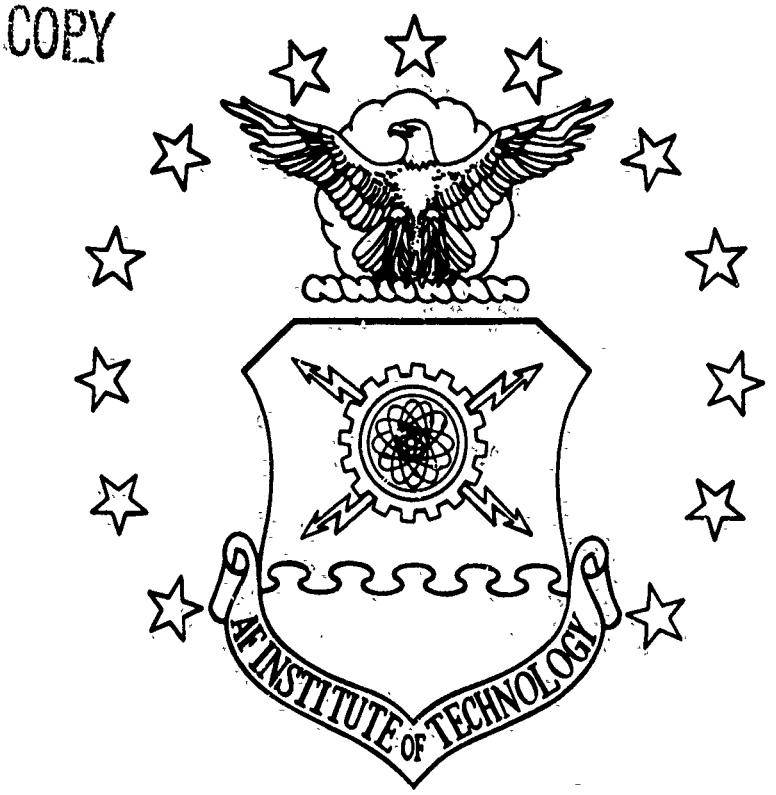

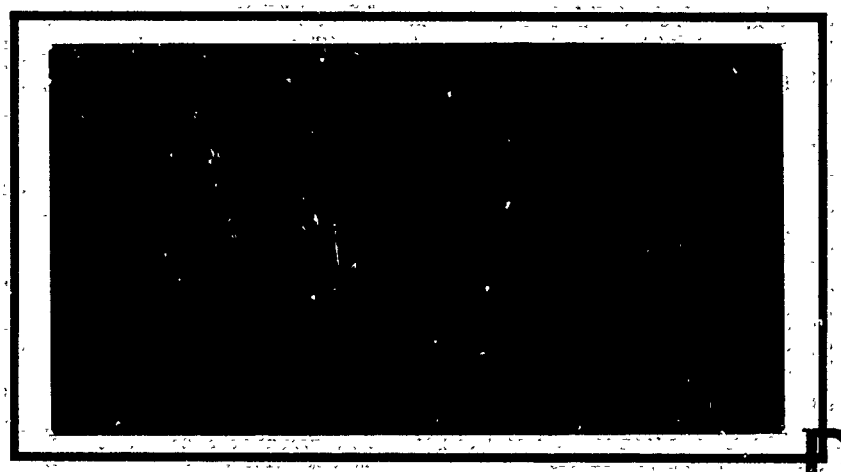

DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GE/ENG/90D-48

QUANTIZATION NOISE CHARACTERISTICS RESULTING FROM GAUSSIAN, NEGATIVE-EXPONENTIAL, AND SINUSOIDAL RANDOM INPUT SIGNALS

THESIS

Van N. Osborne
Captain, USAF

AFIT/GE/ENG/90D-48

Approved for public release; distribution unlimited

AFIT/GE/ENG/90D-48

QUANTIZATION NOISE CHARACTERISTICS RESULTING FROM GAUSSIAN, NEGATIVE-EXPONENTIAL, AND SINUSOIDAL RANDOM INPUT SIGNALS

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Electrical Engineering

Van N. Osborne, B.S., B.S.E.E.
Captain, USAF

December 1990

Preface

The purpose of this study was to investigate the trade-off between the number of quantization levels and the resulting quantization noise characteristics for three classes of commonly occurring signals. Prior efforts, particularly those addressing the related frequency spectra, had been primarily limited to the result of quantizing Gaussian signals.

Theoretical expressions were developed in terms of appropriate orthogonal polynomials. These expressions were used to determine specific noise characteristics resulting from the quantization process over various numbers of bits.

I would like to thank my thesis advisor, Dr. Vittal Pyati, for his guidance. Although his name does not appear in the bibliography, much of Chapter III has directly evolved from earlier, unreleased work that he had performed. I also want to thank my other thesis committee members, Lt Col David Norman and Capt Gregory Warhola, for their help. Finally, I want to express my gratitude to my wife, Diana, and to our two daughters, Vanessa and Rebecca, for their understanding and inspiration as I labored to finish this project.

Van N. Osborne

Table of Contents

List of Figures

List of Tables

Abstract

The purpose of this study was to investigate the trade-off between the number of quantization levels and the resulting noise characteristics for three classes of commonly occurring input signals, namely, those signals possessing Gaussian, negative-exponential and random sinusoidal distributions.

From a literature review, it was noted that much had been done to characterize the mean-squared error resulting from the quantization of a variety of input signal types. However, those efforts to characterize frequency spectra had been limited to the output spectrum resulting from an input with a Gaussian distribution. This study was able to characterize the mean-squared error, output spectrum and error spectrum for each of the three input signal classes considered.

This study derived expressions for each of the entities under consideration by expanding the nonlinear quantization function into a summation of orthogonal polynomials matched to the corresponding input signal distribution. Once accomplished, orthogonality properties were applied to provide usable expressions patterned as sums of intermodulation coefficients.

A set of three Fortran 77 programs were developed – each of which applied to one of the studied input signal classes. Each program required the quantization step size, one appropriate input signal parameter and the number of bits used in the quantization process. When provided each of these required values, the appropriate program produced upon demand either a mean-squared error value and a signal-to-quantization noise ratio or quantizer output spectrum data and quantization error spectrum data. Typical input power spectral densities were applied in order to produce the spectra data.

The study resulted in a set of tables which provided mean-squared error and signal-to-quantization noise ratio data based on various numbers of bits used for

the quantization process. Also, a number of plots displaying the power spectral densities under consideration were produced as based on similar numbers of bits. Among the recommendations provided is to extend the results of this thesis to include the effects of non-uniform quantization, since this thesis strictly considered uniform quantization.

QUANTIZATION NOISE CHARACTERISTICS RESULTING FROM GAUSSIAN, NEGATIVE-EXPONENTIAL, AND SINUSOIDAL RANDOM INPUT SIGNALS

I. Introduction

1.1 Background

It is commonly known that a quantized signal has undergone an irreversible process. The mapping of a signal with a continuous amplitude distribution to a signal with a discrete amplitude distribution introduces error which is referred to as quantization noise. This mapping is a nonlinear function and must be appropriately analyzed for its introduction of noise.

The noise which results from the quantization process affects the quality of the received signal. The number of discrete levels used in the quantization process has a direct bearing on the resulting quantization noise. As this number of levels is allowed to increase without bound, the quantization process becomes a one-to-one correspondence, and the resulting quantization noise disappears.

Perhaps the obvious answer to the quantization noise problem is to increase the number of quantization levels to an arbitrarily chosen large number. Unfortunately, such a solution would cause the cost and complexity of the necessary equipment to increase. In addition, as more levels are used, a larger bit transmittal rate becomes necessary in order to transmit the quantized information.

As the minimum number of necessary quantization levels is determined by considering the maximum allowable quantization noise, another complication arises.

Quantization does not identically affect different types of signals. The characteristics of the quantization noise depend heavily upon the characteristics of the input signal. For example, an input signal with an amplitude distribution evenly spread throughout the domain of the quantization function will result in quantization noise with different characteristics than that noise resulting from an input signal with a Gaussian amplitude distribution.

In order to prudently select the number of quantization levels to use for a given application, it becomes necessary to anticipate the characteristics of the quantization noise. Therefore, it is imperative that a relationship be developed between the number of quantization levels and the resulting noise characteristics for a given set of input signal amplitude distributions. The noise characteristics warranting particular interest are the mean-squared error (also known as and referred to as the normalized noise power) and the noise frequency spectrum. The quantizer output frequency spectrum also deserves consideration.

1.2 Problem

For three classes of input signals, this thesis effort has developed a relationship between the number of quantization levels and the resulting quantization noise characteristics.

1.3 Summary of Current Knowledge

There have been a number of prior studies involving quantization noise. Some of these efforts have included the numerical calculation of the mean-squared error for a variety of input signal classes. Other efforts concentrated on the quantization noise spectra, but were typically limited in scope to the result of an input with a Gaussian amplitude distribution. For a historical survey of past efforts, the reader is referred to Chapter II.

1.4 Assumptions

In order to proceed with this thesis effort, some assumptions were necessary. They were as follows:

1. The quantization process was assumed to be performed by an ideal quantizer, or a quantizer which introduces no nonlinearities to the quantization process other than the intended nonlinear quantization function. The consideration of a quantizer perturbed by the introduction of any additional nonlinearity was beyond the scope of this thesis.
2. The input signal was assumed to be free of noise. This assumption was made in order to concentrate exclusively on quantization noise.
3. The input signal was assumed to be a wide-sense stationary random process. This has been a standard assumption when analyzing in the frequency domain, since a random process possesses a power spectral density if it is wide-sense stationary.

1.5 Scope

This thesis was limited in scope to the analysis of the quantization noise resulting from an ideal, uniform, continuous-time quantizer. Neither nonuniform quantization nor sampling effects have been considered during the development of this analysis.

Quantization noise has been denoted the topic of concern for this thesis. The other noise product of the quantization process, saturation noise, has not been considered independently of quantization noise. This thesis incorporates both types of quantization related noise and does not distinguish between the effects of the two types. Gray and Zeoli have produced a study which optimizes the trade-off between these two results of the quantization process (8).

Finally, this thesis develops the expressions for the quantization noise frequency spectrum. However, the spectrum depends on the spectrum of the input signal. In order to produce output and error spectrums based on the derived equations, it was necessary to consider a single input autocorrelation function for each class of input signals. However, it should be noted that for any given input autocorrelation function, the derived equations may be difficult, if not impossible, to apply.

1.6 Approach

The approach of this thesis effort began by expressing the autocorrelation function of the quantization error in terms of various characteristics of the quantizer input and output. Those terms which included the quantizer output were then treated individually by expressing the output as a nonlinear function of the input.

The treatment of the nonlinear function was dependent upon the amplitude distribution of the input signal. For each of the three classes of inputs considered, the nonlinear function was series-expanded using expansion techniques derived by Barrett and Lampard (2). Once these series expansions were complete, the Fourier transform of the resulting autocorrelation function was determined. The result yielded a quantization noise spectrum and a quantizer output spectrum for each class of inputs considered.

Algorithms for determining the quantization mean-squared error, the quantizer output spectrum and the quantization error spectrum were developed by applying the derived equations. Since the required approach was dependent upon the class of input signal, multiple algorithms were necessary. The algorithms pertaining to the Gaussian and the negative-exponential distributed inputs were based on basic orthogonal polynomial identities as determined by Szegö (16). Each algorithms were then coded in the Fortran 77 computer language and executed to obtain the desired mean-squared error values and quantization related spectra for a variety of numbers of quantization levels.

Finally, it should be noted that the sections of this thesis which consider the Gaussian distributed input tend to parallel Velichkin's earlier work (18) and provide similar results. This effort provides an additional analysis involving the quantization noise spectrum. In addition, the sections which pertain to the other two classes of inputs provide entirely new material and new results.

1.7 Equipment

An ELXSI mainframe computer with a UNIX operating system and a Fortran 77 compiler was used to execute the computer programs developed under this thesis effort.

II. Historical Survey

Quantization has been a familiar topic in the digital communication field for a number of years. As a result, quantization mean-squared error derivations have appeared in many reputable textbooks on the subject matter. This resulting mean-squared error represented the quantization noise power

$$N_q = \frac{q^2}{12} \tag{1}$$

where q is the quantization step size of the quantizer, or the distance between quantization levels. Roden has provided the usual treatment (14:119-121).

This popular result has been based the assumption that the quantization error was uniformly distributed over its range. Unfortunately, this assumption has rarely applied to anything other than a classroom problem. However, this simple expression generally gave a good starting point.

2.1 The Gaussian Distributed Input

Probably the most obvious class of inputs to be considered was that class possessing a Gaussian probability density function. Many types of signals and noise possess an amplitude probability density which very closely resembles such a function.

Max sought to develop an algorithm which would determine the necessary quantization parameters to minimize distortion for both uniformly and Gaussian distributed inputs. His approach was to minimize the expected value of some function of the quantization error. He chose this function to be the square of the quantization error. Consequently, the mean-squared error became the value to be minimized (11:7–9).

Max considered both nonuniform and uniform quantization. In the nonuniform case, he partially differentiated his expression for the mean-squared error with respect to both the input and the output. Next, he equated both results to zero. He then employed iterative numerical techniques in order to solve these resulting equations and yield quantizer outputs and corresponding ranges of inputs. The matching of these outputs to ranges of inputs provided the minimum possible mean-squared error for a given number of quantization levels (11:8-9).

Similarly, Max applied his optimization techniques to the uniform quantization case. He partially differentiated his expression for the mean-squared error with respect to the uniform step size. Once again, he equated the result to zero. However, this time he employed iterative numerical techniques in order to yield the optimal step size. This step size would provide the minimum mean-squared error for a given number of levels (11:9).

The results for both the nonuniform and uniform quantization case were presented in tabular form for the number of output levels ranging from 1 to 36. In each case, the resulting mean-squared error was also determined and given (11:11-12).

2.2 The Rayleigh Distributed Input

As image processing and optical holography research became more common, the Rayleigh probability density function became more applicable to the quantization process. Pearlman and Senge recognized this trend and adapted Max's algorithm to determine the optimal quantization of an input possessing a Rayleigh distribution (12:101).

Pearlman and Senge inserted the Rayleigh probability density function into the equation for the mean-squared error. As Max had done, they also considered both nonuniform and uniform quantization. In both cases, partial differentiations were taken and equated to zero. Iterative Newton-Raphson techniques were used in both cases to determine the optimal step sizes. Also, in both cases, least-squares

curve fitting techniques were also applied to yield general approximation equations for the mean-squared error as a function of the number of quantization levels (under optimal quantization conditions) (12:102–103).

The effort was completed with the inclusion of tables providing the optimal step sizes, resulting mean-squared error, and output entropy for the number of quantization levels ranging from 2 to 64 (12:104–111).

2.3 A Wider Class of Inputs

As applications to the quantization process have increased, a wider class of inputs have become applicable. Lu and Wise applied techniques similar to those used by Max and by Pearlman and Senge to four input distributions: Gaussian, two-sided Rayleigh, Laplace, and two-sided gamma. Each of these considered distributions were two-sided and symmetrical (unlike the classic Rayleigh distribution considered by Pearlman and Senge). However, Lu and Wise considered only uniform quantization (10:471–472).

Lu and Wise were determined to avoid the massive tables provided by earlier investigators of the topic. Therefore, one of their prime objectives was to provide approximation techniques in a compact form. In order to do so, they applied curve fitting techniques to each of their results so that only a short list of parameters would require tabulation. These parameters could then be used to approximate the optimal step size and the resulting mean-squared error for any number of quantization levels ranging from 4 to 1024. Their final results, spanning four different input distributions and the above range of numbers of levels, were then able to fit in three small tables (10:472–473).

2.4 The Frequency Spectrum of the Output

The previously acknowledged efforts were each limited in scope to the consideration of the mean-squared quantization error, or the quantization noise power.

However, the distribution of this power across the frequency spectrum could be just as important, depending upon the particular application.

2.4.1 Early Work When the quantization of speech signals emerged, Bennett became one of the first to successfully characterize the spectrum of a quantized signal. Bennett restricted his consideration to input signals possessing a Gaussian distribution (3:463).

In the interest of examining the frequency spectrum, Bennett understood the importance of characterizing the autocorrelation function of the quantization error. He derived an approximation of such a characterization by employing classical probability density function transformation techniques and by applying Poisson's summation formula. The result was an autocorrelation function of quantization errors in terms of the autocorrelation function of the signal. The Wiener-Khinchine Theorem was then applied to provide an error power spectral density formula (3:463–468).

When Bennett examined the complete spectrum of the result of the quantization process, it became necessary for him to also consider the effects of sampling the original analog signal. While this inclusion tended to create some confusion regarding the effects of only the quantization process, it did provide an understanding of the effects of increased sampling frequency on the signal-to-quantization noise ratio for a given number of bits (3:453).

2.4.2 Other Approaches As the quantization operation became more commonplace, Velichkin also sought to characterize the spectrum of the output of a quantizer. He also limited himself to a Gaussian distributed input. However, unlike Bennett, Velichkin chose to consider the effects of sampling and quantization independently. This modularized his efforts and allowed him to examine the effects of quantization apart from those of the sampling process (18:70).

Velichkin used orthogonal polynomial expansion for the second-order Gaussian

probability distribution to obtain an exact, but computationally intensive equation for the autocorrelation function for the output of the quantizer. Once again, the Wiener-Khinchine theorem was applied to result in the power spectrum of the quantizer output (18:71–73).

Lever was interested in comparing the results of Bennett, Velichkin, and other notable quantization noise spectrum efforts made over the years. Like Bennett, Lever chose to analyze the effects of sampling and quantization jointly, but did so with a sensitivity to prior work which separated the effects of the two processes (9:201–203).

Lever was also able to compare theoretical and experimental signal-to-noise ratio results. He did so under two different circumstances. First, he considered the effects of an ideal quantizer. Next, he considered the effects of a quantizer which had been perturbed by the introduction of an additional nonlinearity. He was able to show that developed theoretical relationships were inadequate to provide accurate estimates when the quantization process was perturbed by an additional nonlinearity (9:203-206).

2.5 Closing Comments Regarding Previous Efforts

There have been a number of studies performed in the area of quantization noise. They have included the minimization of quantization noise for a variety of inputs. These noise minimization studies have often considered nonuniform, as well as uniform quantization. There have also been efforts made in the interest of characterizing the quantization noise spectrum. These efforts have typically been limited to Gaussian inputs undergoing uniform quantization. A matrix of the discussed prior studies and their applicability to the different input distributions and quantization types appears in Table 1.

There exists a relationship between this thesis and these past efforts. This thesis effort has developed quantization noise spectrum expressions resulting from Gaussian inputs, as well as two other classes of inputs. Some of the techniques used

Table 1. Prior Study Applicability Matrix

	Input Distributions			Quantization Type	
	Gaussian	Rayleigh	Others	Uniform	Nonuniform
Mean-Squared Error					
Max	X			X	X
Pearlman and Senge		X		X	X
Lu and Wise	X	Two-Sided	X	X	
Error Spectrum					
Bennett	X			X	
Velichkin	X			X	
Lever	X			X	

to develop these derivations were similar to those used in the prior quantization noise spectrum efforts. As already noted in Chapter I, the thesis effort regarding the Gaussian input specifically tended to parallel Velichkin's efforts (18) and provided similar results. The prior noise minimization studies were also useful as a comparison tool against the calculations resulting from the derived mean-squared error equations.

III. Theoretical Development

3.1 The Quantization Process

In order to appropriately study the effects of quantization, the quantization process itself must be understood on a basic level. Let the time-varying input to the quantizer is denoted as $x(t)$ and the output is denoted as $y(t)$. If $g(x)$ can be determined such that $y(t) = g[x(t)]$, or $y = g(x)$, the relationship between $y(t)$ and $x(t)$ for an ordinary Q-bit uniform quantizer with a step size of q and for $Q = 4$ bits is as illustrated in Figure 1.

For a Q-bit quantizer, there are 2^Q distinct levels with a step size of q between each level. Consequently, the normal operating region of the quantizer exists over a range of $q(2^Q - 1)$. As a result, the relationship illustrated in Figure 1 exists over a range of $x(t)$ from $-\frac{1}{2}q(2^Q - 1)$ to $+\frac{1}{2}q(2^Q - 1)$. Beyond this range of $x(t)$, the quantizer will simply output a level corresponding to $\pm\frac{1}{2}q(2^Q - 1)$ depending on the sign of the input. This phenomenon is known as saturation and $\pm\frac{1}{2}q(2^Q - 1)$ are denoted the saturation levels of the quantizer.

3.2 Quantization Error

3.2.1 Defining the Quantization Error The error resulting from the quantization process is known as the quantization error and can be determined as a function of time by the equation

$$e_q(t) = y(t) - x(t) \tag{2}$$

where once again, $x(t)$ and $y(t)$ are the quantizer input and output, respectively. The quantizer error only exists over the normal operating range of the quantizer.

For $|x(t)| > \frac{1}{2}q(2^Q - 1)$ saturation error results. It can also be determined by Equation 2. The analysis in this thesis incorporates both types of quantization related error, and does not distinguish between the effects of the two.

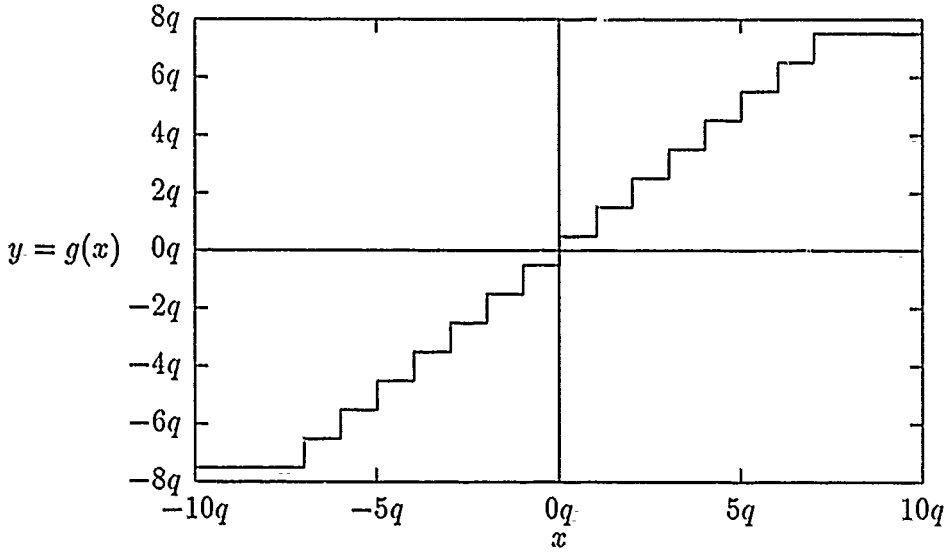

Figure 1. Quantizer Output vs Input for Q = 4 Bits

3.2.2 The General Quantization Noise Autocorrelation Problem

3.2.2.1 The Preliminary Expression In order to determine the power or the spectrum of the quantization noise, $x(t)$, $y(t)$ and $e_q(t)$ must be considered as random processes. As random processes, these functions will be denoted as $X(t)$, $Y(t)$ and $E_q(t)$, respectively. In addition, these random processes are assumed to be stationary in the wide sense, implying that the mean of the random process is constant and its autocorrelation is a function of time differential.

The initial goal is to obtain the autocorrelation of the error, defined as

$$R_{E_q}(\tau) \equiv E\left[E_q(t_1)E_q(t_2)\right] \tag{3}$$

where $\tau = t_2 - t_1$, and E is the expectation operator.

Equation 3 becomes

$$\begin{aligned} R_{E_q}(\tau) &= E\{[Y(t_1) - X(t_1)][Y(t_2) - X(t_2)]\} \\ &= E[Y(t_1)Y(t_2) - X(t_1)Y(t_2) - Y(t_1)X(t_2) + X(t_1)X(t_2)] \\ &= E[Y(t_1)Y(t_2)] - E[X(t_1)Y(t_2)] - E[Y(t_1)X(t_2)] \\ &\quad + E[X(t_1)X(t_2)] \\ &= R_Y(\tau) - R_{XY}(\tau) - R_{YX}(\tau) + R_X(\tau) \end{aligned} \tag{4}$$

where $R_{XY}(\tau)$ is the crosscorrelation function defined as

$$R_{XY}(\tau) \equiv E[X(t_1)Y(t_2)] \tag{5}$$

If $Y(t)$ can be determined as a function of $X(t)$, or if $g(x)$ can be determined such that $y = g(x)$, then the autocorrelation definition can be used to yield

$$R_Y(\tau) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x_1)g(x_2)W(x_1, x_2; \tau)\, dx_1\, dx_2 \tag{6}$$

where $x_1 = x(t_1)$, an observation of the random variable $X(t_1)$. Similarly, $x_2 = x(t_2)$. Also, $W(x_1, x_2; \tau)$ is the joint probability density function applicable to $X(t)$ for the bivariate case.

The second and third terms appearing in Equation 4 require more advanced treatment. Bussgang proved that if an input possesses a Gaussian distributed amplitude, the crosscorrelation of the input and output of a distorting device "will be proportional to the autocorrelation of the input signal" (4:5). Barrett and Lampard broadened the scope of Bussgang's important theorem to include the distributions discussed in this thesis (2). They also provided the expression to determine the constant satisfying

$$R_{XY}(\tau) = cR_X(\tau) \tag{7}$$

as

$$c = \int_{-\infty}^{\infty} g(x_2) W_2(x_2) \left(\frac{x_2 - \mu_2}{\sigma_2^2} \right) dx_2 \tag{8}$$

where $W_2(x_2)$ is the marginal probability density function applicable to $X(t_2)$, and μ_2 and σ_2^2 are the mean and the variance, respectively, of $X(t_2)$ (2:25). It also follows from these results that

$$R_{XY}(\tau) = R_{YX}(\tau) \tag{9}$$

since $R_X(\tau)$ is an autocorrelation and is, therefore, an even function.

By applying Equations 4, 6, 7, 8 and 9, the following general relationship for the autocorrelation of the quantization results:

$$\begin{aligned} R_{E_q}(\tau) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x_1) g(x_2) W(x_1, x_2; \tau)\, dx_1\, dx_2 \\ &\quad + \left[1 - 2 \int_{-\infty}^{\infty} g(x_2) W_2(x_2) \left(\frac{x_2 - \mu_2}{\sigma_2^2} \right) dx_2 \right] R_X(\tau) \end{aligned} \tag{10}$$

3.2.2.2 Treatment of the Nonlinearity Since it is assumed that the probability density functions required to evaluate the expression given as Equation 10 are known, the next obstacle is the determination of an appropriate expression for the nonlinear relationship $y = g(x)$ so that the above integrals can be evaluated.

Thomas provided a general technique for treating a nonlinearity which involves a series expansion of the nonlinearity. The expansion can then be employed to yield an appropriate expression for $y = g(x)$ (17:314–323). However, Thomas' own utilization of his technique is useful only if the input possesses an amplitude distribution which is Gaussian. The general technique, as applied to the quantization problem, is as outlined in the following paragraphs.

The nonlinear function, $g(x)$, can be represented by the series

$$g(x) = \sum_{n=0}^{\infty} c_n \psi_n(x) \tag{11}$$

where $\psi_n(x)$ are orthonormal polynomials with respect to the marginal probability density function, $W(x)$. This implies that

$$\int_{-\infty}^{\infty} W(x)\psi_m(x)\psi_n(x)\,dx = \delta_{mn} \tag{12}$$

is satisfied, where δ_{mn} is the Kronecker delta satisfying

$$\delta_{mn} = \begin{cases} 1 & \text{if } m = n \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

The equation

$$c_n = \int_{-\infty}^{\infty} g(x)W(x)\psi_n(x)\,dx \tag{14}$$

is used to determine c_n. This integral can become quite a formidable task unless $\psi_n(x)$ is carefully chosen with regard to $W(x)$.

