QUANTIZATION NOISE IN A DIGITAL SIGNAL PROCESSOR
FOR SYNTHETIC APERTURE RADARS

by

George Michael Blaszczyinski

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

January 1973

Signature of Author

Department of Electrical Engineering, January 24, 1973

Certified by

Thesis Supervisor

Accepted by

Chairman, Departmental Committee on Graduate Students
QUANTIZATION NOISE IN A DIGITAL SIGNAL PROCESSOR
FOR SYNTHETIC APERTURE RADARS

by

George Michael Balszczynski

Submitted to the Department of Electrical Engineering on January 24, 1973 in partial fulfillment of the requirements for the degree of Master of Science.

ABSTRACT

In a synthetic-aperture radar equipped with a digital signal processor the received signal is quantized and stored in a large computer memory. The subsequent processing may be partially done in a separate arithmetic memory with higher precision (greater number of bits per word). A major operation in the image computation is the Fast Fourier Transform (FFT). This paper contains an analytic description of noise injected into the radar image by (1) the FFT algorithm performed in the arithmetic memory and (2) the initial quantization due to finite precision of the signal memory. The advantages in reducing the noise derived from the freedom to choose different precisions of (1) the signal memory and (2) the arithmetic memory, are considered. The advantages may influence major design decisions.

THESIS SUPERVISOR: Godfrey T. Coate

TITLE: Lecturer - Department of Electrical Engineering
DEDICATION

To My Parents,

whose courageous decision to immigrate to the United States has opened a world of opportunity to me.
ACKNOWLEDGEMENT

This thesis was made possible through the support extended to the Electronic Systems Laboratory of Massachusetts Institute of Technology by the United States Air Force under contract No. F33615-72-C-1503.

Special acknowledgements are extended to Mr. Godfrey Coate whose helpful suggestions and critical reading of the first draft have greatly contributed to this research effort. I also wish to thank Dr. Arthur Lewis for his comments on some of the problems encountered during this research.

The difficult task of typing the report, with its many mathematical expressions, was performed by Mary George, whose cooperation and positive attitude was greatly appreciated. The illustrations were drawn by Arthur Giordani.
# TABLE OF CONTENTS

LIST OF FIGURES ................................. page 7

LIST OF IMPORTANT SYMBOLS .................. 8

CHAPTER I INTRODUCTION ....................... 14

1. Problem Description .......................... 14
2. Background .................................. 15
3. Derivation of Model ......................... 17

CHAPTER II MODEL DESCRIPTION ............... 20

1. List of Variables ............................ 20
2. Quantizer Noise ............................. 20
3. Fast Fourier Transform ..................... 21
4. Gaussian Image ............................. 24

CHAPTER III QUANTIZER NOISE ................ 26

1. Problem Statement ........................... 26
2. Derivation of Characteristic Function ..... 30
3. Derivation of Mean-Square Error .......... 35
4. Extension to Vertical Quantization ........ 37

CHAPTER IV PROPERTIES OF FFT ALGORITHM .. 40

1. Matrix Representation ....................... 40
2. Important Properties ....................... 52
3. Bounds on Largest Magnitude ............... 56

CHAPTER V NOISE INJECTED BY FFT ALGORITHM 64

1. Noise Vector ............................... 64
2. Derivation of Error Variance ............... 78
3. Derivation of Error Bias .................... 87
LIST OF FIGURES

1. Flow Chart of the Model  
   page 19

2. Quantization Schemes  
   27

3. Ranges of Expression (74)  
   67

4. Distribution of the Truncation Error  
   71

5. Example of a Butterfly Operation  
   73

6. Signal Flow in Decimation in Time FFT for 8-Point Transform  
   76

7. Graph of $f(\sigma_x)$ for Gaussian Distribution  
   108

8. Function $f(\sigma_x)$ for $N = 2^6$ and $N = 2^{10}$  
   110

9. Probability Density Function of $\sigma_r$  
   114

10. Probability Density Function of $\sigma_x$  
    114

11. $f(\sigma_x)$ for Poisson and Gaussian Distributions  
    116

12. Schematic Graph of $f(\sigma_x)$  
    118

13. Allowable Regions of $\omega \eta$  
    154

14. Largest $\text{Re}(\omega \eta)$ for Given $\alpha$  
    154

15. Terms in $\text{Re}(\xi_1)/H_{k-2}$  
    155
LIST OF IMPORTANT SYMBOLS

Symbols listed below have the same meaning throughout the entire thesis. Many other symbols are used in the specific derivations, but their meaning is local, defined inside the derivation. The list below is divided into two sections: (1) the list of subscripts and superscripts, and (2) the list of symbols.

The list of subscripts and superscripts

av Subscript, designating a result associated with the white Gaussian image.

c Superscript to a column matrix, forms another column matrix related to the original by (5).

f Superscript, designates a result associated with the decimation in frequency.

i Subscript, designating the imaginary part.

k Subscript, referring to the k'th stage of the FFT.

o Superscript to a complex column matrix, forms another column matrix with real elements, related to the original by (93).

Superscript to a complex square matrix, forms another square matrix with real elements, related to the original by (94).

r Subscript, designating the real part.

t Superscript, designates a result associated with the decimation in time.

T Superscript to a matrix, denotes its transpose.
Subscript to a column matrix, forms another column matrix whose elements constitute a bit-reversed sequence of the elements from the original matrix.

* Superscript, means complex conjugate.

* Subscript, designating a variable normalized with respect to the peak signal.

i, j, k, l, p Subscripts mean i'th, j'th, ... element.

The list of symbols

a Mean of \( p_n(n) \) distribution shown in Fig. 4

\[ a = \left( \frac{-p - 1}{2} \right) (1 + j). \]

\( \bar{a}_k \) Mean of the noise vector, \( \bar{a}_k = \mathbb{E}(n_k) = 2^{-s_k} a \).

b Processor error bias vector

\[ b = \sum_{k=1}^{M} b_k = \sum_{k=1}^{M} \mathbb{E} \{ \varepsilon_{pk} \} \]

\( \bar{b} \) Normalized bias, \( \bar{b} = 2^{-s_M} b \)

\( b_0 \) Largest magnitude point of the bias \( \bar{b} \).

\( B_k \) Matrix of Fourier coefficients defined in (45).

\( e_k \) 2N x 1 column matrix with one in the k'th element and zeroes elsewhere.

\( \text{erfc}(t_0) \) Error function, \( \text{erfc}(t_0) = \int_{t_0}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt \)

\( \mathbb{E} \) Denotes the expected value.

\( E_b \) Energy of bias, \( E_b = \bar{b}^T \bar{b} \).

\( f(\sigma_x) \) Function schematically drawn in Fig. 12

\[ f(\sigma_x) = \mathbb{P}_x \{ 1/2 < H_0 < 1 \mid \sigma_x \} \]

\( F \) Fourier coefficients matrix given by (6).
\( F(\cdot) \)  
Discrete Fourier Transform defined in (3).

\( F_M(\cdot) \)  
Discrete Fourier Transform defined in (29).

\( H_k \)  
Largest absolute value of either real or imaginary part of all elements of \( z_k \),

\[
H_k = \max\{|\zeta_{1r}|, \ldots, |\zeta_{Nr}|, |\zeta_{1i}|, \ldots, |\zeta_{Ni}|\}
\]

\( I \)  
Identity matrix.

\( M \)  
Number of stages of FFT, \( N = 2^M \)

\( M_e(v_1, v_2) \)  
Joint characteristic function of the \( k \)'th element of \( y^c \) with the \( k \)'th element of \( y'^c \), given by (19).

\( M_y(v) \)  
Characteristic function of \( y^c \)

\[
M_y(v) = \mathbb{E}\{\exp(jv^T y^c)\}
\]

\( n \)  
Truncation and shift error.

\( n \)  
2N x 1 column matrix whose elements are integers.

\( n_k \)  
Processor noise vector at the \( k \)'th stage, given by (86).

\( n_s \)  
Shift error.

\( n_t \)  
Truncation error.

\( N \)  
Number of points (samples) in the image or in the signal.

\( N\{m, \Lambda\} \)  
Gaussian distribution with mean \( m \) and covariance matrix \( \Lambda \).

\( P_{\text{subscript}}(\cdot) \)  
Probability density function for variable or variables specified in the subscript.

\( P_k \)  
Signal power after the \( k \)'th stage \( p_k = z_k^T z_k^* \).
\( p_i, p_p, p_s \) Number of bits per word of image memory, processor (arithmetic) memory, and signal memory respectively.

\( p_d, p_n, p_o, p_b, p_m, p_r \) are defined in (239).

\( p_k \) \( N \times N \) complex matrix of Fourier coefficients in the \( k \)'th stage of FFT. Superscripts \( t \) and \( f \) denote the decimation in time and frequency respectively.

\( Pr(\{\}) \) Probability of the event specified inside the brackets.

\( q \) Quantization step size, \( q = 2^{-p_s} \)

\( Q(\()) \) Represents analog to digital quantization.

\( R \) Ratio of r.m.s. signal to r.m.s. noise.

\( s_k \) Value of the characteristic after the \( k \)'th stage of FFT.

\( \text{sinc} (\alpha) \) \( \text{sinc} (\alpha) = \frac{\sin \pi \alpha}{\pi \alpha} \), where \( \alpha \) is a real number.

\( S(\alpha) \) \( S(\alpha) = \prod_{k=1}^{2N} \text{sinc}(\frac{\alpha}{\omega}) \).

\( t(t_0) \) Tail of the input signal distribution

\( t(t_0) = \int_{t_0}^{\infty} [p_x(\tau) + p_x(-\tau)] \, d\tau \)

\( u_o(\()) \) Unit impulse.

\( u_k \) Processor noise vector related to \( n_k \) by:

\( u_k = n_k - E\{n_k\} \).

\( v \) Final processor error vector related to \( \varepsilon_{pk} \) by:

\[ v = \sum_{k=1}^{M} v_k = \sum_{k=1}^{M} \varepsilon_{pk} - E\{\varepsilon_{pk}\} \]

\( v_\ell \) Fourier coefficient \( v_\ell = \exp[-j(\ell-1)2\pi/2^k] \)
Normalized variance $V(\ ) = 2^{-2S_M} \text{Var}(\ )$.

Variance of variable in parentheses.

$(2^{k-1}) \times (2^{k-1})$ diagonal complex matrix of Fourier coefficients defined in (45).

$X$

$N \times 1$ complex column matrix representing signal.

$X_q$

Quantized $x$, $x_q = Q(x)$.

$Y$

$N \times 1$ complex column matrix representing image.

$Y'$

Fourier transform of $x_q$, $Y' = F\{x_q\}$

$z_k$

$N \times 1$ complex column matrix representing the signal after the completion of the $k$'th stage of FFT, $z_k = F_k \ldots F_1 x_0$.

$z_0$

Signal before the start of FFT. For decimation in time $z_0 = x_\Delta$, for decimation in frequency $z_0 = x$.

$a_k$

$2N \times 1$ column matrix formed from the $k$'th row of $F$, $a_k = F^T e_k$.

$b_k$

$k$'th column of $F$, $b_k = F e_k$.

$Y_k$

$N \times 1$ column matrix given by (126).

$\delta(\ )$

Dirac delta function, unit impulse.

$\epsilon$

Final image error.

$\epsilon_p$

Real or imaginary part of any element of $\epsilon_p$.

$\epsilon_q$

Real or imaginary part of any element of $\epsilon_q$.

$\tilde{e}_p$

Processor truncation error vector

$$\tilde{e}_p = \sum_{k=1}^{M} \epsilon_{pk}.$$

$\epsilon_q$

Initial quantization error vector, $\epsilon_q = Y - Y'$.

$\epsilon_{pk}$

$k$'th stage contribution to $\epsilon_p$ related to $n_k$ by (88).
\( z_k \)

An element of \( z_k \).

\( \eta_i \)

An element of \( z_{k-1} \).

\( \lambda^{\text{superscript}}_{mn} \)

The \((m,n)\)'th element of \( A^{\text{subscript}} \) where variable in superscript is the same as in subscript.

\( A^{\text{subscript}} \)

Covariance matrix for variable specified in the subscript.

\( \sigma^{\text{subscript}} \)

Standard deviation of variable specified in the subscript.

\( \sigma_n \)

Standard deviation of \( p_n(n) \) distribution shown in Fig. 4 for \( s_k = 0 \)

\[
\sigma_n = \frac{1}{12} 2^{-2p_p}.
\]

\( \sigma_r \)

Standard deviation of the received signal.

\( \sum_{n} \)

This symbol denotes a repeated sum for \( 2N \) indices varying from \(-\infty\) to \(+\infty\), i.e.

\[
\sum_{n} (\ldots) = \sum_{n_1=-\infty}^{\infty} \cdots \sum_{n_{2N}=-\infty}^{\infty} (\ldots).
\]

\( \omega \)

(1) Quantizing frequency, \( \omega = \frac{2\pi}{q} \).

(2) Fourier coefficient, an element of \( W_k \).

\( \mathbf{1} \)

Column matrix whose elements are all equal to one.

\( i, j, k, l, m, n, p \)

are integers.
1.1 Problem Description

In a digital signal processor for Synthetic Aperture Radar (SAR) (refer to the next section for background) the sampled signal from the receiver is stored in a large digital memory. To compute the image (radar map) the processor correlates the received signal with an appropriate weighting function. For an on-line real-time application the computation time of an image has an upper limit of the order of one second, and only those algorithms which fit within this limit can be used. The scheme considered in this paper involves only point-to-point operations and Fast Fourier Transforms (FFT). Some details of this overall algorithm are described in a later section, but any issues involving justification or necessary approximations required for the implementation of this algorithm are not part of this paper.

The FFT algorithm at any time operates only on a small section of the stored signal. This fact allows a construction of a processor consisting of a large signal memory, and a small arithmetic memory which can be loaded with different sections of the signal. In the smaller memory the FFT operation and various other arithmetic operations can be performed. Such implementation offers various speed, cost and accuracy advantages. In this paper, the advantages derived from the freedom to choose different precisions (number of bits per word) for
signal memory and arithmetic memory, and from different FFT algorithms are discussed. These advantages may influence major design decisions.

1.2 Background

In conventional radars a short radio wave pulse is transmitted and echoes from that pulse are recorded. The intensity of echo at any particular time measures the reflectivity of objects a certain distance away. In the synthetic aperture radar* a periodic sequence of pulses is transmitted and echoes from all those pulses are recorded. To construct a "synthetic aperture" the radar antenna travels a path with respect to the ground, which has a component perpendicular to its beam. Since the transmitted pulses are short bursts of constant carrier frequency, the recorded signal contains not only information about the reflectivity of objects, but also relative phase differences corresponding to different positions of the antenna.** This phase information is used to obtain radar maps of much better azimuth resolution than in ordinary radars where azimuth resolution is determined by the antenna beam width.

The signal processing in SAR consists of the following: First, the signal is shifted down in frequency and adjusted in phase to correct for the deviations in the antenna motion from a smooth reference path. Next, the signal is sampled, quantized, and often pre-filtered in the

---

*For more complete description of SAR principles, see References 3, 8.

**In actual radar, more elaborate modulation can be used, see References 5, 11.
azimuth direction to eliminate undesirable higher frequencies and
decrease the number of samples. Finally it is stored in the memory
as a matrix whose columns contain the sampled echoes from individual
pulses.

Subsequent processing reduces this signal to the final map in
the form of a two-dimensional array. Ideally, for each map point,
the entire stored signal should be correlated with a reference func-
tion corresponding to that point. This reference function is equal
to the signal which would have been received if only one scatterer of
unit reflectivity had been placed in the area mapped in the location
corresponding to that map point. This unfortunately, is too long a
computation for a real-time digital computer. A more practical solution
is to approximate the signal as the kernel of a one-dimensional Fresnel
transform. In the real applications, several such transforms corres-
dponding to sections of image or sections of signal may be necessary.
The inverse transform can be quickly computed by two element-by-
element matrix multiplications and a one-dimensional Fourier transform,
accomplished with FFT algorithm applied to the rows of the signal
matrix. Sometimes, the lines of constant range do not correspond to
the rows of the signal matrix and a small rotation had to precede the
FFT operations. In addition, to control the ringing and other distor-
tions in the image, an appropriate weighting function is applied before
the signal is transformed.
1.3 Derivation of Model

A simple way of studying the noise characteristics of the processor analytically is to assume an independent additive noise. Under this assumption, the noise is modeled as simply an algebraic sum of noise contributions from individual parts of the system. Each contribution is calculated with an assumption that all other parts are noiseless. This approach is valid only if the final distortions in the image are small. Since this is the objective of the design, the additive noise assumption is a reasonable approximation.*

The purpose of this paper is to analyze the noise injected by the initial quantization of the signal and by the FFT algorithm. The noise contributions from other parts of the system are assumed negligible. This simplifies the mathematical model. An FFT operation is applied to each row of the signal matrix separately. If the statistical properties of the image vary little from row to row, then only one representative row need be analyzed. Therefore, the model reduces to noise calculation for an N-point one dimensional image, which is quantized in the frequency domain and transformed with an FFT algorithm. The resulting noise depends on the statistical properties of the image. The model can be qualitatively represented by the flow-chart diagram shown in Fig. 1. The "continuous image" represents the desired image, which the processor tries to reconstruct. The "continuous signal"

*For various approaches in analyzing noise, see References 10, 14, 15, 16, 17, 19, 20.
corresponds to the received signal, which is quantized and stored as "quantized signal", from which the FFT operation reconstructs the image. In the final step, the resulting image is compared with the original.
Fig. 1 Flow Chart of the Model
CHAPTER II

MODEL DESCRIPTION

2.1 List of Variables

The noise injected into the image, in the model described above, depends on:

'precision of signal memory
'quantization method
'precision of arithmetic memory
'FFT algorithm used
'precision of Fourier coefficients
'truncation after arithmetic operations
'precision of image memory
'statistics of image

2.2 Quantizer Noise

In the calculation of noise originating from the initial quantizations of the signal, the quantization method assumed has uniform step size and infinite extent, i.e., no clipping levels.* In addition, the independent additive noise assumption is invoked, by considering the FFT operation to be perfectly accurate. This procedure simplifies the problem to derivation of the mean-square error of an image quantized

*For discussion of alternative quantization methods, see References 4, 12, 18, 21.
in the frequency domain. The error depends on the statistical properties of the image. The resulting expression for the mean square error is given in terms of the joint characteristic function of the image points.

2.3 The Fast Fourier Transform

The FFT algorithm computes a discrete Fourier transform of a sequence of \( N = 2^M \) (\( M \) = integer) point samples. Algorithms for different \( N \) also exist, but they are less frequently used.* The calculation is carried out in \( M \) stages - in each stage the points are accessed only once. The basic element operation of the algorithm is called a "butterfly". It consists of several multiplications and additions operating on two (for \( N = 2^M \)) points of the sequence. Depending on the form of the butterfly operation, there are two distinct FFT algorithms, called "decimation in time" and "decimation in frequency." The important difference between these algorithms occurs when the computation is done in-place. Then the input for the decimation in time, and the output for the decimation in frequency has to be in the "bit-reversed" form. The bit-reversed form is a special ordering of the elements in the sequence, which together with the butterfly operation is described in detail in section 4.1.

The structure of the FFT assumed in this paper is the "in-place" algorithm with \( N = 2^M \) points. This structure covers a large number of

*For more information on transform techniques, see References 2, 6 7, 9, 13.
applications; moreover, extensions of the results to the other structures are straightforward.

The number representation frequently used in digital signal processing is the block-floating-point representation. All numbers are stored in the floating-point format with a common characteristic, i.e., each number is expressed as:

\[(\text{number}) = (\text{mantissa}) \times 2^{\text{characteristic}}\]

where the mantissa is a fraction of magnitude smaller than one, and the characteristic is the same for all the numbers. This representation decreases the size of the memory needed to store an array of numbers, because the characteristic does not have to be stored with each number separately, and reduces the arithmetic operations to fixed-point additions and multiplications. The major complication occurs when the characteristic changes and all mantissas must be shifted to the right or left by an appropriate number of bits. Another difficulty is associated with detecting the need for lowering the characteristic, but this difficulty is not encountered in FFT, as is shown in section 4.3.

In the derivation of the noise contribution of the FFT algorithm, the block-floating-point representation is assumed, with two's complement format used as an example.

At each stage of the FFT, all points of the signal sequence are processed with the butterfly operation. Suppose that the arithmetic operations of the butterfly are carried out with infinite precision and the results truncated to fit the words of arithmetic memory. In general, those results will include irrational numbers (because the
butterfly involves transcendental numbers, e.g. \( \exp \left( \frac{\pi}{4} \right) \); therefore, the truncated parts are not zero. If the signal is considered to be an \( N \)-dimensional vector, then the difference between the signal before truncation and after truncation can be viewed as a noise vector injected into the processing at the given stage. The total noise can thus be modeled as the overall effect of \( M \) noise vectors added to the signal in the corresponding stages of the FFT computation.

In the actual implementation of the butterfly, the infinite precision cannot be attained, but a good approximation can be achieved by using registers several bits longer than the words of arithmetic memory. The neglected butterfly noise is small, but if greater accuracy of the analysis is desired, then this noise can be incorporated into the truncation noise vectors. In fact, the general case can be treated by adjusting the parameters of the truncation noise vectors. In this paper, however, for clarity and simplicity, only the final truncation noise is considered. This is a reasonable approximation in studying the primary effects and in the relation to other approximations.

The butterfly operation uses the coefficient \( \exp \left( \frac{2\pi}{N}k \right) \), where \( k \) is an integer and \( N \) is the number of signal samples. In evaluation of this coefficient a table of the sine function may be used. The number of entries in this table may be smaller than \( N \), without degrading the overall accuracy. This is accomplished by linear interpolation between the available points, provided that their precision is sufficiently high. The smoothness of the sine function assures the validity of such method, and the overall result is an increase in the accuracy of
coefficients, and a decrease in the size of the memory needed to store the sine table. The actual trade-offs of this implementation are not considered in this paper. It is simply assumed that the coefficients can be obtained with arbitrary accuracy.

The conclusion of the last few paragraphs is that the noise injected by the FFT can be modeled by noise vectors added to the signal after each stage of the algorithm. If the noise vectors are much weaker than the signal, then the total noise can be assumed to be an algebraic sum of contributions from individual noise vectors. Furthermore, those noise vectors can be assumed to be random, because their elements vary considerably even for very small changes in the signal. Using this probability approach, the error mean and variance vectors can be calculated.

In Chapter V, the noise vector is derived for images having no finite regions with zero energy in either the spatial or frequency domain. This last assumption makes the noise vectors for strong signals independent (outside of the scale factor) of the signal and of each other, and does represent a normal SAR situation. Using those vectors and some deterministic properties of the FFT, the expressions and bounds for the error mean and variance are derived.

2.4 Gaussian Image

The error expressions derived in Chapter V depend on the values of the characteristic of the block-floating-point representation at the different stages of the FFT. To obtain some concrete results, the
distribution for the characteristic must be known. This distribution can be derived from the probabilistic description of the image. Such a description, which is applicable to SAR, and simple enough for analytic treatment, consists of independent and identically distributed image sample points. The magnitude of each point has a Rayleigh distribution and the phase has a uniform distribution — that is, the real and imaginary components are independent zero-mean Gaussian random variables.*

Using the above image model, the quantizer noise is first derived. Next, the precision requirement for accurate storage of the signal in a block-floating-point memory, and the effect of clipping levels, are discussed. Finally, various shifting policies are considered, and the error mean and variance for the FFT processor are calculated. In conclusion, these results are used in deriving approximate equations for the number of bits needed in the signal and in the arithmetic memory.

*Refer to section 6.1.
CHAPTER III

QUANTIZER NOISE

3.1 Problem Statement

This chapter contains the derivation of an expression for the average quantization noise due to the finite number of bits in the words of the signal memory. Mathematically, this average quantization noise can be measured by the mean-square error of an N-point image sequence quantized in the frequency (i.e., signal) domain. The results obtained depend on the statistical description of the image, and the statistical description used below is the joint characteristic function of all the image points.

