# THE OPTIMUM QUANTIZATION PROCESS

by

JAMES K. ALBRIGHT

B.S., Kansas State University, 1979

---

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Electrical Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1982

Approved by:

Major Professor

TABLE OF CONTENTS

| | | Page |
|---|---|---|
| Chapter I: | Introduction | 1 |
| Chapter II: | The Optimum Quantizer | 2 |
| Appendix 2.1: | Conditions to Ensure Mimimization | 8 |
| Chapter III: | The Optimum Gaussian Quantizer | 10 |
| Appendix 3.1: | Newton's Method for Finding the Zeroes of Functions | 36 |
| Appendix 3.2: | The Secant Method of Determining the Zeroes of Functions | 38 |
| Appendix 3.3: | Program Listing, TABLGEN5 | 40 |
| Chapter IV: | Experimental Results | 47 |
| Appendix 4.1: | Program Listing, NORMGEN | 70 |
| Appendix 4.2: | Program Listing, QUANTIZE1 | 72 |
| Chapter V: | Program Usage | 80 |
| Chapter VI: | Conclusions and Recommendations | 86 |
| References | | 93 |

## LIST OF FIGURES


| Figure | | Page |
|---|---|---|
| 2.1 | Output Levels, General Distribution | 3 |
| 2.2 | Even/Odd Number of Output Levels, Symmetric Distribution | 5 |
| 3.1 | Output Levels, Symmetric Distribution | 11 |
| 3.2 | Infinite Series Convergance | 15 |
| 3.3 | The Optimum Quantizer | 18 |
| 3.4--3.11 | Input/Output Transfer Functions | 23 |
| 3.12 | Mean Square Error | 33 |
| 3.13 | Entropy | 34 |
| 3.14 | Flowchart: The Optimum Quantizer | 41 |
| 4.1 | The Optimum Quantization Process | 48 |
| 4.2--4.9 | Example 1 | 51 |
| 4.10--4.17 | Example 2 | 61 |
| 4.18 | Flowchart: The Optimum Quantization Process | 73 |
| 5.1 | TABLGEN5 Input Sequence | 81 |
| 5.2 | TABLGEN5 Output Sequence | 82 |
| 5.3 | QUANTIZE1 Input/Output Sequence | 85 |
| 6.1 | Summary Input/Output Sequences | 87 |
| 6.2 | Summary Input/Output Histograms | 89 |
| 6.3 | Image Processing Scheme | 91 |
| 6.4 | Zonal Coding | 92 |

## LIST OF TABLES


| Table | | Pag |
|---|---|---|
| 3.1 | Initial Values . . . . . . . . . . . . . . . . . . . . . . . | 13 |
| 3.2 | The Optimum Gaussian Quantizer Parameters . . . . . . . . . . . | 19 |
| 3.3 | Mean Square Error and Entropy . . . . . . . . . . . . . . . . . | 32 |
| 4.1 | Experimental vs. Theoretical Mean Square Error, Example 1 . . | 59 |
| 4.2 | Experimental vs. Theoretical Mean Square Error, Example 2 . . | 69 |

# CHAPTER I

## INTRODUCTION

In practice it is impossible to encode a given data sequence exactly, since data words with infinite lengths would be required. However, very accurate representations of data sequences are achieved in digital computers by resorting to floating-point arithmetic. In contrast, accuracy presents serious problems when one is restricted to using fixed-point arithmetic, as is the case in special purpose hardware. Since fixed-point arithmetic offers faster execution time and more economical hardware, methods have been sought to encode data sequences in an efficient manner using fixed-point arithmetic. One such method is attributed to Max [1]. This method is optimum when the distortion measure is the mean square error criterion, and the data sequence to be encoded can be described by a gaussian random variable.

The purpose of this report is to develop a set of computer programs that implement Max's method on the computer system in the Electrical Engineering Department's Signal Processing Laboratory. The motivation for doing so is that the computer programs can be readily used for encoding purposes in applications such as the compression of image and speech data.

Chapter II is devoted to an introduction of Max's method, while its relevance to encoding gaussian random variables is addressed in Chapter III. Illustrative examples are included along with the computer program documentation presented in Chapter IV. Chapter V supplements the documentation with a user's guide. Some conclusions and recommendations for future work are summarized in Chapter VI.

# CHAPTER II

## THE OPTIMUM QUANTIZER

A quantizer enables one to encode a given discrete-time sequence of numbers, $x(n)$, in binary form. Why would anyone want to quantize? It is because only a finite number of bits are available to encode a given sequence. Thus, in practice, it is impossible to encode $x(n)$ exactly. As a consequence, efficient quantization techniques are sought to use a finite number of bits to encode $x(n)$ in such a way that the resulting digital sequence $\hat{x}(n)$ is a faithful reconstruction of $x(n)$. An optimum quantizer is one which produces minimum distortion with respect to some criterion--e.g., the mean square error. The objective is that the output sequence from the quantizer is as close a representation of the input sequence as possible.

For convenience, we let the input to the quantizer be a continuous random variable X; i.e., X takes a continuum of values which we wish to quantize into a finite set of discrete values $Y_i$. In practice, the entire range of possible input values is broken up into smaller subranges. Each subrange has a lower limit, $X_i$ and upper limit $X_{i+1}$. Any input X between these two limits is mapped into corresponding output $Y_i$. We consider the cost function

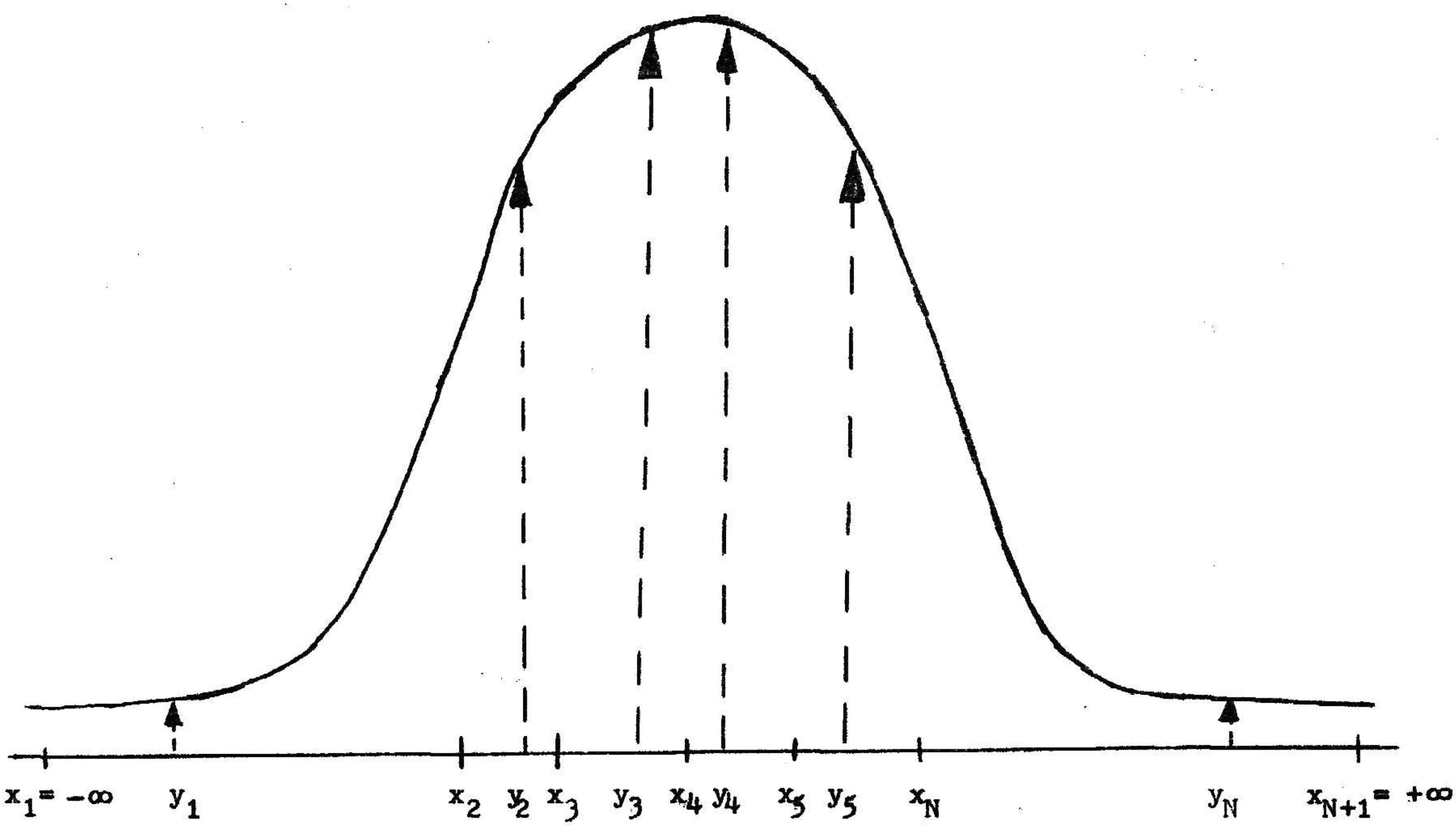
$$E\ ((x-y)^2) = \sum_{i=1}^{N} \int_{x_i}^{x_{i+1}} (x-y_i)^2 p_X(x)\,dx \qquad (2\text{-}1)$$

where $p_X(x)$ denotes the probability density function of X, and N is the number of finite output levels. In the most general case $x_1 = -\infty$ and $x_{N+1} = +\infty$, as illustrated in Figure 2.1. Simplifying for a symmetric distribution, $p_X(x)$, such as the gaussian distribution, $x_{N+1}$ still equals $+\infty$ but

THIS BOOK CONTAINS NUMEROUS PAGES WITH DIAGRAMS THAT ARE CROOKED COMPARED TO THE REST OF THE INFORMATION ON THE PAGE.

THIS IS AS RECEIVED FROM CUSTOMER.

Note that the input abscissa ($x_i$) is continuous, while the output abscissa ($y_i$) is discrete.




OUTPUT LEVELS, GENERAL DISTRIBUTION

FIGURE 2.1

For N EVEN
$$\begin{cases} x_{\frac{N}{2}+1} = 0 \\ y_{\frac{N}{2}-i} = -y_{\frac{N}{2}+(i+1)} \\ x_{\frac{N}{2}-(i-1)} = -x_{\frac{N}{2}+(i+1)} \end{cases}$$

For N ODD
$$\begin{cases} y_{\frac{(N+1)}{2}} = 0 \\ x_{\frac{(N+1)}{2}+i} = -x_{\frac{(N+1)}{2}-(i-1)} \\ y_{\frac{(N+1)}{2}+i} = -y_{\frac{(N+1)}{2}-i} \end{cases}$$

as depicted in Figure 2.2. Equation (2-1) can also be written as

$$E((X-Y)^2) = \sum_{j=2}^{N+1} \int_{x_{j-1}}^{x_j} (x-y_{j-1})^2 p_X(x)\,dx$$

$$= \int_{x_1}^{x_2} (x-y_1)^2 p_X(x)\,dx + \int_{x_2}^{x_3} (x-y_2)^2 p_X(x)\,dx + \dots$$

$$+ \int_{x_{j-1}}^{x_j} (x-y_{j-1})^2 p_X(x)\,dx + \int_{x_j}^{x_{j+1}} (x-y_j)^2 p_X(x)\,dx + \dots$$

$$+ \int_{x_N}^{x_{N+1}} (x-y_N)^2 p_X(x)\,dx \qquad (2\text{-}2)$$

The $x_i$'s and $y_i$'s are determined when (2-2) is minimized. Conditions for a minimum are discussed in Appendix 2.1. Taking the partial with respect to $x_j$, only two terms survive since the other terms are constant with respect to $x_j$ and the partial of a constant is zero. Thus (2-2) yields

OUTPUT LEVELS, SYMMETRIC DISTRIBUTION

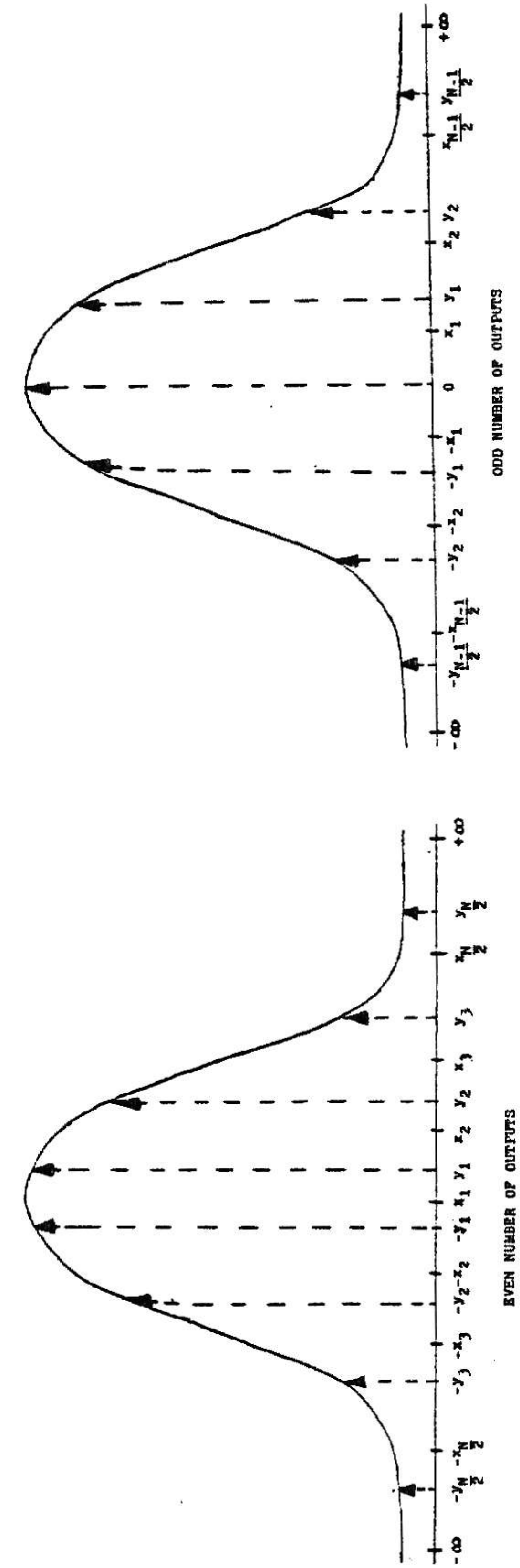



FIGURE 2.2

$$\frac{\partial E((X-Y)^2)}{\partial x_j} = \frac{\partial}{\partial x_j}\int_{x_{j-1}}^{x_j} (x-y_{j-1})^2 p_X(x)dx + \frac{\partial}{\partial x_j}\int_{x_j}^{x_{j+1}} (x-y_j)xp_X(x)dx \tag{2-3}$$

To evaluate the partial derivative in (2-3), we utilize Liebnitz's rule, which states that

$$\frac{d}{dx}\int_{a(x)}^{b(x)} g(z)dz = g(b(x))\,\frac{db(x)}{dx} - g(a(x))\,\frac{da(x)}{dx} \tag{2-4}$$

Noting that $\frac{\partial x_{j-1}}{\partial x_j} = 0$, and applying (2-4) to (2-3) results in

$$\frac{\partial E((X-Y)^2)}{\partial x_j} = (x_j-y_{j-1})^2 p_X(x_j)(1) - (x_{j-1}-y_{j-1})^2 p_X(x_{j-1})\ (0)$$

$$+ (x_{j+1}-y_j)^2 p_X(x_{j+1})(0) - (x_j-y_j)^2 p_X(x_j)\ (1)$$

or

$$\frac{\partial E((X-Y)^2)}{\partial x_j} = (x_j-y_{j-1})^2 p_X(x_j) - (x_j-y_j)^2 p_X(x_j) \tag{2-5}$$

From (2-5) it is clear that one condition for minimizing the mean square error is

$$(x_j-y_{j-1})^2 = (x_j-y_j)^2$$

or

$$y_j = 2x_j - y_{j-1} \quad \text{for } j=2,\ .\ .\ .\ ,N \tag{2-6}$$

Now for the partial with respect to $y_j$. Only one term $y_j$ varies from $y_1$ to $y_N$. Thus (2-1) yields

$$\frac{\partial E((X-Y)^2)}{\partial y_j} = \int_{x_j}^{x_{j+1}} 2(x-y_j)(-1)\,p_X(x)dx \tag{2-7}$$

From (2-7) the second condition for the minimization of the mean square error is obtained as

$$\int_{x_j}^{x_{j+1}} (x-y_j)p_X(x)\,dx = 0 \quad \text{for } j=1, \ldots, N \tag{2-8}$$

Thus to summarize, (2-6) and (2-8) completely describe the optimum quantizer, since their simultaneous solution yields all subranges of the possible input values, $X_i$ and $X_{i+1}$, and their corresponding assigned output value, $Y_i$. A general solution is difficult; i.e., for any $p_X(x)$. However, an iterative procedure [1] can produce solutions for a specified $p_X(x)$. The procedure is to chose initial values $x_1$ and $y_1$, then compute the set of $x_i$ and $y_i$ using (2-6) and (2-8). Once the last values, $x_N$ and $y_N$, have been computed, $y_N$ is compared to the average abscissa between $x_N$ and infinity since (2-8) can be written

$$y_j = \frac{\int_{x_j}^{x_{j+1}} x p_X(x)\,dx}{\int_{x_j}^{x_{j+1}} p_X(x)\,dx} \quad \text{for } j=1, \ldots, N \tag{2-9}$$

where $y_j$ is recognized as the average abscissa corresponding to the centroid of the area between $x_j$ and $x_{j+1}$. If the two values of $y_N$ are within a specified accuracy, then the optimum quantizer has been characterized. Otherwise the initial values, $x_1$ and $y_1$, are chosen again based on the difference between these two values for $y_N$. The remaining $x_i$'s and $y_i$'s are then recalculated. The entire procedure continues until the two values for $y_N$ are within a specified accuracy.

# APPENDIX 2.1

This appendix contains the conditions necessary to ensure minimization of an equation containing two variables. Conditions to ensure minimization of (2-1). A minimum occurs for the critical points where

$$\frac{\partial E}{\partial x_i} = \frac{\partial E}{\partial y_i} = 0$$

if

$$\begin{vmatrix} \dfrac{\partial^2 E}{\partial x_i^2} & \dfrac{\partial^2 E}{\partial y_i \partial x_i} \\ \\ \dfrac{\partial^2 E}{\partial x_i \partial y_i} & \dfrac{\partial^2 E}{\partial^2 y_i^2} \end{vmatrix} > 0$$

and if

$$\frac{\partial^2 E}{\partial x_i^2} > 0, \; \frac{\partial^2 E}{\partial y_i^2} > 0$$

Applying the minimization criteria to (2-1) we obtain

$$\underline{E} = \sum_{i=1}^{N} \int_{x_i}^{x_{i+1}} (y_i - x)^2 p_X(x)\,dx$$

$$\frac{\partial^2 E}{\partial y_i^2} = \int_{x_i}^{x_{i+1}} 2p_X(x)\,dx$$

$$\therefore \; \frac{\partial^2 E}{\partial y_i^2} > 0 \text{ ALWAYS}$$

$$\frac{\partial^2 E}{\partial x_i^{\,2}} = 2p_X(x_i)\ [y_i - y_{i-1}]$$

$$+ [\frac{d}{dx_i}\ p(x_i)]\ [y_i - y_{i-1}]$$

$$[(-1)\ (y_{i-1} + y_i) + 2x_i]$$

$$\frac{\partial^2 E}{\partial x_i \partial yi} = \frac{\partial^2 E}{\partial y_i \partial x_i} = \underbrace{2p(x_i)[y_{i-1} - y_i]}_{\text{ALWAYS NEGATIVE}}$$

# CHAPTER III

# THE OPTIMUM GAUSSIAN QUANTIZER

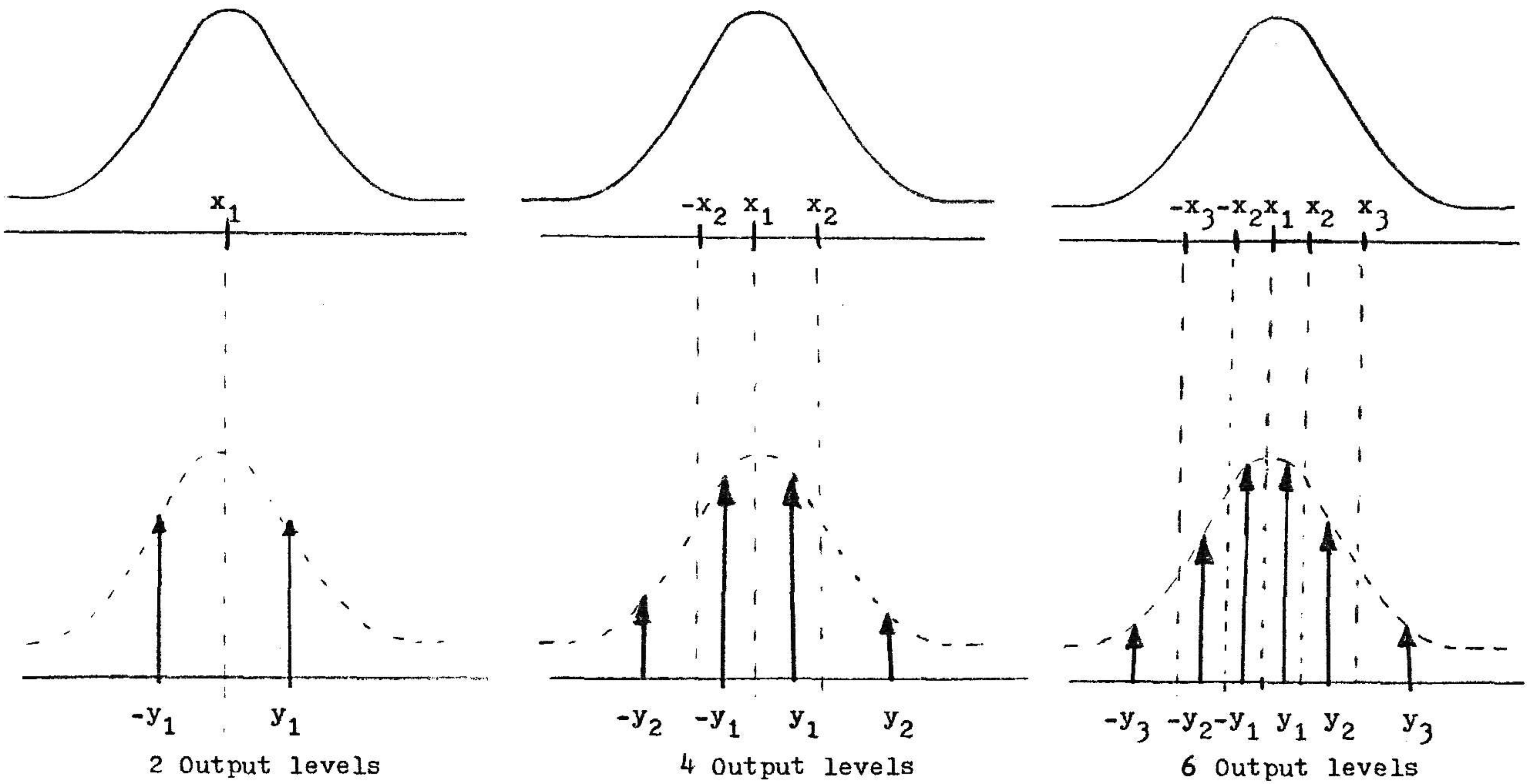
The gaussian distribution is frequently assumed for $p_X(x)$ in (2-6) and (2-8) that yield the optimum quantizer. A set of tables containing the $x_i$ and $y_i$ parameters for the gaussian case were generated by Max [1]. In this chapter we discuss some details related to generating Max's tables.

The critical portion of the iterative procedure to obtain the optimum quantizer for a given number of output levels N, is the initial guess for $x_1$ and $y_1$. It should be pointed out that only one of these values is chosen. The other value is set equal to zero, forcing the output of the quantizer to be symmetric about the origin, as illustrated in Figure 3.1. What about the value to be chosen? To this end, one can refer to Max's tables which are available for values of N from 2 through 36 [1]. It is convenient to obtain an empirical relationship between the initial values published by Max [1] and the number of output levels N. Hein [3] has obtained a handy relationship via a least squares fit of the initial values published by Max [1]. This relationship is given by [3]

$$P_1 = \frac{2.1613}{N} - \frac{1.653}{N^2} \tag{3-1}$$

where $P_1$ is $x_1$ if N is odd, or $y_1$ if N is even. Values for $P_1$ are tabulated and compared to Max's initial values in Table 3.1. Clearly the results compare closely.

The remaining $x_i$'s and $y_i$'s are obtained from (2-6) and (2-8); i.e., EVEN NUMBER OF OUTPUTS ($x_1$ is always zero)

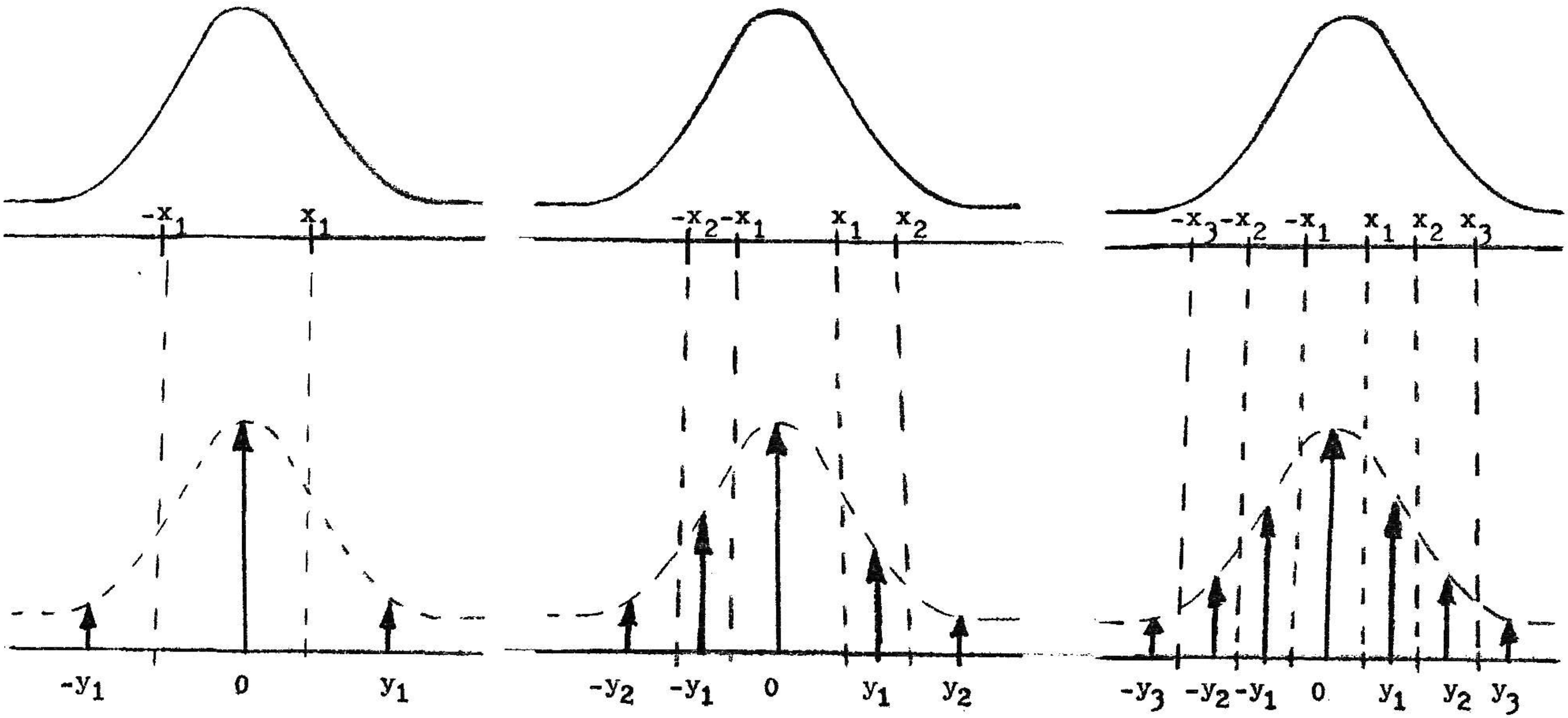



Note $Y_i$ is the centroid of the area between $X_i$ and $X_{i+1}$ .

The positive values of $X_i$ and $Y_i$ are tabulated in Table 3.2, these values are symmetric about the origin.

**FIGURE 3.1**

ODD NUMBER OF OUTPUTS ($y_1$ is always zero)




Note $Y_i$ is the centroid of the area between $X_i$ and $X_{i+1}$ .

