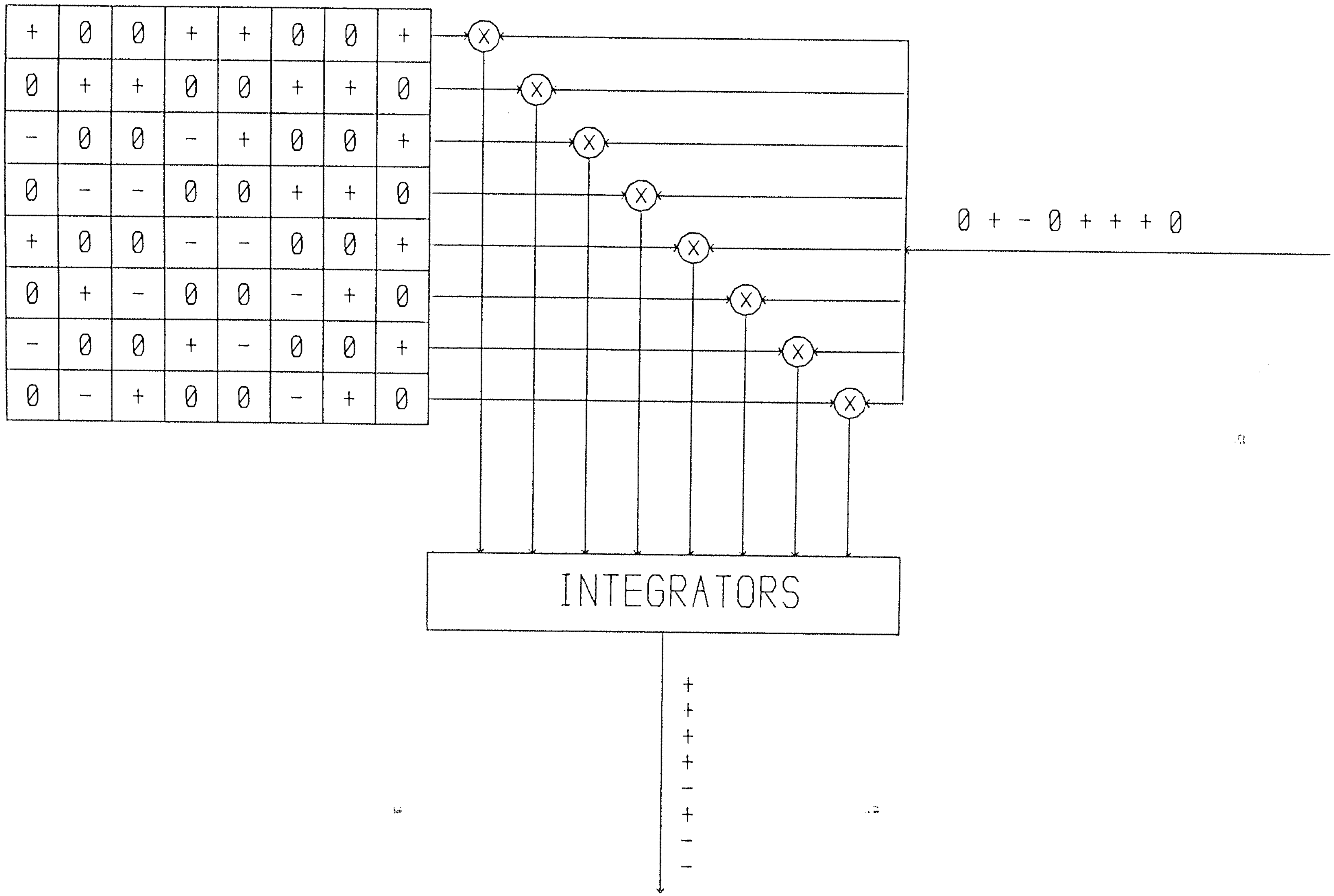


FIG. (5.2) : An example to illustrate the multiplexing proccess.



+	0	0	+	+	0	0	+
0	+	+	0	0	+	+	0
−	0	0	−	+	0	0	+
0	−	−	0	0	+	+	0
+	0	0	−	−	0	0	+
0	+	−	0	0	−	+	0
−	0	0	+	−	0	0	+
0	−	+	0	0	−	+	0

FIG. (5.3) : An example to illustrate the demultiplexing proccess.

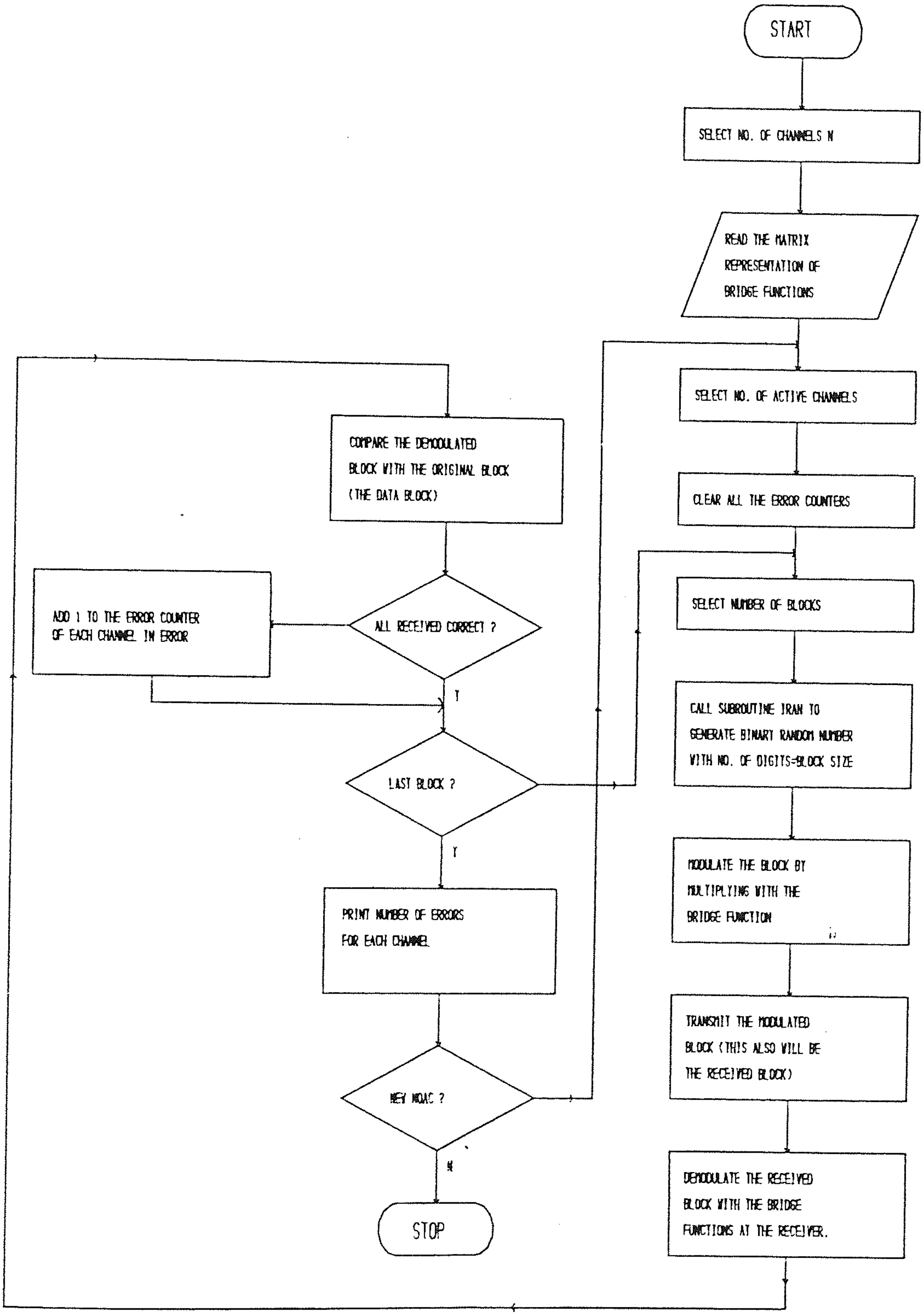


FIG.(5.4): FLOW CHART.

CHAPTER SIX

6

SYSTEM PERFORMANCE AND RESULTS

6.1 INTRODUCTION:

This chapter is devoted to the results obtained from a computer simulation program which determines the performance of a code-division multiplexing system using bridge functions as codes for its channels. The flow chart of the program is very briefly explained first.

The results are divided into three sets:

1. The results obtained from a system which consists of 8 channels. This system uses the set of bridge functions Bri(i,1,3,t).
2. The results obtained from a system which consists of 16 channels. This uses the sets of bridge functions Bri(i,1,4,t), Bri(i,2,4,t) and Bri(i,3,4,t).
3. The results obtained from a system which consists of 32 channels, using the sets of bridge functions Bri(i,1,5,t), Bri(i,2,5,t) and Bri(i,3,5,t).

Each set of these results shows:

a. The probability of channel error against the signal-to-noise ratio for every number of active channels when the system is subject to White Gaussian noise.

b. The error probability of the aggregate signal (the received signal before demodulation) against the channel error, again for different numbers of active channels and for W.G.N.

c. Tables showing number of blocks received in error and the number of blocks which have been corrected. The same tables show the number of blocks received with 1,2,...9 errors and their corresponding number of blocks which have been corrected after demodulation.

The chapter ends with a discussion of the results.

6.2 FLOW CHART OF A COMPUTER PROGRAM:

A computer program has been written in FORTRAN to simulate the performance of the system in the presence of additive white Gaussian noise. A flow chart of the program is shown in figure (6.1). The flow chart is general purpose in the sense that it can be used for different numbers of channels by slightly modifying some of the parameters.

Using the program, results are obtained for the three cases of 8, 16 and 32 channel systems. The results from each case show the trade-off between number of active channels (channels in use) and the probability of error, i.e. the increase in error correcting properties as the number of non-active channels (channels which are turnned off) is increased.

When run the program, reads the matrix representation of the bridge functions set, which is stored in the memory, and then selects the number of active channels. A number of values (sigma) for the variance of a white Gaussian noise generator in the range 0.15 to 1.0 are also stored and these are selected sequentially through the range. The mean value for the noise generator is taken to be zero.

A number of counters are used to store the errors. These are:

1. The aggregate error counter. This stores the errors in the signal stream before it reaches the demultiplexer.

2. The channel error counters. One for each channel, these counters are used to store the number of errors for each channel and for a specific number of blocks to be transmitted.

3. The block error counters. There are two types of these counters, the first is used to count the total number of blocks at the input of the demultiplexer with 1,2,3...9 errors and the second type is used to count the total number of blocks with 1,2...9 errors at the output of the demultiplexer.

All the above counters must be cleared for each value of noise variance (sigma).

Number of blocks to be transmitted can also be selected. This parameter must be high enough for an error to occur in the case of small values of sigma and high redundant blocks.

The data source for the system consists of a binary random number generator (subroutine IRAN) with the number of digits equal to the block size.

A flow chart for the transmission of one block of data is illustrated in figure (6.2). The data block is first modulated by multiplication with the bridge functions as explained in chapter 5. The noise is then added to the block from the white Gaussian noise generator (subroutine RANN) which has two parameters, the mean value which is the mean of a normal distribution (this is zero) and sigma which is the magnitude of one standard deviation. the values returned from RAAN are distributed as folows:

67.77 % of the total number are in the range ± sigma.

27.86 % of the total number are in the range ± 2sigma.

4.18 % of the total number are in the range ± 3sigma.

0.19 % of the total number are in the range ± 4sigma.

The sequence of the generated random number is primed after each complete transmission of the number of blocks to be transmitted. After the addition of noise to the data block the resultant block bits are compared with a threshold level in the following manner: If the amplitude of the bit is greater than +0.5 then a +1 bit is assumed to be sent, if it is less than -0.5 then a -1 bit is assumed to be sent, otherwise a 0 bit is assumed.

Each bit of the block is compared with the corresponding modulated block bit and for every discrepancy the aggregate error counter is incremented by one. Also the total number of errors in the block is counted and the appropriate block error counter is incremented if necessary.

The flow chart for the proccessing of a received block is shown in figure (6.3). The received block is first stored for processing and a correlation is applied for each channel code as explained in chapter 5. The correlation coefficient is compared with a threshold level so that a negative correlation results in the assumption that a -1 bit has been received, otherwise a +1 bit is assumed to be received for the corresponding channel.

A comparison between the bit received and the original transmitted bit is made and, in the case of discrepancy, the corresponding counter is incremented. Also the total number of errors in the block is counted and the corresponding block error counter is incremented.

When the last block has been processed, the contents of all the error counters are printed and stored.

Another value of sigma is then selected and the whole procedure is repeated until the result for the last value of sigma has been printed.

6.3 SIGNAL-TO-NOISE RATIO CALCULATIONS:

The relationship between the signal-to-noise ratio of the transmitted signal and the variance of the White Gaussian distributed noise source will be now derived.

The transmitted signal is a ternary signal where the signal levels are +1, 0 and -1. It has been found by using a computer program that on average, the occurence probability of +1's and -1's are equally probable, and the occurence probability of 0's is twice that of +1's or -1's. Hence if P_0 is the occurrence probability of 0's then

$$P_0 = 2P_{+1} = 2P_{-1} \qquad (6.1)$$

where P_{+1} is the occurence probability of +1's and P_{-1} is the occurence probability of -1's.

From (6.1) it follows that:

$$P_0 = 0.5 \quad , \quad P_{+1} = 0.25 \text{ and } P_{-1} = 0.25 \qquad (6.2)$$

The average power of the signal is given by:

$$S = P_0 \cdot s_0 + P_{+1} \cdot s_{+1} + P_{-1} \cdot s_{-1} \qquad (6.3)$$

$$= 0.5 \cdot (0)^2 + 0.25 \cdot (+1)^2 + 0.25 \cdot (-1)^2 = 0.5$$

The average power N, of White Gausian noise generator with mean $= 0$ and mean squared value σ^2 is given by:

$$N = \sigma^2$$

Hence signal-to-noise ratio

$$(S/N) = 0.5\ \sigma^2 \qquad (6.4)$$

$$(S/N)_{db} = 10\ LOG10\ (0.5\ \sigma^2) \qquad (6.5)$$

6.4 ERROR PROBABILITY:

In a digital communication system digital information is passed from one point to another. The digital communication channel carrying the information may be a cable, radio link or fibre optic channel. At the receiving end there is a decision device which decides, out of the known possible digital states, which state is be ing sent.

In the presence of noise the decision device is liable to make occasional error. The errors depend on the decision device which has been used and, more important, on the type of noise and on the signal-to-noise ratio.

The relationship between probability of error and signal-to-noise ratio in the presence of Additive White Gausian noise will be derived for the transmitted signal. Also a statistical method to measure the error probability will be presented.

6.4.1 Probability of error and signal-to-noise ratio:

The probability density function for a signal centred at level 0 and subject to Gausian noise with a mean squared value σ^2 is given by:

$$P_0(v) = \frac{1}{\sqrt{2\pi}\,\sigma} \operatorname{Exp}(-v^2 / 2\sigma^2) \qquad (6.6)$$

and for a signal centred at level +1, the probability density function is given by:

$$P_1(v) = \frac{1}{\sqrt{2\pi}\,\sigma} \operatorname{Exp}(-(v-1)^2 / 2\sigma^2) \qquad (6.7)$$

and for a signal centred at level -1, is given by:

$$P_{(-1)}(v) = \frac{1}{\sqrt{2\pi}\,\sigma} \operatorname{Exp}(-(v+1)^2 / 2\sigma^2) \qquad (6.8)$$

Figure (6.4) shows these functions. If a threshold level is taken half way between the signal level as indicated in the figure then four regions of error probability are shown, these are:

P_{e01} which is the probability of error in making the decision that a level +1 is present while the present level is 0.

$P_{e0(-1)}$ which is the probability of error in making the decision that a level -1 is present while the present level is 0.

P_{e10} which is the probability of error in making the decision that a level 0 is present while the present level is +1.

$P_{e(-1)0}$ which is the probability of error in making the decision that a level 0 is present while the present level is -1.

These probabilities are given by:

$$P_{e01} = \int_{\mu}^{\infty} \frac{1}{\sqrt{2\pi}\,\sigma} \mathrm{Exp}(-v^2 / 2\sigma^2)\, dv = \mathrm{Erfc}\left[\frac{1}{2.\sigma}\right] \qquad (6.9)$$

In a similar manner, we have

$$P_{e0(-1)} = \mathrm{Erfc}\left[\frac{1}{2.\sigma}\right] \qquad (6.10)$$

$$P_{e10} = \mathrm{Erfc}\left[\frac{1}{2.\sigma}\right] \qquad (6.11)$$

$$P_{e(-1)0} = \mathrm{Erfc}\left[\frac{1}{2.\sigma}\right] \qquad (6.12)$$

The total probability of error P is given by:

$$P = P_0 \,.\, P_{e01} + P_0 \,.\, P_{e0(-1)} + P_1 \,.\, P_{e10} + P_{-1} \,.\, P_{e(-1)0} \qquad (6.13)$$

Where P_0 , P_1 , and P_{-1} are the source digit probability of 0's, 1's and -1's respectively. These probabilities for the system under consideration are as mentioned before:

$$P_0 = 0.5 \;,\; P_1 = 0.25 \;,\; \text{and } P_{-1} = 0.25 \qquad (6.14)$$

By substituting (6.14) and (6.9) to (6.12) in (6.13) the total probability of error is given by:

$$P = 1.5 \;\; \mathrm{Erfc}\left[\frac{1}{2 \cdot \sigma}\right] \qquad (6.15)$$

Because $(S/N) = 1/2\sigma^2$ then $\sigma^2 = 1/2(S/N)$ and hence;

$$P = 1.5 \;\; \mathrm{Erfc}\left[\sqrt{\frac{S}{2N}}\right] \qquad (6.16)$$

6.4.2 MEASUREMENT OF ERROR PROBABILITY:

A statistical method has been used to measure the error probability of the system. In this method a stream of bits is transmitted and the number of errors in the received bits is counted. If the number of bits in the transmitted stream is n and the number of bits received in error is r, then, for a reasonable n and r, the probability of error is approximately equal to r/n.

To get an accurate measurement of error probability, n must be very large. Indeed in theory an infinite number of bits need to be observed to determine unequivocally the value of probability.

In practice the time for observation is limited so that the accuracy of P, the probability of error, is less than 100% accurate.

Statistical theory states that the liklihood, p, of finding r errors in a block of n bits is given by the expression:

$$P = \frac{n!}{r!\,(n-r)!} P_e^{\,r} (1 - P_e)^{n-r} \qquad (6.17)$$

where n! means n.(n-1).(n-2)...3.2.1

For probabilities less than 0.1, confidence limits to the probability value can be determined as follows.

If the result of an experiment is r errors, where r is greater than about 10, then P is estimated as r/n. providing P has a value of about 0.01 or less, then we can be:

1. 95% certain of being whithin approximately $2\sqrt{r}$ of the correct value of r.

2. 99% certain of being whithin 2.58 $\sqrt{r}$ of the correct r.

3. 99.9% certain of being whithin 3.28 $\sqrt{r}$ of the correct r.

For the results in this thesis a minimum number of transmitted bits n=80000 has been used to measure the probability of error. This was found to be the best compromise from the point view of both accuracy and economy of the computation time. Higher numbers, like n = 800000 have been tested and found to give similar results.

6.5 RESULTS OBTAINED FROM THE 8 CHANNELS SYSTEM:

This system uses the set of bridge functions Bri(i,1,3,t) as codes for its channels. The performance of this system has been studied for all possible numbers of active channels. i.e the performance of the system when 1,2,...8 channels are active.

The simulation results for this system are as follows:

1. For each number of active channels the probability of error in the aggregate signal and the probability of error in the channels for different signal-to-noise ratio are presented in tables (6.1).
2. Also for each case two graphs are plotted in figure (6.5). the first graph shows the probability of channel error against the signal to noise ratio. The second graph shows the channel error rate against the aggregate error rate.
3. Tables (6.2.1) - (6.2.8) for each case the table shows in the first column the standard deviation SD. In the 2nd through 10th column headed 1ER, 2ER,...., 8ER the number of blocks transmitted with 1,2,...8 errors and the corresponding number of

blocks received in 1,2,...8 errors are shown for each value of (SD.).

All the above results are based on the transmission of 10000 blocks for each value of (SD.).

6.6 DISCUSSION OF THE RESULTS FROM 8 CHANNELS SYSTEM:

The results show the adaptive property in the relationship between the channel error-correcting properties and the number of active channels. This will be clear from figure (6.5) and tables (6.2.1) - (6.2.8).

In figure (6.5.1) the probability of channel error against (S/N) ratio is plotted for 1, 3 and 7 active channels. From this figure it can be seen that the gain from using 1 channel only and turning off 7 channels is about 10 db, the gain from using 3 active channels and turning off 5 channels is about 3 db.

In the same figure the channel error-rate is plotted against the aggregate error-rate. This graph gives an idea about the comparison between the performance of the present system and the performance of a T.D.M. system. In a T.D.M. system it does not matter if the number of active channels is 1 or 8. The error rate will be the same if there is no error correcting scheme in use, while in the present system the performance gradually declines when the number of active channels is increased until the limit of the system is reached i.e. all the channels are active.

From the figure, when 1 active channel is used and for an aggregate error-rate of 0.1 the channel error-rate is about

0.0001, that is 3 order of magnitude of improvement. When the number of active channels is 3 the graph shows that when the aggregate error-rate is about 0.01the channel error-rate is about 0.0001, that is 2 order of magnitude of improvement. however when the number of active channels is 7, that is near the limit of the system, both the aggregate error-rate and the channel error-rate are almost the same value.

It can be seen from the figure the relationship between the aggregate error-rate and the channel error-rate is linear.

In figure (6.5.2) the probability of channel error-rate against (S/N) ratio is plotted for 2, 4 and 8 channels in use.

The same discussion as for the previous figure (6.5.1) applies here where the performance of the system gradually degrades when more channels are added (turned on). The gain from turning off 4 channels and using only 4 channels is about 3 db and it is about 10 db when turning off 6 channels.

The channel error-rate against the aggregate error-rate are also plotted and shown in the figure for 2, 4 and 8 active channels. Again it is linear. The improvement in performance for 2 active channels is almost the same as for 1 active channel, while the performance is slightly different when using 3 and 4 active channels and this applies also for 7 and 8 active channels.

In figure (6.5.3) the graphs of probability of channel error-rate against the (S/N) ratio and the channel error-rate against the aggregate error-rate are plotted for 1, 2 and 3 active channels. From this figure it is obvious that there is not much difference between using 1 or 2 active channels i.e. the performance of the system does not improve significantly when turning off 1 channel if the previous state is 2 channels active. This can be explained from the point of view of the distance between the codes, where the distance between any two codes of the used set (one with an even order and the other with an odd order) is 8, while the minimum distance between any one of three codes is 4, that is, half the distance of the previous case. Hence if the previous state of the system was 3 active channels and one of them is turned off then a greater improvement is expected. This is clear from the graph where the gain from turning one channel off in this case is about 7 db.

The 2nd graph in figure (6.5-3) shows how the channel error-rate is affected when 1, 2 and 3 active channels are used. From this graph it can be seen that for a given aggregate error-rate 0.1 the channel error-rate is about 0.0001 when 2 channels are active. To get the same channel error-rate when 3 channels are active the aggregate error-rate must be 0.01. Hence the system performance drops dramatically when adding the 3rd channel.

Figure (6.5-4) shows the graphs for using 4, 5 and 6 active channels. From this figure it can be seen that there is no great improvement in the performance of the system when switching from 6 to 5 or even to 4 active channels.

Now we shall discuss the results obtained from the system which are tabulated in tables (6.2). There are 8 tables these are for 1, 2,...8 active channels.

Tables (6.2.1) and (6.2.2) are for 1 and 2 active channels respectively. These two tables show that the system exibits at least 3 error-correcting modes. However, the system can handle most 4th, 5th and 6th order errors and correct them. Some examples are:

In table (6.2.1) when SD. = 0.5 there are 768 blocks with 4 errors, out of these 767 are corrected. Also there are 25 blocks with 6 errors, 23 of these are corrected.

In table (6.2.2) when SD. = .6 there are 87 blocks with 5 errors, 80 of them has been corrected. Also there are 15 blocks with 6 errors, 12 of these are corrected.

Table (6.2.3) shows the results from using 3 active channels, from these results it can be seen that the system exibits a single error correcting mode i.e. every single error which occurs is corrected. Not only that, it can be seen from the table that about 90% of double errors are corrected, and about 70% of triples are also corrected. Some higher order errors are also corrected.

Table (6.2.4) shows the results from using 4 active channels. These results are slightly different from those in table (6.2.3). A single error correcting mode exists as before. For higher order of errors the percentage of error correcting drops as the number of errors increased. For double errors about 85% of these errors are corrected. For triples about 50% are corrected. Some of the higher orders of error are also corrected.

Table (6.2.5) shows the results from using 5 active channels. These are almost the same as the previous case (4 active channels) with only a slight difference. The single error correcting mode also exist~~e~~s here. About 80% of double errors are corrected and 50% of triple errors are also corrected. Some of the higher orders of errors, like 4 and 5 errors are also corrected.

Table (6.2.6) shows the results of using 6 active channels. These results are nearly the same as those belonging to 5 active channels. Again the single error correcting mode exists here.

Table (6.2.7) shows the results from using 7 active channels. These are different from the previous tables where the system in this case can not correct all the single errors i.e. the single error correcting mode also does not exist here. However, about 60% of single error blocks will be corrected, 45% of double errors will be corrected and about 20% of triple errors will also be corrected.

Table (6.2.8) shows the results from using 8 active channels. Here the system is fully loaded (all the channels are in use). Nevertheless the system still can correct 50% of single error and a few percentage of double errors.

As a conclusion from the results of the 8 channels system it can be said that there are 3 distinct criteria of performance. These are:

1. When using 1 or 2 active channels, triple error correcting modes exist, together with some percentage of higher orders of error correcting modes.

2. When using 3, 4, 5 or 6 active channels, single error correcting modes exist. Some percentage of higher orders of error correcting modes also exist.

3. When using 7 or 8 active channels, there is no absolute error correcting mode in existance but some of the errors will nevertheless be corrected.

From these 3 criteria it can be seen that 1, 3 and 7 active channels are the critical numbers of active channels where the performance of the system is degraded from the triple error correcting mode to the single error correcting mode and then to no error correction. Between these modes are various hybrid situations with different error correction properties, depending on the actual data. There is an increase in the error correcting properties as more channels are turned off.

6.7 RESULTS OBTAINED FROM 16 CHANNELS SYSTEM:

This system uses the set of bridge functions Bri(i,1,4,t), Bri(i,2,4,t) and Bri(i,3,4,t) as codes for its channels.

The performance of this system has been studied for all possible number of active channels. i.e. the performance of the system when 1, 2, 3,....,16 channels are active.

The simulation results for this system are grouped into 3 groups these are:

1. Tables (6.3) these tables give the probability of error in the aggregate signal and the probability of error in the channels for different signal-to-noise ratios.

There are 16 tables these are for 1, 2, 3,....,16 active channels.

2. Figures (6.6-1) to (6.6-6). In each figure two graphs are plotted the first graph is a plot of the probability of channel error against the signal-to-noise ratio. The second graph is a plot of the channel error-rate against the aggregate error-rate.

3. Tables (6.4.1) to (6.4.16) These tables show for each state of active channels the following information:

a. Different values of standard deviation ST are shown in column 1.

b. For each value of ST. there are two values in the column headed 1ER, the upper value is the number of blocks transmitted with 1 error and the lower value is the number of these blocks which have been corrected after processing. The remaining columns headed 2ER, 3ER,... ,9ER are the same as 1ER but for 2, 3,...,9 errors. These are based on the transmission of 10000 blocks.

6.8 DISCUSSION OF THE RESULTS FROM 16 CHANNELS SYSTEM:

As in the 8 channels system the results from the 16 channels system show mainly the adaptive property in the relationship between the channel error-correcting properties and the number of active channels. This will be clear through the discussion of figures (6.6) and tables (6.4).

In each figure of figures (6.6-1) to (6.6-6) two graphs are plotted:

1. The probability of channel error against (S/N) ratio.
2. Channel error-rate against aggregate error-rate.

In figure (6.6.1) these graphs are plotted for 1, 5, and 9 active channels. From the first graph of this figure it can be seen that when 9 channels are active then the gain from turning off 4 channels is about 5db. A further turning off more 4 channels will result in a gain of another 7 db. Thus the gain from turning off 8 channels is about 12 db.

From the second graph of this figure it can be seen that when switching from 9 active channels to 5 active channels for an aggregate error-rate = 0.1, then the channel error rate will be changed from 0.02 to 0.0004. Also switching from 5 active channels to 1 active channel for an aggregate error rate =0.3 will result in a change in the channel error rate from 0.03 to 0.00002.

In figure (6.6.2) the graphs are plotted for 2, 6 and 10 active channels. This figure is almost the same as figure (6.6.1) where the similarity is between the curves for 1 and 2 active channels, the curves for 5 and 6 active channels and the curves for 9 and 10 active channels.

In figure (6.6.3) the graphs are plotted for 3, 7 and 11 active channels. from the first graph of this figure it can be seen that switching from 11 active channels to 7 active channels will result in a gain of 2 db for probability of error = 0.0001.

Also, switching from 7 active channels to 3 active channels will results in a gain of 5 db for the same probability of error.

From the second graph it can be seen that when switching from 11 active channels to 7 active channels for an aggregate error-rate = 0.1, the the channel error-rate will be changed from 0.02 to 0.002. Also, when switching from 7 active channels to 3 active channels and for an aggregate error rate = 0.1 then the channel error rate will be changed from 0.002 to 0.00002.

In figure (6.6.4) the graphs are plotted for 4, 8 and 12 active channels.

From the first graph of this figure it can be seen that when switching from 12 active channels to 8 active channels and for a probability of error = 0.001 the the gain will be 2.5 db. Also, when switching from 8 active channels to 4 active channels and for the same probability of error the gain will be 2.5 db.

From the second graph of this figure it can be seen that when switching from 12 active channels to 8 active channels and for an aggregate error rate = 0.1 then the channel error-rate will be changed from 0.03 to 0.005. However when switching from 8 to 4 active channels and for an aggregate error-rate = 0.1 then the channel error-rate will be changed from 0.005 to 0.0002.

In figure (6.6.5) the graphs are plotted for 10, 11 and 13 active channels. From this figure it can be seen that there is no noticeable gain when switching from 13 active channels to 11 or 10 active channels. Also there is no improvement in channel error rate when switching from 13 active channels to 11 or 10 active channels.

In figure (6.6.6) the graphs are plotted for 13, 14 lnd 15 active channels. From the first graph of this figure it can be seen that when switching from 15 to 14 active channels and for a probability of error = 0.0001 then the gain will be about 2 db. However switching from 14 to 13 active channels gives no noticeable gain. From the second graph it can be seen that when switching from 15 to 14 active channels and for an aggregate error rate = 0.07 then the channel error rate will be changed from 0.001 to 0.0001 . However switching from 14 to 13 active channels gives no noticeable improvement in the channel error rate.

Now let us discuss the results which are tabulated in tables (6.4.1) to (6.4.16). These are for 1, 2,...,16 active channels.
Tables (6.4.1) and (6.4.2) are for 1 and 2 active channels. Both tables show that the system exibits 8 error-correction mode when SD (the standard deviation) < 0.5 . However the system can correct some higher orders of errors. This is not shown in the tables since it would be necessary to provide further table to show this. For reasons of economy tables for number of errors greater than 9 have not been included.
Tables (6.4.3) and (6.4.4) are for 3 and 4 active channels. Both tables show a 4 error correction mode. However, the system can also correct most higher order (up to 8) errors.
Tables (6.4.5) and (6.4.6) are for 5 and 6 active channels. Both tables show a triple error correction mode. However about 95% of 4 errors can also be corrected, and about the same percentage

for 5 errors. It is also possible to correct certain higher orders of errors.

Tables (6.4.7) through (6.4.14) are for 7 to 14 active channels. These tables show a single error correction mode. However, higher orders of errors can also be corrected. The tables show the degradation of the percentage of errors which can be corrected when the number of active channels is increased from 7 to 14 active channels.

Tables (6.4.15) and (6.4.16) are for 15 and 16 active channels. In these tables there is no evident absolute error correction mode. However, some errors from certain orders can be corrected, for example, most of the single errors and more than 50% of the double errors can be corrected.

As a conclusion, from the results of the 16 channels system it can be seen that there are 5 distinct criteria of performance these are:

1. When the number of active channels is 1 or 2, the system exibits an 8 error correction mode.
2. When the number of active channels is 3 or 4, the system exibits a 4-error correction mode.
3. When the number of active channels is 5 or 6, the system exibits a triple error correction mode.
4. When the number of active channels is 7 through 14, the system exibits a single error correction mode.

5. When the number of active channels is 15 or 16, no absolute error-correction mode exists but some of the errors can be corrected.

From these 5 criteria it can be seen that 1, 3, 5, 7 and 15 active channels are the critical number of active channels where the performance of the system is degraded from the 8 error correction mode to the 4 error correction mode,to the 3 error correction mode,to the single error correction mode to the no absolute error correction mode. Between these modes there are various hybrid situations with different error correction properties, depending on the actual data.

6.9 RESULTS FROM THE 32 CHANNELS SYSTEM:

This system uses the set of bridge functions Bri(i,1,5,t), Bri(i,2,5,t) and Bri(i,3,5,t) as codes for its channels.

The performance of this system has been studied for all possible numbers of active channels. i.e. the performance of the system when 1, 2, 3,....,32 channels are in use.

As for the 8 and 16 channels systems, the simulation results for the 32 channels system are grouped into 3 groups. These are:

1. Tables (6.5). These tables gives the probability of error in the aggregate signal and the probability of error in the channels for different signal to noise ratios. There are 32 tables of these, one for each state of the active channels.

2. Figures (6.7-1) to (6.7-15) In each figure two graphs are plotted. The first graph is a plot of the probability of channel error against the signal-to-noise ratio (S/N). The second graph

is a plot of the channel error-rate against the aggregate error-rate.

3. Tables (6.6.1) to (6.6.32). There are 32 tables, one for each number of active channels. Each table has the following information:

a. Different values of standard deviation (SD.) are shown in column 1.

b. In column 2, headed 1ER, there are two numbers for each value of (SD.). The upper value is the number of blocks transmitted with a single error and the lower value is the number of blocks received with a single error corrected.

c. Columns 3 through 10, headed 2ER, 3ER,....,9ER, are the same as column 2 but for 2, 3,....,9 errors.

All the above numbers in b and c are based on transmission of 10000 blocks.

6.10 DISCUSSION OF THE RESULTS FROM 32 CHANNELS SYSTEM:

As in the 8 and 16 channels systems the results from 32 channels show an adaptive property in the relationship between the channel error-correcting capabilties and the number of active channels.

This will be clear from figures (6.7-1) to (6.7-15) and tables (6.6.1) to (6.6.32).

As mentioned before, in each figure of (6.7) two graphs are plotted these are:

1. The probability of channel error against (S/N) raio.
2. Channel error-rate against aggregate error-rate.

Because there are 15 figure in figure (6.7) and the discussion of these figures are simillar to those for 8 and 16 channels system, it would be tedious to discuss each figure in the same details as before. We should therefore comment on each figure only briefly since after the explanations assosiated with figures (6.5-1) to (6.5.4) and figures (6.6-1) to (6.6-6), figures (6.7-1) to (6.7-15) becomes self-explanatory.

In figure (6.7-1) the graphs are plotted for 1, 11 and 21 active channels and the graphs in figure (6.7-2) are plotted for 2, 12 and 22 active channels. From these two figures it can be seen that the graphs for 1 and 2 active channels are almost the same. This also also applies for 11 and 12 active channels, but there is a slight difference between 21 and 22 active channels.

In figures (6.7-3) the graphs are plotted for 3, 13 and 23 active channels and the graphs in figure (6.7-4) are plotted for 4, 14 and 24 active channels. From these two figures it can be seen that the graphs for 3 and 4 active channels are sightly different but the graphs for 13 and 14 active channels are almost the same. The similarity between 23 and 24 active channels is clear.

In figure (6.7-5) the graphs are plotted for 5, 15 and 25 active channels and in figure (6.7-6) the graphs are plotted for 6, 16 and 26 active channels. From these two figures it can be seen that there is a difference between the graphs for 5 and 6 active channels while the similarity between those graphs for 15 and 16 active channels and those for 25 and 26 active channels are clear.

In figure (6.7-7) the graphs are plotted for 7, 17 and 27 active channels. From this figure it is clear that there is not much difference between 17 and 27 active channels but there is a significant difference between 7 and 17 active channels.

In figure (6.7-8) the graphs are plotted for 1, 3 and 5 active channels and in figure (6.7-9) the graphs are plotted for 2, 4 and 6 active channels. These two figures show the improvement acheived when the number of active channels goes down from 5 to 3 and then to 1 or from 6 to 4 and then to 2.

In figure (6.7-10) the graphs are plotted for 7, 15 and 27 active channels and figure (6.7-11) shows the graphs for 8, 16 and 28 active channels. From these two figures it can be seen that there is no differnce between 7 and 8 active channels, 15 and 16 are almost the same and 27 and 28 are also simillar.

In figure (6.7-12) the graphs are plotted for 9, 17 and 29 active channels and figure (6.7-13) show the graphs for 10, 18 and 30 active channels. From these two figures the difference between 9 and 10 active channels is obvious while there is no significant difference between 17 and 18 active channels. However, there is a small difference between 29 and 30 active channels.

From figures (6.7-14) and (6.7-15) the similarity between 11 and 12 active channels, 19 and 20 active channels and between 31 and 32 active channels is clear.

Now let us discuss the results in tables (6.6.1) to (6.6.32).

As tables (6.2.1) to (6.2.8) and (6.4.1) to (6.4.16), tables (6.6.1) to (6.6.32) give information about error correction for different number of active channels. There are 32 tables for 1, 2,...,32 active channels. Each table gives the number of blocks with 1 error, 2 errors, ..., 9 errors and the corresponding number of blocks which have been corrected. The first 8 tables, those for 1-8 active channels have been expanded to give information about number of blocks with 10, 11,...,18 errors and the corresponding number of blocks which have been corrected. This is done by providing tables (6.6.1.a)-(6.6.8.a).

The tables will be discussed very briefly :

From tables (6.6.1) and (6.6.1a) it can be seen that for 1 active channel, an 18 error correction mode exists.

Tables (6.6.2) and (6.6.2a) show that the 18 error correction mode also exist for 2 active channels.

From tables (6.6.3) and (6.6.3a) it can be seen that for 3 active channels a 12 error correction mode exists.

From tables (6.6.4) and (6.6.4a) it can be seen that for 4 active channels an 11 error correcting mode exists.

For 5 active channels it can be seen from Table(6.6.5) that there is an 8 error correcting mode.

The 8 error correcting mode also applies for 6 active channels, as can be seen from figure (6.6.6).

For 7 active channels it can be seen from table (6.6.7) that there is a 5 error correction mode.

For 8 active channels it can be seen from table(6.6.8) that there is a 4 error correction mode.

For 9 active channels and up to 14 active channels it can be seen from tables (6.6.9) through (6.6.14) that there is a triple error correction mode for these number of active channels.

For 15 active channels and up to 28 active channels it can be seen that there is a single error correction mode exists.

From 29 active channels and up to the limit of the system, i.e 32 active channels, it can be seen from tables (6.6.29) through (6.6.32) that there is no absolute error correction mode. However, most of the single error will be corrected and some of the higher orders of errors will also be corrected.

In all the above mentioned cases not only the error correction mode which belongs to the case existes but most of the higer order of errors will also be corrected, especialy those which are near to the corresponding error correction mode.

As a conclusion, from the results of the 32 channels system it can be seen that there are 7 distinct criteria of performance. These are:

1. When using 1 or 2 active channels, an 18-error correction mode is exists.
2. When using 3 or 4 active channels, a 12-error correction mode and 11-error correction mode exists.
3. When using 5 or 6 active channels, an 8-error correction mode exists.
4. When using 7 or 8 active channels, a 5-error correction mode and a 4-error correction mode exists.
5. When using 9-14 active channels, a triple error correction mode exists.

6. When using 15-28 active channels, a single error correction mode exists.

7. When using 29-32 active channels, there is no absolute error correction mode but some errors will nevertheless be corrected.

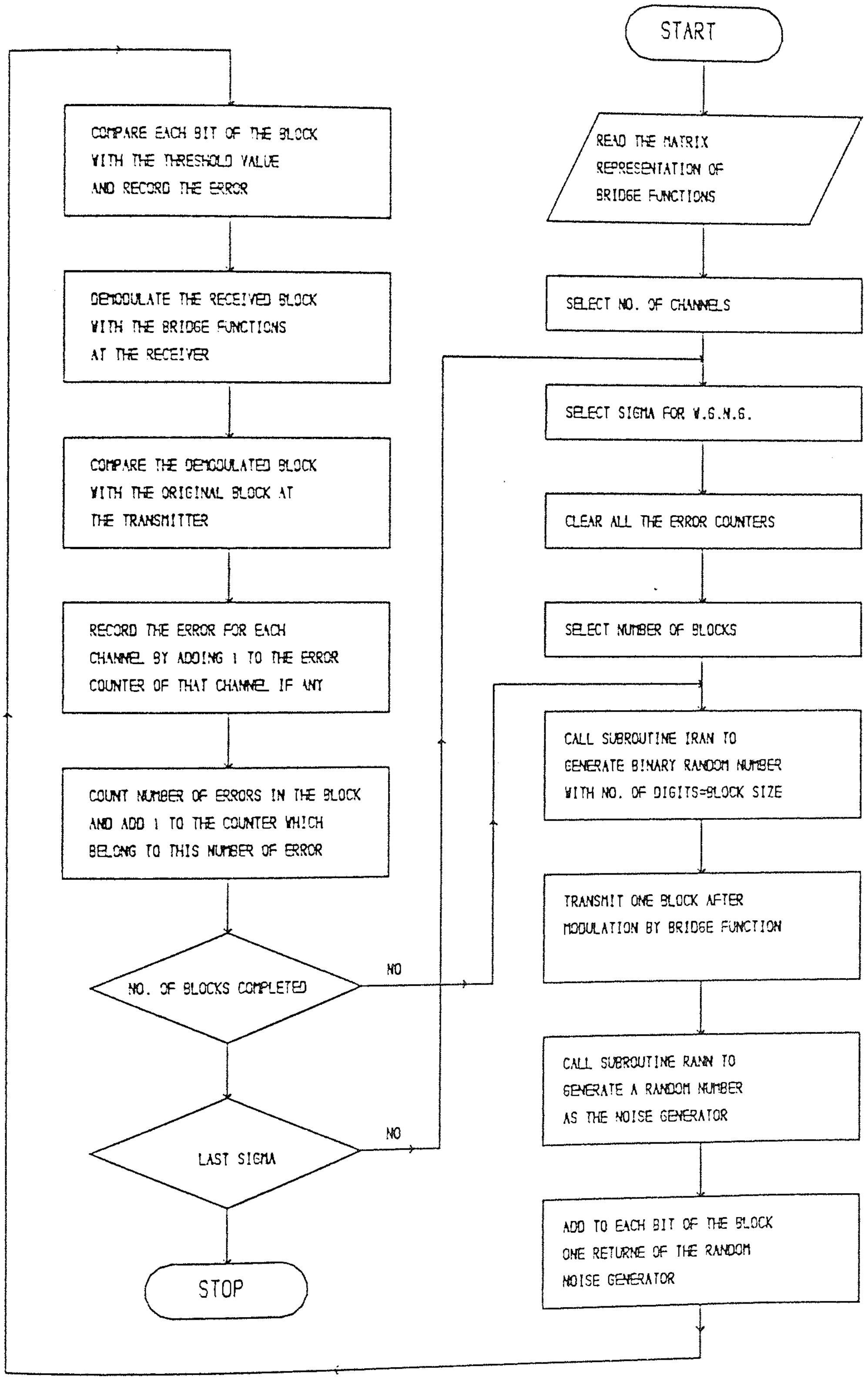


FIG.(6.1): GENERAL FLOW CHART OF THE PROGRAM.