Let $\mathbf{y}$ be an $N$-dimensional complex random vector representing the image, and let $\mathbf{x} = F^{-1}(\mathbf{y})$ be the corresponding signal, where $F\{\}$ represents the discrete Fourier transform and $F^{-1}\{\}$ is its inverse. Let $\mathbf{x}_q = Q\{\mathbf{x}\}$ be the quantized signal, with $Q\{\}$ denoting the quantizer operation, and finally, let $\mathbf{y}'$ be the Fourier transform of $\mathbf{x}_q$.

$$\mathbf{y}' = F(Q(F^{-1}(\mathbf{y})))$$

(1)

Then the mean-square error vector $\mathbf{e}_q^2$ can be simply written as:

$$\mathbf{e}_q = \mathbf{y} - \mathbf{y}'$$

(2a)

$$\mathbf{e}_q^2 = \mathbb{E}\{\text{diag}(\mathbf{e}_q \mathbf{e}_q^T)\}$$

(2b)

-26-
where vectors $\mathbf{y}$ and $\mathbf{y}'$ are considered to be column matrices, superscript $T$ denotes the transpose, and $E\{ \}$ means the expected value.

The quantization method assumed in this paper has uniform intervals between reference levels, is symmetric with respect to negative values, and extends in the range from $-\infty$ to $+\infty$. Depending on the value of the quantizer (analog-to-digital converter) output for zero input, two quantization schemes are possible. These schemes are shown in Fig. 2, and for convenience, are labeled: "vertical quantization" and "horizontal quantization". Extending the analysis below to non-uniform quantization or to quantization with clipping levels is difficult.

(a) Horizontal quantization  
(b) Vertical quantization

Fig. 2. Quantization schemes
The discrete Fourier transform is defined by the following equations:

\[ y_k = \frac{1}{\sqrt{N}} \sum_{n=1}^{N} (e^{-2\pi j/N})^{nk} x_n \quad (3a) \]

\[ x_n = \frac{1}{\sqrt{N}} \sum_{k=1}^{N} (e^{-2\pi j/N})^{-nk} y_k \quad (3b) \]

Symmetric normalization was chosen for simplicity. Another way of expressing the discrete Fourier transform is by writing it as a matrix multiplication:

\[ y^c = F \cdot x^c \quad x^c = F^{-1} \cdot y^c \quad (4) \]

To avoid dealing with complex random variables, the real and imaginary parts of \( x \) and \( y \) are treated separately.

\[ x^c = \begin{bmatrix} x_r \\ x_i \end{bmatrix} \quad y^c = \begin{bmatrix} y_r \\ y_i \end{bmatrix} \quad (5) \]

where \( x_r \) and \( y_r \) are real parts,
\[ x_i \] and \( y_i \) are imaginary parts.

The \( F \) matrix is therefore a \( 2N \) by \( 2N \) real matrix given by:

\[ F = \frac{1}{\sqrt{N}} \begin{vmatrix} F_r & -F_i \\ F_i & F_r \end{vmatrix} \quad (6a) \]

\[ \{F_r\}_{ij} = \cos((2\pi/N)ij) \]
\[ \{F_i\}_{ij} = -\sin((2\pi/N)ij) \quad (6b) \]
Finally, the error vector is redefined to be:

\[ \mathbb{E}_q^2 = \mathbb{E} \{ \text{diag}(y - y')^c (y - y')^c_T \} \]  
(7)

It can also be easily shown that the inverse transform is given by:

\[ F^{-1} = F^T \]  
(8)

Let \( M_y(y) \) be the characteristic function of \( y^c \) given by:

\[ M_y(y) = \mathbb{E} \{ \exp(jy^T y^c) \} \]  
(9)

The mean-square error vector is derived in terms of \( M_y(y) \). Below is the outline of steps to be taken in deriving \( \mathbb{E}_q^2 \).

a) Derive the joint characteristic function of \( x^c \) and \( y^c \),

\[ M_{xy}(x, y) \]

b) Derive the joint characteristic function of \( x_q^c \) and \( y^c \),

\[ M_{x_qy}(x, y) \]

c) Derive the joint characteristic function of \( y^c \) and \( y^c \),

\[ M_{y_y}(y, y) \]

d) Derive the joint characteristic function of the \( k' \)th element of \( y^c \) and the \( k' \)th element of \( y^c \),

\[ M_{y_k y'_k}(y_1, y_2) = M_c(y_1, y_2) \]

e) Differentiate \( M_c(y_1, y_2) \) and set the arguments equal to zero to derive \( \mathbb{E}_q^2 \)

The method used in step (b) is an extension of results derived by Roig and Widrow.
The error $\frac{2}{q}$ is first derived for the horizontal quantization. The results are later extended to the vertical quantization.

3.2 Derivation of Characteristic Function

The $x^c$ and $y^c$ vectors are related through a reversible linear transformation. Therefore, their joint characteristic function is simply:

$$M_{xy}(\nu_x, \nu_y) = E\{\exp(j\nu_x^T x^c + j\nu_y^T y^c)\}$$

$$= E\{\exp(j\nu_x^T F^{-1} y^c + j\nu_y^T y^c)\}$$

$$= E\{\exp(j(\nu_x + \nu_y)^T y^c)\}$$

and from the definition of $M_y(\nu)$:

$$M_{xy}(\nu_x, \nu_y) = M_y(\nu_x + \nu_y) \quad (10)$$

Let $p(\eta, \xi)$ be the joint probability density function of $x^c$ and $y^c$ and let $p_q(\eta, \xi)$ be the joint probability density function of $x_q^c$ and $y^c$. As a function of $\eta_k$, corresponding to the $k$'th element of $x_q$, the function of $p_q(\eta, \xi)$ consists of impulses located at the values representing the individual quantization intervals

$$(n - 1/2)q < \eta_k \leq (n + 1/2)q$$

where $n$ is an integer. In the analysis below, these values are assumed to be at $nq$, i.e. exactly in the middle of each interval. Thus for a one-dimensional real signal the function $p_q(\eta, \xi)$ takes a form:
\[ p_q(\eta, \xi) = \sum_{n=-\infty}^{\infty} \sum_{nq=0}^{(n+1/2)q} \int_{(n-1/2)q}^{(n+1/2)q} \cdots \int_{(n_{2N}-1/2)q}^{(n_{2N}+1/2)q} \frac{2N}{k=1} \frac{\eta_{k-n_k}q}{\frac{\eta_{1+q/2}}{\eta_{1-q/2}}} \frac{\eta_{2N+q/2}}{\eta_{2N-q/2}} \frac{\eta_{2N+q/2}}{\eta_{2N-q/2}} p(\xi, \xi) \, d\xi \]

where \( u_0(\eta) \) is an impulse (Dirac delta function), \( n \) is an integer, and \( q \) is the quantization interval. Extending this to an \( N \)-dimensional complex signal, and using the property that an impulse multiplied by a continuous function is equal to the same impulse multiplied by the value of that function at the point of occurrence of this impulse, results in:

\[
p_q(\eta, \xi) = \sum_{n_{1}=-\infty}^{\infty} \cdots \sum_{n_{2N}=-\infty}^{\infty} \frac{2N}{k=1} \frac{\eta_{k-n_k}q}{\frac{\eta_{1+q/2}}{\eta_{1-q/2}}} \frac{\eta_{2N+q/2}}{\eta_{2N-q/2}} \frac{\eta_{2N+q/2}}{\eta_{2N-q/2}} p(\xi, \xi) \, d\xi \left\{ \sum_{n_{1}=-\infty}^{\infty} \cdots \sum_{n_{2N}=-\infty}^{\infty} \frac{2N}{k=1} \frac{\eta_{k-n_k}q}{\frac{\eta_{1+q/2}}{\eta_{1-q/2}}} \frac{\eta_{2N+q/2}}{\eta_{2N-q/2}} \frac{\eta_{2N+q/2}}{\eta_{2N-q/2}} u_0(\eta_{k-n_k}q) \right\}
\]

\[ p_q(\eta, \xi) = A(\eta, \xi) \cdot B(\eta) \tag{11} \]

where:

\[ A(\eta, \xi) = \int_{\eta_{1-q/2}}^{\eta_{1+q/2}} \cdots \int_{\eta_{2N-q/2}}^{\eta_{2N+q/2}} p(\xi, \xi) \, d\xi \]

\[ B(\eta) = \sum_{n_{1}=-\infty}^{\infty} \cdots \sum_{n_{2N}=-\infty}^{\infty} \frac{2N}{k=1} \frac{\eta_{k-n_k}q}{\frac{\eta_{1+q/2}}{\eta_{1-q/2}}} \frac{\eta_{2N+q/2}}{\eta_{2N-q/2}} \frac{\eta_{2N+q/2}}{\eta_{2N-q/2}} u_0(\eta_{k-n_k}q) \]

\( n_1, \ldots, n_{2N} \) are integers.

Notice that only \( x^c \) is quantized, and that the vectors contain \( 2N \) elements: \( N \) real parts and \( N \) imaginary parts. Using linear system theory, it can be shown that:
\[ M_{xy}(\nu_x, \nu_y) = \frac{1}{(2\pi)^2N} M_A(\nu_x, \nu_y) * M_B(\nu_x) \]  \hspace{1cm} (12) 

where: 
\[ M_A(\nu_x, \nu_y) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} e^{j(\nu_x^T n + \nu_y^T \xi)} A(n, \xi) \, dn \, d\xi \]

\[ M_B(\nu_x) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} e^{j\nu_x^T n} B(n) \, dn \]

and (*) denotes convolution with respect to \( \nu_y \).

Also from the linear system theory:
\[ M_A(\nu_x, \nu_y) = \frac{-1}{(2N)^N} \prod_{k=1}^{2N} (j\nu_{x_k}) \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} e^{j(\nu_x^T n + \nu_y^T \xi)} \left( \frac{\partial^{2N}}{\partial n_1 \cdots \partial n_{2N}} A(n, \xi) \right) \, dn \, d\xi \]  \hspace{1cm} (13)

Using the above identities and some simple but tedious manipulations leads to:

\[ \frac{\partial^{2N}}{\partial n_1 \cdots \partial n_{2N}} A(n, \xi) = \frac{\partial^{2N}}{\partial n_1 \cdots \partial n_{2N}} \int_{n_1-q/2}^{n_1+q/2} \cdots \int_{n_2N-q/2}^{n_2N+q/2} p(\xi, \zeta) \, d\zeta \]

\[ = p(n_1+q/2, n_2+q/2, n_3+q/2, \cdots \, n_{2N}+q/2, \xi) \]

\[ - p(n_1-q/2, n_2+q/2, n_3+q/2, \cdots \, n_{2N}+q/2, \xi) \]

\[ + p(n_1-q/2, n_2-q/2, n_3+q/2, \cdots \, n_{2N}+q/2, \xi) \]

\[ \cdots \]

\[ + p(n_1-q/2, n_2-q/2, n_3-q/2, \cdots \, n_{2N}-q/2, \xi) \]  \hspace{1cm} (14)

(All possible combinations of \( \pm \) signs; \( z^{2N} \) terms)
Substituting (14) into (13) yields:

$$M_A(u_x, v_y) = \frac{-1}{2N} \left( \frac{e^{j\frac{q}{2}(v_{x1} + v_{x2} + \cdots)}}{\Pi_k jv_{xk}} \right)$$

$$= \frac{e^{j\frac{q}{2}(v_{x1} + v_{x2} + \cdots)}}{\Pi_k jv_{xk}} - \cdots \right) M_{xy}(v_x, v_y)$$

$$= q^{2N} \prod_{k=1}^{2N} \frac{\sin(v_{xk} q/2)}{(v_{xk} q/2)} M_{xy}(v_x, v_y)$$

(15)

From the derivation in (11):

$$M_B(v) = \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} e^{jv \sum_{n=-\infty}^{\infty} \sum_{k=1}^{2N} u_o(n_k - n q)} \, dn$$

$$= \left( \frac{2\pi}{q} \right)^{2N} \prod_{n_1=-\infty}^{\infty} \prod_{n_{2N}=-\infty}^{\infty} \frac{2N}{\Pi_k} u_o(v_k + \frac{2\pi}{q} n_k)$$

(16)

Substituting (15) and (16) into (12) produces:

$$M_{xy}^q(u_x, v_y) = \frac{1}{q^{2N}} \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} M_A(v_x - q, v_y) \sum_{n_1=-\infty}^{\infty} \sum_{n_{2N}=-\infty}^{\infty} \frac{2N}{\Pi_k} u_o(\alpha_k + \frac{2\pi}{q} n_k) \, d\alpha$$

$$= \sum_{n_1=-\infty}^{\infty} \sum_{n_{2N}=-\infty}^{\infty} \left( \frac{2N}{\Pi_k} \frac{\sin((v_{xk} - \frac{2\pi}{q} n_k) q/2)}{(v_{xk} - \frac{2\pi}{q} n_k) q/2} \right) M_{xy}(v_x - \frac{2\pi}{q} n_k, v_y)$$

$$+ \cdots \right)$$

$$M_{xy}(v_x, v_y)$$
\[
\sum_{n} S(\nu_x - \omega n) M_{xy} (\nu_x - \omega n, \nu_y)
\]

where

\[
S(\alpha) = \prod_{k=1}^{2N} \text{sinc} (\alpha_k / \omega)
\]

\[
sinc(\alpha) = \frac{\sin(\pi \alpha)}{(\pi \alpha)}
\]

\[
\omega = \frac{2\pi}{q}
\]

\[
n = \begin{bmatrix}
n_1 \\
\vdots \\
n_{2N}
\end{bmatrix}
\]

and the symbol \(\sum_{n}\) means the repeated sum from \(-\infty\) to \(+\infty\); i.e.,

\[
\sum_{n} (\quad) = \sum_{n_1=-\infty}^{\infty} \cdots \sum_{n_{2N}=-\infty}^{\infty} (\quad)
\]

Since \(y^c = F x_q^c\), it follows simply that:

\[
M_{yy}(\nu_x, \nu_y) = E[\exp(j\nu_x^T y^c_c + j\nu_y^T y^c_c)]
\]

\[
= E[\exp(j\nu_x^T y_c^c + j\nu_y^T F x_q^c)]
\]

\[
= E[\exp(j\nu_x^T y_c^c + j(F^{-1}\nu_y)^T x_q^c)]
\]

\[
= M_{xy}^q (F^{-1} \nu_y, \nu_x)
\]
\[ M_{yy'}(\nu_x',\nu_y) = \sum_n S(F^{-1} \nu_y - \omega n) M_y(\nu_x + \nu_y - \omega F n) \]  

(18)

In last step equations (10) and (17) were substituted.

3.3 Derivation of Mean-Square Error

The mean-square error of the \(k\)'th element of \(x\) can be derived from the joint characteristic function of \(y_k\) and \(y'_k\),

\[ M_e(\nu_1, \nu_2) = E[\exp(j\nu_1 y_k^c + j\nu_2 y'_k^c)] \]  

(19)

by using the identity:

\[ \varepsilon_{qk}^2 = E[(y_k^c - y_k^c)^2] \]

\[ = (y_k^c)^2 - 2 y_k^c y_k^c + (y_k^c)^2 \]

\[ = - \left( \frac{\partial^2}{\partial \nu_1^2} - 2 \frac{\partial^2}{\partial \nu_1 \partial \nu_2} + \frac{\partial^2}{\partial \nu_2^2} \right) M_e(\nu_1, \nu_2) \bigg|_{\nu_1 = \nu_2 = 0} \]  

(20)

The function \(M_e(\nu_1, \nu_2)\) is obtained from \(M_{yy}(\nu_x, \nu_y)\) by substituting:

\[ \nu_x = \nu_1 e_k \]

\[ \nu_y = \nu_2 e_k \]  

(21a)
where:

\[ e_k = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \text{ k'th element} \quad (21b) \]

The resulting equation is:

\[ M_e(\nu_1, \nu_2) = \sum_n S(\nu_2 a_k - \omega n) M_y((\nu_1 + \nu_2) e_k - \omega F_n) \quad (22) \]

where:

\[ a_k = F^T e_k \]

The differentiation of the above equation and substitution into equation (20) is done in the Appendix A. The final result is shown below:

\[ \varepsilon_{qk}^2 = \frac{a}{12} + 2 \sum_{\ell=1}^{2N} \left( \frac{a_{k\ell}}{\omega} \right)^2 \sum_{n=-\infty}^{\infty} \frac{(-1)^n}{n^2} M_y(-\omega n \beta_{\ell}) \]

\[ - \sum_{\ell=1}^{2N} \sum_{p=1}^{2N} \left( \frac{a_{k\ell}}{\omega} \right) \left( \frac{a_{kp}}{\omega} \right) \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \frac{(-1)^{n+m}}{nm} M_y(-\omega n \beta_{\ell} - \omega m \beta_p) \quad (23) \]

where:

\[ a_k = F^T e_k \]

\[ \beta_k = F e_k \]
Two important properties of the above expression can be easily observed: 1) The first term of the expression is equal to what would be the mean-square error if the quantization distortion was modeled by a random noise of uniform density added to each point of the signal. Since the quantization distortion is frequently modeled this way with accurate results for small \( q \), this term is the significant term for small \( q \). The remaining terms represent a correction for non-uniform density. 2) The expression for \( \frac{\epsilon^2}{q_k} \) does not involve any derivatives of \( M_y(y) \). This simplifies calculations of \( \frac{\epsilon^2}{q_k} \).

### 3.4 Extension to Vertical Quantization

The mean-square error derived in the previous section applies to horizontal quantization. To make the analysis complete, a similar derivation for vertical quantization is shown below. Vertical quantization, however, is less useful, because of the difficulties in its digital representation, e.g. values representing quantization intervals are \( \pm \frac{1}{2} q, \pm \frac{3}{2} q, \ldots \), which are more difficult to manipulate than integral multiples of \( q \).

The expression corresponding to equation (11) of the previous section is:

\[
p_{q}(n_1,5) = \sum_{n_1=-\infty}^{\infty} \cdots \sum_{n_{2N}=-\infty}^{\infty} \left( \prod_{k=1}^{2N} u_{n}^k(n_k-n_{k-1}q-\frac{q}{2}) \right) \int_{n_1q}^{(n_1+1)q} \cdots \int_{n_{2N}q}^{(n_{2N}+1)q} p_{\xi}(\xi,5) \, d\xi
\]
Equation (16) becomes:

\[
M_{B_1}(\psi) = \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} e^{j\psi T} B_1(\eta) \, d\eta
\]

\[
e^{j\frac{q}{2}} \text{tr}(\psi) \left(\frac{2\pi}{q}\right)^{2N} \sum_{n_1=-\infty}^{\infty} \cdots \sum_{n_{2N}=-\infty}^{\infty} \prod_{k=1}^{2N} u_o (\nu_k + \frac{2\pi}{q} n_k) \]

\[
= \left(\frac{2\pi}{q}\right)^{2N} \sum_{n_1=-\infty}^{\infty} \cdots \sum_{n_{2N}=-\infty}^{\infty} \prod_{k=1}^{2N} (-1)^{n_k} u_o (\nu_k + \frac{2\pi}{q} n_k)
\]

(25)

where:

\[
\text{tr}(\psi) = \sum_{\ell=1}^{2N} \nu_\ell
\]

Following the same steps as for the horizontal quantization leads to the following equivalents of equations (18) and (23):

\[
M_{yy', (\nu_x', \nu_y')} = \sum_{\eta} (-1)^{\text{tr}(\eta)} S \left( F^{-1} \nu_y - \omega \eta \right) M_y (\nu_x + \nu_y + \omega \eta) \]

(26)
\[ \varepsilon_{q k} = \frac{a}{12} + 2 \left( \sum_{\ell=1}^{2N} \frac{\alpha_{k \ell}}{\omega} \right)^2 \left( \sum_{n=-\infty}^{\infty} \frac{1}{n^2} M_y (-\omega n \beta_{\ell}) \right) \]

\[ - 2N \sum_{\ell=1}^{2N} \sum_{p=1}^{2N} \frac{\alpha_{k \ell}}{\omega} \left( \alpha_{k p} / \omega \right) \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \frac{1}{nm} M_y (-\omega n \beta_{\ell} - \omega m \beta_p) \]  

(27)

The only difference between equations (23) and (27) is an alternating sign of the terms in the infinite summation in (23). Both equations share the same properties listed following equation (23).
CHAPTER IV

PROPERTIES OF FFT ALGORITHM

4.1 Matrix Representation

The Fast Fourier Transform operates on a vector of dimension \( N = 2^M \), where \( M \) is an integer. Other FFT algorithms operating on vectors of different dimensions also exist, but they are less frequently used\(^6\), and thus are not considered here. To emphasize the dependence of the FFT on \( M \), the Fourier transform pair is represented by:

\[
Y_k = F_M \{x\}
\]

\[\text{(28)}\]

\[
x = F^{-1}_M \{y\}
\]

The FFT computation is carried out in \( M \) stages — in each stage the values of the signal are accessed from memory only once. Any scaling needed for normalization is usually done after the final stage, and is usually distributed in such a way that the forward transform is complete without scaling, and only the inverse transform requires scaling by \( 1/N \).

\[
y_k = \sum_{n=0}^{N-1} x_n \left( e^{-2\pi j/N} \right)^{nk}
\]

\[\text{(29)}\]

\[
x_n = \frac{1}{N} \sum_{k=0}^{N-1} y_k \left( e^{-2\pi j/N} \right)^{-nk}
\]
Let $z_k$ be the signal after $k$ stages of the FFT. Then the operation performed in the $k'$th stage can be mathematically represented by a matrix multiplication. Let $P_k$ be an $N \times N$ matrix of the $k'$th stage coefficients. Then,

$$z_k = P_k z_{k-1} \quad k = 1, 2, \ldots, M \quad (30)$$

As was mentioned in section 2.2, there are two basic FFT algorithms called "decimation in time" and "decimation in frequency". Each algorithm yields a different $P_k$ matrix, which in subsequent calculations are denoted by $P^t_k$ and $P^f_k$ respectively. The mathematical derivation of those algorithms is shown below. For more detail see reference 6.

**Decimation in Time**

Starting from definition (29) one obtains:

$$y_k = \sum_{n=0}^{N-1} x_n w^{nk}$$

$$= \sum_{\ell=0}^{N/2-1} \left( x_{2\ell} w^{2\ell k} + x_{2\ell+1} w^{(2\ell+1)k} \right)$$

$$= \sum_{\ell=0}^{N/2-1} x_{2\ell} (w^2)^{\ell k} + w^k \sum_{\ell=0}^{N/2-1} x_{2\ell+1} (w^2)^{\ell k} \quad (31)$$

where:

$$w = e^{-2\pi j/N}$$
Let:

$$g_k = \sum_{\ell=0}^{\frac{N}{2}-1} x_{2\ell} (W^2)^{\ell k} = \sum_{\ell=0}^{\frac{N}{2}-1} x_{2\ell} \left( -\frac{2\pi i}{N/2} \right)^{\ell k}$$

$$h_k = \sum_{\ell=0}^{\frac{N}{2}-1} x_{2\ell+1} (W^2)^{\ell k} = \sum_{\ell=0}^{\frac{N}{2}-1} x_{2\ell+1} \left( -\frac{2\pi i}{N/2} \right)^{\ell k}$$

(32)

In the definitions above, the sequence \(\{g_k\}_{k=0}^{N-1}\) is the Fourier transform of \(\{x_{2\ell}\}_{\ell=0}^{N-1}\), and the sequence \(\{h_k\}_{k=0}^{N-1}\) is the Fourier transform of \(\{x_{2\ell+1}\}_{\ell=0}^{N-1}\).