The positive values of $X_i$ and $Y_i$ are tabulated in Table 3.2 , these values are symmetric about the origin

**FIGURE 3.1 (CONT.)**

TABLE 3.1

INITIAL VALUES

| NUMBER OF OUTPUT LEVELS | INITIAL VALUES EQ. (3-1) $P_1 = \frac{2.1613}{N} - \frac{1.653}{N^2}$ | INITIAL VALUES PUBLISHED BY MAX |
|---|---|---|
| 4 | 0.43701 | 0.4528 |
| 5 | 0.36614 | 0.3823 |
| 6 | 0.31430 | 0.3177 |
| 7 | 0.27502 | 0.2803 |
| 8 | 0.24433 | 0.2451 |
| 9 | 0.21974 | 0.2218 |
| 10 | 0.19960 | 0.1996 |
| 11 | 0.18282 | 0.1837 |
| 12 | 0.16863 | 0.1684 |
| 13 | 0.15647 | 0.1569 |
| 14 | 0.14594 | 0.1457 |
| 15 | 0.13674 | 0.1369 |
| 16 | 0.12862 | 0.1284 |
| 17 | 0.12142 | 0.1215 |
| 18 | 0.11497 | 0.1148 |
| 19 | 0.10917 | 0.1092 |
| 20 | 0.10393 | 0.1038 |
| 21 | 0.09917 | 0.09918 |
| 22 | 0.09483 | 0.09469 |
| 23 | 0.09084 | 0.09085 |
| 24 | 0.08718 | 0.08708 |
| 25 | 0.08381 | 0.08381 |
| 26 | 0.08068 | 0.08060 |
| 27 | 0.07778 | 0.07779 |
| 28 | 0.07508 | 0.07502 |
| 29 | 0.07256 | 0.07257 |
| 30 | 0.07021 | 0.07016 |
| 31 | 0.06800 | 0.06802 |
| 32 | 0.06593 | 0.06500 |
| 33 | 0.06398 | 0.06400 |
| 34 | 0.06214 | 0.06212 |
| 35 | 0.06040 | 0.06043 |
| 36 | 0.05876 | 0.05876 |
| 64 | 0.03337 | 0.03342 |
| 128 | 0.01678 | 0.01685 |

$$y_j = 2x_j - y_{j-1} \qquad \text{for } j=2, \ldots, M$$

and

$$\int_{x_j}^{x_{j+1}} (x - y_j) p_X(x) dx = 0 \qquad \text{for } j=1, \ldots, M$$

where

$$M = \begin{cases} \frac{N}{2} & , \text{ N EVEN} \\ \frac{(N-1)}{2} & , \text{ N ODD} \end{cases}$$

due to symmetry, i.e, $x_j = -x_j$ and $y_j = -y_j$, as shown in Figure 3.2. Let $p_X(x)$ be gaussianly distributed

$$p_X(x) = \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}} \quad , \quad |x| < \infty \tag{3-2}$$

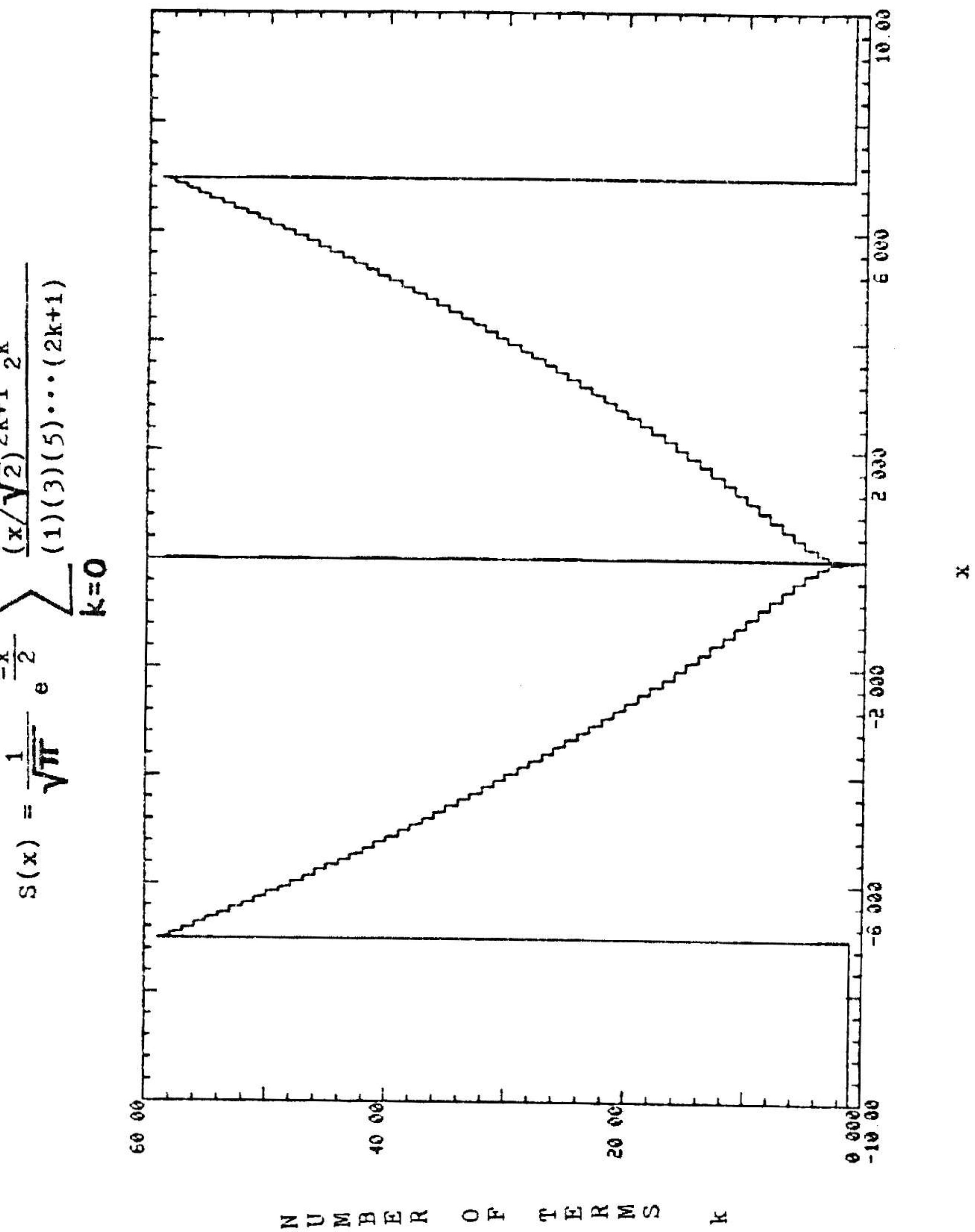
The value of $x_{j+1}$ in (2-8) is determined via an iterative procedure for finding the zeroes of functions, Newton's method [5] which is described in Appendix 3.1. The actual integration in (2-8) is accomplished by expressing (2-8) as

$$\int_{x_j}^{x_{j+1}} x p_X(x) dx - y_j \int_{x_j}^{x_{j+1}} p_X(x) dx = 0 \tag{3-3}$$

where the first part is easily integrable, once it is observed that for our particular choice of $p_X(x)$

$$\frac{dp_X(x)}{dx} = -x p_X(x) \tag{3-4}$$

The second part is evaluated by replacing the integral with its equivalent infinite series representation [4] which converges very rapidly as shown in Figure 3.2. Thus if



$S(x) = \frac{1}{\sqrt{\pi}} e^{\frac{-x^2}{2}} \sum_{k=0}^{\infty} \frac{(x/\sqrt{2})^{2k+1}\ 2^k}{(1)(3)(5)\cdots(2k+1)}$
60.00
40.00
20.00
0.000
-10.00
-6.000
-2.000
2.000
6.000
10.00
x
NUMBER OF TERMS k


FIGURE 3.2

$$S(x) = \frac{1}{\sqrt{\pi}} e^{\frac{-x^2}{2}} \sum_{k=0}^{\infty} \frac{(x/\sqrt{2})^{2k+1} 2^k}{(1)(3)(5)\ldots(2k+1)} \tag{3-5}$$

then

$$\int_{x_j}^{x_{j+1}} p_X(x)dx = S(x_{j+1}) - S(x_j) \tag{3-6}$$

Also, we note that

$$\int_{x_j}^{\infty} p_X(x)dx = S(+\infty) - S(x_j) = 0.5 - S(x_j) \tag{3-7}$$

since

$$\frac{1}{\sqrt{2\pi}} \int_0^{+\infty} e^{\frac{-x^2}{2}} dx = 0.5 \tag{3-8}$$

After the last $x_i$ and $y_i$ are calculated, $y_M$ is also determined by an alternate method. The abscissa of the centroid of the area between $x_M$ and infinity should also equal $y_M$, as seen in (2-9)--i.e.,

$$y_{M_c} = \frac{\int_{x_M}^{\infty} xp_X(x)\,dx}{\int_{x_M}^{\infty} p_X(x)\,dx}$$

The two values for $y_M$ are compared. If they are close enough, then the optimum quantizer has been determined. Otherwise, the initial value must be updated and the $x_i$'s and $y_i$'s again computed. The initial value is updated using an iterative procedure for finding the zeroes of functions, ie force $y_M - y_{M_C}$ toward zero. Newton's method can't be used in this instance because an explicit expression for the first derivative of the function to be zeroed is not available. Thus a similar iterative procedure must be used, the secant method [5] which is described in Appendix 3.2

$$P_1^{**} = \frac{P_1(y_M^* - y_{M_c}^*) - P_1^*(y_M - y_{M_c})}{(y_M^* - y_{M_c}^*) - (y_M - y_{M_c})} \tag{3-9}$$

where $y_M$ is calculated from (2-6) and (2-8), while $y_{M_c}$ is obtained using (2-9). In (3-9),

$$P_1 = \begin{cases} x_1 & \text{if N is odd} \\ y_1 & \text{is N is even} \end{cases}$$

Two asterisks denote the updated value, one asterisk denotes the current value, while no asterisk denotes the previous value. The secant method converges only if the initial values (current and last) are sufficiently close to the actual initial value of the optimum quantizer. However, when it does converge it converges very quickly [5]. For example, with N=128 output levels, the entire iterative process took only six iterations for an accuracy to six decimal places on the Data General Nova 4X computer.

The program to generate the parameters $X_i$ and $Y_i$, which completely describe the optimum gaussian quantizer for a finite number of output levels, N, is listed in Appendix 3.3. Tables generated using this program are summarized in Table 3.2. A graphical representation of the input and output values for the optimum quantizer, tabulated for a specific N in Table 3.2, is perhaps more enlightening. Figures 3.4 thru 3.10 show these input/output transfer functions for N=2 thru N=128 respectively. Figure 3.11 shows the input/output transfer function for N=5.

THE OPTIMUM QUANTIZER

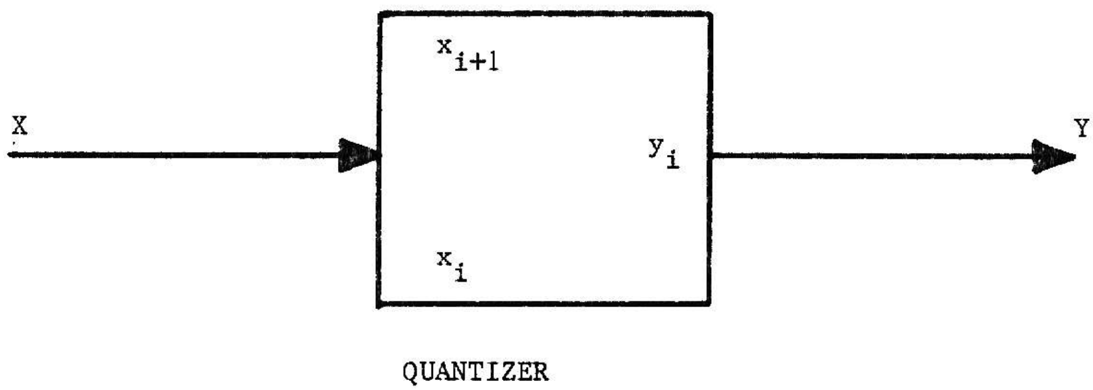



FIGURE 3.3

## TABLE 3.2

N= 2

| J | X(J) | Y(J) |
|---|---|---|
| 1 | .000000 | .797885 |

N= 3

| J | X(J) | Y(J) |
|---|---|---|
| 1 | .612003 | 1.224007 |

N= 4

| J | X(J) | Y(J) |
|---|---|---|
| 1 | .000000 | .452781 |
| 2 | .981600 | 1.510419 |

N= 5

| J | X(J) | Y(J) |
|---|---|---|
| 1 | .382284 | .764568 |
| 2 | 1.244357 | 1.724147 |

N= 6

| J | X(J) | Y(J) |
|---|---|---|
| 1 | .000000 | .317717 |
| 2 | .658912 | 1.000107 |
| 3 | 1.446853 | 1.893598 |

N= 7

| J | X(J) | Y(J) |
|---|---|---|
| 1 | .280289 | .560577 |
| 2 | .874363 | 1.188149 |
| 3 | 1.610763 | 2.033378 |

N= 8

| J | X(J) | Y(J) |
|---|---|---|
| 1 | .000000 | .245095 |
| 2 | .500550 | .756006 |
| 3 | 1.049959 | 1.343911 |
| 4 | 1.747929 | 2.151946 |

N= 9

| J | X(J) | Y(J) |
|---|---|---|
| 1 | .221819 | .443639 |
| 2 | .681218 | .918797 |
| 3 | 1.197595 | 1.476392 |
| 4 | 1.865532 | 2.254671 |

N= 10

| J | X(J) | Y(J) |
|---|---|---|
| 1 | .000000 | .199624 |
| 2 | .404742 | .609859 |
| 3 | .833844 | 1.057829 |
| 4 | 1.324587 | 1.591345 |
| 5 | 1.968230 | 2.345115 |

N= 11

| J | X(J) | Y(J) |
|---|---|---|
| 1 | .183729 | .367458 |
| 2 | .559913 | .752368 |
| 3 | .965599 | 1.178830 |
| 4 | 1.435738 | 1.692646 |
| 5 | 2.059198 | 2.425750 |

N= 12

| J | X(J) | Y(J) |
|---|---|---|
| 1 | .000000 | .168439 |
| 2 | .340144 | .511849 |
| 3 | .694316 | .876784 |
| 4 | 1.081252 | 1.285721 |
| 5 | 1.534384 | 1.783047 |
| 6 | 2.140753 | 2.498459 |

N= 13

| J | X(J) | Y(J) |
|---|---|---|
| 1 | .156887 | .313773 |
| 2 | .476013 | .638252 |
| 3 | .812602 | .986953 |
| 4 | 1.184109 | 1.381266 |
| 5 | 1.622901 | 1.864535 |
| 6 | 2.214547 | 2.564559 |

N= 14

| J | X(J) | Y(J) |
|---|---|---|
| 1 | .000000 | .145708 |
| 2 | .293516 | .441323 |
| 3 | .595885 | .750446 |
| 4 | .918045 | 1.085643 |
| 5 | 1.276590 | 1.467538 |
| 6 | 1.703080 | 1.938623 |
| 7 | 2.281868 | 2.625112 |

N= 15

| J | X(J) | Y(J) |
|---|---|---|
| 1 | .136929 | .273857 |
| 2 | .414311 | .554765 |
| 3 | .702951 | .851138 |
| 4 | 1.013013 | 1.174888 |
| 5 | 1.360479 | 1.546069 |
| 6 | 1.776280 | 2.006491 |
| 7 | 2.343715 | 2.680939 |

N= 16

| J | X(J) | Y(J) |
|---|---|---|
| 1 | .000000 | .128397 |
| 2 | .258224 | .388051 |
| 3 | .522406 | .656762 |
| 4 | .799554 | .942347 |
| 5 | 1.099296 | 1.256245 |
| 6 | 1.437155 | 1.618065 |
| 7 | 1.843554 | 2.069043 |
| 8 | 2.400829 | 2.732615 |

N= 17

| J | X(J) | Y(J) |
|---|---|---|
| 1 | .121497 | .242994 |
| 2 | .366939 | .490883 |
| 3 | .620086 | .749290 |
| 4 | .887446 | 1.025603 |
| 5 | 1.178253 | 1.330902 |
| 6 | 1.507680 | 1.684458 |
| 7 | 1.905731 | 2.127003 |
| 8 | 2.453907 | 2.780811 |

N= 18

| J | X(J) | Y(J) |
|---|---|---|
| 1 | .000000 | .114771 |
| 2 | .230560 | .346348 |
| 3 | .465327 | .584306 |
| 4 | .709087 | .833868 |
| 5 | .967991 | 1.102113 |
| 6 | 1.250977 | 1.399840 |
| 7 | 1.572936 | 1.746031 |
| 8 | 1.963506 | 2.180981 |
| 9 | 2.503445 | 2.825908 |

THE OPTIMUM GAUSSIAN QUANTIZER PARAMETERS

TABLE 3.2 (CONT.)

N=·19

| J | X(J) | Y(J) |
|---|---|---|
| 1 | .109205 | .218409 |
| 2 | .329383 | .440357 |
| 3 | .555079 | .669801 |
| 4 | .790737 | .911674 |
| 5 | 1.042246 | 1.172817 |
| 6 | 1.318314 | 1.463810 |
| 7 | 1.633597 | 1.803385 |
| 8 | 2.017416 | 2.231448 |
| 9 | 2.549850 | 2.868252 |

N= 20

| J | X(J) | Y(J) |
|---|---|---|
| 1 | .000000 | .103765 |
| 2 | .208280 | .312794 |
| 3 | .419643 | .526492 |
| 4 | .637515 | .748539 |
| 5 | .866096 | .983652 |
| 6 | 1.111070 | 1.238487 |
| 7 | 1.380962 | 1.523438 |
| 8 | 1.690218 | 1.856998 |
| 9 | 2.067866 | 2.278734 |
| 10 | 2.593412 | 2.908089 |

N= 21

| J | X(J) | Y(J) |
|---|---|---|
| 1 | .099179 | .198357 |
| 2 | .298857 | .399356 |
| 3 | .502625 | .605894 |
| 4 | .713670 | .821446 |
| 5 | .936004 | 1.050562 |
| 6 | 1.175154 | 1.299746 |
| 7 | 1.439494 | 1.579242 |
| 8 | 1.743296 | 1.907350 |
| 9 | 2.115335 | 2.323319 |
| 10 | 2.634465 | 2.945612 |

N= 22

| J | X(J) | Y(J) |
|---|---|---|
| 1 | .000000 | .094689 |
| 2 | .189945 | .285201 |
| 3 | .382218 | .479235 |
| 4 | .579363 | .679490 |
| 5 | .784387 | .889284 |
| 6 | 1.001159 | 1.113034 |
| 7 | 1.235071 | 1.357108 |
| 8 | 1.494378 | 1.631648 |
| 9 | 1.793210 | 1.954771 |
| 10 | 2.160102 | 2.365433 |
| 11 | 2.673377 | 2.981322 |

N= 23

| J | X(J) | Y(J) |
|---|---|---|
| 1 | .090844 | .181688 |
| 2 | .273545 | .365401 |
| 3 | .459364 | .553328 |
| 4 | .650672 | .748017 |
| 5 | .850341 | .952664 |
| 6 | 1.062122 | 1.171580 |
| 7 | 1.291304 | 1.411028 |
| 8 | 1.546034 | 1.681040 |
| 9 | 1.840308 | 1.999576 |
| 10 | 2.202451 | 2.405326 |
| 11 | 2.710289 | 3.015252 |

N= 24

| J | X(J) | Y(J) |
|---|---|---|
| 1 | .000000 | .087075 |
| 2 | .174590 | .262105 |
| 3 | .350982 | .439859 |
| 4 | .531118 | .622378 |
| 5 | .717236 | .812095 |
| 6 | .912102 | 1.012110 |
| 7 | 1.119376 | 1.226643 |
| 8 | 1.344255 | 1.461868 |
| 9 | 1.594795 | 1.727721 |
| 10 | 1.884872 | 2.042022 |
| 11 | 2.242618 | 2.443213 |
| 12 | 2.745395 | 3.047577 |

N= 25

| J | X(J) | Y(J) |
|---|---|---|
| 1 | .083805 | .167611 |
| 2 | .252210 | .336808 |
| 3 | .423048 | .509288 |
| 4 | .598134 | .686980 |
| 5 | .779603 | .872225 |
| 6 | .970134 | 1.068042 |
| 7 | 1.173310 | 1.278578 |
| 8 | 1.394255 | 1.509932 |
| 9 | 1.640941 | 1.771950 |
| 10 | 1.927137 | 2.082325 |
| 11 | 2.280796 | 2.479268 |
| 12 | 2.778845 | 3.078423 |

N= 26

| J | X(J) | Y(J) |
|---|---|---|
| 1 | .000000 | .080596 |
| 2 | .161540 | .242484 |
| 3 | .324503 | .406522 |
| 4 | .490408 | .574295 |
| 5 | .660970 | .747645 |
| 6 | .838242 | .928839 |
| 7 | 1.024834 | 1.120830 |
| 8 | 1.224257 | 1.327684 |
| 9 | 1.441577 | 1.555469 |
| 10 | 1.684685 | 1.813901 |
| 11 | 1.967265 | 2.120629 |
| 12 | 2.317117 | 2.513606 |
| 13 | 2.810743 | 3.107880 |

N= 27

| J | X(J) | Y(J) |
|---|---|---|
| 1 | .077781 | .155562 |
| 2 | .233977 | .312391 |
| 3 | .392110 | .471828 |
| 4 | .553600 | .635372 |
| 5 | .720083 | .804795 |
| 6 | .893548 | .982302 |
| 7 | 1.076545 | 1.170789 |
| 8 | 1.272530 | 1.374271 |
| 9 | 1.486513 | 1.598736 |
| 10 | 1.726320 | 1.853884 |
| 11 | 2.005543 | 2.157203 |
| 12 | 2.351832 | 2.546462 |
| 13 | 2.841302 | 3.136142 |

TABLE 3.2 (CONT.)

N= 28

| J | X(J) | Y(J) |
|---|---|---|
| 1 | .000000 | .075016 |
| 2 | .150311 | .225606 |
| 3 | .301765 | .377924 |
| 4 | .455574 | .533225 |
| 5 | .613084 | .692943 |
| 6 | .775866 | .858789 |
| 7 | .945854 | 1.032919 |
| 8 | 1.125551 | 1.218183 |
| 9 | 1.318363 | 1.416544 |
| 10 | 1.529252 | 1.639960 |
| 11 | 1.765982 | 1.892004 |
| 12 | 2.042030 | 2.192055 |
| 13 | 2.384871 | 2.577687 |
| 14 | 2.870376 | 3.163065 |

N= 29

| J | X(J) | Y(J) |
|---|---|---|
| 1 | .072566 | .145132 |
| 2 | .218212 | .291292 |
| 3 | .365425 | .439558 |
| 4 | .515340 | .591122 |
| 5 | .669241 | .747360 |
| 6 | .828647 | .909935 |
| 7 | .995448 | 1.080962 |
| 8 | 1.172105 | 1.263248 |
| 9 | 1.361981 | 1.460714 |
| 10 | 1.569994 | 1.679274 |
| 11 | 1.803854 | 1.928434 |
| 12 | 2.076956 | 2.225478 |
| 13 | 2.416638 | 2.607798 |
| 14 | 2.898229 | 3.188661 |

N= 30

| J | X(J) | Y(J) |
|---|---|---|
| 1 | .000000 | .070159 |
| 2 | .140546 | .210932 |
| 3 | .282023 | .353114 |
| 4 | .425417 | .497719 |
| 5 | .571802 | .645884 |
| 6 | .722412 | .798939 |
| 7 | .878724 | .958510 |
| 8 | 1.042590 | 1.126670 |
| 9 | 1.216423 | 1.306177 |
| 10 | 1.403564 | 1.500951 |
| 11 | 1.608897 | 1.716844 |
| 12 | 1.840074 | 1.963203 |
| 13 | 2.110414 | 2.257524 |
| 14 | 2.447124 | 2.636725 |
| 15 | 2.925205 | 3.213685 |

N= 31

| J | X(J) | Y(J) |
|---|---|---|
| 1 | .068008 | .136016 |
| 2 | .204447 | .272878 |
| 3 | .342172 | .411467 |
| 4 | .482105 | .552744 |
| 5 | .625274 | .697803 |
| 6 | .772870 | .847936 |
| 7 | .926335 | 1.004733 |
| 8 | 1.087487 | 1.170240 |
| 9 | 1.258708 | 1.347177 |
| 10 | 1.443307 | 1.539436 |
| 11 | 1.646133 | 1.752830 |
| 12 | 1.874790 | 1.996749 |
| 13 | 2.142526 | 2.288304 |
| 14 | 2.476428 | 2.664553 |
| 15 | 2.951180 | 3.237808 |

N= 32

| J | X(J) | Y(J) |
|---|---|---|
| 1 | .000000 | .065894 |
| 2 | .131975 | .198057 |
| 3 | .264720 | .331384 |
| 4 | .399045 | .466706 |
| 5 | .535825 | .604943 |
| 6 | .676046 | .747149 |
| 7 | .820867 | .894585 |
| 8 | .971699 | 1.048814 |
| 9 | 1.130332 | 1.211851 |
| 10 | 1.299120 | 1.386389 |
| 11 | 1.481341 | 1.576292 |
| 12 | 1.681793 | 1.787293 |
| 13 | 1.908057 | 2.028821 |
| 14 | 2.173340 | 2.317860 |
| 15 | 2.504587 | 2.691315 |
| 16 | 2.976185 | 3.261056 |

N= 33

| J | X(J) | Y(J) |
|---|---|---|
| 1 | .063990 | .127980 |
| 2 | .192322 | .256663 |
| 3 | .321722 | .386781 |
| 4 | .452951 | .519121 |
| 5 | .586844 | .654567 |
| 6 | .724355 | .794143 |
| 7 | .866613 | .939084 |
| 8 | 1.015004 | 1.090925 |
| 9 | 1.171293 | 1.251661 |
| 10 | 1.337808 | 1.423955 |
| 11 | 1.517801 | 1.611646 |
| 12 | 1.716040 | 1.820433 |
| 13 | 1.940066 | 2.059700 |
| 14 | 2.203026 | 2.346353 |
| 15 | 2.531752 | 2.717152 |
| 16 | 3.000350 | 3.283548 |

N= 34

| J | X(J) | Y(J) |
|---|---|---|
| 1 | .000000 | .062118 |
| 2 | .124393 | .186667 |
| 3 | .249426 | .312189 |
| 4 | .375782 | .439375 |
| 5 | .504176 | .568977 |
| 6 | .635414 | .701850 |
| 7 | .770421 | .838992 |
| 8 | .910303 | .981613 |
| 9 | 1.056421 | 1.131229 |
| 10 | 1.210508 | 1.289787 |
| 11 | 1.374882 | 1.459977 |
| 12 | 1.552783 | 1.645589 |
| 13 | 1.748937 | 1.852285 |
| 14 | 1.970850 | 2.089415 |
| 15 | 2.231610 | 2.373805 |
| 16 | 2.557943 | 2.742081 |
| 17 | 3.023688 | 3.305295 |

N= 35

| J | X(J) | Y(J) |
|---|---|---|
| 1 | .060421 | .120842 |
| 2 | .181558 | .242275 |
| 3 | .303594 | .364912 |
| 4 | .427161 | .489409 |
| 5 | .552949 | .616490 |
| 6 | .681736 | .746983 |
| 7 | .814424 | .881865 |
| 8 | .952094 | 1.022324 |
| 9 | 1.096092 | 1.169859 |
| 10 | 1.248128 | 1.326396 |
| 11 | 1.410501 | 1.494606 |
| 12 | 1.586430 | 1.678255 |
| 13 | 1.780614 | 1.882974 |
| 14 | 2.000525 | 2.118076 |
| 15 | 2.259196 | 2.400316 |
| 16 | 2.583252 | 2.766188 |
| 17 | 3.046278 | 3.326368 |

N= 36

| J | X(J) | Y(J) |
|---|---|---|
| 1 | .000000 | .058753 |
| 2 | .117636 | .176520 |
| 3 | .235814 | .295109 |
| 4 | .355105 | .415100 |
| 5 | .476110 | .537119 |
| 6 | .599493 | .661867 |
| 7 | .726008 | .790150 |
| 8 | .856538 | .922926 |
| 9 | .992145 | 1.061365 |
| 10 | 1.134157 | 1.206949 |
| 11 | 1.284267 | 1.361585 |
| 12 | 1.444756 | 1.527927 |
| 13 | 1.618824 | 1.709721 |
| 14 | 1.811144 | 1.912567 |
| 15 | 2.029155 | 2.145743 |
| 16 | 2.285839 | 2.425936 |
| 17 | 2.607726 | 2.789517 |
| 18 | 3.068161 | 3.346805 |

N= 64

| J | X(J) | Y(J) |
|---|---|---|
| 1 | .000000 | .033419 |
| 2 | .066854 | .100290 |
| 3 | .133800 | .167310 |
| 4 | .200947 | .234583 |
| 5 | .268398 | .302212 |
| 6 | .336259 | .370306 |
| 7 | .404641 | .438977 |
| 8 | .473662 | .508347 |
| 9 | .543445 | .578542 |
| 10 | .614123 | .649703 |
| 11 | .685841 | .721980 |
| 12 | .758759 | .795538 |
| 13 | .833050 | .870562 |
| 14 | .908910 | .947259 |
| 15 | .986561 | 1.025864 |
| 16 | 1.066254 | 1.106645 |
| 17 | 1.148279 | 1.189913 |
| 18 | 1.232951 | 1.275989 |
| 19 | 1.320668 | 1.365346 |
| 20 | 1.411925 | 1.458504 |
| 21 | 1.507301 | 1.556097 |
| 22 | 1.607507 | 1.658916 |
| 23 | 1.713393 | 1.767870 |
| 24 | 1.826106 | 1.884341 |
| 25 | 1.947202 | 2.010062 |
| 26 | 2.078662 | 2.147262 |
| 27 | 2.223400 | 2.299539 |
| 28 | 2.385699 | 2.471860 |
| 29 | 2.572499 | 2.673138 |
| 30 | 2.795894 | 2.918650 |
| 31 | 3.080666 | 3.242681 |
| 32 | 3.495806 | 3.748931 |

N=128

| J | X(J) | Y(J) |
|---|---|---|
| 1 | .000000 | .016846 |
| 2 | .033680 | .050513 |
| 3 | .067356 | .084199 |
| 4 | .101058 | .117917 |
| 5 | .134799 | .151681 |
| 6 | .168591 | .185502 |
| 7 | .202448 | .219394 |
| 8 | .236382 | .253371 |
| 9 | .270408 | .287446 |
| 10 | .304539 | .321633 |
| 11 | .338790 | .355946 |
| 12 | .373174 | .390401 |
| 13 | .407706 | .425011 |
| 14 | .442402 | .459793 |
| 15 | .477277 | .494762 |
| 16 | .512348 | .529935 |
| 17 | .547632 | .565329 |
| 18 | .583145 | .600961 |
| 19 | .618907 | .636852 |
| 20 | .654935 | .673019 |
| 21 | .691252 | .709484 |
| 22 | .727876 | .746268 |
| 23 | .764830 | .783393 |
| 24 | .802139 | .820884 |
| 25 | .839825 | .858766 |
| 26 | .877916 | .897066 |
| 27 | .916440 | .935813 |
| 28 | .955425 | .975037 |
| 29 | .994904 | 1.014771 |
| 30 | 1.034911 | 1.055050 |
| 31 | 1.075482 | 1.095913 |
| 32 | 1.116657 | 1.137400 |
| 33 | 1.158478 | 1.179556 |
| 34 | 1.200950 | 1.222343 |
| 35 | 1.244120 | 1.265897 |
| 36 | 1.288086 | 1.310274 |
| 37 | 1.332906 | 1.355537 |
| 38 | 1.378646 | 1.401755 |
| 39 | 1.425380 | 1.449005 |
| 40 | 1.473187 | 1.497370 |
| 41 | 1.522158 | 1.546946 |
| 42 | 1.572393 | 1.597840 |
| 43 | 1.624005 | 1.650171 |
| 44 | 1.677036 | 1.703901 |
| 45 | 1.731625 | 1.759349 |
| 46 | 1.788020 | 1.816691 |
| 47 | 1.846411 | 1.876132 |
| 48 | 1.907021 | 1.937910 |
| 49 | 1.970109 | 2.002307 |
| 50 | 2.035984 | 2.069661 |
| 51 | 2.104834 | 2.140007 |
| 52 | 2.177086 | 2.214164 |
| 53 | 2.253450 | 2.292736 |
| 54 | 2.334611 | 2.376485 |
| 55 | 2.421100 | 2.465715 |
| 56 | 2.513999 | 2.562284 |
| 57 | 2.615103 | 2.667923 |
| 58 | 2.725862 | 2.783802 |
| 59 | 2.849152 | 2.914503 |
| 60 | 2.988916 | 3.063329 |
| 61 | 3.150659 | 3.237989 |
| 62 | 3.345521 | 3.453054 |
| 63 | 3.597598 | 3.742741 |
| 64 | 3.974031 | 4.205320 |

TABLE 3.2 (CONT.)