Since $g(x)$ is discontinuous, the integral appearing as Equation 14 may be approached as the Riemann-Stieltjes integral

$$\int_a^b f(x)\,d\alpha(x)$$

where $f(x)$ corresponds to $g(x)$ and $d\alpha(x)$ corresponds to $W(x)\psi_n(x)\,dx$. In addition, the limits a and b correspond to $-\infty$ and ∞, respectively. Applying the formula for integration by parts applicable to the Riemann-Stieltjes integral (1:144) yields

$$c_n = [g(x)F_n(x)]\Big|_{-\infty}^{\infty} - \int_{-\infty}^{\infty} F_n(x)\,dg(x) \tag{15}$$

where

$$\frac{d}{dx}F_n(x) = W(x)\psi_n(x) \tag{16}$$

By inspection of Figure 1

$$dg(x) = \sum_{i=-M}^{M} q\delta(x - iq)\,dx \tag{17}$$

where $M = 2^{Q-1} - 1$ and $\delta(x)$ is the Dirac delta function which satisfies

$$\delta(x) = \lim_{b\to 0} \frac{1}{|b|} \exp\left[-\pi\left(\frac{x}{b}\right)^2\right] \tag{18}$$

(6:50). Consequently, the sampling property of this delta function can be determined as

$$\begin{aligned} \int_{-\infty}^{\infty} f(x)\delta(x)\,dx &= \lim_{b\to 0} \int_{-\infty}^{\infty} \frac{1}{|b|} \exp\left[-\pi\left(\frac{x}{b}\right)^2\right] f(x)\,dx \\ &= f(0) \end{aligned} \tag{19}$$

for any continuous function, $f(x)$. Therefore, inserting Equation 17 into Equation 15 yields

$$c_n = [g(x)F_n(x)]\Big|_{-\infty}^{\infty} - \int_{-\infty}^{\infty} \sum_{i=-M}^{M} q\delta(x - iq)F_n(x)\,dx \tag{20}$$

Some rearrangement and the use of the sampling property determined as Equation 19 provides the following identity:

$$c_n = [g(x)F_n(x)]\Big|_{-\infty}^{\infty} - \sum_{i=-M}^{M} qF_n(x)\Big|_{x=iq} \tag{21}$$

3.3 The Gaussian Case

3.3.1 The Noise Autocorrelation Problem for a Gaussian Input If the input signal level possesses a Gaussian probability distribution, it becomes necessary to consider both the first and second order probability density functions which possess

the following forms respectively:

$$W(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right) \tag{22}$$

and

$$W(x_1, x_2; \tau) = \frac{1}{2\pi\sigma^2[1-\rho_x^2(\tau)]^{\frac{1}{2}}} \exp\left(-\frac{x_1^2 + x_2^2 - 2x_1x_2\rho_x^2(\tau)}{2\sigma^2[1-\rho_x^2(\tau)]}\right) \tag{23}$$

where the distributions are assumed to be shifted so that the mean of $X(t)$ is zero for all values of t. The variance, σ^2, is equivalent to $R_X(0)$, and the correlation coefficient, $\rho_x(\tau)$, is equivalent to $\frac{R_X(\tau)}{R_X(0)}$, or $\frac{R_X(\tau)}{\sigma^2}$.

The orthogonality property for the Hermite polynomial is

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \exp\left(-\frac{x^2}{2}\right) H_m(x) H_n(x)\, dx = \delta_{mn} n! \tag{24}$$

where $H_n(x)$ is the n-th degree Hermite polynomial defined by

$$H_n(x) \equiv (-1)^n \exp\left(\frac{x^2}{2}\right) \frac{d^n}{dx^n}\left[\exp\left(-\frac{x^2}{2}\right)\right] \tag{25}$$

for non-negative integer values of n (2:27). The change of variables mapping x to $\frac{x}{\sigma}$ as performed on Equation 24 results in

$$\frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{\infty} \exp\left(-\frac{x^2}{2\sigma^2}\right) H_m\left(\frac{x}{\sigma}\right) H_n\left(\frac{x}{\sigma}\right) dx = \delta_{mn} n! \tag{26}$$

which implies that a suitable $\psi_n(x)$ satisfying Equation 12 is

$$\psi_n(x) = \frac{1}{\sqrt{n!}} H_n\left(\frac{x}{\sigma}\right) \tag{27}$$

or

$$\psi_n(x) = \frac{1}{\sqrt{n!}} (-1)^n \exp\left(\frac{x^2}{2\sigma^2}\right) \frac{d^n}{d\left(\frac{x}{\sigma}\right)^n}\left[\exp\left(-\frac{x^2}{2\sigma^2}\right)\right]$$

$$= \frac{\sigma^n}{\sqrt{n!}}(-1)^n \exp\left(\frac{x^2}{2\sigma^2}\right) \frac{d^n}{dx^n}\left[\exp\left(-\frac{x^2}{2\sigma^2}\right)\right] \tag{28}$$

Inserting the expressions given as Equations 22 and 28 into Equation 21 provides the following for $n = 1, 2, 3, \ldots$:

$$\begin{aligned}
c_n &= \left\{\frac{(-1)^n g(x)}{\sqrt{2\pi n!}}\sigma^{n-1}\frac{d^{n-1}}{dx^{n-1}}\left[\exp\left(-\frac{x^2}{2\sigma^2}\right)\right]\right\}\Bigg|_{-\infty}^{\infty} \\
&\quad - \sum_{i=-M}^{M} \frac{q(-1)^n}{\sqrt{2\pi n!}}\sigma^{n-1}\frac{d^{n-1}}{dx^{n-1}}\left[\exp\left(-\frac{x^2}{2\sigma^2}\right)\right]\Bigg|_{x=iq} \\
&= \frac{q}{\sqrt{2\pi n!}}\sum_{i=-M}^{M}\exp\left(-\frac{x^2}{2\sigma^2}\right)(-1)^{n-1}\exp\left(\frac{x^2}{2\sigma^2}\right)\frac{d^{n-1}}{d\left(\frac{x}{\sigma}\right)^{n-1}}\left[\exp\left(-\frac{x^2}{2\sigma^2}\right)\right]\Bigg|_{x=iq} \\
&\quad - \left\{\frac{g(x)}{\sqrt{2\pi n!}}\exp\left(-\frac{x^2}{2\sigma^2}\right)(-1)^{n-1}\exp\left(\frac{x^2}{2\sigma^2}\right)\frac{d^{n-1}}{d\left(\frac{x}{\sigma}\right)^{n-1}}\left[\exp\left(-\frac{x^2}{2\sigma^2}\right)\right]\right\}\Bigg|_{-\infty}^{\infty} \\
&= \frac{q}{\sqrt{2\pi n!}}\sum_{i=-M}^{M}\exp\left(-\frac{(iq)^2}{2\sigma^2}\right)H_{n-1}\left(\frac{iq}{\sigma}\right) \\
&\quad - \left[\frac{g(x)}{\sqrt{2\pi n!}}\exp\left(-\frac{x^2}{2\sigma^2}\right)H_{n-1}\left(\frac{x}{\sigma}\right)\right]\Bigg|_{-\infty}^{\infty}
\end{aligned} \tag{29}$$

Note that the second term vanishes since

$$\lim_{x\to\infty}\frac{p(x)}{\exp(x^2)} = 0 \tag{30}$$

for any polynomial $p(x)$. Therefore, for $n = 1, 2, 3, \ldots$,

$$c_n = \frac{q}{\sqrt{2\pi n!}}\sum_{i=-M}^{M}\exp\left(-\frac{(iq)^2}{2\sigma^2}\right)H_{n-1}\left(\frac{iq}{\sigma}\right) \tag{31}$$

The $n = 0$ case must be considered separately. For $n = 0$, Equation 14 becomes

$$c_0 = \int_{-\infty}^{\infty} g(x)W(x)\,dx \tag{32}$$

since $\psi_0(x) = 1$. Now, since for the Gaussian case, $g(x)$ is an odd function and $W(x)$ is an even function, the product $g(x)W(x)$ is odd and

$$c_0 = 0 \tag{33}$$

An important note regarding the Hermite polynomial, $H_n(x)$, is that if n is even, each term within $H_n(x)$ with a nonzero coefficient possesses an even power of x. Similarly, if n is odd, each term within $H_n(x)$ with a nonzero coefficient possesses an odd power of x. This leads to the following observation:

$$H_n(-x) = \begin{cases} -H_n(x) & \text{n odd} \\ H_n(x) & \text{n even} \end{cases} \tag{34}$$

Applying this property to Equation 31 provides

$$c_n = \begin{cases} \frac{q}{\sqrt{2\pi n!}} \sum_{i=-M}^{M} \exp\left(-\frac{(iq)^2}{2\sigma^2}\right) H_{n-1}\left(\frac{iq}{\sigma}\right) & \text{n odd} \\ 0 & \text{n even} \end{cases} \tag{35}$$

It is now possible to express the nonlinear function, $g(x)$, in the series representation

$$g(x) = \frac{q}{\sqrt{2\pi}} \sum_{k=0}^{\infty} \sum_{i=-M}^{M} \frac{1}{(2k+1)!} \exp\left(-\frac{(iq)^2}{2\sigma^2}\right) H_{2k}\left(\frac{iq}{\sigma}\right) H_{2k+1}\left(\frac{x}{\sigma}\right) \tag{36}$$

The new expression can now be applied to the first portion of the $R_{E_q}(\tau)$ expression, given as Equation 10, or equivalently to the $R_Y(\tau)$ expression, given as Equation 6. But first, $W(x_1, x_2; \tau)$ must be treated appropriately.

In order to simplify the integral given in Equation 6 by taking advantage of the orthogonality property of the Hermite polynomial, $W(x_1, x_2; \tau)$ can be expanded

into the following form:

$$W(x_1,x_2;\tau) = \frac{1}{2\pi\sigma^2}\exp\left[-\frac{1}{2}\left(\frac{x_1^2+x_2^2}{\sigma^2}\right)\right] \bullet \sum_{m=0}^{\infty}\rho_x^m(\tau)\frac{H_m\left(\frac{x_1}{\sigma}\right)H_n\left(\frac{x_2}{\sigma}\right)}{m!} \quad (37)$$

The technique employed to provide this expansion was provided by Barrett and Lampard (2:27).

By utilizing Equations 36 and 37, and by rearranging the orders of summation and integration, Equation 6 now becomes

$$\begin{aligned} R_Y(\tau) = \frac{q^2}{2\pi}\sum_{i=-M}^{M}\sum_{j=-M}^{M}\sum_{k=0}^{\infty}\sum_{l=0}^{\infty}\sum_{m=0}^{\infty}\Bigg\{&\exp\left[-\frac{(iq)^2+(jq)^2}{2\sigma^2}\right] \\ &\bullet\frac{\rho_x^m(\tau)}{m!}H_{2k}\left(\frac{iq}{\sigma}\right)H_{2l}\left(\frac{jq}{\sigma}\right) \\ &\bullet\left[\frac{1}{\sqrt{2\pi}\sigma(2k+1)!}\int_{-\infty}^{\infty}\exp\left(-\frac{x_1^2}{2\sigma^2}\right)H_{2k+1}\left(\frac{x_1}{\sigma}\right)H_m\left(\frac{x_1}{\sigma}\right)dx_1\right] \\ &\bullet\left[\frac{1}{\sqrt{2\pi}\sigma(2l+1)!}\int_{-\infty}^{\infty}\exp\left(-\frac{x_2^2}{2\sigma^2}\right)H_{2l+1}\left(\frac{x_2}{\sigma}\right)H_m\left(\frac{x_2}{\sigma}\right)dx_2\right]\Bigg\} \end{aligned} \quad (38)$$

Utilizing the orthogonality property given by Equation 26 allows the simplification of Equation 38 to

$$\begin{aligned} R_Y(\tau) &= \frac{q^2}{2\pi}\sum_{i=-M}^{M}\sum_{j=-M}^{M}\sum_{k=0}^{\infty}\left\{\frac{\rho_x^{2k+1}(\tau)}{(2k+1)!}\exp\left[-\frac{(iq)^2+(jq)^2}{2\sigma^2}\right]\right. \\ &\quad\left.\bullet H_{2k}\left(\frac{iq}{\sigma}\right)H_{2k}\left(\frac{jq}{\sigma}\right)\right\} \\ &= \frac{q^2}{2\pi}\sum_{k=0}^{\infty}\frac{\rho_x^{2k+1}(\tau)}{(2k+1)!}\left[\sum_{i=-M}^{M}\exp\left(-\frac{(iq)^2}{2\sigma^2}\right)H_{2k}\left(\frac{iq}{\sigma}\right)\right]^2 \end{aligned} \quad (39)$$

which takes the form

$$R_Y(\tau) = \sum_{k=0}^{\infty}a_k R_X^{2k+1}(\tau) \quad (40)$$

where

$$a_k = \frac{q^2}{2\pi(2k+1)!}\left(\frac{1}{\sigma^2}\right)^{2k+1}\left[\sum_{i=-M}^{M}\exp\left(-\frac{(iq)^2}{2\sigma^2}\right)H_{2k}\left(\frac{iq}{\sigma}\right)\right]^2 \tag{41}$$

Now, the second term in Equation 10 may be attacked by first evaluating the integral given by Equation 8. For the case at hand, Equation 8 becomes

$$c = \int_{-\infty}^{\infty} g(x)\left[\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{x^2}{2\sigma^2}\right)\right]\frac{x}{\sigma^2}\,dx \tag{42}$$

Noting that this integrand is an even function of x allows the use of the following equation:

$$c = \frac{2}{\sqrt{2\pi}\sigma}\int_0^{\infty} g(x)\frac{x}{\sigma^2}\exp\left(-\frac{x^2}{2\sigma^2}\right)\,dx \tag{43}$$

Ignoring the form of $g(x)$ derived as Equation 36 and performing the integration given as Equation 43 as a finite sum of integrals over intervals where $g(x)$ is continuous yields

$$\begin{aligned} c &= \sqrt{\frac{2}{\pi}}\frac{1}{\sigma}\left\{\sum_{m=0}^{M}\left[\int_{mq}^{(m+1)q}\left(m+\frac{1}{2}\right)q\frac{x}{\sigma^2}\exp\left(-\frac{x^2}{2\sigma^2}\right)\,dx\right]\right. \\ &\quad \left.+\int_{(M+1)q}^{\infty}\left(M+\frac{1}{2}\right)q\frac{x}{\sigma^2}\exp\left(-\frac{x^2}{2\sigma^2}\right)\,dx\right\} \\ &= \sqrt{\frac{2}{\pi}}\frac{1}{\sigma}\left\langle\sum_{m=0}^{M}\left\{\left(m+\frac{1}{2}\right)q\left[\exp\left(-\frac{m^2q^2}{2\sigma^2}\right)-exp\left(-\frac{(m+1)^2q^2}{2\sigma^2}\right)\right]\right\}\right. \\ &\quad \left.+\left(M+\frac{1}{2}\right)q\exp\left(-\frac{(M+1)^2q^2}{2\sigma^2}\right)\right\rangle \end{aligned} \tag{44}$$

Expanding the summation and collecting terms provides the following expression:

$$c = \sqrt{\frac{2}{\pi}}\frac{q}{\sigma}\left[\frac{1}{2}+\sum_{m=1}^{M}\exp\left(-\frac{m^2q^2}{2\sigma^2}\right)\right] \tag{45}$$

The autocorrelation of the quantization error can now be expressed as a function of the autocorrelation of the input in the following manner:

$$\begin{aligned} R_{E_q}(\tau) &= \frac{q^2}{2\pi\sigma^2}\sum_{k=0}^{\infty}\left\{\frac{1}{(2k+1)!}\left[\sum_{i=-M}^{M}\exp\left(-\frac{(iq)^2}{2\sigma^2}\right)H_{2k}\left(\frac{iq}{\sigma}\right)\right]^2\right. \\ &\quad \left.\bullet\left(\frac{1}{\sigma^2}\right)^{2k+1}R_X^{2k+1}(\tau)\right\} \\ &\quad +\left\{1-\sqrt{\frac{2}{\pi}}\frac{q}{\sigma}\left[1+2\sum_{m=1}^{M}\exp\left(-\frac{(mq)^2}{2\sigma^2}\right)\right]\right\}R_X(\tau) \end{aligned} \tag{46}$$

3.3.2 The Determination of Some Noise Related Figures of Merit for a Gaussian Input

3.3.2.1 The Normalized Noise Power Once Equation 46 has been provided, the determination of an expression for the normalized noise power becomes quite trivial. The normalized noise power, or the mean-squared error, is merely the autocorrelation of the quantization error evaluated at a time differential of zero, or

$$\begin{aligned} N_q &= R_{E_q}(\tau=0) \\ &= \frac{q^2}{2\pi}\sum_{k=0}^{\infty}\left\{\frac{1}{(2k+1)!}\left[\sum_{i=-M}^{M}\exp\left(-\frac{(iq)^2}{2\sigma^2}\right)H_{2k}\left(\frac{iq}{\sigma}\right)\right]^2\right\} \\ &\quad +\sigma^2\left\{1-\sqrt{\frac{2}{\pi}}\frac{q}{\sigma}\left[1+2\sum_{m=1}^{M}\exp\left(-\frac{(mq)^2}{2\sigma^2}\right)\right]\right\} \end{aligned} \tag{47}$$

3.3.2.2 The Signal-to-Quantization Noise Ratio Now that Equation 47 has been provided, a signal-to-quantization noise ratio expression also becomes trivial since the normalized signal power in this case is

$$S = R_X(\tau=0) = \sigma^2 \tag{48}$$

Therefore, the desired ratio then becomes

$$\frac{S}{N_q} = \left\langle \frac{q^2}{2\pi\sigma^2} \sum_{k=0}^{\infty} \left\{ \frac{1}{(2k+1)!} \left[\sum_{i=-M}^{M} \exp\left(-\frac{(iq)^2}{2\sigma^2}\right) H_{2k}\left(\frac{iq}{\sigma}\right) \right]^2 \right\} + 1 - \sqrt{\frac{2}{\pi}} \frac{q}{\sigma} \left[1 + 2 \sum_{m=1}^{M} \exp\left(-\frac{(mq)^2}{2\sigma^2}\right) \right] \right\rangle^{-1} \quad (49)$$

3.3.3 The Determination of the Quantization Noise Spectrum for a Gaussian Input Once the relationship for the autocorrelation of the quantization error has been provided as in Equation 46, the quantization error spectrum can be determined by applying the Wiener-Khinchine relationship. This relationship, given in Shanmugan and Breipohl (15:145), is

$$\begin{aligned} G_{E_q}(f) &= \mathcal{F}\left[R_{N_q}(\tau)\right] \\ &= \int_{-\infty}^{\infty} R_{E_q}(\tau) \exp(-j2\pi f\tau)\, d\tau \end{aligned} \quad (50)$$

where $j = \sqrt{-1}$. In other words, the power spectral density of the quantization error is the Fourier transform of the applicable autocorrelation function.

The Fourier transform operation is well known as being a linear operation. As a result, the application of the Wiener-Khinchine relationship to Equation 46 results in

$$\begin{aligned} G_{E_q}(f) = {} & \frac{q^2}{2\pi} \sum_{k=0}^{\infty} \left\{ \frac{1}{(2k+1)!} \left[\sum_{i=-M}^{M} \exp\left(-\frac{(iq)^2}{2\sigma^2}\right) H_{2k}\left(\frac{iq}{\sigma}\right) \right]^2 \right. \\ & \left. \bullet \left(\frac{1}{\sigma^2}\right)^{2k+1} \mathcal{F}\left[R_X^{2k+1}(\tau)\right] \right\} \\ & + \left\{ 1 - \sqrt{\frac{2}{\pi}} \frac{q}{\sigma} \left[1 + 2 \sum_{m=1}^{M} \exp\left(-\frac{(mq)^2}{2\sigma^2}\right) \right] \right\} \mathcal{F}\left[R_X(\tau)\right] \end{aligned} \quad (51)$$

which takes the form

$$G_{E_q}(f) = \sum_{k=0}^{\infty} a_k \mathcal{F}\left[R_X^{2k+1}(\tau)\right] + b\mathcal{F}\left[R_X(\tau)\right] \tag{52}$$

where a_k is as defined in Equation 41 and

$$b = 1 - \sqrt{\frac{2}{\pi}}\frac{q}{\sigma}\left[1 + 2\sum_{m=1}^{M} \exp\left(-\frac{(mq)^2}{2\sigma^2}\right)\right] \tag{53}$$

If an identical approach is taken regarding Equation 40, the following relationship results for the output spectrum:

$$G_Y(f) = \sum_{k=0}^{\infty} a_k \mathcal{F}\left[R_X^{2k+1}(\tau)\right] \tag{54}$$

The intermodulation coefficients, a_k, will require a certain amount of involved computation – especially since these coefficients converge towards zero somewhat slowly as k increases without bound. However, ignoring tl . problem for the moment, it should be noted that the evaluation of $\mathcal{F}\left[R_X^{2k+1}(\tau)\right]$ as k increases without bound is not a trivial exercise for the general $R_X(\tau)$.

A manageable $R_X(\tau)$ with some application to communications is

$$R_X(\tau) = \exp(-\alpha|\tau|) \tag{55}$$

where α is a positive constant and acts as a damping factor. Inserting this input correlation function into the error power spectral density equation results in

$$\begin{aligned} G_{E_q}(f) &= \sum_{k=0}^{\infty} a_k \mathcal{F}\left\{\exp[-\alpha(2k+1)|\tau|]\right\} + b\mathcal{F}\left\{\exp[-\alpha|\tau|]\right\} \\ &= \sum_{k=0}^{\infty} a_k \left[\frac{2\alpha(2k+1)}{\alpha^2(2k+1)^2 + (2\pi f)^2}\right] + b\left[\frac{2\alpha}{\alpha^2 + (2\pi f)^2}\right] \end{aligned} \tag{56}$$

Similarly,

$$G_Y(f) = \sum_{k=0}^{\infty} a_k \left[\frac{2\alpha(2k+1)}{\alpha^2(2k+1)^2 + (2\pi f)^2} \right] \tag{57}$$

3.4 The Negative-Exponential Case

3.4.1 The Noise Autocorrelation Problem for a Negative-Exponential Input

If a signal with a Gaussian amplitude distribution undergoes a narrow bandpass operation, the resulting envelope has a Rayliegh first order distribution. If following the filtering operation, the signal undergoes a square law detection operation which introduces no time delay, then the signal level of the output possesses the second order probability density

$$\begin{aligned} W(x_1, x_2; \tau) &= \frac{1}{x_0^2[1-\mu^2(\tau)]} I_0 \left[\frac{2\sqrt{x_1 x_2}}{x_0} \frac{\mu(\tau)}{1-\mu^2(\tau)} \right] \\ &\quad \bullet \exp \left[-\frac{x_1 + x_2}{x_0[1-\mu^2(\tau)]} \right] \end{aligned} \tag{58}$$

for $0 \leq (x_1, x_2) < \infty$ (2:27). The parameter x_0 corresponds to the expected value of x. The operation $I_m(x)$ is the m-th order modified (or hyperbolic) Bessel function. The function $\mu(\tau)$ is related to the autocorrelation of $X(t)$. This relationship, as well as its derivation, appears in Section A.1 of this thesis.

The resulting first order probability density function is the familiar negative-exponential density function

$$W(x) = \begin{cases} \frac{1}{x_0} \exp\left(-\frac{x}{x_0}\right) & 0 \leq x < \infty \\ 0 & \text{otherwise} \end{cases} \tag{59}$$

Of particular note is the constraint that x must be non-negative. This constraint necessitates a modification of the quantization process. Since x must be non-negative, there is no need to consider nonlinearities for negative values of x. Now, the appropriate relationship between the output of the quantizer, $y(t)$, and the

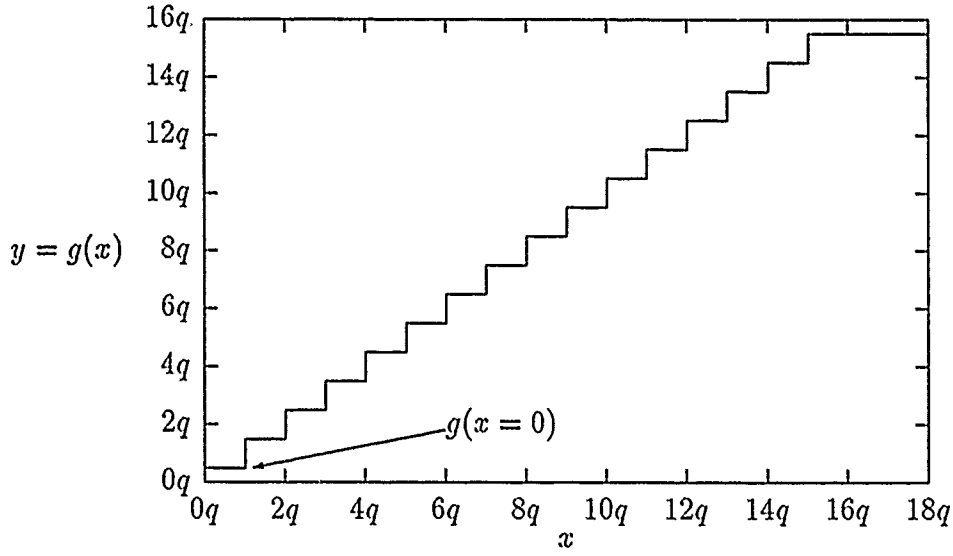

Figure 2. Non-Negative Quantizer Output vs Input for Q = 4 Bits

input, $x(t)$, appears as in Figure 2.

As before, for the Q-bit uniform quantizer, there are 2^Q distinct levels with a step size of q between each level. However, now saturation occurs when $x(t)$ equals $q(2^Q - 1) + g(x = 0)$, or the largest $x(t)$ which equals its corresponding $y(t)$. Therefore, now the operating region of the quantizer exists over a range of $q(2^Q - 1) + g(x = 0)$.

As for the general case, the autocorrelation function for the quantization noise will follow the relationship described in Equation 10. The treatment of the nonlinearity will also match the treatment given in Equations 11 through 15. However, Equation 17 now becomes

$$\frac{d}{dx}[g(x)] = \sum_{i=1}^{N} q\delta(x - iq) \tag{60}$$

where $N = 2^Q - 1$. As a result, Equation 21 now becomes

$$c_n = g(x)F_n(x)|_0^\infty - \sum_{i=1}^{N} qF_n(x)|_{x=iq} \tag{61}$$

where $\psi_n(x)$ are now orthonormal polynomials with respect to the marginal probability density function now under consideration. As before, $F_n(x)$ is as defined in Equation 16.