Substituting (32) into (31) results, for values of \(k\) less than \(\frac{N}{2}\), in:

$$y_k = g_k + W^k h_k \quad \text{for} \quad k < N/2$$

(33)

For values of \(k\) greater than or equal to \(\frac{N}{2}\),

$$g_k = \sum_{\ell=0}^{\frac{N}{2}-1} x_{2\ell} (W^2)^{\ell k}$$

$$= \sum_{\ell=0}^{\frac{N}{2}-1} x_{2\ell} \left( e^{-\frac{2\pi i}{N/2} \ell N/2} - \ell (k-N/2) \right)$$

$$= g_{k-N/2}$$
Similarly,
\[ h_k = h_{k-N/2} \]

Substituting these results into (31) leads, for \( k \geq \frac{N}{2} \), to:
\[ y_k = g_{k-N/2} + w^k h_{k-N/2} \]
\[ = g_{k-N/2} - w^{k-N/2} h_{k-N/2} \quad k \geq \frac{N}{2} \]  \hspace{1cm} (34)

Equations (33) and (34) can be expressed more compactly as a matrix operation on a vector, in agreement with equation (30).
\[ y = P_t z_{M-1} \]  \hspace{1cm} (35)

\[ P_t = \begin{bmatrix} I & W_M \\ I & -W_M \end{bmatrix} \]

where:
\[ W_M = \begin{bmatrix} W^0 \\ W^1 \\ \vdots \\ W^{N-1} \end{bmatrix} \]

and \( z_{M-1} \) is a column matrix
\[ z_{M-1} = \begin{bmatrix} g \\ h \end{bmatrix} \]
with \( \mathbf{g} \) and \( \mathbf{h} \) representing the sequences

\[
\begin{array}{c}
g_k = \frac{N-1}{2}, k=0, 1, \ldots, \frac{N}{2}-1 \\
h_k = \frac{N-1}{2}, k=0, 1, \ldots, \frac{N}{2}-1
\end{array}
\]

\[
\mathbf{g} = 
\begin{bmatrix}
g_0 \\
g_1 \\
\vdots \\
g_{N/2-1}
\end{bmatrix}, \quad \mathbf{h} = 
\begin{bmatrix}
h_0 \\
h_1 \\
\vdots \\
h_{N/2-1}
\end{bmatrix}
\]

\[
\mathbf{g} = \mathcal{F}_{M-1} \left\{ \left\{ x_{2\ell} \right\}_{\ell=0}^{N/2-1} \right\}
\]

\[
\mathbf{h} = \mathcal{F}_{M-1} \left\{ \left\{ x_{2\ell+1} \right\}_{\ell=0}^{N/2-1} \right\}
\]

The vectors \( \mathbf{g} \) and \( \mathbf{h} \), as was shown in (32), are Fourier transforms of order \( M-1 \); each transform operating on half of the input signal. Therefore, in the same way that \( \mathbf{y} \) was expressed as a matrix operating on two Fourier transforms, both \( \mathbf{g} \) and \( \mathbf{h} \) can each be expressed as a matrix operating on two Fourier transforms of order \( M-2 \). This fact allows writing the \( \mathbf{g} \) and \( \mathbf{h} \) without any further derivations:

\[
\mathbf{g} = 
\begin{bmatrix}
I & W_{M-1} \\
I & -W_{M-1}
\end{bmatrix}
\begin{bmatrix}
\mathcal{F}_{M-2}^{(1)} \\
\mathcal{F}_{M-2}^{(2)}
\end{bmatrix}
\]

\[
\mathbf{h} = 
\begin{bmatrix}
I & W_{M-1} \\
I & -W_{M-1}
\end{bmatrix}
\begin{bmatrix}
\mathcal{F}_{M-2}^{(1)} \\
\mathcal{F}_{M-2}^{(2)}
\end{bmatrix}
\]
\[
\begin{bmatrix}
I & W_{M-1} & Z_{M-2}^{(3)} \\
I & -W_{M-1} & Z_{M-2}^{(4)}
\end{bmatrix}
\]

where \( W_{M-1} \) is a diagonal matrix with

\[
(W_{M-1})_{\ell\ell} = e^{-j(2\pi/N)2\ell} \quad \ell = 0, 1, 2, \ldots
\]

and \( Z_{M-2}^{(j)} \)'s are transforms of portions of the input signal.

Combining the above results into a single expression, in accordance with equation (30):

\[
Z_{M-1} = P_{M-1}^t Z_{M-2}
\]

\[
P_{M-1}^t = \begin{bmatrix}
I & W_{M-1} & 0 \\
I & -W_{M-1} & \cdot \cdot \cdot \\
0 & I & W_{M-1} \\
0 & I & -W_{M-1}
\end{bmatrix}
\]

(36)

Let \( B_k^t \) be a \((2^k) \times (2^k)\) matrix given by:

\[
B_k^t = \begin{bmatrix}
I & W_k \\
I & -W_k
\end{bmatrix}
\]
where

$$W_k = \begin{pmatrix} \nu_1 & 0 \\ \nu_2 & 0 \\ \nu_2 & 0 \\ \vdots & \vdots \\ 0 & \nu_2^{k-1} \end{pmatrix} \text{ diagonal matrix}$$

$$\nu_\ell = e^{-j2\pi/2^k}(\ell-1)$$

The $W_k$ defined above reduces to the previously defined $W_M$ and $W_{M-1}$ for $k = M$ and $k = M-1$. Therefore, the $P_{M-1}^t$ matrix can be written as:

$$P_{M-1}^t = \begin{pmatrix} B_{M-1}^t & 0 \\ 0 & B_{M-1}^t \end{pmatrix}$$

Carrying out this procedure one step further gives:

$$P_{M-2}^t = P_{M-2}^t z_{M-3}$$

where:

$$P_{M-2}^t = \begin{pmatrix} B_{M-2}^t & 0 \\ B_{M-2}^t & B_{M-2}^t \end{pmatrix}$$

Continuing in this manner results in $M$ stages

$$Y = P_M^t \cdots P_1^t z_0$$  \hspace{1cm} (37)
where:

\[
\begin{vmatrix}
B_k^t \\
B_k^t \\
0 \\
0
\end{vmatrix}
\]

The vector \( z_0 \) is formed from \( x \) through a one-to-one correspondence (mapping) of elements. By carefully following the derivation of (37), it can be shown that this correspondence is obtained by reversing the order of the bits in the binary indices of the vector elements.* If for example, \( x \) and \( z_0 \) have each 32 elements, then the fifth element of \( x \) has a binary index 00101 (=5). Reversing the bits of that index results in 10100 (=20). Therefore the fifth element of \( x \) is equal to the twentieth element of \( z_0 \). In subsequent derivations, the subscript \( \Delta \) is used to denote this rearrangement of elements by the bit reversal of binary indices:

\[
z_0 = x_\Delta
\]  
(38)

Clearly, if \( z_0 = x_\Delta \), then \((z_0)_\Delta = x\).

Decimation in Frequency

Starting from definition (29) one obtains

\[
y_k = \sum_{n=0}^{N-1} x_n w^{nk}
\]
\[
\frac{N-1}{2} \sum_{k=0}^{N-1} \left( x_{\ell} W^{\ell k} + x_{\ell+N/2} W^{(\ell+N/2)k} \right)
\]

\[
= \sum_{\ell=0}^{N-1} \left( x_{\ell} + (-1)^k x_{\ell+N/2} \right) W^{\ell k}
\] (30)

Divide the set \{y_k\}_{k=0}^{N-1} into two sets: \{y_{2k}\}_{k=0}^{N-1}, \{y_{2k+1}\}_{k=0}^{N-1}. Then

\[
\frac{N-1}{2} \sum_{k=0}^{N-1} (x_{\ell} + x_{\ell+N/2}) (W^2)^{\ell k} = \sum_{\ell=0}^{N-1} (x_{\ell} - x_{\ell+N/2}) W^{\ell} (W^2)^{\ell k}
\] (40)

Let:

\[
\{x_{\ell} + x_{\ell+N/2}\}_{\ell=0}^{N-1} = g
\]

\[
\{(x_{\ell} - x_{\ell+N/2}) W^{\ell}\}_{\ell=0}^{N-1} = h
\]

Then equation (40) can be more compactly expressed by:

\[
\{y_{2k}\}_{k=0}^{N-1} = F_{M-1}\{g\}
\]

(41)

\[
\{y_{2k+1}\}_{k=0}^{N-1} = F_{M-1}\{h\}
\]
and \( g \) and \( h \) are obtained by matrix operation on \( x \)

\[
\begin{bmatrix} g \\ h \end{bmatrix} = z_1 = P_{-1}^f x
\]

where:

\[
P_{-1}^f = \begin{bmatrix} I & I \\ W_{M} & -W_{M} \end{bmatrix}
\]

Next, the set \( \{y_{2k}\}_{k=0}^{N-1} \) can be divided into two sets \( \{y_{4k}\}_{k=0}^{N-1} \)

\[
\{y_{4k+2}\}_{k=0}^{N-1}, \text{ and similarly the set } \{y_{2k+1}\}_{k=0}^{N-1}
\]

two sets \( \{y_{4k+1}\}_{k=0}^{N-1}, \{y_{4k+3}\}_{k=0}^{N-1} \). These divisions are analogous to the first division of \( \{y_k\}_{k=0}^{N-1} \), and therefore the next stage of the transform can be written without any further derivations:

\[
z_2 = P_{-2}^f z_1
\]

\[
P_{-2}^f = \begin{bmatrix} I & I & \cdots & 0 \\ W_{M-1} & -W_{M-1} & \cdots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ 0 & \cdots & 0 & I & I \\ W_{M-1} & -W_{M-1} \end{bmatrix}
\]
This procedure can be carried out until the complete transform is obtained. It again requires $M$ stages, and the final result is in the bit reversed form:

$$y_{\Delta} = P_M^f \cdots P_1^f x$$  

(44)

The explicit form of $P_k^f$ is specified below in the summary.

Summarizing

$$y = P_M^t \cdots P_1^t x_{\Delta} \quad \text{dec. in time}$$  

(45)

$$y_{\Delta} = P_M^f \cdots P_1^f x \quad \text{dec. in frequency}$$

where:

$$P_k^t = \begin{bmatrix}
B_k^t & 0 & 0 & \cdots \\
0 & B_k^t & 0 & \\
0 & 0 & B_k^t & \\
\vdots & \vdots & \vdots & \\
\end{bmatrix}$$

$$P_k^f = \begin{bmatrix}
B_k^f & 0 & 0 & \cdots \\
0 & B_k^f & 0 & \\
0 & 0 & B_k^f & \\
\vdots & \vdots & \vdots & \\
\end{bmatrix}$$
\( P_k \) is an \( N \times N \) matrix

\( B_{k}^{t} \) is an \( (2^k) \times (2^k) \) matrix.

\( B_{k}^{f} \) is an \( (2^{M-k+1}) \times (2^{M-k+1}) \) matrix.

\[
B_{k}^{t} = \begin{bmatrix} I & W_k \\ I & -W_k \end{bmatrix} \quad \text{dec. in time}
\]

\[
B_{h}^{f} = \begin{bmatrix} I & I \\ W_{M-k+1} & -W_{M-k+1} \end{bmatrix} \quad \text{dec. in frequency}
\]

\[
W_{k} = \begin{bmatrix} v_1 & 0 & \cdots & 0 \\ 0 & v_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & v_{2^{k-1}} \end{bmatrix} \quad \text{diagonal matrix}
\]

\[ v_{\ell} = e^{-j(2\pi/2^k)\ell(\ell-1)} \]

\[
z_{k} = P_{k}^{t} \cdots P_{1}^{t} z_{\Delta} , \quad z_{0} = x_{\Delta} \quad \text{dec. in time}
\]

\[
z_{k} = P_{k}^{f} \cdots P_{1}^{f} z , \quad z_{0} = x \quad \text{dec. in frequency}
The form of \( P_k \) allows a fast in-place computation of Fourier transform.

### 4.2 Important Properties

The inverse of \( P_k \) is given by (the superscript is omitted whenever the relation holds for both decimations):

\[
P_k^{-1} = \frac{1}{2} P_k^{*T}
\]

(46)

where * denotes complex conjugate and T denotes transpose.

Proof:

\[
\frac{1}{2} P_k P_k^{*T} = \frac{1}{2}
\]

For decimation in time:

\[
B_k^t B_k^{*T} = \begin{bmatrix}
I + W \frac{W}{k-k} & I - W \frac{W}{k-k} \\
I - W \frac{W}{k-k} & I + W \frac{W}{k-k}
\end{bmatrix}
\]

For decimation in frequency:

\[
B_k^{fT} B_k^{f*} = \begin{bmatrix}
2I & 0 \\
0 & 2W^{M-k+1} \frac{W^*}{M-k+1}
\end{bmatrix}
\]
Next:

\[
\begin{bmatrix}
\nu_1^* & \nu_1^* & 0 \\
\nu_2^* & \nu_2^* & 0 \\
0 & 0 & \ddots
\end{bmatrix}
\]

where:

\[
\nu_i \nu_i^* = e^{-j(2\pi/2^k)i} e^{j(2\pi/2^k)i} = 1
\]

thus:

\[
WW^* = I
\]

\[
BB^* = 2I
\]

and

\[
\frac{1}{2} p_k p_k^T = I
\]

where

\[
\frac{1}{2} p_k^T p_k = p_k^{-1}
\]

Let \( p_k \) be the energy of the signal after the \( k \)'th stage, defined by:

\[
p_k = z_k^T z_k^*
\]

(47)

where \( z_k \) is the signal after the \( k \)'th stage. Then the energy \( p_k \) is deterministically related to the energy of the previous \((k-1)\)'st stage by:

\[
p_k = 2p_{k-1}
\]

(48)
Proof:

\[ p_k = z_k^T z_k^* \]

\[ = (p_k z_{k-1})^T (p_k z_{k-1})^* \]

\[ = \frac{z_{k-1}^T}{z_{k-1}} (p_k^T p_k z_{k-1}) \]

\[ = \frac{z_{k-1}^T}{z_{k-1}} (p_k^T p_k^*) z_{k-1} \]

\[ = 2 \frac{z_{k-1}^T}{z_{k-1}} z_{k-1} \]

\[ = 2p_{k-1} \]

\[ p_o = \frac{z_{k}^T}{z_{k}} \frac{z_{k}^*}{z_{k}^*} \]

Another relation which can be easily verified is:

\[ p_k^T = (p_{M-k})^T \] \hfill (49a)

\[ p_M^T \cdots p_1^T = (p_M^T \cdots p_1)^T \] \hfill (49b)

The matrix \( p_k \) has the property that it independently operates on pairs of elements. Each such operation is called a "butterfly", and it can be easily derived from equation (45) as shown below. First, divide the vectors \( z_k \) and \( z_{k-1} \) into sets of smaller vectors according to the following expression:

\[ z_k = p_k z_{k-1} \]
\[
\begin{pmatrix}
E_k & 0 & \cdots & z_{k-1}^{(1)} \\
0 & B_k & \cdots & z_{k-2}^{(2)} \\
\vdots & \vdots & \ddots & \vdots \\
\vdots & \vdots & \cdots & \vdots
\end{pmatrix} = 
\begin{pmatrix}
E_k & z_{k-1}^{(1)} \\
E_k & z_{k-1}^{(2)} \\
\vdots & \vdots \\
\vdots & \vdots
\end{pmatrix} = 
\begin{pmatrix}
z_k^{(1)} \\
z_k^{(2)} \\
\vdots \\
\vdots
\end{pmatrix}
\]

Substituting appropriate \( B_k \) matrix yields:

\[
z_k^{(j)} = \begin{pmatrix}
I & W_k \\
I & -W_k
\end{pmatrix} z_{k-1}^{(j)} \quad \text{dec. in time}
\]

\[
z_k^{(j)} = \begin{pmatrix}
I & I \\
\lambda_{k-1} & -\lambda_{k-1}
\end{pmatrix} z_{k-1}^{(j)} \quad \text{dec. in frequency}
\]

Next, denote by \( \zeta_1, \zeta_2, \ldots \) the elements of \( z_k \) and by \( \eta_1, \eta_2, \ldots \) the elements of \( z_{k-1} \). The index of \( \zeta_1 \) or \( \eta_1 \) is used only to distinguish between the elements, and has no relation to the positions (indices) of those elements in \( z_k \) or \( z_{k-1} \). Using this notation and above identities, the butterfly operation can be written as:
\[ \zeta_1 = \eta_1 + \omega \eta_2 \]
\[ \text{dec. in time} \quad (50a) \]
\[ \zeta_2 = \eta_1 - \omega \eta_2 \]
\[ \zeta_1 = \eta_1 + \eta_2 \]
\[ \text{dec. in frequency} \quad (50b) \]
\[ \zeta_2 = \omega \eta_1 - \omega \eta_2 \]

where \( \omega \) is an element of \( W_k \).

4.3 Bounds on Largest Magnitude

The signal \( z_k \) is assumed to be stored in the digital memory in a block-floating-point representation (section 2.3) with the real and imaginary parts occupying separate words. The common characteristic of the \( z_k \) is related to the largest absolute value of all the numbers stored in the memory. In the subsequent derivations, this value is designated by the symbol \( H_k \):

\[ H_k = \max(|\zeta_{1r}|, \ldots, |\zeta_{Nr}|, |\zeta_{1i}|, \ldots, |\zeta_{Ni}|) \quad (51) \]

where

\[ z_k = \{\zeta_{\ell}\}_{\ell=0}^{N-1} = \{\zeta_{1r} + j \zeta_{1i}\}_{\ell=0}^{N-1} \]

\( \zeta_{1r}, \zeta_{1i} \) real
By calculating the upper and lower bounds on the $H_k$, the behavior of the characteristic can be computed. This information is later used to obtain the largest and the smallest possible quantization errors.

The bounds on $H_k$ are first calculated using equation (48). Next, more accurate derivation yields better bounds for one and two successive stages of the FFT. Finally, those results are used to evaluate the bounds on the characteristic.

Consider all the possible signals $x$ with a given initial energy $p_o = x^T x$. Then according to the equation (48), the output energy is $p_M = 2^M x^T x$. For all such signals the largest attainable value of $H$ occurs if the whole output energy ends up at one point of $M$ the output signal, and moreover, if it ends up in either the real or imaginary part of that point

$$H_M \leq \sqrt{x^T x} (2^{M/2}) \quad (52)$$

Similarly, the smallest possible value of $H_M$ occurs if the output energy is evenly distributed between all the points of the output signal, and is equally split between the real and imaginary parts of each point.

$$H_M \geq \sqrt{x^T x} (2^M (1/2N)) \quad (53)$$
By analogous arguments and some manipulations of $P_k$ matrices, it can be shown that for the $k'$th stage, the $H_k$ can be related to the energy of signal at an arbitrary other stage by the following inequality:

$$\frac{1}{\sqrt{2}} \sqrt{z_{k'}^T z_{k'}} \leq H_k \leq 2^{\frac{k'-\ell}{2}} \sqrt{z_{k'}^T z_{k'}}$$

(54)

$k, \ell = 0, 1, 2, \ldots, M; \ k > \ell$

Unfortunately, the value of the characteristic cannot be easily related to the energy. However, for a given value of $H_\ell$, the energy $z_{\ell}^T z_{\ell}^*$ can be bounded from both sides. For all possible signals $z_{\ell}$ with $H_\ell$, the largest energy occurs if the magnitudes of the real and imaginary parts of all the elements of $z_{\ell}$ are equal to $H_\ell$, and the smallest energy occurs if only one element has energy $H_\ell^2$ in either its real or imaginary part, and all other elements are zero.

$$H_\ell^2 \leq z_{\ell}^T z_{\ell}^* \leq 2NH_\ell^2$$

Furthermore, by noticing that the energy from the $\ell'$th stage can be transferred to the $k'$th stage only between sets of $2^{k'-\ell}$ elements, the following inequality can be obtained from (54).

$$\frac{H_\ell}{\sqrt{2}} \leq H_k \leq 2^{\frac{k'-\ell}{2}} H_\ell, \ k > \ell$$

(55)
This result is obtained from the conservation of energy equation (48).

For one or two successive stages, a more detailed examination presented in Appendix B results in several closer bounds. Those bounds are listed below:

For the decimation in time the higher bounds for one and two consecutive stages are:

\[ H_k \leq (1 + \sqrt{2}) H_{k-1} \]  \hspace{1cm} (56)

\[ H_k \leq 5.29 H_{k-2} \]  \hspace{1cm} (57)

For the decimation in frequency a better lower bound for one stage is given by:

\[ H_k \geq \frac{2}{1 + \sqrt{2}} H_{k-1} \]  \hspace{1cm} (58)

The lower bounds for the decimation in time and higher bounds for the decimation in frequency are given by (55).

The elements of \( P_k \) matrix for \( k = 1, 2 \) in decimation in time and for \( K = M, M - 1 \) in decimation in frequency are \( +1, -1, +j, -j \).

Using this property it can be shown that:

\[ H_0 \leq H_1 \leq 2H_0 \hspace{1cm} \text{dec. in time} \]  \hspace{1cm} (59a)

\[ H_1 \leq H_2 \leq 2H_1 \]

\[ H_{M-1} \leq H_M \leq 2H_{M-1} \hspace{1cm} \text{dec. in frequency} \]  \hspace{1cm} (59b)

\[ H_{M-2} \leq H_{M-1} \leq 2H_{M-2} \]
The above results are used in obtaining the bounds on the number of changes of the characteristic of the block-floating-point representation between the stages of the FFT. The value of the characteristic is a function of $H_k$. By convention, the fractional point of this representation (analogous to the decimal point) is assumed to the left of all the significant digits, i.e. all mantissas are fractions of magnitude smaller than one. Then, after the $k^{th}$ stage and before the $(k+1)^{st}$ stage, the characteristic $s_k$ always satisfies the inequality

$$2^{s_k - 1} \leq H_k < 2^{s_k} \quad (60)$$

This is exactly true only if the mantissas are stored with infinite precision, i.e. infinite number of bits.

By combining the lower bound on $H_k$ from (55) with the inequality (60) it can be easily verified that the decrease of the characteristic from any stage to any later stage (if a decrease should occur) cannot be greater than one. From (55):

$$\frac{H_k}{\sqrt{2}} \leq H_k, \quad k > \ell$$

and from (60)

$$2^{s_{\ell} - 1} \leq H_{\ell}, \quad H_k < 2^{s_k}$$

combining these results yields:

$$2^{s_{\ell} - 3/2} < 2^{s_k}, \quad k > \ell \quad (61)$$
Taking the logarithm of both sides proves the derived result:

\[ s_k - s_\ell > -\frac{3}{2}; \quad k > \ell \] (62)

In a similar way it can be shown that for \( n \) stages of the transform, the characteristic cannot increase by more than \( (n+1) \).

From (55)

\[ H_k \leq \frac{1}{2} \left( k - \ell + \frac{1}{2} \right) H_\ell; \quad k > \ell \]

and from (60)

\[ \frac{s_k - 1}{2} \leq H_k \leq \frac{s_\ell}{2} \]

Combining these results and taking the logarithm yields:

\[ \frac{s_k - 1}{2} \leq \frac{s_\ell + k - \ell + \frac{1}{2}}{2}; \quad k > \ell \] (63)

\[ s_k - s_\ell < k - \ell + \frac{3}{2}; \quad k > \ell \] (64)

Since the characteristic can have only integral values, the above results can also be expressed by:

\[ -1 \leq (s_k - s_\ell) \leq k - \ell + 1 \quad k > \ell \] (65)

Above relation was derived from (55) and the definition of the characteristic.

The more accurate bounds derived in the Appendix B do not invalidate the inequality (65). Following the method used in deriving (62) and (64) yields:
a) For inequality (56)

\[ \frac{s_{k-1}}{2} < \frac{s_{k-1}}{2} (1+\sqrt{2}) \]

(66)

\[ s_k - s_{k-1} < 2.26 \]

b) For inequality (57)

\[ \frac{s_{k-1}}{2} < 5.29 \frac{s_{k-2}}{2} \]

(67)

\[ s_k - s_{k-2} < 3.4 \]

c) For inequality (58)

\[ \frac{s_k}{2} > \frac{2}{1+\sqrt{2}} \frac{s_{k-1}}{2} \]

(68)

\[ s_k - s_{k-1} > -1.26 \]

Since the characteristic can only have integral values, these agree with (65).

The inequality (65) allows a decrease of the characteristic by one. However, from an ensemble of all possible input signals, very few of them will result in a decrease in the characteristic. Therefore, it is customary in practical applications to provide only for increases
in the characteristic. This procedure avoids the considerable
difficulty involved in implementing the mechanism of decreasing the
characteristic and the associated left shifts, and has little or no
effect on the quality of the resulting image. Therefore,

\[ 0 \leq s_k - s_\ell \leq k - \ell + 1 \quad k > \ell \]  

(69)
CHAPTER V

NOISE INJECTED BY FFT ALGORITHM

5.1 Noise Vector

In the previous chapter some important properties of the FFT algorithm were derived. In this chapter these properties will be used to estimate the noise introduced by the actual implementation with finite precision registers and memory words. The major aspects of this noise were already outlined in section 2.3; the details and the mathematical formulations will be considered now.

This noise, which is caused by the truncation - the discarding of the least significant bits in order to contain the value in the register, is the difference between the truncated value and the accurate value of the signal. This difference is propagated through the remaining computations and produces a distortion in the final image.