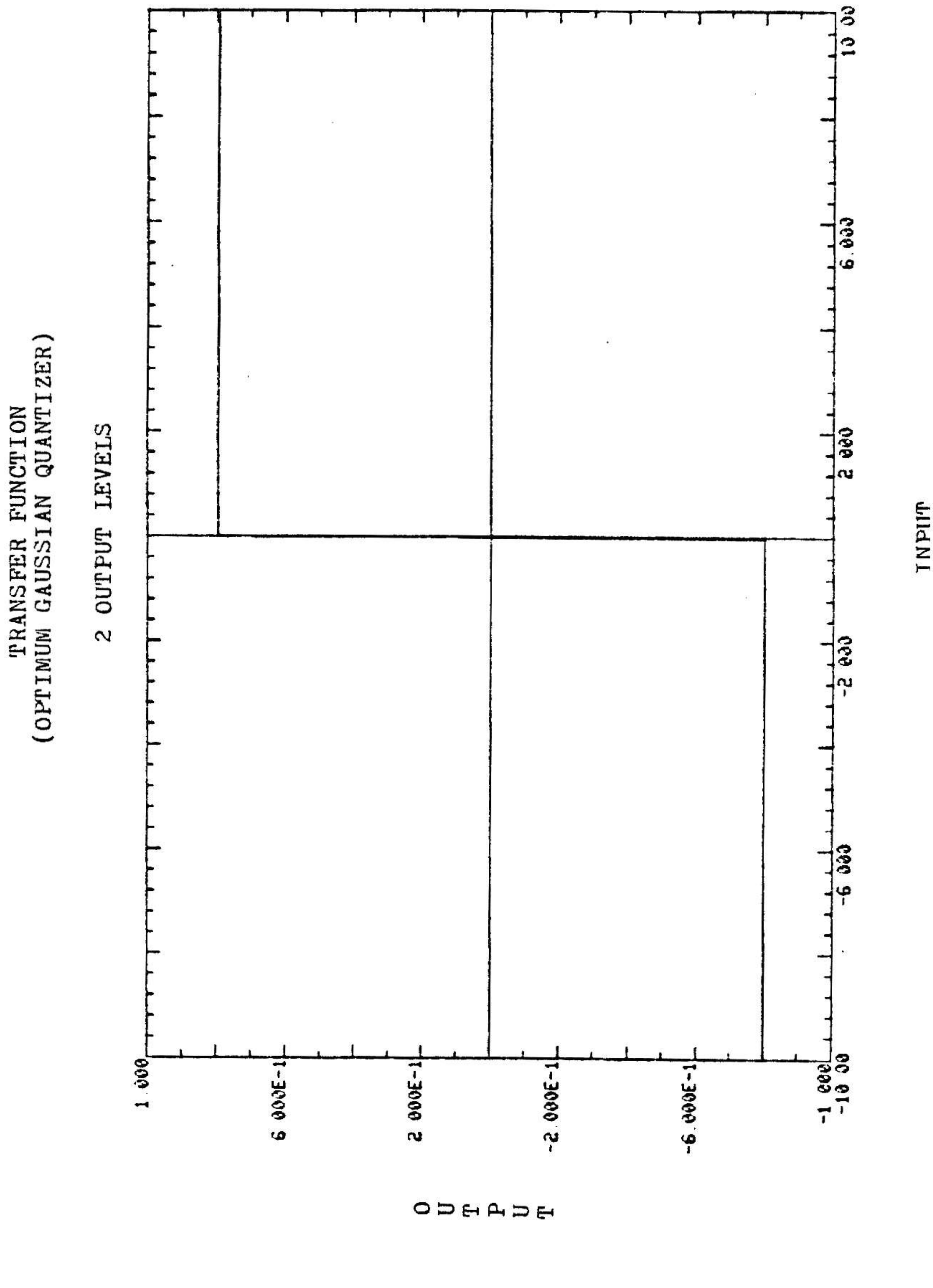

TRANSFER FUNCTION
(OPTIMUM GAUSSIAN QUANTIZER)
2 OUTPUT LEVELS
INPUT
OUTPUT
1.000
6.000E-1
2.000E-1
-2.000E-1
-6.000E-1
-1.000
-10.00
-6.000
-2.000
2.000
6.000
10.00

FIGURE 3.4

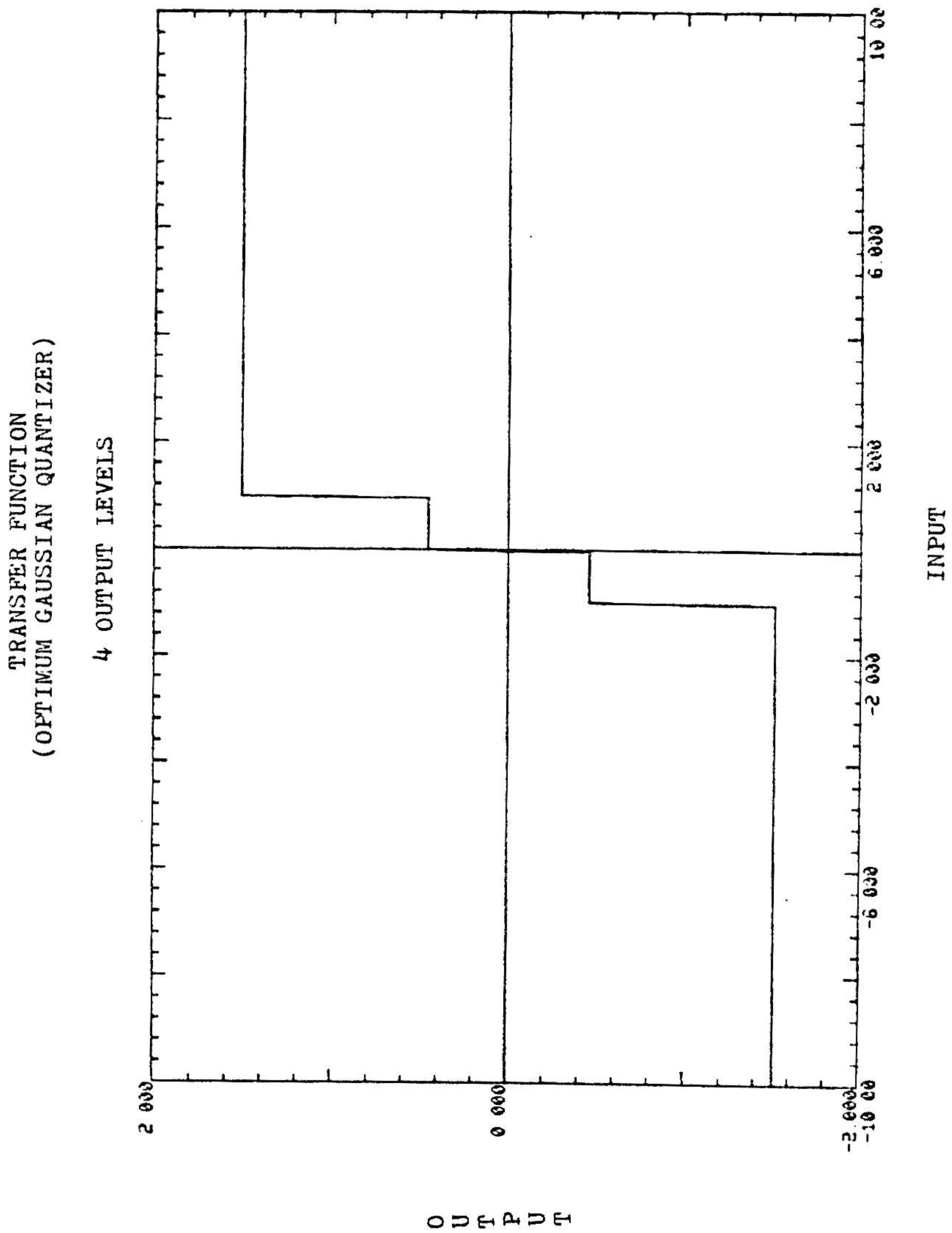

TRANSFER FUNCTION
(OPTIMUM GAUSSIAN QUANTIZER)
4 OUTPUT LEVELS
INPUT
OUTPUT
2.000
0.000
-2.000
-10.00
-6.000
-2.000
2.000
6.000
10.00

FIGURE 3.5

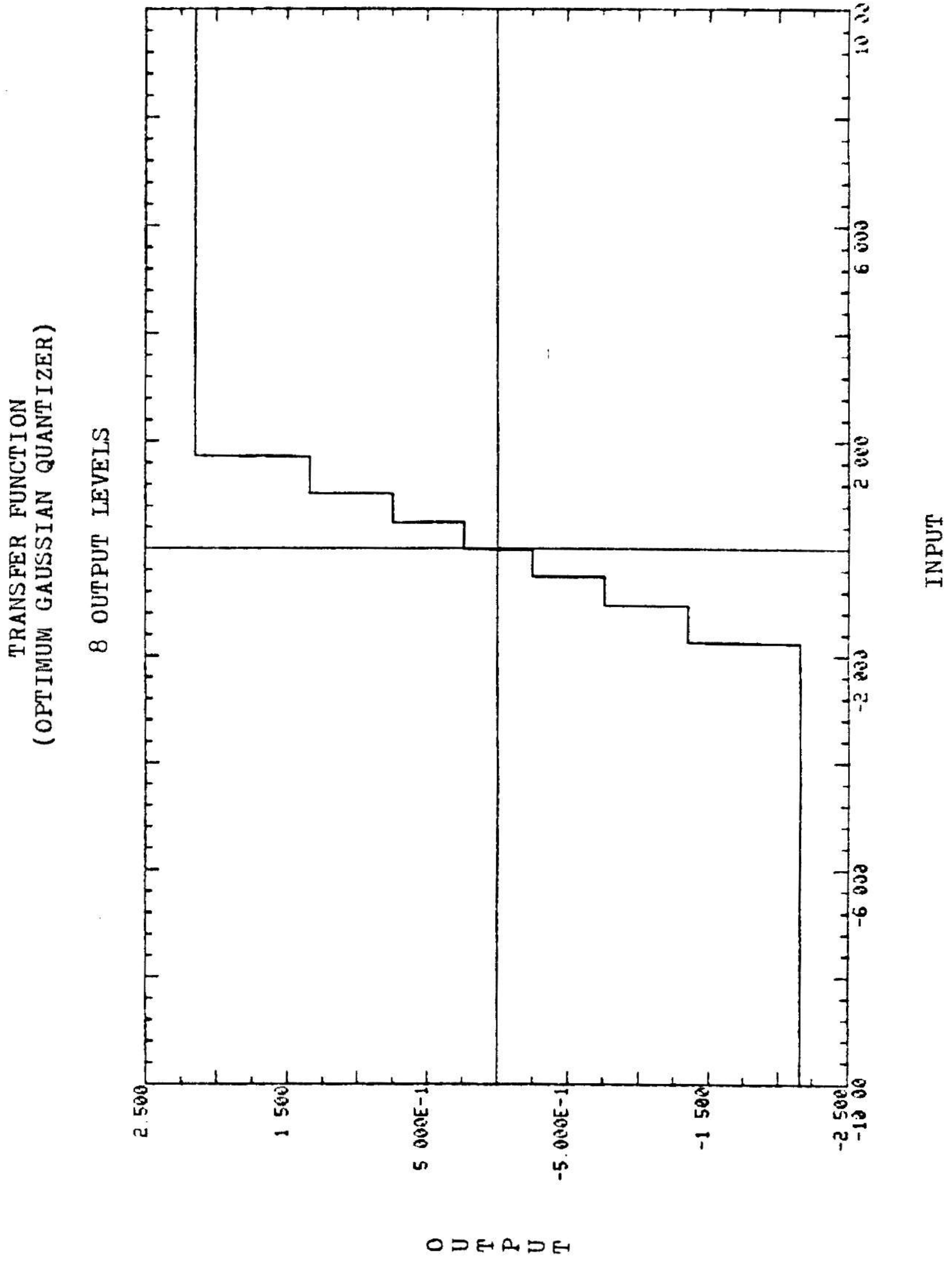

TRANSFER FUNCTION
(OPTIMUM GAUSSIAN QUANTIZER)
8 OUTPUT LEVELS
INPUT
OUTPUT
2.500
1.500
5.000E-1
-5.000E-1
-1.500
-2.500
-10.00
-6.000
-2.000
2.000
6.000

FIGURE 3.6

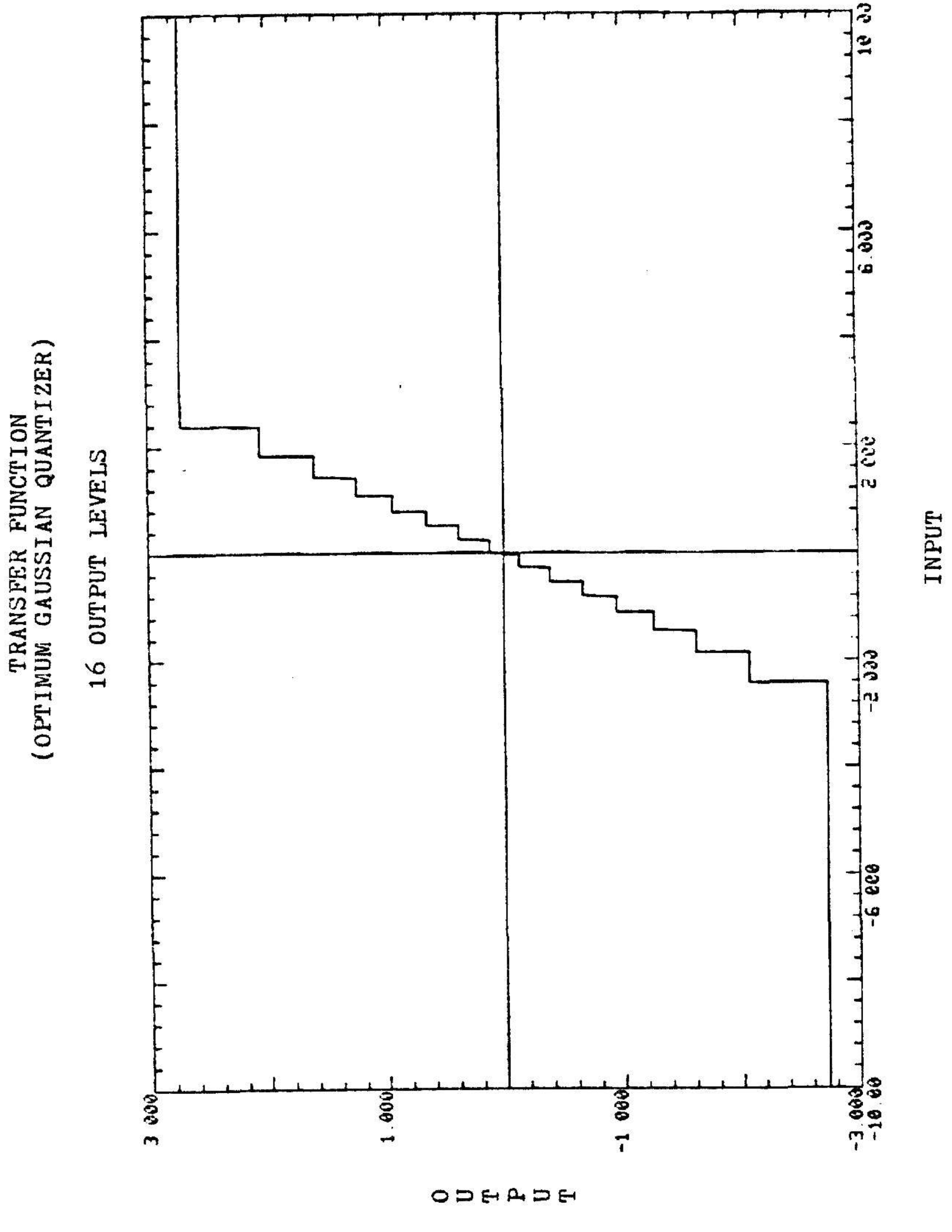

TRANSFER FUNCTION
(OPTIMUM GAUSSIAN QUANTIZER)
16 OUTPUT LEVELS
OUTPUT
INPUT
3.000
1.000
-1.000
-3.000
-10.00
-6.000
-2.000
2.000
6.000
10.00

FIGURE 3.7

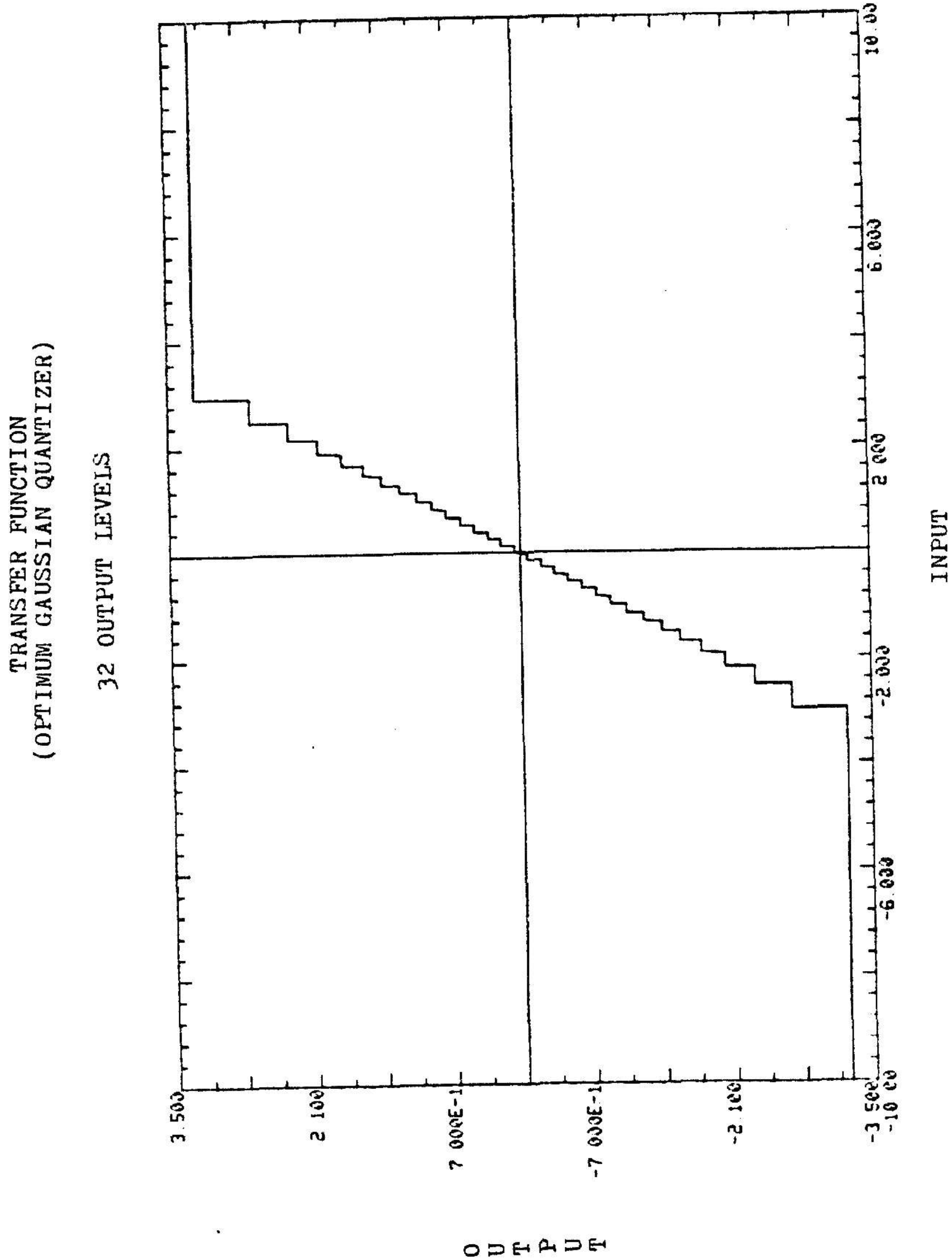

TRANSFER FUNCTION
(OPTIMUM GAUSSIAN QUANTIZER)
32 OUTPUT LEVELS
INPUT
OUTPUT
3.500
2.100
7.000E-1
-7.000E-1
-2.100
-3.500
-10.00
-6.000
-2.000
2.000
6.000
10.00

FIGURE 3.8

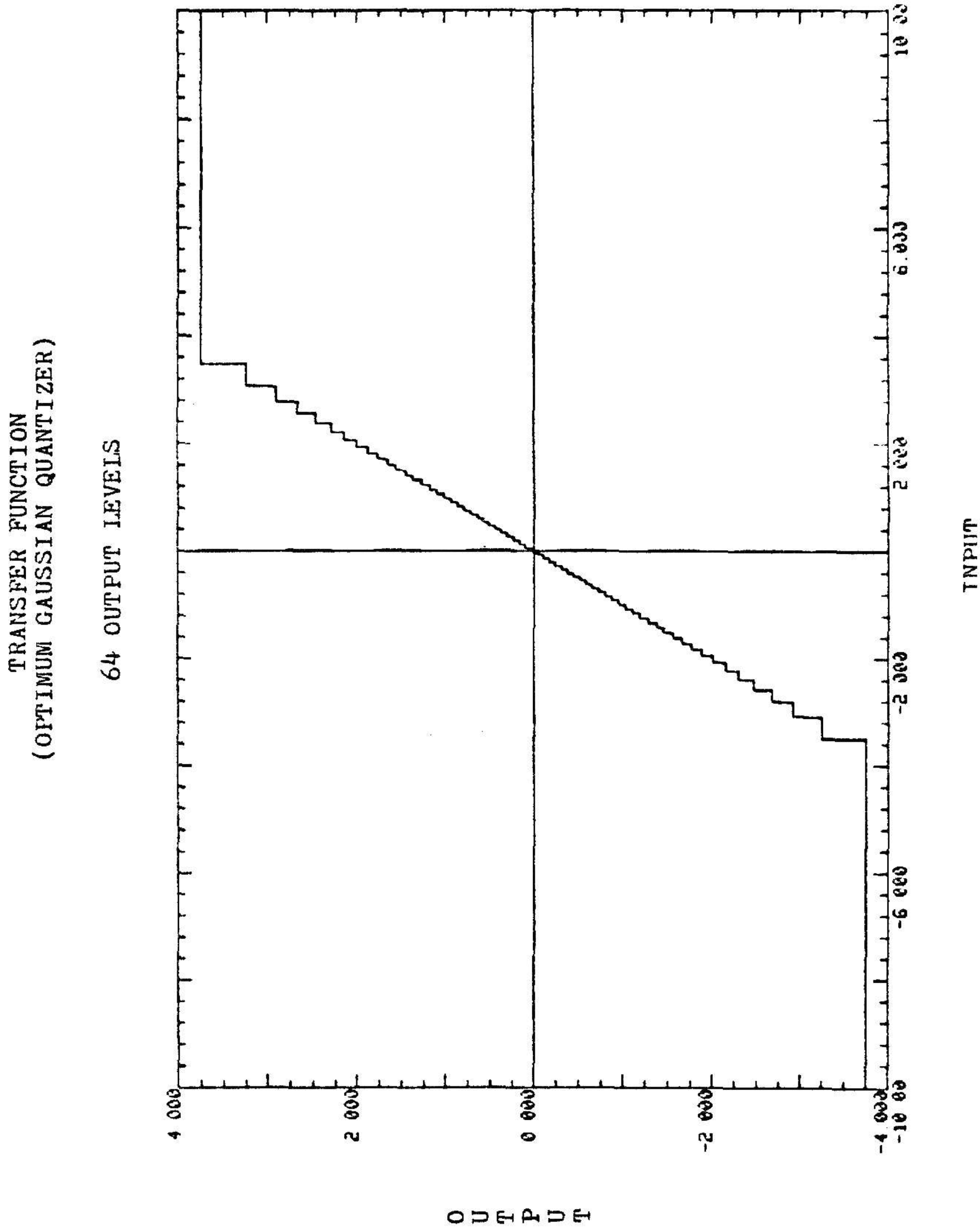

TRANSFER FUNCTION
(OPTIMUM GAUSSIAN QUANTIZER)
64 OUTPUT LEVELS
4.000
2.000
0.000
-2.000
-4.000
OUTPUT
-10.00
-6.000
-2.000
2.000
6.000
10.00
INPUT

FIGURE 3.9

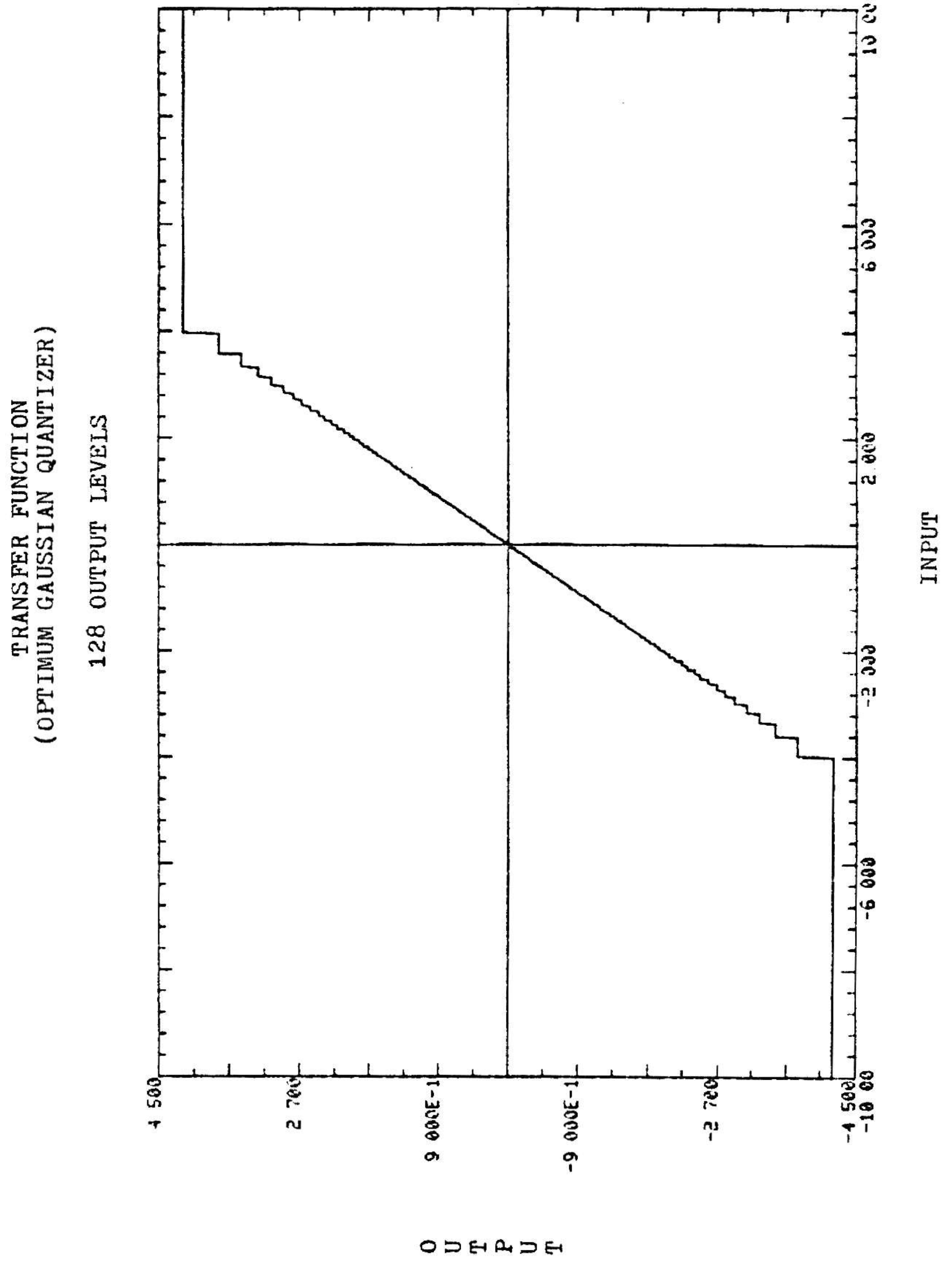

TRANSFER FUNCTION
(OPTIMUM GAUSSIAN QUANTIZER)
128 OUTPUT LEVELS
INPUT
OUTPUT
-10 00
-6 000
-2 000
2 000
6 000
10 00
-4 500
-2 700
-9 000E-1
9 000E-1
2 700
4 500