The orthogonality property for the Laguerre polynomial is

$$\int_0^\infty e^{-x} L_m(x) L_n(x)\, dx = \delta_{mn} \tag{62}$$

where $L_n(x)$ is the n-th degree Laguerre polynomial defined by

$$L_n(x) = \frac{e^x}{n!} \frac{d^n}{dx^n}(x^n e^{-x}) \tag{63}$$

for non-negative integer values of n (2:28). The change of variables mapping x to $\frac{x}{x_0}$ as performed on Equation 62 results in

$$\int_0^\infty \frac{1}{x_0} \exp\left(-\frac{x}{x_0}\right) L_m\left(\frac{x}{x_0}\right) L_n\left(\frac{x}{x_0}\right) dx = \delta_{mn} \tag{64}$$

which implies that a suitable $\psi_n(x)$ satisfying Equation 12 is

$$\psi_n(x) = L_n\left(\frac{x}{x_0}\right) \tag{65}$$

or

$$\begin{aligned} \psi_n(x) &= \frac{1}{n!} \exp\left(\frac{x}{x_0}\right) \frac{d^n}{d\left(\frac{x}{x_0}\right)^n} \left[\left(\frac{x}{x_0}\right)^n \exp\left(-\frac{x}{x_0}\right)\right] \\ &= \frac{x_0^n}{n!} \exp\left(\frac{x}{x_0}\right) \frac{d^n}{dx^n} \left[\left(\frac{x}{x_0}\right)^n \exp\left(-\frac{x}{x_0}\right)\right] \end{aligned} \tag{66}$$

Inserting the expressions given as Equations 59 and 66 into Equation 61 provides the following for $n = 1, 2, 3, \ldots$:

$$\begin{aligned} c_n &= \left\{ \frac{x_0^{n-1} g(x)}{n!} \frac{d^{n-1}}{dx^{n-1}} \left[\left(\frac{x}{x_0} \right)^n \exp\left(-\frac{x}{x_0} \right) \right] \right\} \Bigg|_0^\infty \\ &\quad - \sum_{i=1}^{N} \frac{q x_0^{n-1}}{n!} \frac{d^{n-1}}{dx^{n-1}} \left[\left(\frac{x}{x_0} \right)^n \exp\left(-\frac{x}{x_0} \right) \right] \Bigg|_{x=iq} \end{aligned} \tag{67}$$

Now, noting that

$$\begin{aligned} &\frac{d^n}{dx^n} \left[\left(\frac{x}{x_0} \right)^n \exp\left(-\frac{x}{x_0} \right) \right] = \\ &\quad \frac{1}{x_0} \left\{ n \frac{d^{n-1}}{dx^{n-1}} \left[\left(\frac{x}{x_0} \right)^{n-1} \exp\left(-\frac{x}{x_0} \right) \right] - \frac{d^{n-1}}{dx^{n-1}} \left[\left(\frac{x}{x_0} \right)^n \exp\left(-\frac{x}{x_0} \right) \right] \right\} \end{aligned} \tag{68}$$

provides the identity

$$\begin{aligned} &\frac{d^{n-1}}{dx^{n-1}} \left[\left(\frac{x}{x_0} \right)^n \exp\left(-\frac{x}{x_0} \right) \right] = \\ &\quad n \frac{d^{n-1}}{dx^{n-1}} \left[\left(\frac{x}{x_0} \right)^{n-1} \exp\left(-\frac{x}{x_0} \right) \right] - x_0 \frac{d^n}{dx^n} \left[\left(\frac{x}{x_0} \right)^n \exp\left(-\frac{x}{x_0} \right) \right] \end{aligned} \tag{69}$$

Inserting this identity into Equation 67 provides the following for $n = 1, 2, 3, \ldots$:

$$\begin{aligned} c_n &= \left\langle \frac{g(x)}{n!} \left\{ n \frac{d^{n-1}}{d\left(\frac{x}{x_0}\right)^{n-1}} \left[\left(\frac{x}{x_0} \right)^{n-1} \exp\left(-\frac{x}{x_0} \right) \right] \right. \right. \\ &\quad \left. \left. - \frac{d^n}{d\left(\frac{x}{x_0}\right)^n} \left[\left(\frac{x}{x_0} \right)^n \exp\left(-\frac{x}{x_0} \right) \right] \right\} \right\rangle \Bigg|_0^\infty \\ &\quad - \left\langle \sum_{i=1}^{N} \frac{q}{n!} \left\{ n \frac{d^{n-1}}{d\left(\frac{x}{x_0}\right)^{n-1}} \left[\left(\frac{x}{x_0} \right)^{n-1} \exp\left(-\frac{x}{x_0} \right) \right] \right. \right. \\ &\quad \left. \left. - \frac{d^n}{d\left(\frac{x}{x_0}\right)^n} \left[\left(\frac{x}{x_0} \right)^n \exp\left(-\frac{x}{x_0} \right) \right] \right\} \right\rangle \Bigg|_{x=iq} \\ &= \sum_{i=1}^{N} q \exp\left(-\frac{iq}{x_0} \right) \left[L_n\left(\frac{iq}{x_0} \right) - L_{n-1}\left(\frac{iq}{x_0} \right) \right] \end{aligned}$$

$$- \left\{ g(x) \exp\left(-\frac{x}{x_0}\right) \left[L_n\left(\frac{x}{x_0}\right) - L_{n-1}\left(\frac{x}{x_0}\right) \right] \right\} \Big|_0^\infty$$

$$= \sum_{i=0}^{N} \eta_i q \exp\left(-\frac{iq}{x_0}\right) \left[L_n\left(\frac{iq}{x_0}\right) - L_{n-1}\left(\frac{iq}{x_0}\right) \right] \tag{70}$$

where

$$\eta_i = \begin{cases} \frac{g(x=0)}{q} & i = 0 \\ 1 & i = 1, 2, 3, \ldots \end{cases} \tag{71}$$

The $n = 0$ case must be considered separately. For $n = 0$, Equation 14 becomes

$$c_0 = \int_0^\infty g(x) W(x)\, dx \tag{72}$$

since $\psi_0(x) = 1$. Therefore,

$$\begin{aligned} c_0 &= \sum_{i=0}^{N} \int_{iq}^{(i+1)q} [g(x=0) + iq] \frac{1}{x_0} \exp\left(-\frac{x}{x_0}\right) dx \\ &\quad + \int_{(N+1)q}^{\infty} [g(x=0) + (N+1)q] \frac{1}{x_0} \exp\left(-\frac{x}{x_0}\right) dx \\ &= \sum_{i=0}^{N} [g(x=0) + iq] \left[\exp\left(-\frac{iq}{x_0}\right) - \exp\left(-\frac{(i+1)q}{x_0}\right) \right] \\ &\quad + [g(x=0) + (N+1)q] \exp\left(-\frac{(N+1)q}{x_0}\right) dx \end{aligned} \tag{73}$$

Expanding and combining terms yields

$$c_0 = \sum_{i=0}^{N} \eta_i q \exp\left(-\frac{iq}{x_0}\right) \tag{74}$$

Therefore, Equation 70 also applies to the $n = 0$ case if the understanding is made that the $L_{n-1}(x)$ terms vanish when $n = 0$.

It is now possible to express the nonlinear function, $g(x)$, in the series representation

$$g(x) = \sum_{k=0}^{\infty} qL_k\left(\frac{x}{x_0}\right) \sum_{i=0}^{N} \eta_i \exp\left(-\frac{iq}{x_0}\right) \left[L_k\left(\frac{iq}{x_0}\right) - L_{k-1}\left(\frac{iq}{x_0}\right)\right] \tag{75}$$

Similar to the earlier Gaussian analysis, the next step is to apply the expression for the nonlinearity to the equation for the autocorrelation of the quantizer output, $R_Y(\tau)$. But also as in the Gaussian analysis, $W(x_1, x_2; \tau)$ must be expanded in order to take advantage of the orthogonality of the polynomials used. Barrett and Lampard (2:28) showed that $W(x_1, x_2; \tau)$ can be expanded into the following form:

$$W(x_1, x_2; \tau) = \frac{1}{x_0^2} \exp\left(-\frac{x_1 + x_2}{x_0}\right) \sum_{m=0}^{\infty} \mu^{2m}(\tau) L_m\left(\frac{x_1}{x_0}\right) L_n\left(\frac{x_2}{x_0}\right) \tag{76}$$

By utilizing Equations 75 and 76, and by rearranging the orders of summation and integration, Equation 6 now becomes

$$\begin{aligned} R_Y(\tau) \;=\; & q^2 \sum_{i=0}^{N} \sum_{j=0}^{N} \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} \sum_{m=0}^{\infty} \left\{ \eta_i \eta_j \exp\left(-\frac{iq + jq}{x_0}\right) \mu^{2m}(\tau) \right. \\ & \bullet \left[L_k\left(\frac{iq}{x_0}\right) - L_{k-1}\left(\frac{iq}{x_0}\right)\right] \left[L_l\left(\frac{jq}{x_0}\right) - L_{l-1}\left(\frac{jq}{x_0}\right)\right] \\ & \bullet \left[\int_0^{\infty} \frac{1}{x_0} \exp\left(-\frac{x_1}{x_0}\right) L_k\left(\frac{x_1}{x_0}\right) L_m\left(\frac{x_1}{x_0}\right) dx_1\right] \\ & \left. \bullet \left[\int_0^{\infty} \frac{1}{x_0} \exp\left(-\frac{x_2}{x_0}\right) L_l\left(\frac{x_2}{x_0}\right) L_m\left(\frac{x_2}{x_0}\right) dx_2\right] \right\} \end{aligned} \tag{77}$$

The orthogonality principle given by Equation 64 allows the simplification of the previous expression to

$$\begin{aligned} R_Y(\tau) \;=\; & q^2 \sum_{i=0}^{N} \sum_{j=0}^{N} \sum_{k=0}^{\infty} \left\{ \eta_i \eta_j \exp\left(\frac{iq + jq}{x_0}\right) \mu^{2k}(\tau) \right. \\ & \left. \bullet \left[L_k\left(\frac{iq}{x_0}\right) - L_{k-1}\left(\frac{iq}{x_0}\right)\right] \left[L_k\left(\frac{jq}{x_0}\right) - L_{k-1}\left(\frac{jq}{x_0}\right)\right] \right\} \end{aligned}$$

$$= q^2 \sum_{k=0}^{\infty} \mu^{2k}(\tau) \left\{ \sum_{i=0}^{N} \eta_i \exp\left(-\frac{iq}{x_0}\right) \left[L_k\left(\frac{iq}{x_0}\right) - L_{k-1}\left(\frac{iq}{x_0}\right) \right] \right\}^2 \quad (78)$$

which takes the form

$$R_Y(\tau) = \sum_{k=0}^{\infty} a_k \mu^{2k}(\tau) \quad (79)$$

where

$$a_k = q^2 \left\{ \sum_{i=0}^{N} \eta_i \exp\left(-\frac{iq}{x_0}\right) \left[L_k\left(\frac{iq}{x_0}\right) - L_{k-1}\left(\frac{iq}{x_0}\right) \right] \right\}^2 \quad (80)$$

Now, the other term in the error autocorrelation expression, Equation 10, may be approached. First, the constant, c, may be evaluated by performing the integral given in Equation 8. For the applicable probability density function, Equation 8 becomes

$$c = \int_0^{\infty} g(x) \left[\frac{1}{x_0} \exp\left(-\frac{x}{x_0}\right) \right] \left(\frac{x - x_0}{\sigma_x^2} \right) dx \quad (81)$$

where $\sigma_x^2 = x_0^2$. Ignoring the form of $g(x)$ derived as Equation 75 and performing the integration given as Equation 81 as a finite sum of integrals over intervals where $g(x)$ is continuous yields

$$\begin{aligned} c &= \sum_{n=0}^{N} \left[\int_{nq}^{(n+1)q} (n + \eta_0)\, q \left(\frac{1}{x_0}\right) \exp\left(-\frac{x}{x_0}\right) \left(\frac{x - x_0}{x_0^2}\right) dx \right] \\ &\quad + \int_{(N+1)q}^{\infty} (N + \eta_0)\, q \frac{1}{x_0} \exp\left(-\frac{x}{x_0}\right) \left(\frac{x - x_0}{x_0^2}\right) dx \\ &= \sum_{n=0}^{N} \left\{ (n + \eta_0) \frac{q^2}{x_0^2} \left[n \exp\left(-\frac{nq}{x_0}\right) - (n+1) \exp\left(-\frac{(n+1)q}{x_0}\right) \right] \right\} \\ &\quad + (N + \eta_0) \frac{q^2}{x_0^2} (N+1) \exp\left(-\frac{(N+1)q}{x_0}\right) \end{aligned} \quad (82)$$

Expanding the summation and collecting terms provides the following expression:

$$c = \frac{q^2}{x_0^2} \sum_{n=1}^{N} n \exp\left(-\frac{nq}{x_0}\right) \quad (83)$$

The autocorrelation of the quantization error can now be expressed in the following form:

$$R_{E_q}(\tau) = \sum_{k=0}^{\infty} a_k \mu^{2k}(\tau) + (1-2c)R_X(\tau) \tag{84}$$

which introduces a new problem. The relationship between $R_X(\tau)$ and $\mu(\tau)$ remains to be determined. To determine this relationship, the following integral may be considered:

$$\begin{aligned} R_X(\tau) &= \int_0^{\infty}\int_0^{\infty} x_1 x_2 W(x_1, x_2; \tau)\, dx_1\, dx_2 \\ &= \int_0^{\infty}\int_0^{\infty} \frac{x_1 x_2}{x_0^2[1-\mu^2(\tau)]} I_0\left[\frac{2\sqrt{x_1 x_2}}{x_0}\frac{\mu(\tau)}{1-\mu^2(\tau)}\right] \\ &\quad \bullet \exp\left(-\frac{x_1+x_2}{x_0[1-\mu^2(\tau)]}\right) dx_1\, dx_2 \end{aligned} \tag{85}$$

The reduction of this integral to a simple function of $\mu(\tau)$ is quite involved and appears in Section A.1 of this thesis. The resulting relationship is as follows:

$$R_X(\tau) = x_0^2[1+\mu^2(\tau)] \tag{86}$$

Incorporating Equations 80, 83 and 86 into Equation 84 allows the expression of the autocorrelation of the quantization error as a function of $\mu(\tau)$ in the following manner:

$$\begin{aligned} R_{E_q}(\tau) &= q^2 \sum_{k=0}^{\infty} \mu^{2k}(\tau) \left\{ \sum_{i=0}^{N} \eta_i \exp\left(-\frac{iq}{x_0}\right)\left[L_k\left(\frac{iq}{x_0}\right) - L_{k-1}\left(\frac{iq}{x_0}\right)\right]\right\}^2 \\ &\quad + \left[x_0^2 - 2q^2 \sum_{n=1}^{N} n \exp\left(-\frac{nq}{x_0}\right)\right][1+\mu^2(\tau)] \end{aligned} \tag{87}$$

where, as before, $N = 2^Q - 1$.

3.4.2 The Determination of Some Noise Related Figures of Merit for a Negative-Exponential Input

3.4.2.1 The Normalized Noise Power From the expression provided above as Equation 87, the determination of an expression for the normalized noise power can be made. This normalized noise power, or the mean-squared error, is the autocorrelation of the quantization error evaluated at a time differential of zero.

The determination of $\mu^2(\tau)$ at a time differential of zero is accomplished as follows:

$$\begin{aligned} \mu^2(\tau = 0) &= \frac{R_X(\tau = 0)}{x_0^2} - 1 \\ &= \frac{E(x^2)}{x_0^2} - 1 \\ &= 1 \end{aligned} \tag{88}$$

Therefore, the normalized noise power can be expressed as

$$\begin{aligned} N_q &= q^2 \sum_{k=0}^{\infty} \left\{ \sum_{i=0}^{N} \eta_i \exp\left(-\frac{iq}{x_0}\right) \left[L_k\left(\frac{iq}{x_0}\right) - L_{k-1}\left(\frac{iq}{x_0}\right) \right] \right\}^2 \\ &\quad + 2\left[x_0^2 - 2q^2 \sum_{n=1}^{N} n \exp\left(-\frac{nq}{x_0}\right) \right] \end{aligned} \tag{89}$$

3.4.2.2 The Signal-to-Quantization Noise Ratio With N_q given in Equation 89, the only remaining entity required to produce a signal-to-quantization noise ratio is the normalized signal power. This quantity is determined as follows:

$$\begin{aligned} S &= R_X(\tau = 0) \\ &= E(x^2) \\ &= 2x_0^2 \end{aligned} \tag{90}$$

Creating the desired ratio then yields

$$\frac{S}{N_q} = \left\langle \frac{q^2}{2x_0^2} \sum_{k=0}^{\infty} \left\{ \sum_{i=0}^{N} \eta_i \exp\left(-\frac{iq}{x_0}\right) \left[L_k\left(\frac{iq}{x_0}\right) - L_{k-1}\left(\frac{iq}{x_0}\right) \right] \right\}^2 + 1 - \frac{2q^2}{x_0^2} \sum_{n=1}^{N} n \exp\left(-\frac{nq}{x_0}\right) \right\rangle^{-1} \tag{91}$$

3.4.3 The Determination of the Quantization Noise Spectrum for a Negative-Exponential Input Now that the autocorrelation of the quantization error is available as Equation 87, the Wiener-Khinchine relationship can be applied in a similar manner as for the case of the Gaussian input. Applying this relationship and recognizing once again that the Fourier transform operation is a linear operation yields the following expression:

$$G_{E_q}(f) = q^2 \sum_{k=0}^{\infty} \mathcal{F}[\mu^{2k}(\tau)] \left\{ \sum_{i=0}^{N} \eta_i \exp\left(\frac{iq}{x_0}\right) \left[L_k\left(\frac{iq}{x_0}\right) - L_{k-1}\left(\frac{iq}{x_0}\right) \right] \right\}^2 + \left[1 - \frac{2q^2}{x_0^2} \sum_{n=1}^{N} n \exp\left(-\frac{nq}{x_0}\right) \right] \mathcal{F}[R_X(\tau)] \tag{92}$$

which takes the form

$$G_{E_q}(f) = \sum_{k=0}^{\infty} a_k \mathcal{F}[\mu^{2k}(\tau)] + b\mathcal{F}[R_X(\tau)] \tag{93}$$

where a_k is as defined in Equation 80, and

$$b = 1 - \frac{2q^2}{x_0^2} \sum_{n=1}^{N} n \exp\left(-\frac{nq}{x_0}\right) \tag{94}$$

If an identical approach is taken regarding Equation 79, the following relationship results for the output spectrum:

$$G_Y(f) = \sum_{k=0}^{\infty} a_k \mathcal{F}[\mu^{2k}(\tau)] \tag{95}$$

As with the case of the Gaussian input, the evaluation of the $\mathcal{F}[\mu^{2k}(\tau)]$ term as k increases without bound is perhaps impossible for the general $\mu^2(\tau)$. A manageable $\mu^2(\tau)$ which satisfies Equation 88 is

$$\mu^2(\tau) = \exp(-\alpha|\tau|) \tag{96}$$

which is identical to the input autocorrelation function considered for the Gaussian input. The corresponding input autocorrelation function is

$$R_X(\tau) = x_0^2[1 + \exp(-\alpha|\tau|)] \tag{97}$$

Applying these functions to the error power spectral density equation results in

$$\begin{aligned} G_{E_q}(f) &= \sum_{k=0}^{\infty} a_k \mathcal{F}[\exp(-\alpha k|\tau|)] + b\mathcal{F}\{x_0^2[1 + \exp(-\alpha|\tau|)]\} \\ &= \sum_{k=1}^{\infty} a_k \left[\frac{2\alpha k}{\alpha^2 k^2 + (2\pi f)^2}\right] + bx_0^2 \left[\frac{2\alpha}{\alpha^2 + (2\pi f)^2}\right] \\ &\quad + (a_0 + bx_0^2)\delta(f) \end{aligned} \tag{98}$$

Similarly,

$$G_Y(f) = a_0\delta(f) + \sum_{k=1}^{\infty} a_k \left[\frac{2\alpha k}{\alpha^2 k^2 + (2\pi f)^2}\right] \tag{99}$$

It should be noted that for the general case,

$$G_{E_q}(f) = \sum_{k=1}^{\infty} a_k \mathcal{F}[\mu^{2k}(\tau)] + bx_0^2 \mathcal{F}[\mu^2(\tau)] + (a_0 + bx_0^2)\delta(f) \tag{100}$$

Since the value of b is usually negative and tends to approach -1 as the number of quantization levels increases, care must be taken to ensure that the value $a_0 + bx_0^2$ provides a valid power spectral density quantity. In other words, the condition

$$a_0 + bx_0^2 \geq 0 \tag{101}$$

must be met.

Using Equation 80,

$$a_0 = q^2 \left[\sum_{i=0}^{N} \eta_i \exp\left(-\frac{iq}{x_0}\right) \right]^2 \tag{102}$$

Using this identity and inserting Equation 94 into the required condition yields the following requirement:

$$g(x=0) \geq \left[2q^2 \sum_{n=1}^{N} n \exp\left(-\frac{nq}{x_0}\right) - x_0^2 \right]^{\frac{1}{2}} - q \sum_{i=1}^{N} \exp\left(-\frac{iq}{x_0}\right) \tag{103}$$

Since the choice of a $g(x=0)$ value directly affects the a_0 term, the quantization noise can be minimized if the $g(x=0)$ value is chosen to satisfy Equation 103 at equality. Consequently, the identity

$$a_0 = -bx_0^2 \tag{104}$$

can be assumed, unless Equation 103 at equality provides either a complex or a negative value for $g(x=0)$. If this becomes the case, $g(x=0)$ can be chosen as a zero value, and hence

$$a_0 = q^2 \left[\sum_{i=1}^{N} \exp\left(-\frac{iq}{x_0}\right) \right]^2 \tag{105}$$

3.5 The Sinusoidal Case with Random Phase

3.5.1 The Noise Autocorrelation Problem for a Sinusoidal Input with Random Phase If the input to the quantizer consists of a signal which possesses the characteristics of a sine wave with constant amplitude and frequency but with a uniformly random phase, it can be modeled as the following random process:

$$X(t) = A\cos(\omega_0 t + \Phi) \tag{106}$$

where the probability density function pertaining to the random phase is

$$W(\phi) = \begin{cases} \frac{1}{2\pi} & -\pi < \phi \leq \pi \\ 0 & \text{elsewhere} \end{cases} \tag{107}$$

Through the utilization of the corresponding characteristic function, Barrett and Lampard (2:28) showed that the resulting second order probability density function for the signal level of the quantizer input is

$$\begin{aligned} W(x_1, x_2; \tau) &= \frac{1}{\pi^2} \left[\frac{1}{(A^2 - x_1^2)^{\frac{1}{2}}} \right] \left[\frac{1}{(A^2 - x_2^2)^{\frac{1}{2}}} \right] \\ &\quad \cdot \sum_{m=0}^{\infty} \epsilon_m T_m \left(\frac{x_1}{A} \right) T_m \left(\frac{x_2}{A} \right) \cos(m\omega_0 \tau) \end{aligned} \tag{108}$$

for $|x_1| < A$ and $|x_2| < A$, where

$$\epsilon_m = \begin{cases} 1 & m = 0 \\ 2 & m = 1, 2, 3, \ldots \end{cases} \tag{109}$$

and $T_m(x)$ is the Tchebycheff polynomial of the first kind, defined by

$$T_m(x) = \cos[m \arccos(x)] \tag{110}$$

The resulting first order probability density function is

$$W(x) = \begin{cases} \frac{1}{\pi(A^2 - x^2)^{\frac{1}{2}}} & |x| < A \\ 0 & \text{elsewhere} \end{cases} \tag{111}$$

The quantization operation to be acted upon this input is identical to the operation used upon the Gaussian input. However, now the level of the input signal is constrained to an absolute value less than A, whereas the Gaussian and negative-exponential inputs were allowed to approach unbounded levels.

As for the general case, the autocorrelation function for the quantization noise will follow the relationship described in Equation 10, with new and appropriate limits placed on the integral. Equations 11 through 21 apply in a similar manner.

The orthogonality property for the Tchebycheff polynomial is

$$\frac{1}{\pi}\int_{-1}^{1} \epsilon_n T_m(x) T_n(x) \left(1-x^2\right)^{-\frac{1}{2}}\, dx = \delta_{mn} \tag{112}$$

(2:28). The change of variables mapping x to $\frac{x}{A}$ results in

$$\int_{-A}^{A} \epsilon_n T_m\left(\frac{x}{A}\right) T_n\left(\frac{x}{A}\right) \left[\frac{1}{\pi\left(A^2-x^2\right)^{\frac{1}{2}}}\right] dx = \delta_{mn} \tag{113}$$

which implies that a suitable $\psi_n(x)$ satisfying Equation 12 is

$$\psi_n(x) = \sqrt{\epsilon_n} T_n\left(\frac{x}{A}\right) \tag{114}$$

or

$$\psi_n(x) = \sqrt{\epsilon_n} \cos\left[n \arccos\left(\frac{x}{A}\right)\right] \tag{115}$$

Inserting the expressions given as Equations 111 and 115 into Equation 21 provides the following:

$$c_n = g(x)\sqrt{\epsilon_n} F_n(x)\Big|_{-A}^{A} - \sum_{i=-M}^{M} q\sqrt{\epsilon_n} F_n(x)\Big|_{x=iq} \tag{116}$$

where

$$\frac{d}{dx}F_n(x) = \left[\frac{1}{\pi\left(A^2-x^2\right)^{\frac{1}{2}}}\right] \cos\left[n \arccos\left(\frac{x}{A}\right)\right] \tag{117}$$

Some modification to this equation reveals that

$$c_n = \sum_{i=-M}^{M} \frac{q\sqrt{\epsilon_n}}{n\pi} K_n(x)\Big|_{x=iq} - \frac{g(x)\sqrt{\epsilon_n}}{n\pi} K_n(x)\Big|_{-A}^{A} \tag{118}$$

where

$$\frac{d}{dx}K_n(x) = \left[-\frac{n}{(A^2 - x^2)^{\frac{1}{2}}}\right] \cos\left[n \arccos\left(\frac{x}{A}\right)\right] \tag{119}$$

Still further simplification yields

$$\begin{aligned} c_n &= \sum_{i=-M}^{M} \frac{q\sqrt{\epsilon_n}}{n\pi} \sin\left[n \arccos\left(\frac{iq}{A}\right)\right] \\ &\quad - \left\{\frac{g(x)\sqrt{\epsilon_n}}{n\pi} \sin\left[n \arccos\left(\frac{x}{A}\right)\right]\right\}\Bigg|_{-A}^{A} \\ &= \frac{q\sqrt{\epsilon_n}}{n\pi} \sum_{i=-M}^{M} U_n\left(\frac{iq}{A}\right) \end{aligned} \tag{120}$$

where $U_n(x)$ is the Tchebycheff polynomial of the second kind, defined by

$$U_n(x) = \sin[n \arccos(x)] \tag{121}$$

It is interesting to note that

$$\begin{aligned} U_n(-x) &= \sin[n \arccos(-x)] \\ &= \sin\{n[\pi - \arccos(x)]\} \\ &= \sin(n\pi)T_n(x) - \cos(n\pi)U_n(x) \\ &= (-1)^{n+1} U_n(x) \end{aligned} \tag{122}$$

Applying this property to Equation 120 provides

$$c_n = \begin{cases} \frac{q\sqrt{\epsilon_n}}{n\pi} \sum_{i=-M}^{M} U_n\left(\frac{iq}{A}\right) & \text{n, odd} \\ 0 & \text{n, even} \end{cases} \tag{123}$$

It is now possible to express the nonlinear function, $g(x)$, in the series representation

$$g(x) = \frac{2q}{\pi}\sum_{k=0}^{\infty}\sum_{i=-M}^{M}\frac{1}{2k+1}U_{2k+1}\left(\frac{iq}{A}\right)T_{2k+1}\left(\frac{x}{A}\right) \tag{124}$$

where, as in the Gaussian case, $M = 2^{Q-1} - 1$.