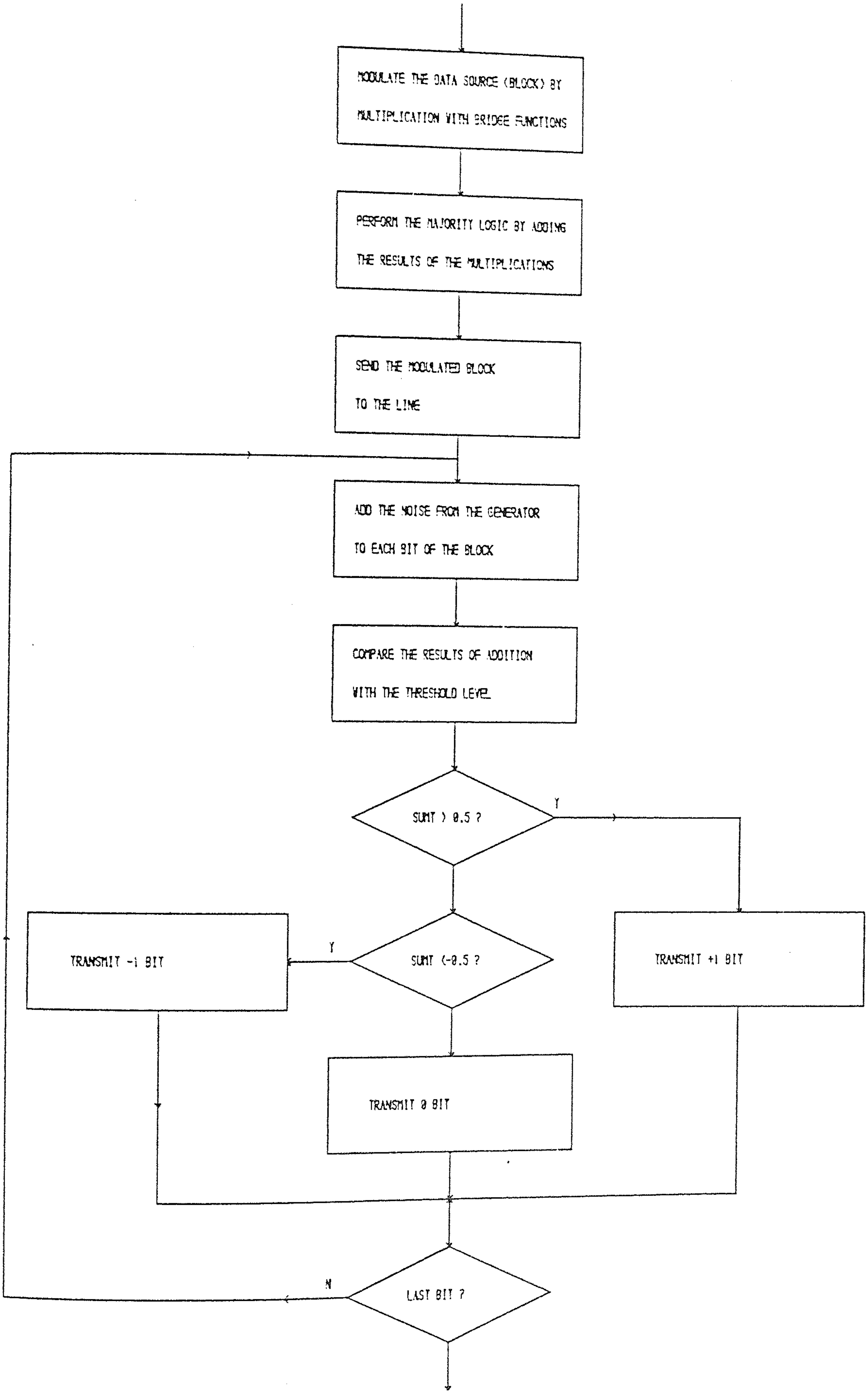


FIG.(6.2) FLOW CHART TO ILLUSTRATE THE TRANSMIT BLOCK.

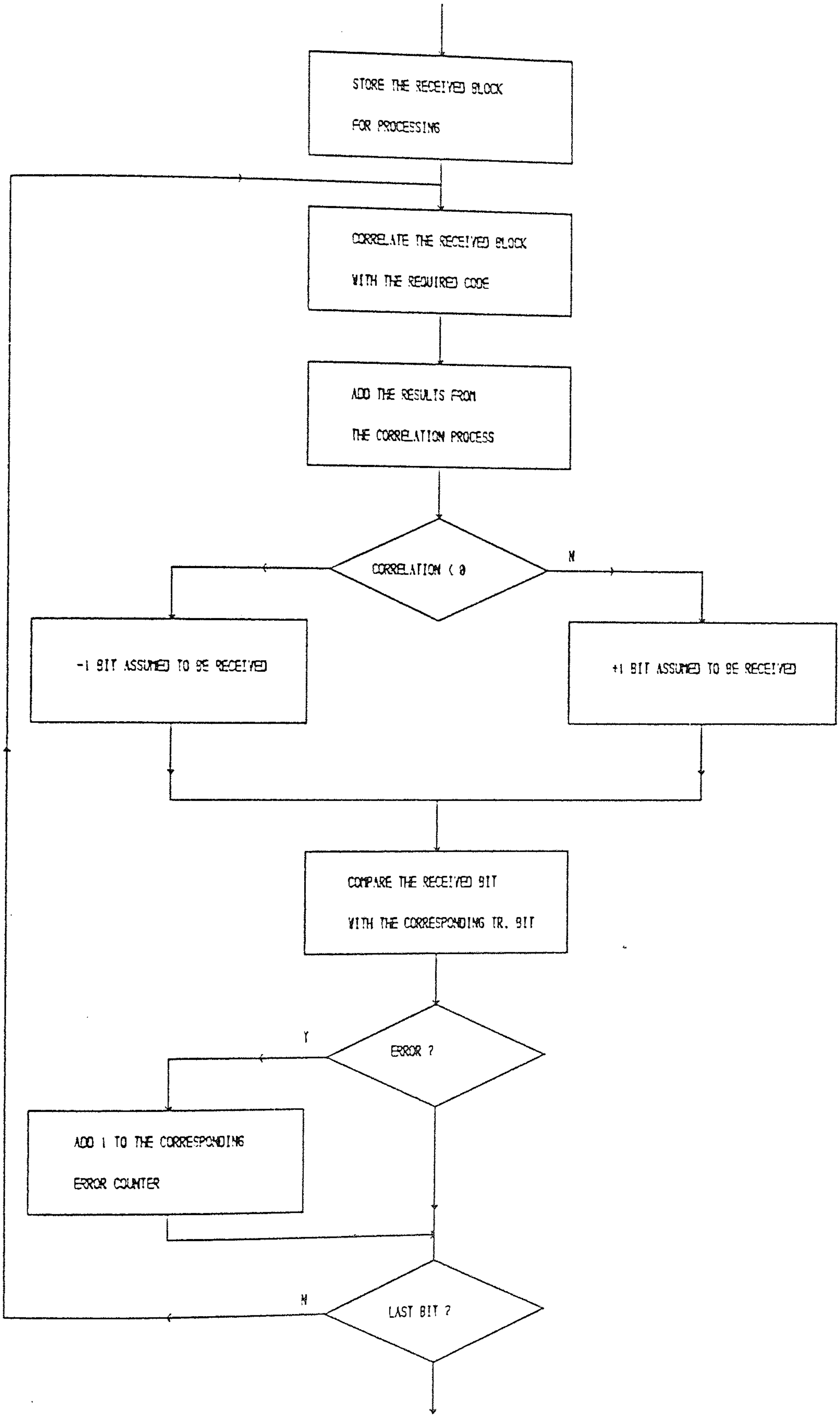


FIG.(6.3): FLOW CHART TO ILLUSTRATE RECEIVEBLOCK

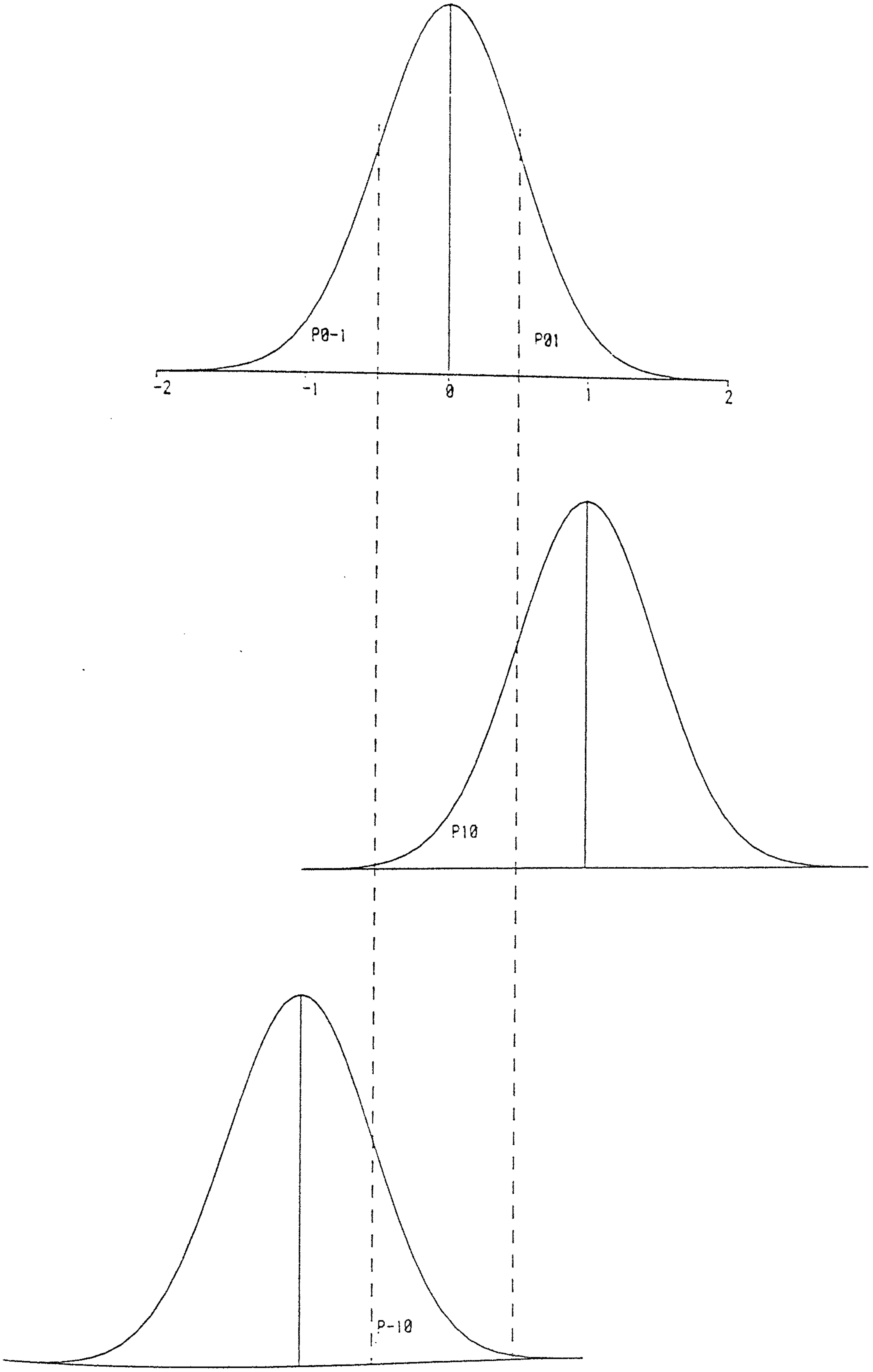


FIG.(6.4):Probability density functions for ternary signal

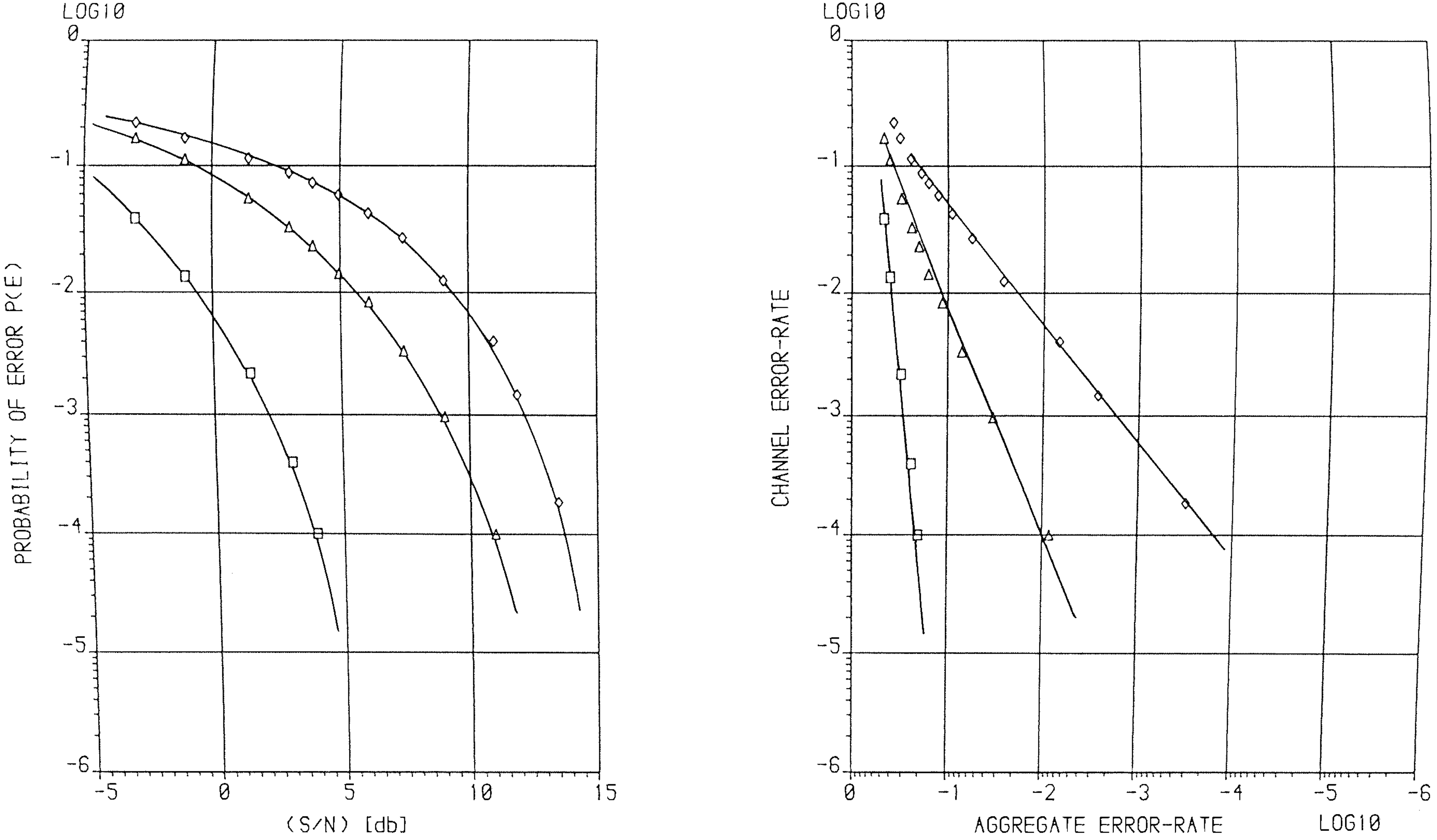


FIG.(6.5 -1):Probability of error for Gaus. noise using Bridge functions of 8 bits as codes

□□□ 1 CHANNEL　△△△ 3 CHANNELS　◇◇◇ 7 CHANNELS

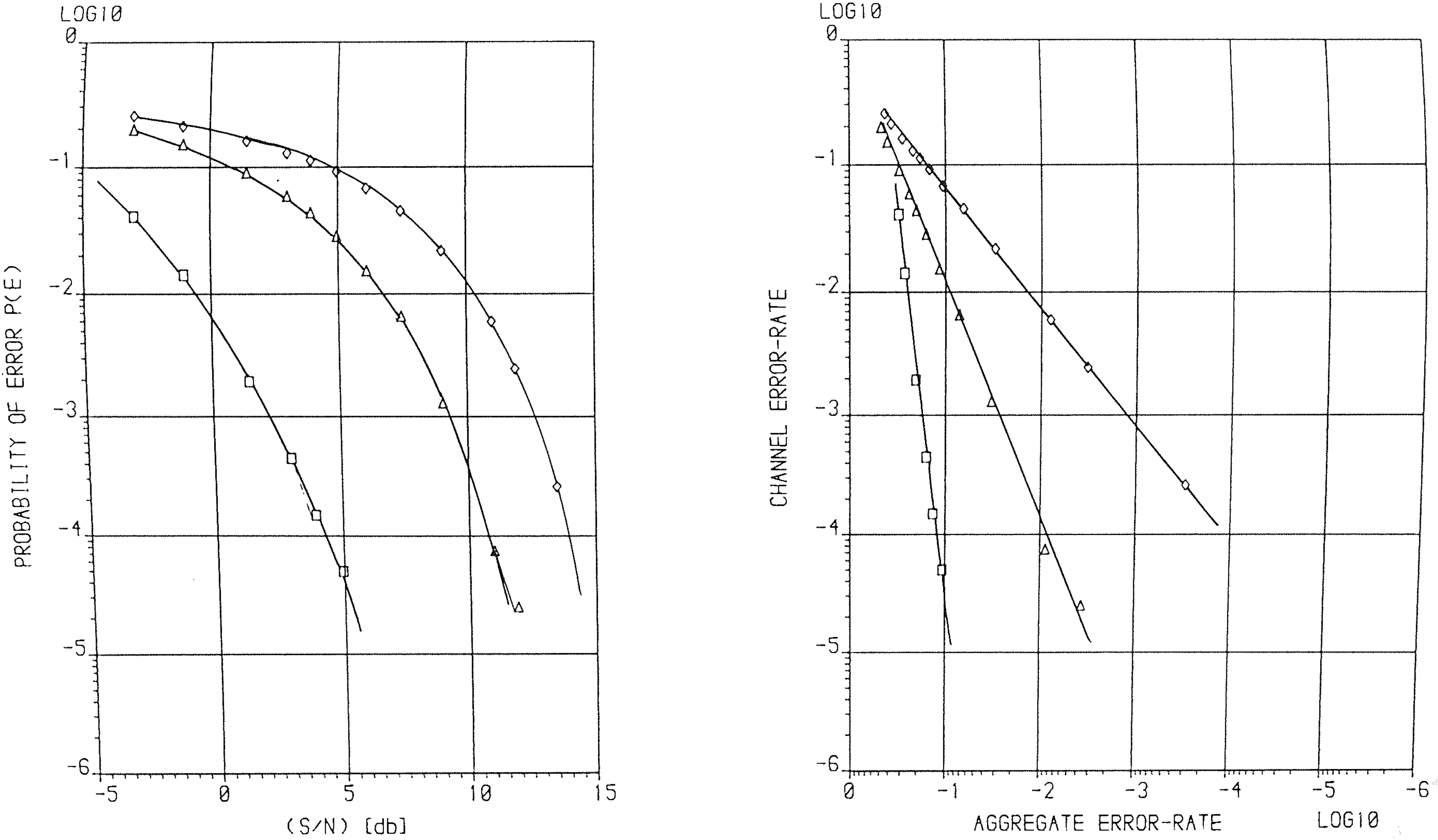


FIG.(6.5 -2):Probability of error for Gaus. noise using Bridge functions of 8 bits as codes

□□□ 2 CHANNELS △△△ 4 CHANNELS ◇◇◇ 8 CHANNELS

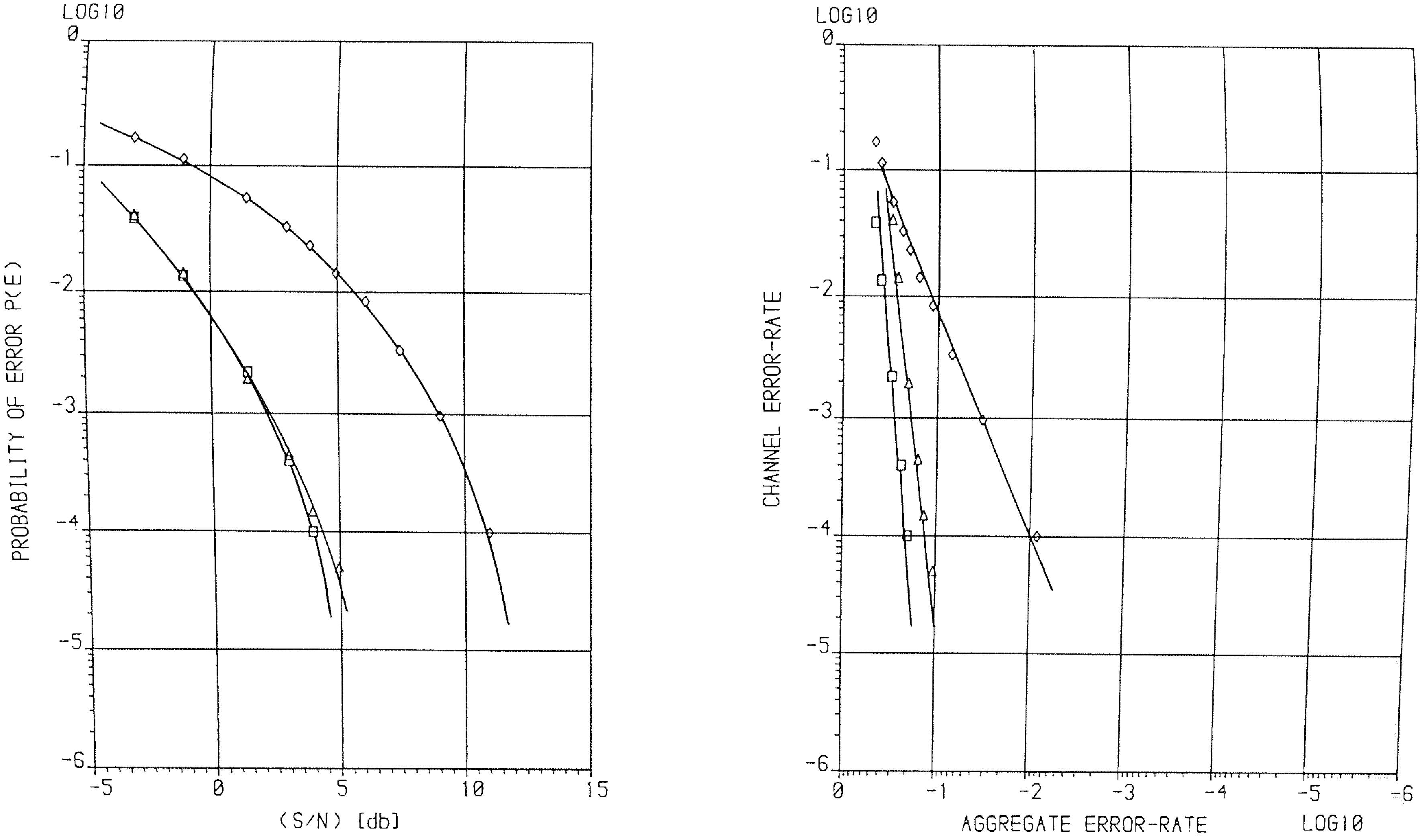


FIG.(6.5 -3):Probability of error for Gaus. noise using Bridge functions of 8 bits as codes

□□□ 1 CHANNEL △△△ 2 CHANNELS ◇◇◇ 3 CHANNELS

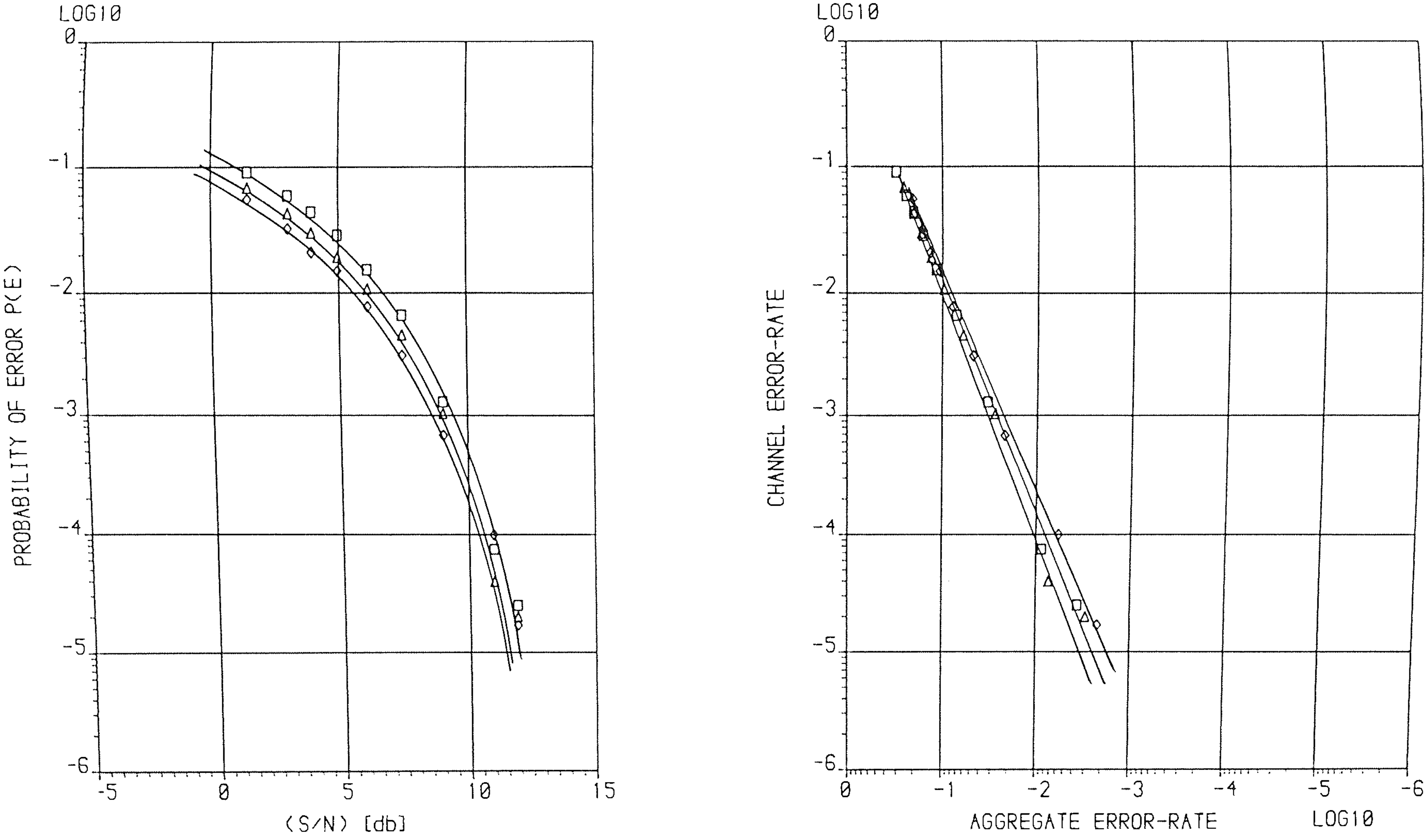


FIG.(6.5-4):Probability of error for Gaus. noise using Bridge functions of 8 bits as codes

□□□ 4 CHANNELS   △△△ 5 CHANNELS   ◇◇◇ 6 CHANNELS

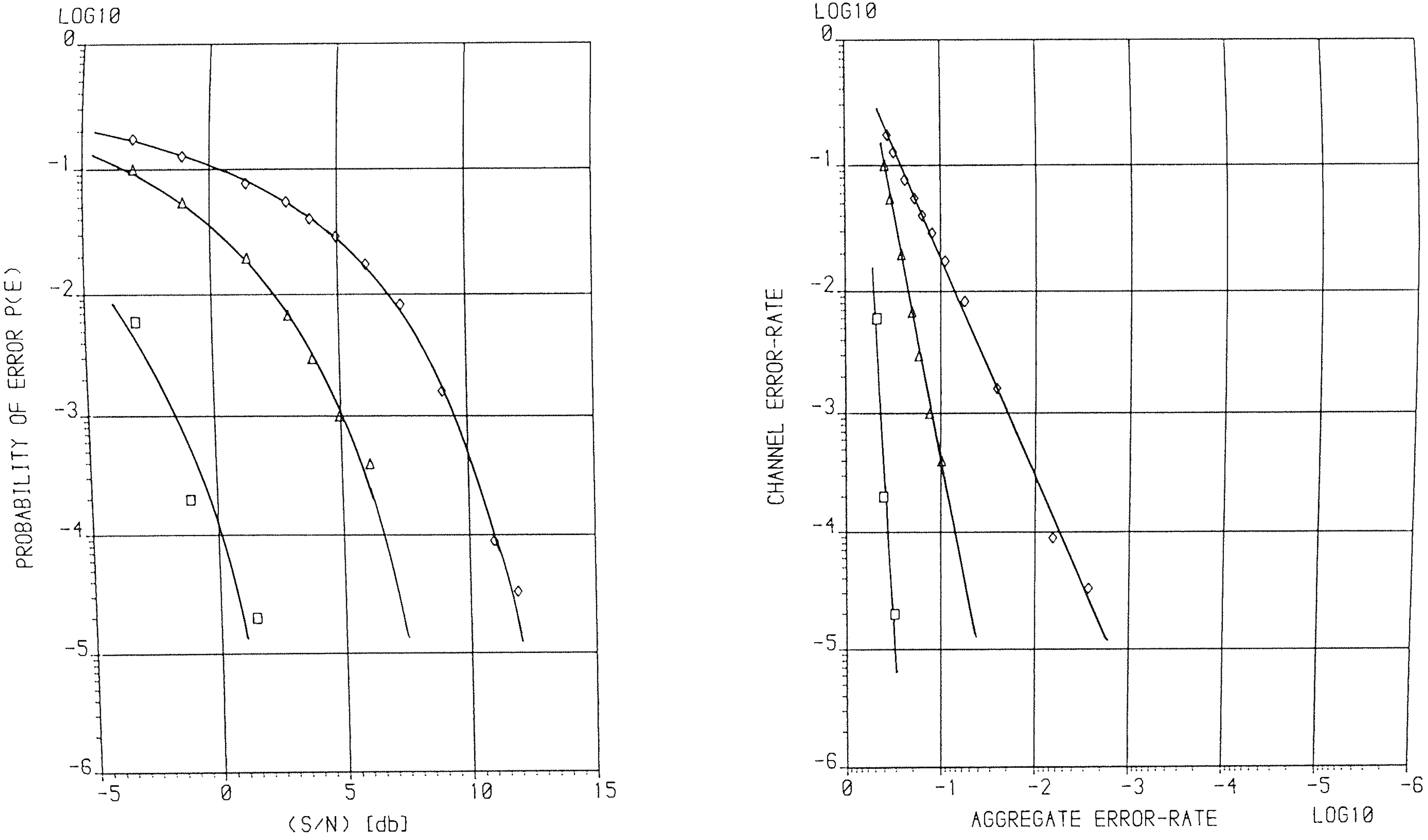


FIG.(6.6-1):Probability of error for Gaus. noise using Bridge functions of 16 bits as codes

□□□ 1 CHANNEL △△△ 5 CHANNELS ◇◇◇ 9 CHANNELS

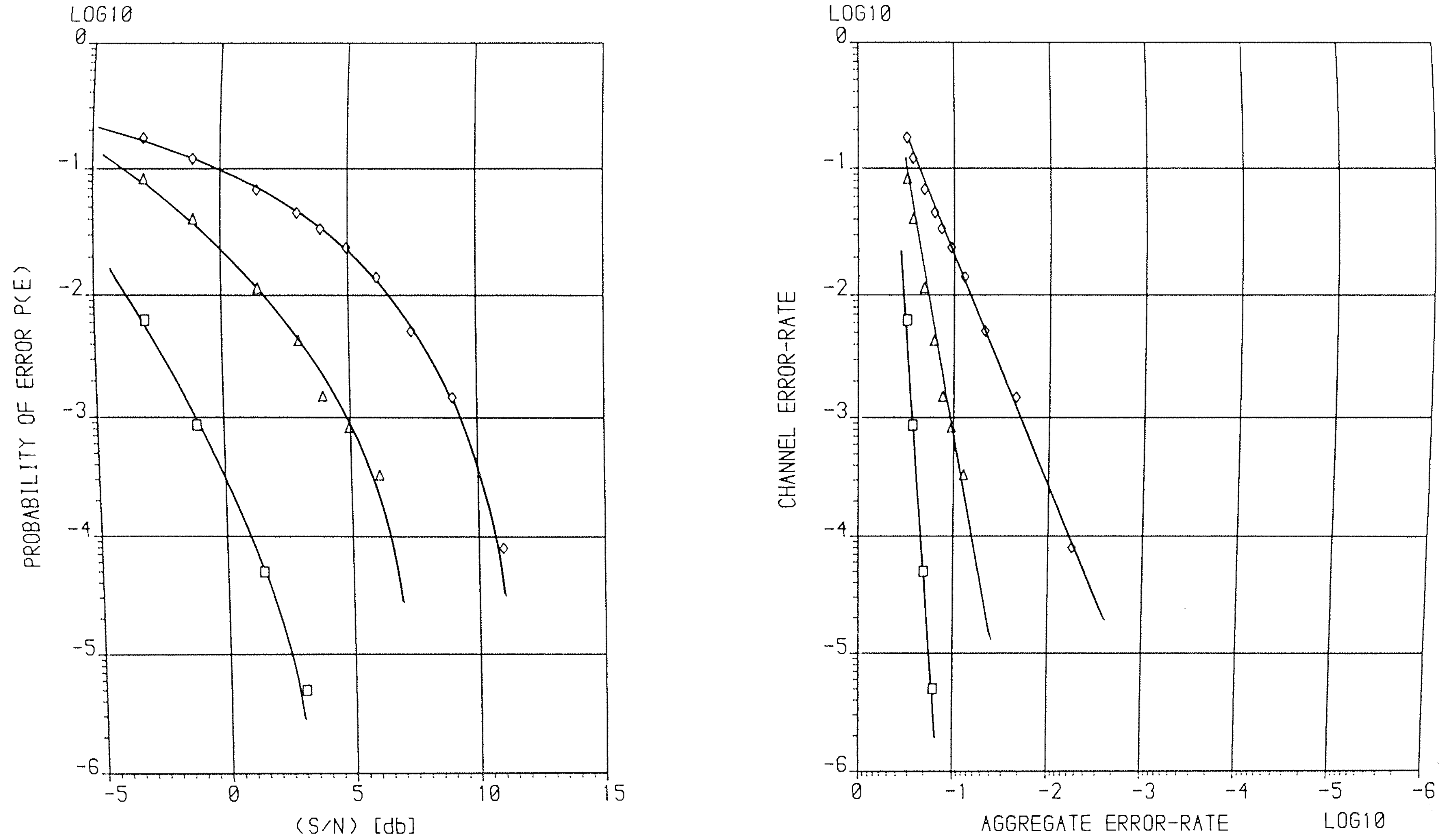


FIG.(6.6-2):Probability of error for Gaus. noise using Bridge functions of 16 bits as codes

□□□ 2 CHANNELS　　△△△ 6 CHANNELS　　◇◇ 90 CHANNELS

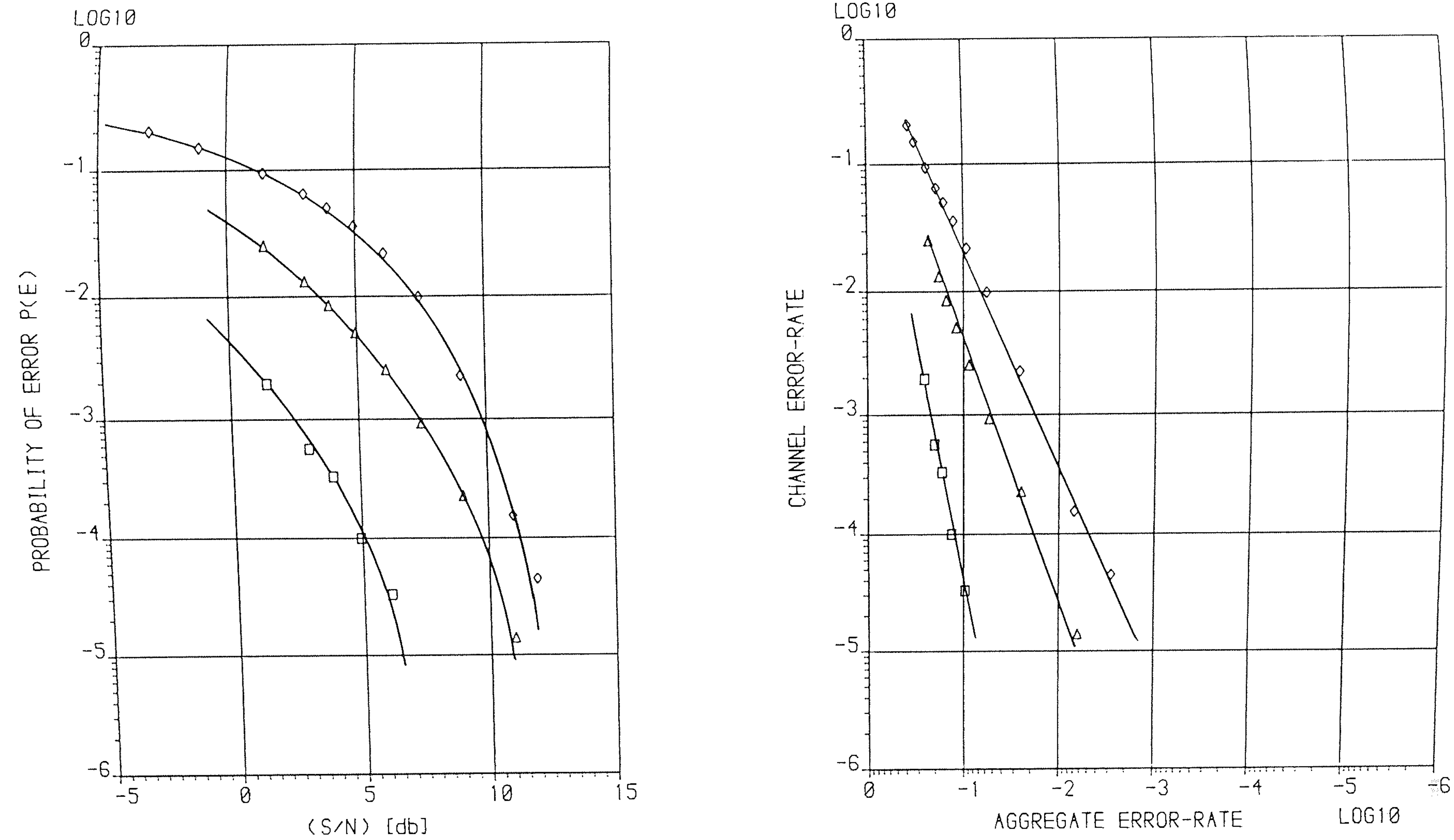


FIG.(6.6-3):Probability of error for Gaus. noise using Bridge functions of 16 bits as codes

□ □ □ 3 CHANNELS △ △ △ 7 CHANNELS ◇ ◇ 91 CHANNELS

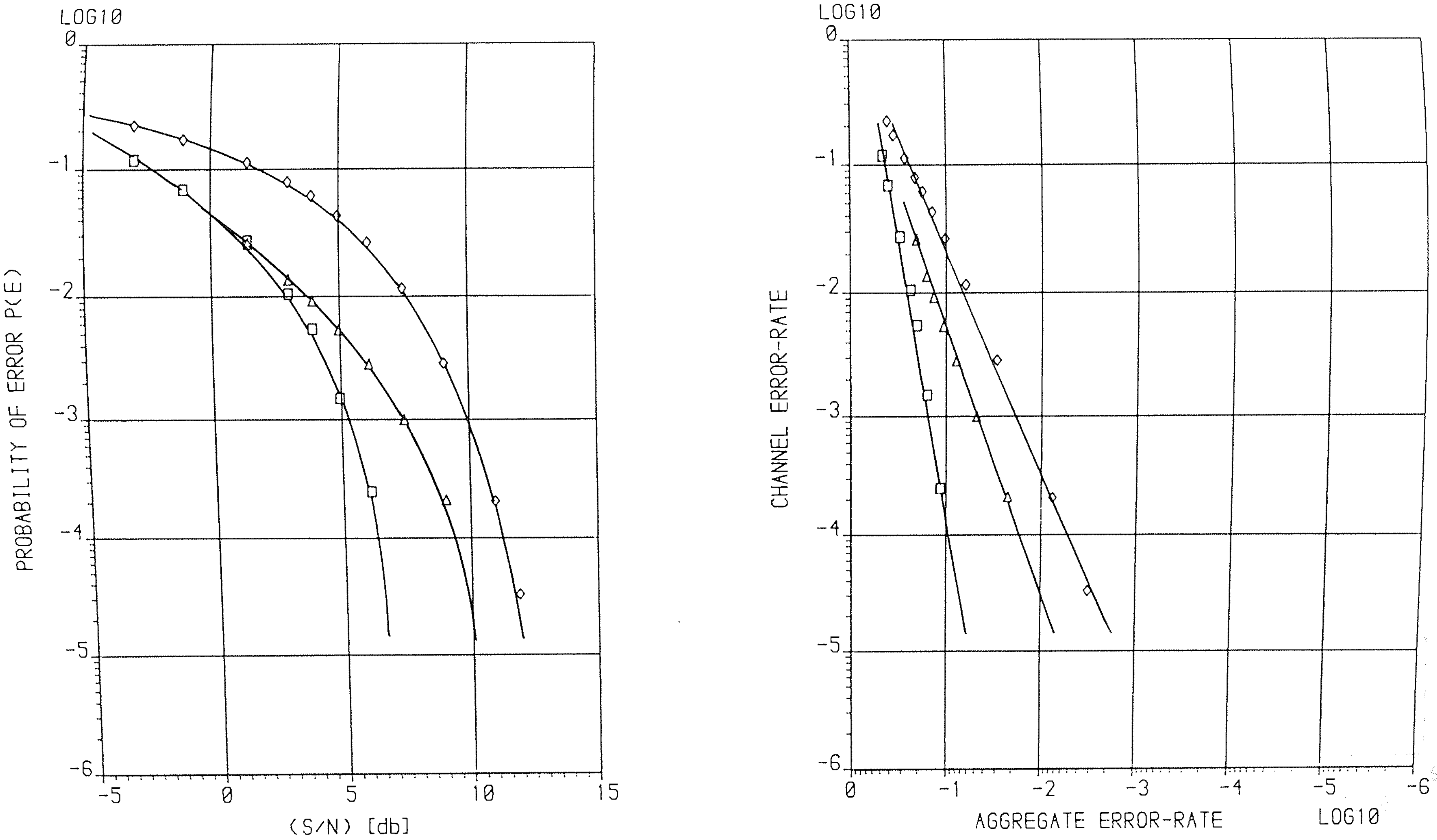


FIG.(6.6-4):Probability of error for Gaus. noise using Bridge functions of 16 bits as codes