FIGURE 3.10

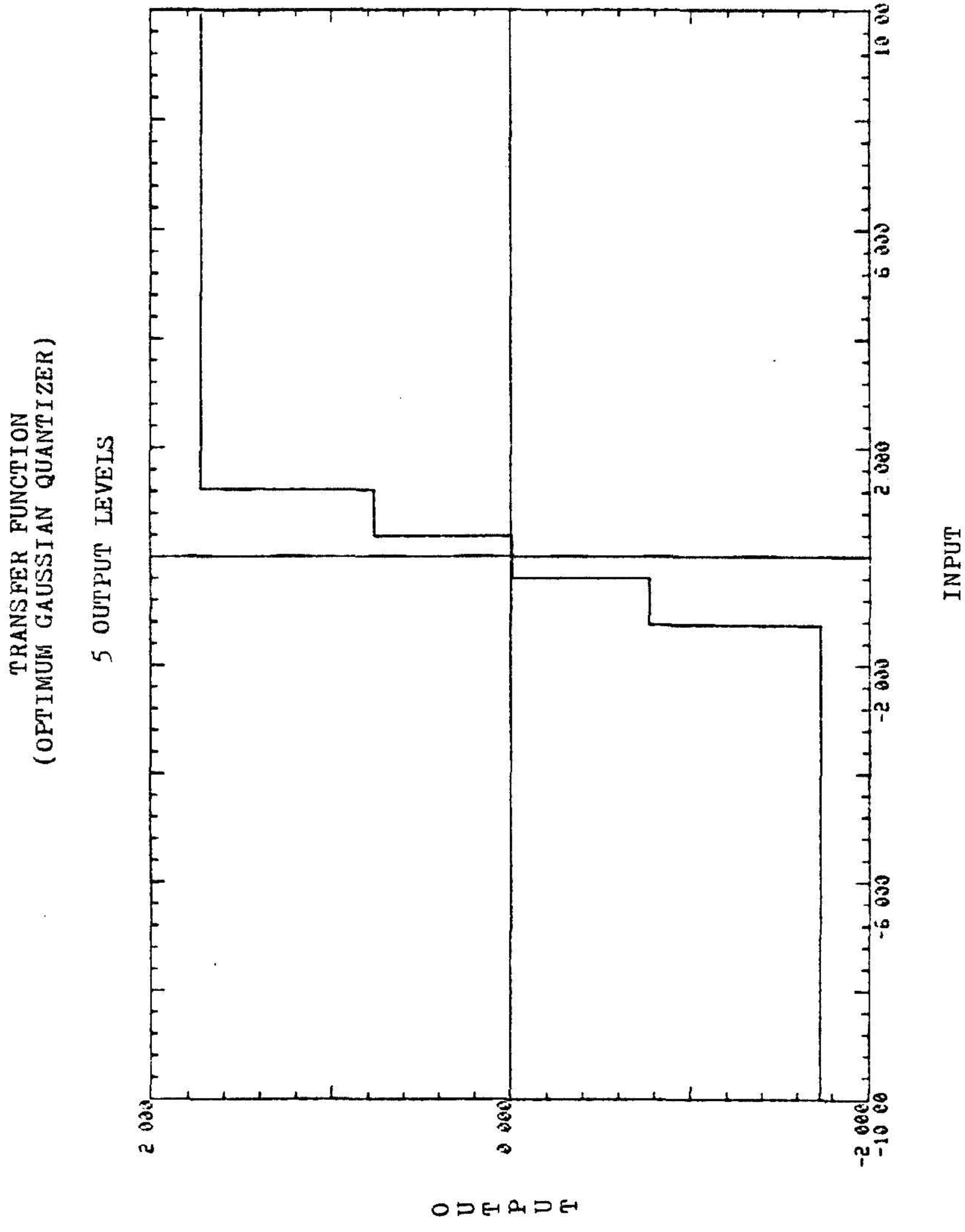

TRANSFER FUNCTION
(OPTIMUM GAUSSIAN QUANTIZER)
5 OUTPUT LEVELS
INPUT
OUTPUT
-10.00
-6.000
-2.000
2.000
6.000
10.00
-2.000
0.000
2.000

FIGURE 3.11

## Entropy Considerations

The notion of entropy is useful in assessing the performance of the optimum quantizer. The entropy of the output levels is given by [6]

$$\text{ENTROPY} = \begin{cases} -2 \sum_{i=1}^{M} (\text{Probability of } y_i)\log_2(\text{Probability of } y_i), & N \text{ EVEN} \\ -(2 \text{ Probability of } y_1)\log_2(2 \text{ Probability of } y_1) \\ -2 \sum_{i=2}^{M} (\text{Probability of } y_i)\log_2(\text{Probability of } y_i), & N \text{ ODD} \end{cases} \tag{3-10}$$

where entropy is defined as the average information (uncertainty) of the output levels. The smaller the uncertainty in the output, for a specified N, the better the performance. In (3-10) we note that

$$P(y_i) = \int_{x_i}^{x_{i+1}} p_X(x)dx \tag{3-11}$$

Values of the mean square error and entropy, obtained from (2-1) and (3-10) respectively, for the numbers of output levels 2 through 128 are tabulated in Table 3.3, and plotted in Figures 3.12 and 3.13. A comment about how these numerical values were obtained is now in order. The mean square error was calculated form (2-1)--i.e.,

$$E((X - Y)^2) = 2\sum_{j=2}^{M+1} \int_{x_{j-1}}^{x_j} (x - y_{j-1})^2 p_X(x)dx$$

This integral can be expressed as the sum of three integrals since $(x - y_{j-1})^2$ can be written $y_{j-1}^2 - 2xy_{j-1} + x^2$. Thus

$$E((X - Y)^2) = 2\sum_{j=2}^{M+1} [y_{j-1}^2 \int_{x_{j-1}}^{x_j} p_X(x)dx - 2y_{j-1} \int_{x_{j-1}}^{x_j} xp_X(x)dx + \int_{x_{j-1}}^{x_j} x^2 p_X(x)dx]$$

# ILLEGIBLE DOCUMENT

THE FOLLOWING DOCUMENT(S) IS OF POOR LEGIBILITY IN THE ORIGINAL

THIS IS THE BEST COPY AVAILABLE

## TABLE 3.3

## MEAN SQUARE ERROR AND ENTROPY

| NUMBER OF OUTPUT LEVELS | MSE | ENTROPY | NUMBER OF OUTPUT LEVELS | MSE | ENTROPY |
|---|---|---|---|---|---|
| 2 | .363380 | 1.0000 | 65 | .000623 | 5.7319 |
| 3 | .190174 | 1.5358 | 66 | .000604 | 5.7536 |
| 4 | .117481 | 1.9111 | 67 | .000587 | 5.7750 |
| 5 | .079941 | 2.2029 | 68 | .000571 | 5.7961 |
| 6 | .057977 | 2.4428 | 69 | .000554 | 5.8169 |
| 7 | .044000 | 2.6469 | 70 | .000539 | 5.8374 |
| 8 | .034547 | 2.8249 | 71 | .000524 | 5.8576 |
| 9 | .027853 | 2.9827 | 72 | .000510 | 5.8775 |
| 10 | .022936 | 3.1246 | 73 | .000496 | 5.8971 |
| 11 | .019219 | 3.2535 | 74 | .000483 | 5.9166 |
| 12 | .016339 | 3.3717 | 75 | .000470 | 5.9357 |
| 13 | .014062 | 3.4807 | 76 | .000458 | 5.9545 |
| 14 | .012231 | 3.5820 | 77 | .000446 | 5.9731 |
| 15 | .010736 | 3.6766 | 78 | .000434 | 5.9915 |
| 16 | .009500 | 3.7653 | 79 | .000424 | 6.0097 |
| 17 | .008466 | 3.8488 | 80 | .000413 | 6.0276 |
| 18 | .007592 | 3.9277 | 81 | .000403 | 6.0453 |
| 19 | .006847 | 4.0025 | 82 | .000393 | 6.0628 |
| 20 | .006207 | 4.0736 | 83 | .000383 | 6.0800 |
| 21 | .005653 | 4.1413 | 84 | .000374 | 6.0971 |
| 22 | .005170 | 4.2059 | 85 | .000366 | 6.1140 |
| 23 | .004745 | 4.2678 | 86 | .000357 | 6.1298 |
| 24 | .004371 | 4.3271 | 87 | .000349 | 6.1472 |
| 25 | .004040 | 4.3840 | 88 | .000341 | 6.1630 |
| 26 | .003745 | 4.4388 | 89 | .000335 | 6.1797 |
| 27 | .003481 | 4.4916 | 90 | .000327 | 6.1956 |
| 28 | .003245 | 4.5425 | 91 | .000320 | 6.2114 |
| 29 | .003032 | 4.5916 | 92 | .000313 | 6.2270 |
| 30 | .002839 | 4.6392 | 93 | .000307 | 6.2424 |
| 31 | .002663 | 4.6852 | 94 | .000300 | 6.2577 |
| 32 | .002504 | 4.7297 | 95 | .000294 | 6.2728 |
| 33 | .002358 | 4.7730 | 96 | .000288 | 6.2877 |
| 34 | .002225 | 4.8150 | 97 | .000282 | 6.3025 |
| 35 | .002102 | 4.8557 | 98 | .000276 | 6.3172 |
| 36 | .001989 | 4.8954 | 99 | .000271 | 6.3317 |
| 37 | .001886 | 4.9340 | 100 | .000265 | 6.3460 |
| 38 | .001791 | 4.9715 | 101 | .000260 | 6.3602 |
| 39 | .001703 | 5.0082 | 102 | .000255 | 6.3743 |
| 40 | .001620 | 5.0440 | 103 | .000250 | 6.3882 |
| 41 | .001544 | 5.0788 | 104 | .000245 | 6.4020 |
| 42 | .001473 | 5.1128 | 105 | .000240 | 6.4157 |
| 43 | .001406 | 5.1461 | 106 | .000236 | 6.4292 |
| 44 | .001344 | 5.1786 | 107 | .000231 | 6.4426 |
| 45 | .001296 | 5.2104 | 108 | .000227 | 6.4559 |
| 46 | .001232 | 5.2414 | 109 | .000223 | 6.4691 |
| 47 | .001180 | 5.2719 | 110 | .000218 | 6.4821 |
| 48 | .001132 | 5.3017 | 111 | .000214 | 6.4950 |
| 49 | .001087 | 5.3309 | 112 | .000210 | 6.5078 |
| 50 | .001045 | 5.3595 | 113 | .000207 | 6.5205 |
| 51 | .001005 | 5.3876 | 114 | .000203 | 6.5336 |
| 52 | .000968 | 5.4151 | 115 | .000200 | 6.5455 |
| 53 | .000933 | 5.4421 | 116 | .000198 | 6.5581 |
| 54 | .000899 | 5.4686 | 117 | .000194 | 6.5704 |
| 55 | .000867 | 5.4947 | 118 | .000191 | 6.5826 |
| 56 | .000837 | 5.5202 | 119 | .000188 | 6.5946 |
| 57 | .000808 | 5.5453 | 120 | .000184 | 6.6065 |
| 58 | .000781 | 5.5700 | 121 | .000181 | 6.6185 |
| 59 | .000755 | 5.5943 | 122 | .000178 | 6.6302 |
| 60 | .000730 | 5.6182 | 123 | .000175 | 6.6402 |
| 61 | .000706 | 5.6416 | 124 | .000173 | 6.6535 |
| 62 | .000684 | 5.6648 | 125 | .000170 | 6.6649 |
| 63 | .000663 | 5.6875 | 126 | .000167 | 6.6764 |
| 64 | .000642 | 5.7099 | 127 | .000165 | 6.6877 |
| | | | 128 | .000163 | 6.6989 |

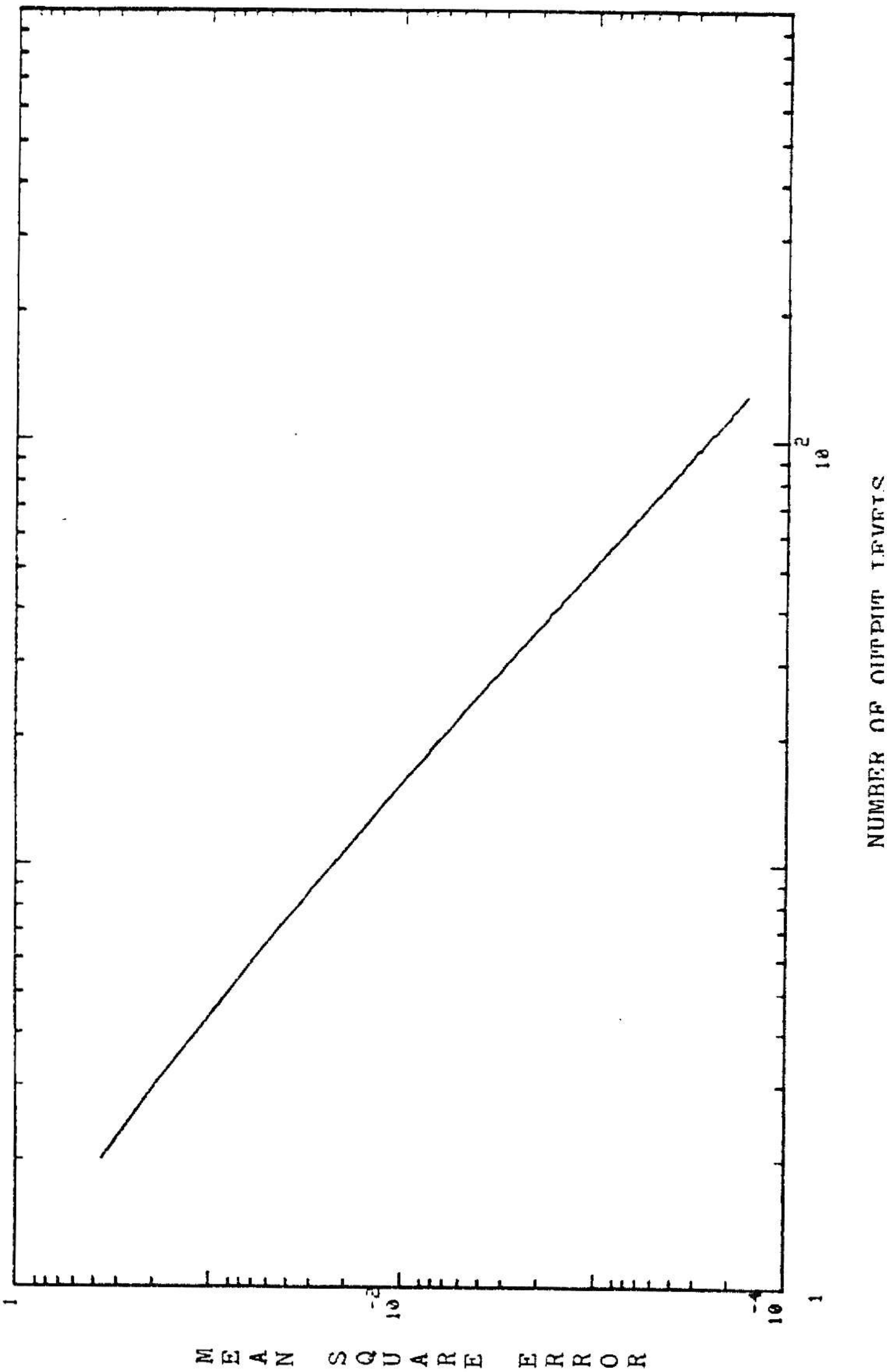



FIGURE 3.12

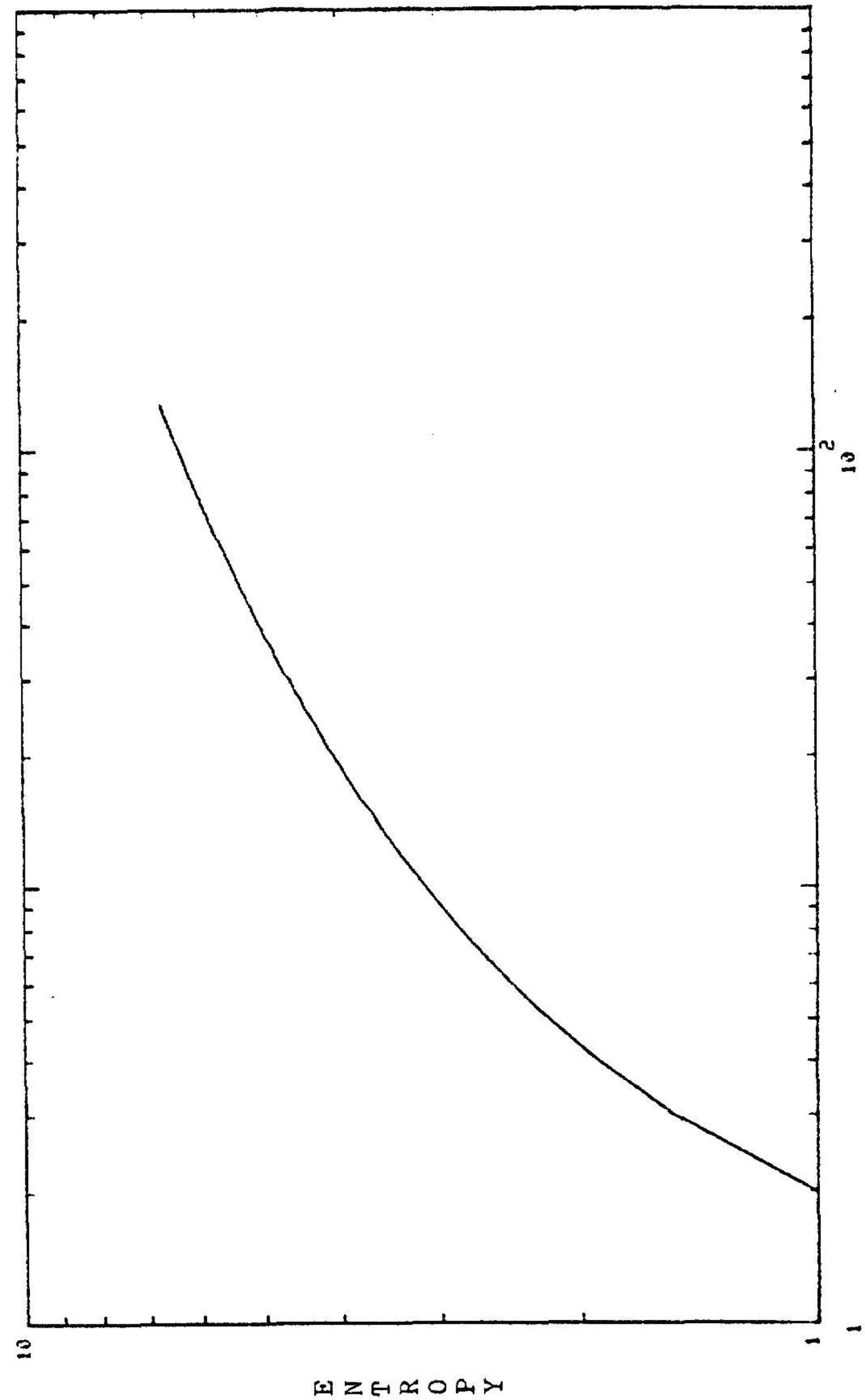



FIGURE 3.13

The forms of the first two integrals were discussed earlier in (3-3). The last integral can be integrated by parts as follows:

$$\int u \frac{dv}{dx} dx = uv - \int v \frac{du}{dx} dx \tag{3-13}$$

where u=x and $\frac{dv}{dx} = xp_x(x)$. This yields

$$\int_{x_{j-1}}^{x_j} x^2 p_X(x) dx = -xp_X(x) + \int_{x_{j-1}}^{x_j} p_X(x) dx \tag{3-14}$$

Note that (3-14) is valid only when $p_X(x)$ is gaussian, and that the last integral is of the form previously discussed in (3-3).

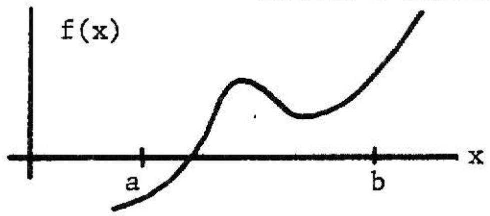
The tabulated values for mean square error and entropy in Table 3.3 compare very closely with those published by Max [1].

# APPENDIX 3.1

This appendix contains the iterative procedure to determine the zeroes of functions known as Newton's method.

Note if the initial value is not sufficiently close to the actual zero this method will not converge. If no prior knowledge is known about the location of the zero(es) it is suggested that another method that always converges (but is very slow) such as the bisection method be used for a few iterations to insure that the initial guess is sufficiently close to the actual zero. Newton's method may then be used for subsequent iterations to speed the process up.

NEWTON'S METHOD




SOLVE $f(x) = 0$

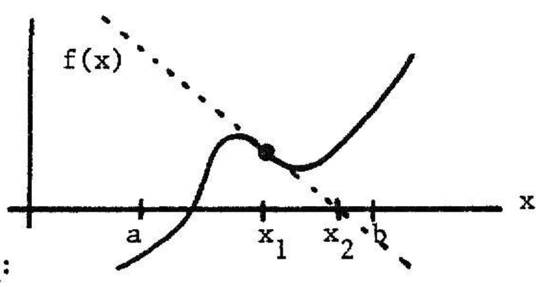
(1) Pick any $x_1$ in interval $\varepsilon[a,b]$.

(2) Draw tangent to graph at $f(x_1)$.

(3) Where tangent crosses horizontal axis is $x_2$.

(4) Draw tangent to Graph at $f(x_2)$ etc.

** If tangent to graph lies outside the interval [a,b] this method fails.




Derivation:

$$y - f(x_n) = [f^{(1)}(x_n)](x - x_n)$$

Next x is where $y = 0$

Thus

$$x_{n+1} = x_n - \frac{f(x_n)}{f^{(1)}(x_n)}$$

Iterative Procedure to solve (2-8)

$$\int_{x_j}^{x_{j+1}} (x - y_j) p_X(x) dx = 0$$

where $x_j$, $y_j$ are known solve for $x_{j+1}$

* Use Newton's method for finding zeroes of functions

$$g(x) = 0 = \int_{x_j}^{x_{j+1}} (x - y_j) p_X(x) dx = 0$$

$$\hat{x}_{j+1} = x_{j+1} - \frac{g(x_{j+1})}{g^{(1)}(x_{j+1})}$$

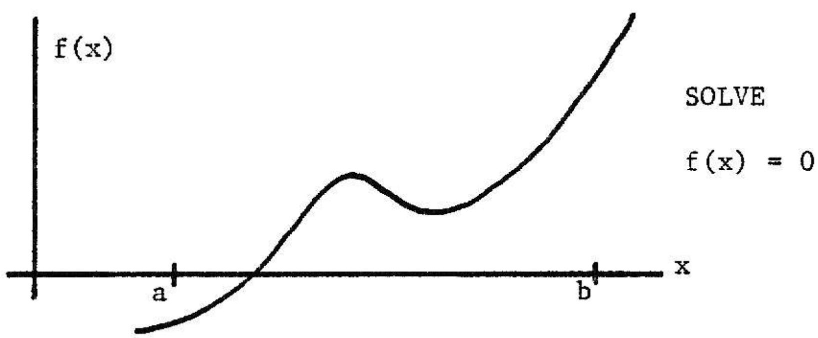
$$\hat{x}_{j+1} = x_{j+1} - \frac{\int_{x_j}^{x_{j+1}} (x - y_j) p_X(x) dx}{(x_{j+1} - y_j) p_X(x_{j+1})}$$

where $(x_{j+1})_{initial} = 2y_j - x_j$

# APPENDIX 3.2

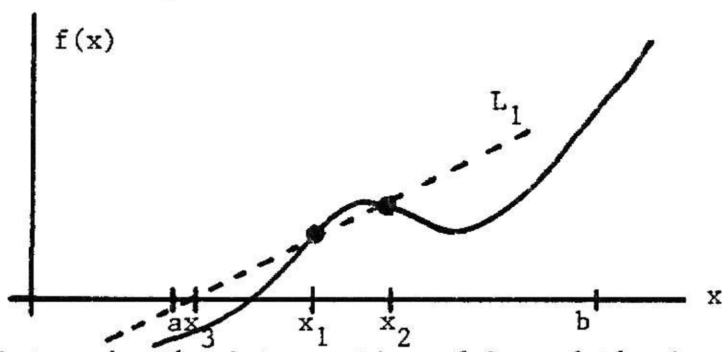
This appendix contains the iterative procedure to determine the zeroes of functions known as the secant method.

Note if the initial value is not close to the zero this method will not converge. If no prior knowledge is known about the location of the zero(es) it is suggested that another method that always converges (but very slow) such as the bisection method be used for a few iterations to insure that the initial guess is sufficiently close to the actual zero. The secant method may be used for subsequent iterations to speed the process up.



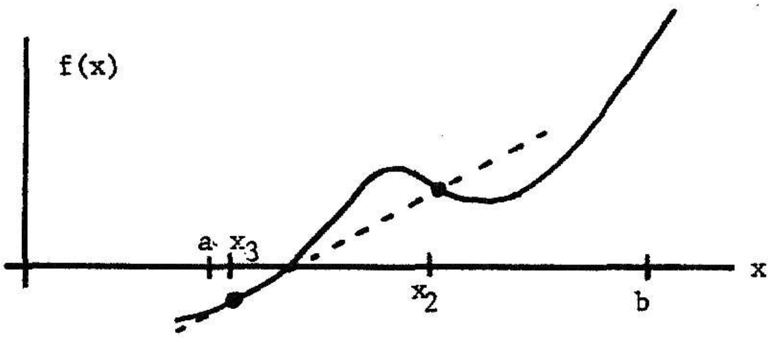

(1) Pick any two points in the interval $\varepsilon[a,b]$, ie, $x_1$ and $x_2$

(2) Connect $(x_1, f(x_1))$ and $(x_2, f(x_2))$ by a straight line $L_1$



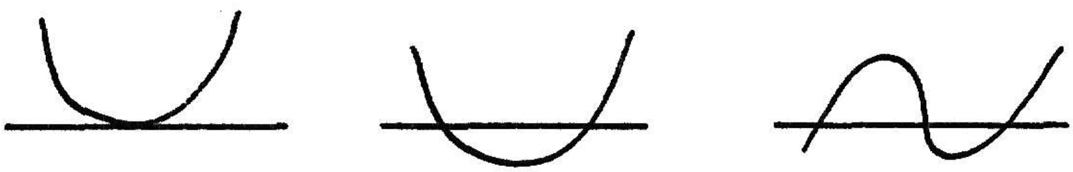

(3) Let $x_3$ be the intersection of $L_1$ and the interval [a,b] (ie, the horizontal axis)

(4) Connect $(x_2, f(x_2))$ and $(x_3, f(x_3)$ etc.




Caution: Secant line may not cross horizontal axis within the interval [a,b]

** The sequence $\{x_n\}$ will converge to a zero of f(x) if $x_1$ and $x_2$ are close enough to a zero and if the zero is a simple zero



Not a simple zero (slope = 0) Simple zero Simple zero

Derivation:

$$y - f(x_1) = \frac{f(x_2) - f(x_1)}{x_2 - x_1} (x - x_1)$$

Next x is where y = 0 rearranging,

$$x = x_1 - \frac{f(x_1)\ [x_1 - x_2]}{f(x_2) - f(x_1)}$$

# APPENDIX 3.3

This appendix contains the program to generate the parameters $X_i$ and $Y_i$, which completely describe the optimum gaussian quantizer for a finite number of output levels, N.