The truncation occurs after the arithmetic operations and in the evaluation of algebraic coefficients. After each of these operations, the eliminated bits may either be used in rounding-off the resulting number, or simply ignored. Rounding-off requires two additional operations: test and add, and the resulting gains in accuracy may not justify it. If the values are not rounded-off and the two's-complement arithmetic is used, then the truncation error is always less than the least significant
bit. Defining this error as the truncated value minus the accurate value forces it to be negative:

\[-\frac{p}{2}p + s \quad < n_t \quad \leq 0\]  

(70)

where:  
\(n_t\) = truncation error  
\(s\) = current value of characteristic  
\(p\) = precision, number of significant bits

On the other hand, if the values are rounded-off, then the error is bounded by:

\[-\frac{p}{2}p + s - 1 \quad < n_t \quad \leq 2^{p}p + s - 1\]  

(71)

The basic operation of the FFT is the butterfly operation. It consists of fetching two complex numbers from the arithmetic memory, operating on them with an appropriate Fourier coefficient, and storing them again in the arithmetic memory. The accuracy of this operation may be improved by using registers longer than the words of arithmetic memory by several bits. This way the truncation errors propagated to the end of the butterfly operation can be made smaller than the least significant bit of the arithmetic memory. After the final truncation necessary to permit placing the results back in the arithmetic memory, the only truncation noise present is that resulting from the final truncation. This procedure not only improves the performance of the processor, but also simplifies the analysis of the total noise. The additional hardware required for the few longer registers needed in computing the butterflies is a small increase in cost and complexity. Therefore, this scheme is assumed in the further analysis. For a different analysis,
but without the above assumption, consult reference 20.

The next problem in analyzing the noise injected by the processor is the effect of the shifts due to the changes of the characteristic. As was indicated in the previous section, the shifts can be only to the right, with at most two shifts per stage. Since each word of the arithmetic memory has only \( p \) bits; a shift to the right always annihilates the least significant bit of the value stored in that word. Therefore, the errors injected by the shifts are very similar to the truncation errors. If no rounding-off is performed, then the shift error is equal to \(-2^{p+s-1}\) if the bit shifted out was 1, and is equal to 0 if the bit shifted out was 0, where \( s \) is the characteristic after the shift.

Consider a processor which does not round-off the values before storing them in the arithmetic memory, but simply truncates them to the required number of bits. For such a processor a right shift in the \( k \)th stage is equivalent to a truncation by one more bit in all the butterfly operations of that stage. Similarly, the effect of two right shifts is the same as the effect of two bit longer truncations. In both of these cases the final combined truncation and shifting error is less than (in magnitude) the least significant bit of the signal at the completion of that stage. Mathematically, this error is bounded by:

\[
-2^{p+s_k} < n \leq 0
\]

where \( s_k \) is now the characteristic immediately after the completion of the \( k \)'th stage.
The problem is more complicated for a processor which does perform the rounding-off. The errors which can occur in such a processor are illustrated by an example below, in which $p_p$ is assumed to be 3.

Let $\mu$ be a number which can be expressed in the binary format by:

$$\mu = 0.b_1b_2$$  \hspace{1cm} (73)

where $b_1$ and $b_2$ represent bits, i.e. $b_1, b_2 = 0,1$. Let $\phi$ be the actual value of the signal, $0 \leq \phi < 1$, requiring in general an arbitrary number of bits for accurate representation. Finally, let $\phi_q$ be the value stored in the memory corresponding to $\phi$. Consider $\phi$ in the following ranges shown below in binary notation. These ranges are also illustrated in Fig. 3.

1) $\mu - 0.0001 \leq \phi < \mu$
2) $\mu \leq \phi < \mu + 0.0001$
3) $\mu + 0.0001 \leq \phi < \mu + 0.001$
4) $\mu + 0.001 \leq \phi < \mu + 0.0011$

![Diagram showing ranges](image)

Range given by (76)

Fig. 3. Ranges of expression (74)
The rounding-off to 3 significant bits of \( \phi \) results in the following values of \( \phi_q \):

\[
\phi_q = 0.b_1b_20 = \mu \quad \text{ranges 1, 2}
\]

\[
\phi_q = 0.b_1b_21 = \mu + 0.001 \quad \text{ranges 3, 4}
\]  

(75)

Next, consider a single right shift and its effect on the combined range of 3 and 4, given by:

\[
\mu + 0.0001 \leq \phi < \mu + 0.0011
\]  

(76)

If no rounding-off is performed during the shift then the range given by (76) is represented by \( \mu \). If the rounding-off is attempted, then this range is represented by \((\mu + 0.01)\). However, both \( \mu \) and \((\mu + 0.01)\) are equally far away from the region given by (76), and therefore no accuracy has been gained by rounding-off. Since rounding-off takes additional operations, no rounding-off during the shift is assumed. Thus the entire range

\[
\mu - 0.0001 \leq \phi < \mu + 0.0011
\]  

(77)

is represented by \( \mu \) (if the processor does provide rounding-off prior to the shift).

If the processor does not perform any rounding-off, the \( \mu \) would represent a range given by:

\[
\mu \leq \phi < \mu + 0.01
\]  

(78)

Therefore, the rounding-off after butterflies increases the accuracy, but due to the shifts, this increase is not as large as for the ideal case,
where \( \mu \) would represent a region given by:

\[
\mu - 0.001 \leq \phi < \mu + 0.001 \quad (79)
\]

The undesirable effects of the shifts on the accuracy suggest a better implementation of the processor. The butterfly function operates on a pair of signal points from the arithmetic memory. For each such pair, a change of characteristic may occur before its butterfly operation, during its butterfly operation, or after the operation. Again, at most two such changes are possible in one stage. Each change requires a shift in all the words of the arithmetic memory. However, for each pair the shift does not have to occur immediately after the change; it may be delayed by various algorithms. If the change for that pair occurs before its butterfly operation, then the shift may be delayed and executed immediately after the butterfly. This is easy to implement and it even improves the speed of the processor. Also, the shift delayed this way becomes the part of the final rounding-off ending the butterfly operation. The resulting values are simply rounded-off by one more bit. This procedure increases the accuracy of the processor, because the rounded-off value corresponds to region (79) rather than (77).

Similarly, the change of characteristic during the butterfly operation also results in rounding-off by one more bit. Finally, if the change occurs after the butterfly operation has been completed, then the accuracy is clearly improved by delaying the shift until the next stage. Unfortunately, delaying the shift until the next stage is more complicated, and the accuracy gained is not as large as in the first two cases. Therefore, it may be more practical not to include such a delay in the processor.
Concluding the last few paragraphs it is evident that the shifting and truncation errors in the k'th stage for a well-designed processor with rounding-off can be approximately bounded by:

\[-p + s_k - 1 \leq n \leq 2 p + s_k - 1\]

(80)

where \(s_k\) is the characteristic after completion of the k'th stage. The error may be slightly larger if the change of the characteristic occurs after the butterfly operation.

In analyzing the total noise injected into the image, each truncation and shifting error will be modeled as a random variable added to the signal. This approach is justified by the fact that for sufficiently strong signals, the signal and the error are approximately independent. This means that there may be many signals, all corresponding to the same area mapped, differing only by few quantization levels at any point, and all producing nearly identical images, but each resulting in a very different set of truncation and shifting errors. Therefore, a whole ensemble of random errors can be associated with each such image.

In a typical butterfly operation, a \(p\)-bit number from the arithmetic memory is multiplied by a many bit approximation of an irrational Fourier coefficient, and then truncated to \(p\) bits. The resulting error can be anywhere within the interval \((-2 p + s_k, 0]\) (assuming no rounding-off). Therefore, an appropriate distribution for this error is a uniform distribution shown in Fig. 4.
The mean and the variance of the distribution shown in Fig. 4 are

\[
\text{mean} = \frac{-p + s_k - 1}{2} = a = \frac{s_k}{2}
\]

(81)

\[
\text{variance} = \frac{1}{12} \left( \frac{-p + s_k}{2} \right)^2 = \sigma_n^2 = 2s_k
\]

(82)

where \(a\) and \(\sigma_n^2\) are the mean and the variance for \(s_k = 0\). For a processor with rounding-off the mean is zero.

For the decimation in time, the butterfly operation can be expressed by:
\[ \begin{align*}
\zeta_{1r} &= \eta_{1r} + \omega_r \eta_{2r} - \omega_i \eta_{2i} \\
\zeta_{1i} &= \eta_{1i} + \omega_i \eta_{2r} + \omega_r \eta_{2i} \\
\zeta_{2r} &= \eta_{1r} - \omega_r \eta_{2r} + \omega_i \eta_{2i} \\
\zeta_{2i} &= \eta_{1i} - \omega_i \eta_{2r} - \omega_r \eta_{2i}
\end{align*} \] (83)

The output of butterfly operation consists of four numbers: \( \zeta_{1r} \), \( \zeta_{1i} \), \( \zeta_{2r} \), \( \zeta_{2i} \), and a truncation and shifting error is associated with each of those numbers. To continue the analysis, the relations between the values of the four errors resulting from the butterfly must be determined. These relations will be established by considering an example in which \( \eta_1 = 0 \) and \( \eta_2 = 4 + j2 \). This situation is illustrated in Fig. 5. If the polar angle of \( \omega \left(=\tan^{-1} \frac{\omega_i}{\omega_r}\right) \) is not a multiple of \( \frac{\pi}{2} \), then \( \omega_r \) and \( \omega_i \) are irrational. The value of \( \zeta_{1r} \) can fall anywhere inside one quantization interval creating a random truncation error, and similarly the value of \( \zeta_{1i} \) can fall anywhere inside one quantization interval creating its truncation error. Furthermore, the errors associated with \( \zeta_{1r} \) and \( \zeta_{1i} \) can be assumed uncorrelated (if \( \tan^{-1} \frac{\omega_i}{\omega_r} \) is not a multiple of \( \frac{\pi}{2} \)). Using identical arguments, the errors associated with pairs \( (\zeta_{2r}, \zeta_{2i}) \), \( (\zeta_{1r}, \zeta_{2i}) \), \( (\zeta_{1i}, \zeta_{2r}) \) can be assumed uncorrelated. However, the errors of \( (\zeta_{1r}, \zeta_{2r}) \) and \( (\zeta_{1i}, \zeta_{2i}) \) are algebraically related (their sum is equal to one quantization interval if no rounding-off is performed). The relations between all truncation errors can be expressed by a correlation matrix shown below, in which \( X \) means a value different than zero.
Fig. 5 Example of a Butterfly Operation
\[ \begin{array}{c|cccc}
\zeta_{1r} & \zeta_{1i} & \zeta_{2r} & \zeta_{2i} \\
\hline
\zeta_{1r} & X & 0 & X & 0 \\
\zeta_{1i} & 0 & X & 0 & X \\
\zeta_{2r} & X & 0 & X & 0 \\
\zeta_{2i} & 0 & X & 0 & X \\
\end{array} \] (84)

The distribution for truncation errors and correlation matrix above does not change if the value of \( n_1 \) is different from zero.

For the decimation in frequency, the butterfly operation can be expressed by:

\[
\begin{align*}
\zeta_{1r} &= n_{1r} + n_{2r} \\
\zeta_{1i} &= n_{1i} + n_{2i} \\
\zeta_{2r} &= \omega_r (n_{1r} - n_{2r}) - \omega_i (n_{1i} - n_{2i}) \\
\zeta_{2i} &= \omega_i (n_{1r} - n_{2r}) + \omega_r (n_{1i} - n_{2i}) 
\end{align*}
\] (85)

From the above equations it is evident that all errors are uncorrelated, but errors for \( \zeta_{1r} \) and \( \zeta_{1i} \) are not uniformly distributed. However, those errors will be approximated by random variables with uniform distribution. This assumption will make the analysis of the decimation in frequency a little less accurate.

As was shown above, the random errors added to the signal in a given stage are not all uncorrelated for the decimation in time. However, the following argument permits assuming those errors to be uncorrelated. Consider a pair of signal points immediately after the
completion of an FFT stage, which are the results of a single butterfly operation from that stage. According to the previous analysis, each of these points has a random error added as a result of truncation and shifting. Now consider a single point of the final image. By following the signal flow in the processor, it can be seen that the noise contribution to the single point in the image can come from either of the points from the previously considered stage, but never from both of the points together. A flow diagram for the decimation in time is shown in Fig. 6. Therefore, assuming the two random errors to be uncorrelated has no effect on the final noise as far as any of the image points is concerned. This assumption allows a calculation of the final error mean and variance for any one point in the image, but prohibits calculation of any statistics relating more than one point. Following this idea, and noticing that the results of other butterflies involve multiplications by different irrational numbers, and therefore, can also be assumed uncorrelated, permits all the errors in that stage to be considered uncorrelated.

The truncation and shifting errors for the entire stage can therefore be modeled by a random vector added to the signal vector at that stage. Let $\mathbf{n}_k$ represent this vector, where $k$ indicates the $k$'th stage. The distribution of the real and imaginary part of each element of this vector is shown in Fig. 4, and all these random variables are uncorrelated. The collection of $M$ such vectors, where $M$ is the number of stages, constitutes all the noise injected by the processor in computing the image.
The noise from the top branch of second butterfly in the first stage propagates to $y(0)$, $y(2)$, $y(4)$, $y(6)$; from the bottom half to $y(1)$, $y(3)$, $y(5)$, $y(7)$.

Fig. 6 Signal Flow in Decimation in Time FFT for 8-Point Transform
The noise vectors for different stages can also be assumed uncorrelated, because the irrational Fourier coefficients are multiplied by different numbers in each stage of the transform.

Next, some inaccuracies and drawbacks of the above model are briefly described.

If the signal has a value zero at one point, then the truncation and shifting error at this point will also be zero. If a significant number of the input signal points or the image points are zero, then a considerable number of the truncation and shifting errors will also be zero, and the total noise injected by the processor will be noticeably smaller from the noise predicted by the model. Fortunately, such signals are unusual in SAR applications.

Another discrepancy of the model occurs in the first two stages of the decimation in time and in the last two stages of the decimation in frequency. In those stages no multiplication by irrational numbers occur and the only sources of noise are the shifts. This deficiency can be removed by more detailed, but very long and complicated analysis. Such analysis, however, will not change the final results significantly, and it may cloud the basic issues.
5.2 Derivation of Error Variance

Restating the definition of the random error vector for the k'th stage described in the previous section:

\[ n_k = a_k + u_k \]  \hspace{1cm} (86)

where:

\[ a_k = 2^s_k a \]

\[ a = \begin{cases} 
  0 & \text{with rounding-off} \\
  (1+j) \begin{pmatrix} \frac{-p-1}{p} \\ \frac{2s_k}{p} \end{pmatrix} & \text{no rounding-off}
\end{cases} \]

\[ E(u_k) = 0 \]

\[ u_k, u_\ell \text{ uncorrelated for } k \neq \ell \]

The elements of Re\((u_k)\) and Im\((u_k)\) are uncorrelated within each vector and between vectors, each with uniform distribution between \(-2^{-p+s_k-1}\) and \(+2^{-p+s_k-1}\). The variance of this distribution is \(2^s_k \sigma_n^2\), where

\[ \sigma_n^2 = \frac{2-2p}{12}. \]

As was argued in section 2.3, the total noise injected into the image can be approximated by an algebraic sum of the noise contributions from the individual noise vectors, if the noise vectors are much weaker than the signal. Let \(e_p\) represent the error in the image, where each element of \(e_p\) corresponds to one point of the image. Then:
\( \epsilon_p = \sum_{k=1}^{M} \epsilon_{pk} \)  \hspace{1cm} (87)

where \( \epsilon_{pk} \) is the error contribution from the \( k \)'th stage, given by:

\( \epsilon_{pk} = \frac{p_t \cdots p_t}{M} \frac{n_{k-1}}{n_k} \) \hspace{1cm} \text{dec. in time} \hspace{1cm} (88a)

\( \epsilon_{pk} = (p_{M}^{f} \cdots p_{k-1}^{f}) \Delta \frac{n_{k}}{} \) \hspace{1cm} \text{dec. in frequency} \hspace{1cm} (88b)

To separate the bias from the random noise, let

\( b_k = \mathbb{E}\{\epsilon_{pk}\} \)  \hspace{1cm} (89)

and

\( v_k = \epsilon_{pk} - b_k \)  \hspace{1cm} (90)

finally,

\( b = \sum_{k=1}^{M} b_k \)  \hspace{1cm} (91)

\( v = \sum_{k=1}^{M} v_k \)  \hspace{1cm} (92)

Before proceeding to the calculation of the variance of \( v \) (i.e. the mean square error), make the following definitions:

Let \( \xi \) be any \( N \times 1 \) complex column matrix and \( \xi^0 \) be a \( 2N \times 1 \) real column matrix related to \( \xi \) by:

\[ \begin{align*}
\xi &= \begin{bmatrix}
\xi_1^r \\
\xi_1^i \\
\vdots \\
\xi_N^r \\
\xi_N^i
\end{bmatrix} \\
\xi^0 &= \begin{bmatrix}
\xi_1^r \\
\xi_1^i \\
\vdots \\
\xi_N^r \\
\xi_N^i
\end{bmatrix}
\end{align*} \]
\[
\xi^0 = \begin{bmatrix}
\xi_{1r} \\
\xi_{1i} \\
\xi_{2r} \\
\xi_{2i} \\
\vdots \\
\vdots
\end{bmatrix}
\]

where \( \xi^0 \) is the \( \ell \)'th element of \( \xi \) and \( r \) and \( i \) indicate the real and imaginary part. Let \( \Xi \) be any \( N \times N \) complex square matrix and \( \Xi^0 \) be a \( 2N \times 2N \) real square matrix formed from \( \Xi \) by replacing each element of \( \Xi \) by a \( 2 \times 2 \) matrix given by

\[
\begin{bmatrix}
\xi_{k\ell r} & -\xi_{k\ell 1} \\
\xi_{k\ell 1} & \xi_{k\ell r}
\end{bmatrix}
\]

where \( \xi_{k\ell} \) is the \((k,\ell)\)'th element of \( \Xi \).

Using above definitions, it is easy to verify the following identities:

\[
(\Xi^{-1})^0 = (\Xi^0)^{-1}
\]

\[
(\Xi^*)^0 = (\Xi^0)^T
\]

\[
(\Xi_1 \Xi_2)^0 = \Xi_1^0 \Xi_2^0
\]

\[
(\Xi \xi)^0 = \Xi^0 \xi^0
\]
These identities together with the above definitions permit
manipulations of complex column and square matrices by using only real
matrices. To make the set of definitions complete, let \((\xi^T)^0\) be a
1 x 2N real row matrix given by:

\[
(\xi^T)^0 = (\xi^0)^T
\]  

(99)

The purpose of rearranging the real and imaginary parts of complex
matrices in the manner shown above, is to simplify the derivation of
covariance matrix of \(v\), which is designated by the symbol \(K\) and given by:

\[
K = E\{(v^*)^0 (v^0)^T\}
\]  

(100)

Because the \(v_k\)'s are uncorrelated, \(K\) can be expressed by:

\[
K = \sum_{k=1}^{M} K_k
\]  

(101)

where

\[
K_k = E\{(v_k^*)^0 (v_k^0)^T\}
\]  

(102)

Let \(\omega\) represent the noise in the memory immediately after the completion
of the last FFT stage, and before rearranging the points into the bit reversed
form for the decimation in frequency. Then:

\[
v_k = \omega_k \quad \text{dec. in time}
\]  

(103)

\[
v_k = (\omega_k)_\Delta \quad \text{dec. in frequency}
\]

and

\[
\omega_k = \frac{P}{M} \cdots P_{k+1} u_k
\]  

(104)
where superscripts of $P_k$'s have been omitted. Let

$$
\mathbb{P}_k = E\{(u_k^*)^0 (u_k^0)^T\}
$$

$$
= E\{(P_M^* \cdots P_{k+1}^*)^0 (u_k^T p_{k+1}^T \cdots p_M^T)^0\}
$$

$$
= (P_M^* \cdots P_{k+1}^*)^0 E\{(u_k^*)^0 (u_k^0)^T\}(P_{M}^T \cdots P_{k+1}^T)^0
$$

$$
= 2^{2s_k} \sigma_n^2 (P_M^* \cdots P_{k+1}^*)^0 (P_M^T \cdots P_{k+1}^T)^0
$$

$$
= 2^{2s_k} \sigma_n^2 2^{M-k} I
$$

(105)

In the above derivation, the identity (46) was used, and the fact that $E\{(u_k^*)^0 (u_k^0)^T\}$ is an identity matrix multiplied by the variance $2^{2s_k} \sigma_n^2$.

Since $\mathbb{P}_k$ is an identity matrix multiplied by a scalar, it is equal to the $\mathbb{P}_k$ matrix for both decimations. Finally,

$$
\mathbb{K} = \sum_{k=1}^{M} \sigma_n^2 2^{s_k} 2^{M-k} I
$$

$$
= \sigma_n^2 2^M \sum_{k=1}^{M} 2^{s_k-k} I
$$

(106)

The above equation shows that the assumed model predicts the elements of the final error vector to be uncorrelated, each with a variance given by:

$$
\text{Var}(\varepsilon_p) = \sigma_n^2 2^M \sum_{k=1}^{M} 2^{s_k-k}
$$

(107)
where $\varepsilon_p$ is either real or imaginary part of any element of error vector $\varepsilon$. Although according to (106) the elements of $\mathbf{v}$ are uncorrelated, this conclusion may not be true in the real case, because the assumption of uncorrelated elements of $u_k$ does not allow any statements relating different elements of $\mathbf{v}$. However, the result is valid for any single element of $\mathbf{v}$. Therefore, equation (107) can be used to measure the intensity of the noise injected by the processor.

The quantity of greater interest is the "normalized variance" given by:

$$V(\varepsilon) = \frac{\text{Var}(\varepsilon)}{2s_M}$$

which represents the distribution of the image noise independently of the final characteristic $s_M$, i.e. the square root of $V(\varepsilon)$ provides a measure of noise level in reference to a maximum image output level of one.

$$V(\varepsilon_p) = \sigma_n^2 \left( \frac{M-2s_M}{2} \right) \sum_{k=1}^{M} \frac{2s_k}{2}$$

where

$$\sigma_n^2 = \frac{1}{12} 2^{2p-p}$$

The normalized variance, which from now on will be referred to simply as the variance, depends on the set $\{s_k\}_{k=1}^{M}$ representing the characteristic and the corresponding shifts. To analyze this dependence, first consider the effect of removing one shift. Symbolically, this can be represented by:

$$s_k + s_k \quad k < n$$

$$s_k + s_k - 1 \quad k \geq n$$
where \( n \) is an arbitrary \( n \)'th stage which contains at least one shift. Let \( V(\varepsilon_p) \) be the original variance and \( V(\varepsilon^*_p) \) be the variance with the one shift removed. Then:

\[
V(\varepsilon_p) = \sigma_n^2 M^{-2s_M} \sum_{k=1}^{M} 2s_k - k
= A \left( \sum_{k=1}^{n-1} 2s_k - k + \sum_{k=n}^{M} 2s_k - k \right)
= B + C
\]

where:

\[
A = \sigma_n^2 M^{-2s_M} \quad B = A \sum_{k=1}^{n-1} 2s_k - k \quad C = A \sum_{k=n}^{M} 2s_k - k
\]

and:

\[
V(\varepsilon^*_p) = \sigma_n^2 M^{-2(s_M-1)} \left( \sum_{k=1}^{n-1} 2s_k - k + \sum_{k=n}^{M} 2(s_k-1) - k \right)
= 4B + C
\]

Since \( B > 0 \) and \( C > 0 \), then:

\[
V(\varepsilon^*_p) > V(\varepsilon_p)
\]
The above inequality indicates that removing one shift increases the variance. This result will be later referred to as property 1. Recall that the variance was normalized with respect to the peak signal, and not the mean signal power. Since the removal of one shift also increases the mean signal power, the consequences of increased error variance may be compensated by the increase in signal power. This effect will be considered in Chapter VII.

Next, consider the effect of moving a shift from an arbitrary \(n'\)th stage to the \((n + 1)'\)st stage, i.e.

\[
\begin{align*}
\sigma_n &+ \sigma_{n-1} \\
\sigma_k &+ \sigma_k & k \neq n
\end{align*}
\]

Let \(V(\varepsilon_p)\) be the original variance, \(V(\varepsilon_p^2)\) be the variance after moving one shift.

\[
V(\varepsilon_p) = \sigma_n^2 2 \left( \sum_{k=1}^{M-2s_M} \frac{M}{2} + \frac{2s_{k-k}}{2} \right)
\]

\[
= A \left( \sum_{k=1}^{M} 2s_{k-k} + 2s_{n-n} \right)_{k \neq n} + 2s_{n-n}
\]

and

\[
V(\varepsilon_p^2) = D + A 2n^{2(s_n-1)-n}
\]

\[
= D + \frac{A}{4} 2s_{n-n}
\]

where:

\[
D = A \sum_{k=1}^{M} \frac{2s_{k-k}}{2}
\]
Comparing \( V(\varepsilon_p) \) and \( V(\varepsilon_p^2) \) yields:

\[
V(\varepsilon_p) > V(\varepsilon_p^2)
\]  

(113)

This results in property 2: moving a shift one stage later decreases the error variance.

From the above results, the bounds on the variance can be obtained. By applying property 1, it is evident that the largest variance will occur for the signal which causes no shifts, i.e., \( s_k = 0 \). Substituting into (109) yields:

\[
V(\varepsilon_p)_{\text{max}} = \sigma_n^2 2^M \sum_{k=1}^{M} 2^{-k}
\]

\[
= \sigma_n^2 2^M (1-2^{-M})
\]

(114)

\[
= \sigma_n^2 (2^M - 1)
\]

The smallest variance will occur for the signal which causes \((M + 1)\) shifts. The last stage must have two shifts, otherwise moving one shift to the last stage would further decrease the variance, as was shown in property 2. In all other stages only one shift is allowed. Thus \( s_k = k \) for \( k < 1 \), \( s_M = M + 1 \), and the minimum variance is

\[
V(\varepsilon_p)_{\text{min}} = \sigma_n^2 (2^{M-2(M+1)}) \left( \sum_{k=1}^{M-1} 2^{2k-k} + 2^{2(M+1)-M} \right)
\]
\[-87-\]

\[= \sigma_n^2 \left( 2^{-M-2} \sum_{k=1}^{M-1} 2^k + 2^{M+2} \right)\]

\[= \sigma_n^2 (\frac{5}{4} - 2^{-M-1})\]  

(115)

**Rewriting above results:**

\[V(c)_{\text{max}} = \sigma_n^2 (2^M - 1)\]  

(116)

\[V(c)_{\text{min}} = \sigma_n^2 (\frac{5}{4} - 2^{-M-1})\]  

(117)

---

### 5.3 Derivation of Error Bias

Rewriting pertinent equations:

\[b = \sum_{k=1}^{M} b_k\]  

(118a)

\[b_k = p_{M}^{t} \cdots p_{k+1}^{t} a_k \quad \text{dec. in time}\]  

(118b)

\[b_k = \left( p_{M}^{f} \cdots p_{k+1}^{f} a_k \right) \Delta \quad \text{dec. in frequency}\]  

(118c)

\[a_k = 2^{s_k} a_1\]  

(118d)

\[a = (1+j) 2^{-p-1}\]  

(118e)
Decimation in time

Let \( d_k \) be a signal which in ideal case would produce \( a_k \) in the \( k \)'th stage:

\[
d_k = (p_{-1}^t \cdots p_{-k}^t)^{-1} a_k \tag{119}
\]

then

\[
b_k = p_{-1}^t \cdots p_{-M}^t d_k
\]

and, according to (45)

\[
b_k = f_M [(d_k)_\Delta] \tag{120}
\]

where symbols \( f_M \) and \( \Delta \) were defined in section 4.1.

The signal \( d_k \) can be derived easily by noticing that \( (p_{-1}^t \cdots p_{-k}^t)^{-1} \) is equivalent to \( 2^{M-k} \) parallel inverse Fourier transforms (section 4.1). Let \( c_k \) be a column matrix of dimension \( 2^k \), given by:

\[
c_k = 2^s_k \ a \ 1 \tag{121}
\]

and let \( \delta_k \) be related to \( c_k \) by:

\[
\delta_k = [f_k^{-1} (c_k)] \tag{122}
\]

then \( d_k \) is a concatenation of \( \delta_k \) matrices:
\[
\delta_k = \begin{bmatrix}
\delta_k \\
\delta_k \\
\vdots \\
\vdots \\
\end{bmatrix}
\] (123)

Calculating \( \delta_k \) yields:

\[
\delta_k = [F_k^{-1} (2^{s_k} a 1)]_\Delta
\]

\[
= 2^{s_k} a \begin{bmatrix}
1 \\
0 \\
\vdots \\
0 \\
\end{bmatrix}
\] (124)

The reversed bit form of \( d_k \) becomes

\[
(d_k)_\Delta = 2^{s_k} a \begin{bmatrix}
1 \\
0 \\
\vdots \\
0 \\
\end{bmatrix}
\] (125)

Define \( \gamma_k \) as a column matrix given by:

\[
\gamma_k
\]
\[ Y_k = \begin{pmatrix} 1 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 2^M - 2^k \end{pmatrix} \text{ elements} \] (126)

Substituting into (120) produces:

\[ b_k = f_M [2^{s_k} a Y_{M-k}] \] (127)

Finally, summing up the \( b_k \)'s from all the stages yields:

\[ b = \sum_{k=1}^{M} f_M (2^{s_k} a Y_{M-k}) = f_M \{ a \sum_{k=1}^{M} 2^{s_k} Y_{M-k} \} \] (128)

Decimation in frequency:

As was shown in section 4.1, the matrix \( (P^f_M \ldots P^f_{k+1}) \) consists of \( 2^k \) parallel Fourier transforms. Therefore, the bias can be expressed by the following identity:

\[ b_k = \begin{pmatrix} \delta_k \\ \vdots \\ \delta_k \\ \vdots \\ \Delta \end{pmatrix} \] (129)
where

\[ \delta'_k = [F_{M-k} \ (c'_k)]_\Delta \]  \hspace{1cm} (130)

and \( c'_k \) is a column matrix of dimension \( 2^{M-k} \) given by

\[ c'_k = 2^k \ a \ 1 \]  \hspace{1cm} (131)

Evaluation of \( \delta'_k \) yields:

\[ \delta'_k = [F_{M-k} \ (2^k \ a \ 1)]_\Delta \]

\[
\begin{bmatrix}
1 \\
0 \\
\vdots \\
0 \\
\end{bmatrix} = 2^k \ a \ 2^{M-k} \\
\begin{bmatrix}
1 \\
0 \\
\vdots \\
0 \\
\end{bmatrix} \Delta \\
\begin{bmatrix}
1 \\
0 \\
\vdots \\
0 \\
\end{bmatrix}
\]

Substituting into (129) yields:

\[ b_k = 2^{M+s_k-k} \ a \ \gamma_k \]  \hspace{1cm} (133)

where \( \gamma_k \) is given by (126).
Finally,

\[ b = a \sum_{k=1}^{M} (2^{M-k}) 2^{s_k} y_k \]  \hspace{1cm} (134)

Let \( b_* \) denote the normalized bias defined by:

\[ b_* = 2^{-s_M} b \]  \hspace{1cm} (135)

and let superscripts \( t \) and \( f \) denote the decimation in time and the decimation in frequency respectively.

Then,

\[ b_{t*} = F_M \{ a \sum_{k=1}^{M} (2^{s_k-s_M}) y_{M-k} \} \]  \hspace{1cm} (136a)

\[ b_{f*} = a \sum_{k=1}^{M} (2^{M-k}) (2^{s_k-s_M}) y_k \]  \hspace{1cm} (136b)

From this point on, the normalized bias will be referred to simply as bias, in the same way as normalized variance is referred to as variance.

Let \( E_b \) be the energy of the bias defined by:

\[ E_b = b_{t*}^T b_{f*} \]  \hspace{1cm} (137)

then \( E_b \) is given by:
\[ E_b = |a|^2 \sum_{k=1}^{2(M-s_M)} \frac{2^{2s_k-k}}{2} \]  

for both decimations. An interesting relation which can be obtained by comparing (138) with (109) is

\[ \frac{(E_b/N)}{2\mu(c)} = \frac{|a|^2}{2\sigma^2_n} = 3 \]  

An important property of the bias vector \( b_\star \) is the value of the element of largest magnitude in \( b_\star \). Let the symbol \( b_0 \) denote this element. For the decimation in time:

\[ b_{\star}^t = a \sum_{k=1}^{M} 2^{s_k-s_M} f_M(x_{M-k}) \]

Using identity (29) yields:

\[ f_M(x_k) = \left\{ \sum_{n=0}^{2^{k-1}} e^{\frac{-j2\pi n\ell}{N}} \right\}_{\ell=0}^{N-1} \]

This set has its maximum at \( \ell = 0 \) for all \( k \). Therefore,

\[ b_0 = a \sum_{k=1}^{M} 2^{s_k-s_M} 2^{M-k} \]  

For the decimation in frequency, \( b_0 \) also corresponds to the first element of \( b_\star \), and is given by the same equation. Therefore, for both decimations, the largest value of bias is given by:
To analyze the dependence of $b_0$ on the set $\{s_k\}_{k=0}^M$, a derivation similar to the one used in analyzing the variance has to be carried out. The results of such derivation show that the properties 1 and 2 of section 5.2 also hold true for $b_0$. Thus $(b_0)_{\text{max}}$ corresponds to $s_k = 0$ for all $k$ and $(b_0)_{\text{min}}$ corresponds to $s_k = k$ for $k < M$ and $s_M = M + 1$. Substituting into (141) yields:

\[
\begin{align*}
(b_0)_{\text{max}} &= a(2^M - 1) \\
(b_0)_{\text{min}} &= \frac{a}{2}(M+1)
\end{align*}
\]
CHAPTER VI

GAUSSIAN IMAGE

6.1 Gaussian Image Assumption

The equations derived in Chapters 3 and 5 can be used to estimate the quantization errors due to (1) the initial quantization of the signal and (2) the truncations during the processing. These equations are given in terms of the quantities which depend on the signal. To apply these equations, a statistical model of the signal will be used. This model will be derived from a probabilistic model of the true image, i.e., without the quantization distortions. In section 1.3, the model of the image was reduced to a set of $N$ complex random samples described by a joint probability density function. In this section several more simplifying assumptions will be introduced.

The first assumption is to make all the samples independent. A processor designed for images which satisfy this idealized assumption, will well resolve images consisting of many sharp points and edges which are not arranged in strongly periodic patterns. The consequences of this assumption will be discussed in greater detail in later sections. Next, assume that each point of the image is a result of scattering from many objects located inside one resolution area on the ground. Furthermore, assume that all objects inside that area scatter with equal intensity and their average separation is of the order of one
wave-length. This results in a well known Rayleigh distribution of
the amplitude and a uniform distribution of the phase, with the
amplitude and the phase independent.

\[ p_r(r) = \frac{r}{\sigma^2} e^{-r^2/2\sigma^2} \]  \hspace{1cm} (143)

\[ p_\theta(\theta) = \begin{cases} \frac{1}{2\pi} & -\pi \leq \theta \leq \pi \\ 0 & \text{otherwise} \end{cases} \]  \hspace{1cm} (144)

where \( z = re^{i\theta} \) is an image point, \( r \) is the amplitude with the
standard deviation \( \sigma \), \( \theta \) is the phase, and \( p_r(r) \) and \( p_\theta(\theta) \) represent
probability density functions. An important aspect of this idealized
model is that the resulting distributions are equivalent to the
\textit{independent} Gaussian distributions for the real and imaginary parts
of \( z \), and in this way greatly simplify the subsequent analysis. The
approximations used so far are reasonable for a preliminary study.
Later, some deviations from the Gaussian image will be considered.

The joint probability density function of \( r \) and \( \theta \) is simply
the product \( p_r(r) p_\theta(\theta) \). Let \( z_r = \text{Re}(z) \) and \( z_i = \text{Im}(z) \). The
joint density function of \( z_r \) and \( z_i \) becomes:

\[ p_{z_r z_i}(z_r, z_i) = \frac{1}{|\det J|} p_r([z_r^2 + z_i^2]^{1/2}) p_\theta(\tan^{-1}(z_i/z_r)) \]
\[
\frac{1}{r} \left( \frac{1}{\sqrt{2\pi} \sigma} e^{-z_1^2 / 2\sigma^2} \right) \left( \frac{1}{\sqrt{2\pi} \sigma} e^{-z_1^2 / 2\sigma^2} \right)
\]

where \( J_f \) is the Jacobian matrix

\[
J_f = \begin{bmatrix}
\cos \theta & -r \sin \theta \\
\sin \theta & r \cos \theta
\end{bmatrix}
\]

Equation (145) indicates that \( z_1 \) and \( z_1 \) are identically distributed independent zero mean Gaussian random variables with the standard deviation \( \sigma \). Let \( y \) be an \( N \times 1 \) complex column matrix representing the image. Such one-dimensional image representation was discussed in Section 1.3. The joint probability density function of the elements of \( y \) is a jointly Gaussian density, frequently represented by:

\[
p_y(y) = N(m_y, \Lambda_y)
\]

where \( m_y \) is the mean and \( \Lambda_y \) is the covariance matrix of \( y \). For the case discussed above, the mean and the covariance are given by:

\[
m_y = 0
\]

\[
\Lambda_y = \sigma^2 I
\]
The image model can be made more general by assuming an arbitrary covariance matrix $\Lambda_y$. The actual form of $\Lambda_y$ depends on how the real and imaginary parts of $y$ are arranged. If the definition (5) is used, then $\Lambda_y = E[y^c y^{cT}]$ and if the definition (84) is used then $\Lambda_y = E[y^0 y^{0T}]$.

In agreement with the previous chapters, let $x$ be the received signal, which in the model is the inverse Fourier transform of $y$. Since the Fourier transform is a linear operation, the probability density function of $x$ is also jointly Gaussian, completely described by its covariance matrix $\Lambda_x$.

$$p_x(x) = N[0, \Lambda_x]$$  \hspace{1cm} (148)

Using the definitions (4), (5) and (6) of Chapter 3, the signal $x$ is related to $y$ by:

$$x^c = F^{-1} y^c$$

and $\Lambda_x$ is given by:

$$\Lambda_x = E[x^c x^{cT}]$$

$$= F^{-1} E[y^c y^{cT}] (F^{-1})^T$$

$$= F^T \Lambda_y F$$  \hspace{1cm} (149)

where in the last step the equation (8) was used. The above relations can be viewed as a two dimensional inverse Fourier transform; i.e. $\Lambda_x$ can be obtained from $\Lambda_y$ by first taking the Fourier transform of all the columns of $\Lambda_y$ and then of all the rows of the resulting matrix.
If $A_y = \sigma^2 I$, then $A_x$ is

$$A_x = F^T \sigma^2 I F = \sigma^2 I$$ (150)

Therefore, an image consisting of independent Gaussian samples results in a signal also consisting of independent Gaussian samples.

### 6.2 Quantizer Noise

The total error introduced into the image can be approximated by an algebraic sum of the initial quantization error and the error injected by the FFT processor (section 1.3). In Chapter 3 the formula for initial quantization errors was derived, and in Chapter 5 the errors injected by the processor were calculated. In this section, the results of Chapter 3 will be applied to the Gaussian image.

Using the definition (5), the joint probability density function and the characteristic function of $y$ become:

$$P_y(y^c) = \frac{1}{(2\pi)^N |A_y|^{2N}} e^{-\frac{1}{2} y^c A_y^{-1} y^c}$$ (151)

$$M_y(y) = e^{-\frac{1}{2} y^T A_y y}$$ (152)

where:

$$A_y = E\{y^c y^c^T\}$$ (153)

For the horizontal quantization (see Fig. 2) the error is given by (23).
\[ \varepsilon_{qk} = \frac{q^2}{12} + 2 \sum_{\ell=1}^{2N} \frac{\alpha_{k\ell}}{n} \left[ \sum_{n=-\infty}^{\infty} \frac{(-1)^n}{n^2} M_y(-\omega_n \beta_{\ell}) \right] \\
\qquad - \frac{2N}{p} \sum_{\ell=1}^{2N} \frac{\alpha_{k\ell}}{\omega} \frac{\alpha_{kp}}{\omega} \left[ \sum_{n=-\infty}^{\infty} \frac{(-1)^{n+m}}{n m} M_y(-\omega_n \beta_{\ell} - \omega_m \beta_p) \right] \]

where:

\[ \alpha_k = F^T e_k \]

\[ \beta_k = F e_k \]

Substituting (152) into parts of (23) yields:

\[ M_y(-\omega_n \beta_{\ell}) = \exp\left\{ -\frac{1}{2} \omega_n^2 \frac{\beta_{\ell}}{n} \Lambda_y \beta_{\ell} \right\} \]

\[ = \exp\left\{ -\frac{1}{2} \omega_n^2 e_{\ell}^T F \Lambda_y F e_{\ell} \right\} \]

\[ = \exp\left\{ -\frac{1}{2} \omega_n^2 e_{\ell}^T \Lambda_{x} e_{\ell} \right\} \]

\[ = \exp\left\{ -\frac{1}{2} \omega_n^2 \lambda_{\ell\ell}^x \right\} \quad (154) \]

where \( \lambda_{mn}^x \) is the \((m,n)\)'th element of \( \Lambda_x \), and \( \Lambda_x \) is given by (149).

Substituting (152) into the argument of the second summation...
\[
M_y (\omega n \beta_\ell - \omega m \beta_p) = \exp\left\{ -\frac{1}{2} \omega^2 \left[ \beta_\ell^T \gamma \beta_\ell + \beta_p^T \gamma \beta_p + nm (\beta_\ell^T \gamma \beta_\ell + \beta_p^T \gamma \beta_p) \right] \right\} \\
= \exp\left\{ -\frac{1}{2} \omega^2 \left( n^2 \lambda_{\ell\ell}^x + m^2 \lambda_{pp}^x \right) \right\} \exp\{-\omega^2 n \ m \ \lambda_{\ell p}^x\} \tag{155}
\]

and,

\[
\sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \frac{(-1)^{n+m}}{n \ m} M_y (\omega n \beta_\ell - \omega m \beta_p) \\
\sum_{n \neq 0} \sum_{m \neq 0}
\]

\[
= \sum_{n=1}^{\infty} \sum_{m=1}^{\infty} \frac{(-1)^{n+m}}{n \ m} \exp\left\{ -\frac{1}{2} \omega^2 \left( n^2 \lambda_{\ell\ell}^x + m^2 \lambda_{pp}^x \right) \right\} \exp\{-\omega^2 n \ m \ \lambda_{\ell p}^x\} \tag{156}
\]

Substituting above into (23) yields:

\[
\frac{e^2}{q_k} = \frac{a^2}{12} + 4 \sum_{L=1}^{2N} \left( \frac{\alpha_{kL}}{\omega} \right)^2 \sum_{n=1}^{\infty} \frac{(-1)^n}{n^2} e^{-\frac{1}{2} \omega^2 \lambda_{\ell\ell}^x} n^2
\]

\[
+ 4 \sum_{L=1}^{2N} \sum_{p=1}^{2N} \frac{\alpha_{kL}}{\omega} \frac{\alpha_{kp}}{\omega} \sum_{n=1}^{\infty} \sum_{m=1}^{\infty} \frac{(-1)^{n+m}}{n \ m} e^{-\frac{1}{2} \omega^2 (n^2 \lambda_{\ell\ell}^x + m^2 \lambda_{pp}^x)} \sinh (\omega^2 n \ m \ \lambda_{\ell p}^x) \tag{157}
\]
For the vertical quantization the resulting expression is very similar:

\[
\overline{\epsilon_{qk}^2} = \frac{q^2}{12} + 4 \sum_{\ell=1}^{2N} \left( \frac{\alpha_{k\ell}}{\omega} \right)^2 \sum_{n=1}^{\infty} \frac{1}{n^2} e^{-\frac{1}{2} \omega^2 \lambda_{x,\ell}^2} n^2
\]

\[+ 4 \sum_{\ell=1}^{2N} \sum_{p=1 \atop p \neq \ell}^{2N} \left( \frac{\alpha_{k\ell}}{\omega} \right) \left( \frac{\alpha_{kp}}{\omega} \right) \sum_{n=1}^{\infty} \sum_{m=1}^{\infty} \frac{1}{n m} \]

\[\cdot \frac{1}{2} \omega^2 (n^2 \lambda_{x,\ell\ell}^x + m^2 \lambda_{x,pp}^x) \sinh(\omega^2 n m \lambda_{x,\ell p}^x) \quad (158)\]

If the image points are independent and of same variance, then \( \Lambda_x = \sigma^2 \mathbb{I} \) as was shown in (150). Assuming \( \sigma = 1 \), and setting \( \lambda_{x,\ell\ell}^x = 1 \) and \( \lambda_{x,\ell p}^x = 0 \) in the above expression for the horizontal quantization results in:

\[
\overline{\epsilon_{qk}^2} = \frac{q^2}{12} + 4 \sum_{\ell=0}^{2N} \left( \frac{\alpha_{k\ell}}{\omega} \right)^2 \sum_{n=1}^{\infty} \frac{(-1)^n}{n^2} e^{-\frac{1}{2} \omega^2 n^2}
\]

\[= \frac{q^2}{12} + 4 \sum_{n=1}^{\infty} \frac{(-1)^n}{n^2} \frac{1}{\omega^2} e^{-\frac{1}{2} \omega^2 n^2}
\]

\[
\overline{\epsilon_{qk}^2} = q^2 \left[ \frac{1}{12} + \frac{1}{\pi^2} \sum_{n=1}^{\infty} \frac{(-1)^n}{n^2} e^{-2\pi^2 n^2 / q^2} \right] \quad (159)
\]

Similarly, for the vertical quantization:
\[
\varepsilon_{qk}^2 = q^2 \left( \frac{1}{12} + \frac{1}{\pi^2} \sum_{n=1}^{\infty} \frac{1}{n^2} e^{-2\pi^2 n^2/q^2} \right)
\] (160)

An important property of the above expressions is that the second term is only significant for the values of \( \frac{q}{\sigma} \) greater than 3. In the actual applications the magnitude of \( q \) will be much smaller, therefore the mean square error \( \varepsilon_{qk}^2 \) can be accurately approximated by:

\[
\varepsilon_{qk}^2 = \frac{q^2}{12}
\] (161)

An identical result could have been obtained if the initial quantization error was modeled by a random noise added to the input signal, where the amount added to each point of the signal was independent with uniform distribution from \(-\frac{q}{2}\) to \(+\frac{q}{2}\). Since equations (159) and (160) are exact, the above model can be used for an independent Gaussian image, and any deviations from (161) are caused by the deviations of the image from the Gaussian assumption [see comment following equation (23)].

Another important property of the above expression is the uniformity of the error, i.e., all points of the image contain the same amount of noise.

For small values of \( q \), both expressions for \( \varepsilon_{qk}^2 \) are equal to \( \frac{q^2}{12} \). For large values of \( q \), the second term of (160) becomes:
\[ \frac{q^2}{\pi^2} \sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{q^2}{\pi^2} \left( \frac{\pi^2}{6} \right) \]  

(162)

and \( \varepsilon_{qk}^2 \) for vertical quantization becomes \( \frac{q^2}{4} \), a value that agrees with a simple hand-calculation. In the case of horizontal quantization, the limit of \( \varepsilon_{qk}^2 \) becomes:

\[
\lim_{q \to \infty} \varepsilon_{qk}^2 = \lim_{q \to \infty} q \left( \frac{1}{12} + \frac{1}{\pi} \sum_{n=1}^{\infty} \frac{(-1)^n}{n^2} e^{-\frac{2\pi^2}{q^2}} \right)
\]

\[ = \lim_{\omega \to 0} \frac{4}{\omega} \left( \frac{1}{12} + \sum_{n=1}^{\infty} \frac{(-1)^n}{n^2} e^{-\frac{1}{2} \omega^2 n^2} \right) \]

\[ = \lim_{\omega \to 0} \sum_{n=1}^{\infty} (-1)^n \frac{1}{\omega^2 n^2} \left( e^{-\frac{1}{2} (\omega n)^2} - 1 \right) \]

\[ = \lim_{\omega \to 0} \sum_{n=1}^{\infty} (-1)^n g(\omega n) \]

where:

\[ g(x) = \frac{4}{x} \left( e^{-\frac{1}{2} x^2} - 1 \right) \]

and \( \frac{\pi}{12} \) was converted into an infinite summation shown below:

\[ \frac{\pi}{12} = - \sum_{n=1}^{\infty} \frac{(-1)^n}{n^2} \]

Let \( \omega = \Delta x \). Then
\[ \lim_{q \to \infty} \varepsilon_{qk}^2 = \lim_{\Delta x \to 0} \sum_{n=1}^{\infty} \frac{g(2n\Delta x) - g(2n\Delta x - \Delta x)}{\Delta x} \Delta x \]

\[ = \int_{0}^{\infty} g'(2x) \, dx \]

\[ = -\frac{1}{2} g(2x) \bigg|_{0}^{\infty} \]

\[ = -\frac{1}{2} \frac{4}{4x^2} \left( (1 - \frac{1}{2} 4x^2 + \cdots) - 1 \right) \bigg|_{0}^{\infty} \]

\[ = 1 \quad (163) \]

This result is one check of the correctness of (159).

The equation (163) was derived using a symmetric definition for the discrete Fourier transform given by (3). In deriving the variance and the bias for the processor error, a nonsymmetric definition (29) was used. The difference between the two definitions is only in the scale factor \(2^{M/2}\). Therefore, the variance due to the initial quantization with the nonsymmetric definition (29) is simply \(2^M \frac{q^2}{12}\) and the normalized variance \(V(\varepsilon_q)\), analogous to the definition (109) for \(V(\varepsilon_p)\), is given by:

\[ V(\varepsilon_q) = 2^M \frac{q^2}{12} \quad (164) \]
6.3 Distribution of Standard Deviation

The remaining four sections of this chapter are devoted to analyzing the errors injected by the processor in computing a Gaussian image with independent points. The equations for the variance \( V(\epsilon_p) \), the bias \( b_\epsilon \), and the largest bias point \( b_\sigma \), all depend on the set of characteristics \( \{ s_k \}_{k=0}^M \). The values of \( s_k \)'s depend on the mean amplitude of the stored signal, and the properties of that amplitude are the topic of this section.

As was shown in (150), an independent-point Gaussian image, which from now on will be called a white Gaussian image, corresponds to a white Gaussian signal. The signal has to be stored in \( N \) block-floating-point registers. Suppose that the initial characteristic \( s_0 \) is equal to zero, and let \( \sigma_X \) be the standard derivation of the signal. Calculate the probability that this signal can be stored in the registers without a need to change the characteristic \( s_0 \). This is equivalent to calculating the probability that \( H_0 \), which is the largest magnitude of the elements of \( \mathbf{x}^0 \), defined by (51), is between \( \frac{1}{2} \) and 1. The \( \mathbf{x}^0 \) represents the signal according to (84). Using an error function defined by:

\[
erfc(t_0) = \int_{t_0}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt
\]  

(165)

the probability that \( \frac{1}{2} < H_0 < 1 \) becomes:
\[
\Pr\left(\frac{1}{2} \leq H_0 < 1 \mid \sigma_x\right) = \Pr\left(H_0 < 1 \mid \sigma_x\right) - \Pr\left(H_0 < \frac{1}{2} \mid \sigma_x\right)
\]

\[
= \left[1 - 2 \operatorname{erfc}\left(\frac{1}{\sigma_x}\right)\right]^{2N} - \left[1 - 2 \operatorname{erfc}\left(\frac{1}{2\sigma_x}\right)\right]^{2N} \quad (166)
\]

Define \(f(\sigma_x)\) to be:

\[
f(\sigma_x) = \Pr\left(\frac{1}{2} \leq H_0 < 1 \mid \sigma_x\right)
\]

Then for the Gaussian distribution:

\[
f(\sigma_x) = \left[1 - 2 \operatorname{erfc}\left(\frac{1}{\sigma_x}\right)\right]^{2N} = \left[1 - 2 \operatorname{erfc}\left(\frac{1}{2\sigma_x}\right)\right]^{2N} \quad (168)
\]

The graph of \(f(\sigma_x)\) for \(N = 2^{10}\) is shown in Fig. 7. The individual points of the graph were obtained from the following approximations:

Let:

\[
\beta(\sigma_x) = \left[1 - 2 \operatorname{erfc}\left(\frac{1}{\sigma_x}\right)\right]^{2N} \quad (169)
\]

Taking logarithm of both sides:

\[
\frac{1.15}{N} \log \beta(\sigma_x) = \ln[1 - 2 \operatorname{erfc}(1/\sigma_x)]
\]

where \(\ln\) is the natural logarithm, and \(\log\) is the logarithm of base 10. Expanding in a power series:
Fig. 7 Graph of $f(\sigma_X)$ for Gaussian Distribution
\[
\frac{1.15}{N} \log \beta(\sigma_x) = -[2 \text{ erfc}(1/\sigma_x)] - \frac{1}{2} [2 \text{ erfc}(1/\sigma_x)]^2 - \cdots
\]

The magnitude of the left-hand side of the above equation is very small for values of \(\beta(\sigma_x)\) between 0.1 and 1, and \(N = 2^{10}\). Therefore, taking only the first term of the power expansion results in a very good approximation.

\[
\frac{1.15}{N} \log \frac{1}{\beta(\sigma_x)} = 2 \text{ erfc}(1/\sigma_x)
\]

\[
\beta(\sigma_x) = 10 \frac{2N}{1.15} \text{ erfc}(1/\sigma_x)
\]

\[
f(\sigma_x) = 10 \frac{2N}{1.15} \text{ erfc}(1/\sigma_x) - 10 \frac{2N}{1.15} \text{ erfc}(1/2\sigma_x)
\]

In the actual computation, equation (170) was more useful than equation (171). The interpretation of \(f(\sigma_x)\) is as follows: If the standard deviation \(\sigma_x\) of the signal is between 0.16 and 0.26 then the signal can be contained in the block-floating-point registers without any shifts with very high probability. If \(\sigma_x\) is greater than 0.32, then most likely it will have to be scaled down by one or more shifts, and if \(\sigma_x\) is smaller than 0.12, then it will have to be scaled up. The standard deviation for which the probability of scaling down is \(\frac{1}{2}\) is equal to twice the standard deviation for which the probability of scaling up is \(\frac{1}{2}\).
For larger $N$ the function $f(\sigma_x)$ comes closer to the origin, and its sides become steeper. However, the $f(\sigma_x)$ is rather insensitive to large changes of $N$, providing that $N$ is large enough. In Fig. 8 the cases of $N = 2^{10}$ and $N = 2^6$ are compared.

![Graph showing $f(\sigma_x)$ for $N = 2^6$ and $N = 2^{10}$](image)

Fig. 8. Function $f(\sigma_x)$ for $N = 2^6$ and $N = 2^{10}$

Another probability function, which is as important as $f(\sigma_x)$, is the distribution of the standard deviation $\sigma_x$ given that $H_0$ is between $\frac{1}{2}$ and 1. Suppose that the received signal has some standard deviation $\sigma_x$. As this signal is loaded into the memory, the initial characteristic $s_o$ is adjusted so that at least one sample has 1 in its most-significant bit. The resulting standard deviation of the signal contained in the memory can only have values given by:
\[ \sigma_x = \frac{\sigma_r}{2^i} \quad i = 0, \pm 1, \pm 2, \ldots \quad (172) \]

The probability that \( \sigma_x \) has a particular value \( \sigma_r/2^i \) is:

\[ p_r(\sigma_x = \sigma_r/2^i) = \Pr\left(\frac{1}{2} \leq H_0 < 1 \mid \sigma_x = 2^{-i} \sigma_r\right) = f(2^{-i} \sigma_r) \quad (173) \]

Therefore, the probability density function of \( \sigma_x \) is given by:

\[ p_{\sigma_x}(\sigma_x \mid \sigma_r) = \sum_{i=-\infty}^{\infty} f(2^{-i} \sigma_r) \delta(\sigma_x - 2^{-i} \sigma_r) \quad (174) \]

where \( \delta(\ ) \) is a Dirac delta function.

The standard deviation of the input signal \( \sigma_r \) can be adjusted by varying the gain to obtain the best performance. The issues involved in adjusting \( \sigma_r \) are discussed later; for now, assume that \( \sigma_r \) can have any value. Since the actual value of \( s_0 \) is unimportant (it merely represents a scale factor which does not affect the quality of the image), the probability distribution of \( \sigma_r \) need to be specified only in the interval \([\frac{1}{2}, 1)\). If the actual \( \sigma_r \) falls outside this interval, then changing \( s_0 \) by an appropriate integer will always yield a value inside this interval. Furthermore, the distribution of \( \sigma_r \) should not be changed for different \( s_0 \); that is, dividing all samples by \( 2^{-i} \), \( i = 0, \pm 1, \pm 2, \ldots \) should not change the form of the probability density function of \( \sigma_r \). Mathematically, this can be expressed by:
\[ p_{\sigma_r}(\sigma_r) = \frac{p_{\sigma_r}(2^i \sigma_r)}{\int p_{\sigma_r}(2^i \sigma_r) \, d\sigma_r} \quad i = 0, \pm 1, \pm 2, \ldots \]  

(175)

where \( p_{\sigma_r}(\sigma_r) \) is the probability density function of \( \sigma_r \). Integrating the denominator and letting \( c = 2^i \) yields:

\[ p_{\sigma_r}(\sigma_r) = c \, p_{\sigma_r}(c \, \sigma_r) \]  

(176)

Differentiating with respect to \( c \) and solving the resulting differential equation produces:

\[ p_{\sigma_r}(c \, \sigma_r) + c \, p'_{\sigma_r}(c \, \sigma_r) \, \sigma_r = 0 \]

\[ \frac{p'_{\sigma_r}(c \, \sigma_r)}{p_{\sigma_r}(c \, \sigma_r)} = - \frac{1}{c \, \sigma_r} \]

\[ \ln p_{\sigma_r}(\sigma_r) = - \ln \sigma_r + \text{const.} \]

\[ p_{\sigma_r}(\sigma_r) = \frac{\text{const.}}{\sigma_r} \]  

(177)

The constant of integration can be evaluated by normalizing \( p_{\sigma_r}(\sigma_r) \).

The final result is:
\[ p_{\sigma_r}(\sigma_r) = \frac{1}{\sigma_r \ln 2} ; \quad \frac{1}{2} \leq \sigma_r < 1 \]  

(178)

The density given by (178) is shown in Fig. 9. For probability density function of \( \sigma_r \) shown in Fig. 9, the probability density function of \( \sigma_x \) becomes:

\[ p_{\sigma_x}(\sigma_x) = \int_{-\infty}^{+\infty} p_{\sigma_x}(\sigma_x | \sigma_r) \ p_{\sigma_r}(\sigma_r) \ d\sigma_r \]

\[ = \sum_{i=-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(2^{-i} \sigma_r) \delta(\sigma_x - 2^{-i} \sigma_r) \ p_{\sigma_r}(\sigma_r) \ d\sigma_r \]

\[ = \sum_{i=-\infty}^{\infty} 2^i \int_{-\infty}^{+\infty} f(2^{-i} \sigma_r) \ p_{\sigma_r}(\sigma_r) \ \delta(\sigma_x - 2^{-i} \sigma_r) \ d(2^{-i} \sigma_r) \]

\[ = \sum_{i=-\infty}^{\infty} 2^i \ f(\sigma_x) \ p_{\sigma_r}(2^i \sigma_x) \]

\[ = f(\sigma_x) \sum_{i=-\infty}^{\infty} 2^i \frac{1}{2^i \sigma_x \ln 2} \quad \left[ \frac{1}{2} \leq 2^i \sigma_x < 1 \right] \]

\[ = \frac{f(\sigma_x)}{\sigma_x \ln 2} \]

(179)

The function \( p_{\sigma_x}(\sigma_x) \) for the case of \( N = 2^{10} \) is shown in Fig. 10 as the upper boundary of the shaded area.
Fig. 9 Probability Density Function of $\sigma_r$.

Fig. 10 Probability Density Function of $\sigma_x$. 

6.4 Signal in Block-Floating-Point Memory

The function \( f(\sigma_x) \) plays an important role in analyzing the shift patterns in FFT. In the previous section \( f(\sigma_x) \) was derived for a white Gaussian signal. To study the dependence of \( f(\sigma_x) \) on a particular distribution, consider a signal with independent, identically distributed points, but with an arbitrary distribution. For such a signal the function \( f(\sigma_x) \) can be expressed by an equation analogous to (171).

\[
f(\sigma_x) = 10 \left( \frac{N}{1.15} t(1/\sigma_x) \right) - 10 \left( \frac{N}{1.15} t(1/2\sigma_x) \right) \tag{180}
\]

where:

\[
t(t_o) = \int_{t_o}^{\infty} [p_x(\tau) + p_x(-\tau)] d\tau \tag{181}
\]

and \( p_x(\tau) \) is the probability density function of the signal. Due to large value of the coefficients \( \frac{N}{1.15} \) in the exponents of (180), the function \( f(\sigma_x) \) switches between 0 and 1 in short intervals of \( \sigma_x \). This can be expressed more quantitatively by a ratio:

\[
\frac{\alpha_1 - \alpha_2}{\alpha_2} \tag{182}
\]

where \( \alpha_1 \) and \( \alpha_2 \) are solutions to:
\begin{align}
0.1 &= 10 \frac{N}{1.15} t(1/\alpha_1) \\
0.9 &= 10 \frac{N}{1.15} t(1/\alpha_2)
\end{align} (183a, 183b)

If the ratio \( \frac{\alpha_1 - \alpha_2}{\alpha_2} \) is small then \( f(\alpha_x) \) has a shape similar to the one shown in Fig. 7. Closer examination of (181) and (183a), (183b) shows that the probability density functions which yield the smallest ratios are those whose values decrease most rapidly for large values of \( x \). To analyze a bad case, consider the Poisson distribution. The tail of this distribution decreases rather slowly:

\[ t(t_o) = e^{-t_o} \] (184)

and the ratio for \( N = 2^{10} \) is 0.44. For comparison, the form of \( f(\sigma_x) \) for Gaussian and Poisson distribution are shown in Fig. 11.

![Fig. 11 f(\sigma_x) for Poisson and Gaussian distributions](image-url)
The shape of \( f(\sigma_x) \) does not change much for different distributions. In many of the subsequent derivations the shape of \( f(\sigma_x) \) is important, but the actual distance of the interval where \( f(\sigma_x) \approx 1 \) from the origin is not important. Therefore, assuming the Gaussian distribution will produce accurate results even if the actual distribution is not Gaussian. The important properties of \( f(\sigma_x) \) are:

a) There is a significant interval of \( \sigma_x \) in which \( f(\sigma_x) \approx 1 \).

b) The transitions of \( f(\sigma_x) \) from 0 to 1 and from 1 to 0 (or more accurately, from 0.1 to 0.9, and from 0.9 to 0.1) usually occur in small intervals of \( \sigma_x \). The interval for the transition 1 \( \rightarrow \) 0 is twice as large as for the transition 0 \( \rightarrow \) 1.

c) The transition 1 \( \rightarrow \) 0 occurs twice as far from the origin as the transition 0 \( \rightarrow \) 1.

The function \( f(\sigma_x) \) can be schematically represented by the diagram shown in Fig. 12. The values of \( a \) and \( b \) in Fig. 12 are given by:

\[
\frac{\alpha_1 - \alpha_2}{2} = 2b \tag{185}
\]

\[
\frac{\alpha_1 + \alpha_2}{2} = 2a \tag{186}
\]

where \( \alpha_1 \) and \( \alpha_2 \) were defined in (183).
For the distributions with slowly decreasing tails the significant region of \( f(\sigma_x) \) moves closer to the origin. For smaller \( N \) the significant region of \( f(\sigma_x) \) moves away from the origin, and the transition regions become wider. If the points of the signal are not completely independent, but have a narrow covariance function, then the significant region also moves away from the origin and the transition regions increase.

The number of bits per word of memory required to store the signal accurately depends on the signal standard deviation \( \sigma_x \). The mean amplitude of the quantizer noise which distorts the signal is of the order of the least-significant bit. For smaller \( \sigma_x \) and a fixed r.m.s.
signal-to-noise ratio, the value of the least-significant bit must be decreased. This is equivalent to adding more bits. As was shown in Fig. 11, for some distributions \( f(\sigma_x) \) has non-zero values for very small \( \sigma_x \). Since

\[
p_{\sigma_x}(\sigma_x) = \frac{f(\sigma_x)}{\sigma_x \ln 2}
\]

the resulting values of \( \sigma_x \) for those distributions will be also small. This situation can be redeemed by allowing the values of the signal to be clipped above a certain range. This eliminates the occurrences of very large magnitudes, which otherwise would cause additional left shifts, resulting in a much smaller standard deviation. This also improves the shape of \( f(\sigma_x) \).

The error introduced by the clipping can be estimated from the probability density function of the signal points. On the average, this error spreads uniformly throughout the final image, and can be modeled by an independent additive noise. The largest tolerable value of this noise can be used to determine the clipping level.
6.5 Shift Patterns

Continuing with the previously stated assumptions of white Gaussian input signal and small distortions due to truncations, the distribution for the $z_k$ can be derived as follows: Let $z_k$ be the signal in the k'th stage. Since $z_k$ is related to $x$ through a matrix multiplication, the distribution of elements of $z_k$ is also Gaussian, completely described by the covariance matrix $\Lambda_{z_k}$.

$$ p_{z_k}(z_k) = N\{0, \Lambda_{z_k}\} \quad (187) $$

and

$$ \Lambda_{z_k} = E\{z_k^o z_k^{oT}\} \quad (188) $$

Using equations (45) and (46), and identities (88), (89), and (90):

$$ \Lambda_{z_k} = (p_k^o \cdots p_1^o) E\{z_0^o z_0^{oT}\} (p_1^{oT} \cdots p_k^{oT}) $$

$$ = (p_k^o \cdots p_1^o) \sigma_x^2 I (p_k^{*o} \cdots p_1^{*o}) $$

$$ = \sigma_x^2 (p_k^o \cdots p_1^o) (g^k) [(p_1^{-1})^o \cdots (p_k^{-1})^o] $$

$$ = 2^k \sigma_x^2 I \quad (189) $$
The above equation indicates that $z_{k}$ is a white Gaussian signal with variance $\sigma_{z}^{2}$ given by:

$$
\sigma_{z}^{2} = 2^{k} \sigma_{x}^{2}
$$

(190)

The probability that $s_{k}$ has a particular value $i$ is thus given by:

$$
\Pr\{s_{k} = i\} = \Pr\{2^{i-1} \leq H_{k} < 2^{i}\}
$$

$$
= \Pr\{H_{k} < 2^{i}\} - \Pr\{H_{k} < 2^{i-1}\}
$$

$$
= [1 - 2\text{erfc}(2^{i}/\sigma_{z})]^{2N} - [1 - 2\text{erfc}(2^{i-1}/\sigma_{z})]^{2N}
$$

$$
= f(\sigma_{z}/2^{i})
$$

$$
= f(2^{-i+k/2} \sigma_{x})
$$

(191)

where $H_{k}$ was defined in (51) and (60). Equation (191) is sufficient for analyzing the properties of $s_{k}$.

Suppose that $\sigma_{x}$ is inside the significant region of $f(\sigma_{x})$ (Fig. 12). According to (191), the probability that $s_{2}$ is equal to $i$ is $f(2^{i-1} \sigma_{x})$. Thus the probability that $s_{2} = 1$ is one (assuming $f(\sigma_{x})$ shown in Fig. 12) and the probability that $s_{2} \neq 1$ is zero. Similarly, $s_{4} = 2$, $s_{6} = 3$, and so on. The values of $s_{1}$, $s_{3}$, $s_{5}$,... depend on which part of the significant region contains the value of $\sigma_{x}$. 
If $\sigma_x$ is near the end of the region on the origin side then $s_1 = 0$, $s_3 = 1$, $s_5 = 2$, ..., and if it is near the end away from the origin, then $s_1 = 1$, $s_3 = 2$, $s_5 = 3$. For values of $\sigma_x$ close to the center of the significant region, there will be some non-zero probability that $s_k = \frac{k-1}{2}$ and some non-zero probability that $s_k = \frac{k+1}{2}$ for $k = 1, 3, 5, ...$.

From the above discussion it is easy to see that if $\sigma_x$ falls into the first transition region then $s_k$ is equal to $\frac{k-1}{2}$ for odd $k$, and with finite probabilities $s_k$ is either $\frac{k-2}{2}$ or $\frac{k}{2}$ for even $k$. Finally, for $\sigma_x$ in the second transition region $s_k = \frac{k+1}{2}$ for odd $k$, and $s_k = \frac{k}{2}$ or $s_k = \frac{k+2}{2}$ for even $k$.

The above results can be summarized in the following description of the possible shift patterns. Each pattern is represented by a sequence of numbers. Each number corresponds to one stage, and is equal to the probability of one shift in that stage. The left-most number corresponds to the first stage, the right-most number corresponds to the last stage. In all patterns shown below, $a$ and $b$ satisfy the following conditions:

$$0 < a, b < 1$$ \hspace{1cm} (192)

$$a + b = 1$$ \hspace{1cm} (193)
\sigma_x \text{ left transition region} \\
\phantom{0} 0 \ a\ b\ a\ b\ a\ b \ldots \quad (194a)

\sigma_x \text{ left side of the significant region} \\
\phantom{0} 0\ 1\ 0\ 1\ 0\ 1\ldots \quad (194b)

\sigma_x \text{ center of the significant region} \\
\phantom{0} a\ b\ a\ b\ a\ b\ a\ b\ a\ b \ldots \quad (194c)

\sigma_x \text{ right side of the significant region} \\
\phantom{0} 1\ 0\ 1\ 0\ 1\ 0 \ldots \quad (194d)

\sigma_x \text{ right transition region} \\
\phantom{0} 1\ a\ b\ a\ b\ a\ b\ a\ b \ldots \quad (194e)

6.6 Error Variance and Bias

The white Gaussian image model can now be used to derive the error variance and the bias injected by the processor into the computed image. The shift pattern used in the derivation is (194d) for even M. Using properties 1 and 2 of Section 5.2, it can be easily verified that for even M this pattern results in a variance \( V(\epsilon_p) \) (normalized with respect to the peak signal) which is larger than variances of patterns represented by (194b) and (194c). Patterns represented by (194a) and (194e) can in rare cases produce larger variance. This fact, combined with the fact that \( \sigma_x \) is less likely to be in one of the transition regions than in significant region, makes the occurrence of larger variance quite infrequent. Therefore, assuming pattern (194d) corresponds to images which create relatively large errors, and occur frequently enough to be considered.
The sequence \( \{s_k\} \) is given by:

\[
\begin{align*}
    s_k &= \begin{cases}
        \frac{k+1}{2} & k = 1, 3, 5, \ldots \\
        \frac{k}{2} & k = 2, 4, 6, \ldots
    \end{cases} \\
    &\quad (195)
\end{align*}
\]

where (1) refers to the sequence \( k = 1, 3, 5, \ldots \) and (2) refers to the sequence \( k = 2, 4, 6, \ldots \). Substituting into (109) yields the variance:

\[
\begin{align*}
    \operatorname{V}(\varepsilon_p)_{av} &= \sigma_n^2 \left[ 2^{M-2(M/2)} \right] \left( \sum_{k=1}^{(1)} 2^{(k+1)-k} + \sum_{k=2}^{(2)} 2^{k-k} \right) \\
    &= \sigma_n^2 \left[ 2 \sum_{k=1}^{(1)} + \sum_{k=2}^{(2)} \right] \\
    &= \frac{3}{2} M \sigma_n^2 \\
    &\quad (196)
\end{align*}
\]

Substituting into (141) yields the largest bias point:

\[
\begin{align*}
    (b_o)_{av} &= a 2^{M/2} \left( \sum_{k=1}^{(1)} 2^{k+1} - k + \sum_{k=2}^{(2)} 2^k - k \right) \\
    &= a 2^{M/2} \left( 2^{M/2} - \sum_{k=1}^{M/2} 2^{-k} + \sum_{k=1}^{M/2} 2^{-k} \right) \\
    &= 3 a (2^{M/2} - 1) \\
    &\quad (197)
\end{align*}
\]
CHAPTER VII

PRECISION REQUIREMENTS

7.1 Signal to Noise Ratio

The model of the processor derived in Chapter 1 consisted of two memories: a large signal memory where the received signal is stored, and a small arithmetic memory, where the signal is processed. The signal memory is responsible for the initial quantization errors. The arithmetic memory is assumed to have more bits per word than the signal memory.

After the signal is processed in the arithmetic memory, it can be stored into either a third "image" memory, or back into the signal memory. For generality, the image memory will be considered different from the signal memory.

A major design problem is to determine the number of bits per word of memory required in (1) the signal memory, (2) the arithmetic memory, and (3) the image memory (if different from the signal memory). An often used prespecified design criterion is the smallest tolerable signal-to-noise ratio of the final image, designated by $R$ and usually expressed in decibels by:

$$20 \log_{10} R$$  \hspace{1cm} (198)

$R$ is assumed to be the ratio of the r.m.s. signal to the r.m.s. noise, and therefore can be expressed in terms of the standard
deviation of the image points and the square root of the total mean-square error. Using variables normalized with respect to the peak signal \(2^{-s_M}\) the signal to noise ratio becomes:

\[
R = \left( \frac{\bar{v}(y)}{\bar{e}^2} \right)^{1/2}
\]

(199)

where \(\bar{v}(y)\) is the variance of \(y\) and \(\bar{e}^2\) is the mean-square error. Taking the logarithm of both sides and defining \(p_r\) to be

\[
p_r = \log_2 R
\]

(200)

results in:

\[
\frac{1}{2} \log_2 \bar{e}^2 - \frac{1}{2} \log_2 \bar{v}(y) = p_r
\]

(201)

This inequality related the mean-square error to the signal power.

Before inequality (201) can be used to bound the mean-square error of the image, the signal power \(\sigma_y^2\) has to be determined. First, assume that the standard deviation of the received signal \(\sigma_r\) is not controlled, and therefore can have any value with equal probability. This assumption was used in deriving equation (179). Using this assumption and equations (179), (190) and (191), the probability density function of the normalized image variance \(\bar{v}(y)\) can be derived as follows: \(\bar{v}(y)\) is given by:

\[
\bar{v}(y) = 2^{-2s_M} \sigma_y^2
\]

(202)
Using equation (190)

\[ V(y) = 2^{M-2s} \frac{\sigma^2_x}{\sigma^2} \]  \(203\)

The probability density function of \(V(y)\) given a value of \(\sigma_x\) is:

\[ p_V[V^{1/2}(y) | \sigma_x] = \sum_i \Pr(s_M = i | \sigma_x) \delta[2^{M/2} - 1 - V^{1/2}(y)] \]  \(204\)

The probability density function of \(V(y)\) can therefore be expressed by:

\[ p_V[V^{1/2}(y)] = \int p_V[V^{1/2}(y) | \sigma_x] p_{\sigma_x}(\sigma_x) \, d\sigma_x \]  \(205\)

Substituting (179) and (191) yields:

\[ p_V[V^{1/2}(y)] = \int \sum_i \delta[2^{M/2} - i \sigma_x - V^{1/2}(y)] \frac{f(\sigma_x)}{\sigma_x \ln 2} \, d\sigma_x \]

\[ = f(V^{1/2}(y)) \left( \sum_i \frac{f(2^{-M/2} + i \frac{V^{1/2}(y)}{\sigma_y \ln 2})}{2^{M/2} + i} \right) \]  \(206\)

The summation in the brackets has only one or two significant terms.

However, the important property of equation (206) is the fact that \(V^{1/2}(y)\) can only have values for which \(f(V^{1/2}(y)) \neq 0\). Since the signal to noise ratio has to be greater than \(R\) even for the smallest \(V^{1/2}(y)\), the mean-square error has to be restricted by:

\[ \frac{1}{2} \log_2 \frac{1}{\epsilon^2} \leq \log_2 [V^{1/2}(y)]_{\text{min}} - p_r \]  \(207\)

where \([V^{1/2}(y)]_{\text{min}}\) is equal to the smallest value of \(V^{1/2}(y)\) for which \(f(V^{1/2}(y)) \neq 0\) in Fig. 12. For a Gaussian image and \(N = 2^{10}\).
\[
[V^{1/2}(y)]_{\text{min}} = 0.12 \quad \text{and} \quad \log_2 [V^{1/2}(y)]_{\text{min}} = -3.1.
\]

Inequality (207) was derived for \( \sigma_r \) not controlled. If \( \sigma_r \) can be controlled, then the bound on the mean-square error can be made less restrictive. The most desirable standard deviation for a signal which is limited in range is usually taken to have a value equal to one third of the range. Since the range of the image was normalized to 1, the variance of \( y \) has to be less than \((\frac{1}{3})^2\).

Unfortunately, it is difficult to adjust the input signal variance \( \sigma_x^2 \) so that \( V(y) = (\frac{1}{3})^2 \), because the final characteristic \( s_M \) will in general have non-zero probability of being equal to one of two consecutive integers. Therefore, a safety factor of 2 has to be used and the inequality (201) becomes:

\[
\frac{1}{2} \log_2 \varepsilon^2 \leq \log_2(1/6) - p_r
\]

(208)

Inequalities (207) and (208) can be expressed by the following: Let \( p_m \) be given by:

\[
p_m = \log_2 \text{(peak margin factor)}
\]

(209)

where the peak margin factor is chosen by the designer. If \( \sigma_r \) is controlled, then a good value for the peak margin factor is 6, otherwise a good value is equal to \([V(y)]_{\text{min}}^{-1/2}\). Finally:

\[
\frac{1}{2} \log_2 \varepsilon^2 \leq p_m - p_r
\]

(210)

Let \( p_1 \) be the number of bits per word of the image memory. Thus the quantization noise of the image memory can be approximated
by \( \frac{1}{\sqrt{12}} 2^{-p_1} \). To determine the value of \( p_1 \), it seems reasonable to set this quantization noise to be of the order of one half of the total mean-square error of the processor, that is:

\[
\frac{1}{12} 2^{-2p_1} = \epsilon^2/2 \tag{211}
\]

Substituting (211) into (210) yields:

\[
p_1 \geq p_m - p_r - 1.8 + \frac{1}{2} \tag{212}
\]

Let

\[
p_d = p_m + p_r - 1.3 \tag{213}
\]

then

\[
p_1 \geq p_d \tag{214}
\]

7.2 Gaussian Image

In this section the results of the previous chapter are converted into the precision requirements. The relations below are expressed in terms of the following symbols representing the number of bits per word of memory.

\[
p_i = \text{number of bits in image memory}
\]

\[
p_s = \text{number of bits in signal memory}
\]

\[
p_p = \text{number of bits in arithmetic memory} \tag{215}
\]

The values of \( p_i \), \( p_s \), and \( p_p \) can all be different; in general \( p_i \) is equal to \( p_s \) and \( p_p \) is greater than both \( p_i \) and \( p_s \). In the previous
section a method for determining $p_1$ was presented. Now the relation between $p_s$, $p_p$, and $p_1$ will be derived, followed by a summary of the results.

The error variance due to the initial quantization, given by (164) when applied to the special case of the shift pattern (194d) used in the section 6.6 becomes:

$$V(\varepsilon_q)_{av} = \frac{q^2}{12}$$

(216)

The quantization step size $q$ is simply:

$$q = 2^{-p_s}$$

(217)

The errors due to the initial quantization and the processor truncation are therefore characterized by the following equations:

$$V(\varepsilon_q)_{av} = \frac{1}{12} 2^{-2p_s}$$

(218)

$$V(\varepsilon_p)_{av} = \frac{3}{2} M \left( \frac{1}{12} 2^{-2p_p} \right)$$

(219)

and the largest bias point is given by

$$(b_o)_{av} = \frac{3}{2} \left( 2^{M/2} - 1 \right) 2^{-p_p}$$

(220)

where only the real part of $(b_o)_{av}$ is considered.

The processor may be designed to calculate the bias $b_o$ and subtract it in the final operation. If this is done, then the total error is characterized by the sum of $V(\varepsilon_q)$ and $V(\varepsilon_p)$. If the bias is not subtracted then the $b_o$ must also enter the error calculations. The subsequent analysis is done for the three cases, depending on the presence and the origin of the bias in the image:
(a) no bias, because all values have been rounded-off.
(b) no bias, because it has been subtracted.
(c) with bias.

A reasonable assignment of the prespecified tolerable error is to divide it equally between the initial quantization errors and the processor truncation errors. For cases (a) and (b) this can be expressed by:

\[ V(e_q)_{av} = V(e_p)_{av} \quad \text{case (a, b)} \quad (221) \]

For case (c) the bias is represented by its largest magnitude point \( b_0 \) given by (141). Thus:

\[ V(e_q)_{av} = [v^{1/2}(e_p)_{av} + (b_0)_{av}]^2 \quad \text{case (c)} \quad (222) \]

By making the two errors approximately equal, the relations between \( p_s \) and \( p_p \) can be obtained. Substituting equations (218), (219), and (220) yields:

\[ \sqrt{1/12} 2^{-p_s} = \sqrt{3M/2} \sqrt{1/12} 2^{-p_p} \quad \text{case (a, b)} \quad (223) \]

\[ \sqrt{1/12} 2^{-p_s} = \sqrt{3M/2} \sqrt{1/12} 2^{-p_p} + \frac{3}{2} (2^{M/2} - 1) 2^{-p_p} \quad \text{case (c)} \quad (224) \]

Taking the logarithm of both sides of (223) and letting:

\[ p_n = \frac{1}{2} \log_2 (3M/2) \quad (225) \]

results in:

\[ p_p - p_s = p_n \quad \text{case (a, b)} \quad (226) \]
Rearranging equation (224) produces:

\[ 2^{-p_p} = [\sqrt{3M/2} + \sqrt{1/12} \ (3/2) \ (2^{M/2} - 1)] \ 2^{-p_p} \]

The ratio of the first term in the brackets to the second term decreases with increasing M. For \( M = 6 \) it is less than 0.1.

Let:

\[ p_o = \log_2 [\sqrt{3M/2} + 3^{3/2} (2^{M/2} - 1)] \] \hspace{1cm} (227)

then:

\[ p_p - p_s = p_o \quad \text{case (c)} \] \hspace{1cm} (228)

To determine the relation between \( p_s \) and \( p_p \), consider first the case (a), in which bias does not occur. In equation (211) half of the tolerable mean-square error was assigned to the quantization noise of the image memory. Therefore, the remaining half corresponds to the initial quantization and truncation errors:

\[ \frac{1}{2} \overline{\epsilon^2} = V(\epsilon_q) + V(\epsilon_p) \]

\[ = 2V(\epsilon_p) \]

\[ = 2(3M/2) (1/12) 2^{-2p_p} \] \hspace{1cm} (229)
Substituting into (210), and using the definition of \( p_n \) (225) yields:

\[
p_p = p_m + p_r - 1.3 + p_n + \frac{1}{2}
\] (230)

Replacing the first three terms with (213) gives:

\[
p_p = p_d + p_n + \frac{1}{2} \quad \text{case (a)}
\] (231)

Next, consider the case (b) in which the bias is subtracted in the final operation. This case is very similar to the case (a). However, the largest bias point \( b_o \) can become large enough to cause additional shifts, and in this way decrease the signal to noise ratio. This can be prevented by using enough bits to contain the sum of both the bias and the signal. Mathematically, this corresponds to replacing the value of \( p_d \) in expression (231) with:

\[
\max(p_d, p_b) + 1
\] (232)

where:

\[
p_b = \log_2[(3/2) (2^m/2 - 1)]
\] (233)

The 1 added to \( \max(p_d, p_b) \) in the expression (232) protects against the shift in the case when the DC value of the signal reaches its maximum. Since this is very unlikely, a more reasonable substitution is:

\[
\max(p_d, p_b + 1)
\] (234)

Using this in expression (231) results in:

\[
p_p = \max(p_d, p_b + 1) + p_n + \frac{1}{2} \quad \text{case (b)}
\] (235)
Finally, for the case (c) where the bias is not subtracted, the normalized mean-square error is:

\[
\frac{1}{2} \varepsilon^2 = 2 \left[ V(e_p)_{av} + (b_0)_{av} \right]^2
\]

\[
= 2 \left[ \sqrt{3M/2} + 3^{3/2} \left( 2^{M/2} - 1 \right) \right]^2 \frac{1}{12} \frac{1}{2} 2^{-2p_p} \quad (236)
\]

Substituting into (210), and using the definition (227) yields:

\[
p_p = p_m + p_r - 1.3 + p_o + \frac{1}{2} \quad (237)
\]

Replacing the first three terms with (213) gives:

\[
p_p = p_d + p_o + \frac{1}{2} \quad \text{case (c)} \quad (238)
\]

A summary of these results appears on the next page.
Summary

\[ p_1 \geq p_d \]
\[ p_s \geq p_d + \frac{1}{2} \]
\[ p_p \geq p_d + p_n + \frac{1}{2} \quad \text{case (a)} \]
\[ p_p \geq \max(p_d, p_b + 1) + p_n + \frac{1}{2} \quad \text{case (b)} \]
\[ p_p \geq p_d + p_o + \frac{1}{2} \quad \text{case (c)} \]

where

\[ p_d = p_m + p_r + \frac{1}{2} \left[ 1 - \log_2 (12) \right] \]
\[ p_n = \frac{1}{2} \log_2 (3M/2) \]
\[ p_o = \log_2 \left[ \sqrt{3M/2} + 3^{3/2} (2^{M/2} - 1) \right] \]
\[ p_b = \log_2 \left[ (3/2) (2^{M/2} - 1) \right] \]
\[ p_m = - \log_2 \left[ V^{1/2} (y)_{\min} \right] \]
\[ p_r = \log_2 R \]

To compare the significance of the various quantities in the summary above, let \( R = 100, p_m = \log_2 6, \) and \( M = 10. \) Then

\[ p_r = 6.7 \quad (240a) \]
\[ p_m = 2.6 \quad (240b) \]
\[ p_b = 5.6 \quad (240c) \]
\[ p_o = 7.4 \quad (240d) \]
\[ p_n = 2.0 \quad (240e) \]
\[ p_d = 8.0 \quad (240f) \]
Substituting above into expressions for \( p_i, p_p, p_s \) and rounding-off to the next higher integer yields:

\[
\begin{align*}
p_i &= 8 & (241a) \\
p_s &= 9 & (241b) \\
p_p &= 11 \text{ case } (a, b) & (241c) \\
p_p &= 16 \text{ case } (c) & (241d)
\end{align*}
\]

For \( M = 6 \) and the same values of \( R \) and \( p_m \):

\[
\begin{align*}
p_b &= 3.4 & (242a) \\
p_o &= 5.3 & (242b) \\
p_n &= 1.6 & (242c)
\end{align*}
\]

And the values of \( p_i, p_s, \) and \( p_p \) become:

\[
\begin{align*}
p_i &= 8 & (243a) \\
p_s &= 9 & (243b) \\
p_p &= 11 \text{ case } (a, b) & (243c) \\
p_p &= 14 \text{ case } (c) & (243d)
\end{align*}
\]

7.3 Comments

The expressions for \( p_i, p_p, \) and \( p_s \) in the previous section were derived for an idealized image consisting of independent points whose real and imaginary parts were independent Gaussian random variables. These expressions are a good approximation to the actual situation only if the images processed do not contain bright periodic patterns or single very bright points. Presence of bright periodic patterns will decrease the number of shifts and the presence of single bright points will increase the number of shifts. If the power of such periodic patterns or
single bright points is significant, then the resulting errors may be
better described by $V(c_p)^{\text{max}}$ and $V(c_p)^{\text{min}}$ respectively.

The regular patterns of shifts shown in (194) suggest that
a possible way of implementing the FFT processor is to have a fixed shift
pattern.

$$101010\ldots$$ (244)

This in general would require a few more bits per word in the memory,
but would simplify the construction and increase the speed of the
processor. It would also permit easy elimination of the image bias,
because the bias can be precomputed and simply subtracted in the final
operation.

Another possible approach to the problem of eliminating the bias
is to have the bias precomputed for several different shift patterns.
The number of such precomputed bias signals can be reduced by noticing
that only the last few shifts are significant. Also fixing the shift
pattern for the last few stages attains the same result. Since subtracting
the bias significantly reduces the number of bits for the processor, finding
a good procedure to accomplish this can be beneficial.

Most of the results derived in this paper are based on the
assumption of an independent additive noise. Since the actual distortions
are not independent and not additive, the obtained results are not good
for small number of bits or very weak signals. The derivations for the
error bias may be inaccurate even for a large number of bits and strong
signals, because the mathematical model used in less faithful for those
derivations. To verify the results for error bias, a simulation test may
be required. The remaining results can be used only after the assumptions
have been verified.
Appendix A

Derivation of Equation (23)

The mean square error of the k'th element of \( y^C \) is given by:

\[
\varepsilon_{qk}^{-2} = - \left( \frac{\partial^2}{\partial \nu_1^2} - 2 \frac{\partial^2}{\partial \nu_1 \partial \nu_2} + \frac{\partial^2}{\partial \nu_2^2} \right) M_e(\nu_1, \nu_2) \bigg|_{\nu_1 = \nu_2 = 0} \tag{245}
\]

\[
M_e(\nu_1, \nu_2) = \sum_n S(\nu_2 \alpha_k - \omega n) M_y[(\nu_1 + \nu_2) e_k - \omega F_n] \tag{246}
\]

where all the symbols above are defined in Chapter 3. Differentiating \( M_e(\nu_1, \nu_2) \) yields:

\[
\frac{\partial^2}{\partial \nu_1 \partial \nu_2} M_e(\nu_1, \nu_2)
\]

\[
= \sum_n \frac{\partial}{\partial \nu_2} S(\nu_2 \alpha_k - \omega n) \frac{\partial}{\partial \nu_1} M_y[(\nu_1 + \nu_2) e_k - \omega F_n]
\]

\[
+ \sum_n S(\nu_2 \alpha_k - \omega n) \frac{\partial^2}{\partial \nu_1 \partial \nu_2} M_y[(\nu_1 + \nu_2) e_k - \omega F_n] \tag{247}
\]

\[
\frac{\partial^2}{\partial \nu_2^2} M_e(\nu_1, \nu_2)
\]

\[
= \sum_n \frac{\partial^2}{\partial \nu_2^2} S(\nu_2 \alpha_k - \omega n) M_y[(\nu_1 + \nu_2) e_k - \omega F_n]
\]
\[ + \sum_{n} \frac{3}{\partial \nu_2} S(\nu_2 \alpha_k - \omega_n) \frac{3}{\partial \nu_2} M_y[(\nu_1 + \nu_2)e_k - \omega \Phi_n] \]

\[ + \sum_{n} S(\nu_2 \alpha_k - \omega_n) \frac{\partial^2}{\partial \nu_2^2} M_y[(\nu_1 + \nu_2)e_k - \omega \Phi_n] \quad (248) \]

Letting

\[ M_{ij}(n) = \frac{\partial^2}{\partial \nu_1^i \partial \nu_2^j} M_y[(\nu_1 + \nu_2)e_k - \omega \Phi_n] \bigg|_{\nu_1 = \nu_2 = 0} \]

\[ (249) \]

reduces above expressions to:

\[ \frac{\partial^2}{\partial \nu_1 \partial \nu_2} M_e(\nu_1, \nu_2) \bigg|_0 \]

\[ = \sum_{n} \frac{3}{\partial \nu_2} S(\nu_2 \alpha_k - \omega_n) \bigg|_{0} M_{10}(n) \]

\[ + \sum_{n} S(-\omega_n) M_{11}(n) \quad (250) \]

\[ \frac{\partial^2}{\partial \nu_2^2} M_e(\nu_1, \nu_2) \bigg|_0 \]

\[ = \sum_{n} \frac{\partial^2}{\partial \nu_2^2} S(\nu_2 \alpha_k - \omega_n) \bigg|_{0} M_{00}(n) \]
\[ + 2 \sum_{n} \frac{\partial}{\partial \nu_2} \left. S(\nu_2 \alpha_k - \omega n) \right|_0^{M_{01}(n)} \]

\[ + \sum_{n} S(-\omega n) M_{02}(n) \]  \hspace{1cm} (251)

The derivative of \( \text{sinc}(x) \) is:

\[ \frac{d}{dx} \text{sinc}(x) = \frac{1}{x} [\cos(\pi x) - \text{sinc}(x)] \]  \hspace{1cm} (252)

Let:

\[ \phi_i = \frac{\alpha_k \nu_2}{\omega} - n_i, \quad i = 1, 2, \ldots, 2N \]  \hspace{1cm} (253)

Applying the identity (252) and the definition of the \( S(\nu_1 \alpha_k - \omega n) \) produces:

\[ \frac{\partial}{\partial \nu_2} S(\nu_2 \alpha_k - \omega n) = 2N \sum_{\ell=1}^{2N} \left( \frac{\alpha_k \ell}{\omega} \right) \frac{1}{\phi_\ell} \left[ \cos(\pi \phi_\ell) - \text{sinc}(\phi_\ell) \right] \sum_{p=1}^{2N} \frac{1}{\text{sinc}(\phi_p)} \]  \hspace{1cm} (254)

\[ \frac{\partial^2}{\partial \nu_2^2} S(\nu_2 \alpha_k - \omega n) = 2N \sum_{\ell=1}^{2N} \sum_{p=1}^{2N} \left( \frac{\alpha_k \ell}{\omega} \right) \left( \frac{\alpha_k p}{\omega} \right) \frac{1}{\phi_\ell \phi_p} \]

\[ \left[ \cos(\pi \phi_\ell) - \text{sinc}(\phi_\ell) \right] \left[ \cos(\pi \phi_p) - \text{sinc}(\phi_p) \right] \prod_{r=1}^{2N} \frac{1}{\text{sinc}(\phi_r)} \]
\[ + \sum_{\ell=1}^{2N} -2 \left( \frac{\alpha_k}{\omega} \right)^2 \frac{1}{\phi_\ell} \left[ \cos(p\phi_\ell) - \text{sinc}(\phi_\ell) \right] \prod_{p=1}^{2N} \frac{\text{sin}(\pi \phi_p)}{\text{sinc}(\phi_p)} \]

\[ - \sum_{\ell=1}^{2N} \left( \frac{\alpha_k}{\omega} \right)^2 \frac{1}{\phi_\ell} \pi \frac{\text{sin}(\pi \phi_\ell)}{\text{sinc}(\phi_\ell)} \prod_{p=1, p \neq \ell}^{2N} \frac{\text{sinc}(\phi_p)}{\text{sinc}(\phi_p)} \]  

(255)

Setting \( \nu_1 = \nu_2 = 0 \) and noticing that:

\[ \frac{1}{n} \left[ \cos(\pi n) - \text{sinc}(n) \right] = \begin{cases} \frac{(-1)^n}{n} & n \neq 0 \\ 0 & n = 0 \end{cases} \]  

(256)

\[ \frac{1}{n^2} \left[ \cos(\pi n) - \text{sinc}(n) \right] = \begin{cases} \frac{(-1)^n}{n^2} & n \neq 0 \\ \frac{\pi^2}{3} & n = 0 \end{cases} \]  

(257)

\[ \delta(-\omega_n) = \sum_{k=1}^{2N} \delta(n_k) \]  

(258)

\[ \delta(n_k) = \begin{cases} 1 & n_k = 0 \\ 0 & n_k \neq 0 \end{cases} \]  

(259)
Results in:

\[
\frac{\partial}{\partial \nu_2} S(v_2 \alpha_k - \omega n) \bigg|_0 = \sum_{\ell=1}^{2N} - \left( \frac{\alpha_{k\ell}}{\nu_2} \right) f_1(n_\ell) \sum_{p=1}^{2N} \delta(n_p) \prod_{r=1}^{2N} \delta(n_r) \prod_{p \neq \ell} \delta(n_p) \prod_{r \neq p, \ell} \delta(n_r) \tag{260}
\]

\[
\frac{\partial^2}{\partial \nu_2^2} S(v_2 \alpha_k - \omega n) \bigg|_0 = \left( \frac{\alpha_{k\ell}}{\nu_2} \right)^2 f_2(n_\ell) - \frac{\pi^2}{2} \delta(n_\ell) \prod_{p=1}^{2N} \delta(n_p) \prod_{p \neq \ell} \delta(n_p) \prod_{r \neq p, \ell} \delta(n_r) \prod_{r \neq p, \ell} \delta(n_r) \tag{261}
\]

where:
\[
\begin{align*}
\frac{\partial^2}{\partial \nu_1 \partial \nu_2} M_e(\nu_1, \nu_2) \bigg|_0 &= \frac{2N}{\omega} \sum_{\ell=1}^{2N} \left( \frac{a_{k\ell}}{\omega} \right) \sum_{n_{\ell}} f_1(n_{\ell}) M_{10}(n_{\ell}) + M_{11}(0) \\
\frac{\partial^2}{\partial \nu_2^2} M_e(\nu_1, \nu_2) \bigg|_0 &= \sum_{\ell=1}^{2N} \sum_{p=1}^{2N} \left( \frac{a_{k\ell}}{\omega} \right) \left( \frac{a_{kp}}{\omega} \right) \sum_{n_{\ell}} \sum_{n_p} f_1(n_{\ell}) f_1(n_p) M_{00}(n_{\ell}, n_p) \\
&- 2 \sum_{\ell=1}^{2N} \left( \frac{a_{k\ell}}{\omega} \right)^2 \sum_{n_{\ell}} f_2(n_{\ell}) M_{00}(n_{\ell})
\end{align*}
\]
\begin{equation}
+ 2 \sum_{\ell=1}^{2N} \left( \frac{\alpha_{k\ell}}{\omega} \right) \sum_{n_{\ell}} f_1(n_{\ell}) M_{01}(n_{\ell})
\end{equation}

\begin{equation}
+ M_{02}(0)
\end{equation}

where:

\begin{equation}
M_{1j}(n_{\ell}) = M_{1j}(n_{\ell} e_{\ell})
\end{equation}

\begin{equation}
M_{1j}(n_{\ell}, n_p) = M_{1j}(n_{\ell} e_{\ell}) + n_p e_{p}
\end{equation}

Computing $M_{11}(0)$:

\begin{equation}
M_{11}(0) = \frac{a^2}{\frac{\partial}{\partial v_1} \frac{\partial}{\partial v_2}} M_y \left[ (v_1 + v_2) e_k \right] \bigg|_0
\end{equation}

\begin{equation}
= \frac{a^2}{\frac{\partial}{\partial (v_1 + v_2)^2}} M_y \left[ (v_1 + v_2) e_k \right] \bigg|_0
\end{equation}

\begin{equation}
= - \lambda_k^Y
\end{equation}

where $\lambda_k^Y$ is the variance of the k'th element of $\textbf{y}^c$. From the above equation it is evident that:
\[ M_{11}(0) = M_{20}(0) = M_{02}(0) \]  

(269)

and those three terms cancel out in (245). Therefore, the equation for \( \varepsilon_{qk}^2 \) reduces to:

\[
\varepsilon_{qk}^2 = - \sum_{\ell=1}^{2N} \sum_{p=1}^{2N} \left( \frac{\alpha_{k\ell}}{\omega} \right) \left( \frac{\alpha_{kp}}{\omega} \right) \sum_{n_\ell} f_1(n_\ell) f_1(n_p) M_{00}^{(n_\ell, n_p)}
\]

\[
+ 2 \sum_{\ell=1}^{2N} \left( \frac{\alpha_{k\ell}}{\omega} \right)^2 \sum_{n_\ell} f_2(n_\ell) M_{00}^{(n_\ell)}
\]

\[
+ 2 \sum_{\ell=1}^{2N} \left( \frac{\alpha_{k\ell}}{\omega} \right) \sum_{n_\ell} f_1(n_\ell) \left[ M_{01}^{(n_\ell)} - M_{10}^{(n_\ell)} \right]
\]  

(270)

The last term of the above expression is zero because:

\[
\frac{\partial}{\partial v_1} M_y \left[ (v_1 + v_2) e_k - \omega F_n \right] = \frac{\partial}{\partial (v_1 + v_2)} M_y \left[ (v_1 + v_2) e_k - \omega F_n \right]
\]

\[
= \frac{\partial}{\partial v_2} M_y \left[ (v_1 + v_2) e_k - \omega F_n \right]
\]  

(271)

and thus \( M_{01}^{(n_\ell)} = M_{10}^{(n_\ell)} \).

Letting

\[
\beta_{\ell} = \beta e_\ell = 1, 2, \ldots, 2N
\]  

(272)
allows $M_{00}(n_\ell)$ and $M_{00}(n_\ell, n_p)$ to be written as:

$$M_{00}(n_\ell) = M_y(-\omega n_\ell p_\ell)$$

$$M_{00}(n_\ell, n_p) = M_y[-\omega(n_\ell p_\ell + n_p p_\ell)]$$

(273)

Substituting above into (270) results in:

$$\sqrt{2} \varepsilon_{qk} = 2 \sum_{\ell=1}^{2N} \left( \frac{\alpha_{k\ell}}{\omega} \right)^2 \sum_{n_\ell} f_2(n_\ell) M_y(-\omega n_\ell p_\ell)$$

$$- \sum_{\ell=1}^{2N} \sum_{p=1}^{2N} \left( \frac{\alpha_{k\ell}}{\omega} \right) \left( \frac{\alpha_{kp}}{\omega} \right) \sum_{n_\ell} \sum_{n_p} f_1(n_\ell) f_1(n_p) M_y(-\omega n_\ell p_\ell - \omega n_p p_\ell)$$

(275)

The first term of (275) reduces to:

$$2 \sum_{\ell=1}^{2N} \left( \frac{\alpha_{k\ell}}{\omega} \right)^2 \sum_{n_\ell} f_2(n_\ell) M_y(-\omega n_\ell p_\ell)$$

$$= 2 \sum_{\ell=1}^{2N} \left( \frac{\alpha_{k\ell}}{\omega} \right)^2 \left( \frac{\pi^2}{6} + \sum_{n_\ell} \frac{n_\ell}{n_\ell} M_y(-\omega n_\ell p_\ell) \right)$$

$$n_\ell \neq 0$$

$$= \frac{\pi^2}{3\omega} (\mathbf{F} e_k)^T (\mathbf{F} e_k) + 2 \sum_{\ell=1}^{2N} \left( \frac{\alpha_{k\ell}}{\omega} \right)^2 \sum_{n_\ell} \frac{n_\ell}{n_\ell} M_y(-\omega n_\ell p_\ell)$$

$$n_\ell \neq 0$$
\[
\epsilon^2_{qk} = \frac{q^2}{12} + 2 \sum_{\ell=1}^{2N} \left( \frac{\alpha_{k\ell}}{\omega} \right)^2 \sum_{\begin{subarray}{c} n_{\ell} \\ n_{\ell} \neq 0 \end{subarray}} \frac{(-1)^n}{n_{\ell}} M_y (-\omega n_{\ell} \beta_{\ell}) (276)
\]

where in the last step the identity (8) was used. Substituting the above result into (275) and replacing \( n_{\ell} \) and \( n_p \) with \( n \) and \( m \) yields the desired result:

\[
\epsilon^2_{qk} = \frac{q^2}{12} + 2 \sum_{\ell=1}^{2N} \left( \frac{\alpha_{k\ell}}{\omega} \right)^2 \sum_{n=\infty}^{\infty} \frac{(-1)^n}{n_{\ell}} M_y (-\omega n_{\ell} \beta_{\ell}) (23)
\]
APPENDIX B

BOUNDS ON \( H_k \)

The derivation shown below represents a typical approach to obtaining the bounds on \( H_k \), which is used in many articles dealing with the subject of quantization noise in the FFT. The results of this derivation are stated in equation (56), (57), and (58) of section 4.3.

The derivations are based on equations (50a) and (50b) of section 4.2. The equations are rewritten below.

\[
\begin{align*}
\zeta_1 &= \eta_1 + \omega \eta_2 \\
\zeta_2 &= \eta_1 - \omega \eta_2
\end{align*}
\]

\( \text{dec. in time} \) (50a)

\[
\begin{align*}
\xi_1 &= \eta_1 + \eta_2 \\
\xi_2 &= \omega \eta_1 - \omega \eta_2
\end{align*}
\]

\( \text{dec. in frequency} \) (50b)

where: \( \zeta_1, \zeta_2, \ldots \) are elements of \( z_k \), \( \eta_1, \eta_2, \ldots \) are elements of \( z_{k-1} \), and \( \omega \) is an element of \( W_k \). The index of \( \zeta_1 \) and \( \eta_1 \) is used only to distinguish between the elements, and has no relation to the position of those elements in \( z_k \) or \( z_{k-1} \).
In calculation of bounds on $H_k$, the values of $\eta_1$ and $\eta_2$ are assumed arbitrary, confined only by the requirement that the magnitudes of their real and imaginary parts be less than $H_{k-1}$. By systematically varying $\eta_1$, $\eta_2$, and $\omega$ the extrema of $H_k$ can be found. The subsequent derivations are carried out separately for the decimation in time and the decimation in frequency.

**Decimation in time**

\[
\zeta_1 = \eta_1 + \omega \eta_2 \\
\zeta_2 = \eta_2 - \omega \eta_2
\]

The regions of the allowable values of $\eta_1$ and $\eta_2$ (set by $H_{k-1}$) are not changed by any number of 90° rotations of these regions in the complex plane, and therefore it can be assumed without the loss of generality, that the largest or the smallest value of $H_k$ is associated with the real part of $\zeta_1$. To obtain the largest $\text{Re}(\zeta_1)$, notice that $\text{Re}(\eta_1)$ and $\text{Re}(\omega \eta_2)$ can be maximized independently. Clearly, the maximum of $\text{Re}(\eta_1)$ is $H_{k-1}$. The maximum of $\text{Re}(\omega \eta_2)$ occurs if $|\text{Re}(\eta_2)| = |\text{Im}(\eta_2)|$ and $\omega$ is chosen to make the polar angle of $\omega \eta_2$ equal to zero. Then:

\[
H_k \leq (1 + \sqrt{2}) H_{k-1}
\]

(277)

The above inequality gives a better bound than inequality (55), in which $H_k$ is bounded by $2\sqrt{2} H_{k-1}$. 
Since a one stage butterfly operation provides a better bound than inequality (55), it is possible that similar arguments for two consecutive stages of the FFT will also result in a better bound than inequality (55). To investigate this, consider the product $P_k P_{k-1}$. An equation analogous to equation on page 55 is:

$$
\begin{align*}
\frac{z^{(1)}}{z_{k}} &= \begin{bmatrix} I & W_k \\ I & -W_k \end{bmatrix} \begin{bmatrix} B_{k-1} & 0 \\ 0 & B_{k-1} \end{bmatrix} \begin{bmatrix} z^{(1)}_{k-2} \\ z_{k-2} \end{bmatrix} \\
&= \begin{bmatrix} B_{k-1} & \frac{W_k B_{k-1}}{B_{k-1}} \\ B_{k-1} & -\frac{W_k B_{k-1}}{B_{k-1}} \end{bmatrix} \begin{bmatrix} z^{(1)}_{k-2} \\ z_{k-2} \end{bmatrix}
\end{align*}
$$

(278)

Notice that:

$$
W_k = \begin{bmatrix} V & 0 \\ 0 & -jV \end{bmatrix}
$$

(279)

where $V$ is diagonal with:

$$
\{v\}_{\ell \ell} = e^{-j(2\pi/2^k)\ell}
$$

and
\[
\begin{bmatrix}
I & U \\
I & -U
\end{bmatrix}
\]

where \(U\) is diagonal with
\[
 \{u\}_{\ell \ell} = e^{-j(2\pi/2^k)2\ell}
\]

The product \(W_k B_{k-1}\) is:
\[
\begin{bmatrix}
V & 0 \\
0 & jV
\end{bmatrix}
\begin{bmatrix}
I & U \\
I & -U
\end{bmatrix}
\]
\[
= \begin{bmatrix}
V \\
-jV
\end{bmatrix}
\begin{bmatrix}
Z \\
-jZ
\end{bmatrix}
\]

where \(Z = V U\) is diagonal with:
\[
\{z\}_{\ell \ell} = e^{-j(2\pi/2^k)3\ell}
\]

Substituting (279), (280), and (281) into (278) yields:
\[
\begin{bmatrix}
I & U & V & Z \\
I & -U & -jV & jZ \\
I & U & -V & -Z \\
I & -U & jV & -jZ
\end{bmatrix}
\]

\[
\begin{bmatrix}
I & U & V & Z \\
I & -U & -jV & jZ \\
I & U & -V & -Z \\
I & -U & jV & -jZ
\end{bmatrix}
\]

\[
\begin{bmatrix}
I & U & V & Z \\
I & -U & -jV & jZ \\
I & U & -V & -Z \\
I & -U & jV & -jZ
\end{bmatrix}
\]

\[
\begin{bmatrix}
I & U & V & Z \\
I & -U & -jV & jZ \\
I & U & -V & -Z \\
I & -U & jV & -jZ
\end{bmatrix}
\]

\[
\begin{bmatrix}
I & U & V & Z \\
I & -U & -jV & jZ \\
I & U & -V & -Z \\
I & -U & jV & -jZ
\end{bmatrix}
\]

\[
\begin{bmatrix}
I & U & V & Z \\
I & -U & -jV & jZ \\
I & U & -V & -Z \\
I & -U & jV & -jZ
\end{bmatrix}
\]

\[
\begin{bmatrix}
I & U & V & Z \\
I & -U & -jV & jZ \\
I & U & -V & -Z \\
I & -U & jV & -jZ
\end{bmatrix}
\]
The two stage operation can be expressed by (using notation defined for eq. (50)):

\begin{align*}
\xi_1 &= \eta_1 + \omega \eta_2 + \omega^2 \eta_3 + \omega^3 \eta_4 \\
\xi_2 &= \eta_1 - \omega \eta_2 - j\omega^2 \eta_3 + j\omega^3 \eta_4 \\
\xi_3 &= \eta_1 + \omega \eta_2 - \omega^2 \eta_3 - \omega^3 \eta_4 \\
\xi_4 &= \eta_1 - \omega \eta_2 + j\omega^2 \eta_3 - j\omega^3 \eta_4
\end{align*}

where \( \eta_1, \eta_2, \eta_3, \eta_4 \), are elements of \( z_{k-2} \).

Because of the symmetry of the allowable regions for \( \eta_1, \eta_2, \eta_3, \) and \( \eta_4 \), and the form of equations (283), the problem of finding maximum \( H_k \) is invariant to 90° rotations. Therefore, it can be assumed that the maximum \( H_k \) is equal to the largest possible value of \( \text{Re}(\xi_1) \). The factors \( \omega, \omega^2, \omega^3 \) rotate the vector representations in the complex plane of \( \eta_1, \eta_2, \eta_3 \), by angles \( \alpha, 2\alpha, 3\alpha \) respectively, where \( \alpha \) is the polar angle of \( \omega \) (the magnitude of \( \omega \) equals one). Maximizing \( \text{Re}(\xi_1) \) requires that all terms have angles between \(-\frac{\pi}{4}\) and \(+\frac{\pi}{4}\). If one of the terms has a different angle, then there exist a corresponding \( \eta \), which is a reflection of the original \( \eta \) around one of the axes of symmetry of the allowable region, which results in larger \( \text{Re}(\xi_1) \). This is demonstrated in Fig. 13, where for simplicity \( H_k \) is assumed equal to one. For each \( \alpha \), by restricting the value of each term to a region between \(-\frac{\pi}{4}\) to \(+\frac{\pi}{4}\), the allowable region for each \( \eta \) is specified. Rotating this region to the final position shows that the value of \( \eta \) giving the greatest contribution
to the $\text{Re}(\zeta_1)$ always satisfies the relation

$$|\text{Re}(\eta)| = |\text{Im}(\eta)|$$

This is demonstrated in Fig. 14.

Using the properties derived above, an expression for $\text{Re}(\zeta_1)$ can be written as a function of $\alpha$:

Let:

$$D = \frac{\text{Re}(\zeta_1)}{H_{k-2}}$$

$$\omega = e^{-j\alpha}$$

for $0 \leq \alpha \leq \frac{\pi}{6}$

$$\eta_2 = \eta_3 = \eta_4 = \sqrt{2} \ e^{j\pi/4}$$

$$D = 1 + \sqrt{2} \left[ \cos\left(\frac{\pi}{4} - \alpha\right) + \cos\left(\frac{\pi}{4} - 2\alpha\right) + \cos\left(\frac{\pi}{4} - 3\alpha\right) \right]$$

for $\frac{\pi}{6} = \alpha = \frac{\pi}{4}$

$$\eta_2 = \eta_3 = \sqrt{2} \ e^{j\pi/4}, \quad \eta_4 = \sqrt{2} \ e^{3j\pi/4}$$

$$D = 1 + \sqrt{2} \left[ \cos\left(\frac{\pi}{4} - \alpha\right) + \cos\left(\frac{\pi}{4} - 2\alpha\right) + \cos\left(\frac{3\pi}{4} - 3\alpha\right) \right]$$

Other cases are symmetric.

It can be easily shown that neither $\alpha = 0$, $\frac{\pi}{6}$, nor $\frac{\pi}{4}$ corresponds to the maximum. By inspection of Fig. 15, the local maxima occur
Fig. 13 Allowable Regions of \( \omega \eta \)

Fig. 14 Largest Re(\( \omega \eta \)) for Given \( \alpha \)
Fig. 15. Terms in $\text{Re}(\zeta_1)/H_{k-2}$

approximately at $\alpha = \frac{\pi}{9}$ and $\alpha = \frac{2\pi}{9}$. Expanding in power series around $\alpha = \frac{\pi}{4}$ gives:

$$\alpha = \frac{\pi}{4} - \delta$$

$$D = 1 + \sqrt{2} \left[ (1 - \frac{1}{2} \delta^2) + \frac{1}{\sqrt{2}} (1 + 2\delta - 2\delta^2) + (1 - \frac{9}{2} \delta^2) \right]$$

$$\frac{3D}{3\delta} = \sqrt{2} - 2(5 + \sqrt{2}) \delta = 0$$

$$\delta = \frac{1}{2 + 5\sqrt{2}} = \frac{1}{9.08}$$

$$D = 5.04$$
Expanding in power series around $\alpha = 0$ gives:

\[ D = 1 + \left( (1 + \alpha - \frac{1}{2} \alpha^2) + (1 - 2\alpha - 2\alpha^2) + (1 + 3\alpha - \frac{9}{2} \alpha^2) \right) \]

\[ = 4 + 6\alpha - 7\alpha^2 \]

\[ \frac{\partial D}{\partial \alpha} = 6 - 14\alpha = 0 \]

\[ \alpha = \frac{3}{7} \]

\[ D = 5.29 \]  \hspace{1cm} (285)

Thus the maximum occurs at $\alpha = \frac{3}{7}$, and

\[ H_k = 5.3 H_{k-2} \]  \hspace{1cm} (286)

Above expression is a closer bound than inequality (55), which predicts that $H_k \leq 4\sqrt{2} H_{k-2} = 5.66 H_{k-2}$. However, the improvement is not as significant as in (277), and this trend will continue for more than two shifts. The direct calculation of the maximum $H_k$ becomes overly complicated for more than two stages, and is not justified in view of the fact that inequality (55) provides a very good approximation. The results of (55), (277) and (286) are listed below:
<table>
<thead>
<tr>
<th>θ of Stages</th>
<th>Maximum $H_k$</th>
<th>Decimal Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$1 + \sqrt{2}$</td>
<td>2.41</td>
</tr>
<tr>
<td>2</td>
<td>$-\quad$</td>
<td>5.3</td>
</tr>
<tr>
<td>3</td>
<td>$8\sqrt{2}$</td>
<td>11.32</td>
</tr>
<tr>
<td>4</td>
<td>$16\sqrt{2}$</td>
<td>22.63</td>
</tr>
<tr>
<td>5</td>
<td>$32\sqrt{2}$</td>
<td>45.25</td>
</tr>
</tbody>
</table>

To compute the minimum $H_k$ for one stage, first assume that

$$H_{k-1} = \max(|\eta_{2r}|, |\eta_{21}|)$$

i.e., $H_{k-1}$ is associated with $\eta_2$. Then for each value of $\eta_2$, $\text{Re}(\zeta_1)$ is minimized if $\eta_1 = 0$. Next, observe that if $\eta_2 = H_{k-1}$ and $\omega = e^{j\pi/4}$, then $H_k = H_{k-1}/\sqrt{2}$. Since this agrees with inequality (55) no closer lower bound on $H_k$ can be found using this technique. Therefore, the lower bound is given by inequality (55).

**Decimation in Frequency**

$$\zeta_1 = \eta_1 + \eta_2$$

(50b)

$$\zeta_2 = \omega\eta_1 + \omega\eta_2$$

The search for maximum $H_k$ for the decimation in frequency is much simpler. Clearly the maximum $H_k$ is associated with $\zeta_2$. 
Let \( \eta_1 = -\eta_2 = \sqrt{2} e^{j\pi/4} H_{k-1} \) and let \( \omega = e^{-j\pi/4} \), then \( H_k = 2\sqrt{2} \).

Since this is also the maximum value specified by (55), no better bound can be found using above technique.

For two stages, using a method analogous to the steps taken in deriving (278) through (282):

\[
\begin{bmatrix}
B_{k-1} & 0 & I & I \\
0 & B_{k-1} & W_{M-k} & -W_{M-k}
\end{bmatrix}
\begin{bmatrix}
Z_k^{(j)}
\end{bmatrix}
= \begin{bmatrix}
B_{k-1} & B_{k-1} \\
B_{k-1} W_{M-k} & -B_{k-1} W_{M-k}
\end{bmatrix}
\begin{bmatrix}
Z_{k-2}^{(j)}
\end{bmatrix}
\]

(287)

\[
\begin{bmatrix}
I & I & V & 0 \\
U & -U & 0 & -jV
\end{bmatrix}
\begin{bmatrix}
B_{k-1} W_{M-k}
\end{bmatrix}
= \begin{bmatrix}
V & -jZ \\
Z & jZ
\end{bmatrix}
\]

(288)
where \( U, V, Z \) are defined with equations (279), (280), (281).

Finally:

\[
\begin{bmatrix}
I & I & I & I \\
U & -U & U & -U \\
V & -jV & -V & jV \\
Z & jZ & -Z & -jZ
\end{bmatrix}
\begin{array}{c}
\frac{\zeta(j)}{z_{k-2}}
\end{array}
\tag{289}
\]

Using notation defined for eq. (57), the two stage operation can be expressed by:

\[
\begin{align*}
\zeta_1 &= \eta_1 + \eta_2 + \eta_3 + \eta_4 \\
\zeta_2 &= \omega (\eta_1 - \eta_2 + \eta_3 - \eta_4) \\
\zeta_3 &= \omega^2 (\eta_1 - j\eta_2 - \eta_3 + j\eta_4) \\
\zeta_4 &= \omega^3 (\eta_1 + j\eta_2 - \eta_3 - j\eta_4)
\end{align*}
\tag{290}
\]

For \( \eta_1 = \eta_3 = -\eta_2 = -\eta_4 = \sqrt{2} e^{j\pi/4} \) and \( \omega = e^{-j\pi/4} \), \( \text{Re}(\zeta_2) = 4\sqrt{2} \).

Since this is equal to the bound set by inequality (55), no better bound can be found in this approach.

To find the minimum of \( H_k \) for one stage, first observe that choosing \( \eta_1 = H_{k-1} \) and \( \eta_2 = 0 \) gives smaller \( H_k \) than choosing \( \eta_1 = \eta_2 \) or \( \eta_1 = -\eta_2 \). Thus in minimizing \( H_k \), the difference \( |\eta_1 - \eta_2| \) is not zero, and therefore \( \omega \) can be chosen to make \( |\text{Re}[\omega(\eta_1-\eta_2)]| \) equal to \( |\text{Im}[\omega(\eta_1-\eta_2)]| \). It can be assumed without loss of generality that \( H_k \) corresponds to the real part of either \( \zeta_1 \) or \( \zeta_2 \) and \( H_{k-1} \) is
equal to Re(\(\eta_1\)). Let \(-a\) be the real part of \(\eta_2\). Then the smallest possible values of Re(\(\zeta_1\)) and Re(\(\zeta_2\)) (which are greater than Im(\(\zeta_1\)), Im(\(\zeta_2\))), are:

\[
\text{Re}(\zeta_1) = H_{k-1} - a
\]

\[
\text{Re}(\zeta_2) = \frac{1}{\sqrt{2}} (H_{k-1} + a)
\]

(291)

Minimum \(H_k\) results if \(\text{Re}(\zeta_1) = \text{Re}(\zeta_2)\).

Therefore:

\[
H_k \geq \frac{2}{\sqrt{2} + 1} H_{k-1}
\]

(292)

The imaginary parts of \(\eta_1\) and \(\eta_2\) can only increase \(H_k\), so they can be assumed zero.

**Summarizing**

The bounds on \(H_k\) are accurately expressed by the inequality (55).

\[
\frac{H_{\ell}}{\sqrt{2}} \leq H_k \leq \frac{H_{k-\ell} + 1}{2} \frac{1}{H_{\ell}}
\]

For the decimation in time the expressions (277) and (286) give closer higher bounds for one and two consecutive stages:
\[ H_k \leq (1 + \sqrt{2}) H_{k-1} \]

\[ H_k \leq 5.3 H_{k-2} \]

For the decimation in frequency a better lower bound for one stage is given by (292).

\[ H_k \geq \frac{2}{1 + \sqrt{2}} H_{k-1} \]
REFERENCES