□ □ □ 4 CHANNELS ▵ ▵ ▵ 8 CHANNELS ◇ ◇ ◇12 CHANNELS

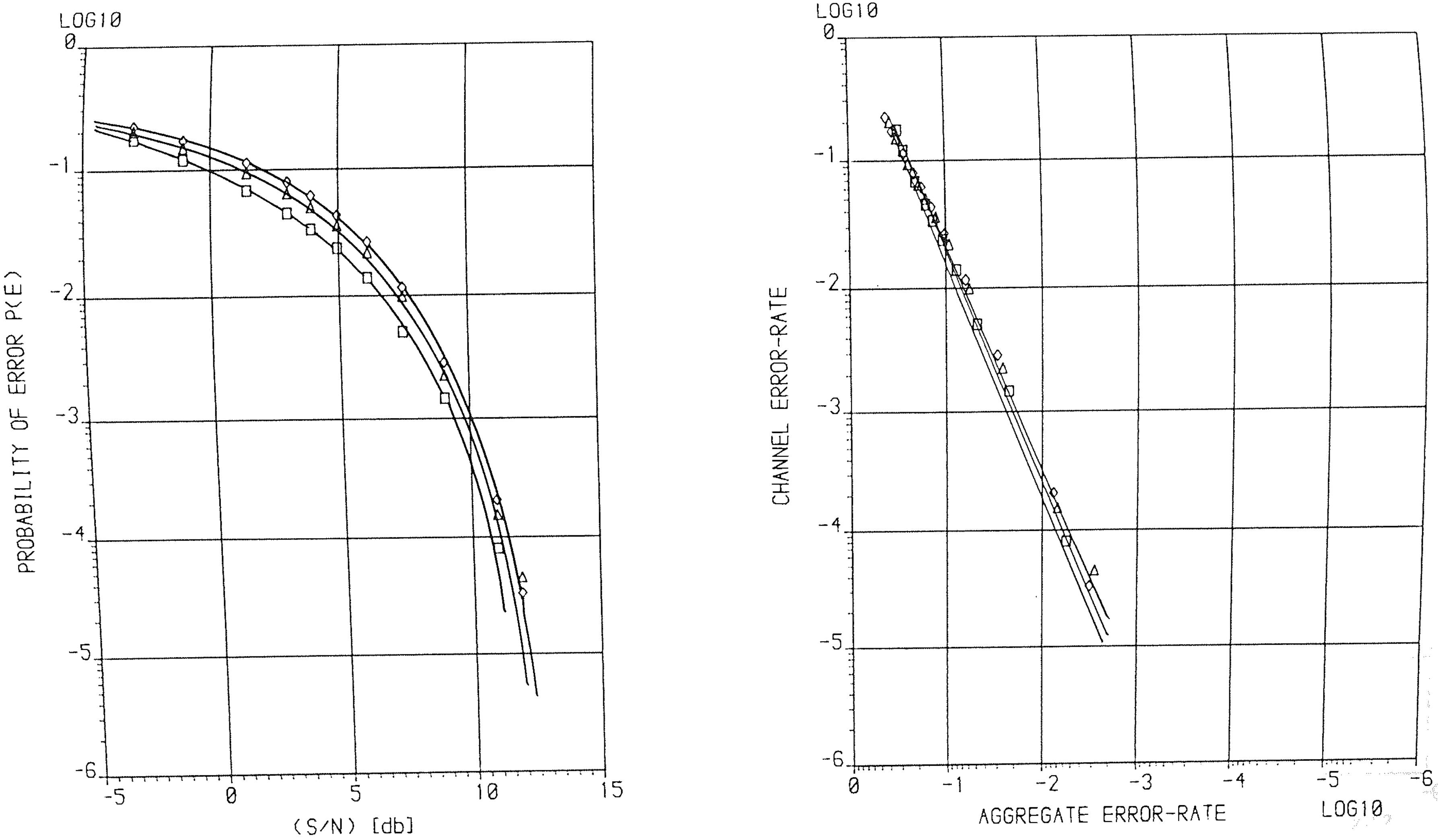


FIG.(6.6-5):Probability of error for Gaus. noise using Bridge functions of 16 bits as codes

□ □ □ 10 CHANNELS   △ △ △ 11 CHANNELS   ◇ ◇ 93 CHANNELS

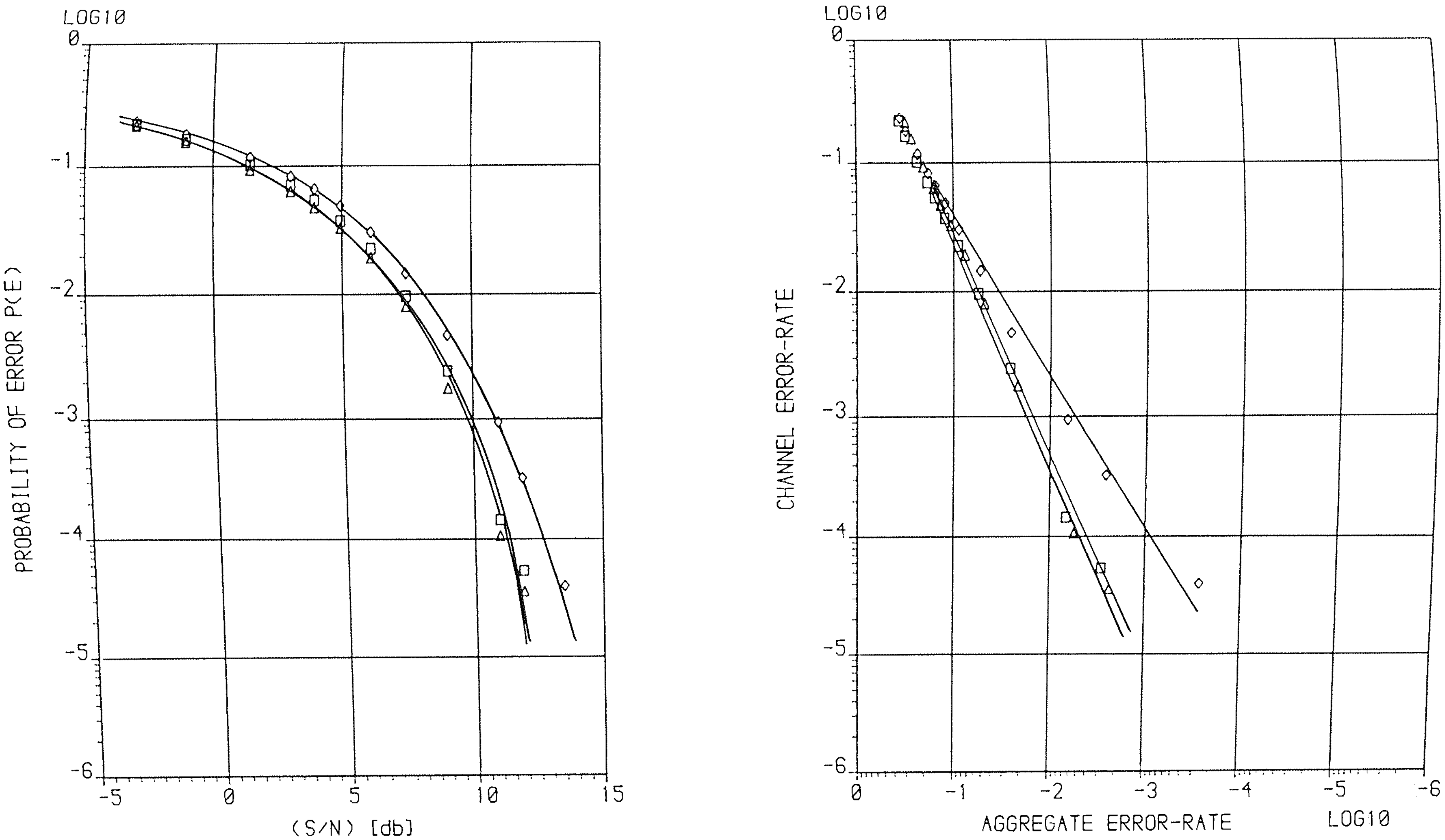


FIG.(6.6-6):Probability of error for Gaus. noise using Bridge functions of 16 bits as codes

□ □-□13 CHANNELS △ △ △14 CHANNELS ◇ ◇ ◇15 CHANNELS

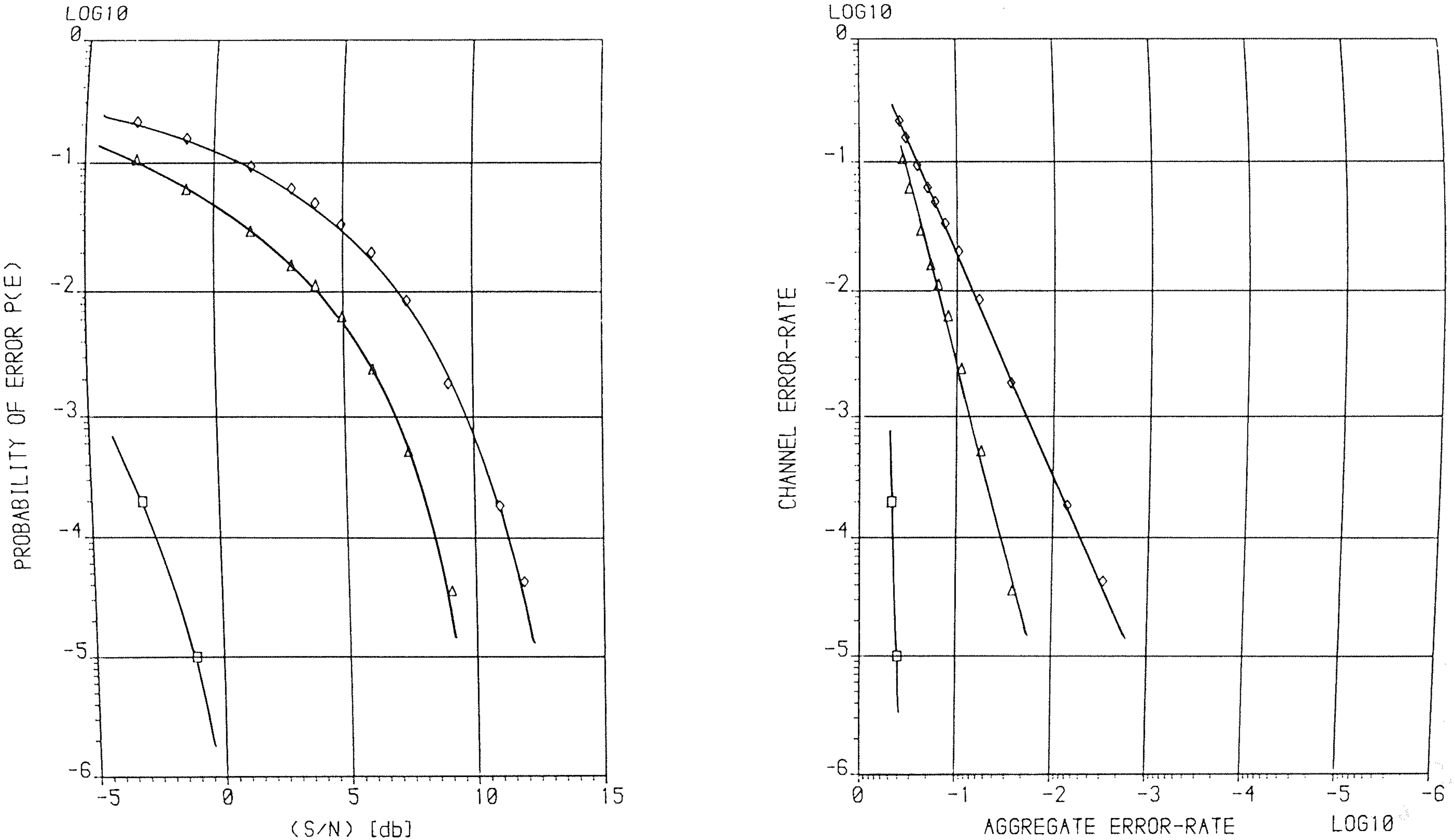


FIG.(6.7 -1):Probability of error for Gaus. noise using Bridge functions of 32 bits as codes

□ □ □ 1 CHANNEL   △ △ △ 11 CHANNELS   ◇ ◇ ◇ 21 CHANNELS

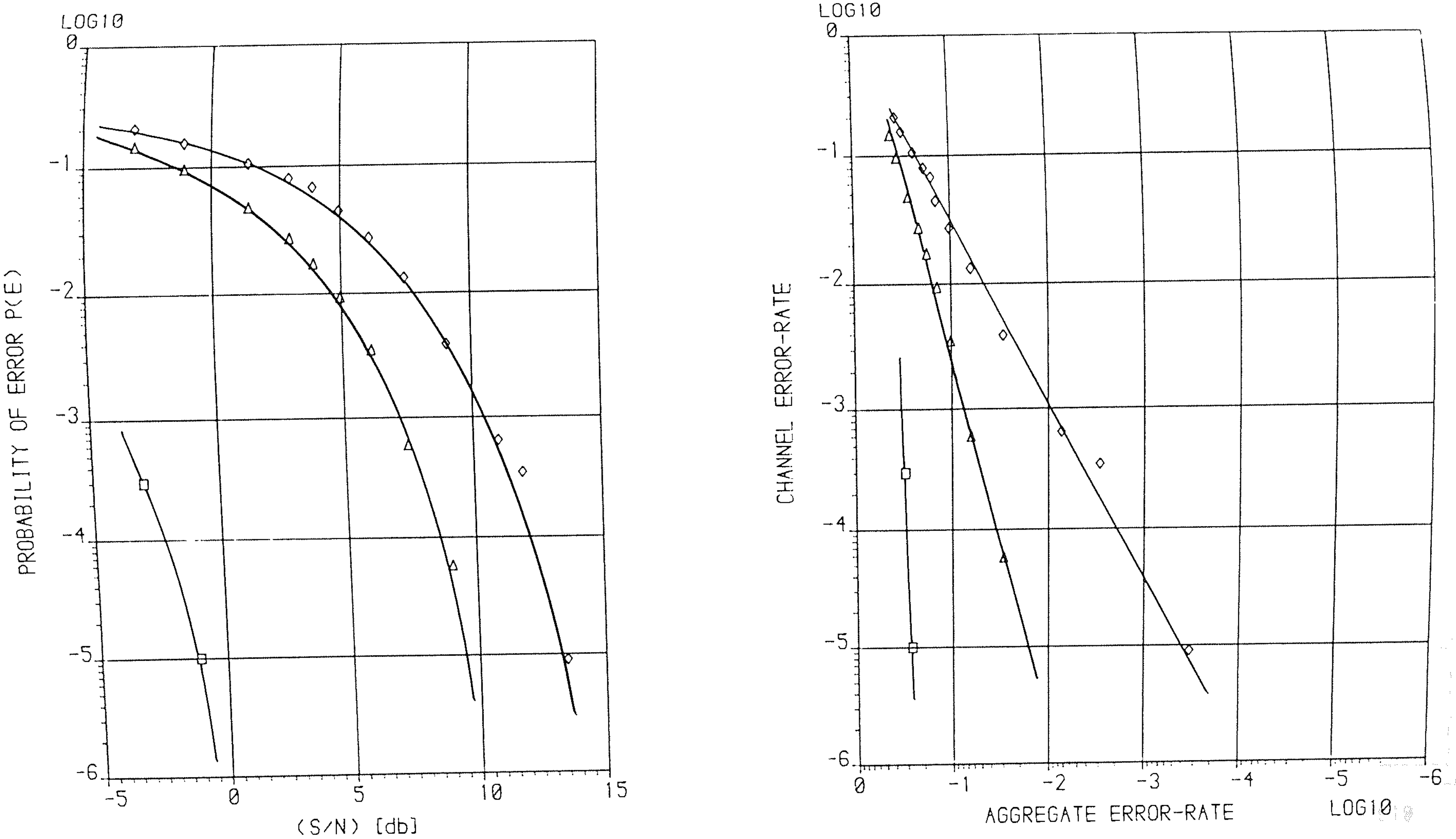


FIG.(6.7 -2):Probability of error for Gaus. noise using Bridge functions of 32 bits as codes

□□□ 2 CHANNELS △△△12 CHANNELS ◇◇◇ 22 CHANNELS

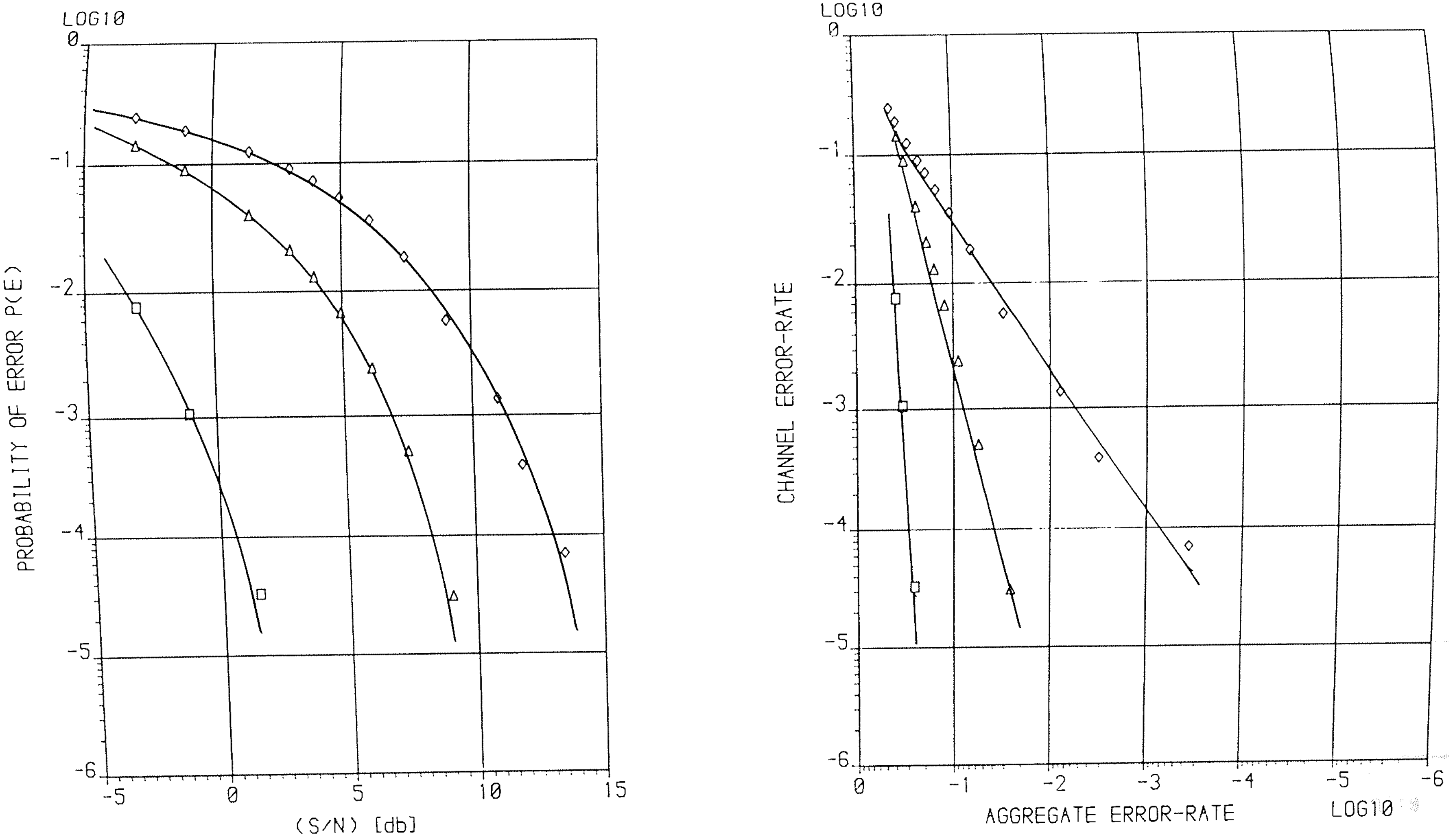


FIG.(6.7 -3):Probability of error for Gaus. noise using Bridge functions of 32 bits as codes

□ □ □ 3 CHANNELS   △ △ △ 13 CHANNELS   ◇ ◇ ◇ 23 CHANNELS

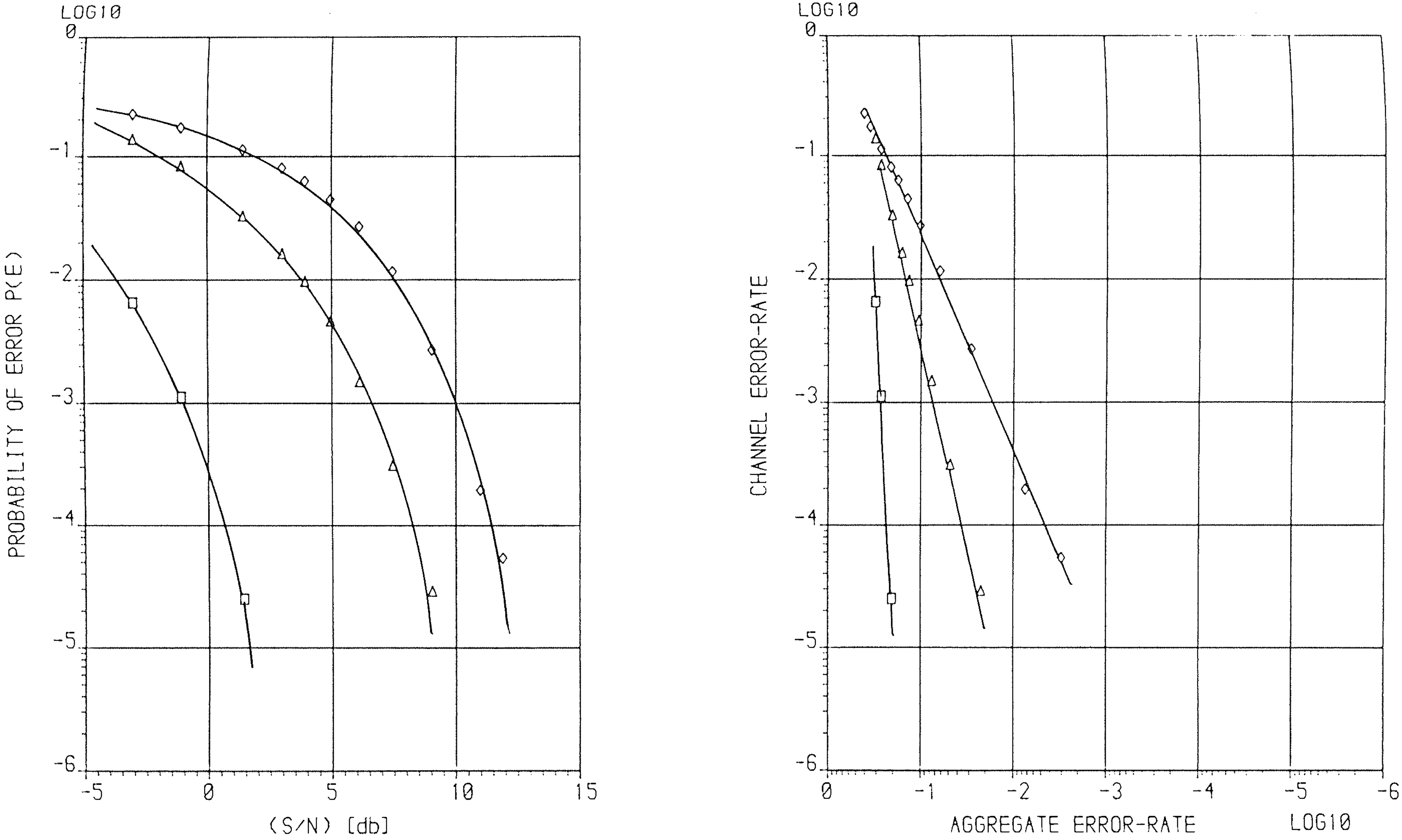


FIG.(6.7 -4):Probability of error for Gaus. noise using Bridge functions of 32 bits as codes

□ □ □ 4 CHANNELS   △ △ △ 14 CHANNELS   ◇ ◇ ◇ 24 CHANNELS

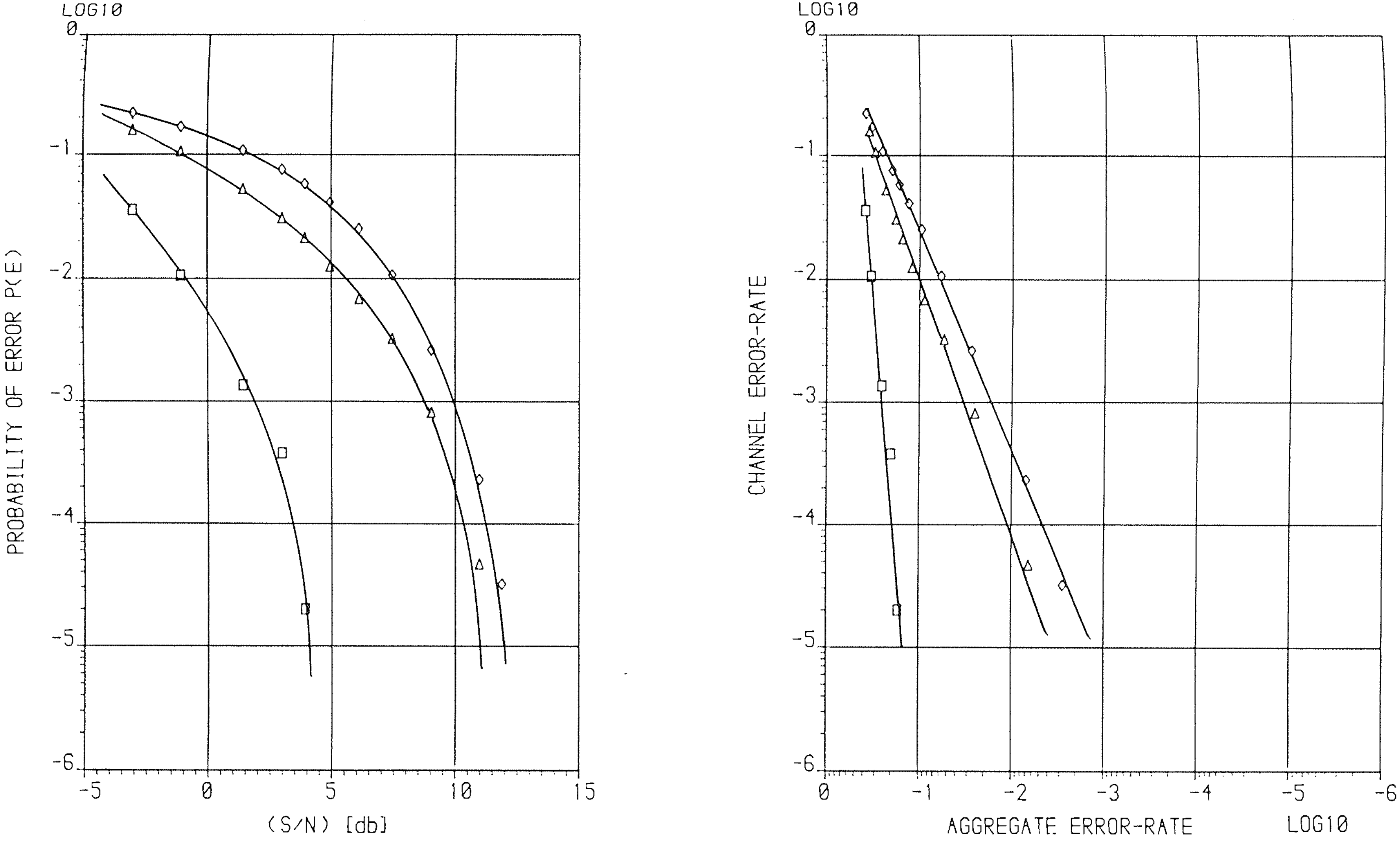


FIG.(6.7 -5):Probability of error for Gaus. noise using Bridge functions of 32 bits as codes

□ □ □ 5 CHANNELS ∆ ∆ ∆ 15 CHANNELS ◇ ◇ ◇ 25 CHANNELS

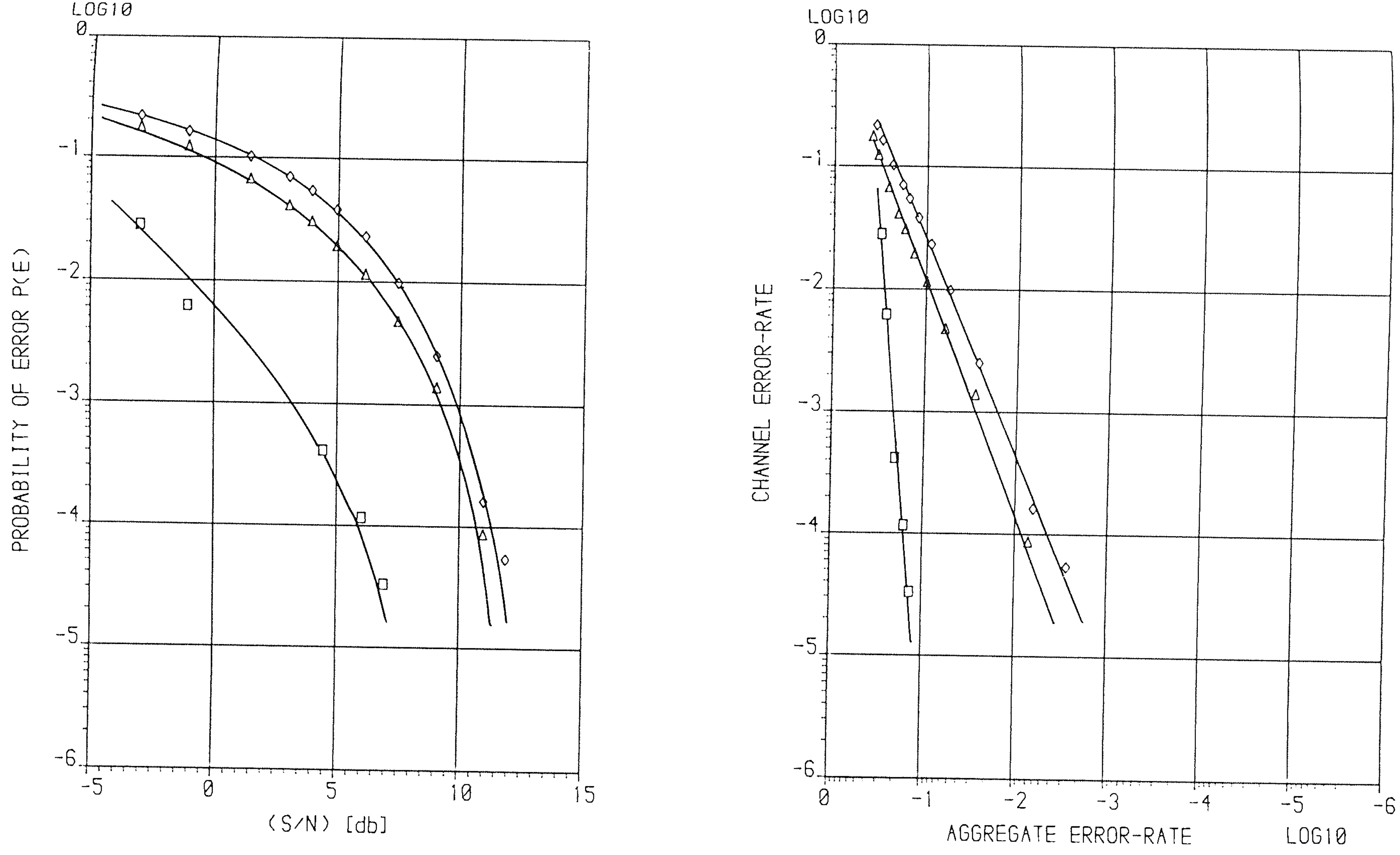


FIG.(6.7 -6):Probability of error for Gaus. noise using Bridge functions of 32 bits as codes

□ □ □ 6 CHANNELS   △ △ △ 16 CHANNELS   ◇ ◇ ◇ 26 CHANNELS

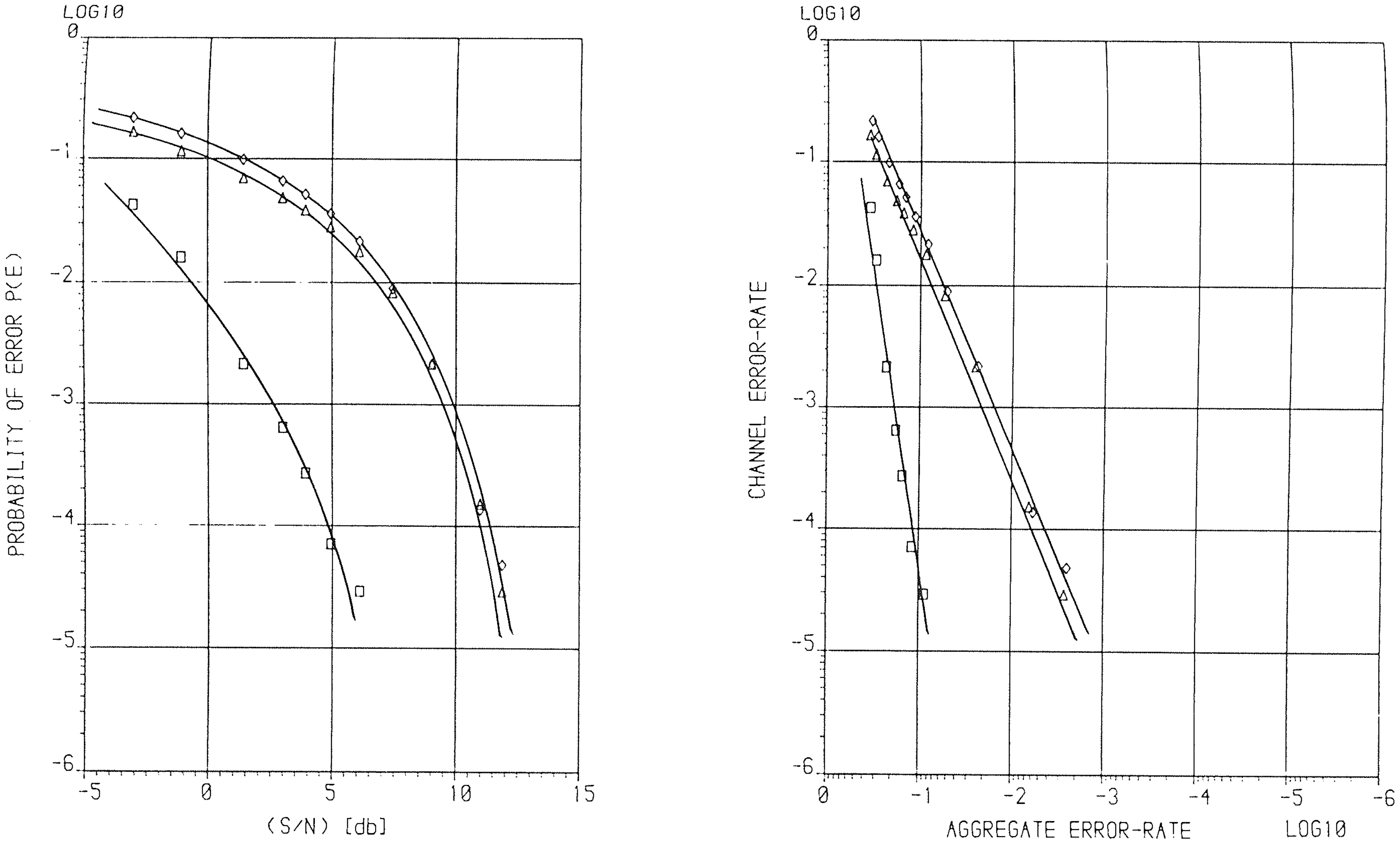


FIG.(6.7 -7):Probability of error for Gaus. noise using Bridge functions of 32 bits as codes

□ □ □ 7 CHANNELS △ △ △ 17 CHANNELS ◇ ◇ ◇ 27 CHANNELS

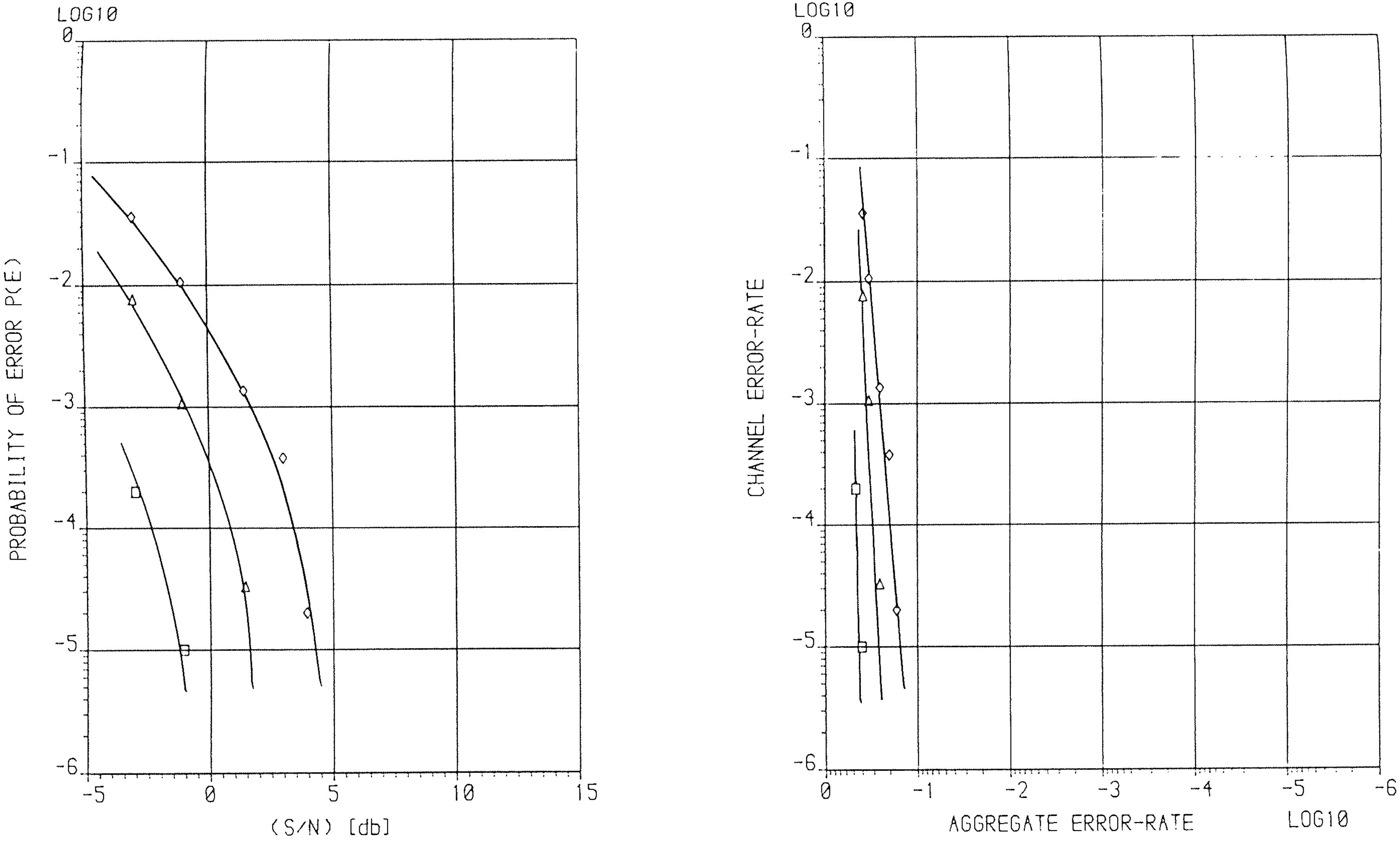


FIG.(6.7 -8):Probability of error for Gaus. noise using Bridge functions of 32 bits as codes

□□□ 1 CHANNEL ∆∆∆ 3 CHANNELS ◇◇◇ 5 CHANNELS

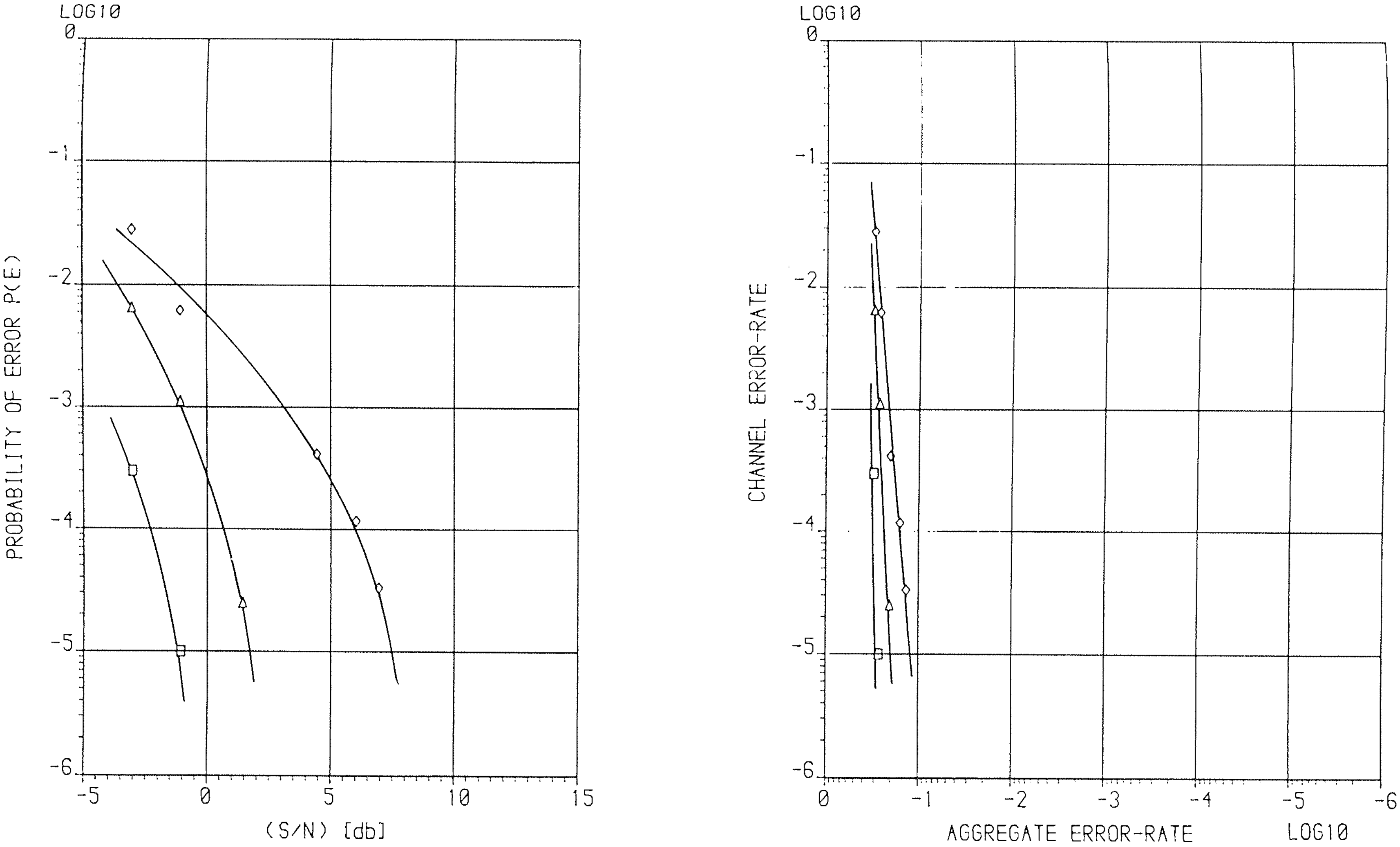


FIG.(6.7 -9):Probability of error for Gaus. noise using Bridge functions of 32 bits as codes