By utilizing the expressions given as Equations 108 and 124, and by rearranging the orders of summation and integration, Equation 6 becomes

$$\begin{aligned} R_Y(\tau) = \frac{4q^2}{\pi^2}\sum_{i=-M}^{M}\sum_{j=-M}^{M}\sum_{k=0}^{\infty}\sum_{l=0}^{\infty}\sum_{m=0}^{\infty}\Bigg\{ & \frac{\cos(m\omega_0\tau)}{\epsilon_m(2k+1)(2l+1)} \\ & \bullet U_{2k+1}\left(\frac{iq}{A}\right)U_{2l+1}\left(\frac{jq}{A}\right) \\ & \bullet\left[\int_{-A}^{A}\epsilon_m T_{2k+1}\left(\frac{x_1}{A}\right)T_m\left(\frac{x_1}{A}\right)\left(\frac{1}{\pi\left(A^2-x_1^2\right)^{\frac{1}{2}}}\right)dx_1\right] \\ & \bullet\left[\int_{-A}^{A}\epsilon_m T_{2l+1}\left(\frac{x_2}{A}\right)T_m\left(\frac{x_2}{A}\right)\left(\frac{1}{\pi\left(A^2-x_2^2\right)^{\frac{1}{2}}}\right)dx_2\right]\Bigg\} \end{aligned} \tag{125}$$

Utilizing the orthogonality property given by Equation 113 allows the simplification of Equation 125 to

$$\begin{aligned} R_Y(\tau) &= \frac{2q^2}{\pi^2}\sum_{i=-M}^{M}\sum_{j=-M}^{M}\sum_{k=0}^{\infty}\left\{\frac{\cos[(2k+1)\omega_0\tau]}{(2k+1)^2}\right. \\ &\quad \left.\bullet\, U_{2k+1}\left(\frac{iq}{A}\right)U_{2k+1}\left(\frac{jq}{A}\right)\right\} \\ &= \frac{2q^2}{\pi^2}\sum_{k=0}^{\infty}\left\{\frac{\cos[(2k+1)\omega_0\tau]}{(2k+1)^2}\left[\sum_{i=-M}^{M}U_{2k+1}\left(\frac{iq}{A}\right)\right]^2\right\} \end{aligned} \tag{126}$$

which takes the form

$$R_Y(\tau) = \sum_{k=0}^{\infty}a_k\cos[(2k+1)\omega_0\tau] \tag{127}$$

where

$$a_k = \frac{2q^2}{\pi^2}\frac{1}{(2k+1)^2}\left[\sum_{i=-M}^{M} U_{2k+1}\left(\frac{iq}{A}\right)\right]^2 \tag{128}$$

Now, the second term in Equation 10 must be attacked by first evaluating the integral given by Equation 8. For the case at hand, Equation 8 becomes

$$c = \int_{-A}^{A} g(x)\left[\frac{1}{\pi\left(A^2-x^2\right)^{\frac{1}{2}}}\right]\frac{x}{\sigma^2}\,dx \tag{129}$$

where σ^2 is the normalized power of the input, which for a sinusoid with amplitude A is $\frac{A^2}{2}$. Noting that the integrand is an even function of x, allows the use of the following equation:

$$c = \frac{4}{A^2}\int_{0}^{A} g(x)\left[\frac{x}{\pi\left(A^2-x^2\right)^{\frac{1}{2}}}\right]dx \tag{130}$$

Ignoring the form of $g(x)$ derived as Equation 124 and performing the integration given as Equation 130 as a finite sum of integrals over intervals where $g(x)$ is continuous yields

$$\begin{aligned} c &= \frac{4}{A^2}\left\{\sum_{m=0}^{M}\left[\int_{mq}^{(m+1)q}\left(m+\tfrac{1}{2}\right)q\left(\frac{x}{\pi\left(A^2-x^2\right)^{\frac{1}{2}}}\right)dx\right]\right. \\ &\quad \left.+\int_{(M+1)q}^{A}\left(M+\frac{1}{2}\right)q\left(\frac{x}{\pi\left(A^2-x^2\right)^{\frac{1}{2}}}\right)dx\right\} \\ &= \frac{4q}{\pi A^2}\left\{\sum_{m=0}^{M}\left[\left(m+\frac{1}{2}\right)\left([A^2-(mq)^2]^{\frac{1}{2}}-[A^2-(m+1)^2q^2]^{\frac{1}{2}}\right)\right]\right. \\ &\quad \left.+\left(M+\frac{1}{2}\right)\left([A^2-(M+1)^2q^2]^{\frac{1}{2}}\right)\right\} \end{aligned} \tag{131}$$

Expanding the summation and collecting terms provides the following expression:

$$c = \frac{4q}{\pi A^2}\left[\frac{A}{2}+\sum_{m=1}^{M}\left(A^2-m^2q^2\right)^{\frac{1}{2}}\right] \tag{132}$$

The autocorrelation of the quantization error can now be expressed in the following form:

$$R_{E_q}(\tau) = \sum_{k=0}^{\infty} a_k cos[(2k+1)\omega_0\tau] + (1-2c)R_X(\tau) \tag{133}$$

which introduces another new problem. The relationship between $R_X(\tau)$ and $\cos(\omega_0\tau)$ remains to be determined. To determine this relationship, the following expression must be considered:

$$\begin{aligned} R_X(\tau) &= \int_{-A}^{A}\int_{-A}^{A} x_1 x_2 W(x_1, x_2; \tau)\, dx_1\, dx_2 \\ &= \sum_{m=0}^{\infty}\left\{\frac{\epsilon_m}{\pi^2}\cos(m\omega_0\tau)\left[\int_{-A}^{A}\frac{x_1}{(A^2-x_1^2)^{\frac{1}{2}}}T_m\left(\frac{x_1}{A}\right)dx_1\right]\right. \\ &\quad \left.\bullet\left[\int_{-A}^{A}\frac{x_2}{(A^2-x_2^2)^{\frac{1}{2}}}T_m\left(\frac{x_2}{A}\right)dx_2\right]\right\} \\ &= \sum_{m=0}^{\infty}\left\{\frac{\epsilon_m}{\pi^2}\cos(m\omega_0\tau)\left[\int_{-A}^{A}\frac{x}{(A^2-x^2)^{\frac{1}{2}}}T_m\left(\frac{x}{A}\right)dx_1\right]^2\right\} \end{aligned} \tag{134}$$

The reduction of this expression to a simple function of $\cos(m\omega_0\tau)$ is quite involved and therefore appears in Section A.2 of this thesis. The simple function of $\cos(m\omega_0\tau)$ is as follows:

$$R_X(\tau) = \frac{A^2}{2}\cos(\omega_0\tau) \tag{135}$$

Incorporating Equations 126, 132 and 135 into Equation 10 allows the expression of the autocorrelation of the quantization error as a function of $\cos(m\omega_0\tau)$ in the following manner:

$$\begin{aligned} R_{E_q}(\tau) &= \frac{2q^2}{\pi^2}\sum_{k=0}^{\infty}\left\{\frac{\cos[(2k+1)\omega_0\tau]}{(2k+1)^2}\left[\sum_{i=-M}^{M}U_{2k+1}\left(\frac{iq}{A}\right)\right]^2\right\} \\ &\quad + \left\{\frac{A^2}{2} - \frac{4q}{\pi}\left[\frac{A}{2} + \sum_{m=1}^{M}\left(A^2 - m^2q^2\right)^{\frac{1}{2}}\right]\right\}\cos(\omega_0\tau) \end{aligned} \tag{136}$$

where, as before, $M = 2^{Q-1} - 1$.

3.5.2 The Determination of Some Noise Related Figures of Merit for a Sinusoidal Input with Random Phase

3.5.2.1 The Normalized Noise Power From the expression provided above as Equation 136, the determination of an expression for the normalized noise power can be made. As mentioned for the previous two cases, this normalized noise power, or mean-squared error, is the autocorrelation of the quantization error evaluated at a time differential of zero, or

$$\begin{aligned} N_q &= R_{E_q}(\tau = 0) \\ &= \frac{2q^2}{\pi^2} \sum_{k=0}^{\infty} \left\{ \frac{1}{(2k+1)^2} \left[\sum_{i=-M}^{M} U_{2k+1} \left(\frac{iq}{A} \right) \right]^2 \right\} \\ &\quad + \frac{A^2}{2} - \frac{4q}{\pi} \left[\frac{A}{2} + \sum_{m=1}^{M} \left(A^2 - m^2 q^2 \right)^{\frac{1}{2}} \right] \end{aligned} \tag{137}$$

3.5.2.2 The Signal-to-Quantization Noise Ratio Now that Equation 137 has been provided, a signal-to-quantization ratio derivation becomes trivial, since the normalized signal power is

$$S = R_X(\tau = 0) = \frac{A^2}{2} \tag{138}$$

Creating the desired ratio then yields

$$\begin{aligned} \frac{S}{N_q} &= \left\langle \frac{4q^2}{A^2\pi^2} \sum_{k=0}^{\infty} \left\{ \frac{1}{(2k+1)^2} \left[\sum_{i=-M}^{M} U_{2k+1} \left(\frac{iq}{A} \right) \right]^2 \right\} \right. \\ &\quad \left. + 1 - \frac{4q}{\pi} \left[\frac{1}{A} + \frac{2}{A^2} \sum_{m=1}^{M} \left(A^2 - m^2 q^2 \right)^{\frac{1}{2}} \right] \right\rangle^{-1} \end{aligned} \tag{139}$$

3.5.3 The Determination of the Quantization Noise Spectrum for a Sinusoidal Input with Random Phase Now that the autocorrelation of the quantization error

is available as Equation 136, the Wiener-Khinchine relationship can be applied in a similar manner as for the previous two cases. Applying this relationship and recognizing that the Fourier transform operation is a linear operation yields the following expression:

$$\begin{aligned} G_{E_q}(f) &= \frac{2q^2}{\pi^2}\sum_{k=0}^{\infty}\left\{\frac{1}{(2k+1)^2}\left[\sum_{i=-M}^{M} U_{2k+1}\left(\frac{iq}{A}\right)\right]^2 \mathcal{F}\{\cos[2\pi(2k+1)f_0\tau]\}\right\} \\ &\quad + \left\{\frac{A^2}{2} - \frac{4q}{\pi}\left[\frac{A}{2} + \sum_{m=1}^{M}\left(A^2 - m^2q^2\right)^{\frac{1}{2}}\right]\right\}\mathcal{F}\{\cos(2\pi f_0\tau)\} \\ &= \frac{q^2}{\pi^2}\sum_{k=0}^{\infty}\left\{\left[\sum_{i=-M}^{M} U_{2k+1}\left(\frac{iq}{A}\right)\right]^2 \right. \\ &\quad \left. \bullet \frac{1}{(2k+1)^2}\{\delta[f-(2k+1)f_0] + \delta[f+(2k+1)f_0]\}\right\} \\ &\quad + \left\{\frac{A^2}{4} - \frac{2q}{\pi}\left[\frac{A}{2} + \sum_{m=1}^{M}\left(A^2 - m^2q^2\right)^{\frac{1}{2}}\right]\right\} \\ &\quad \bullet[\delta(f-f_0) + \delta(f+f_0)] \end{aligned} \tag{140}$$

which takes the form

$$\begin{aligned} G_{E_q}(f) &= \frac{1}{2}\sum_{k=0}^{\infty} a_k\{\delta[f-(2k+1)f_0] + \delta[f+(2k+1)f_0]\} \\ &\quad + \frac{b}{2}[\delta(f-f_0) + \delta(f+f_0)] \\ &= \frac{1}{2}\sum_{k=1}^{\infty} a_k\{\delta[f-(2k+1)f_0] + \delta[f+(2k+1)f_0]\} \\ &\quad + \frac{1}{2}(a_0+b)[\delta(f-f_0) + \delta(f+f_0)] \end{aligned} \tag{141}$$

where a_k is as defined in Equation 128 and

$$b = \frac{A^2}{2} - \frac{4q}{\pi^2}\left[\frac{A}{2} + \sum_{m=1}^{M}\left(A^2 - m^2q^2\right)^{\frac{1}{2}}\right] \tag{142}$$

If an identical approach is taken regarding Equation 128, the following relationship results for the output spectrum:

$$G_Y(f) = \frac{1}{2}\sum_{k=0}^{\infty} a_k\{\delta[f-(2k+1)f_0] + \delta[f+(2k+1)f_0]\} \tag{143}$$

Note that in order for Equation 141 to represent a valid power spectral density, the condition

$$a_0 + b \geq 0 \tag{144}$$

must be met.

Since

$$\begin{aligned} U_1(x) &= \sin(\arccos x) \\ &= \left(1-x^2\right)^{\frac{1}{2}} \end{aligned} \tag{145}$$

Equation 128 can be used to reveal that

$$a_0 = \frac{8q^2}{\pi^2 A^2}\left[\frac{A}{2} + \sum_{i=1}^{M}\left(A^2 - i^2q^2\right)^{\frac{1}{2}}\right]^2 \tag{146}$$

Making the appropriate substitutions into the condition given as Equation 144 yields

$$\frac{8q^2}{\pi^2 A^2}\left[\frac{A}{2} + \sum_{i=1}^{M}\left(A^2 - i^2q^2\right)^{\frac{1}{2}}\right]^2 - \frac{4q}{\pi}\left[\frac{A}{2} + \sum_{i=1}^{M}\left(A^2 - i^2q^2\right)^{\frac{1}{2}}\right] + \frac{A^2}{2} \geq 0 \tag{147}$$

The application of the quadratic equation reveals that an equivalent condition is

$$\frac{A}{2} + \left(A^2 - i^2q^2\right)^{\frac{1}{2}} \geq \frac{A^2\pi}{4q} \tag{148}$$

or

$$\frac{q}{A} + \frac{2q}{A}\sum_{i=1}^{M}\left(1 - \frac{i^2q^2}{A}\right)^{\frac{1}{2}} \geq \frac{\pi}{2} \tag{149}$$

In order for the preceding expressions regarding the sinusoidal input with random phase to be valid, care must be taken to ensure that this condition on the ratio of q to A is satisfied. Also, a second condition exists which must be satisfied. This condition, namely

$$\frac{q}{A} \leq \frac{1}{M} \tag{150}$$

can be noted by studying many of the previous expressions.

IV. Computer Implementation and Computations

4.1 The Gaussian Case

4.1.1 Approaching the Gaussian Input Equations When approaching the problem of attacking the equations derived for the Gaussian input problem, the first obstacle to overcome is the determination of each required intermodulation coefficient, a_k, by using Equation 41, repeated here as

$$a_k = \frac{q^2}{2\pi}\left(\frac{1}{\sigma^2}\right)^{2k+1}\left[\sum_{i=-M}^{M}\exp\left(-\frac{(iq)^2}{2\sigma^2}\right)\frac{H_{2k}\left(\frac{iq}{\sigma}\right)}{\sqrt{(2k+1)!}}\right]^2 \tag{151}$$

Studying Equation 151 draws particular attention to the ratio of $H_{2k}\left(\frac{iq}{\sigma}\right)$ to $\sqrt{(2k+1)!}$. Both of these terms increase without bound as k increases without bound. Therefore, somehow this ratio must be approached carefully.

Szegö developed an approximation for a related Hermite polynomial (16:194) which is as follows:

$$\frac{\Gamma\left(\frac{n}{2}+1\right)}{\Gamma(n+1)}\exp\left(-\frac{x^2}{2}\right)H_n^s(x) \approx \cos\left[(2n+1)^{\frac{1}{2}}x - n\frac{\pi}{2}\right] + \frac{x^3}{6(2n+1)^{\frac{1}{2}}}\sin\left[(2n+1)^{\frac{1}{2}}x - n\frac{\pi}{2}\right] \tag{152}$$

or, for n being even,

$$H_n^s(x) \approx \frac{n!}{\left(\frac{n}{2}\right)!}\exp\left(\frac{x^2}{2}\right)(-1)^{\frac{n}{2}} \bullet\left\{\cos\left[(2n+1)^{\frac{1}{2}}x\right] + \frac{x^3}{6(2n+1)^{\frac{1}{2}}}\sin\left[(2n+1)^{\frac{1}{2}}x\right]\right\} \tag{153}$$

where

$$H_n^s(x) = (-1)^n e^{x^2}\frac{d^n}{dx^n}e^{-x^2} \tag{154}$$

Relating the approximation given as Equation 153 to the version of the Hermite polynomial used in this thesis reveals that

$$\begin{aligned} H_n(x) &= 2^{-\frac{n}{2}} H_n^s\left(\frac{x}{\sqrt{2}}\right) \\ &\approx \frac{n!}{\left(\frac{n}{2}\right)! 2^{\frac{n}{2}}} \exp\left(\frac{x^2}{4}\right)(-1)^{\frac{n}{2}} \\ &\quad \bullet \left\{ \cos\left[(n+\frac{1}{2})^{\frac{1}{2}}x\right] + \frac{x^3}{24(n+\frac{1}{2})^{\frac{1}{2}}} \sin\left[(n+\frac{1}{2})^{\frac{1}{2}}x\right] \right\} \end{aligned} \tag{155}$$

for n being even.

Returning to the ratio under consideration

$$\frac{H_{2k}(x)}{\sqrt{(2k+1)!}} \approx \frac{(2k)!}{k!2^k} \exp\left(\frac{x^2}{4}\right)(-1)^k C_{2k}(x) \tag{156}$$

where

$$C_{2k}(x) = \cos\left[(2k+\frac{1}{2})^{\frac{1}{2}}x\right] + \frac{x^3}{24(2k+\frac{1}{2})^{\frac{1}{2}}} \sin\left[(2k+\frac{1}{2})^{\frac{1}{2}}x\right] \tag{157}$$

In order to simplify this ratio, Stirling's approximation can be utilized. This approximation is

$$k! \approx e^{-k}k^k(2\pi k)^{\frac{1}{2}} \tag{158}$$

(5:29). It is applicable to large k, and is accurate to within 0.5% for values of k larger than 16. Utilizing this approximation reveals that

$$\begin{aligned} \frac{H_{2k}(x)}{\sqrt{(2k+1)!}} &\approx \frac{\left[e^{-2k}(2k)^{2k}(4\pi k)^{\frac{1}{2}}\right]^{\frac{1}{2}}}{e^{-k}k^k(2\pi k)^{\frac{1}{2}}2^k(2k+1)^{\frac{1}{2}}} \exp\left(\frac{x^2}{4}\right)(-1)^k C_{2k}(x) \\ &\approx \left[\frac{1}{(2k+1)(\pi k)^{\frac{1}{2}}}\right]^{\frac{1}{2}} \exp\left(\frac{x^2}{4}\right)(-1)^k C_{2k}(x) \end{aligned} \tag{159}$$

Now, for values of k larger than 16, the following approximation for a_k can be used:

$$a_k \approx \hat{a}_k = \frac{q^2}{2\pi}\left(\frac{1}{\sigma^2}\right)^{2k+1}\left\langle \sum_{i=-M}^{M}\left[\frac{1}{(2k+1)(\pi k)^{\frac{1}{2}}}\right]^{\frac{1}{2}}\exp\left(\frac{(iq)^2}{4\sigma^2}\right)\right.$$
$$\bullet\left\{\cos\left[\left(2k+\frac{1}{2}\right)^{\frac{1}{2}}\left(\frac{iq}{\sigma}\right)\right]\right.$$
$$\left.\left.+\frac{\left(\frac{iq}{\sigma}\right)^3}{24\left(2k+\frac{1}{2}\right)^{\frac{1}{2}}}\sin\left[\left(2k+\frac{1}{2}\right)^{\frac{1}{2}}\left(\frac{iq}{\sigma}\right)\right]\right\}\right\rangle^2 \qquad (160)$$

For values of k smaller than or equal to 16, the exact expression for a_k, given as Equation 151, can be used. Generating the Hermite polynomials necessary to use the exact equation becomes a simple matter if the following recurrence relationship (13:2402) is recognized:

$$H_{n+1}(x) = xH_n(x) - nH_{n-1}(x) \qquad (161)$$

where $H_0(x) = 1$ and $H_1(x) = x$.

Allowing a'_k to be equivalent to $(\sigma^2)^{2k+1}a_k$, with a_k as calculated using the exact equation, the number of necessary computations for determining a'_k can be reduced by noting the property given as Equation 34. This property allows the modification of Equation 151 to

$$a'_k = \frac{q^2}{2\pi(2k+1)!}\left[H_{2k}(0) + 2\sum_{i=1}^{M}\exp\left(-\frac{(iq)^2}{2\sigma^2}\right)H_{2k}\left(\frac{iq}{\sigma}\right)\right]^2 \qquad (162)$$

Now, if $\hat{a}'_k$ is equivalent to $(\sigma^2)^{2k+1}\hat{a}_k$, with $\hat{a}_k$ as calculated using Equation 160, then the expression used to calculate the normalized noise power for the quantization

noise can now be derived from Equation 47 to be

$$\begin{aligned} N_q &\approx \sum_{k=0}^{16} a'_k + \sum_{k=17}^{R} \hat{a}'_k \\ &\quad + \sigma^2 \left\{ 1 - \sqrt{\frac{2}{\pi}} \frac{q}{\sigma} \left[1 + 2 \sum_{m=1}^{M} \exp\left(-\frac{(mq)^2}{2\sigma^2} \right) \right] \right\} \\ &\approx \sum_{k=0}^{16} a'_k + \sum_{k=17}^{R} \hat{a}'_k + \sigma^2 b \end{aligned} \tag{163}$$

where b is as defined in Equation 53. The upper limit, R, can be determined by repeatedly increasing its value by a factor of 10 until further increases have no substantial effect on the result. For the purposes of providing data for this thesis, the values $R = 10^4$, $R = 10^5$, and $R = 10^6$ were used. Computer memory limitations placed a constraint on the $R = 10^6$ value, although, for each case, this was sufficient for providing a reasonable approximation. Particular attention was paid to the order of the summation. The intermodulation coefficients were summed in the order of decreasing values of k in an attempt to sum smaller values of a_k first and reduce the effects of computer roundoff.

Once N_q is determined, the signal-to-quantization noise ratio is easily obtained by dividing the N_q value into the input signal power, $S = \sigma^2$.

In the process of calculating N_q, it became necessary to determine each a'_k, each $\hat{a}'_k$, and the value b. These same values can be used to provide the power spectral densities of the quantization error and of the quantizer output. For the chosen input autocorrelation function, Equations 55, 57, and 56 reveal the following one-sided power spectral densities:

$$G_X(f) = \frac{4}{\alpha} \left[\frac{1}{1 + \left(2\pi \frac{f}{\alpha} \right)^2} \right] \tag{164}$$

$$G_Y(f) \approx \frac{4}{\alpha} \left[\sum_{k=0}^{R} \frac{(2k+1)a_k}{(2k+1)^2 + \left(2\pi \frac{f}{\alpha} \right)^2} \right] \tag{165}$$

$$G_{E_q}(f) \approx \frac{4}{\alpha}\left[\sum_{k=0}^{R}\frac{(2k+1)a_k}{(2k+1)^2+\left(2\pi\frac{f}{\alpha}\right)^2} + \frac{b}{1+\left(2\pi\frac{f}{\alpha}\right)^2}\right] \tag{166}$$

In order to reduce redundancy, each of the above power spectral densities are one-sided and apply only to $f \geq 0$. Note that these power spectral densities can be plotted in increments of $\frac{1}{\alpha}$ versus a horizontal axis of $\frac{f}{\alpha}$. Therefore, at this point, α does not require further specification. R takes on the same value as that used to determine the final $\hat{a}'_k$ for earlier consideration.

Finally, since for the chosen input autocorrelation function, $R_X(\tau = 0) = \sigma^2 = 1$. This implies that $a_k = a'_k$ and $\hat{a}_k = \hat{a}'_k$. Therefore, all quantities needed to determine the power spectral densities are identical to those used to determine the normalized noise power.

4.1.2 Programming for Gaussian Input Results The computer program subroutines which compute the normalized noise power, signal-to-quantization noise ratio, and each of the relevant power spectral densities pertaining to the chosen input autocorrelation function have been coded into the Fortran 77 computer language and can be found in Section B.1 of this thesis. This code applies to the $R = 10,000$ case only. Only minor modifications are necessary to increase R.

A synopsis of the subroutines which have been used to produce the results given later in this thesis appears in Table 2.

4.1.3 The Gaussian Input Results Once the required subroutines were coded, they were used to generate the normalized noise powers and signal-to-quantization noise ratios for the quantization process ranging from 1 to 8 bits. In order to obtain the normalized noise powers, a unit input standard deviation was assumed. However, the calculated signal-to-quantization noise ratios are valid regardless of the input

Table 2. Subroutines Used for the Gaussian Input Case

Subroutine	Purpose	Called By
nopoga	Calculates the normalized noise power and the signal-to-quantization noise ratio.	User. This subroutine must be provided the quantization step size, the standard deviation of the Gaussian input, and the number of bits used for quantization.
pospga	Determines the power spectral densities of the quantization error, the quantizer output, and the quantizer input for the chosen input autocorrelation function and places them in files named gauerr, gauout, and gauin, respectively.	User. This subroutine must be provided the quantization step size, the standard deviation of the Gaussian input, and the number of bits used for quantization.
dtakga	Determines all values of a'_k and $\hat{a}'_k$ for $k = 0$ through $k = R$. For the case provided in Section B.1, $R = 10,000$.	nopoga or pospga.
clakg1	Calculates the value of a'_k for $k = 0$ through $k = 16$.	dtakga.
geth2k	Determines the configuration of $H_{2k}(x)$ and $H_{2k-1}(x)$ if given $H_{2k-2}(x)$ and $H_{2k-3}(x)$ or if $k = 0$.	clakg1.
evevpl	Evaluates a polynomial possessing nonzero coefficients for only the even powers of the polynomial argument.	clakg1.
facto	Provides the factorial of an integer.	clakg1.
clakg2	Calculates the value of $\hat{a}'_k$ for $k = 17$ through $k = R$. For the case provided in Section B.1, $R = 10,000$.	dtakga.

Table 3. Calculated Noise Related Figures of Merit for a Gaussian Input

Number of Bits	R	Step Size (units)	Normalized Noise Power (units squared)	Signal-to-Quantization Noise Ratio (dB)
1	10^4	1.60	$3.61 \cdot 10^{-1}$	4.42
2	10^4	1.00	$1.17 \cdot 10^{-1}$	9.32
3	10^5	0.590	$3.71 \cdot 10^{-2}$	14.3
4	10^5	0.339	$1.13 \cdot 10^{-2}$	19.5
5	10^6	0.191	$3.46 \cdot 10^{-3}$	24.6
6	10^6	0.106	$1.02 \cdot 10^{-3}$	29.9
7	10^6	0.0586	$2.95 \cdot 10^{-4}$	35.3
8	10^6	0.0313	$8.14 \cdot 10^{-5}$	40.9

standard deviation. The optimal quantization step size was determined to three significant digits by repeated program execution over a simple iterative process. The data obtained, along with the value of R used to obtain the data, appears in Table 3.

The necessary subroutines were also used to determine the input, the output, and the error power spectral densities for the quantization process ranging from the use of 1 to 5 bits. The input power spectral density for the chosen input autocorrelation function appears in Figure 3.

Figure 4 illustrates the trend of the output power spectral density as the number of bits employed increases from 1 to 5. Likewise, Figure 5 illustrates the trend of the error power spectral density as the number of bits used increases in a like manner.