THE OPTIMUM QUANTIZER

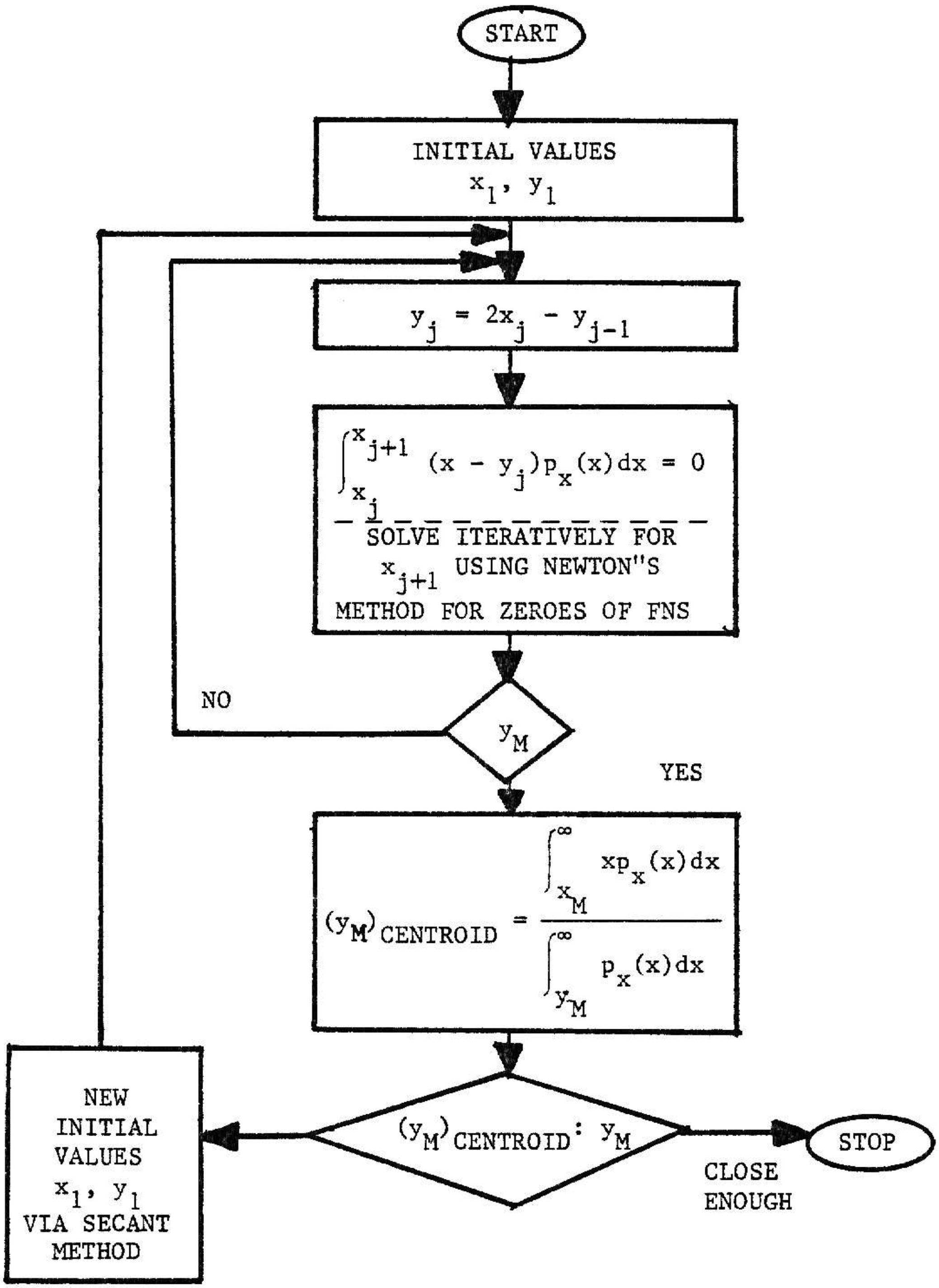



FIGURE 3.14

```
TABLGEN5.FR             7/15/1982   0: 1:32   DIR DPO        Page   1

C       THIS ROUTINE COMPUTES THE CUT-POINTS AND REP-VALUES FOR
C       A GUASSIAN PROBABILITY DENSITY AND MSE DISTORTION MEASURE.
C
        COMPILER FREE
        DOUBLE PRECISION X,Y,MSE,ENTROPY,AREA1,AREA2
        DOUBLE PRECISION X0,X1,X2,P0,P2,XN,YN
        DOUBLE PRECISION P1,Y0,GAP,Z0,Z1,W0,W1,AREA,TAIL,A0,A1,W2,DABS,Q1,Q0
        REAL MSE1,ENTR1
        DIMENSION TABLEG(128,2)
        CALL OPENW(4,'MEAN SQUARED ERROR OUTPUT FILENAME ?  ',0,SIZE)
        CALL OPENW(5,'ENTROPY OUTPUT FILENAME ?  ',0,SIZE)
        ITTO=10
        ITTI=11
        ITTTY=12
C
C       GET THE NUMBER OF LEVELS
C
        WRITE(ITTO,5)
 5      FORMAT(' ')
 5000   ACCEPT 'NUMBER OF OUTPUT LEVELS ?  ', NLEV
        N=NLEV/2
C
C       ESTIMATE THE VALUE OF THE FIRST CUT-POINT OR REP-VALUE
C
        P1=2.1613D0/DBLE(FLOAT(NLEV))-1.653D0/DBLE(FLOAT(NLEV)**2.0)
C
C       PERFORM TEN ITERATIONS
C
        WRITE(ITTTY,6000) NLEV
 6000   FORMAT('1',12X,I3,'  OUTPUT LEVELS')
        DO 1073 II=1,10
        WRITE(ITTTY,6500) NLEV
 6500   FORMAT('0',50X,'N=',I3)
        WRITE(ITTTY,2040) II
 2040   FORMAT(' ',12X,'ITERATION NUMBER',I3,Z)
        WRITE(ITTTY,2045)
 2045   FORMAT('+',4X,'J',7X,'X(J)',12X,'Y(J)')
C
C       INITIALIZE THE FIRST CUT-POINT AND REP-VALUE
C
        IF(N*2.EQ.NLEV) GO TO 1100
C
C       ODD NUMBER OF LEVELS
C
        X0=P1
        Y0=2.0D0*P1
        GO TO 1110
 1100   CONTINUE
C
C       EVEN NUMBER OF LEVELS
C
        X0=0.0D0
        Y0=P1
 1110   CONTINUE
        I=1
C
C       THE FIRST CUT-POINT VALUE
```

```
TABLGEN5.FR            7/15/1982   0: 1:32   DIR DP0        Page  2

C
      WRITE(ITTTY,2000) I,X0,Y0
 2000 FORMAT(' ',I36,F13.6,F16.6)
      TABLEG(I,1)=X0
      TABLEG(I,2)=Y0
      IF(NLEV.LT.4)  GO TO 1000
C
C     DETERMINE THE REST OF THE CUT-POINTS AND REP-VALUES
C
      DO 1000 I=2,N
      CALL CUTPT(X0,Y0,X1)
C
      X0=X1
      Y0=2.0D0*X0-Y0
      WRITE(ITTTY,2000) I,X0,Y0
      TABLEG(I,1)=X0
      TABLEG(I,2)=Y0
 1000 CONTINUE
C
C     USING AN ALTERNATE METHOD,
C     DETERMINE THE LAST REP-VALUE, Y(N), BY COMPUTING
C     THE CENTROID OF THE AREA FROM X(N) TO INFINITY
C
      XN=X0
      CALL ERF(XN,AREA)
      YN=DEXP(-XN**2/2.0D0)/DSQRT(2.0D0*3.141592D0)/(0.5D0-AREA)
      WRITE(ITTTY,6700) YN
 6700 FORMAT(' ',17X,'SECANT METHOD......',6X,'Y(N) =',1X,F16.6)
C
C     COMPUTE THE DIFFERENCE BETWEEN THE TWO VALUES AND UPDATE
C     THE STARTING VALUE, P1
C
      Q1=Y0-YN
      IF(DABS(Q1/YN).LT.1.0D-6) GO TO 3000
      IF(II.EQ.1) P2=0.99D0*P1
      IF(II.NE.1) P2=(P1*Q0-P0*Q1)/(Q0-Q1)
      P0=P1
      P1=P2
      Q0=Q1
 1073 CONTINUE
C
C     NOW THAT THE OPTIMUM PARAMETERS HAVE BEEN
C     DETERMINED, CALCULATE THE MEAN SQUARED
C     ERROR FOR THE CURRENT NUMBER OF
C     OUTPUT LEVLELS
C
 3000 MSE=0.0D0
      IF (NLEV.EQ.2) GO TO 6500
      IF (N*2.EQ.NLEV) GO TO 7000
C
C     ODD NUMBER OF LEVELS
C
      X=TABLEG(1,1)
      Y=0.0D0
      CALL ERF(X,AREA)
      MSE=2.0D0*(((-X+2.0D0*Y)*DEXP(-X**2/2.0D0)
     & -2.0D0*Y)/DSQRT(2.0D0*3.141592D0)
```

```
     &  +(1.0D0+Y**2)*AREA)
        IF (NLEV.EQ.3) GO TO 8500
C
C       ODD OR EVEN NUMBER OF LEVELS
C
 7000   DO 8000 I=1,N-1
        X1=TABLEG(I,1)
        X2=TABLEG(I+1,1)
        Y=TABLEG(I,2)
        CALL ERF(X1,AREA1)
        CALL ERF(X2,AREA2)
        AREA=AREA2-AREA1
        MSE=2.0D0*(((-X2+2.0D0*Y)*DEXP(-X2**2/2.0D0)
     &  -(-X1+2.0D0*Y)*DEXP(-X1**2/2.0D0))
     &  /DSQRT(2.0D0*3.141592D0)
     &  +(1.0D0+Y**2)*AREA)+MSE
 8000   CONTINUE
C
C       INTEGRATE FROM X(N) TO INFINITY
C
 8500   X=TABLEG(N,1)
        Y=TABLEG(N,2)
        CALL ERF(X,AREA)
        MSE=2.0D0*(-(-X+2.0D0*Y)*DEXP(-X**2/2.0D0)
     &  /DSQRT(2.0D0*3.141592D0)
     &  +(1.0D0+Y**2)*(0.5D0-AREA))+MSE
        MSE1=SNGL(MSE)
        WRITE (ITTTY,1) NLEV,MSE
 1      FORMAT('-',12X,'MEAN SQUARED ERROR FOR ',I3,' OUTPUT LEVELS = ',F9.6)
        WRITE BINARY(4) MSE1
C
C       CALCULATE THE ENTROPY
C
        ENTROPY=0.0D0
        IF(NLEV.EQ.2) GO TO 9300
        IF (N*2.EQ.NLEV) GO TO 9000
C
C       ODD NUMBER OF LEVELS
C
        X=TABLEG(1,1)
        CALL ERF(X,AREA)
        ENTROPY=(2.0D0*AREA*DLOG(2.0D0*AREA))/DLOG(2.0D0)
        IF (NLEV.EQ.3) GO TO 9300
C
C       ODD OR EVEN NUMBER OF LEVELS
C
 9000   DO 9500 I=1,N-1
        X1=TABLEG(I,1)
        X2=TABLEG(I+1,1)
        CALL ERF(X1,AREA1)
        CALL ERF(X2,AREA2)
        AREA=AREA2-AREA1
        ENTROPY=2.0D0*(AREA*DLOG(AREA))/DLOG(2.0D0)+ENTROPY
 9500   CONTINUE
C
C       INTEGRATE FROM X(N) TO INFINITY
C
 9300   X=TABLEG(N,1)
        CALL ERF(X,AREA)
        AREA=0.5D0-AREA
        ENTROPY=-(2.0D0*(AREA*DLOG(AREA))/DLOG(2.0D0)+ENTROPY)
        ENTR1=SNGL(ENTROPY)
        WRITE (ITTTY,2) NLEV,ENTROPY
 2      FORMAT('-',12X,'ENTROPY FOR ',13X,I3,' OUTPUT LEVELS = ',F9.6)
        WRITE BINARY(5) ENTR1
        IF (NLEV.GT.0) GO TO 5000
        STOP
        END
```

```
CUTPT2.FR              7/15/1982   0: 0:22   DIR DP0          Page   1

C       THIS SUBROUTINE COMPUTES THE NEXT CUT-POINT, X1, GIVEN THE
C       PREVIOUS CUT-POINT AND REP-VALUE
C
        SUBROUTINE CUTPT(X0,Y0,X1)
        DOUBLE PRECISION X0,Y0,X1,EPS,AREA,AREA0,AREA1,DELX1,PROB,FUNC,DABS
        DOUBLE PRECISION TAIL
        EPS=1.0D-9
C
C       INTEGRATE THE FUNCTION FROM X0 TO Y0
C
        CALL FPX(X0,Y0,AREA0)
C
        X1=2.0D0*Y0-X0
C
C       ITERATE UP TO 100 TIMES TO DETERMINE X1
C
        DO 1000 I=1,100
        CALL FPX(X1,Y0,AREA1)
        AREA=AREA1-AREA0
        DELX1=-(AREA)/(PROB(X1)*FUNC(X1,Y0))
        IF(DABS(DELX1/X1).LT.EPS) RETURN
        X1=X1+DELX1
 1000   CONTINUE
        RETURN
        END
```

```
ERF2.FR                7/15/1982   0: 0:21   DIR DP0          Page   1

C       THIS SUBROUTINE EVALUATES THE INTEGRAL OF THE
C       GAUSSIAN PROBABILITY DISTRIBUTION VIA AN
C       INFINITE SERIES REPRESENTATION
C
C
        SUBROUTINE ERF(X,AREA)
        DOUBLE PRECISION X,W,AREA,AREA1,DEXP,DSQRT,DBLE,DABS
        W=1.0D0
        AREA1=X*DEXP(-X**2/2.0D0)/DSQRT(2.0D0*3.141592D0)
        DO 100 J=1,100
        K=2*J+1
        W=W*DBLE(FLOAT(K))
        AREA=(X/DSQRT(2.0D0))**K*2.0D0**J/W*DEXP(-X**2/2.0D0)/DSQRT(3.141592D0)+AREA1
        IF (DABS(AREA1-AREA).LT.1.0D-9) RETURN
        AREA1=AREA
  100   CONTINUE
        TYPE "NEED MORE TERMS OF THE INFINITE SERIES"
        RETURN
        END
```

```
FPX.FR                 7/15/1982   0: 0:25   DIR DP0          Page   1

        SUBROUTINE FPX(X,Y,AREA)
        DOUBLE PRECISION X,Y,AREA,DEXP,DSQRT,A
        CALL ERF(X,A)
        AREA=(1-DEXP(-X**2/2.0D0))/DSQRT(2.0D0*3.141592D0)-Y*A
        RETURN
        END
```

```
PROB.FR                7/15/1982    0: 0:24    DIR DF0         Page   1

      DOUBLE PRECISION FUNCTION PROB(X)
      DOUBLE PRECISION X,DEXP,DSQRT
      PROB=DEXP(-X**2/2.0D0)/DSQRT(2.0D0*3.141592D0)
      RETURN   -
      END
```

```
FUNC.FR                7/15/1982    0: 0:24    DIR DF0         Page   1

      DOUBLE PRECISION FUNCTION FUNC(X,YO)
      DOUBLE PRECISION X,YO
      FUNC=X-YO
      RETURN
      END
```

# CHAPTER IV

## EXPERIMENTAL RESULTS

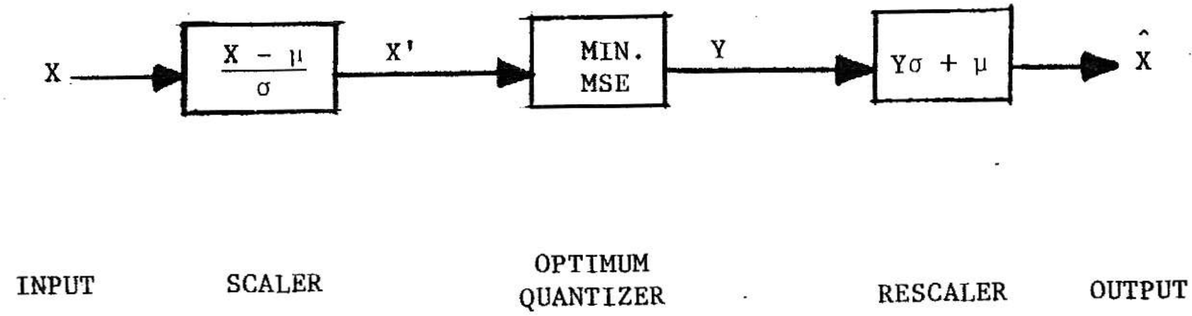
The experimental results reported in this chapter were carried out at the Signal Processing Laboratory of the Electrical Engineering Department at KSU.

A random gaussian input sequence of specified mean and variance was generated using the program NORMGEN, listed in Appendix 4.1. The mean and variance of this input sequence were then calculated via the program, XBAR. These actual values are necessary since the input sequence is scaled to zero mean and unit variance, as depicted in Figure 4.1. The scaled sequence is then processed through the quantizer. The quantizer's output is also gaussianly distributed with zero mean and unit variance. Hence it must be rescaled to the actual mean and variance of the input sequence. The mean square error was then calculated to access whether the rescaled output sequence, for the given number of output levels N, is a faithful reproduction of the input sequence. The mean square error can be calculated at two places in Figure 4.1; at X', Y and at X, $\hat{X}$. Both results are useful. The first, at X', Y, has been discussed earlier in (2-1). It provides an experimental measure for comparison against the theoretical values tabulated in Table 3.3. The second, at X, $\hat{X}$, establishes how faithful a reproduction the rescaled quantizer output sequence is of the input sequence. Theses two mean square errors are related. Using the notation of Figure 4.1 we have

$$x = x'\sigma + \mu$$

$$\hat{x} = y\sigma + \mu$$

$$\frac{(x - \hat{x})^2}{\text{NPTS}} = \frac{((x'\sigma + \mu) - (y\sigma + \mu))^2}{\text{NPTS}} \qquad (4\text{-}1)$$




$X$: Input random variable
$\mu$: Mean of X
$\sigma$: Standard deviation of X
$\hat{X}$: Reconstruction of X; i.e., quantizer output

THE OPTIMUM QUANTIZATION PROCESS

FIGURE 4.1

where NPTS is the total number of data points. Rewriting (4-1) produces the desired relationship

$$(x - \hat{x})^2 = \sigma^2(x' - y)^2 \tag{4-2}$$

The entire optimum quantization process of Figure 4.1 is implemented by the program QUANTIZE1, listed in Appendix 4.2.

Two examples of the entire quantization process are now presented. The first example has gaussian input sequence with mean 0.044654 and variance 0.93953. The second example has a gaussian input sequence with mean 250.11 and variance 5.6372. Both examples follow the same format. The input sequence and the rescaled quantizer output sequences are shown for 2,4,8,16,32,64, and 128 output levels, which correspond to 1,2,3,4,5,6, and 7 output bits respectively. Perhaps more informative of what is actually happening are the histograms of the rescaled quantizer outputs, each of which consist of 2048 data points. The histograms were formed by dividing the range between the minimum and maximum values of each sequence into 450 bins, and then determining how many of the 2048 data points fall in each of these 450 bins. The program HISTPLOT was used to compute these histograms. We observe that the histograms resemble gaussian distributions as the number of output levels is increased. Finally the mean square error is calculated for each of the output sequences relative to the input sequence. These values are compared to the theoretical values published by Max [1], except N=64 and N=128 which Max did not calculate. These two exceptions were earlier tabulated in Table 3.3. The mean square error is then rescaled, due to its dependence on the variance of the input sequence, via the relationship obtained in (4-2).

## EXAMPLE 1

A random gaussian input sequence with a mean and variance equal to 0.044654 and 0.93953, respectively. Note that the input sequence is in floating point arithmetic, while the output sequences are in fixed point arithmetic.

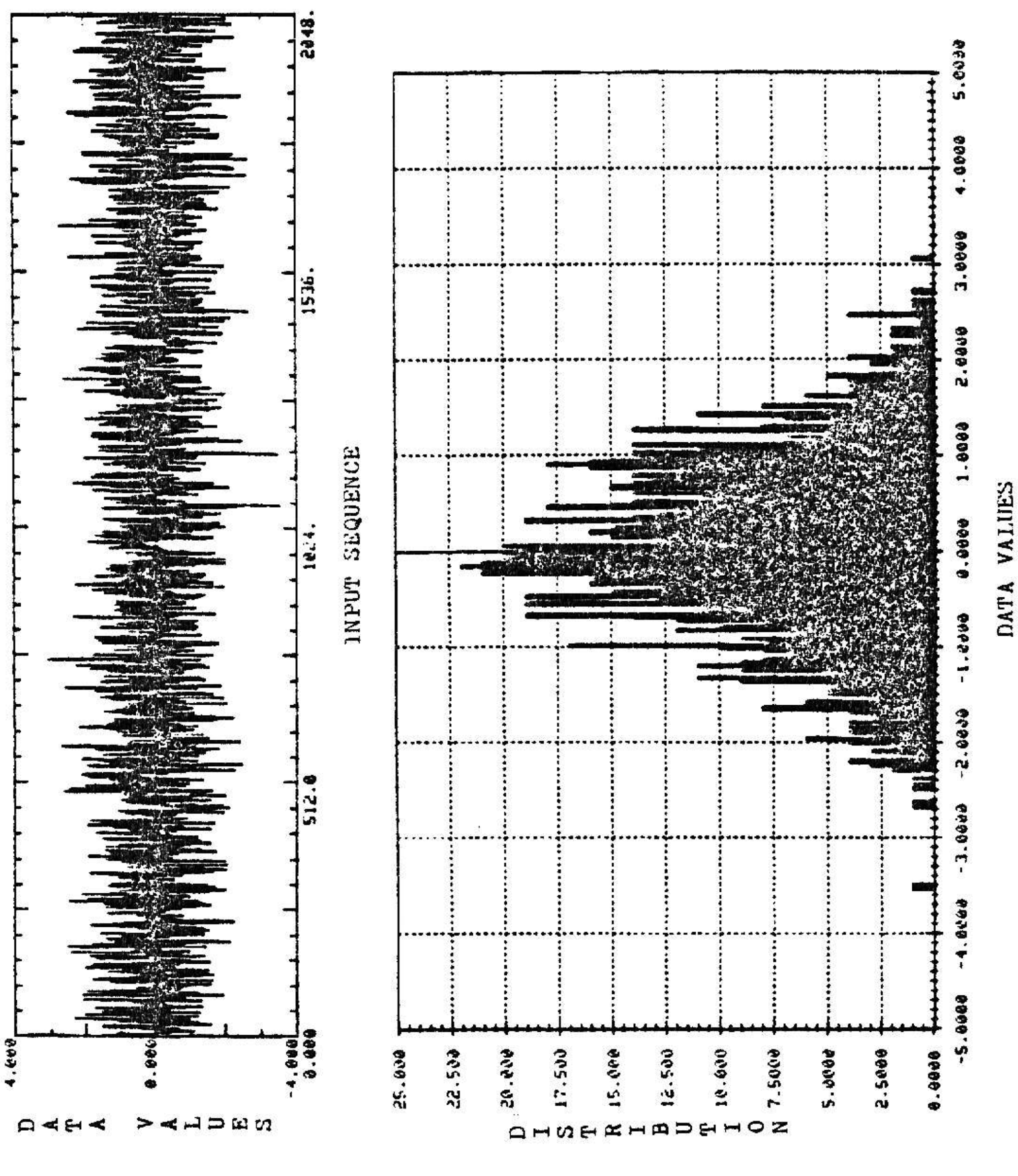



FIGURE 4.2

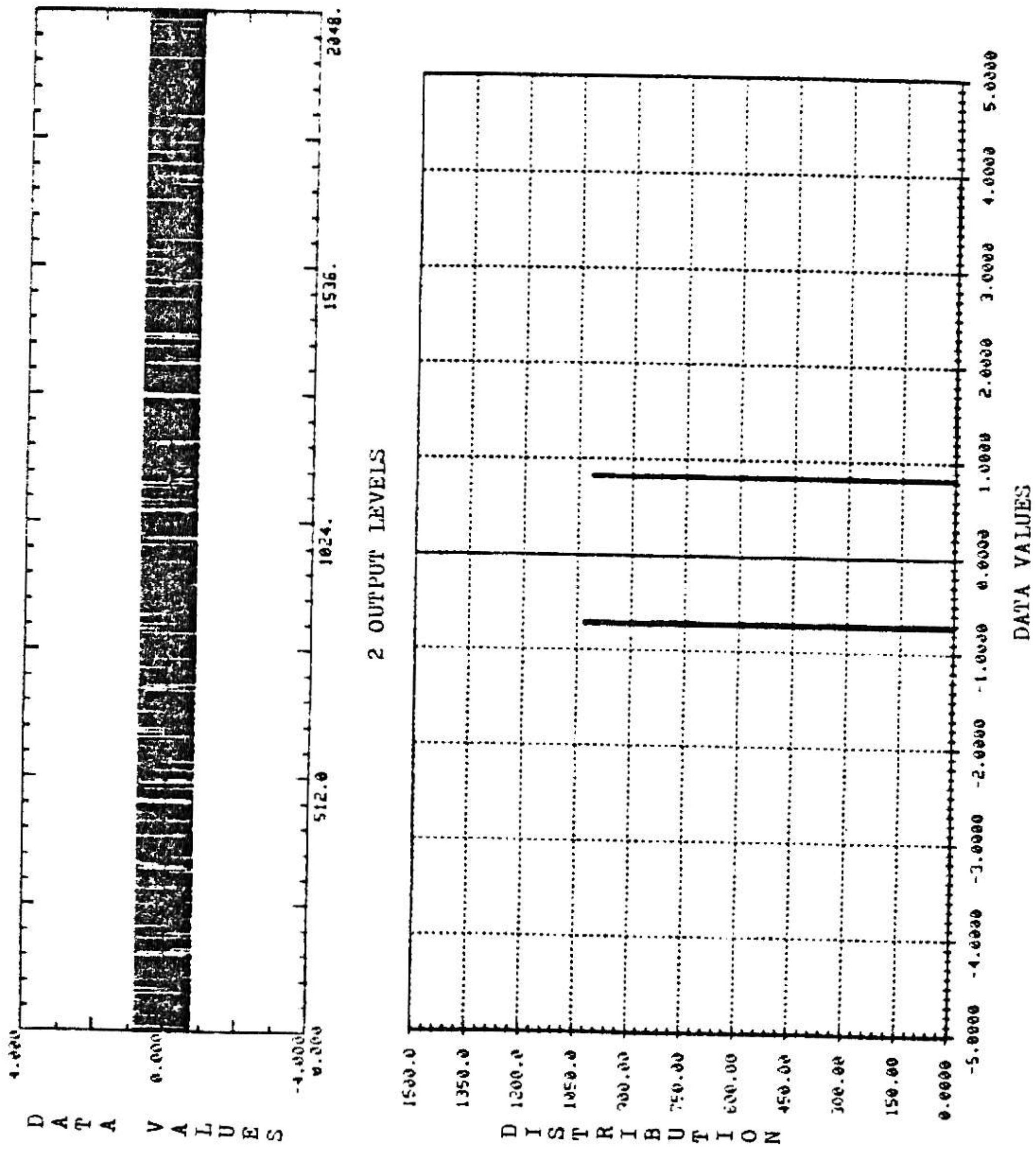
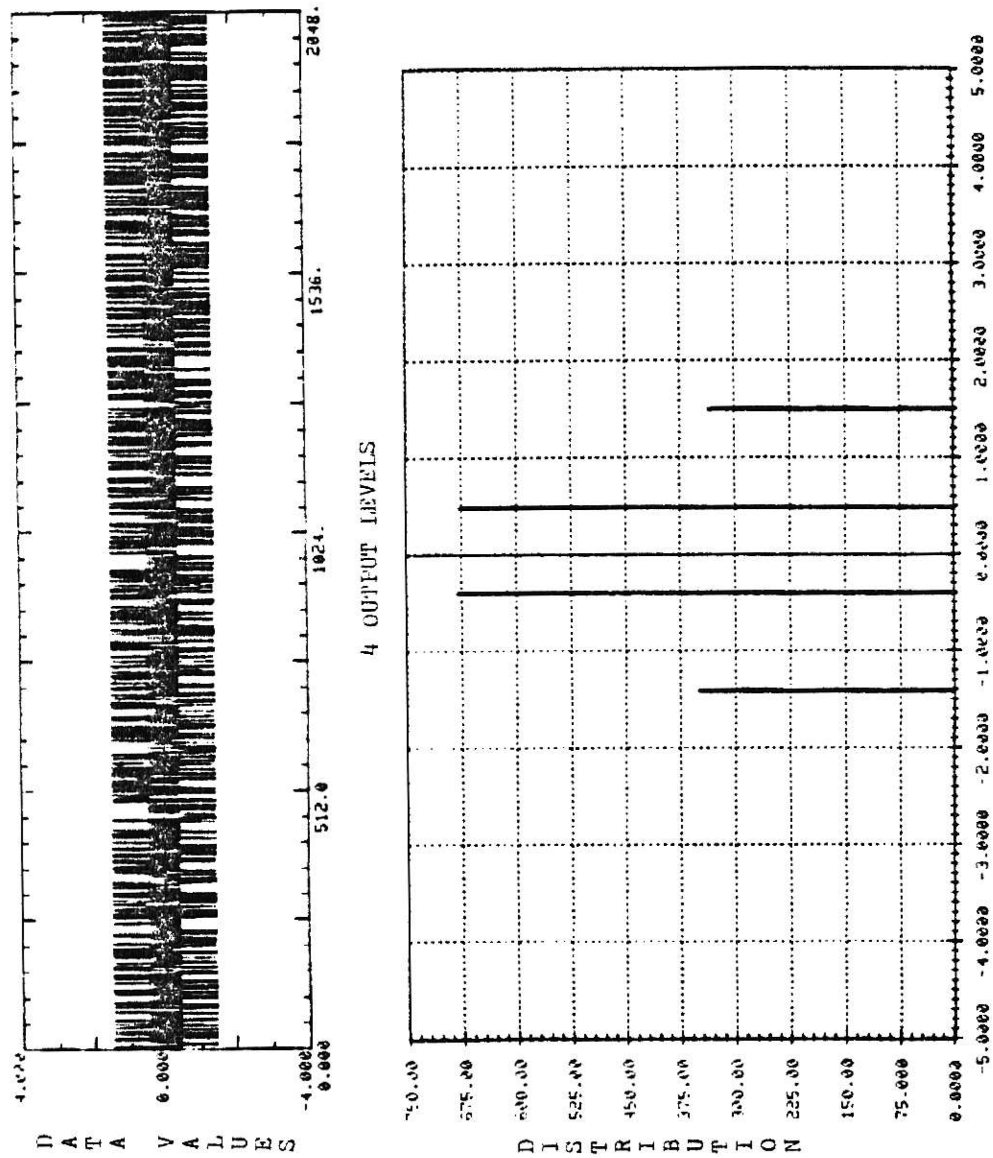



FIGURE 4.3



DATA VALUES
512.0
1024.
1536.
2048.
4 OUTPUT LEVELS
DISTRIBUTION
DATA VALUES


FIGURE 4.4

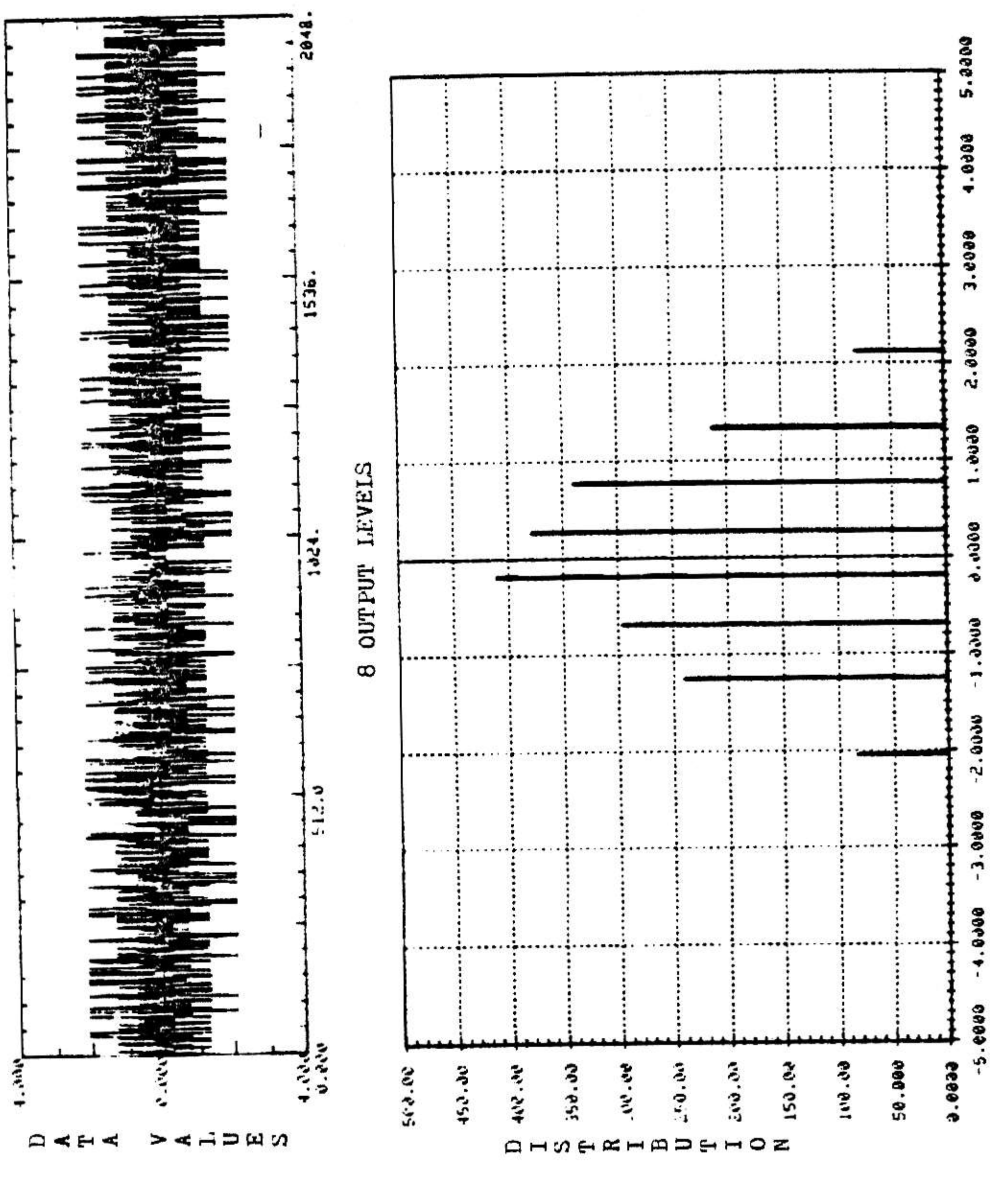



FIGURE 4.5

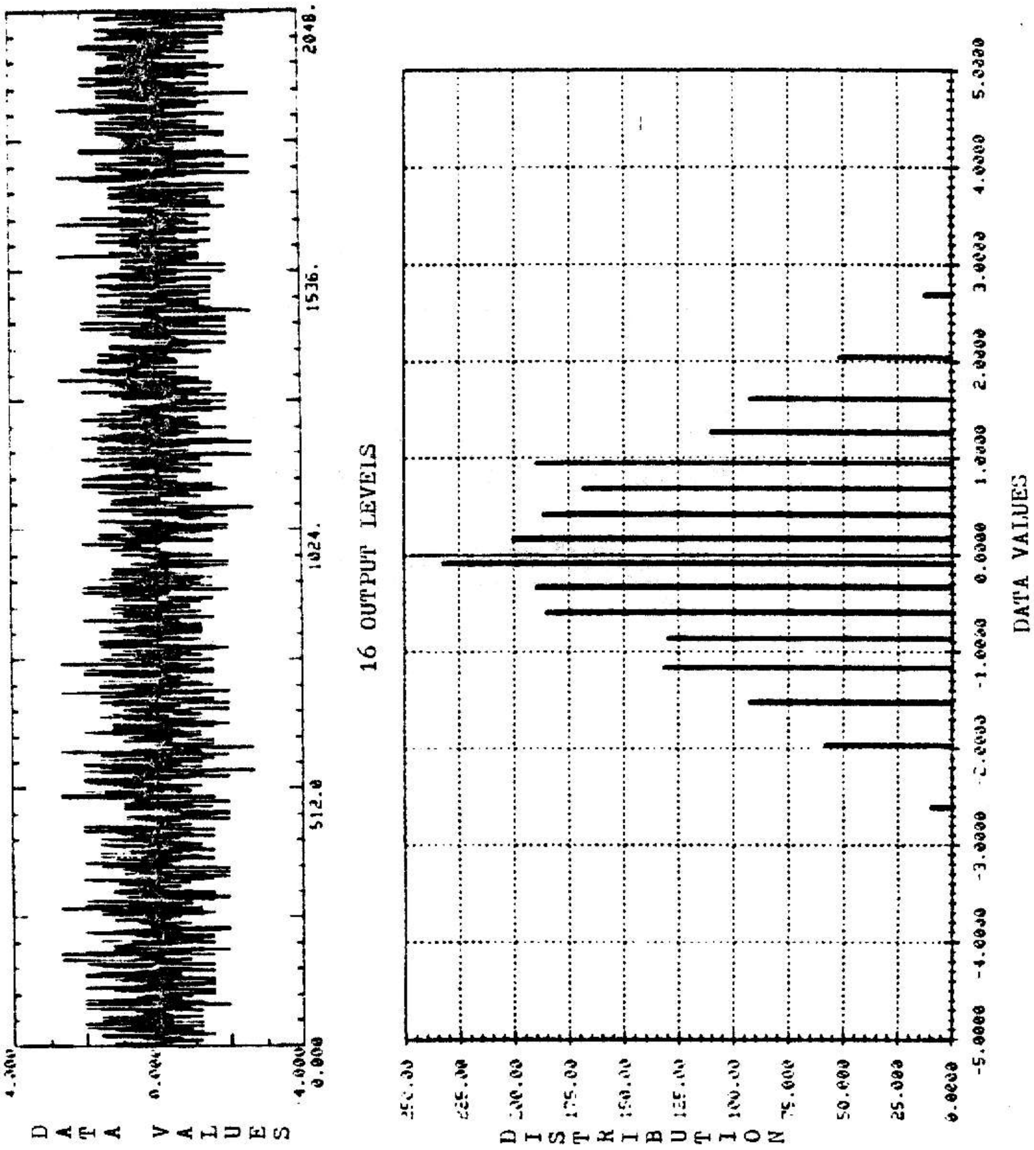
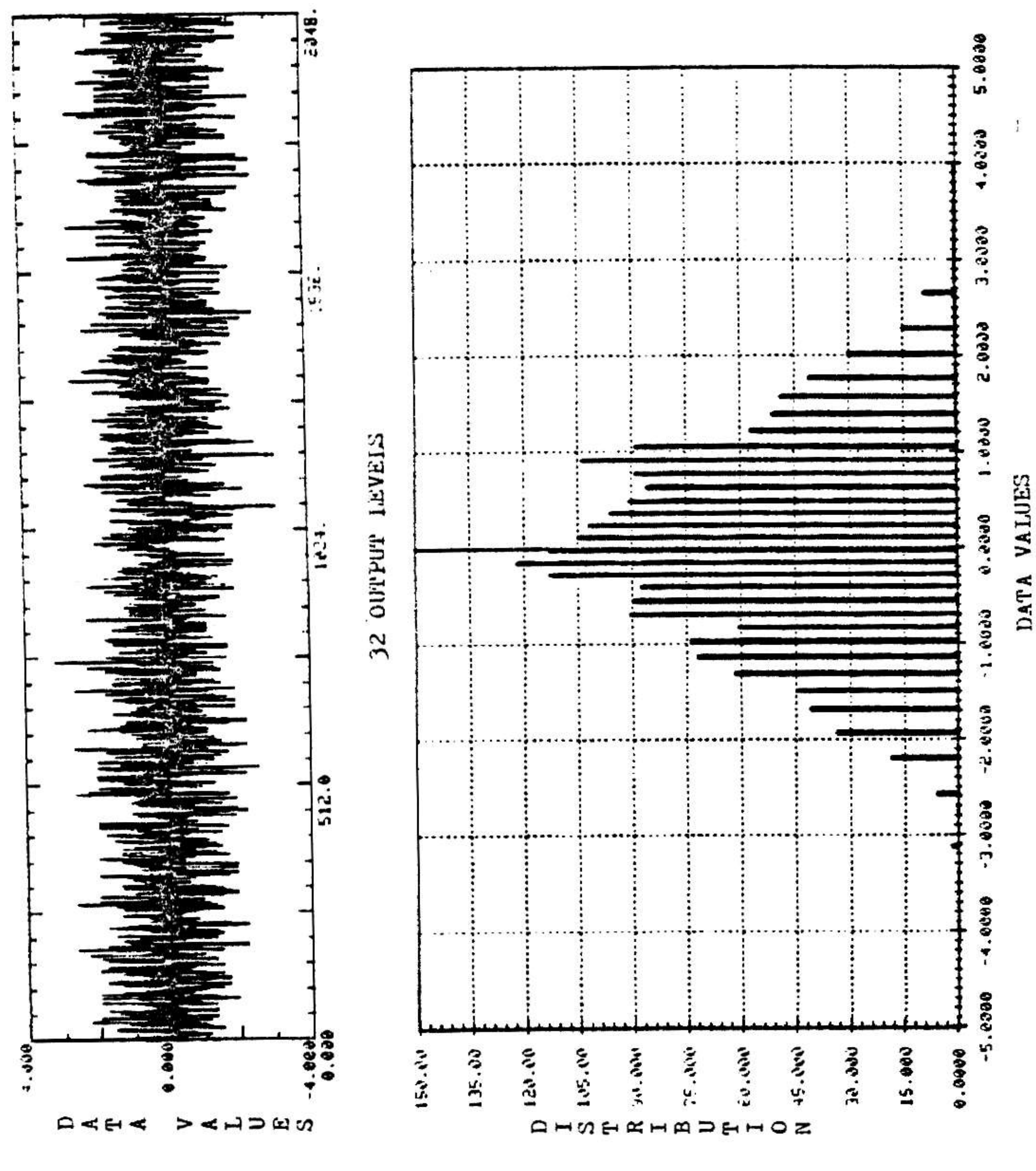



FIGURE 4.6



DATA VALUES
512.0
2048.
32 OUTPUT LEVELS
DISTRIBUTION
DATA VALUES
-5.0000
-4.0000
-3.0000
-2.0000
1.0000
2.0000
3.0000
5.0000
0.0000
15.000
135.00


FIGURE 4.7

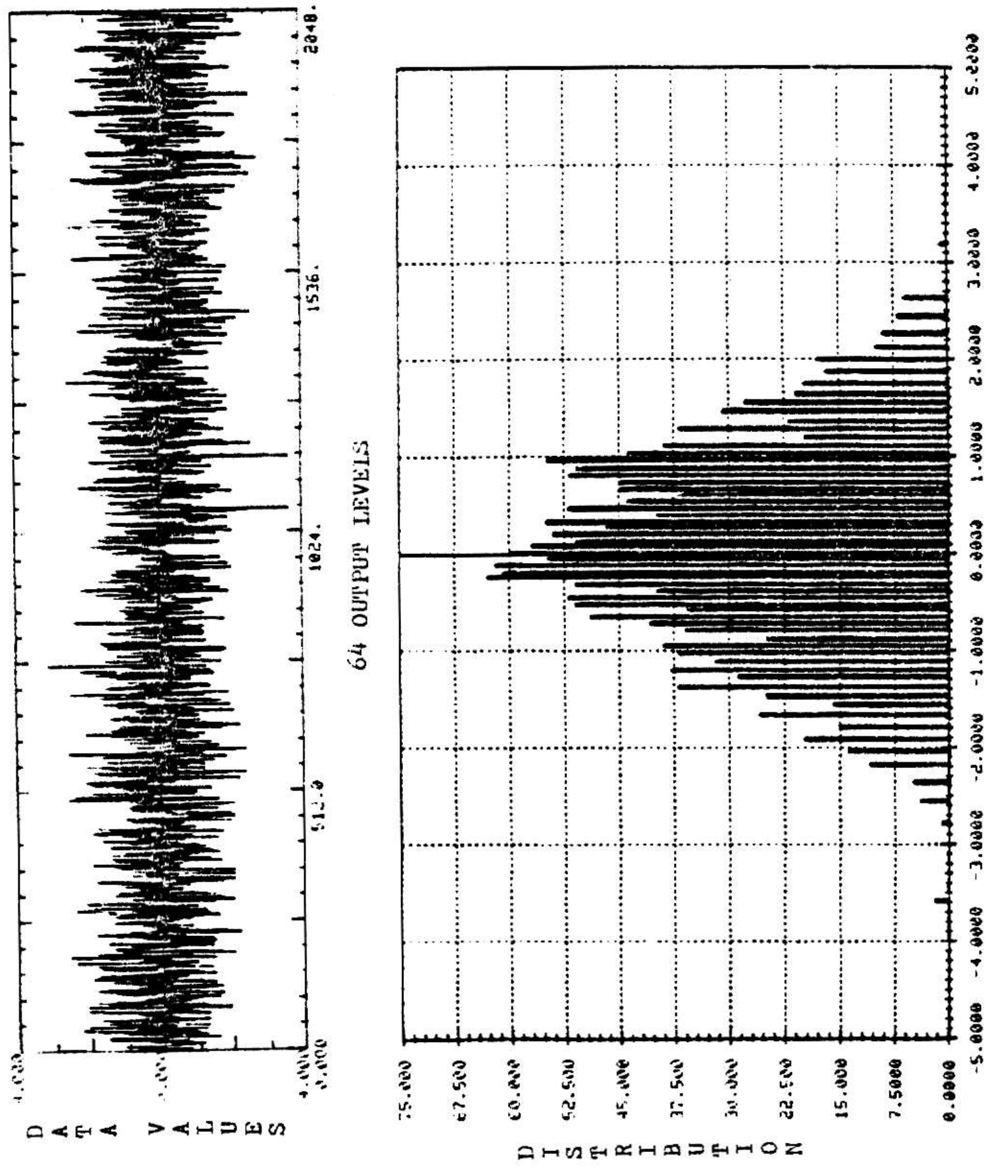



FIGURE 4.8

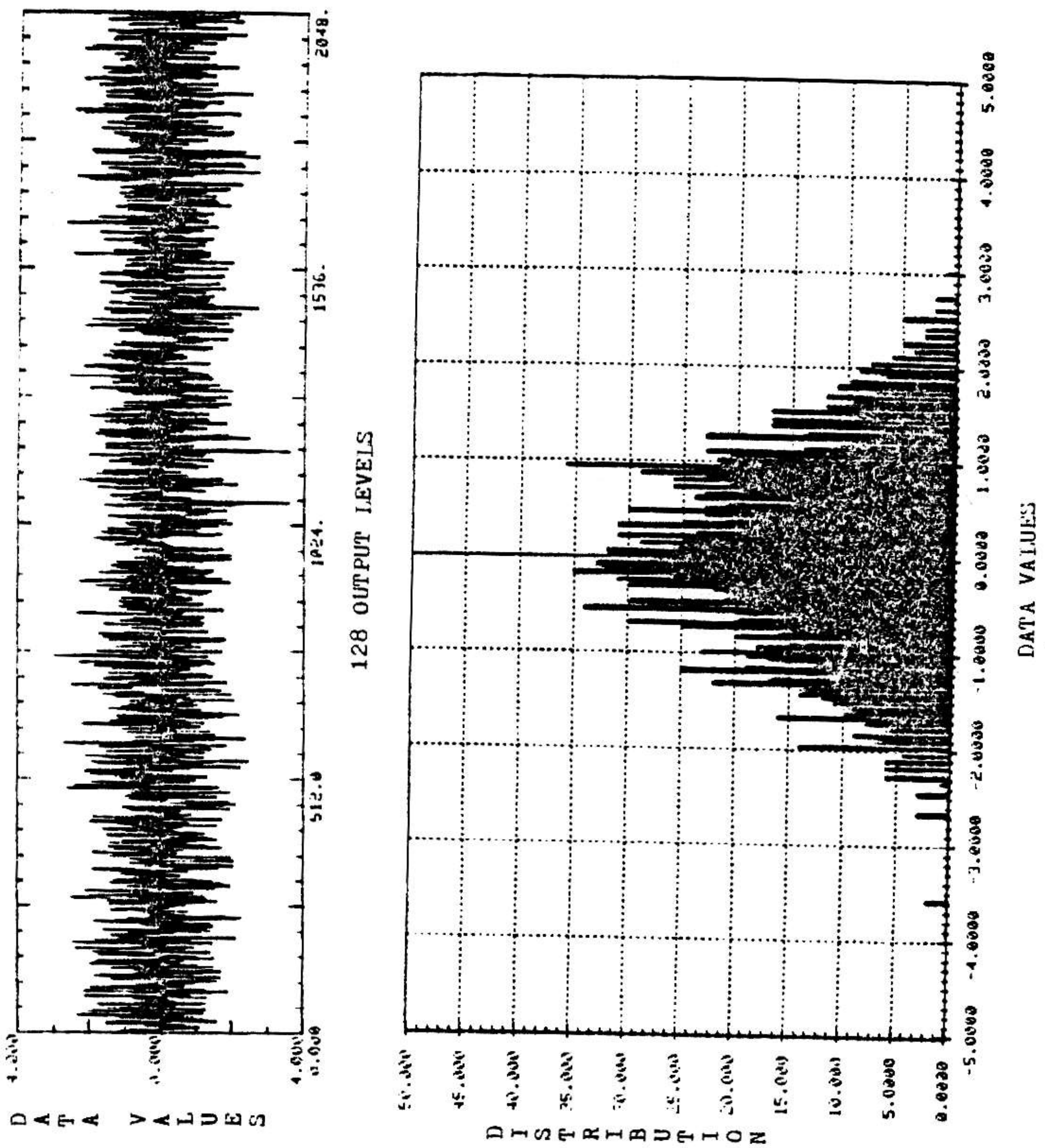



FIGURE 4.9

TABLE 4.1

EXPERIMENTAL VS. THEORETICAL
MEAN SQUARE ERROR
(EXAMPLE 1)

| NUMBER OF OUTPUT LEVELS | THEORETICAL MSE [1] | EXPERIMENTAL MSE | RESCALED EXPERIMENTAL MSE |
|---|---|---|---|
| 2 | 0.3634 | 0.382545 | 0.337679 |
| 4 | 0.1175 | 0.120184 | 0.106088 |
| 8 | 0.03454 | 0.034559 | 0.030506 |
| 16 | 0.009497 | 0.010177 | 0.008983 |
| 32 | 0.002499 | 0.002453 | 0.002165 |
| 64 | 0.000642 | 0.000666 | 0.000588 |
| 128 | 0.000162 | 0.000159 | 0.000140 |

## EXAMPLE 2

A random gaussian input sequence with mean and variance equal to 250.11 and 5.6372 respectively. Note that the input sequence is in floating point arithmetic, while the output sequences are in fixed point arithmetic.

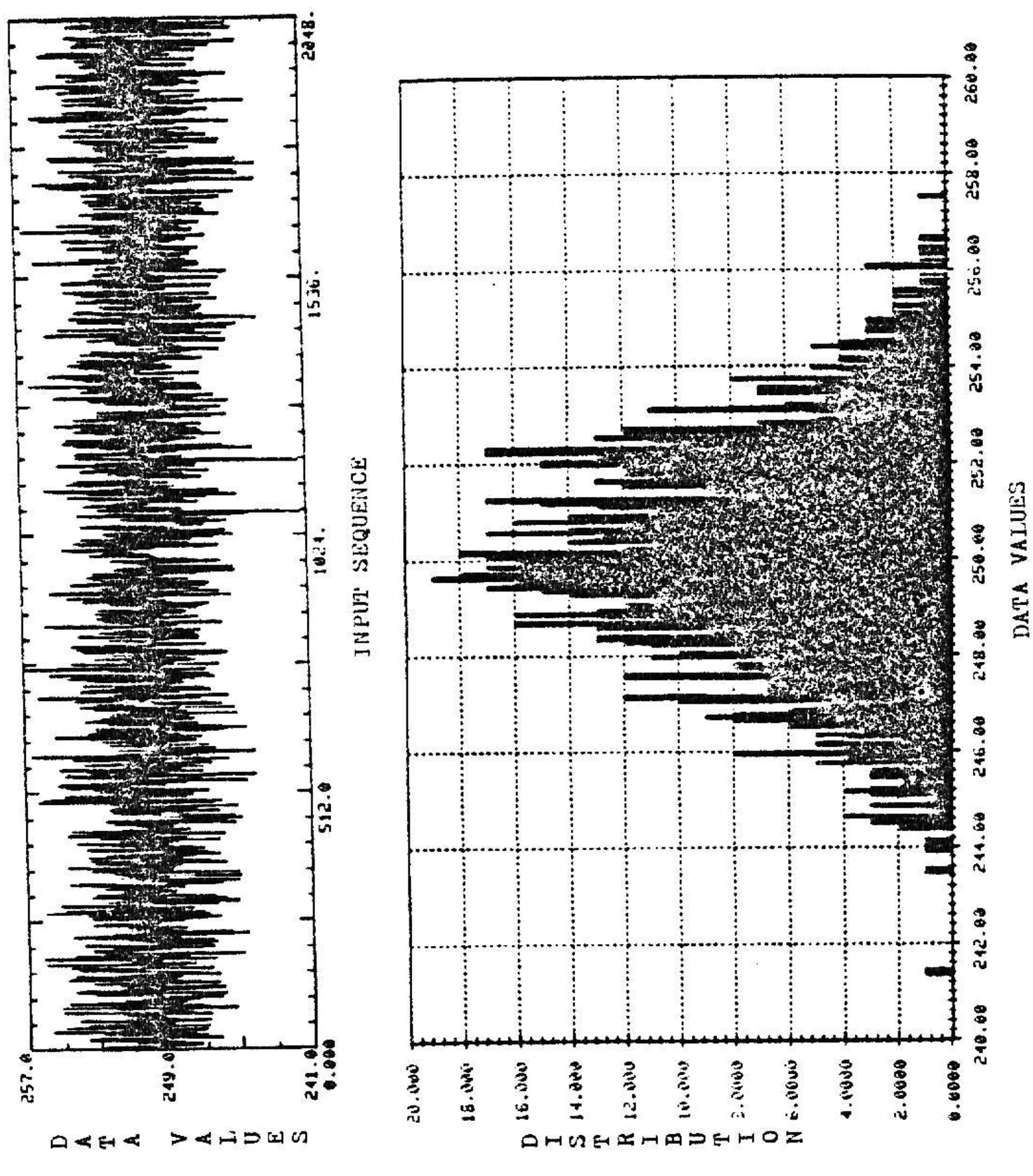



FIGURE 4.10

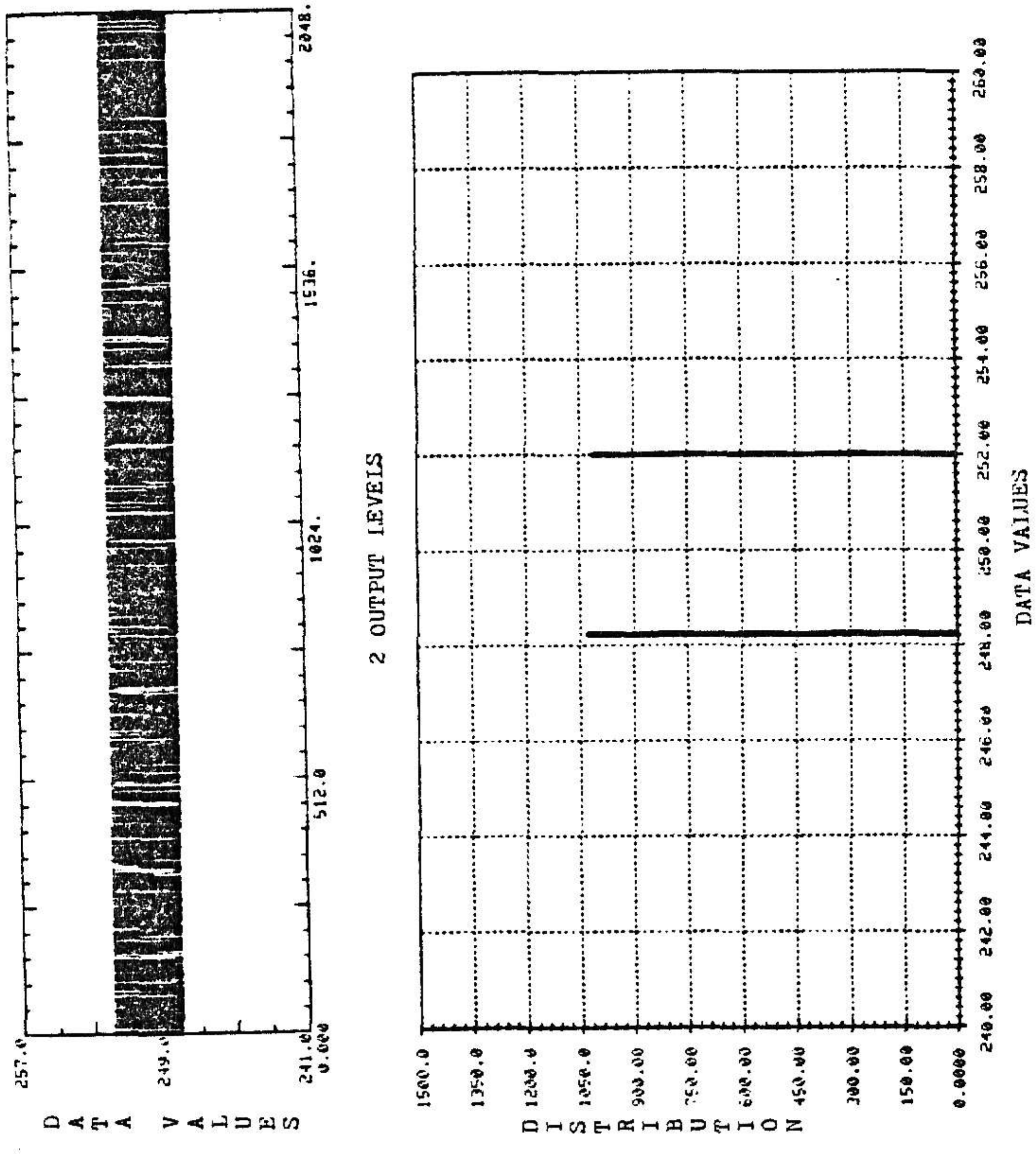
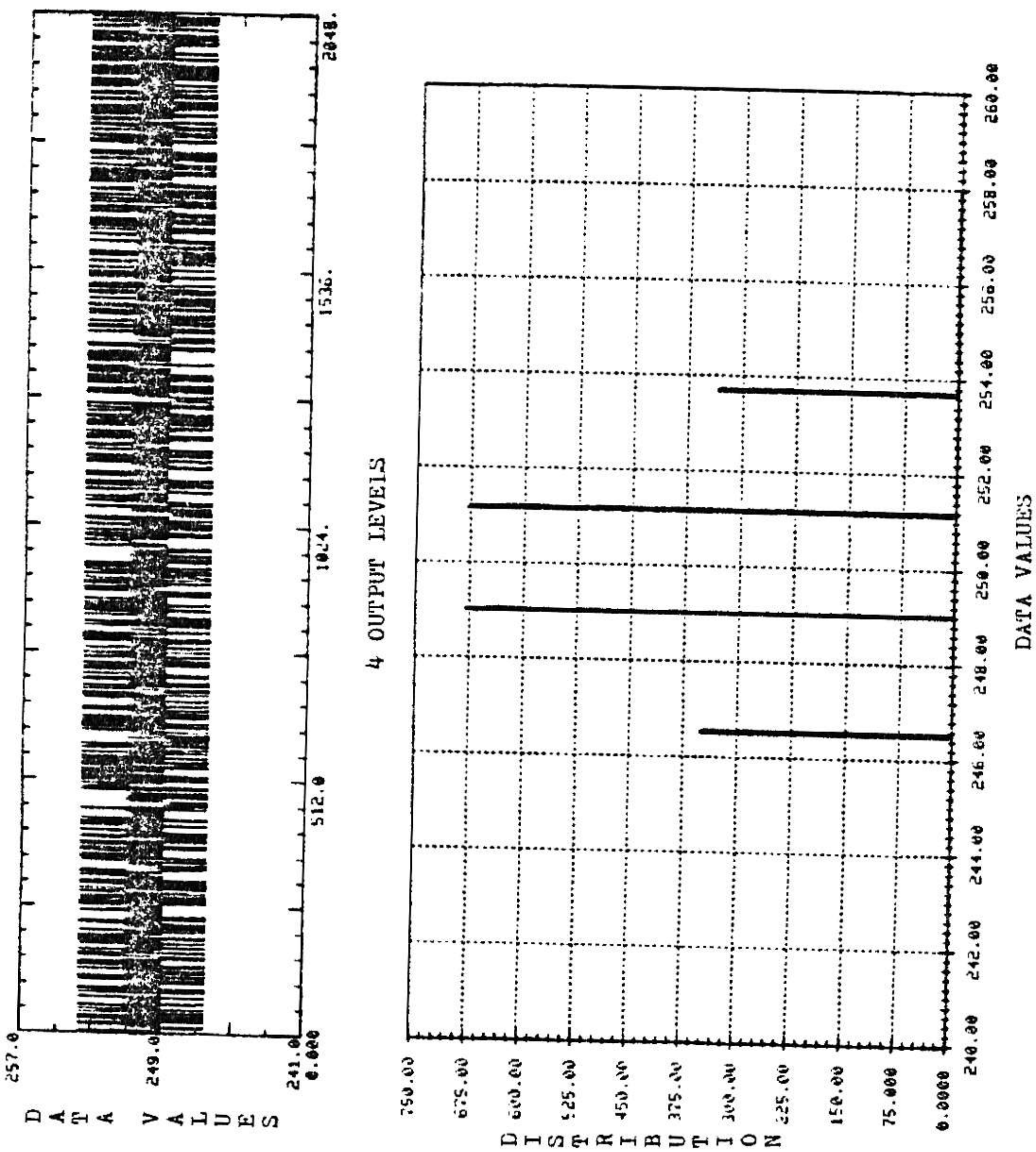

DATA VALUES
257.0
249.0
241.0
0.000
512.0
1024.
1536.
2048.
2 OUTPUT LEVELS
DISTRIBUTION
1500.0
1350.0
1200.0
1050.0
900.00
600.00
450.00
300.00
150.00
0.0000
240.00
242.00
244.00
246.00
248.00
250.00
252.00
254.00
256.00
258.00
260.00
DATA VALUES

FIGURE 4.11



DATA VALUES
257.0
249.0
241.0
0.000
512.0
1024.
1536.
2048.
4 OUTPUT LEVELS
DISTRIBUTION
DATA VALUES


FIGURE 4.12

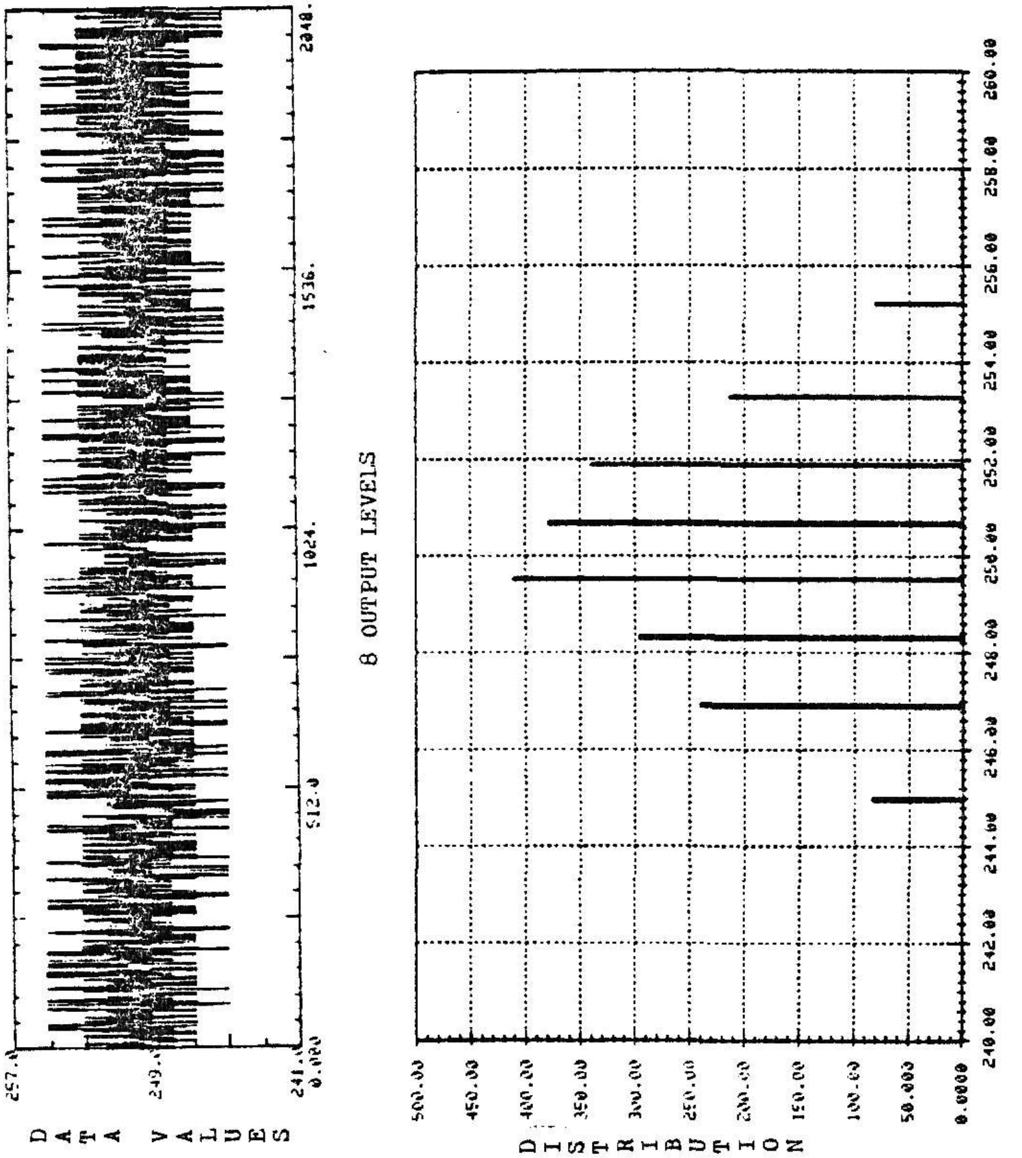



FIGURE 4.13

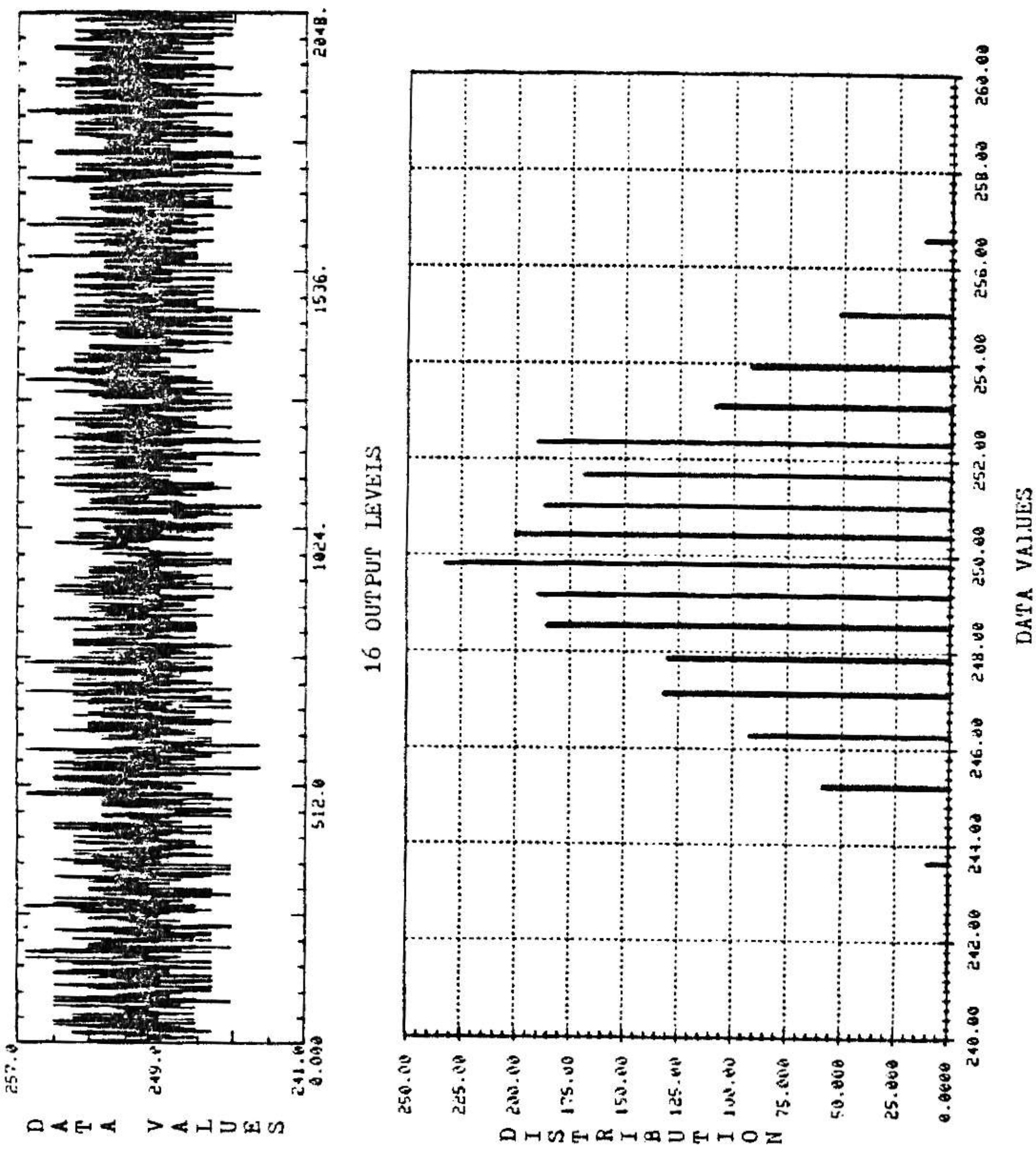



FIGURE 4.14

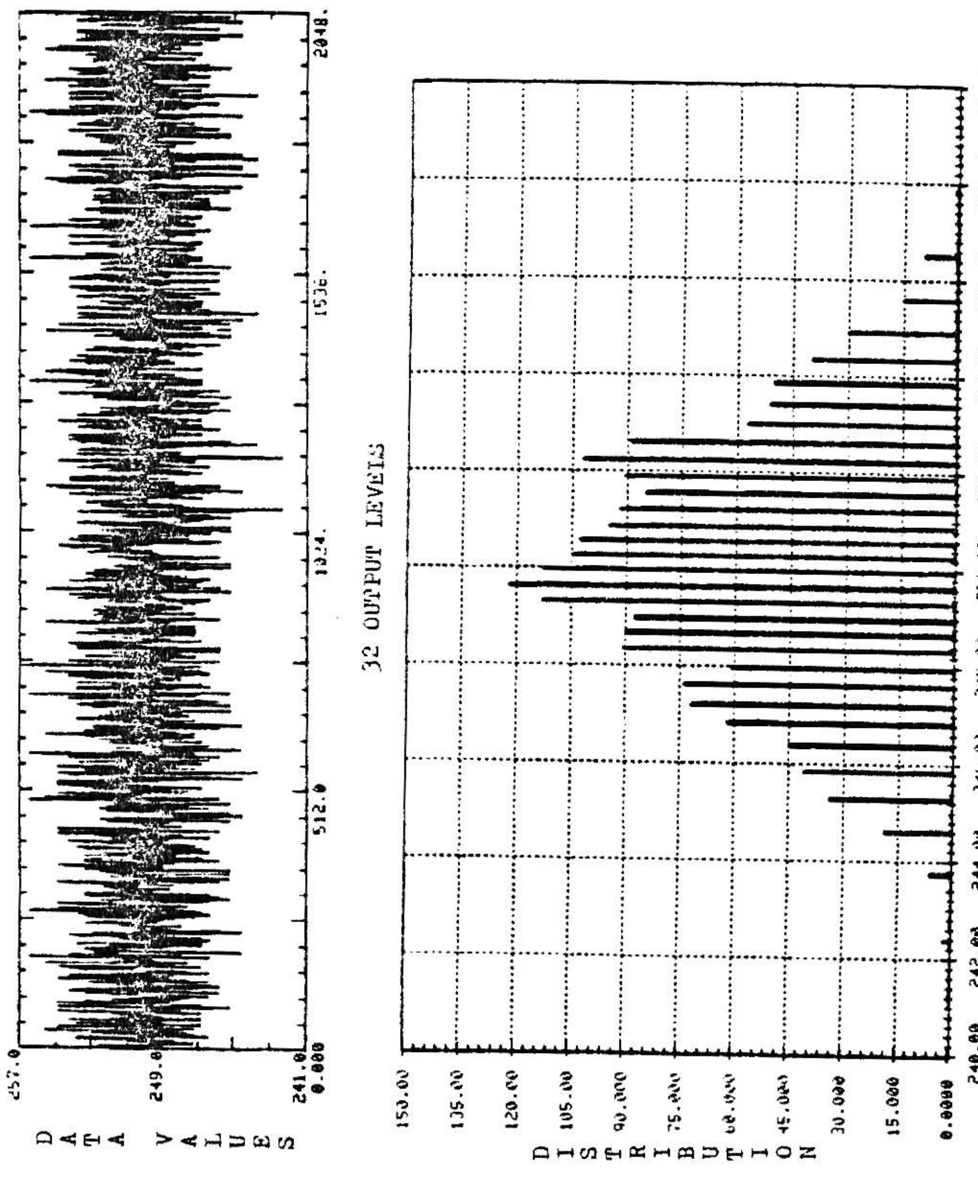



FIGURE 4.15

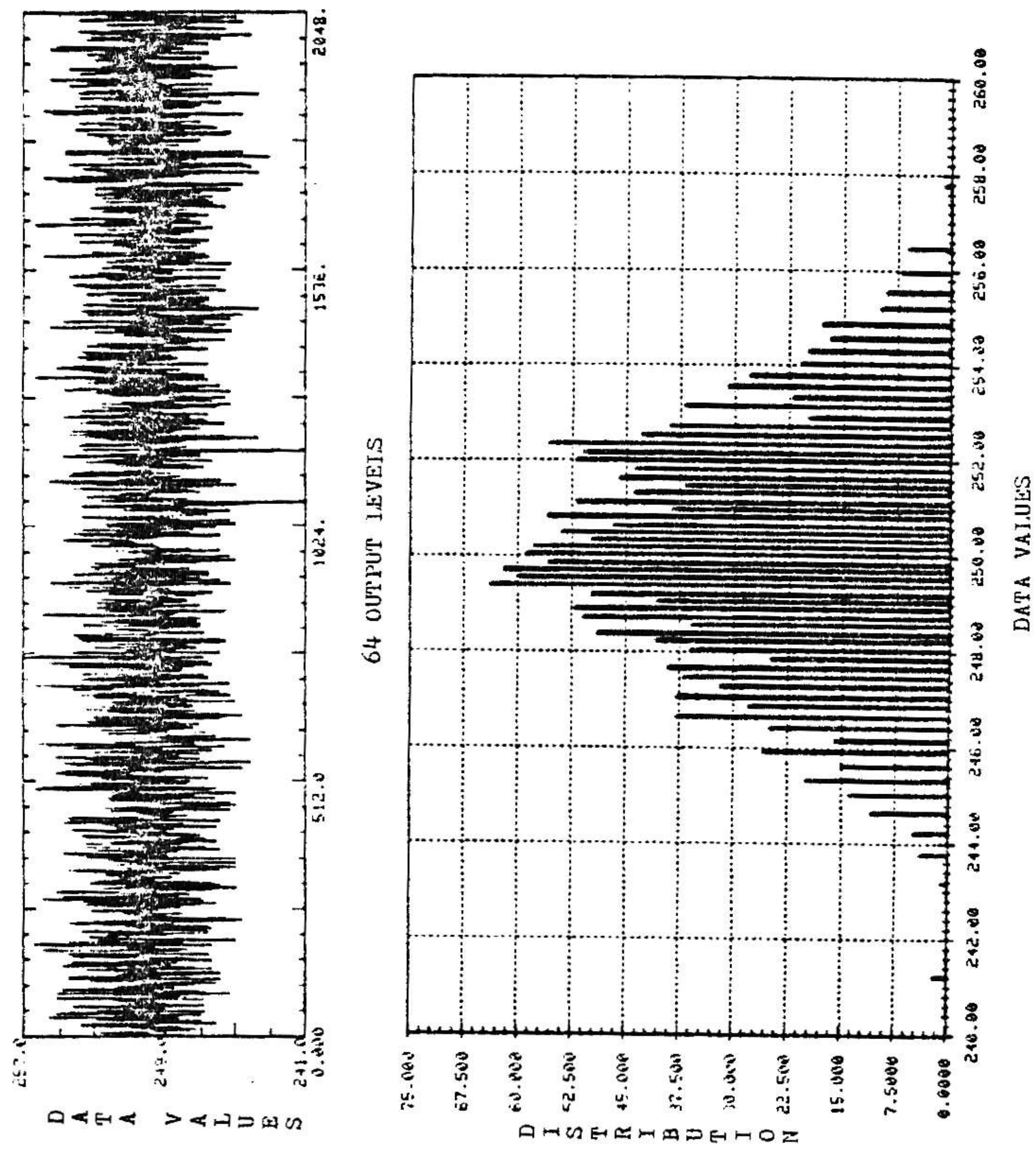



FIGURE 4.16

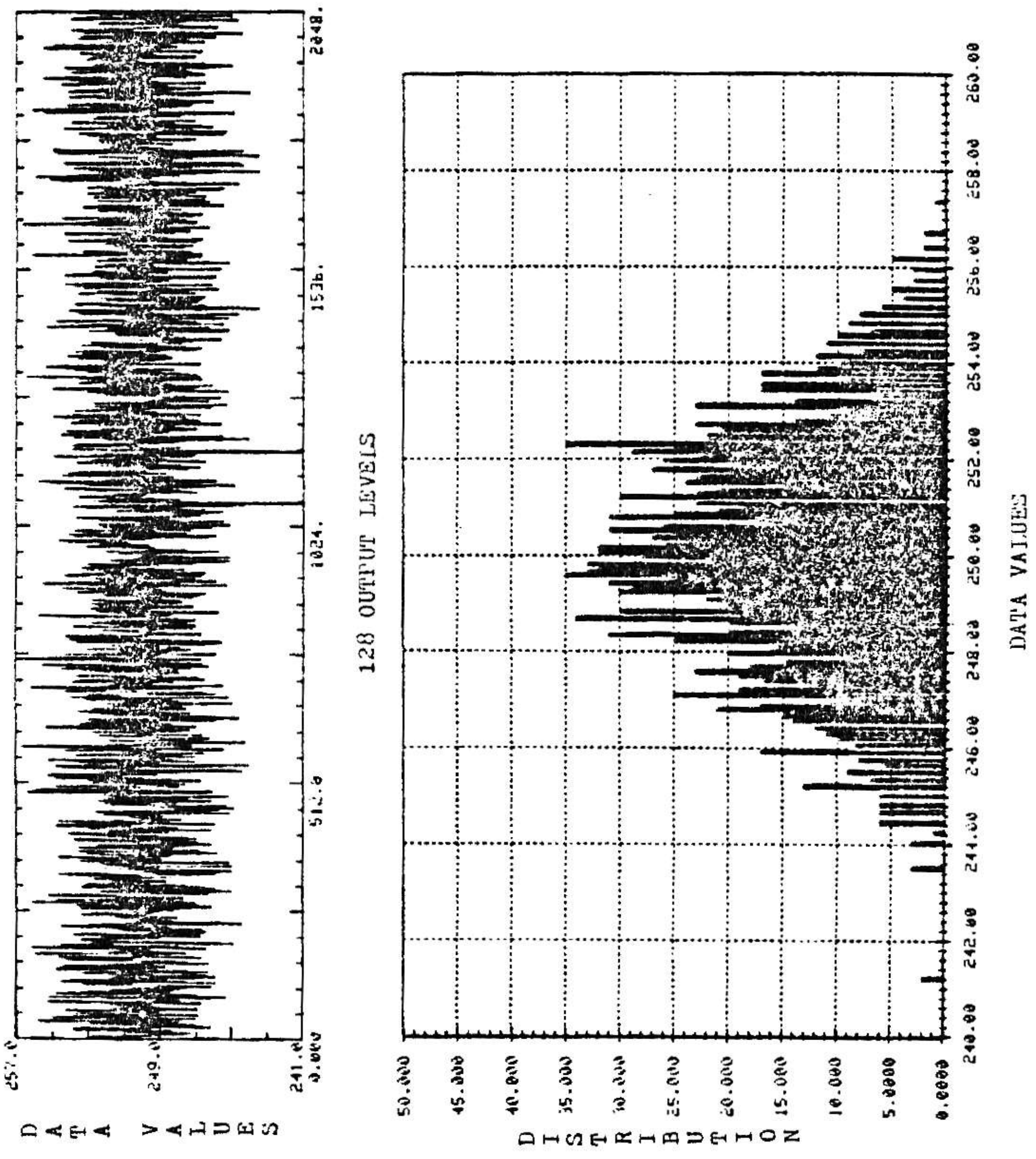



FIGURE 4.17

TABLE 4.2

EXPERIMENTAL VS. THEORETICAL
MEAN SQUARE ERROR
(EXAMPLE 2)

| NUMBER OF OUTPUT LEVELS | THEORETICAL MSE [1] | EXPERIMENTAL MSE | RESCALED EXPERIMENTAL MSE |
|---|---|---|---|
| 2 | 0.3634 | 0.063750 | 2.025848 |
| 4 | 0.1175 | 0.020033 | 0.636594 |
| 8 | 0.03454 | 0.005759 | 0.182994 |
| 16 | 0.009497 | 0.001697 | 0.053914 |
| 32 | 0.002499 | 0.000409 | 0.012989 |
| 64 | 0.000642 | 0.000111 | 0.003526 |
| 128 | 0.000162 | 0.000027 | 0.000843 |

# APPENDIX 4.1

This appendix contains the program to generate a random gaussian data sequence of specified mean and variance.

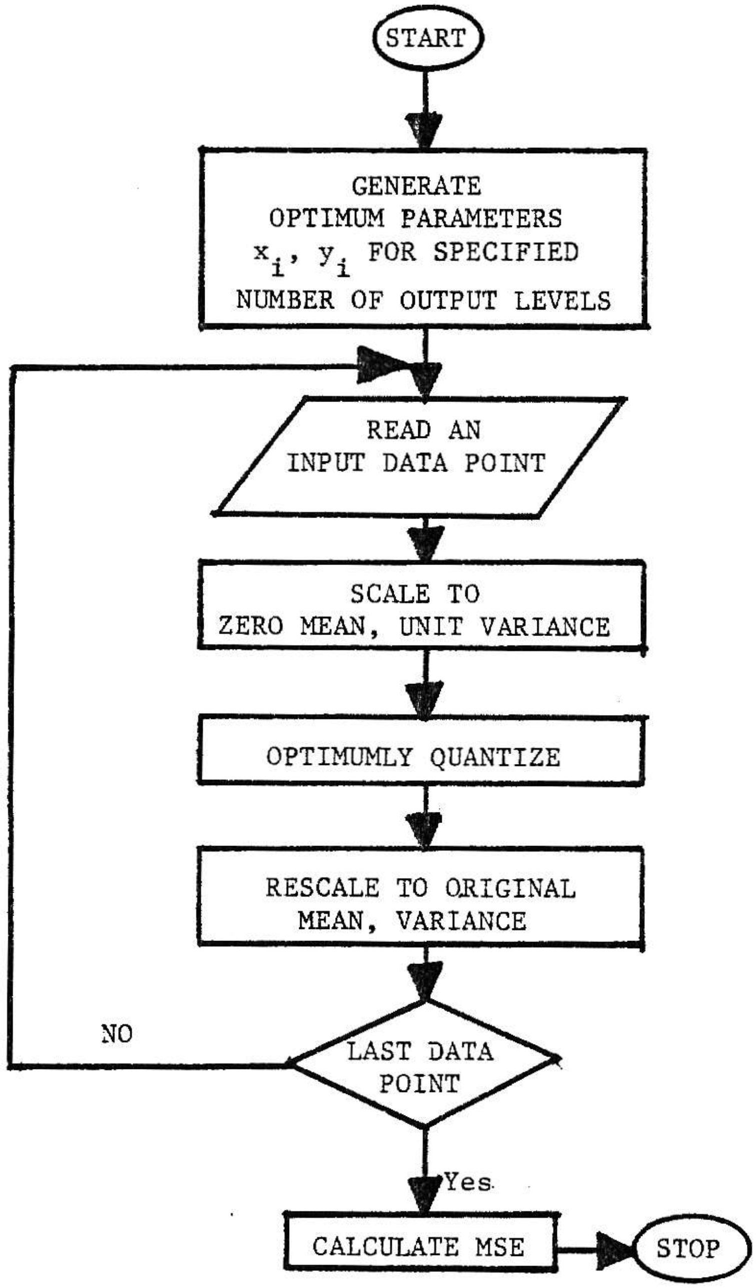
```
NORMGEN.FR              7/15/1982   0: 5:56   DIR DP0        Page  1

C       THIS PROGRAM GENERATES A RANDOM GAUSSIAN
C       DATA FILE OF SPECIFIED MEAN AND VARIANCE.
C
        REAL MEAN
        CALL OPENW(0,'OUTPUT FILENAME ?  ',4,SIZE)
        ACCEPT 'NUMBER OF DATA POINTS IN THE OUTPUT FILE ?  ', NPTS
        ACCEPT 'DESIRED MEAN AND VARIANCE OF THE OUTPUT FILE: MEAN,VAR ?  ', MEAN,VAR
C
        DO 100 I=1,NPTS
        X=GAUSS(VAR,MEAN)
        CALL WRITR(0,I,X,1,IERR)
        CALL CHECK(IERR)
 100    CONTINUE
        STOP
        END
```

# APPENDIX 4.2

This appendix contains the program to optimumly quantize a guassian input data sequence. This program also calculates the mean square error for the entire quantization process; i.e., the mean square error with respect to the input and output shown in Figure 4.1 for the entire sequence of data points.

FIGURE 4.18

THE OPTIMUM QUANTIZATION PROCESS



```
C       THIS SUBROUTINE QUANTIZES A NORMALLY DISTRIBUTED
C       INPUT DATA FILE INTO AN OUTPUT FILE
C       CONTAINING A SPECIFIED NUMBER OF ASSIGNED
C       OUTPUT VALUES (IE, LEVELS).
C
C
C       ASSIGNMENT OF VARIABLES USED IN THIS SUBROUTINE
C
C       NLEV...THE NUMBER OF OUTPUT LEVELS
C       .......NOTE: 2**B=NLEV  WHERE B IS THE NUMBER OF
C                    OUTPUT BITS
C
C       MEAN...THE MEAN OF THE INPUT DATA FILE
C
C       VAR...THE VARIANCE OF THE INPUT DATA FILE
C
C       TABLEG...QUANTIZATION TABLE (JOEL MAX, M.I.T. 1961)
C       .........NOTE: TABLEG(ROW,COLUMN) WHERE COLUMN 1 ARE
C                      THE CUT POINTS OR XI'S, AND COLUMN 2
C                      ARE THE REP-VALUES OR YI'S
C
C       X(I)...THE INPUT DATA VALUES
C
C       Y(I)...THE QUANTIZED OUTPUT VALUES
C
C
C
        COMPILER FREE
        REAL MEAN, MSE
        DIMENSION TABLEG(128,2)
        DIMENSION X(2048), Y(2048)
        CALL OPENR(2,'INPUT FILENAME ?  ',0,SIZE)
        CALL OPENW(3,'OUTPUT FILENAME ?  ',0,SIZE)
 2      FORMAT(' NUMBER OF OUTPUT LEVELS ?  ',Z)
 3      FORMAT(' VARIANCE OF THE INPUT FILE ?  ',Z)
 4      FORMAT(' MEAN OF THE INPUT FILE ?  ',Z)
C
        ITTO=10
        ITTI=11
        ITTTY=12
        WRITE(ITTO,2)
        READ(ITTI) NLEV
        WRITE(ITTO,3)
        READ(ITTI) VAR
        WRITE(ITTO,4)
        READ(ITTI) MEAN
        MSE=0.0
C
C       GENERATE TABLE
C
        CALL TABLE(NLEV,TABLEG)
C
C       READ INPUT FILE INTO TABLE
C
        DO 100 I=1,2048
        READ BINARY(2,END=990) X(I)
        NPTS=I
        CALL LOOKUP(NLEV,MEAN,VAR,TABLEG,I,X,Y)
 100    CONTINUE
        TYPE 'INPUT FILE LENGTH MAY EXCEED 2048 POINTS'
 990    CONTINUE
C
C       CALCULATE MEAN SQUARED ERROR
C
        DO 995 I=1,NPTS
        MSE=((Y(I)-X(I))**2)/NPTS+MSE
        WRITE BINARY(3) Y(I)
 995    CONTINUE
        WRITE(ITTO,1) MSE
 1      FORMAT(' MEAN SQUARED ERROR =  ',F16.6)
C
        CALL CLOSE(2,IERR)
        CALL CLOSE(3,IERR)
        STOP
        END
```

```
C       THIS SUBROUTINE COMPUTES THE CUT-POINTS AND REP-VALUES FOR
C       A GUASSIAN PROBABILITY DENSITY AND MSE DISTORTION MEASURE.
C
        SUBROUTINE TABLE(NLEV,TABLEG)
        COMPILER FREE
        DOUBLE PRECISION X,Y,MSE,ENTROPY,AREA1,AREA2
        DOUBLE PRECISION X0,X1,X2,P0,P2,XN,YN
        DOUBLE PRECISION P1,Y0,GAP,Z0,Z1,W0,W1,AREA,TAIL,A0,A1,W2,DABS,Q1,Q0
        DIMENSION TABLEG(128,2)
        ITT0=10
        ITT1=11
        ITTTY=12
C
C       GET THE NUMBER OF LEVELS
C
        N=NLEV/2
C
C       ESTIMATE THE VALUE OF THE FIRST CUT-POINT OR REP-VALUE
C
        P1=2.1613D0/DBLE(FLOAT(NLEV))-1.653D0/DBLE(FLOAT(NLEV)**2.0)
C
C       PERFORM TEN ITERATIONS
C
        DO 1073 II=1,10
C
C       INITIALIZE THE FIRST CUT-POINT AND REP-VALUE
C
        IF(N*2.EQ.NLEV) GO TO 1100
C
C       ODD NUMBER OF LEVELS
C
        X0=P1
        Y0=2.0D0*P1
        GO TO 1110
 1100   CONTINUE
C
C       EVEN NUMBER OF LEVELS
C
        X0=0.0D0
        Y0=P1
 1110   CONTINUE
        I=1
C
C       THE FIRST CUT-POINT VALUE
C
        TABLEG(I,1)=X0
        TABLEG(I,2)=Y0
        IF(NLEV.LT.4)  GO TO 1000
C
C       DETERMINE THE REST OF THE CUT-POINTS AND REP-VALUES
C
        DO 1000 I=2,N
        CALL CUTPT(X0,Y0,X1)
C
        X0=X1
        Y0=2.0D0*X0-Y0
        TABLEG(I,1)=X0
```

```
TABLGENS.FR             7/15/1982   0: 1:47   DIR DP0         Page  2

        TABLEG(I,2)=Y0
 1000   CONTINUE
C
C       USING AN ALTERNATE METHOD,
C       DETERMINE THE LAST REP-VALUE, Y(N), BY COMPUTING
C       THE CENTROID OF THE AREA FROM X(N) TO INFINITY
C
        XN=X0
        CALL ERF(XN,AREA)
        YN=DEXP(-XN**2/2.0D0)/DSQRT(2.0D0*3.14159D0)/(0.5D0-AREA)
C
C       COMPUTE THE DIFFERENCE BETWEEN THE TWO VALUES AND UPDATE
C       THE STARTING VALUE, P1
C
        Q1=Y0-YN
        IF(DABS(Q1/YN).LT.1.0D-6) RETURN
        IF(II.EQ.1) P2=0.99D0*P1
        IF(II.NE.1) P2=(P1*Q0-P0*Q1)/(Q0-Q1)
        P0=P1
        P1=P2
        Q0=Q1
 1073   CONTINUE
        STOP
        END
```

```
CUTPT2.FR              7/15/1982   0: 1: 8   DIR DP0          Page   1

C       THIS SUBROUTINE COMPUTES THE NEXT CUT-POINT, X1, GIVEN THE
C       PREVIOUS CUT-POINT AND REP-VALUE
C
        SUBROUTINE CUTPT(X0,Y0,X1)
        DOUBLE PRECISION X0,Y0,X1,EPS,AREA,AREA0,AREA1,DELX1,PROB,FUNC,DABS
        DOUBLE PRECISION TAIL
        EPS=1.0D-9
C
C       INTEGRATE THE FUNCTION FROM X0 TO Y0
C
        CALL FPX(X0,Y0,AREA0)
C
        X1=2.0D0*Y0-X0
C
C       ITERATE UP TO 100 TIMES TO DETERMINE X1
C
        DO 1000 I=1,100
        CALL FPX(X1,Y0,AREA1)
        AREA=AREA1-AREA0
        DELX1=-(AREA)/(PROB(X1)*FUNC(X1,Y0))
        IF(DABS(DELX1/X1).LT.EPS) RETURN
        X1=X1+DELX1
 1000   CONTINUE
        RETURN
        END
```

```
ERF2.FR                7/15/1982   0: 1: 6   DIR DP0          Page   1

C       THIS SUBROUTINE EVALUATES THE INTEGRAL OF THE
C       GAUSSIAN PROBABILITY DISTRIBUTION VIA AN
C       INFINITE SERIES REPRESENTATION
C
C
        SUBROUTINE ERF(X,AREA)
        DOUBLE PRECISION X,W,AREA,AREA1,DEXP,DSQRT,DBLE,DABS
        W=1.0D0
        AREA1=X*DEXP(-X**2/2.0D0)/DSQRT(2.0D0*3.141592D0)
        DO 100 J=1,100
        K=2*J+1
        W=W*DBLE(FLOAT(K))
        AREA=(X/DSQRT(2.0D0))**K*2.0D0**J/W*DEXP(-X**2/2.0D0)/DSQRT(3.141592D0)+AREA
        IF (DABS(AREA1-AREA).LT.1.0D-9) RETURN
        AREA1=AREA
 100    CONTINUE
        TYPE 'NEED MORE TERMS OF THE INFINITE SERIES'
        RETURN
        END
```

```
FPX.FR                 7/15/1982   0: 1:11   DIR DP0          Page   1

        SUBROUTINE FPX(X,Y,AREA)
        DOUBLE PRECISION X,Y,AREA,DEXP,DSQRT,A
        CALL ERF(X,A)
        AREA=(1-DEXP(-X**2/2.0D0))/DSQRT(2.0D0*3.141592D0)-Y*A
        RETURN
        END
```

FUNC.FR 7/15/1982 0: 1:10 DIR DP0 Page 1

```
      DOUBLE PRECISION FUNCTION FUNC(X,YO)
      DOUBLE PRECISION X,YO
      FUNC=X-YO
      RETURN
      END
```

PROB.FR 7/15/1982 0: 1: 9 DIR DP0 Page 1

```
      DOUBLE PRECISION FUNCTION PROB(X)
      DOUBLE PRECISION X,DEXP,DSQRT
      PROB=DEXP(-X**2/2.0D0)/DSQRT(2.0D0*3.14159D0)
      RETURN
      END
```

```
      SUBROUTINE LOOKUP(NLEV,MEAN,VAR,TABLEG,I,X,Y)
      COMPILER FREE
      REAL MEAN
      DIMENSION TABLEG(129,2)
      DIMENSION X(2048), Y(2048)
C
      ITTO=10
      ITTI=11
      ITTTY=12
C
      MAXJ=NLEV/2
      LAST=MAXJ-1
C     SCALE INPUT DATA TO ZERO MEAN AND UNIT VARIANCE
C
      XSCALE=(X(I)-MEAN)/SQRT(VAR)
C
C     LOOKUP TABLE VALUES ARE POSITIVE
C
      XSCAL=ABS(XSCALE)
C
C     ODD NUMBER OF LEVELS
C
      IF(XSCAL.GE.TABLEG(1,1)) GO TO 50
      Y(I)=0.0
      RETURN
C
C     EVEN OR ODD NUMBER OF LEVELS
C
 50   DO 100 J=1,LAST
      IF(XSCAL.GE.TABLEG(J+1,1)) GO TO 100
      J2=J
      GO TO 200
 100  CONTINUE
C
      J2=MAXJ
C
 200  YTABLE=TABLEG(J2,2)*XSCALE/XSCAL
C
C     NOTE  XSCALE/XSCAL EQUALS +1 OR -1
C
C     RESCALE TO ORIGINAL MEAN AND VARIANCE
C
      Y(I)=(YTABLE*SQRT(VAR))+MEAN
C
      RETURN
      END
```

# CHAPTER V

## PROGRAM USAGE

Two programs are used. The first, TABLGEN5, generates the optimum gaussian quantizer parameters, $X_i$ and $Y_i$, in tabular form. The second, QUANTIZE1, takes a random gaussian input sequence, scales it to a normalized gaussian sequence and then passes the entire sequence through the optimum quantizer for a specified number of output levels. The quantizer output is then rescaled to the original mean and variance of the input sequence.

The first program, TABLGEN5, generates the optimum parameters. There are three inputs for which the program will prompt the user. The first two inputs are the names of output files. One of these stores the value of the mean square error for the specified number of output levels. These values can be retrieved at a later time and plotted as in Figure 3.12. Similiarly the second output filename is used to store the value of entropy being calculated. The last input that is entered is the number of output levels desired for the optimum quantizer. All values are entered from the CRT terminal. A typical input sequence is illustrated in Figure 5.1. The output consists of the values $x_i$ and $y_i$, ending with the values for the mean square error and entropy for the specific number of output levels being calculated, as exemplified in Figure 5.2. The output is printed on the line printer.

The second program, QUANTIZE1, requires the prior use of a library program, XBAR, to determine the mean and variance of the gaussian input sequence. QUANTIZE1 requires five inputs to the CRT terminal, all of which are prompted for: the <u>output sequence filename</u>, the <u>input filename</u> of the random gaussian sequence, the <u>mean</u> and <u>variance</u> of the input sequence obtained from XBAR, and finally the desired <u>number of output</u>

FIGURE 5.1

TABLGEN5
Input Sequence

```
TABLGEN5

MEAN SQUARE ERROR OUTPUT FILENAME ?  MSE.FP

ENTROPY OUTPUT FILENAME ? ENTROPY.FP

NUMBER OF OUTPUT LEVELS ? 4
NUMBER OF OUTPUT LEVELS ? 5
INT
R
```

# FIGURE 5.2

## TABLGEN5
## Output Sequence

```
 4  OUTPUT LEVELS

                                          N=  4
ITERATION NUMBER  1     J         X(J)              Y(J)
                        1       .000000           .437013
                        2       .941286          1.445560
     SECANT METHOD......        Y(N) =           1.478315

                                          N=  4
ITERATION NUMBER  2     J         X(J)              Y(J)
                        1       .000000           .432642
                        2       .930265          1.427887
     SECANT METHOD......        Y(N) =           1.469573

                                          N=  4
ITERATION NUMBER  3     J         X(J)              Y(J)
                        1       .000000           .453042
                        2       .982275          1.511508
     SECANT METHOD......        Y(N) =           1.510958

                                          N=  4
ITERATION NUMBER  4     J         X(J)              Y(J)
                        1       .000000           .452776
                        2       .981588          1.510400
     SECANT METHOD......        Y(N) =           1.510409

                                          N=  4
ITERATION NUMBER  5     J         X(J)              Y(J)
                        1       .000000           .452730
                        2       .981600          1.510419
     SECANT METHOD......        Y(N) =           1.510419


MEAN SQUARED ERROR FOR   4 OUTPUT LEVELS =   .117481


ENTROPY FOR                 4 OUTPUT LEVELS =  1.911098
```

# FIGURE 5.2 (CONT.)

## TABLGEN5
## Output Sequence

```
  5  OUTPUT LEVELS

                                          N=  5
ITERATION NUMBER  1     J        X(J)                Y(J)
                        1      .366140              .732280
                        2     1.181830             1.631381
     SECANT METHOD......        Y(N) =             1.672620

                                          N=  5
ITERATION NUMBER  2     J        X(J)                Y(J)
                        1      .362479              .724957
                        2     1.157906             1.610854
     SECANT METHOD......        Y(N) =             1.661201

                                          N=  5
ITERATION NUMBER  3     J        X(J)                Y(J)
                        1      .382718              .765436
                        2     1.246066             1.726697
     SECANT METHOD......        Y(N) =             1.725562

                                          N=  5
ITERATION NUMBER  4     J        X(J)                Y(J)
                        1      .382272              .764544
                        2     1.244311             1.724079
     SECANT METHOD......        Y(N) =             1.724110

                                          N=  5
ITERATION NUMBER  5     J        X(J)                Y(J)
                        1      .382284              .764568
                        2     1.244358             1.724149
     SECANT METHOD......        Y(N) =             1.724149


MEAN SQUARED ERROR FOR   5 OUTPUT LEVELS =   .079941


ENTROPY FOR                5 OUTPUT LEVELS =  2.202916
```

levels from the optimum quantizer. The output file is written to permanent storage as previously mentioned. The only other output is the value of the mean square error, as mentioned in Appendix 4.2, to the CRT terminal. A typical input/output sequence is illustrated in Figure 5.3.

FIGURE 5.3

QUANTIZE1
Input/Output Sequence

```
QUANTIZE1

INPUT FILENAME ?  G2048.01

OUTPUT FILENAME ?  G64.01
 NUMBER OF OUTPUT LEVELS ?  64
 VARIANCE OF THE INPUT FILE ?   93953
 MEAN OF THE INPUT FILE ?  .044654
INPUT FILE LENGTH MAY EXCEED 2048 POINTS
 MEAN SQUARED ERROR =            .000530

STOP
R
```

# CHAPTER VI

## CONCLUSIONS AND RECOMMENDATIONS

The programs developed in this report illustrate the accuracy problems encountered when using fixed-point arithmetic as exemplified in Figures 6.1 and 6.2 (Note only the input sequence is in floating point arithmetic). However, since special purpose fixed-point hardware in the field can be implemented faster and more economically, these optimal programs are of value.

The computer programs presented in this report can be utilized as a subroutine in applications such as speech and image data compression. In both instances the quantization occurs after the raw data has been transformed (FFT, DCT, etc.). Figure 6.3 [7] depicts the image processing scheme incorporating the optimum quantization process of Figure 4.1. It should be noted that not all pixels are quantized with the same number of output levels; for instance zonal coding, Figure 6.4 may be the scheme employed for compression [7]. Thus a subroutine that can quantize, encode, various output levels is of great value in such an application. Finally it should be pointed out that in processing the smaller MxM pixel blocks of Figure 6.3, all the elements are assumed to have a gaussian distribution, $p_X(x)$, except the very first element (1,1) which is the uniformly distributed dc term. Therefore our gaussian quantizer will apply to all but one term. For this singular term, we could go back and calculate optimum parameters for the uniform distribution, $p_X(x)$. However, the internal hardware of the computer itself assigns numbers based on the uniform distribution. Thus, with our gaussian parameters and the inherent uniform distribution of the computer, we have all the ingredients to implement the optimum quantizer for the image data compression scheme of Figure 6.3.

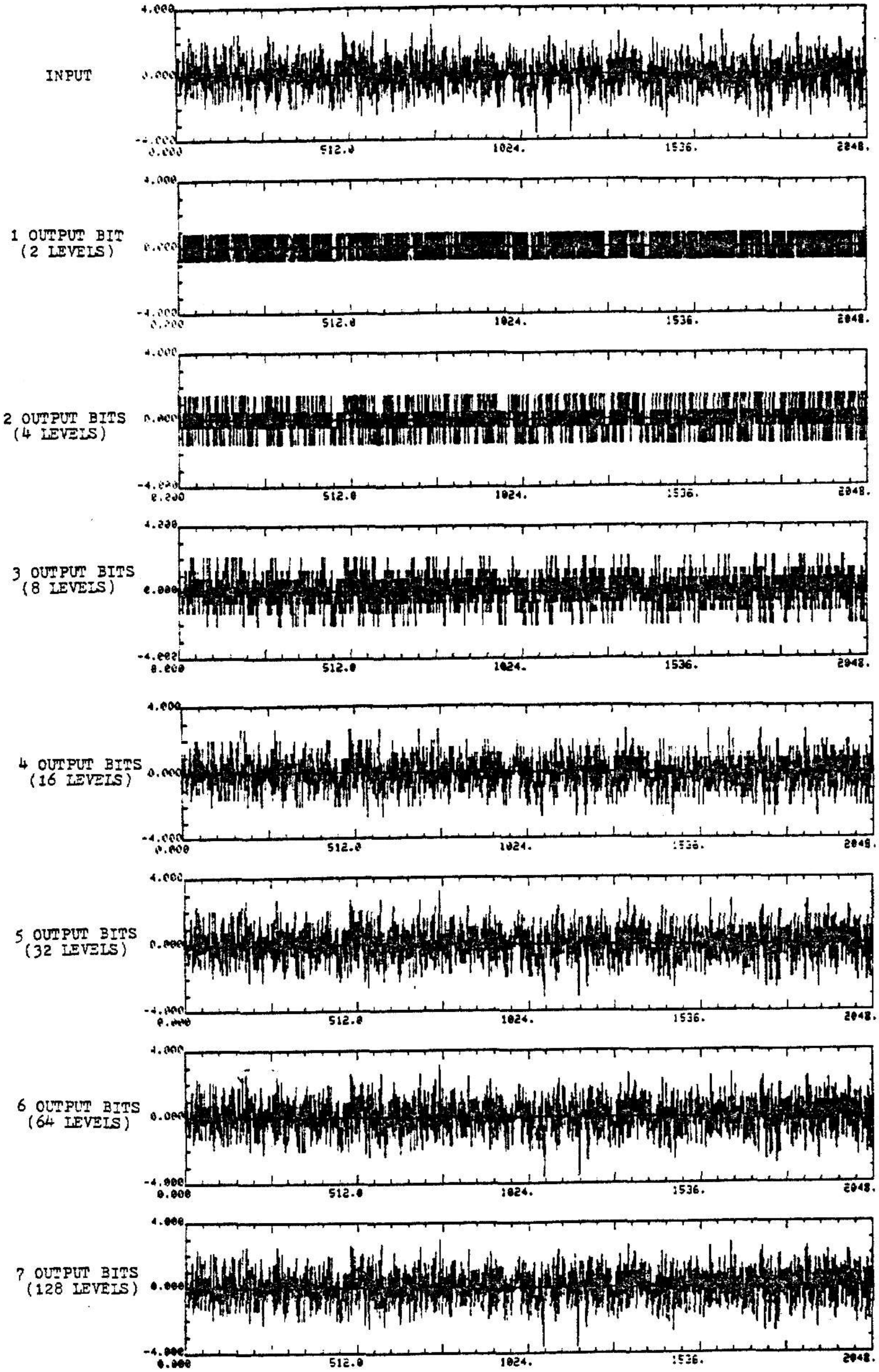
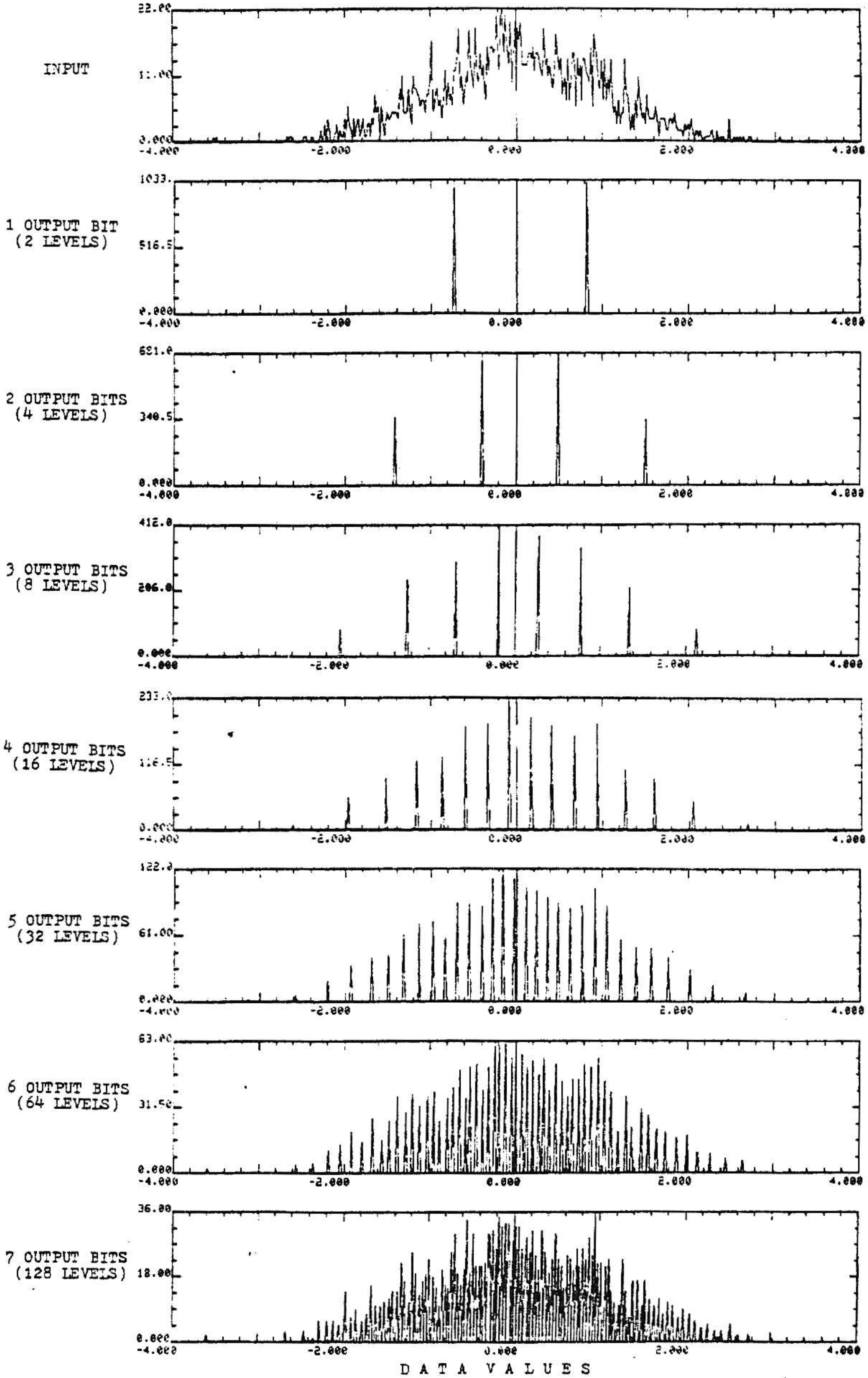

INPUT
1 OUTPUT BIT
(2 LEVELS)
2 OUTPUT BITS
(4 LEVELS)
3 OUTPUT BITS
(8 LEVELS)
4 OUTPUT BITS
(16 LEVELS)
5 OUTPUT BITS
(32 LEVELS)
6 OUTPUT BITS
(64 LEVELS)
7 OUTPUT BITS
(128 LEVELS)
4.000
0.000
-4.000
512.0
1024.
1536.
2048.

FIGURE 6.1


INPUT
1 OUTPUT BIT
(2 LEVELS)
2 OUTPUT BITS
(4 LEVELS)
3 OUTPUT BITS
(8 LEVELS)
4 OUTPUT BITS
(16 LEVELS)
5 OUTPUT BITS
(32 LEVELS)
6 OUTPUT BITS
(64 LEVELS)
7 OUTPUT BITS
(128 LEVELS)
-4.000
-2.000
0.000
2.000
4.000
D A T A   V A L U E S

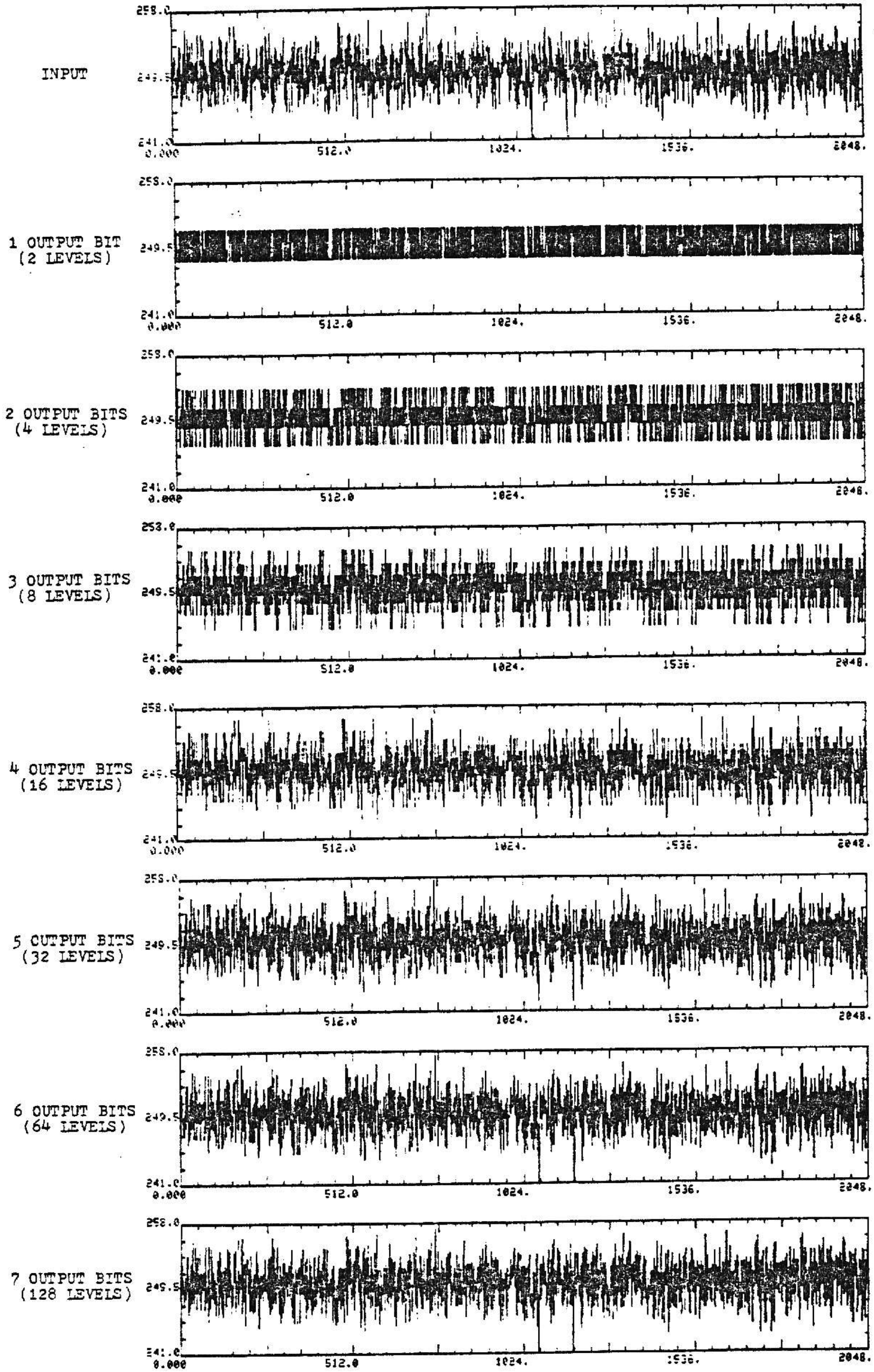

INPUT
1 OUTPUT BIT
(2 LEVELS)
2 OUTPUT BITS
(4 LEVELS)
3 OUTPUT BITS
(8 LEVELS)
4 OUTPUT BITS
(16 LEVELS)
5 OUTPUT BITS
(32 LEVELS)
6 OUTPUT BITS
(64 LEVELS)
7 OUTPUT BITS
(128 LEVELS)
258.0
249.5
241.0
0.000
512.0
1024.
1536.
2048.

FIGURE 6.2

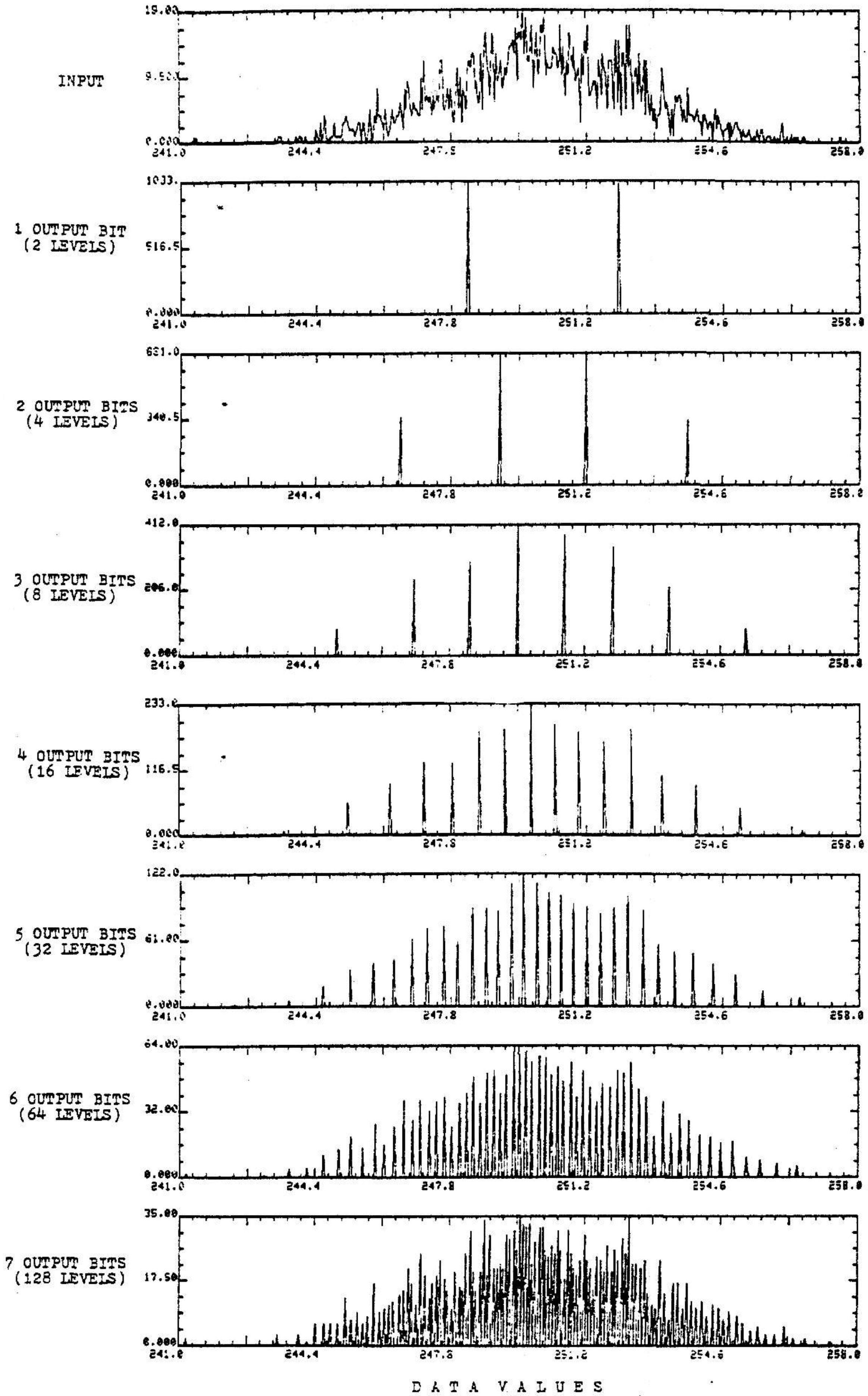



FIGURE 6.2 (CONT.)

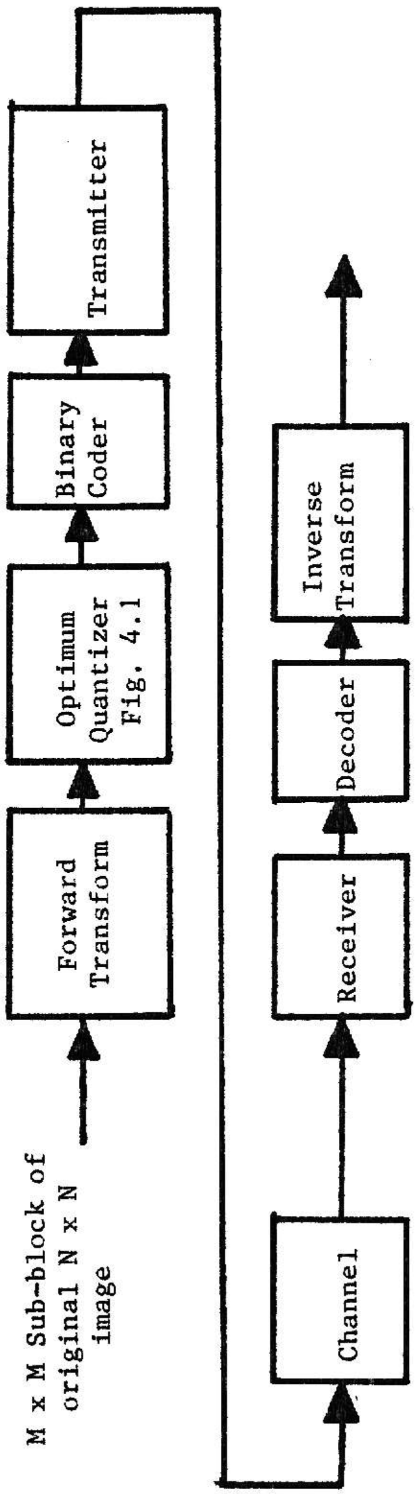



IMAGE PROCESSING SCHEME

FIGURE 6.3

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 8 | 8 | 8 | 7 | 7 | 7 | 5 | 5 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 8 | 8 | 7 | 6 | 5 | 5 | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 2 | 2 | 2 |
| 8 | 7 | 6 | 4 | 4 | 4 | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 7 | 6 | 4 | 3 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 7 | 5 | 4 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 5 | 4 | 2 | 2 | 2 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 3 | 3 | 2 | 2 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 3 | 3 | 2 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 3 | 2 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 3 | 2 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 3 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 3 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

ZONAL CODING

Typical numbers of output bits assigned for zonal transform coding. A 16 x 16 block is illustrated, each element is a pixel.

FIGURE 6.4

# REFERENCES

[1] J. Max, "Quantizing for Minimum Distortion," IRE Transactions On Informative Theory, Vol. IT-6, pp. 7-12, March 1960.

[2] A. Papoulis, Probability Random Variables, and Stochastic Processes, New York: McGraw-Hill, pp. 122-123, 1965.

[3] D. Hein, personal correspondence.

[4] S. Meyer, Data Analysis For Scientists and Engineers, New York: Wiley, pp. 229-231, 1975.

[5] C. Boor and S. Conte, Elementary Numerical Analysis: An Algorithmic Approach, New York: McGraw-Hill, pp. 32-33 and pp. 59-60, 1972.

[6] W. Pratt, Digital Image Processing, New York: Wiley, p. 185, 1978.

[7] W. Pratt, Digital Image Processing, New York: Wiley, pp. 670-675, 1978.

# THE OPTIMUM QUANTIZATION PROCESS

by

JAMES K. ALBRIGHT

B.S., Kansas State University, 1979

---

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Electrical Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1982

# ABSTRACT

The objective of this work is to mathematically describe the optimum quantizer. Once the results have been obtained for the general optimum quantizer, a specific case is considered. A digital computer implementation then establishes experimental values based on the mathematical derivation for the optimum quantizer. The experimental and theoretical values are then compared. The specific case considered is the optimum quantization process for a random gaussian sequence.