□□□ 2 CHANNELS   △△△ 4 CHANNELS   ◇◇◇ 6 CHANNELS

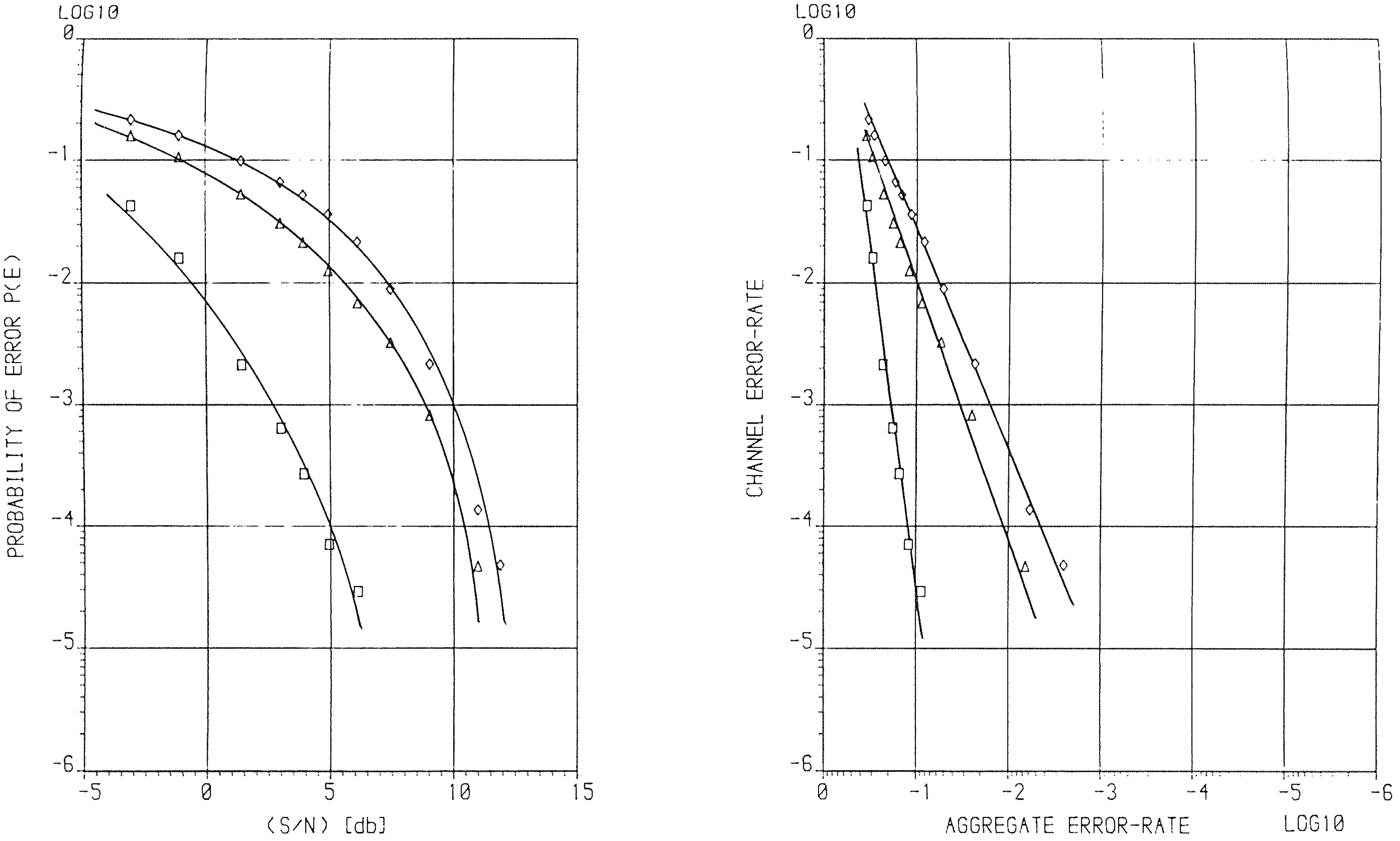


FIG.(6.7-10):Probability of error for Gaus. noise using Bridge functions of 32 bits as codes

□ □ □ 7 CHANNELS ∆ ∆ ∆ 15 CHANNELS ◇ ◇ ◇ 27 CHANNELS

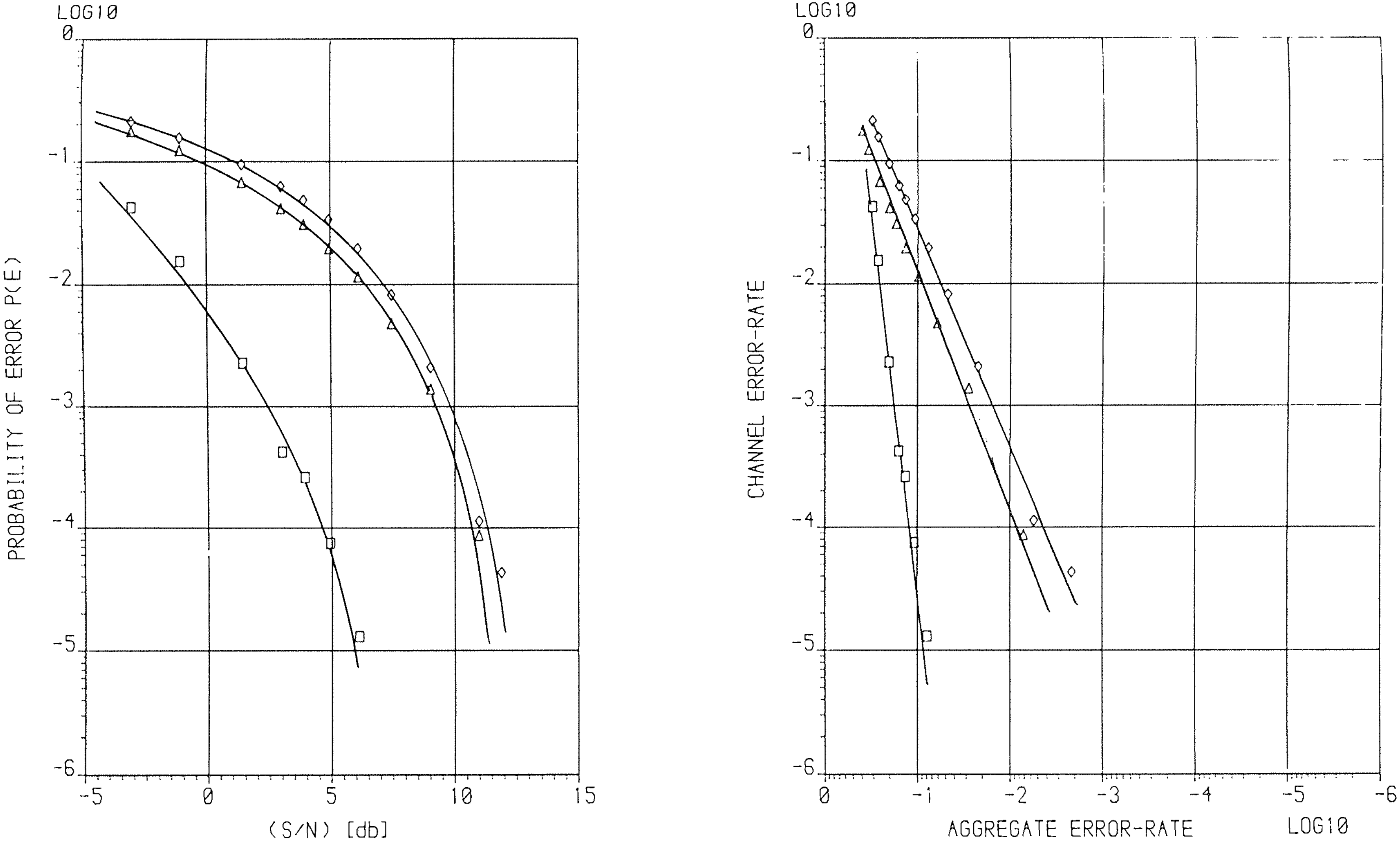


FIG.(6.7-11):Probability of error for Gaus. noise using Bridge functions of 32 bits as codes

□ □ □ 8 CHANNELS △ △ △ 16 CHANNELS ◇ ◇ ◇ 28 CHANNELS

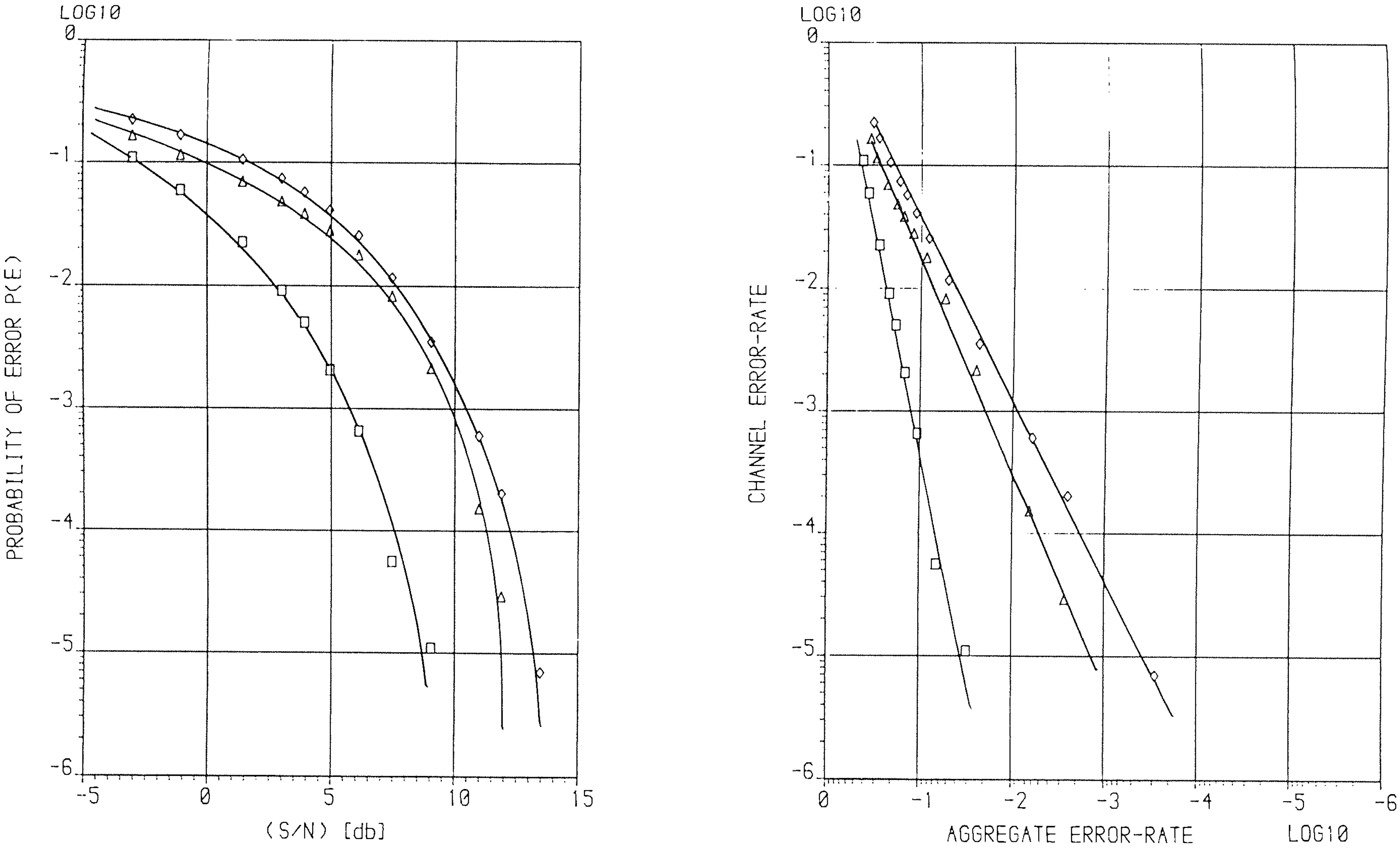


FIG.(6.7-12):Probability of error for Gaus. noise using Bridge functions of 32 bits as codes

□□□ 9 CHANNELS △△△17 CHANNELS ◇◇◇ 29 CHANNELS

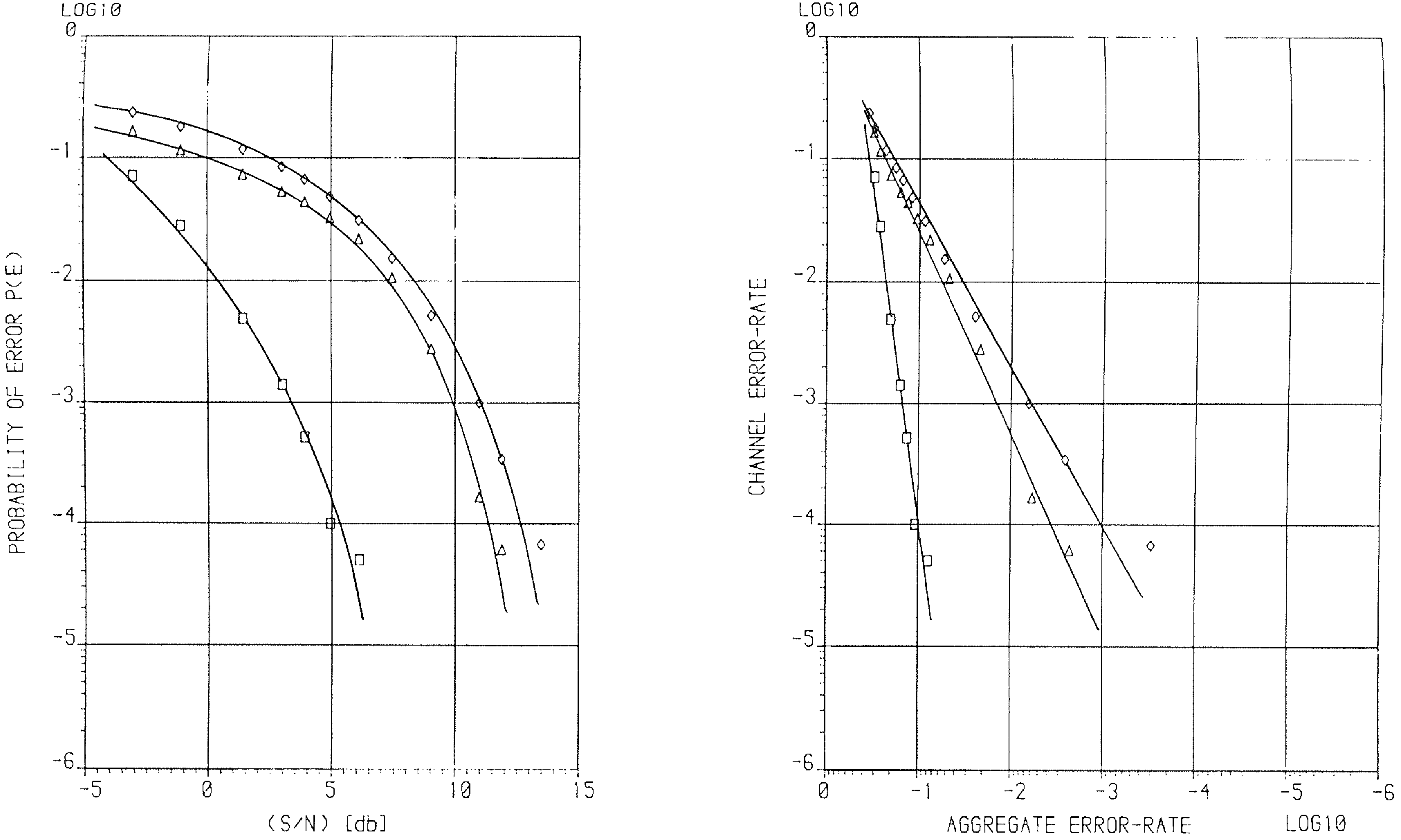


FIG.(6.7-13):Probability of error for Gaus. noise using Bridge functions of 32 bits as codes

□ □ □ 10 CHANNELS △ △ △ 18 CHANNELS ◇ ◇ ◇ 30 CHANNELS

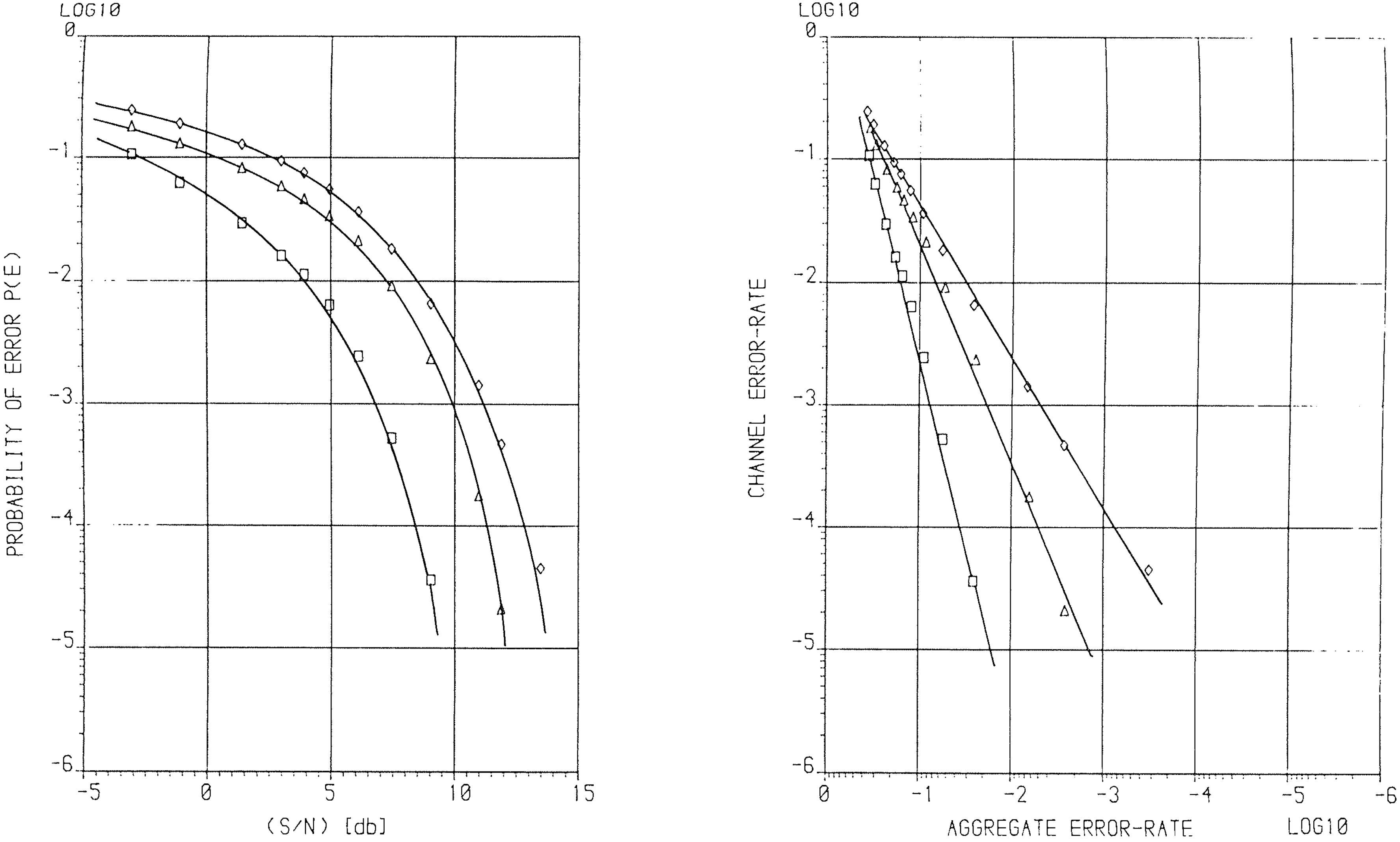


FIG.(6.7-14):Probability of error for Gaus. noise using Bridge functions of 32 bits as codes

□ □ □ 11 CHANNELS ∆ ∆ ∆ 19 CHANNELS ◇ ◇ ◇ 31 CHANNELS

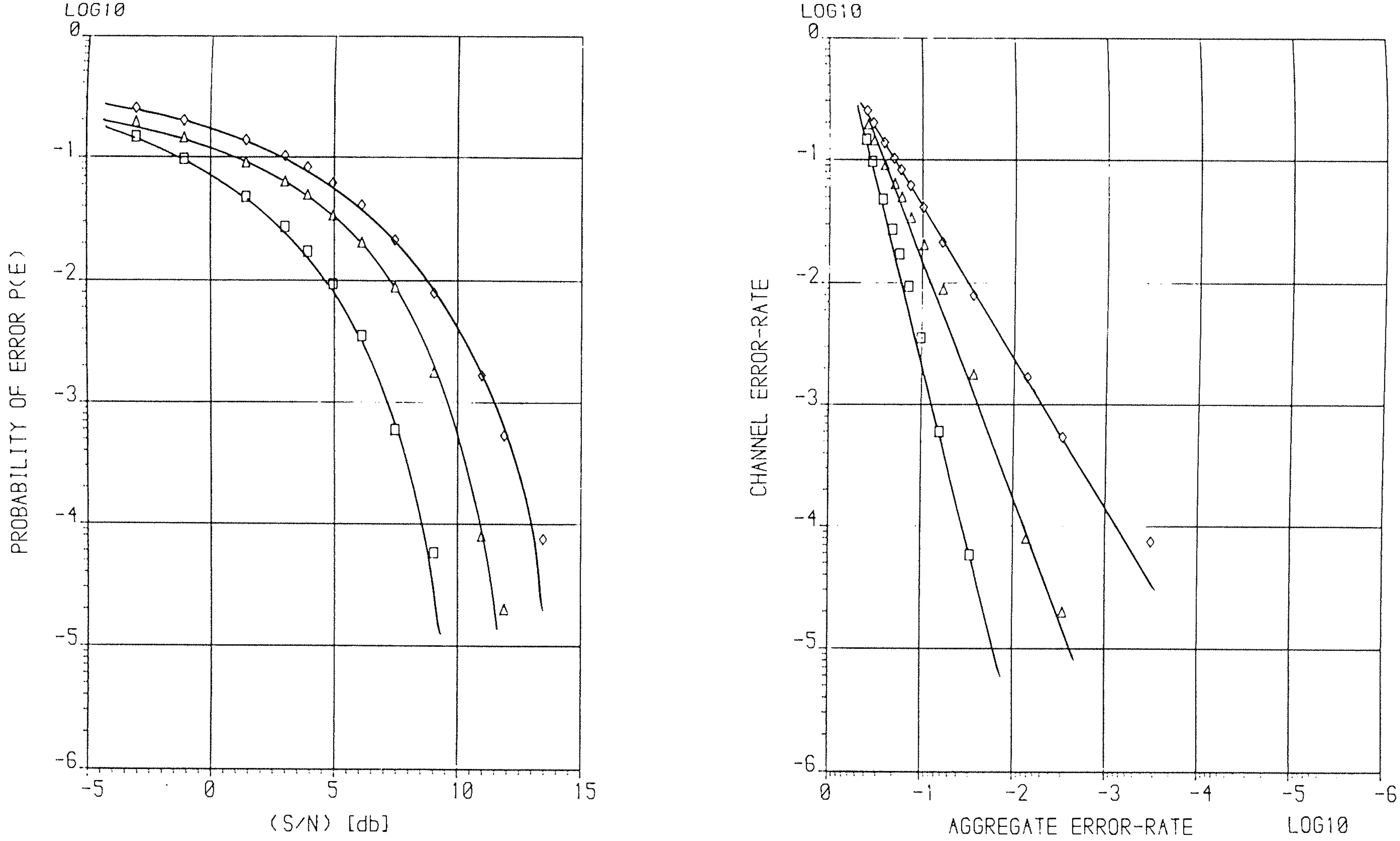


FIG.(6.7-15):Probability of error for Gaus. noise using Bridge functions of 32 bits as codes

□□□ 12 CHANNELS △△△ 20 CHANNELS ◇◇◇ 32 CHANNELS

Tables:(6.1)

(S/N)db	AG.ERR.	CH.ERR.
13.47	.31E -3	.00E 0
11.88	.36E -2	.00E 0
10.97	.83E -2	.00E 0
9.03	.33E -1	.00E 0
7.45	.72E -1	.00E 0
6.11	.11E 0	.00E 0
4.95	.16E 0	.00E 0
3.93	.20E 0	.10E -3
3.01	.24E 0	.40E -3
1.43	.31E 0	.22E -2

1 CHANNEL

(S/N)db	AG.ERR.	CH.ERR.
13.47	.22E -3	.00E 0
11.88	.24E -2	.00E 0
10.97	.57E -2	.00E 0
9.03	.22E -1	.00E 0
7.45	.47E -1	.00E 0
6.11	.76E -1	.00E 0
4.95	.11E 0	.50E -4
3.93	.14E 0	.15E -3
3.01	.16E 0	.45E -3
1.43	.20E 0	.19E -2

2 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
13.47	.34E -3	.00E 0
11.88	.35E -2	.00E 0
10.97	.84E -2	.10E -3
9.03	.33E -1	.97E -3
7.45	.72E -1	.34E -2
6.11	.12E 0	.84E -2
4.95	.16E 0	.14E -1
3.93	.20E 0	.23E -1
3.01	.24E 0	.33E -1
1.43	.31E 0	.56E -1

3 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
13.47	.29E -3	.00E 0
11.88	.36E -2	.25E -4
10.97	.87E -2	.75E -4
9.03	.33E -1	.13E -2
7.45	.72E -1	.66E -2
6.11	.12E 0	.15E -1
4.95	.16E 0	.29E -1
3.93	.20E 0	.44E -1
3.01	.24E 0	.60E -1
1.43	.30E 0	.91E -1

4 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
13.47	.24E -3	.00E 0
11.88	.29E -2	.20E -4
10.97	.73E -2	.40E -4
9.03	.28E -1	.10E -2
7.45	.61E -1	.46E -2
6.11	.97E -1	.11E -1
4.95	.13E 0	.19E -1
3.93	.17E 0	.30E -1
3.01	.20E 0	.43E -1
1.43	.25E 0	.69E -1

5 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
13.47	.25E -3	.00E 0
11.88	.22E -2	.17E -4
10.97	.58E -2	.10E -3
9.03	.22E -1	.68E -3
7.45	.47E -1	.31E -2
6.11	.78E -1	.78E -2
4.95	.11E 0	.15E -1
3.93	.13E 0	.21E -1
3.01	.16E 0	.33E -1
1.43	.20E 0	.56E -1

6 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
13.47	.31E -3	.19E -3
11.88	.27E -2	.15E -2
10.97	.69E -2	.40E -2
9.03	.27E -1	.12E -1
7.45	.56E -1	.27E -1
6.11	.91E -1	.42E -1
4.95	.13E 0	.59E -1
3.93	.16E 0	.74E -1
3.01	.19E 0	.80E -1
1.43	.24E 0	.11E 0

7 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
13.47	.30E -3	.26E -3
11.88	.33E -2	.25E -2
10.97	.80E -2	.60E -2
9.03	.31E -1	.22E -1
7.45	.65E -1	.45E -1
6.11	.11E 0	.68E -1
4.95	.15E 0	.92E -1
3.93	.18E 0	.11E 0
3.01	.22E 0	.13E 0
1.43	.28E 0	.16E 0

8 CHANNELS

Table:(G.2.1)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER
0.15	25	0	0	0	0	0	0	0
	25	0	0	0	0	0	0	0
0.18	280	4	0	0	0	0	0	0
	280	4	0	0	0	0	0	0
0.20	618	22	2	0	0	0	0	0
	618	22	2	0	0	0	0	0
0.25	2075	244	18	1	0	0	0	0
	2075	244	18	1	0	0	0	0
0.30	3422	926	149	13	1	0	0	0
	3422	926	149	13	1	0	0	0
0.35	3973	1787	460	58	8	0	0	0
	3973	1787	460	58	8	0	0	0
0.40	3842	2557	974	211	23	0	0	0
	3842	2557	974	211	23	0	0	0
0.45	3389	2996	1471	458	89	6	0	0
	3389	2996	1471	459	88	6	0	0
0.50	2915	3143	1925	768	179	25	0	0
	2915	3144	1925	767	181	23	0	0
0.60	1905	3012	2637	1381	498	93	9	0
	1905	3014	2642	1380	498	88	8	0

Table:(G.2.2)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER
0.15	18	0	0	0	0	0	0	0
	18	0	0	0	0	0	0	0
0.18	188	3	0	0	0	0	0	0
	188	3	0	0	0	0	0	0
0.20	423	15	0	0	0	0	0	0
	423	15	0	0	0	0	0	0
0.25	1499	112	6	0	0	0	0	0
	1499	112	6	0	0	0	0	0
0.30	2682	447	44	3	1	0	0	0
	2682	447	44	3	1	0	0	0
0.35	3517	1012	167	17	0	0	0	0
	3517	1012	167	17	0	0	0	0
0.40	3923	1632	361	56	7	0	0	0
	3923	1632	362	55	7	0	0	0
0.45	3920	2141	672	134	10	3	0	0
	3920	2142	671	135	10	2	0	0
0.50	3745	2493	975	234	17	3	0	0
	3745	2494	982	226	17	3	0	0
0.60	3337	2937	1557	471	87	15	0	0
	3337	2948	1561	466	80	12	0	0

Table:(6.2.3)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER
0.15	27	0	0	0	0	0	0	0
	27	0	0	0	0	0	0	0
0.18	270	5	0	0	0	0	0	0
	270	5	0	0	0	0	0	0
0.20	638	17	0	0	0	0	0	0
	641	14	0	0	0	0	0	0
0.25	2113	250	18	0	0	0	0	0
	2127	236	13	0	0	0	0	0
0.30	3388	939	141	7	1	0	0	0
	3432	913	113	6	0	0	0	0
0.35	3974	1783	485	53	6	0	0	0
	4064	1778	401	33	6	0	0	0
0.40	3849	2566	968	208	23	2	0	0
	3974	2577	875	155	8	1	0	0
0.45	3451	2938	1506	441	92	9	0	0
	3612	2982	1436	311	56	2	0	0
0.50	2849	3193	1986	728	194	19	1	0
	3044	3314	1885	582	105	3	1	0
0.60	1894	3010	2662	1394	485	95	5	2
	2084	3306	2605	1194	279	25	1	1

Table:(6.2.4)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER
0.15	23	0	0	0	0	0	0	0
	23	0	0	0	0	0	0	0
0.18	282	4	0	0	0	0	0	0
	283	3	0	0	0	0	0	0
0.20	658	19	1	0	0	0	0	0
	660	18	0	0	0	0	0	0
0.25	2070	262	21	0	0	0	0	0
	2108	232	11	0	0	0	0	0
0.30	3397	934	147	12	0	0	0	0
	3555	842	74	7	0	0	0	0
0.35	4038	1778	485	55	8	0	0	0
	4361	1677	288	29	0	0	0	0
0.40	3837	2514	994	198	24	0	0	0
	4300	2523	627	89	4	0	0	0
0.45	3353	2996	1506	435	93	8	0	0
	3922	3146	1094	185	22	1	0	0
0.50	2824	3151	2009	756	171	18	0	0
	3415	3553	1552	348	36	2	0	0
0.60	1910	3046	2576	1423	452	95	10	1
	2462	3884	2210	776	133	10	0	0

Table:(6.2.5)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER
0.15	19	0	0	0	0	0	0	0
	19	0	0	0	0	0	0	0
0.18	232	2	0	0	0	0	0	0
	233	1	0	0	0	0	0	0
0.20	566	11	0	0	0	0	0	0
	568	9	0	0	0	0	0	0
0.25	1800	180	18	0	0	0	0	0
	1833	151	9	0	0	0	0	0
0.30	3147	734	76	6	0	0	0	0
	3296	611	36	3	0	0	0	0
0.35	3820	1435	288	45	3	0	0	0
	4101	1278	166	14	0	0	0	0
0.40	4001	2008	642	101	9	0	0	0
	4405	2007	359	22	0	0	0	0
0.45	3720	2705	1036	241	30	2	0	0
	4312	2650	632	83	3	0	0	0
0.50	3420	3027	1467	421	89	11	0	0
	4120	3131	952	150	16	0	0	0
0.60	2662	3131	2180	882	220	38	2	0
	3475	3559	1580	341	41	1	0	0

Table:(6.2.6)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER
0.15	20	0	0	0	0	0	0	0
	20	0	0	0	0	0	0	0
0.18	174	1	0	0	0	0	0	0
	175	0	0	0	0	0	0	0
0.20	442	10	0	0	0	0	0	0
	446	5	0	0	0	0	0	0
0.25	1535	101	3	1	0	0	0	0
	1559	72	2	0	0	0	0	0
0.30	2653	499	44	4	0	0	0	0
	2755	403	17	0	0	0	0	0
0.35	3547	1015	181	30	0	0	0	0
	3777	868	79	6	0	0	0	0
0.40	3880	1656	385	51	3	0	0	0
	4272	1438	164	6	1	0	0	0
0.45	3960	2165	625	123	13	1	0	0
	4462	1971	315	27	0	1	0	0
0.50	3774	2537	933	259	32	2	0	0
	4453	2366	481	58	9	1	0	0
0.60	3264	3048	1514	457	105	11	1	0
	4183	2997	778	115	10	2	0	0

Table:(6.2.7)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER
0.15	25	0	0	0	0	0	0	0
	16	0	0	0	0	0	0	0
0.18	213	1	0	0	0	0	0	0
	148	0	0	0	0	0	0	0
0.20	507	21	1	0	0	0	0	0
	352	9	0	0	0	0	0	0
0.25	1760	166	9	1	0	0	0	0
	1325	77	2	0	0	0	0	0
0.30	2977	639	80	5	0	0	0	0
	2423	291	10	0	0	0	0	0
0.35	3638	1351	263	36	4	0	0	0
	3272	691	51	4	0	0	0	0
0.40	3915	1998	591	99	10	0	0	0
	3853	1153	109	8	0	0	0	0
0.45	3808	2447	1009	207	32	1	0	0
	4142	1587	222	20	0	0	0	0
0.50	3492	2798	1293	429	80	10	0	0
	4195	1987	378	37	0	0	0	0
0.60	2730	3088	2006	831	194	37	0	0
	4067	2652	661	105	6	2	0	0

Table:(6.2.8)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER
0.15	24	0	0	0	0	0	0	0
	12	0	0	0	0	0	0	0
0.18	252	5	0	0	0	0	0	0
	128	2	0	0	0	0	0	0
0.20	598	21	1	0	0	0	0	0
	310	1	0	0	0	0	0	0
0.25	1933	234	12	1	0	0	0	0
	1099	44	0	0	0	0	0	0
0.30	3259	779	123	10	0	0	0	0
	2072	197	5	0	0	0	0	0
0.35	3889	1576	405	62	7	0	0	0
	3003	525	23	2	0	0	0	0
0.40	3882	2221	870	178	19	1	0	0
	3515	889	56	1	0	0	0	0
0.45	3440	2699	1325	358	85	6	0	0
	3697	1272	151	18	0	0	0	0
0.50	3079	2993	1683	604	160	25	1	0
	3903	1675	279	23	1	0	0	0
0.60	2301	2942	2317	1178	414	83	6	1
	3892	2332	529	77	5	0	0	0

Tables:<G.3>

(S/N)db	AG.ERR.	CH.ERR.
10.97	.10E -1	.00E 0
9.03	.34E -1	.00E 0
7.45	.72E -1	.00E 0
6.11	.12E 0	.00E 0
4.95	.16E 0	.00E 0
3.93	.20E 0	.00E 0
3.01	.24E 0	.00E 0
1.43	.31E 0	.20E -4
-1.07	.40E 0	.20E -3
-3.01	.47E 0	.60E -2

1 CHANNEL

(S/N)db	AG.ERR.	CH.ERR.
10.97	.64E -2	.00E 0
9.03	.23E -1	.00E 0
7.45	.46E -1	.00E 0
6.11	.77E -1	.00E 0
4.95	.10E 0	.00E 0
3.93	.14E 0	.00E 0
3.01	.16E 0	.50E -5
1.43	.20E 0	.50E -4
-1.07	.27E 0	.86E -3
-3.01	.31E 0	.63E -2

2 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
13.47	.26E -3	.00E 0
11.88	.28E -2	.00E 0
10.97	.70E -2	.00E 0
9.03	.28E -1	.00E 0
7.45	.60E -1	.00E 0
6.11	.96E -1	.33E -4
4.95	.13E 0	.10E -3
3.93	.17E 0	.33E -3
3.01	.20E 0	.57E -3
1.43	.26E 0	.20E -2

3 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
13.47	.35E -3	.00E 0
11.88	.35E -2	.00E 0
10.97	.90E -2	.00E 0
9.03	.35E -1	.00E 0
7.45	.72E -1	.00E 0
6.11	.12E 0	.25E -3
4.95	.16E 0	.15E -2
3.93	.20E 0	.55E -2
3.01	.23E 0	.10E -1
1.43	.30E 0	.27E -1

4 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
13.47	.32E -3	.00E 0
11.88	.27E -2	.00E 0
10.97	.67E -2	.00E 0
9.03	.20E -1	.00E 0
7.45	.60E -1	.00E 0
6.11	.98E -1	.40E -3
4.95	.13E 0	.10E -2
3.93	.17E 0	.30E -2
3.01	.20E 0	.68E -2
1.43	.26E 0	.20E -1

5 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
13.47	.22E -3	.00E 0
11.88	.22E -2	.00E 0
10.97	.64E -2	.00E 0
9.03	.23E -1	.00E 0
7.45	.46E -1	.00E 0
6.11	.78E -1	.33E -3
4.95	.11E 0	.83E -3
3.93	.13E 0	.15E -2
3.01	.16E 0	.43E -2
1.43	.20E 0	.11E -1

6 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
13.47	.25E -3	.00E 0
11.88	.25E -2	.00E 0
10.97	.63E -2	.14E -4
9.03	.25E -1	.23E -3
7.45	.54E -1	.93E -3
6.11	.88E -1	.26E -2
4.95	.12E 0	.51E -2
3.93	.15E 0	.85E -2
3.01	.18E 0	.13E -1
1.43	.23E 0	.25E -1

7 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
13.47	.21E -3	.00E 0
11.88	.24E -2	.00E 0
10.97	.57E -2	.00E 0
9.03	.22E -1	.21E -3
7.45	.48E -1	.10E -2
6.11	.78E -1	.29E -2
4.95	.11E 0	.54E -2
3.93	.14E 0	.92E -2
3.01	.16E 0	.14E -1
1.43	.20E 0	.26E -1

8 CHANNELS

Tables:(6.3)

(S/N)db	AG.ERR.	CH.ERR.
13.47	.30E -3	.00E 0
11.88	.31E -2	.00E 0
10.97	.80E -2	.33E -4
9.03	.31E -1	.11E -2
7.45	.66E -1	.52E -2
6.11	.11E 0	.13E -1
4.95	.15E 0	.23E -1
3.93	.19E 0	.36E -1
3.01	.22E 0	.50E -1
1.43	.28E 0	.78E -1

9 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
13.47	.22E -3	.00E 0
11.88	.23E -2	.00E 0
10.97	.56E -2	.80E -4
9.03	.22E -1	.15E -2
7.45	.48E -1	.51E -2
6.11	.78E -1	.14E -1
4.95	.11E 0	.24E -1
3.93	.14E 0	.34E -1
3.01	.16E 0	.45E -1
1.43	.20E 0	.68E -1

10 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
13.47	.24E -3	.00E 0
11.88	.25E -2	.90E -5
10.97	.61E -2	.27E -4
9.03	.25E -1	.80E -3
7.45	.54E -1	.37E -2
6.11	.87E -1	.94E -2
4.95	.12E 0	.17E -1
3.93	.15E 0	.26E -1
3.01	.18E 0	.37E -1
1.43	.23E 0	.60E -1

11 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
13.47	.24E -3	.00E 0
11.88	.24E -2	.00E 0
10.97	.57E -2	.92E -4
9.03	.22E -1	.94E -3
7.45	.49E -1	.31E -2
6.11	.77E -1	.79E -2
4.95	.11E 0	.15E -1
3.93	.13E 0	.23E -1
3.01	.16E 0	.32E -1
1.43	.20E 0	.53E -1

12 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
13.47	.24E -3	.00E 0
11.88	.28E -2	.54E -4
10.97	.68E -2	.15E -3
9.03	.26E -1	.25E -2
7.45	.55E -1	.97E -2
6.11	.90E -1	.23E -1
4.95	.12E 0	.37E -1
3.93	.16E 0	.54E -1
3.01	.19E 0	.71E -1
1.43	.24E 0	.10E 0

13 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
13.47	.22E -3	.00E 0
11.88	.23E -2	.36E -4
10.97	.55E -2	.11E -3
9.03	.22E -1	.18E -2
7.45	.48E -1	.81E -2
6.11	.77E -1	.19E -1
4.95	.11E 0	.33E -1
3.93	.13E 0	.48E -1
3.01	.16E 0	.64E -1
1.43	.20E 0	.94E -1

14 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
13.47	.27E -3	.40E -4
11.88	.26E -2	.33E -3
10.97	.66E -2	.94E -3
9.03	.26E -1	.47E -2
7.45	.53E -1	.15E -1
6.11	.88E -1	.30E -1
4.95	.12E 0	.49E -1
3.93	.15E 0	.66E -1
3.01	.18E 0	.83E -1
1.43	.23E 0	.12E 0

15 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
13.47	.33E -3	.87E -4
11.88	.30E -2	.61E -3
10.97	.71E -2	.16E -2
9.03	.29E -1	.78E -2
7.45	.61E -1	.22E -1
6.11	.99E -1	.41E -1
4.95	.14E 0	.62E -1
3.93	.17E 0	.86E -1
3.01	.21E 0	.10E 0
1.43	.26E 0	.14E 0

16 CHANNELS

Table :(G.4.1)

ST.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.20	137	11	1	0	0	0	0	0	0
	137	11	1	0	0	0	0	0	0
0.25	320	86	14	1	0	0	0	0	0
	320	86	14	1	0	0	0	0	0
0.30	388	219	82	17	1	0	0	0	0
	388	219	82	17	1	0	0	0	0
0.35	283	294	181	78	20	2	1	0	0
	283	294	181	78	20	2	1	0	0
0.40	183	279	249	147	62	18	5	3	0
	183	279	249	147	62	18	5	3	0
0.45	10426	20832	25020	20435	12370	5654	2036	542	96
	10426	20832	25020	20435	12370	5654	2036	542	96
0.50	5762	14466	21945	23078	17242	9966	4419	1477	416
	5762	14466	21945	23078	17242	9966	4419	1477	416
0.60	1843	6316	13378	20215	21895	17365	10804	5277	1930
	1843	6316	13378	20215	21895	17366	10803	5277	1931
0.80	15	128	424	938	1674	2075	1940	1452	804
	15	128	424	938	1674	2075	1940	1452	805
1.00	3	25	142	427	935	1589	2007	1954	1496
	3	25	142	427	935	1590	2007	1963	1500

Table :(G.4.2)

ST.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.20	96	3	0	0	0	0	0	0	0
	96	3	0	0	0	0	0	0	0
0.25	258	49	4	0	0	0	0	0	0
	258	49	4	0	0	0	0	0	0
0.30	366	141	26	4	0	0	0	0	0
	366	141	26	4	0	0	0	0	0
0.35	367	250	77	28	6	0	0	0	0
	367	250	77	28	6	0	0	0	0
0.40	316	312	153	51	9	4	1	0	0
	316	312	153	51	9	4	1	0	0
0.45	24594	28637	20786	10644	3978	1154	288	48	4
	24594	28637	20786	10644	3978	1154	288	48	4
0.50	18601	26809	23890	14662	6832	2394	649	178	21
	18601	26809	23890	14662	6832	2394	650	177	21
0.60	1040	2053	2503	2006	1254	590	220	48	19
	1040	2053	2503	2006	1254	590	220	48	19
0.80	3911	10861	18598	22371	19574	13066	6841	2857	923
	3911	10861	18598	22374	19582	13098	6845	2847	909
1.00	186	652	1344	2052	2029	1711	1156	527	237
	186	652	1344	2053	2044	1720	1163	526	218

Table :(6.4.3)

ST.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.20	1183	74	2	0	0	0	0	0	0
	1183	74	2	0	0	0	0	0	0
0.25	3241	809	128	21	1	0	0	0	0
	3241	809	128	21	1	0	0	0	0
0.30	3760	2223	777	208	30	5	0	0	1
	3760	2223	777	208	30	5	0	0	1
0.35	2896	2988	1748	717	242	60	7	3	0
	2896	2988	1751	716	242	58	7	3	0
0.40	1828	2744	2426	1448	684	214	63	7	1
	1828	2744	2428	1463	675	207	62	7	1
0.45	1025	2138	2520	2097	1190	566	182	41	14
	1025	2138	2522	2109	1183	573	173	37	13
0.50	577	1469	2268	2249	1717	989	429	150	42
	577	1469	2274	2256	1730	983	434	132	35
0.60	180	600	1404	2026	2159	1766	1041	545	182
	180	600	1410	2052	2187	1782	1039	503	170
0.80	19	101	420	991	1674	2040	1915	1490	836
	19	104	435	1044	1745	2141	1978	1417	735
1.00	0	28	146	443	936	1565	1991	1984	1527
	0	29	155	492	1053	1749	2196	1979	1388

Table :(6.4.4)

ST.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.20	128	13	1	0	0	0	0	0	0
	128	13	1	0	0	0	0	0	0
0.25	324	90	17	0	0	0	0	0	0
	324	90	17	0	0	0	0	0	0
0.30	371	224	84	18	1	0	0	0	0
	371	224	84	18	1	0	0	0	0
0.35	294	283	185	77	23	3	0	0	0
	294	283	185	78	22	3	0	0	0
0.40	202	258	241	145	59	29	4	1	0
	202	258	242	146	58	29	4	0	0
0.45	98	204	273	197	126	55	14	8	0
	98	204	276	200	129	48	14	6	0
0.50	64	165	228	220	172	94	33	11	6
	64	165	228	230	179	84	30	11	2
0.60	23	61	135	216	209	179	101	48	17
	23	62	139	223	234	164	95	44	9
0.80	7	9	39	105	143	222	201	137	95
	7	10	40	113	180	238	205	124	60
1.00	0	6	17	49	99	174	202	173	151
	0	7	18	69	112	229	209	187	111