4.2 The Negative-Exponential Case

4.2.1 Approaching the Negative-Exponential Input Equations As was the case with anticipating a Gaussian input, the problem of attacking the negative-exponential input problem begins with approaching the intermodulation coefficients. The equa-

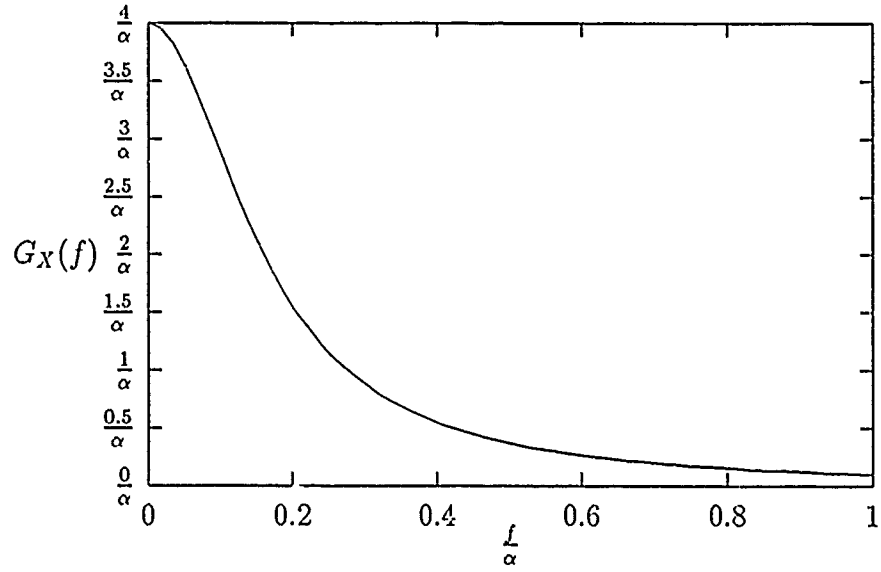

Figure 3. The Power Spectral Density of the Gaussian Input Signal

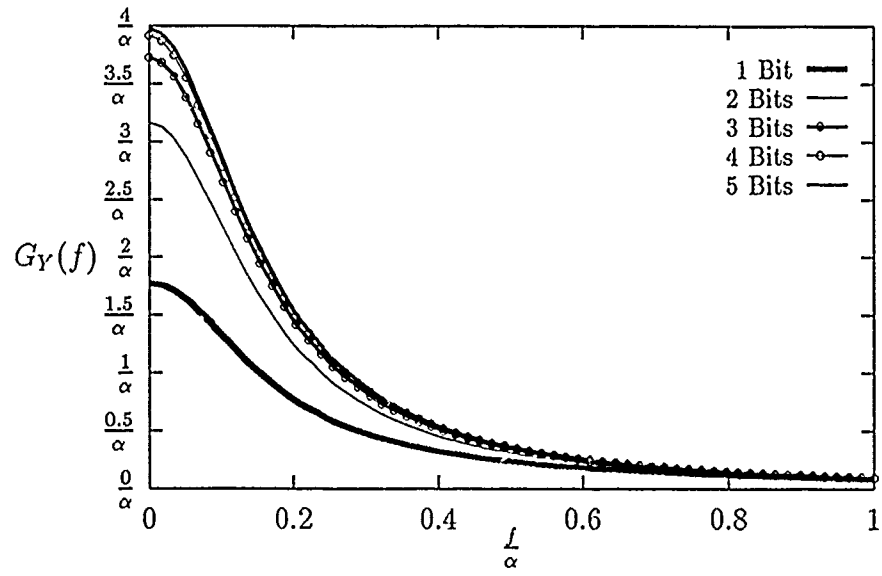

Figure 4. The Power Spectral Density of the Quantizer Output for a Gaussian Input

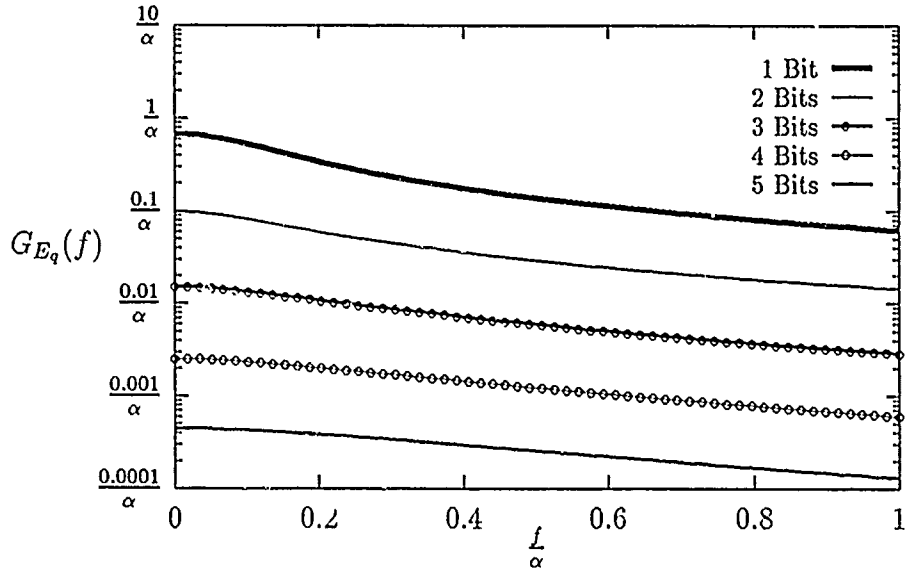

Figure 5. The Power Spectral Density of the Quantization Noise for a Gaussian Input

tion for determining these coefficients appears as Equation 80 and is repeated here as

$$a_k = q^2 \left\{ \sum_{i=0}^{N} \eta_i \exp\left(-\frac{iq}{x_0}\right) \left[L_k\left(\frac{iq}{x_0}\right) - L_{k-1}\left(\frac{iq}{x_0}\right) \right] \right\}^2 \tag{167}$$

Studying this expression draws particular attention to the subtraction operation between the two Laguerre polynomials. Both of these terms become difficult to evaluate as k increases without bound. Therefore, this expression must be approached carefully.

Szegö developed an approximation for the Laguerre polynomial (16:192) which is as follows:

$$L_n(x) \approx \pi^{-\frac{1}{2}} \exp\left(\frac{x}{2}\right) (nx)^{-\frac{1}{4}} \cos\left[2(nx)^{\frac{1}{2}} - \frac{\pi}{4}\right] \tag{168}$$

This approximation becomes increasingly accurate as n becomes large, but the accuracy occurs somewhat more slowly than the approximation for the Hermite polynomial given as Equation 152 (16:192). However, the approximation, which possesses

an error term on the order of $n^{-3/4}$, is sufficiently accurate for values of n such that $n \geq 50$. Therefore, for values of k larger than 50, the following approximation can be used:

$$\begin{aligned} a_k \approx \hat{a}_k &= q^2 \left\langle \sum_{i=1}^{N} \exp\left(-\frac{iq}{x_0}\right) \left\{ \pi^{-\frac{1}{2}} \exp\left(\frac{iq}{2x_0}\right) \left(\frac{kiq}{x_0}\right)^{-\frac{1}{4}} \right.\right. \\ &\quad \bullet \cos\left[2\left(\frac{kiq}{x_0}\right)^{\frac{1}{2}} - \frac{\pi}{4}\right] \\ &\quad - \pi^{-\frac{1}{2}} \exp\left(\frac{iq}{2x_0}\right) \left(\frac{(k-1)iq}{x_0}\right)^{-\frac{1}{4}} \\ &\quad \left.\left. \bullet \cos\left[2\left(\frac{(k-1)iq}{x_0}\right)^{\frac{1}{2}} - \frac{\pi}{4}\right] \right\} \right\rangle^2 \\ &= \frac{q^{\frac{3}{2}} x_0^{\frac{1}{2}}}{\pi} \left\langle \sum_{i=1}^{N} \exp\left(-\frac{iq}{2x_0}\right) \left\{ (ik)^{-\frac{1}{4}} \cos\left[2\left(\frac{kiq}{x_0}\right)^{\frac{1}{2}} - \frac{\pi}{4}\right] \right.\right. \\ &\quad \left.\left. - [i(k-1)]^{-\frac{1}{4}} \cos\left[2\left(\frac{(k-1)iq}{x_0}\right)^{\frac{1}{2}} - \frac{\pi}{4}\right] \right\} \right\rangle^2 \end{aligned} \tag{169}$$

For values of k such that $k \leq 50$, the exact expression of a_k , given as Equation 167, can be used. Generating the Laguerre polynomial difference terms necessary to use the exact equation can be accomplished if the following identity, as provided by Szegö (16:97), is applied:

$$L_k(x) = \sum_{r=0}^{k} \binom{k}{r} \frac{(-x)^r}{r!} \tag{170}$$

Therefore, for $k \geq 1$,

$$\begin{aligned} &L_k(x) - L_{k-1}(x) \\ &\quad = \frac{(-x)^k}{k!} + \sum_{r=0}^{k-1} \left\{ \left[\binom{k}{r} - \binom{k-1}{r} \right] \frac{(-x)^r}{r!} \right\} \end{aligned}$$

$$= \frac{(-x)^k}{k!} + \sum_{r=1}^{k-1} \binom{k-1}{r-1} \frac{(-x)^r}{r!}$$

$$= \sum_{r=1}^{k} \binom{k-1}{r-1} \frac{(-x)^r}{r!} \tag{171}$$

Now, if a_k and $\hat{a}_k$ are determined using the appropriate equations, the expression used to calculate the normalized noise power can be derived from Equation 89 as

$$\begin{aligned} N_q &\approx \sum_{k=0}^{50} a_k + \sum_{k=51}^{R} \hat{a}_k \\ &\quad + 2\left[x_0^2 - 2q^2 \sum_{n=1}^{N} n \exp\left(-\frac{nq}{x_0}\right)\right] \\ &\approx \sum_{k=0}^{50} a_k + \sum_{k=51}^{R} \hat{a}_k + 2bx_0^2 \end{aligned} \tag{172}$$

where b is as defined in Equation 94. The upper limit, R, can be determined by repeatedly increasing its value by a factor of 10 until further increases have no substantial effect on the result. For the purposes of providing data for this thesis, the values $R = 10^4$, $R = 10^5$, and $R = 10^6$ were used. Computer memory limitations placed a constraint on the $R = 10^6$ value, although, for each case, this was sufficient for providing a reasonable approximation. Particular attention was paid to the order of the summation. The intermodulation coefficients were summed in the order of decreasing values of k in an attempt to sum smaller values of a_k first and reduce the effects of computer roundoff.

Recall that before this expression for N_q can be valid, the value $g(x = 0)$ must be chosen to satisfy the condition given as Equation 103. Therefore, before Equation 172 can be applied, $g(x = 0)$ must be determined by applying Equation 103 or by letting $g(x = 0) = 0$ as appropriate. For further details, please refer to the text accompanying Equation 103.

Once N_q is determined, the signal-to-quantization noise ratio is easily obtained by dividing the value N_q into the input signal power, $S = 2x_0^2$.

In the process of calculating N_q, it became necessary to determine an appropriate $g(x = 0)$, each a_k, each $\hat{a}_k$, and the value b. These values can be used to provide the power spectral densities of the quantization error and of the quantizer output. For the chosen input autocorrelation function, Equations 97, 99, and 98 reveal the following one-sided power spectral densities:

$$G_X(f) = \frac{x_0^2}{\alpha}\delta\left(\frac{f}{\alpha}\right) + \frac{4}{\alpha}\left[\frac{1}{1+\left(2\pi\frac{f}{\alpha}\right)^2}\right] \tag{173}$$

$$G_Y(f) \approx \frac{a_0}{\alpha}\delta\left(\frac{f}{\alpha}\right) + \frac{4}{\alpha}\left[\sum_{k=1}^{R}\frac{ka_k}{k^2+\left(2\pi\frac{f}{\alpha}\right)^2}\right] \tag{174}$$

$$G_{E_q}(f) \approx \left(\frac{a_0+bx_0^2}{\alpha}\right)\delta\left(\frac{f}{\alpha}\right) + \frac{4}{\alpha}\left[\sum_{k=1}^{R}\frac{ka_k}{k^2+\left(2\pi\frac{f}{\alpha}\right)^2} + \frac{bx_0^2}{1+\left(2\pi\frac{f}{\alpha}\right)^2}\right] \tag{175}$$

In order to reduce redundancy, each of the above power spectral densities are one-sided and apply only to $f \geq 0$. Also note that as for the case of a Gaussian input, these power spectral densities can be plotted in increments of $\frac{1}{\alpha}$ versus a horizontal axis of $\frac{f}{\alpha}$. Therefore, at this point, α does not require further specification. R takes on the same value as that used to determine the final $\hat{a}_k$ for earlier consideration.

4.2.2 Programming for Negative-Exponential Input Results The computer program subroutines which compute the normalized noise power, signal-to-quantization noise ratio, and each of the relevant power spectral densities pertaining to the chosen input autocorrelation function have been coded into the Fortran 77 computer language and can be found in Section B.2 of this thesis. This code applies to the $R = 10,000$ case only. Only minor modifications are necessary to increase R.

A synopsis of the subroutines which have been used to produce the results given later in this thesis appears in Table 4.

4.2.3 The Negative-Exponential Input Results Once the required subroutines were coded, they were used to generate the normalized noise powers and signal-to-quantization noise ratios for the quantization process ranging from 1 to 8 bits. In order to obtain the normalized noise powers, a unit input mean was assumed. However, the calculated signal-to-quantization noise ratios are valid regardless of the input mean. The optimal quantization step size was determined to three significant digits by repeated program execution over a simple iterative process. Furthermore, the optimal value for the quantizer output corresponding to the first quantization level was determined as well. The data obtained, along with the value of R used to obtain the data, appears in Table 5.

The necessary subroutines were also used to determine the input, the output, and the error power spectral densities for the quantization process ranging from the use of 1 to 5 bits. The input power spectral density for the chosen input autocorrelation function appears in Figure 6.

Figures 7 through 11 illustrate the trend of the output power spectral density as the number of bits employed increases from 1 to 5. Note that the delta function appearing at $\frac{f}{\alpha} = 0$ increases and approaches the value $\frac{x_0^2}{\alpha}$, or $\frac{1}{\alpha}$, as the number of bits used for quantization increases. Figure 12 illustrates the trend of the error power spectral density as the number of bits used increases in a like manner.

Table 4. Subroutines Used for the Negative-Exponential Input Case

Subroutine	Purpose	Called By
nopone	Calculates the normalized noise power and the signal-to-quantization noise ratio.	User. This subroutine must be provided the quantization step size, the mean of the negative-exponential input, and the number of bits used for quantization.
pospne	Determines the power spectral densities of the quantization error, the quantizer output, and the quantizer input for the chosen input autocorrelation function and places them in files named nexerr, nexout, and nexin, respectively.	User. This subroutine must be provided the quantization step size, the mean of the negative-exponential input, and the number of bits used for quantization.
dtakne	Determines all values of a_k and $\hat{a}_k$ for $k = 0$ through $k = R$. For the case provided in Section B.2, $R = 10,000$.	nopone or pospne.
getg0	Determines the appropriate $g(x = 0)$ in order to minimize error while satisfying the necessary conditions for valid computations.	dtakne.
clakn1	Calculates the value of a_k for $k = 0$ through $k = 50$.	dtakne.
elpdt	Evaluates the Laguerre polynomial difference term for a particular argument. This term is the evaluation of $L_k(x) - L_{k-1}(x)$ for the given argument.	clakn1.
comb	Evaluates the combination function of "m choose n". In other words, it determines the number of ways that n items can be selected from m total items.	elpdt.
dfacto	Finds the factorial of its first argument and returns the factorial as its second argument. The second is in double precision format in order to allow larger values.	elpdt.
clakn2	Calculates the value of $\hat{a}_k$ for $k = 51$ through $k = R$. For the case provided in Section B.2, $R = 10,000$.	dtakne.

Table 5. Calculated Noise Related Figures of Merit for a Negative-Exponential Input

Number of Bits	R	First Quantization Level Output Value (units)	Step Size (units)	Normalized Noise Power (units squared)	Signal-to-Quantization Noise Ratio (dB)
1	10^4	0.001	1.91	$3.77 \cdot 10^{-1}$	7.24
2	10^4	0.244	1.08	$1.37 \cdot 10^{-1}$	11.7
3	10^5	0.221	0.660	$4.96 \cdot 10^{-2}$	16.1
4	10^5	0.161	0.400	$1.70 \cdot 10^{-2}$	20.7
5	10^6	0.104	0.235	$5.75 \cdot 10^{-3}$	25.4
6	10^6	0.065	0.139	$1.86 \cdot 10^{-3}$	30.3
7	10^6	0.038	0.0784	$5.81 \cdot 10^{-4}$	35.4
8	10^6	0.023	0.0465	$1.81 \cdot 10^{-4}$	40.4

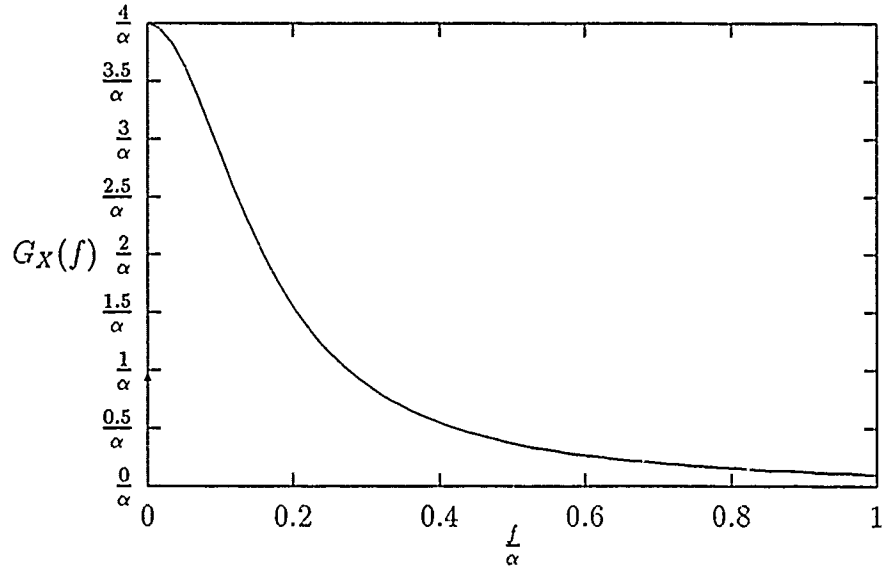

Figure 6. The Power Spectral Density of the Negative-Exponential Input Signal

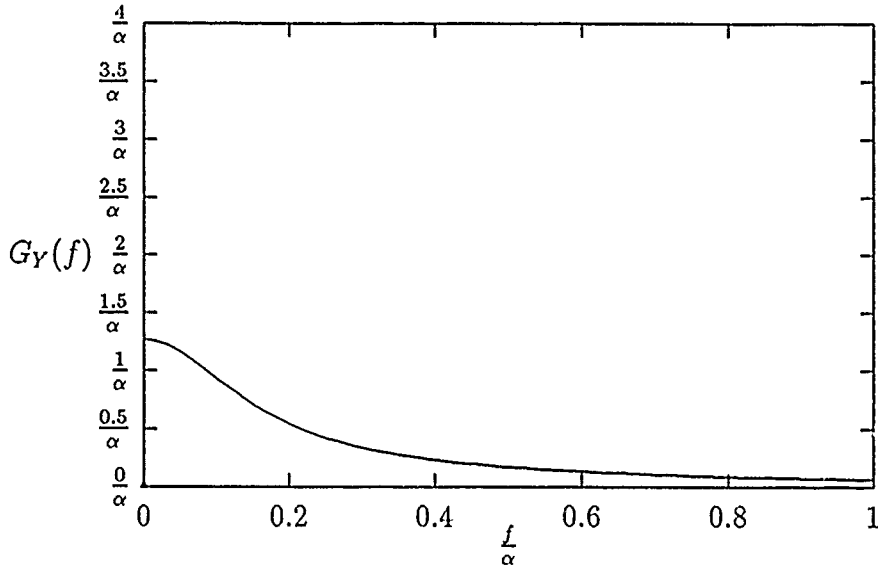

Figure 7. The Power Spectral Density of a 1-Bit Quantizer Output for a Negative-Exponential Input

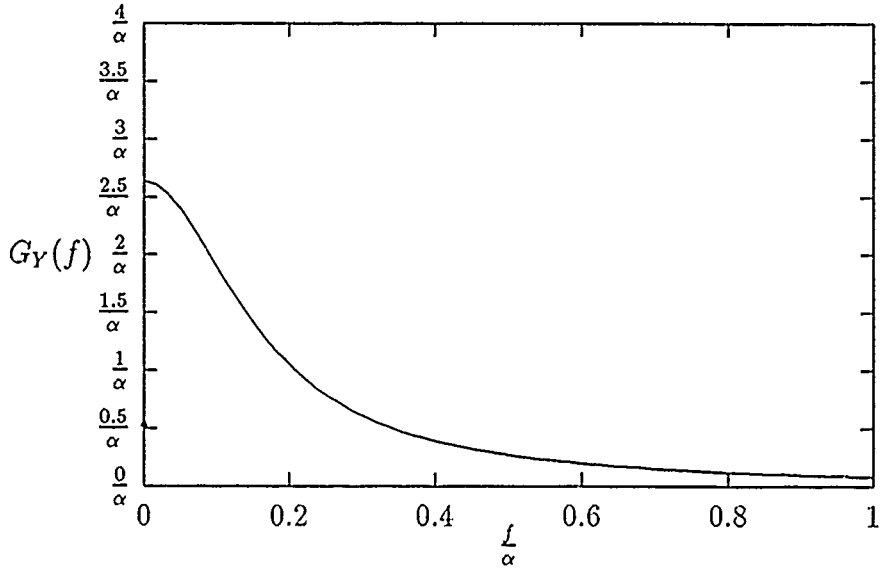

Figure 8. The Power Spectral Density of a 2-Bit Quantizer Output for a Negative Exponential Input

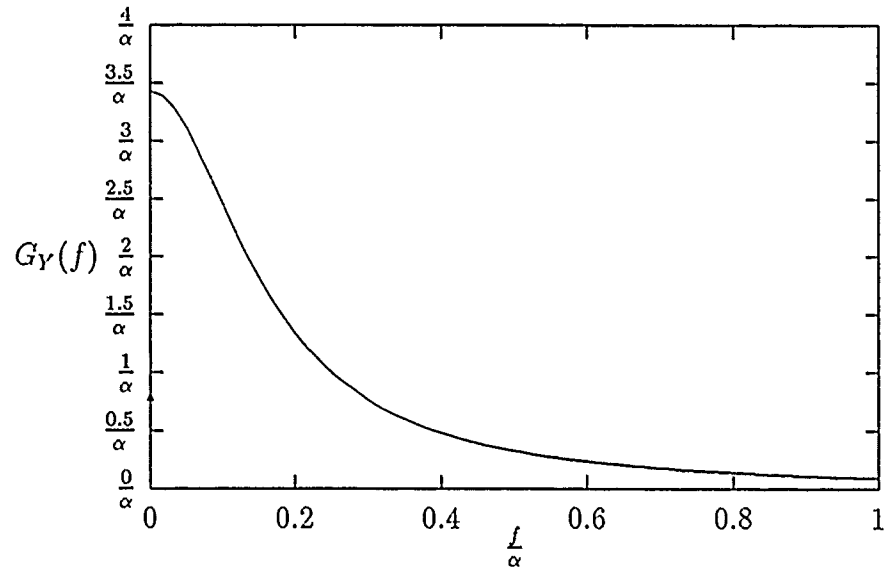

Figure 9. The Power Spectral Density of a 3-Bit Quantizer Output for a Negative-Exponential Input

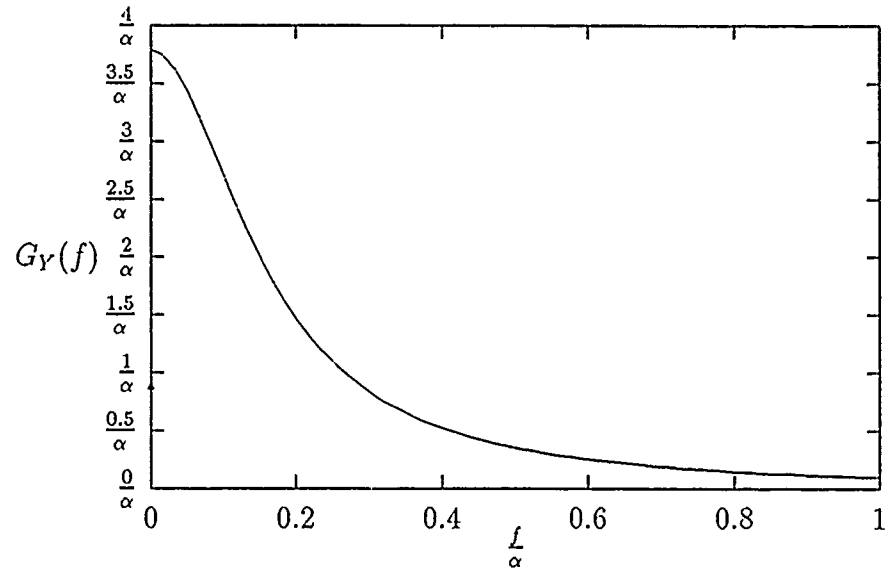

Figure 10. The Power Spectral Density of a 4-Bit Quantizer Output for a Negative-Exponential Input

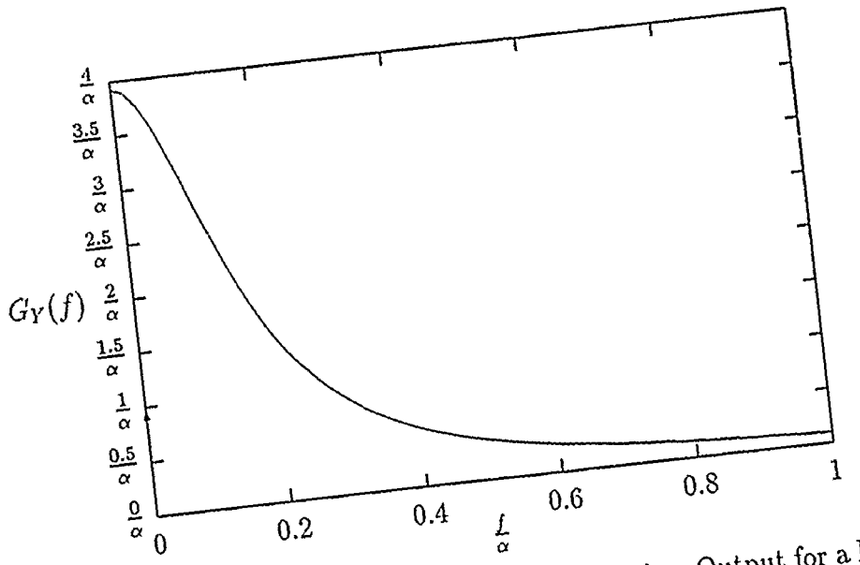

Figure 11. The Power Spectral Density of a 5-Bit Quantizer Output for a Negative-Exponential Input

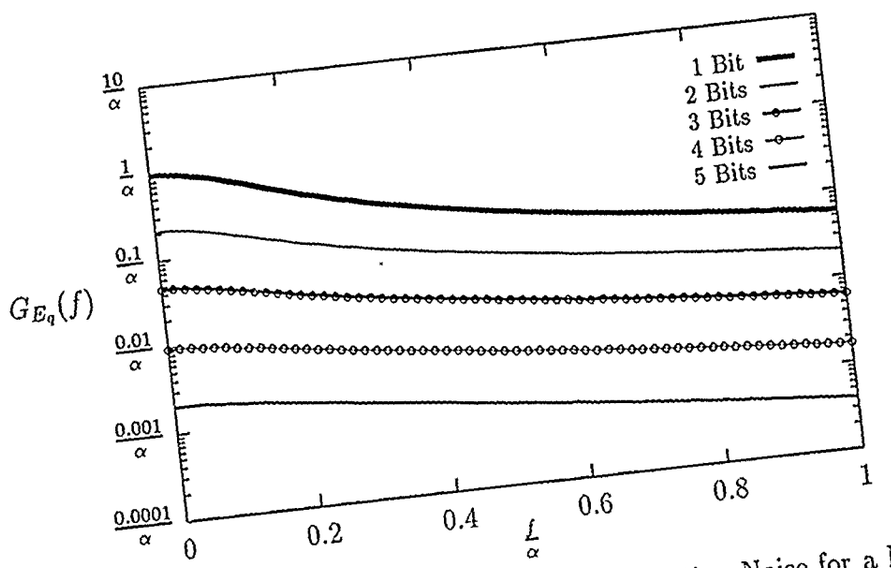

Figure 12. The Power Spectral Density of the Quantization Noise for a Negative-Exponential Input

4.3 The Sinusoidal Case with Random Phase

4.3.1 Approaching the Random Sinusoidal Input Equations For the case of this class of inputs, computations become more direct and straightforward than for the prior two cases. As before, the problem begins with approaching the intermodulation coefficients. The equation for determining these coefficients appears as Equation 128, and is repeated here as

$$a_k = \frac{2q^2}{\pi^2}\frac{1}{(2k+1)^2}\left[\sum_{i=-M}^{M} U_{2k+1}\left(\frac{iq}{A}\right)\right]^2 \tag{176}$$

Applying the property given as Equation 122 to Equation 176 reveals that

$$a_k = \frac{2q^2}{\pi^2}\frac{1}{(2k+1)^2}\left[U_{2k+1}(0) + 2\sum_{i=1}^{M} U_{2k+1}\left(\frac{iq}{A}\right)\right]^2 \tag{177}$$

where

$$U_{2k+1}(x) = \sin[(2k+1)\arccos(x)] \tag{178}$$

Now, using Equation 177, the expression used to calculate the normalized noise power can be derived from Equation 137 as

$$\begin{aligned} N_q &\approx \sum_{k=0}^{R} a_k + \frac{A^2}{2} - \frac{4q}{\pi}\left[\frac{A}{2} + \sum_{m=1}^{M}\left(A^2 - m^2q^2\right)^{\frac{1}{2}}\right] \\ &\approx \sum_{k=0}^{R} a_k + b \end{aligned} \tag{179}$$

where b is as defined in Equation 142. The upper limit, R, can be determined by repeatedly increasing its value by a factor of 10 until further increases have no substantial effect on the result. For the purpose of providing data for this thesis, the values $R = 10^4$ and $R = 10^5$ were used. Particular attention was paid to the order of the summation. The intermodulation coefficients were summed in the order of decreasing values of k in an attempt to sum smaller values of a_k first and reduce the

effects of computer roundoff.