Table :(G.4.5)

ST.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.20	97	5	0	0	0	0	0	0	0
	97	5	0	0	0	0	0	0	0
0.25	271	66	16	0	0	0	0	0	0
	271	66	16	0	0	0	0	0	0
0.30	378	182	53	14	0	0	0	0	0
	378	182	53	14	0	0	0	0	0
0.35	314	267	152	51	9	2	0	0	0
	314	267	153	50	10	1	0	0	0
0.40	244	295	227	96	24	11	0	0	0
	244	295	228	98	22	10	0	0	0
0.45	153	275	239	157	76	36	6	2	0
	153	275	245	156	73	35	6	1	0
0.50	110	215	247	211	114	46	21	5	2
	110	215	254	211	114	46	18	3	0
0.60	39	111	194	252	204	114	46	22	6
	39	113	204	269	199	109	36	17	2
0.80	4	34	119	169	230	164	162	74	25
	4	36	127	202	243	191	126	46	17
1.00	8	12	54	119	175	217	194	126	52
	8	17	67	156	228	233	169	82	23

Table :(G.4.6)

ST.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.20	94	4	0	0	0	0	0	0	0
	94	4	0	0	0	0	0	0	0
0.25	245	54	7	0	0	0	0	0	0
	245	54	7	0	0	0	0	0	0
0.30	357	142	27	3	0	0	0	0	0
	357	142	27	3	0	0	0	0	0
0.35	376	234	99	21	4	1	0	0	0
	376	234	101	19	4	1	0	0	0
0.40	335	261	157	60	18	4	1	0	0
	335	261	159	59	18	4	0	0	0
0.45	263	300	198	95	34	11	2	1	0
	263	300	203	92	34	9	2	1	0
0.50	194	262	244	141	69	23	7	3	0
	194	262	247	149	65	19	6	1	0
0.60	111	192	241	216	119	66	14	9	4
	112	196	250	223	116	53	13	6	3
0.80	41	113	196	222	173	136	70	34	8
	42	120	217	240	195	120	46	11	4
1.00	24	65	132	185	228	173	105	58	18
	25	79	176	243	242	135	73	18	3

Table :(G.4.7)

ST.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.20	1196	71	2	0	0	0	0	0	0
	1197	69	1	0	0	0	0	0	0
0.25	3175	843	121	16	1	0	0	0	0
	3218	796	101	13	1	0	0	0	0
0.30	3749	2210	802	198	39	6	1	0	0
	3899	2154	691	159	12	2	1	0	0
0.35	2878	2964	1797	717	231	49	12	3	1
	3119	3065	1645	563	138	24	5	1	0
0.40	1817	2700	2452	1540	634	215	61	16	2
	2079	2954	2425	1287	439	118	25	4	0
0.45	1015	2098	2494	2069	1274	545	201	38	10
	1253	2474	2696	1909	908	321	84	12	2
0.50	560	1492	2167	2341	1735	960	419	168	36
	765	1940	2530	2295	1414	608	202	54	10
0.60	183	591	1406	2024	2138	1820	1055	477	204
	345	976	1845	2451	2078	1347	604	204	51
0.80	23	121	400	959	1613	2046	2033	1464	827
	126	372	904	1680	2207	2098	1485	723	273
1.00	5	21	137	415	987	1588	1995	1946	1499
	65	211	617	1146	1930	2138	1902	1197	551

Table :(G.4.8)

ST.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.20	124	13	0	0	0	0	0	0	0
	127	9	0	0	0	0	0	0	0
0.25	282	80	20	3	0	0	0	0	0
	297	68	18	0	0	0	0	0	0
0.30	338	207	62	25	1	0	0	0	0
	384	184	49	11	0	0	0	0	0
0.35	324	260	164	56	14	7	0	1	0
	372	285	118	31	7	2	0	0	0
0.40	234	284	215	117	48	15	1	2	0
	289	331	195	67	23	6	1	0	0
0.45	130	226	246	184	99	44	19	8	2
	186	291	244	153	54	14	7	1	0
0.50	107	170	231	195	141	74	42	11	3
	145	225	284	183	92	24	14	3	0
0.60	49	101	170	203	185	135	81	38	17
	65	162	251	221	161	73	40	10	5
0.80	14	40	79	146	161	173	156	113	71
	17	65	163	207	222	179	87	41	13
1.00	4	19	50	104	130	148	173	146	98
	9	50	95	193	216	191	125	82	28

Table :(G.4. 9)

ST.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.15	43	0	0	0	0	0	0	0	0
	43	0	0	0	0	0	0	0	0
0.18	411	9	0	0	0	0	0	0	0
	414	6	0	0	0	0	0	0	0
0.20	917	58	0	0	0	0	0	0	0
	923	51	0	0	0	0	0	0	0
0.25	2814	513	73	7	0	0	0	0	0
	2882	454	46	4	0	0	0	0	0
0.30	3772	1701	446	92	13	2	0	0	0
	4049	1512	314	44	6	0	0	0	0
0.35	3445	2711	1145	373	91	16	3	0	1
	3903	2625	867	211	36	4	0	0	0
0.40	2791	2901	1917	873	279	63	11	3	0
	3413	3003	1599	526	129	19	2	0	0
0.45	1889	2814	2364	1392	610	205	54	9	1
	2499	3125	2164	984	333	63	11	0	0
0.50	1336	2305	2502	1891	1007	420	144	33	8
	1928	2855	2526	1422	587	139	40	1	1
0.60	678	1517	2275	2207	1654	945	407	155	31
	1146	2304	2656	2003	1090	439	117	19	1

Table :(G.4.10)

ST.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.15	36	0	0	0	0	0	0	0	0
	36	0	0	0	0	0	0	0	0
0.18	355	10	0	0	0	0	0	0	0
	355	10	0	0	0	0	0	0	0
0.20	798	44	2	0	0	0	0	0	0
	802	38	2	0	0	0	0	0	0
0.25	2541	434	54	1	0	0	0	0	0
	2597	363	36	0	0	0	0	0	0
0.30	3719	1349	331	43	5	1	0	0	0
	3893	1206	228	29	1	0	0	0	0
0.35	3620	2349	905	254	67	7	0	1	0
	3960	2247	656	137	20	1	0	0	0
0.40	3162	2857	1551	636	164	41	8	1	0
	3673	2857	1218	358	58	14	2	0	0
0.45	2422	2910	2127	1061	394	113	22	3	0
	3010	3109	1792	688	166	24	3	0	0
0.50	1883	2706	2386	1454	669	229	61	17	2
	2499	3141	2179	961	281	71	15	1	1
0.60	1048	2095	2430	2073	1198	586	238	65	15
	1707	2748	2564	1526	667	192	52	11	1

Table :(6.4.11)

ST.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.15	41	0	0	0	0	0	0	0	0
	41	0	0	0	0	0	0	0	0
0.18	422	13	0	0	0	0	0	0	0
	423	10	0	0	0	0	0	0	0
0.20	966	64	3	0	0	0	0	0	0
	977	53	1	0	0	0	0	0	0
0.25	2803	555	78	2	1	0	0	0	0
	2914	412	38	1	0	0	0	0	0
0.30	3710	1743	513	90	8	1	2	0	0
	4073	1570	249	27	1	1	1	0	0
0.35	3477	2566	1220	430	119	27	5	2	0
	4147	2486	802	174	32	3	1	0	0
0.40	2737	2850	1923	912	329	94	13	5	1
	3597	3131	1395	433	86	11	0	0	0
0.45	1928	2664	2338	1429	691	248	66	17	4
	2751	3312	2036	753	260	40	6	0	0
0.50	1319	2270	2362	1869	1059	492	173	49	13
	2094	3086	2403	1249	397	113	23	3	1
0.60	683	1454	2122	2099	1649	1047	491	198	58
	1345	2495	2672	1812	899	267	73	13	3

Table :(6.4.12)

ST.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.15	54	0	0	0	0	0	0	0	0
	54	0	0	0	0	0	0	0	0
0.18	487	8	2	0	0	0	0	0	0
	488	7	1	0	0	0	0	0	0
0.20	1085	62	2	0	0	0	0	0	0
	1095	47	1	0	0	0	0	0	0
0.25	3001	683	105	6	0	0	0	0	0
	3159	538	44	0	0	0	0	0	0
0.30	3816	1913	588	117	24	3	0	1	0
	4324	1662	289	28	2	1	0	0	0
0.35	3211	2905	1423	546	126	30	5	1	0
	4116	2914	811	159	20	6	1	0	0
0.40	2341	2910	2213	1113	407	142	28	8	0
	3313	3508	1554	429	88	18	0	0	0
0.45	1508	2469	2537	1744	845	330	110	26	2
	2568	3543	2118	849	223	45	9	0	0
0.50	929	1967	2403	2180	1306	636	253	69	19
	1865	3315	2500	1285	452	105	19	0	0
0.60	346	1109	1835	2276	1961	1355	688	260	83
	1095	2608	2841	1927	923	294	66	10	1

Table :(6.4.13)

ST.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.15	38	0	0	0	0	0	0	0	0
	38	0	0	0	0	0	0	0	0
0.18	432	8	2	0	0	0	0	0	0
	435	6	1	0	0	0	0	0	0
0.20	972	53	1	0	0	0	0	0	0
	979	42	0	0	0	0	0	0	0
0.25	2825	545	73	4	0	0	0	0	0
	2958	403	26	0	0	0	0	0	0
0.30	3772	1677	436	81	19	1	1	0	0
	4215	1377	193	18	3	0	0	0	0
0.35	3512	2708	1174	364	80	10	4	1	0
	4396	2510	550	88	9	3	1	0	0
0.40	2763	2929	1929	847	267	81	15	2	0
	3823	3240	1099	280	42	3	0	0	0
0.45	1923	2758	2403	1381	594	209	59	13	0
	3147	3550	1630	519	117	23	4	0	0
0.50	1350	2268	2591	1850	985	395	145	32	7
	2588	3499	2047	841	231	42	7	1	0
0.60	606	1560	2204	2237	1717	966	389	153	34
	1712	3160	2579	1340	524	132	28	3	1

Table :(6.4.14)

ST.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.15	35	0	0	0	0	0	0	0	0
	35	0	0	0	0	0	0	0	0
0.18	355	9	0	0	0	0	0	0	0
	358	5	0	0	0	0	0	0	0
0.20	807	37	2	0	0	0	0	0	0
	812	30	0	0	0	0	0	0	0
0.25	2494	421	38	5	0	0	0	0	0
	2577	314	9	0	0	0	0	0	0
0.30	3637	1416	307	57	9	2	0	0	0
	3991	1095	122	8	0	0	0	0	0
0.35	3645	2361	861	238	57	13	0	1	0
	4361	1993	362	49	5	0	0	0	0
0.40	3203	2834	1557	560	185	40	5	2	0
	4260	2788	671	134	15	5	0	0	0
0.45	2525	2896	2051	1049	401	121	22	4	0
	3836	3214	1100	293	45	10	2	0	0
0.50	1939	2576	2389	1528	702	223	65	18	1
	3399	3409	1511	418	98	21	2	0	0
0.60	1073	2066	2411	2052	1254	589	215	55	18
	2638	3338	1863	766	254	65	12	0	0

Table :(6.4.15)

ST.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.15	43	0	0	0	0	0	0	0	0
	41	0	0	0	0	0	0	0	0
0.18	405	6	0	0	0	0	0	0	0
	385	5	0	0	0	0	0	0	0
0.20	941	55	1	0	0	0	0	0	0
	911	35	0	0	0	0	0	0	0
0.25	2708	589	64	9	0	0	0	0	0
	2730	411	9	0	0	0	0	0	0
0.30	3670	1597	433	77	9	3	0	0	0
	3963	1172	110	6	0	0	0	0	0
0.35	3502	2507	1207	361	80	20	2	0	0
	4393	2134	347	35	5	1	0	0	0
0.40	2732	2925	1850	844	270	71	12	1	1
	4050	2868	679	131	18	1	0	0	0
0.45	2069	2795	2250	1321	595	195	57	13	4
	3710	3206	1133	242	48	6	0	0	0
0.50	1459	2396	2479	1719	951	427	126	31	2
	3219	3451	1458	440	108	18	1	0	0
0.60	736	1566	2247	2193	1562	896	390	174	42
	2508	3242	1999	816	254	60	6	3	0

Table :(6.4.16)

ST.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.15	53	0	0	0	0	0	0	0	0
	48	0	0	0	0	0	0	0	0
0.18	457	9	0	0	0	0	0	0	0
	424	6	0	0	0	0	0	0	0
0.20	989	67	4	0	0	0	0	0	0
	918	43	0	0	0	0	0	0	0
0.25	2945	652	83	16	0	0	0	0	0
	2862	386	16	1	0	0	0	0	0
0.30	3726	1855	582	139	16	1	1	0	0
	4052	1204	124	9	0	0	0	0	0
0.35	3266	2773	1364	500	143	28	3	2	0
	4163	2271	356	29	2	1	0	0	0
0.40	2455	2827	2076	1069	413	119	33	9	1
	3997	2848	685	114	19	1	0	0	0
0.45	1697	2447	2381	1596	903	343	94	31	5
	3535	3180	1136	250	39	6	2	0	0
0.50	1093	2055	2356	2013	1248	608	275	68	15
	3103	3336	1561	413	92	21	1	1	0
0.60	522	1248	1920	2156	1774	1176	654	290	112
	2366	3229	2072	814	235	59	12	2	0

Tables:(6.5)

(S/N)db	AG.ERR.	CH.ERR.
10.97	.85E -2	.00E 0
9.03	.34E -1	.00E 0
7.45	.72E -1	.00E 0
6.11	.12E 0	.00E 0
4.95	.16E 0	.00E 0
3.93	.20E 0	.00E 0
3.01	.24E 0	.00E 0
1.43	.31E 0	.00E 0
-1.07	.40E 0	.10E -4
-3.01	.46E 0	.20E -3

1 CHANNEL

(S/N)db	AG.ERR.	CH.ERR.
10.97	.57E -2	.00E 0
9.03	.22E -1	.00E 0
7.45	.48E -1	.00E 0
6.11	.78E -1	.00E 0
4.95	.11E 0	.00E 0
3.93	.13E 0	.00E 0
3.01	.16E 0	.00E 0
1.43	.20E 0	.00E 0
-1.07	.27E 0	.10E -4
-3.01	.31E 0	.30E -3

2 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
10.97	.70E -2	.00E 0
9.03	.28E -1	.00E 0
7.45	.60E -1	.00E 0
6.11	.97E -1	.00E 0
4.95	.13E 0	.00E 0
3.93	.17E 0	.00E 0
3.01	.20E 0	.00E 0
1.43	.26E 0	.33E -4
-1.07	.34E 0	.11E -2
-3.01	.39E 0	.77E -2

3 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
10.97	.57E -2	.00E 0
9.03	.22E -1	.00E 0
7.45	.48E -1	.00E 0
6.11	.78E -1	.00E 0
4.95	.11E 0	.00E 0
3.93	.14E 0	.00E 0
3.01	.16E 0	.00E 0
1.43	.20E 0	.25E -4
-1.07	.27E 0	.11E -2
-3.01	.31E 0	.66E -2

4 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
10.97	.71E -2	.00E 0
9.03	.28E -1	.00E 0
7.45	.60E -1	.20E -4
6.11	.96E -1	.00E 0
4.95	.13E 0	.20E -4
3.93	.17E 0	.20E -4
3.01	.20E 0	.38E -3
1.43	.26E 0	.14E -2
-1.07	.34E 0	.11E -1
-3.01	.39E 0	.36E -1

5 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
13.98	.56E -2	.00E 0
12.04	.22E -1	.00E 0
10.46	.48E -1	.00E 0
9.12	.78E -1	.00E 0
7.96	.11E 0	.00E 0
6.94	.14E 0	.33E -4
6.02	.16E 0	.12E -3
4.44	.20E 0	.42E -3
-1.07	.27E 0	.62E -2
-3.01	.31E 0	.28E -1

6 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
10.97	.64E -2	.00E 0
9.03	.25E -1	.00E 0
7.45	.54E -1	.00E 0
6.11	.87E -1	.29E -4
4.95	.12E 0	.71E -4
3.93	.15E 0	.27E -3
3.01	.18E 0	.64E -3
1.43	.23E 0	.21E -2
-1.07	.30E 0	.16E -1
-3.01	.35E 0	.43E -1

7 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
10.97	.56E -2	.00E 0
9.03	.22E -1	.00E 0
7.45	.48E -1	.00E 0
6.11	.78E -1	.13E -4
4.95	.11E 0	.75E -4
3.93	.14E 0	.26E -3
3.01	.16E 0	.42E -3
1.43	.20E 0	.23E -2
-1.07	.27E 0	.16E -1
-3.01	.31E 0	.43E -1

8 CHANNELS

Tables:(6.5)

(S/N)db	AG.ERR.	CH.ERR.
10.97	.76E -2	.00E 0
9.03	.31E -1	.11E -4
7.45	.66E -1	.56E -4
6.11	.11E 0	.66E -3
4.95	.15E 0	.21E -2
3.93	.19E 0	.50E -2
3.01	.22E 0	.91E -2
1.43	.28E 0	.23E -1
-1.07	.37E 0	.60E -1
-3.01	.43E 0	.11E 0

9 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
10.97	.57E -2	.00E 0
9.03	.22E -1	.00E 0
7.45	.49E -1	.00E 0
6.11	.78E -1	.50E -4
4.95	.11E 0	.10E -3
3.93	.13E 0	.52E -3
3.01	.16E 0	.14E -2
1.43	.21E 0	.49E -2
-1.07	.27E 0	.28E -1
-3.01	.31E 0	.71E -1

10 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
10.97	.64E -2	.00E 0
9.03	.26E -1	.36E -4
7.45	.56E -1	.53E -3
6.11	.90E -1	.24E -2
4.95	.12E 0	.64E -2
3.93	.16E 0	.11E -1
3.01	.19E 0	.16E -1
1.43	.24E 0	.30E -1
-1.07	.31E 0	.63E -1
-3.01	.36E 0	.11E 0

11 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
10.97	.76E -2	.00E 0
9.03	.29E -1	.50E -4
7.45	.63E -1	.60E -3
6.11	.10E 0	.35E -2
4.95	.14E 0	.94E -2
3.93	.18E 0	.17E -1
3.01	.21E 0	.27E -1
1.43	.27E 0	.48E -1
-1.07	.35E 0	.97E -1
-3.01	.41E 0	.15E 0

12 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
10.97	.64E -2	.00E 0
9.03	.26E -1	.31E -4
7.45	.56E -1	.51E -3
6.11	.89E -1	.24E -2
4.95	.12E 0	.68E -2
3.93	.16E 0	.13E -1
3.01	.19E 0	.21E -1
1.43	.24E 0	.40E -1
-1.07	.31E 0	.91E -1
-3.01	.36E 0	.14E 0

13 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
10.97	.56E -2	.00E 0
9.03	.22E -1	.29E -4
7.45	.48E -1	.31E -3
6.11	.77E -1	.15E -2
4.95	.11E 0	.47E -2
3.93	.13E 0	.98E -2
3.01	.16E 0	.16E -1
1.43	.20E 0	.33E -1
-1.07	.27E 0	.84E -1
-3.01	.31E 0	.14E 0

14 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
10.97	.65E -2	.47E -4
9.03	.25E -1	.82E -3
7.45	.54E -1	.33E -2
6.11	.88E -1	.69E -2
4.95	.12E 0	.13E -1
3.93	.15E 0	.21E -1
3.01	.18E 0	.31E -1
1.43	.23E 0	.53E -1
-1.07	.30E 0	.11E 0
-3.01	.35E 0	.16E 0

15 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
10.97	.71E -2	.87E -4
9.03	.28E -1	.14E -2
7.45	.61E -1	.48E -2
6.11	.99E -1	.12E -1
4.95	.14E 0	.20E -1
3.93	.17E 0	.31E -1
3.01	.20E 0	.42E -1
1.43	.26E 0	.68E -1
-1.07	.34E 0	.12E 0
-3.01	.39E 0	.18E 0

16 CHANNELS

Tables:(6.5)

(S/N)db	AG.ERR.	CH.ERR.
13.47	.32E -3	.00E 0
11.88	.27E -2	.29E -4
10.97	.65E -2	.15E -3
9.03	.25E -1	.22E -2
7.45	.55E -1	.83E -2
6.11	.88E -1	.18E -1
4.95	.12E 0	.28E -1
3.93	.15E 0	.39E -1
3.01	.18E 0	.49E -1
1.43	.23E 0	.70E -1

17 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
13.47	.27E -3	.00E 0
11.88	.23E -2	.61E -4
10.97	.58E -2	.17E -3
9.03	.22E -1	.28E -2
7.45	.48E -1	.11E -1
6.11	.70E -1	.22E -1
4.95	.11E 0	.33E -1
3.93	.13E 0	.44E -1
3.01	.16E 0	.53E -1
1.43	.21E 0	.74E -1

18 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
13.47	.30E -3	.00E 0
11.88	.26E -2	.21E -4
10.97	.64E -2	.18E -3
9.03	.25E -1	.24E -2
7.45	.54E -1	.92E -2
6.11	.87E -1	.21E -1
4.95	.12E 0	.34E -1
3.93	.15E 0	.47E -1
3.01	.18E 0	.59E -1
1.43	.23E 0	.83E -1

19 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
13.47	.35E -3	.00E 0
11.88	.29E -2	.20E -4
10.97	.71E -2	.80E -4
9.03	.27E -1	.18E -2
7.45	.60E -1	.89E -2
6.11	.96E -1	.20E -1
4.95	.13E 0	.34E -1
3.93	.17E 0	.50E -1
3.01	.20E 0	.64E -1
1.43	.25E 0	.91E -1

20 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
13.47	.32E -3	.00E 0
11.88	.29E -2	.43E -4
10.97	.71E -2	.19E -3
9.03	.28E -1	.19E -2
7.45	.59E -1	.85E -2
6.11	.97E -1	.20E -1
4.95	.13E 0	.33E -1
3.93	.17E 0	.49E -1
3.01	.20E 0	.63E -1
1.43	.25E 0	.94E -1

21 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
13.47	.33E -3	.90E -5
11.88	.29E -2	.35E -3
10.97	.73E -2	.64E -3
9.03	.29E -1	.39E -2
7.45	.63E -1	.13E -1
6.11	.10E 0	.27E -1
4.95	.14E 0	.44E -1
3.93	.16E 0	.60E -1
3.01	.19E 0	.80E -1
1.43	.24E 0	.11E 0

22 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
13.47	.36E -3	.70E -4
11.88	.32E -2	.39E -3
10.97	.79E -2	.14E -2
9.03	.31E -1	.58E -2
7.45	.66E -1	.18E -1
6.11	.11E 0	.35E -1
4.95	.15E 0	.53E -1
3.93	.19E 0	.73E -1
3.01	.22E 0	.91E -1
1.43	.28E 0	.12E 0

23 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
13.47	.34E -3	.00E 0
11.88	.30E -2	.54E -4
10.97	.75E -2	.20E -3
9.03	.29E -1	.27E -2
7.45	.63E -1	.12E -1
6.11	.10E 0	.27E -1
4.95	.14E 0	.45E -1
3.93	.18E 0	.63E -1
3.01	.21E 0	.81E -1
1.43	.27E 0	.11E 0

24 CHANNELS

Tables:(6.5)

(S/N)db	AG.ERR.	CH.ERR.
13.47	.28E -3	.00E 0
11.88	.28E -2	.32E -4
10.97	.70E -2	.23E -3
9.03	.27E -1	.26E -2
7.45	.59E -1	.11E -1
6.11	.96E -1	.25E -1
4.95	.13E 0	.41E -1
3.93	.17E 0	.58E -1
3.01	.20E 0	.76E -1
1.43	.25E 0	.11E 0

25 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
13.47	.26E -3	.00E 0
11.88	.27E -2	.54E -4
10.97	.64E -2	.16E -3
9.03	.26E -1	.25E -2
7.45	.55E -1	.98E -2
6.11	.90E -1	.23E -1
4.95	.12E 0	.38E -1
3.93	.16E 0	.55E -1
3.01	.19E 0	.71E -1
1.43	.24E 0	.10E 0

26 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
13.47	.25E -3	.00E 0
11.88	.25E -2	.48E -4
10.97	.59E -2	.14E -3
9.03	.24E -1	.22E -2
7.45	.52E -1	.89E -2
6.11	.83E -1	.22E -1
4.95	.12E 0	.36E -1
3.93	.15E 0	.52E -1
3.01	.17E 0	.66E -1
1.43	.22E 0	.99E -1

27 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
13.47	.26E -3	.00E 0
11.88	.21E -2	.43E -4
10.97	.55E -2	.11E -3
9.03	.22E -1	.21E -2
7.45	.48E -1	.83E -2
6.11	.77E -1	.20E -1
4.95	.11E 0	.34E -1
3.93	.14E 0	.49E -1
3.01	.16E 0	.63E -1
1.43	.20E 0	.94E -1

28 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
13.47	.28E -3	.70E -5
11.88	.25E -2	.20E -3
10.97	.61E -2	.60E -3
9.03	.23E -1	.35E -2
7.45	.51E -1	.12E -1
6.11	.83E -1	.26E -1
4.95	.11E 0	.41E -1
3.93	.14E 0	.58E -1
3.01	.17E 0	.74E -1
1.43	.22E 0	.11E 0

29 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
13.47	.30E -3	.67E -4
11.88	.26E -2	.34E -3
10.97	.64E -2	.10E -2
9.03	.25E -1	.52E -2
7.45	.54E -1	.15E -1
6.11	.88E -1	.31E -1
4.95	.12E 0	.48E -1
3.93	.15E 0	.67E -1
3.01	.18E 0	.84E -1
1.43	.23E 0	.12E 0

30 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
13.47	.32E -3	.45E -4
11.88	.27E -2	.47E -3
10.97	.68E -2	.14E -2
9.03	.26E -1	.66E -2
7.45	.58E -1	.18E -1
6.11	.94E -1	.37E -1
4.95	.13E 0	.56E -1
3.93	.16E 0	.75E -1
3.01	.19E 0	.94E -1
1.43	.25E 0	.13E 0

31 CHANNELS

(S/N)db	AG.ERR.	CH.ERR.
13.47	.32E -3	.75E -4
11.88	.29E -2	.54E -3
10.97	.72E -2	.17E -2
9.03	.28E -1	.80E -2
7.45	.61E -1	.21E -1
6.11	.99E -1	.41E -1
4.95	.14E 0	.62E -1
3.93	.17E 0	.83E -1
3.01	.20E 0	.10E 0
1.43	.26E 0	.14E 0

32 CHANNELS

Table :(6.G.1)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.20	2083	202	27	2	0	0	0	0	0
	2083	202	27	2	0	0	0	0	0
0.25	3710	2063	677	169	32	4	0	0	0
	3710	2063	677	169	32	4	0	0	0
0.30	2268	2723	2178	1193	504	182	41	18	2
	2268	2723	2178	1193	504	182	41	18	2
0.35	730	1630	2248	2081	1488	962	414	179	50
	730	1630	2248	2081	1488	962	414	179	50
0.40	183	664	1254	1818	1958	1687	1184	663	330
	183	664	1254	1818	1958	1687	1184	663	330
0.45	43	206	581	1113	1493	1756	1677	1313	882
	43	206	581	1113	1493	1756	1677	1313	882
0.50	15	63	221	549	886	1400	1666	1665	1330
	15	63	221	549	886	1400	1666	1665	1330
0.60	0	9	35	107	276	545	862	1306	1455
	0	9	35	107	276	545	862	1306	1455
0.80	0	0	4	4	13	51	127	300	578
	0	0	4	4	13	51	127	300	578
1.00	0	0	0	0	1	11	15	48	127
	0	0	0	0	1	11	15	48	127

Table :(6.G.2)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.20	1486	154	7	0	0	0	0	0	0
	1486	154	7	0	0	0	0	0	0
0.25	3523	1279	291	29	12	0	0	0	0
	3523	1279	291	29	12	0	0	0	0
0.30	3377	2641	1315	464	122	35	2	2	1
	3377	2641	1315	464	122	35	2	2	1
0.35	2069	2592	2257	1379	618	247	76	23	6
	2069	2592	2257	1379	618	247	76	23	6
0.40	1006	1980	2272	2003	1303	730	318	104	33
	1006	1980	2272	2003	1303	730	318	104	33
0.45	485	1226	1755	2039	1785	1324	732	356	135
	485	1226	1755	2039	1785	1324	732	356	135
0.50	217	683	1267	1814	1873	1646	1152	685	390
	217	683	1267	1814	1873	1646	1152	685	390
0.60	54	245	546	1065	1492	1703	1576	1320	918
	54	245	546	1065	1492	1703	1576	1320	918
0.80	1	37	112	300	617	1091	1291	1536	1520
	1	37	112	300	617	1091	1291	1536	1520
1.00	0	6	29	115	283	614	844	1232	1381
	0	6	29	115	283	614	844	1232	1381

Table:(6.6.1.a)

SD.	10ER	11ER	12ER	13ER	14ER	15ER	16ER	17ER	18ER
0.20	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
0.25	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
0.30	1	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0
0.35	18	5	0	0	0	0	0	0	0
	18	5	0	0	0	0	0	0	0
0.40	147	56	18	3	1	0	0	0	0
	147	56	18	3	1	0	0	0	0
0.45	517	252	109	40	7	5	2	0	0
	517	252	109	40	7	5	2	0	0
0.50	957	648	345	155	67	16	13	1	1
	957	648	315	155	67	16	13	1	1
0.60	1530	1353	1054	676	439	211	88	36	13
	1530	1353	1051	676	439	211	88	36	13
0.80	827	1156	1470	1477	1303	1071	719	468	252
	827	1156	1470	1477	1303	1071	719	468	252
1.00	315	537	874	1164	1368	1453	1333	1086	798
	315	537	874	1164	1368	1453	1333	1086	798

Table:(6.6.2.a)

SD.	10ER	11ER	12ER	13ER	14ER	15ER	16ER	17ER	18ER
0.20	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
0.25	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
0.30	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
0.35	1	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0
0.40	6	3	1	0	0	0	0	0	0
	6	3	1	0	0	0	0	0	0
0.45	44	16	8	1	0	0	0	0	0
	44	16	8	1	0	0	0	0	0
0.50	151	54	19	6	1	1	0	0	0
	151	54	19	6	1	1	0	0	0
0.60	582	288	122	65	12	7	0	0	0
	582	288	122	65	12	7	0	0	0
0.80	1287	948	604	365	170	70	30	15	3
	1287	948	604	365	170	70	30	15	3
1.00	1526	1332	1087	704	439	206	118	56	20
	1526	1332	1087	705	439	205	119	57	18

Table :(6.6.3)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.20	2027	296	26	0	1	0	0	0	0
	2027	296	26	0	1	0	0	0	0
0.25	3696	2033	682	150	32	7	0	0	0
	3696	2033	682	150	32	7	0	0	0
0.30	2282	2761	2097	1267	489	151	52	11	2
	2282	2761	2097	1267	489	151	52	11	2
0.35	725	1640	2221	2064	1592	891	418	170	56
	725	1640	2221	2064	1592	891	418	170	56
0.40	208	640	1291	1825	1974	1698	1122	670	320
	208	640	1291	1825	1974	1690	1122	670	320
0.45	48	223	594	1021	1549	1716	1728	1319	900
	48	223	594	1021	1549	1716	1728	1319	908
0.50	10	68	240	519	918	1353	1688	1660	1324
	10	68	240	519	918	1353	1688	1660	1324
0.60	3	4	32	111	278	511	941	1300	1465
	3	4	32	111	278	511	941	1300	1465
0.80	0	0	1	4	11	44	123	279	563
	0	0	1	4	11	44	123	279	563
1.00	0	0	0	0	2	7	21	48	122
	0	0	0	0	2	7	21	48	123

Table (6.6.4)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.20	2027	282	24	0	0	0	0	0	0
	2027	282	24	0	0	0	0	0	0
0.25	3666	2007	727	140	33	3	0	1	0
	3666	2007	727	140	33	3	0	1	0
0.30	2289	2762	2139	1211	514	159	35	20	1
	2289	2762	2139	1211	514	159	35	20	1
0.35	768	1678	2168	2094	1545	894	416	176	60
	768	1678	2168	2094	1545	894	416	176	60
0.40	226	653	1279	1744	2006	1703	1135	664	327
	226	653	1279	1744	2006	1703	1135	664	328
0.45	52	235	571	1024	1479	1772	1721	1348	859
	52	235	571	1024	1479	1772	1721	1349	859
0.50	10	73	230	519	955	1386	1659	1593	1393
	10	73	230	519	955	1386	1660	1593	1398
0.60	1	3	38	101	267	529	854	1239	1480
	1	3	38	101	267	529	854	1242	1485
0.80	0	1	0	3	14	54	135	205	533
	0	1	0	3	14	54	135	206	541
1.00	0	0	0	0	1	6	10	54	132
	0	0	0	0	1	6	10	54	139

Table:(G.G.3.a)

SD.	10ER	11ER	12ER	13ER	14ER	15ER	16ER	17ER	18ER
0.20	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
0.25	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
0.30	1	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0
0.35	18	2	0	0	0	0	0	0	0
	18	2	0	0	0	0	0	0	0
0.40	148	55	20	4	1	0	0	0	0
	148	55	20	4	1	0	0	0	0
0.45	474	247	118	36	11	5	2	0	0
	474	248	118	35	11	5	2	0	0
0.50	1015	619	311	155	80	25	13	0	0
	1015	619	312	156	79	24	13	0	0
0.60	1490	1379	1053	681	419	207	72	31	20
	1492	1379	1052	680	420	206	74	29	21
0.80	874	1159	1409	1487	1362	1128	717	450	205
	876	1166	1417	1489	1376	1134	704	445	206
1.00	294	522	857	1192	1363	1520	1356	1066	735
	297	530	872	1229	1380	1564	1371	1070	708

Table:(G.G.4.a)

SD.	10ER	11ER	12ER	13ER	14ER	15ER	16ER	17ER	18ER
0.20	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
0.25	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
0.30	1	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0
0.35	11	2	3	0	0	0	0	0	0
	11	2	3	0	0	0	0	0	0
0.40	151	58	20	4	2	0	0	0	0
	150	58	20	4	2	0	0	0	0
0.45	529	233	116	39	14	1	0	0	0
	531	233	115	38	13	1	0	0	0
0.50	985	595	343	166	59	21	8	1	2
	983	598	340	167	55	21	7	1	2
0.60	1558	1407	1023	723	420	215	90	33	10
	1570	1404	1032	715	420	207	82	35	9
0.80	873	1167	1416	1491	1373	1043	757	441	227
	879	1189	1454	1502	1410	1064	708	401	215
1.00	292	530	846	1198	1432	1501	1318	1054	736
	304	558	882	1267	1536	1521	1354	1019	692

Table :(6.6.6)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.20	1483	136	14	0	0	0	0	0	0
	1483	136	14	0	0	0	0	0	0
0.25	3525	1257	264	44	4	0	0	0	0
	3525	1257	264	44	4	0	0	0	0
0.30	3367	2626	1302	495	133	28	7	1	0
	3367	2626	1302	495	133	28	7	1	0
0.35	1986	2707	2258	1304	691	221	62	24	1
	1986	2707	2258	1304	691	221	62	24	1
0.40	1039	1936	2266	1986	1336	708	307	121	39
	1039	1936	2266	1986	1336	708	307	121	39
0.45	441	1173	1803	2052	1867	1269	703	375	147
	441	1173	1803	2052	1867	1269	703	376	147
0.50	247	695	1295	1751	1914	1611	1134	731	350
	247	695	1295	1751	1914	1612	1133	732	350
0.60	65	196	590	1037	1490	1736	1618	1362	951
	65	196	590	1037	1490	1736	1620	1365	948
0.80	6	26	116	311	637	1019	1375	1575	1533
	6	26	116	311	638	1026	1382	1602	1545
1.00	1	8	35	105	249	538	891	1191	1473
	1	8	35	106	250	550	923	1233	1596

Table :(6.6.5)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.20	1783	219	18	1	0	0	0	0	0
	1783	219	18	1	0	0	0	0	0
0.25	3729	1673	443	100	17	0	0	0	0
	3729	1673	443	100	17	0	0	0	0
0.30	2808	2766	1800	864	266	80	18	5	1
	2808	2766	1800	864	266	80	18	6	0
0.35	1299	2275	2402	1749	1100	548	181	58	13
	1299	2275	2402	1749	1100	548	181	58	13
0.40	482	1172	1843	2142	1835	1243	656	330	144
	482	1172	1843	2142	1835	1243	656	331	143
0.45	161	568	1139	1616	1959	1664	1369	799	409
	161	568	1139	1616	1959	1664	1369	799	409
0.50	56	222	579	1095	1558	1774	1641	1253	888
	56	222	579	1095	1558	1774	1642	1254	893
0.60	8	52	174	380	734	1162	1587	1600	1457
	8	52	174	380	734	1162	1587	1605	1466
0.80	0	1	17	47	127	302	549	923	1280
	0	1	17	47	127	302	553	930	1296
1.00	0	0	2	5	16	84	200	375	718
	0	0	2	5	16	84	203	387	756

Table:(6.6.5.a)

SD.	10ER	11ER	12ER	13ER	14ER	15ER	16ER	17ER	18ER
0.20	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
0.25	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
0.30	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
0.35	4	0	1	0	0	0	0	0	0
	4	0	1	0	0	0	0	0	0
0.40	52	14	2	1	0	0	0	0	0
	52	14	2	1	0	0	0	0	0
0.45	197	74	25	5	3	0	0	0	0
	198	73	25	5	3	0	0	0	0
0.50	518	235	112	40	13	7	2	0	0
	513	237	109	39	14	7	1	0	0
0.60	1118	841	440	253	125	49	16	2	1
	1118	833	442	253	127	41	15	1	1
0.80	1469	1485	1282	1054	719	420	183	88	29
	1486	1510	1314	1047	715	376	164	69	30
1.00	1031	1326	1410	1477	1175	941	610	320	205
	1083	1395	1485	1531	1152	907	530	275	129

Table:(6.6.6.a)

SD.	10ER	11ER	12ER	13ER	14ER	15ER	16ER	17ER	18ER
0.20	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
0.25	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
0.30	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
0.35	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
0.40	10	4	1	0	9	0	0	0	0
	10	4	1	0	0	0	0	0	0
0.45	44	15	7	1	1	0	0	0	0
	43	15	7	1	1	0	0	0	0
0.50	139	59	22	9	1	0	0	0	0
	139	61	19	9	1	0	0	0	0
0.60	486	266	132	40	18	3	2	0	0
	487	268	131	38	17	3	1	0	0
0.80	1230	932	615	362	154	72	22	10	3
	1237	926	619	340	138	62	17	7	0
1.00	1506	1387	1031	707	427	241	139	54	12
	1580	1402	1006	626	365	201	78	36	3

Table :(G.G.7)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.20	1710	187	18	0	0	0	0	0	0
	1710	187	18	0	0	0	0	0	0
0.25	3597	1576	419	71	13	2	1	0	0
	3597	1576	422	68	14	1	1	0	0
0.30	2866	2769	1630	757	247	76	23	6	2
	2866	2769	1648	748	254	66	20	3	2
0.35	1527	2337	2246	1661	971	485	203	55	14
	1527	2338	2275	1684	972	462	191	38	14
0.40	675	1413	1923	2012	1609	1106	625	290	133
	675	1414	1953	2067	1621	1106	591	270	108
0.45	293	760	1292	1635	1058	1538	1171	733	394
	293	762	1320	1682	1920	1588	1138	681	351
0.50	125	418	843	1257	1480	1674	1431	1214	698
	125	418	851	1307	1552	1748	1483	1149	660
0.60	24	122	328	601	955	1273	1457	1465	1213
	24	122	331	624	997	1356	1555	1519	1226
0.80	3	13	62	182	323	582	832	1037	1260
	3	13	63	185	342	609	900	1138	1379
1.00	1	2	8	56	138	289	472	666	860
	1	2	8	59	140	308	516	772	1005

Table :(G.G.8)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.20	1927	265	28	0	0	0	0	0	0
	1927	265	28	0	0	0	0	0	0
0.25	3671	1816	596	124	23	2	1	0	0
	3671	1816	600	124	19	2	1	0	0
0.30	2518	2713	1947	1010	428	127	55	9	0
	2518	2714	1971	1011	419	122	46	6	0
0.35	1110	1915	2155	1882	1307	695	352	167	58
	1110	1915	2203	1947	1297	662	337	139	39
0.40	435	981	1582	1853	1711	1381	936	523	262
	435	981	1635	1916	1784	1427	880	454	229
0.45	189	503	903	1351	1561	1601	1361	1005	704
	189	503	926	1431	1660	1700	1382	957	620
0.50	81	210	470	881	1230	1426	1496	1325	1051
	81	211	494	944	1305	1572	1567	1373	998
0.60	13	74	170	359	640	852	1158	1237	1300
	13	74	172	385	691	954	1309	1367	1406
0.80	5	9	25	76	185	353	493	657	896
	5	9	27	79	199	389	559	795	1058
1.00	0	3	9	27	72	155	230	374	505
	0	3	9	27	74	170	264	467	630

Table:(6.6.7.a)

SD.	10ER	11ER	12ER	13ER	14ER	15ER	16ER	17ER	18ER
0.20	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
0.25	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
0.30	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
0.35	5	1	1	0	0	0	0	0	0
	3	1	1	0	0	0	0	0	0
0.40	40	13	6	0	0	0	0	0	0
	25	12	3	0	0	0	0	0	0
0.45	158	81	23	9	4	0	0	0	0
	129	60	18	6	1	0	0	0	0
0.50	417	249	123	30	14	7	1	0	0
	370	201	82	20	13	4	1	0	0
0.60	1007	659	433	257	116	51	23	13	0
	960	612	344	207	72	31	16	1	0
0.80	1241	1181	1008	861	575	390	232	122	57
	1366	1244	1017	745	453	305	139	60	30
1.00	1103	1139	1182	1087	1008	708	555	375	174
	1271	1343	1302	1134	872	591	354	175	92

Table:(6.6.8.a)

SD.	10ER	11ER	12ER	13ER	14ER	15ER	16ER	17ER	18ER
0.20	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
0.25	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
0.30	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
0.35	17	5	3	0	0	0	0	0	0
	13	3	1	0	0	0	0	0	0
0.40	145	56	24	5	2	1	0	0	0
	108	30	14	4	0	0	0	0	0
0.45	429	185	113	44	13	7	1	1	0
	350	150	67	29	4	3	0	0	0
0.50	776	483	285	143	66	36	17	9	1
	674	413	202	88	35	21	9	0	0
0.60	1226	944	699	543	368	237	99	46	21
	1221	886	659	423	247	120	43	19	7
0.80	998	1147	1110	1062	824	741	526	106	201
	1189	1270	1269	976	784	632	364	190	121
1.00	640	892	951	1018	1037	929	861	720	556
	908	1066	1176	1206	1073	963	707	556	327

Table :(G.G. 9)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.20	1674	187	15	0	0	0	0	0	0
	1674	187	15	0	0	0	0	0	0
0.25	3670	1483	416	83	10	2	0	0	0
	3670	1483	421	78	11	1	0	0	0
0.30	3013	2731	1618	680	255	64	10	0	0
	3013	2734	1631	683	245	57	8	0	0
0.35	1553	2427	2274	1672	963	431	141	44	13
	1553	2430	2322	1685	951	413	124	29	13
0.40	625	1438	2115	2156	1622	1009	554	237	80
	625	1442	2165	2220	1660	983	500	193	61
0.45	260	751	1423	1805	1880	1612	1104	638	287
	260	754	1479	1910	1960	1592	1045	575	237
0.50	116	378	796	1341	1780	1803	1493	1084	627
	116	380	840	1457	1882	1866	1490	1002	531
0.60	16	101	274	561	983	1459	1654	1574	1255
	16	102	299	635	1138	1622	1786	1553	1192
0.80	0	7	31	110	244	497	884	1211	1476
	1	8	40	156	324	679	1103	1456	1667
1.00	1	0	7	22	62	206	368	648	1000
	1	1	9	40	121	320	604	983	1389

Table :(G.G.10)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.20	1510	150	9	0	0	0	0	0	0
	1510	150	9	0	0	0	0	0	0
0.25	3523	1261	309	42	4	0	0	0	0
	3523	1261	309	42	4	0	0	0	0
0.30	3297	2696	1324	491	142	30	4	1	0
	3297	2696	1324	491	142	30	4	1	0
0.35	1959	2700	2260	1346	632	239	75	22	5
	1959	2700	2260	1346	634	239	74	21	5
0.40	964	1841	2365	2032	1314	732	318	122	28
	964	1841	2365	2033	1316	733	315	122	27
0.45	525	1156	1835	2041	1798	1243	763	334	146
	525	1156	1835	2041	1805	1250	757	337	142
0.50	242	644	1323	1863	1838	1629	1128	683	381
	242	644	1323	1864	1843	1641	1143	668	364
0.60	53	215	581	1014	1457	1742	1678	1322	882
	53	215	581	1019	1470	1770	1724	1324	879
0.80	6	34	121	283	606	977	1449	1551	1519
	6	34	122	295	643	1073	1597	1687	1625
1.00	0	7	45	95	267	541	893	1161	1490
	0	7	49	116	325	700	1164	1504	1759

Table :(6.6.11)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.20	1664	175	7	0	0	0	0	0	0
	1664	175	7	0	0	0	0	0	0
0.25	3607	1502	400	66	17	2	0	1	0
	3607	1503	401	64	18	1	0	1	0
0.30	2979	2742	1646	683	233	61	13	3	0
	2979	2745	1669	679	215	59	11	3	1
0.35	1571	2300	2315	1735	902	425	164	44	14
	1571	2306	2358	1768	887	403	130	39	8
0.40	715	1431	2025	2048	1583	1068	537	273	102
	715	1434	2093	2102	1634	1027	514	223	53
0.45	307	815	1392	1830	1737	1572	1096	650	337
	307	816	1434	1932	1831	1610	1058	572	264
0.50	128	396	891	1357	1680	1617	1475	1108	643
	128	397	918	1440	1800	1751	1440	1051	568
0.60	34	124	326	689	1034	1339	1539	1471	1225
	34	126	339	729	1141	1496	1700	1534	1250
0.80	5	20	56	164	377	595	904	1095	1280
	5	20	59	189	420	716	1125	1399	1554
1.00	0	3	11	54	138	277	513	775	970
	0	3	12	66	191	422	760	1101	1392

Table :(6.6.12)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.20	1897	231	19	1	1	0	0	0	0
	1897	231	19	1	1	0	0	0	0
0.25	3704	1815	510	92	16	3	0	0	0
	3704	1815	513	93	12	3	0	0	0
0.30	2646	2824	1911	916	333	103	31	6	2
	2646	2828	1930	914	326	98	23	5	2
0.35	1114	2093	2321	1973	1184	613	253	102	26
	1114	2096	2384	2016	1189	569	228	71	14
0.40	400	1021	1701	2080	1910	1410	750	308	161
	400	1023	1754	2207	2002	1393	669	310	113
0.45	153	458	933	1529	1823	1809	1418	932	515
	153	460	972	1658	2022	1917	1369	772	417
0.50	45	179	474	888	1400	1754	1638	1433	1017
	45	181	504	1015	1614	1951	1706	1388	829
0.60	8	32	110	293	575	1000	1396	1618	1558
	8	33	132	355	731	1281	1704	1795	1531
0.80	1	2	10	31	70	222	470	791	1119
	1	4	11	41	131	442	782	1232	1592
1.00	0	0	2	3	13	59	122	269	500
	0	0	2	11	40	136	321	712	1120

Table :(G.6.13)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9FR
0.20	1663	175	16	0	0	0	0	0	0
	1663	175	16	0	0	0	0	0	0
0.25	3657	1537	370	83	9	2	0	0	0
	3657	1537	373	81	8	2	0	0	0
0.30	3023	2780	1612	683	231	52	18	3	0
	3023	2782	1630	685	224	43	14	1	0
0.35	1580	2374	2362	1641	907	420	151	40	12
	1581	2375	2410	1685	901	370	130	32	6
0.40	621	1491	2093	2112	1618	1078	482	229	97
	621	1498	2179	2207	1665	1023	397	180	66
0.45	240	739	1428	1835	1946	1605	1105	573	293
	241	744	1495	1979	2076	1673	957	495	216
0.50	99	355	819	1355	1789	1765	1497	1025	678
	100	361	874	1506	1984	1931	1461	912	521
0.60	16	89	266	619	978	1480	1665	1499	1307
	18	92	300	762	1218	1767	1848	1571	1146
0.80	0	6	27	99	261	506	883	1268	1391
	0	8	51	174	480	868	1462	1702	1664
1.00	0	1	2	20	67	203	394	668	1032
	1	5	18	67	244	550	934	1385	1674

Table :(G.6.14)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.20	1515	119	7	1	0	0	0	0	0
	1515	119	7	1	0	0	0	0	0
0.25	3609	1201	303	47	3	0	0	0	0
	3609	1202	305	44	3	0	0	0	0
0.30	3392	2559	1319	518	137	31	3	2	0
	3392	2561	1335	517	126	26	2	2	0
0.35	2035	2724	2190	1368	620	238	57	19	6
	2035	2727	2236	1391	596	216	41	12	3
0.40	1005	2008	2266	1964	1281	716	332	119	24
	1005	2014	2344	2041	1314	642	272	76	14
0.45	510	1126	1807	2146	1710	1268	767	337	152
	511	1137	1885	2280	1836	1264	602	262	82
0.50	240	662	1269	1766	1915	1636	1194	702	344
	241	674	1356	1973	2070	1707	1058	559	223
0.60	53	219	573	1045	1489	1746	1619	1281	937
	55	239	645	1285	1750	2036	1631	1193	649
0.80	7	25	114	282	610	1060	1371	1529	1549
	12	47	204	524	1138	1633	1756	1741	1293
1.00	0	10	36	129	255	564	832	1238	1405
	1	30	147	393	801	1329	1632	1741	1517

Table :(6.6.15)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.20	1635	208	12	0	0	0	0	0	0
	1636	207	10	0	0	0	0	0	0
0.25	3651	1498	363	60	8	2	0	0	0
	3692	1472	341	55	6	1	0	0	0
0.30	3094	2688	1563	694	204	54	15	2	2
	3174	2682	1529	660	172	41	8	1	0
0.35	1670	2384	2256	1680	896	406	135	43	15
	1737	2456	2334	1686	817	326	89	20	5
0.40	703	1438	2081	2054	1586	988	537	230	85
	827	1501	2238	2145	1626	906	377	144	45
0.45	331	840	1396	1856	1858	1460	1090	603	299
	354	882	1574	2111	2015	1477	889	396	171
0.50	151	451	890	1407	1790	1632	1367	1064	613
	170	475	1027	1697	2055	1856	1301	770	378
0.60	27	127	353	683	1123	1421	1587	1434	1170
	33	146	445	954	1477	1879	1737	1422	920
0.80	1	18	58	190	376	689	976	1263	1309
	4	36	135	369	726	1208	1591	1686	1522
1.00	0	5	16	55	138	299	504	787	1085
	5	24	61	220	514	894	1276	1483	1633

Table :(6.6.16)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.20	1806	207	17	0	0	0	0	0	0
	1812	197	17	0	0	0	0	0	0
0.25	3674	1678	468	104	25	3	1	0	0
	3732	1621	446	91	17	4	0	0	0
0.30	2754	2731	1766	872	364	94	25	5	0
	2860	2776	1762	802	286	70	12	0	0
0.35	1333	2142	2274	1767	1120	586	271	97	18
	1432	2256	2398	1803	1014	467	162	37	7
0.40	516	1189	1775	1948	1742	1277	790	369	179
	568	1301	1986	2163	1819	1147	575	205	81
0.45	212	606	1128	1544	1751	1589	1271	870	499
	238	676	1336	1870	1976	1706	1158	595	264
0.50	92	321	604	1082	1464	1603	1519	1181	936
	102	362	767	1331	1893	1911	1505	1096	578
0.60	16	80	216	450	760	1079	1345	1509	1332
	19	97	305	668	1152	1641	1729	1582	1220
0.80	2	11	35	105	232	416	631	884	1085
	3	26	81	243	519	901	1196	1508	1494
1.00	1	3	10	36	83	160	201	518	690
	2	14	52	141	337	592	949	1203	1413

Table :(6.6.17)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.15	100	1	0	0	0	0	0	0	0
	100	1	0	0	0	0	0	0	0
0.18	787	35	0	0	0	0	0	0	0
	790	31	0	0	0	0	0	0	0
0.20	1674	181	12	0	0	0	0	0	0
	1690	166	8	0	0	0	0	0	0
0.25	3717	1398	387	73	10	0	0	0	0
	3873	1320	302	52	5	0	0	0	0
0.30	3048	2834	1578	645	205	59	11	1	0
	3432	2880	1372	477	137	25	5	0	0
0.35	1549	2468	2308	1721	871	397	110	32	13
	1953	2871	2236	1404	597	222	53	12	2
0.40	670	1568	2067	2081	1597	1006	502	232	91
	900	2035	2400	2060	1349	649	207	81	28
0.45	301	754	1376	1861	2019	1599	1025	579	276
	444	1153	1899	2191	1917	1259	617	302	109
0.50	117	387	859	1430	1829	1789	1466	1007	590
	203	699	1302	2009	2030	1610	1114	608	251
0.60	22	86	297	670	1059	1499	1605	1619	1237
	48	240	625	1217	1665	1861	1668	1210	758

Table :(6.6.18)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.15	86	0	0	0	0	0	0	0	0
	86	0	0	0	0	0	0	0	0
0.18	653	37	1	0	0	0	0	0	0
	661	29	0	0	0	0	0	0	0
0.20	1554	144	9	0	0	0	0	0	0
	1571	128	4	0	0	0	0	0	0
0.25	3483	1231	283	40	7	0	1	0	0
	3716	1093	165	19	5	0	0	0	0
0.30	3394	2636	1328	443	140	24	4	1	1
	3979	2606	964	253	51	9	0	0	0
0.35	1993	2665	2199	1439	608	244	76	19	3
	2608	3122	1887	990	311	100	29	3	1
0.40	988	1906	2286	2018	1310	737	322	111	32
	1510	2633	2403	1687	845	378	116	31	12
0.45	406	1171	1817	2059	1821	1280	725	346	141
	900	1945	2312	2053	1376	738	338	124	42
0.50	248	678	1283	1706	1947	1634	1164	741	326
	555	1304	1869	2117	1771	1190	643	295	125
0.60	60	204	585	1098	1432	1673	1621	1316	946
	201	622	1242	1776	1820	1619	1198	776	425

Table :(6.6.19)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.15	95	0	0	0	0	0	0	0	0
	95	0	0	0	0	0	0	0	0
0.18	747	38	2	0	0	0	0	0	0
	749	35	2	0	0	0	0	0	0
0.20	1645	182	10	0	0	0	0	0	0
	1667	163	4	0	0	0	0	0	0
0.25	3631	1412	380	76	9	0	0	0	0
	3802	1330	278	47	3	0	0	0	0
0.30	3111	2803	1496	635	202	62	10	3	1
	3533	2849	1267	432	114	19	5	1	0
0.35	1648	2413	2298	1633	898	336	148	33	12
	2160	2844	2323	1289	500	160	40	9	1
0.40	699	1614	2106	1985	1572	981	492	236	94
	1094	2270	2406	1924	1198	567	206	66	22
0.45	334	848	1438	1856	1859	1542	1015	563	286
	612	1401	2070	2141	1757	1053	525	214	81
0.50	135	445	950	1485	1793	1650	1405	989	575
	336	964	1584	2023	1952	1421	938	444	207
0.60	39	130	361	701	1130	1480	1547	1485	1203
	133	407	887	1419	1766	1781	1481	1044	578

Table :(6.6.20)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.15	112	0	0	0	0	0	0	0	0
	112	0	0	0	0	0	0	0	0
0.18	831	38	2	0	0	0	0	0	0
	833	30	0	0	0	0	0	0	0
0.20	1776	226	16	0	0	0	0	0	0
	1782	221	12	0	0	0	0	0	0
0.25	3659	1621	468	89	23	1	1	0	0
	3770	1610	382	61	11	0	0	0	0
0.30	2881	2743	1752	804	288	88	26	5	2
	3214	2893	1620	596	155	30	4	2	0
0.35	1400	2127	2324	1760	1122	545	203	82	22
	1746	2600	2457	1625	748	265	68	15	1
0.40	543	1306	1695	2055	1760	1198	706	361	152
	767	1834	2231	2215	1506	772	338	116	30
0.45	212	622	1176	1592	1820	1601	1252	830	483
	407	1068	1828	2082	1925	1331	765	350	113
0.50	106	289	693	1079	1492	1661	1553	1245	836
	223	632	1248	1814	1914	1768	1160	681	317
0.60	18	93	204	490	821	1185	1415	1391	1380
	79	268	608	1153	1547	1738	1673	1237	867

Table :(6.6.21)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.15	103	0	0	0	0	0	0	0	0
	103	0	0	0	0	0	0	0	0
0.10	837	44	2	1	0	0	0	0	0
	841	40	1	1	0	0	0	0	0
0.20	1810	209	13	1	0	0	0	0	0
	1820	191	11	0	0	0	0	0	0
0.25	3776	1669	422	97	9	2	0	0	0
	3804	1601	357	65	5	0	0	0	0
0.30	2832	2829	1781	773	298	76	10	4	0
	3136	2974	1626	585	154	29	4	0	0
0.35	1314	2134	2366	1863	1123	523	194	64	16
	1654	2576	2569	1599	774	240	62	12	1
0.40	476	1160	1897	2156	1794	1226	700	319	116
	746	1675	2472	2252	1543	711	292	88	24
0.45	160	545	1068	1722	1949	1710	1270	822	416
	338	1007	1820	2313	1977	1355	702	294	105
0.50	66	237	573	1108	1586	1801	1661	1222	831
	186	558	1262	1885	2088	1796	1126	634	291
0.60	10	28	154	398	747	1191	1555	1627	1469
	77	214	613	1147	1696	1975	1678	1236	718

Table :(6.6.22)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.15	104	1	0	0	0	0	0	0	0
	103	1	0	0	0	0	0	0	0
0.10	840	42	2	1	0	0	0	0	0
	819	36	1	1	0	0	0	0	0
0.20	1870	210	16	1	0	0	0	0	0
	1845	185	10	1	0	0	0	0	0
0.25	3737	1776	511	113	13	1	0	0	0
	3028	1674	386	58	6	0	0	0	0
0.30	2689	2738	1958	918	301	95	28	8	1
	3067	2869	1690	607	148	29	7	1	0
0.35	1162	2014	2345	1934	1233	627	244	95	31
	1612	2520	2560	1649	754	255	78	8	2
0.40	385	1036	1735	1979	1896	1379	817	443	165
	735	1684	2381	2227	1535	799	270	74	29
0.45	250	773	1332	1887	1897	1561	1132	611	328
	887	1746	2155	2047	1463	772	368	133	46
0.50	101	355	848	1385	1744	1741	1533	1068	634
	524	1200	1869	2109	1765	1177	710	290	119
0.60	21	93	249	563	1001	1480	1614	1679	1236
	212	567	1145	1727	1910	1741	1282	732	373

Table :(6.G.23)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.15	115	0	0	0	0	0	0	0	0
	110	0	0	0	0	0	0	0	0
0.18	903	47	4	1	0	0	0	0	0
	875	42	1	1	0	0	0	0	0
0.20	1963	251	17	1	0	0	0	0	0
	1891	208	12	0	0	0	0	0	0
0.25	3795	1860	550	133	16	2	0	0	0
	3842	1708	394	69	6	0	0	0	0
0.30	2533	2829	1940	991	411	132	29	8	1
	2970	2916	1626	626	172	25	5	0	0
0.35	1018	1904	2243	1981	1362	753	312	119	41
	1565	2528	2531	1619	773	238	52	15	1
0.40	346	877	1510	2042	1884	1470	959	482	249
	786	1660	2363	2249	1519	694	295	83	24
0.45	89	362	772	1411	1712	1763	1478	1082	660
	384	1007	1798	2260	1949	1306	698	322	108
0.50	39	144	372	767	1265	1577	1672	1464	1165
	230	592	1249	1861	2065	1717	1146	665	250
0.60	4	12	82	216	470	848	1203	1448	1504
	78	249	648	1116	1614	1837	1763	1275	775

Table :(6.G.24)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.15	106	1	0	0	0	0	0	0	0
	106	1	0	0	0	0	0	0	0
0.18	866	49	2	0	0	0	0	0	0
	872	41	1	0	0	0	0	0	0
0.20	1906	219	18	0	0	0	0	0	0
	1930	191	13	0	0	0	0	0	0
0.25	3766	1706	509	105	16	4	0	0	0
	4003	1627	373	43	5	1	0	0	0
0.30	2644	2771	1886	913	374	90	23	3	1
	3127	3076	1667	574	111	25	5	0	0
0.35	1143	2023	2419	1884	1202	634	255	85	15
	1658	2839	2684	1570	591	192	33	7	1
0.40	399	1038	1714	2029	1864	1384	811	413	162
	759	1904	2585	2369	1372	582	206	51	15
0.45	128	459	987	1519	1846	1723	1426	916	554
	339	1197	2095	2447	1872	1141	564	210	56
0.50	44	167	484	964	1376	1732	1724	1360	960
	148	672	1482	2215	2219	1598	930	475	148
0.60	5	34	117	273	546	1008	1408	1619	1564
	70	238	722	1412	1908	2010	1674	1064	519

Table :(G.G.25)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.15	89	0	0	0	0	0	0	0	0
	89	0	0	0	0	0	0	0	0
0.18	800	42	2	0	0	0	0	0	0
	805	37	1	0	0	0	0	0	0
0.20	1772	216	10	1	0	0	0	0	0
	1807	175	7	1	0	0	0	0	0
0.25	3664	1626	464	91	22	2	0	0	0
	3882	1531	335	40	6	0	0	0	0
0.30	2876	2826	1765	770	256	77	20	6	1
	3418	3063	1459	428	86	18	1	0	0
0.35	1308	2231	2388	1821	1071	505	197	72	14
	1870	3005	2607	1324	460	139	36	7	0
0.40	503	1243	1906	2118	1789	1213	643	311	125
	963	2170	2765	2211	1056	467	137	42	8
0.45	171	557	1217	1608	1973	1681	1256	804	389
	445	1392	2344	2449	1752	948	395	144	32
0.50	58	262	617	1129	1592	1851	1603	1241	796
	257	869	1857	2294	2097	1418	717	288	101
0.60	9	47	174	421	787	1200	1524	1647	1450
	124	424	991	1751	2100	1903	1330	754	351

Table :(G.G.26)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.15	82	0	0	0	0	0	0	0	0
	82	0	0	0	0	0	0	0	0
0.18	776	43	3	0	0	0	0	0	0
	781	35	2	0	0	0	0	0	0
0.20	1632	190	9	1	0	0	0	0	0
	1654	160	8	1	0	0	0	0	0
0.25	3595	1548	415	79	15	0	0	0	0
	3806	1454	272	33	4	0	0	0	0
0.30	3050	2768	1582	695	238	53	15	1	1
	3561	2917	1309	359	74	14	1	1	0
0.35	1535	2357	2329	1753	917	394	160	41	9
	2151	3144	2384	1193	338	81	23	3	0
0.40	659	1468	1986	2155	1678	1026	534	234	85
	1185	2472	2752	1977	923	297	85	31	2
0.45	283	743	1303	1833	1937	1627	1119	627	296
	660	1663	2573	2323	1525	768	242	80	18
0.50	103	396	776	1351	1736	1841	1509	996	684
	363	1196	2116	2413	1874	1167	517	183	62
0.60	14	89	264	559	1060	1378	1659	1594	1299
	214	663	1317	2013	2092	1684	1084	525	229

Table :(G.G.27)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.15	79	1	0	0	0	0	0	0	0
	79	1	0	0	0	0	0	0	0
0.18	721	34	4	0	0	0	0	0	0
	724	29	2	0	0	0	0	0	0
0.20	1531	163	7	1	0	0	0	0	0
	1548	141	4	1	0	0	0	0	0
0.25	3532	1366	323	58	11	1	0	0	0
	3711	1266	207	22	1	0	0	0	0
0.30	3271	2625	1517	585	170	39	6	0	0
	3779	2749	1191	271	41	4	1	0	0
0.35	1785	2558	2312	1525	765	317	107	31	2
	2475	3253	2226	890	257	56	10	3	0
0.40	816	1657	2206	2089	1505	897	393	183	61
	1464	2828	2642	1720	690	240	49	8	1
0.45	362	910	1612	1976	1807	1445	940	469	213
	887	2073	2703	2238	1218	448	173	44	14
0.50	140	546	1021	1611	1903	1700	1291	845	455
	561	1582	2409	2354	1649	837	310	108	28
0.60	27	144	396	748	1185	1635	1676	1591	1097
	351	925	1730	2152	2059	1396	736	348	135

Table :(G.G.28)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.15	83	0	0	0	0	0	0	0	0
	83	0	0	0	0	0	0	0	0
0.18	640	22	0	0	0	0	0	0	0
	644	14	0	0	0	0	0	0	0
0.20	1504	116	9	1	0	0	0	0	0
	1515	101	5	1	0	0	0	0	0
0.25	3418	1289	304	46	10	0	0	0	0
	3594	1144	205	18	1	0	0	0	0
0.30	3334	2598	1374	439	121	36	5	0	0
	3851	2708	930	193	31	2	0	0	0
0.35	2025	2616	2234	1405	601	239	78	11	5
	2763	3229	2051	718	164	30	3	0	0
0.40	1024	1878	2282	2857	1257	729	336	119	39
	1727	2974	2685	1411	472	166	29	3	0
0.45	458	1108	1796	2125	1832	1281	724	360	150
	1118	2338	2757	2027	945	325	108	30	2
0.50	226	641	1328	1863	1871	1626	1138	670	343
	843	1860	2644	2220	1286	583	218	75	10
0.60	67	219	580	1095	1494	1669	1626	1314	926
	576	1320	2130	2207	1684	1052	494	174	54

Table :(6.6.29)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.15	90	0	0	0	0	0	0	0	0
	89	0	0	0	0	0	0	0	0
0.18	720	37	3	0	0	0	0	0	0
	706	33	0	0	0	0	0	0	0
0.20	1613	160	7	0	0	0	0	0	0
	1587	128	2	0	0	0	0	0	0
0.25	3620	1307	333	51	9	0	0	0	0
	3748	1128	191	12	1	0	0	0	0
0.30	3218	2651	1404	605	165	36	9	2	0
	3776	2587	992	247	19	9	0	0	0
0.35	1800	2533	2359	1436	747	331	112	27	8
	2672	3183	2017	744	191	32	3	2	0
0.40	868	1716	2214	2012	1494	871	394	157	47
	1765	2903	2602	1422	493	141	24	8	1
0.45	406	1021	1619	1967	1807	1363	900	469	241
	1165	2368	2751	1879	909	349	129	24	5
0.50	162	538	1087	1629	1869	1648	1331	822	488
	795	1910	2612	2192	1282	565	220	75	17
0.60	39	155	410	824	1309	1632	1697	1399	1070
	560	1253	2010	2251	1780	1074	478	189	67

Table :(6.6.30)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.15	94	1	0	0	0	0	0	0	0
	88	1	0	0	0	0	0	0	0
0.18	763	31	0	0	0	0	0	0	0
	735	21	0	0	0	0	0	0	0
0.20	1675	177	9	0	0	0	0	0	0
	1617	134	2	0	0	0	0	0	0
0.25	3678	1409	384	65	12	0	0	0	0
	3756	1166	206	12	1	0	0	0	0
0.30	3078	2681	1559	677	218	54	14	2	0
	3717	2552	1061	231	28	7	0	0	0
0.35	1591	2428	2365	1597	855	399	158	47	14
	2640	3137	2041	761	172	37	4	2	0
0.40	709	1495	2129	2055	1607	1027	500	220	61
	1729	2830	2591	1420	508	139	32	9	1
0.45	308	859	1422	1852	1850	1480	1052	592	313
	1186	2265	2739	1804	911	355	116	23	4
0.50	117	441	879	1453	1760	1718	1413	1009	604
	831	1828	2581	2198	1302	565	218	69	17
0.60	26	112	303	652	1131	1444	1651	1517	1204
	579	1234	1951	2250	1789	1072	502	182	77

Table (G.G.31)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.15	103	0	0	0	0	0	0	0	0
	97	0	0	0	0	0	0	0	0
0.18	777	40	2	0	0	0	0	0	0
	730	27	0	0	0	0	0	0	0
0.20	1753	197	12	0	0	0	0	0	0
	1649	140	4	0	0	0	0	0	0
0.25	3724	1521	438	77	13	0	0	0	0
	3783	1208	207	15	1	0	0	0	0
0.30	2945	2681	1702	770	269	71	20	3	1
	3621	2548	1119	243	30	9	0	0	0
0.35	1434	2256	2353	1729	1000	471	204	68	22
	2575	3116	2003	792	174	35	4	1	0
0.40	570	1312	1992	2060	1707	1129	630	296	104
	1720	2763	2583	1429	492	130	31	6	0
0.45	229	708	1256	1741	1829	1541	1207	742	375
	1162	2272	2630	1906	923	336	118	29	4
0.50	75	337	713	1291	1647	1689	1496	1157	757
	827	1778	2538	2192	1342	551	217	59	17
0.60	16	71	245	500	917	1263	1561	1595	1292
	594	1241	1892	2241	1801	1053	521	195	76

Table (G.G.32)

SD.	1ER	2ER	3ER	4ER	5ER	6ER	7ER	8ER	9ER
0.15	102	0	0	0	0	0	0	0	0
	92	0	0	0	0	0	0	0	0
0.18	841	45	3	0	0	0	0	0	0
	784	35	1	0	0	0	0	0	0
0.20	1816	219	14	0	0	0	0	0	0
	1676	158	4	0	0	0	0	0	0
0.25	3739	1615	490	92	21	1	0	0	0
	3756	1256	221	17	1	0	0	0	0
0.30	2801	2722	1793	872	320	86	27	4	2
	3538	2590	1104	253	36	8	0	0	0
0.35	1274	2104	2315	1887	1079	572	246	90	34
	2493	3050	2002	827	184	32	4	0	0
0.40	502	1123	1852	2035	1767	1237	768	390	129
	1716	2722	2530	1423	513	134	31	5	0
0.45	180	579	1110	1551	1813	1636	1323	839	476
	1223	2191	2616	1887	904	352	102	24	5
0.50	59	261	571	1119	1494	1681	1506	1290	898
	839	1799	2478	2129	1358	572	203	56	20
0.60	14	49	171	406	747	1119	1382	1585	1381
	594	1226	1898	2202	1757	1096	535	205	64

CHAPTER

SEVEN

7

CONCLUSIONS AND FURTHER WORK

7.1 CONCLUSIONS:

The object of this research was to investigate possible code-division multiplex techniques for use in data transmission systems. The work predominantly related to the important problem in code-division multiplexing techniques of finding a suitable code set which can be used as codes for the channels in the system and which can accomodate a larger number of channels. The work also considered the performance of systems using these codes over channels impaired by Gaussian noise. Most data communication sytems are subject to such an impairment.

A systematic study was carried out on techniques of multiplexing data in general and on code division multiplexing in particular. This led to the investigation of the suitability of a group of functions, called bridge functions, for use as codes in code division multiplexing systems. These functions are three valued functions which take the values 0, +1 and -1 .

The present work has contributed to the development of the theory and practice of code-division multiplexing in that a new coding strategy has been proposed in which binary data can be multiplexed by first transforming the binary data into ternary data at the multiplexer. The opposite transformation (i.e. transformation from ternary into binary) is then carried out at the demultiplexer.

The necessary requirement for a set of codes or functions to be suitable as codes for a code-division multiplexing system which uses this strategy has been derived and a further requirement concerning the selection of a subset in the suitable set has been discussed.

A comprehensive study of three systems has been carried out to show the performance when using bridge functions as codes in code division multiplexing systems based on the new strategy. The study also shows that with the new strategy a further important problem in code-division multiplexing can be solved, this is the demand for a greater number of useable data channels.

The first system to be studied was an 8 channel system which uses the bridge function Bri(i,1,3,t). The results from this system show that this system has an interesting property concerning the adaptive relationship between the channel error correcting properties and the number of active channels. When the number of active channels is less than the full load the system shows two modes of error correction. These are a triple-error correcting mode and a single error correcting mode. The triple-error correcting mode occurs when 1 or 2 channels are active, while the single error correcting mode occurs when 3 or 4 channels are active. However, the system shows a continuous adaptability of error correcting capability when the number of active channels changes. The limit of the system is reached when all the 8 channels are active.

The second system was a 16 channel system which uses the bridge functions Bri(i,1,4,t) and Bri(i,2,4,t). The results from this system also show the adaptive relationship between the channel error correcting properties and the number of active channels. There are four error correcting modes associated with this system depending on the number of active channels. These are 8-error correcting mode, a 4-error correcting mode, a triple-error correcting mode and a single-error correcting mode.

The third system was a 32 channel system which uses the set of bridge functions Bri(i,1,5,t), Bri(i,2,5,t) and Bri(i,3,5,t). The results from this system also show the adaptive relationship between the channel error correcting properties and the number of active channels. There are 6 error correcting modes associated with this system. These modes are an 18-error correcting mode, an 11-error correcting mode, an 8-error correcting mode, a 5 error correcting mode, a triple-error correcting mode and a single-error correcting mode.

In all three systems in an r-error correcting mode, besides correcting all blocks with r errors, blocks with r+1, r+2, ...,N (where N is the maximum number of channels which can be used in the system) will be partially corrected, i.e. some r+1, r+2, ...,N errors will be corrected in a proportion of cases which decreases as the number of errors increases. Hence, the proportion will be greater for those errors near to r errors, like r+1, r+2 and less for those errors near to N.

In all the three systems, besides the above mentioned error correcting modes there are various hybrid situations between these modes with different error correction properties.

By using the new coding strategy it is possible to design systems with a relatively high number of data channels. However, a system with an even higher number of data channels can be achieved using the 32 channel system and applying a two-stage transformation process. The maximum number of data channels can then be increased from 7x7 = 49 channels using 7-truncated Walsh functions, to 32x32 = 1024 channels using the new technique. This is a great improvement in code-division multiplexing capability.

Because of the inherent automatic error-correction in the proposed system when the system is not fully loaded, it is possible to use these sytems for multiplexing a number of channels less than the maximum number of channels which can be used and reserve the rest of the channels for error correction. for example the 8 channel system can be used for multiplexing 4 channels and the 4 channels left can be used for error correction. This arrangement will give a system with a single-error correction mode as can be seen from the results obtained when using 4 channels.

The 16 channel system also can be used to provide a sytem with a single-error correction mode. This system is able to multiplex

14 channels and the two redundant channels will provide the single-error correction mode.

The 32 channel system also can be used to provide a system with a single-error correction mode if 4 channels are made redundant channels. This sytem is thus able to multiplex 28 channels with a single-error correction mode.

Other systems with different error correction modes are feasable.

Comparing with other systems like the TDM, the number of redundant channels which are needed for a single-error correction mode is almost the same as in the proposed systems. However, the advantage of the proposed system over TDM is the simplicity of the proposed system because there is no need for a coding and decoding scheme for the error correction.

Another application of the proposed system is the possibility of using it in cryptographic data communication. This follows from an important property of the code-division multiplexing system in which each data vector is mapped to a unique vector at the multiplexer and remapped to the original one at the demultiplexer. Thus for each code there is a one-to-one mapping between the original and the transformed vector. However, any change among the rows and / or columns maps the original vector onto an altogether different vector. This property of the system establishes it as suitable not only as a multiplexing technique but also a method for cryptographic data communication.

7.2 SUGGESTIONS FOR FURTHER WORK:

The present work has contributed to the development of the theory and practice of code-division multiplexing by introducing a new coding strategy which enables the use of a relatively higher number of data channels than any so far described in the literature. There are two areas in which further work could be undertaken.

SIMULATION:

The proposed strategy can be applied not only to bridge functions but also to any set of three-valued functions. Hence, a simulation study based on the proposed strategy but using different functions but simillar to bridge functions would be useful.

PRACTICE:

The development and testing of a practical system based on the the theoretical and simulation work given in this thesis needs to be carried out to show the commercial viability of the proposed system.

APPENDICES

INTRODUCTION TO APPENDICES:

1. Appendix A contains 3 programmes written in FORTRAN, these programmes are for the generation of the matrix representation of the sets of bridge functions bri(i,j,3,t),bri(i,j,4,t) and bri(i,j,5,t) where j takes the values 0, 1, 2 and 3 in the first set, 0, 1, 2, 3, and 4 in the second set and 0, 1, 2, 3, 4 and 5 in the third set.

2. Appendix B contains 3 programmes written in FORTRAN and using GINO-F library routines for drawing the waveforms of those bridge functions mentioned in 1.

3. Appendix C contains the following figures:
   A. Figures (A.j) for j = 0, 1, 2 and 3.
   B. Figures (B.j) for j = 0, 1, 2, 3 and 4.
   C. Figures (C.j) for j = 0, 1, 2, 3, 4, and 5.

These figures show the waveforms of those bridge functions mentioned in 1.

APPENDIX A
PROGRAMMES FOR MATRICES REPRESENTATION

```
C  The following programme is for generating the matrix
C  representation of a bridge function Bri(i,j,p,t)
C  Where i is the order of the bridge function and
C  j is the shift information j=NOS in the programme
C  p is follows from N=2**p where N is the number of
C  bridge functions in this programme N=8.
       DIMENSION NC(8,8),N(5)
       WRITE(7,5)
  5    FORMAT(30H ENTER NO. OF BRIDGE FUNCTIONS)
       READ(7,15) NOB
  15   FORMAT(I3)
       WRITE(7,1000)
 1000  FORMAT(21H  ENTER NO. OF SHIFTS)
       READ(7,1100) NOS
 1100  FORMAT(I2)
       NOC=3-NOS
       DO 60 I1=1,NOB
       IF(NOS.EQ.3) THEN
       DO 10 J1=1,2**NOS
       IF(I1.NE.J1) THEN
       NC(I1,J1)=0
       ELSE
       NC(I1,J1)=1
       END IF
   10  CONTINUE
       ELSE IF(NOS.EQ.2) THEN
       DO 20 J1=1,2**NOS
       IF(I1.NE.J1.AND.I1.NE.(J1+2**NOS)) THEN
       NC(I1,J1)=0
       ELSE
       NC(I1,J1)=1
       END IF
   20  CONTINUE
       ELSE IF (NOS.EQ.1) THEN
       DO 30 J1=1,2**NOS
       IF (I1.LE.4) THEN
        IF(I1.NE.J1.AND.I1.NE.(J1+2**NOS)) THEN
        NC(I1,J1)=0
        ELSE
        NC(I1,J1)=1
        END IF
       ELSE
       NC(I1,J1)=NC(I1-4,J1)
       END IF
   30   CONTINUE
       ELSE
       NC(I1,1)=1
       END IF
   60  CONTINUE
       IF(NOS.EQ.3) GO TO 190
       DO 180 I=0,NOB-1
```

```
      DO 70 I1=0,2
      I0=I1+1
      I2=2**I1
      N(I0)=IAND(I,I2)
   70 CONTINUE
        IF(N(3).EQ.0) THEN
        N1=0
        DO 80 J2=2**(NOS+1),2**NOS+1,-1
        NC(I+1,J2)=NC(I+1,N1+1)
        N1=N1+1
   80   CONTINUE
         ELSE
         N2=0
         DO 90 J3=2**(NOS+1),2**NOS+1,-1
        NC(I+1,J3)=-NC(I+1,N2+1)
         N2=N2+1
   90    CONTINUE
        END IF
         IF(NOS.EQ.2) GO TO 180
        IF(N(2).EQ.0) THEN
        N3=0
        DO 100 J4=2**(NOS+2),2**(NOS+1)+1,-1
       NC(I+1,J4)=NC(I+1,N3+1)
        N3=N3+1
  100   CONTINUE
         ELSE
         N4=0
         DO 110 J5=2**(NOS+2),2**(NOS+1)+1,-1
        NC(I+1,J5)=-NC(I+1,N4+1)
         N4=N4+1
  110    CONTINUE
        END IF
         IF(NOS.EQ.1) GO TO 180
        IF(N(1).EQ.0) THEN
       N9=0
       DO 160 J10=NOB,5,-1
       NC(I+1,J10)=NC(I+1,N9+1)
       N9=N9+1
  160  CONTINUE
       ELSE
       N10=0
      DO 170 J11=NOB,5,-1
       NC(I+1,J11)=-NC(I+1,N10+1)
       N10=N10+1
  170  CONTINUE
       END IF
  180  CONTINUE
  190 WRITE(15,1205) ((NC(I,J),J=1,NOB),I=1,NOB)
 1205 FORMAT(8I3)
       STOP
       END
C  *****************************************
```

```
C  The following programme is for generating the matrix
C  representation of a bridge function Bri(i,j,p,t)
C  Where i is the order of the bridge function and
C  j is the shift information j=NOS in the programme
C  p is follows from N=2**p where N is the number of
C  bridge functions in this brogramme N=16.
      DIMENSION NC(16,16),N(5)
      WRITE(7,5)
  5   FORMAT(30H ENTER NO. OF BRIDGE FUNCTIONS)
      READ(7,15) NOB
 15   FORMAT(I3)
      WRITE(7,1000)
1000  FORMAT(21H  ENTER NO. OF SHIFTS)
      READ(7,1100) NOS
1100  FORMAT(I2)
      NOC=4-NOS
      DO 60 I1=1,NOB
      IF(NOS.EQ.4) THEN
      DO 10 J1=1,2**NOS
      IF(I1.NE.J1) THEN
      NC(I1,J1)=0
      ELSE
      NC(I1,J1)=1
      END IF
 10   CONTINUE
      ELSE IF(NOS.EQ.3) THEN
      DO 20 J1=1,2**NOS
      IF(I1.NE.J1.AND.I1.NE.(J1+2**NOS)) THEN
      NC(I1,J1)=0
      ELSE
      NC(I1,J1)=1
      END IF
 20   CONTINUE
      ELSE IF (NOS.EQ.2) THEN
      DO 30 J1=1,2**NOS
      IF (I1.LE.8) THEN
       IF(I1.NE.J1.AND.I1.NE.(J1+2**NOS)) THEN
       NC(I1,J1)=0
       ELSE
       NC(I1,J1)=1
       END IF
      ELSE
      NC(I1,J1)=NC(I1-8,J1)
      END IF
 30    CONTINUE
      ELSE IF (NOS.EQ.1) THEN
      DO 40 J1=1,2**NOS
       IF(I1.LE.4) THEN
       IF(I1.NE.J1.AND.I1.NE.(J1+2**NOS)) THEN
       NC(I1,J1)=0
       ELSE
       NC(I1,J1)=1
       END IF
       ELSE IF (I1.LE.8) THEN
```

```
       NC(I1,J1)=NC(I1-4,J1)
       ELSE
       NC(I1,J1)=NC(I1-8,J1)
       END IF
   40  CONTINUE
      ELSE
      NC(I1,1)=1
      END IF
   60 CONTINUE
      IF(NOS.EQ.4) GO TO 190
      DO 180 I=0,NOB-1
      DO 70 I1=0,3
      I0=I1+1
      I2=2**I1
      N(I0)=IAND(I,I2)
   70 CONTINUE
        IF(N(4).EQ.0) THEN
        N1=0
        DO 80 J2=2**(NOS+1),2**NOS+1,-1
        NC(I+1,J2)=NC(I+1,N1+1)
        N1=N1+1
   80   CONTINUE
         ELSE
         N2=0
         DO 90 J3=2**(NOS+1),2**NOS+1,-1
        NC(I+1,J3)=-NC(I+1,N2+1)
         N2=N2+1
   90    CONTINUE
        END IF
         IF(NOS.EQ.3) GO TO 180
        IF(N(3).EQ.0) THEN
        N3=0
        DO 100 J4=2**(NOS+2),2**(NOS+1)+1,-1
       NC(I+1,J4)=NC(I+1,N3+1)
        N3=N3+1
  100   CONTINUE
         ELSE
         N4=0
         DO 110 J5=2**(NOS+2),2**(NOS+1)+1,-1
        NC(I+1,J5)=-NC(I+1,N4+1)
         N4=N4+1
  110    CONTINUE
        END IF
         IF(NOS.EQ.2) GO TO 180
        IF(N(2).EQ.0) THEN
        N5=0
        DO 120 J6=2**(NOS+3),2**(NOS+2)+1,-1
        NC(I+1,J6)=NC(I+1,N5+1)
        N5=N5+1
  120   CONTINUE
        ELSE
        N6=0
        DO 130 J7=2**(NOS+3),2**(NOS+2)+1,-1
        NC(I+1,J7)=-NC(I+1,N6+1)
```

```
        N6=N6+1
  130   CONTINUE
        END IF
      IF(NOS.EQ.1) GO TO 180
       IF(N(1).EQ.0) THEN
       N9=0
       DO 160 J10=NOB,9,-1
       NC(I+1,J10)=NC(I+1,N9+1)
       N9=N9+1
  160  CONTINUE
       ELSE
       N10=0
      DO 170 J11=NOB,9,-1
       NC(I+1,J11)=-NC(I+1,N10+1)
       N10=N10+1
  170  CONTINUE
       END IF
  180  CONTINUE
  190 IF(NOB.EQ.32)THEN
      WRITE(15,1200) ((NC(I,J),J=1,NOB),I=1,NOB)
 1200 FORMAT(32I2)
      ELSE
      WRITE(15,1205) ((NC(I,J),J=1,NOB),I=1,NOB)
 1205 FORMAT(16I3)
      END IF
       STOP
       END
C  ******************************************


C  The following programme is for generating the matrix
C  representation of a bridge function Bri(i,j,p,t)
C  Where i is the order of the bridge function and
C  j is the shift information j=NOS in the programme
C  p is follows from N=2**p where N is the number of
C  bridge functions in this brogramme N=32.
      DIMENSION NC(32,32),N(5)
      WRITE(7,5)
  5   FORMAT(30H ENTER NO. OF BRIDGE FUNCTIONS)
      READ(7,15) NOB
  15  FORMAT(I3)
      WRITE(7,1000)
      DIMENSION NC(32,32),N(5)
      WRITE(7,5)
  5   FORMAT(30H ENTER NO. OF BRIDGE FUNCTIONS)
      READ(7,15) NOB
  15  FORMAT(I3)
      WRITE(7,1000)
 1000 FORMAT(21H  ENTER NO. OF SHIFTS)
      READ(7,1100) NOS
 1100 FORMAT(I2)
      NOC=5-NOS
      DO 60 I1=1,NOB
      IF(NOS.EQ.5) THEN
```

```
      DO 10 J1=1,2**NOS
      IF(I1.NE.J1) THEN
      NC(I1,J1)=0
      ELSE
      NC(I1,J1)=1
      END IF
   10 CONTINUE
      ELSE IF(NOS.EQ.4) THEN
      DO 20 J1=1,2**NOS
      IF(I1.NE.J1.AND.I1.NE.(J1+2**NOS)) THEN
      NC(I1,J1)=0
      ELSE
      NC(I1,J1)=1
      END IF
   20 CONTINUE
      ELSE IF (NOS.EQ.3) THEN
      DO 30 J1=1,2**NOS
      IF (I1.LE.16) THEN
       IF(I1.NE.J1.AND.I1.NE.(J1+2**NOS)) THEN
       NC(I1,J1)=0
       ELSE
       NC(I1,J1)=1
       END IF
      ELSE
      NC(I1,J1)=NC(I1-16,J1)
      END IF
   30  CONTINUE
      ELSE IF (NOS.EQ.2) THEN
      DO 40 J1=1,2**NOS
       IF(I1.LE.8) THEN
       IF(I1.NE.J1.AND.I1.NE.(J1+2**NOS)) THEN
       NC(I1,J1)=0
       ELSE
       NC(I1,J1)=1
       END IF
       ELSE IF (I1.LE.16) THEN
       NC(I1,J1)=NC(I1-8,J1)
       ELSE
       NC(I1,J1)=NC(I1-16,J1)
       END IF
   40  CONTINUE
      ELSE IF(NOS.EQ.1) THEN
       DO 50 J1=1,2**NOS
       IF(I1.LE.4) THEN
       IF(I1.NE.J1.AND.I1.NE.(J1+2**NOS)) THEN
       NC(I1,J1)=0
       ELSE
       NC(I1,J1)=1
       END IF
       ELSE IF (I1.LE.8) THEN
       NC(I1,J1)=NC(I1-4,J1)
       ELSE IF(I1.LE.16) THEN
       NC(I1,J1)=NC(I1-8,J1)
       ELSE
```

```
       NC(I1,J1)=NC(I1-16,J1)
       END IF
 50    CONTINUE
      ELSE
      NC(I1,1)=1
      END IF
 60   CONTINUE
      IF(NOS.EQ.5) GO TO 190
      DO 180 I=0,31
      DO 70 I1=0,4
      I0=I1+1
      I2=2**I1
      N(I0)=IAND(I,I2)
 70   CONTINUE
        IF(N(5).EQ.0) THEN
        N1=0
        DO 80 J2=2**(NOS+1),2**NOS+1,-1
        NC(I+1,J2)=NC(I+1,N1+1)
        N1=N1+1
 80     CONTINUE
         ELSE
         N2=0
         DO 90 J3=2**(NOS+1),2**NOS+1,-1
        NC(I+1,J3)=-NC(I+1,N2+1)
         N2=N2+1
 90      CONTINUE
        END IF
         IF(NOS.EQ.4) GO TO 180
        IF(N(4).EQ.0) THEN
        N3=0
        DO 100 J4=2**(NOS+2),2**(NOS+1)+1,-1
       NC(I+1,J4)=NC(I+1,N3+1)
        N3=N3+1
100     CONTINUE
         ELSE
         N4=0
         DO 110 J5=2**(NOS+2),2**(NOS+1)+1,-1
        NC(I+1,J5)=-NC(I+1,N4+1)
         N4=N4+1
110      CONTINUE
        END IF
         IF(NOS.EQ.3) GO TO 180
        IF(N(3).EQ.0) THEN
        N5=0
        DO 120 J6=2**(NOS+3),2**(NOS+2)+1,-1
        NC(I+1,J6)=NC(I+1,N5+1)
        N5=N5+1
120     CONTINUE
        ELSE
        N6=0
        DO 130 J7=2**(NOS+3),2**(NOS+2)+1,-1
        NC(I+1,J7)=-NC(I+1,N6+1)
        N6=N6+1
130     CONTINUE
```

```
        END IF
       IF(NOS.EQ.2) GO TO 180
       IF(N(2).EQ.0) THEN
       N7=0
       DO 140 J8=2**(NOS+4),2**(NOS+3)+1,-1
       NC(I+1,J8)=NC(I+1,N7+1)
       N7=N7+1
 140   CONTINUE
       ELSE
       N8=0
       DO 150 J9=2**(NOS+4),2**(NOS+3)+1,-1
       NC(I+1,J9)=-NC(I+1,N8+1)
       N8=N8+1
 150   CONTINUE
       END IF
      IF(NOS.EQ.1) GO TO 180
       IF(N(1).EQ.0) THEN
       N9=0
       DO 160 J10=NOB,17,-1
       NC(I+1,J10)=NC(I+1,N9+1)
       N9=N9+1
 160   CONTINUE
       ELSE
       N10=0
       DO 170 J11=NOB,17,-1
       NC(I+1,J11)=-NC(I+1,N10+1)
       N10=N10+1
 170   CONTINUE
       END IF
 180   CONTINUE
 190  IF(NOB.EQ.32)THEN
      WRITE(15,1200) ((NC(I,J),J=1,NOB),I=1,NOB)
1200  FORMAT(32I2)
      ELSE
      WRITE(15,1205) ((NC(I,J),J=1,NOB),I=1,NOB)
1205  FORMAT(16I3)
      END IF
       STOP
       END
C   ******************************************
```