Recall that before this expression for N_q can be valid, the ratio of q to A must meet certain criteria as defined by the conditions given as Equation 149 and 150. Therefore, before Equation 179 can be applied, the ratio of q to A must be tested for applicability.

Once N_q is determined, the signal-to-quantization noise ratio is easily obtained by dividing the value of N_q into the input signal power, $S = \frac{A^2}{2}$.

In the process of calculating N_q, it becomes necessary to determine each a_k and the value b. These same values can be used to provide the power spectral densities of the quantization error and of the quantizer output. Equations 135, 143, and 141 reveal the following one-sided power spectral densities:

$$G_X(f) = \frac{A^2}{2}\delta(f - f_0) \tag{180}$$

$$G_Y(f) \approx \sum_{k=0}^{T} a_k \delta[f - (2k+1)f_0] \tag{181}$$

$$G_{E_q}(f) \approx \sum_{k=1}^{T} a_k \delta[f - (2k+1)f_0] + (a_0 + b)\delta(f - f_0) \tag{182}$$

In order to reduce redundancy, each of the above power spectral densities are one-sided and apply only to $f \geq 0$. Also note that for a random sinusoidal input, these power spectral densities can be plotted versus a horizontal axis in increments of f_0. Therefore, at this point, f_0 does not require further specification. Finally, the value T can be limited in order to produce an uncrowded input. The limited value of T will produce a spectrum for $0 \leq f \leq (2T+1)f_0$. The value of T must be chosen to be smaller than that of R.

4.3.2 Programming for Random Sinusoidal Input Results The computer program subroutines which compute the normalized noise power, signal-to-quantization

Table 6. Subroutines Used for the Random Sinusoidal Input Case

Subroutine	Purpose	Called By
noposi	Calculates the normalized noise power and the signal-to-quantization noise ratio.	User. This subroutine must be provided the quantization step size, the amplitude of the random sinusoidal input, and the number of bits used for quantization.
pospsi	Determines the power spectral densities of the quantization error, the quantizer output, and the quantizer input for the chosen input autocorrelation function and places them in files named sinerr, sinout, and sinin, respectively.	User. This subroutine must be provided the quantization step size, the amplitude of the random sinusoidal input, and the number of bits used for quantization.
testra	Tests the given ratio of the quantization step size to the amplitude of the random sinusoidal input. If the ratio given does not allow valid results, an appropriate message is printed and subsequent calculations are forgone.	noposi or pospsi.
dtaksi	Determines all values of a_k for $k = 0$ through $k = R$. For the case provided in Section B.3, $R = 10,000$.	noposi or pospsi.
claksi	Calculates the value of a_k for $k = 0$ through $k = R$. For the case provided in Section B.3, $R = 10,000$.	dtaksi.
evuk	Evaluates the Tchebycheff polynomial of the second kind, $U_k(x)$.	claksi.

noise ratio, and each of the relevant power spectral densities have been coded into the Fortran 77 computer language and can be found in Section B.3 of this thesis. This code applies to the $R = 10,000$ and $T = 20$ case only. Only minor modifications are necessary to increase R or change T.

A synopsis of the subroutines which have been used to produce the results given later in this thesis appears in Table 6.

4.3.3 The Random Sinusoidal Results Once the required subroutines were coded, they were used to generate the normalized noise powers and signal-to-quantization noise ratios for the quantization process ranging to 8 bits. In order to

Table 7. Calculated Noise Related Figures of Merit for a Random Sinusoidal Input

Number of Bits	R	Step Size (units)	Normalized Noise Power (units squared)	Signal-to-Quantization Noise Ratio (dB)
1	10^4	1.58	$1.18 \cdot 10^{-1}$	6.26
2	10^4	0.607	$2.24 \cdot 10^{-2}$	13.5
3	10^4	0.274	$5.12 \cdot 10^{-3}$	19.9
4	10^4	0.131	$1.25 \cdot 10^{-3}$	26.0
5	10^4	0.0639	$3.10 \cdot 10^{-4}$	32.1
6	10^5	0.0316	$7.81 \cdot 10^{-5}$	38.1
7	10^5	0.0158	$2.12 \cdot 10^{-5}$	43.7
8	10^5	0.00784	$5.01 \cdot 10^{-6}$	50.0

obtain the normalized noise powers, a unit amplitude was assumed for the random sinusoidal input. However, the calculated signal-to-quantization noise ratios are valid regardless of the input amplitude. The optimal quantization step size was determined to three significant digits by repeated program execution over a simple iterative process. The data obtained, along with the value of R used to obtain the data, appears in Table 7.

The necessary subroutines were also used to determine the input, the output, and the error power spectral densities for the quantization process ranging from the use of 1 to 5 bits. The input power spectral density for a unit input amplitude appears in Figure 13.

Figures 14 through 18 illustrate the trend of the resulting output power spectral density as the number of bits employed increases from 1 to 5. Figures 19 through 23 illustrate the trend of the error power spectral density as the number of bits used increases in a like manner.

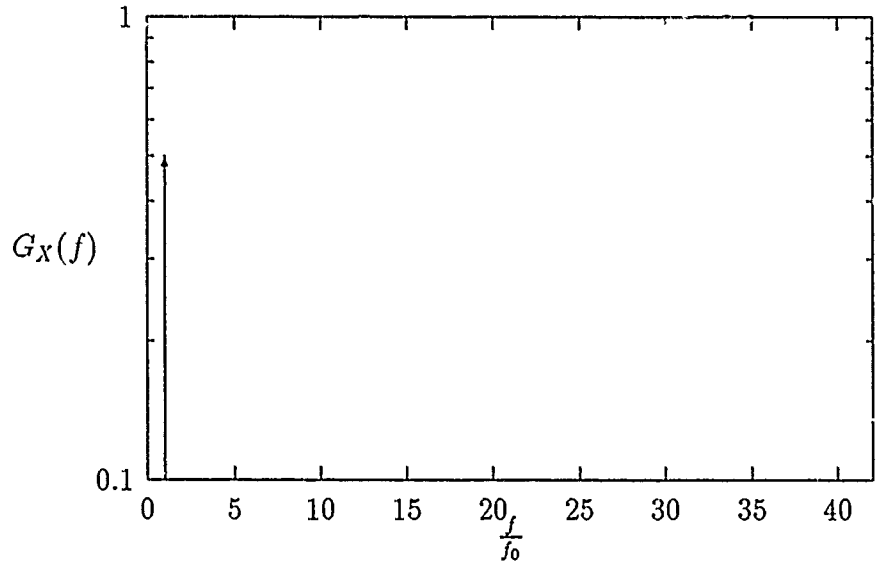

Figure 13. The Power Spectral Density of the Random Sinusoidal Input Signal

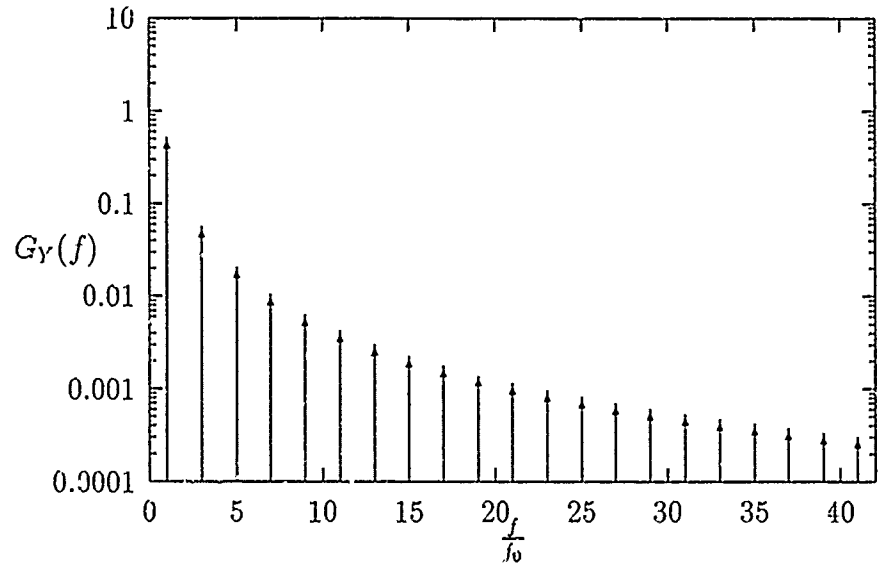

Figure 14. The Power Spectral Density of a 1-Bit Quantizer Output for a Random Sinusoidal Input

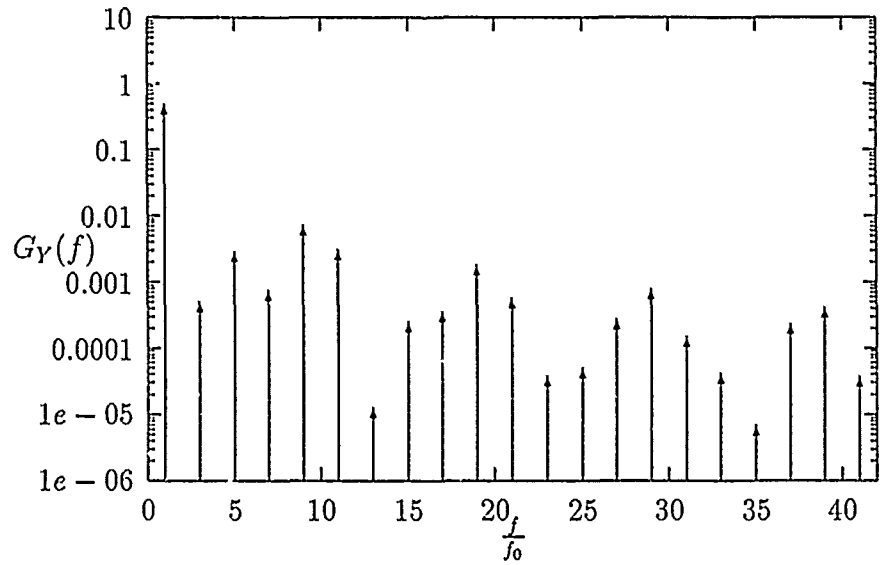

Figure 15. The Power Spectral Density of a 2-Bit Quantizer Output for a Random Sinusoidal Input

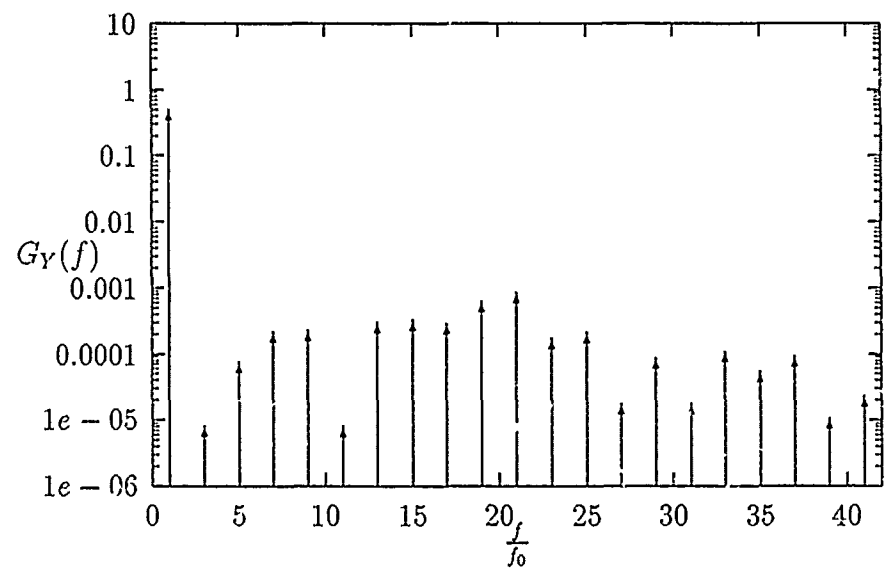

Figure 16. The Power Spectral Density of a 3-Bit Quantizer Output for a Random Sinusoidal Input

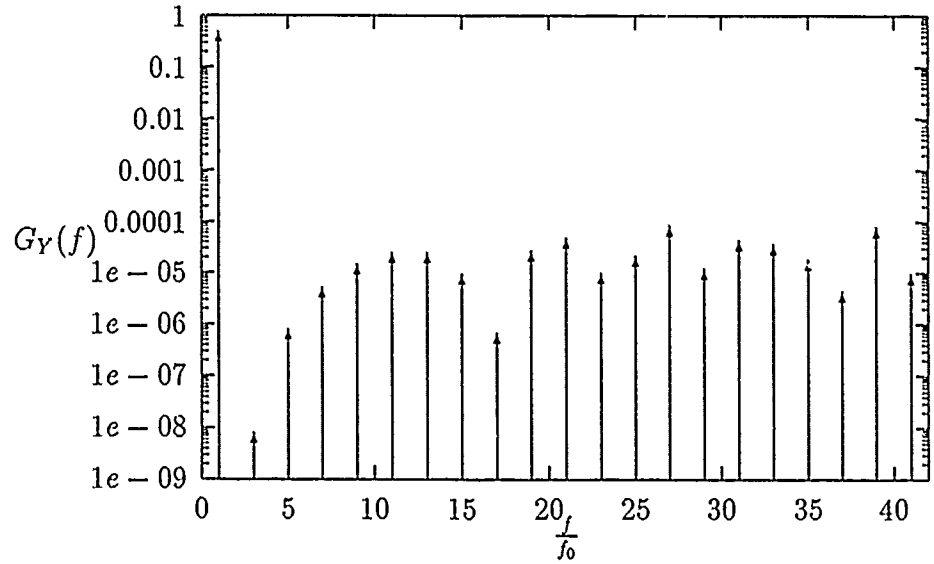

Figure 17. The Power Spectral Density of a 4-Bit Quantizer Output for a Random Sinusoidal Input

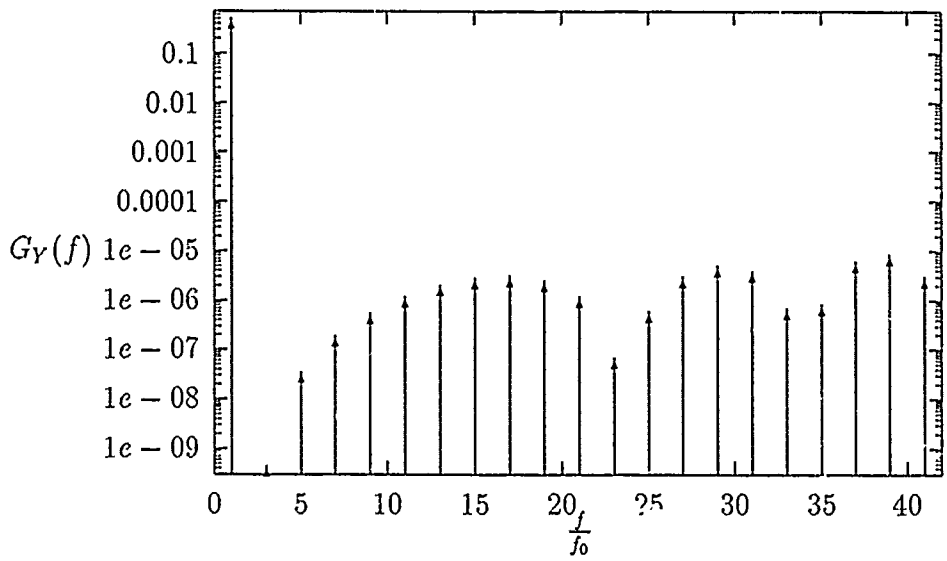

Figure 18. The Power Spectral Density of a 5-Bit Quantizer Output for a Random Sinusoidal Input

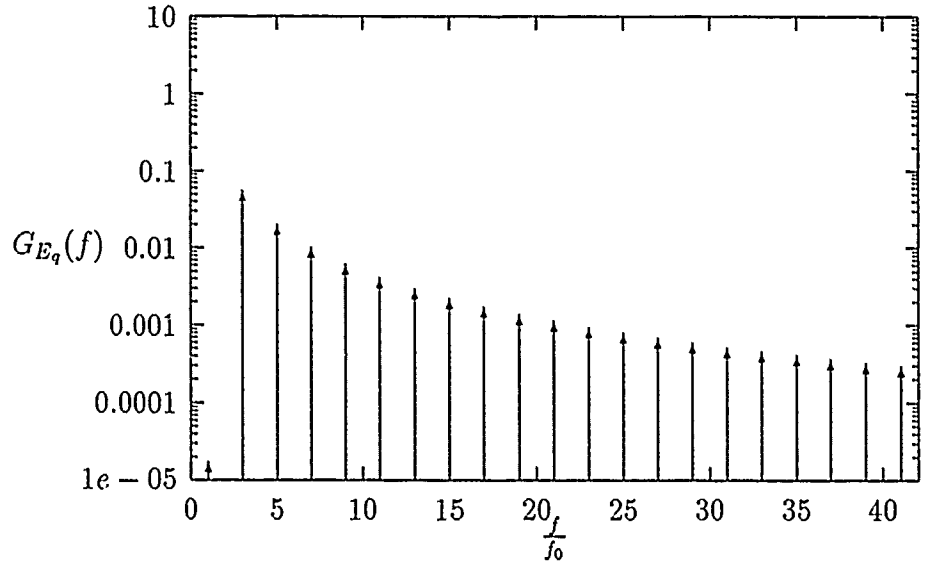

Figure 19. The Power Spectral Density of the 1-Bit Quantization Noise for a Random Sinusoidal Input

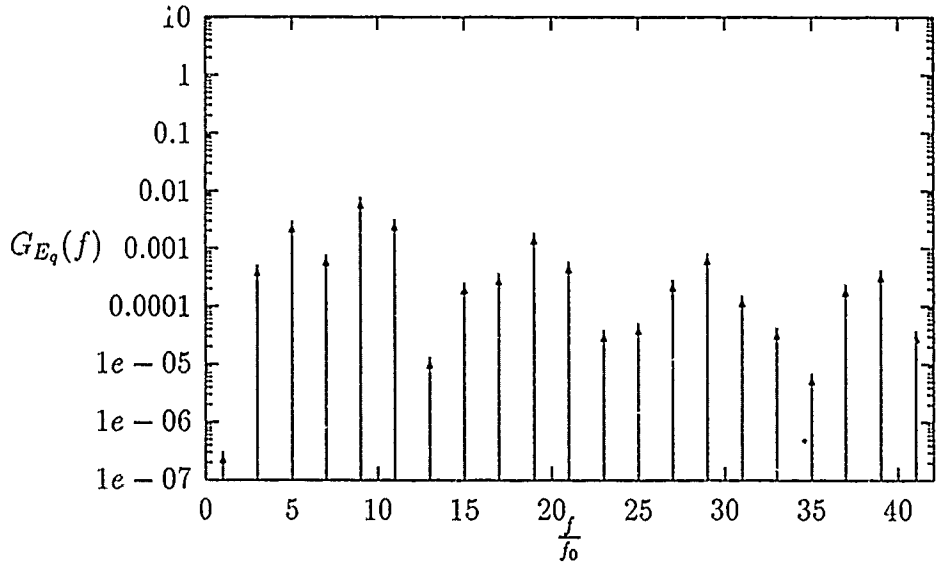

Figure 20. The Power Spectral Density of the 2-Bit Quantization Noise for a Random Sinusoidal Input

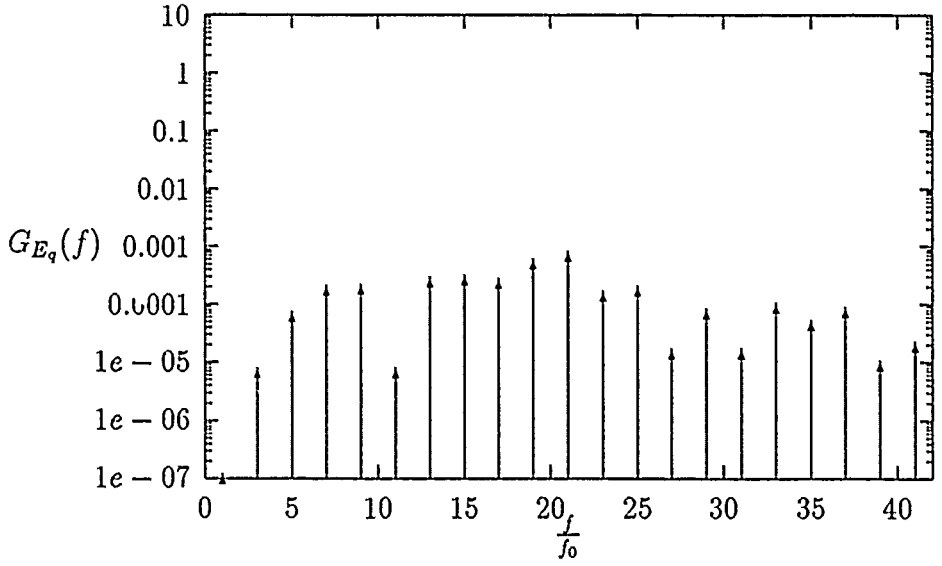

Figure 21. The Power Spectral Density of the 3-Bit Quantization Noise for a Random Sinusoidal Input

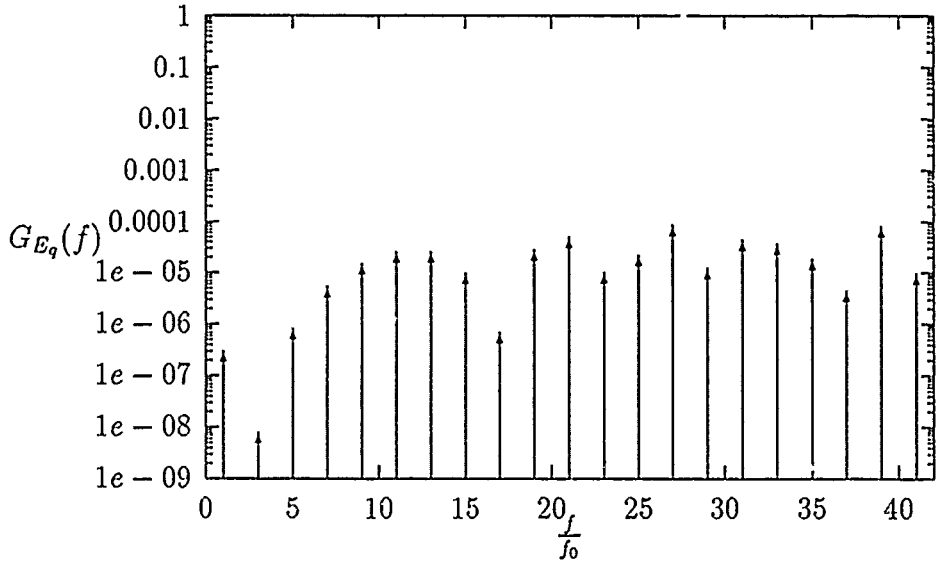

Figure 22. The Power Spectral Density of the 4-Bit Quantization Noise for a Random Sinusoidal Input

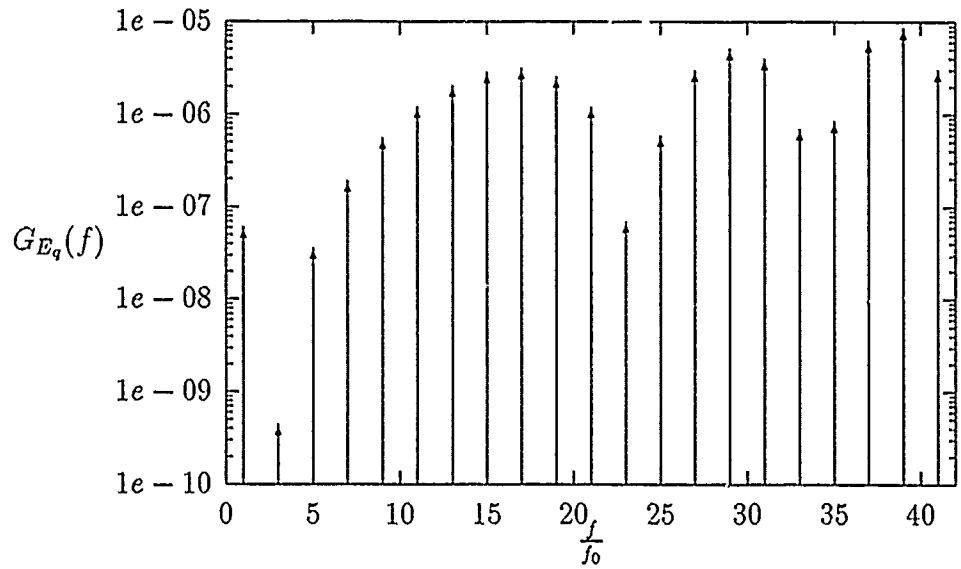

Figure 23. The Power Spectral Density of the 5-Bit Quantization Noise for a Random Sinusoidal Input

V. Conclusions and Recommendations

5.1 Conclusions

For three classes of input signals, this thesis has developed a relationship between the number of quantization levels and the resulting noise characteristics. For each case, this relationship was characterized by expressions for the normalized noise power, the signal-to-quantization noise ratio, the quantization error power spectral density and the quantizer output power spectral density. These expressions were in turn used to obtain the results given in Tables 3, 5 and 7. Furthermore, by assuming the input power spectral densities shown in Figures 3, 6 and 13, the quantization error power spectral densities were determined to appear as shown in Figures 5, 12 and 19 through 23 as applicable to the corresponding input signal classification. Similarly, the quantizer output power spectral densities were determined to appear as shown in Figures 4, 7 through 11 and 14 through 18.

In actuality, this thesis effort resulted in no unexpected results. However, there was a discovery that certain requirements regarding quantization parameters were to be met before the derived theoretical expressions became valid. This discovery was not anticipated. These requirements are summarized in Table 8.

By comparing the signal-to-quantization noise ratios of the three classes of inputs, it is evident that the Gaussian and negative-exponential input cases result in

Table 8. Quantization Parameter Requirements for Valid Expressions

Input Signal Distribution	Parameter Affected	Condition Required
Gaussian	None	N/A
Negative-Exponential	$g(x=0)$	See Equation 103
Random Sinusoidal	$\frac{q}{A}$	See Equations 149 and 150

similar ratios, particularly as the number of bits used in the quantization process increases. In contrast, the random sinusoidal input case fares much better. The reason for this better performance is that the random sinusoidal case has no possibility of providing an input larger than A. The other two cases must account for the probability of inputs approaching infinity. Therefore, as the number of bits used increases, the optimal quantization step size leads to a higher saturation level. This trend is not necessary for the case of the random sinusoidal input. Consequently, the optimal step sizes are much smaller and lead to a more accurate representation of the input.

5.2 Recommendations

It is recommended that further studies be directed toward incorporating the effects of sampling into the results obtained by this thesis. Recall that this thesis was limited in scope to a continuous-time quantizer.

It is also recommended that further work be done to extend the results of this thesis effort to include non-uniform quantization. It is well known that non-uniform quantization, although more difficult to theoretically analyze, provides better noise performance than uniform quantization.

Finally, further efforts regarding the consideration of additional classes of input signals are also recommended. The three input signal classifications studied by this thesis are only a subset of the signals relevant to today's applications.

Appendix A. *Key Derivations*

A.1 The Determination of a Simple Relationship between $R_X(\tau)$ and $\mu(\tau)$ for a Negative-Exponential Input

The integral under consideration for determining the subject relationship was expressed as Equation 85 and is restated here as

$$R_X(\tau) = \int_0^\infty \int_0^\infty \frac{x_1 x_2}{x_0^2[1-\mu^2(\tau)]} I_0\left[\frac{2\sqrt{x_1 x_2}}{x_0}\frac{\mu(\tau)}{1-\mu^2(\tau)}\right] \cdot \exp\left(-\frac{x_1+x_2}{x_0[1-\mu^2(\tau)]}\right) dx_1\, dx_2 \tag{183}$$

By rearranging the double integral and by letting

$$\alpha = \frac{1}{x_0[1-\mu^2(\tau)]} \tag{184}$$

and

$$\begin{aligned}\beta &= \frac{\sqrt{x_2}}{x_0}\frac{\mu(\tau)}{1-\mu^2(\tau)} \\ &= \alpha\sqrt{x_2}\mu(\tau)\end{aligned} \tag{185}$$

Equation 183 can be written as

$$R_X(\tau) = \frac{\alpha}{x_0}\int_0^\infty \left\{x_2 \exp(-\alpha x_2)\int_0^\infty [x_1 \exp(-\alpha x_1) I_0(2\beta\sqrt{x_1})\, dx_1]\, dx_2\right\} \tag{186}$$

Applying an integral identity given by Gradshteyn and Ryzhik (7:720) results in

$$R_X(\tau) = \frac{\alpha}{x_0}\int_0^\infty x_2 \exp(-\alpha x_2)\left[\frac{1}{\beta\alpha^{\frac{3}{2}}}\exp\left(\frac{\beta^2}{2\alpha}\right) \cdot M_{-\frac{3}{2},0}\left(\frac{\beta^2}{\alpha}\right)\right] dx_2 \tag{187}$$

where $M_{\lambda,\mu}(z)$ is the Whittaker function (7:1059) defined by

$$M_{\lambda,\mu}(z) \equiv z^{\mu+\frac{1}{2}} \exp\left(-\frac{z}{2}\right) \Phi\left(\mu - \lambda + \frac{1}{2}, 2\mu + 1; z\right) \tag{188}$$

The degenerate hypergeometric function (7:1058), $\Phi(\xi, \gamma; z)$, is defined by

$$\begin{aligned} \Phi(\xi,\gamma;z) \equiv\ & 1 + \frac{\xi}{\gamma}\frac{z}{1!} + \frac{\xi(\xi+1)}{\gamma(\gamma+1)}\frac{z^2}{2!} \\ & + \frac{\xi(\xi+1)(\xi+2)}{\gamma(\gamma+1)(\gamma+2)}\frac{z^3}{3!} + \cdots \end{aligned} \tag{189}$$

Applying the identities given as Equations 188 and 189 to Equation 187 and letting $z = \frac{\beta^2}{\alpha}$ yields

$$\begin{aligned} R_X(\tau) &= \frac{1}{\alpha x_0}\int_0^\infty x_2 \exp(-\alpha x_2)\left[1 + 2z + \frac{3z^2}{2!} + \frac{4z^3}{3!} + \cdots\right] dx_2 \\ &= \frac{1}{\alpha x_0}\int_0^\infty x_2 \exp(-\alpha x_2)\left[\sum_{n=0}^{\infty}\frac{(n+1)z^n}{n!}\right] dx_2 \\ &= \frac{1}{\alpha x_0}\int_0^\infty x_2 \exp(-\alpha x_2)\left\{\frac{d}{dz}\left[\sum_{n=0}^{\infty}\frac{z^{n+1}}{n!}\right]\right\} dx_2 \\ &= \frac{1}{\alpha x_0}\int_0^\infty x_2 \exp(-\alpha x_2)\left\{\frac{d}{dz}\left[z\sum_{n=0}^{\infty}\frac{z^n}{n!}\right]\right\} dx_2 \\ &= \frac{1}{\alpha x_0}\int_0^\infty x_2 \exp(-\alpha x_2)\left[\frac{d}{dz}(ze^z)\right], dx_2 \\ &= \frac{1}{\alpha x_0}\int_0^\infty x_2 \exp(-\alpha x_2)(1+z)e^z\, dx_2 \end{aligned} \tag{190}$$

Recalling Equation 185 and recalling that $z = \frac{\beta^2}{\alpha}$ reveals that

$$\begin{aligned} R_X(\tau) &= \frac{1}{\alpha x_0}\int_0^\infty \Big\{x_2 \exp(-\alpha x_2)[1 + \alpha\mu^2(\tau)x_2] \\ &\quad \bullet \exp[\alpha\mu^2(\tau)x_2]\Big\}\, dx_2 \\ &= \frac{1}{\alpha x_0}\int_0^\infty [x_2 + \alpha\mu^2(\tau)x_2^2]\exp\{-\alpha[1-\mu^2(\tau)]x_2\}\, dx_2 \\ &= \frac{1}{\alpha x_0}\left[\int_0^\infty x_2 \exp\{-\alpha[1-\mu^2(\tau)]x_2\}\, dx_2\right] \end{aligned}$$

$$+\frac{\mu^2(\tau)}{x_0}\int_0^\infty x_2^2\exp\{-\alpha[1-\mu^2(\tau)]x_2\}\,dx_2$$
$$=\frac{1}{\alpha x_0}\left[\frac{1}{\alpha^2[1-\mu^2(\tau)]^2}\right]+\frac{\mu^2(\tau)}{x_0}\left[\frac{2}{\alpha^3[1-\mu^2(\tau)]^3}\right] \tag{191}$$

Finally, recalling Equation 184 reveals that

$$R_X(\tau)=x_0^2[1+\mu^2(\tau)] \tag{192}$$

A.2 The Determination of a Simple Relationship between $R_X(\tau)$ and $\cos(m\omega_0\tau)$ for a Sinusoidal Input with Random Phase

The expression under consideration for determining the subject relationship was expressed as Equation 134 and is restated here as

$$R_X(\tau)=\sum_{m=0}^{\infty}\frac{\epsilon_m}{\pi^2}\cos(m\omega_0\tau)\mathcal{I}_m^2 \tag{193}$$

where

$$\begin{aligned}\mathcal{I}_m &= \int_{-A}^{A}\frac{x}{(A^2-x^2)^{\frac{1}{2}}}T_m\left(\frac{x}{A}\right)dx\\ &= \int_{-A}^{A}\frac{x}{(A^2-x^2)^{\frac{1}{2}}}\cos\left[m\arccos\left(\frac{x}{A}\right)\right]dx\end{aligned} \tag{194}$$

Applying the trigonometric substitution $x=A\cos\theta$ to Equation 194 yields

$$\begin{aligned}\mathcal{I}_m &= A\int_0^\pi\cos\theta\cos(m\theta)\,d\theta\\ &= \begin{cases}\frac{A\pi}{2} & m=1\\ 0 & \text{elsewhere}\end{cases}\end{aligned} \tag{195}$$

Applying this result to Equation 193 provides the following simple expression:

$$R_X(\tau)=\frac{A^2}{2}\cos(\omega_0\tau) \tag{196}$$

Appendix B. *Computer Programming Source Code*

B.1 Source Code Used for the Gaussian Input Results

```
************************************************************************
*   SUBROUTINE NOPOGA                                                  *
************************************************************************
*     This subroutine determines the normalized noise power and the    *
*     signal-to-quantization noise ratio (in dBs) when given the       *
*     quantization step size, the standard deviation of the Gaussian   *
*     input, and the number of bits used in the quantization process.  *
************************************************************************
*     Variables:                                                       *
*         q       :   The quantization step size                       *
*         sig     :   The standard deviation of the Gaussian input     *
*         noofbt  :   The number of bits used in the quantization      *
*                         process                                      *
*         akpa    :   The array which ultimately contains all of the   *
*                         desired a sub k primes                       *
*         b       :   An additional quantity later required to pro-    *
*                         vide the power spectral density of the       *
*                         quantization noise                           *
*         nopo    :   The calculated normalized noise power            *
*         sinora  :   The signal-to-quantization noise ratio in its    *
*                         dimensionless state                          *
*         sinodb  :   The signal-to-quantization noise ratio in dBs    *
*         lev     :   The number of quantization levels                *
*         k       :   The parameter which indicates the desired        *
*                         a sub k prime                                *
************************************************************************
      subroutine nopoga(q,sig,noofbt)
          real q, sig, akpa(0:10000), b, nopo, sinora, sinodb
          integer noofbt, lev, k
          call dtakga(q,sig,noofbt,akpa,b)
          lev = 2**(noofbt)
          nopo = 0.
          do 20 k = 10000, 0, -1
              nopo = nopo + akpa(k)
   20     continue
          nopo = nopo + (sig**2 * b)
          write(6,30) 'The normalized noise power for ', lev,
     +         ' levels with a step size of ', q, ' units'
          write(6,40) '   and a Gaussian input with a standard ',
     +         'deviation of ', sig, ' units '
          write(6,50) '   is ', nopo, ' units squared.'
          sinora = (sig**2)/nopo
          sinodb = 10. * log10(sinora)
```

```
      write(6,60) 'The resulting signal-to-quantization noise ',
     +     'ratio is ', sinodb, ' dB.'
      write(6,*) ' '
   30 format (1x, a31, i3, a28, f7.4, a6)
   40 format (1x, a40, a13, f5.2, a7)
   50 format (1x, a6, g10.3, a15)
   60 format (1x, a43, a9, f5.2, a4)
      end
*************************************************************************
*   SUBROUTINE POSPGA                                                   *
*************************************************************************
*     This subroutine outputs the data necessary to plot the power      *
*     spectral density of the quantization noise for a Gaussian input   *
*     with a specified autocorrelation.  This data allows the plotting  *
*     of the power spectral density in increments of 1/alpha versus     *
*     the horizontal axis of freq/alpha, where alpha is a damping       *
*     factor pertaining to the specified autocorrelation function.      *
*************************************************************************
*     Variables:                                                        *
*         q       :   The quantization step size                        *
*         sig     :   The standard deviation of the Gaussian input      *
*         noofbt  :   The number of bits used in the quantization       *
*                         process                                       *
*         pi      :   The standard constant                             *
*         aka     :   The array which stores the previosly calculated   *
*                         a sub k's                                     *
*         b       :   A quantity calculated earlier which is necessary *
*                         to provide the power spectral density of the  *
*                         quantization noise                            *
*         falph   :   The frequency divided by the parameter alpha      *
*         psdo    :   The output power spectral density in increments   *
*                         of 1/alpha for a particular falph             *
*         psde    :   The error power spectral density in increments    *
*                         of 1/alpha for a particular falph             *
*         psdi    :   The input power spectral density in increments    *
*                         of 1/alpha for a particular falph             *
*         indf    :   An index used to iterate through falph's          *
*         k       :   The parameter which indicates the desired         *
*                         a sub k prime                                 *
*         dukpl1  :   The value 2k + 1                                  *
*************************************************************************
      subroutine pospga(q,sig,noofbt)
         real q, sig, pi, aka(0:10000), b, falph, psdo, psde, psdi
         integer noofbt, indf, k, dukpl1
         pi = 3.1416
         q = q/sig
         sig = 1.
         call dtakga(q,sig,noofbt,aka,b)
         open (unit=10,file='gauerr')
         open (unit=11,file='gauout')
         open (unit=12,file='gauin')
```

```
      do 20 indf = 0, 236
          falph = indf/59.
          psdo = 0.
          do 10 k = 10000, 0, -1
              dukpl1 = 2 * k + 1
              psdo = psdo + (dukpl1 * aka(k))/
     +                    (dukpl1**2 + (2. * pi * falph)**2)
   10     continue
          psde = 4. * (psdo +
     +         (b/(1. + (2. * pi * falph)**2)))
          psdo = psdo * 4.
          psdi = 4./(1. + (2. * pi * falph)**2)
          write(10,30) falph, psde
          write(11,30) falph, psdo
          write(12,30) falph, psdi
   20     continue
   30     format (1x, f8.6, 5x, e12.5)
          close (unit=10)
          close (unit=11)
          close (unit=12)
      end
*************************************************************************
*   SUBROUTINE DTAKGA                                                   *
*************************************************************************
*     This subroutine determines all values of a sub k prime for k = 0 *
*     through k = 10,000.  It also produces the constant b, which,      *
*     along with the a sub k primes, is necessary to determine the      *
*     quantization noise spectrum.                                      *
*************************************************************************
*     Variables:                                                        *
*         q       :   The quantization step size                        *
*         sig     :   The standard deviation of the Gaussian input      *
*         noofbt  :   The number of bits used in the quantization       *
*                         process                                       *
*         akpa    :   The array which ultimately contains all of the    *
*                         desired a sub k primes                        *
*         b       :   An additional quantity later required to pro-     *
*                         vide the power spectral density of the        *
*                         quantization noise                            *
*         pi      :   The usual constant                                *
*         qsigra  :   The ratio of the step size to the standard        *
*                         deviation of the Gaussian input               *
*         sumex   :   The sum of the iterated exponential terms         *
*         exarg   :   The iterated argument of the exponential term     *
*                         necessary to determine b                      *
*         h2kmn1  :   The polynomial array representation of the        *
*                         Hermite polynomial of degree 2k - 1           *
*         h2k     :   The polynomial array representation of the        *
*                         Hermite polynomial of degree 2k               *
*         m       :   The number transitions between quantization       *
*                         levels in the positive (or negative) non-     *
```

```
*                        zero range                                    *
*        ind     :   An index used in the process                      *
*        k       :   The parameter which indicates the desired         *
*                        a sub k prime                                 *
************************************************************************
      subroutine dtakga(q,sig,noofbt,akpa,b)
          real q, sig, akpa(0:10000), b, pi, qsigra, sumex, exarg
          double precision h2kmn1(0:31), h2k(0:32)
          integer noofbt, m, ind, k
          pi = 3.1416
          m = 2**(noofbt - 1) - 1
          qsigra = q/sig
          do 10 ind = 0, 31
              h2kmn1(ind) = 0.
              h2k(ind) = 0.
   10     continue
          h2k(32) = 0.
          do 20 k = 0, 16
              call clakg1(akpa(k),h2kmn1,h2k,qsigra,q,m,k)
   20     continue
          do 30 k = 17, 10000
              call clakg2(akpa(k),qsigra,q,m,k)
   30     continue
          sumex = 0.
          exarg = (-(qsigra**2))/2.
          do 40 ind = 1, m
              sumex = sumex + exp((ind**2) * exarg)
   40     continue
          b = 1. - sqrt(2./pi) * qsigra * (1 + 2. * sumex)
      end
************************************************************************
*   SUBROUTINE CLAKG1                                                  *
************************************************************************
*     This subroutine calculates the exact value of a sub k prime and  *
*     is to be used on values of k such that 0 <= k <= 16.  Larger      *
*     values of k will result in overflow during calculations.  Also,   *
*     for larger values of k, the approximation subroutine CLAKP2 is    *
*     quite sufficient.                                                 *
************************************************************************
*     Variables:                                                        *
*         akp     :   The desired value a sub k prime                   *
*         h2kmn1  :   The polynomial array representation of the        *
*                         Hermite polynomial of degree 2k - 1           *
*         h2k     :   The polynomial array representation of the        *
*                         Hermite polynomial of degree 2k               *
*         qsigra  :   The ratio of the step size to the standard        *
*                         deviation of the Gaussian input               *
*         q       :   The quantization step size                        *
*         m       :   The number transitions between quantization       *
*                         levels in the positive (or negative) non-     *
*                         zero range                                    *
```

```
*        k       :   The parameter which indicates the desired       *
*                        a sub k prime                                *
*        pi      :   The standard constant                           *
*        msum    :   The total evaluation of the summation term      *
*        polarg  :   The iterated argument of the Hermite polynomial *
*                        in the summation term of the appropriate     *
*                        a sub k prime equation                       *
*        polqty  :   The iterated evaluation of the Hermite poly-     *
*                        nomial in the summation term of the approp-  *
*                        riate a sub k prime equation                 *
*        sumadd  :   An intermediate value used to obtain msum       *
*        squqty  :   The squared quantity which includes all terms    *
*                        involving Hermite polynomial evaluations     *
*        facqty  :   The factorial of 2k + 1                          *
*        ord     :   The degree of the Hermite polynomials to be      *
*                        evaluated                                    *
*        i       :   An index used in the process                     *
*        dukpl1  :   The value corresponding to 2k + 1                *
**********************************************************************
      subroutine clakg1(akp,h2kmn1,h2k,qsigra,q,m,k)
          real akp, qsigra, q, pi, msum, polarg, polqty, sumadd, squqty,
     +            facqty
          double precision h2kmn1(0:31), h2k(0:32)
          integer m, k, ord, i, dukpl1
          pi = 3.1416
          ord = 2 * k
          call geth2k(h2kmn1,h2k,k)
          msum = 0.
          do 10 i = 1, m
              polarg = i * qsigra
              call evevpl(h2k,ord,polarg,polqty)
              sumadd = polqty * exp((-(polarg**2))/2.)
              msum = msum + sumadd
   10     continue
          polarg = 0.
          call evevpl(h2k,ord,polarg,polqty)
          squqty = (polqty + 2. * msum)**2
          dukpl1 = ord + 1
          call facto(dukpl1,facqty)
          akp = (squqty/facqty) * (q**2/(2. * pi))
      end
**********************************************************************
*   SUBROUTINE GETH2K                                                *
**********************************************************************
*     This subroutine determines the configuration of the Hermite    *
*     polynomials of degree 2k and of degree 2k - 1.  In order to do *
*     so, it must be fed the Hermite polynomials of degree 2k - 2    *
*     and of degree 2k - 3 (the previously determined Hermite poly-  *
*     nomials).                                                      *
**********************************************************************
*     Variables:                                                     *
```

```
*         h2kmn1  :   The polynomial array representation of the       *
*                         Hermite polynomial of degree 2k - 3 (when    *
*                         called) and then of degree 2k - 1 (when      *
*                         returned)                                    *
*         h2k     :   The polynomial array representation of the       *
*                         Hermite polynomial of degree 2k - 2 (when    *
*                         called) and then of degree 2k (when          *
*                         returned)                                    *
*         k       :   The parameter which indicates the desired degree *
*         n       :   The degree of the desired h2kmn1                 *
*         npl1    :   The degree of the desired h2k                    *
*         coind   :   An index used in the process                     *
************************************************************************
      subroutine geth2k(h2kmn1,h2k,k)
          double precision h2kmn1(0:31), h2k(0:32)
          integer k, n, npl1, coind
          if (k .eq. 0) then
              h2k(0) = 1.
              return
          else
              n = 2 * k - 1
              npl1 = 2 * k
              do 10 coind = 1, n, 2
                  h2kmn1(coind) = h2k(coind - 1) - (dble(n - 1.)
     +                              * h2kmn1(coind))
   10         continue
              do 20 coind = 2, npl1, 2
                  h2k(coind) = h2kmn1(coind - 1)
     +                              - (dble(n) * h2k(coind))
   20       continue
              h2k(0) = (-n) * h2k(0)
          end if
      end
************************************************************************
*   SUBROUTINE EVEVPL                                                  *
************************************************************************
*     This subroutine evaluates a polynomial possessing nonzero        *
*     coefficients for even powers of the polynomial argument only.    *
*     The maximum degree of the polynomial which can be handled by     *
*     this subroutine is degree 32.                                    *
************************************************************************
*     Variables:                                                       *
*         polarr  :   The polynomial array representation --           *
*                         polarr(n) is the coefficient                 *
*                         of the n-th power of the polynomial          *
*                         argument                                     *
*         ord     :   The degree of the polynomial to be evaluated     *
*         polarg  :   The argument of the polynomial                   *
*         polans  :   The evaluation of the polynomial for the         *
*                         argument                                     *
*         poladp  :   Iterations of polans in double precision form    *
```

```
*         indexp  :   An index used in the process                    *
*************************************************************************
      subroutine evevpl(polarr,ord,polarg,polans)
          real polarg, polans
          double precision polarr(0:32), poladp
          integer ord, indexp
          poladp = polarr(0)
          do 10 indexp = 2, ord, 2
              poladp = poladp + polarr(indexp) * polarg**indexp
   10     continue
          polans = poladp
      end
*************************************************************************
*   SUBROUTINE FACTO                                                    *
*************************************************************************
*     This subroutine finds the factorial of its first argument and     *
*     returns the factorial as its second argument.                     *
*************************************************************************
*     Variables:                                                        *
*         facarg  :   The integer whose factorial is to be found        *
*         facans  :   The calculated factorial of facarg                *
*         facind  :   An index used in the process                      *
*************************************************************************
      subroutine facto(facarg,facans)
          real facans
          integer facarg, facind
          facans = 1.
          if (facarg .le. 1) return
          do 10 facind = 2, facarg
              facans = facans * facind
   10     continue
      end
*************************************************************************
*   SUBROUTINE CLAKG2                                                   *
*************************************************************************
*     This subroutine calculates the approximate value of a sub k       *
*     prime and is to be used on values of k such that                  *
*     17 <= k <= 10,000.  Smaller values of k should be referred to     *
*     CLAKP1 for an exact calculation.  Larger values of k will be      *
*     insignificant.                                                    *
*************************************************************************
*     Variables:                                                        *
*         akp     :   The desired value a sub k prime                   *
*         qsigra  :   The ratio of the step size to the standard        *
*                         deviation of the Gaussian input               *
*         q       :   The quantization step size                        *
*         m       :   The number transitions between quantization       *
*                         levels in the positive (or negative) non-     *
*                         zero range                                    *
*         k       :   The parameter which indicates the desired         *
*                         a sub k prime                                 *
```

```
*         pi      :   The standard constant                              *
*         cosarg  :   The square of 2k + 1/2 - used to increment the     *
*                         argument of the cosine and sine terms          *
*         msum    :   The total evaluation of the summation term         *
*         mqty    :   An intermediate value used to obtain msum          *
*         sinfac  :   The factor applied to the incremented sine term    *
*         squqty  :   The squared quantity which includes the            *
*                         summation of the cosine and sine terms         *
*         i       :   An index used in the process                       *
*         dukpl1  :   The value 2k + 1                                   *
************************************************************************
      subroutine clakg2(akp,qsigra,q,m,k)
          real akp, qsigra, q, pi, cosarg, msum, mqty, sinfac, squqty
          integer m, k, i, dukpl1
          pi = 3.1416
          cosarg = sqrt(2. * k + 0.5)
          msum = 0.
          do 10 i = 1, m
              mqty = cos(cosarg * i * qsigra)
              sinfac = ((i * qsigra)**3)/(24. * cosarg)
              mqty = mqty + sinfac * sin(cosarg * i * qsigra)
              msum = msum + mqty * exp((-((i * qsigra)**2))/4.)
   10     continue
          squqty = (1. + 2. * msum)**2
          dukpl1 = (2 * k) + 1
          akp = ((q**2)/(2. * pi * dukpl1 * sqrt(pi * k))) * squqty
      end
```