APPENDIX B
PROGRAMMES FOR DRAWING BRIDGE FUNCTIONS

```
C  The following programme is for drawing the waveform of
C  a bridge function bri(i,j,p,t) where i is the order
C  of the bridge function, j is the shift information
C  and p follows from N=2**p where N is the number of
C  bridge functions. In this programme N=8.
C  The programme also write the matrix representation.
C  The programme first reads the matrix representation
C  from a file stored in the memory.
      DIMENSION NC(8,8)
      READ(16,1000) ((NC(I,J),J=1,8),I=0,7)
      WRITE(7,1010)
      READ(7,1020) NOB
      WRITE(7,1030)
      READ(7,1040) NOS
      WRITE(7,1045)
 1045 FORMAT(4H NOF)
      READ(7,1040) NOF
      WRITE(7,1050)
      READ(7,1060) MOD
      CALL GINO
      CALL SHIFT2(20.,10.)
      IF(MOD.EQ.1) THEN
      CALL SCALE(.5)
      END IF
      CALL CHASIZ(2.,2.5)
      DO 110 I=0,NOB-1
      CALL MOVTO2(260.-12*I,170.)
      CALL CHAANG(-90.)
      CALL CHAHOL(6Hbri(*.)
      CALL CHAINT(I,1)
      CALL CHAHOL(9H,1,3,t)*.)
      CALL MOVTO2(260.-12*I,140.)
      IF (NC(I,1).EQ.0.) THEN
      CALL LINBY2(0.,-12.)
      ELSE
      CALL MOVBY2(5.,0.)
      CALL LINBY2(0.,-12.)
      END IF
      DO 100 J=2,NOB
      IF (NC(I,J).EQ.NC(I,J-1)) THEN
      CALL LINBY2(0.,-12.)
      ELSE
      IF (NC(I,J).EQ.1) THEN
        IF(NC(I,J-1).EQ.0) THEN
        CALL LINBY2(5.,0.)
        CALL LINBY2(0.,-12.)
        ELSE
        CALL LINBY2(10.,0.)
        CALL LINBY2(0.,-12.)
        END IF
```

```
      ELSE IF(NC(I,J).EQ.0) THEN
        IF(NC(I,J-1).EQ.1) THEN
        CALL LINBY2(-5.,0.)
        CALL LINBY2(0.,-12.)
        ELSE
        CALL LINBY2(5.,0.)
        CALL LINBY2(0.,-12.)
         END IF
      ELSE
        IF(NC(I,J-1).EQ.0) THEN
        CALL LINBY2(-5.,0.)
        CALL LINBY2(0.,-12.)
        ELSE
        CALL LINBY2(-10.,0.)
        CALL LINBY2(0.,-12.)
        END IF
      END IF
      END IF
  100 CONTINUE
  110 CONTINUE
      DO 130 I=0,7
      CALL MOVTO2(155.-10.*I,127.)
      DO 120 J=1,8
      IF (NC(I,J).EQ.0) THEN
      CALL CHAHOL(6H   0*.)
      ELSE IF(NC(I,J).GT.0) THEN
      CALL CHAHOL(6H   +*.)
      ELSE
      CALL CHAHOL(6H   -*.)
      END IF
  120 CONTINUE
  130 CONTINUE
      CALL MOVTO2(60.,140.)
      CALL CHAHOL(8HFIG. 2*.)
      CALL CHAHOL(32H : The bridge functions bri(i,*.)
      CALL CHAINT(NOS,1)
      CALL CHAHOL(7H,3,t)*.)
      CALL MOVTO2(52.,122.)
      CALL CHAHOL(34Hand their matrix representation.*.)
      CALL DEVEND
      STOP
 1000 FORMAT(8I3)
 1010 FORMAT(30H ENTER NO. OF BRIDGE FUNCTIONS)
 1020 FORMAT(I3)
 1030 FORMAT(20H ENTER NO. OF SHIFTS)
 1040 FORMAT(I1)
 1050 FORMAT(35H  TYPE: 1 FOR TERMINAL 2 FOR PLOTER)
 1060 FORMAT(I2)
      END
C ***************************************
```

```
C  The following programme is for drawing the waveform of
C  a bridge function bri(i,j,p,t) where i is the order
C  of the bridge function, j is the shift information
C  and p follows from N=2**p where N is the number of
C  bridge functions. In this programme N=16.
C  The programme also write the matrix representation.
C  The programme first reads the matrix representation
C  from a file stored in the memory.
      DIMENSION NC(16,16)
      READ(16,1000) ((NC(I,J),J=1,16),I=1,16)
      WRITE(7,1010)
      READ(7,1020) NOB
      WRITE(7,1030)
      READ(7,1040) NOS
      WRITE(7,1045)
 1045 FORMAT(4H NOF)
      READ(7,1040) NOF
      WRITE(7,1050)
      READ(7,1060) MOD
      CALL GINO
      IF(MOD.EQ.1) THEN
      CALL SCALE(.5)
      END IF
      CALL CHASIZ(2.,1.5)
      DO 110 I=1,NOB
      CALL MOVTO2(300.-8*I,176.)
      CALL CHAANG(-90.)
      CALL CHAHOL(6Hbri(*.)
      CALL CHAINT(I-1,2)
      CALL CHAHOL(3H,*.)
      CALL CHAINT(NOS,1)
      CALL CHAHOL(7H,4,t)*.)
      CALL MOVTO2(300.-8*I,150.)
      IF (NC(I,1).EQ.0.) THEN
      CALL LINBY2(0.,-6.)
      ELSE
      CALL MOVBY2(3.5,0.)
      CALL LINBY2(0.,-6.)
      END IF
      DO 100 J=2,NOB
      IF (NC(I,J).EQ.NC(I,J-1)) THEN
      CALL LINBY2(0.,-6.)
      ELSE
      IF (NC(I,J).EQ.1) THEN
        IF(NC(I,J-1).EQ.0) THEN
        CALL LINBY2(3.5,0.)
        CALL LINBY2(0.,-6.)
        ELSE
        CALL LINBY2(7.,0.)
        CALL LINBY2(0.,-6.)
        END IF
      ELSE IF(NC(I,J).EQ.0) THEN
        IF(NC(I,J-1).EQ.1) THEN
        CALL LINBY2(-3.5,0.)
```

```
        CALL LINBY2(0.,-6.)
        ELSE
        CALL LINBY2(3.5,0.)
        CALL LINBY2(0.,-6.)
         END IF
      ELSE
        IF(NC(I,J-1).EQ.0) THEN
        CALL LINBY2(-3.5,0.)
        CALL LINBY2(0.,-6.)
        ELSE
        CALL LINBY2(-7.,0.)
        CALL LINBY2(0.,-6.)
        END IF
      END IF
      END IF
  100 CONTINUE
  110 CONTINUE
      DO 130 I=1,NOB
      CALL MOVTO2(155.-5.*I,165.)
      DO 120 J=1,NOB
      IF (NC(I,J).EQ.0) THEN
      CALL CHAHOL(5H  0*.)
      ELSE IF(NC(I,J).GT.0) THEN
      CALL CHAHOL(5H  +*.)
      ELSE
      CALL CHAHOL(5H  -*.)
      END IF
  120 CONTINUE
  130 CONTINUE
      CALL MOVTO2(65.,176.)
      CALL CHASIZ(2.5,2.)
      CALL CHAHOL(9HFIG.(B.*.)
      CALL CHAINT(NOF,2)
      CALL CHAHOL(32H): The bridge functions Bri(i,*.)
      CALL CHAINT(NOS,1)
      CALL CHAHOL(7H,4,t)*.)
      CALL MOVTO2(58.,148.)
      CALL CHAHOL(34HAnd their matrix representation.*.)
      CALL DEVEND
      STOP
 1000 FORMAT(16I3)
 1010 FORMAT(30H ENTER NO. OF BRIDGE FUNCTIONS)
 1020 FORMAT(I3)
 1030 FORMAT(20H ENTER NO. OF SHIFTS)
 1040 FORMAT(I1)
 1050 FORMAT(35H  TYPE: 1 FOR TERMINAL 2 FOR PLOTER)
 1060 FORMAT(I2)
      END
C ****************************************
```

```
C   The following programme is for drawing the waveform of
C   a bridge function bri(i,j,p,t) where i is the order
C   of the bridge function, j is the shift information
C   and p follows from N=2**p where N is the number of
C   bridge functions. In this programme N=32.
C   The programme reads the matrix representation from
C   a file stored in the memory.
      DIMENSION NC(32,32)
      READ(16,1000) ((NC(I,J),J=1,32),I=1,32)
      WRITE(7,1010)
      READ(7,1020) NOB
      WRITE(7,1030)
      READ(7,1040) NOS
      WRITE(7,1050)
      READ(7,1060) MOD
      WRITE(7,1045)
 1045 FORMAT(4H NOF)
      READ(7,1040) NOF
      CALL GINO
      IF(MOD.EQ.1) THEN
      CALL SCALE(.5)
      END IF
      CALL SCALE(.9)
      CALL CHASIZ(1.5,1.5)
      DO 110 I=1,NOB
      CALL MOVTO2(300.-8*I,175.)
      CALL CHAANG(-90.)
      CALL CHAHOL(6Hbri(*.)
      CALL CHAINT(I-1,2)
      CALL CHAHOL(3H,*.)
      CALL CHAINT(NOS,1)
      CALL CHAHOL(7H,5,t)*.)
      CALL MOVTO2(300.-8*I,150.)
      IF (NC(I,1).EQ.0.) THEN
      CALL LINBY2(0.,-4.)
      ELSE
      CALL MOVBY2(3.5,0.)
      CALL LINBY2(0.,-4.)
      END IF
      DO 100 J=2,NOB
      IF (NC(I,J).EQ.NC(I,J-1)) THEN
      CALL LINBY2(0.,-4.)
      ELSE
      IF (NC(I,J).EQ.1) THEN
        IF(NC(I,J-1).EQ.0) THEN
        CALL LINBY2(3.5,0.)
        CALL LINBY2(0.,-4.)
        ELSE
        CALL LINBY2(7.,0.)
        CALL LINBY2(0.,-4.)
        END IF
      ELSE IF(NC(I,J).EQ.0) THEN
        IF(NC(I,J-1).EQ.1) THEN
        CALL LINBY2(-3.5,0.)
```

```
        CALL LINBY2(0.,-4.)
        ELSE
        CALL LINBY2(3.5,0.)
        CALL LINBY2(0.,-4.)
         END IF
      ELSE
        IF(NC(I,J-1).EQ.0) THEN
        CALL LINBY2(-3.5,0.)
        CALL LINBY2(0.,-4.)
        ELSE
        CALL LINBY2(-7.,0.)
        CALL LINBY2(0.,-4.)
        END IF
      END IF
      END IF
  100 CONTINUE
  110 CONTINUE
      CALL MOVTO2(30.,155.)
      CALL CHAHOL(9HFIG.(A.*.)
      CALL CHAINT(NOF,2)
      CALL CHAHOL(32H): The bridge functions bri(i,*.)
      CALL CHAINT(NOS,1)
      CALL CHAHOL(7H,5,t)*.)
      CALL CHAMOD
      CALL DEVEND
      STOP
 1000 FORMAT(32I2)
 1010 FORMAT(30H ENTER NO. OF BRIDGE FUNCTIONS)
 1020 FORMAT(I3)
 1030 FORMAT(20H ENTER NO. OF SHIFTS)
 1040 FORMAT(I1)
 1050 FORMAT(35H  TYPE: 1 FOR TERMINAL 2 FOR PLOTER)
 1060 FORMAT(I2)
      END
      DIMENSION NC(32,32)
      READ(16,1000) ((NC(I,J),J=1,32),I=1,32)
      WRITE(7,1010)
      READ(7,1020) NOB
      WRITE(7,1030)
      READ(7,1040) NOS
      WRITE(7,1050)
      READ(7,1060) MOD
      CALL GINO
      CALL CHASIZ(4.5,3.5)
      CALL SCALE(.95)
      CALL CHAANG(-90.)
      DO 110 I=1,NOB
      IF(MOD.EQ.1) THEN
      CALL SCALE(.7)
      CALL MOVTO2(150.-5*I,180.)
      ELSE
       CALL MOVTO2(300.-6*I,160.)
      END IF
      DO 100 J=1,NOB
```

```
      IF(NC(I,J).EQ.0) THEN
      CALL CHAHOL(3H0*.)
      ELSE IF(NC(I,J).EQ.1) THEN
      CALL CHAHOL(3H+*.)
      ELSE
      CALL CHAHOL(3H-*.)
      END IF
  100 CONTINUE
  110 CONTINUE
      CALL MOVTO2(85.,155.)
      CALL CHASIZ(3.,2.5)
      CALL CHAHOL(42HFIG.(      ):The matrix representation of*.)
      CALL MOVTO2(75.,140.)
      CALL CHAHOL(25HBridge functions BRI(i,*.)
      CALL CHAINT(NOS,1)
      CALL CHAHOL(7H,5,t)*.)
      CALL CHAMOD
      CALL DEVEND
      STOP
 1000 FORMAT(32I2)
 1010 FORMAT(30H ENTER NO. OF BRIDGE FUNCTIONS)
 1020 FORMAT(I3)
 1030 FORMAT(20H ENTER NO. OF SHIFTS)
 1040 FORMAT(I1)
 1050 FORMAT(35H  TYPE: 1 FOR TERMINAL 2 FOR PLOTER)
 1060 FORMAT(I2)
      END
```

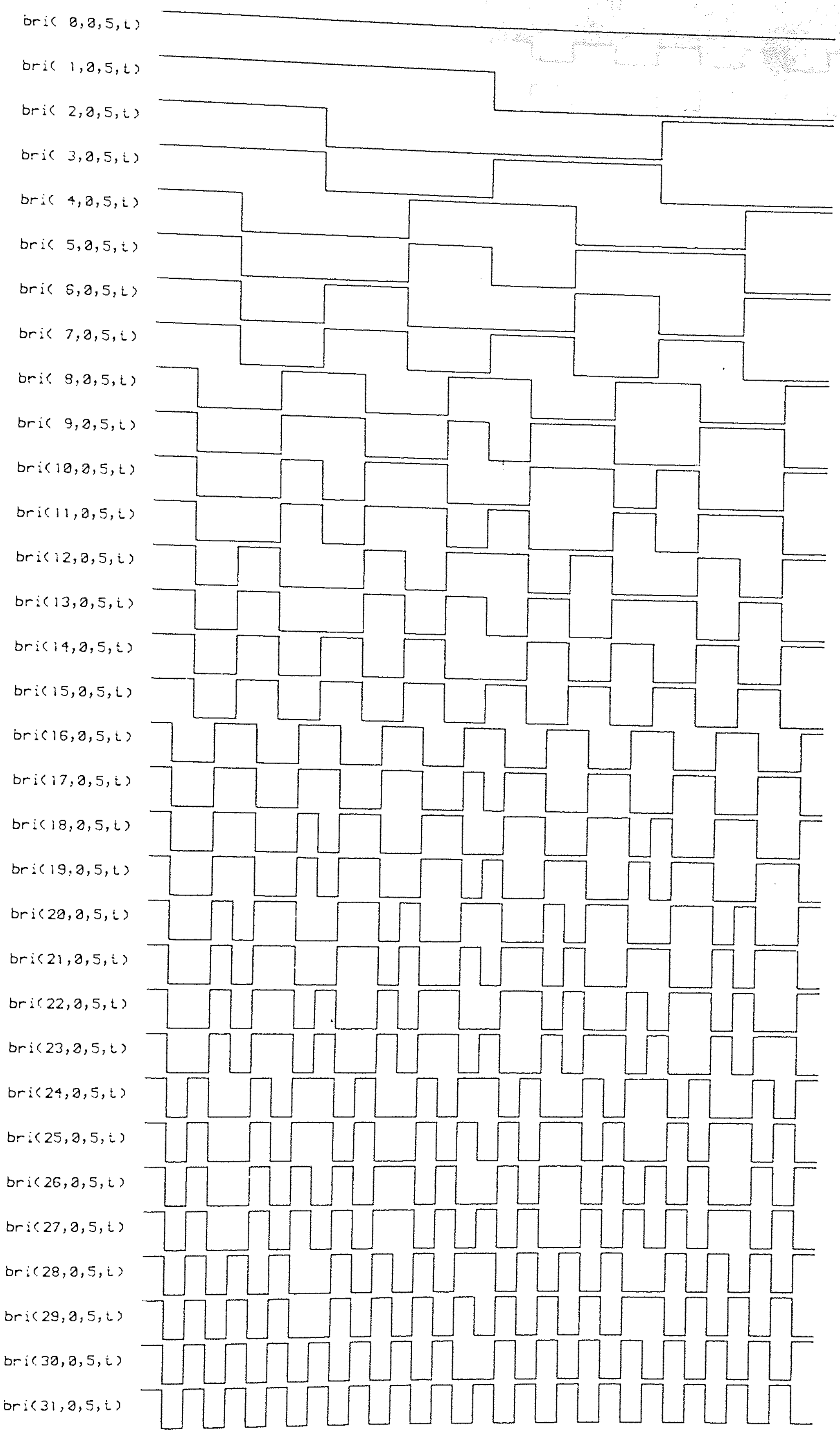


FIG.(A. 0): The bridge Functions bri(i.0,5,t)

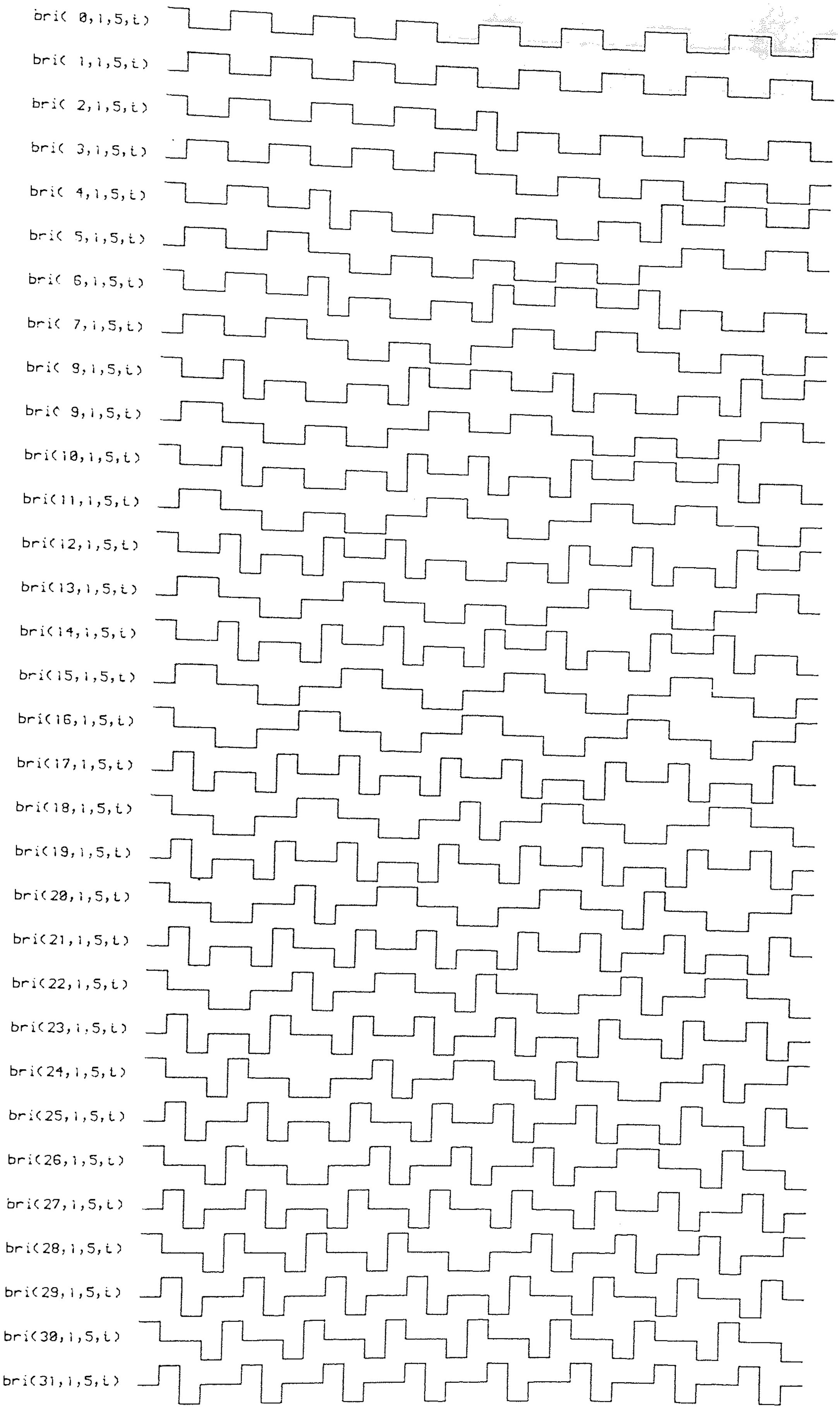


FIG.(A. 1): The bridge Functions bri(i,1,5,t)

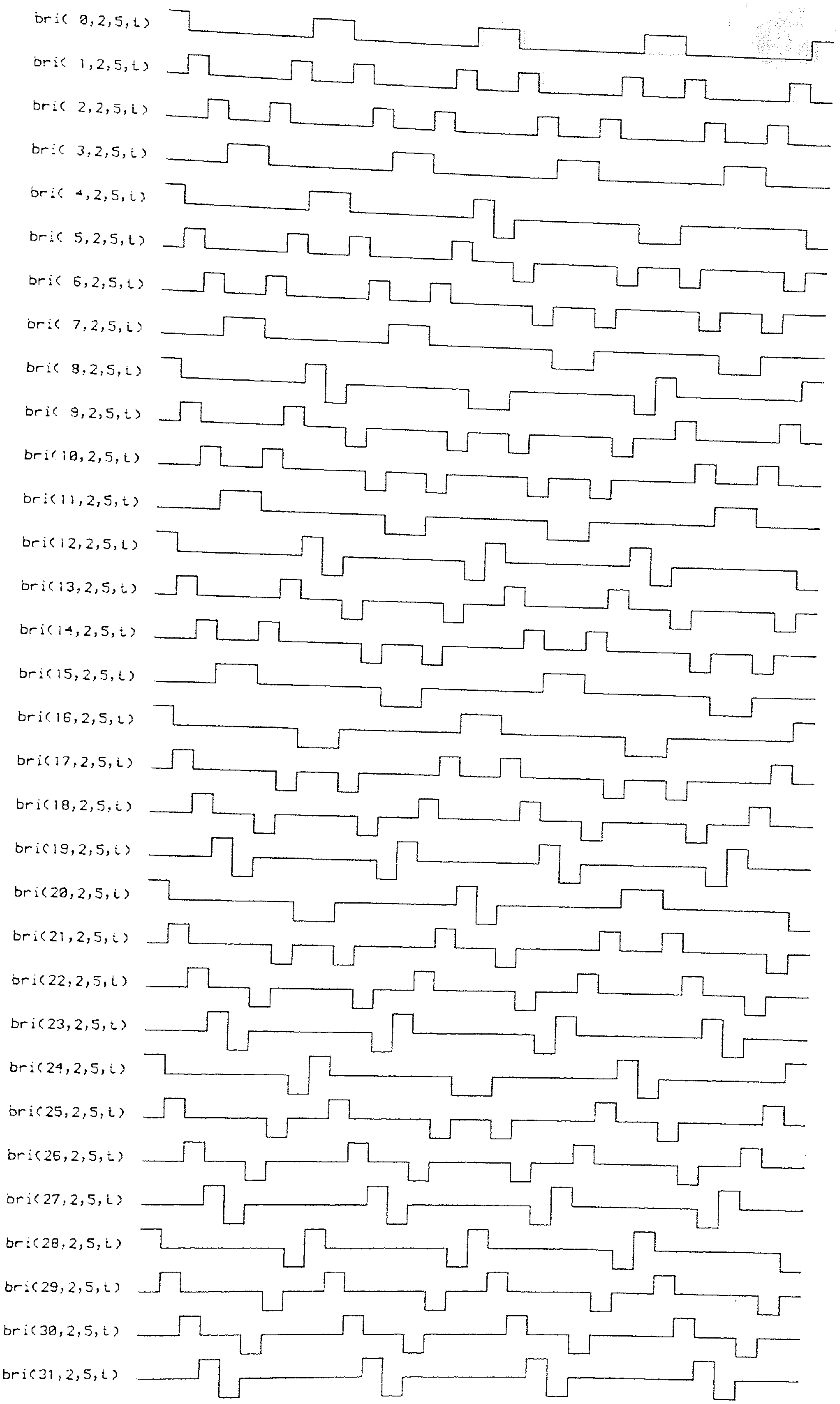


FIG.(A. 2): The bridge Functions bri(i.2,5,t)

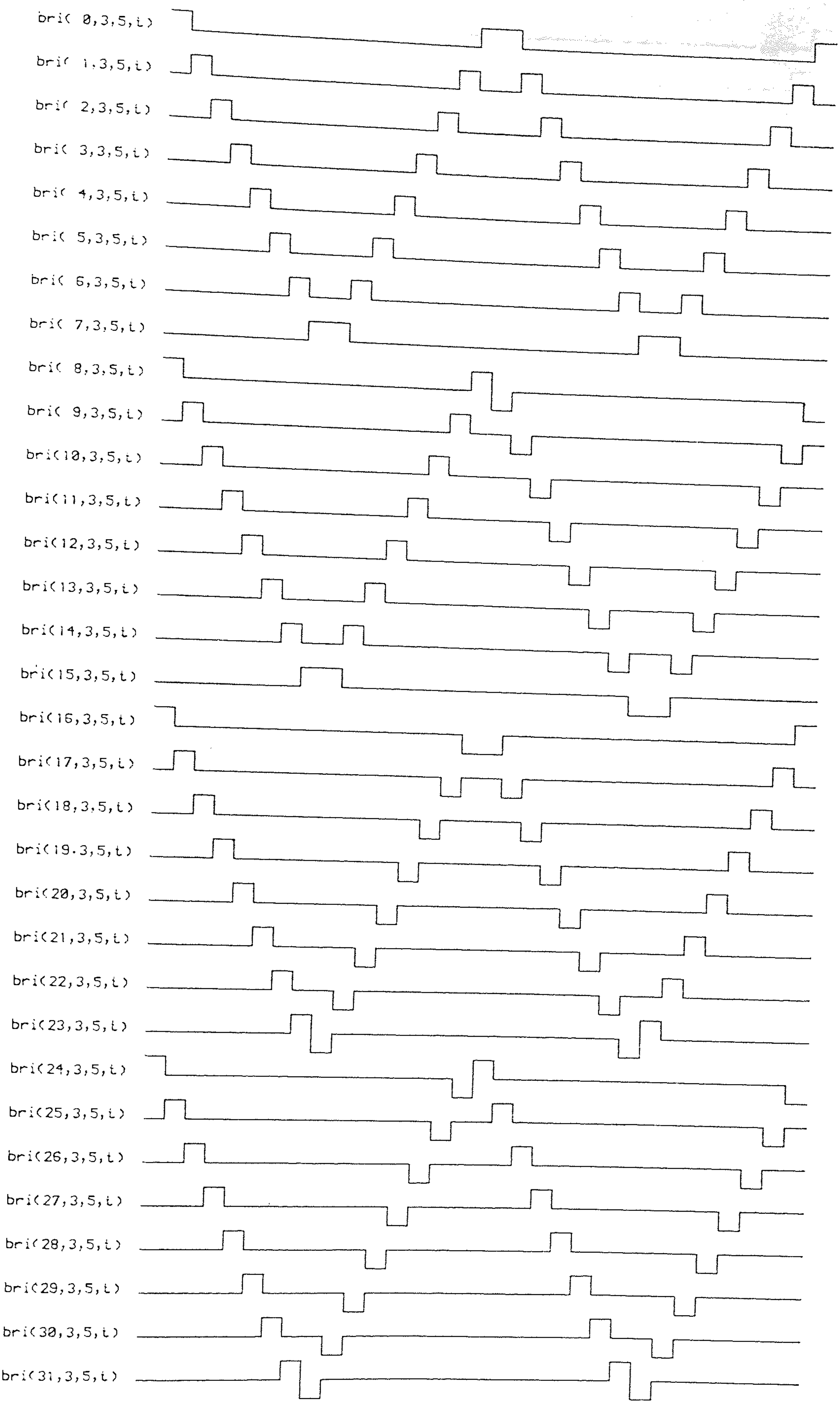


FIG.(A. 3): The bridge functions bri(i,3,5,t)

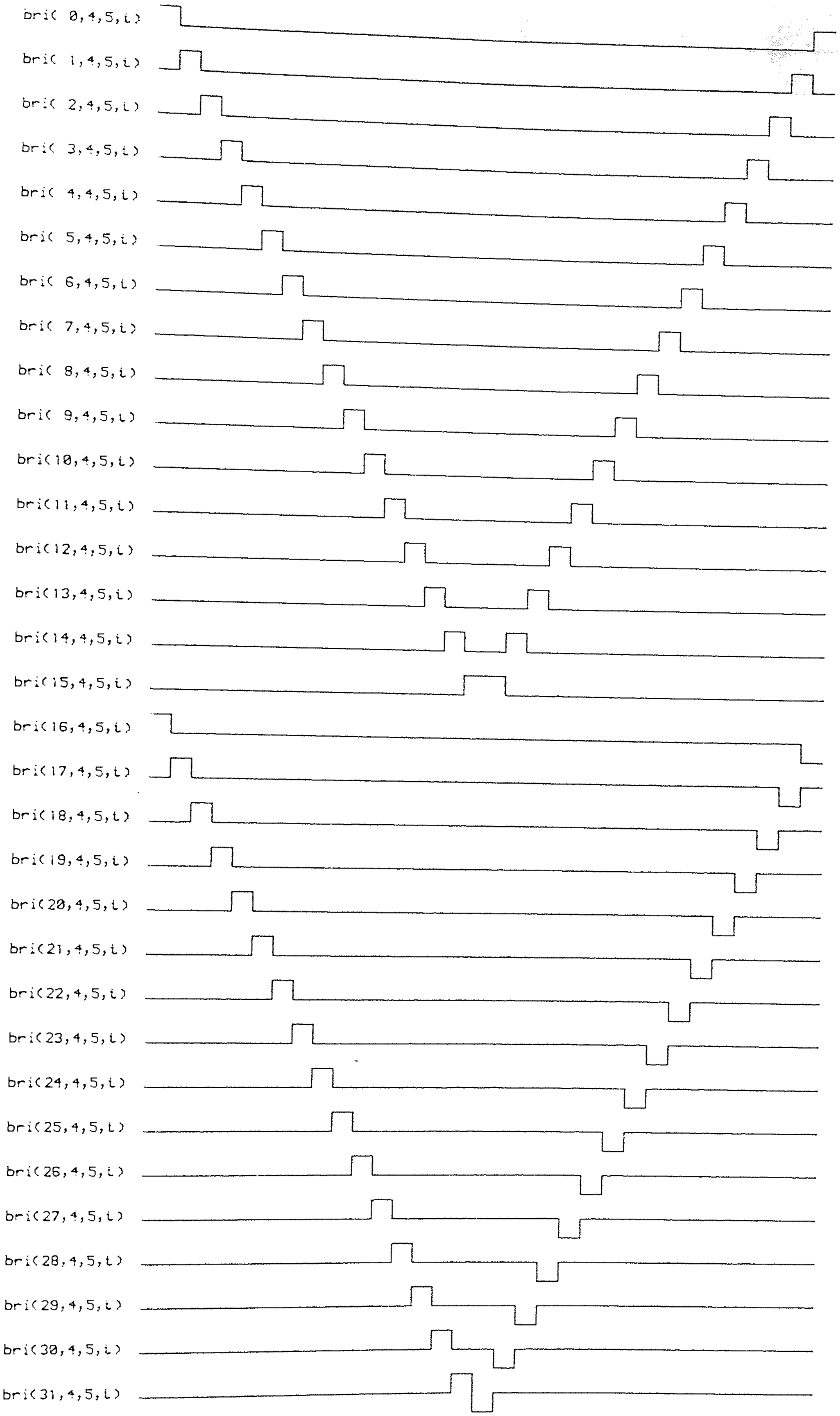


FIG.(A. 4): The bridge Functions bri(i,4,5,t)

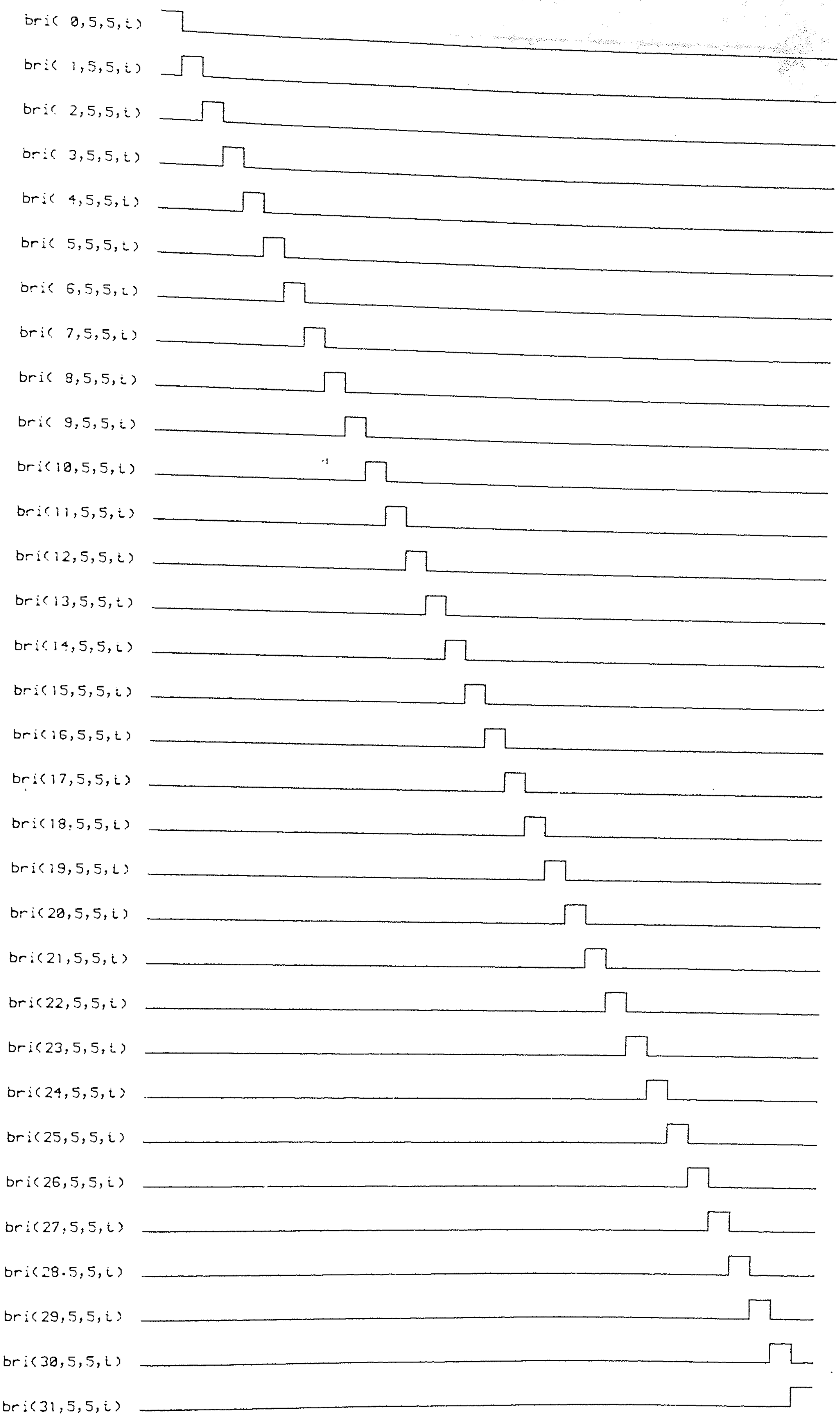
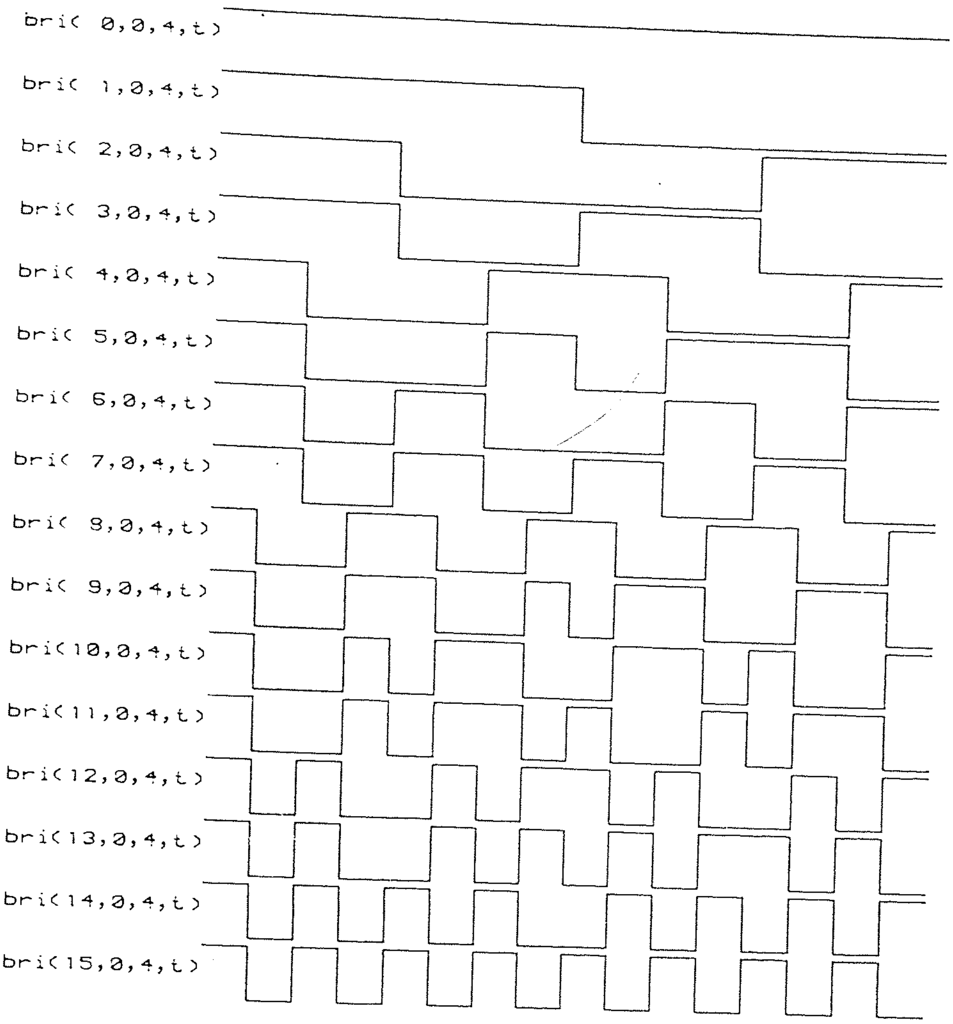


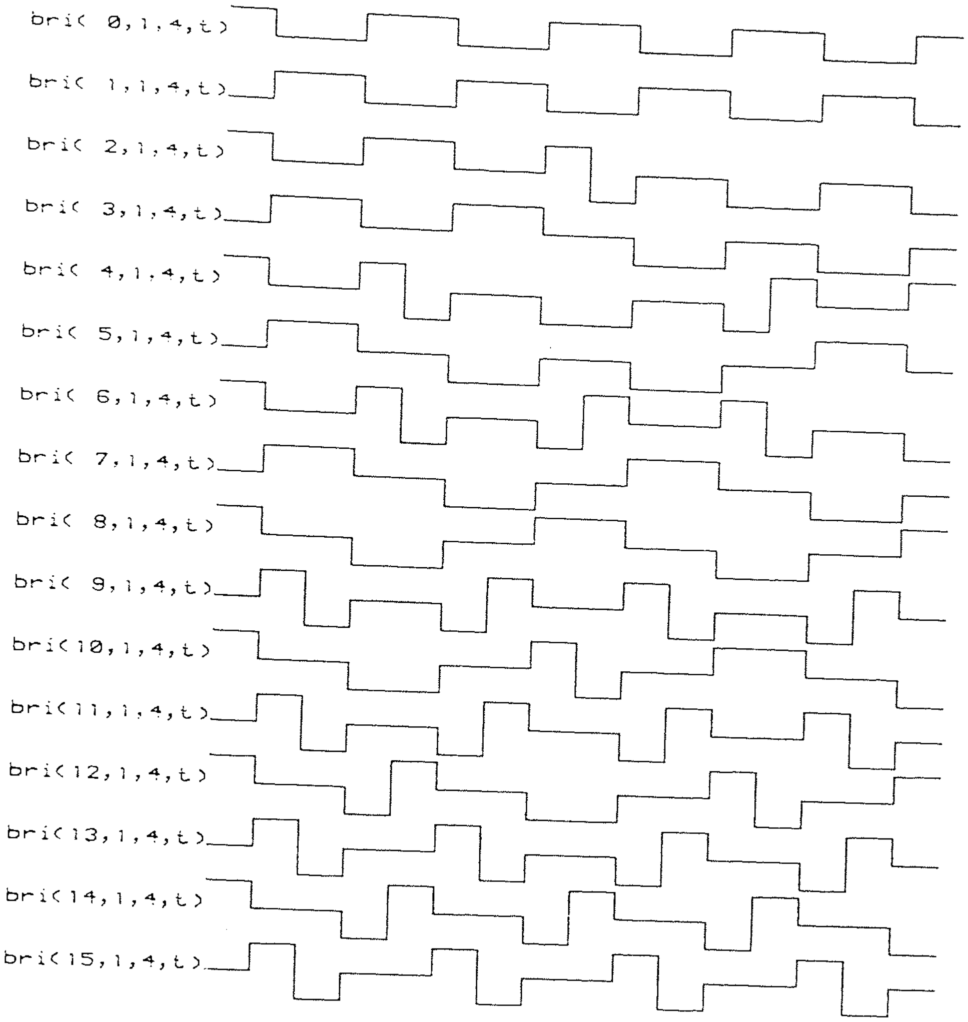
FIG.(A. 5): The bridge Functions bri(i,5,5,t)



$$\begin{matrix}
+ & + & + & + & + & + & + & + & + & + & + & + & + & + & + & + \\
+ & + & + & + & + & + & + & + & - & - & - & - & - & - & - & - \\
+ & + & + & + & - & - & - & - & - & - & - & - & + & + & + & + \\
+ & + & + & + & - & - & - & - & + & + & + & + & - & - & - & - \\
+ & + & - & - & - & - & + & + & + & + & - & - & - & - & + & + \\
+ & + & - & - & - & - & + & + & - & - & + & + & + & + & - & - \\
+ & + & - & - & + & + & - & - & - & - & + & + & - & - & + & + \\
+ & + & - & - & + & + & - & - & + & + & - & - & + & + & - & - \\
+ & - & - & + & + & - & - & + & + & - & - & + & + & - & - & + \\
+ & - & - & + & + & - & - & + & - & + & + & - & - & + & + & - \\
+ & - & - & + & - & + & + & - & - & + & + & - & + & - & - & + \\
+ & - & - & + & - & + & + & - & + & - & - & + & - & + & + & - \\
+ & - & + & - & - & + & - & + & + & - & + & - & - & + & - & + \\
+ & - & + & - & - & + & - & + & - & + & - & + & + & - & + & - \\
+ & - & + & - & + & - & + & - & - & + & - & + & - & + & - & + \\
+ & - & + & - & + & - & + & - & + & - & + & - & + & - & + & -
\end{matrix}$$

FIG.(B. 0): The bridge Functions Bri(i,0,4,t)

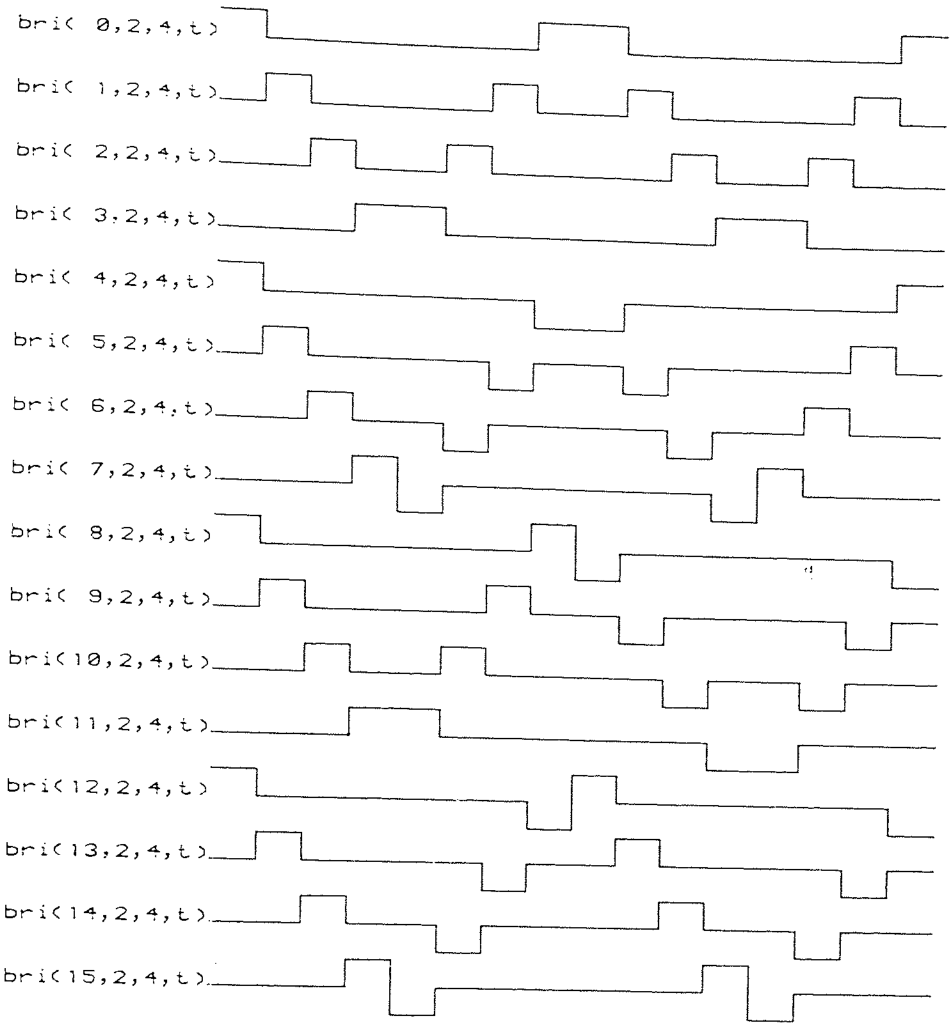
And their matrix representation.



+	0	0	+	+	0	0	+	+	0	0	+	+	0	0	+
0	+	+	0	0	+	+	0	0	+	+	0	0	+	+	0
+	0	0	+	+	0	0	+	-	0	0	-	-	0	0	-
0	+	+	0	0	+	+	0	0	-	-	0	0	-	-	0
+	0	0	+	-	0	0	-	-	0	0	-	+	0	0	+
0	+	+	0	0	-	-	0	0	-	-	0	0	+	+	0
+	0	0	+	-	0	0	-	+	0	0	+	-	0	0	-
0	+	+	0	0	-	-	0	0	+	+	0	0	-	-	0
+	0	0	-	-	0	0	+	+	0	0	-	-	0	0	+
0	+	-	0	0	-	+	0	0	+	-	0	0	-	+	0
+	0	0	-	-	0	0	+	-	0	0	+	+	0	0	-
0	+	-	0	0	-	+	0	0	-	+	0	0	+	-	0
+	0	0	-	+	0	0	-	-	0	0	+	-	0	0	+
0	+	-	0	0	+	-	0	0	-	+	0	0	-	+	0
+	0	0	-	+	0	0	-	+	0	0	-	+	0	0	-
0	+	-	0	0	+	-	0	0	+	-	0	0	+	-	0

FIG.(B. 1): The bridge Functions Bri(i,1,4,t)

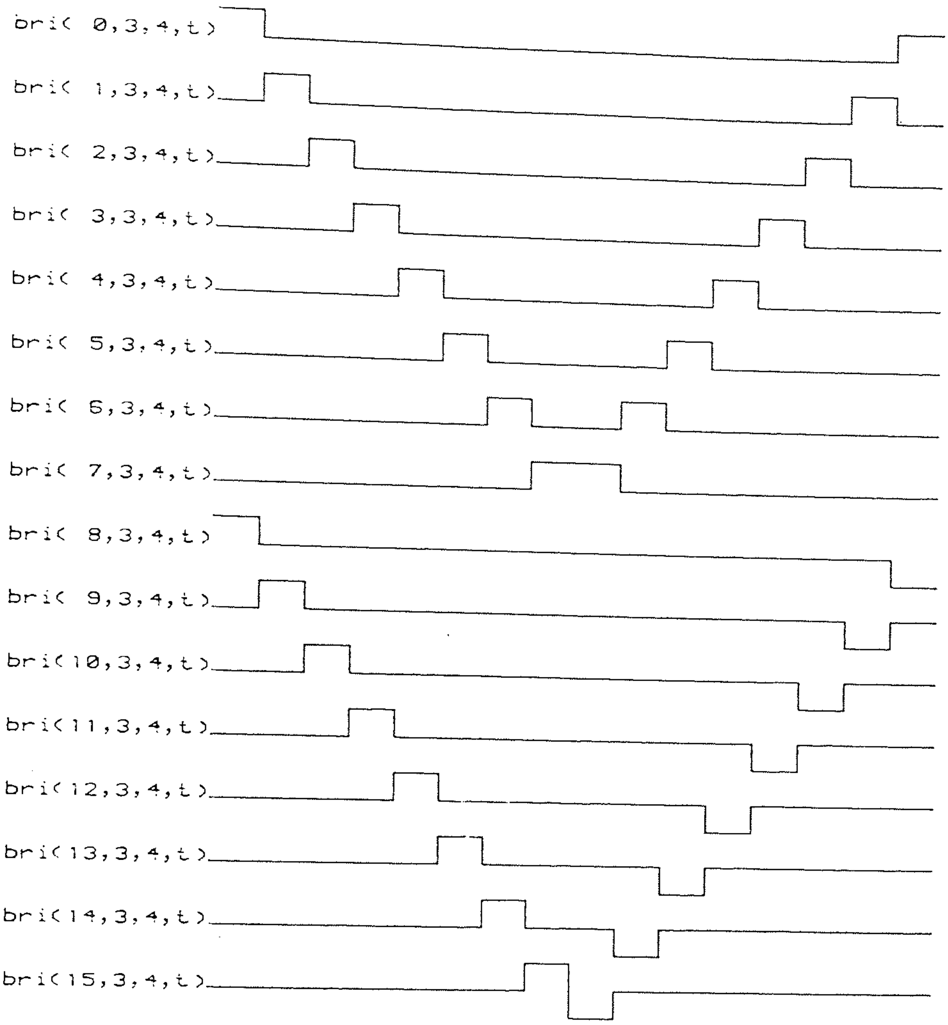
And their matrix representation.



+	0	0	0	0	0	0	+	+	0	0	0	0	0	0	+
0	+	0	0	0	0	+	0	0	+	0	0	0	0	+	0
0	0	+	0	0	+	0	0	0	0	+	0	0	+	0	0
0	0	0	+	+	0	0	0	0	0	0	+	+	0	0	0
+	0	0	0	0	0	0	−	−	0	0	0	0	0	0	+
0	+	0	0	0	0	−	0	0	−	0	0	0	0	+	0
0	0	+	0	0	−	0	0	0	0	−	0	0	+	0	0
0	0	0	+	−	0	0	0	0	0	0	−	+	0	0	0
+	0	0	0	0	0	0	+	−	0	0	0	0	0	0	−
0	+	0	0	0	0	+	0	0	−	0	0	0	0	−	0
0	0	+	0	0	+	0	0	0	0	−	0	0	−	0	0
0	0	0	+	+	0	0	0	0	0	0	−	−	0	0	0
+	0	0	0	0	0	0	−	+	0	0	0	0	0	0	−
0	+	0	0	0	0	−	0	0	+	0	0	0	0	−	0
0	0	+	0	0	−	0	0	0	0	+	0	0	−	0	0
0	0	0	+	−	0	0	0	0	0	0	+	−	0	0	0

FIG.(B. 2): The bridge Functions Bri(i,2,4,t)

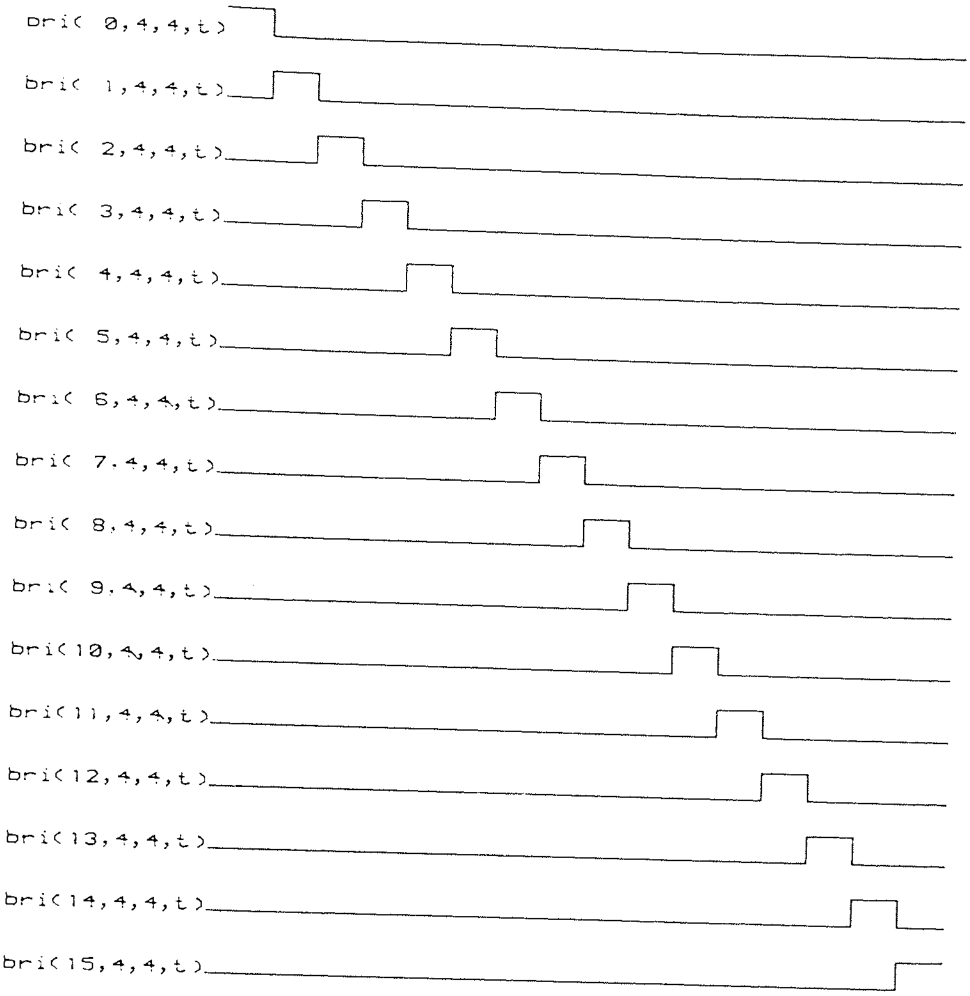
And their matrix representation.



$$
\begin{matrix}
+ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & + \\
0 & + & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & + & 0 \\
0 & 0 & + & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & + & 0 & 0 \\
0 & 0 & 0 & + & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & + & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & + & 0 & 0 & 0 & 0 & 0 & 0 & + & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & + & 0 & 0 & 0 & 0 & + & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & + & 0 & 0 & + & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & + & + & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
+ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & - \\
0 & + & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & - & 0 \\
0 & 0 & + & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & - & 0 & 0 \\
0 & 0 & 0 & + & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & - & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & + & 0 & 0 & 0 & 0 & 0 & 0 & - & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & + & 0 & 0 & 0 & 0 & - & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & + & 0 & 0 & - & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & + & - & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{matrix}
$$

FIG.(B. 3): The bridge functions Bri(i,3,4,t)

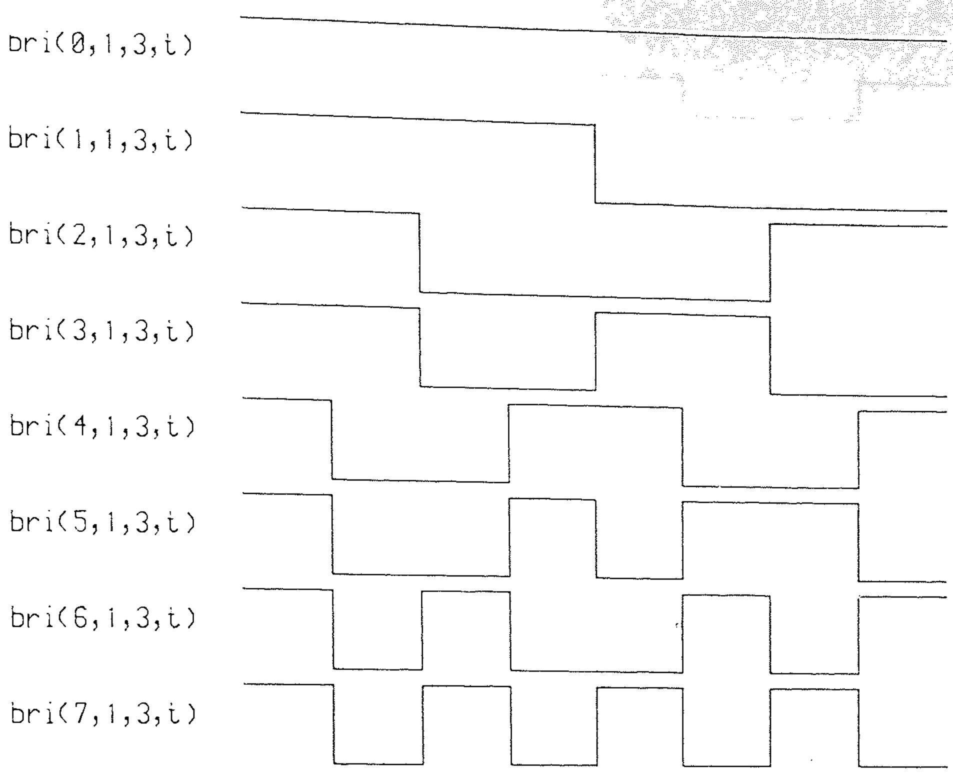
And their matrix representation.



```
+ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 + 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 + 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 + 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 + 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 + 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 + 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 + 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 + 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 + 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 + 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 + 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 + 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 + 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 + 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 +
```

FIG.(B. 4): The bridge Functions Bri(i,4,4,t)

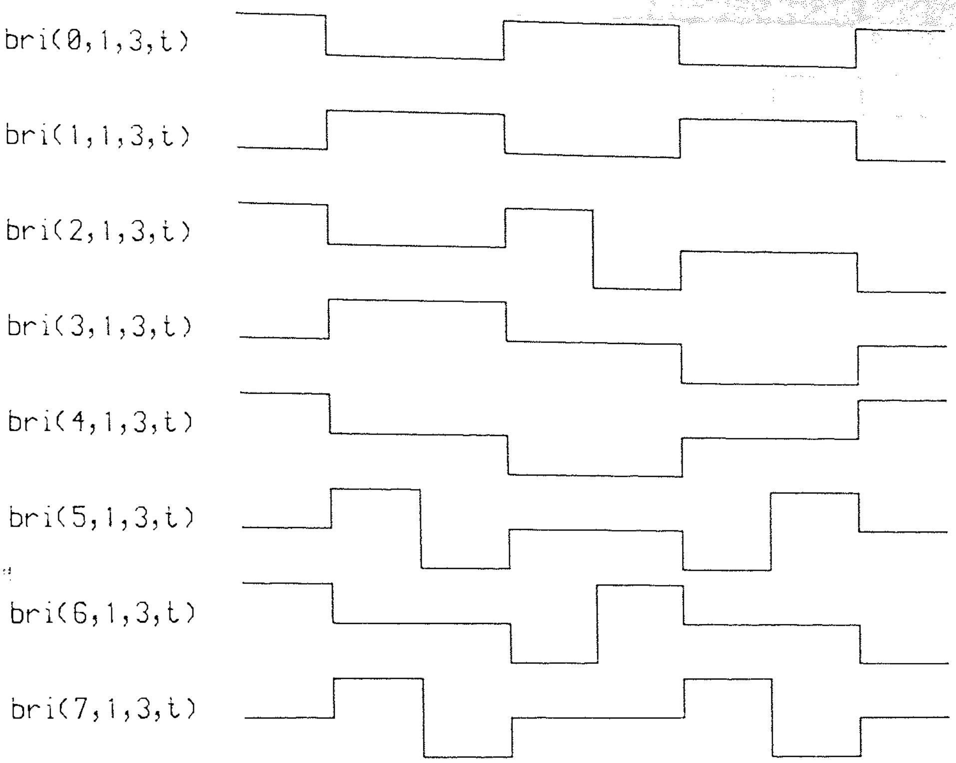
And their matrix representation.



$$\begin{matrix} + & + & + & + & + & + & + & + \\ + & + & + & + & - & - & - & - \\ + & + & - & - & - & - & + & + \\ + & + & - & - & + & + & - & - \\ + & - & - & + & + & - & - & + \\ + & - & - & + & - & + & + & - \\ + & - & + & - & - & + & - & + \\ + & - & + & - & + & - & + & - \end{matrix}$$

FIG.(C.0): The bridge functions bri(i,0,3,t)

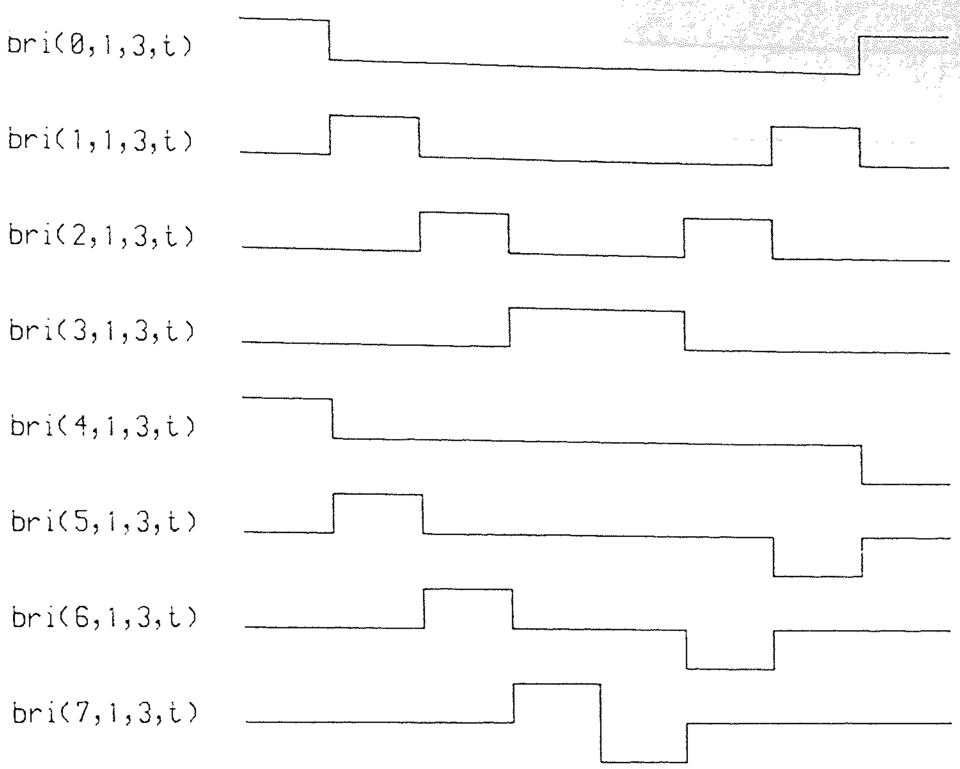
And their matrix representation.



$$\begin{matrix}
+ & 0 & 0 & + & + & 0 & 0 & + \\
0 & + & + & 0 & 0 & + & + & 0 \\
+ & 0 & 0 & + & - & 0 & 0 & - \\
0 & + & + & 0 & 0 & - & - & 0 \\
+ & 0 & 0 & - & - & 0 & 0 & + \\
0 & + & - & 0 & 0 & - & + & 0 \\
+ & 0 & 0 & - & + & 0 & 0 & - \\
0 & + & - & 0 & 0 & + & - & 0
\end{matrix}$$

FIG.(C.1): The bridge functions bri(i,1,3,t)

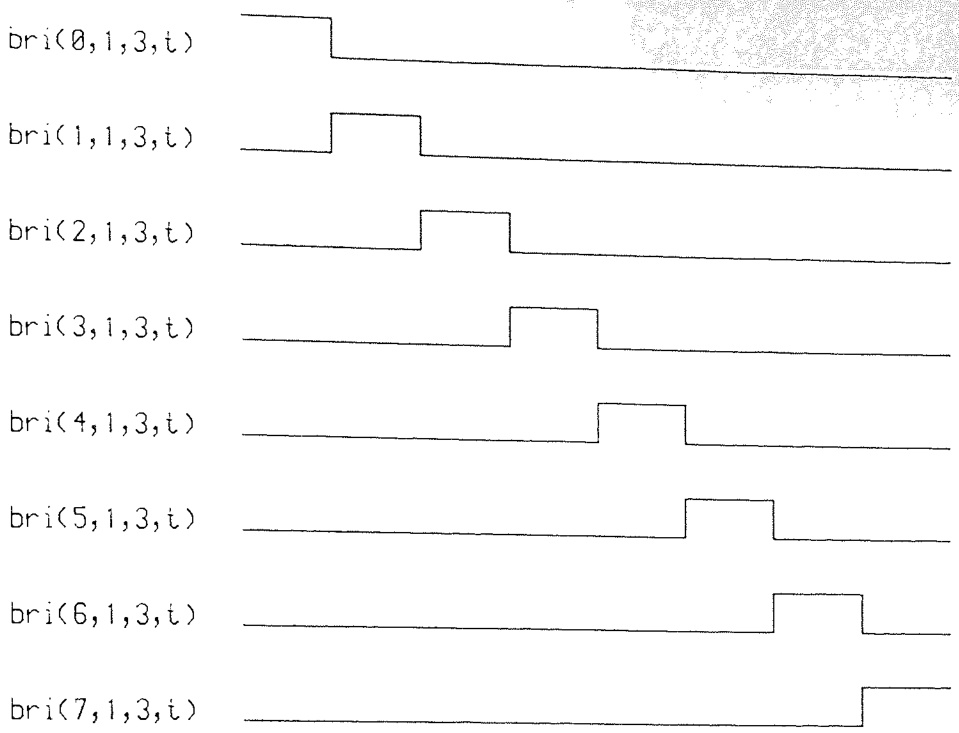
And their matrix representation.



$$\begin{matrix}
+ & 0 & 0 & 0 & 0 & 0 & 0 & + \\
0 & + & 0 & 0 & 0 & 0 & + & 0 \\
0 & 0 & + & 0 & 0 & + & 0 & 0 \\
0 & 0 & 0 & + & + & 0 & 0 & 0 \\
+ & 0 & 0 & 0 & 0 & 0 & 0 & - \\
0 & + & 0 & 0 & 0 & 0 & - & 0 \\
0 & 0 & + & 0 & 0 & - & 0 & 0 \\
0 & 0 & 0 & + & - & 0 & 0 & 0
\end{matrix}$$

FIG.(C.2): The bridge functions bri(i,2,3,t)

And their matrix representation.



$$\begin{matrix}
+ & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & + & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & + & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & + & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & + & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & + & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & + & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & +
\end{matrix}$$

FIG.(C.3): The bridge Functions bri(i,3,3,t)

And their matrix representation.

REFERENCES

REFERENCES

[1] Ballard A.H.,
"New multiplex technique for telemetry",
PROC. of the National Telemetring Conference, paper 6-2
May 1962 Washington, pp. 1-16.

[2] Ballard A.H.,
"Orthogonal multiplexing" The new era of communication
IEEE Space & Aeronautics part 2 nov. 1962, pp. 51-60.

[3] Barrett R.,
"Correlation multiplex data transmission system",
Electronics letters, Vol. 4, No. 24, 29th nov. 1968,
PP. 538-539.

[4] Barrett R., and Gordon J.A.,
"British patent no. 1456 846".

[5] Barrett R.,
"The use of Walsh and other nonsinusoidal functions in
communication signal processing",
1973 Proceedings of theory and applications of Walsh
functions and other nonsinusoidal functions.,
A colloquium June 28-29 1973, pp. 345-367,

[6] Beauchamp, K.G.,
"Walsh functions and their applications",
Academic press, 1975, pp. 20-30 .

[7] Brigham, E.R., Snaith M.J. and Wilock D.M.,
The post office electrical engineers journal vol.69, 1976
pp. 93-102

[8] Chakravarthy C.R.,
"Applications of Walsh functions for multiplex
communication system",
Electro-technology 1962, vol.16, no.2, pp. 94-102.

[9] Chang C. and Liu Z.,
"Telemetry system based on Walsh functions"
PROC. of INT. Telemetring CONF. San Diego CA.
oct. 1980 , pp. 617-623.

[10] Changxin, F.,
"On even active channel majority multiplexing",
IEEE Inter. symposium on Electromagnetic Compatibility ,
1980, pp. 349-353.

[11] Davidson I.A.,
"The multiplexing of signals using orthogonal functions",
Proc. IEEE 1972, International conference on
communication, pp. 38-7,38-10.

[12] Flood J.E,
"Principles of multiplex communication",
Nato advanced study institutes series 271, series E:
Applied sciences, No.12, pp.271-287.

[13] Ghani F., and Ahmad A.,
"A data transmission system with signal processing shared between the transmitter and receiver",
IEEE international symposium on Electromagnetic compatibility 1983, pp. 954-57.

[14] Ghani F., Clark A. and Ahmad A.,
"Pre-processing of signal in digital data transmission",
IERE conference on digital processing of signals on communications Loughborough 1981 pp.119-127.

[15] Gibbs, J.E.,
"Discrete complex Walsh transforms.",
1970 proceedings: applications of Walsh functions, Washington D.C., AD 707431, pp. 106-122.

[16] Glenn A.,
"Code-division multiplex system",
IEEE International convention record vol.12, pt.E, 1964, pp. 53-61.

[17] Goldberg B.G.,
"Code-division multiplexing by frequency shift biphase modulated M-sequence",
IEEE trans. on aerospace and electronic systems, vol.AES-17 no.2, March 1981, pp. 303-304.

[18] Gordon, J.A., and Barrett, R.,
"Correlation-recovered addaptive majority multiplexing",
PROC. IEE VOL.118, NO.3/4, march/april, 1971, pp.417-422.

[19] Gregg W.D.,
"Analog and digital communication", Chapter 10,
John Wiley and sons, newyork 1977.

[20] Haider, I.,
"Internal report on CDM techniques",
1983, Department of Electrical Engineering, Aston university.

[21] Hamsher D.H.,
"Communication system engineering handbook" Chapter 10,
Mc Graw Hill book company 1967.

[22] Harmuth, H.F.,
"Transmission of information by orthogonal functions",
2nd edtion, Springer_Verlag, 1972, pp.22-30.

[23] Harmuth, H.F.,
"SEQUENCY THEORY foundations and applications",
Academic Press, 1977, pp. 46-51 .

[24] Harmuth H.F.,
"Survey of the theory & applications of Walsh functions",
Proc. 5th annual Princeton conference on information sciences & systems 1971, pp. 25-33.

[25] Hashim A.,
"Digital code-division multiplexing",
International Zurich seminar on digital communication 1982, pp. F(5)/1-5.

[26] Insam E.,
"Multiplexing system for Walsh waves",
Electronics letters vol.10, no.11, 30th may 1974. pp. 212-213.

[27] Ince A.N.,
"code-division multiple access and message multiplexing for satallite systems",
International Zurich seminar on digital communication 1980, pp. C7.1-C7-9.

[28] Jones P.S.,
"Code-division-multiplexed delta modulation for the subscriber loop",
"2nd International symposium on subscriber loop and services, may 1976, pp. 148-152.

[29] Lackey R.B.,
"The wonderful world of Walsh functions",
Proc. of IEEE International conference on communication June 1972, pp. 38-1,38-5.

[30] Leakey D.M.,
"New techniques in multiplexing" alternative to FDM&TDM,
GEC journal of science and technology vol.40, no.2, 1973, pp. 59-64.

[31] Lindner J.,
"the performance of code-division multiplexing with pulse position modulation",
Agard conference proceedings no. 239, 5-9th june 1978, pp. 35/1-11.

[32] Maseng T.,
"performance analysis of majority logic multiplex system"
IEEE trans. on communication vol. COM-28, no.9, 1980, pp. 1675-1680.

[33] Mukherjee A., and Mokhopadhyah D.,
"A method for increasing the number of majority multiplexed channels",
Proc. IEEE vol.66, no.9, Sep. 1978, pp.1096-1097.

[34] Pearce T.H.,
"Multiplexing techniques",
SERT journal vol.2, june 1977, pp. 136-139.

[35] Qishan Z., Mingrui Z. and Yaokun L.,
"A sequency-divisio multiplex system",
Proc. of International telemetring conference 1981, pp.1363-1368.

[36] Qishan Z., and Zhihua L.
"New orthogonal multiplexing system",
International telemetring conference 1982, pp.175-183.

[37] Rajan J.A.,
"Adaptive aquisition of multiple access codes",
IEEE international conference on communications, ICC 82, the digital revolution pt.A, vol.3(7E 1/1-4), 1982.

[38] Sansone, G.,
"Orthogonal functions"
Interscience publishers, Inc., 1959

[39] Schmidt W.,
"The application of TDMA to intelsat satellite services",
COMSAT technical review vol.3, no.2, 1973, pp.257-275.

[40] Schneider K.,
"Optimum demodulation of code-division multiplex signals"
National telecommunications conference record, pt.II, 20.2/1-4, 1978.

[41] Schneider K.,
"New approaches to code-division multiplexing",
IEEE Canadian communications and power conference, 1978, Montreal, pp.75-8

[42] Schneider K.,
"Optimum detection of code-division multiplex signals",
IEEE trans. on Aerospace and electronic systems,
Vol. AES-15, no.1, Jan. 1979, pp. 181-185.

[43] Schriber H. H.,
"Theory and applications of sequency multiplexing",
1973 Proceedings of theory and applications of Walsh functions and other nonsinusoidal functions,
A colloquium 28-29 June 1973, pp. 317-344,
The Hatfield Polytechnic, Hatfield, UK.

[44] Stanley, William D.,
"Electronic communication systems", chapter 6
Reston publishing company 1982.

[45] Stein S. and Jones J.,
"Modern communication principles with application to digital signaling", Chapter 8 (multiplexing techniques),
Mc Graw-Hill book company 1967.

[46] Swick, D.A.,
"Generation of Walsh functions",
IEEE Tran. on Inform. theory, VOL. IT-15, NO.1, pp.167-168, 1969

[47] Titsworth, R.C.,
"A Boolean-function-multiplexed-telemetry data system"
IEEE trans., 1963, SET-9, pp. 42-45 .

[48] Walsh, J.L.,
"A closed set of orthogonal functions",
Amer. J. Math. 45, 1923, pp. 5-24.

[49] Zheng H. and Youwei Y.,
"An addaptive multiplex delta modulation system",
IEEE international symposium on EMC 1980, pp. 346-348.

[50] Zhihua , Li and Qishan ,Zhang,
"Ordering of Walsh functions",
IEEE Trans. on Electromagnetic Compatibility, VOL. EMC-25 , No.2, 1983, pp. 115-119.

[51] Zhihua, L. and Qishan, Z.,
"Introduction to Bridge functions",
IEEE trans. on Electromagnitic Compatibility, VOL. EMC-25 , NO.4, NOV. 1983, pp. 459-464.

BIBLIOGRAPHY

BIBLIOGRAPHY

1. Ahmed N.,
"Walsh functions: some properties and applications"
IEEE Int. symposium on electromagnetic compatibility 1980
pp. 275-281.

2. Aronstein R. H.,
"Comparison of orthogonal and block codes"
PROC. IEEE Vol. 110, No.11, Nov. 1963, pp.1965-1966.

3. Berlekamp, E. R.,
"Algebraic coding theory",
Mc Graw-Hill, New York 1968.

4. Besslich P. W.,
"Walsh functions generators for minimum orthogonality error",
IEEE Tran. on electromagnetic compatibility, Vol. EMC-15,
No. 4, Nov.1973, pp. 177-179.

5. Bramhall J.N.,
"Bibliography on Walsh & Walsh related functions",
1973 proceedings of theory and applications of Walsh functions and other nonsinusoidal functions, pp.417-460.

6. Carey M. J., Constantinides A. G.,
"Cross-talk in multiplex systems using Walsh functions",
PROC. IEE, Vol.125, No.12, 1978, pp. 1329-1335.

7. Chaiko K.I.,
"The use of Walsh functions for data transmission over channels with short-term interruptions",
Trans. in Telecommun. and Radio Eng., part 2, Vol.35,
No.1, 1980, pp.79-83.

8. Clark, A. P.,
"Principles of digital data transmission",
Pentech Press 1976.

9. Clark, A. P.,
"Advanced data transmission systems",
Pentech press, London 1977.

10 Coates R. F. W.,
"Modern communication systems", 2nd edition,
The Macmillan press Ltd., 1982.

11. Feldman, C.B., and Bennett, W.R.,
"Bandwidth and transmission performance",
The Bell System Technical Journal, Vol.28, pp. 440-595,
July 1949.

12. Fernandez L. C., Rao K. R.,
"Design of a synchronous Walsh function generator",
IEEE Trans. on elecromagnetic compatibility, Vol. EMC-19,
No.4, 1977, pp. 407-410.

13. Ferrel G. S.,
"Introduction to communication systems",
Addison-Wesley publishing company 1982.

14. Figueiras Vidal A. R.,
"Interpulse interference effects on orthogonal division multiplexing systems",
2 nd symposium on EMC, 28-30 June 1977, pp. 31-34.

15. Figueiras Vidal A. R.,
"Noise effects on orthogonal division multiplexing systems.",
2 nd symposium on EMC, 28-30 June 1977, pp.35-36.

16. Gaubatz D. A. and Kitai R.,
"A programable Walsh function generator for orthogonal sequency pairs",
IEEE Trans. on EMC. May 1974, pp. 134-136.

17. Ghani F,
"Serial digital communication systems with signal arranged in orthogonal sets",
Ph.D. thesis, Loughborough University, March 1974.

18. Girasimov I. and Smolov V.,
"Two-dimensional Bridge functional transforms of time-pulse type", IEEE
Trans. on automation & remote control vol.38, no.1, pt.2
pp. 120-126.

19. Glick D., Goldberg R., and Mushinsky B.,
"Providing telco customers continuous data services",
Telephony nov. 1983, pp. 46-54.

20. Harris F.,
"The DFT applied to time-domain signal processing",
IEEE communication magazine May 1982, pp. 13-22.

21. Harmuth H. F.,
"Orthogonal codes",
The Institute of Electrical Engineers, Monograph No. 369E
March 1960, pp. 242-248.

22. Harmuth H. F.,
"Applications of Walsh functions in communications",
IEEE Spectrum Nov. 1969, pp. 82-91.

23 Haykin S.,
"Communication systems", 2nd edition,
John Wiley & sons 1983.

24. Helm H. A., Bellcomm, Inc., Washington D. C.,
"Group codes and Walsh functions",
Applications of Walsh functions, 1971 symposium proceedings, pp. 78-83.

25. Hubner H.,
"Multiplex systems using sums of Walsh functionsas carriers",
Walsh functions symposium, Washington 1971, pp. 180-191

26. Jullien, G. A.,
"Digital spectrum analysis",
short course in digital signal proccessing at the University of Aston in Birmingham, Sep. 1978.

27. Kaitai R. and Yuen C.K.,
"Walsh functions generators",
1973 proceedings of theory and applications of Walsh functions and other nonsinusoidal functions, pp. 297-315.

28. Lathi B. P.,
"Modern digital and analog communication systems",
Holt, Rinehart and Wiston CBS college publishing 1983.

29. Losev V. V.,
"Digital signals based on Walsh's function for a multichannel data-transmission system",
Radio engineering and electronic physics, Vol 24, No. 11, 1979, pp. 42-45.

30. Lynn P. A.,
"The analysis and processing of signals", 2nd edition,
The Macmillan press Ltd., 1982.

31. Magnuski, H.,
"Address communications systems", Chapter 18 from
Communication system engineering edited by Hamsher D. H.,
Mc Graw-Hill New York 1980.

32. Manassewitch, V.,
"Frequency synthesizers theory and design",
John Wiley, New York, 1980.

33. Marshall, C. J.,
"Principles of digital communications",
Mc Graw-Hill, New York, 1980.

34. Miller G. M.,
"Modern electronic communication", 2nd edition,
John Wiley and sons 1983.

35. Moharier P.,
"Walsh-Fourier & trigonometric secrecy coding",
Journal of Institution of electronics & telecommunication engineers vol.20, no.5, 1974.

36. Neto I., Dimond K. R.,
"D* functions generator",
Electronics Lett. Vol. 11, No.19, 1975, pp. 461-462.

37. Rabiner, L. R. and Gold, B.,
"Theory and applications of digital signal processing",
Printice Hall, New Jersy 1975.

38. Ridinbo G. R. and Cheung W. Y.,
"Signal processing techniques in error control systems",
ICASSP 80, PROC. IEEE 1980, pp. 968-973.

39. Roddam T.,
"The function of functions",
Wireless world Vol.87, No. 1551 Dec. 1981 pp. 36-39.

40. Roddam T.,
Wireless world Vol.87, no. 1552 Jan. 1982 pp. 47-49.

41. Rosenfeld Y.,
"Universality of the Bridge functions and a semiempirical freezing criterion in 2&3 dimensions"
Pysical review, vol. 24, no.5, nov. 1981.

42. Sannino M.,
"Multiple-output Walsh function generation for minimum orthogonality error",
IEEE Trans. on Instrumentation and measurement, Vol. IM-27, No.1, March 1978, pp. 29-32.

43. Schwartz M.,
"Information transmission, Modulation and noise",
3rd edition, Mc Graw-Hill 1981.

44. Schwartz, M., Bennett, W. R. and Stein, S.,
"Communication systems and techniques",
Mc Graw-Hill, New York 1966.

45. Schiling, J.,
"Principles of communication systems",
Mc Graw-Hill, New York 1971.

46. Scolaro R. J.,
"Error probabilities for orthogonal and multiplex systems in the presence of intersymbol interference and Gaussian noise", IEEE Trans. on Commun. Vol. Com-25, No.5, 1977, pp. 549-557.

47. Shanmugam K. S.,
"Digital and analog communication systems",
John Wiley and sons 1979.

48. Shannon, C. E.,
"Communication in the presence of noise",
PROC. IRE, Vol. 37, pp. 10-12, Jan. 1949.

49. Shore J. E.,
"On the application of Haar functions",
IEEE Trans. on Commun. Vol. COM-21, No.3, 1973,
pp. 209-216

50. Suran J.and Marolf R.,
"Integrated circuits and integrated systems",
Proc. of IEEE, pp.1661-1668.

51. Thomson K. and Rao K.,
"Analyzing a biorthogonal information channel by the Walsh-Hadamard transform",
Computer & Electronic engineer vol.4, no.2, pp.119-132,
1977.

52. Yuen C. K.,
"New Walsh function generator",
Electronics Lett. 7th Oct. 1971 Vo. 7 No. 20, pp.605-607

53. Yuen C. K.,
"Interchannel interference in Walsh multiplex systems",
IEEE Trans. on Commun. Vol. COM-22, No.2, Feb. 1974,
pp. 255-261.

54. 1970 Proceedings of applications of Walsh functions,
Nat. Techn. Info. service AD-707 431 US Department of
Commerce Springfield, Verg., 22151, USA.

55. 1971 Proceedings of applications of Walsh functions,
Nat. Techn. Info. service AD-727 000 US Department of
Commerce Springfield, Verg., 22151, USA.

56. 1971 Proceedings of theory and applications of Walsh functions, A symposium June 29-30 1971, the Hatfield polytechnic, Hatfield, Herts. UK. Library service.

57. 1972 Proceedings of applications of Walsh functions,
Nat. Techn. Info. service AD-744 650 US Department of
Commerce Springfield, Verg., 22151, USA.

58. 1973 Proceedings of applications of Walsh functions,
Nat. Techn. Info. service AD-763 000 US Department of
Commerce Springfield, Verg., 22151, USA.

59. 1973 Proceedings of Theory and applications of Walsh functions and other nonsinusoidal functions.
A colloquium June 28-29 1973.
The Hatfield Polytechnic, Hatfield, Herts. UK library service.