B.2 Source Code Used for the Negative-Exponential Input Results

```
************************************************************************
*   SUBROUTINE NOPONE                                                  *
************************************************************************
*     This subroutine determines the normalized noise power and the    *
*     signal-to-quantization noise ratio (in dBs) when given the       *
*     quantization step size, the mean of the negative-exponential     *
*     input, and the number of bits used in the quantization process.  *
************************************************************************
*     Variables:                                                       *
*         q       :   The quantization step size                       *
*         x0      :   The mean of the negative-exponential input       *
*         noofbt  :   The number of bits used in the quantization      *
*                         process                                      *
*         aka     :   The array which ultimately contains all of the   *
*                         desired a sub k's                            *
*         b       :   An additional quantity later required to pro-    *
*                         vide the power spectral density of the       *
```

```
*                       quantization noise                              *
*       g0      :   The desired value for the quantizer output          *
*                       corresponding to the first level divided by     *
*                       the step size                                   *
*       nopo    :   The calculated normalized noise power               *
*       sinora  :   The signal-to-quantization noise ratio in its       *
*                       dimensionless state                             *
*       sinodb  :   The signal-to-quantization noise ratio in dBs       *
*       lev     :   The number of quantization levels                   *
*       k       :   The parameter which indicates the desired           *
*                       a sub k                                         *
************************************************************************
      subroutine nopone(q,x0,noofbt)
          double precision q, x0, aka(10000), b, g0, nopo, sinora,
     +        sinodb
          integer noofbt, lev, k
          call dtakne(q,x0,noofbt,aka,b,g0)
          lev = 2**(noofbt)
          nopo = 0.
          do 20 k = 10000, 0, -1
              nopo = nopo + aka(k)
   20     continue
          nopo = nopo + (2. * (x0**2) * b)
          write(6,30) 'The normalized noise power for ', lev,
     +                ' levels with a step size of ', q,
     +                ' units'
          write(6,40) '   and a negative-exponential input with a ',
     +                'mean of ', x0, ' units is '
          write(6,50) '   ', nopo, ' units squared.'
          sinora = (2. * x0**2)/nopo
          sinodb = 10. * log10(sinora)
          write(6,60) 'The resulting signal-to-quantization noise ',
     +                'ratio is ', sinodb, ' dB.'
          write(6,70) 'The quantizer output for an input of 0 units',
     +        ' is ', g0 * q, ' units.'
          write(6,*) ' '
   30     format (1x, a31, i3, a28, f7.4, a6)
   40     format (1x, a43, a8, f5.2, a10)
   50     format (1x, a3, g10.3, a15)
   60     format (1x, a43, a9, f5.2, a4)
   70     format (1x, a44, a4, f5.3, a7)
      end
************************************************************************
*   SUBROUTINE POSPNE                                                  *
************************************************************************
*     This subroutine outputs the data necessary to plot the power     *
*     spectral density of the quantization noise for a Negative-Expo-  *
*     nential input with a specified autocorrelation.  This data       *
*     allows the plotting of the power spectral density in increments  *
*     of 1/alpha versus the horizontal axis of freq/alpha, where alpha *
*     is a damping factor pertaining a function of the specified       *
```

```
*     autocorrelation function.                                        *
*************************************************************************
*     Variables:                                                       *
*          q       :   The quantization step size                      *
*          x0      :   The mean of the negative-exponential input      *
*          noofbt  :   The number of bits used in the quantization     *
*                          process                                     *
*          pi      :   The standard constant                           *
*          aka     :   The array which stores the previosly calculated *
*                          a sub k's                                   *
*          b       :   A quantity calculated earlier which is necessary *
*                          to provide the power spectral density of the *
*                          quantization noise                          *
*          g0      :   The desired value for the quantizer output      *
*                          corresponding to the first level divided by *
*                          the step size                               *
*          falph   :   The frequency divided by the parameter alpha    *
*          psdo    :   The output power spectral density in increments *
*                          of 1/alpha for a particular falph           *
*          psde    :   The error power spectral density in increments  *
*                          of 1/alpha for a particular falph           *
*          psdi    :   The input power spectral density in increments  *
*                          of 1/alpha for a particular falph           *
*          psdde   :   The level of the delta function which           *
*                          accompanies the error power spectral density *
*          psddo   :   The level of the delta function which           *
*                          accompanies the output power spectral       *
*                          density                                     *
*          psddi   :   The level of the delta function which           *
*                          accompanies the input power spectral        *
*                          density                                     *
*          indf    :   An index used to iterate through falph's        *
*          k       :   The parameter which indicates the desired       *
*                          a sub k prime                               *
*************************************************************************
      subroutine pospne(q,x0,noofbt)
          double precision q, x0, pi, aka(0:10000), b, g0, falph, psdo,
     +         psde, psdi, psdde, psddo, psddi
          integer noofbt, indf, k
          pi = 3.1416
          call dtakne(q,x0,noofbt,aka,b,g0)
          open (unit=13,file='nexerr')
          open (unit=14,file='nexout')
          open (unit=15,file='nexin')
          do 20 indf = 0, 236
              falph = indf/59.
              psdo = 0.
              do 10 k = 10000, 1, -1
                  psdo = psdo + (k * aka(k))/
     +                        (k**2 + (2. * pi * falph)**2)
   10         continue
```

```
            psde = 4. * ( psdo +
     +           (b * x0**2)/(1. + (2. * pi * falph)**2))
            psdo = psdo * 4.
            psdi = 4. * (x0**2)/(1. + (2. * pi * falph)**2)
            if (indf .eq. 0) then
                psdde = aka(0) + (b * x0**2)
                psddo = aka(0)
                psddi = x0**2
                write(13,30) falph, psde, psdde
                write(14,30) falph, psdo, psddo
                write(15,30) falph, psdi, psddi
            else if (indf .eq. 1) then
                psdde = 0.
                psddo = 0.
                psddi = 0.
                write(13,30) falph, psde, psdde
                write(14,30) falph, psdo, psddo
                write(15,30) falph, psdi, psddi
            else
                write(13,40) falph, psde
                write(14,40) falph, psdo
                write(15,40) falph, psdi
            end if
   20       continue
   30       format (1x, f8.6, 5x, e12.5, 5x, e12.5)
   40       format (1x, f8.6, 5x, e12.5)
            close (unit=13)
            close (unit=14)
            close (unit=15)
        end
*************************************************************************
*   SUBROUTINE DTAKNE                                                   *
*************************************************************************
*     This subroutine determines all values of a sub k for k = 0        *
*     through k = 10,000.                                               *
*************************************************************************
*     Variables:                                                        *
*         aka     :   The array which ultimately contains all of the    *
*                         desired a sub k's                             *
*         qx0ra   :   The ratio of the step size to the mean of the     *
*                         negative-exponential input                    *
*         x0      :   The mean of the negative-exponential input        *
*         n       :   The number of transitions between quantization    *
*                         levels                                        *
*         k       :   The parameter which indicates the desired         *
*                         a sub k                                       *
*         ind     :   An index used in the process                      *
*         g0      :   The desired value for the quantizer output        *
*                         corresponding to the first level divided by   *
*                         the step size                                 *
*         q       :   The quantization step size                        *
```

```
*         b       :   An additional quantity later required to pro-    *
*                         vide the power spectral density of the       *
*                         quantization noise                           *
**********************************************************************
      subroutine dtakne(q,x0,noofbt,aka,b,g0)
          double precision q, x0, aka(0:10000), b, g0, qx0ra
          integer noofbt, n, k
          qx0ra = q/x0
          n = 2**(noofbt) - 1
          call getg0(g0,q,x0,qx0ra,n,b)
          do 20 k = 0, 50
              call clakn1(aka(k),qx0ra,q,n,k,g0)
   20     continue
          do 30 k = 51, 10000
              call clakn2(aka(k),qx0ra,q,n,k,x0,g0)
   30     continue
          end
**********************************************************************
*   SUBROUTINE GETG0                                                 *
**********************************************************************
*     This subroutine determines what the quantizer output corre-    *
*     sponding to the first level should be in order to negate or    *
*     minimize the dc or average value of the quantization error.  In *
*     other words, it determines the appropriate g(x = 0).           *
*     It also produces the constant b, which, along with the a sub k's,*
*     is necessary to determine the quantization noise spectrum.     *
**********************************************************************
*     Variables:                                                     *
*         g0      :   The desired value for the quantizer output     *
*                         corresponding to the first level divided by *
*                         the step size q                            *
*         q       :   The quantization step size                     *
*         x0      :   The mean of the negative-exponential input     *
*         qx0ra   :   The ratio of the step size to the mean of the  *
*                         negative-exponential input                 *
*         n       :   The number transitions between quantization    *
*                         levels                                     *
*         b       :   An additional quantity later required to pro-  *
*                         vide the power spectral density of the     *
*                         quantization noise                         *
*         sumex   :   The sum of the iterated exponential terms      *
*         exarg   :   The iterated argument of the exponential term  *
*                         necessary to determine b                   *
*         srbx0   :   An intermediate value used to determine g0     *
*         exind   :   An index used in the process                   *
**********************************************************************
      subroutine getg0(g0,q,x0,qx0ra,n,b)
          double precision g0, q, x0, qx0ra, b, sumex, exarg, srbx0
          integer n, exind
          g0 = 0.
          sumex = 0.
```

```
          exarg = -qx0ra
          do 10 exind = 1, n
              sumex = sumex + exind * exp(exind * exarg)
   10     continue
          b = 1. - 2. * (qx0ra**2) * sumex
          if (b .lt. 0.) then
              sumex = 0.
              do 20 exind = 1, n
                  sumex = sumex + exp(exind * exarg)
   20         continue
              srbx0 = (x0 * sqrt(-b))/q
              if (sumex .lt. srbx0) g0 = srbx0 - sumex
          end if
      end
*************************************************************************
*   SUBROUTINE CLAKN1                                                   *
*************************************************************************
*     This subroutine calculates the exact value of a sub k and         *
*     is to be used on values of k such that 0 <= k <= 50.  Larger       *
*     values of k will result in overflow during calculations.  Also,    *
*     for larger values of k, the approximation subroutine CLAKN2 is     *
*     quite sufficient.                                                  *
*************************************************************************
*     Variables:                                                         *
*         ak      :   The desired value a sub k                          *
*         qx0ra   :   The ratio of the step size to the mean of the      *
*                         negative-exponential input                     *
*         q       :   The quantization step size                         *
*         n       :   The number of transitions between quantization     *
*                         levels in the positive (or negative) non-      *
*                         zero range                                     *
*         k       :   The parameter which indicates the desired          *
*                         a sub k                                        *
*         g0      :   The desired value for the quantizer output         *
*                         corresponding to the first level divided by    *
*                         the step size                                  *
*         nsum    :   The total evaluation of the summation term         *
*         polarg  :   The iterated argument of the Laguerre poly-        *
*                         nomials in the summation term of the           *
*                         appropriate a sub k prime equation             *
*         polqt   :   The iterated evaluation of the difference of the   *
*                         two relevant Laguerre polynomials              *
*         sumadd  :   An intermediate value used to obtain nsum          *
*         ind     :   An index used in the process                       *
*************************************************************************
      subroutine clakn1(ak,qx0ra,q,n,k,g0)
          double precision ak, qx0ra, q, g0, nsum, polarg, polqt, sumadd
          integer n, k, ind
          nsum = 0.
          do 10 ind = 0, n
              polarg = ind * qx0ra
```

```
            call elpdt(k,polarg,polqt)
            sumadd = exp(-polarg) * polqt
            if (ind .eq. 0) sumadd = g0 * sumadd
            nsum = nsum + sumadd
   10     continue
          ak = (q * nsum)**2
      end
*************************************************************************
*   SUBROUTINE ELPDT                                                    *
*************************************************************************
*     This subroutine evaluates the Laguerre polynomial difference     *
*     term for a particular argument.  This term is the evaluation     *
*     of the Laguerre polynomial of degree k minus the evaluation      *
*     of the Laguerre polynomial of degree k - 1 for the given argu-   *
*     ment.                                                            *
*************************************************************************
*     Variables:                                                       *
*         k       :   The parameter which indicates the desired degree *
*         polarg  :   The iterated argument of the Laguerre poly-      *
*                         nomials in the summation term of the         *
*                         appropriate a sub k prime equation           *
*         polqt   :   The iterated evaluation of the difference of the *
*                         two relevant Laguerre polynomials            *
*         r       :   An index through the appropriate summation term  *
*         kmn1    :   The value of k minus 1                           *
*         rmn1    :   The value of r minus 1                           *
*         cterm   :   The evaluation of the combination term           *
*         rfact   :   The evaluation of r factorial                    *
*************************************************************************
      subroutine elpdt(k,polarg,polqt)
          integer k, r, kmn1, rmn1
          double precision polarg, polqt, cterm, rfact
          if (k .eq. 0) then
              polqt = 1.
              return
          else
              kmn1 = k - 1
              polqt = 0.
              do 10 r = 1, k
                  rmn1 = r - 1
                  call comb(kmn1,rmn1,cterm)
                  call dfacto(r,rfact)
                  polqt = polqt + ((cterm/rfact) * ((-polarg)**r))
   10         continue
          end if
      end
*************************************************************************
*   SUBROUTINE COMB                                                     *
*************************************************************************
*     This subroutine evaluates the combination function of            *
*     "m choose n".  In other words, it determines the number of ways  *
```

```
*     that n items can be selected from m total items.                *
**********************************************************************
*     Variables:                                                       *
*         m       :   The number of items selected from                *
*         n       :   The number of items selected                     *
*         cterm   :   The result of the operation                      *
*         mmnn    :   The value of m minus n                           *
*         lo      :   The smallest value between n and mmnn            *
*         hi      :   The largest value betwe  n + 1 and mmnn + 1      *
*         ind     :   An index used in the process                     *
**********************************************************************
      subroutine comb(m,n,cterm)
          integer m, n, mmnn, lo, hi, ind
          double precision cterm
          mmnn = m - n
          if (mmnn .gt. n) then
              lo = n
              hi = mmnn + 1
          else
              lo = mmnn
              hi = n + 1
          end if
          cterm = 1.
          do 10 ind = m, hi, -1
              if (lo .gt. 1) then
                  cterm = cterm * (dble(ind)/lo)
                  lo = lo - 1
              else
                  cterm = cterm * ind
              end if
   10     continue
      end
**********************************************************************
*   SUBROUTINE DFACTO                                                  *
**********************************************************************
*     This subroutine finds the factorial of its first argument and    *
*     returns the factorial as its second argument.  The second        *
*     is in double precision format in order to allow larger values.   *
**********************************************************************
*     Variables:                                                       *
*         k       :   The integer whose factorial is to be found       *
*         kfact   :   The calculated factorial of facarg               *
*         kind    :   An index used in the process                     *
**********************************************************************
      subroutine dfacto(k,kfact)
          integer k, kind
          double precision kfact
          kfact = 1.
          if (k .le. 1) return
          do 10 kind = 2, k
              kfact = kfact * kind
```

```
   10     continue
      end
*****************************************************************************
*   SUBROUTINE CLAKN2                                                       *
*****************************************************************************
*     This subroutine calculates the approximate value of a sub k           *
*     and is to be used on values of k such that                            *
*     51 <= k <= 10,000.  Smaller values of k should be referred to         *
*     CLAKN1 for an exact calculation.  Larger values of k will be          *
*     insignificant.                                                        *
*****************************************************************************
*     Variables:                                                            *
*         ak      :   The desired value a sub k                             *
*         qx0ra   :   The ratio of the step size to the mean of the         *
*                         negative-exponential input                        *
*         q       :   The quantization step size                            *
*         x0      :   The mean of the negative-exponential input            *
*         g0      :   The desired value for the quantizer output            *
*                         corresponding to the first level divided by       *
*                         the step size                                     *
*         pi      :   The standard constant                                 *
*         piov4   :   The standard constant, pi, divided by 4               *
*         cosar1  :   The the term used to increment the argument           *
*                         of the first cosine term                          *
*         cosar2  :   The the term used to increment the argument           *
*                         of the second cosine term                         *
*         nsum    :   The total evaluation of the summation term            *
*         nqty    :   An intermediate value used to obtain msum             *
*         n       :   The number transitions between quantization           *
*                         levels                                            *
*         k       :   The parameter which indicates the desired             *
*                         a sub k prime                                     *
*         ind     :   An index used in the process                          *
*****************************************************************************
      subroutine clakn2(ak,qx0ra,q,n,k,x0,g0)
          double precision ak, qx0ra, q, x0, g0, pi, piov4, cosar1,
     +            cosar2, nsum, nqty
          integer n, k, ind
          pi = 3.1416
          piov4 = pi/4.
          cosar1 = 2. * sqrt(k * qx0ra)
          cosar2 = 2. * sqrt((k - 1.) * qx0ra)
          nsum = 0.
          do 10 ind = 1, n
              nqty = (x0/((ind * k)**(0.25)))
     +                * cos((sqrt(ind * 1.) * cosar1) - piov4)
              nqty = nqty - ((1./((ind * (k - 1))**(0.25)))
     +                * cos((sqrt(ind * 1.) * cosar2) - piov4))
              nsum = nsum + (nqty * exp(-((ind * qx0ra)/2.)))
   10     continue
          ak = (q**(1.5)) * sqrt(x0) * (1./pi) * nsum**2
```

```
      end
```

B.3 Source Code Used for the Random Sinusoidal Input Results

```
************************************************************************
*   SUBROUTINE NOPOSI                                                  *
************************************************************************
*     This rubroutine determines the normalized noise power and the    *
*     signal-to-quantization noise ratio (in dBs) when given the       *
*     a sub k's as determined by DTAKSI, the quantization step size,   *
*     the amplitude of the sinusoidal input, and the number of bits    *
*     used in the quantization process.                                *
************************************************************************
*     Variables:                                                       *
*         q       :   The quantization step size                       *
*         a       :   The ampltitude of the sinusoidal input           *
*         noofbt  :   The number of bits used in the quantization      *
*                         process                                      *
*         aka     :   The array which ultimately contains all of the   *
*                         desired a sub k's                            *
*         b       :   An additional quantity later required to pro-    *
*                         vide the power spectral density of the       *
*                         quantization noise                           *
*         nopo    :   The calculated normalized noise power            *
*         sinora  :   The signal-to-quantization noise ratio in its    *
*                         dimensionless state                          *
*         sinodb  :   The signal-to-quantization noise ratio in dBs    *
*         k       :   The parameter which indicates the desired        *
*                         a sub k prime                                *
*         lev     :   The number of quantization levels                *
*         valid   :   A logical indication of whether the qara is a    *
*                         valid ratio                                  *
************************************************************************
      subroutine noposi(q,a,noofbt)
          real q, a, aka(0:10000), b, nopo, sinora, sinodb
          integer noofbt, k, lev
          logical valid
          call testra(q,a,noofbt,valid)
          if (valid) then
              call dtaksi(q,a,noofbt,aka,b)
              lev = 2**(noofbt)
              nopo = 0.
              do 20 k = 10000, 0, -1
                  nopo = nopo + aka(k)
   20         continue
              nopo = nopo + b
              write(6,30) 'The normalized noise power for ', lev,
```

```
     +            ' levels with a step size of ', q, ' units '
                write(6,40) '   and a random sinusoidal input with an ',
     +            'amplitude of ', a, ' units '
                write(6,50) '   is ', nopo, ' units squared.'
                sinora = ((a**2)/2)/nopo
                sinodb = 10. * log10(sinora)
                write(6,60) 'The resulting signal-to-quantization noise ',
     +            'ratio is ', sinodb, ' dB.'
                write(6,*) ' '
            end if
   30       format (1x, a31, i3, a28, f7.4, a7)
   40       format (1x, a41, a13, f5.2, a7)
   50       format (1x, a6, g10.3, a15)
   60       format (1x, a43, a9, f5.2, a4)
        end
*************************************************************************
*   SUBROUTINE POSPSI                                                   *
****** *  ********************************************************************
*     This subroutine outputs the data necessary to plot the power      *
*     spectral density of the quantization noise for a random           *
*     sinusoidal input with a specified autocorrelation.  This data     *
*     allows the plotting of the power spectral density versus the      *
*     horizontal axis of frequency in increments of the frequency of    *
*     the random sinusoidal input.                                      *
*************************************************************************
*     Variables:                                                        *
*         q       :   The quantization step size                        *
*         a       :   The ampltitude of the sinusoidal input            *
*         noofbt  :   The number of bits used in the quantization       *
*                         process                                       *
*         aka     :   The array which stores the previosly calculated   *
*                         a sub k's                                     *
*         b       :   A quantity calculated earlier which is necessary *
*                         to provide the power spectral density of the *
*                         quantization noise                            *
*         psde    :   The error power spectral density                  *
*         psdo    :   The output power spectral density                 *
*         psdi    :   The input power spectral density                  *
*         psd0    :   A value set to zero which aids in the plotting    *
*                         of the delta functions                        *
*         indf    :   An index used to iterate through freq's           *
*         freq    :   The frequency in increments of the input          *
*                         frequency                                     *
*         valid   :   A logical indication of whether the qara is a     *
*                         valid ratio                                   *
*************************************************************************
      subroutine pospsi(q,a,noofbt)
          real q, a, aka(0:10000), b, psde, psdo, psdi, psd0
          integer noofbt, indf, freq
          logical valid
          call testra(q,a,noofbt,valid)
```

```
        if (valid) then
            open (unit=16,file='sinerr')
            open (unit=17,file='sinout')
            open (unit=18,file='sinin')
            call dtaksi(q,a,noofbt,aka,b)
            psde = aka(0) + b
            psdo = aka(0)
            psdi = (a**2)/2.
            psd0 = 0.
            freq = 1
            write(16,20) freq, psd0
            write(16,20) freq, psde
            write(16,20) freq, psd0
            write(17,20) freq, psd0
            write(17,20) freq, psdo
            write(17,20) freq, psd0
            write(18,20) freq, psd0
            write(18,20) freq, psdi
            write(18,20) freq, psd0
            do 10 indf = 1, 20
                freq = 2 * indf + 1
                psde = aka(indf)
                psdo = aka(indf)
                write(16,20) freq, psd0
                write(16,20) freq, psde
                write(16,20) freq, psd0
                write(17,20) freq, psd0
                write(17,20) freq, psdo
                write(17,20) freq, psd0
   10       continue
            close (unit=16)
            close (unit=17)
            close (unit=18)
        end if
   20   format (1x, i3, 5x, e12.5)
      end
*************************************************************************
*   SUBROUTINE TESTRA                                                   *
*************************************************************************
*     This subroutine tests the given ratio of the quantization step    *
*     size to the amplitude of the random sinusoidal input in order     *
*     to determine if subsequent calculations are valid.                *
*************************************************************************
*     Variables:                                                        *
*         q       :   The quantization step size                        *
*         a       :   The ampltitude of the sinusoidal input            *
*         noofbt  :   The number of bits used in the quantization       *
*                         process                                       *
*         valid   :   A logical indication of whether the qara is a     *
*                         valid ratio                                   *
*         qara    :   The ratio of the step size to the amplitude of    *
```

```
*                       the sinusoidal input                          *
*       mrec    :   The reciprocal of m                               *
*       pi      :   The usual constant                                *
*       piov2   :   The value of pi divided by 2                      *
*       ssum    :   The iterated sum of square roots used as a        *
*                       test quantity                                 *
*       coqty   :   The qunatity derived from ssum which is compared  *
*                       to piov2                                      *
*       m       :   The number transitions between quantization       *
*                       levels in the positive (or negative) non-     *
*                       zero range                                    *
*       ind     :   An index used in the process                      *
************************************************************************
      subroutine testra(q,a,noofbt,valid)
          real q, a, qara, mrec, pi, piov2, ssum, coqty
          integer noofbt, m, ind
          logical valid
          valid = .true.
          qara = q/a
          m = 2**(noofbt - 1) - 1
          mrec = 999999
          if (m .ne. 0) mrec = 1./m
          if (qara .gt. mrec) then
              write(6,*) 'The q to A ratio is too large!'
              valid = .false.
          else
              pi = 3.1416
              piov2 = pi/2.
              ssum = 0.
              do 10 ind = 1, m
                  ssum = ssum + sqrt(1. - (ind * qara)**2)
   10         continue
              coqty = qara * (1 + 2. * ssum)
              if (coqty .lt. piov2) then
                  write(6,*) 'The q to A ratio is not appropriate!'
                  valid = .false.
              end if
          end if
      end
************************************************************************
*   SUBROUTINE DTAKSI                                                  *
************************************************************************
*     This subroutine determines all values of a sub k for k = 0       *
*     through k = 10,000.  It also produces the constant b, which,     *
*     along with the a sub k's, is necessary to determine the          *
*     quantization noise spectrum.                                     *
************************************************************************
*     Variables:                                                       *
*         aka     :   The array which ultimately contains all of the   *
*                         desired a sub k's                            *
*         qara    :   The ratio of the step size to the amplitude of   *
```

```
*                       the random sinusoidal input                    *
*       q       :   The quantization step size                         *
*       m       :   The number transitions between quantization        *
*                       levels in the positive (or negative) non-      *
*                       zero range                                     *
*       k       :   The parameter which indicates the desired          *
*                       a sub k prime                                  *
*       valid   :   A logical indication of whether the qara is a      *
*                       valid ratio                                    *
************************************************************************
      subroutine dtaksi(q,a,noofbt,aka,b)
          real q, a, aka(0:10000), b, pi, qara, sumex
          integer noofbt, m, k, ind
          pi = 3.1416
          qara = q/a
          m = 2**(noofbt - 1) - 1
          do 10 k = 0, 10000
              call claksi(aka(k),qara,m,k,q)
   10     continue
          sumex = 0.
          do 20 ind = 1, m
              sumex = sumex + sqrt(a**2 - (ind * q)**2)
   20     continue
          b = (a**2)/2. - 4. * q * (sumex + a/2.)/pi
      end
************************************************************************
*   SUBROUTINE CLAKSI                                                  *
************************************************************************
*     This subroutine calculates the exact value of a sub k for k      *
*     such that 0 <= k <= 10000.  Larger values of k will result in    *
*     negligible values.                                               *
************************************************************************
*     Variables:                                                       *
*       ak      :   The desired value a sub k                          *
*       qara    :   The ratio of the step size to the amplitude of     *
*                       the random sinusoidal input                    *
*       m       :   The number transitions between quantization        *
*                       levels in the positive (or negative) non-      *
*                       zero range                                     *
*       k       :   The parameter which indicates the desired          *
*                       a sub k                                        *
*       q       :   The quantization step size                         *
*       pi      :   The standard constant                              *
*       msum    :   The total evaluation of the summation term         *
*       polarg  :   The iterated argument of the Tchebycheff          *
*                       polynomial in the summation term of the        *
*                       appropriate a sub k equation                   *
*       polqty  :   The iterated evaluation of the Tchebycheff         *
*                        polynomial in the summation term of the       *
*                        appropriate a sub k equation                  *
*       squqty  :   The squared quantity which includes all terms      *
```

```
*                        involving Tchebycheff polynomial evaluations *
*         dukpl1  :   The value corresponding to 2k + 1                *
*         i       :   An index used in the process                     *
**************************************************************************
      subroutine claksi(ak,qara,m,k,q)
          real ak, qara, q, pi, msum, polarg, polqty, squqty
          integer m, k, dukpl1, i
          pi = 3.1416
          dukpl1 = 2 * k + 1
          msum = 0.
          do 10 i = 1, m
              polarg = i * qara
              call evuk(dukpl1,polarg,polqty)
              msum = msum + polqty
   10     continue
          polarg = 0.
          call evuk(dukpl1,polarg,polqty)
          squqty = (polqty + 2. * msum)**2
          ak = 2. * ((q/(pi * dukpl1))**2) * squqty
      end
**************************************************************************
*   SUBROUTINE EVUK                                                      *
**************************************************************************
*     This subroutine evaluated the Tchebycheff polynomial of the        *
*     second kind, U sub k of the argument, given the argument and       *
*     the degree, k, of the polynomial.                                  *
**************************************************************************
*     Variables:                                                         *
*         k       :   The parameter which indicates the desired        *
*                         a sub k                                        *
*         polarg  :   The argument of the polynomial                   *
*         polqty  :   The evaluation of the polynomial for the         *
*                         argument                                       *
**************************************************************************
      subroutine evuk(k,polarg,polqty)
          real polarg, polqty
          integer k
          polqty = sin(k * acos(polarg))
      end
```

Bibliography

1. Apostol, Tom M. *Mathematical Analysis* (Second Edition). Reading MA: Addison-Wesley, 1974.
2. Barrett, J. F. and D. G. Lampard. "An Expansion for Some Second-Order Probability Distributions," *IRE Transactions on Information Theory, IT-1*: 10–15 (January 1955).
3. Bennett, W. R. "Spectra of Quantized Signals," *Bell Systems Technical Journal, 27*: 446–472 (July 1948).
4. Bussgang, Julian J. "Crosscorrelation Functions of Amplitude-Distorted Gaussian Signals," *Massachusetts Institute of Technology Research Laboratory of Electronics Technical Report No. 216* (March 26, 1952).
5. Clarke, A. Bruce and Ralph L. Disney. *Probability and Random Processes for Engineers and Scientists.* New York: John Wiley and Sons, 1970.
6. Gaskill, Jack D. *Linear Systems, Fourier Transforms and Optics.* New York: John Wiley and Sons, 1978.
7. Gradshteyn, I. S. and I. M. Ryzhik. *Tables of Integrals, Series, and Products* (Second Edition). New York: Academic Press, 1980.
8. Gray, Glenn A. and Gene W. Zeoli. "Quantization and Saturation Noise Due to Analog-to-Digital Conversion," *IEEE Transactions on Aerospace and Electronic Systems, AES-7*: 222–223 (January 1971).
9. Lever, K. V. "Quantising Noise Spectra," *Mathematical Topics in Telecommunications,* Volume 2, edited by Kenneth W. Cattermole and John J. O'Reilly. New York: John Wiley and Sons, 1984.
10. Lu, Fu-Sheng and Gary L. Wise. "A Simple Approximation for Minimum Mean-Square Error Symmetric Uniform Quantization," *IEEE Transactions on Communications, COM-32*: 470–474 (April 1984).
11. Max, Joel. "Quantizing for Minimum Distortion," *IRE Transactions on Information Theory, IT-6*: 7–12 (March 1960).
12. Pearlman, William A. and George H. Senge. "Optimal Quantization of the Rayleigh Probability Distribution," *IEEE Transactions on Communications, COM-27*: 101–112 (January 1979).
13. Robertson, G. H. "Computer Study of Quantizer Output Spectra," *Bell Systems Technical Journal, 48*: 2393–2403 (September 1969).
14. Roden, Martin S. *Digital Communication Systems Design.* Englewood Cliffs NJ: Prentice Hall, 1988.

15. Shanmugan, K. Sam and A. M. Breipohl. *Random Signals: Detection, Estimation and Data Analysis.* New York: John Wiley and Sons, 1988.

16. Szegö, Gábor. *Orthogonal Polynomials.* New York: American Mathematical Society, 1939.

17. Thomas, John B. *Statistical Communication Theory.* New York: John Wiley and Sons, 1969.

18. Velichkin, A. I. "Correlation Function and Spectral Density of a Quantized Process," *Telecommunications and Radio Engineering. Part II: Radio Engineering, 17*: 70–77 (July 1962).

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1990	Master's Thesis

4. TITLE AND SUBTITLE
QUANTIZATION NOISE CHARACTERISTICS RESULTING FROM GAUSSIAN, NEGATIVE-EXPONENTIAL, AND SINUSOIDAL RANDOM INPUT SIGNALS

5. FUNDING NUMBERS

6. AUTHOR(S)
Van N. Osborne, Captain, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GE/ENG/90D-48

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The purpose of this study was to investigate the trade-off between the number of quantization levels and the resulting noise characteristics for three classes of commonly occurring input signals, namely, those signals possessing Gaussian, negative-exponential and random sinusoidal distributions. This study derived expressions for the mean-squared error, output spectrum and error spectrum by expanding the nonlinear quantization function into a summation of orthogonal polynomials matched to the corresponding input signal distribution. Once accomplished, orthogonality properties were applied to provide usable expressions. A set of three Fortran 77 programs were developed – each of which applied to one of the studied input signal classes. The appropriate program produced upon demand either a mean-squared error value and a signal-to-quantization noise ratio or quantizer output spectrum data and quantization error spectrum data. Typical input power spectral densities were applied in order to produce the spectra data. The study resulted in a set of tables which provided mean-squared error and signal-to-quantization noise ratio data based on various numbers of bits used for the quantization process. Also, a number of plots displaying the power spectral densities under consideration were produced as based on similar numbers of bits.

Keywords: Theses

14. SUBJECT TERMS	15. NUMBER OF PAGES
Quantization, Noise, Spectra, Power Spectra, Probability Density Functions, Digital Communications (JN)	117
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL