Performance Analysis of a Multistage Multicarrier Demultiplexer/Demodulator (M-MCDD) in the Presence of Interference and Quantization Error

David Salim Salhany

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Applied Science at
Concordia University
Montréal, Québec, Canada

May 2000

0-612-47831-9

Abstract

"Performance Analysis of a Multistage Multicarrier Demultiplexer/Demodulator (M-MCDD) in the Presence of Interference and Quantization Error"

David Salim Salhany

One of several proposed multicarrier demultiplexer/demodulator (MCDD) structures is studied for application to satellite on-board regenerative repeaters. The structure considered here is the multistage multicarrier demultiplexer/demodulator (M-MCDD). The M-MCDD has been chosen as the preferred structure for demultiplexing signals composed of $N=2^l$ channels due to its high level of modularity, low computational complexity and low power consumption. Originally proposed by Göckler in 1992, the M-MCDD is considered one of the best available MCDDs to demultiplex composite MF-TDMA signals in the presence of additive white Gaussian noise (AWGN) and adjacent channel interference (ACI). Adjacent channel interference (ACI) is the result of imperfect filtering, which can be induced by any filtering method, and insufficient channel separation within the aggregate input signal. An AWGN analysis is carried out considering the presence of ACI within the M-MCDD. Once the effect of ACI on the performance of the M-MCDD is determined, the AWGN analysis will be extended by assuming that the halfband filter coefficients are quantized in value as opposed to being real-valued. This will also introduce quantization noise into the demultiplexed output. Due to the complexity induced by ACI and the combination of ACI with quantization noise, we rely on analytical techniques to arrive at approximate numerical solutions, with calculable error margins, for the performance of the M-MCDD. One of the important applications of the numerical solutions will be to determine the filter design for the M-MCDD.

Acknowledgements

I would like to thank Dr. Tho Le Ngoc of Concordia University, Dr. Norman Secord and Dr. Chun Loo, both of the Communication Research Council, for their intuition, insight and expertise donated to this research project. The project would not be the success that it is without their hard work and dedicated support.

Dedication

I dedicate my years of hard work and accomplishments at Concordia University to my mother, Nelda, my brother, Kevin and my girlfriend, Monica. Thank you for your patience these last two years.

I love you all.

Table of Contents

Page

Page

List of Figures

Figure Page

List of Tables

List of Key Symbols Defined

The M-MCDD Structure and Signals

$M_{a,b}$	-The module of stage "a" in position "b" from top to bottom
l	-The number of stages in the M-MCDD
$N=2^l$	-The total number of channels to be demultiplexed within the M-MCDD
A(f)	-Frequency response of the analog anti-aliasing filter (AAF)
B	-Bandwidth of each individual channel
NB	-Total one-sided bandwisth filtered out by A(f)
$D(j\omega)$	-Frequency response of the digital anti-aliasing filter (DAF)
$h(nT_s)$	-The impulse response of the FIR halfband filter having $2L+1$ taps
$H(j\omega)$	-Corresponding frequency response of the FIR halfband filter
$Ltap$	-The number of non-zero filter taps per halfband filter
$2L+1$	-Total number of filter taps per hlafband filter. Filter taps range from $-L$ to L
$g(t)$	-The impulse response of the channel detection filter
$Itap$	-The number of filter taps in the channel detection filter
$h_s\,(t)$	-The transmit filter
f_c	-Frequency spacing between the carriers
$T_c = 1/f_c$	-Corresponding sampling period
$f_s = 2N\cdot f_c$	-Initial sampling frequency for an N channel composite signal
$T_s = 1/f_s$	-Corresponding sampling period
T_b	-The symbol duration
k	-Position of the desired channel from the output of the M-MCDD
α	-Position of the desired symbol on the output channel of the M-MCDD
$f_k = (k+\frac{1}{2})\cdot f_c$	-Center frequency of the k^{th} channel
$\omega = 2\pi f_s$	-Normalized frequency index
$x(t)$	-Analog MF-TDMA input signal
$x(nT_s)$	-The sampled input MF-TDMA signal
X(f)	-Analog spectrum of $x(t)$
$s_k(t)$	-k^{th} FDMA signal

A_k	-The carrier amplitude of the k^{th} channel
$a_{k,i}$	-The data phase in the i^{th} symbol interval on the k^{th} channel
$a_{k,\alpha}$	-Data phase of the desired symbol
$a_{q,j}$ and b_{qj}	-The in-phase and quadrature data phase in the j^{th} symbol interval on the q^{th} channel
γ_k	-The random timing offset on the k^{th} channel that ranges between $-T/2$ to $T/2$.
ϕ_k	-The random phase offset on the k^{th} channel that ranges between 0 and 2π.
$f(\gamma_k)$ and $f(\phi_k)$	-The corresponding probability density functions
$z(t)$	-Additive White Gaussian Noise (AWGN) process.
$f_z(z)$	-The probability density function of $z(t)$ (Gaussian distributed)
σ^2	-Variance of $z(t)$
$z_k(\cdot)$	-The output WGN process from the k^{th} channel
$z_k^I(\cdot)$	-The in-phase component of $z_k(\cdot)$
σ_o^2	-Variance of $z_k(lT_b)$
$x'_{1,1}(nT_S)$	-Sampled signal at the output of the A/D converter
n_j	-Summing index from stage j
$x_{a,b}(n(2^{(a-1)}T_S))$	-The input signal of stage "a", module "b"
$X_{a,b}(2^{(a-1)}T_S\omega))$	-Spectrum of the sampled input signal to stage "a", module "b"
$x_{a,b,H}(n(2^a T_S))$	-The output highpass part of input signal to stage "a", module "b"
$x_{a,b,L}(n(2^a T_S))$	-The output lowpass part of input signal to stage "a", module "b"
$y_k(nT_c)$	-The demultiplexed k^{th} channel output signal
$Y_k(T_c\omega)$	-Spectrum of the k^{th} channel output signalcorresponding to $Y_k(f)$
$r_k(lT_b)$	-The final output signal of the k^{th} channel from the channel detection filter
$I_2 - I_1 + 1$	-Number of summation terms used to obtain $r_k(lT_b)$
β_l and μ_l	-The basepoint index and the fractional interval, respectively

Interference Analysis and The Gauss Quadrature Rule Technique

$u(0)$	-Actual desired symbol to be demodulated
$u(\cdot)$	-Intersymbol interference (ISI) on the k^{th} channel
$v(\cdot)$	-Interchannel interference (ICI) on all other interfering channels

$v^I(\cdot)$	-In-phase component of ICI
$v^Q(\cdot)$	-Quadrature component of ICI
η	-Interference variable
$f_\eta(\eta)$	-The probability density function of the interference
$m_i(\eta)$	-i^{th} moment of the interference
ξ or ξ_η	-Fixed value in the range of the interference
$\xi_{\eta,MAX}$	-The maximum value of the interference
$\xi_{\eta,MIN}$	-The minimum value of the interference
η_i	-An interference term of η
η'	-The truncated interference variable
η''	-The collection of interfering symbols not considered by η'
m_l and m_u	-The number of significant interfering symbols below and above the α^{th} desired symbol on the desired channel, respectively.
$m_{l,j}$ and $m_{u,j}$	-The number of significant interfering symbols below and above the α^{th} desired symbol on the j^{th} channel, respectively.
K	-The total number of interference samples considered in η'
η_1	-The ISI portion of η'
η_2	-The ICI portion of η'
$\eta_{1,i}$	-Individual terms of the ISI portion of η'
$\eta_{2,i}$	-Individual terms of the ICI portion of η'
$E[\eta^\rho_{1,i}]$	-ρ^{th} moment of the i^{th} term of the intersymbol interference (ISI)
$E[\eta^\rho_{2,i}]$	-ρ^{th} moment of the i^{th} term of the interchannel interference (ICI)
w_i and x_i	-The weights and abscissas of the Gauss Quadrature Rule
M	-Number of terms used in the GQR
R_M	-The truncation error due to using $2M+1$ moments to evaluate the probability of bit error

The Beaulieu Fourier Series Expansion Technique

c_m	-The coefficients in the Fourier Series representation of $Q(x)$

$\varepsilon(x)$	-The error in using the Beaulieu FSE to approximate $Q(x)$
$\Phi_\eta(\omega)$	-The characteristic function of the interference variable, η
$\Phi^{*}_{\eta}(-m\omega)$	-The truncated characteristic function
$\Phi^{**}_{\eta}(-m\omega)$	-The characteristic function comprising all the interfering symbols not included in the truncated characteristic function
$P(\cdot)$	-A square wave of period T used to gate $f_z\ (z)$
β	-The error associated with using the Fourier Series approximation for $Q(x)$
R_M	-The error associated with using M terms in the summation for the Fourier Series
Δ	-The error due to using K interference terms in $\Phi_\eta(\omega)$
$\mathrm{M_{BEAU}}$	-Number of terms taken in the Fourier Series Expansion
T	-The period of the square wave gating function

Quantization Analysis and The Gram-Charlier Series Expansion Technique [♦]

$\hat{h}_j(n_j 2^j T_s)$	-Quantized halfband filter coeffieicnts from stage j
$\hat{H}(j\omega)$	-The discrete Fourier Transform (DFT) for $\hat{h}_j(n_j 2^j T_s)$
$\tilde{h}_j(n_j 2^j T_s)$	-The sequence of quantization error random variables from each stage
$\tilde{H}(j\omega)$	-The discrete Fourier Transform (DFT) for $\tilde{h}_j(n_j 2^j T_s)$
Δ	-The quantization step size
B	-Number of quantization bits used to represent the halfband filter coefficients
$\mathrm{h_{total}}$	-Total number of halfband filter coefficients that are quantized
Ψ	-Total number of quantization bits in the M-MCDD
$\hat{y}_k(mT_c)$	-The demultiplexed $\mathrm{k^{th}}$ channel output signal considering quantized halfband filter coefficients
$\tilde{y}_k(mT_c)$	-The demultiplexed $\mathrm{k^{th}}$ channel output signal resulting from quantization error
$y'_k(mT_c)$	-The demultiplexed $\mathrm{k^{th}}$ channel output signal corresponding to the cross-product

[♦] Please note that when the context of discussion involves quantization, we use q to represent the quantization variable. Thus, so as not to confuse our choice of variables for the interference, the channel representation is given as Q when not discussing the k^{th} channel. Also, quadrature components are shown as $\boldsymbol{Q}$.

between quantization error and interference

$\hat{r}_k(lT_b)$ -Final output signal from the channel detection filter assuming quantized halfband halfband filter coefficients

$\tilde{r}_k(lT_b)$ -Final output due to quantization error

$r'_k(lT_b)$ -Final output corresponding to cross product between interference and quantization noise

$\tilde{u}'(\cdot)$ -Combination of ISI with quantization noise

$\tilde{v}'(\cdot)$ -Combination of ICI with quantization noise

$\tilde{u}'_{\max}(\cdot)$ -Maximum value of the ISI/quantization noise with respect to $\tilde{h}_j(n_j 2^j T_s)$

$\tilde{u}'_{\min}(\cdot)$ -Minimum value of the ISI/quantization noise with respect to $\tilde{h}_j(n_j 2^j T_s)$

$\tilde{v}'_{\max}(\cdot)$ -Maximum value of the ICI/quantization noise with respect to $\tilde{h}_j(n_j 2^j T_s)$

$\tilde{v}'_{\min}(\cdot)$ -Minimum value of the ICI/quantization noise with respect to $\tilde{h}_j(n_j 2^j T_s)$

$\tilde{z}_k'^{\,I}(\cdot)$ -Combination of Gaussian noise from the kth channel (in-phase component) with quantization noise

η_q -Quantization noise variable

$f(\eta, \eta_q)$ -Joint probability density function of the quantization noise and interference

$f_{\eta q}(\eta_q)$ -Marginal probability density function of the quantization noise

$m_i(\eta_q)$ -i^{th} moment of the quantization noise

$\xi_{\eta q}$ -Fixed value in the range of the quantization error

$\xi_{\eta_q, MAX}$ -Maximum value of the quantization noise

$\xi_{\eta_q, MIN}$ -Minimum value of the quantization noise

R -Correlation between interference and quantization noise

a_i -Recursive Gram-Charlier coefficients for representing $f_\eta(\eta)$

b_i -Recursive Gram-Charlier coefficients for representing $f_{\eta q}(\eta_q)$

η_q^* -Truncated quantization noise variable

$\eta_{q.I}$ -The ISI component of η_q^*

$\eta_{q.I.i}$ -Individual terms of the ISI component of η_q^*

$E[\eta^{\rho}_{q.1,i}]$ -ρ^{th} moment of the i^{th} term of the ISI component of η_q^*

$\eta_{q,2}$ -The ICI component of η_q^*

$\eta_{q,2,Q,i}$ -Individual terms of the ICI component of η_q^* , i^{th} term on the Q^{th} channel ($Q \neq k$).

$E[\eta^{\rho}_{q,2,Q,i}]$ -ρ^{th} moment of the i^{th} term on the Q^{th} channel of the ICI component of η_q^*

$uu(\cdot), vv(\cdot), zz(\cdot)$ -Quantities defined to simplify the expressions for the moments of the quantization noise

$\hbar_j^{lp}(n_j \cdot 2^j T_s)$ -Summation of halfband filter coefficient-random variable combinations for the lowpass output channels

$\hbar_j^{hp}(n_j \cdot 2^j T_s)$ -Summation of halfband filter coefficient-random variable combinations for the highpass output channels

M -Number of points taken in the Taylor Series Expansion for $Q(\cdot)$

R_M -Truncation error due to using only $M+1$ terms in the Taylor Series

R'_{2K_i} -Truncation error from upper bounding each Gram-Charlier Series to $2K_i$ terms

$2K_1$ -Number of moments used in the interference, less one

$2K_2$ -Number of moments used in the quantization error, less one

$2K_3$ -Number of terms used in the orthogonal polynomial expansion of $f(\eta, \eta_q)$, less one

Performance Evaluation In The Presence of Interference and/or Quantization Noise

P_b -Probability of bit error

$P_{b,\eta}$ -The probability of bit error due to interference

$\Delta_{BER,\eta q}$ -Expression for BER degradation factor due to quantization noise

Δ -Amount of degradation in performance (relative to ideal QPSK demodulation) due to interference

Δ_q -Amount of degradation in performance (relative to infinite quantization) due to quantization

Δ_t -Amount of total degradation in performance (relative to ideal QPSK demodulation) due to interference and quantization

M_{op} -Computational complexity of the M-MCDD

Chapter One

Introduction

Broadband satellite communications systems are currently required to handle large amounts of multimedia traffic, therefore greater power efficiency and bandwidth utilization are demanded [1-3]. This means that satellites will be used as regenerative repeaters with multi-beam antennas and baseband switching resulting in low cost cellular and personal communication service (PCS) systems [1-3]. This function is possible by considering the on-board processing (OBP) satellite communications system, shown in Figure 1-1.

The communication process shown in Figure 1-1 can be summarized as follows. Cellular and PCS subscribers communicate through different channels to a single earth station within each spotbeam. This earth station uplinks the MF-TDMA signal to the satellite. Individual carriers are then regenerated from the MF-TDMA signal on board the satellite. In the process of Figure 1-1, we will focus on the process occurring on the uplink, that is, involving the MF-TDMA signal [1] [4-5].

The regeneration of a carrier from an MF-TDMA signal is very similar to a spectral analysis in which a wideband signal is filtered, sampled and divided into uniform subbands. At the center of each of these subbands will be one of the desired signals. In the context of OBP, this "spectral analyzer" is most often referred to as a *Multi-Carrier Demultiplexer/ Demodulator* (MCDD) [2]. There have been several efficient structures proposed in the past to perform the demultiplexing process [1] [4] [6]. One of these structures is the *Multistage Multicarrier Demultiplexer/Demodulator* (M-MCDD) [6-8]. The M-MCDD has several advantages over previously designed structures, the most notably being its *modularity* and *compactness*. Design complexity is also reduced in comparison with the *Single Stage Method* (SSM) which uses more modules for the demultiplexing process. These advantages result in the M-MCDD being a relatively simple and cost effective solution for a demultiplexer [1] [6-7].

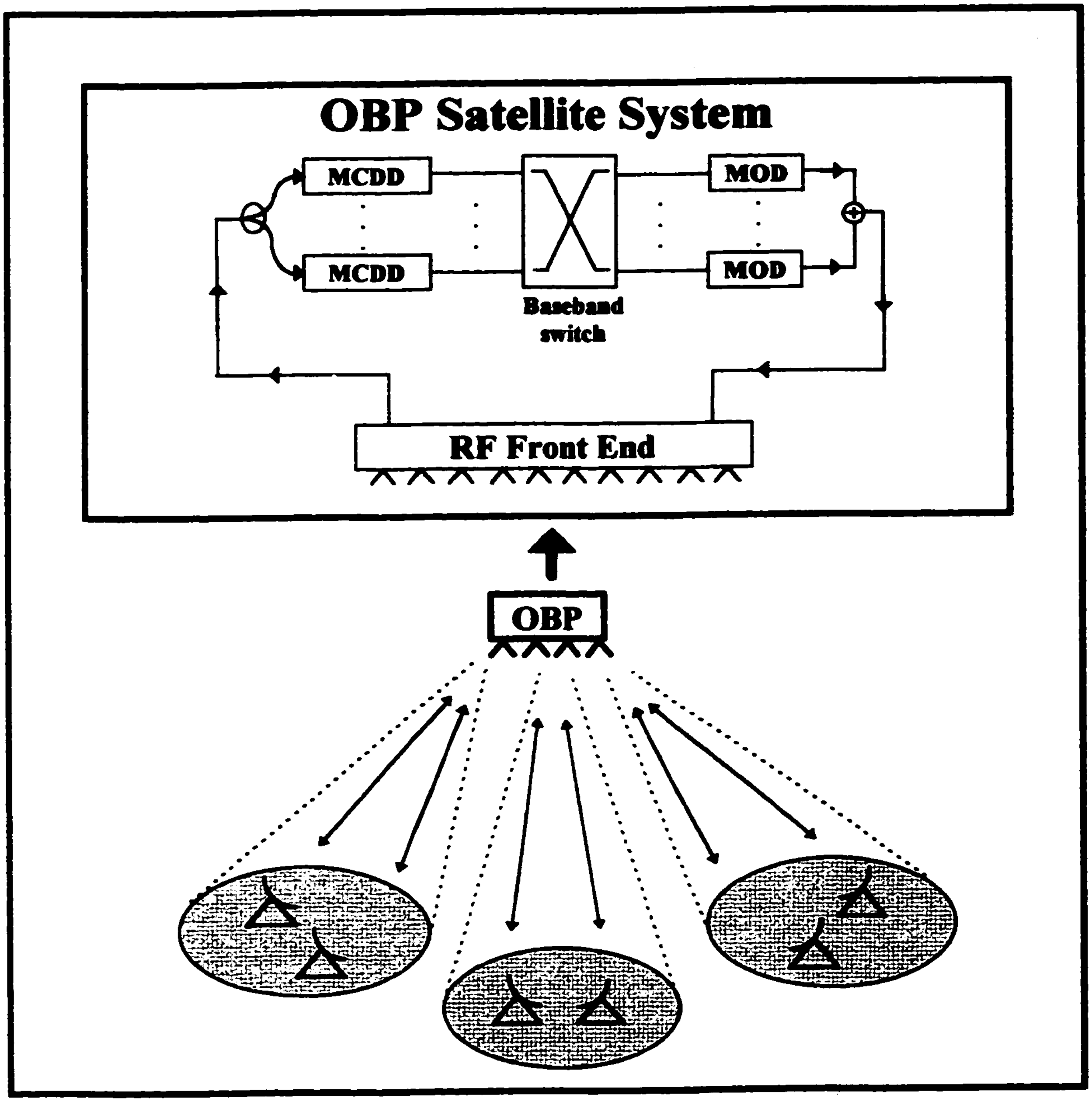


Figure 1-1. Overview of the On-Board Processing (OBP) Satellite Communication System [1].

The design of the M-MCDD uses digital signal processing (DSP) hardware involving finite impulse response (FIR) filters [6] [9]. Because these digital filters are of finite length, the presence of sidelobes in the stopband region of the filter will cause imperfect filtering of the desired channel [2] [9]. This, in turn, will induce adjacent channel interference (ACI) to appear in the demultiplexed output which may degrade the performance [2] [10-12]. Therefore, to reduce the amount of ACI, it is desirable to increase the filter lengths in order to reduce the size of the sidelobes. However, the result of increasing the filter lengths is an increase in the complexity of the M-MCDD. Therefore, there is a tradeoff in increasing the filter lengths, that is, performance versus complexity of the M-MCDD [9]. To date, there exists actual hardware implementations of the M-MCDD, however, tools which can be useful in choosing design criteria for the M-MCDD, such as filtering requirements, have not been developed [13-14]. Thus, the aim of this research is to evaluate the effect of interference on the performance of the M-MCDD and to develop an analytical model to determine the performance given a fixed number of filter taps. It will then be clear how this performance and complexity evaluation can be used to choose filtering requirements for the M-MCDD.

There are two possibilities for the evaluation of the performance of the M-MCDD and they are simulation and analysis* [2] [13]. Most of the work to date on performance evaluation of MCDD structures has involved simulation. This method, however, is known to be very time consuming, especially for evaluation at low bit error rates [2]. Typically, a simulation program can take up to a full day or even a week depending on the desired symbol error rate (SER) to be evaluated. The development of analytical tools for performance evaluation is a valuable asset since an analytical tool is significantly faster and more accurate than a simulation program [13]. Thus, in this research, it will be shown that an additive white Gaussian noise (AWGN) analysis can be carried out to evaluate the performance of the M-MCDD in the presence of ACI. And, as stated, an analytical model with which one may evaluate the performance of the M-MCDD will be developed. It should be noted that we do not present any simulation results for comparison.

The main reason why analytical techniques have been avoided is the degree of difficulty in arriving at a solution. To elaborate, imperfect filtering and insufficient channel separation cause ACI to appear in the demodulated output in the form of intersymbol interference (ISI) and interchannel interference (ICI) [2] [10-11]. The presence of ACI makes the probability of bit error expression difficult to solve since the probability density function involved in the expression is unknown and virtually impossible to find [2] [12]. Therefore, to evaluate the analytical model with known accuracy, we use numerical approximation techniques [2]. The analysis is later extended beyond the complexity of ACI to include quantization noise resulting from finite length halfband filter coefficients. Consequently, the analytical model is extended to evaluate the effect of quantization noise as well as ACI on the performance of the M-MCDD. To arrive at a numerical solution for the analytical model, the numerical approximation techniques considered are the Gauss

* In this thesis, no numerical comparison with simulation is done. This work is actually a continuation of previous analysis done on the PPM and SSM [2].

Quadrature Rule (GQR) and the Fourier Series Expansion technique of Beaulieu (FSE). Both techniques will be used to evaluate the effect of ACI on the performance of the M-MCDD [10-12] [15-17]. To extend the evaluation to include the effect of quantization noise on the performance, the Gram-Charlier Series expansion is considered [14] [17-22]. These techniques have been chosen to solve the problem because, generally, they are sufficiently fast while maintaining a desired level of precision in the numerical results [10-12] [14-22].

The objectives of the research within this thesis may be outlined as follows:

1. The first objective is to evaluate the performance of the Multistage Multicarrier Demultiplexer/Demodulator (M-MCDD) in the presence of Additive White Gaussian Noise (AWGN) and Adjacent Channel Interference (ACI) to develop an analytical model. This analytical model will in turn help determine the number of filter taps required in the halfband filters and channel detection filter for the design of the M-MCDD.
2. Secondly, to extend the AWGN analysis and analytical model in order to investigate the effect of quantization noise on the BER performance of the M-MCDD. The quantization noise is a result of finite word length halfband filter coefficients [28]. In the end, this analysis will help us choose the number of quantization bits per halfband filter tap.

The body of the thesis is organized into five chapters. In chapter two, an introduction to the concept of group demodulation and the M-MCDD structure as background information is given [1] [4] [6-7]. As well, previously analyzed MCDDs are briefly described and it is shown why the M-MCDD is the preferred group demodulator [1-2] [4] [6] [13]. It is also made clear what are the issues in the research which need to be solved. In chapter three, the analysis of the M-MCDD in the presence of Adjacent Channel Interference (ACI) is carried out [9-13]. From this analysis, performance analysis results will be presented on two fronts. On one hand, we will look at the numerical accuracy of the performance of the M-MCDD. For example, between the GQR technique and Beaulieu's FSE technique, we will vary design parameters and show how the performance varies between both techniques. We will also decide which of the two is the preferred method of numerical approximation based on their CPU timing requirements [15-16]. Also, the choice of several important parameters for the numerical analysis will be made [2]. On the other hand, we look at the numerical results as a method to evaluate the system performance in order to effectively choose the number of filter taps for the M-MCDD [6-7] [13] [26-27]. In particular, we will try to choose values for the halfband filter taps and channel detection filter taps based on a desired performance level, that is, a maximum allowable BER level, and degradation relative to ideal QPSK demodulation. This may result in several possible scenarios to choose from. To impose further limitations on our choice, the computational complexity of the M-MCDD as a function of the number of filter taps will be evaluated [9] [25]. Thus, based on a maximum tolerable level of computational complexity, it is possible to restrict our choice for the filter characteristics needed to meet design criteria. If, in the end, we still have several possibilities to consider, the choice will ultimately come down to which scenario achieves the better design criteria, depending on what criteria is a priority in the design. It will also be clear why a tradeoff exists between performance and complexity [2] [6-7] [9] [13] [25]. Finally, case studies based on all of the above design criteria will be given showing how to choose the desired number of filter taps. In chapter four, the AWGN analysis of the M-MCDD is extended to include the effect of finite halfband filter coefficient quantization thus quantization noise as well as ACI appear in the demodulated output [9] [14] [17-22] [25-26] [28]. The amount of degradation in performance compared with the corresponding infinite quantization scenario is studied. The variation in the amount of degradation as the filter lengths vary is looked at [14] [28]. As well, to further analyze the system performance given from the previous chapter, it will be made clear how to determine the minimum number of quantization bits per filter tap based on maximum tolerable degradation levels relative to infinite quantization. The case studies from the previous chapter will also be extended to take into account these degradation levels due to using a finite number of quantization bits as opposed to an infinite number. From this, it will be shown how to choose the number of quantization bits per halfband filter tap. It can be done based on the maximum tolerable degradation levels and the relative improvement in degradation as we increase the number of quantization bits while considering the potential increase in cost in doing so. Finally, conclusions are made in chapter five.

Chapter Two

Group Demodulation Techniques and Structures

2.1: Background on Multicarrier Demodulation

New On-Board Processing systems on satellites require the processing of N MF-TDMA signals composed of frequency multiplexed channels. We require a signal processing method capable of discriminating the individual channels and distinguishing the desired symbol on the desired channel. We therefore consider a spectral analysis process in which a wideband signal is filtered, sampled and divided into uniform subbands which are then frequency shifted to baseband. This spectral analysis process is known as multicarrier or group demodulation [1-2] [4] [6].

Multicarrier demodulation consists of two distinct processes and they are demultiplexing and demodulation [4]. The focus of the thesis will be on spectral analyzers which mainly serve to demultiplex MF-TDMA signals. These architectures are better known as Multicarrier Demultiplexers/Demodulators (MCDD) [2] [6-7]. The operation of the MCDD is to separate the channels of the input MF-TDMA signal, perform channel detection and to supply each channel to a demodulator for symbol detection using synchronization techniques [1] [4] [23-24]. The typical baseband model for a signal composed of eight channels as input to the MCDD is shown in Figure 2-1. Here, the frequency spacing between the carriers is f_c , the initial sampling frequency is $f_s = 2N \cdot f_c$, to avoid aliasing once the MF-TDMA signal is sampled, and the center frequency of the k^{th} channel is $f_k = (k+\frac{1}{2}) \cdot f_c$ [2] [9].

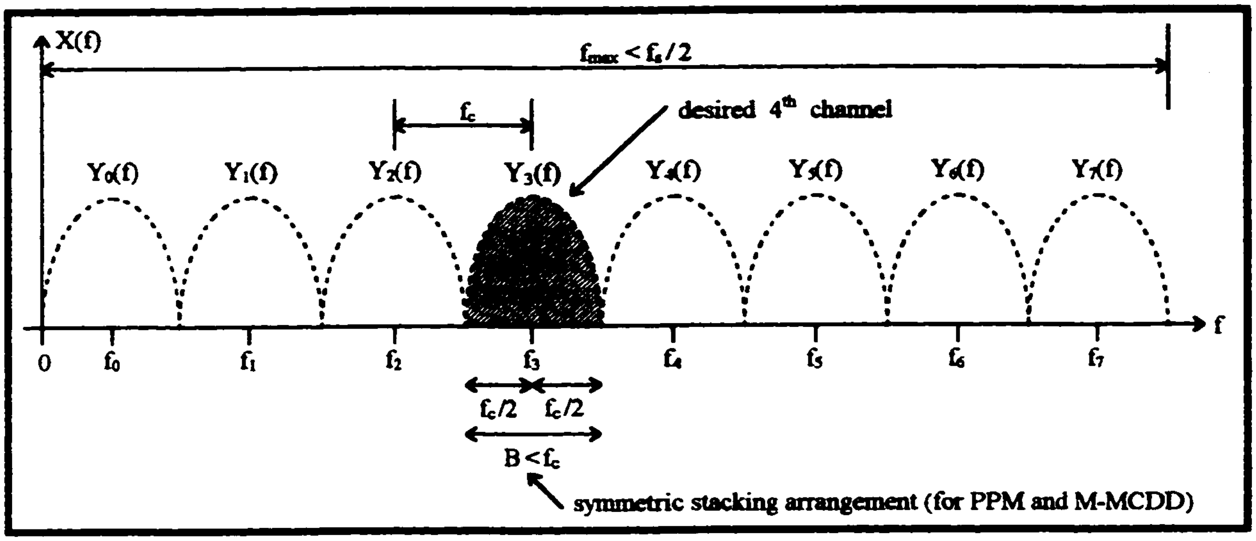


Figure 2-1. Baseband model for *N=8* channels as input to the MCDD [1] [2].

In the demodulation process, the demodulator receives the output QPSK signals from the MCDD [2]. The purpose of the demodulator is to take these QPSK signals and extract the essential information from them, namely the carrier frequency, the clock frequency and the symbol and timing offsets in order to detect the desired symbol on the desired channel. We can find this information through coherent demodulation, which is used in satellite communications in order to achieve the required bit error rate with an acceptable signal to noise ratio (SNR).

There are reasons why we use group demodulation to obtain the individual channels from an MF-TDMA signal [1-2] [4]. On one hand, we may perform frequency demultiplexing to obtain each individual channel, followed by demodulation. To achieve frequency demultiplexing, we directly implement a bank of digital filters followed by a decimator, which is used as a down-sampler [1] [25-26]. The frequency demultiplexer may be realized through the direct method or the Single Stage Method (SSM) where we implement a digital filter for each channel to be demultiplexed [1] [6]. This, however, is not the most convenient solution because depending on the number of channels to be demultiplexed, we can have many receivers which will introduce a high level of complexity. As a cost-effective alternative to the SSM, we can resort to a multicarrier group receiver where demultiplexing and part of the demodulation process are carried out simultaneously thereby reducing complexity and payload [1] [4]. There are two alternatives for the design of MCDDs and they are the Polyphase/FFT Method (PPM) and the Multistage Multicarrier Demultiplexer/Demodulator (M-MCDD) [1] [6-8]. Because the M-MCDD has some important advantages over the SSM and PPM, the focus of the thesis will be on the M-MCDD [6-7] [13]. In the next section, however, the SSM and PPM will be briefly described. Apart from these two possibilities, there are also other techniques which have been described. One of these is the Chirp Fourier Transform (CFT) Technique [2].

2.2 : Background on Previously Analyzed MCDDs

2.2.1: The Single Stage Method (SSM)

The Single Stage Method (SSM) is the most primitive way of implementing a frequency demultiplexer. The way in which the SSM functions is that it demultiplexes a composite MF-TDMA signal on a channel-by-channel basis. This method of demultiplexing has two distinct advantages, namely, a high degree of modularity and flexibility. The SSM will be modular if f_c is the same between each channel. In this case, the demultiplexing requirements for each channel will be the same. Only varying channel bandwidths will cause deviations between the lowpass filtering requirements for each channel. This is because the individual channel filters need to be adjusted according to the filter requirements of each corresponding channel. This, along with the fact that the number of channels permitted is unconditional, shows a high level of design flexibility [1] [6] [8]. The SSM also has drawbacks. One major drawback of the SSM is its extremely high level of complexity. For example, to implement an N channel demultiplexer, you need exactly N modules to obtain each channel giving the SSM a relatively large complexity and payload. The SSM also has a high level of computational complexity in comparison with other structures [1] [6].

The SSM used for demultiplexing an N channel MF-TDMA signal is illustrated in Figure 2-2. Here, $x(nT_s)$ represents the sampled input MF-TDMA signal, $h(nT_s)$ is a lowpass filter, N is the decimation rate corresponding to the number of channels to be demultiplexed and $y_k(nT_c)$ is the demultiplexed k^{th} channel output signal. From $x(nT_s)$, we wish to obtain the output $y_k(nT_c)$ and this can be done in either of two ways. The first method is to frequency shift $x(nT_s)$ to baseband and then extract $y_k(nT_c)$ using the corresponding lowpass filter. The second method is to filter $x(nT_s)$ in the bandpass region and then frequency shift the output signal to baseband [1] [9] [23]. We see from Figure 2-2 the first method is used. In either case, however, we also require a decimation of the signal by a factor of $2N$ in order to reduce the sampling frequency and to meet the Nyquist Criterion [1] [9] [23] [26].

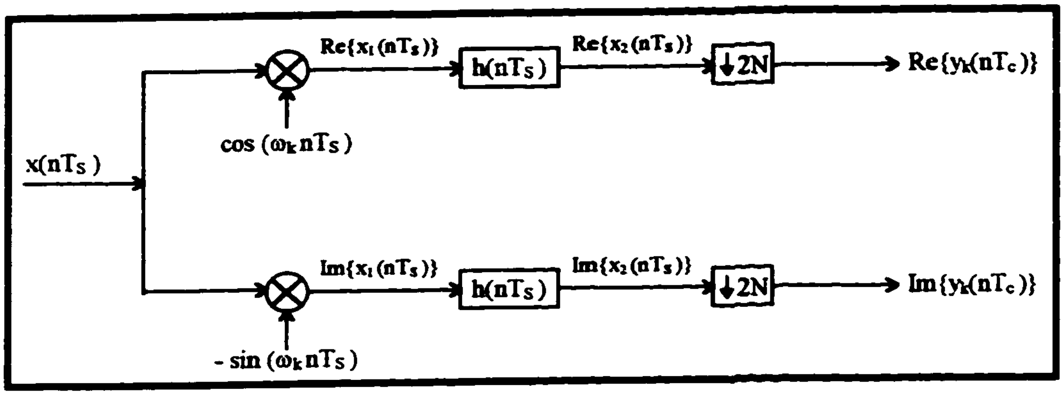


Figure 2-2. Single Stage Method (SSM) for demultiplexing *k*th channel of *N* channel signal [1-2].

The normalized frequency here is $\omega_k = 2\pi f_k$ [1] [9]. To shift $x(nT_s)$ to baseband, homodyne demultiplexing is performed by multiplying $x(nT_s)$ by $e^{-j\omega_k nT_s}$ [2] [9]. Also, to distinguish the real and imaginary outputs in Figure 2-2, the sinusoidal components are separated. The kth channel output signal of the SSM, $y_k(nT_c)$, is expressed as [1]:

$$y_k(nT_c) = \sum_{i=0}^{L} h(iT_c) \cdot x(n2NT_s - iT_c) \cdot e^{-j2\pi f_k(n2NT_s - iT_c)} \tag{2-1}$$

,where f_k, f_c and f_s have been previously given. Also, the corresponding sampling periods are $T_c = 1/f_c$ and $T_s = 1/f_s$, respectively. It is also assumed that the lowpass filter, $h(nT_s)$, and the signal $x(nT_s)$ are of equal length, LT_c, where L is the length of each channel [2]. Concerning the computational complexity of the SSM, the operation rate per unit time is given as [6]:

$$M_{SSM} = KW^2 \cdot \frac{W(N-4) - 2B(N+2)}{(W-B)(W-2B)} \tag{2-2}$$

, where :

$$K = -\frac{2}{3} \cdot \log(5\delta_1\delta_2) \tag{2-3}$$

, where δ_1 and δ_2 denote the acceptable passband and stopband ripples, respectively of the lowpass filter, $h(nT_s)$. Although it is not clear now, this represents a high level of computational complexity which can be greatly reduced in the polyphase method [1] [6]. The comparison of the operation rates for the different MCDDs, based on published results, will be shown after all three structures are presented [6].

2.2.2 : The Polyphase/FFT Method (PPM)

The Polyphase/FFT Method (PPM) is more interesting than the SSM because the PPM is more modular but involves less structural complexity [1-2] [6]. It will also be seen later on that the operation rate is about 10% of the SSM operation rate. By operation rate, we mean the number of filtering operations done per unit time [6]. Although better than the SSM, the PPM also has slight disadvantages. Unlike the SSM, we are constrained to the number of channels we can demultiplex. In general, the number of channels that can be demultiplexed must be a power of two, that is $N=2^{J}$, since the FFT processor is designed primarily for this number of channels [2] [6] [9]. We also require uniform channel spacing and symmetric stacking arrangement of the channels in order for the polyphase filters to be able to filter out the desired channel [1-2].

The PPM consists of two main parts, and they are the polyphase filters and the FFT processor. Each channel to be demultiplexed has its own branch containing a polyphase filter. Thus, for N channels to be demultiplexed, the PPM consists of N polyphase filters in N branches and one FFT processor [1-2] [6] [25]. Thus, our objective is to arrive at a single unified structure that can demultiplex a group of N channels with far smaller complexity than the SSM. The derivation of the PPM structure is done directly from one SSM module, depicted in Figure 2-2. Also, the demultiplexed output of the PPM can be derived from (2-1). If we define $\bar{x}_p(mT_c)$ as the p^{th} multiplexed input channel and $\bar{p}_p(mT_c)$ as the polyphase filter for the corresponding p^{th} channel, then the demultiplexed output for the desired k^{th} channel of the PPM is [2]:

$$y_k(mT_c) = (-1)^m \sum_{p=0}^{N-1} W_N^{-2kp} \cdot W_N^{-p} \cdot \left\{ \bar{p}_p(mT_c) * \bar{x}_p(mT_c) \right\} \tag{2-4}$$

, where * denotes digital convolution and $W_N = e^{j\frac{\pi}{N}}$. The PPM used to demultiplex an N channel MF-TDMA signal is shown in Figure 2-3.

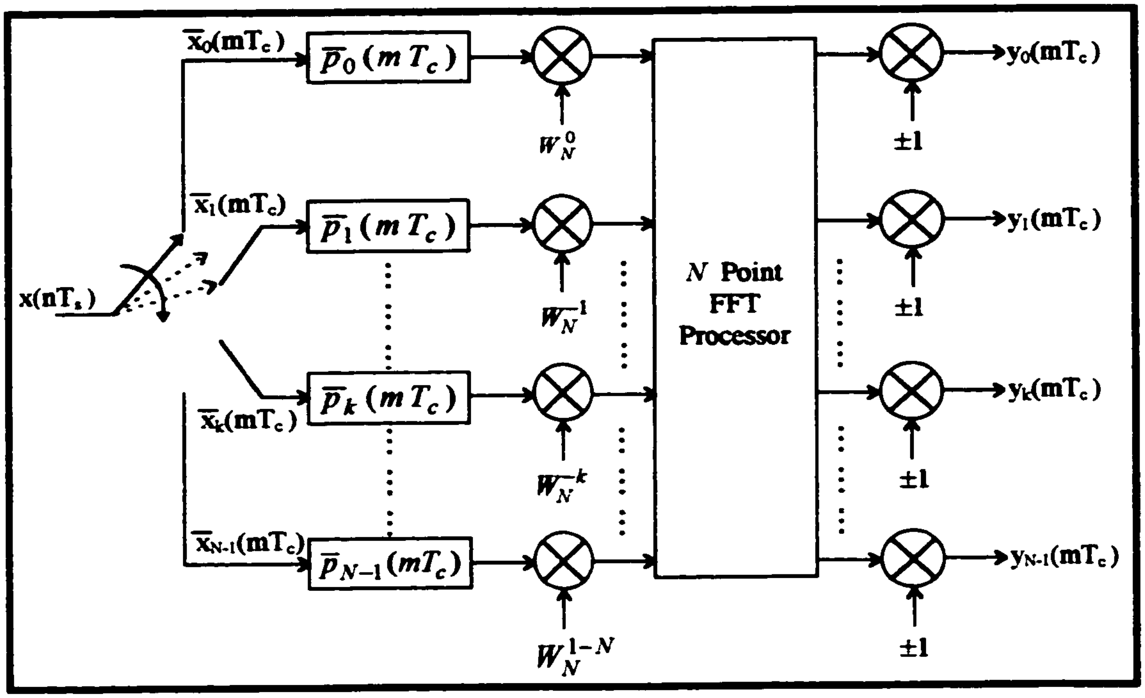


Figure 2-3. Structure of the Polyphase/FFT Method (PPM) used to demultiplex N channels [2].

The operation rate of the PPM used to determine its computational complexity is expressed as [6]:

$$M_{PPM} = 2W \cdot \left[\frac{1}{N} \cdot \left(\frac{\frac{2}{3} \cdot \log\left(\frac{1}{10\delta_1\delta_2} \right) \cdot 2NW}{W - 2B} \right) + 4\log_2 N \right] \tag{2-5}$$

, where W is the channel spacing which is uniform in this case, B is the polyphase filter bandwidth, N is the number of channels to be demultiplexed, δ_1 and δ_2 is the maximum allowable passband and stopband ripple, respectively, for $\overline{p}_p(mT_c)$. The computational efficiency of this method over the SSM will be shown.

The filters, $\overline{p}_p(mT_c)$, are called polyphase filters since this set of filters have the same amplitude response but different phase responses [25-26]. Although in (2-4) it has been assumed that the polyphase filters have an infinite impulse response (IIR), in reality they will be implemented as finite impulse response (FIR) filters. As an L-tap FIR filter, the amplitude response of $\overline{p}_p(mT_c)$ will approximate an allpass function. Assuming that the original filter,

$h(nT_s\ )$, has a linear phase response, the filter $\bar{p}_p(mT_c)$ will also have a linear phase response whose slope is proportional to p/N [2] [9] [25]. The name polyphase network thus comes from the fact that different paths in the network have different phase responses but the same amplitude response [25-26]. In the SSM, we recall that the lowpass filter, $h(nT_s)$, was used to filter out the individual channels. Because the polyphase filters were derived directly from these lowpass filters, the function of the polyphase filter bank is to extract the individual QPSK signals in each branch. Following the polyphase filter bank is the second part of the PPM which is the FFT processor. Finally, the set of alternating sign changes after the FFT processor is just a technicality of the derivation [2].

The process occurring in the PPM structure will now be summarized from left to right. The input MF-TDMA signal, $x(nT_s)$, with input sampling rate f_s , is subject to the commutator arm which rotates clockwise starting from $x_0\ (mT_c)$. To achieve the initial decimation from f_s to $f_s/2$, the arm rotates clockwise as indicated in Figure 2-3, with every sample of $x(nT_s)$, thus dropping every second sample [2]. Once we separate all samples of $x(nT_s)$, we see $1/2N$ of the total amount of samples in each branch thus the sampling rate is decimated from $f_s = 2N\cdot f_c$ to f_c [2] [25]. This concept of decimation in the time domain will be elaborated in section 2.3.2. The data in each branch is then passed through the corresponding polyphase filter which performs the channel filtering. The QPSK signals are then passed through the FFT processor and then toggled in sign before arriving at the output.

2.3 : The Multistage Multicarrier Demultiplexer/Demodulator

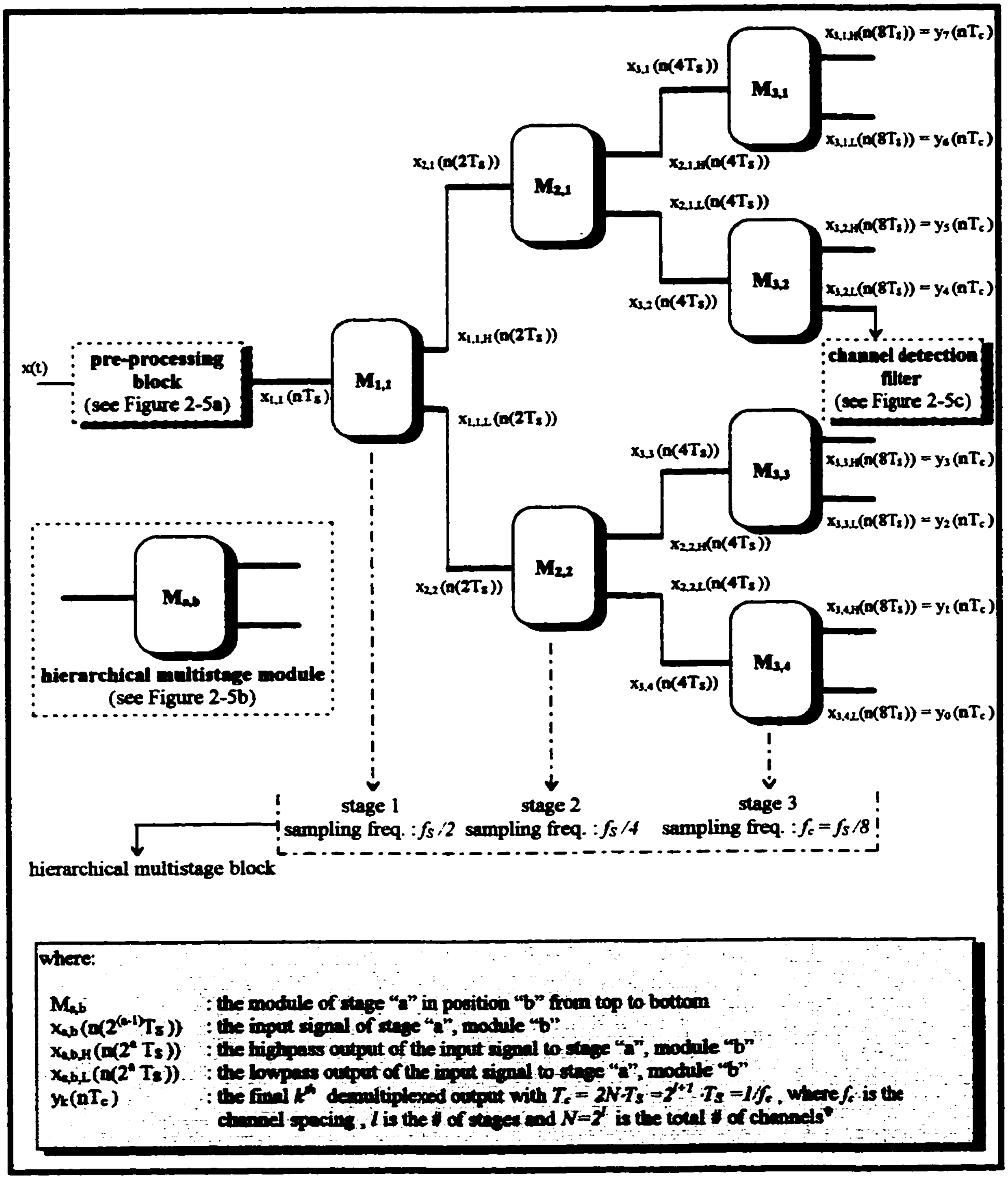
2.3.1 : Concept

We recall for the PPM that its usefulness is exploited so long as we have a uniform channel spacing, a symmetric stacking arrangement of the channels and the number of channels to be demultiplexed, N, is a power of two [1-2] [6]. The Multistage Multicarrier Demultiplexer/ Demodulator (M-MCDD) was developed using the same assumptions as the PPM for the input MF-TDMA signal [6-7]. Under these assumptions, the M-MCDD demultiplexes the symmetrical MF-TDMA signal by recursively dividing its spectrum into two halves, a highpass part and a lowpass part above and below the spectral midpoint π, respectively, through multiple stages of filtering and decimation until each individual channel is obtained at the output [1] [6] [13] [25]. Because the MF-TDMA signal spectrum keeps splitting into two, we must in the end have a number of channels, *N*, being a power of two [6-7]. So if the number of channels we wish to demultiplex is $N=2^l$, we require an *l* stage M-MCDD. Each stage of the M-MCDD contains identical modules which perform the demultiplexing process. Each module will be modeled as a linear time invariant system and will take a single input and produce two outputs in two separate branches corresponding to the highpass and lowpass portion of the input spectrum [6-7] [25]. The k^{th} stage of the M-MCDD ($0<k<l$) will contain 2^{k-1} modules and because the process through the M-MCDD is recursive, the 2^{k-1} modules, having a total of 2^{k-1} inputs, will produce 2^k outputs. It follows that through the final l^{th} stage, we need exactly *N/2* modules to provide *N* demultiplexed channels at the output.

Using identical modules offers some advantages over other MCDDs, the most notably being a high degree of modularity. Its modularity is attributed to the inclusion of identical components in every module such as the halfband filters and decimators, both of which will be described later on. Its modularity results in the M-MCDD being a relatively simple and cost-effective solution for a demultiplexer [1] [6-7] [13]. Other advantages of this demultiplexing method are its high level of design feasibility because of its simplicity due to modularity and its structural compactness resulting in a reduction in payload in comparison to the other MCDDs [6-7]. The M-MCDD uses *N-1* modules to demultiplex *N* channels [1] [6-7]. Furthermore, the *N-1* modules of the M-MCDD are arranged in a tree-like structure using $l=log_2N$ stages [1] [6-7] [25-26]. Also, the M-MCDD has a relatively low computational or structural complexity [6-7]. This is demonstrated in Table 2-1 in section 2.3.3. The low structural complexity of the M-MCDD provides a low power consumption. Based on implementations of the M-MCDD, it is known that the M-MCDD could be designed using 2µm CMOS technology with a power consumption of less than 50mW per channel [6].

2.3.2 : Operation and Structure

The essential elements of the M-MCDD structure will be considered here [6-8]. Figure 2-4 shows the overall M-MCDD structure consisting of three stages to demultiplex *N=8* channels. Also, Figures 2-5(a,b,c) show the structure of the pre-processing block, the hierarchical multistage module used in each stage of demultiplexing and the channel detection filter, respectively.

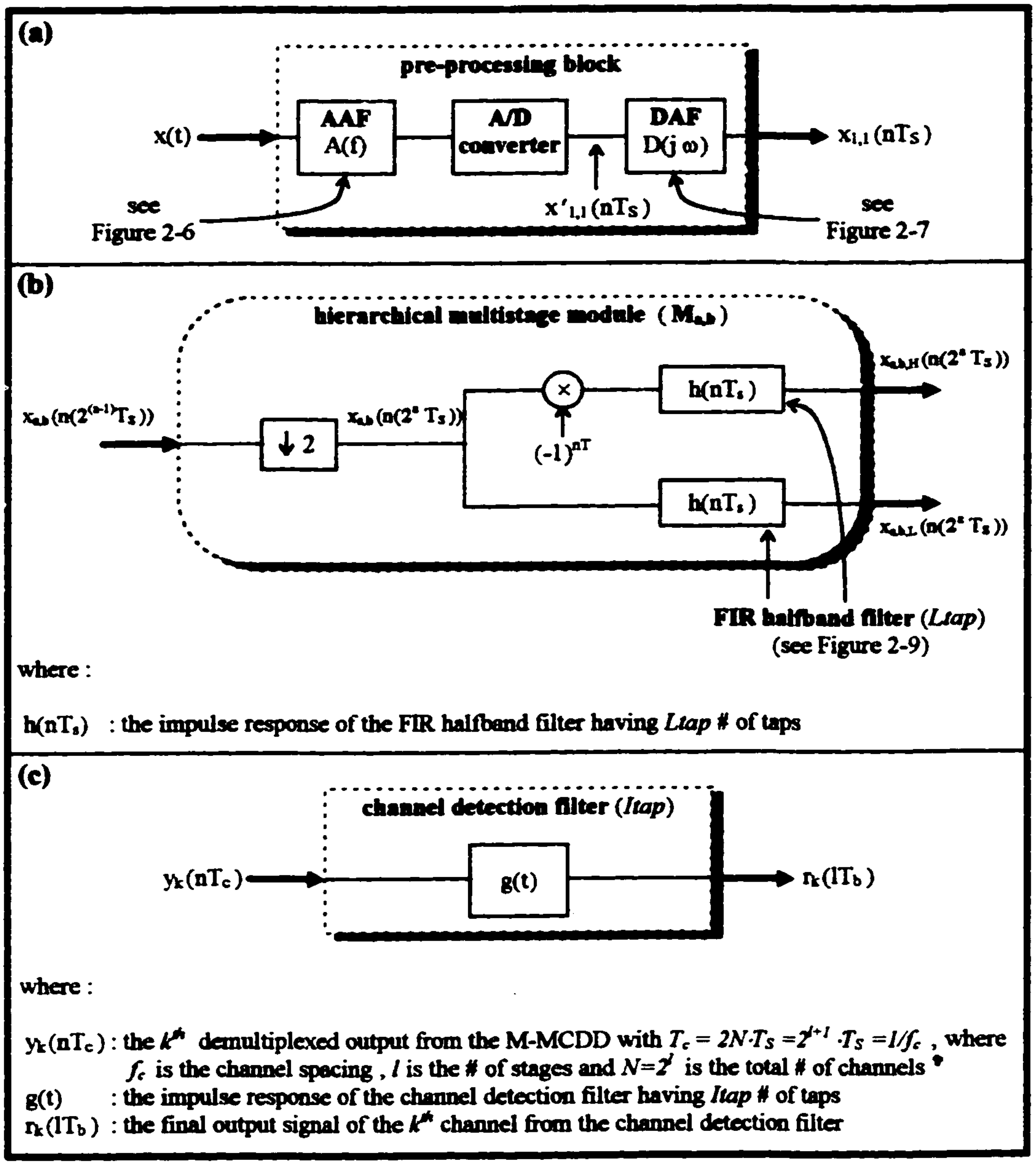

x(t)
pre-processing block (see Figure 2-5a)
$x_{1,1}(nT_s)$
$M_{1,1}$
$x_{1,1,H}(n(2T_s))$
$x_{1,1,L}(n(2T_s))$
$x_{2,1}(n(2T_s))$
$M_{2,1}$
$x_{2,1,H}(n(4T_s))$
$x_{2,1,L}(n(4T_s))$
$x_{2,2}(n(2T_s))$
$M_{2,2}$
$x_{2,2,H}(n(4T_s))$
$x_{2,2,L}(n(4T_s))$
$x_{3,1}(n(4T_s))$
$M_{3,1}$
$x_{3,1,H}(n(8T_s)) = y_7(nT_c)$
$x_{3,1,L}(n(8T_s)) = y_6(nT_c)$
$x_{3,2}(n(4T_s))$
$M_{3,2}$
$x_{3,2,H}(n(8T_s)) = y_5(nT_c)$
$x_{3,2,L}(n(8T_s)) = y_4(nT_c)$
channel detection filter (see Figure 2-5c)
$x_{3,3}(n(4T_s))$
$M_{3,3}$
$x_{3,3,H}(n(8T_s)) = y_3(nT_c)$
$x_{3,3,L}(n(8T_s)) = y_2(nT_c)$
$x_{3,4}(n(4T_s))$
$M_{3,4}$
$x_{3,4,H}(n(8T_s)) = y_1(nT_c)$
$x_{3,4,L}(n(8T_s)) = y_0(nT_c)$
$M_{a,b}$
hierarchical multistage module (see Figure 2-5b)
stage 1 sampling freq. : $f_S/2$
stage 2 sampling freq. : $f_S/4$
stage 3 sampling freq. : $f_c = f_S/8$
hierarchical multistage block

where:

$M_{a,b}$	: the module of stage "a" in position "b" from top to bottom
$x_{a,b}(n(2^{(a-1)}T_S))$	: the input signal of stage "a", module "b"
$x_{a,b,H}(n(2^a T_S))$	: the highpass output of the input signal to stage "a", module "b"
$x_{a,b,L}(n(2^a T_S))$	: the lowpass output of the input signal to stage "a", module "b"
$y_k(nT_c)$	: the final k^{th} demultiplexed output with $T_c = 2N \cdot T_S = 2^{l+1} \cdot T_S = 1/f_c$, where f_c is the channel spacing, l is the # of stages and $N=2^l$ is the total # of channels*

Figure 2-4. The overall structure for the M-MCDD for the case of $N=2^3=8$ channels [1] [6] [13].

* The assumption here is that an extra decimation process is present after the final stage. To make the model modular, this assumption has not been taken in the analysis and as a result, $y_k(nT_c)$ is bandlimited to π



where :

$h(nT_s)$: the impulse response of the FIR halfband filter having *Ltap* # of taps

where :

$y_k(nT_c)$: the k^{th} demultiplexed output from the M-MCDD with $T_c = 2N \cdot T_S = 2^{l+1} \cdot T_S = 1/f_c$, where f_c is the channel spacing , l is the # of stages and $N=2^l$ is the total # of channels [♦]

g(t) : the impulse response of the channel detection filter having *Itap* # of taps

$r_k(lT_b)$: the final output signal of the k^{th} channel from the channel detection filter

Figure 2-5(a,b,c). Inset showing the components of the : (a) pre-processing block
(b) hierarchical multistage module($M_{a,b}$)
(c) channel detection filter [1] [6] [13].

The M-MCDD consists of two blocks: pre-processing and hierarchical multistage, as shown in Figure 2-4. The implementation of the M-MCDD requires a pre-processing stage, shown in Figure 2-5(a), consisting of three main components: the analog anti-aliasing filter (AAF), the

[♦] The assumption here is that an extra decimation process is present after the final stage. To make the model modular, this assumption has not been taken in the analysis and as a result, $y_k(nT_c)$ is bandlimited to π

analog-to-digital converter (A/D) and the digital anti-aliasing filter (DAF) [6] [13]. We use a lowpass analog anti-aliasing filter (AAF) to meet the Nyquist Criterion. In other words, the AAF is used to bandlimit the analog MF-TDMA input signal, $x(t)$, before sampling to avoid aliasing [9]. The analog MF-TDMA input signal should ideally have a maximum one-sided frequency of $f_{max} = NB$, where N represents the number of channels to be demultiplexed and B is the corresponding bandwidth for each channel. Here, with a symmetric stacking arrangement and to obtain sufficient channel separation, we have $B<f_c$. [1-2] Thus, to bandlimit $x(t)$ to $f_{max} = NB$, we choose the cutoff frequency of the AAF to be $f_{max,AAF} = NB$ [6-7]. Figure 2-6 illustrates the AAF frequency response.

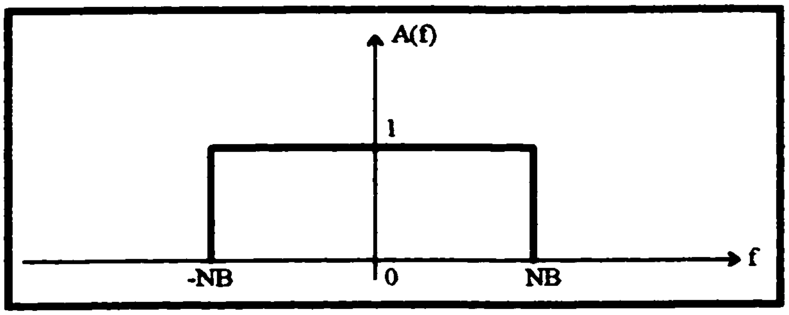


Figure 2-6. Frequency response of the analog anti-aliasing filter (AAF) [6-7].

Because the input to the demultiplexer , $x(t)$, is analog, it will have to be sampled before it can be further processed. We use an A/D converter with a sampling rate of f_s thus resulting in the digital signal $x(nT_s)$ [9]. From what is known about the characteristic of the AAF, and using the Nyquist criterion, we can determine the minimum value of the initial sampling frequency, f_{Si} , for the A/D converter. It can be determined from the maximum frequency of $X(f)$ after filtering through the AAF [6-7] [9]. According to the Nyquist criterion, the minimum sampling rate must be at least twice the maximum frequency of the input signal and since, after the AAF, $X(f)$ has $f_{max} = NB$, we determine that $f_{S,min} > 2NB$ [9]. From actual implementations of the M-MCDD, the A/D conversion of the input signal is done by using a fast sample-and-hold cascaded by an slow analog-to-digital converter (A/D). Because of the slow A/D converter, we introduce an oversampling of $f_{S,min}$ by a factor of two over the Nyquist Rate, thus the initial sampling rate, f_{Si} , is chosen to be $f_{Si} = 1/T_{Si} = 4NB$ [6] [9]. In the analysis, however, this factor has not been taken into account. We therefore accept $f_s = 1/T_s = 2Nf_c$ as the initial sampling frequency, assuming that the symmetrically stacked signals occupy the entire bandwidth allocated to them, that is, $B=f_c$. This, however, will not affect the analysis since there is no dependency on how many samples are taken per symbol.

The third component of the pre-processing block is the lowpass digital anti-aliasing filter (DAF), which has a cutoff frequency of $\omega=\pi$. The reason for introducing the DAF after sampling the input MF-TDMA signal is to avoid aliasing of the signal once decimation occurs through the first stage. Once $x(nT_s)$ passes through the DAF, it is no longer a real signal but it is rather a complex signal since only the portion of the signal between 0 and π is retained [9] [25-26]. Note that this bandlimiting to π is possible without the loss of information since the analog input MF-TDMA signal is real thus once sampled, $x'_{1,1}(nT_s)$ is symmetric about π [6-7] [9]. It is also a result of the assumed symmetric stacking arrangement of the channels [1] [9]. The DAF can ideally be shown as a brickwall filter bandlimited to π, and this is done by Figure 2-7 [6] [9] [25]. From Figure 2-7, we note that the frequency range has been normalized such that $\omega=2\pi f_s$ [9]. Because the function of the DAF is similar to that of the halfband filter, that is, to filter out the lowpass spectrum of the input signal, for the simplicity of analysis, we can approximate the filter characteristic of the DAF by that of a halfband filter which is used in each stage of demultiplexing [13].

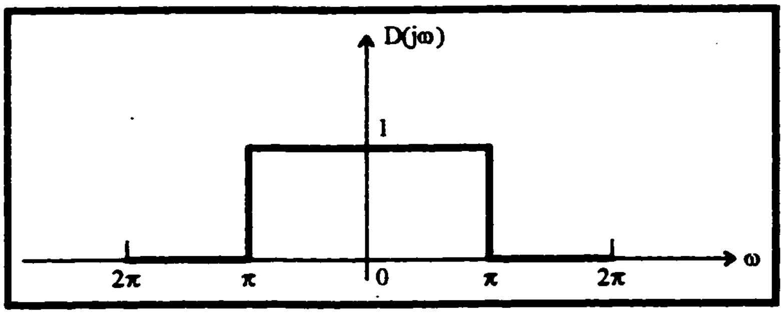


Figure 2-7. Ideal filter characteristics for the digital anti-aliasing filter (DAF) [6] [9] [25].

The hierarchical multistage block involves identical modules in every stage of demultiplexing. The process within each module is essentially broken down into two major parts: decimation and filtering [6-7]. Decimation is an essential component of the M-MCDD structure for the same reasons it was essential for the SSM and the PPM structures. That is, its purpose is to convert the composite input signal to baseband by reducing its sampling frequency [6-7] [25].

The decimation process can be visualized in both the time and frequency domains. In the frequency domain, the decimation process extends the bandwidth of its input spectrum by the same factor that the input sampling rate is reduced by. For example, if we are decimating a signal whose sampling rate is f_S so that its new sampling rate will become $f_{S,new} = f_S / D$, its spectrum will then be bandlimited to $D \cdot \omega_{max}$, given that it was originally bandlimited to $\omega_{max} < 2\pi/D$ [25]. Now, we must choose the appropriate decimation factor for each stage of the M-MCDD. After any stage of demultiplexing, the output signal is bandlimited to $\omega_{max} = \pi$. We require that the spectrum after decimation occupy the entire frequency range, that is, from 0 to 2π. Since we need the frequency range to double, we conclude then that the decimation rate required in every stage of demultiplexing is $D=2$ [6-7] [25].

In the time domain and in context of the M-MCDD, the decimation process may be visualized by a division of a sampled sequence in two separate branches, similar to the process occurring in the PPM [2] [25]. Within the M-MCDD, the decimation by a factor of two discards exactly half of the input samples to every stage which are not part of the desired output. Therefore, after decimation in each stage, the lowpass and highpass output sequences each contain exactly half of the input sequence, thus both outputs have the same number of samples. At the same time, however, the highpass and lowpass output sequences are mutually exclusive in their content. If the lowpass sequence is chosen to be $x_{LD}[n]=x[2n]$, the highpass sequence will be $x_{HD}[n] = x[2n+1]$, where $n=0,\pm 1, \pm 2, \pm 3, \ldots$ Figure 2-8 demonstrates decimation in the time domain [25].

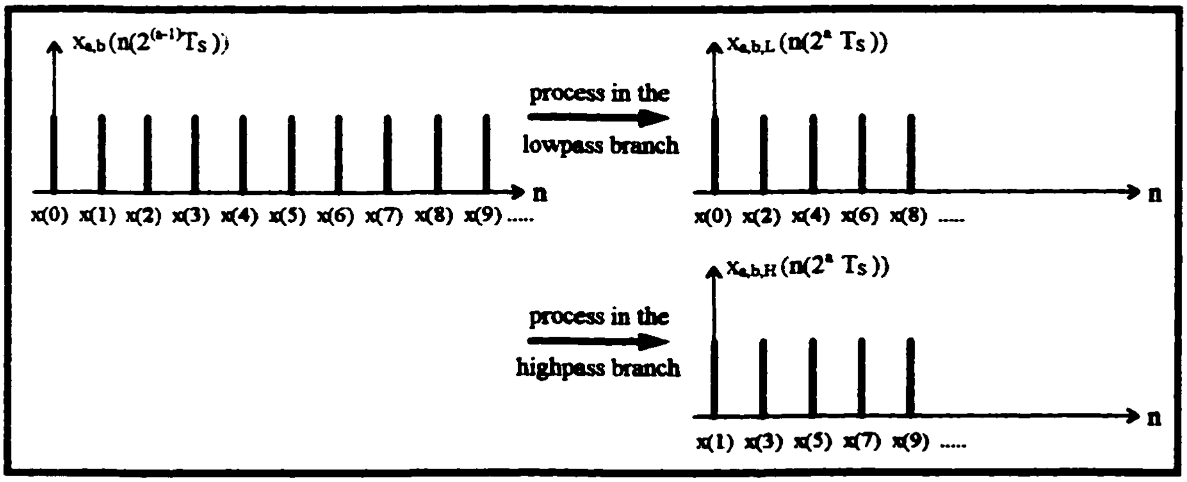

$x_{a,b}(n(2^{(a-1)}T_S))$
process in the
lowpass branch
n
x(0) x(1) x(2) x(3) x(4) x(5) x(6) x(7) x(8) x(9) .....
$x_{a,b,L}(n(2^a T_S))$
n
x(0) x(2) x(4) x(6) x(8) .....
$x_{a,b,H}(n(2^a T_S))$
process in the
highpass branch
n
x(1) x(3) x(5) x(7) x(9) .....

Figure 2-8. Decimation and filtering process in the time domain for module $M_{a,b}$ [25].

Having shown the decimation process in the modules, we will discuss the other process occurring, that is, filtering. The filtering requirements are achieved by using halfband filters which divide the input signal spectrum into the highpass and lowpass portions in two separate branches of the module [6-7]. The impulse response of the FIR halfband filter is [25]:

$$h(nT_s)=\begin{cases}\frac{1}{2} & ;n=0\\ \dfrac{\sin(\pi nT_s/2)}{\pi nT_s} & ;n=\pm1,\pm3,\pm5\ldots.\\ 0 & ;n=\pm2,\pm4,\pm6\ldots.\end{cases} \tag{2-6}$$

with $h(nT_s)$ being an even function. Note that T_s is the initial sampling period. The impulse and frequency responses of the FIR halfband filter are illustrated in Figure 2-9 [27].

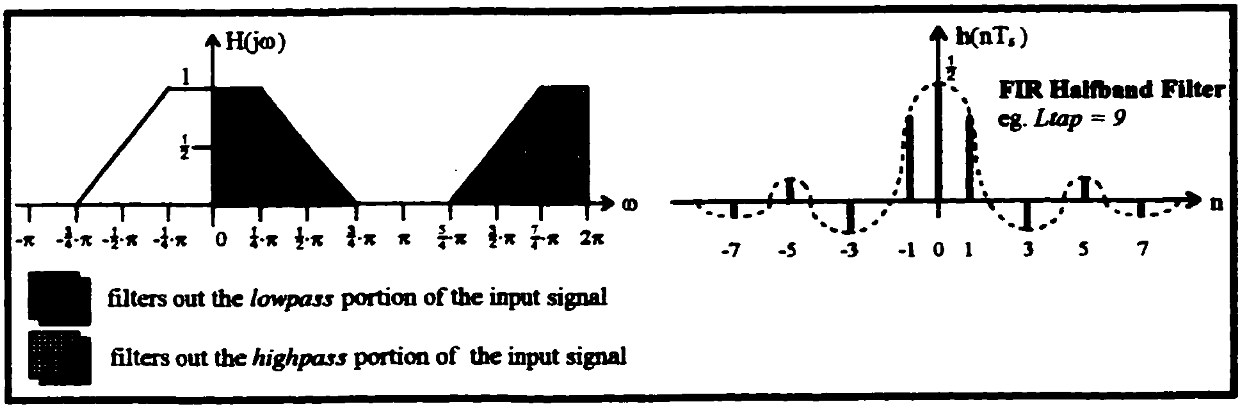


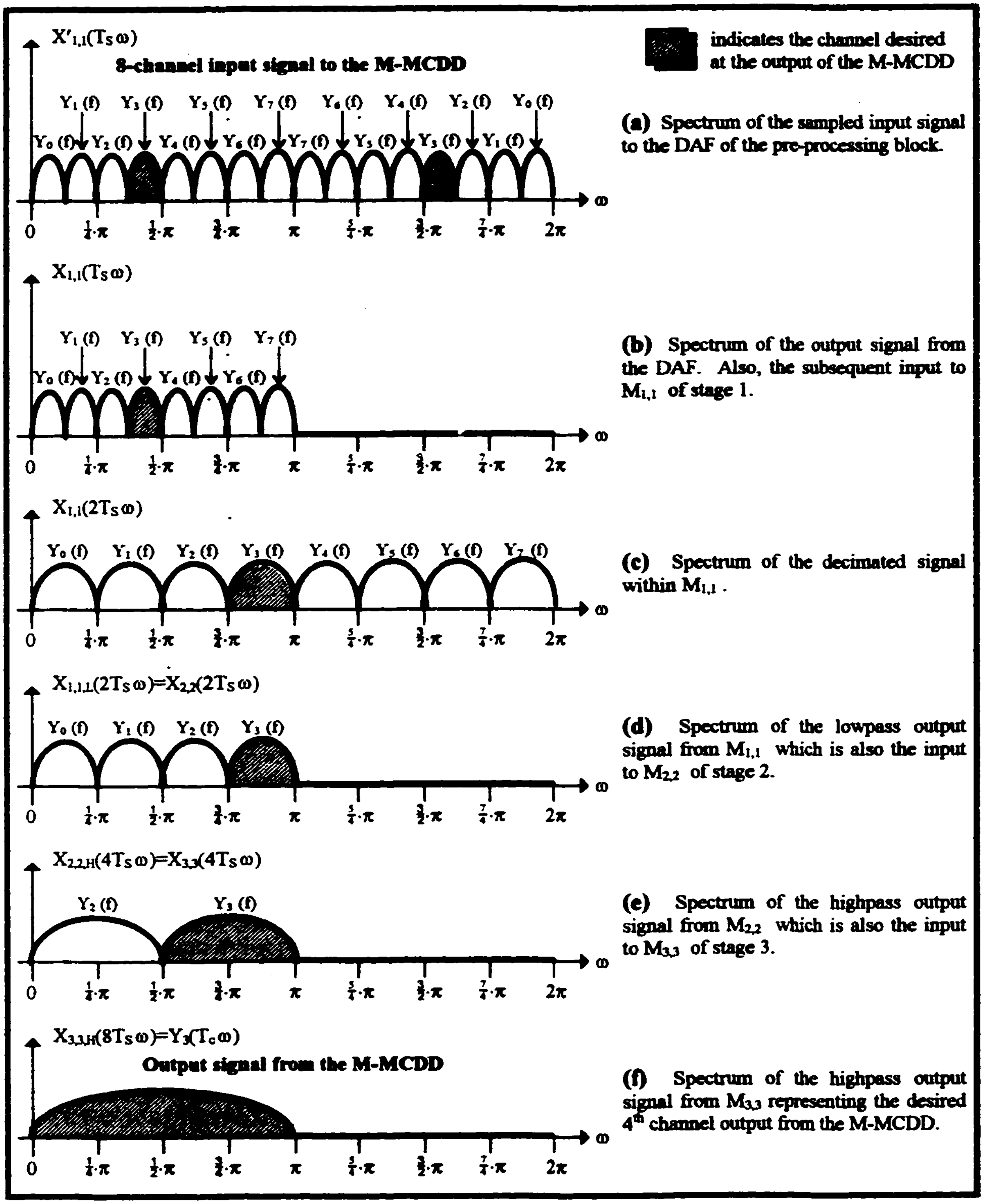
Figure 2-9. Frequency and impulse response of the FIR halfband filter [27].

The decimation and filtering operations for a three stage M-MCDD are demonstrated in Figure 2-10 [6] [25-26]. From this figure, we note the redundant process occurring through each stage [6]. Notice that whenever the highpass portion of the spectrum is desired, the filtered spectrum must be shifted by $\pm\pi$ in order to prepare it for the decimation process of the following stage. In other words, we require a shifting of the highpass spectrum such that $\omega_{min}=0$, where it originally would be $\omega_{min}=\pi$. After shifting, this would be the same ω_{min} as the corresponding lowpass portion prior to shifting. Because of this shifting, the highpass and lowpass filtering components within each module are different yet related. Their relationship is shown by Figure 2-5(b) and is also indicated by [1] [6] :

$$h_H(nT_s)=(-1)^n\cdot h_L(nT_s)=(-1)^n\cdot h(nT_s) \tag{2-7}$$

The process that proceeds the hierarchical multistage block is the channel detection filter stage, shown in Figure 2-5(c). This consists of a channel detection filter, $g(t)$, which has the k^{th} channel demultiplexed output from the hierarchical multistage block, $y_k(nT_c)$, as an input. As part of the demodulation process, this filter does a sample rate conversion to detect the k^{th} channel signal [2] [4] [26]. The corresponding output is $r_k(lT_b)$. The analysis involving the channel detection filter stage is elaborated upon in section 3.1.1.

As an extra note, in Figures 2-4 and 2-10, a carrier spacing of $f_c=f_s/N$ is reflected at the output [6]. However, the output sampling frequency must be $f_c=f_s/2N$ in order to satisfy the Nyquist Criterion and to ensure the output signal is real [9]. Thus, to make the process complete, we need another decimation process with $D=2$ after the final stage but this has not been taken into account in the analysis so as not to effect the modularity of the M-MCDD [6-7] [9] [25].



▨ indicates the channel desired at the output of the M-MCDD

(a) Spectrum of the sampled input signal to the DAF of the pre-processing block.

(b) Spectrum of the output signal from the DAF. Also, the subsequent input to $M_{1,1}$ of stage 1.

(c) Spectrum of the decimated signal within $M_{1,1}$.

(d) Spectrum of the lowpass output signal from $M_{1,1}$ which is also the input to $M_{2,2}$ of stage 2.

(e) Spectrum of the highpass output signal from $M_{2,2}$ which is also the input to $M_{3,3}$ of stage 3.

(f) Spectrum of the highpass output signal from $M_{3,3}$ representing the desired 4th channel output from the M-MCDD.

Figure 2-10(a-f). The process occurring through each M-MCDD stage until the output [6] [25-26].

2.3.3 : Comparison of MCDDs

To compare the operation rates of the SSM, PPM and the Multistage Multicarrier Demultiplexer/Demodulator (M-MCDD), Table 2-1 has been compiled. The operation rates for the SSM and PPM is based on (2-2) and (2-5), respectively. That of the M-MCDD is an approximation based on an architectural analysis. The assumptions made in Table 2-1 are a 32-channel demultiplexer, uniform channel bandwidth of 8.4kHz and A/D converter wordlength of 12 bits [6].

Table 2-1. Comparison of the three MCDDs presented [6].

Method	Channel spacing	No of channels	Decimation rate	operation rate
SSM	no restriction	no restriction	no restriction	1,167,230.4/ms
PPM	uniform	power of two	2 × (number of channels)	143,135.2/ms
M-MCDD	uniform	power of two	power of two	88,267.2/ms

It is clear from this table, under the specified assumptions, that there is a significant improvement in the computational or structural complexity. It is also clear from the table that each method is useful under certain limiting conditions. For the SSM, there are few, if not any, limiting conditions for the MF-TDMA input signal since each channel is demultiplexed independently [1] [6]. For the PPM and M-MCDD, its usefulness is exploited so long as we have a uniform channel spacing, a symmetric stacking arrangement of the channels and the number of channels to be demultiplexed, *N*, is a power of two [1-2] [6]. Generally, the PPM structure performs the same functions as the SSM, that is, filtering and decimation to obtain the individual channels from the MF-TDMA input signal, however as a unified structure, the PPM is more compact than the SSM [1-2]. From what was stated of the M-MCDD, however, it is preferred because it is more modular, feasible, compact and power efficient than both the SSM and PPM [6-7]. In support of its relative compactness, the M-MCDD uses *N-1* modules to demultiplex *N* channels, whereas the SSM uses *N* individual circuits to achieve the same end [1] [6-7]. From Table 2-1, we see it is also preferred due to the fact that it is more efficient computationally.

2.4 : Statement of the Research Problem

An important consideration in the implementation of OBP satellite communication systems is the design of group demodulators [1]. The group demodulators, or MCDDs, are used to process MF-TDMA signals with far smaller complexity than using individual channel demultiplexers [1-2] [4] [6-8] [25-26] We are faced with several types of MCDDs which can be used in OBP satellite communications systems. The first of these is the SSM, which represents the individual channel demultiplexer and has a relatively high structural complexity and payload [1] [4] [6-8] [25-26] The other two types of MCDDs considered are the PPM and the M-MCDD [1] [6] [8] [25-26]. The M-MCDD has been chosen as preferred structure to perform group demodulation because of its modularity, compactness, low complexity and low power consumption in comparison to the SSM and PPM [6]. The M-MCDD must be designed such that it will permit the detection of the desired symbol on the desired channel with a minimum probability of error. That is, it must minimize the ability of the demodulation process to detect a symbol other than the desired symbol. When attempting to detect the desired symbol, two sources of interference which appear in the demodulated output may cause the data to be misinterpreted. These two sources are intersymbol interference (ISI) and interchannel interference (ICI) [2] [10-12]. Interchannel interference arises between adjacent channels because of imperfect filtering and insufficient channel separation in the MCDD [2]. Concerning imperfect filtering, an issue in the design of the M-MCDD is that the halfband filters and the channel detection filter are finite impulse response (FIR) filters as opposed to infinite impulse response (IIR) filters [2] [6] [25-27]. The finite number of filter taps or delays cause a non-zero stopband ripple to occur within the rejection region. As a result, interchannel interference (ICI) or "crosstalk" will occur since we cannot have perfect filtering of the channels. This phenomenon is illustrated in Figure 2-11 [2] [9]. It is desirable to reduce the amount of ICI and to do so, we must increase the filter lengths [9]. Consequently, the more filter taps we use in the design of the M-MCDD, the closer its performance will be to that of ideal QPSK demodulation [2]. However, by increasing the number of filter taps, we will increase the structural and computational complexity of the M-MCDD [6]. The design problem for the M-MCDD is thus a tradeoff between reducing the degradation caused by crosstalk while keeping the complexity of the M-MCDD at an acceptable level.

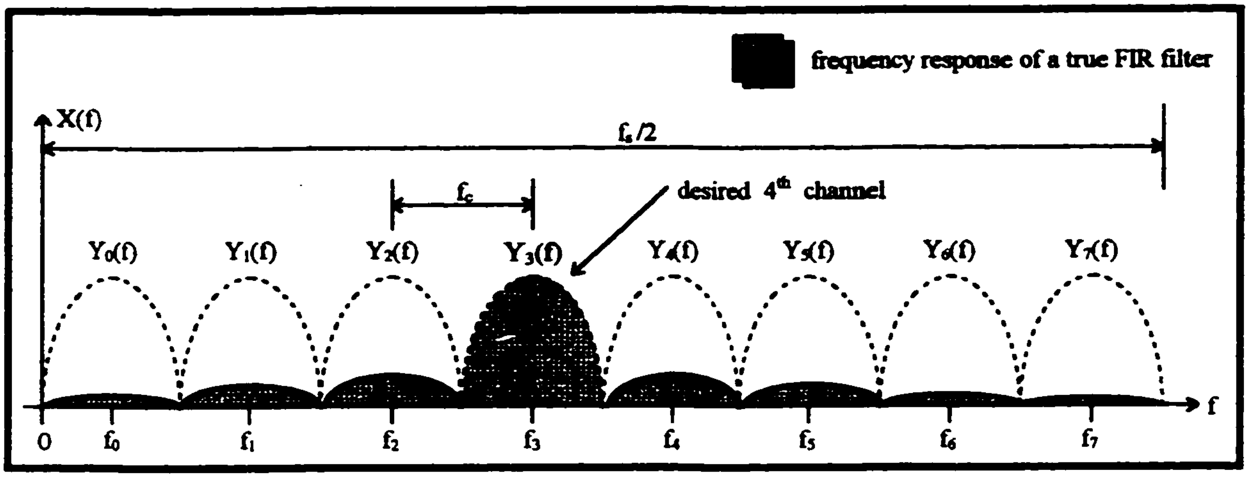


Figure 2-11. The true response of an FIR filter resulting in crosstalk due to imperfect filtering [2].

In addition to the imperfect filtering of the channels, ICI or crosstalk is also the result of insufficient channel separation between adjacent channels [2]. A non-ideal case of Figure 2-1 is shown by Figure 2-12 where we see a crossover between adjacent channels. This is one reason for requiring a symmetric stacking arrangement of the channels in the input MF-TDMA signal, that is, to guarantee sufficient channel separation between adjacent channels [1-2].

Whereas ICI or crosstalk occurs between adjacent channels, intersymbol interference (ISI) occurs within each individual channel. The symbols which are demodulated interact with each other, similar to what is seen at the output of a matched filter. When ISI occurs between adjacent symbols, detection of the desired symbol becomes difficult and obscured [23-24]. Together, ISI and ICI comprise adjacent channel interference (ACI) and it is this phenomenon that will be investigated further in this thesis since it significantly effects the performance of the M-MCDD and in effect plays an important role in its design [2] [10-12].

Although the M-MCDD is not a new concept and actual structural implementations exist, the effect of ACI on its performance has not been determined [6] [13-14]. We are faced with two different methods capable of analyzing the effect of interference on the performance of the M-MCDD. These two methods are analysis and simulation [2] [13]. Most of the work to date on performance evaluation of MCDD structures has involved simulation. This method, however, is known to be very time consuming, especially for evaluation at low bit error rates [2]. For example, to obtain acceptable simulation measurements, we require 100 errors being recorded. Thus, a typical simulation at a symbol error rate (SER) of 10^{-4} requires a million symbols or a full day simulation. A SER of 10^{-6} requires one hundred times as many symbols which may take up to a week. Thus, the development of analytical tools for performance evaluation is a valuable asset since an analytical tool is significantly faster and more accurate than a simulation program [2] [10-11] [13]. Therefore, in this research, we will use an additive white Gaussian noise (AWGN) analysis in the presence of ACI to determine the probability of bit error expression used to evaluate the performance of the M-MCDD. It should be noted that we do not compare our analytical results with a baseline simulation result. Our purpose is simply to develop an analytical model.

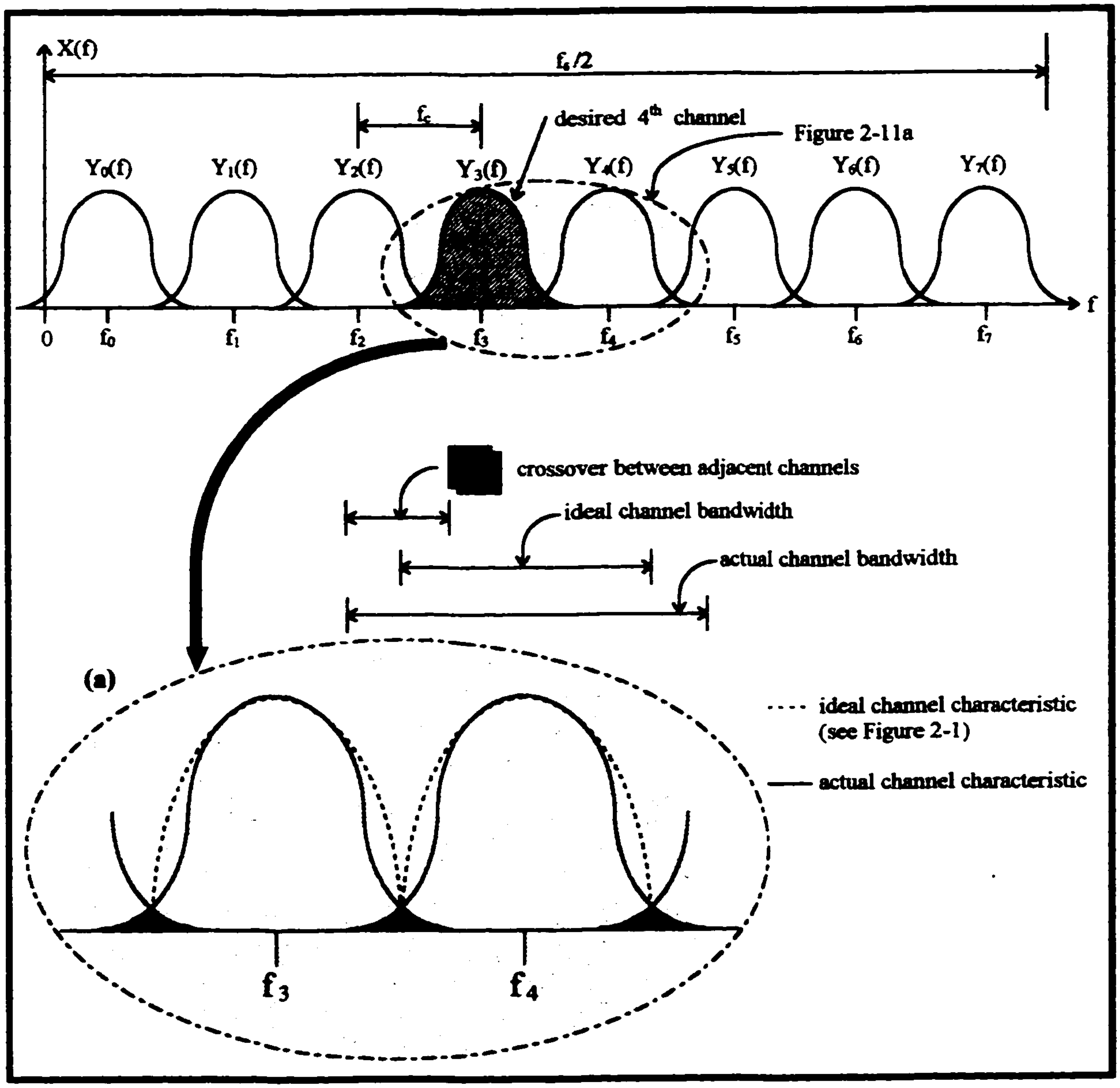


Figure 2-12. Input MF-TDMA signal showing insufficient channel separation causing ICI (a). Inset showing difference between ideal and actual channel characteristic [2].

Until present, analytical techniques have been avoided because of difficulties in arriving at a numerical solution for the probability of bit error. These difficulties arise since the probability density function involved in the probability of bit error expression is unknown and virtually impossible to find. The probability density function is unknown because it involves an infinitely large amount of interfering symbols on each channel [2] [10-12]. In order to arrive at a solution, we rely on numerical approximation techniques to solve the probability of bit error [2]. Two techniques used to solve the effect of interference on the performance of the M-MCDD are the Gauss Quadrature Rule (GQR) [10-12] [15] [17] and the Fourier Series Expansion technique of Beaulieu (FSE) [16]. Analytical tools will be developed using these two techniques. From these tools, BER curves will be determined for the performance analysis of the M-MCDD. Both analytical tools will then be studied based on results of different BER curves. We will vary the design parameters and see the effect on the convergence of the BER in both cases. It will then be concluded which of the two techniques is better suited to analyze the performance of the M-MCDD based on the CPU timing requirements [2].

In addition to the numerical accuracy of the results, the development of analytical tools will also enable us to evaluate the system performance of the M-MCDD to be able to wisely determine its filtering requirements. That is, given a certain performance and degradation level a design engineer wishes to achieve in the implementation of the M-MCDD, the results of the system performance for the M-MCDD will tell what are the possible filter lengths we can choose from [10-12].

Another issue in the design of the M-MCDD is that we cannot achieve infinite precision in the filter taps for the digital filters. That is to say, the coefficients for the filter taps cannot be real-valued [9]. Therefore, we are not only faced with a choice of the number of filter taps for the M-MCDD but we must also choose the number of quantization bits per filter tap based on a maximum allowed degradation in performance relative to the performance of the M-MCDD using infinite quantization or real-valued filter tap coefficients [9] [28]. In particular, the AWGN analysis will be extended so as to include the effect of halfband filter coefficient quantization on the performance of the M-MCDD. To evaluate the amount of degradation in performance due to quantization noise, the analytical tool will be extended using the Gram-Charlier Series Expansion Technique [2] [14] [17-22].

Thus, to resummarize the two objectives of the research stated in the introduction and this section, we first wish to evaluate the performance of the M-MCDD in the presence of AWGN and

ACI to develop an analytical model. This will help determine the number of filter taps in the halfband filters and channel detection filter for the design of the M-MCDD. Secondly, we will assume that the halfband filter coefficients are represented by a finite number of quantization bits. Thus, we will extend the AWGN analysis and our analytical model so as to include the effect of quantization noise on the performance of the M-MCDD. The extended analytical model will help us determine the number of quantization bits per halfband filter tap. In relation to the above objectives, to effectively choose the number of filter taps and quantization bits per halfband filter in the design of the M-MCDD, we will use the numerical results, compiled graphically, as well as the computational complexity. The computational complexity will be analyzed based on derived expressions. Therefore, the requirements of a design engineer can be satisfied based on a desired performance and degradation levels and maximum allowable computational complexity.

Chapter Three

Effects of Adjacent Channel Interference on Group Demodulation Performance

3.1 : AWGN Analysis of the M-MCDD Structure

3.1.1 : Input FDMA signal and Rate Conversion Stage

The input to the M-MCDD is a baseband signal consisting of N FDMA signals, as shown in Figure 2-1, and noise :

$$x(t) = \sum_{k=0}^{N-1} s_k(t) + z(t) \tag{3-1}$$

Assuming that the k^{th} FDMA signal is QPSK, it will have the form :

$$s_k(t) = A_k \cdot e^{j2\pi f_k t} \cdot \sum_{i=-\infty}^{\infty} e^{j[\alpha_{k,i}+\phi_k]} \cdot h_s(t - iT_b - \gamma_k) \tag{3-2}$$

where A_k is the carrier amplitude, $f_k = (k+\frac{1}{2})\cdot f_c$ is the carrier frequency of the desired k^{th} channel, $\alpha_{k,i}$ is the data phase in the i^{th} symbol interval, γ_k and ϕ_k represent random timing and phase offsets, respectively, $h_s(t)$ is the transmit filter, T_b is the symbol duration and $z(t)$ is an additive white Gaussian noise (AWGN) process.

After the channels are demultiplexed, a channel detection filter stage, as is shown in Figure 2-4 and Figure 2-5(c), is required to convert from the channel sample rate $f_c = 1/T_c$ to the symbol rate $f_b = 1/T_b < f_c$ in order to detect the desired k^{th} channel. The output signal may be expressed as [2] [25-26] :

$$r_k(lT_b) = r_k[(\beta_l + \mu_l)T_c] = \sum_{\xi=I_1}^{I_2} y_k[(\beta_l - \xi)T_c] \cdot g[(\xi + \mu_l)T_c] \tag{3-3}$$

where $g(t)$ is the impulse response of the channel detection filter, $y_k(nT_c)$ is the k^{th} channel demultiplexed output from the M-MCDD and β_l and μ_l are the basepoint index and the fractional interval, respectively, and are defined by [2] :

$$\beta_l = \operatorname{int}\left[\frac{l \cdot T_b}{T_c}\right] \tag{3-4}$$

$$\mu_l = \frac{l \cdot T_b}{T_c} - \beta_l \tag{3-5}$$

3.1.2 : Noiseless k^{th} output signal

The first step in analyzing the M-MCDD structure is to derive the form of the k^{th} desired output channel. This is possible by considering the structure presented in Figure 2-4. If we assume that we have an l-stage M-MCDD structure which will produce $N=2^l$ output signals, the analog input signal is bandlimited to NB and the spectral characteristics of the DAF approximate those of $h(nT_s)$, we obtain (see Appendix 3.1.1 for derivation[†]) [13] :

$$y_k(mT_c) = \begin{cases} (-1)^m \sum_{n_0=-L}^{L} \cdots\cdots \sum_{n_l=-L}^{L} \left\{ \begin{array}{l} \prod_{j=0}^{l}\left(h(n_j \cdot 2^j T_s)\right) \cdot \\ x((2^l m - \sum_{j=0}^{l} 2^j n_j)T_s) \cdot e^{j\pi(k+\frac{1}{2})\cdot \sum_{j=0}^{l} n_j} \end{array} \right\} & ; \quad k = 0 \ , \ k \ even \\ (-1)^m \sum_{n_0=-L}^{L} \cdots\cdots \sum_{n_l=-L}^{L} \left\{ \begin{array}{l} (-1)^{n_0 T_s} \cdot \prod_{j=0}^{l}\left(h(n_j \cdot 2^j T_s)\right) \cdot \\ x((2^l m - \sum_{j=0}^{l} 2^j n_j)T_s) \cdot e^{j\pi(k+\frac{1}{2})\cdot \sum_{j=0}^{l} n_j} \end{array} \right\} & ; \quad k \ odd \end{cases} \tag{3-6}$$

where $T_c = 2^l \cdot T_s$ is the sampling period at the output of the demultiplexer and the halfband filter has exactly $2L+1$ taps.

3.1.3 : Final Output

Combining (3-1) through (3-3) and (3-6), we arrive at an expression for the final output from the channel detection filter stage assuming k^{th} desired channel and α^{th} desired symbol (see Appendix 3.1.2 for derivation):

[†] For simplicity, all derivations are given in the Appendix

$$
\begin{aligned}
r_k(lT_b) = A_k \cdot e^{j\phi_k}\left[e^{j\theta_{k,\alpha}} \cdot u(0) + \sum_{t\neq\alpha} e^{j\theta_{k,t}} \cdot u(\alpha T_b - iT_b)\right] \\
+ \sum_{q\neq k} A_q \cdot e^{j\phi_q} \sum_{t} e^{j\theta_{q,t}} \cdot v(\alpha T_b - iT_b) + z_k(lT_b)
\end{aligned}
\tag{3-7}
$$

where the *intersymbol interference* (ISI) from the k^{th} channel is :

$$
u(\alpha T_b - iT_b) = \begin{cases}
\sum\limits_{\xi=I_1}^{I_2}\sum\limits_{n_0}\cdots\cdots\sum\limits_{n_l}\left\{\begin{array}{l}
\prod\limits_{j=0}^{l}\left(h(n_j\cdot 2^j T_s)\right)\cdot \\
h_s[(2^l(\beta_l-\xi)-\sum\limits_j 2^j n_j)\cdot T_s - iT_b - \gamma_k]\cdot \\
g[(\xi+\mu_l)T_c]
\end{array}\right\} & ; \quad k=0 \ , \ k \ even \\
\sum\limits_{\xi=I_1}^{I_2}\sum\limits_{n_0}\cdots\cdots\sum\limits_{n_l}\left\{\begin{array}{l}
(-1)^{n_0\cdot T_s}\cdot\prod\limits_{j=0}^{l}\left(h(n_j\cdot 2^j T_s)\right)\cdot \\
h_s[(2^l(\beta_l-\xi)-\sum\limits_j 2^j n_j)\cdot T_s - iT_b - \gamma_k]\cdot \\
g[(\xi+\mu_l)T_c]
\end{array}\right\} & ; \quad k \ odd
\end{cases}
\tag{3-8}
$$

and the *interchannel interference* (ICI) from all other interfering channels is :

$$
v(\alpha T_b - iT_b) = \begin{cases}
\sum\limits_{\xi=I_1}^{I_2}\sum\limits_{n_0}\cdots\cdots\sum\limits_{n_l}\left\{\begin{array}{l}
\prod\limits_{j=0}^{l}\left(h(n_j\cdot 2^j T_s)\right)\cdot \\
h_s[(2^l(\beta_l-\xi)-\sum\limits_j 2^j n_j)\cdot T_s - iT_b - \gamma_q]\cdot \\
g[(\xi+\mu_l)T_c]\cdot e^{j\pi(k-q)\cdot\sum_j n_j}
\end{array}\right\} & ; \quad k=0 \ , \ k \ even \\
\sum\limits_{\xi=I_1}^{I_2}\sum\limits_{n_0}\cdots\cdots\sum\limits_{n_l}\left\{\begin{array}{l}
(-1)^{n_0\cdot T_s}\cdot\prod\limits_{j=0}^{l}\left(h(n_j\cdot 2^j T_s)\right)\cdot \\
h_s[(2^l(\beta_l-\xi)-\sum\limits_j 2^j n_j)\cdot T_s - iT_b - \gamma_q]\cdot \\
g[(\xi+\mu_l)T_c]\cdot e^{j\pi(k-q)\cdot\sum_j n_j}
\end{array}\right\} & ; \quad k \ odd
\end{cases}
\tag{3-9}
$$

3.1.4 : Additive White Gaussian Noise (AWGN) Analysis

Referring to (3-7), the output WGN process from the k^{th} channel is :

$$z_k(lT_b)=\begin{cases}\displaystyle\sum_{\xi=I_1}^{I_2}(-1)^{\beta_l-\xi}\sum_{n_0}\cdots\cdots\sum_{n_l}\left\{\begin{array}{l}z[(2^l(\beta_l-\xi)-\sum_j 2^j n_j)\cdot T_s]\cdot\\ \prod_{j=0}^{l}\left(h(n_j\cdot 2^j T_s)\right)\cdot\\ g[(\xi+\mu_l)T_c]\cdot e^{j\pi(k+\frac{1}{2})\cdot\sum_j n_j}\end{array}\right\} & ;\quad k=0\ ,\ k\ even\\ \displaystyle\sum_{\xi=I_1}^{I_2}(-1)^{\beta_l-\xi}\sum_{n_0}\cdots\cdots\sum_{n_l}\left\{\begin{array}{l}z[(2^l(\beta_l-\xi)-\sum_j 2^j n_j)\cdot T_s]\cdot\\ (-1)^{n_0\cdot T_s}\cdot\prod_{j=0}^{l}\left(h(n_j\cdot 2^j T_s)\right)\cdot\\ g[(\xi+\mu_l)T_c]\cdot e^{j\pi(k+\frac{1}{2})\cdot\sum_j n_j}\end{array}\right\} & ;\quad k\ odd\end{cases}\tag{3-10}$$

Since $z(t)$ is a zero-mean Gaussian process with variance σ^2, $z_k(lT_b)$ is also zero-mean Gaussian with variance σ_o^2 given by (see Appendix 3.1.3 for derivation) [13] :

$$\begin{aligned}\sigma_o^2&=\sigma^2\cdot\sum_{\xi=I_1}^{I_2}\left|g[(\xi+\mu_l)T_c]\right|^2\cdot\sum_{n_0}\left|h(n_0T_s)\right|^2\cdot\sum_{n_1}\left|h(n_1\cdot 2T_s)\right|^2\cdot\cdots\cdots\sum_{n_l}\left|h(n_l\cdot 2^lT_s)\right|^2\\ &=\sigma^2\cdot\sum_{\xi=I_1}^{I_2}\left|g[(\xi+\mu_l)T_c]\right|^2\cdot\prod_{j=0}^{l}\left(\sum_{n_j}\left|h(n_j\cdot 2^jT_s)\right|^2\right)\end{aligned}\tag{3-11}$$

3.1.5 : Probability of Bit Error

In deriving the probability of bit error, we may solely consider the *in-phase* component of $r_k(lT_b)$. This is due to the assumption that the data symbols $a_{k,i}$ are assumed independent and equiprobable. This is a fair assumption in this analysis since the QPSK system we are dealing with is uncoded. It is also because the in-phase and quadrature components of $z_k(lT_b)$ are independent and identically distributed (i.i.d.) so the probability of bit error is symmetric about both components [10-11]. Thus, under the assumption that we have perfect phase synchronization on the desired k^{th} channel ($\phi_k=0$) and the desired symbol is the a^{th}, we obtain :

$$
\begin{aligned}
r_k^I(lT_b) = {} & A_k \cdot \begin{bmatrix} a_{k,\alpha} \cdot u(0) \\ +\sum_{i\neq\alpha} a_{k,i} \cdot u(\alpha T_b - iT_b) \end{bmatrix} \\
& + \sum_{q\neq k} A_q \cdot \sum_i \left\{ \begin{array}{l} a_{q,i} \cdot \begin{bmatrix} v^I(\alpha T_b - iT_b) \cdot \cos\phi_q \\ -v^Q(\alpha T_b - iT_b) \cdot \sin\phi_q \end{bmatrix} \\ -b_{q,i} \cdot \begin{bmatrix} v^I(\alpha T_b - iT_b) \cdot \sin\phi_q \\ +v^Q(\alpha T_b - iT_b) \cdot \cos\phi_q \end{bmatrix} \end{array} \right\} + z_k^I(lT_b)
\end{aligned}
\tag{3-12}
$$

where the in-phase and quadrature components of the ICI from the q^{th} channel, respectively, are:

$$
v^I(\alpha T_b - iT_b) = \left\{ \begin{array}{ll}
\sum_{\xi=I_1}^{I_2} \sum_{n_0} \cdots\cdots \sum_{n_l} \left\{ \begin{array}{l} \prod_{j=0}^{l} \left(h(n_j \cdot 2^j T_s) \right) \cdot \\ h_s[(2^l(\beta_l - \xi) - \sum_j 2^j n_j) \cdot T_s - iT_b - \gamma_q] \cdot \\ g[(\xi + \mu_l)T_c] \cdot \cos(\pi(k-q) \cdot \sum_j n_j) \end{array} \right\} & ; \quad k = 0 \text{ , } k \textit{ even} \\
\sum_{\xi=I_1}^{I_2} \sum_{n_0} \cdots\cdots \sum_{n_l} \left\{ \begin{array}{l} (-1)^{n_s \cdot T_s} \cdot \prod_{j=0}^{l} \left(h(n_j \cdot 2^j T_s) \right) \cdot \\ h_s[(2^l(\beta_l - \xi) - \sum_j 2^j n_j) \cdot T_s - iT_b - \gamma_q] \cdot \\ g[(\xi + \mu_l)T_c] \cdot \cos(\pi(k-q) \cdot \sum_j n_j) \end{array} \right\} & ; \quad k \textit{ odd}
\end{array} \right.
\tag{3-13}
$$

$$
v^Q(\alpha T_b - iT_b) = \left\{ \begin{array}{ll}
\sum_{\xi=I_1}^{I_2} \sum_{n_0} \cdots\cdots \sum_{n_l} \left\{ \begin{array}{l} \prod_{j=0}^{l} \left(h(n_j \cdot 2^j T_s) \right) \cdot \\ h_s[(2^l(\beta_l - \xi) - \sum_j 2^j n_j) \cdot T_s - iT_b - \gamma_q] \cdot \\ g[(\xi + \mu_l)T_c] \cdot \sin(\pi(k-q) \cdot \sum_j n_j) \end{array} \right\} & ; \quad k = 0 \text{ , } k \textit{ even} \\
\sum_{\xi=I_1}^{I_2} \sum_{n_0} \cdots\cdots \sum_{n_l} \left\{ \begin{array}{l} (-1)^{n_s \cdot T_s} \cdot \prod_{j=0}^{l} \left(h(n_j \cdot 2^j T_s) \right) \cdot \\ h_s[(2^l(\beta_l - \xi) - \sum_j 2^j n_j) \cdot T_s - iT_b - \gamma_q] \cdot \\ g[(\xi + \mu_l)T_c] \cdot \sin(\pi(k-q) \cdot \sum_j n_j) \end{array} \right\} & ; \quad k \textit{ odd}
\end{array} \right.
\tag{3-14}
$$

To be able to compute (3-13) and (3-14), we need to expand the cos(·) and sin(·) terms in those expressions, respectively. They may be expanded using simple trigonometric identities and simplified using the fact that the operations in the stages of the M-MCDD may be interchanged without loss of generality since we are concerned mainly with the final numerical result [6-7] [34]. It will be seen later on how (3-15) and (3-16) contribute to the effect of aliasing through each stage of demultiplexing. Thus, the expansion of these terms yields:

$$cos(\pi \cdot (k-q) \cdot \sum_{j=0}^{l} n_j) = \begin{cases} \sum_{i=0}^{(l/2)-1} \binom{l+1}{2i+1} \cdot (-1)^{\frac{l}{2}-i} \cdot \begin{Bmatrix} \prod_{k=0}^{2i} cos(\pi \cdot (k-q) \cdot n_k) \cdot \\ \prod_{m=2i+1}^{l} sin(\pi \cdot (k-q) \cdot n_m) \end{Bmatrix} & ; \quad l \text{ even} \\ + \prod_{k=0}^{l} cos(\pi \cdot (k-q) \cdot n_k) & \\ & \\ (-1)^{\frac{l+1}{2}} \cdot \prod_{k=0}^{l} sin(\pi \cdot (k-q) \cdot n_k) & \\ + \sum_{i=1}^{(l-1)/2} \binom{l+1}{2i} \cdot (-1)^{\frac{l+1}{2}-i} \cdot \begin{Bmatrix} \prod_{k=0}^{2i-1} cos(\pi \cdot (k-q) \cdot n_k) \cdot \\ \prod_{m=2i}^{l} sin(\pi \cdot (k-q) \cdot n_m) \end{Bmatrix} & ; \quad l \text{ odd} \\ + \prod_{k=0}^{l} cos(\pi \cdot (k-q) \cdot n_k) & \end{cases} \tag{3-15}$$

$$sin(\pi \cdot (k-q) \cdot \sum_{j=0}^{l} n_j) = \begin{cases} \sum_{i=0}^{(l/2)-1} \binom{l+1}{2i+1} \cdot (-1)^{i} \cdot \begin{Bmatrix} \prod_{k=0}^{2i} sin(\pi \cdot (k-q) \cdot n_k) \cdot \\ \prod_{m=2i+1}^{l} cos(\pi \cdot (k-q) \cdot n_m) \end{Bmatrix} & ; \quad l \text{ even} \\ + (-1)^{\frac{l}{2}} \cdot \prod_{k=0}^{l} sin(\pi \cdot (k-q) \cdot n_k) & \\ & \\ \sum_{i=0}^{(l-1)/2} \binom{l+1}{2i+1} \cdot (-1)^{i} \cdot \begin{Bmatrix} \prod_{k=0}^{2i} sin(\pi \cdot (k-q) \cdot n_k) \cdot \\ \prod_{m=2i+1}^{l} cos(\pi \cdot (k-q) \cdot n_m) \end{Bmatrix} & ; \quad l \text{ odd} \end{cases} \tag{3-16}$$

The in-phase component of $z_k(lT_b)$ follows from (3-10) in a similar manner and is :

$$z_k^I(lT_b) = \begin{cases} \sum_{\xi=I_1}^{I_2} (-1)^{\beta_l-\xi} \sum_{n_0} \ldots\ldots \sum_{n_l} \left\{ \begin{array}{l} z[(2^l(\beta_l-\xi)-\sum_j 2^j n_j)\cdot T_s]\cdot \\ \prod_{j=0}^{l}\left(h(n_j\cdot 2^j T_s)\right)\cdot \\ g[(\xi+\mu_l)T_c]\cdot\cos(\pi(k+\frac{1}{2})\cdot\sum_j n_j) \end{array} \right\} & ; \quad k=0 \ , \ k \ even \\ \sum_{\xi=I_1}^{I_2} (-1)^{\beta_l-\xi} \sum_{n_0} \ldots\ldots \sum_{n_l} \left\{ \begin{array}{l} z[(2^l(\beta_l-\xi)-\sum_j 2^j n_j)\cdot T_s]\cdot \\ (-1)^{n_0\cdot T_s}\cdot\prod_{j=0}^{l}\left(h(n_j\cdot 2^j T_s)\right)\cdot \\ g[(\xi+\mu_l)T_c]\cdot\cos(\pi(k+\frac{1}{2})\cdot\sum_j n_j) \end{array} \right\} & ; \quad k \ odd \end{cases} \tag{3-17}$$

From (3-12), we can denote the combined ISI from the k^{th} channel and the ICI by the interference variable η :

$$\eta = \sum_{i\neq\alpha} A_k \cdot a_{k,i}\cdot u(\alpha T_b - iT_b) + \sum_{q\neq k} A_q \cdot \sum_i \left\{ \begin{array}{l} a_{q,i}\cdot\begin{bmatrix} v^I(\alpha T_b - iT_b)\cdot\cos\phi_q \\ -v^Q(\alpha T_b - iT_b)\cdot\sin\phi_q \end{bmatrix} \\ -b_{q,i}\cdot\begin{bmatrix} v^I(\alpha T_b - iT_b)\cdot\sin\phi_q \\ +v^Q(\alpha T_b - iT_b)\cdot\cos\phi_q \end{bmatrix} \end{array} \right\} \tag{3-18}$$

Thus, the sufficient statistic becomes the in-phase component of $r_k(lT_b)$ and is compactly denoted as :

$$r_k^I(lT_b) = A_k \cdot a_{k,\alpha}\cdot u(0) + \eta + z_k^I(lT_b) \tag{3-19}$$

If we average the bit error rate over the desired data symbol $a_{k,\alpha}\in\{-1,+1\}$ bearing in mind that the p.d.f of η is even [10-11], we obtain the probability of bit error as (see Appendix 3.1.4 for derivation):

$$P_b = \int_{-\infty}^{+\infty} P_{b|\eta}\cdot f_\eta(\eta)\cdot d\eta = \int_{-\infty}^{+\infty}\left\{\int_{-T/2}^{T/2} Q\left(\frac{A_k u(0)-\eta}{\sigma_o}\right)\cdot f(\gamma_k)\cdot d\gamma_k\right\}\cdot f_\eta(\eta)\cdot d\eta \tag{3-20}$$

The expression given in (3-20) must be evaluated numerically in order to characterize the performance of the M-MCDD. The problem in finding its analytical solution is that the probability density function of the interference, $f_\eta(\eta)$, is unknown and virtually impossible to find. We thus rely on numerical approximation techniques to evaluate the integral given in (3-20) [2].

The numerical approximation techniques considered to evaluate the performance of the M-MCDD in the presence of ACI are the Gauss Quadrature Rule technique [10-12] [15] [17] and the Fourier Series Expansion technique of Norman Beaulieu [16]. Ultimately, one of the two techniques will be chosen to carry out the performance evaluation of the M-MCDD. Their candidacy will be based on the CPU time requirements. From both techniques, we will look at the overall BER results relative to ideal QPSK demodulation and their relative numerical accuracy. The Gauss Quadrature Rule technique is considered because its weights and abscissas can be obtained easily, thus generating BER curves generally requires little CPU time [2] [10-12] [15] [17]. On the other hand, the Fourier Series Expansion technique of Beaulieu is a candidate method because it uses the characteristic function of the interference. This, unlike the probability density function of the interference, can be obtained much easier meaning a lower level of CPU processing power is required to generate the BER curves [2] [16].

3.2 : The Gauss Quadrature Rule Technique

3.2.1 : Concept

The Gauss Quadrature Rule (GQR), proposed by Golub and Welsch [15], has been one technique adopted for the numerical approximation. For purposes of completeness, a step-by-step description of the GQR algorithm which was suggested by Benedetto *et al.* [10-12] is given below.

The approximation of P_b is based on evaluating the $Q(x)$ function at specified points and taking a linear combination of these values. If we consider the expression in (3-20) with a fixed timing offset and a closed interval $\eta \in [a,b]$ on which η is well-defined and non-zero, (3-20) may be approximated by [2] [10]:

$$\int_a^b Q\left(\frac{A_k u(0)-\eta}{\sigma_o}\right)\cdot f_\eta(\eta)\cdot d\eta \approx \sum_{i=1}^{M} w_i \cdot Q\left(\frac{A_k u(0)-x_i}{\sigma_o}\right) \tag{3-21}$$

The x_i are called the *abscissas* of the formula and the w_i are called the *weights*. Together, the set $\{w_i, x_i\}_{i=1}^{M}$ is called a *Gauss Quadrature Rule* (GQR) corresponding to the weight function $f_\eta(\eta)$, as such formulas were first studied by Gauss. The weight function $f_\eta(\eta)$ must satisfy several conditions for the approximation of (3-17) to be valid. The function $f_\eta(\eta)$ must be non-negative and integrable over $\eta \in [a,b]$ with [10]:

$$\int_a^b f_\eta(\eta) d\eta > 0 \tag{3-22}$$

and the integrals:

$$\int_a^b |\eta|^k f_\eta(\eta)\,d\eta \qquad (3\text{-}23)$$

must be definite[1] and finite.

The central problem is to find the weights and abscissas of the quadrature rule. The algorithm of Benedetto et al. [10-12] due to Golub and Welsch [15] computes the set $\{w_i, x_i\}_{i=1}^{M}$ using the first $2M+1$ moments of the interference. The algorithm proceeds as follows:

1. The unknown weight function, $f_\eta(\eta)$, can be approximated by a set of polynomials $z_0(x)$, $z_1(x)$, ... that are orthonormal with respect to $f_\eta(\eta)$. These polynomials satisfy a *three-term recursive relationship* [15], that is:

   $$\eta\cdot z_{j-1}(\eta) = \beta_{j-1}\cdot z_{j-2}(\eta) + \alpha_j\cdot z_{j-1}(\eta) + \beta_j\cdot z_j(\eta) \qquad (3\text{-}24)$$

   It is possible to find the coefficients α_j and β_j for the recurrence relation by knowing only the first $2M+1$ moments of the interference.

2. First, form an *(M+1)* by *(M+1) Hankel matrix* of the moments, denoted by **X** :

   $$X = \{x_{ij}\}_{i,j=1}^{M+1} \quad , \quad x_{ij} = M_{i+j-2} \qquad (3\text{-}25)$$

   where:

   $$M_k = \int_a^b \eta^k \cdot f_\eta(\eta)\cdot d\eta \qquad (3\text{-}26)$$

   is the k^{th} moment of the interference for $0 \le k \le 2M$.

3. Next, we perform the *Cholesky decomposition*, that is, $\mathbf{X}=\mathbf{Y}^T\mathbf{Y}$ of the Hankel matrix to obtain an upper triangular matrix **Y** of the form:

   $$Y = \{y_{ij}\}_{i,j=1}^{M+1} \quad , \quad y_{ii} = \sqrt{x_{ii} - \sum_{k=1}^{i-1} y_{ki}^2} \quad , \quad y_{ij} = \frac{x_{ij} - \sum_{k=1}^{i-1} y_{ki}\cdot y_{kj}}{y_{ii}}\ ;\ (i<j) \qquad (3\text{-}27)$$

4. Once the upper triangular matrix **Y** is found, the coefficients α_j and β_j of the three-term recursive relationship are given by:

[1] *Definite* in this sense means positive, non-zero and well-defined in the interval $\eta \in [a,b]$ [19]

$$\alpha_j = \frac{y_{j,j+1}}{y_{j,j}} - \frac{y_{j-1,j}}{y_{j-1,j-1}} \quad , \quad j = 1,2,.....,M \tag{3-28}$$

$$\beta_j = \frac{y_{j+1,j+1}}{y_{j,j}} \quad , \quad j = 1,2,.....,M-1 \tag{3-29}$$

with $y_{0,0}$ $=1$ and $y_{0,1}$ $=0$.

5. The weights and abscissas of the quadrature rule are then found from the eigenvalues and eigenvectors of the symmetric tridiagonal matrix $\mathbf{Z}$ of the form :

$$\mathbf{Z} = \begin{bmatrix} \alpha_1 & \beta_1 & 0 & \ldots\ldots\ldots\ldots\ldots\ldots & & & 0 \\ \beta_1 & \alpha_2 & \beta_2 & 0 & \ldots\ldots\ldots\ldots & & 0 \\ 0 & \beta_2 & \alpha_3 & \beta_3 & 0 & \ldots\ldots & 0 \\ 0 & 0 & \beta_3 & \alpha_4 & \beta_4 & \ldots\ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & \vdots & \vdots & \beta_{m-2} & \alpha_{m-1} & \beta_{m-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \beta_{m-1} & \alpha_m \end{bmatrix} \tag{3-30}$$

6. If we let $u_{j,1}$ denote the first component of the j^{th} orthonormalized eigenvector of $\mathbf{Z}$ and let λ_j denote the corresponding j^{th} eigenvalue, then:

$$\begin{aligned} w_j &= u_{j,1}^2 \cdot M_0 \\ x_i &= \lambda_j \end{aligned} \tag{3-31}$$

where M_0 is the mean or zeroth moment of the interference.

3.2.2 : Evaluation of the Moments of the Interference

The moments of the interference can be computed in a recursive manner using a method due to Prabhu [29]. Consider that the interference variable η is the sum of K separate interference terms,

$$\eta = \sum_{i=1}^{K} \eta_i \tag{3-32}$$

and let γ_n denote the partial sum of the first n terms,

$$\gamma_n = \sum_{i=1}^{n} \eta_i \tag{3-33}$$

It is then possible to show that the following recursion holds [29]:

$$E[\gamma_n^k] = \sum_{i=0}^{k} \binom{k}{i} E[\gamma_{n-1}^i] \cdot E[\eta_n^{k-i}] \tag{3-34}$$

where E[·] denotes expectation. The k^{th} moment is then given by :

$$M_k = E[\gamma_K^k] = \sum_{i=0}^{k} \binom{k}{i} E[\gamma_{K-1}^i] \cdot E[\eta_K^{k-i}] \tag{3-35}$$

In evaluating the moments of the interference M_k , we must evaluate the moments of the individual terms of the interference. There are two sets of terms involved in the computation of the individual moments; one set for the intersymbol interference (ISI), $E[\eta^{\rho}_{1,i}]$, and one for the interchannel interference (ICI) on the q^{th} channel, $E[\eta^{\rho}_{2,q,i}]$ [10-11] [16]. Since both involve an infinite number of interfering symbols, they will have to be truncated in order to obtain an expression which is computationally tractable [2]. Let $R = m_l + m_u$ be the number of ISI terms which contribute significantly to the interference. Also, let $R_j = m_{l,j} + m_{u,j} + 1$ be the corresponding number of significantly interfering ICI terms on the j^{th} channel. Here, $m_{l,j}$ and $m_{u,j}$ represent the number of interfering symbols below and above the α^{th} desired symbol on the j^{th} channel, respectively. Likewise, m_l and m_u are the number of interfering symbols on the desired channel. The interference variable η of (3-18) is then approximated by :

$$\begin{aligned}\eta' &= \sum_{\substack{i=\alpha-m_l \\ i\neq\alpha}}^{\alpha+m_u} A_k \cdot a_{k,i} \cdot u(\alpha T_b - iT_b) + \sum_{q\neq k} A_q \cdot \sum_{i=\alpha-m_{l,q}}^{\alpha+m_{u,q}} \left\{ \begin{matrix} a_{q,i} \cdot \begin{bmatrix} v^I(\alpha T_b - iT_b)\cdot\cos\phi_q \\ -v^Q(\alpha T_b - iT_b)\cdot\sin\phi_q \end{bmatrix} \\ -b_{q,i} \cdot \begin{bmatrix} v^I(\alpha T_b - iT_b)\cdot\sin\phi_q \\ +v^Q(\alpha T_b - iT_b)\cdot\cos\phi_q \end{bmatrix} \end{matrix} \right\} \\ &= \eta_1 + \eta_2 \end{aligned} \tag{3-36}$$

and the total number of interference samples is given by:

$$K = R + \sum_{j\neq k} R_j \tag{3-37}$$

The first summation term in η' represents the ISI and may be denoted by η_1 . Similarly, the second term represents the ICI and may be denoted as η_2 . If we examine the individual ISI and ICI terms of η' (ie. for fixed i), we can derive the expression for the moments of the individual interference terms. It is possible to evaluate the moments of the ISI and ICI individually since the data symbols $a_{q,i}$ and $b_{q,i}$ are i.i.d [10-11]. Thus, it can be seen straightforwardly from (3-36) that the individual moments for the ISI are (see Appendix 3.2 for derivations):

$$E[\eta_{1,i}^{\rho}] = \tfrac{1}{2} \cdot A_k^{\rho} \sum_{a_{k,i}} \int_{-T/2}^{T/2} \left(a_{k,i} \cdot u(\alpha T_b - iT_b) \right)^{\rho} f(\gamma_k) \cdot d\gamma_k \tag{3-38}$$

and the individual moments for the ICI on the q^{th} channel are (see Appendix 3.2 for derivations):

$$E[\eta_{2,q,i}^{\rho}] = \tfrac{1}{8\pi} \cdot A_q^{\rho} \sum_{a_{q,i}, b_{q,i}} \int_{\substack{0 \\ \phi_q}}^{2\pi} \int_{\substack{-T/2 \\ \gamma_q}}^{T/2} \left\{ \begin{array}{l} a_{q,i} \cdot \begin{bmatrix} v^I(\alpha T_b - iT_b) \cdot \cos\phi_q \\ -v^Q(\alpha T_b - iT_b) \cdot \sin\phi_q \end{bmatrix} \\ -b_{q,i} \cdot \begin{bmatrix} v^I(\alpha T_b - iT_b) \cdot \sin\phi_q \\ +v^Q(\alpha T_b - iT_b) \cdot \cos\phi_q \end{bmatrix} \end{array} \right\}^{\rho} \cdot f(\gamma_q) \cdot d\gamma_q \cdot d\phi_q \tag{3-39}$$

In both cases, since the data symbols $a_{k,i}$ are equiprobable and chosen from the set $\{-1, +1\}$, it can be seen straightforwardly that the odd order moments will be zero [2] [10-11]. Since only the even order moments remain from (3-38) and (3-39), we can simplify them to obtain for the individual ISI moments (see Appendix 3.2 for derivation) :

$$E[\eta_{1,i}^{2\rho}] = A_k^{2\rho} \int_{-T/2}^{T/2} \left(u(\alpha T_b - iT_b) \right)^{2\rho} f(\gamma_k) \cdot d\gamma_k \tag{3-40}$$

and for the individual ICI moments (see Appendix 3.2 for derivation) :

$$E[\eta_{2,q,i}^{2\rho}] = \tfrac{1}{2\pi} \cdot A_q^{2\rho} \int_{\substack{0 \\ \phi_q}}^{2\pi} \int_{\substack{-T/2 \\ \gamma_q}}^{T/2} \left\{ \begin{array}{l} v^I(\alpha T_b - iT_b) \cdot \left(\cos\phi_q + \sin\phi_q \right) \\ +v^Q(\alpha T_b - iT_b) \cdot \left(\cos\phi_q - \sin\phi_q \right) \end{array} \right\}^{2\rho} \cdot f(\gamma_q) \cdot d\gamma_q \cdot d\phi_q \tag{3-41}$$

, respectively.

3.2.3 : Evaluation of the Truncation Errors from the GQR Technique

There are two types of truncations errors that occur in using this method to evaluate the probability of bit error for the M-MCDD : one is due to using the first *2M+1* moments and the second is due to the truncation of the infinite series of interfering symbols in η [2].

It can be shown that the truncation error due to using *2M+1* moments to evaluate P_b is [10, thm. 1] [30, pp. 782, 787, 934] (see Appendix 3.3 for derivation) :

$$|R_M| < C \cdot \int_{-T/2}^{T/2} \exp\left(\frac{-(A_k u(0) - \xi)^2}{4\sigma_o^2}\right) \cdot f(\gamma_k) \cdot d\gamma_k \qquad (3\text{-}42)$$

, where ξ is a fixed value in the range of the interference variable η and:

$$C = \frac{B\sqrt{(2M+1)!} \cdot \prod_{i=1}^{M} \beta_i^2}{\sqrt{2\pi} \cdot (2M)! \cdot \sigma_o^{2M}} \qquad (3\text{-}43)$$

and $B \approx 1.086435$ [10]. To maximize the bound of (3-42), we must maximize ξ, and this can be shown to occur when (see Appendix 3.3 for derivation) :

$$\begin{aligned}\xi_{\eta,MAX} &= \max_{\gamma_k} \sum_{\substack{i=\alpha-m_l \\ i\neq\alpha}}^{\alpha+m_u} \left|A_k \cdot u(\alpha T_b - iT_b)\right| \\ &\quad + \sqrt{2}\sum_{q\neq k} A_q \max_{\gamma_q} \sum_{i=\alpha-m_{l,q}}^{\alpha+m_{u,q}} \left|v^I(\alpha T_b - iT_b)\right| + \left|v^Q(\alpha T_b - iT_b)\right|\end{aligned} \qquad (3\text{-}44)$$

It is also useful to define the minimum value of ξ for purposes which will be clarified later on. Thus, the minimum value of ξ can be expressed as (see Appendix 3.3 for derivation):

$$\begin{aligned}\xi_{\eta,MIN} &= -\min_{\gamma_k} \sum_{\substack{i=\alpha-m_l \\ i\neq\alpha}}^{\alpha+m_u} \left|A_k \cdot u(\alpha T_b - iT_b)\right| \\ &\quad - \sqrt{2}\sum_{q\neq k} A_q \min_{\gamma_q} \sum_{i=\alpha-m_{l,q}}^{\alpha+m_{u,q}} \left|v^I(\alpha T_b - iT_b)\right| + \left|v^Q(\alpha T_b - iT_b)\right|\end{aligned} \qquad (3\text{-}45)$$

The second type of truncation error is due to considering a limited number of interfering symbols on the desired channel and on the interfering channels. This truncation error is more difficult to classify however, it has been shown how the effect of this truncation error on the probability of bit error can be bounded within a certain range [11]. First, we define the collection of interfering symbols *not* considered by the truncated series of (3-36) as:

$$\eta'' = \sum_{\substack{i<\alpha-m_l \\ i>\alpha+m_u}} A_k \cdot a_{k,i} \cdot u(\alpha T_b - iT_b) + \sum_{q\neq k} A_q \cdot \sum_{\substack{i<\alpha-m_{l,q} \\ i>\alpha+m_{u,q}}} \left\{\begin{aligned} a_{q,i} &\cdot \begin{bmatrix} v^I(\alpha T_b - iT_b) \cdot \cos\phi_q \\ -v^Q(\alpha T_b - iT_b) \cdot \sin\phi_q \end{bmatrix} \\ -b_{q,i} &\cdot \begin{bmatrix} v^I(\alpha T_b - iT_b) \cdot \sin\phi_q \\ +v^Q(\alpha T_b - iT_b) \cdot \cos\phi_q \end{bmatrix} \end{aligned}\right\} \qquad (3\text{-}46)$$

such that $\eta = \eta' + \eta''$. For a fixed timing phase, $\gamma_{k,}$ it has been shown that the true probability of bit error lies in the interval [11, Eq. 49] :

$$\begin{aligned}\int_a^b Q\left(\frac{A_k u(0)-\eta'}{\sigma_o}\right)\cdot f_{\eta'}(\eta')\cdot d\eta' &\leq \int_a^b Q\left(\frac{A_k u(0)-\eta}{\sigma_o}\right)\cdot f_{\eta}(\eta)\cdot d\eta \\ &\leq \left(1-\frac{\sigma_r^2}{\sigma_0^2}\right)^{-1/2} \int_a^b Q\left(\frac{A_k u(0)-\eta'}{\sigma_o\cdot\left(1-\frac{\sigma_r^2}{\sigma_0^2}\right)^{-1/2}}\right)\cdot f_{\eta'}(\eta')\cdot d\eta'\end{aligned} \tag{3-47}$$

, where $\sigma_r^{\,2}$ is a quantity such that [11] :

$$E\{\exp(A\eta'')\}\leq \exp\left(\frac{A^2}{2}\cdot\sigma_r^2\right) \quad , \quad \text{for } \forall \text{ A} \tag{3-48}$$

It should be noted that (3-47) is valid as long as the eye pattern remains open or the peak distortion is less than one [11]. Thus, it is possible to show that the value of $\sigma_r^{\,2}$ can be obtained and upper bounded by (see Appendix 3.3 for derivation):

$$\begin{aligned}\sigma_r^2 &= \max_{\gamma_k} A_k^2 \sum_{\substack{i<\alpha-m_l \\ i>\alpha+m_u}} \left|u(\alpha T_b - iT_b)\right|^2 \\ &\quad + \max_{\gamma_q\ \phi_q} \sum_{q\neq k} A_q^2 \sum_{\substack{i<\alpha-m_{l,q} \\ i>\alpha+m_{u,q}}} \left|\begin{matrix} v^I(\alpha T_b - iT_b)\cdot(\cos\phi_q + \sin\phi_q) \\ +v^Q(\alpha T_b - iT_b)\cdot(\cos\phi_q - \sin\phi_q)\end{matrix}\right|^2 \\ &\leq \max_{\gamma_k} A_k^2 \sum_{\substack{i<\alpha-m_l \\ i>\alpha+m_u}} \left|u(\alpha T_b - iT_b)\right|^2 \\ &\quad + \sum_{q\neq k} A_q^2\cdot \max_{\gamma_q} \sum_{\substack{i<\alpha-m_{l,q} \\ i>\alpha+m_{u,q}}} \left|v^I(\alpha T_b - iT_b)\right|^2 + \left|+v^Q(\alpha T_b - iT_b)\right|^2\end{aligned} \tag{3-49}$$

If we choose the number of interfering samples, *K*, in (3-37) to be large enough such that σ_r remains small, the upper and lower bounds on the probability of bit error stated by (3-47) will converge. It is known that if $\sigma_r^2 / \sigma_0^2 \approx 10^{-2}$, the separation between the upper and lower bounds will be to the order of 0.05dB [11].

3.3 : Beaulieu's Fourier Series Expansion Technique

3.3.1 : Concept

In Beaulieu's Fourier Series Expansion (FSE) Technique, the complementary distribution function of the noise is approximated by a Fourier Series of the form [2]:

$$Q(x) = \sum_{m=-\infty}^{+\infty} c_m e^{jm\omega x} + \varepsilon(x) \tag{3-50}$$

, where $\varepsilon(x)$ represents the error in using the FSE to approximate $Q(x)$. If (3-50) is substituted into the probability of bit error expression of (3-20), assuming that the noise has a variance of one, that is $\sigma_0{}^2 = 1$, we can rewrite the bit error rate expression as [31]:

$$\begin{aligned} P_b &= \int_{-\infty}^{+\infty}\left\{\int_{-T/2}^{T/2}\left(\sum_{m=-\infty}^{+\infty} c_m e^{jm\omega(A_k u(0)-\eta)} + \varepsilon(A_k u(0)-\eta)\right)\cdot f(\gamma_k)\cdot d\gamma_k\right\}\cdot f_\eta(\eta)\cdot d\eta \\ &= \sum_{m=-\infty}^{+\infty} c_m \cdot\left\{\int_{-T/2}^{T/2} e^{jm\omega A_k u(0)}\cdot f(\gamma_k)\cdot d\gamma_k\right\}\cdot\int_{-\infty}^{+\infty} e^{-jm\omega\eta}\cdot f_\eta(\eta)\cdot d\eta \\ &\quad + \int_{-\infty}^{+\infty}\left\{\int_{-T/2}^{T/2}\varepsilon(A_k u(0)-\eta) f(\gamma_k) d\gamma_k\right\}\cdot f_\eta(\eta) d\eta \end{aligned} \tag{3-51}$$

The characteristic function of the interference variable, η, is denoted by $\Phi_\eta(\omega)$ and is defined as [31]:

$$\Phi_\eta(\omega) = E\left\{e^{j\omega\eta}\right\} = \int_{-\infty}^{+\infty} e^{j\omega\eta}\cdot f_\eta(\eta) d\eta \tag{3-52}$$

By substituting (3-52) into (3-51), the probability of bit error can consequently be expressed as :

$$\begin{aligned} P_b &= \sum_{m=-\infty}^{+\infty} c_m \cdot\left\{\int_{-T/2}^{T/2} e^{jm\omega A_k u(0)}\cdot f(\gamma_k)\cdot d\gamma_k\right\}\cdot\Phi_\eta(-m\omega) \\ &\quad + \int_{-\infty}^{+\infty}\left\{\int_{-T/2}^{T/2}\varepsilon(A_k u(0)-\eta) f(\gamma_k) d\gamma_k\right\}\cdot f_\eta(\eta) d\eta \end{aligned} \tag{3-53}$$

To have (3-53) become computationally tractable, we must limit the range of the interference and truncate the infinite Fourier Series. If the ISI is finite, then η is bounded, that is $a \le \eta \le b$ [10-11]. Also, we can truncate the Fourier Series to $M+1$ terms, thus we write [16]:

$$\begin{aligned} P_b &= \sum_{m=0}^{M} c_m \cdot\left\{\int_{-T/2}^{T/2} e^{jm\omega A_k u(0)}\cdot f(\gamma_k)\cdot d\gamma_k\right\}\cdot\Phi_\eta(-m\omega) \\ &\quad + \int_{a}^{b}\left\{\int_{-T/2}^{T/2}\varepsilon(A_k u(0)-\eta) f(\gamma_k) d\gamma_k\right\}\cdot f_\eta(\eta) d\eta \end{aligned} \tag{3-54}$$

In Beaulieu's FSE technique, the coefficients c_m in the Fourier Series representation of $Q(x)$ have been determined such that the error term $\varepsilon(x)$ is bounded. In this method, the Fourier Series is combined with a Chernoff Bound and is summarized by Figure 3-1 and the following inequality [16]:

$$E\{P(z-z_0)\} > Q(z_0) - Q(z_0+\tfrac{T}{2}) \quad \Rightarrow \quad Q(z_0) = E\{P(z-z_0)\} + \alpha \tag{3-55}$$

, where :

$$\alpha < Q(\tfrac{T}{2} - |z_0|) \tag{3-56}$$

, where $z(\cdot)$ is the noise process having zero mean and unit variance, $Q(\cdot)$ is the complementary distribution function of the noise and $P(\cdot)$ is a square wave of period T which is used to gate the probability density function of $z(\cdot)$, $f_z(z)$. $P(\cdot)$ and its Fourier Series representation is given as [16]:

$$P(z) = \begin{cases} 1 & ; \quad lT < z < \frac{(2l+1)T}{2} \\ 0 & ; \quad \frac{(2l+1)T}{2} < z < (l+1)T \\ \frac{1}{2} & ; \quad z = \frac{lT}{2} \end{cases} \quad \Rightarrow \quad P(z) = \frac{1}{2} + \frac{2}{\pi} \sum_{\substack{m=1 \\ m \text{ odd}}}^{+\infty} \frac{\sin(2m\pi x/T)}{m} \tag{3-57}$$

for $l=0, \pm 1, \pm 2, \pm 3, \ldots$

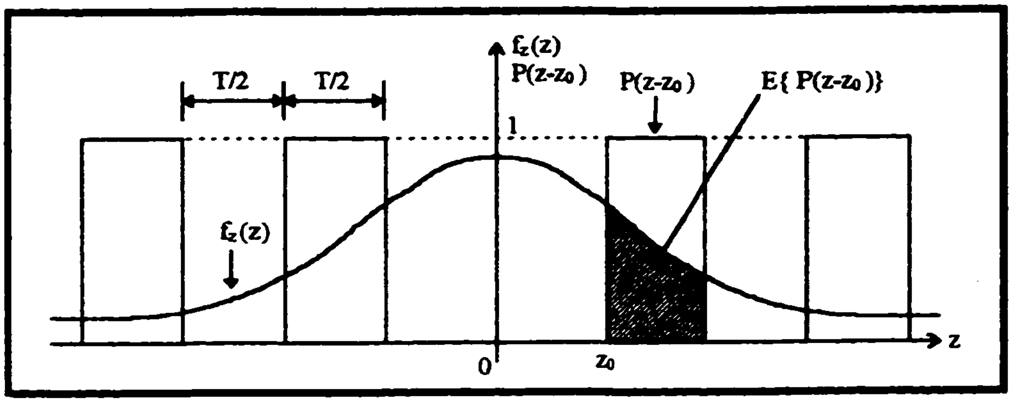


Figure 3-1. The function, $P(z)$, used to gate the p.d.f. of the noise, $f_z(z)$ [16].

By combining (3-55) through (3-57), it has been shown that the Fourier Series approximation to $Q(x)$ is given by [16]:

$$Q(x) = \tfrac{1}{2} - \tfrac{2}{\pi} \sum_{\substack{m=1 \\ m \text{ odd}}}^{+\infty} \frac{e^{-m^2\omega^2/2}}{m} \cdot \sin(m\omega x) + \varepsilon(x) \tag{3-58}$$

, where:

$$|\varepsilon(x)| < Q(\tfrac{T}{2} - |x|) \tag{3-59}$$

$$E\{P(z-z_0)\} > Q(z_0) - Q(z_0 + \tfrac{T}{2}) \;\Rightarrow\; Q(z_0) = E\{P(z-z_0)\} + \alpha \qquad (3\text{-}55)$$

, where :

$$\alpha < Q(\tfrac{T}{2} - |z_0|) \qquad (3\text{-}56)$$

, where *z(·)* is the noise process having zero mean and unit variance, *Q(·)* is the complementary distribution function of the noise and *P(·)* is a square wave of period *T* which is used to gate the probability density function of *z(·)*, f_z *(z)*. *P(·)* and its Fourier Series representation is given as [16]:

$$P(z) = \begin{cases} 1 & ; \; lT < z < \frac{(2l+1)T}{2} \\ 0 & ; \; \frac{(2l+1)T}{2} < z < (l+1)T \\ \frac{1}{2} & ; \; z = \frac{lT}{2} \end{cases} \;\Rightarrow\; P(z) = \frac{1}{2} + \frac{2}{\pi} \sum_{\substack{m=1 \\ m\ \mathrm{odd}}}^{+\infty} \frac{\sin(2m\pi x/T)}{m} \qquad (3\text{-}57)$$

for *l=0, ±1, ±2, ±3, ...*

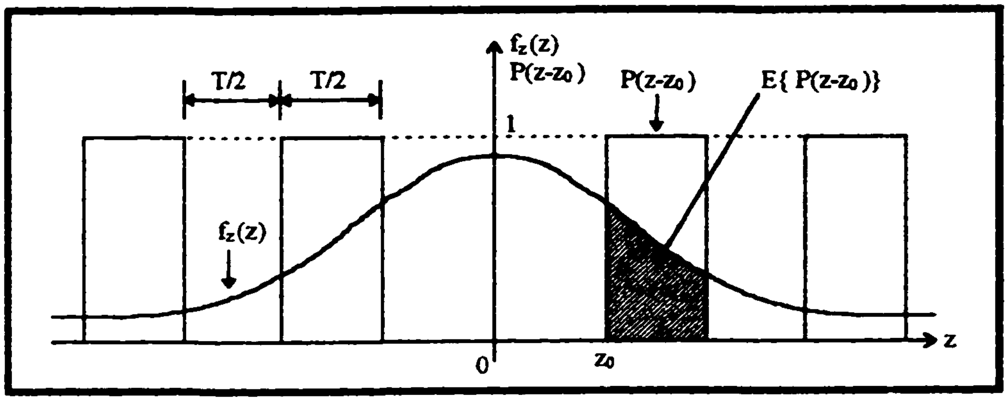


Figure 3-1. The function, *P(z)*, used to gate the p.d.f. of the noise, f_z *(z)* [16].

By combining (3-55) through (3-57), it has been shown that the Fourier Series approximation to *Q(x)* is given by [16]:

$$Q(x) = \tfrac{1}{2} - \tfrac{2}{\pi} \sum_{\substack{m=1 \\ m\ \mathrm{odd}}}^{+\infty} \frac{e^{-m^2\omega^2/2}}{m} \cdot \sin(m\omega x) + \varepsilon(x) \qquad (3\text{-}58)$$

, where:

$$|\varepsilon(x)| < Q(\tfrac{T}{2} - |x|) \qquad (3\text{-}59)$$

, and the normalized frequency is $\omega = 2\pi / T$. Also, (3-58) will be truncated so as to include the first M terms of the series. The advantage to this approach is that while $f_\eta(\eta)$ is very hard to find, the characteristic function, $\Phi_\eta(\omega)$, may be easily obtained [16].

3.3.2 : Final Bit Error Rate Expression

If we can assume with no loss of generality that the probability density functions $f(\gamma_k)$ and $f(\eta)$ are evenly distributed, we can rearrange the index of summation for the probability of bit error from (3-54), so we have [16]:

$$P_b = c_0 + \sum_{m=1}^{M} \int_{-T/2}^{T/2} \left\{ \begin{matrix} c_m \cdot e^{jm\omega A_k u(0)} \cdot \Phi_\eta(-m\omega) \\ +c_{-m} \cdot e^{-jm\omega A_k u(0)} \cdot \Phi_\eta(m\omega) \end{matrix} \right\} \cdot f(\gamma_k) \cdot d\gamma_k + \int_a^b \left\{ \int_{-T/2}^{T/2} \varepsilon(A_k u(0) - \eta) f(\gamma_k) d\gamma_k \right\} \cdot f_\eta(\eta) d\eta \qquad (3\text{-}60)$$

Furthermore, if $f(\eta)$ is even, $\Phi_\eta(-m\omega) = \Phi_\eta(m\omega)$ and by substituting the coefficients for the Fourier Series of Q(x) from (3-58) and the error upper bound of (3-59) into (3-60), we obtain the final probability of bit error expression as [16]:

$$P_b = \tfrac{1}{2} - \tfrac{2}{\pi} \sum_{\substack{m=1 \\ m\ \text{odd}}}^{M} \frac{e^{-m^2\omega^2/2}}{m} \cdot \left\{ \int_{-T/2}^{T/2} \sin(m\omega A_k u(0)) \cdot f(\gamma_k) \cdot d\gamma_k \right\} \cdot \Phi_\eta(-m\omega) + R_M + \beta + \Delta \qquad (3\text{-}61)$$

, where R_M, β and Δ all represent truncation errors which will be discussed in detail in the next section. The magnitude of the error in Beaulieu's FSE technique is thus bounded by [2]:

$$|\text{error}| < |R_M| + |\beta| + |\Delta| \qquad (3\text{-}62)$$

3.3.3 : Evaluation of the Truncation Errors From Beaulieu's Method

We first define the truncation error β which represents the error associated with using the Fourier Series approximation for $Q(x)$ [2]. To obtain β in (3-61), we substitute (3-59) into (3-60) and once simplified, the expression for β is upper bounded by [16]:

$$|\beta| < \int_{-T/2}^{T/2} Q\left(\tfrac{T}{2} - A_k u(0) - \xi_{\max}\right) \cdot f(\gamma_k) d\gamma_k \qquad (3\text{-}63)$$

In (3-63), the upper bound has been maximized by appropriately choosing the maximum value of the interference, $\xi_{\eta_{max}}$, given by (3-44) [11].

The second truncation error, R_M , is the error associated with using M terms in the summation for the Fourier Series as opposed to an infinite series [2]. The resulting upper bound for R_M is (see Appendix 3.4 for derivation) :

$$R_M < \frac{\sqrt{2\pi} \cdot T}{\pi^2 \cdot M} \cdot Q\left(\frac{2\pi M}{T}\right) \tag{3-64}$$

The third and final truncation error, Δ , is due to using a finite number of interference terms, K , defined by (3-37), in the characteristic function for the interference [2]. To derive the form of Δ , we need to know the truncated characteristic function based on using a finite number of interfering symbols and then subtract it from the ideal characteristic function of (3-52). The upper bound on Δ is thus (see Appendix 3.4 for derivation) :

$$\begin{aligned} |\Delta| \leq \frac{\omega^2}{\pi} \cdot & \left\{ \sum_{\substack{i=\alpha-m_l \\ i\neq\alpha}}^{\alpha+m_u} A_k^2 \cdot u^2(\alpha T_b - iT_b) + \sum_{q\neq k} A_q^2 \sum_{i=\alpha-m_{l,q}}^{\alpha+m_{u,q}} \left|v^I(\alpha T_b - iT_b)\right|^2 + \left|v^Q(\alpha T_b - iT_b)\right|^2 \right\} \\ & \cdot \sum_{\substack{m=1 \\ m\ odd}}^{M} m \cdot e^{-m^2\omega^2/2} \cdot \sin(m\omega\, A_k u(0)) \cdot \Phi_\eta^*(-m\omega) \end{aligned} \tag{3-65}$$

Although the truncation error of (3-65), by definition, is identical to the truncation error of (3-47) from the GQR technique, they in fact will be different in value [2]. This is because the truncation error of (3-65) involves the characteristic function of the interference while that of (3-47) is derived from the interference variable, η'' , of (3-46) [2] [11].

3.4 : Numerical Results for the Effects of Adjacent Channel Interference

3.4.1 : Discussion of Numerical Algorithms

3.4.1.1 : The Gauss Quadrature Rule (GQR) Algorithm

The computational algorithm which uses the Gauss Quadrature Rule (GQR) technique to evaluate the M-MCDD is summarized by Figure 3-7. In that flow diagram, we see several major blocks which constitute the general part of the algorithm implemented in MATLAB [32]. Due to spacial constraints, Figure 3-7 is split into two parts. Figure 3-7(a) shows the first three blocks of the algorithm while the subsequent portion is shown by Figure 3-7(b).

The first major part is a preparation stage needed to define the coefficients and arrays involved and to allow the user to give some input to some important parameters. At first, the user is prompted to enter some values. The values which are user-defined are *Nterms* (number of terms in the GQR), *Ltap* (number of non-zero filter taps per halfband filter), *Itap* (number of filter taps in the channel detection filter), *mL*, *mU* (number of interfering symbols below and above the desired symbol, respectively), *l* (the number of stages for the M-MCDD) and *k* (the position of the desired channel). These parameters are constrained to a certain range according to the limitations of the program, which will be elaborated later on in the discussion. Within this stage of the program, coefficients and arrays are also defined. Some of the important coefficients defined are f_s (normalized frequency spacing between the carriers) set at 1.5 times the symbol rate of the signals ($f_c = 1.5 \cdot f_b$) and *beta* (rolloff factor for the root raised cosine response used for the transmit and channel detection filters) set at 0.5. Although f_s and *beta* were not varied in the analysis, from previous work, it is expected that if we increase the channel spacing and lower the rolloff factor, the resulting performance of the M-MCDD would improve [2]. That is, for a given filter length, the amount of signal-to-noise ratio (SNR) needed to achieve a specified performance level would decrease. Also, to simplify the algorithm, *base_offset* (the base timing offset) was set to 1/3. With this set value, the signals are all assumed to be time synchronized with the group demodulator. With perfect timing and a channel spacing of $f_s = 1.5$, there are only two possibilities for the timing parameters β_l and μ_l , shown in (3-4) and (3-5), respectively. These possibilities are [2] :

$$\beta_l \cdot T_c = l \cdot T_b \quad and \quad \mu_l \cdot T_c = 0 \tag{3-66}$$

or

$$\beta_l \cdot T_c = l \cdot T_b - \frac{T_b}{3} \quad and \quad \mu_l \cdot T_c = \frac{T_b}{3} \tag{3-67}$$

Normally, the probability of bit error is computed for each state and then an average is taken but because in initial trials it was seen that the probability of bit error was unchanged from one state to the other, only the latter state was used in the computation thus also saving computation time [2]. This accounts for the base timing offset of 1/3. We also define the size of the arrays for the filter coefficients, the moments and the magnitude of the interfering symbols.

In the next block, three sets of filter coefficients are computed. These filter coefficients are for the transmit filter, halfband filter, and channel detection filter. The transmit filter and channel detection filter both have a root raised cosine spectrum [2]. Ideally, the cascade of these filters would produce a full raised cosine spectral response [23-24]. The root raised cosine filters are first specified in the frequency domain and then an inverse fast Fourier Transform (FFT) is used to obtain the time domain impulse response coefficients. A large inverse FFT of 8192 frequency samples was used in order to better approximate the true impulse response [2] [32]. It is then truncated to the desired number of filter taps or transmit filter coefficients, as the case may be. For simplicity, the coefficients have also been normalized to have unit energy so that $\sigma_o^2 = \sigma_i^2$ in (3-11) [2]. In the case of the channel detection filter, a hamming window is applied to reduce the size of the sidelobes [9] [25]. By doing this, we effectively reduce the amount of ICI. Once this windowing is applied, it is necessary to rescale these channel detection filter coefficients to have unit energy, as is indicated within the second block in Figure 3-7(a) [2]. Figures 3-2 and 3-3 demonstrate the generation of the channel detection filter coefficients for the cases *Itap=12* and *Itap=15*, respectively. Also shown in these figures is the corresponding frequency response of the filters [9]. Likewise, Figure 3-4 shows the transmit filter spectrum for the case *Ltap=9* and *Itap=15*.

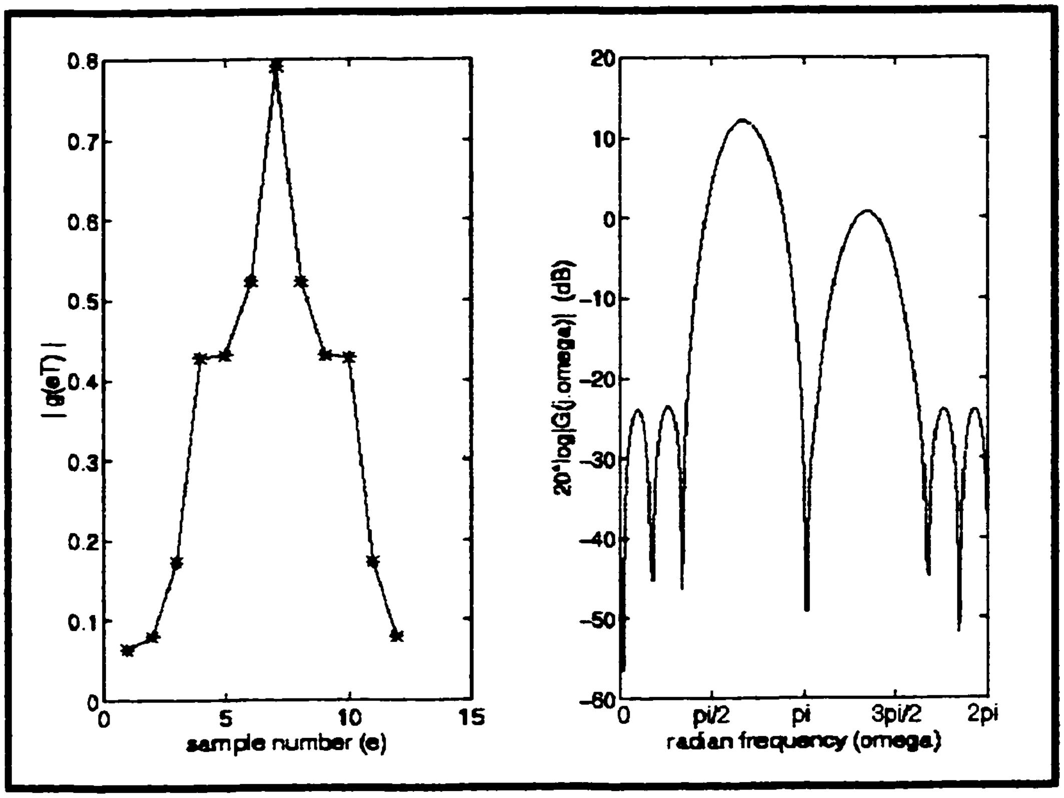


Figure 3-2. Channel detection filter coefficients and spectrum using *Itap=12*. [9]

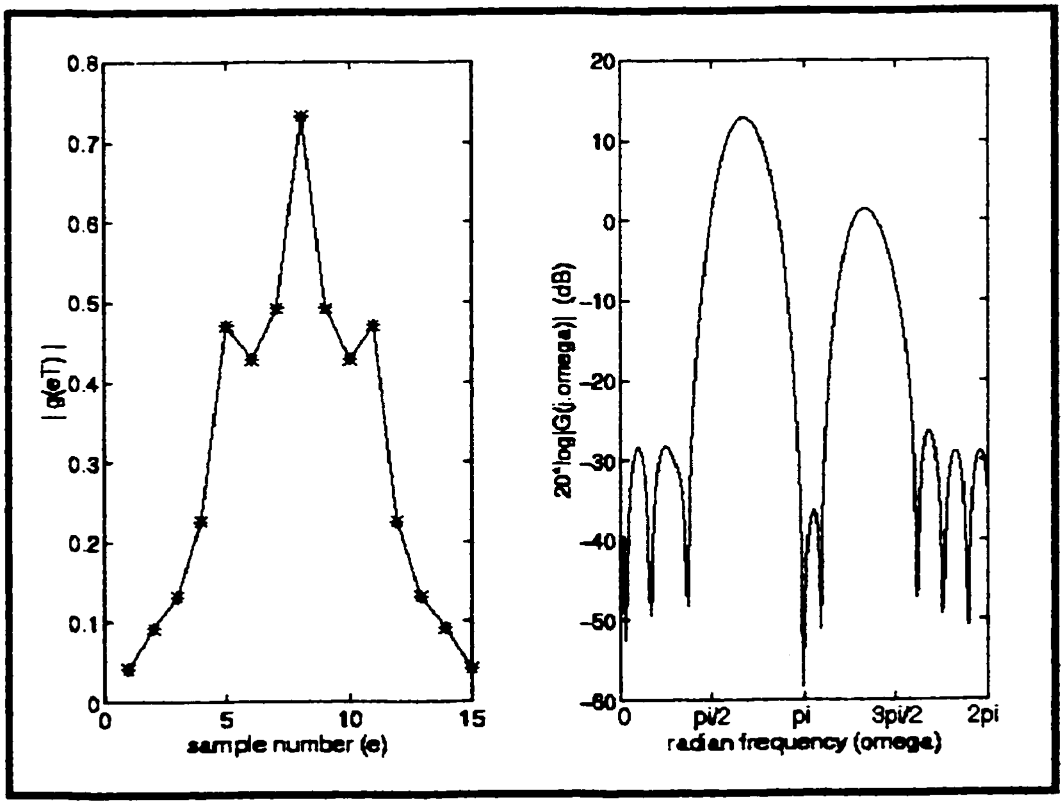


Figure 3-3. Channel detection filter coefficients and spectrum using *Itap=15*. [9]

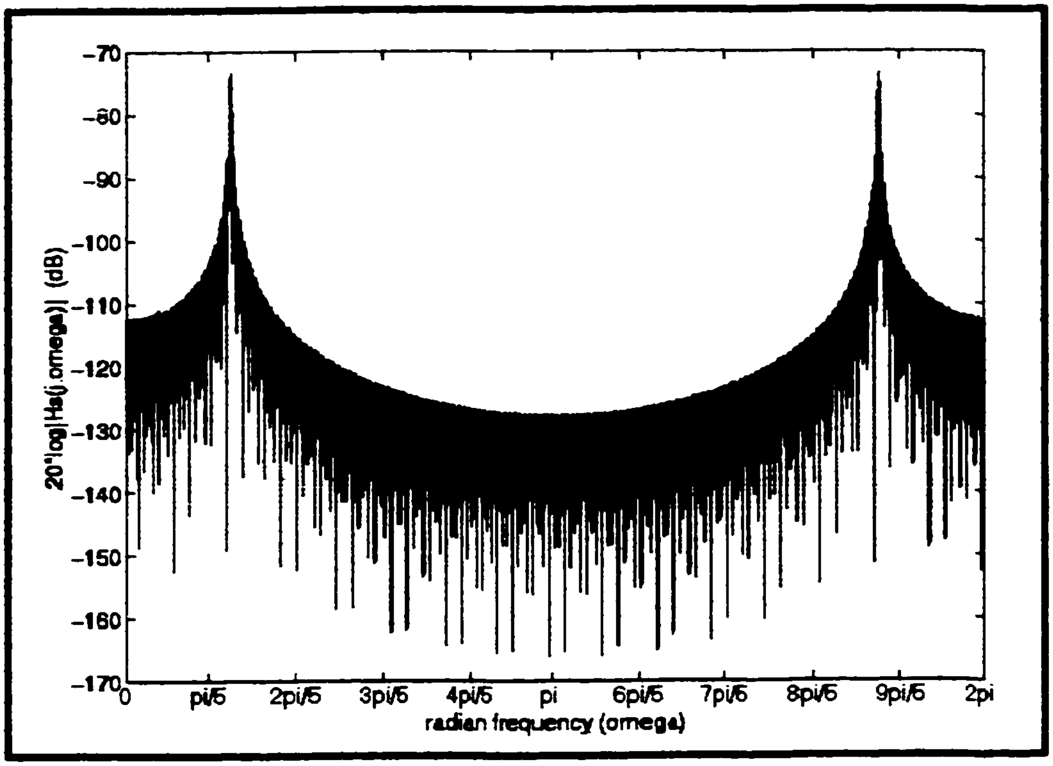


Figure 3-4. Transmit filter spectrum from the scenario *Ltap= 9* and *Itap=15.* [9]

The halfband filter coefficients were produced by FIR3 which is an FIR filter design algorithm created by Burrus [32-33]. This technique was adopted because of its claim that no complex windowing technique was needed to produce an FIR filter impulse response and also, FIR3 was seen to be more numerically stable and closer to the true impulse response than the other algorithms inherent in MATLAB, such as FIR1 and FIR2 [†] [32-33]. In this design technique, points of the filter spectrum are specified and using an inverse FFT, the time domain samples are obtained [32]. To produce the halfband filter response, four evenly spaced frequency domain samples were used since they produced the time domain samples closest to the true halfband filter impulse response. Because there are *(Ltap-1)/2* filter taps below and above the center tap of each halfband filter, respectively, the center tap is set to be *(Ltap+1)/2,* corresponding to the tap *h(0)* . The transmit filter and channel detection filter have also been offset accordingly to be aligned with the center tap of the halfband filters. For purposes of illustration, the values of *Ltap* were limited between 5 and 13. There were not any numerical limitations in exceeding the specified range for *Ltap*. Table 3-1 shows the numerical generation of the halfband filter coefficients using FIR 3 and compares the result to the true filter coefficients (*Ltap=∞*) [25] [32-33]. Figures

[†] By stability, we mean the number of filter taps which FIR3 can generate is much larger than the other algorithms available [32-33].

3-5 and 3-6 show the coefficients graphically along with the corresponding spectrum for *Ltap=5* and *Ltap=13*, respectively.

Table 3-1. Comparison of actual and FIR3 halfband filter coefficients, *h(nT$_s$)* [25] [32-33].

n	0	1	3	5	7	9	11
Ltap = ∞	7.0711e-01	4.5016e-01	-1.5005e-01	9.0032e-02	-6.4308e-02	5.0018e-02	-4.0923e-02
Ltap = 5	7.6275e-01	4.5072e-01	-7.7151e-02	-	-	-	-
Ltap = 7	7.5665e-01	4.5332e-01	-8.9710e-02	1.4016e-02	-	-	-
Ltap = 9	7.4870e-01	4.5594e-01	-1.0517e-01	2.7726e-02	-4.0322e-03	-	-
Ltap = 11	7.4360e-01	4.5711e-01	-1.1445e-01	3.7310e-02	-9.4372e-03	1.2562e-03	-
Ltap = 13	7.3998e-01	4.5767e-01	-1.2066e-01	4.4390e-02	-1.4345e-02	3.3404e-03	-4.1048e-04

From Table 3-1, as *Ltap* increases, we see that the generated coefficients converge to the true values. Figures 3-5 and 3-6 demonstrate this further and the spectrums show a difference of 40dB in stopband rejection between *Ltap=5* and *Ltap=13* therefore an assumably large difference in adjacent channel crosstalk between the two cases [9] [26].

From the illustrations of the halfband filter and channel detection filter coefficients, we see that the step incrementation of *Ltap* is two while for that for *Itap* is three, respectively. For *Ltap*, this incrementation is clear however, the step increase in *Itap* has been used here based on previous analysis of other MCDDs and is solely intended for purposes of illustration of the numerical results [2] [27]. It should be noted, however, that any conclusions drawn upon concerning *Ltap* and *Itap* in the numerical results do not take into consideration this difference in incrementation.

The third major block is used to produce the intersymbol interference (ISI) and interchannel interference (ICI) coefficients on a symbol-per-symbol basis. These interference values are needed to produce the moments of the interference [2]. The first step is to produce the desired symbol, *u0*, which is assumed to be the center symbol on the k^{th} channel since we assume *mL=mU*. Because of the decimation factor of 2^j involved in each j^{th} stage of demultiplexing, the computation of all the coefficients are done in three parts, one for the part corresponding to the zeroeth coefficient of the halfband filter or the center tap, another part for the positive coefficients above the center tap and a third part for the negative coefficients below the center tap. The next step is to compute the interference on this desired symbol. To do so, a loop with a variable based on *Nchannel* (the total number of channels to be demultiplexed) is performed. If the k^{th} channel comes up, we compute the *u(·)* array, corresponding to the ISI. With all other interfering channels, we compute *vI(·)* and *vQ(·)* for the ICI. The interference coefficients are computed in a manner similar to how the desired symbol was computed.

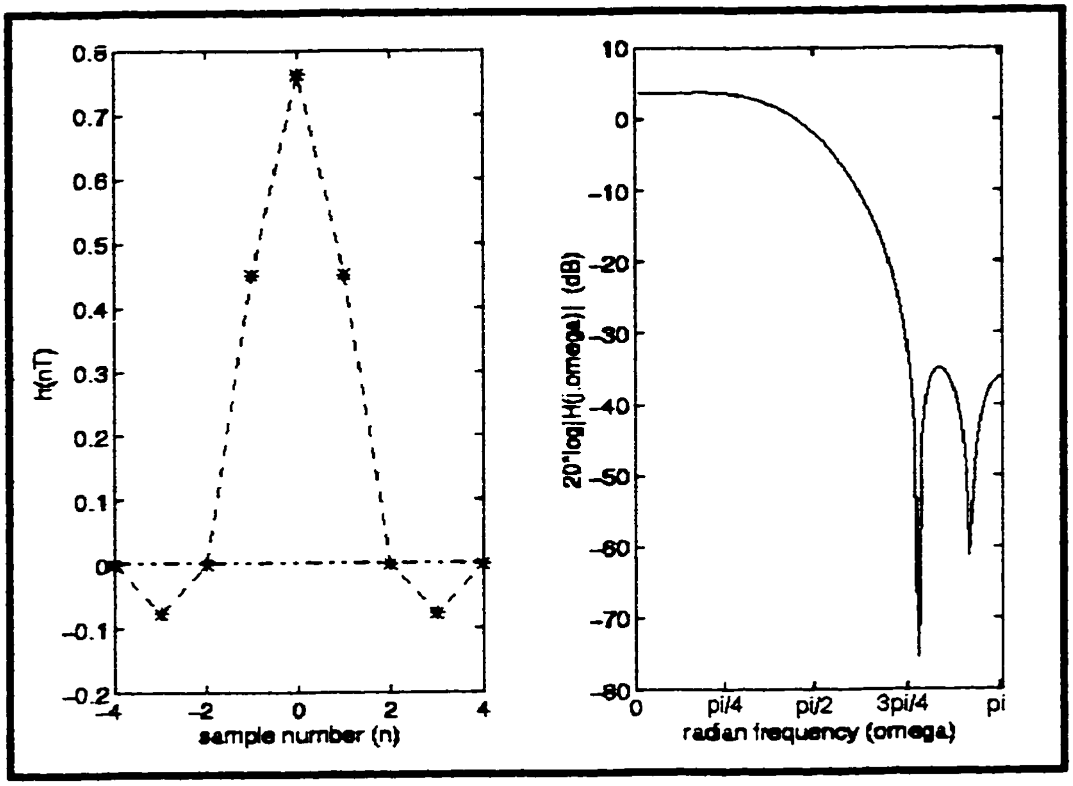


Figure 3-5. **Halfband filter coefficients and spectrum from FIR3 using** *Ltap=5*. [9] [25] [32-33]

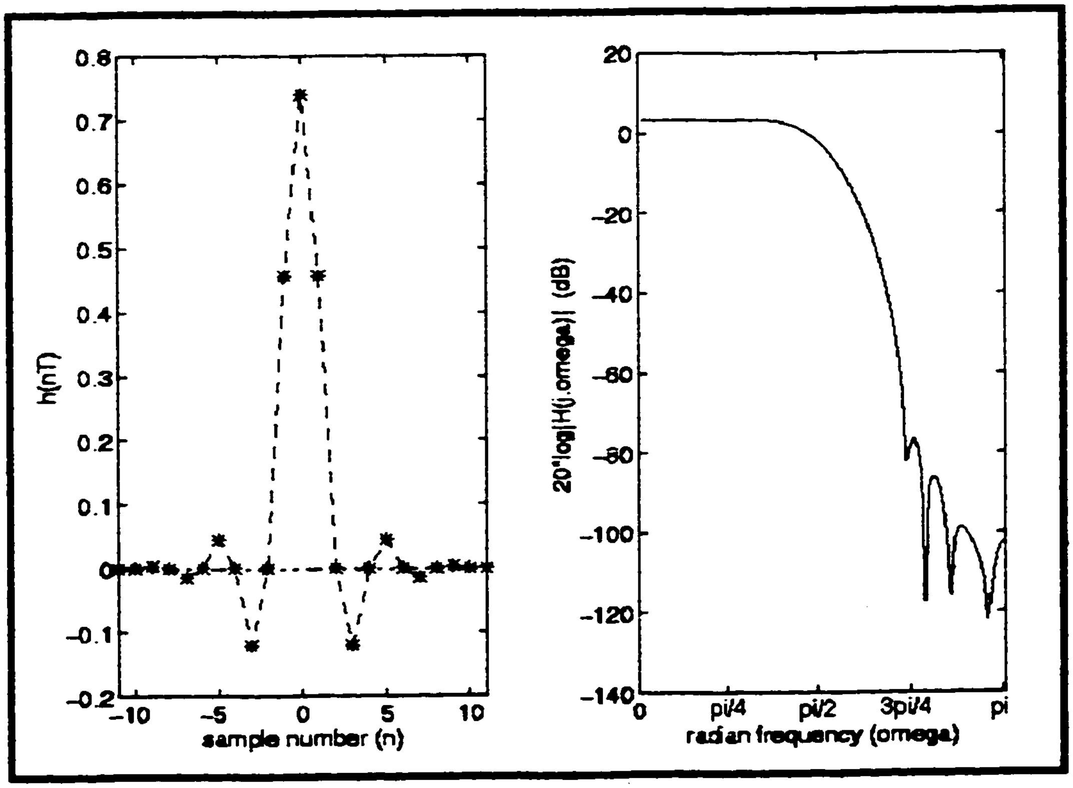


Figure 3-6. **Halfband filter coefficients and spectrum from FIR3 using** *Ltap=13*. [9] [25] [32-33]

The subsequent steps are summarized in Figure 3-7(b). Thus, the next step computes the maximum interference and noise adjustment factor according to (3-44) and (3-11), respectively. As mentioned, if the filter coefficients are scaled so as to have unit energy, the noise adjustment factor should theoretically come out to one since $\sigma_o^2 = \sigma^2$ [2]. Numerically, we expect it to come out to be very close to one.

What follows is the computation of the moments of the interference consisting of a few main steps. We first initialize the interference array so that the zeroeth moment is equal to one and all other moments are equal to zero. We then compute the interference of each ISI term according to (3-40), assuming a fixed timing offset. The individual ICI terms are then calculated according to (3-41), also assuming a fixed timing offset. In initial computations, the phase offset was assumed to be random to see what effect it would have on the performance. As a result, the phase offset was left as a random component. To integrate over the random phase, a *NI*=200 point integral was used with the alternate extended Simpson's Rule [2]. The total moments are then computed by collecting the moments of the individual symbols in Prabhu's Technique, specified in section 3.2.2 [29]. This method produces *Nterms* non-zero moments, not including the zeroeth moment equal to one, but a *2*Nterms+1* length array is returned containing also the zero-valued odd order moments. The weights, abscissas and the coefficient *C* of (3-43) used to compute the truncation error due to using only *Nterms* number of points in the GQR, are all computed from the GQR subroutine, described in section 3.2.1. The GQR subroutine has the *2*Nterms+1* array of the moments as an input parameter [15]. Once the weights and abscissas are obtained, the bit error rate (BER) and truncation error, R_M , from (3-21) and (3-42), respectively, are easily obtained. Here, it is assumed that only BERs above 10^{-15} are significant and if the BER for a particular SNR falls below 10^{-15} , the computation will not proceed. Although the error associated with the truncated series of (3-36) has not been computed. However, the total number of interfering symbols, *K*, in (3-37) has been selected to be large enough such that this error, shown in (3-47), is considered negligible [11]. As it will be discussed in section 3.4.2.2, *K* is chosen to be 167 interfering symbols over eight channels. That implies approximately twenty-one symbols per channel.

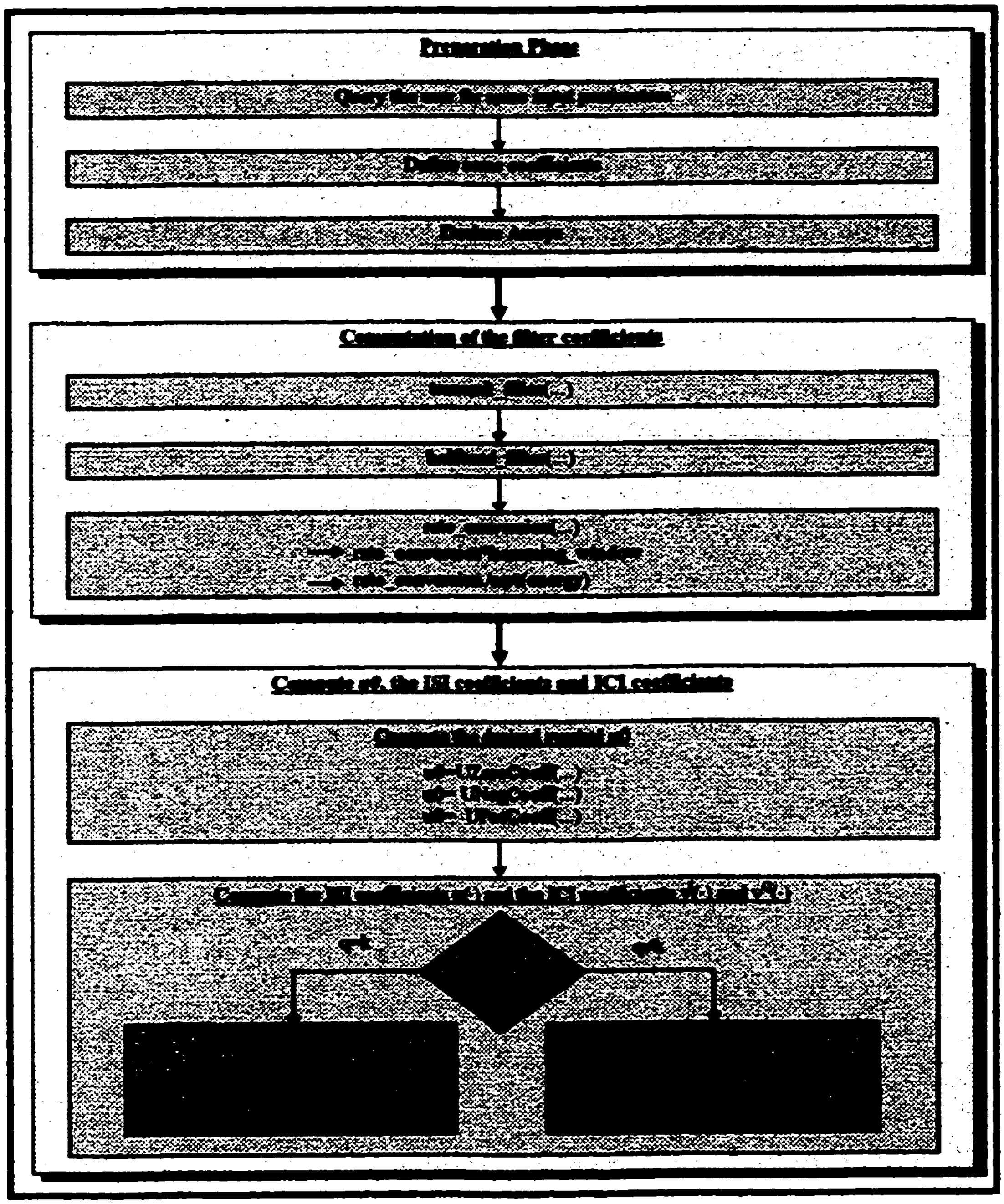
3.4.1.2 : Limitations of the GQR Algorithm

The GQR algorithm, although useful and effective in computing the performance of the M-MCDD, has its own limitations. For the number of interfering symbols, its optimum range is between ten and fifty. Any value below ten produced BERs of value zero and anything above fifty caused an insignificant change in the BER, that is accurate to four decimal places. Also, the number of points in the GQR, *Nterms*, could not exceed twenty-six or else the program would not execute. This was because the symmetric tridiagonal matrix **Z** of (3-30) could not produce the eigenvalues and eigenvectors to give the weights and abscissas, respectively. However, it was found that exceeding eighteen terms was unnecessary, even for low BERs, since any value of *Nterms* exceeding 18 would cause an underflow in the double precision arithmetic [2].

3.4.1.3 : The Beaulieu Fourier Series Expansion (FSE) Technique

For reasons which will be clarified later, the algorithm based on Beaulieu's Fourier Series Expansion (FSE) technique was developed primarily to verify the results of the GQR technique, that is, to verify the probability of bit error and to see if a comparable CPU time requirement and accuracy may be observed [16]. Therefore, in this program, the amount of input parameters to be specified by the user is limited to only *Ltap* and *Itap* for the number of non-zero filter taps for the halfband filters and number of filter taps for the channel detection filter, respectively. Of the possible values to choose from, there are a total of seven scenarios which the user may look at and compare to the previous results of the GQR algorithm. The general flow diagram of the technique is shown in Figure 3-8, making use of the first three blocks from Figure 3-7(a) of the GQR algorithm. The Beaulieu FSE algorithm is quite similar to the GQR algorithm, the only major exception being the computation of the BER and truncation error. This part of the Beaulieu FSE algorithm will now be elaborated [2] [16].

To compute the BER and truncation errors using this method, we run through a loop for each SNR value and another inner loop for each value of M (the number of terms in the Fourier Series of $Q(\cdot)$), which is a function of the SNR. Because an integration must be performed for the characteristic function of the interference, for each iteration, this algorithm is expected to take much more computation time to achieve the same end as the GQR algorithm [2]. This increased requirement in CPU time will be seen later on.



Preparation Phase
Computation of the filter coefficients
Compute a0, the ISI coefficients and ICI coefficients

Figure 3-7(a). Three initial stages common to both the GQR and Beaulieu FSE algorithms.

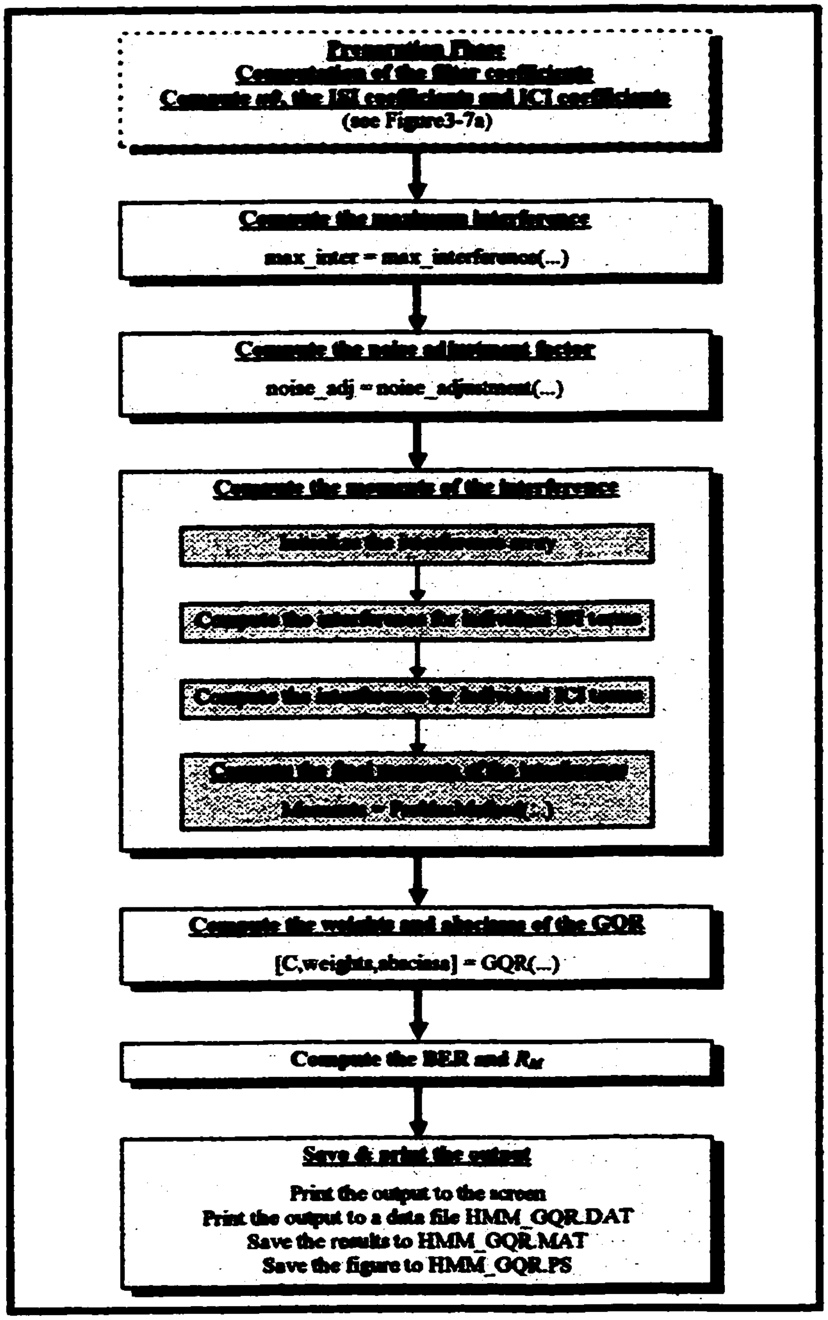

Preparation Phase
Computation of the filter coefficients
Compute a0, the ISI coefficients and ICI coefficients
(see Figure3-7a)
Compute the maximum interference
max_inter = max_interference(...)
Compute the noise adjustment factor
noise_adj = noise_adjustment(...)
Compute the moments of the interference
Initialize the interference array
Compute the interference for individual ISI terms
Compute the interference for individual ICI terms
Compute the final moments of the interference
Moments = [illegible](...)
Compute the weights and abscissa of the GQR
[C,weights,abscissa] = GQR(...)
Compute the BER and R_M
Save & print the output
Print the output to the screen
Print the output to a data file HMM_GQR.DAT
Save the results to HMM_GQR.MAT
Save the figure to HMM_GQR.PS

Figure 3-7(b). Flow diagram for the Gauss Quadrature Rule Algorithm

For each iteration, we first compute the three truncation errors which are *CFTruncError* (the error due to using a finite number of interfering symbols in the characteristic function), *FourierApproxError* (the error due to using the Fourier Series as an approximation to $Q(\cdot)$) and *SeriesTruncError* (the error due to using a finite number of terms, M, in the Fourier Series of $Q(\cdot)$). Actually, *CFTruncError* is computed near the end of the loop but at the beginning of this block, a parameter needed to compute this error is calculated. To compute *FourierApproxError* , we must specify the value of T, that is, the period of the square wave gating function [16]. Likewise, for *SeriesTruncError* , we must specify the value of M. The criterion used to choose the values of M and T are further discussed in section 3.4.2.2 on numerical accuracy.

Once M and T are calculated, we go through each iteration of the Fourier Series up to M to obtain the final BER and truncation error. Through each loop, we compute the characteristic function to update P_b and *CFTruncError* , according to (3-61) and (3-65), respectively. Since the computation of the characteristic function is in the frequency domain while for the moments it is in the time domain, the computation of the characteristic function of the interference is similar to that of the corresponding moments [31]. Therefore, we compute the characteristic function for the individual ISI terms and then for the individual ICI terms. Just as was the case with the moments, we integrate over the random phase using the alternate extended Simpson's Rule [2]. In this calculation, however, instead of using Prabhu's Method as was done for the moments, we multiply each individual characteristic function to obtain the final characteristic function. This can be done because of the assumption of the data symbols to be identically and independently distributed (i.i.d.) [10-11]. Once we complete all M iterations of the Fourier Series, we arrive at the final *CFTruncError* value and that of the BER for a specific SNR. This process continues for each SNR value and the calculation is complete once all the truncation errors and BERs are computed for every value of SNR, that is for $0\text{dB} \leq \text{SNR} \leq 15\text{dB}$, or the BER becomes insignificantly small. In this algorithm, $\text{BER}<10^{-8}$ is considered insignificantly small.

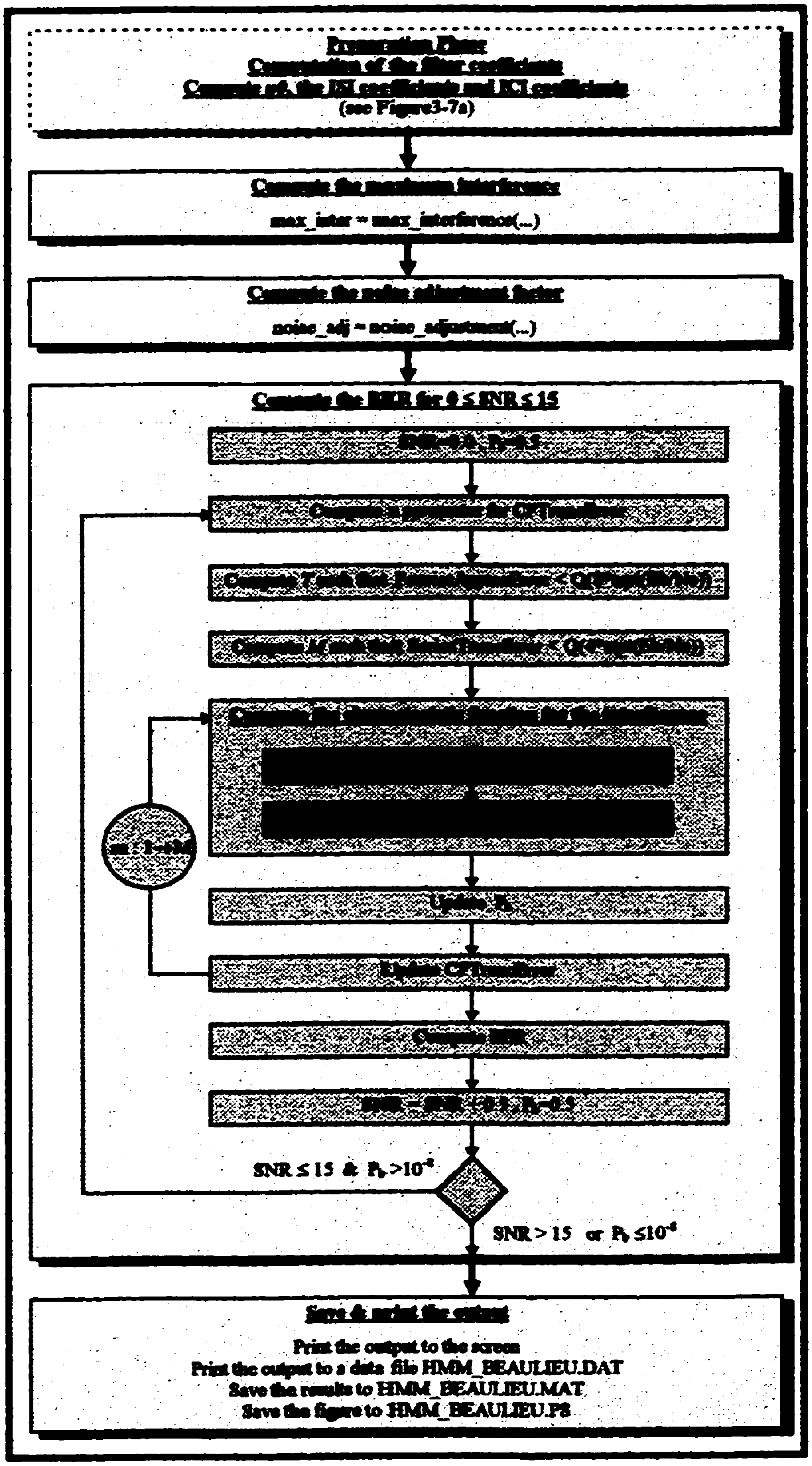

Preparation Phase
Computation of the filter coefficients
Compute a0, the ISI coefficients and ICI coefficients
(see Figure3-7a)
Compute the maximum interference
max_inter = max_interference(...)
Compute the noise adjustment factor
noise_adj = noise_adjustment(...)
Compute the BER for 0 ≤ SNR ≤ 15
SNR=0.0, Pb=0.5
Update Pb
Update CF transform
Compute BER
SNR = SNR + 0.5, Pb=0.5
SNR ≤ 15 & Pb >10^-6
SNR > 15 or Pb ≤10^-6
Save & print the output
Print the output to the screen
Print the output to a data file HMM_BEAULIEU.DAT
Save the results to HMM_BEAULIEU.MAT
Save the figure to HMM_BEAULIEU.PS

Figure 3-8. Flow diagram for Beaulieu's FSE Algorithm.

3.4.2 : Performance Results of the M-MCDD

3.4.2.1 : Computation and Organization of Results

Each expression containing the terms $cos(\pi \cdot (k-q) \cdot \sum_{j=0}^{l} n_j)$ and $sin(\pi \cdot (k-q) \cdot \sum_{j=0}^{l} n_j)$ requiring numerical computation will have to be put in a calculable form. Substituting (3-15) and (3-16) into each of these expressions containing the cos(·) and sin(·) terms, respectively, will give a final set of equations which are computable by various software packages. As mentioned in section 3.4.1.1, the software package chosen to carry out the numerical computation was MATLAB. That is, MATLAB was used to code the GQR technique. With MATLAB, the algorithm for the GQR could be made more compact by creating subroutines as user-defined functions. There are also many "canned" routines inherent to MATLAB which can be used to make the code of the algorithm more compact [32]. One clear disadvantage of MATLAB, however, is its lack of speed when executing FOR loops. It would be possible to reduce CPU time requirements of the algorithm by using MEX files which would contain these FOR loops computed in C language [32]. Such a possibility, however, was not investigated.

The performance analysis numerical results will be presented on two fronts. In section 3.4.2.2, we will present the numerical accuracy of these results. To show the numerical accuracy of the performance results, we shall look at the results from the GQR technique and the Beaulieu FSE in terms of relative accuracy of the BER, and relevant truncation errors. We will also compare the CPU execution time. Also, the accuracy of the results will be investigated by varying several important parameters of the algorithms, namely, *mL*, *mU* (the number of interfering symbols), *k* (desired channel position) and *M* (number of terms). From the discussion, it will be shown how the value of these parameters were set in order to determine the system performance of the M-MCDD.

In section 3.4.2.3, we present the results of the system performance based on the GQR technique using associated values of the parameters determined in section 3.4.2.2. This section will show how the values of *Ltap* and *Itap* can be chosen based on a desired BER performance and degradation in SNR relative to ideal QPSK demodulation one would like to achieve while keeping the computational complexity at an acceptable level. System performance results will be presented using BER curves, isometric contours and computational complexity. These results will be used to specify the design criteria of the M-MCDD through illustrative examples.

3.4.2.2 : Numerical Accuracy

Tables 3-2 and 3-3 show the results from two scenarios for both the GQR and Beaulieu FSE Technique for *Ltap=9* , *Itap=3* and *Ltap=5* , *Itap=9*, respectively. The remaining parameters were kept at the following values: *mL=mU=10, l=3, k=4,* f_s *=1.5* and *beta=0.5*. The corresponding BER curves resulting from both algorithms are shown in Figures 3-9 and 3-10 depicting the information shown in both tables, respectively.

From the BER curves of Figures 3-9 and 3-10, we see that the results from the GQR and Beaulieu FSE technique are very close. We can also make observations as to the numerical results we obtain using both techniques bearing in mind that our purpose is not to compare the two sets of results. From previous literature, it is expected that a lower relative truncation error can be achieved with the Beaulieu Technique than with the GQR technique but from these numerical results, this is not the case [15-16]. The curves from the GQR were produced using up to eighteen terms, depending on the SNR, since the BER curve would not change substantially by using more terms whereas for the Beaulieu Technique, many more terms were required. Using more terms would directly affect the CPU time requirements for the algorithms therefore the CPU time required for the Beaulieu Technique is much higher than that required for the GQR.

From Tables 3-2 and 3-3, only one truncation error for each technique is reported. In the case of the GQR, only the truncation error R_M , for using a limited number of points in the GQR, is reported because the other truncation error, that is for using a limited number of interfering symbols, is expected to be very small from previous work thus insignificant in value. The truncation error R_M in the GQR can also be regarded as the error for only using $2M+1$ moments to evaluate P_b [2]. As for the Beaulieu FSE Technique, only the truncation error due to using a finite number of interfering symbols in the characteristic function of the interference, Δ, is reported since the remaining two truncation errors, namely the error due to using the Fourier Series to approximate $Q(\cdot)$ and the error due to using M terms in the Fourier Series expansion for $Q(\cdot)$, were calculated through the Beaulieu FSE algorithm and determined not to contribute to the overall error magnitude since they were insignificantly small in value. Another factor contributing to the overall error in both algorithms is the error due to using the alternate extended Simpson's Rule for the numerical integration over the random phase [2]. This error was not taken into account in the computation of the truncation errors.

Table 3-2. Comparison of results from GQR and Beaulieu FSE for the case *Ltap=9* and *Itap=3*.

	GQR (CPU time → 8:58.5167)			Beaulieu FSE (CPU time → 11:43:38.0833)		
Eb/No (dB)	M	P_b	R_M	(M_{BEAU}, T)	P_b	Δ
0	3	9.3448e-02	5.3211e-022	(13,20)	8.1384e-02	2.1177e-002
1	4	7.2599e-02	2.0666e-021	(17,22)	6.3403e-02	2.2889e-002
2	4	5.4562e-02	3.8995e-021	(21,25)	4.8251e-02	2.3803e-002
3	4	3.9634e-02	2.9655e-021	(27,28)	3.5966e-02	2.3819e-002
4	5	2.7849e-02	7.1836e-022	(33,31)	2.6372e-02	2.2986e-002
5	5	1.8979e-02	4.1217e-023	(41,35)	1.9133e-02	2.1477e-002
6	6	1.2607e-02	3.8579e-025	(51,39)	1.3830e-02	1.9539e-002
7	6	8.2210e-03	3.6836e-028	(63,43)	1.0036e-02	1.7429e-002
8	6	5.3087e-03	1.9868e-032	(79,49)	7.3654e-03	1.5355e-002
9	7	3.4280e-03	2.8763e-038	(97,54)	5.5019e-03	1.3458e-002
10	9	2.2349e-03	4.3804e-046	(123,61)	4.2046e-03	1.1807e-002
11	10	1.4836e-03	2.1579e-056	(155,68)	3.2982e-03	1.0421e-002
12	12	1.0095e-03	7.7910e-070	(197,77)	2.6602e-03	9.2850e-003
13	17	7.0716e-04	3.1806e-087	(245,86)	2.2061e-03	8.3699e-003
14	17	5.1110e-04	1.3960e-109	(307,96)	1.8791e-03	7.6402e-003
15	18	3.8206e-04	3.4060e-138	(387,108)	1.6405e-03	7.0619e-003

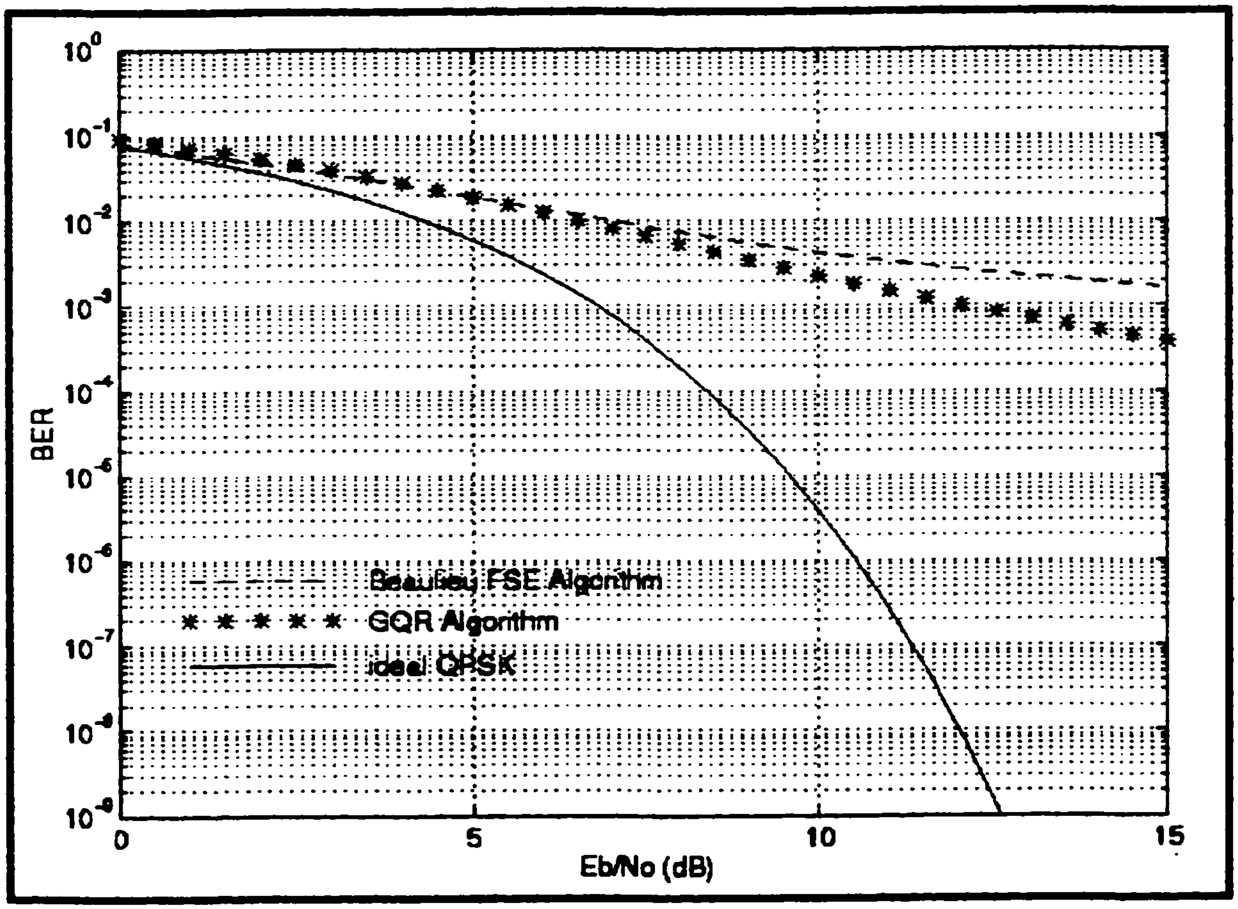


Figure 3-9. Comparison of GQR and Beaulieu FSE algorithm for the case *Ltap=9* and *Itap=3*.

Table 3-3. Comparison of results from GQR and Beaulieu FSE for the case *Ltap=5* and *Itap=9*.

	GQR (CPU time → 13:26.3333)			Beaulieu FSE (CPU time → 11:25:16.2333)		
Eb/No (dB)	M	P_b	R_M	(M_{BEAU}, T)	P_b	Δ
0	3	8.3385e-02	2.1067e-027	(13,20)	7.2091e-02	1.1005e-002
1	4	6.1466e-02	8.2147e-026	(17,22)	5.3069e-02	1.1737e-002
2	4	4.2841e-02	2.8279e-024	(21,25)	3.7355e-02	1.1872e-002
3	4	2.7968e-02	8.3216e-023	(27,28)	2.5071e-02	1.1331e-002
4	5	1.6930e-02	2.0099e-021	(33,31)	1.6026e-02	1.0166e-002
5	5	9.4060e-03	3.7860e-020	(41,35)	9.7683e-03	8.5632e-003
6	6	4.7507e-03	5.2147e-019	(51,39)	5.7000e-03	6.7829e-003
7	6	2.1645e-03	4.8434e-018	(63,43)	3.2072e-03	5.0778e-003
8	6	8.8558e-04	2.7392e-017	(79,49)	1.7580e-03	3.6236e-003
9	7	3.2547e-04	8.2961e-017	(97,54)	9.5082e-04	2.4934e-003
10	9	1.0815e-04	1.1446e-016	(123,61)	5.1462e-04	1.6770e-003
11	10	3.2949e-05	5.8691e-017	(155,68)	2.8274e-04	1.1184e-003
12	12	9.3997e-06	8.6559e-018	(197,77)	1.5977e-04	7.4961e-004
13	17	2.5809e-06	2.6592e-019	(245,86)	9.3851e-05	5.1101e-004
14	17	7.0391e-07	1.1336e-021	(307,96)	5.7759e-05	3.5757e-004
15	18	1.9699e-07	4.0218e-025	(387,108)	3.7411e-05	2.5842e-004

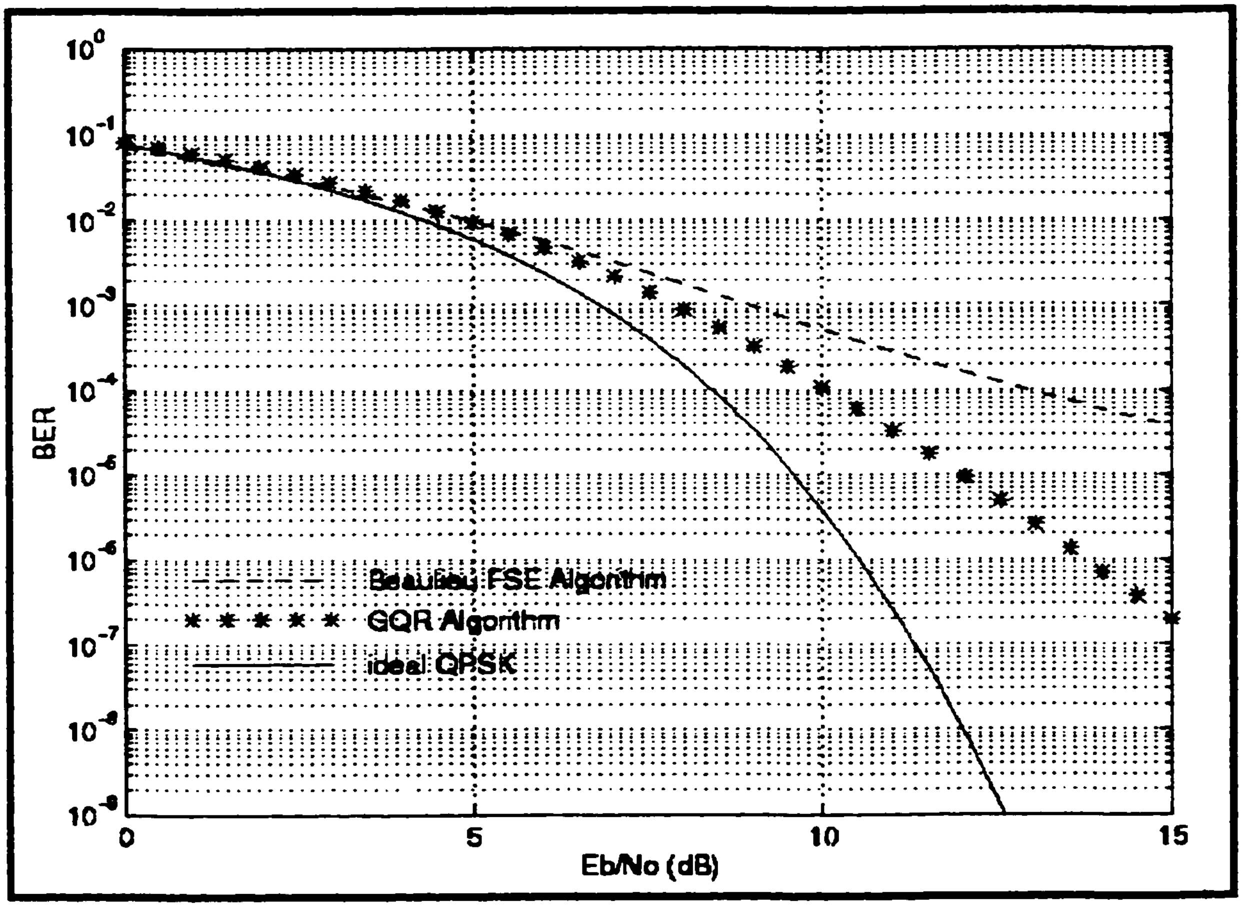


Figure 3-10. Comparison of GQR and Beaulieu FSE algorithm for the case *Ltap=5* and *Itap=9*.

If we look at the truncation errors of each algorithm, we can make several observations from Tables 3-2 and 3-3. We find that the truncation error associated with the GQR slowly approaches zero for low BERs while the error associated with the Beaulieu technique does not go below 10^{-5}. Thus, in the case of the Beaulieu FSE technique, for BERs which are below 10^{-5}, this would serve to indicate a lack of numerical accuracy. Unfortunately, at these points we wish a high level of numerical accuracy since errors can occur quite easily for low BERs. It should be noted that it would be possible to achieve a slightly better accuracy if more interfering symbols were used in the characteristic function, but this was not done. As mentioned, mL and mU were fixed at ten interfering symbols each for both techniques. In the case of the GQR, we can see a high level of accuracy relative to the BER values since the truncation error is much smaller than the corresponding BER.

Generally, we see that the number of GQR terms required to achieve the desired level of accuracy for the GQR technique is small while the number of terms required in the Fourier Series for the Beaulieu FSE technique is quite large. In the case of Beaulieu's FSE technique, there are two parameters that need to be determined for each SNR to set the number of terms in the Fourier Series. The first value is T, that is, the period of the square wave gating function [16]. The value of T was chosen such that [2]:

$$|\beta| < Q\left(\tfrac{T}{2} - A_k u(0) + \xi_{\max}\right) < Q\left(8 \cdot \sqrt{\frac{E_b}{N_o}}\right) \tag{3-68}$$

, referring to (3-63) with a fixed timing offset. The value of M_{BEAU} may then be chosen as a function of the value of T. Thus, the value of M_{BEAU} was chosen such that [2]:

$$R_M < \frac{\sqrt{2\pi} \cdot T}{\pi^2 \cdot M} \cdot Q\left(\frac{2\pi M}{T}\right) < Q\left(4 \cdot \sqrt{\frac{E_b}{N_o}}\right) \tag{3-69}$$

, referring to (3-64). The reason why the bound in (3-68) was set at $Q(8 \cdot \sqrt{E_b/N_o})$ to choose the value of T was that from previous work done, it was observed that by choosing T too small, the Fourier Series would not converge. Thus, to keep the value of T small while allowing the series to converge, the conservative approach of (3-68) is used [2]. In general, it is better to use a larger T value and use more terms in the Series rather than using smaller parameters to try to save time but then we would have to rerun the lengthy algorithm because the program would fail. Although this criteria for choosing T allows the Series to converge, it is not optimum, that is, it may be possible to find a smaller T value and a smaller number of terms, M_{BEAU}, such that the Series would converge, but such values of T and M_{BEAU} were not sought after [2]. The values of M_{BEAU} and T are also used in calculating the truncation errors *SeriesTruncError* and *FourierApproxError* , respectively, as mentioned in section 3.4.1.3.

In the case of the GQR, although the truncation error is very small, it is possible to make this error smaller by increasing the number of terms in the GQR however, the probability of bit error will eventually not change if *Nterms* is increased beyond a certain level. Thus, to determine a sufficient number of terms in the GQR, it is more useful to look at the convergence of the BER for a particular SNR value instead of attempting to minimize the corresponding truncation error. Therefore, in order to achieve the level of precision required for the probability of bit error, the number of GQR points had to be chosen for each signal-to-noise (SNR) as shown in Table 3-4 and Figure 3-11 for the case *Ltap=9* and *Itap=12*. The number of GQR points, M , was chosen by determining when P_b was precise to four decimal places [2]. In the results from the GQR that follow, it may be noticed that the number of terms used for each SNR correspond for each case of *Ltap* and *Itap* chosen. Please note that the above method for determining M was used for each case and the fact that they all correspond is simply coincidence.

Table 3-4. Variation in BER for *Ltap=9* and *Itap=12* to determine the choice for *M*.

	Note that bolded BERs indicate convergence for a particular SNR value.					
	P_b	P_b	P_b	P_b	P_b	P_b
Nterms	SNR=10dB	SNR=11B	SNR=12B	SNR=13B	SNR=14B	SNR=15B
3	2.7178e-05	4.2416e-06	4.3879e-07	2.6535e-08	8.0492e-10	1.0176e-11
4	2.9821e-05	5.3519e-06	7.1143e-07	6.2917e-08	3.1998e-09	7.8739e-11
5	3.0062e-05	5.5923e-06	8.2848e-07	9.2179e-08	6.8088e-09	2.8116e-10
6	3.0049e-05	5.6052e-06	8.5134e-07	1.0460e-07	9.7694e-09	5.9674e-10
7	3.0047e-05	5.6024e-06	8.5175e-07	1.0704e-07	1.1136e-08	8.8456e-10
8	3.0047e-05	5.6022e-06	8.5128e-07	1.0701e-07	1.1440e-08	1.0402e-09
9	**3.0047e-05**	5.6023e-06	8.5128e-07	1.0693e-07	1.1435e-08	1.0836e-09
10	**3.0047e-05**	**5.6023e-06**	8.5129e-07	1.0693e-07	1.1419e-08	1.1419e-08
11	**3.0047e-05**	**5.6023e-06**	8.5129e-07	1.0693e-07	1.1419e-08	1.0813e-09
12	**3.0047e-05**	**5.6023e-06**	**8.5129e-07**	1.0693e-07	1.1420e-08	1.0809e-09
13	**3.0047e-05**	**5.6023e-06**	**8.5129e-07**	1.0693e-07	1.1420e-08	1.0811e-09
14	**3.0047e-05**	**5.6023e-06**	**8.5129e-07**	1.0693e-07	1.1420e-08	1.0812e-09
15	**3.0047e-05**	**5.6023e-06**	**8.5129e-07**	1.0693e-07	1.1420e-08	1.0811e-09
16	**3.0047e-05**	**5.6023e-06**	**8.5129e-07**	1.0693e-07	1.1420e-08	1.0811e-09
17	**3.0047e-05**	**5.6023e-06**	**8.5129e-07**	**1.0693e-07**	**1.1420e-08**	1.0811e-09
18	**3.0047e-05**	**5.6023e-06**	**8.5129e-07**	**1.0693e-07**	**1.1420e-08**	**1.0811e-09**
19	**3.0047e-05**	**5.6023e-06**	**8.5129e-07**	**1.0693e-07**	**1.1420e-08**	**1.0811e-09**
20	**3.0047e-05**	**5.6023e-06**	**8.5129e-07**	**1.0693e-07**	**1.1420e-08**	**1.0811e-09**

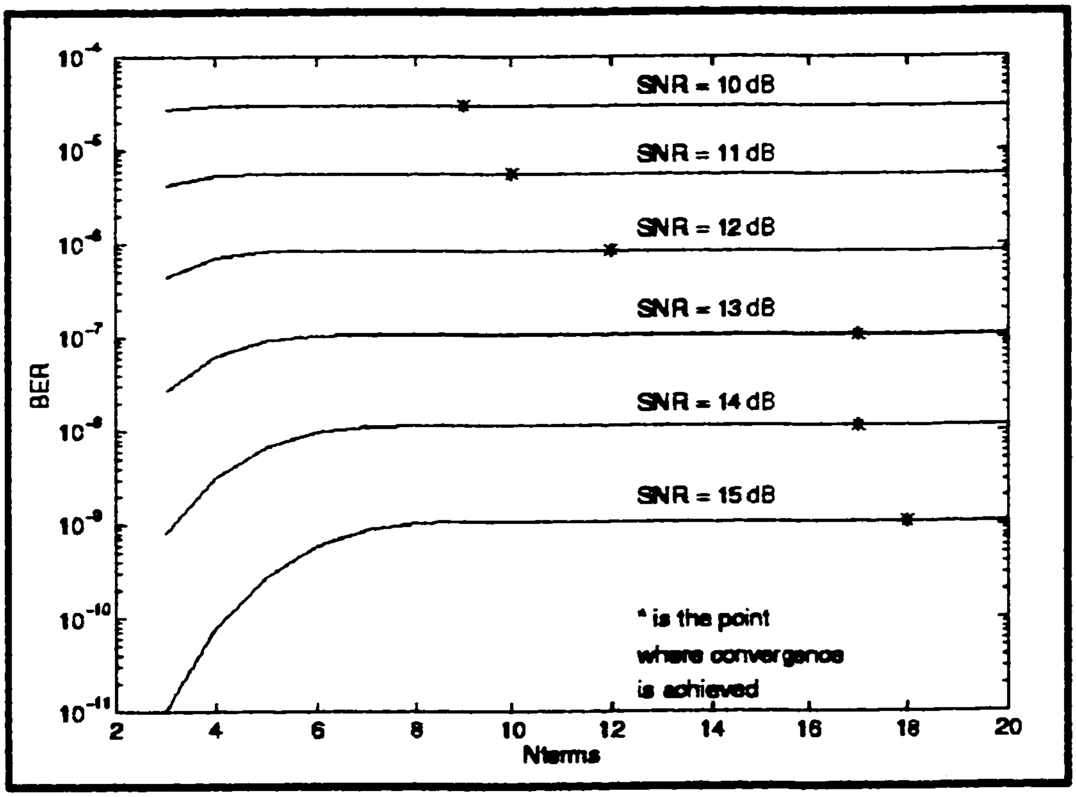


Figure 3-11. Variation in BER for *Ltap=9* and *Itap=12* to determine the choice for *M*.

From the results, the advantage which the GQR has over the Beaulieu FSE technique is its lower CPU time requirements. Because of this, we choose the GQR as our preferred method to evaluate the performance of the M-MCDD. While the GQR technique generally requires minutes to compute a BER curve, the Beaulieu FSE technique requires many hours to do the same job. The main reason for the large difference in CPU time between both techniques, other than the large difference in the number of terms required by each algorithm, is that in the GQR technique, all the numerical integrations are performed with the computation of the moments of the interference [2] [15]. The moments are computed fairly quickly since use is made of Prabhu's Technique and once these moments are obtained, the weights and abscissas are found and the BER curve is generated [29]. For the Beaulieu FSE technique, however, a numerical integration must be performed to determine the characteristic function of the interference for each term in the Fourier Series and at each SNR value [2] [16]. Thus, the total number of integrations that are needed for each scenario is the sum of all the M_{BEAU} values from each table. With the amount of time required for each numerical integration being ten to fifteen seconds on average, it is no wonder that the entire Beaulieu algorithm requires ten to eleven hours of CPU time to compute!

Based on the results of the GQR algorithm using up to *Nterms=18* GQR points, we will look at the effect of varying the parameters *mL*, *mU* and *k* on the numerical accuracy. For the results which follow and for the system performance, it is assumed we have an *l=3* stage M-MCDD used to demultiplex an eight channel MF-TDMA signal (*Nchannels=8*). As for the value of the desired channel, *k*, it was observed that varying *k* between one and eight had no effect on the performance of the M-MCDD. Although this may seem unusual, it is likely due to the initial assumptions made on the composite input signal, that is, the fact that the frequency spacing between the carriers is uniform. For the results which follow, it is assumed that the desired channel is in the middle of the eight channel MF-TDMA signal, that is, *k=4*.

The values of *mL* and *mU* affected the BER in a direct manner. That is to say, if more interfering symbols were introduced into the demodulated output, the BER would increase but increasing *mL* and *mU* beyond fifty would cause an insignificant change in the BER. Likewise, if fewer interfering symbols were used, the BER would be lower, however the numerical results would be less accurate. The minimum values of *mL* and *mU* was ten because any values less than ten caused numerical instability in the BER and truncation error. Tables 3-5 and 3-6 show the effect of varying *mL* and *mU* on the BER and accuracy, respectively. Particularly, *mL* and *mU* are varied between ten and fifty for the scenario *Ltap=9* and *Itap=9*. Also, Figure 3-12 graphically depicts the increase in the BER from *mL=mU=10* to *mL=mU=50* according to what is shown numerically in Table 3-5.

Table 3-5. Comparison of BERs using different *mL* and *mU* for the case *Ltap=9* and *Itap=9*.

	mL=mU=10 CPU time → 17:31.3	*mL=mU=20* CPU time → 34:01.9	*mL=mU=30* CPU time → 54:32.1	*mL=mU=50* CPU time → 88:56.7
Eb/No (dB)	P_b	P_b	P_b	P_b
0	8.2199e-02	8.4742e-02	8.6132e-02	8.6248e-02
1	6.0164e-02	6.2959e-02	6.4491e-02	6.4620e-02
2	4.1493e-02	4.4394e-02	4.5994e-02	4.6129e-02
3	2.6666e-02	2.9480e-02	3.1050e-02	3.1183e-02
4	1.5772e-02	1.8293e-02	1.9727e-02	1.9849e-02
5	8.4710e-03	1.0532e-02	1.1742e-02	1.1846e-02
6	4.0742e-03	5.5941e-03	6.5317e-03	6.6137e-03
7	1.7315e-03	2.7334e-03	3.3975e-03	3.4570e-03
8	6.4279e-04	1.2297e-03	1.6601e-03	1.6999e-03
9	2.0685e-04	5.1245e-04	7.6899e-04	7.9377e-04
10	5.7597e-05	2.0009e-04	3.4238e-04	3.5688e-04
11	1.3961e-05	7.4481e-05	1.4913e-04	1.5724e-04
12	2.9918e-06	2.7017e-05	6.4835e-05	6.9255e-05
13	5.8146e-07	9.7891e-06	2.8724e-05	3.1119e-05
14	1.0608e-07	3.6329e-06	1.3216e-05	1.4532e-05
15	1.8919e-08	1.4126e-06	6.4149e-06	7.1603e-06

Table 3-6. Comparison of R_M using different *mL* and *mU* for the case *Ltap=9* and *Itap=9*.

	mL=mU=10 CPU time → 17:31.3	*mL=mU=20* CPU time → 34:01.9	*mL=mU=30* CPU time → 54:32.1	*mL=mU=50* CPU time → 88:56.7
Eb/No (dB)	R_M	R_M	R_M	R_M
0	2.4383e-029	1.4466e-027	4.0645e-029	1.3541e-033
1	1.1561e-027	1.7432e-026	7.7046e-029	1.6558e-034
2	5.0909e-026	1.3683e-025	5.8930e-029	4.0181e-036
3	2.0423e-024	6.2615e-025	1.4378e-029	1.2728e-038
4	7.2871e-023	1.4524e-024	8.3253e-031	3.1072e-042
5	2.2431e-021	1.4323e-024	7.8822e-033	3.0098e-047
6	5.7341e-020	4.8128e-025	7.6354e-036	5.0170e-054
7	1.1600e-018	4.1690e-026	4.1937e-040	5.0264e-063
8	1.7481e-017	6.5542e-028	6.2117e-046	8.0517e-075
9	1.8182e-016	1.2021e-030	9.7362e-054	3.8934e-090
10	1.1857e-015	1.4750e-034	4.9731e-064	6.9682e-110
11	4.2967e-015	6.0096e-040	1.8793e-077	3.2869e-135
12	7.4299e-015	3.3669e-047	8.1257e-095	1.4681e-167
13	5.0624e-015	8.5494e-057	3.8338e-117	9.4283e-209
14	1.0678e-015	2.4328e-069	1.0244e-145	4.4698e-261
15	5.1478e-017	1.3359e-085	3.7411e-182	*****underflow****

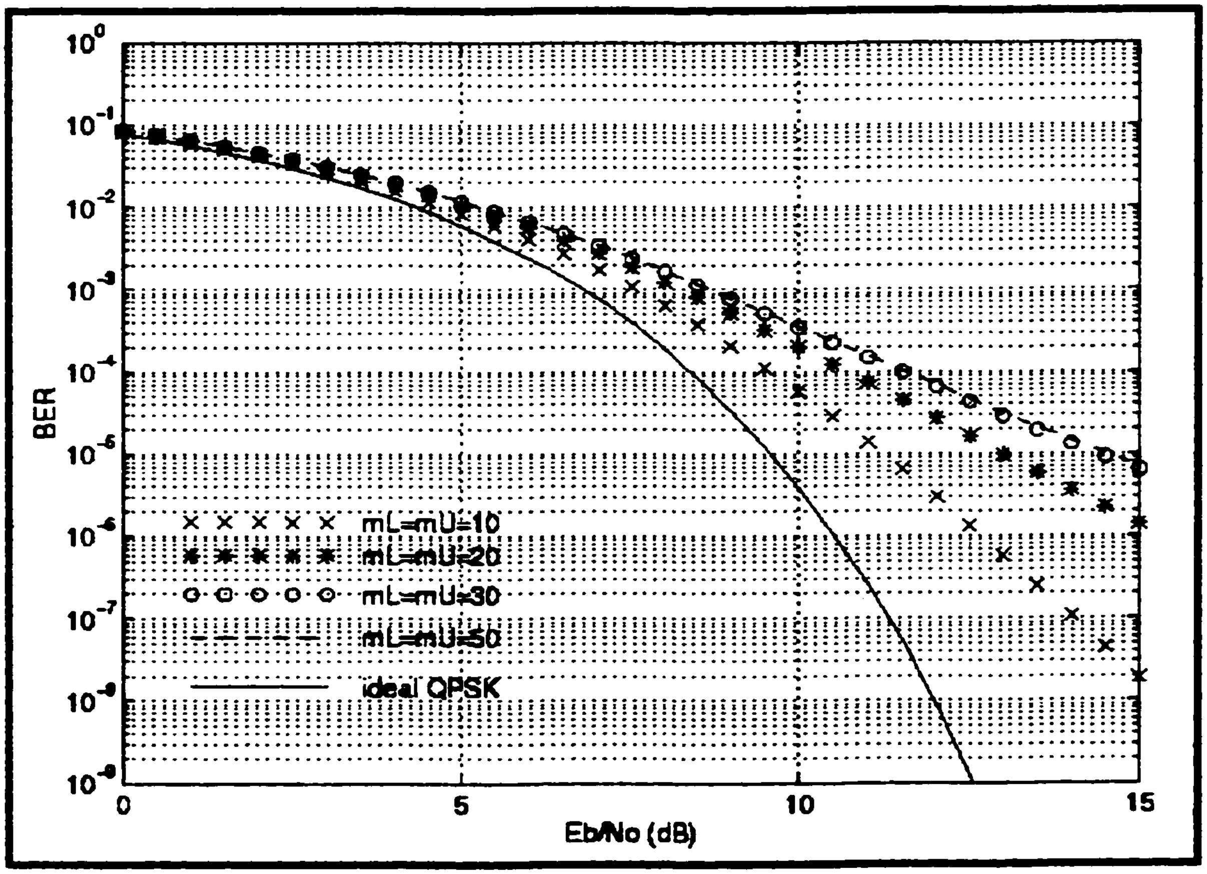


Figure 3-12. Comparison of BERs using different *mL* and *mU* for the case *Ltap=9* and *Itap=9*.

We notice from these tables and Figure 3-12 that when the number of interfering symbols increases, the BER also increases however, at the same time, the associated truncation error is lower especially for larger SNR values. We also notice from these results that although we obtain a better accuracy by using a larger number of interfering symbols, as *mL* and *mU* approach fifty, the BER experiences a smaller increase. From Figure 3-12, we easily see that there is little difference in BER between the case *mL=mU=30* and *mL=mU=50*. If we were to surpass fifty, the BER would not change significantly. This slower increase in the BER as *mL* and *mU* increase is due to the decaying values of the interfering symbols as they diverge further from the desired symbol. To show this decay in value, Figure 3-13 shows the change in the profile of the ISI as *mL* and *mU* increase from thirty to fifty. Likewise, Figures 3-14 and 3-15 show a similar change in profile of the ICI in its in-phase and quadrature components, respectively, for an adjacent channel.

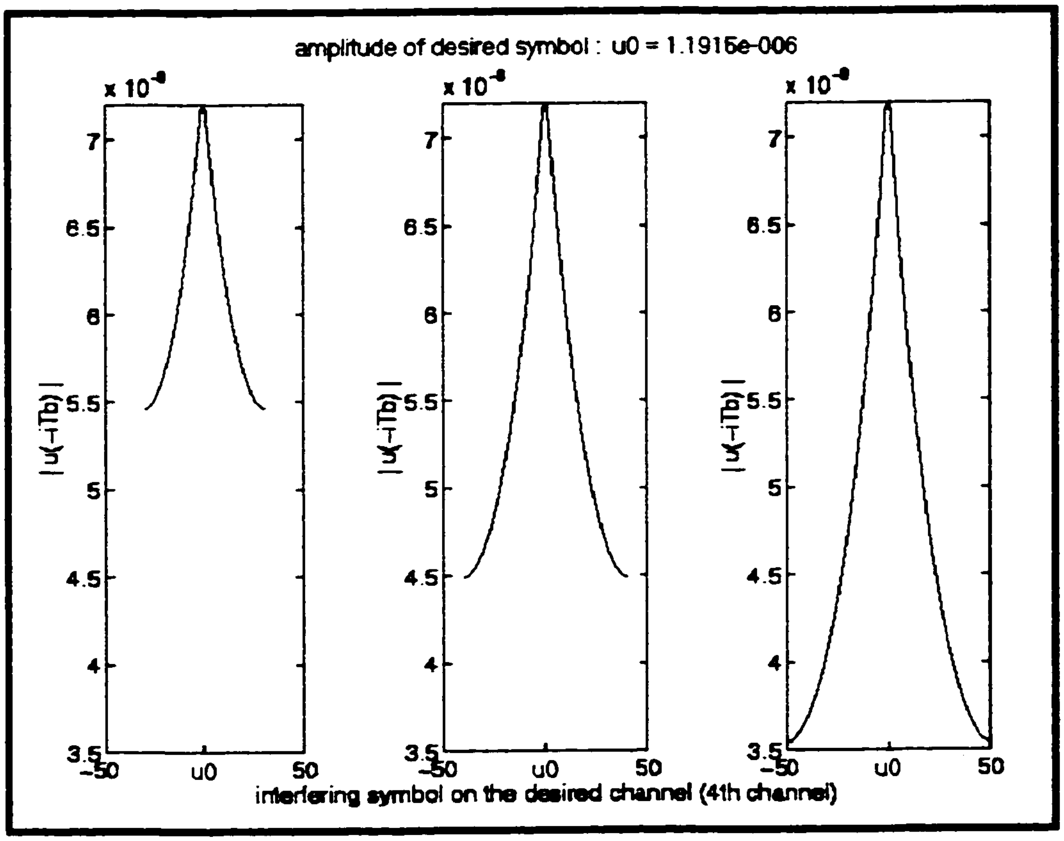


Figure 3-13. Decay of the ISI coefficients as *mL* and *mU* increase from 30 to 50.

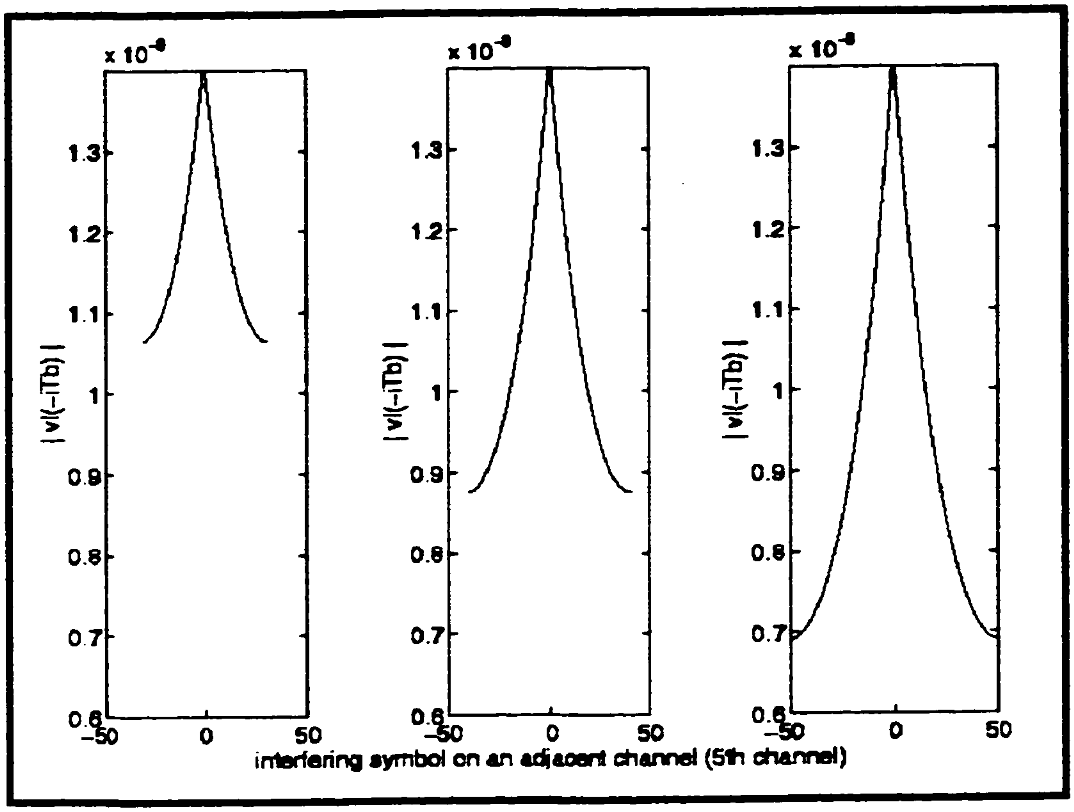


Figure 3-14. Decay of the in-phase ICI coefficients as *mL* and *mU* increase from 30 to 50.

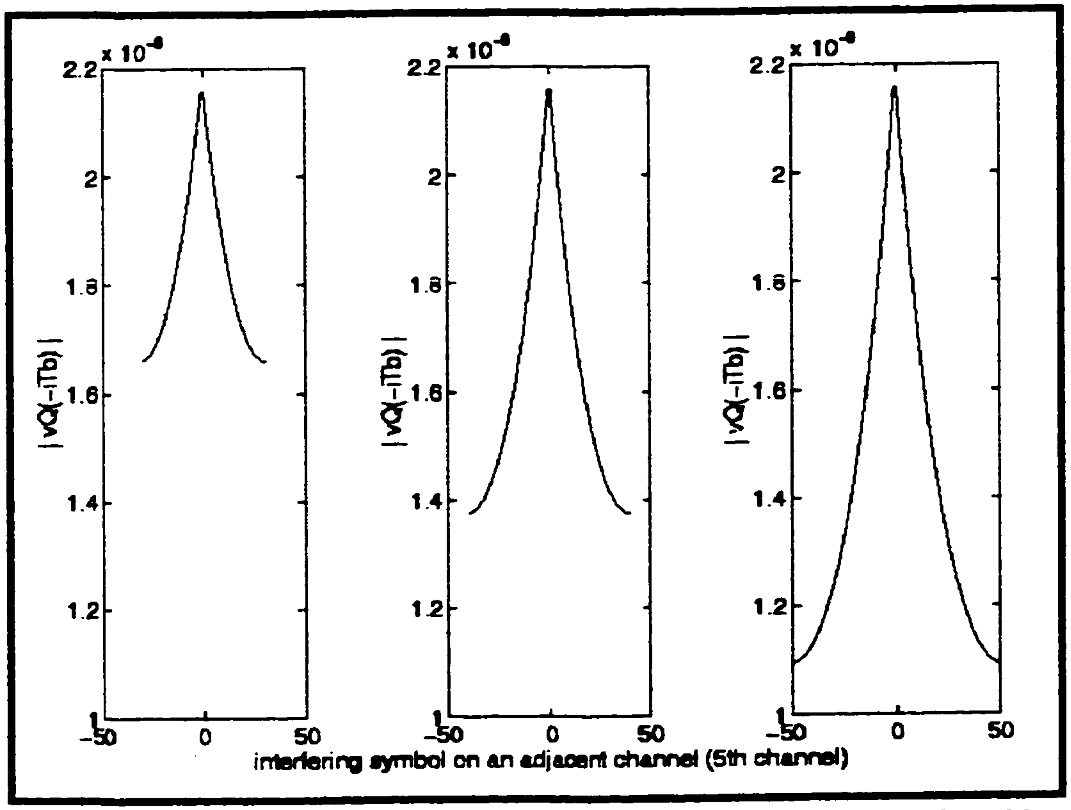


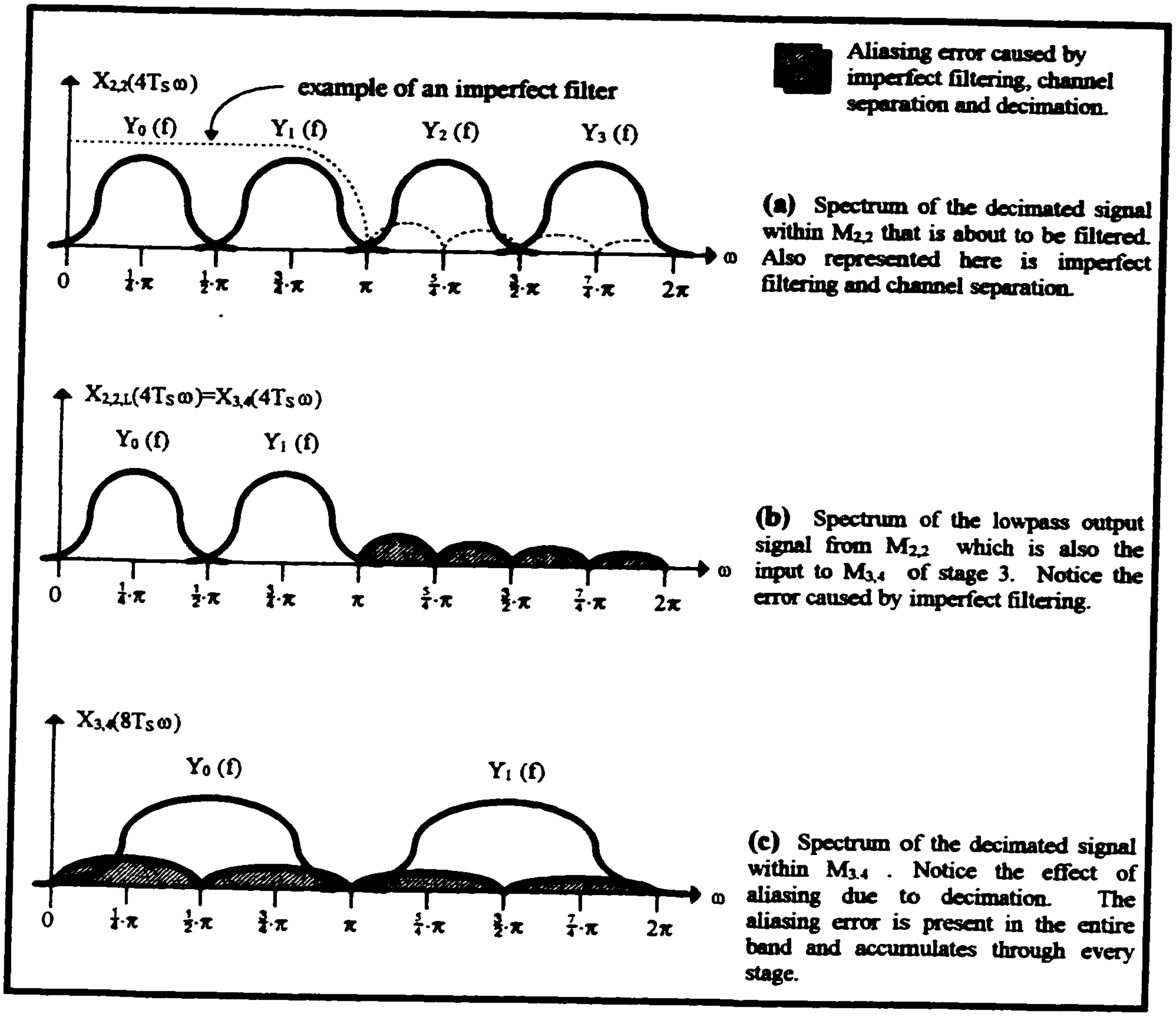
Figure 3-15. Decay of the quadrature ICI coefficients as *mL* and *mU* increase from 30 to 50.

From Figures 3-13 to 3-15, we see the large decay in value of the interfering symbols as *mL* and *mU* approach fifty. In fact, if in Figures 3-13 to 3-15, we were to increase *mL* and *mU* beyond fifty, we would see an even larger decay in the extreme interfering symbols which in effect would not contribute significantly to the overall interference. Since the value of the interference has a direct effect on the outcome of the BER curve, it can be seen why there is little difference between the case *mL=mU=30* and *mL=mU=50* in Figure 3-12 and why any value beyond fifty would give the same result. However, using a larger number of interfering symbols significantly increases the CPU time requirements since there are more iterations to go through in the algorithm to compute the total interference. Therefore, in choosing a proper number of interfering symbols to analyze the system performance of the M-MCDD, we must consider the trade-off that exists between the CPU time and the numerical accuracy, based on P_b and R_M, when the number of interfering symbols is increased. Thus, for the results that follow, the case of *mL=mU=10* was chosen since the results for both the BER and R_M for the scenarios investigated were reasonably accurate compared to higher *mL* and *mU* values while the CPU time required to compute such cases was fairly short. Despite the fact that in the case of *Ltap=9* and *Itap=9* it was seen that *mL=mU=30* was a wiser choice based on BER, R_M accuracy and CPU time requirements, for a majority of the remaining scenarios *mL=mU=10* was seen to be better for the same reasons.

3.4.2.3 : System Performance

The types of filter taps associated with the M-MCDD which will be used in its design are the non-zero filter taps per halfband filter (*Ltap*) and the channel detection filter taps (*Itap*). We will use various design tools to evaluate the system performance of the M-MCDD. First, we will look at BER curves showing the results of the performance analysis of the M-MCDD as *Ltap* and *Itap* vary, respectively, and how the CPU timing requirements of the GQR algorithm is dependent upon the number of filter taps selected. We will then look at the theoretical computational complexity of the M-MCDD as a function of *Ltap* and *Itap*. The computational complexity is based on analytical results of the ISI and ICI and those theoretical results will be used to verify the actual CPU time requirements of the GQR algorithm. We will also look at a set of isometric contours which will show the proper combinations of *Ltap* and *Itap* to achieve a specific performance and various degradation levels. The desired BER performance levels which will be looked at are 10^{-4}, 10^{-5}, and 10^{-6}. Using these design tools, illustrative examples will be given to show how we can select the proper number of filter taps to achieve the desired performance and degradation levels, relative to an ideal demodulation of QPSK signals, while maintaining an acceptable level of complexity.

Before discussing how these numerical results may be used to select the number of filter taps, a comment on the overall system performance will be made. Compared to the performance of previously analyzed structures, the M-MCDD exhibits more degraded performance [2] [6-7]. The M-MCDD is at a slight disadvantage in comparison to the SSM and PPM because the performance of the M-MCDD is affected by aliasing [2] [6]. Referring to the process shown by Figure 2-10, as the multicarrier signal is processed through each stage, the decimation process not only affects the portion of the spectrum in the interval $[0, \pi]$ but the part of the spectrum containing the error terms within $[\pi, 2\pi]$ is folded over on top of the desired spectrum [6] [25-26]. This aliasing effect is present in every stage of demultiplexing. As a result, the interference is literally added on top of itself through each stage therefore, this will tend to degrade the performance of the M-MCDD [6]. This aliasing effect is shown qualitatively by Figure 3-16, which demonstrates the process shown by Figure 2-10 also taking into consideration the aliasing error caused by imperfect filtering, channel separation and decimation [6] [26]. This aliasing effect may manifest itself quantitatively in the analysis of the M-MCDD through (3-15) and (3-16) showing the expansion of the $\cos(\cdot)$ and $\sin(\cdot)$ terms of the ICI expressed in (3-13) and (3-14), respectively. (3-15) and (3-16) contribute numerous summation terms to the ICI, cumulatively through each stage. This cumulative summation was not present in either the SSM nor the PPM. Thus, it may be responsible for the higher BERs seen for the M-MCDD compared to the SSM and PPM [2] [6].



▨ Aliasing error caused by imperfect filtering, channel separation and decimation.

(a) Spectrum of the decimated signal within $M_{2,2}$ that is about to be filtered. Also represented here is imperfect filtering and channel separation.

(b) Spectrum of the lowpass output signal from $M_{2,2}$ which is also the input to $M_{3,4}$ of stage 3. Notice the error caused by imperfect filtering.

(c) Spectrum of the decimated signal within $M_{3,4}$. Notice the effect of aliasing due to decimation. The aliasing error is present in the entire band and accumulates through every stage.

Figure 3-16(a-c). The effects of aliasing due to imperfect filtering and decimation [6] [26].

Numerical results of the BER and truncation error from the GQR algorithm for varying *Ltap* and *Itap* values are shown in Tables 3-7 and 3-8, respectively. Table 3-7 shows the variation in performance as *Ltap* increases from eleven to thirteen while keeping *Itap* constant at nine. Similarly, Table 3-8 shows the variation in performance as *Itap* increases from nine to twelve while keeping *Ltap* constant at nine. In all four cases, the number of terms in the GQR is shown for each value of SNR from 0dB to 15dB. Also, the truncation error is observed to be relatively unchanged in both tables although a slight decrease in truncation error can be observed, depending on the SNR, as *Ltap* and *Itap* increase for those particular cases. Between Tables 3-7 and 3-8, the only exception is for the higher SNRs, where we notice an increase in truncation error. This goes against our expectations, that is, for larger filter lengths, we expect a performance closer to ideal QPSK with truncation errors of smaller magnitude. This is especially true for very low BERs, that is generally, below 10^{-6}, however in these cases, we observe the opposite occurring. BER curves are given in Figures 3-17 and 3-18 showing a summary of the numerical results which are partially shown in Tables 3-7 and 3-8, respectively. Figure 3-17 shows the variation in performance as a function of *Ltap* keeping *Itap=9*. Likewise, Figure 3-18 shows the variation in performance as a function of *Itap* keeping *Ltap=9*. Each figure shows five different BER curves corresponding to different filter tap values.

Table 3-7. Variation in performance of the M-MCDD as *Ltap* increases keeping *Itap=9*.

		Ltap=11 and *Itap=9*		*Ltap=13* and *Itap=9*	
Eb/No (dB)	M	P_b	R_M	P_b	R_M
0	3	8.1610e-02	1.3034e-030	8.1120e-02	5.9466e-032
1	4	5.9519e-02	6.7748e-029	5.8981e-02	3.2579e-030
2	4	4.0827e-02	3.3487e-027	4.0272e-02	1.7208e-028
3	4	2.6026e-02	1.5539e-025	2.5496e-02	8.6804e-027
4	5	1.5209e-02	6.6590e-024	1.4745e-02	4.1321e-025
5	5	8.0239e-03	2.5816e-022	7.6597e-03	1.8286e-023
6	6	3.7594e-03	8.8227e-021	3.5074e-03	7.3823e-022
7	6	1.5381e-03	2.5725e-019	1.3876e-03	2.6548e-020
8	6	5.4090e-04	6.1422e-018	4.6479e-04	8.2541e-019
9	7	1.6134e-04	1.1402e-016	1.2936e-04	2.1365e-017
10	9	4.0490e-05	1.5421e-015	2.9472e-05	4.3911e-016
11	10	8.5430e-06	1.3997e-014	5.4499e-06	6.7498e-015
12	12	1.5296e-06	7.6902e-014	8.1904e-07	7.1981e-014
13	17	2.3729e-07	2.2456e-013	1.0141e-07	4.8444e-013
14	17	3.2954e-08	2.9592e-013	1.0635e-08	1.8263e-012
15	18	4.2764e-09	1.4321e-013	9.8479e-10	3.3197e-012

Table 3-8. Variation in performance of the M-MCDD as *Itap* increases keeping *Ltap=9*.

		Ltap=9 and *Itap=9*		*Ltap=9* and *Itap=12*	
Eb/No (dB)	M	P_b	R_M	P_b	R_M
0	3	8.2199e-02	2.4383e-029	8.1149e-02	4.6402e-032
1	4	6.0164e-02	1.1561e-027	5.9013e-02	2.5386e-030
2	4	4.1493e-02	5.0909e-026	4.0305e-02	1.3385e-028
3	4	2.6666e-02	2.0423e-024	2.5527e-02	6.7369e-027
4	5	1.5772e-02	7.2871e-023	1.4772e-02	3.1980e-025
5	5	8.4710e-03	2.2431e-021	7.6809e-03	1.4102e-023
6	6	4.0742e-03	5.7341e-020	3.5220e-03	5.6681e-022
7	6	1.7315e-03	1.1600e-018	1.3962e-03	2.0270e-020
8	6	6.4279e-04	1.7481e-017	4.6905e-04	6.2581e-019
9	7	2.0685e-04	1.8182e-016	1.3110e-04	1.6056e-017
10	9	5.7597e-05	1.1857e-015	3.0047e-05	3.2633e-016
11	10	1.3961e-05	4.2967e-015	5.6023e-06	4.9463e-015
12	12	2.9918e-06	7.4299e-015	8.5129e-07	5.1825e-014
13	17	5.8146e-07	5.0624e-015	1.0693e-07	3.4111e-013
14	17	1.0608e-07	1.0678e-015	1.1420e-08	1.2505e-012
15	18	1.8919e-08	5.1478e-017	1.0811e-09	2.1943e-012

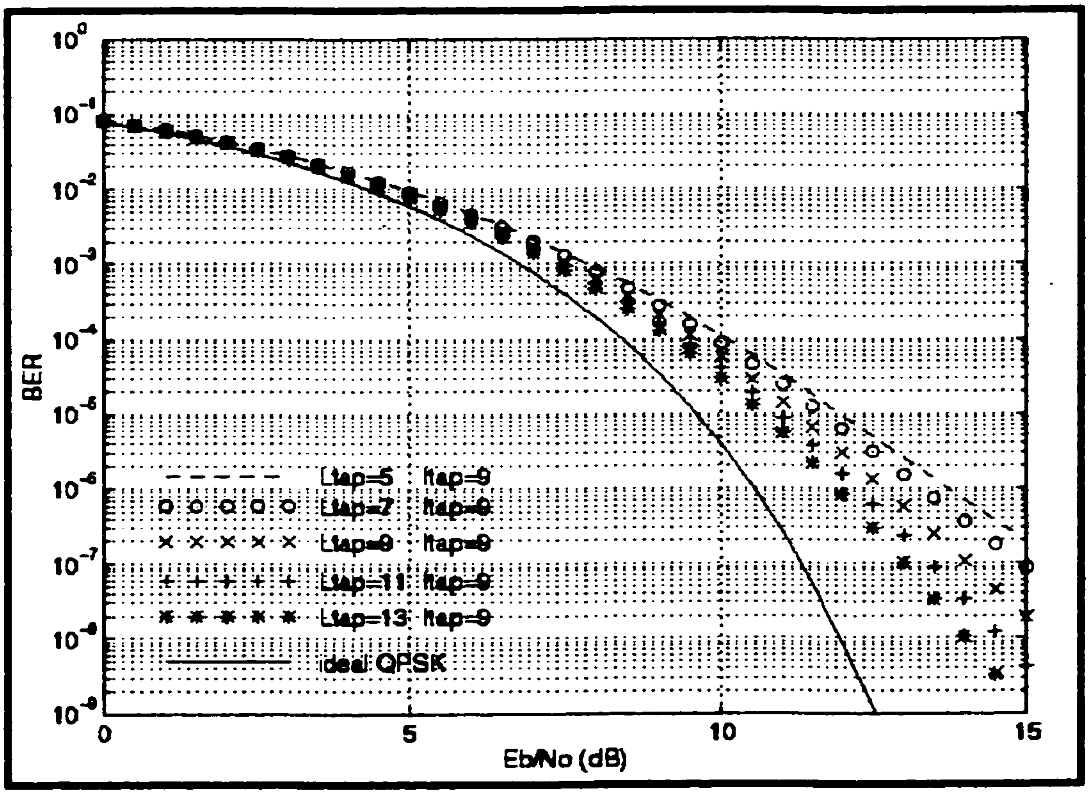


Figure 3-17. Performance of the M-MCDD as a function of *Ltap* for *Itap=9*.

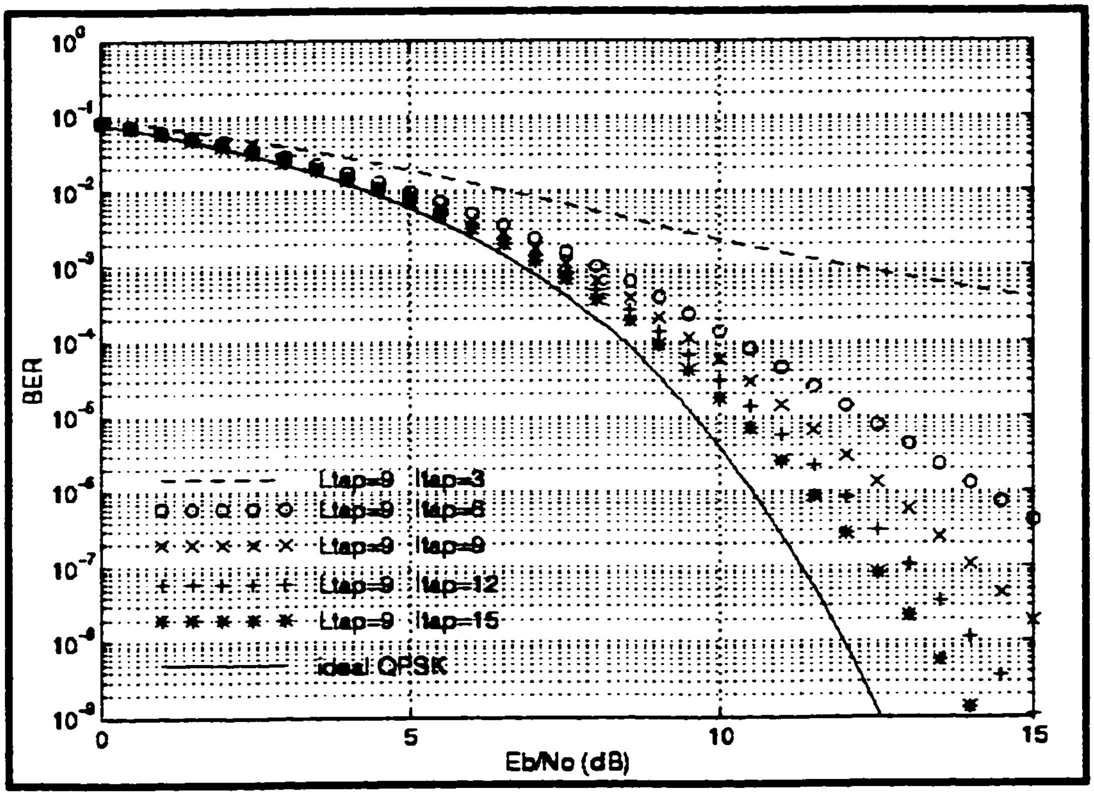


Figure 3-18. Performance of the M-MCDD as a function of *Itap* for *Ltap=9*.

The trade-off seen in the selection of the number of filter taps from Figures 3-17 and 3-18 is that if we wish the performance of the M-MCDD to converge to that of ideal QPSK demodulation, we must increase the filter complexity. That is to say, if we require more accurate results, we need to increase filter complexity which consequently increases the overall computational complexity of the entire M-MCDD. The computational complexity will be shown numerically later in this section. In terms of the execution of the GQR algorithm, an increased filter complexity means more CPU time and power to process. This will also be shown later on.

As for the selection of *Ltap* and *Itap*, the first observation from Figures 3-17 and 3-18 is that although for most of the BER curves the improvement is gradual, as the number of filter taps increase, we see a clear disadvantage in using *Itap=3* channel detection filter taps. Clearly, the degradation in performance associated with *Itap=3* is too great and consequently, one should avoid using this small a number of filter taps when designing the channel detection filter. A more general observation from Figures 3-17 and 3-18 is that an increase in *Itap* resulted in a more significant improvement in performance compared to an increase in *Ltap*. To demonstrate this, Table 3-9 has been compiled based on the data given in Tables 3-7 and 3-8. It shows the relative improvement in performance in dB as *Ltap* and *Itap* increase, respectively. To obtain the specific P_b points in Table 3-9, linear interpolation was used in Tables 3-7 and 3-8 [34].

Table 3-9. Improvement in performance in dB associated with increasing *Ltap* and *Itap* [34].

	Ltap=11 → *Ltap=13* *Itap=9*	*Ltap=9* *Itap=9* → *Itap=12*
$P_b = 10^{-3}$	0.10	0.23
$P_b = 10^{-4}$	0.20	0.42
$P_b = 10^{-5}$	0.25	0.54
$P_b = 10^{-6}$	0.44	0.86
$P_b = 10^{-7}$	0.68	1.00

From Table 3-9, we observe that for a fixed performance level, an increase in *Itap* results in almost double the improvement in dB as does the equivalent increase in *Ltap*. One way to explain this observation in terms of the computational algorithm is that the summation through *Itap* is an outer loop in the algorithms while the summation through *Ltap* is an inner loop, referring to equations (3-8) and (3-9), for example. To further support this observation, we see a larger improvement in performance when increasing between *Itap=3* and *Itap=15* for *Ltap=9* than by increasing between *Ltap=5* and *Ltap=13* for *Itap=9*, referring to Figures 3-18 and 3-17, respectively. Thus, we see a clear advantage if we increase *Itap* over *Ltap*, that is, a larger improvement in performance of the M-MCDD.

There is, however, a disadvantage associated with increasing *Itap*. From Table 3-10 and Figure 3-19, we can compare the increase in CPU time requirements for the GQR algorithm as *Ltap* and *Itap* increase individually. Clearly, there is a larger increase in CPU time when *Itap* is increased than when *Ltap* is increased. Therefore, the disadvantage of increasing *Itap* over *Ltap* is a more substantial increase in CPU time requirements for the GQR algorithm to execute. The reason for this occurrence, again, is due to the summation of *Itap* being an outer loop while that of *Ltap* is an inner loop. We would therefore expect more iterations or computations if *Itap* were to increase as opposed to *Ltap*. This phenomenon brings us back to the trade-off that was previously stated, that is, performance versus complexity. Although this applies for an increase of *Ltap* and/or *Itap*, it is more accentuated as *Itap* increases.

Table 3-10. CPU time requirements for various *Ltap* and *Itap* values.

Ltap	Itap	CPU time (min : sec)	Ltap	Itap	CPU time (min : sec)
5	9	13 : 26.3333	9	3	08 : 58.5167
7	9	14 : 17.1167	9	6	13 : 05.1833
9	9	17 : 31.2667	9	9	17 : 31.2667
11	9	22 : 34.2000	9	12	25 : 20.3333
13	9	25 : 24.2500	9	15	29 : 00.4000

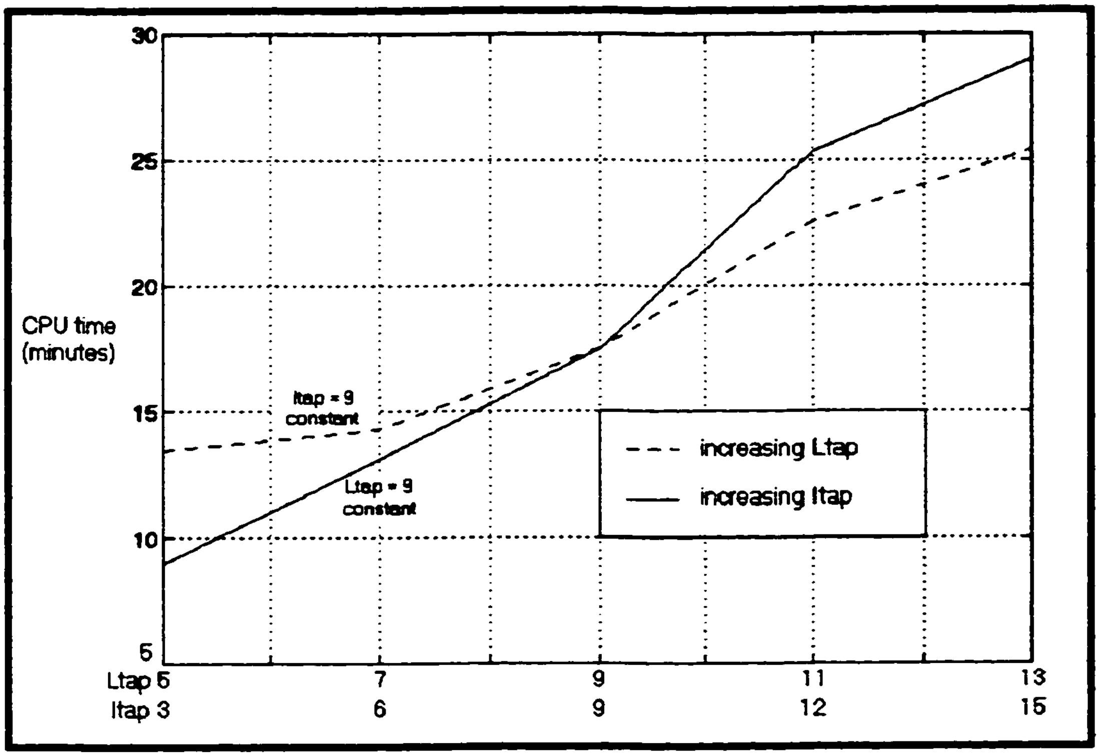


Figure 3-19. Comparison of CPU time for the GQR algorithm as *Ltap* and *Itap* vary.

This increase in CPU time can be verified by looking at the theoretical computational complexity. The computational complexity, which is based on the number of operations, that is the total number of additions and multiplications, required to compute the ISI and ICI, referring to (3-8), (3-13) and (3-14), respectively [6]. In general, we can analyze the complexity to determine the total number of operations required to compute the interference. So given the fact that *mL=mU=10* and *l*=3 were fixed for the performance of the M-MCDD, the complexity can be determined as a function of *Ltap* and *Itap*. Thus, the total number of operations required to compute the interference has been determined to be (see Appendix A3.5 for more detail) :

$$M_{op} = (36{,}435 \cdot Itap \cdot Ltap) - (46{,}845 \cdot Itap) - 314 \tag{3-70}$$

The complexity of the M-MCDD of (3-70), based on different *Ltap* and *Itap* values, is shown in Table 3-11 and summarized graphically in Figure 3-20. If we compare the computational complexity of Figure 3-20 to the CPU time requirements of Figure 3-19, we see that they are very close in shape. This similarity between both sets of curves should serve to verify both results. The computational complexity of the M-MCDD, based on the computation of the interference, is indeed comparable to the CPU timing requirements for the GQR algorithm since a majority of the computation time of the GQR algorithm goes towards computing the ISI and ICI. Using these values for the computational complexity, we can determine how much complexity we are willing to endure for the possible values of *Ltap* and *Itap* given in the isometric contours.

Table 3-11. Computational complexity for various *Ltap* and *Itap* values.

Ltap	*Itap*	number of operations (M_{op})	*Ltap*	*Itap*	number of operations (M_{op})
5	9	1,217,656	9	3	842,896
7	9	1,873,486	9	6	1,686,106
9	9	2,529,316	9	9	2,529,316
11	9	3,185,146	9	12	3,372,526
13	9	3,840,976	9	15	4,215,736

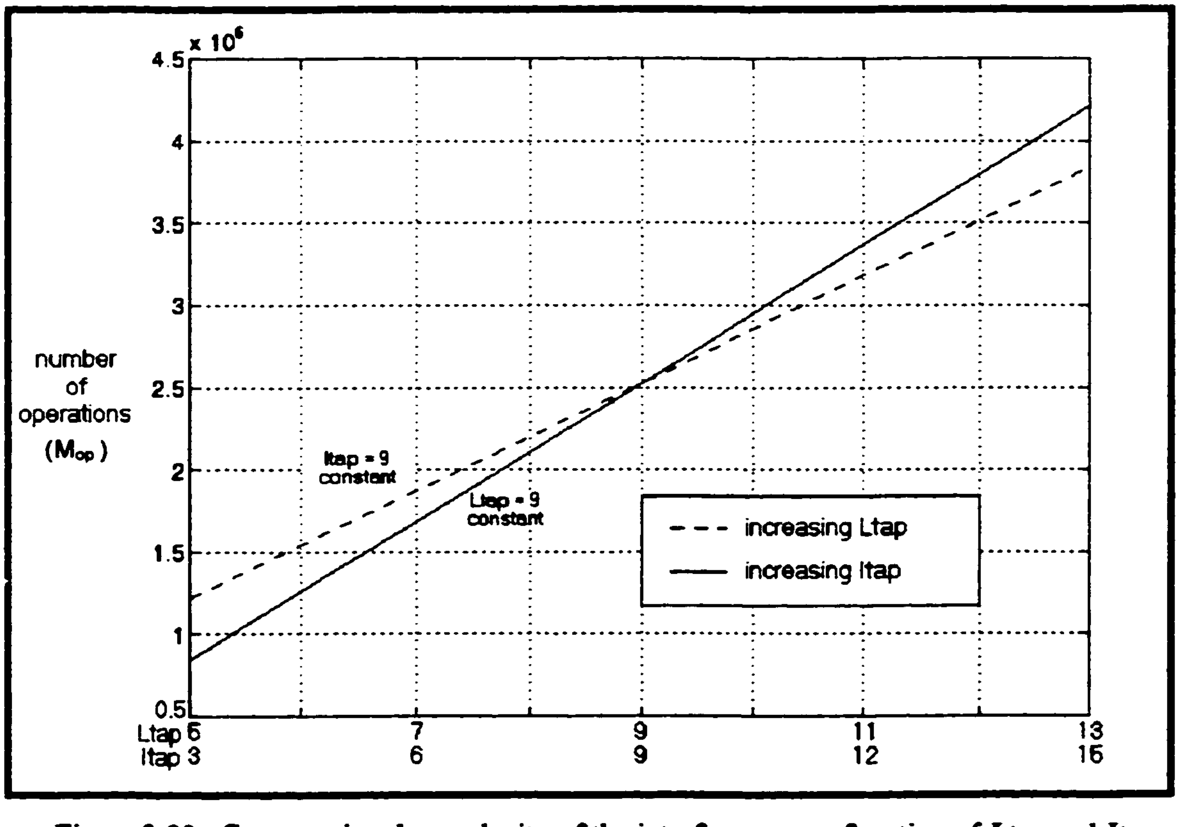


Figure 3-20. Computational complexity of the interference as a function of *Ltap* and *Itap*.

We will now use the isometric BER × SNR contours to interpret the results from Tables 3-7 and 3-8 and Figures 3-17 and 3-18, respectively, in order to effectively choose the values of *Ltap* and *Itap* based on desired performance and degradation levels. It should be noted that the isometric contours summarize data from many more scenarios than what is shown by Tables 3-7 and 3-8 and Figures 3-17 and 3-18. Figures 3-21 through 3-23 show three examples of isometric contours for different BER performance levels. Table 3-12 shows the same data but in numerical form.

Table 3-12. Values of *Ltap* and *Itap* to achieve performance and degradation level.

(*Ltap*,*Itap*)	1 dB degradation	1.5 dB degradation	2 dB degradation
$P_b = 10^{-4}$	(11,9) (9,12) (7,15)	(11,6) (7,9) (5,12)	(5,6)
$P_b = 10^{-5}$	(13,9) (11,12) (9,15)	(13,6) (9,9) (7,12)	(9,6) (5,9)
$P_b = 10^{-6}$	(13,12) (11,15)	(11,9) (9,12)	(11,6) (7,9)

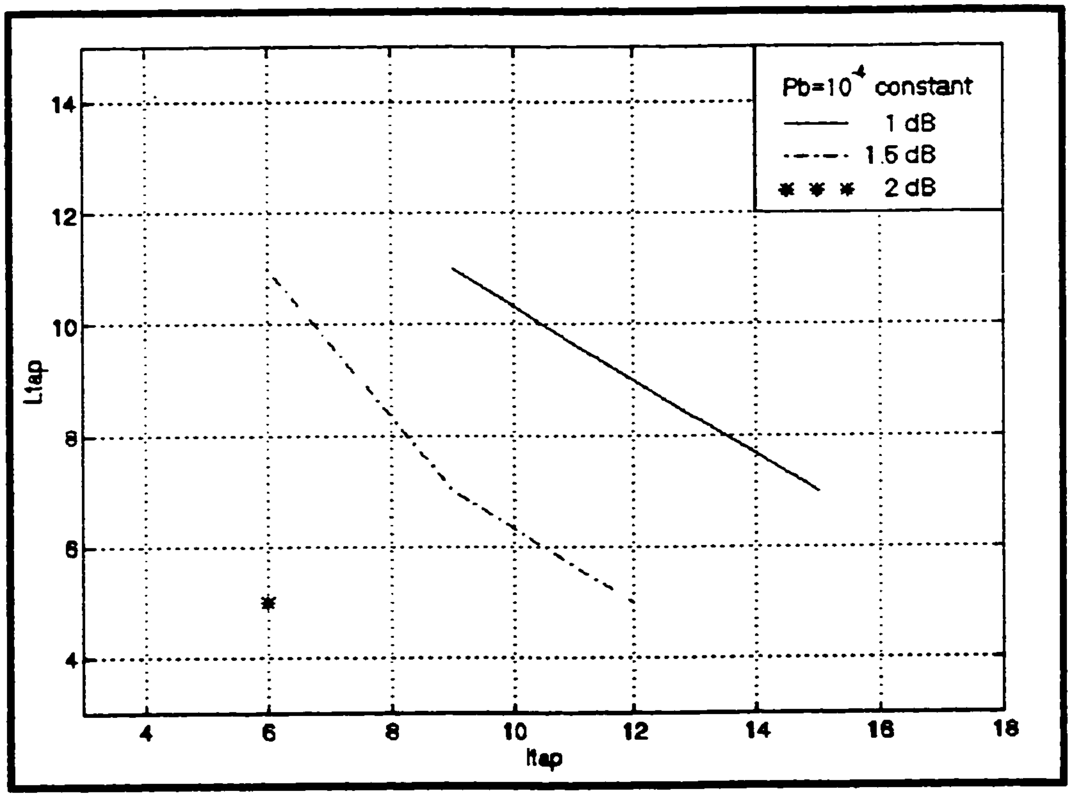


Figure 3-21. Isometric contour for $P_b=10^{-4}$ and various degradation levels.

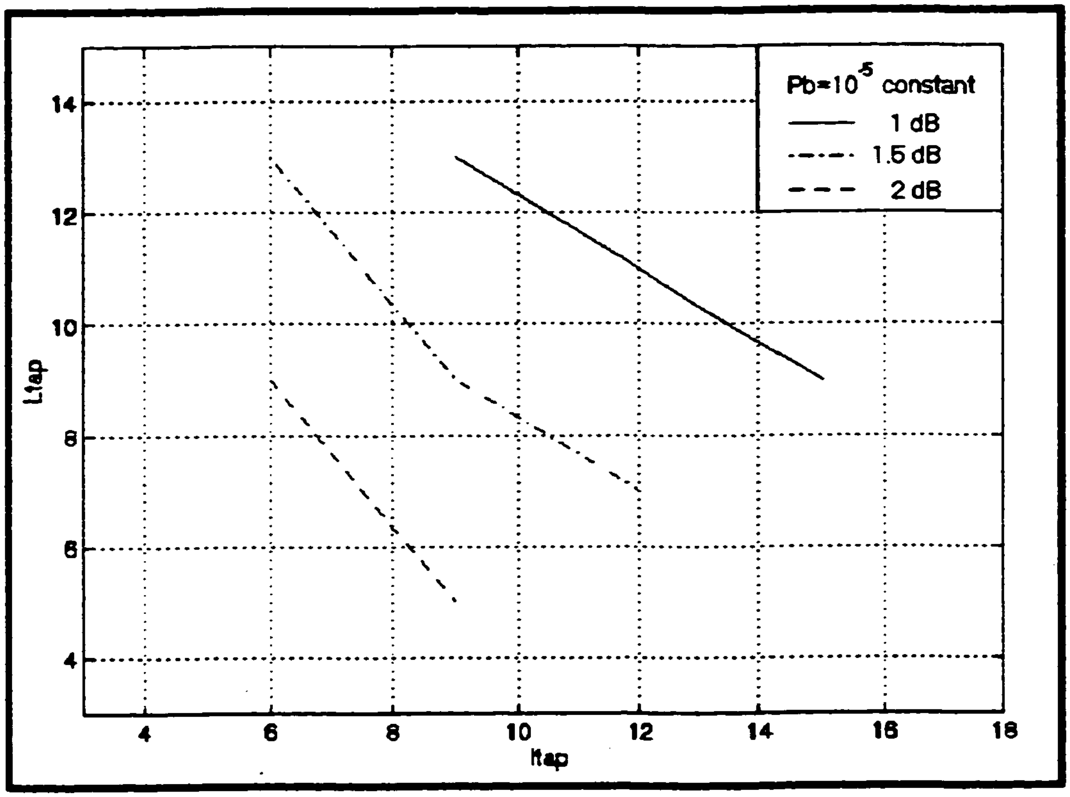


Figure 3-22. Isometric contour for $P_b=10^{-5}$ and various degradation levels.

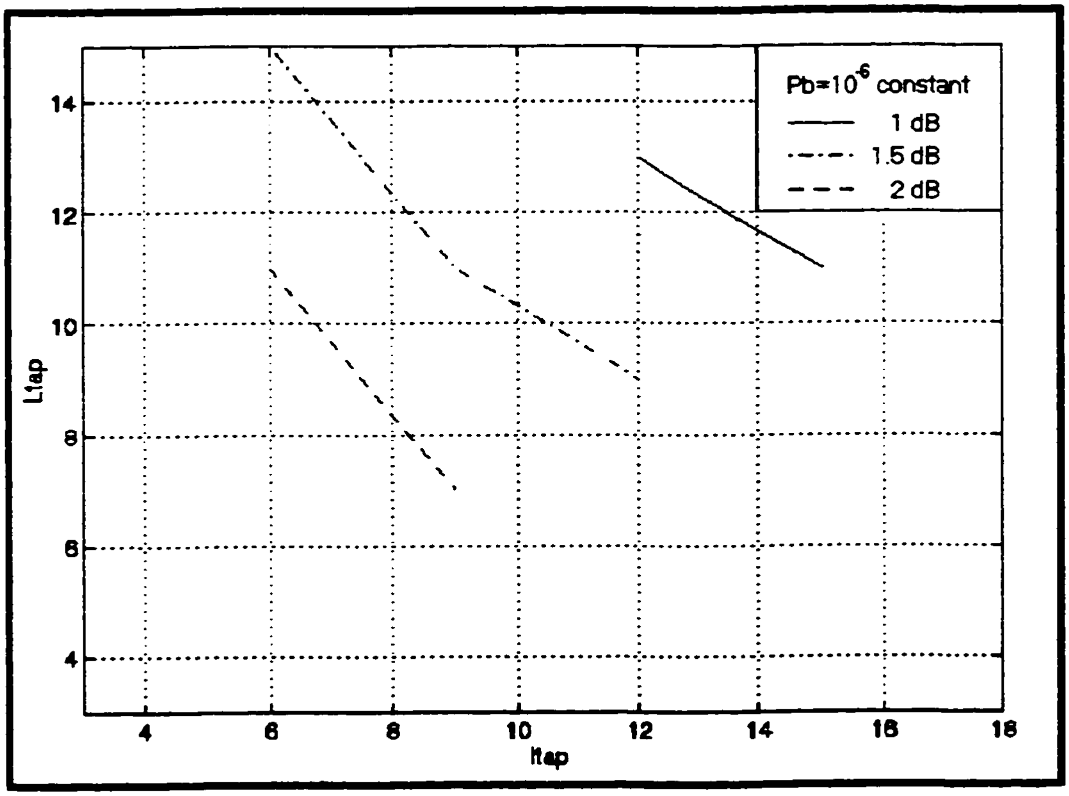


Figure 3-23. Isometric contour for $P_b=10^{-6}$ and various degradation levels.

The manner in which these isometric contours were compiled was basically the same as how the data for Table 3-9 was obtained, that is, by observing performance analysis results for many different scenarios. The difference is that in these figures, the amount of degradation in dB was measured by observation from the BER curves, such as those shown by Figures 3-17 and 3-18, whereas the amount of improvement in dB, shown in Table 3-9, was determined by linear interpolation of the numerical results of Tables 3-7 and 3-8 [34]. Another important difference is that Table 3-9 shows the relative improvement in performance between two specific M-MCDD scenarios while the isometric contours show the amount of degradation relative to ideal QPSK demodulation.

The isometric contours will tell us what choices of *Ltap* and *Itap*, as a pair of values, are available to satisfy the design criteria. What remains to be seen is how one can use these isometric contours along with the other results to specifically choose *Ltap* and *Itap* among several possible pairs. If we wish to suffer less degradation in performance compared to ideal QPSK demodulation, it is better to keep *Itap* larger than *Ltap*. On the other hand, keeping *Itap* larger than *Ltap* will cause a larger computational complexity of the M-MCDD. The trade-off here in choosing *Ltap* and *Itap* is performance and/or degradation improvement versus complexity. The difference in increasing *Itap* over *Ltap* is that an increase in *Itap* has a greater effect on the performance and complexity as opposed to an increase in *Ltap*. Thus, this trade-off is more profound with an increase in *Itap*.

Some illustrative examples will now be provided on how to choose *Ltap* and *Itap* taking into consideration this trade-off that exists.

Example 3-1.

Our objective is to achieve the best possible performance level. We specify a degradation level of 1.5dB (Δ = 1.5dB) and we are willing to endure a complexity in the range of $2\cdot10^6$ to $2.5\cdot10^6$ operations ($2\cdot10^6 < M_{op} < 2.5\cdot10^6$). Table 3-13 below summarizes the result of a search through Table 3-12 for the possible combinations. Also, (3-70) has been used to calculate the complexity.

Table 3-13. Possible *Ltap* and *Itap* combinations to satisfy conditions in Example 3-1.

	(*Ltap*,*Itap*)	M_{op}
$P_b = 10^{-4}$	(11,6)	2,123,326
$P_b = 10^{-5}$	(7,12)	2,498,086

So, if we wish to achieve the best performance level, clearly our choice is (7,12). However, from Table 3-13, we see that (11,6) gives us a lower complexity but at a worse performance level. This shows the trade-off that exists between performance and complexity.

Example 3-2.

In the next example, our objective is to achieve the lowest possible degradation level. We specify a performance level of 10^{-5} ($P_b = 10^{-5}$) and we are willing to endure a complexity in the range of $1.5 \cdot 10^6$ to $2.5 \cdot 10^6$ operations ($1.5 \cdot 10^6 < M_{op} < 2.5 \cdot 10^6$). Table 3-14 summarizes the possibilities.

Table 3-14. Possible *Ltap* and *Itap* combinations to satisfy conditions in Example 3-2.

	(*Ltap*,*Itap*)	M_{op}
Δ=1.5dB	(7,12)	2,498,086
Δ=2.0dB	(9,6)	1,686,106

From Table 3-14, (7,12) will give us the lowest level of degradation however, (9,6) gives us a much lower complexity level at a higher degradation level. This shows the trade-off between degradation and complexity.

Example 3-3.

This final example will be somewhat more involved. Here, we will try to achieve the best possible performance and/or degradation levels with the lowest possible complexity. We are looking for performance levels between 10^{-5} and 10^{-6} ($10^{-5} < P_b < 10^{-6}$) and degradation levels between 1dB and 1.5dB (1dB$<\Delta<$1.5dB). We also specify a maximum complexity level of $4 \cdot 10^6$ operations ($M_{op} < 4 \cdot 10^6$), which is quite a loose requirement.

First, we try to see if we can satisfy the best performance *and* degradation levels according to the specifications. Table 3-15 shows the complexity levels for the combinations satisfying a performance level of $P_b = 10^{-6}$ and a degradation level of Δ=1dB.

Table 3-15. *Ltap* and *Itap* combinations to satisfy best performance and degradation levels.

	(Ltap,Itap)	M_{op}
$P_b = 10^{-6}$, $\Delta = 1.0$dB	(13,12)	5,121,406
$P_b = 10^{-6}$, $\Delta = 1.0$dB	(11,15)	5,308,786

From Table 3-15, we see that the complexities are larger than $M_{op} = 4 \cdot 10^6$ operations so this does not satisfy our design criteria. We therefore have to look at the best possible performance *or* degradation levels, that is, $P_b = 10^{-6}$ *or* $\Delta = 1.0$dB, respectively. Table 3-16 summarizes the possible scenarios that meet our specifications.

Table 3-16. *Ltap* and *Itap* combinations to satisfy best performance or degradation levels.

Bolded values indicate the criteria of the problem satisfied		
	(Ltap,Itap)	M_{op}
$P_b = 10^{-5}$, **$\Delta = 1.0$dB**	(13,9)	3,840,976
$P_b = 10^{-6}$, $\Delta = 1.5$dB	(11,9)	3,185,146
$P_b = 10^{-6}$, $\Delta = 1.5$dB	(9,12)	3,372,526

From Table 3-16, we see that (13,9) satisfies the lowest degradation level and (11,9) best satisfies $P_b = 10^{-6}$ as the desired performance level due to its lower complexity. However, if our objective is to achieve the lowest complexity of the two, we choose the (11,9) scenario which, at the same time, achieves the best possible performance level.

3.4.2.4 : Summary of the Numerical Results

The numerical results for the performance analysis was presented on two fronts. On one hand, we presented the numerical accuracy of the results. The accuracy was shown by observing the results of the GQR technique and those of the Beaulieu FSE technique [15-16]. The BER and truncation errors from both techniques were observed. The BER curves were very close in value. Looking at the truncation errors, that of the GQR becams exponentially lower in value with SNR while that of the Beaulieu FSE remained at a high level. We preferred the GQR technique over the Beaulieu FSE technique based on its relative speed of execution. It was found that the GQR technique took significantly less CPU time than the Beaulieu FSE.

Following this, we presented the system performance of the M-MCDD based on the BER curves generated by varying *Ltap* and *Itap*. As a first point, it was noted that the M-MCDD was at a slight disadvantage to the SSM and PPM because, unlike these previously analyzed structures, the performance of the M-MCDD was affected by aliasing. Therefore, in comparison to the SSM and PPM, the performance of the M-MCDD is more degraded [2] [6-7] [25-26]. Although the analysis did not directly take into account the effect of aliasing, as shown in Figure 3-16, we saw from (3-15) and (3-16) numerous summation terms resulting from the stages of demultiplexing. These summation terms were not present in previous analytical work on the SSM and PPM [1] [2]. Thus, our discussion included how this in fact may have been where the aliasing affected the analysis however, this cannot be proven.

Also within this section, it was shown how the number of filter taps could be chosen based on desired performance and degradation levels, relative to ideal QPSK demodulation, and maximum allowable computational complexity. To facilitate the choice of *Ltap* and *Itap* using this criteria, the information in the BER curves was compiled and summarized using BER×SNR isometric contours. These graphs show the proper choices for *Ltap* and *Itap* to achieve the desired performance and degradation levels. Also, a table and graph of the computational complexity of the GQR algorithm was shown. It was based on the total number of additions and multiplications needed to compute the ISI and ICI. From the results, we saw a trade-off in selecting the values of *Ltap* and *Itap*. To improve the performance relative to ideal QPSK demodulation, we are better off increasing *Itap* since we can see a larger improvement in performance by increasing *Itap* compared to a corresponding increase in *Ltap*. If we do decide to increase *Itap* over *Ltap*, however, we would be introducing complexity into the system at a faster rate. The trade-off between performance and complexity actually exists for varying *Itap* as well as *Ltap* values however, increasing *Itap* has a greater effect on performance and complexity than does an increase in *Ltap*. To summarize, the value of *Itap* and *Ltap* should be chosen so as to satisfy the performance and/or degradation requirements, with respect to ideal QPSK demodulation, while not exceeding the maximum allowable computational complexity.

Practically speaking, to choose the values of *Ltap* and *Itap*, we can refer to the methodology we used in the examples presented at the end of this section. A designer should specify what is the design criteria desired, that is, worse case performance, degradation levels and/or maximum complexity levels. At the same time, the objective or intent of the design should be made clear. This means that there must be one criteria that is a priority in satisfying over the others. For example, a designer should be clear in specifying the objective, be it the best possible performance level, degradation level and/or lowest complexity level. Specifying one design objective will usually result in one or more possible scenarios that can satisfy the situation, however, the possibilities are more limited if more than one design objective is specified. Thus, based on the design intent and the worse case criteria specified, we may refer to the numerical data and choose the possibilities for the design. Our final choice would ultimately be based on which scenario achieves the better design criteria.

Chapter Four

Effects of Filter Coefficient Quantization

4.1 : Model of Halfband Filter With Quantized Coefficients

To introduce the model used for the quantized halfband filter coefficients, we first look at the discrete Fourier Transform (DFT) for the halfband filter. It is [9]:

$$H(j\omega) = \sum_{n=-L}^{L} h(nT_s) \cdot e^{-j\omega n} \qquad (4\text{-}1)$$

where $h(nT_s\ )$ has been defined as the halfband filter impulse response [27]. Now suppose that the coefficients of $h(nT_s\ )$ are quantized, resulting in a new set of coefficients $\hat{h}(nT_s)$ defined by:

$$\hat{h}(nT_s) = h(nT_s) + \tilde{h}(nT_s) \qquad (4\text{-}2)$$

The DFT for the quantized system is [9]:

$$\hat{H}(j\omega) = \sum_{n=-L}^{L} \hat{h}(nT_s) \cdot e^{-j\omega n} \Big|_{\hat{h}(nT_s)\ non-random} = H(j\omega) + \tilde{H}(j\omega) \qquad (4\text{-}3)$$

where:

$$\tilde{H}(j\omega) = \sum_{n=-L}^{L} \tilde{h}(nT_s) \cdot e^{-j\omega n} \Big|_{\tilde{h}(nT_s)\ non-random} \qquad (4\text{-}4)$$

Here, the quantity $\tilde{h}(nT_s)$ represents the sequence of *quantization error random variables* with zero mean and variance $\Delta^2\ /12$, where Δ represents the quantization step size. Its corresponding system function is represented by the DFT $\tilde{H}(j\omega)$, defined by (4-4), assuming of course that $\tilde{h}(nT_s)$ is non-random [9] [31]. The set $\tilde{h}(nT_s)$ is sampled from the quantization error random process $\tilde{h}(t)$ whose model is assumed to have the following properties:

1. The quantization error sequence $\tilde{h}(nT_s)$ is a sampled from a stationary random process, $\tilde{h}(t)$.
2. $\tilde{h}(nT_s)$ is uncorrelated with the unquantized halfband filter sequence, $h(nT_s)$.

3. The random variables of the quantization error process are uncorrelated. The quantization error is thus a *white-noise process*.
4. The probability distribution of the quantization error process is *uniform* over the range of quantization error. Thus, the quantization error is also a *stationary process* [31].

The frequency response of the quantized system is linearly related to the quantization errors in the impulse response. Thus, the quantized system may be represented by Figure 4-2, which shows the unquantized system in parallel with the quantization error system [9] [14]. If B-bits of quantization are used to represent the halfband filter, we will need $(B+1)$ length words (including the sign bit). In that case, $\tilde{h}(nT_s)$ can be represented by a *uniform distribution* with zero mean and variance $2^{-2B}/12$, as illustrated by Figure 4-1 [9] [31].

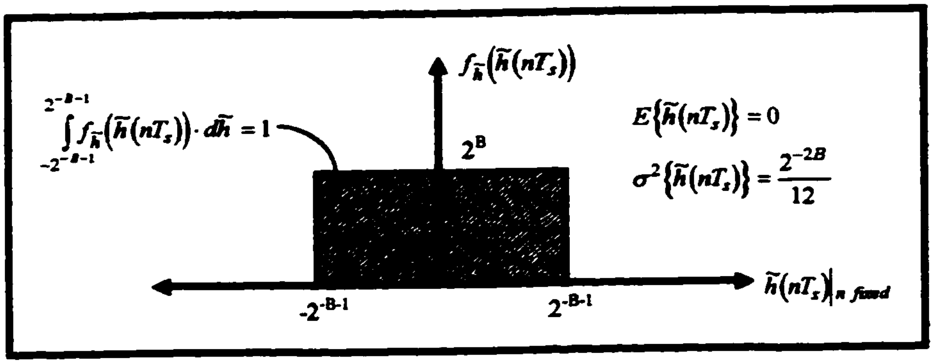


Figure 4-1. Distribution of the quantization noise process assuming B-bits of quantization [9] [31].

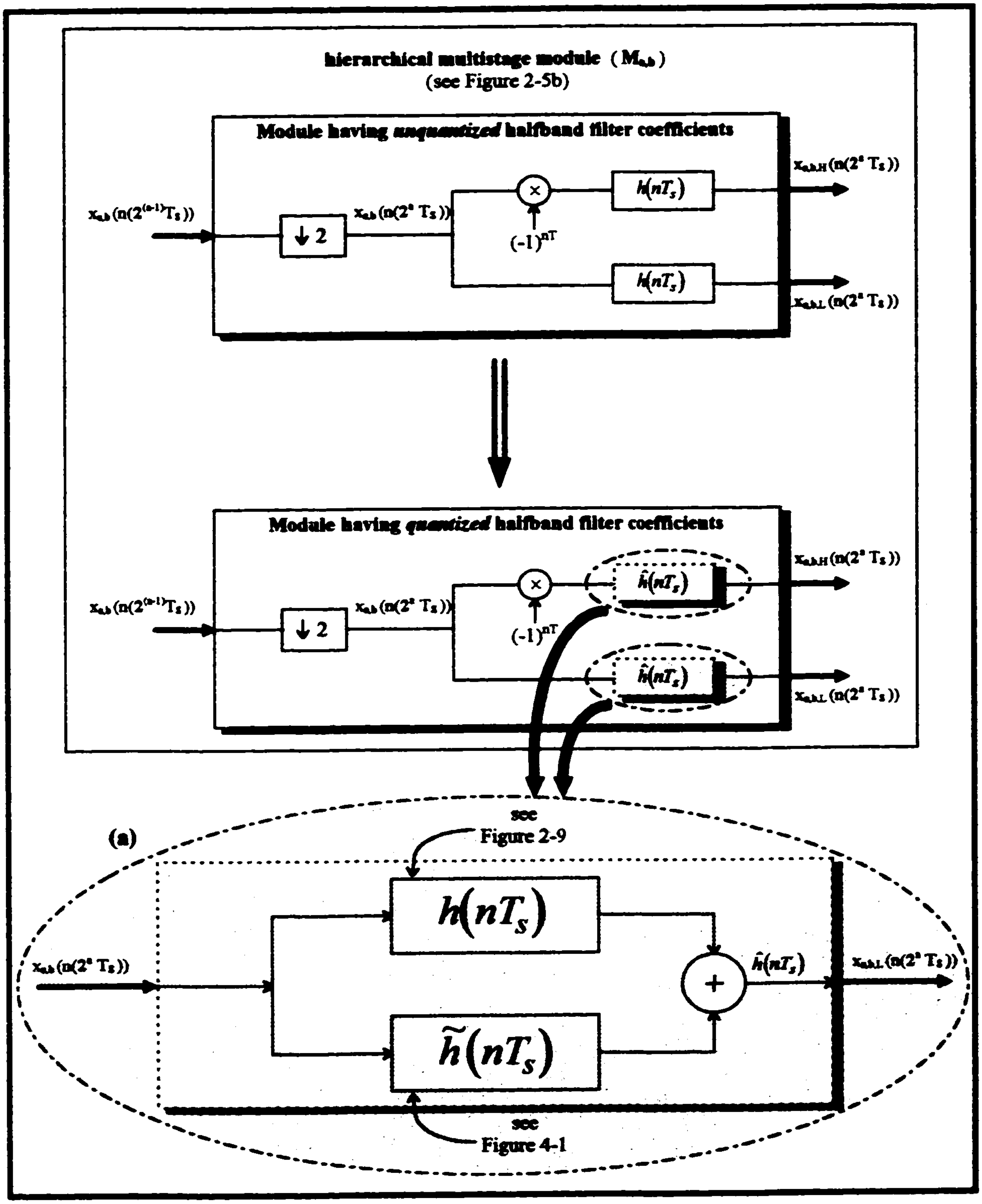


Figure 4-2. The transition from real to quantized coefficients for $h(nT_s)$ in the hierarchical module (a). Model for the quantized halfband filter coefficients used in the analysis. [1] [6] [9] [14]

4.2 : Analysis of Quantization for the M-MCDD Structure

4.2.1 : Output Signal from Channel Detection Filter

In this situation, the halfband filter coefficients are quantized thus replaced by the system represented by Figure 4-2 while the input MF-TDMA signal is kept the same as is indicated by (3-1) and (3-2). So, after the channels are demultiplexed, the output from the *channel detection* filtering stage assuming quantized halfband filter coefficients, is [2] [14] [25-26]:

$$\hat{r}_k(lT_b) = \hat{r}_k[(\beta_l + \mu_l)T_c] = \sum_{\xi = I_1}^{I_2} \hat{y}_k[(\beta_l - \xi)T_c] \cdot g[(\xi + \mu_l)T_c] \tag{4-5}$$

where $g(t)$ is the impulse response of the channel detection filter and β_l and μ_l are the *basepoint index* and the *fractional interval*, defined by (3-4) and (3-5), respectively [2]. The output of the channel detection filter stage is quantized since its input, $\hat{y}_k(nT_c)$, which is the k^{th} channel output signal from the M-MCDD, is also quantized due to filtering through the quantized halfband filters [9].

The first step in analyzing the M-MCDD structure in the presence of quantization noise is to derive the form of the k^{th} desired output channel. If we follow the same procedure as was done to derive (3-6) further assuming that the spectral characteristics of the DAF approximate those of $\hat{h}(nT_s)$ and the quantized halfband filter coefficient processes at every stage, $\hat{h}_j(n_j \cdot 2^j T_s) \mid \forall j, j = 0,1,\cdots,l$, have the same stochastic characteristics, we obtain [14] [31]:

$$\hat{y}_k(mT_c) = \begin{cases} (-1)^m \displaystyle\sum_{n_0=-L}^{L} \cdots\cdots \sum_{n_l=-L}^{L} \left\{ \begin{array}{l} \displaystyle\prod_{j=0}^{l} \left(\hat{h}_j(n_j \cdot 2^j T_s) \right) \\ \cdot x((2^l m - \displaystyle\sum_{j=0}^{l} 2^j n_j)T_s) \cdot e^{j\pi(k+\frac{1}{2}) \cdot \sum_{j=0}^{l} n_j} \end{array} \right\} & ; \begin{array}{l} k = 0\ , \\ k \textit{ even} \end{array} \\ (-1)^m \displaystyle\sum_{n_0=-L}^{L} \cdots\cdots \sum_{n_l=-L}^{L} \left\{ \begin{array}{l} (-1)^{n_0 \cdot T_s} \cdot \displaystyle\prod_{j=0}^{l} \left(\hat{h}_j(n_j \cdot 2^j T_s) \right) \\ \cdot x((2^l m - \displaystyle\sum_{j=0}^{l} 2^j n_j)T_s) \cdot e^{j\pi(k+\frac{1}{2}) \cdot \sum_{j=0}^{l} n_j} \end{array} \right\} & ; k \textit{ odd} \end{cases} \tag{4-6}$$

where $T_c = 2^l \cdot T_s$ is the sampling period at the output of the demultiplexer and the halfband filter has exactly $2L+1$ taps. We take note that from (4-2), the quantized coefficients, $\hat{h}(n_j \cdot 2^j T_s)$, are represented by $\hat{h}_j(n_j \cdot 2^j T_s)$ in (4-6) in order to show that the error processes in each stage are independent. Therefore, we can subsequently represent the error processes $\tilde{h}(n_j \cdot 2^j T_s)$ uniquely from each stage by $\widetilde{h}_j(n_j \cdot 2^j T_s)$ for ∀j, j=0,1,..., l [31]. Thus, if we substitute (4-2) into (4-6) and make use of the fact that the filtering operations in every stage are *identical* due to the modular nature of the process, we can conclude without loss of generality that (see Appendix 4.1.1 for derivation) [1] [6] [13]:

$$\hat{y}_k(mT_c) = y_k(mT_c) + \widetilde{y}_k(mT_c) + y_k'(mT_c) \tag{4-7}$$

where $y_k(mT_c)$ is the real valued output of the M-MCDD assuming no quantization and is represented by (3-6). $\widetilde{y}_k(mT_c)$ is the output of the M-MCDD resulting from quantization error and is represented by (see Appendix 4.1.1 for derivation):

$$\widetilde{y}_k(mT_c) = \begin{cases} (-1)^m \sum_{n_0=-L}^{L} \cdots\cdots \sum_{n_l=-L}^{L} \left\{ \begin{array}{l} \prod_{j=0}^{l} \left(\widetilde{h}_j(n_j \cdot 2^j T_s) \right) \\ \cdot x((2^l m - \sum_{j=0}^{l} 2^j n_j) T_s) \cdot e^{j\pi(k+\frac{1}{2}) \cdot \sum_{j=0}^{l} n_j} \end{array} \right\} & ; \begin{array}{l} k = 0\ , \\ k \textit{ even} \end{array} \\ (-1)^m \sum_{n_0=-L}^{L} \cdots\cdots \sum_{n_l=-L}^{L} \left\{ \begin{array}{l} (-1)^{n_0 \cdot T_s} \cdot \prod_{j=0}^{l} \left(\widetilde{h}_j(n_j \cdot 2^j T_s) \right) \\ \cdot x((2^l m - \sum_{j=0}^{l} 2^j n_j) T_s) \cdot e^{j\pi(k+\frac{1}{2}) \cdot \sum_{j=0}^{l} n_j} \end{array} \right\} & ; \ k \textit{ odd} \end{cases} \tag{4-8}$$

and finally, $y_k'(mT_c)$ is the output corresponding to the cross-product between quantization and interference. It is denoted as (see Appendix 4.1.1 for derivation) :

$$y_k'(mT_c)=\begin{cases}(-1)^m\sum_{a=1}^{l}\binom{l+1}{a}\sum_{n_0=-L}^{L}\cdots\sum_{n_l=-L}^{L}\left\{\begin{array}{l}\prod_{j=0}^{a-1}\left(h(n_j\cdot 2^j T_s)\right)\cdot\prod_{j=a}^{l}\left(\widetilde{h}_j(n_j\cdot 2^j T_s)\right)\\ \cdot x((2^l m-\sum_{j=0}^{l}2^j n_j)T_s)\cdot e^{j\pi(k+\frac{1}{2})\cdot\sum_{j=0}^{l}n_j}\end{array}\right\} & ;\ \begin{array}{l}k=0\ ,\\ k\ even\end{array}\\ (-1)^m\sum_{a=1}^{l}\binom{l+1}{a}\sum_{n_0=-L}^{L}\cdots\sum_{n_l=-L}^{L}\left\{\begin{array}{l}(-1)^{n_0\cdot T_s}\cdot\left(\begin{array}{l}\prod_{j=0}^{a-1}\left(h(n_j\cdot 2^j T_s)\right)\\ \cdot\prod_{j=a}^{l}\left(\widetilde{h}_j(n_j\cdot 2^j T_s)\right)\end{array}\right)\\ \cdot x((2^l m-\sum_{j=0}^{l}2^j n_j)T_s)\cdot e^{j\pi(k+\frac{1}{2})\cdot\sum_{j=0}^{l}n_j}\end{array}\right\} & ;\ k\ odd\end{cases}$$

$$=\begin{cases}(-1)^m\sum_{n_0=-L}^{L}\cdots\sum_{n_l=-L}^{L}\left\{\begin{array}{l}\left\{\sum_{a=1}^{l}\binom{l+1}{a}\cdot\left(\begin{array}{l}\prod_{j=0}^{a-1}\left(h(n_j\cdot 2^j T_s)\right)\\ \cdot\prod_{j=a}^{l}\left(\widetilde{h}_j(n_j\cdot 2^j T_s)\right)\end{array}\right)\right\}\\ \cdot x((2^l m-\sum_{j=0}^{l}2^j n_j)T_s)\cdot e^{j\pi(k+\frac{1}{2})\cdot\sum_{j=0}^{l}n_j}\end{array}\right\} & ;\ \begin{array}{l}k=0\ ,\\ k\ even\end{array}\\ (-1)^m\sum_{n_0=-L}^{L}\cdots\sum_{n_l=-L}^{L}\left\{\begin{array}{l}\left\{\sum_{a=1}^{l}\binom{l+1}{a}(-1)^{n_0\cdot T_s}\cdot\left(\begin{array}{l}\prod_{j=0}^{a-1}\left(h(n_j\cdot 2^j T_s)\right)\\ \cdot\prod_{j=a}^{l}\left(\widetilde{h}_j(n_j\cdot 2^j T_s)\right)\end{array}\right)\right\}\\ \cdot x((2^l m-\sum_{j=0}^{l}2^j n_j)T_s)\cdot e^{j\pi(k+\frac{1}{2})\cdot\sum_{j=0}^{l}n_j}\end{array}\right\} & ;\ k\ odd\end{cases}\qquad\text{(4-9)}$$

Combining (3-1), (3-2), (4-5) and (4-6), we arrive at an expression for the quantized final output from the channel detection stage, similar to the form of (4-7), assuming k^{th} desired channel and α^{th} desired symbol (see Appendix 4.1.2 for derivation) :

$$\hat{r}_k(lT_b)=r_k(lT_b)+\widetilde{r}_k(lT_b)+r_k'(lT_b)\qquad\text{(4-10)}$$

where :

$r_k(lT_b)$ is the real valued final output assuming no quantization and represented by (3-7)

$\widetilde{r}_k(lT_b)$ is the final output due to quantization error

$r_k'(lT_b)$ is the cross product result between interference and quantization noise.

4.2.2 : Effect of Quantization on the Final Output

As was done in section 3.1.5, we may consider solely the in-phase component of $\hat{r}_k(lT_b)$ if the system is assumed uncoded [10-11]. Thus, the latter two terms, $\tilde{r}_k^{I}(lT_b)$ and $r_k'^{I}(lT_b)$, of (4-10) may both be combined to effectively represent the total effect of quantization on the output of the channel detection filter. These two quantities may be expressed by (see Appendix 4.1.2 for derivation) :

$$\begin{aligned}\tilde{r}_k^{I}(lT_b) = A_k \cdot &\left[a_{k,\alpha} \cdot \tilde{u}(0) + \sum_{i \neq \alpha} a_{k,i} \cdot \tilde{u}(\alpha T_b - iT_b)\right] \\ &+ \sum_{q \neq k} A_q \cdot \sum_i \left\{\begin{matrix} a_{q,i} \cdot \begin{bmatrix} \tilde{v}^{I}(\alpha T_b - iT_b) \cdot \cos\phi_q \\ -\tilde{v}^{Q}(\alpha T_b - iT_b) \cdot \sin\phi_q \end{bmatrix} \\ -b_{q,i} \cdot \begin{bmatrix} \tilde{v}^{I}(\alpha T_b - iT_b) \cdot \sin\phi_q \\ +\tilde{v}^{Q}(\alpha T_b - iT_b) \cdot \cos\phi_q \end{bmatrix} \end{matrix}\right\} + \tilde{z}_k^{I}(lT_b)\end{aligned} \tag{4-11}$$

$$\begin{aligned}r_k'^{I}(lT_b) = A_k \cdot &\left[a_{k,\alpha} \cdot u'(0) + \sum_{i \neq \alpha} a_{k,i} \cdot u'(\alpha T_b - iT_b)\right] \\ &+ \sum_{q \neq k} A_q \cdot \sum_i \left\{\begin{matrix} a_{q,i} \cdot \begin{bmatrix} v'^{I}(\alpha T_b - iT_b) \cdot \cos\phi_q \\ -v'^{Q}(\alpha T_b - iT_b) \cdot \sin\phi_q \end{bmatrix} \\ -b_{q,i} \cdot \begin{bmatrix} v'^{I}(\alpha T_b - iT_b) \cdot \sin\phi_q \\ +v'^{Q}(\alpha T_b - iT_b) \cdot \cos\phi_q \end{bmatrix} \end{matrix}\right\} + z_k'^{I}(lT_b)\end{aligned} \tag{4-12}$$

The result in (4-11) represents the effect of quantization alone while (4-12) represents the effect of the cross-products between both quantization and interference [14]. Once both results from (4-11) and (4-12) are combined to represent the total effect of quantization on the final output, they may be combined into a single quantization noise variable, η_q , which may be denoted as :

$$\begin{aligned}\eta_q = \tilde{r}_k^{I}(lT_b) + r_k'^{I}(lT_b) \doteq A_k &\left[a_{k,\alpha} \cdot \tilde{u}'(0) + \sum_{i \neq \alpha} a_{k,i} \cdot \tilde{u}'(\alpha T_b - iT_b)\right] \\ &+ \sum_{q \neq k} A_q \cdot \sum_i \left\{\begin{matrix} a_{q,i} \cdot \begin{bmatrix} \tilde{v}'^{I}(\alpha T_b - iT_b) \cdot \cos\phi_q \\ -\tilde{v}'^{Q}(\alpha T_b - iT_b) \cdot \sin\phi_q \end{bmatrix} \\ -b_{q,i} \cdot \begin{bmatrix} \tilde{v}'^{I}(\alpha T_b - iT_b) \cdot \sin\phi_q \\ +\tilde{v}'^{Q}(\alpha T_b - iT_b) \cdot \cos\phi_q \end{bmatrix} \end{matrix}\right\} + \tilde{z}_k'^{I}(lT_b)\end{aligned} \tag{4-13}$$

where the combinations of ISI, ICI on the q^{th} channel and Gaussian noise with quantization noise, respectively, are represented by (see Appendix 4.1.2 for derivation):

$$\widetilde{u}'(\alpha T_b - iT_b) = \begin{cases} \sum\limits_{\xi=I_1}^{I_2} \sum\limits_{n_0} \cdots\cdots \sum\limits_{n_l} \left\{ \begin{array}{l} h_s[(2^l(\beta_l - \xi) - \sum\limits_j 2^j n_j) \cdot T_s - iT_b - \gamma_k] \\ \cdot g[(\xi + \mu_l)T_c] \\ \cdot \hbar_j^{lp}(n_j \cdot 2^j T_s) \end{array} \right\} & ; \quad \begin{array}{l} k = 0 \;, \\ k \text{ even} \end{array} \\ \sum\limits_{\xi=I_1}^{I_2} \sum\limits_{n_0} \cdots\cdots \sum\limits_{n_l} \left\{ \begin{array}{l} h_s[(2^l(\beta_l - \xi) - \sum\limits_j 2^j n_j) \cdot T_s - iT_b - \gamma_k] \\ \cdot g[(\xi + \mu_l)T_c] \\ \cdot \hbar_j^{hp}(n_j \cdot 2^j T_s) \end{array} \right\} & ; \quad k \text{ odd} \end{cases} \tag{4-14}$$

$$\widetilde{v}'^I(\alpha T_b - iT_b) = \begin{cases} \sum\limits_{\xi=I_1}^{I_2} \sum\limits_{n_0} \cdots\cdots \sum\limits_{n_l} \left\{ \begin{array}{l} h_s[(2^l(\beta_l - \xi) - \sum\limits_j 2^j n_j) \cdot T_s - iT_b - \gamma_q] \\ \cdot g[(\xi + \mu_l)T_c] \cdot \cos(\pi(k-q) \cdot \sum\limits_j n_j) \\ \cdot \hbar_j^{lp}(n_j \cdot 2^j T_s) \end{array} \right\} & ; \quad \begin{array}{l} k = 0 \;, \\ k \text{ even} \end{array} \\ \sum\limits_{\xi=I_1}^{I_2} \sum\limits_{n_0} \cdots\cdots \sum\limits_{n_l} \left\{ \begin{array}{l} h_s[(2^l(\beta_l - \xi) - \sum\limits_j 2^j n_j) \cdot T_s - iT_b - \gamma_q] \\ \cdot g[(\xi + \mu_l)T_c] \cdot \cos(\pi(k-q) \cdot \sum\limits_j n_j) \\ \cdot \hbar_j^{hp}(n_j \cdot 2^j T_s) \end{array} \right\} & ; \quad k \text{ odd} \end{cases} \tag{4-15}$$

$$\widetilde{v}'^Q(\alpha T_b - iT_b) = \begin{cases} \sum\limits_{\xi=I_1}^{I_2} \sum\limits_{n_0} \cdots\cdots \sum\limits_{n_l} \left\{ \begin{array}{l} h_s[(2^l(\beta_l - \xi) - \sum\limits_j 2^j n_j) \cdot T_s - iT_b - \gamma_q] \\ \cdot g[(\xi + \mu_l)T_c] \cdot \sin(\pi(k-q) \cdot \sum\limits_j n_j) \\ \cdot \hbar_j^{lp}(n_j \cdot 2^j T_s) \end{array} \right\} & ; \quad \begin{array}{l} k = 0 \;, \\ k \text{ even} \end{array} \\ \sum\limits_{\xi=I_1}^{I_2} \sum\limits_{n_0} \cdots\cdots \sum\limits_{n_l} \left\{ \begin{array}{l} h_s[(2^l(\beta_l - \xi) - \sum\limits_j 2^j n_j) \cdot T_s - iT_b - \gamma_q] \\ \cdot g[(\xi + \mu_l)T_c] \cdot \sin(\pi(k-q) \cdot \sum\limits_j n_j) \\ \cdot \hbar_j^{hp}(n_j \cdot 2^j T_s) \end{array} \right\} & ; \quad k \text{ odd} \end{cases} \tag{4-16}$$

$$\tilde{z}_k'^{I}(lT_b) = \begin{cases} \sum_{\xi=I_1}^{I_2} (-1)^{\beta_l-\xi} \sum_{n_0} \ldots\ldots \sum_{n_l} \left\{ \begin{aligned} & z[(2^l(\beta_l-\xi) - \sum_j 2^j n_j)\cdot T_s] \\ & \cdot g[(\xi+\mu_l)T_c]\cdot \cos\left(\pi(k+\frac{1}{2})\cdot \sum_j n_j\right) \\ & \cdot \hbar_j^{lp}(n_j \cdot 2^j T_s) \end{aligned} \right\} & ; \quad k=0\ , \ k \text{ even} \\ \\ \sum_{\xi=I_1}^{I_2} (-1)^{\beta_l-\xi} \sum_{n_0} \ldots\ldots \sum_{n_l} \left\{ \begin{aligned} & z[(2^l(\beta_l-\xi) - \sum_j 2^j n_j)\cdot T_s] \\ & \cdot g[(\xi+\mu_l)T_c]\cdot \cos\left(\pi(k+\frac{1}{2})\cdot \sum_j n_j\right) \\ & \cdot \hbar_j^{hp}(n_j \cdot 2^j T_s) \end{aligned} \right\} & ; \quad k \text{ odd} \end{cases} \tag{4-17}$$

, where :

$$\hbar_j^{lp}(n_j \cdot 2^j T_s) = \prod_{j=0}^{l}\left(\tilde{h}_j(n_j \cdot 2^j T_s)\right) + \sum_{a=1}^{l}\binom{l+1}{a}\prod_{j=0}^{a-1}\left(h(n_j \cdot 2^j T_s)\right)\cdot \prod_{j=a}^{l}\left(\tilde{h}_j(n_j \cdot 2^j T_s)\right) \tag{4-18}$$

and :

$$\begin{aligned} \hbar_j^{hp}(n_j \cdot 2^j T_s) = {} & (-1)^{n_0 \cdot T_s} \cdot \prod_{j=0}^{l}\left(\tilde{h}_j(n_j \cdot 2^j T_s)\right) \\ & + \sum_{a=1}^{l}\binom{l+1}{a}(-1)^{n_0 \cdot T_s} \cdot \prod_{j=0}^{a-1}\left(h(n_j \cdot 2^j T_s)\right)\cdot \prod_{j=a}^{l}\left(\tilde{h}_j(n_j \cdot 2^j T_s)\right) \end{aligned} \tag{4-19}$$

4.2.3 : Probability of Bit Error Analysis

The interference variable, η , from (3-18) and the quantization noise variable, η_q , from (4-13) may be used to compactly represent $\hat{r}_k^I(lT_b)$, which is the in-phase component of (4-10). As a result, we obtain the sufficient statistic which differs from the sufficient statistic of (3-19) in that we have now taken into consideration the effects of quantization noise for the halfband filter coefficients. The sufficient statistic, resulting from the in-phase component of the final output and is useful in deriving the probability of bit error, becomes [14]:

$$\hat{r}_k^I(lT_b) = A_k \cdot a_{k,\alpha} \cdot u(0) + \eta + \eta_q + z_k^I(lT_b) \tag{4-20}$$

It can be shown that both η_q and η can be well-approximated by Gaussian random variables (see Appendix 4.1.4 for explanation). If from this point on we may assume that both variables are standardized, that is, both have a zero mean and a variance of one, then both variables are well-approximated by a standard normal distribution. Thus, we represent η and η_q, respectively, by [31]:

$$\left(\frac{\eta - m_1(\eta)}{m_2(\eta) - m_1^2(\eta)}\right) \quad ; \quad \left(\frac{\eta_q - m_1(\eta_q)}{m_2(\eta_q) - m_1^2(\eta_q)}\right) \tag{4-21}$$

where $m_1\ (\cdot)$ and $m_2\ (\cdot)$ are the first and second moments about the origin, respectively. For each variable, $m_1\ (\cdot)$ and $m_2\ (\cdot)$ are denoted by the i^{th} moment equation with $i{=}1,2$ [31] :

$$m_i(\eta) = \int_{-\infty}^{+\infty} \eta^i \cdot f_\eta(\eta)\, d\eta \tag{4-22}$$

$$m_i(\eta_q) = \int_{-\infty}^{+\infty} \eta_q{}^i \cdot f_{\eta_q}(\eta_q)\, d\eta_q \tag{4-23}$$

If we average the bit error rate over the desired data symbol $a_k,\ _a\in\{-1,+1\}$, similar to how (3-20) was derived, bearing in mind that the marginal probability density functions of η and η_q are even [10-11] and for a fixed timing offset on the desired channel, the probability of bit error in the presence of interference and quantization noise may be represented by (see Appendix 4.1.3 for derivations) :

$$P_b = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} P_{b|\eta,\eta_q} \cdot f(\eta,\eta_q)\cdot d\eta d\eta_q = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} Q\left(\frac{A_k u(0) - \eta - \eta_q}{\sigma_o}\right)\cdot f(\eta,\eta_q)\cdot d\eta d\eta_q \tag{4-24}$$

As was done in solving the probability of bit error of (3-20), we must resort to numerical techniques to solve (4-24). The reason for this is that the joint probability density function, $f(\eta.\eta_q)$, is unknown and virtually impossible to find due to the innumerable number of interference terms [2]. Because η and η_q may be well-approximated by Gaussian random variables, we can expand $f(\eta,\eta_q\ )$ in terms of standard normal density functions. Thus, a *Gram-Charlier Series* expansion of $f(\eta.\eta_q\ )$ has been chosen to solve (4-24) [19-22]. The advantage which the Gram-Charlier Series possesses is that its coefficients can be generated quite easily and recursively using the moments of the interference and quantization noise. Also, this technique can compute the probability of bit error efficiently without the need of exhaustively going through all interfering symbols [21-22].

4.3 : Gram-Charlier Series Expansion

4.3.1 : Expansion of Joint Probability Density Function

The joint probability density function may be expanded using an orthogonal polynomial expansion denoted by [18] :

$$f(\eta,\eta_q) = f_\eta(\eta)\cdot f_{\eta_q}(\eta_q)\cdot\sum_{\Omega=0}^{\infty} c_\Omega\cdot P_{\eta,\Omega}(\eta)\cdot P_{\eta_q,\Omega}(\eta_q) \tag{4-25}$$

where the polynomials $P(\cdot)$ are orthogonal with the joint probability density function and the coefficients c_Ω must be determined. If both marginal density functions can be approximated by standard normal Gaussian density functions (see Appendix 4.1.4 for explanation) and the joint density function is assumed to be well-approximated by a two-dimensional joint Gaussian distribution, then the quantities from (4-25) may be expressed as [18] [31]:

$$P_{\eta,\Omega}(\eta) = H_\Omega(\eta) \quad ; \quad P_{\eta_q,\Omega}(\eta_q) = H_\Omega(\eta_q) \quad ; \quad c_\Omega = \frac{R^\Omega}{\Omega!} \tag{4-26}$$

where R represents the correlation between η and η_q and is defined by [31] :

$$R = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \eta\cdot\eta_q\cdot f(\eta,\eta_q)\cdot d\eta d\eta_q \tag{4-27}$$

and $H_\Omega(x)$ is the Ω^{th} Hermitian polynomial defined as [18] :

$$H_\Omega(x) = (-1)^\Omega\cdot e^{\frac{x^2}{2}}\cdot\frac{d^\Omega}{dx^\Omega}\left(e^{\frac{-x^2}{2}}\right) \tag{4-28}$$

4.3.2 : Expansion of Marginal Probability Density Functions

4.3.2.1 : Representation and Series Expansion

The marginal density functions may be found from the joint interference-quantization noise density function as [31] :

$$f_\eta(\eta) = \int_{-\infty}^{+\infty} f(\eta,\eta_q)d\eta_q \tag{4-29}$$

$$f_{\eta_q}(\eta_q)=\int_{-\infty}^{+\infty} f(\eta,\eta_q)\,d\eta \qquad (4\text{-}30)$$

We may represent the marginal density functions by Gram-Charlier Series, both of similar form, since $f_\eta(\eta)$ and $f_{\eta_q}(\eta_q)$ are well-approximated by Gaussian densities (see Appendix 4.1.4 for explanation). If the joint density is well-approximated by a jointly Gaussian density function, we can express both marginal densities in the following series representation [18-19] :

$$f_\eta(\eta)=f(\eta)\cdot\left\{1+\sum_{r=3}^{\infty}(-1)^r\cdot\frac{a_r}{r!}\cdot H_r(\eta)\right\} \qquad (4\text{-}31)$$

$$f_{\eta_q}(\eta_q)=f(\eta_q)\cdot\left\{1+\sum_{r=3}^{\infty}(-1)^r\cdot\frac{b_r}{r!}\cdot H_r(\eta_q)\right\} \qquad (4\text{-}32)$$

where the reference function is the standard normal law [19] [31] :

$$f(x)=\frac{1}{\sqrt{2\pi}}\cdot e^{-\frac{x^2}{2}}\ ;-\infty\le x\le+\infty \qquad (4\text{-}33)$$

The coefficients a_r and b_r may be calculated recursively through the relationships [19] :

$$a_i=(-1)^i\cdot m_i(\eta)-\sum_{w=1}^{i}(-1)^w\cdot\binom{i}{w}\cdot c_w\cdot a_{i-w}\quad ;\quad a_o=1,a_1=a_2=0\quad ;\quad 3\le i\le\infty \qquad (4\text{-}34)$$

$$b_i=(-1)^i\cdot m_i(\eta_q)-\sum_{w=1}^{i}(-1)^w\cdot\binom{i}{w}\cdot c_w\cdot b_{i-w}\quad ;\quad b_o=1,b_1=b_2=0\quad ;\quad 3\le i\le\infty \qquad (4\text{-}35)$$

where [31] :

$$c_\alpha=\begin{cases}(\alpha-1)!! & ;\alpha\ even\\ 0 & ;\alpha\ odd\end{cases} \qquad (4\text{-}36)$$

and [14] [18] :

$$(\alpha-1)!!=(\alpha-1)\cdot(\alpha-3)\cdots\cdots5\cdot3\cdot1 \qquad (4\text{-}37)$$

4.3.2.2 : Convergence of the Gram-Charlier Series Method

For the numerical computation of $f_\eta(\eta)$ and $f_\eta(\eta_q)$ through the Gram-Charlier Series representation, it is necessary to verify that the series converge. A sufficient, but not necessary, condition for the convergence of the Gram-Charlier Series is known to be [20] :

$$E\left\{e^{\frac{\eta^2}{4}}\right\}<\infty\quad ;\quad E\left\{e^{\frac{\eta_q^2}{4}}\right\}<\infty \qquad (4\text{-}38)$$

It can be shown in a straightforward manner that the above expressions are true (see Appendix 4.2 for proof).

4.3.3 : Upper Bound on the Q-function

The Q-function from (4-24) may be upper bounded using an exponential function [23] :

$$Q\left(\frac{A_k u(0)-\eta-\eta_q}{\sigma_o}\right)<\frac{1}{2}\cdot\exp\left\{-\frac{1}{2}\cdot\left(\frac{A_k u(0)-\eta-\eta_q}{\sigma_o}\right)^2\right\} \tag{4-39}$$

(4-39) can be expanded straightforwardly to obtain a function of η and another function of η_q. . A third term in the expansion is a function of both variables and is not separable by straightforward means. Once this third term is expanded using a Taylor Series, the final result on the upper bound of the Q-function is (see Appendix 4.3 for derivation) [34] :

$$\begin{aligned}Q\left(\frac{A_k u(0)-\eta-\eta_q}{\sigma_o}\right)<\sum_{m=0}^{\infty}\frac{(-1)^m}{2\cdot m!}\cdot\exp\left\{-\frac{1}{2}\cdot\left(\frac{\frac{A_k}{2}u(0)-\eta}{\sigma_o}\right)^2\right\}\cdot\left(\frac{\frac{A_k}{2}u(0)-\eta}{\sigma_o}\right)^m\\ \cdot\exp\left\{-\frac{1}{2}\cdot\left(\frac{\frac{A_k}{2}u(0)-\eta_q}{\sigma_o}\right)^2\right\}\cdot\left(\frac{\frac{A_k}{2}u(0)-\eta_q}{\sigma_o}\right)^m\end{aligned} \tag{4-40}$$

4.3.4 : Final Expression for the Probability of Bit Error

Once (4-25), (4-26), (4-31), (4-32) and (4-40) are substituted into the probability of bit error expression of (4-24) , a final expression may be obtained representing an upper bound for the probability of bit error. Thus, assuming that $|\eta|>0$ and $|\eta_q|>0$ [14] :

$$P_b<\sum_{m,\Omega=0}^{\infty}\left\{\begin{aligned}&\frac{(-1)^m\cdot R^{\Omega}}{2\cdot m!\,\Omega!}\cdot\int_{-\infty}^{+\infty}\left\{\begin{aligned}&\exp\left\{-\frac{1}{2}\left(\frac{\frac{A_k}{2}u(0)-\eta}{\sigma_o}\right)^2\right\}\cdot\left(\frac{\frac{A_k}{2}u(0)-\eta}{\sigma_o}\right)^m\\&\cdot H_{\Omega}(\eta)\cdot\left\{1+\sum_{r=3}^{\infty}(-1)^r\cdot\frac{a_r}{r!}\cdot H_r(\eta)\right\}\end{aligned}\right\}\cdot f(\eta)\,d\eta\\&\cdot\int_{-\infty}^{+\infty}\left\{\begin{aligned}&\exp\left\{-\frac{1}{2}\left(\frac{\frac{A_k}{2}u(0)-\eta_q}{\sigma_o}\right)^2\right\}\cdot\left(\frac{\frac{A_k}{2}u(0)-\eta_q}{\sigma_o}\right)^m\\&\cdot H_{\Omega}(\eta_q)\cdot\left\{1+\sum_{r=3}^{\infty}(-1)^r\cdot\frac{b_r}{r!}\cdot H_r(\eta_q)\right\}\end{aligned}\right\}\cdot f(\eta_q)\,d\eta_q\end{aligned}\right\} \tag{4-41}$$

4.3.5 : Evaluation of the Moments for the Quantization Noise

4.3.5.1 : Moment Evaluation

The moments of the quantization noise can be derived in a manner similar to that used to derive the moments of the interference bearing in mind that there is an added level of randomality in the halfband filters' quantization error. If we truncate the number of interfering symbols on the desired channel and on the interfering channels to $R = m_l + m_u + 1$ and $R_Q = m_{lQ} + m_{uQ} + 1$, respectively, to give a total number of interfering symbols $K = R + \sum R_Q$, then η_q becomes* :

$$\begin{aligned}\eta_q^{*} &= A_k \sum_{l=\alpha-m_l}^{\alpha+m_u} a_{k,l} \cdot \tilde{u}'(\alpha T_b - iT_b) \\ &+ \sum_{Q \neq k} A_Q \cdot \sum_{l=\alpha-m_{l,Q}}^{\alpha+m_{u,Q}} \begin{Bmatrix} a_{Q,l} \cdot [\tilde{v}'^{I}(\alpha T_b - iT_b) \cdot \cos\phi_Q - \tilde{v}'^{\mathcal{Q}}(\alpha T_b - iT_b) \cdot \sin\phi_Q] \\ -b_{Q,l} \cdot [\tilde{v}'^{I}(\alpha T_b - iT_b) \cdot \sin\phi_Q + \tilde{v}'^{\mathcal{Q}}(\alpha T_b - iT_b) \cdot \cos\phi_Q] \end{Bmatrix} + \tilde{z}_k'^{I}(lT_b)\end{aligned} \tag{4-42}$$

If, for fixed i, we can denote the ISI component by $\eta_{q.1.i}$ and the ICI component on the Q^{th} channel by $\eta_{q.2.Q.i}$, we can derive the expressions for the moments of the quantization noise individually. As was the case for the interference, it would be possible to evaluate the moments of the ISI and ICI individually since the data symbols $a_{Q,i}$ and $b_{Q,i}$ are i.i.d [10-11]. These individual moments would be combined by Prabhu Technique, indicated by (3-32) through (3-35), to arrive at the final moments for the quantization noise [29]. Furthermore, since only the even order moments are non-zero [14] and using Schwartz's Inequality to separate the interference and quantization noise variables, we obtain (see Appendix 4.4 for derivation) [34] :

$$E\left\{\eta_{q,1,l}^{v}\right\} \leq \begin{cases} 2^{\Psi} A_k^{v} \sum\limits_{a_{k,l}} a_{k,l}^{v} \cdot \begin{Bmatrix} \int\limits_{-T/2}^{T/2} \left(uu(\alpha T_b - iT_b)\right)^{v/2} \cdot f(\gamma_k)\, d\gamma_k \\ \cdot \int\limits_{-2^{-B-1}}^{2^{-B-1}} \cdots \int\limits_{-2^{-B-1}}^{2^{-B-1}} \left\{ \sum\limits_{n_0} \cdots \sum\limits_{n_l} \left(\hbar_j^{lp}(n_j \cdot 2^{j} T_s) \right)^2 \right\}^{v/2} d\tilde{h} \cdots d\tilde{h} \end{Bmatrix} & ; \quad \begin{matrix} k = 0\,, \\ k\ even \end{matrix} \\ 2^{\Psi} A_k^{v} \sum\limits_{a_{k,l}} a_{k,l}^{v} \cdot \begin{Bmatrix} \int\limits_{-T/2}^{T/2} \left(uu(\alpha T_b - iT_b)\right)^{v/2} \cdot f(\gamma_k)\, d\gamma_k \\ \cdot \int\limits_{-2^{-B-1}}^{2^{-B-1}} \cdots \int\limits_{-2^{-B-1}}^{2^{-B-1}} \left\{ \sum\limits_{n_0} \cdots \sum\limits_{n_l} \left(\hbar_j^{lp}(n_j \cdot 2^{j} T_s) \right)^2 \right\}^{v/2} d\tilde{h} \cdots d\tilde{h} \end{Bmatrix} & ; \quad k\ odd \end{cases} \tag{4-43}$$

* Please note that when the context of discussion involves quantization, we use q to represent the quantization variable. Thus, so as not to confuse our choice of variables from previous chapters, the channel representation is given as Q when not discussing the k^{th} channel. Also, quadrature components are shown as $\mathcal{Q}$.

$$E\left\{\eta^{v}_{q,2,Q,l}\right\} \le \begin{cases} 2^{\Psi}\cdot A^{v}_{Q}\displaystyle\sum_{\substack{a_{Q,l}\\ b_{Q,l}}}\left\{\int_{-T/2}^{T/2}\begin{Bmatrix} uu(\alpha T_b - iT_b) \\ -a_{Q,l}\cdot b_{Q,l}\cdot vv(\alpha T_b - iT_b)\end{Bmatrix}^{v/2}\cdot f(\gamma_Q)\,d\gamma_Q\right\} & \\ \cdot\displaystyle\int_{-2^{-B-1}}^{2^{-B-1}}\cdots\int_{-2^{-B-1}}^{2^{-B-1}}\left\{\sum_{n_0}\cdots\cdots\sum_{n_l}\left(\hbar^{lp}_{j}(n_j\cdot 2^{j}T_s)\right)^2\right\}^{v/2} d\tilde{h}\cdots d\tilde{h} & ;\quad \begin{matrix} k=0\ , \\ k \text{ even}\end{matrix} \\ 2^{\Psi}\cdot A^{v}_{Q}\displaystyle\sum_{\substack{a_{Q,l}\\ b_{Q,l}}}\left\{\int_{-T/2}^{T/2}\begin{Bmatrix} uu(\alpha T_b - iT_b) \\ -a_{Q,l}\cdot b_{Q,l}\cdot vv(\alpha T_b - iT_b)\end{Bmatrix}^{v/2}\cdot f(\gamma_Q)\,d\gamma_Q\right\} & \\ \cdot\displaystyle\int_{-2^{-B-1}}^{2^{-B-1}}\cdots\int_{-2^{-B-1}}^{2^{-B-1}}\left\{\sum_{n_0}\cdots\cdots\sum_{n_l}\left(\hbar^{lp}_{j}(n_j\cdot 2^{j}T_s)\right)^2\right\}^{v/2} d\tilde{h}\cdots d\tilde{h} & ;\quad k \text{ odd}\end{cases} \tag{4-44}$$

, where, for simplicity of representation, the following quantities have been defined :

$$uu(\alpha T_b - iT_b) = \sum_{\xi=I_1}^{I_2}\sum_{n_0}\cdots\sum_{n_l}\begin{Bmatrix} h_s^2[(2^{l}(\beta_l - \xi) - \sum_j 2^{j} n_j)\cdot T_s - iT_b - \gamma_k] \\ \cdot g^2[(\xi + \mu_l)T_c] \end{Bmatrix} \tag{4-45}$$

$$vv(\alpha T_b - iT_b) = \sum_{\xi=I_1}^{I_2}\sum_{n_0}\cdots\sum_{n_l}\begin{Bmatrix} h_s^2[(2^{l}(\beta_l - \xi) - \sum_j 2^{j} n_j)\cdot T_s - iT_b - \gamma_Q] \\ \cdot g^2[(\xi + \mu_l)T_c]\cdot\cos(2\pi(k-Q)\cdot\sum_j n_j) \end{Bmatrix} \tag{4-46}$$

$$\begin{aligned} zz(\alpha T_b) &= \sum_{\xi=I_1}^{I_2}\sum_{n_0}\cdots\sum_{n_l}\begin{Bmatrix} z^2[(2^{l}(\beta_l - \xi) - \sum_j 2^{j} n_j)\cdot T_s]\cdot g^2[(\xi+\mu_l)T_c] \\ \cdot\cos^2\left(\pi(k+\frac{1}{2})\cdot\sum_j n_j\right) \end{Bmatrix} \\ &= \sum_{\xi=I_1}^{I_2}\sum_{n_0}\cdots\sum_{n_l}\begin{Bmatrix} z^2[(2^{l}(\beta_l - \xi) - \sum_j 2^{j} n_j)\cdot T_s]\cdot g^2[(\xi+\mu_l)T_c] \\ \left\{\frac{1}{2}\left(1+\cos\left(2\pi(k+\frac{1}{2})\cdot\sum_j n_j\right)\right)\right\} \end{Bmatrix} \end{aligned} \tag{4-47}$$

, and :

$$\Psi = B\cdot h_{total} - 1 \tag{4-48}$$

, where the total number of halfband filter coefficients which are quantized are :

$$h_{total} = (2L+1)(l+1) \tag{4-49}$$

, and $Ltap$ is chosen such that the total number of filter taps per halfband filter is 2L+1 for $L \in \mathrm{N}$ [34]. (4-49) is obtained by considering a single arbitrary channel path through the M-MCDD. For example, to obtain $y_0(nT_c)$, we go through h_{total} halfband filter coefficients, referring to Figure 2-4, including the effect of the DAF.

We can simplify (4-43) and (4-44) knowing that only the even order moments are non-zero [14]. In that case, (4-43) and (4-44) respectively become :

$$E\left\{\eta_{q,1,j}^{2v}\right\} \leq \begin{cases} 2^{\Psi+1} \cdot A_k^{2v} \cdot \left\{ \begin{aligned} &\int_{-T/2}^{T/2} \left(uu(\alpha T_b - iT_b)\right)^v \cdot f(\gamma_k)\, d\gamma_k \\ &\cdot \int_{-2^{-\beta-1}}^{2^{-\beta-1}} \cdots \int_{-2^{-\beta-1}}^{2^{-\beta-1}} \left\{ \sum_{n_0} \cdots\cdots \sum_{n_l} \left(\hbar_j^{lp}(n_j \cdot 2^j T_s) \right)^2 \right\}^v d\tilde{h} \cdots d\tilde{h} \end{aligned} \right\} & ; \quad k = 0\ , \ k \text{ even} \\ \\ 2^{\Psi-1} \cdot A_k^{2v} \cdot \left\{ \begin{aligned} &\int_{-T/2}^{T/2} \left(uu(\alpha T_b - iT_b)\right)^v \cdot f(\gamma_k)\, d\gamma_k \\ &\cdot \int_{-2^{-\beta-1}}^{2^{-\beta-1}} \cdots \int_{-2^{-\beta-1}}^{2^{-\beta-1}} \left\{ \sum_{n_0} \cdots\cdots \sum_{n_l} \left(\hbar_j^{hp}(n_j \cdot 2^j T_s) \right)^2 \right\}^v d\tilde{h} \cdots d\tilde{h} \end{aligned} \right\} & ; \quad k \text{ odd} \end{cases} \tag{4-50}$$

$$E\left\{\eta_{q,2,Q,j}^{2v}\right\} \leq \begin{cases} 2^{\Psi+1} \cdot A_Q^{2v} \cdot \int_{-T/2}^{T/2} \left\{ \begin{Bmatrix} uu(\alpha T_b - iT_b) \\ -vv(\alpha T_b - iT_b) \end{Bmatrix}^v + \begin{Bmatrix} uu(\alpha T_b - iT_b) \\ +vv(\alpha T_b - iT_b) \end{Bmatrix}^v \right\} \cdot f(\gamma_Q)\, d\gamma_Q & \\ \cdot \int_{-2^{-\beta-1}}^{2^{-\beta-1}} \cdots \int_{-2^{-\beta-1}}^{2^{-\beta-1}} \left\{ \sum_{n_0} \cdots\cdots \sum_{n_l} \left(\hbar_j^{lp}(n_j \cdot 2^j T_s) \right)^2 \right\}^v d\tilde{h} \cdots d\tilde{h} & ; \quad k = 0\ , \ k \text{ even} \\ \\ 2^{\Psi+1} \cdot A_Q^{2v} \cdot \int_{-T/2}^{T/2} \left\{ \begin{Bmatrix} uu(\alpha T_b - iT_b) \\ -vv(\alpha T_b - iT_b) \end{Bmatrix}^v + \begin{Bmatrix} uu(\alpha T_b - iT_b) \\ +vv(\alpha T_b - iT_b) \end{Bmatrix}^v \right\} \cdot f(\gamma_Q)\, d\gamma_Q & \\ \cdot \int_{-2^{-\beta-1}}^{2^{-\beta-1}} \cdots \int_{-2^{-\beta-1}}^{2^{-\beta-1}} \left\{ \sum_{n_0} \cdots\cdots \sum_{n_l} \left(\hbar_j^{hp}(n_j \cdot 2^j T_s) \right)^2 \right\}^v d\tilde{h} \cdots d\tilde{h} & ; \quad k \text{ odd} \end{cases} \tag{4-51}$$

As for the Gaussian noise component, it is possible to simplify since the even order moments of a Gaussian random variable are known, given that $z(\cdot) \sim N(0, \sigma^2)$ [31]. Thus, the even order moments become (see Appendix 4.4 for derivation) :

$$E\left\{\left(\tilde{z}_k'^{I}(lT_b)\right)^{v}\right\} \le \begin{cases} 2^{\varphi+1}\cdot\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\left(zz(\alpha T_b)\right)^{v/2}\cdot\left(f(z)\right)^{h_{max}-(I_2-I_1+1)}dz\cdots dz \\ \cdot\int_{-2^{-B-1}}^{2^{-B-1}}\cdots\int_{-2^{-B-1}}^{2^{-B-1}}\left\{\sum_{n_0}\cdots\cdots\sum_{n_l}\left(\tilde{h}_j^{lp}(n_j\cdot 2^{j}T_s)\right)^2\right\}^{v/2}d\tilde{h}\cdots d\tilde{h} & ; \begin{matrix} k=0\ , \\ k\ even \end{matrix} \\ 2^{\varphi+1}\cdot\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\left(zz(\alpha T_b)\right)^{v/2}\cdot\left(f(z)\right)^{h_{max}-(I_2-I_1+1)}dz\cdots dz \\ \cdot\int_{-2^{-B-1}}^{2^{-B-1}}\cdots\int_{-2^{-B-1}}^{2^{-B-1}}\left\{\sum_{n_0}\cdots\cdots\sum_{n_l}\left(\tilde{h}_j^{lp}(n_j\cdot 2^{j}T_s)\right)^2\right\}^{v/2}d\tilde{h}\cdots d\tilde{h} & ; \ k\ odd \end{cases} \tag{4-52}$$

4.3.5.2 : Correlation Between the Interference and Quantization Noise

As for the correlation between η and η_q , it is seen to consist of two individual components. Using the same arguments as previous, these two components are (see Appendix 4.5 for derivation) :

$$E\left\{\eta_{1J}\cdot\eta_{q.1J}\right\} \le \begin{cases} 2^{\varphi+1}\cdot\int_{-T/2}^{T/2}\left\{A_k^2\cdot u(\alpha T_b - iT_b)\cdot\sqrt{uu(\alpha T_b - iT_b)}\right\}\cdot f(\gamma_k)d\gamma_k \\ \cdot\int_{-2^{-B-1}}^{2^{-B-1}}\cdots\int_{-2^{-B-1}}^{2^{-B-1}}\sqrt{\sum_{n_0}\cdots\cdots\sum_{n_l}\left(\tilde{h}_j^{lp}(n_j\cdot 2^{j}T_s)\right)^2}\cdot d\tilde{h}\cdots d\tilde{h} & ; \begin{matrix} k=0\ , \\ k\ even \end{matrix} \\ 2^{\varphi+1}\cdot\int_{-T/2}^{T/2}\left\{A_k^2\cdot u(\alpha T_b - iT_b)\cdot\sqrt{uu(\alpha T_b - iT_b)}\right\}\cdot f(\gamma_k)d\gamma_k \\ \cdot\int_{-2^{-B-1}}^{2^{-B-1}}\cdots\int_{-2^{-B-1}}^{2^{-B-1}}\sqrt{\sum_{n_0}\cdots\cdots\sum_{n_l}\left(\tilde{h}_j^{lp}(n_j\cdot 2^{j}T_s)\right)^2}\cdot d\tilde{h}\cdots d\tilde{h} & ; \ k\ odd \end{cases} \tag{4-53}$$

$$E\left\{\eta_{2J}\cdot\eta_{q.2J}\right\} \le \begin{cases} 2^{\varphi+1}\sum_{Q\neq k}A_Q^2\cdot\int_{-T/2}^{T/2}\left\{\begin{matrix} v^{I}(\alpha T_b - iT_b)\cdot\sqrt{\frac{1}{2}\cdot\begin{pmatrix} uu(\alpha T_b - iT_b) \\ +vv(\alpha T_b - iT_b)\end{pmatrix}} \\ +v^{Q}(\alpha T_b - iT_b)\cdot\sqrt{\frac{1}{2}\cdot\begin{pmatrix} uu(\alpha T_b - iT_b) \\ -vv(\alpha T_b - iT_b)\end{pmatrix}} \end{matrix}\right\}\cdot f(\gamma_Q)d\gamma_Q \\ \cdot\int_{-2^{-B-1}}^{2^{-B-1}}\cdots\int_{-2^{-B-1}}^{2^{-B-1}}\sqrt{\sum_{n_0}\cdots\cdots\sum_{n_l}\left(\tilde{h}_j^{lp}(n_j\cdot 2^{j}T_s)\right)^2}\cdot d\tilde{h}\cdots d\tilde{h} & ; \begin{matrix} k=0\ , \\ k\ even \end{matrix} \\ 2^{\varphi+1}\sum_{Q\neq k}A_Q^2\cdot\int_{-T/2}^{T/2}\left\{\begin{matrix} v^{I}(\alpha T_b - iT_b)\cdot\sqrt{\frac{1}{2}\cdot\begin{pmatrix} uu(\alpha T_b - iT_b) \\ +vv(\alpha T_b - iT_b)\end{pmatrix}} \\ +v^{Q}(\alpha T_b - iT_b)\cdot\sqrt{\frac{1}{2}\cdot\begin{pmatrix} uu(\alpha T_b - iT_b) \\ -vv(\alpha T_b - iT_b)\end{pmatrix}} \end{matrix}\right\}\cdot f(\gamma_Q)d\gamma_Q \\ \cdot\int_{-2^{-B-1}}^{2^{-B-1}}\cdots\int_{-2^{-B-1}}^{2^{-B-1}}\sqrt{\sum_{n_0}\cdots\cdots\sum_{n_l}\left(\tilde{h}_j^{lp}(n_j\cdot 2^{j}T_s)\right)^2}\cdot d\tilde{h}\cdots d\tilde{h} & ; \ k\ odd \end{cases} \tag{4-54}$$

4.3.5.3 : Computation of Quantization Moments and Correlation

Use has been made of certain recursions and known facts to effectively compute the correlation and moments of the quantization noise. Thus, in calculating (4-43) through (4-54), use can be made of the fact that the odd order moments are all zero and that the even order moments of a normal distribution with zero mean and variance σ^2 and that of a uniform distribution are known and well-defined. In the case of a uniform distribution, since the probability density function is simply a constant within the range of the values of the random variable, the integral expression is quite trivial in its solution. In both cases, the moments of $z(\cdot)\sim N(0,\sigma^2)$ and $\widetilde{h}(\cdot)$ *~Uniform*$(0,2^{-2B}/12)$ to compute (4-43) through (4-54) can be expressed as [31] :

$$\int_{-\infty}^{+\infty} z^{2\lambda}[(2^{l}(\beta_l-\xi)-\sum_j 2^j n_j)\cdot T_s]\cdot f(z)\,dz=(2\lambda-1)!!\cdot\sigma^2 \;\; ; \; 1\le\lambda\le K_2 \qquad (4\text{-}55)$$

$$\int_{-2^{-B-1}}^{2^{-B-1}} \widetilde{h}_j^{\varphi}(n_j\cdot 2^j T_s)\cdot f_{\widetilde{h}}(\widetilde{h})\cdot d\widetilde{h}=\begin{cases}0 & ;\;\; \varphi\ odd\\ \dfrac{2^{\varphi(-B-1)}}{\varphi+1} & ;\;\; \varphi\ even\end{cases} \quad (2\le\varphi\le 2K_2) \qquad (4\text{-}56)$$

, where in the above integrals ξ, j, and n_j are set to a fixed value. Also, we assume that we are using only the first $2K_1$ moments of the interference and the first $2K_2$ moments of the quantization error, corresponding to the upper bounding of each Gram-Charlier series to the $2K^{th}$ term. The truncation of the Gram-Charlier Series for the probability of bit error expression of (4-41) will be discussed in the next section.

4.3.6 : Truncation Errors From the Gram-Charlier Series Expansion

4.3.6.1 : Truncated probability of Bit Error Expression

In order to evaluate the probability of bit error, we have to truncate the series expressions shown in (4-41). We will assume that we are using the following [14]:

- the first $2K_1+1$ moments of the interference
- the first $2K_2+1$ moments of the quantization
- the first $2K_3+1$ terms of the orthogonal expansion
- the first $M+1$ terms of the Taylor Series Expansion

The reason for truncating each Gram-Charlier Series to $2K_i$ terms is in order to keep each upper bound at an even value, since only the even order moments of the interference and quantization are non-zero [13-14]. Also, analogous to the discussion in chapter three on the interference, we are using K_I in place of M.

As a result of these truncations, we experience truncation errors [2]. In particular, the effects of two truncation errors in the numerical calculation of P_b will be discussed : the error due to the truncation of each Gram-Charlier Series [35] and the error due to the truncation of the Taylor Series representation of $Q(\cdot)$ [30]. The expression for the probability of bit error can be numerically evaluated by considering :

$$P_b < \sum_{m=0}^{M}\sum_{\Omega=0}^{2K_1}\left\{\begin{array}{l} \dfrac{(-1)^m\cdot R^{\Omega}}{2\cdot m!\Omega!}\cdot\displaystyle\int_{\xi_{\eta,\min}}^{\xi_{\eta,\max}}\left\{\begin{array}{l}\exp\left\{-\dfrac{1}{2}\left(\dfrac{\frac{A_b}{2}u(0)-\eta}{\sigma_o}\right)^2\right\}\cdot\left(\dfrac{\frac{A_b}{2}u(0)-\eta}{\sigma_o}\right)^m \\ \cdot H_{\Omega}(\eta)\cdot\left\{1+\displaystyle\sum_{r=3}^{2K_1}(-1)^r\cdot\dfrac{a_r}{r!}\cdot H_r(\eta)\right\}\end{array}\right\}\cdot f(\eta)\,d\eta \\ \cdot\displaystyle\int_{\xi_{\eta_q,\min}}^{\xi_{\eta_q,\max}}\left\{\begin{array}{l}\exp\left\{-\dfrac{1}{2}\left(\dfrac{\frac{A_b}{2}u(0)-\eta_q}{\sigma_o}\right)^2\right\}\cdot\left(\dfrac{\frac{A_b}{2}u(0)-\eta_q}{\sigma_o}\right)^m \\ \cdot H_{\Omega}(\eta_q)\cdot\left\{1+\displaystyle\sum_{r=3}^{2K_2}(-1)^r\cdot\dfrac{b_r}{r!}\cdot H_r(\eta_q)\right\}\end{array}\right\}\cdot f(\eta_q)\,d\eta_q \end{array}\right\} \tag{4-57}$$

$$+R_M+R'_{2K_1,\eta}+R'_{2K_2,\eta_q}$$

where : R_M is the truncation error due to using only $(M+1)$ terms in the Taylor Series.

R'_{2K_i} is the truncation error from upper bounding each Gram-Charlier Series to $2K_i$ terms.

ξ_η and ξ_{η_q} are fixed values of the interference and quantization error, respectively.

The expressions for the truncation errors will be given in section 4.3.6.4.

4.3.6.2 : Computation and Simplification of Bit Error Expression

To compute the BER in the presence of quantization noise, we will put (4-57) in a simpler form. First of all, to numerically evaluate (4-57), we use Schwartz's Inequality to separate the interference from the quantization [34]. From this, we obtain:

$$P_b < \sqrt{\sum_{m=0}^{M}\sum_{\Omega=0}^{2K_1}\frac{1}{(m!)^2}\cdot\left(\int_{\xi_{\eta,MIN}}^{\xi_{\eta,MAX}}\left\{\begin{array}{l}\exp\left\{-\frac{1}{2}\left(\frac{\frac{A_s}{2}u(0)-\eta}{\sigma_o}\right)^2\right\}\cdot\left(\frac{\frac{A_s}{2}u(0)-\eta}{\sigma_o}\right)^m \\ \cdot H_\Omega(\eta)\cdot\left\{1+\sum_{r=3}^{2K_1}(-1)^r\cdot\frac{a_r}{r!}\cdot H_r(\eta)\right\}\cdot f(\eta)\,d\eta\end{array}\right\}\right)^2}$$
$$\cdot\sqrt{\sum_{m=0}^{M}\sum_{\Omega=0}^{2K_1}\frac{R^{2\Omega}}{4\cdot(\Omega!)^2}\cdot\left(\int_{\xi_{\eta_q,MIN}}^{\xi_{\eta_q,MAX}}\left\{\begin{array}{l}\exp\left\{-\frac{1}{2}\left(\frac{\frac{A_s}{2}u(0)-\eta_q}{\sigma_o}\right)^2\right\}\cdot\left(\frac{\frac{A_s}{2}u(0)-\eta_q}{\sigma_o}\right)^m \\ \cdot H_\Omega(\eta_q)\cdot\left\{1+\sum_{r=3}^{2K_2}(-1)^r\cdot\frac{b_r}{r!}\cdot H_r(\eta_q)\right\}\cdot f(\eta_q)\,d\eta_q\end{array}\right\}\right)^2} \tag{4-58}$$

Focusing on the quantization part of (4-58), if we take isolate the term corresponding to *m=0* and *Ω=0*, we further simplify (4-57) for numerical evaluation :

$$P_b < \sqrt{\sum_{m=0}^{M}\sum_{\Omega=0}^{2K_1}\frac{1}{(m!)^2}\cdot\left(\int_{\xi_{\eta,MIN}}^{\xi_{\eta,MAX}}\left\{\begin{array}{l}\exp\left\{-\frac{1}{2}\left(\frac{\frac{A_s}{2}u(0)-\eta}{\sigma_o}\right)^2\right\}\cdot\left(\frac{\frac{A_s}{2}u(0)-\eta}{\sigma_o}\right)^m \\ \cdot H_\Omega(\eta)\cdot\left\{1+\sum_{r=3}^{2K_1}(-1)^r\cdot\frac{a_r}{r!}\cdot H_r(\eta)\right\}\cdot f(\eta)\,d\eta\end{array}\right\}\right)^2}$$
$$\cdot\sqrt{\begin{array}{l}\left(1+\frac{1}{4}\cdot\sum_{r=3}^{2K_2}\left(\int_{\xi_{\eta_q,MIN}}^{\xi_{\eta_q,MAX}}\exp\left\{-\frac{1}{2}\left(\frac{\frac{A_s}{2}u(0)-\eta_q}{\sigma_o}\right)^2\right\}\cdot(-1)^r\cdot\frac{b_r}{r!}\cdot H_r(\eta_q)\cdot f(\eta_q)\,d\eta_q\right)^2\right. \\ \left.+\sum_{\substack{m=1\\ m=0,\Omega\neq0\\ m\neq0,\Omega=0}}^{M}\sum_{\Omega=1}^{2K_1}\frac{R^{2\Omega}}{4\cdot(\Omega!)^2}\cdot\left(\int_{\xi_{\eta_q,MIN}}^{\xi_{\eta_q,MAX}}\left\{\begin{array}{l}\exp\left\{-\frac{1}{2}\left(\frac{\frac{A_s}{2}u(0)-\eta_q}{\sigma_o}\right)^2\right\}\cdot\left(\frac{\frac{A_s}{2}u(0)-\eta_q}{\sigma_o}\right)^m \\ \cdot H_\Omega(\eta_q)\cdot\left\{1+\sum_{r=3}^{2K_2}(-1)^r\cdot\frac{b_r}{r!}\cdot H_r(\eta_q)\right\}\cdot f(\eta_q)\,d\eta_q\end{array}\right\}\right)^2\right.\end{array}} \tag{4-59}$$

So, from the upper bound on the BER of (4-59), we see we have two distinguishable parts. From (4-59), we can say :

$$P_b < P_{b,\eta}\cdot\Delta_{BER,\eta_q} \tag{4-60}$$

, where :

$$P_{b,\eta} = \sqrt{\sum_{m=0}^{M}\sum_{\Omega=0}^{2K_1}\frac{1}{(m!)^2}\cdot\left(\int_{\xi_{\eta,MIN}}^{\xi_{\eta,MAX}}\left\{\begin{array}{l}\exp\left\{-\frac{1}{2}\left(\frac{\frac{A_s}{2}u(0)-\eta}{\sigma_o}\right)^2\right\}\cdot\left(\frac{\frac{A_s}{2}u(0)-\eta}{\sigma_o}\right)^m \\ \cdot H_\Omega(\eta)\cdot\left\{1+\sum_{r=3}^{2K_1}(-1)^r\cdot\frac{a_r}{r!}\cdot H_r(\eta)\right\}\cdot f(\eta)\,d\eta\end{array}\right\}\right)^2} \tag{4-61}$$

represents the BER due to interference only and :

$$\Delta_{BER.\eta_q} = \sqrt{\begin{array}{l} \left(1+\frac{1}{4}\cdot\sum_{r=3}^{2K_1}\left(\int_{\xi_{\eta_q,MIN}}^{\xi_{\eta_q,MAX}} \exp\left\{-\frac{1}{2}\left(\frac{\frac{A_k}{2}u(0)-\eta_q}{\sigma_o}\right)^2\right\}\cdot(-1)^r\cdot\frac{b_r}{r!}\cdot H_r(\eta_q)\cdot f(\eta_q)\,d\eta_q\right)\right)^2 \\ +\left(\sum_{\substack{m=1 \\ m=0,\Omega=0 \\ m=0,\Omega=0}}^{M}\sum_{\Omega=1}^{2K_1}\frac{R^{2\Omega}}{4\cdot(\Omega!)^2}\cdot\left(\int_{\xi_{\eta_q,MIN}}^{\xi_{\eta_q,MAX}}\left\{\begin{array}{l}\exp\left\{-\frac{1}{2}\left(\frac{\frac{A_k}{2}u(0)-\eta_q}{\sigma_o}\right)^2\right\}\cdot\left(\frac{\frac{A_k}{2}u(0)-\eta_q}{\sigma_o}\right)^m \\ \cdot H_\Omega(\eta_q)\cdot\left\{1+\sum_{r=3}^{2K_1}(-1)^r\cdot\frac{b_r}{r!}\cdot H_r(\eta_q)\right\}\cdot f(\eta_q)\,d\eta_q\end{array}\right\}\right)\right)^2\end{array}} \tag{4-62}$$

represents the BER degradation due to the presence of quantization noise. In section 4.4.1, we will discuss shortcuts in computing the BER expression using the above simplification process.

4.3.6.3 : Maxima and Minima of the Quantization Noise

The maxima and minima of the interference and the quantization noise are used in the probability of bit error expression of (4-57) thus need to be defined. The maxima and minima of the interference were derived previously and are as indicated by (3-44) and (3-45), respectively. The maxima and minima of the quantization noise, respectively, can be expressed as (see Appendix 4.6 for derivation) :

$$\begin{aligned}\xi_{\eta_q,MAX} &= \max_{\gamma_k}\sum_{i=\alpha-m_l}^{\alpha+m_u}\left|A_k\cdot\tilde{u}'_{\max}(\alpha T_b-iT_b)\right| \\ &\quad+\sqrt{2}\sum_{q\neq k}A_q\max_{\gamma_q}\sum_{i=\alpha-m_{l,q}}^{\alpha+m_{u,q}}\left|\tilde{v}'^{I}_{\max}(\alpha T_b-iT_b)\right|+\left|\tilde{v}'^{Q}_{\max}(\alpha T_b-iT_b)\right|+\max\tilde{z}'^{I}_k(lT_b)\end{aligned} \tag{4-63}$$

$$\begin{aligned}\xi_{\eta_q,MIN} &= -\min_{\gamma_k}\sum_{i=\alpha-m_l}^{\alpha+m_u}\left|A_k\cdot\tilde{u}'_{\min}(\alpha T_b-iT_b)\right| \\ &\quad-\sqrt{2}\sum_{q\neq k}A_q\min_{\gamma_q}\sum_{i=\alpha-m_{l,q}}^{\alpha+m_{u,q}}\left|\tilde{v}'^{I}_{\min}(\alpha T_b-iT_b)\right|+\left|\tilde{v}'^{Q}_{\min}(\alpha T_b-iT_b)\right|+\min\tilde{z}'^{I}_k(lT_b)\end{aligned} \tag{4-64}$$

, where $\tilde{u}'_{\max}(\alpha T_b-iT_b)$, $\tilde{v}'^{I}_{\max}(\alpha T_b-iT_b)$, $\tilde{v}'^{Q}_{\max}(\alpha T_b-iT_b)$ and $\max\tilde{z}'^{I}_k(lT_b)$ are the maximum values of the ISI, ICI on the q^{th} channel and Gaussian noise, respectively, with respect to the halfband filter coefficient quantization error, $\tilde{h}_j(n_j\cdot 2^j T_s)$. Likewise, $\tilde{u}'_{\min}(\alpha T_b-iT_b)$, $\tilde{v}'^{I}_{\min}(\alpha T_b-iT_b)$, $\tilde{v}'^{Q}_{\min}(\alpha T_b-iT_b)$ and $\min\tilde{z}'^{I}_k(lT_b)$ are the minimum values of the ISI, ICI on the

q^{th} channel and Gaussian noise, respectively. These quantities are (see Appendix 4.6 for explanation) :

$$
\begin{array}{l}
\widetilde{u}'_{\max}(\alpha T_b - iT_b) = -\widetilde{u}'_{\min}(\alpha T_b - iT_b) = \\
= \begin{cases}
2^{-B-1}\cdot\sum\limits_{\xi=I_1}^{I_2}\sum\limits_{n_0}\cdots\sum\limits_{n_l} h_s[(2^l(\beta_l-\xi)-\sum\limits_j 2^j n_j)\cdot T_s - iT_b - \gamma_k]\cdot g[(\xi+\mu_l)T_c] \\
+\sum\limits_{a=1}^{l}\binom{l+1}{a}\sum\limits_{\xi=I_1}^{I_2}\left\{\begin{array}{l}
\sum\limits_{n_0}\cdots\sum\limits_{n_{a-1}}\left\{\begin{array}{l}\prod\limits_{j=0}^{a-1}\left(h(n_j\cdot 2^j T_s)\right) \\ \cdot h_s[(2^l(\beta_l-\xi)-\sum\limits_j 2^j n_j)\cdot T_s - iT_b - \gamma_k] \\ \cdot g[(\xi+\mu_l)T_c]\end{array}\right\} \\
\sum\limits_{n_a}\cdots\sum\limits_{n_l}\left\{\begin{array}{l}\left(2^{-B-1}\right)\cdot \\ h_s[(2^l(\beta_l-\xi)-\sum\limits_j 2^j n_j)\cdot T_s - iT_b - \gamma_k] \\ \cdot g[(\xi+\mu_l)T_c]\end{array}\right\}
\end{array}\right\} & ; \quad k=0\ , \ k \text{ even} \\
2^{-B-1}\cdot\sum\limits_{\xi=I_1}^{I_2}\sum\limits_{n_0}\cdots\sum\limits_{n_l} h_s[(2^l(\beta_l-\xi)-\sum\limits_j 2^j n_j)\cdot T_s - iT_b - \gamma_k]\cdot g[(\xi+\mu_l)T_c] \\
+\sum\limits_{a=1}^{l}\binom{l+1}{a}\sum\limits_{\xi=I_1}^{I_2}\left\{\begin{array}{l}
\sum\limits_{n_0}\cdots\sum\limits_{n_{a-1}}\left\{\begin{array}{l}(-1)^{n_0\cdot T_s}\cdot\prod\limits_{j=0}^{a-1}\left(h(n_j\cdot 2^j T_s)\right) \\ \cdot h_s[(2^l(\beta_l-\xi)-\sum\limits_j 2^j n_j)\cdot T_s - iT_b - \gamma_k] \\ \cdot g[(\xi+\mu_l)T_c]\end{array}\right\} \\
\sum\limits_{n_a}\cdots\sum\limits_{n_l}\left\{\begin{array}{l}\left(2^{-B-1}\right)\cdot \\ h_s[(2^l(\beta_l-\xi)-\sum\limits_j 2^j n_j)\cdot T_s - iT_b - \gamma_k] \\ \cdot g[(\xi+\mu_l)T_c]\end{array}\right\}
\end{array}\right\} & ; \quad k \text{ odd}
\end{cases}
\end{array}
\qquad (4\text{-}65)
$$

$$\tilde{v}'^{\,I}_{\max}(\alpha T_b - iT_b) = -\tilde{v}'^{\,I}_{\min}(\alpha T_b - iT_b) =$$

$$= \begin{cases} 2^{-B-1}\cdot\displaystyle\sum_{\xi=I_1}^{I_2}\sum_{n_0}\cdots\sum_{n_l}\left\{\begin{array}{l} h_s[(2^l(\beta_l-\xi)-\sum_j 2^j n_j)\cdot T_s - iT_b - \gamma_q]\cdot g[(\xi+\mu_l)T_c] \\ \cdot\cos(\pi(k-q)\cdot\sum_j n_j) \end{array}\right\} \\ +\displaystyle\sum_{a=1}^{l}\binom{l+1}{a}\sum_{\xi=I_1}^{I_2}\left\{\begin{array}{l} \displaystyle\sum_{n_0}\cdots\sum_{n_{a-1}}\left\{\begin{array}{l} \prod_{j=0}^{a-1}\left(h(n_j\cdot 2^j T_s)\right) \\ \cdot h_s[(2^l(\beta_l-\xi)-\sum_j 2^j n_j)\cdot T_s - iT_b - \gamma_q] \\ \cdot g[(\xi+\mu_l)T_c]\cdot\cos(\pi(k-q)\cdot\sum_j n_j) \end{array}\right\} \\ \displaystyle\sum_{n_a}\cdots\sum_{n_l}\left\{\begin{array}{l} \left(2^{-B-1}\right)\cdot \\ h_s[(2^l(\beta_l-\xi)-\sum_j 2^j n_j)\cdot T_s - iT_b - \gamma_q] \\ \cdot g[(\xi+\mu_l)T_c]\cdot\cos(\pi(k-q)\cdot\sum_j n_j) \end{array}\right\} \end{array}\right\} & ;\quad k=0\ ,\ k\ \mathit{even} \\ 2^{-B-1}\cdot\displaystyle\sum_{\xi=I_1}^{I_2}\sum_{n_0}\cdots\sum_{n_l}\left\{\begin{array}{l} h_s[(2^l(\beta_l-\xi)-\sum_j 2^j n_j)\cdot T_s - iT_b - \gamma_q]\cdot g[(\xi+\mu_l)T_c] \\ \cdot\cos(\pi(k-q)\cdot\sum_j n_j) \end{array}\right\} \\ +\displaystyle\sum_{a=1}^{l}\binom{l+1}{a}\sum_{\xi=I_1}^{I_2}\left\{\begin{array}{l} \displaystyle\sum_{n_0}\cdots\sum_{n_{a-1}}\left\{\begin{array}{l} (-1)^{n_0\cdot T_s}\cdot\prod_{j=0}^{a-1}\left(h(n_j\cdot 2^j T_s)\right) \\ \cdot h_s[(2^l(\beta_l-\xi)-\sum_j 2^j n_j)\cdot T_s - iT_b - \gamma_q] \\ \cdot g[(\xi+\mu_l)T_c]\cdot\cos(\pi(k-q)\cdot\sum_j n_j) \end{array}\right\} \\ \displaystyle\sum_{n_a}\cdots\sum_{n_l}\left\{\begin{array}{l} \left(2^{-B-1}\right)\cdot \\ h_s[(2^l(\beta_l-\xi)-\sum_j 2^j n_j)\cdot T_s - iT_b - \gamma_q] \\ \cdot g[(\xi+\mu_l)T_c]\cdot\cos(\pi(k-q)\cdot\sum_j n_j) \end{array}\right\} \end{array}\right\} & ;\quad k\ \mathit{odd} \end{cases} \tag{4-66}$$

$$
\tilde{v}'^{Q}_{\max}(\alpha T_b - iT_b) = -\tilde{v}'^{Q}_{\min}(\alpha T_b - iT_b) =
$$

$$
= \begin{cases}
2^{-B-1}\cdot\sum_{\xi=I_1}^{I_2}\sum_{n_0}\cdots\sum_{n_l}\left\{\begin{array}{l} h_s[(2^l(\beta_l-\xi)-\sum_j 2^j n_j)\cdot T_s - iT_b - \gamma_q]\cdot g[(\xi+\mu_l)T_c] \\ \cdot\sin(\pi(k-q)\cdot\sum_j n_j) \end{array}\right\} \\
+\sum_{a=1}^{l}\binom{l+1}{a}\sum_{\xi=I_1}^{I_2}\left\{\begin{array}{l} \sum_{n_0}\cdots\sum_{n_{a-1}}\left\{\begin{array}{l} \prod_{j=0}^{a-1}\left(h(n_j\cdot 2^j T_s)\right) \\ \cdot h_s[(2^l(\beta_l-\xi)-\sum_j 2^j n_j)\cdot T_s - iT_b - \gamma_q] \\ \cdot g[(\xi+\mu_l)T_c]\cdot\sin(\pi(k-q)\cdot\sum_j n_j) \end{array}\right\} \\ \sum_{n_a}\cdots\sum_{n_l}\left\{\begin{array}{l} \left(2^{-B-1}\right)\cdot \\ h_s[(2^l(\beta_l-\xi)-\sum_j 2^j n_j)\cdot T_s - iT_b - \gamma_q] \\ \cdot g[(\xi+\mu_l)T_c]\cdot\sin(\pi(k-q)\cdot\sum_j n_j) \end{array}\right\} \end{array}\right\} & ; \quad \begin{array}{l} k=0\ , \\ k \text{ even} \end{array} \\
2^{-B-1}\cdot\sum_{\xi=I_1}^{I_2}\sum_{n_0}\cdots\sum_{n_l}\left\{\begin{array}{l} h_s[(2^l(\beta_l-\xi)-\sum_j 2^j n_j)\cdot T_s - iT_b - \gamma_q]\cdot g[(\xi+\mu_l)T_c] \\ \cdot\sin(\pi(k-q)\cdot\sum_j n_j) \end{array}\right\} \\
+\sum_{a=1}^{l}\binom{l+1}{a}\sum_{\xi=I_1}^{I_2}\left\{\begin{array}{l} \sum_{n_0}\cdots\sum_{n_{a-1}}\left\{\begin{array}{l} (-1)^{n_0\cdot T_s}\cdot\prod_{j=0}^{a-1}\left(h(n_j\cdot 2^j T_s)\right) \\ \cdot h_s[(2^l(\beta_l-\xi)-\sum_j 2^j n_j)\cdot T_s - iT_b - \gamma_q] \\ \cdot g[(\xi+\mu_l)T_c]\cdot\sin(\pi(k-q)\cdot\sum_j n_j) \end{array}\right\} \\ \sum_{n_a}\cdots\sum_{n_l}\left\{\begin{array}{l} \left(2^{-B-1}\right)\cdot \\ h_s[(2^l(\beta_l-\xi)-\sum_j 2^j n_j)\cdot T_s - iT_b - \gamma_q] \\ \cdot g[(\xi+\mu_l)T_c]\cdot\sin(\pi(k-q)\cdot\sum_j n_j) \end{array}\right\} \end{array}\right\} & ; \quad k \text{ odd}
\end{cases}
\tag{4-67}
$$

$$\max \tilde{z}_k'^{I}(lT_b) = -\min \tilde{z}_k'^{I}(lT_b) =$$

$$= \begin{cases} 3.291\sigma\cdot\left(\begin{array}{l} \left|2^{-B-1}\right|\cdot\sum_{\xi=I_1}^{I_2}(-1)^{\beta_l-\xi}\cdot\sum_{n_0}\cdots\sum_{n_l} g[(\xi+\mu_l)T_c]\cdot\cos\left(\pi(k+\frac{1}{2})\cdot\sum_j n_j\right) \\ +\sum_{a=1}^{l}\binom{l+1}{a}\sum_{\xi=I_1}^{I_2}\left\{\begin{array}{l} \sum_{n_0}\cdots\sum_{n_{a-1}}\left\{\begin{array}{l}\prod_{j=0}^{a-1}\left(h(n_j\cdot 2^j T_s)\right) \\ \cdot g[(\xi+\mu_l)T_c]\cdot\cos\left(\pi(k+\frac{1}{2})\cdot\sum_j n_j\right)\end{array}\right\} \\ \sum_{n_a}\cdots\sum_{n_l}\left\{\begin{array}{l}\left|2^{-B-1}\right| \\ \cdot g[(\xi+\mu_l)T_c]\cdot\cos\left(\pi(k+\frac{1}{2})\cdot\sum_j n_j\right)\end{array}\right\}\end{array}\right\}\end{array}\right) & ;\quad k=0\ ,\ k\ \text{even} \\ \\ 3.291\sigma\cdot\left(\begin{array}{l} \left|2^{-B-1}\right|\cdot\sum_{\xi=I_1}^{I_2}(-1)^{\beta_l-\xi}\cdot\sum_{n_0}\cdots\sum_{n_l} g[(\xi+\mu_l)T_c]\cdot\cos\left(\pi(k+\frac{1}{2})\cdot\sum_j n_j\right) \\ +\sum_{a=1}^{l}\binom{l+1}{a}\sum_{\xi=I_1}^{I_2}\left\{\begin{array}{l} \sum_{n_0}\cdots\sum_{n_{a-1}}\left\{\begin{array}{l}(-1)^{n_0\cdot T_s}\cdot\prod_{j=0}^{a-1}\left(h(n_j\cdot 2^j T_s)\right) \\ \cdot g[(\xi+\mu_l)T_c]\cdot\cos\left(\pi(k+\frac{1}{2})\cdot\sum_j n_j\right)\end{array}\right\} \\ \sum_{n_a}\cdots\sum_{n_l}\left\{\begin{array}{l}\left|2^{-B-1}\right| \\ \cdot g[(\xi+\mu_l)T_c]\cdot\cos\left(\pi(k+\frac{1}{2})\cdot\sum_j n_j\right)\end{array}\right\}\end{array}\right\}\end{array}\right) & ;\quad k\ \text{odd} \end{cases} \qquad \textbf{(4-68)}$$

If the timing phase can be assumed to be constant on the desired channel and the interfering channels, then it is easy to see from (3-44), (3-45), (4-63) and (4-64) that :

$$\xi_{\eta,MAX} = -\xi_{\eta,MIN} \qquad \textbf{(4-69)}$$

$$\xi_{\eta_q,MAX} = -\xi_{\eta_q,MIN} \qquad \textbf{(4-70)}$$

4.3.6.4 : Expressions for the Truncation Errors

The upper bound on the truncation error for using only the first $2K_i$ terms in each of the Gram-Charlier Series can be maximized by considering (3-44) and (4-63), respectively. Thus, the maximum truncation error in this case has been shown to be [35] :

$$\left|R'_{2K_i,\eta}\right| < C'_\eta\cdot\int_{-T/2}^{T/2}\exp\left(\frac{-(\frac{A_k}{2}u(0)-\xi_{\eta,MAX})^2}{4\sigma_o^2}\right)\cdot f(\gamma_k)\,d\gamma_k \qquad \textbf{(4-71)}$$

$$\left|R'_{2K_1,\eta_q}\right| < C'_{\eta_q} \cdot \int_{-T/2}^{T/2} \exp\left(\frac{-(\frac{A_k}{2}u(0)-\xi_{\eta_q,MAX})^2}{4\sigma_o^2}\right) \cdot f(\gamma_k)\,d\gamma_k \tag{4-72}$$

where [35] :

$$C'_\mu = \frac{(2K-1)!!}{(2K+2)!} \cdot \sqrt{4K+2} \cdot \frac{1}{2\sigma_o^{2(K+1)}} \cdot \frac{m_{2K+2}(\mu)}{\left(1-\dfrac{m_2(\mu)}{2(K+1)\cdot\sigma_o^2}\right)} \tag{4-73}$$

The truncation error for using only *(M+1)* terms in the Taylor Series for *Q(·)* is (see Appendix 4.7 for derivation) [30, pg. 880] :

$$|R_M| \le \frac{1}{(M+1)!} \cdot \left(\frac{-(\frac{A_k}{2}u(0)-\xi_{\eta,MAX})}{\sigma_o}\right)^{M+1} \cdot \left(\frac{-(\frac{A_k}{2}u(0)-\xi_{\eta_q,MAX})}{\sigma_o}\right)^{M+1} \tag{4-74}$$

4.4 : Performance Results of the M-MCDD Resulting From Filter Coefficient Quantization

4.4.1 : Discussion of Gram-Charlier Series Expansion Algorithm

The algorithm used to evaluate the amount of degradation in performance due to halfband filter coefficient quantization using the Gram-Charlier Series Expansion technique is shown in Figure 4-3. Only the major deviations from the GQR algorithm will be discussed here and these deviations are within the loops for the calculation of the interference and quantization coefficients (Figure 4-3(a)), the computation of the moments and correlation (Figure 4-3(b)) and the final calculation of the BER (Figure 4-3(b)).

First, we will observe how the calculation of the quantization coefficients were integrated into the program so as to save as much processing power as possible. It should be noted that the block diagram in Figure 4-3 for the computation of the interference and quantization coefficients is greatly simplified yet contains all the essential information of what the algorithm is doing. As was stated, the ISI and ICI were computed to be able to calculate the moments of the interference. Similar to the calculation of the moments of the interference, where the vectors $u[.]$, $v^I[.]$, $v^Q[.]$ are precomputed to save computation time, similar functions can be defined to ease the computation of the moments of the quantization noise. Various linear combinations of (4-45) and (4-46) can be used in calculating the ISI, ICI and Gaussian noise aggregated with the quantization noise as well as the correlation between the interference and quantization noise. In addition, time can be saved since the vectors defined by (4-45) and (4-46) may be computed within the same loops as those used to compute the ISI and ICI of the interference. The effects of quantization on the desired symbol are computed using the quantities *uq0_max* and *uu0* . These are placed within the arrays *uq_max[·]* and *uu[·]* , respectively, after the first *mL* interfering symbols below the desired symbol. Most of the remaining quantities computed are needed to determine $\tilde{u}'_{\max}(\cdot), \tilde{v}'^{I}_{\max}(\cdot)$ $, \tilde{v}'^{Q}_{\max}(\cdot)$ according to (4-65), (4-66) and (4-67), respectively. The quantities used to calculate $\tilde{u}'_{\max}(\cdot)$ are *uq_max[·]* , *tmp_sum* and *uq_max_tmp*. For $\tilde{v}'^{I}_{\max}(\cdot)$, the quantities *vqI_max[·]* , *tmp_sumI* and *vqI_max_tmp* are used. Likewise, for $\tilde{v}'^{Q}_{\max}(\cdot)$, the quantities *vqQ_max[·]* , *tmp_sumQ* and *vqQ_max_tmp* are used. The quantities *zq_max* , *tmp_sum_z* and *zq_max_tmp* are used in computing $max\, \tilde{z}'^{I}_{k}(lT_b)$ of (A4-73) from Appendix 4.6. In turn, $max\, \tilde{z}'^{I}_{k}(lT_b)$ is used to compute $\xi_{\eta_q,MAX}$ and $\xi_{\eta_q,MIN}$ indicated by (4-63) and (4-64), respectively. This calculation is shown in Figure 4-3(a) within the module entitled 'compute zq_max'. The maximum value of the Gaussian noise aggregated with quantization noise is computed within the ICI loop but is only executed if the first interfering symbol below the desired symbol is being computed. The reason why this calculation was placed within the ICI loop is the need for the decimation factor, 2^J, and the calculation of $\cos(\pi(k+\frac{1}{2})\cdot\sum n_j)$, indicated by (3-15).

The computation of the moments of the quantization and the correlation between the interference and quantization is a new part of this algorithm which was not present in the GQR algorithm. The quantization array for the moments is defined at first. This quantization array is computed using MAPLE_V and corresponds to the calculation of the following integral in the case *k=0* and *k even* [36] :

$$\int_{-2^{-B-1}}^{2^{-B-1}}\cdots\int_{-2^{-B-1}}^{2^{-B-1}}\left\{\sum_{n_0}\cdots\cdots\sum_{n_J}\left\{\tilde{h}_j^{bp}(n_j\cdot 2^J T_s)\right\}^2\right\}^{v/2} d\tilde{h}\cdots d\tilde{h} \tag{4-75}$$

, where $\tilde{h}_j^{bp}(n_j\cdot 2^J T_s)$ is defined by (4-18). (4-75) is used in computing the moments (4-43) through (4-54). There were certain limitations that were encountered in the calculation of (4-75) from above which will be discussed shortly. Using this array and *uu[·]* and *vv[·]* of (4-45) and (4-46), respectively, we can calculate the ISI, ICI and Gaussian noise moments of the quantization. Unlike the calculation of the ICI moments of the interference, no randomality with the phase offset was assumed in the calculation of the ICI moments of the quantization. As for the Gaussian noise moment, the portion aggregated with quantization noise corresponding to:

$$\int_{-\infty}^{+\infty}\cdots\int_{-\infty}^{+\infty}\left(zz(\alpha T_b)\right)^{v/2}\cdot\left(f(z)\right)^{(I_2-I_1+1)(l+1)(2L+1)} dz\cdots dz \tag{4-76}$$

, where $zz(\alpha T_b)$, defined by (4-47), was computed using MATLAB but making use of (4-55), the integration process was not needed [32]. To compute the correlation between the interference and quantization, a new array, *u_tmp[·]*, needed to be defined and this array included the desired symbol, *u0*, in its *(mL+1)*th position. In contrast, the ISI array, *u[·]*, did not include the desired symbol. The final moments are then compiled using Prabhu's Method [29]. Since the correlation is a second order moment, the third component of the output vector was chosen since it corresponded to the value of the correlation [31].

As was seen in section 4.3.6.2, the BER in the presence of quantization noise was put in a simpler and a more programmable form. It will now be shown how it is possible to simplify the algorithm based on the results from the interference only scenario and by precomputing some necessary BER coefficients. Consider the expression for $P_{b,\eta}$ of (4-61). For an ideal infinite quantization scenario, we eliminate all forms of quantization noise and we are therefore in the presence of interference only. Therefore, for infinite quantization, that is for B→ ∞ , $\xi_{\eta_q,MAX}$ of (4-63) approaches zero thus $\xi_{\eta_q,MIN}$ of (4-64) also approaches zero. Therefore, as B→ ∞ , the integrals in the Δ_{BER,η_q} expression of (4-62) approach zero, thus $\Delta_{BER,\eta_q} \rightarrow 1$ and so the BER for an interference only scenario can be represented simply by (4-61). Because the performance of the M-MCDD due to interference has been analyzed using the GQR technique, it would be desirable to make use of these previous results here. Therefore, the results from the GQR have been used in this algorithm to represent the results of the BER expression of (4-61). Of course, it is expected that the true numerical results of (4-61) would agree with those of the GQR algorithm but this has not been verified. This portion of the GQR algorithm to compute the infinite quantization BER is indicated in Figure 4-3(b).

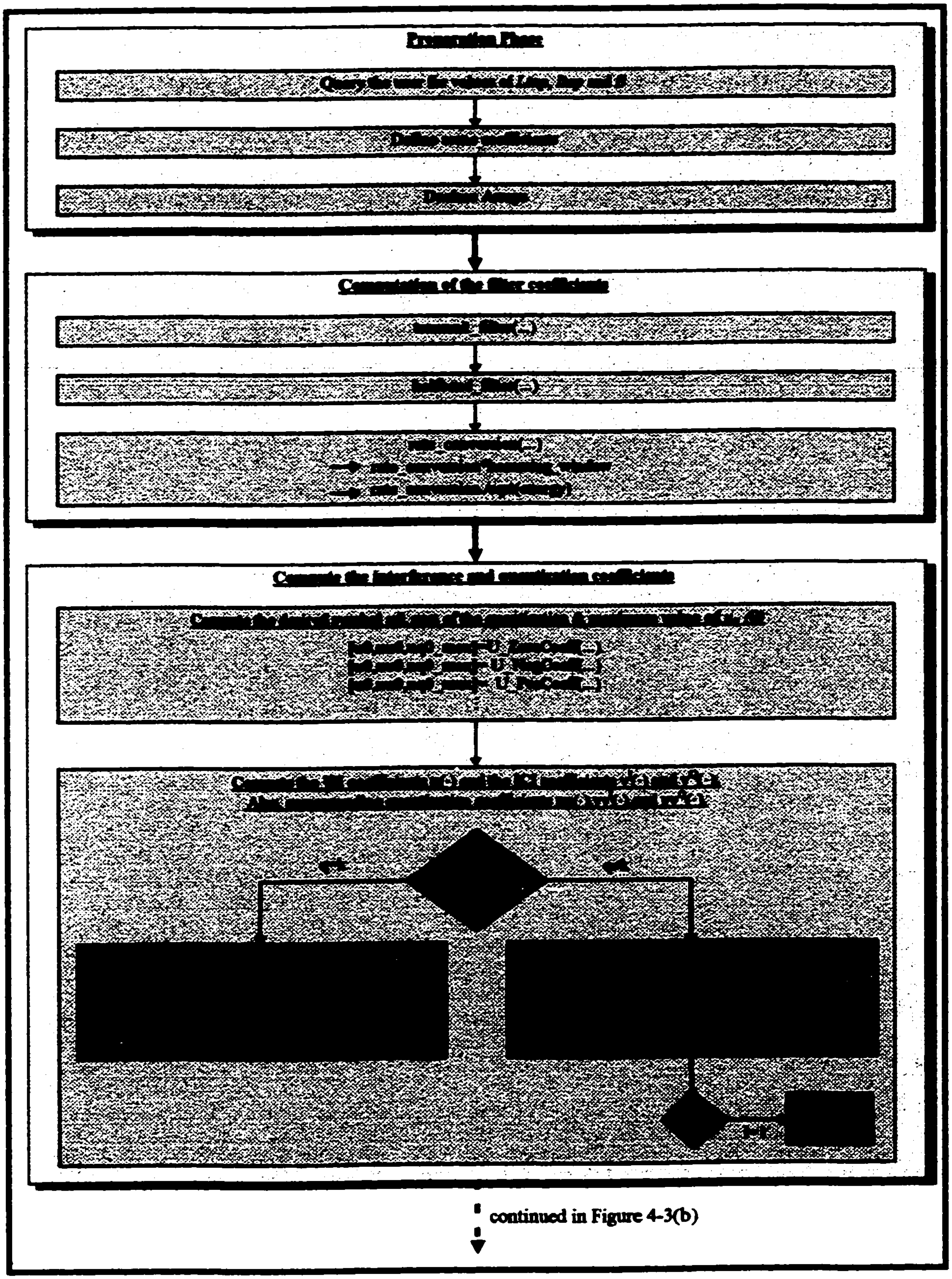
What remains to be computed is the BER degradation of (4-62) to determine the amount of degradation in the BER due to quantization noise for a finite value of *B*. Since (4-62) requires solving complex integral expressions involving Hermite polynomials, Δ_{BER,η_q} was numerically evaluated using MAPLE_V [36]. Because the remaining part of the algorithm is programmed in MATLAB, at first it seemed necessary to have an interaction between MATLAB and MAPLE_V. It seemed necessary to be able to interact between both numerical evaluation tools because we would need to import and export several variables to and from MAPLE_V, respectively. For example, to compute (4-62), MAPLE_V would require the values of b_i , *R*, $\xi_{\eta_q,MAX}$ and *u0* and the result of the program, to be sent back to MATLAB, would be the array of BER degradation coefficients. Unfortunately, these two numerical tools are not easily compatible therefore importing and exporting variables between both is not done easily. Also, each execution of the MAPLE_V algorithm took in excess of three hours which is even longer than the execution time for the overall MATLAB program [32] [36]! Therefore, instead of creating a direct interface between MATLAB and MAPLE_V, another alternative was thought of which would effectively eliminate the need for this interface. What was done to make the overall Gram-Charlier Series algorithm work swiftly and smoothly was that a limited number of scenarios were selected and the MATLAB program was executed for each one using the corresponding parameters. For each scenario, the output values of the necessary variables of the MATLAB program were used as a new set of input parameters to the MAPLE_V algorithm, which was then executed. Once all the scenarios of interest were investigated, a matrix of BER degradation coefficients, representing the outputs of the MAPLE_V algorithm, was compiled. Therefore, by having such a matrix which could easily be loaded by the main program in MATLAB, in addition to eliminating the need for an interface between MATLAB and MAPLE_V, we also save a great deal of CPU time [32] [36]. A total of approximately eleven scenarios were selected and the selection was based on performance with respect to infinite quantization and CPU time requirements of the GQR. For example, the scenarios involving *Itap=3* were not selected.

4.4.2 : Limitations of the Gram-Charlier Series Expansion Algorithm

There are a few limitations in the main algorithm which may have affected the outcome of the numerical results. The effect of these limitations on the results will be further outlined in the next section. The lack of optimization of the number of terms used in the Taylor Series expansion of $Q(\cdot)$, represented by M , may have affected the overall accuracy. Also, a likely contributing factor to the final outcome of this algorithm is the lack of higher order moments because the program makes use of the moments of the quantization noise in order to generate the BER degradation coefficients. In fact, due to limited resources, we were limited to the fourth order moments for the quantization noise. This is due to the long computation times and lack of system resources in computing (4-75) and (4-76). The integral expression of (4-75) to compute the quantization array for the moments was computed using MAPLE_V for various *Ltap* values while keeping B as an open variable. Due the length of time required to calculate the quantization array for the scenarios of interest, in some cases up to 30 minutes, they have been precomputed using MAPLE_V [36]. The portion of the Gaussian noise moment of the quantization noise, corresponding to (4-76), encountered a situation similar to the quantization array of (4-75). Typically, to evaluate the higher order moments of $\tilde{z}_k'^{I}(lT_b)$, (4-76) would have to be expanded in order to make its computation simpler but by doing so, the algorithm to compute the moment would become very lengthy in terms of size and time. For example, a fourth order moment of $\tilde{z}_k'^{I}(lT_b)$ would typically take around 40 minutes to compute while a sixth order moment would take over 10 hours to compute!

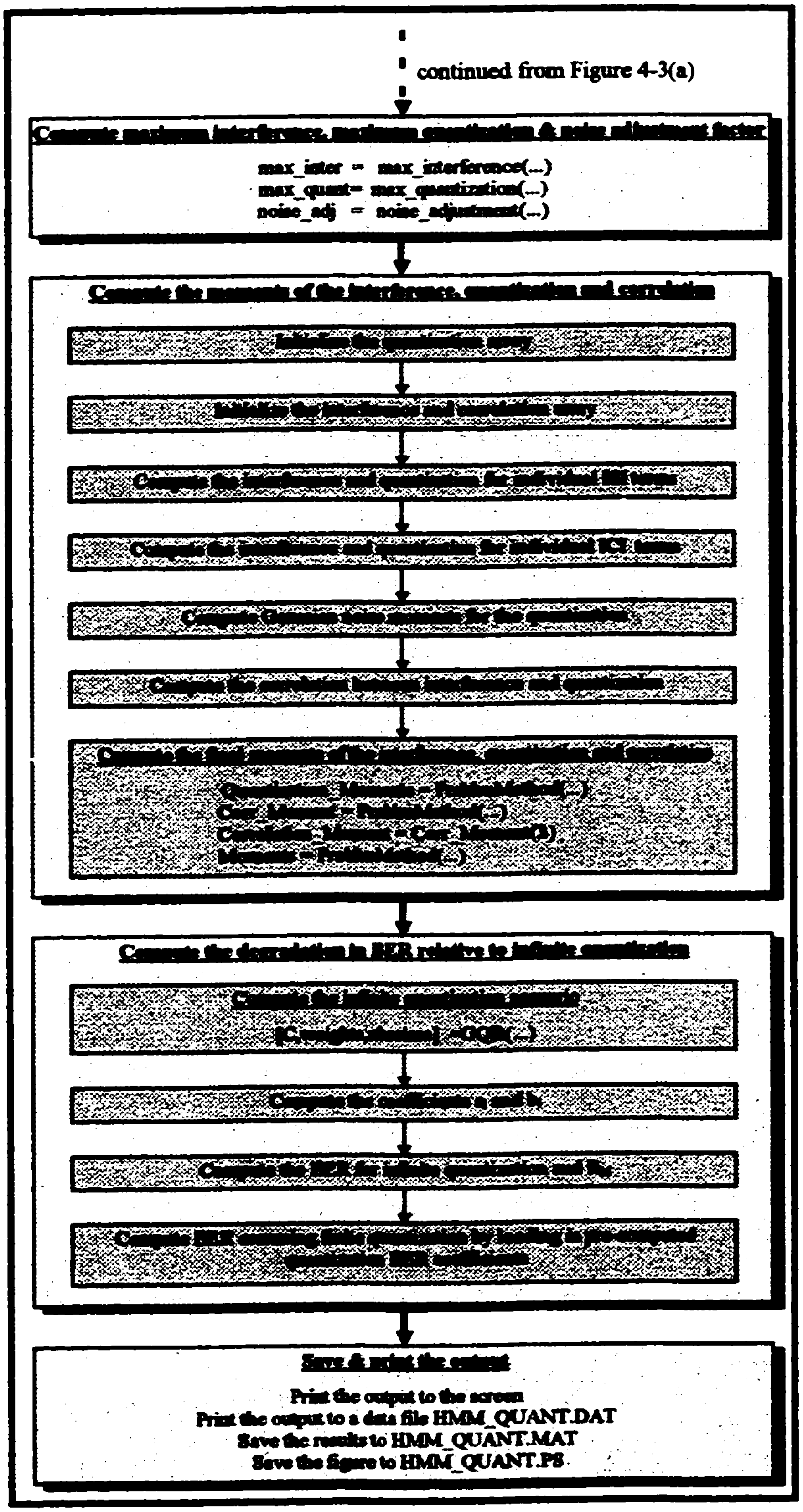
The MAPLE_V algorithm, used to compute (4-62), also has several limitations which should be mentioned. These limitations stem from the fact that the computation of each integral expression in MAPLE_V takes a significant amount of time [36]. In fact, in more complicated cases, it would not be unusual to have a single integral take several minutes to compute. Thus, in general it took on average three hours for each execution of this algorithm but this is also considering the limited number of iterations that the loops were allowed to go through to calculate the individual integrals. The value of r in (4-62) was only run up to and including four while in the other iterative loop, only the case of $m=M$ and $\Omega=3$ was calculated since the corresponding integral in particular involved a lengthy computation. Unlike what was done in the Beaulieu FSE Technique, where the corresponding value of M was selected based on an upper bounding of the associated truncation error, in this case the value of M was initially fixed for each SNR value in the scenarios investigated and no optimization for the value of M was done although the algorithm executed properly and satisfactory results were obtained for the values of M initially selected. Since only the halfband filter coefficients were quantized, the BER degradation would mainly be affected by the parameters associated with the halfband filters, in this case, B and $Ltap$. The value of $Itap$, however, does still have a role in this analysis. The effect of all these parameters on the numerical results will be clarified in the next section.

Despite these limitations with the MAPLE_V algorithm, there are some advantages compared to previous calculations. Unlike the quantization array for the moments as described by (4-75), the computation of the complex integral expressions of (4-62) went smoothly. The main reason being that these integral expressions do not expand to as many terms as was the case in (4-75), therefore there was less memory needed, and there was a single variable of integration in this case. In (4-75), depending on the scenario, it would not be unrealistic to have 50 numerical integrations to perform. Although from (4-75) we see that the integration is straightforward, MAPLE_V has difficulties in performing them [36].



Preparation Phase
Declare Arrays
Computation of the filter coefficients
Compute the interference and quantization coefficients
continued in Figure 4-3(b)

Figure 4-3(a). First half of flow diagram for the Gram-Charlier Series Expansion Algorithm.



continued from Figure 4-3(a)
Compute maximum interference, maximum quantization & noise adjustment factor
max_inter = max_interference(...)
max_quant= max_quantization(...)
noise_adj = noise_adjustment(...)
Compute the moments of the interference, quantization and correlation
Initialize the quantization array
Initialize the interference and correlation array
Compute the interference and quantization for individual ISI terms
Compute the interference and quantization for individual ICI terms
Compute Gaussian noise moments for the quantization
Compute the correlation between interference and quantization
Compute the final moments of the interference, quantization and correlation
Quantization_Moments = [illegible](...)
Corr_Moment = [illegible](...)
Correlation_Moment = Corr_Moment(3)
Moments = [illegible](...)
Compute the degradation in BER relative to infinite quantization
Compute the infinite quantization moments
[C,weights,abscissas] =GQR(...)
Compute the coefficients a and b
Compute the BER for infinite quantization and Pe
Compute BER assuming finite quantization by loading in pre-computed quantization BER coefficients
Save & print the output
Print the output to the screen
Print the output to a data file HMM_QUANT.DAT
Save the results to HMM_QUANT.MAT
Save the figure to HMM_QUANT.PS

Figure 4-3(b). Second half of flow diagram for the Gram-Charlier Series Expansion Algorithm.

4.4.3 : Numerical Results of the M-MCDD Considering Quantization

4.4.3.1 : Organization of Results

The analytical tool using the Gram-Charlier Series technique has produced performance degradation curves due to the presence of quantization noise resulting from the numerical quantization of the halfband filter coefficients. These degradation curves show the amount of degradation in SNR, relative to infinite quantization, for varying numbers of quantization bits per halfband filter coefficient, *B*. From these degradation curves, we will first look into the numerical accuracy of the results, meaning the truncation errors and CPU time will be analyzed. We will then extend the evaluation of the system performance of the M-MCDD by looking at the amount of degradation as a function of the number of quantization bits for different scenarios. Thus, for a fixed scenario, if we know the degradation level we are willing to endure, we can successfully select the number of quantization bits. As mentioned in the previous section, however, there were limiting factors which may have affected the overall accuracy. As a result, the upper bound on the BER is not smooth for performance levels between 10^{-8} and 10^{-4} , thus in evaluating the amount of degradation, we look at a fixed SNR for finite quantization instead of a fixed performance level. Also, observations will be made from the degradation curves as to the effect of the values of *Ltap*, *Itap* and *B* on the degradation for finite quantization.

4.4.3.2 : Presentation of BER Curves

Degradation curves will now be presented for three different scenarios to show the effect of finite quantization on the performance of the M-MCDD. Table 4-1 and Figure 4-4 show the numerical and graphical depiction, respectively, of the performance degradation due to the presence of quantization noise for the scenario $Ltap=13$, $Itap=9$. We recall that *Ltap* are the non-zero taps per halfband filter while *Itap* are the filters taps in the channel detection filter. In this case, the number of quantization bits per halfband filter coefficient, *B*, is increased from four to eight to show the reduction in performance degradation resulting from an increase in the quantization error levels. By increasing *B*, we are limiting the maximum error, referring to the p.d.f. in Figure 4-1. Also, the value of the quantization array will be smaller with an increase in *B*, referring to (4-75). Table 4-1 shows also the number of points taken in the Taylor Series Expansion, *M*, for each SNR and also the truncation error and CPU time associated with choosing the number of quantization bits as $B=4$ and $B=8$. Likewise, Figure 4-4 shows the corresponding BER curves for finite quantization for the cases between $B=4$ and $B=8$ to show the reduced degradation levels. Also shown for comparison are the results for infinite quantization resulting from the GQR technique and also the ideal QPSK demodulation BER curve. Similarly, the results of two other scenarios are shown for comparison to the *Ltap=13* , *Itap=9* case. These two scenarios are *Ltap=9* , *Itap=12*, shown in Table 4-2 and Figure 4-5, and *Ltap=9* , *Itap=15*, shown in Table 4-3 and Figure 4-6.

Table 4-1. Degradation in BER due to filter quantization for the scenario *Ltap=13* and *Itap=9*.

		$B = \infty$	$B = 4$ CPU time → 126 : 38.9667		$B = 8$ CPU time → 143 : 27.4833	
Eb/No (dB)	M	P_b	P_b	R_M	P_b	R_M
0	4	8.1120e-02	9.7791e-02	1.5200e-04	8.1121e-02	1.5652e-04
1	6	5.8981e-02	8.4512e-02	1.4554e-55	5.8983e-02	1.5164e-55
2	8	4.0272e-02	6.3503e-02	3.2364e-107	4.0274e-02	3.4119e-107
3	10	2.5496e-02	4.5547e-02	1.8753e-159	2.5497e-02	2.0003e-159
4	12	1.4745e-02	3.1722e-02	3.0504e-212	1.4747e-02	3.2922e-212
5	14	7.6597e-03	2.0981e-02	1.4674e-265	7.6613e-03	1.6024e-265
6	16	3.5074e-03	1.2546e-02	2.1695e-319	3.5087e-03	2.3972e-319
7	18	1.3876e-03	6.6461e-03	**underflow**	1.3886e-03	**underflow**
8	20	4.6479e-04	3.5551e-03	**underflow**	4.6564e-04	**underflow**
9	22	1.2936e-04	1.5310e-03	**underflow**	1.2994e-04	**underflow**
10	24	2.9472e-05	6.4171e-04	**underflow**	2.9916e-05	**underflow**
11	26	5.4499e-06	2.8949e-04	**underflow**	5.9229e-06	**underflow**
12	28	8.1904e-07	1.6952e-04	**underflow**	1.5863e-06	**underflow**
13	30	1.0141e-07	1.7775e-04	**underflow**	1.4280e-06	**underflow**
14	32	1.0635e-08	8.9800e-05	**underflow**	7.1970e-07	**underflow**
15	34	9.8479e-10	4.9617e-05	**underflow**	3.9761e-07	**underflow**

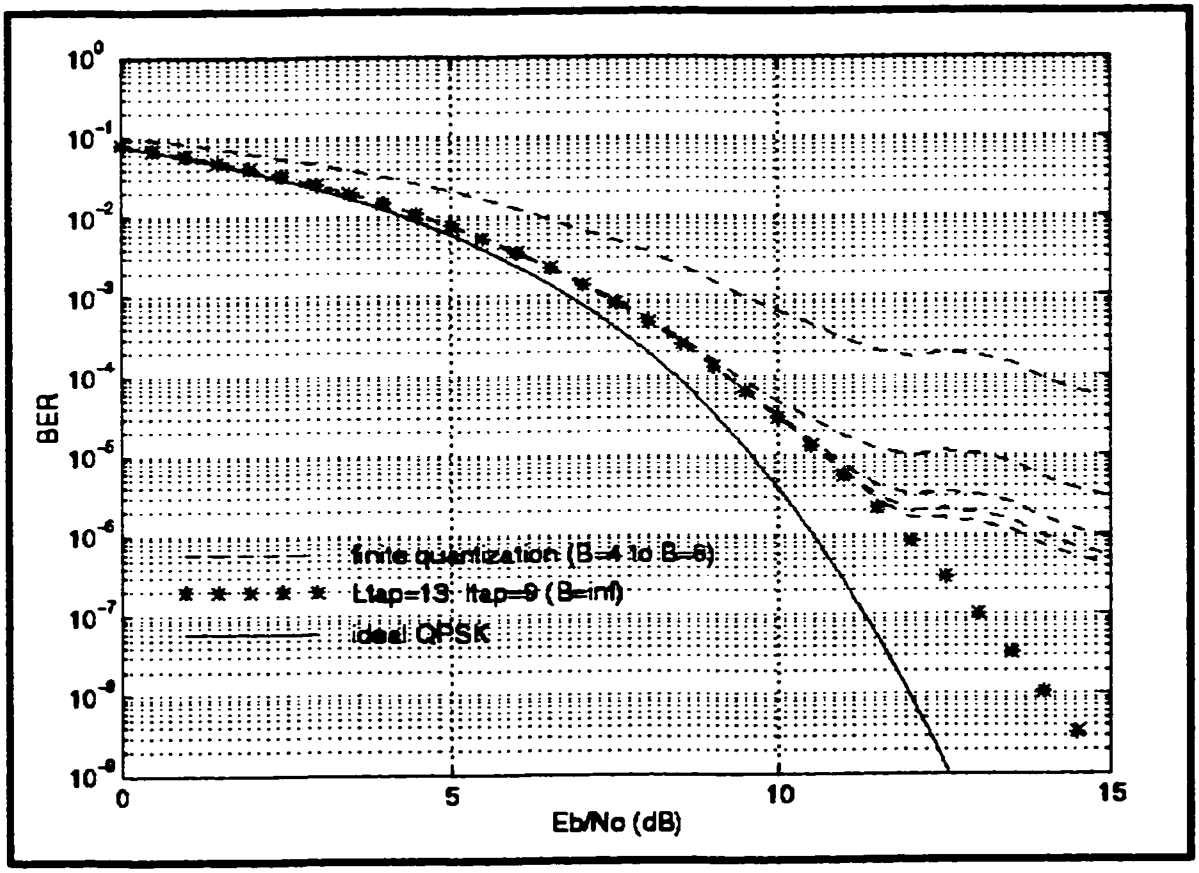


Figure 4-4. Degradation in BER using *B=4* to *B=8* quantization bits for *Ltap=13* and *Itap=9*.

Table 4-2. Degradation in BER due to filter quantization for the scenario *Ltap=9* and *Itap=12*.

		$B = \infty$	$B = 4$ CPU time → 124 : 56.1167		$B = 8$ CPU time → 142 : 38.7167	
Eb/No (dB)	M	P_b	P_b	R_M	P_b	R_M
0	4	8.1149e-02	8.5623e-02	1.6614e-02	8.1150e-02	1.7466e-02
1	6	5.9013e-02	6.6328e-02	1.0401e-52	5.9014e-02	1.1155e-52
2	8	4.0305e-02	4.7203e-02	1.5122e-103	4.0306e-02	1.6546e-103
3	10	2.5527e-02	3.1759e-02	5.7288e-155	2.5528e-02	6.3950e-155
4	12	1.4772e-02	2.0400e-02	6.0925e-207	1.4774e-02	6.9384e-207
5	14	7.6809e-03	1.2446e-02	1.9162e-259	7.6822e-03	2.2263e-259
6	16	3.5220e-03	6.9988e-03	1.8523e-312	3.5230e-03	2.1957e-312
7	18	1.3962e-03	3.5556e-03	**underflow**	1.3970e-03	**underflow**
8	20	4.6905e-04	1.8393e-03	**underflow**	4.6975e-04	**underflow**
9	22	1.3110e-04	7.8409e-04	**underflow**	1.3157e-04	**underflow**
10	24	3.0047e-05	3.2816e-04	**underflow**	3.0413e-05	**underflow**
11	26	5.6023e-06	1.4887e-04	**underflow**	5.9971e-06	**underflow**
12	28	8.5129e-07	8.8102e-05	**underflow**	1.5267e-06	**underflow**
13	30	1.0693e-07	9.3713e-05	**underflow**	1.3523e-06	**underflow**
14	32	1.1420e-08	4.8213e-05	**underflow**	6.9364e-07	**underflow**
15	34	1.0811e-09	2.7236e-05	**underflow**	3.9178e-07	**underflow**

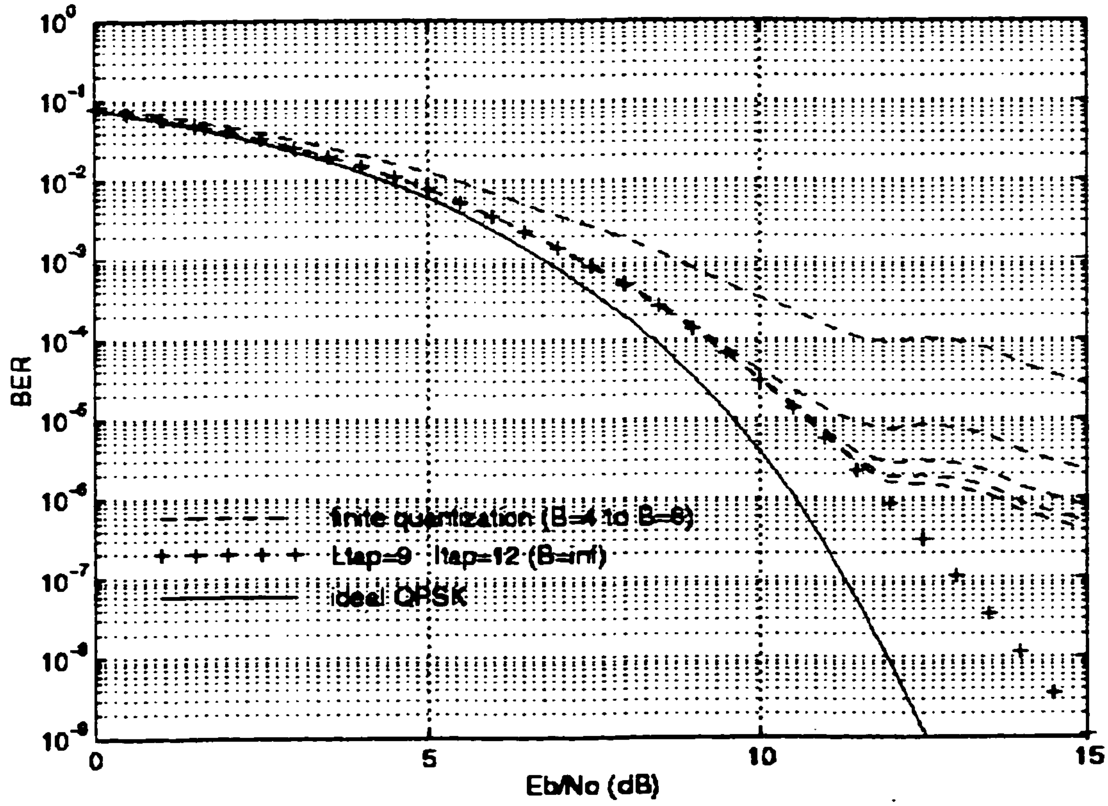


Figure 4-5. Degradation in BER using *B=4* to *B=8* quantization bits for *Ltap=9* and *Itap=12*.

Table 4-3. **Degradation in BER due to filter quantization for the scenario** ***Ltap=9*** **and** ***Itap=15.***

		$B = \infty$	$B = 4$		$B = 8$	
			CPU time → 142 : 07.0167		CPU time → 163 : 26.7000	
Eb/No (dB)	M	P_b	P_b	R_M	P_b	R_M
0	4	8.0375e-02	8.4807e-02	1.9245e+01	8.0376e-02	1.9612e+01
1	6	5.8166e-02	6.5376e-02	2.0252e-48	5.8168e-02	2.0795e-48
2	8	3.9434e-02	4.6184e-02	4.9493e-98	3.9436e-02	5.1203e-98
3	10	2.4698e-02	3.0727e-02	3.1518e-148	2.4699e-02	3.2854e-148
4	12	1.4053e-02	1.9407e-02	5.6342e-199	1.4054e-02	5.9176e-199
5	14	7.1231e-03	1.1542e-02	2.9786e-250	7.1243e-03	3.1521e-250
6	16	3.1440e-03	6.2476e-03	4.8400e-302	3.1449e-03	5.1608e-302
7	18	1.1777e-03	2.9991e-03	**underflow**	1.1783e-03	**underflow**
8	20	3.6399e-04	1.4273e-03	**underflow**	3.6453e-04	**underflow**
9	22	9.0134e-05	5.3906e-04	**underflow**	9.0458e-05	**underflow**
10	24	1.7375e-05	1.8976e-04	**underflow**	1.7586e-05	**underflow**
11	26	2.5418e-06	6.7544e-05	**underflow**	2.7210e-06	**underflow**
12	28	2.7705e-07	2.8673e-05	**underflow**	4.9685e-07	**underflow**
13	30	2.2340e-08	1.9579e-05	**underflow**	2.8252e-07	**underflow**
14	32	1.3440e-09	5.6745e-06	**underflow**	8.1637e-08	**underflow**
15	34	6.2002e-11	1.5619e-06	**underflow**	2.2468e-08	**underflow**

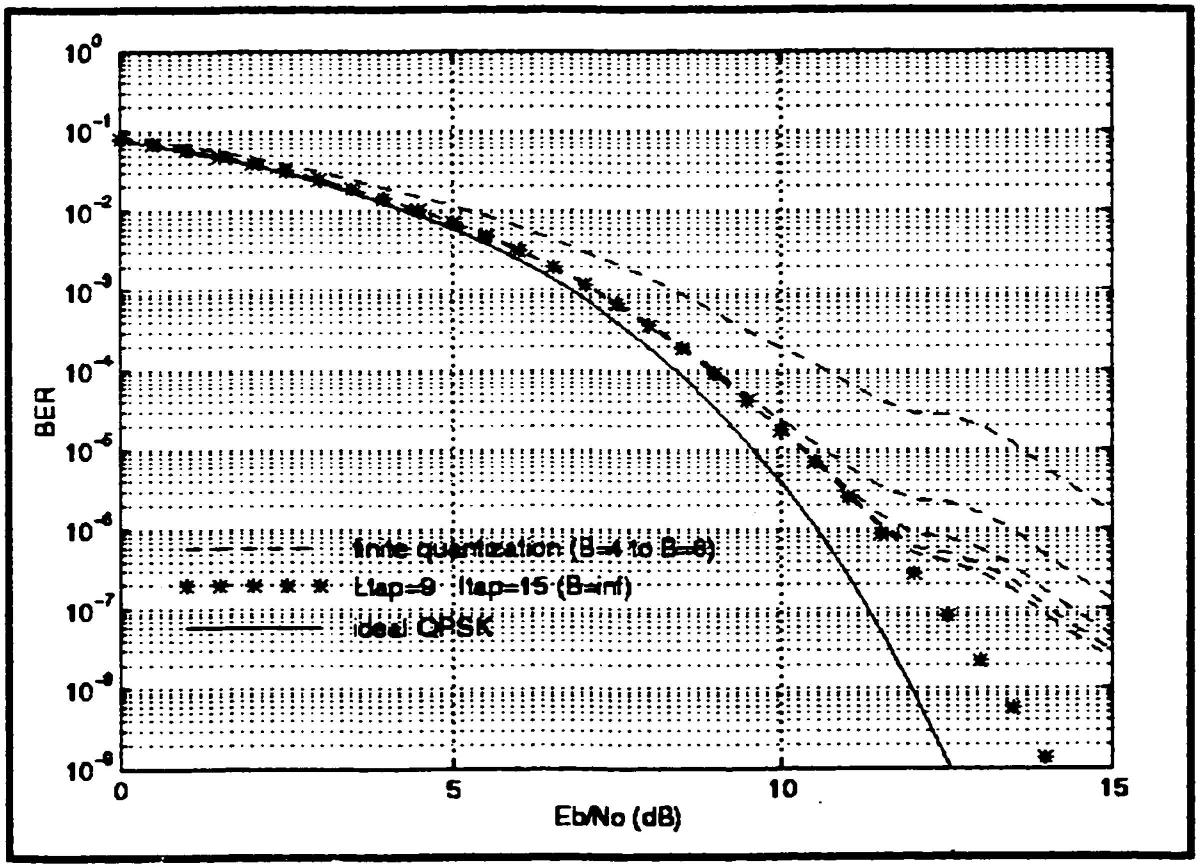


Figure 4-6. **Degradation in BER using** ***B=4*** **to** ***B=8*** **quantization bits for** ***Ltap=9*** **and** ***Itap=15.***

4.4.3.3 : Numerical Accuracy

We will first make some observations as to the numerical accuracy common to all three scenarios presented in the previous section. The truncation error due to using a finite number of terms in the Taylor Series, R_M, in all three scenarios is seen to become exponentially smaller with an increase in SNR until, at 7dB, we see an underflow in its value. The only unusual observation is in the case of *Ltap=9*, *Itap=15* at 0dB, referring to Table 4-3. At that particular point, we see that R_M actually exceeds the value of P_b. Because all the truncation errors are relatively small, this may either indicate an instability in R_M for 0dB or possibly that P_b is inaccurate at this SNR value. But because in the other scenarios the truncation error is not as large at 0dB, we may side with the fact that there is some instability in R_M for *Ltap=9*, *Itap=15* at 0dB. We can also make observations for the CPU time requirements of the Gram-Charlier Series algorithm. First of all, in comparing the CPU times from Tables 4-1 through 4-3 for either *B=4* or *B=8* with those of the corresponding scenarios in Table 3-10 for the GQR technique, we see that the difference in CPU times between Tables 4-1 through 4-3 is proportional to the difference in CPU times in Table 3-10. This is expected because the Gram-Charlier Series algorithm makes use of the GQR program to generate the infinite quantization BER and also the moments of the quantization noise are generated in a manner similar to those of the interference in the GQR algorithm. Thus, a similar proportion of CPU times is expected between both algorithms. Secondly, we note that in the Gram-Charlier Series algorithm, the number of quantization bits, *B*, is used as a constant in generating the moments of the quantization noise therefore it is not expected that the CPU time would vary greatly with the selection of *B*. Indeed, from Tables 4-1 through 4-3, we see that in comparing *B=4* to *B=8* in each case, there is not a significant increase in CPU time. The CPU time is seen to increase slightly but for the most part, the percentage increase is relatively small. Also, as mentioned in section 4.4.2, the number of points used in the Taylor Series was fixed for each SNR since satisfactory results were obtained. Perhaps more accurate results could have been obtained, as well as a smoother BER curve for higher SNRs, if the number of points were increased but this was not looked into.

4.4.3.4 : System Performance

Before looking into the choice of *B* and how we may go about this, we will make some observations on the effect of *Ltap* and *Itap* on these numerical results based on what is shown for all three scenarios. First, we compare the cases of *Ltap=13*, *Itap=9* and *Ltap=9*, *Itap=12* for a change in the value of *Ltap*. In comparing Figures 4-4 and 4-5, respectively, we see that there is more degradation in the case of $Ltap=13$ if we compare between $B=4$ and $B=\infty$. However, at the same time, as B increases from four to eight, we see a larger improvement in performance in the case of $Ltap=13$ until at $B=8$, in both cases, the BER curves are almost the same. This can also be verified by looking at Tables 4-1 and 4-2, respectively. The $B=\infty$ BERs in the $Ltap=13$ case are slightly smaller in value but the corresponding $B=4$ BERs are higher valued, thus creating a larger difference between $B=4$ and $B=\infty$ in the $Ltap=13$ case. In both cases, for $B=8$, we see little difference. What this indicates is that there is a direct effect of *Ltap* on the BER due to quantization. As the value of *Ltap* increases, we are introducing more halfband filter coefficients which need to be quantized. Quantization noise will result when each halfband filter tap is numerically quantized to B significant bits [9]. Thus, because there are l stages of demultiplexing, for a larger value of *Ltap*, more quantization noise appears at the output of the channel detection filter [14]. Thus, as *Ltap* increases, we see a significant increase in degradation due to quantization. This is more apparent for $B=4$. But if we were to increase B from four to five, the amount of quantization noise for each coefficient would be reduced by as much as half, referring to Figure 4-1 [9] [31]. Therefore, if we were to have more coefficients, the reduction in total system quantization noise between $B=4$ and $B=5$ would be larger. That is why a larger *Ltap* value results in a more significant improvement in performance between $B=4$ and $B=5$. If we then go to higher B values, the reduction in quantization noise decreases until we get to $B=8$ where, in comparing the cases $Ltap=13$ and $Ltap=9$, we see very little difference in BER.

We can now compare the cases of $Itap=12$ and $Itap=15$ with $Ltap=9$ constant to see the effect of increasing *Itap*. In comparing Tables 4-2 and 4-3, respectively, for the $B=\infty$ BER, we see the performance is improved in the case of $Itap=15$. By interpolation, we can see approximately 0.82dB improvement at 10^{-5}. Likewise, if we compare the $B=4$ curve in both cases, we can see a 1dB improvement at 10^{-4}. We look at 10^{-4} in this case because at this BER, the curves are not distorted. In the case of the $B=8$ curves, because there is no distortion at 10^{-5}, we compare the two and see a 0.836dB improvement at 10^{-5} [34]. From this, we can see that there is an improvement in performance in the finite quantization BER which is consistent with the corresponding improvement in the $B=\infty$ BER. Qualitatively, if we were to compare both sets of results for the difference between $B=4$ and $B=\infty$, we would see no significant difference between the two. This, together with the fact that we observe no difference in improvement with an increase in B between both scenarios, means that increasing *Itap* in this analysis does not affect the finite quantization BER as does *Ltap* but rather creates a significant improvement in the BER values for $B=\infty$. This significant improvement for $B=\infty$. will decrease the finite quantization BER values correspondingly. We recall that this was one characteristic of *Itap* observed in Chapter Three, meaning that an increase in *Itap* created a more significant improvement in performance compared to an increase in *Ltap*. This means that *Itap* affects the overall performance rather than just mainly the degradation due to quantization, which is what was observed with *Ltap*. We can also observe that an increase in *Itap* will cause the finite quantization BER curves to be less distorted for SNRs above 12dB.

To summarize the effects of increasing *Ltap* and *Itap* on the degradation and to further summarize more of the results of the Gram-Charlier Series algorithm, three degradation versus *B* curves have been compiled. These curves depict the degradation in SNR with respect to the corresponding infinite quantization, that is, the additional SNR required to achieve a desired performance level, versus the number of quantization bits per halfband filter coefficient. To evaluate how much degradation with respect to infinite quantization is experienced as *B* is varied, we look at a specific M-MCDD scenario, that is, for one value of *Ltap* and *Itap* as well as a fixed performance level. The degradation versus *B* curves, shown in Figures 4-7 to 4-9 were generated by observing performance degradation curves, such as those shown in Figures 4-4 through 4-6. Table 4-4 summarizes the degradation as a function of *Ltap*, *Itap* and *B* while Figures 4-7 through 4-9 each show the corresponding plots for varying *Ltap* values (*Ltap=5*, *9* and *13*) while keeping *Itap* at the constant value of nine, twelve and fifteen in each plot, respectively. In each of these plots, because of the distorted BER curves for finite quantization, we look at the amount of degradation at a fixed SNR of 13dB. At 13dB, the BERs vary between 10^{-3} and 10^{-4}.

Table 4-4. Degradation (relative to $B=\infty$) in dB due to finite quantization for SNR=13dB.

	Ltap = 5					Ltap = 9					Ltap = 13				
B	4	5	6	7	8	4	5	6	7	8	4	5	6	7	8
Itap = 9	2.95	1.75	1.46	1.30	1.24	4.00	2.20	1.65	1.40	1.28	4.21	2.37	1.80	1.45	1.32
Itap = 12	2.82	1.72	1.45	1.27	1.14	3.83	2.13	1.58	1.32	1.18	4.10	2.30	1.70	1.37	1.22
Itap = 15	2.65	1.49	1.23	1.04	0.96	3.35	1.86	1.36	1.10	1.00	3.85	2.06	1.54	1.20	1.06

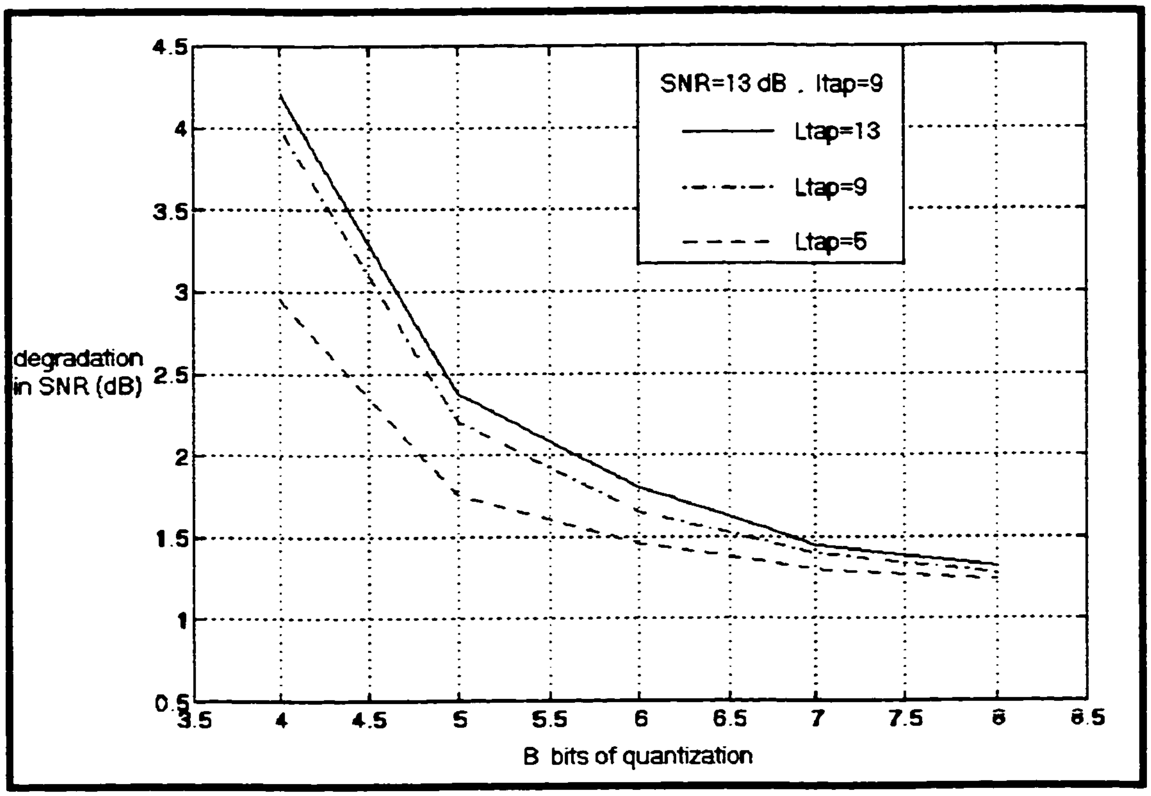


Figure 4-7. Degradation in performance relative to $B=\infty$ for various *Ltap* values keeping *Itap=9*.

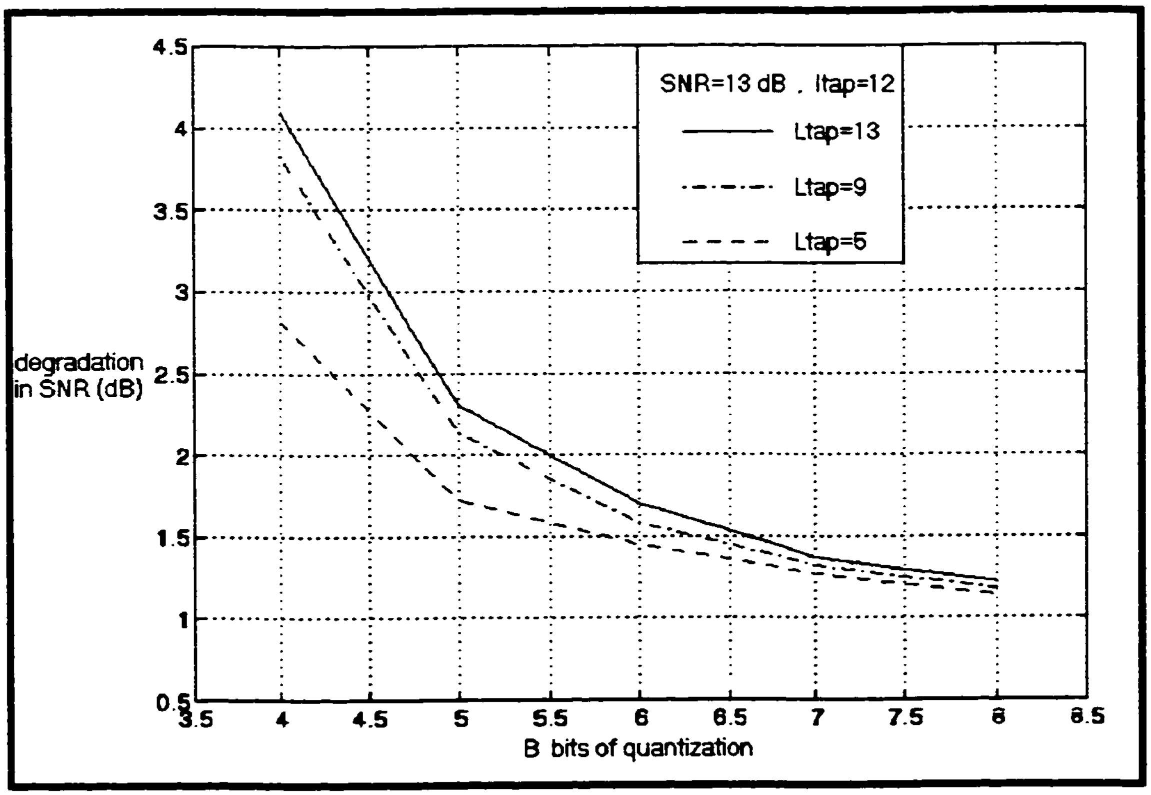


Figure 4-8. Degradation in performance relative to $B=\infty$ for various *Ltap* values keeping *Itap=12*.

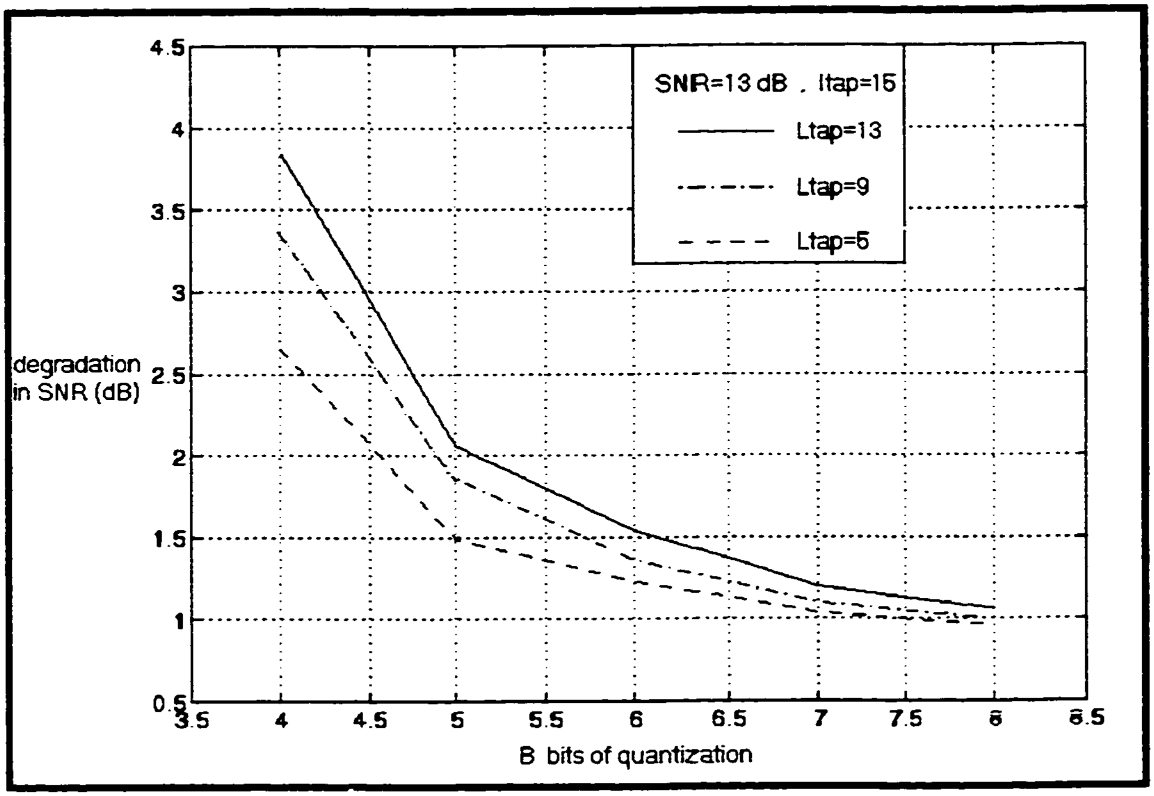


Figure 4-9. Degradation in performance relative to $B=\infty$ for various *Ltap* values keeping *Itap=15*.

From each of these plots, we clearly see that the degradation decays exponentially with *B*. We can also verify the fact that for small *B* values (ie. *B=4*), there is a significant degradation in performance as *Ltap* increases but as *B* increases, this change in degradation decreases until at *B=8*, we see very little difference in degradation between the *Ltap* values. This value of *B* can be considered large enough so as not to make a significant difference in degradation with an increase in *Ltap*. Also, for large *Ltap* values (ie. *Ltap=13*), we clearly see a larger improvement in performance from *B=4* to *B=5* compared to the smaller *Ltap* values. In the case of *Itap*, between all three curves, we see no difference in the relative shape of the different *Ltap* curves but only in their relative position. That is, as *Itap* increases, we generally see less degradation in performance however, it does not make a difference in improving performance between values of *Ltap*. The reason why we see less degradation in performance is because the smaller $B=\infty$ BER values cause the finite quantization curves to be lower on the BER scale as well. Therefore, an increase in *Itap* does not affect the finite quantization as does an increase in *Ltap*. Depending on the value of *B*, an increase in *Ltap* has a more direct effect on the degradation due to quantization than does an increase in *Itap*. Also, because as $B \rightarrow \infty$ we expect no quantization error, in all three plots, we expect the degradation to slowly approach zero as $B \rightarrow \infty$. This can be seen by observing Figures 4-7 through 4-9.

Apart from verifying the effects of varying *Ltap* and *Itap* on the degradation due to quantization, the degradation vs. *B* plots of Figures 4-7 through 4-9 can be used as design tools when having to select the value of *B*. Two specific examples will now be provided based on the examples provided in Chapter Three on how to choose *Ltap* and *Itap*. In these examples, it has been assumed that the desired SNR level is fixed at 13dB for finite quantization and at this level, the BER can be anywhere between 10^{-8} and 10^{-4}. Recall, this was the assumption used in generating Figures 4-7 to 4-9.

Example 4-1.

We look back to Example 3-3 where we try to achieve the best possible degradation and performance levels. The desired levels were 1dB to 1.5dB and 10^{-5} to 10^{-6} , respectively. If we remove the restriction of a maximum complexity level, we can then focus on achieving $P_b = 10^{-6}$ and Δ=1dB. From Table 3-15, we can then choose the combination *Ltap=13* , *Itap=12* since its complexity level is lower.

Now, if we assume we have finite halfband filter coefficient quantization, then we must also tolerate a further degradation relative to infinite quantization. We will specify a maximum tolerable degradation of 1dB due to quantization, that is Δ_q =1dB, giving us a total of 2dB degradation relative to ideal QPSK demodulation (Δ_t =2dB) . Thus, at this degradation level, we see from Table 4-4 that the minimum degradation we can achieve is Δ_q =1.22dB and this is when using *B=8* quantization bits. Since this is closest to the specified 1dB degradation, we can indeed choose *B=8* for the scenario *Ltap=13* , *Itap=12*. Therefore, using *Ltap=13* , *Itap=12* with *B=8* quantization bits, we can achieve a performance level of $P_b = 10^{-6}$ with Δ_t =2.22dB degradation relative to ideal QPSK demodulation.

Example 4-2.

Also from Example 3-3, if we do specify a maximum complexity level of M_{op}=4·10^{6} operations while trying to achieve the best possible performance level, we can choose *Ltap=9* , *Itap=12*. This scenario, according to Table 3-16, achieves $P_b = 10^{-6}$ with Δ=1.5dB degradation. We note from Table 3-16 that *Ltap=11* , *Itap=9* is also a possibility since this also achieves $P_b = 10^{-6}$ however, we choose the former scenario in accordance with the data available in Table 4-4, also noting that there is not much savings in complexity in choosing the latter case.

If we now assume that we have finite halfband filter coefficient quantization using between *B=4* and *B=8* bits and that we want the degradation due to quantization to be less than Δ_q =1.5dB, giving us a total of Δ_t =3dB degradation relative to ideal QPSK demodulation, the possible choices for *B* from Table 4-4 are summarized in Table 4-5 below.

Table 4-5. Degradation levels for different *B* values for *Ltap=9* , *Itap=12*.

B	Δ_q	$\Delta_{q,B-1} - \Delta_{q,B}$
6	1.58	-
7	1.32	0.26
8	1.18	0.14

So, we would be able to choose *B=6* , *7* or *8* to achieve no more than Δ_t =3dB total degradation. In the case of *B=6*, we achieve slightly over 3dB, that is, Δ_t =3.08dB. The deciding factor may be the cost of having each of the three *B* values implemented and, depending on the design specifications, to see which alternative is more feasible. However, judging by the reduction in degradation as we increase the number of quantization bits, it is shown from Table 4-5 that there is more improvement by increasing from *B=6* to *B=7*, giving 0.26dB improvement, than by increasing from *B=7* to *B=8*, giving only 0.14dB improvement. Therefore, using this criteria, we may choose *B=7* quantization bits giving a slightly reduced degradation compared to *B=6* while not using as many quantization bits as in the case of *B=8*. Thus, if we choose *B=7* quantization bits for the scenario *Ltap=9* , *Itap=12*, we can achieve $P_b = 10^{-6}$ at Δ_t =2.82dB degradation relative to ideal QPSK demodulation.

To summarize the results of both examples, Table 4-6 is compiled below.

Table 4-6. Summary of the results from Examples 4-1 and 4-2.

Example	(*Ltap* , *Itap*)	*B*	P_b	Δ	Δ_q	$\Delta_t = \Delta + \Delta_q$
4-1	(13,12)	8	10^{-6}	1dB	1.22dB	2.22dB
4-2	(9,12)	7	10^{-6}	1.5dB	1.32dB	2.82dB

4.4.4 : Summary of Numerical Results

To summarize the effect of *Ltap* and *Itap* on the finite quantization BER analysis, we look at the degradation vs. *B* curves of Figure 4-7 through 4-9. We clearly see the degradation in performance as *Ltap* is increased due to the augmented quantization noise to the entire system resulting from more halfband filter taps. At the same time, for increasing *Ltap* values, we see the larger improvement in performance as *B* is increased from four to eight by the decreasing slope of the curves. In fact, we see a more significant improvement in performance for small values of *B* as *B* increases. We also see that as *B* approaches eight, the change in degradation decreases, especially for smaller *Ltap* values. This is expected because as *B* increases, we are introducing less and less quantization noise per halfband filter coefficient. Once the value of *B* reaches eight, we don't see a great change in degradation. In fact, it was found that there was no need to increase *B* beyond eight since we would not make any significant observations regarding performance. At the same time, even if *Ltap* increases from five to thirteen, the amount of degradation will not increase substantially for $B=8$. From this analysis, however, it is expected that if *B* was allowed to approach infinity, the amount of quantization noise which would be added to the system would become smaller and smaller until no quantization noise would be present. This would mean that at $B=\infty$, the amount of degradation relative to infinite quantization would be 0dB for any value of *Ltap* chosen. The effect of *Itap* on the BER performance due to quantization is not the same as that of *Ltap*. Comparing any two *Itap* cases, we observe no difference in a relative change of *Ltap* and *B*. An increase in *Itap*, however, did decrease the overall degradation but this is expected to be due to the vast improvement in the infinite quantization BER. This was actually an observation made in Chapter Three as we compared an increase in *Itap* versus an increase in *Ltap*. The effect of a change in *Ltap* and/or *B* on the quantization is therefore more direct than the effect of a change in *Itap*. That is not to say that their change has more of an impact on the overall degradation but rather we can observe more of a relative difference in degradation due to quantization if *Ltap* and/or *B* were to change than if *Itap* were to change, referring to Figures 4-7 through 4-9. Logically, this would be expected since the quantization noise in the system comes from a quantization of the halfband filter coefficients, whose quantity is affected by *Ltap*. Also, *B* affects the level of quantization of these coefficients. The value of *Itap* would have an impact on the overall performance of the M-MCDD. More specifically, it mainly effects the infinite quantization BER as opposed to the degradation due to quantization. All of this is supported by the numerical results presented.

As for the execution time of the algorithm, we can generally say that the CPU time required for the Gram-Charlier Series algorithm was quite large, although all attempts were made to minimize the CPU time as much as possible. In several of the more complicated cases, for example, *Ltap=11, Itap=15* and *Ltap=13, Itap=12* both with *B=8*, over three hours of CPU time was required to execute the algorithm. This length of CPU time, however, is not reflected in the results shown in Tables 4-1 through 4-3. For the simpler scenarios such as *Ltap=5* or *7* with *Itap=3* or *Itap=6* and *B=4*, it would take up to forty minutes of CPU time to complete the algorithm. In contrast, from Table 3-10, the execution time for the GQR algorithm was seen in general not to exceed thirty minutes. As for the accuracy of this method, we see that only the truncation error due to using $M+1$ terms in the Taylor Series for $Q(\cdot)$, R_M , has been calculated [30]. This is because truncation errors due to the upper bounding of each Gram-Charlier Series were initially calculated and observed to be small relative to R_M. Although these truncation errors were not presented in Tables 4-1 to 4-3, we can predict their magnitude by making an observation for the truncation error R_M. We see that it falls significantly with an increasing SNR. In fact, for SNRs above 6dB, we see an underflow occurring in R_M. This shows a high level of accuracy in the corresponding BER results, except possibly at 0dB SNR where R_M was seen to exceed a value of one in Table 4-3. In that case, an instability in R_M at 0dB may be observed. This accuracy for higher SNRs may be of some surprise since we easily see the distortion of the upper bound on the finite quantization probability of bit error for SNR values above 11dB. The reasons for this distortion will be discussed next. It was also observed that the analytical tool using the Gram-Charlier Series technique worked best when *Itap* was greater than nine. If *Itap* was selected to be less than nine, error floors would occur in the finite quantization BER curve.

From the results presented in Figures 4-4 to 4-6, we see that the upper bound on the BER for finite quantization is somewhat distorted for high SNRs since at these SNRs, the BERs do not decay logarithmically. There are several factors which may contribute to this occurrence. A contributing factor from the analysis may be the numerous inequalities that were used over and over again. Inequalities such as Schwartz's Inequality were used so many times in order to separate the interference and quantization variables [34]. Thus, this may have affected the upper bound for certain portions of the curve or for various SNR values. This also may have to do with the number of terms chosen in the Taylor Series expansion, since the number of terms was fixed and was not optimized. A lack of higher order moments for the quantization, however, could very well be the main contributing factor. We were generally limited to the fourth order moments in the calculation of the Gaussian noise moment of (4-52) and the ISI and ICI quantization moments, represented by (4-50) and (4-51), respectively, due to our limited resources.

To conclude, we now have the tools with which we may choose the number of quantization bits per halfband filter tap, B , knowing the values of *Ltap* and *Itap* to achieve the performance and degradation levels we want. We may then choose the value of B based on the maximum amount of degradation in the performance of the M-MCDD due to quantization noise we are willing to accept for BERs below 10^{-4}. Particularly in this analysis, we can conclude that the preferred values of B to choose from are between five and seven. Generally speaking, if we use $B=4$, we will suffer too much degradation compared to taking B a step higher to $B=5$. At the same time, if we compare $B=7$ to $B=8$, we see that there is very little improvement in performance if we select $B=8$. If it is absolutely necessary to minimize the degradation levels at the expense of potentially incurring a higher design cost, it would be better to use $B=8$, however this may not be merited. If one is willing to experience a slightly higher level of degradation, it would be recommended to keep $B=7$ in order not to potentially drive up design costs unnecessarily.

Chapter Five

Conclusion

The M-MCDD has been chosen as the preferred structure for the implementation of on-board processing (OBP) satellite communication systems because of its modularity, compactness, low structural and computational complexity and low power consumption, in comparison to previously designed multicarrier demodulators (MCD), namely the Single Stage Method (SSM) and Polyphase/FFT Method (PPM) [1-2] [6-7]. Although the concept of the M-MCDD as a group demodulator and the analytical techniques discussed in this thesis are well known [6-8] [10-12] [15-22] [25-26], the contribution of this research lies in the application of these analytical techniques to evaluate the performance of the M-MCDD [13-14]. This research has provided two key contributions which served to evaluate the performance of the M-MCDD. These two contributions are the presentation of the numerical accuracy of the results and the presentation of the system performance of the M-MCDD. Conclusions based on these results will be given.

The first contribution is that we have studied the numerical accuracy of the results provided. These numerical results were obtained through an AWGN analysis to evaluate the performance of the M-MCDD in the presence of ACI and also in the presence of quantization noise [9-11] [13]. Analytical techniques were needed because introducing interference and quantization noise into the demultiplexed output made solving the probability density functions virtually impossible because of the infinitely large amount of interfering symbols which needed to be considered [10-12].

Although analysis was chosen as the method to obtain the numerical results, there were actually two possibilities in effectively choosing the filtering requirements for the M-MCDD. The other possibility was to use simulation [2] [13]. The reason why analysis was chosen as the preferred method of evaluation for the M-MCDD is because simulation, which has been the method used to date, is known to be very time consuming. This is especially true for evaluation at low bit error rates [2]. The development of analytical tools for performance evaluation is a valuable asset since an analytical tool is significantly faster and more accurate than a simulation program [13]. Having said this, however, we did not go through and compare actual numerical results between simulation and analysis.

Thus, from the analysis, we have developed analytical tools to evaluate the derived probability of bit error expressions numerically. Particularly, the candidate analytical techniques which have been studied to evaluate the performance of the M-MCDD in the presence of ACI were the Gauss Quadrature Rule (GQR) Technique [10-12] [15] [17] and the Beaulieu Fourier Series Expansion (FSE) Technique [16]. The GQR technique was chosen as a candidate because the weights and abscissas, needed to determine the BER, could be generated quite easily thereby requiring a low level of processing power in its execution [2] [10-12] [15] [17]. The Beaulieu FSE technique was also considered because it involved a computation of the characteristic function of the interference, which is easily obtainable in contrast to the p.d.f. of the interference, which is next to impossible to obtain [2] [16]. These two techniques were compared in terms of their speed of execution. This comparison was done to determine which of the two techniques was better suited to carry out the performance analysis of the M-MCDD [2]. From the numerical accuracy of the results presented, it was concluded that the GQR Technique was undoubtedly better suited for the performance evaluation due to its relatively low CPU time requirements. We also looked at the BER curves and error measurements resulting from both techniques. From the numerical accuracy, we can also see how the parameters involved in the performance evaluation of the M-MCDD were selected. For example, *Nterms*, *mL* and *mU* were chosen so as to achieve the best possible accuracy while keeping the CPU timing requirements as low as possible [2].

To evaluate the performance of the M-MCDD in the presence of quantization noise as well as AWGN, the AWGN in the presence of interference was extended to investigate the effect of using finite halfband filter coefficient quantization on the BER performance of the M-MCDD. The Gram-Charlier Series Expansion Technique [14] [17-22] was used to evaluate the degradation in performance due to the presence of quantization noise. It was chosen due to its simplicity in generating its coefficients mathematically from the moments of the quantization, thereby requiring a low amount of CPU time in generating the BER results [21-22]. Overall, the analytical tools allowed us to evaluate the performance of the M-MCDD as a function of the number of non-zero halfband filter taps, *Ltap*, the number of channel detection filter taps, *Itap*, and the number of quantization bits per halfband filter coefficient, *B*. And although there were limitations cited to several techniques which may have affected the performance results of the M-MCDD, the results obtained were considered acceptable. That is to say, compared to theory, the analytical model behaved as expected. As the number of filter taps were increased, the performance of the M-MCDD converged to the ideal case.

The numerical results were also presented on another front, namely to determine the system performance of the M-MCDD based on a desired performance and degradation level. Therefore, the second contribution of this research is that we have provided tools with which a designer may choose the number of filter taps per halfband filter and channel detection filter as well as the number of quantization bits per halfband filter coefficient. The halfband filters and channel detection filter, which are a part of the M-MCDD, are FIR filters therefore their lengths or number of taps is finite [2] [6-7] [9]. Thus, the design problem we are faced with is how to effectively choose the number of filter taps required for the halfband filters (*Ltap*) and channel detection filter (*Itap*) in order to achieve a target performance. In addition, since infinite word length or real-valued filter coefficients are impossible to realize, we are required to know the finite number of quantization bits per halfband filter tap to achieve a desired degradation level relative to the infinite quantization scenario [9] [28]. Thus, using the numerical results, the number of filter taps and quantization bits per halfband filter coefficient could be chosen.

The BER curves have been compiled to form design tools such as BER×SNR isometric contours and degradation versus *B* curves. Thus, using the isometric contours, we may choose the values of *Ltap* and *Itap* as a pair in order to achieve the performance level and degradation level relative to ideal QPSK. Also, computational complexity was used as a design criteria and a trade off was seen between performance improvement and complexity in the selection of *Ltap* and *Itap*. Thus, we can also choose from the possible pairs of values of *Ltap* and *Itap* by referring to the complexity involved in computing the BER of the M-MCDD. That is to say, from the performance and degradation levels we wish to achieve, we may have several candidate *Ltap* and *Itap* pairs to choose from. If we specify a maximum level of computational complexity of the M-MCDD we wish to achieve, the choices may be more limited, making our decision much more focused. And, once the values of *Ltap* and *Itap* are chosen, we may choose the number of quantization bits per halfband filter coefficient in order to achieve the degradation level relative to infinite quantization, by referring to the curves depicting the degradation in SNR versus the number of quantization bits, *B*.

From the numerical results, we have made some general conclusions as to the better choices for the filter taps and quantization bits. We have concluded that using small lengths for the channel detection filter should be avoided due to the high level of degradation in performance associated with these filter lengths. For example, using a filter length of three resulted in exceptionally high degradation, relative to ideal QPSK, compared to using filter lengths of six or larger. When choosing the halfband filter lengths, we did not find any outlying values that caused significant degradation in performance, unlike what was observed concerning the length of the channel detection filter. Also, for the number of quantization bits, it was concluded that using four quantization bits per halfband filter coefficient resulted in very high degradation compared to using five bits, while using eight quantization bits showed negligible improvement in performance compared to using seven bits. Thus, it would not merit using eight bits as opposed to seven bits due to the higher design cost that potentially may be incurred in using that many number of bits. Therefore, the best range of bits to choose from, in general, was between five and seven.

In summary, the theme of the research behind this thesis is how to design a Multistage Multicarrier Demultiplexer/Demodulator (M-MCDD) group demodulator for the implementation of on-board processing (OBP) satellite communication systems [1] [6-8]. We achieved the two main objectives stated at the beginning of this thesis. On one hand, we evaluated the performance of the M-MCDD in the presence of AWGN and ACI to develop an analytical model. This model was used to effectively choose the filtering requirements of the M-MCDD. Secondly, the effect of finite word length halfband filter coefficients on the performance of the M-MCDD was investigated in order to extend this analytical model. We were then able to effectively choose the number of quantization bits per halfband filter tap. To summarize these objectives, the interest was to design the M-MCDD based on the number of filter taps and quantization bits per halfband filter tap in order to achieve a desired level of performance and degradation.

References

[1] W. F. Yim et al, "Multi-Carrier Demodulators for On-Board Processing Satellites", *Int. Jour. of Sat. Comm.*, vol. 6, pp. 243-251, 1988.

[2] N. P. Secord, "Analysis of Interference Effects in Satellite On-Board Regenerative Repeaters", *CRC report# CRC-RP-96-002*, April 1996.

[3] M. J. Riezenman, "Communications", *IEEE Spectrum* , vol. 35 no. 1, pp. 29-36, January 1998.

[4] E. Del Re and R. Fantacci, "Alternatives for On-Board Digital Multi-Carrier Demodulation", *Int. Jour. of Sat. Comm.*, vol. 6, pp.267-281, 1988.

[5] J. Gilderson and J.Cherkaoui, "On-Board Swtiching for ATM Via Satellite", *IEEE Communication Magazine*, vol. 35 no. 7, pp. 66-70, July 1997.

[6] H. Göckler and H. Eyssele, "Study of On-Board Digital FDM-Demultiplexing for Mobile SCPC Satellite Communications - Parts I and II", *European Trans. Telecomm. Systems*, vol. 3, pp.7-30, 1992.

[7] H. Göckler, "A Modular Multistage Approach to Digital FDM Demultiplexing for Mobile SCPC Satellite Communications", *Int. Jour. of Sat. Comm.*, vol. 6, pp.283-288, 1988.

[8] H. Scheuermann and H. Göckler, "A Comprehensive Survey of Digital Transmultiplexing Methods", *Proc. of the IEEE*, vol. 69, no. 11, 1981, pp.1419-1450, *Proc. of the IEEE*, vol. 70, no. 7, 1982, pp.774-775.

[9] A.V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, Prentice Hall, New Jersey, 1989.

[10] S. Benedetto et al., "Error probability in the Presence of Intersymbol Interference and Additive Noise For Mulitlevel Digital Signals", *IEEE Trans. Commun. Tech.*, vol. COM-21, pp.

181-190, 1973.

[11] S. Benedetto et al., "Combined Effects of Intersymbol, Interchannel and Co-channel Interference in M-ary CPSK Systems", *IEEE Trans. Commun. Tech.*, vol. COM-21, pp. 997-1008, 1973.

[12] S. Benedetto et al., "Moment-based Performance Evaluation of Digital Transmission Systems", *IEE Proceedings - Part I*, vol. 139, no. 3, pp.258-266, June 1992.

[13] D.Salhany et al., "Performance Analysis of a Multistage Multicarrier Demultiplexer /Demodulator", ICC'97, June 8-12 1997, Montreal.

[14] D.Salhany et al., "Effects of Filter Quantization on a Multistage Multicarrier Demultiplexer/Demodulator", ICC'98, June 7-11 1998, Atlanta.

[15] G. H. Golub and J. H. Welsch, "Calculation of Gauss Quadrature Rules", *Math. Comput.*, vol. 23, pp. 221-230, 1969.

[16] N. C. Beaulieu, "The Evaluation of Error Probabilities For Intersymbol and Cochannel Interference", *IEEE Trans. Commun.*, vol. COM-39, pp. 1740-1749, December 1991.

[17] M. Vandroogenbroek et al., "Error Probability Computation in Optical Fiber Links", AGARD SSP Symposium Proceedings, vol. CP-574, pp. 39-1 - 39-4, September 1995.

[18] A. Zayesdny et al, *Engineering Applications of Stochastic Processes (Theory, Problems and Solutions)*, Research Studies Press, John Wiley, New York, 1989.

[19] L. C. Calvez and R. Genin, "Recursive Methods for Numerical Computation of the Coefficients of a Generalization of the Gram-Charlier Series", *Proc. of the IEEE*, vol. 64, pp. 1254-1255, August 1976.

[20] E. Biglieri, "A Recursive Method for Computing the Coefficients of the Gram-Charlier Series", *Proc. of the IEEE*, vol. 61, pp. 251-252, February 1973.

[21] J. C. Cartledge and L. W. Coathup, "A Gram-Charlier Series Method of Calculating the Probability of Bit Error in Lightwave Transmission Systems", *Journal of Lightwave Technology*,

vol. LT-4, no. 12, pp. 1736-1739, December 1986.

[22] M. Mansuripur et al., "Fiber Optics Receiver Error Rate Prediction Using the Gram-Charlier Series", *IEEE Trans. Commun.*, vol. COM-28, no. 3, pp. 402-407, March 1980.

[23] R.D. Gitlin et al, *Data Communications Principles*, Plenum Press, New York, 1992.

[24] J. G. Proakis and M. Salehi, *Communications Systems Engineering* , Prentice-Hall, New Jersey, 1994.

[25] Mitra, Sanjit K., *Digital Signal Processing : A Computer-Based Approach*, McGraw-Hill, New York, 1996.

[26] R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing*, Prentice Hall, New Jersey, 1983.

[27] M. Bellanger, *Digital Processing of Signals : Theory and Practice* , Wiley Publications, New York, 1988.

[28] B. Liu, "Effect of Finite Word Length on the Accuracy of Digital Filters - A Review", *IEEE Trans. Circ. Theory* , CT-18, no. 6, pp. 670-677, November 1971.

[29] V. K. Prahbu, "Some Considerations of Error Bounds in Digital Systems", *Bell Sys. Tech. Jour.*, vol. 50, pp. 3127-3151, December 1971.

[30] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions*, Dover Publications, New York, 1972.

[31] A. Papoulis, *Probability, Random Variables and Stochastic Processes*, Third Edition, McGraw-Hill, New York, 1991.

[32] *MATLAB Reference Guide*, Version 4.2c, The MathWorks inc., Natick, MA., 1994.

[33] C. S. Burrus, "Multiband Least Squares FIR Filter Design", *IEEE Trans. on Signal Proc.*, vol. 43, no. 2, pp. 412-420, February 1995.

[34] E. Kreyszig, *Advanced Engineering Mathematics*, Sixth Edition, John Wiley, New York,

1988.

[35] E. Y. Ho and Y. S. Yeh, “A New Approach for Evaluating the Error Probability in the Presence of Intersymbol Interference and Additive Gaussian Noise”, *Bell Sys. Tech. Jour.*, vol. 49, pp. 2249-2265, November 1970.

[36] *MAPLE_V Reference Guide*, Release 5, Waterloo Maple inc., Waterloo, 1997.

Appendix

Details of Derivations

This appendix provides the detailed derivations for Chapters Three and Four. For convenience, the section is numbered as Ax.y.z where x is the chapter number, y is the major chapter section and z is the particular section being dealt with. It includes the following sections:

A3 : Derivations from Chapter Three

This part contains the derivations from Chapter Three on the topics:

A3.1 : Details of the derivations from various sections of the AWGN Analysis

A3.2 : Derivation of the Moments of the Interference

A3.3 : Details on the Truncation Errors From the GQR Technique

A3.4 : Details on the Truncation Errors of the Beaulieu FSE Technique

A3.5 : Details on the Computational Complexity of the M-MCDD

A4 : Derivations From Chapter Four

This part contains the derivations from Chapter Four on the topics:

A4.1 : Details of Quantization Noise Analysis from various sections for the Probability of Bit Error

A4.2 : Proof of the Convergence of the Gram-Charlier Series

A4.3 : Expansion of the Upper Bound on $Q(\cdot)$

A4.4 : Derivation of the Moments of the Quantization Noise

A4.5 : Derivation of the Correlation Between the Quantization and Interference

A4.6 : Derivation of the Maxima and Minima of the Quantization Noise

A4.7 : Derivation of the Truncation Error For the Finite Taylor Series of $Q(\cdot)$

A3 : Derivations From Chapter Three

A3.1 : Details of the AWGN Analysis

A3.1.1 : Noiseless k^{th} output (Derivation of (3-6))

Figure A-1 presents a more general yet simplified M-MCDD model from Figure 2-4 having l stages which was used to derive the noiseless k^{th} output channel and the final output signal.

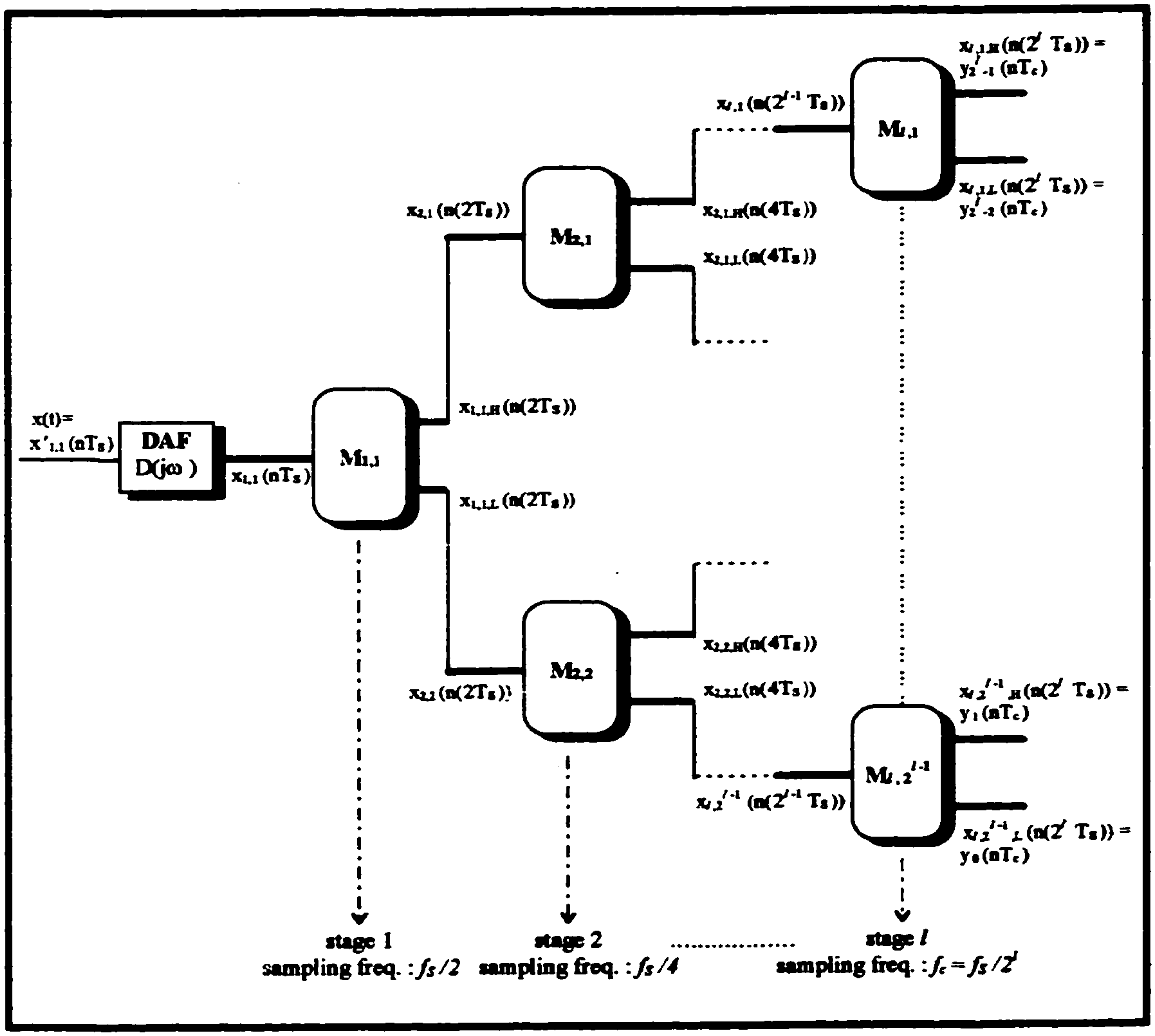


Figure A-1. The general l-stage M-MCDD used in the analysis of the final output [1] [6] [13].

From the M-MCDD structure presented in Figure A-1, we can equate the output from stage i to the corresponding input to stage $i+1$. Using the notation presented in Figure A-1, we can see that:

$$x_{i,j,H}(n\cdot 2^i T_s) = x_{i+1,2j-1}(n\cdot 2^i T_s) \tag{A3-1}$$

$$x_{i,j,L}(n\cdot 2^i T_s) = x_{i+1,2j}(n\cdot 2^i T_s) \tag{A3-2}$$

, where $1 \le j \le 2^{i-1}$. Thus, if we rearrange the indices from (A3-1) and (A3-2) to generalize the output from stage i and noting that the desired output channel is modulated unto the carrier $f_k = (k+\frac{1}{2})\cdot f_c$, we obtain for the highpass and lowpass output from stage i, respectively (consequently the same as the input to stage $i+1$):

$$x_{i+1,b}(m\cdot 2^i T_s) = \begin{cases} \begin{array}{l} x_{i,\lfloor \frac{b+1}{2} \rfloor,H}(m\cdot 2^i T_s) \\ = (-1)^m \cdot \displaystyle\sum_{n=-L}^{L} \left\{ \begin{array}{l} x_{i,\lfloor \frac{b+1}{2} \rfloor}\big((m\cdot 2^i - n\cdot 2^{i-1})\cdot T_s\big)\cdot \\ (-1)^{n\cdot 2^{i-1}T_s}\cdot h(n\cdot 2^{i-1}T_s)\cdot e^{j\pi(k+\frac{1}{2})n} \end{array} \right\} \end{array} & ; \quad \textit{highpass output from stage } i \\ \\ \begin{array}{l} x_{i,\lfloor \frac{b+1}{2} \rfloor,L}(m\cdot 2^i T_s) \\ = (-1)^m \cdot \displaystyle\sum_{n=-L}^{L} \left\{ \begin{array}{l} x_{i,\lfloor \frac{b+1}{2} \rfloor}\big((m\cdot 2^i - n\cdot 2^{i-1})\cdot T_s\big)\cdot \\ h(n\cdot 2^{i-1}T_s)\cdot e^{j\pi(k+\frac{1}{2})n} \end{array} \right\} \end{array} & ; \quad \textit{lowpass output from stage } i \end{cases} \tag{A3-3}$$

, where $1 \le b \le 2^i$. We will consider the effect of the DAF later so we will derive the desired output in terms of $x_{1,1}(\cdot)$. Once the effect of the DAF is substituted, we will obtain the form of the desired output in terms of $x(\cdot)$. Thus, if we proceed through the first stage by substituting $i=1$ into (A3-3), bearing in mind that we can have either a highpass or lowpass output, respectively, we obtain :

$$i = 1 \Rightarrow \begin{cases} \begin{array}{l} x_{2,1}(m\cdot 2T_s) = x_{1,1,H}(m\cdot 2T_s) \\ = (-1)^m \cdot \displaystyle\sum_{n_1=-L}^{L} \left\{ \begin{array}{l} (-1)^{n_1\cdot T_s}\cdot h(n_1\cdot T_s)\cdot \\ x_{1,1}\big((2m-n_1)\cdot T_s\big)\cdot e^{j\pi(k+\frac{1}{2})n_1} \end{array} \right\} \end{array} & ; \quad \textit{highpass output from stage } 1 \\ \\ \begin{array}{l} x_{2,2}(m\cdot 2T_s) = x_{1,1,L}(m\cdot 2T_s) \\ = (-1)^m \cdot \displaystyle\sum_{n_1=-L}^{L} \left\{ \begin{array}{l} h(n_1\cdot T_s)\cdot \\ x_{1,1}\big((2m-n_1)\cdot T_s\big)\cdot e^{j\pi(k+\frac{1}{2})n_1} \end{array} \right\} \end{array} & ; \quad \textit{lowpass output from stage } 1 \end{cases} \tag{A3-4}$$

If we now similarly proceed through the second stage, we will look at two cases. In the first, we will proceed from the highpass output from stage one, through $\mathbf{M}_{2,1}$, while in the second case, we will proceed from the lowpass output of stage one, through $\mathbf{M}_{2,2}$. Thus in each case, we respectively get :

$$i=2\ \ (M_{2,1}) \Rightarrow \begin{cases} x_{3,1}(m\cdot 4T_s) = x_{2,1,H}(m\cdot 4T_s) & \\ = (-1)^m \cdot \sum\limits_{n_2=-L}^{L} \left\{ \begin{matrix} (-1)^{n_2\cdot 2T_s}\cdot h(n_2\cdot 2T_s)\cdot \\ x_{2,1}\left((4m-2n_2)\cdot T_s\right)\cdot e^{j\pi(k+\frac{1}{2})\cdot n_2} \end{matrix} \right\} & ; \begin{matrix} \textit{highpass output} \\ \textit{from stage } 2 \end{matrix} \\ & \\ x_{3,2}(m\cdot 4T_s) = x_{2,1,L}(m\cdot 4T_s) & \\ = (-1)^m \cdot \sum\limits_{n_2=-L}^{L} \left\{ \begin{matrix} h(n_2\cdot 2T_s)\cdot \\ x_{2,1}\left((4m-2n_2)\cdot T_s\right)\cdot e^{j\pi(k+\frac{1}{2})\cdot n_2} \end{matrix} \right\} & ; \begin{matrix} \textit{lowpass output} \\ \textit{from stage } 2 \end{matrix} \end{cases} \tag{A3-5}$$

$$\Rightarrow \quad (-1)^m \cdot \sum_{n_2=-L}^{L} h(n_2\cdot 2T_s)\cdot x_{2,1}\left((4m-2n_2)\cdot T_s\right)\cdot e^{j\pi(k+\frac{1}{2})n_2}$$

$$= (-1)^m \cdot \sum_{n_1=-L}^{L}\sum_{n_2=-L}^{L} \left\{ \begin{matrix} (-1)^{n_1\cdot T_s}\cdot h(n_1\cdot T_s)\cdot h(n_2\cdot 2T_s)\cdot \\ x_{1,1}\left((4m-n_1-2n_2)\cdot T_s\right)\cdot e^{j\pi(k+\frac{1}{2})\cdot(n_1+n_2)} \end{matrix} \right\}$$

$$i=2\ \ (M_{2,2}) \Rightarrow \begin{cases} x_{3,3}(m\cdot 4T_s) = x_{2,2,H}(m\cdot 4T_s) & \\ = (-1)^m \cdot \sum\limits_{n_2=-L}^{L} \left\{ \begin{matrix} (-1)^{n_2\cdot 2T_s}\cdot h(n_2\cdot 2T_s)\cdot \\ x_{2,2}\left((4m-2n_2)\cdot T_s\right)\cdot e^{j\pi(k+\frac{1}{2})\cdot n_2} \end{matrix} \right\} & ; \begin{matrix} \textit{highpass output} \\ \textit{from stage } 2 \end{matrix} \\ & \\ x_{3,4}(m\cdot 4T_s) = x_{2,2,L}(m\cdot 4T_s) & \\ = (-1)^m \cdot \sum\limits_{n_2=-L}^{L} \left\{ \begin{matrix} h(n_2\cdot 2T_s)\cdot \\ x_{2,2}\left((4m-2n_2)\cdot T_s\right)\cdot e^{j\pi(k+\frac{1}{2})\cdot n_2} \end{matrix} \right\} & ; \begin{matrix} \textit{lowpass output} \\ \textit{from stage } 2 \end{matrix} \end{cases} \tag{A3-6}$$

$$\Rightarrow \quad (-1)^m \cdot \sum_{n_2=-L}^{L} h(n_2\cdot 2T_s)\cdot x_{2,2}\left((4m-2n_2)\cdot T_s\right)\cdot e^{j\pi(k+\frac{1}{2})n_2}$$

$$= (-1)^m \cdot \sum_{n_1=-L}^{L}\sum_{n_2=-L}^{L} \left\{ \begin{matrix} h(n_1\cdot T_s)\cdot h(n_2\cdot 2T_s)\cdot \\ x_{1,1}\left((4m-n_1-2n_2)\cdot T_s\right)\cdot e^{j\pi(k+\frac{1}{2})\cdot(n_1+n_2)} \end{matrix} \right\}$$

This process holds for every stage thereafter until $i=l$. It is possible to reverse the order of decimation and filtering without loss of generality due to the linearity of the system however, in this analysis, decimation is assumed to precede filtering through each stage. Therefore, from (A3-5) and (A3-6), we see that the $(-1)^{nT}$ factor present in the highpass output from every stage disappears, except from the final stage creating two distinguishable cases in the end. Thus, to simplify the remainder of the analysis, only the lowpass outputs from each remaining stage will be looked at. Thus, proceeding recursively through stage *l-1*, we get for each stage:

$$i=3\Rightarrow\begin{cases} x_{4,4}(m\cdot 8T_s)=x_{3,2,L}(m\cdot 8T_s) & \\ =(-1)^m\cdot\sum_{n_3=-L}^{L}\begin{Bmatrix} h(n_3\cdot 4T_s)\cdot \\ x_{3,2}\left((8m-4n_3)\cdot T_s\right)\cdot e^{j\pi(k+\frac{1}{2})n_3}\end{Bmatrix} & ;\quad \textit{highpass output from stage } 3 \\ =(-1)^m\cdot\sum_{n_1}\sum_{n_2}\sum_{n_3}\begin{Bmatrix}(-1)^{n_1\cdot T_s}\cdot h(n_1\cdot T_s)\cdot h(n_2\cdot 2T_s)\cdot h(n_3\cdot 4T_s)\cdot \\ x_{1,1}\left((8m-n_1-2n_2-4n_3)\cdot T_s\right)\cdot e^{j\pi(k+\frac{1}{2})(n_1+n_2+n_3)}\end{Bmatrix} & \\ & \\ x_{4,8}(m\cdot 8T_s)=x_{3,4,L}(m\cdot 8T_s) & \\ =(-1)^m\cdot\sum_{n_3=-L}^{L}\begin{Bmatrix} h(n_3\cdot 4T_s)\cdot \\ x_{3,4}\left((8m-4n_3)\cdot T_s\right)\cdot e^{j\pi(k+\frac{1}{2})n_3}\end{Bmatrix} & ;\quad \textit{lowpass output from stage } 3 \\ =(-1)^m\cdot\sum_{n_1}\sum_{n_2}\sum_{n_3}\begin{Bmatrix} h(n_1\cdot T_s)\cdot h(n_2\cdot 2T_s)\cdot h(n_3\cdot 4T_s)\cdot \\ x_{1,1}\left((8m-n_1-2n_2-4n_3)\cdot T_s\right)\cdot e^{j\pi(k+\frac{1}{2})(n_1+n_2+n_3)}\end{Bmatrix} & \end{cases}\qquad \textbf{(A3-7)}$$

$$\vdots$$

$$\vdots$$

$$\vdots$$

$$\vdots$$

$$i=l-1\Rightarrow\begin{cases} x_{l,2^{l-2}}(m\cdot 2^{l-1}T_s)=x_{l-1,2^{l-3},L}(m\cdot 2^{l-1}T_s) & \\ =(-1)^m\cdot\sum_{n_{l-1}=-L}^{L}\begin{Bmatrix} h(n_{l-1}\cdot 2^{l-2}T_s)\cdot \\ x_{l-1,2^{l-3}}\left((2^{l-1}m-2^{l-2}n_{l-1})\cdot T_s\right) \\ \cdot e^{j\pi(k+\frac{1}{2})n_{l-1}}\end{Bmatrix} & ;\quad \textit{highpass output from stage } l-1 \\ =(-1)^m\cdot\sum_{n_1}\sum_{n_2}\cdots\sum_{n_{l-1}}\begin{Bmatrix}(-1)^{n_1\cdot T_s}\cdot h(n_1\cdot T_s)\cdot h(n_2\cdot 2T_s)\cdots h(n_{l-1}\cdot 2^{l-2}T_s)\cdot \\ x_{1,1}\left((2^{l-1}m-n_1-2n_2-\cdots-2^{l-2}n_{l-1})\cdot T_s\right) \\ \cdot e^{j\pi(k+\frac{1}{2})(n_1+\cdots+n_{l-1})}\end{Bmatrix} & \\ & \\ x_{l,2^{l-1}}(m\cdot 2^{l-1}T_s)=x_{l-1,2^{l-2},L}(m\cdot 2^{l-1}T_s) & \\ =(-1)^m\cdot\sum_{n_{l-1}=-L}^{L}\begin{Bmatrix} h(n_{l-1}\cdot 2^{l-2}T_s)\cdot \\ x_{l-1,2^{l-2}}\left((2^{l-1}m-2^{l-2}n_{l-1})\cdot T_s\right) \\ \cdot e^{j\pi(k+\frac{1}{2})n_{l-1}}\end{Bmatrix} & ;\quad \textit{lowpass output from stage } l-1 \\ =(-1)^m\cdot\sum_{n_1}\sum_{n_2}\cdots\sum_{n_{l-1}}\begin{Bmatrix} h(n_1\cdot T_s)\cdot h(n_2\cdot 2T_s)\cdots h(n_{l-1}\cdot 2^{l-2}T_s)\cdot \\ x_{1,1}\left((2^{l-1}m-n_1-2n_2-\cdots-2^{l-2}n_{l-1})\cdot T_s\right) \\ \cdot e^{j\pi(k+\frac{1}{2})(n_1+\cdots+n_{l-1})}\end{Bmatrix} & \end{cases}\qquad \textbf{(A3-8)}$$

Thus, after stage *l*, we obtain the desired output signal which in general can be expressed as :

$$y_k(mT_c)=\begin{cases}(-1)^m\sum_{n_1=-L}^{L}\cdots\cdots\sum_{n_l=-L}^{L}\left\{\begin{array}{l}\prod_{j=1}^{l}\left(h(n_j\cdot 2^{j-1}T_s)\right)\cdot\\ x_{1,1}((2^l m-\sum_{j=1}^{l}2^{j-1}n_j)T_s)\cdot e^{j\pi(k+\frac{1}{2})\cdot\sum_{j=1}^{l}n_j}\end{array}\right\} & ;\quad k=0\ ,\ k\ even\\[2ex] (-1)^m\sum_{n_1=-L}^{L}\cdots\cdots\sum_{n_l=-L}^{L}\left\{\begin{array}{l}(-1)^{n_1\cdot T_s}\cdot\prod_{j=1}^{l}\left(h(n_j\cdot 2^{j-1}T_s)\right)\cdot\\ x_{1,1}((2^l m-\sum_{j=1}^{l}2^{j-1}n_j)T_s)\cdot e^{j\pi(k+\frac{1}{2})\cdot\sum_{j=1}^{l}n_j}\end{array}\right\} & ;\quad k\ odd\end{cases}\tag{A3-9}$$

, where $T_c = 2^l \cdot T_s$. Now, we must take into account the effect of the DAF to be able to arrive at an expression in terms of the MF-TDMA sampled input signal, *x(nT$_s$)*. If we assume that *x(t)* is bandlimited to *NB* from the start then we can say :

$$x'_{1,1}((2^l m-\sum_{j=1}^{l}2^{j-1}n_j)T_s\}=x((2^l m-\sum_{j=1}^{l}2^{j-1}n_j)T_s)\tag{A3-10}$$

, referring to the quantities in Figure A-1. This signal passes through the DAF whose impulse response has been assumed to be approximated by a prototype FIR halfband filter (see Chapter two for more detail), thus [13]:

$$\begin{aligned}x_{1,1}(nT_s)&=\{x'_{1,1}*h\}(nT_s)\\ &=\sum_{p=-L}^{L}h(pT_s)\cdot x'_{1,1}(n\cdot T_s-p\cdot T_s)=\sum_{p=-L}^{L}h(pT_s)\cdot x(n\cdot T_s-p\cdot T_s)\end{aligned}\tag{A3-11}$$

If we substitute (A3-11) into (A3-9) and rearrange the original expression so that all filtering operations are accounted for, we get :

$$y_k(mT_c)=\begin{cases}(-1)^m\cdot\left\{\begin{array}{l}\sum_{n_0=-L}^{L}h(n_0T_s)\sum_{n_1=-L}^{L}h(n_1 2T_s)\sum_{n_2=-L}^{L}\cdots\\ \cdots\sum_{n_l=-L}^{L}h(n_l 2^l T_s)\cdot x((2^l m-\sum_{j=0}^{l}2^j n_j)T_s)\cdot e^{j\pi(k+\frac{1}{2})\cdot\sum_{j=0}^{l}n_j}\end{array}\right\} & ;\quad k=0\ ,\ k\ even\\[2ex] (-1)^m\cdot\left\{\begin{array}{l}\sum_{n_0=-L}^{L}(-1)^{n_0T_s}\cdot h(n_0T_s)\sum_{n_1=-L}^{L}h(n_1 2T_s)\sum_{n_2=-L}^{L}\cdots\\ \cdots\sum_{n_l=-L}^{L}h(n_l 2^l T_s)\cdot x((2^l m-\sum_{j=0}^{l}2^j n_j)T_s)\cdot e^{j\pi(k+\frac{1}{2})\cdot\sum_{j=0}^{l}n_j}\end{array}\right\} & ;\quad k\ odd\end{cases}\tag{A3-12}$$

Once (A3-12) is simplified, (3-6) is obtained.

A3.1.2 : Final Output and Interference Terms (Derivation of (3-7))

First, we manipulate (3-1) and (3-2) in order to obtain the same index as is indicated by *x(·)* in (3-6). With the proper index, (3-2) becomes :

$$
\begin{aligned}
s_k\left((2^l m-\sum_{j=0}^{l}2^j n_j)\cdot T_s\right) &= \sum_{i=-\infty}^{\infty}\left\{\begin{array}{l} A_k\cdot e^{j[2\pi f_k\left((2^l m-\sum_{j=0}^{l}2^j n_j)\cdot T_s\right)+\alpha_{k,j}+\phi_k]}\cdot \\ h_s(\left((2^l m-\sum_{j=0}^{l}2^j n_j)\cdot T_s\right)-iT_b-\gamma_k) \end{array}\right\} \\
&= \sum_{i=-\infty}^{\infty}\left\{\begin{array}{l} A_k\cdot e^{j[2\pi(k+\frac{1}{2})\cdot f_c\left(m\cdot 2^l T_s-\sum_{j=0}^{l}n_j\cdot 2^j T_s\right)+\alpha_{k,j}+\phi_k]}\cdot \\ h_s(\left((2^l m-\sum_{j=0}^{l}2^j n_j)\cdot T_s\right)-iT_b-\gamma_k) \end{array}\right\} \\
&= \sum_{i=-\infty}^{\infty}\left\{\begin{array}{l} A_k\cdot e^{j[2\pi(k+\frac{1}{2})\cdot f_c\left(m\cdot T_c-\sum_{j=0}^{l}n_j\cdot T_c\right)+\alpha_{k,j}+\phi_k]}\cdot \\ h_s(\left((2^l m-\sum_{j=0}^{l}2^j n_j)\cdot T_s\right)-iT_b-\gamma_k) \end{array}\right\} \\
&= (-1)^m\cdot\sum_{i=-\infty}^{\infty}\left\{\begin{array}{l} A_k\cdot e^{-j\pi(k+\frac{1}{2})\cdot\sum_{j=0}^{l}n_j}\cdot e^{j(\alpha_{k,j}+\phi_k)}\cdot \\ h_s(\left((2^l m-\sum_{j=0}^{l}2^j n_j)\cdot T_s\right)-iT_b-\gamma_k) \end{array}\right\}
\end{aligned}
\qquad \text{(A3-13)}
$$

If we use the same index in (3-1) and then substitute (A3-13), we obtain:

$$
\begin{aligned}
x\left((2^l m-\sum_{j=0}^{l}2^j n_j)\cdot T_s\right) &= \sum_{k=0}^{N-1}s_k\left((2^l m-\sum_{j=0}^{l}2^j n_j)\cdot T_s\right)+z\left((2^l m-\sum_{j=0}^{l}2^j n_j)\cdot T_s\right) \\
&= \sum_{k=0}^{N-1}(-1)^m\cdot\sum_{i=-\infty}^{\infty}\left\{\begin{array}{l} A_k\cdot e^{-j\pi(k+\frac{1}{2})\cdot\sum_{j=0}^{l}n_j}\cdot e^{j(\alpha_{k,j}+\phi_k)} \\ \cdot h_s(\left((2^l m-\sum_{j=0}^{l}2^j n_j)\cdot T_s\right)-iT_b-\gamma_k) \\ +z\left((2^l m-\sum_{j=0}^{l}2^j n_j)\cdot T_s\right) \end{array}\right\} \qquad \text{(A3-14)} \\
&= (-1)^m\cdot\sum_{k=0}^{N-1}\left\{\begin{array}{l} A_k\cdot e^{j\phi_k}\cdot\sum_{i=-\infty}^{\infty}e^{j\alpha_{k,j}}\cdot e^{-j\pi(k+\frac{1}{2})\cdot\sum_{j=0}^{l}n_j}\cdot h_s(\left((2^l m-\sum_{j=0}^{l}2^j n_j)\cdot T_s\right)-iT_b-\gamma_k) \\ +z\left((2^l m-\sum_{j=0}^{l}2^j n_j)\cdot T_s\right) \end{array}\right\}
\end{aligned}
$$

Next, we substitute (A3-14) into (3-6). To simplicity the analysis, we will consider only the lowpass outputs from the final stage since the steps to follow for the corresponding highpass outputs are exactly the same. The lowpass outputs from the final stage are represented by the first form of (3-6), for *k=0* and *k even*. Therefore, from that part of the analysis, we get:

$$\begin{aligned}
y_k(mT_c) &= (-1)^m \sum_{n_0=-L}^{L} \cdots\cdots \sum_{n_l=-L}^{L} \prod_{j=0}^{l} \left(h(n_j \cdot 2^j T_s)\right) e^{j\pi(k+\frac{1}{2})\cdot\sum_{j=0}^{l} n_j} \\
&\quad \cdot \left((-1)^m \cdot \left\{ \begin{array}{l} \displaystyle\sum_{k=0}^{N-1} A_k \cdot e^{j\phi_k} \cdot \sum_{i=-\infty}^{\infty} \left\{ \begin{array}{l} e^{j\alpha_{k,i}} \cdot e^{-j\pi(k+\frac{1}{2})\cdot\sum_{j=0}^{l} n_j} \cdot \\ h_s\left(\left((2^l m - \displaystyle\sum_{j=0}^{l} 2^j n_j)\cdot T_s\right) - iT_b - \gamma_k\right) \end{array} \right] \\ +z\left((2^l m - \displaystyle\sum_{j=0}^{l} 2^j n_j)\cdot T_s\right) \end{array} \right] \right) \\
&= \sum_{n_0=-L}^{L} \cdots\cdots \sum_{n_l=-L}^{L} \left\{ \begin{array}{l} \displaystyle\prod_{j=0}^{l} \left(h(n_j \cdot 2^j T_s)\right) \cdot \\ \displaystyle\sum_{k=0}^{N-1} \left\{ \begin{array}{l} A_k \cdot e^{j\phi_k} \\ \cdot \displaystyle\sum_{i=-\infty}^{\infty} \left\{ \begin{array}{l} e^{j\alpha_{k,i}} \cdot h_s\left(\left((2^l m - \displaystyle\sum_{j=0}^{l} 2^j n_j)\cdot T_s\right) - iT_b - \gamma_k\right) \cdot \\ e^{j\pi(k-q)\cdot\sum_{j=0}^{l} n_j} \end{array} \right] \end{array} \right] \end{array} \right\} \qquad \text{(A3-15)} \\
&\quad + (-1)^m \cdot \sum_{n_0=-L}^{L} \cdots\cdots \sum_{n_l=-L}^{L} \left\{ \begin{array}{l} \displaystyle\prod_{j=0}^{l} \left(h(n_j \cdot 2^j T_s)\right) \cdot \\ z\left((2^l m - \displaystyle\sum_{j=0}^{l} 2^j n_j)\cdot T_s\right) \cdot e^{j\pi(k+\frac{1}{2})\cdot\sum_{j=0}^{l} n_j} \end{array} \right\}
\end{aligned}$$

From (A3-15), if we isolate the desired term for the k^{th} channel, we obtain for $y_k(mT_c)$:

$$
\begin{aligned}
y_k(mT_c) = {} & A_k \cdot e^{j\phi_k} \cdot \sum_i e^{ja_{k,i}} \sum_{n_0} \cdots\cdots \sum_{n_l} \left\{ \begin{array}{l} \prod_{j=0}^{l} \left(h(n_j \cdot 2^j T_s) \right) \\ \cdot h_s[(2^l m - \sum_j 2^j n_j) \cdot T_s - iT_b - \gamma_k] \end{array} \right\} \\
& + \sum_{q \neq k} A_q \cdot e^{j\phi_q} \cdot \sum_i e^{ja_{q,i}} \sum_{n_0} \cdots\cdots \sum_{n_l} \left\{ \begin{array}{l} \prod_{j=0}^{l} \left(h(n_j \cdot 2^j T_s) \right) \\ \cdot h_s[(2^l m - \sum_j 2^j n_j) \cdot T_s - iT_b - \gamma_q] \\ \cdot e^{j\pi(k-q) \cdot \sum_j n_j} \end{array} \right\} \\
& + (-1)^m \cdot \sum_{n_0} \cdots\cdots \sum_{n_l} \left\{ \begin{array}{l} \prod_{j=0}^{l} \left(h(n_j \cdot 2^j T_s) \right) \\ \cdot z[(2^l m - \sum_j 2^j n_j) \cdot T_s] \cdot e^{j\pi(k+\frac{1}{2}) \cdot \sum_j n_j} \end{array} \right\}
\end{aligned}
\qquad \text{(A3-16)}
$$

From (A3-16), if we substitute the appropriate indices according to those defined in (3-3), we obtain:

$$
\begin{aligned}
y_k[(\beta_l - \xi)T_c] = {} & A_k \cdot e^{j\phi_k} \cdot e^{ja_{k,\alpha}} \sum_{n_0} \cdots\cdots \sum_{n_l} \left\{ \begin{array}{l} \prod_{j=0}^{l} \left(h(n_j \cdot 2^j T_s) \right) \\ \cdot h_s[(2^l (\beta_l - \xi) - \sum_j 2^j n_j) \cdot T_s - \alpha T_b - \gamma_k] \end{array} \right\} \\
& + A_k \cdot e^{j\phi_k} \cdot \sum_{i \neq \alpha} e^{ja_{k,i}} \sum_{n_0} \cdots\cdots \sum_{n_l} \left\{ \begin{array}{l} \prod_{j=0}^{l} \left(h(n_j \cdot 2^j T_s) \right) \\ \cdot h_s[(2^l (\beta_l - \xi) - \sum_j 2^j n_j) \cdot T_s - iT_b - \gamma_k] \end{array} \right\} \\
& + \sum_{q \neq k} A_q \cdot e^{j\phi_q} \cdot \sum_i e^{ja_{q,i}} \sum_{n_0} \cdots\cdots \sum_{n_l} \left\{ \begin{array}{l} \prod_{j=0}^{l} \left(h(n_j \cdot 2^j T_s) \right) \\ \cdot h_s[(2^l (\beta_l - \xi) - \sum_j 2^j n_j) \cdot T_s - iT_b - \gamma_q] \\ \cdot e^{j\pi(k-q) \cdot \sum_j n_j} \end{array} \right\} \\
& + (-1)^{(\beta_l - \xi)} \cdot \sum_{n_0} \cdots\cdots \sum_{n_l} \left\{ \begin{array}{l} \prod_{j=0}^{l} \left(h(n_j \cdot 2^j T_s) \right) \cdot \\ z[(2^l (\beta_l - \xi) - \sum_j 2^j n_j) \cdot T_s] \cdot e^{j\pi(k+\frac{1}{2}) \cdot \sum_j n_j} \end{array} \right\} .
\end{aligned}
\qquad \text{(A3-17)}
$$

, where the first term in (A3-17) represents the α^{th} desired symbol on the k^{th} desired channel, the second term represents the ISI on the k^{th} channel, the third term represents the ICI on all other interfering channels and the final term is the Gaussian noise. Substituting (A3-17) into (3-3) gives us for the output of the channel detection filter :

$$
\begin{aligned}
r_k(lT_b) = {} & A_k \cdot e^{j\phi_k} \cdot e^{j\alpha_{k,\alpha}} \sum_{\xi=I_1}^{I_2} \sum_{n_0} \cdots\cdots \sum_{n_l} \left\{ \begin{array}{l} \prod_{j=0}^{l} \left(h(n_j \cdot 2^j T_s) \right) \\ \cdot h_s[(2^l(\beta_l - \xi) - \sum_j 2^j n_j) \cdot T_s - \alpha T_b - \gamma_k] \\ \cdot g[(\xi + \mu_l) T_c] \end{array} \right\} \\
& + A_k \cdot e^{j\phi_k} \cdot \sum_{i \neq \alpha} e^{j\alpha_{k,i}} \sum_{\xi=I_1}^{I_2} \sum_{n_0} \cdots\cdots \sum_{n_l} \left\{ \begin{array}{l} \prod_{j=0}^{l} \left(h(n_j \cdot 2^j T_s) \right) \\ \cdot h_s[(2^l(\beta_l - \xi) - \sum_j 2^j n_j) \cdot T_s - iT_b - \gamma_k] \\ \cdot g[(\xi + \mu_l) T_c] \end{array} \right\} \\
& + \sum_{q \neq k} A_q \cdot e^{j\phi_q} \cdot \sum_i e^{j\alpha_{q,i}} \sum_{\xi=I_1}^{I_2} \sum_{n_0} \cdots\cdots \sum_{n_l} \left\{ \begin{array}{l} \prod_{j=0}^{l} \left(h(n_j \cdot 2^j T_s) \right) \\ \cdot h_s[(2^l(\beta_l - \xi) - \sum_j 2^j n_j) \cdot T_s - iT_b - \gamma_q] \\ \cdot g[(\xi + \mu_l) T_c] \cdot e^{j\pi(k-q) \cdot \sum_j n_j} \end{array} \right\} \\
& + \sum_{\xi=I_1}^{I_2} (-1)^{(\beta_l - \xi)} \cdot \sum_{n_0} \cdots\cdots \sum_{n_l} \left\{ \begin{array}{l} \prod_{j=0}^{l} \left(h(n_j \cdot 2^j T_s) \right) \\ \cdot z[(2^l(\beta_l - \xi) - \sum_j 2^j n_j) \cdot T_s] \\ \cdot g[(\xi + \mu_l) T_c] \cdot e^{j\pi(k+\frac{1}{2}) \cdot \sum_j n_j} \end{array} \right\}
\end{aligned}
\tag{A3-18}
$$

Once we repeat the above steps for the highpass outputs from the final stage, we will have two distinguishable cases for (A3-18), one for *k=0* and *k even* and another for *k odd*. From there, we may define (3-8) and (3-9) as the intersymbol interference (ISI) on the k^{th} channel and the interchannel interference (ICI) on all other interfering channels, respectively. Also, we may define (3-10) as the output WGN process resulting from the k^{th} channel. Once these quantities are defined and substituted into (A3-18), we obtain (3-7) exactly.

A3.1.3 : Output Noise Variance (Derivation of (3-11))

In order to derive (3-11), we will focus on the form of the lowpass outputs from the output WGN process of (3-10), corresponding to *k=0* and *k even*. In the case of the highpass outputs, because we are looking at the second order moment about the origin, the $(-1)^{nT}$ factor will disappear thus for both cases the final result will be the same . Since *z(t)* is a stationary Gaussian process, we assume that *z(t)~N(0,σ^2).* Since *z(t)* has a mean of zero, the variance is simply obtained by considering the second order moment about the origin, thus [31]:

$$
\begin{aligned}
E(z_k^2) &= E(z_k \cdot z_k^*) = \sigma_o^2 \\
&= E\left\{
\begin{array}{l}
\displaystyle\sum_{\xi_1=I_1}^{I_2}(-1)^{\beta_l-\xi_1}\sum_{n_0}\cdots\cdots\sum_{n_l}\left\{
\begin{array}{l}
z[(2^l(\beta_l-\xi_1)-\sum_j 2^j n_j)\cdot T_s] \\
\cdot\displaystyle\prod_{j=0}^{l}\left(h(n_j\cdot 2^j T_s)\right) \\
\cdot g[(\xi_1+\mu_l)T_c]\cdot e^{j\pi(k+\frac{1}{2})\cdot\sum_j n_j}
\end{array}\right\} \\
\displaystyle\sum_{\xi_2=I_1}^{I_2}(-1)^{\beta_l-\xi_2}\sum_{n_0'}\cdots\cdots\sum_{n_l'}\left\{
\begin{array}{l}
z[(2^l(\beta_l-\xi_2)-\sum_j 2^j n_j')\cdot T_s] \\
\cdot\displaystyle\prod_{j=0}^{l}\left(h(n_j'\cdot 2^j T_s)\right) \\
\cdot g[(\xi_2+\mu_l)T_c]\cdot e^{-j\pi(k+\frac{1}{2})\cdot\sum_j n_j'}
\end{array}\right\}
\end{array}\right\}
\end{aligned}
\qquad \text{(A3-19)}
$$

Next, we get :

$$
E(z_k\cdot z_k^*) = \sum_{\xi_1=I_1}^{I_2}\sum_{n_0}\cdots\sum_{n_l}\sum_{\xi_2=I_1}^{I_2}\sum_{n_0'}\cdots\sum_{n_l'}\left\{
\begin{array}{l}
(-1)^{\xi_1+\xi_2}\cdot\displaystyle\prod_{j=0}^{l}\left(h(n_j\cdot 2^j T_s)\right)\cdot\prod_{j=0}^{l}\left(h(n_j'\cdot 2^j T_s)\right) \\
\cdot g[(\xi_1+\mu_l)T_c]\cdot g[(\xi_2+\mu_l)T_c] \\
\cdot E\left\{
\begin{array}{l}
z[(2^l(\beta_l-\xi_1)-\sum_j 2^j n_j)\cdot T_s] \\
z[(2^l(\beta_l-\xi_2)-\sum_j 2^j n_j')\cdot T_s]
\end{array}\right\}
\end{array}\right\}
\qquad \text{(A3-20)}
$$

Since *z(t)* is also a white process, the correlation function for *z(t)* is [31]:

$$
R_z(\xi_1-\xi_2) = \begin{cases} \sigma^2\cdot\delta(\xi_1-\xi_2)\ ; & \xi_1=\xi_2 \\ \quad 0 \quad ; & other \end{cases}
\qquad \text{(A3-21)}
$$

Once (A3-21) is substituted into (A3-20) and substituting $\xi_1 = \xi_2 = \xi$, we get :

$$
\begin{aligned}
\sigma_o^2 = \sigma^2 &\cdot \sum_{\xi=I_1}^{I_2}\left|g[(\xi+\mu_l)T_c]\right|^2 \\
&\cdot\prod_{j=0}^{l}\left(\sum_{n_j}\left|h(n_j\cdot 2^j T_s)\right|^2 + \sum_{n_j}\sum_{n_j'\neq n_j} h(n_j\cdot 2^j T_s)\cdot h(n_j'\cdot 2^j T_s)\right)
\end{aligned}
\qquad \text{(A3-22)}
$$

From (A3-22), the cross-terms of the halfband filter coefficients fall out due to the equivalent noise bandwidth of $h(n_j \cdot 2^j T_s)$ [27]. Without these cross-terms, the form of (3-11) is seen from (A3-22).

A3.1.4 : Derivation of the Probability of Bit Error (Derivation of (3-20))

In order to show that (3-20) is true, we start from the sufficient statistic of (3-19). From (3-19), the variable η is random with respect to the carrier phase ϕ_q and timing phase γ_q . The variable $u(0)$ is random with respect to the timing phase γ_k . Since $a_{k\ \alpha}$ is also randomly chosen between $\{\pm 1\}$ with equal probability, it can be shown that :

$$P_{b|a_{k,\alpha}=+1} = P_{b|a_{k,\alpha}=-1} \tag{A3-23}$$

We will prove (A3-23) by looking at two cases : $a_{k\ \alpha} = -1$ and $a_{k\ \alpha} = +1$. In the first case, for $a_{k\ \alpha} = -1$, assuming a constant carrier phase, we have :

$$\begin{aligned}
P_{b|\eta,a_{k,\alpha}=-1} &= \int_{-T/2}^{T/2} \Pr\left\{-A_k \cdot u(0) + \eta\ + z_k^I(lT_b) \geq 0\right\} \cdot f(\gamma_k) d\gamma_k \\
&= \int_{-T/2}^{T/2} \Pr\left\{z_k^I(lT_b) \geq A_k \cdot u(0) - \eta\ \right\} \cdot f(\gamma_k) d\gamma_k \\
&= \int_{-T/2}^{T/2} \Pr\left\{N(0,\sigma_o^2) \geq A_k \cdot u(0) - \eta\ \right\} \cdot f(\gamma_k) d\gamma_k \\
&= \int_{-T/2}^{T/2} \Pr\left\{N(0,1) \geq \frac{A_k \cdot u(0) - \eta}{\sigma_o}\right\} \cdot f(\gamma_k) d\gamma_k
\end{aligned} \tag{A3-24}$$

thus :

$$P_{b|\eta,\eta_q,a_{k,\alpha}=-1} = \int_{-T/2}^{+T/2} Q\left(\frac{A_k u(0) - \eta}{\sigma_o}\right) \cdot f(\gamma_k) \cdot d\gamma_k \tag{A3-25}$$

Now, for $a_{k\ \alpha} = +1$:

$$\begin{aligned}
P_{b|\eta,a_{k,\alpha}=+1} &= \int_{-T/2}^{T/2} \Pr\left\{A_k \cdot u(0) + \eta\ + z_k^I(lT_b) \leq 0\right\} \cdot f(\gamma_k) d\gamma_k \\
&= \int_{-T/2}^{T/2} \Pr\left\{z_k^I(!T_b) \leq -A_k \cdot u(0) - \eta\ \right\} \cdot f(\gamma_k) d\gamma_k \\
&= \int_{-T/2}^{T/2} \Pr\left\{N(0,\sigma_o^2) \leq -A_k \cdot u(0) - \eta\ \right\} \cdot f(\gamma_k) d\gamma_k \\
&= \int_{-T/2}^{T/2} \Pr\left\{N(0,1) \geq \frac{A_k \cdot u(0) + \eta}{\sigma_o}\right\} \cdot f(\gamma_k) d\gamma_k
\end{aligned} \tag{A3-26}$$

Since it was assumed earlier that the data symbols $a_{k,i}$ are independent and equiprobable, η may therefore be a positive or negative quantity with a probability of $\frac{1}{2}$ thus η has an even probability density function [10-11]. It thus follows from (A3-26) that :

$$P_{b|\eta,a_{k,\alpha}=+1} = \int_{-T/2}^{+T/2} Q\left(\frac{A_k u(0) - \eta}{\sigma_o}\right) \cdot f(\gamma_k) \cdot d\gamma_k \tag{A3-27}$$

So since :

$$P_{b|\eta} = \frac{1}{2} \cdot \left(P_{b|\eta,a_{k,\alpha}=-1} + P_{b|\eta,a_{k,\alpha}=+1}\right) \tag{A3-28}$$

we have that :

$$P_{b|\eta} = \int_{-T/2}^{+T/2} Q\left(\frac{A_k u(0) - \eta}{\sigma_o}\right) \cdot f(\gamma_k) \cdot d\gamma_k \tag{A3-29}$$

where $f(\gamma_k)$ is the p.d.f. of the timing phase on the desired k^{th} channel. To obtain the unconditional probability of bit error, we must average over the probability density function of the interference. Let $f_\eta(\eta)$ be the p.d.f. of the interference η. Thus, the unconditional probability of bit error becomes :

$$P_b = \int_{-\infty \atop \eta}^{+\infty} \left\{ \int_{-T/2 \atop \gamma_k}^{+T/2} Q\left(\frac{A_k u(0) - \eta}{\sigma_o}\right) \cdot f(\gamma_k) \cdot d\gamma_k \right\} \cdot f(\eta) \cdot d\eta \tag{A3-30}$$

If in (A3-30) we assume that we have a fixed timing offset γ_k , we see straightforwardly that (A3-30) becomes the probability of bit error expression that is shown by (3-20).

A3.2 : Derivation of the Moments of the Interference (Derivation of (3-38) through (3-41))

First, we start with the truncated expression for the interference variable of (3-36) and define η_1 as the ISI and η_2 as the ICI. These two quantities, respectively, are :

$$\eta_1 = \sum_{i=\alpha-m_l \atop i\neq\alpha}^{\alpha+m_u} A_k \cdot a_{k,i} \cdot u(\alpha T_b - iT_b) \tag{A3-31}$$

$$\eta_2 = \sum_{q\neq k} A_q \cdot \sum_{i=\alpha-m_l}^{\alpha+m_u} \left\{ \begin{array}{l} a_{q,i} \cdot [v^I(\alpha T_b - iT_b) \cdot \cos\phi_q - v^Q(\alpha T_b - iT_b) \cdot \sin\phi_q] \\ -b_{q,i} \cdot [v^I(\alpha T_b - iT_b) \cdot \sin\phi_q + v^Q(\alpha T_b - iT_b) \cdot \cos\phi_q] \end{array} \right\} \tag{A3-32}$$

If we look at η' from (3-36) on a symbol by symbol basis, that is, for fixed i and fixed q^{th} channel in the case of the ICI, we can say:

$$\eta'_i = \eta_{1,i} + \eta_{2,q,i} = \eta'\big|_{i\ fixed} \tag{A3-33}$$

From (A3-31), we can see easily how (3-38) was derived. Now, we will show why the odd order moments are zero. Of course, one should expect this result since the data symbols $a_{k,i}$ are equiprobable and chosen from the set *{-1,+1}* [10]. Thus, if the order of ρ from (3-38) is odd, we have:

$$\begin{aligned} E[\eta_{1,i}^{\rho}] &= \tfrac{1}{2}\cdot A_k^{\rho}\sum_{a_{k,i}}\int_{-T/2}^{T/2} a_{k,i}{}^{\rho}\cdot u^{\rho}(\alpha T_b - iT_b)\cdot f(\gamma_k)\cdot d\gamma_k \\ &= \tfrac{1}{2}\cdot A_k^{\rho}\cdot\int_{-T/2}^{T/2}\big(u^{\rho}(\alpha T_b - iT_b) - u^{\rho}(\alpha T_b - iT_b)\big)\cdot f(\gamma_k)\cdot d\gamma_k \\ &= 0 \end{aligned} \tag{A3-34}$$

As for the even order moments of the ISI, as shown in (3-40), this can be easily derived. Consider the order of the moment to be $\rho' = 2\rho$, thus :

$$\begin{aligned} E[\eta_{1,i}^{2\rho}] &= \tfrac{1}{2}\cdot A_k^{2\rho}\cdot\int_{-T/2}^{T/2} 2\cdot u^{2\rho}(\alpha T_b - iT_b)\cdot f(\gamma_k)\cdot d\gamma_k \\ &= A_k^{2\rho}\cdot\int_{-T/2}^{T/2}\big(u(\alpha T_b - iT_b)\big)^{2\rho}\cdot f(\gamma_k)\cdot d\gamma_k \end{aligned} \tag{A3-35}$$

We see that this is exactly the form of (3-40). As for the moments of the ICI, we also see quite easily how (3-39) could be derived by simply considering all the random quantities from (A3-32). We will show why the odd order moments are zero. First, consider an arbitrary term in the polynomial presented in (3-39), that is to say [34]:

$$\begin{aligned} &\sum_{a_{q,i},b_{q,i}}\left\{\begin{matrix} a_{q,i}\cdot[v^{I}(\alpha T_b - iT_b)\cdot\cos\phi_q - v^{Q}(\alpha T_b - iT_b)\cdot\sin\phi_q] \\ -b_{q,i}\cdot[v^{I}(\alpha T_b - iT_b)\cdot\sin\phi_q + v^{Q}(\alpha T_b - iT_b)\cdot\cos\phi_q] \end{matrix}\right\}^{\rho} \\ &\rightarrow(-1)^{\beta}\cdot\binom{\rho}{\beta}\cdot\sum_{a_{q,i},b_{q,i}}\begin{matrix} a_{q,i}^{\rho-\beta}\cdot[v^{I}(\alpha T_b - iT_b)\cdot\cos\phi_q - v^{Q}(\alpha T_b - iT_b)\cdot\sin\phi_q]^{\rho-\beta} \\ \cdot b_{q,i}^{\beta}\cdot[v^{I}(\alpha T_b - iT_b)\cdot\sin\phi_q + v^{Q}(\alpha T_b - iT_b)\cdot\cos\phi_q]^{\beta} \end{matrix} \end{aligned} \tag{A3-36}$$

, where $0 \le \beta \le \rho$. If we consider β odd then, we obtain for the individual terms :

$$\left\{\begin{matrix} -a_{q,i}^{\rho-\beta}\cdot\begin{bmatrix} v^{I}(\alpha T_b - iT_b)\cdot\cos\phi_q \\ -v^{Q}(\alpha T_b - iT_b)\cdot\sin\phi_q \end{bmatrix}^{\rho-\beta}\cdot\begin{bmatrix} v^{I}(\alpha T_b - iT_b)\cdot\sin\phi_q \\ +v^{Q}(\alpha T_b - iT_b)\cdot\cos\phi_q \end{bmatrix}^{\beta} \\ +a_{q,i}^{\rho-\beta}\cdot\begin{bmatrix} v^{I}(\alpha T_b - iT_b)\cdot\cos\phi_q \\ -v^{Q}(\alpha T_b - iT_b)\cdot\sin\phi_q \end{bmatrix}^{\rho-\beta}\cdot\begin{bmatrix} v^{I}(\alpha T_b - iT_b)\cdot\sin\phi_q \\ +v^{Q}(\alpha T_b - iT_b)\cdot\cos\phi_q \end{bmatrix}^{\beta} \end{matrix}\right\} = 0 \tag{A3-37}$$

and if we consider β even, we have $\rho-\beta$ odd thus :

$$\left\{\begin{array}{l} \begin{bmatrix} v^I(\alpha T_b - iT_b)\cdot\cos\phi_q \\ -v^Q(\alpha T_b - iT_b)\cdot\sin\phi_q \end{bmatrix}^{\rho-\beta} \cdot b_{q,i}^{\beta} \cdot \begin{bmatrix} v^I(\alpha T_b - iT_b)\cdot\sin\phi_q \\ +v^Q(\alpha T_b - iT_b)\cdot\cos\phi_q \end{bmatrix}^{\beta} \\ -\begin{bmatrix} v^I(\alpha T_b - iT_b)\cdot\cos\phi_q \\ -v^Q(\alpha T_b - iT_b)\cdot\sin\phi_q \end{bmatrix}^{\rho-\beta} \cdot b_{q,i}^{\beta} \cdot \begin{bmatrix} v^I(\alpha T_b - iT_b)\cdot\sin\phi_q \\ +v^Q(\alpha T_b - iT_b)\cdot\cos\phi_q \end{bmatrix}^{\beta} \end{array}\right\} = 0 \qquad \text{(A3-38)}$$

thus, the odd order moments are zero. As for the even order moments, we will show how (3-41) was derived. First, for the case where β is odd, we have $\rho-\beta$ odd also thus we may refer to (A3-38) to see why the result will be zero. As for the case where β is even, we have $\rho-\beta$ even also thus we have :

$$4\cdot\binom{\rho}{\beta}\cdot\begin{pmatrix} v^I(\alpha T_b - iT_b)\cdot\cos\phi_q \\ -v^Q(\alpha T_b - iT_b)\cdot\sin\phi_q \end{pmatrix}^{\rho-\beta}\cdot\begin{pmatrix} v^I(\alpha T_b - iT_b)\cdot\sin\phi_q \\ +v^Q(\alpha T_b - iT_b)\cdot\cos\phi_q \end{pmatrix}^{\beta} \qquad \text{(A3-39)}$$

Once, we consider all values of β that is for $0\le\beta\le\rho$, we obtain the polynomial expression :

$$\begin{aligned} &4\cdot\begin{pmatrix} v^I(\alpha T_b - iT_b)\cdot\cos\phi_q - v^Q(\alpha T_b - iT_b)\cdot\sin\phi_q \\ +v^I(\alpha T_b - iT_b)\cdot\sin\phi_q + v^Q(\alpha T_b - iT_b)\cdot\cos\phi_q \end{pmatrix}^{\rho} \\ &= 4\cdot\begin{pmatrix} v^I(\alpha T_b - iT_b)\cdot\left(\cos\phi_q + \sin\phi_q\right) \\ +v^Q(\alpha T_b - iT_b)\cdot\left(\cos\phi_q - \sin\phi_q\right) \end{pmatrix}^{\rho} \end{aligned} \qquad \text{(A3-40)}$$

Thus, if we substitute (A3-40) for the polynomial expression in (3-39) and consider the order of the moment to be $\rho'=2\rho$, we obtain (3-41) exactly.

A3.3 : Details on the Truncation Errors From the GQR Technique (Derivation of (3-42), (3-44), (3-45), (3-49))

First, we will derive the truncation error due to using only the first $2M+1$ moments of the interference. To proceed, we must make use of the following theorem to arrive at an upper bound for the truncation error [10,thm 1]:

$$\begin{aligned} P_b &= \int_{-\infty}^{+\infty}\left\{\int_{-T/2}^{T/2} Q\left(\frac{A_k u(0)-\eta}{\sigma_o}\right)\cdot f(\gamma_k)\cdot d\gamma_k\right\}\cdot f_\eta(\eta)\cdot d\eta \\ &\approx \sum_{i=1}^{M} w_i\cdot\left\{\int_{-T/2}^{T/2} Q\left(\frac{A_k u(0)-x_i}{\sigma_o}\right)\cdot f(\gamma_k)\cdot d\gamma_k\right\} + R_M \end{aligned} \qquad \text{(A3-41)}$$

, where:

$$R_M = \frac{1}{K_M^2 \cdot (2M)!} \cdot \int_{-T/2}^{T/2} Q^{(2M)}\left(\frac{A_k u(0) - \eta}{\sigma_o}\right) \cdot f(\gamma_k) \cdot d\gamma_k \tag{A3-42}$$

and:

$$K_M^{-2} = \prod_{i=1}^{M} \beta_i^2 \tag{A3-43}$$

Here, β_i represents the coefficients of the three-term recursive relationship of (3-24), $Q^{(2M)}(\cdot)$ represents the $(2M)^{th}$ derivative of the complementary distribution function [30], $\sigma_0{}^2$ is the output noise variance defined in (3-11) and ξ is a quantity which lies in the range of the interference variable η. Assuming that the interference is bounded, we have $a \leq \xi \leq b$ [2]. So, if we substitute (A3-43) into (A3-42), we get :

$$R_M = \frac{\prod_{i=1}^{M} \beta_i^2}{(2M)!} \cdot \int_{-T/2}^{T/2} Q^{(2M)}\left(\frac{A_k u(0) - \eta}{\sigma_o}\right) \cdot f(\gamma_k) \cdot d\gamma_k \tag{A3-44}$$

The definition of the complementary distribution function is [31]:

$$Q(x) = \int_{x}^{+\infty} Z(t) \cdot dt \tag{A3-45}$$

, where :

$$Z(t) = \frac{1}{\sqrt{2\pi}} e^{-t^2/2} \tag{A3-46}$$

The n^{th} derivative of the complementary distribution function is given by [30, p. 934] :

$$Q^{(n)}(x) = -Z^{(n-1)}(x) = (-1)^n \cdot Z(x) \cdot H_{n-1}(x) \tag{A3-47}$$

, where $H_n(x)$ is the n^{th} Hermite polynomial which satisfies the recursive relationship [30, p.782] :

$$H_{n+1}(x) = x \cdot H_n(x) + n \cdot H_{n-1}(x) \tag{A3-48}$$

If we substitute (A3-47) into (A3-44) while making use of (A3-46), we obtain :

$$R_M = \frac{\prod_{i=1}^{M}\beta_i^2}{(2M)!}\cdot\int_{-T/2}^{T/2}\frac{(-1)^{2M}}{\sigma_0^{2M}}Z\left(\frac{A_k u(0)-\xi}{\sigma_o}\right)\cdot\left|H_{2M-1}\left(\frac{A_k u(0)-\xi}{\sigma_o}\right)\right|\cdot f(\gamma_k)\cdot d\gamma_k$$

$$= \frac{\prod_{i=1}^{M}\beta_i^2}{(2M)!\cdot\sigma_0^{2M}}\cdot\int_{-T/2}^{T/2}\frac{1}{\sqrt{2\pi}}\exp\left(\frac{-(A_k u(0)-\xi)^2}{2\sigma_0^2}\right)\cdot\left|H_{2M-1}\left(\frac{A_k u(0)-\xi}{\sigma_o}\right)\right|\cdot f(\gamma_k)\cdot d\gamma_k \qquad \text{(A3-49)}$$

$$= \frac{\prod_{i=1}^{M}\beta_i^2}{\sqrt{2\pi}\cdot(2M)!\cdot\sigma_0^{2M}}\cdot\int_{-T/2}^{T/2}\exp\left(\frac{-(A_k u(0)-\xi)^2}{2\sigma_0^2}\right)\cdot\left|H_{2M-1}\left(\frac{A_k u(0)-\xi}{\sigma_o}\right)\right|\cdot f(\gamma_k)\cdot d\gamma_k$$

The following upper bound on $H_n(x)$ is known [30, p. 787] :

$$|H_n(x)| < B\cdot e^{x^2/4}\cdot\sqrt{n!} \qquad \text{(A3-50)}$$

, where $B \approx 1.086435$. (A3-50) is then substituted into (A3-49) and thus, the truncation error is bounded by :

$$|R_M| < \frac{B\sqrt{(2M+1)!}\cdot\prod_{i=1}^{M}\beta_i^2}{\sqrt{2\pi}\cdot(2M)!\cdot\sigma_o^{2M}}\cdot\int_{-T/2}^{T/2}\left\{\begin{matrix}\exp\left(\dfrac{-(A_k u(0)-\xi)^2}{2\sigma_o^2}\right)\\ \cdot\exp\left(\dfrac{(A_k u(0)-\xi)^2}{4\sigma_o^2}\right)\end{matrix}\right\}\cdot f(\gamma_k)\cdot d\gamma_k \qquad \text{(A3-51)}$$

Once (A3-51) is simplified, its form will correspond exactly to that of (3-42). To maximize the upper bound of (3-42), we must maximize ξ thus we will now show that (3-44) is true [2]. We first maximize the ISI term in (3-36), corresponding to (A3-31). To do so, we maximize the random quantities present in (A3-31). Its maximum value is :

$$max\ \eta_1 = \max_{\gamma_k}\sum_{\substack{i=\alpha-m_l\\ i\neq\alpha}}^{\alpha+m_u}\left|A_k\cdot u(\alpha T_b - iT_b)\right| \qquad \text{(A3-52)}$$

Next, we maximize the ICI term of (3-36), corresponding to (A3-32), thus :

$$max\ \eta_2 = \sum_{q\neq k}A_q\max_{\gamma_q,\phi_q}\sum_{i=\alpha-m_{l,q}}^{\alpha+m_{u,q}}\left|\begin{matrix}v^I(\alpha T_b - iT_b)\cdot(\cos\phi_q+\sin\phi_q)\\ +v^Q(\alpha T_b - iT_b)\cdot(\cos\phi_q-\sin\phi_q)\end{matrix}\right|. \qquad \text{(A3-53)}$$

We can now maximize (A3-53) with respect to ϕ_q . It is easily seen that (A3-53) is maximized if $|\phi_q| = \pi/4$ and by setting this value, (A3-53) becomes :

$$max\ \eta_2 = \sqrt{2}\cdot\sum_{q\neq k}A_q\max_{\gamma_q}\sum_{i=\alpha-m_{l,q}}^{\alpha+m_{u,q}}\left|v^I(\alpha T_b - iT_b)\right| + \left|v^Q(\alpha T_b - iT_b)\right| \qquad \text{(A3-54)}$$

Summing together (A3-52) and (A3-54) will give us (3-44). To minimize the interference, we must minimize the ISI and ICI components likewise. To do so, it is easy to see from (A3-31) that we must minimize with respect to the timing phase on the k^{th} channel. Also, the ISI is minimum if all interfering symbols are most negative, therefore:

$$min\ \eta_1 = -\min_{\gamma_k} \sum_{\substack{i=\alpha-m_l \\ i\neq\alpha}}^{\alpha+m_u} \left| A_k \cdot u(\alpha T_b - iT_b) \right| \tag{A3-55}$$

The same applies to the ICI component of (A3-32). We minimize with respect to the timing phase, set the interfering symbols to their most negative value and set $|\phi_q| = \pi/4$ to maximize the magnitude of the ICI. By doing so, the minimum ICI value is :

$$min\ \eta_2 = -\sqrt{2} \cdot \sum_{q\neq k} A_q \min_{\gamma_q} \sum_{i=\alpha-m_{l,q}}^{\alpha+m_{u,q}} \left| v^I(\alpha T_b - iT_b) \right| + \left| v^Q(\alpha T_b - iT_b) \right| \tag{A3-56}$$

Finally, summing together (A3-55) and (A3-56) will give us (3-45).

We will now derive the form of the second type of truncation error, that is, the truncation error due to considering a limited number of interfering symbols on the desired channel and on the interfering channels [2]. We first start with the upper bound of (3-48) [11]. If we expand the left-hand side of (3-48) using (3-46), we obtain :

$$\begin{aligned} E\{\exp(A\eta'')\} &= E\left\{ \exp\left(\begin{aligned} & A \cdot \sum_{\substack{i<\alpha-m_l \\ i>\alpha+m_u}} A_k \cdot a_{k,i} \cdot u(\alpha T_b - iT_b) \\ & + A \cdot \sum_{q\neq k} A_q \cdot \sum_{\substack{i<\alpha-m_{l,q} \\ i>\alpha+m_{u,q}}} \left\{ \begin{aligned} & a_{q,i} \cdot \begin{bmatrix} v^I(\alpha T_b - iT_b) \cdot \cos\phi_q \\ -v^Q(\alpha T_b - iT_b) \cdot \sin\phi_q \end{bmatrix} \\ & -b_{q,i} \cdot \begin{bmatrix} v^I(\alpha T_b - iT_b) \cdot \sin\phi_q \\ +v^Q(\alpha T_b - iT_b) \cdot \cos\phi_q \end{bmatrix} \end{aligned} \right\} \end{aligned} \right) \right\} \\ &= E\left\{ \begin{aligned} & \prod_{\substack{i<\alpha-m_l \\ i>\alpha+m_u}} \exp\left(A \cdot A_k \cdot a_{k,i} \cdot u(\alpha T_b - iT_b) \right) \\ & \cdot \prod_{q\neq k} \prod_{\substack{i<\alpha-m_{l,q} \\ i>\alpha+m_{u,q}}} \exp\left(A \cdot A_q \cdot \left\{ \begin{aligned} & a_{q,i} \cdot \begin{bmatrix} v^I(\alpha T_b - iT_b) \cdot \cos\phi_q \\ -v^Q(\alpha T_b - iT_b) \cdot \sin\phi_q \end{bmatrix} \\ & -b_{q,i} \cdot \begin{bmatrix} v^I(\alpha T_b - iT_b) \cdot \sin\phi_q \\ +v^Q(\alpha T_b - iT_b) \cdot \cos\phi_q \end{bmatrix} \end{aligned} \right\} \right) \end{aligned} \right\} \end{aligned} \tag{A3-57}$$

If the data symbols are independent, we get from (A3-57) [31]:

$$E\{\exp(A\eta'')\} = \prod_{\substack{i<\alpha-m_l \\ i>\alpha+m_u}} E\left\{ \exp\left(A \cdot \eta_{1,i} \right) \right\} \cdot \prod_{q\neq k} \prod_{\substack{i<\alpha-m_{l,q} \\ i>\alpha+m_{u,q}}} E\left\{ \exp\left(A \cdot \eta_{2,q,i} \right) \right\} \tag{A3-58}$$

, where $\eta_{1,i}$ and $\eta_{2,q,i}$ have been defined as the ISI of the interference in (A3-31) and the ICI of the interference in (A3-32), respectively, for fixed i. The first expectation may be upper bounded as [11] :

$$E\left\{\exp(A\,\eta_{1,i})\right\} \le \exp\left(\frac{A^2}{2}\cdot E\left\{\eta_{1,i}^2\right\}\right)$$
$$= \exp\left(\frac{A^2}{2}\cdot A_k^2\cdot \int_{-T/2}^{T/2}\left(u(\alpha T_b - iT_b)\right)^2\cdot f(\gamma_k)\cdot d\gamma_k\right) \tag{A3-59}$$

, and if γ_k is fixed, (A3-59) becomes :

$$E\left\{\exp(A\,\eta_{1,i})\right\} \le \exp\left(\frac{A^2}{2}\cdot A_k^2\cdot \left|u(\alpha T_b - iT_b)\right|^2\right) \tag{A3-60}$$

The ICI from (A3-58) may be upper bounded in a similar fashion [11] :

$$E\left\{\exp(A\,\eta_{2,q,i})\right\} \le \exp\left(\frac{A^2}{2}\cdot E\left\{\eta_{2,q,i}^2\right\}\right)$$
$$= \exp\left(\frac{A^2}{2}\cdot A_q^2\cdot \int\limits_{\substack{0\\ \phi_q}}^{2\pi}\int\limits_{\substack{-T/2\\ \gamma_q}}^{T/2}\left\{\begin{array}{l} v^I(\alpha T_b - iT_b)\cdot\begin{pmatrix}\cos\phi_q\\ +\sin\phi_q\end{pmatrix}\\ +v^Q(\alpha T_b - iT_b)\cdot\begin{pmatrix}\cos\phi_q\\ -\sin\phi_q\end{pmatrix}\end{array}\right\}^2 \begin{array}{c}\cdot f(\gamma_q)\cdot f(\phi_q)\\ \cdot d\gamma_q\cdot d\phi_q\end{array}\right) \tag{A3-61}$$

As was done for the ISI in (A3-59), we may fix γ_q and ϕ_q and so (A3-61) becomes :

$$E\left\{\exp(A\,\eta_{2,q,i})\right\} \le \exp\left(\frac{A^2}{2}\cdot A_q^2\cdot \left|\begin{array}{l} v^I(\alpha T_b - iT_b)\cdot\left(\cos\phi_q + \sin\phi_q\right)\\ +v^Q(\alpha T_b - iT_b)\cdot\left(\cos\phi_q - \sin\phi_q\right)\end{array}\right|^2\right) \tag{A3-62}$$

So, if we substitute (A3-60) and (A3-62) into (A3-58), we get :

$$E\left\{\exp(A\,\eta'')\right\} \le \exp\left\{\frac{A^2}{2}\left[\begin{array}{l} A_k^2 \sum\limits_{\substack{i<\alpha-m_l\\ i>\alpha+m_u}}\left|u(\alpha T_b - iT_b)\right|^2\\ +\sum\limits_{q\neq k} A_q^2 \sum\limits_{\substack{i<\alpha-m_{l,q}\\ i>\alpha+m_{u,q}}}\left|\begin{array}{l} v^I(\alpha T_b - iT_b)\cdot\left(\cos\phi_q + \sin\phi_q\right)\\ +v^Q(\alpha T_b - iT_b)\cdot\left(\cos\phi_q - \sin\phi_q\right)\end{array}\right|^2\end{array}\right]\right\} \tag{A3-63}$$

If we then compare the form of (3-48) to that of (A3-63), we see straightforwardly, for fixed γ_q and ϕ_q :

$$
\begin{aligned}
\sigma_r^2 &= A_k^2 \sum_{\substack{i<\alpha-m_l \\ i>\alpha+m_u}} \left|u(\alpha T_b - iT_b)\right|^2 \\
&\quad + \sum_{q\neq k} A_q^2 \sum_{\substack{i<\alpha-m_{l,q} \\ i>\alpha+m_{u,q}}} \left| \begin{array}{l} v^I(\alpha T_b - iT_b)\cdot\left(\cos\phi_q + \sin\phi_q\right) \\ +v^Q(\alpha T_b - iT_b)\cdot\left(\cos\phi_q - \sin\phi_q\right) \end{array} \right|^2
\end{aligned}
\tag{A3-64}
$$

In order for (A3-64) to be unconditioned on γ_q and ϕ_q , we choose their maximum values in order to maximize $\sigma_r{}^2$. Thus, using the maximum values of γ_q and ϕ_q , we must find :

$$
\begin{aligned}
\sigma_r^2 &= \max_{\gamma_k} A_k^2 \sum_{\substack{i<\alpha-m_l \\ i>\alpha+m_u}} \left|u(\alpha T_b - iT_b)\right|^2 \\
&\quad + \max_{\gamma_q,\phi_q} \sum_{q\neq k} A_q^2 \sum_{\substack{i<\alpha-m_{l,q} \\ i>\alpha+m_{u,q}}} \left| \begin{array}{l} v^I(\alpha T_b - iT_b)\cdot\left(\cos\phi_q + \sin\phi_q\right) \\ +v^Q(\alpha T_b - iT_b)\cdot\left(\cos\phi_q - \sin\phi_q\right) \end{array} \right|^2
\end{aligned}
\tag{A3-65}
$$

Once we choose ϕ_q appropriately, as was done previously, we will have the upper bound of (3-49).

A3.4 : Details on the Truncation Errors of the Beaulieu FSE Technique (Derivation of (3-64), (3-65))

First, we observe the derivation of the truncation error, R_M , due to using only M terms in the Fourier Series. This truncation error is easily defined as the terms neglected in the finite summation of the probability of bit error expression of (3-61) [2] [16]. R_M is thus defined as :

$$
\left|R_M\right| = \left| -\tfrac{2}{\pi} \sum_{\substack{m=M+1 \\ m\,\mathrm{odd}}}^{+\infty} \frac{e^{-m^2\omega^2/2}}{m} \cdot \left\{ \int_{-T/2}^{T/2} \sin(m\omega\, A_k u(0)) \cdot f(\gamma_k)\cdot d\gamma_k \right\} \cdot \Phi_\eta(-m\omega) \right|
\tag{A3-66}
$$

We can upper bound this truncation error as follows [16] :

$$
\begin{aligned}
\left|R_M\right| &\leq \frac{2}{\pi}\cdot \sum_{\substack{m=M+1 \\ m\,\mathrm{odd}}}^{+\infty} \frac{e^{-m^2\omega^2/2}}{m} < \frac{2}{\pi}\cdot \int_M^{+\infty} \frac{e^{-\alpha^2\omega^2/2}}{\alpha} d\alpha < \frac{2}{\pi M}\cdot \int_M^{+\infty} e^{-\alpha^2\omega^2/2}\cdot d\alpha \\
&< \frac{\sqrt{2\pi}\cdot T}{\pi^2\cdot M}\cdot Q\!\left(\frac{2\pi M}{T}\right)
\end{aligned}
\tag{A3-67}
$$

From (3-63) and (A3-67), we see a tradeoff between M and T. (3-63) indicates that in order to keep β small, we need to increase T. However, (A3-67) indicates that by increasing T, we need to increase M also to keep R_M small. But from the bit error rate expression of (3-61) we see that by increasing M, we are taking into account more terms in the Fourier Series thus more time would be required to compute the bit error rate. Thus, for a better accuracy, we need to increase computation time significantly. The problem is to find the best values for *M* and *T* for a specified level of accuracy. As it was discussed in section 3.4.2.2, it is possible to make an optimum choice for both *M* and *T*.

To understand how the truncation error, Δ , due to using only a finite number of interference terms was derived, we will first look into the truncated characteristic function, that is, giving consideration to a finite number of interfering symbols. Assuming that data symbols are transmitted independently, the characteristic function of the interference, $\Phi_\eta(\omega)$, is the product of an infinite number of symbols on the desired and interfering channels [10] [31]. If we consider *K* interfering symbols in total, given by (3-37), the characteristic function can be approximated by :

$$\Phi^*_\eta(-m\omega) = \prod_{\substack{i=\alpha-m_i \\ i\neq\alpha}}^{\alpha+m_i} \Phi_{\eta_{1,i}}(-m\omega) \cdot \prod_{\substack{q=0 \\ q\neq k}}^{N-1} \prod_{i=\alpha-m_{i,q}}^{\alpha+m_{i,q}} \Phi_{\eta_{2,q,i}}(-m\omega) \tag{A3-68}$$

, where the ISI characteristic function of each ISI term can be easily seen from (A3-31). It is represented by [31]:

$$\Phi_{\eta_{1,i}}(-m\omega) = E\left\{e^{-jm\omega\eta_{1,i}}\right\} = \int_{-T/2}^{T/2} \cos\left(m\omega\, A_k \cdot u(\alpha T_b - iT_b)\right) \cdot f(\gamma_k) \cdot d\gamma_k \tag{A3-69}$$

Likewise, the characteristic function of each ICI term can be easily seen from (A3-32) and thus is [31]:

$$\Phi_{\eta_{2,q,i}}(-m\omega) = E\left\{e^{-jm\omega\eta_{2,q,i}}\right\}$$
$$= \frac{1}{2\pi} \int_{\substack{0 \\ \phi_q}}^{2\pi} \int_{\substack{-T/2 \\ \gamma_q}}^{T/2} \left\{ \begin{array}{l} \cos\left(m\omega\, A_q \cdot \begin{pmatrix} v^I(\alpha T_b - iT_b) \cdot \cos\phi_q \\ -v^Q(\alpha T_b - iT_b) \cdot \sin\phi_q \end{pmatrix} \right) \\ \cdot \cos\left(m\omega\, A_q \cdot \begin{pmatrix} v^I(\alpha T_b - iT_b) \cdot \sin\phi_q \\ -v^Q(\alpha T_b - iT_b) \cdot \cos\phi_q \end{pmatrix} \right) \end{array} \right\} \cdot f(\gamma_q) \cdot d\gamma_q \cdot d\phi_q \tag{A3-70}$$

Now we can use (A3-68) in the probability of bit error expression of (3-61), thus :

$$P_b = \frac{1}{2} - \frac{2}{\pi} \sum_{\substack{m=1 \\ m\ \mathrm{odd}}}^{M} \frac{e^{-m^2\omega^2/2}}{m} \cdot \left\{ \int_{-T/2}^{T/2} \sin(m\omega\, A_k u(0)) \cdot f(\gamma_k) \cdot d\gamma_k \right\} \cdot \Phi^*_\eta(-m\omega)$$
$$+ R_M + \beta + \Delta \tag{A3-71}$$

The truncation error Δ is defined in a similar manner to R_M . Δ simply represents the difference between the ideal characteristic function of (3-52) and the truncated characteristic function of (A3-68). Thus, Δ is defined as:

$$
\begin{aligned}
|\Delta| &= \tfrac{2}{\pi}\sum_{\substack{m=1\\ m\ \mathrm{odd}}}^{M}\frac{e^{-m^2\sigma^2/2}}{m}\cdot\left\{\int_{-T/2}^{T/2}\sin(m\omega\, A_k u(0))\cdot f(\gamma_k)\cdot d\gamma_k\right\}\cdot\left(\Phi_\eta(-m\omega)-\Phi_\eta^{*}(-m\omega)\right) \\
&= \tfrac{2}{\pi}\sum_{\substack{m=1\\ m\ \mathrm{odd}}}^{M}\frac{e^{-m^2\sigma^2/2}}{m}\cdot\left\{\int_{-T/2}^{T/2}\sin(m\omega\, A_k u(0))\cdot f(\gamma_k)\cdot d\gamma_k\right\}\cdot\Phi_\eta^{*}(-m\omega)\left(\Phi_\eta^{**}(-m\omega)-1\right)
\end{aligned}
\qquad \text{(A3-72)}
$$

, where $\Phi_\eta^{**}(-m\omega)$ is the characteristic function comprising all the interfering symbols not included in the truncated characteristic function. It is defined as :

$$
\Phi_\eta^{**}(-m\omega)=\prod_{\substack{i<\alpha-m_l\\ i>\alpha+m_u}}\Phi_{\eta_{1,i}}(-m\omega)\cdot\prod_{\substack{q=0\\ q\neq k}}^{N-1}\prod_{\substack{j<\alpha-m_{l,q}\\ j>\alpha+m_{u,q}}}\Phi_{\eta_{2,q,j}}(-m\omega) \qquad \text{(A3-73)}
$$

We also note that :

$$
\Phi_\eta(\omega)=\Phi_\eta^{*}(-m\omega)\cdot\Phi_\eta^{**}(-m\omega) \qquad \text{(A3-74)}
$$

It can be shown how (A3-72) may be simplified [16]. To do so, we will assume that the timing and phase offsets are fixed. Assuming fixed timing and phase offsets, the characteristic functions for the ISI and ICI from (A3-69) and (A3-70), respectively, become :

$$
\Phi_{\eta_{1,i}}(-m\omega)=\cos(a_i) \qquad \text{(A3-75)}
$$

$$
\Phi_{\eta_{2,q,j}}(-m\omega)=\frac{1}{2}\cdot\left(\cos(b_{q,j})+\cos(c_{q,j})\right) \qquad \text{(A3-76)}
$$

, where :

$$
\begin{aligned}
a_i &= m\omega\, A_k\cdot u(\alpha T_b-iT_b) \\
b_{q,j} &= m\omega\, A_q\cdot\left\{v^I(\alpha T_b-iT_b)\cdot\left(\cos\phi_q+\sin\phi_q\right)+v^Q(\alpha T_b-iT_b)\cdot\left(\cos\phi_q-\sin\phi_q\right)\right\} \\
c_{q,j} &= m\omega\, A_q\cdot\left\{v^I(\alpha T_b-iT_b)\cdot\left(\cos\phi_q-\sin\phi_q\right)-v^Q(\alpha T_b-iT_b)\cdot\left(\cos\phi_q+\sin\phi_q\right)\right\}
\end{aligned}
\qquad \text{(A3-77)}
$$

Thus, (A3-73) becomes [34]:

$$
\Phi_\eta^{**}(-m\omega)=\frac{1}{2}\cdot\prod_{\substack{i<\alpha-m_l\\ i>\alpha+m_u}}\cos(a_i)\cdot\prod_{\substack{q=0\\ q\neq k}}^{N-1}\prod_{\substack{j<\alpha-m_{l,q}\\ j>\alpha+m_{u,q}}}\left(\cos(b_{q,J})+\cos(c_{q,J})\right) \qquad \text{(A3-78)}
$$

Using the fact that [16] :

$$
\prod_{k=I}^{+\infty}\cos(a_k)\geq 1-\frac{1}{2}\cdot\sum_{k=I}^{+\infty}a_k^2 \quad ; \quad |a_k|\leq 1 \qquad \text{(A3-79)}
$$

, we get :

$$\left|\Phi_\eta^{**}(-m\omega)-1\right| \le \frac{1}{2}\cdot\sum_{\substack{i=\alpha-m_l\\ i\ne\alpha}}^{\alpha+m_u} a_i^2 + \frac{1}{4}\cdot\sum_{q\ne k}\sum_{i=\alpha-m_{l,q}}^{\alpha+m_{u,q}} b_{q,i}^2 + \frac{1}{4}\cdot\sum_{q\ne k}\sum_{i=\alpha-m_{l,q}}^{\alpha+m_{u,q}} c_{q,i}^2 \qquad \text{(A3-80)}$$

If we substitute (A3-80) into (A3-72), we obtain for Δ :

$$|\Delta| \le \frac{2}{\pi}\cdot\sum_{\substack{m=1\\ m\ \mathrm{odd}}}^{M}\frac{e^{-m^2\omega^2/2}}{m}\cdot\sin(m\omega\,A_k u(0))\cdot\Phi_\eta^{*}(-m\omega)\cdot\left\{\begin{matrix}\frac{1}{2}\cdot\sum\limits_{\substack{i=\alpha-m_l\\ i\ne\alpha}}^{\alpha+m_u} a_i^2\\ +\frac{1}{4}\cdot\sum\limits_{q\ne k}\sum\limits_{i=\alpha-m_{l,q}}^{\alpha+m_{u,q}} b_{q,i}^2+\frac{1}{4}\cdot\sum\limits_{q\ne k}\sum\limits_{i=\alpha-m_{l,q}}^{\alpha+m_{u,q}} c_{q,i}^2\end{matrix}\right\} \qquad \text{(A3-81)}$$

Likewise, if we substitute (A3-77) into (A3-81) and simplify the expression by taking the quantities within the parentheses out of the summation over *m*, we get an expression closer to that of (3-65). It is :

$$|\Delta| \le \frac{\omega^2}{\pi}\cdot\left\{\begin{matrix}\sum\limits_{\substack{i=\alpha-m_l\\ i\ne\alpha}}^{\alpha+m_u} A_k^2\cdot u^2(\alpha T_b-iT_b)\\ +\frac{1}{2}\cdot\sum\limits_{q\ne k}A_q^2\sum\limits_{i=\alpha-m_{l,q}}^{\alpha+m_{u,q}}\left\{\begin{matrix}v^I(\alpha T_b-iT_b)\cdot\left(\cos\phi_q+\sin\phi_q\right)\\ +v^Q(\alpha T_b-iT_b)\cdot\left(\cos\phi_q-\sin\phi_q\right)\end{matrix}\right\}^2\\ +\frac{1}{2}\cdot\sum\limits_{q\ne k}A_q^2\sum\limits_{i=\alpha-m_{l,q}}^{\alpha+m_{u,q}}\left\{\begin{matrix}v^I(\alpha T_b-iT_b)\cdot\left(\cos\phi_q-\sin\phi_q\right)\\ -v^Q(\alpha T_b-iT_b)\cdot\left(\cos\phi_q+\sin\phi_q\right)\end{matrix}\right\}^2\end{matrix}\right\} \qquad \text{(A3-82)}$$

$$\cdot\sum_{\substack{m=1\\ m\ \mathrm{odd}}}^{M} m\cdot e^{-m^2\omega^2/2}\cdot\sin(m\omega\,A_k u(0))\cdot\Phi_\eta^{*}(-m\omega)$$

Once the squared ICI terms in (A3-82) are expanded and simplified, the form of (3-65) for Δ is obtained.

A3.5 : Details on the Computational Complexity of the M-MCDD (Derivation of (3-70))

The computational complexity of the M-MCDD, used to help determine the values of *Ltap* and *Itap*, is based on the total number of operations (additions and multiplications) needed to compute the ISI of (3-8) and the ICI of (3-13) and (3-14) in its in-phase and quadrature components, respectively [6]. The procedure to derive (3-70) is to first look at the number of operations needed to compute the ISI and then to look at the number of operations to compute the

ICI. First, for the ISI of (3-8), we assume a fixed interfering symbol we are trying to compute. From (3-8), we see that there are two non-trivial multiplications per iteration. Now, the question is how many iterations does (3-8) go through. To determine the total number of iterations, we must bear in mind that there are *Ltap* non-zero filter taps per halfband filter thus the total number of filter taps (including the zero filter taps) per halfband filter is *2Ltap-3*, that is *Ltap-2* filter taps on each side plus a zero-position filter tap. Refer to Figure 2-9 for more detail [27]. When we go through the n_0 loop in (3-8), there is no decimation present. Therefore, we have *Ltap* iterations which result in the non-trivial multiplications. When we go through all other n_j iterative loops, where we have decimation, all the halfband filter taps are non-zero so there are *2Ltap-3* iterations which result in the non-trivial multiplications. Therefore, from the n_0 loop where there are *Ltap* iterations to consider, there are exactly *2Ltap* non-trivial multiplications and *Ltap-1* additions giving a total of *3Ltap-1* operations from this loop. From all other iterative loops where there are *2Ltap-3* iterations to consider, there are exactly *4Ltap-6* non-trivial multiplications and *2Ltap-4* additions giving a total of *6Ltap-10* operations from each of these loops. Therefore, the total number of operations from the n_j loops where $0 \le j \le l$, including the l additions from one iteration to the next, is $3 \cdot Ltap - 1 + l \cdot (6Ltap - 10) + l$. We then go through *Itap* iterations, therefore the number of operations after *Itap* iterations, including the *Itap-1* additions from one iteration to the next, is $Itap \cdot (3 \cdot Ltap - 1 + l \cdot (6Ltap - 10) + l) + Itap - 1$. If we then consider all *mL+mU* interfering symbols on the desired k^{th} channel, the total number of operations to compute the ISI of (3-8) becomes :

$$M_{ISI} = (mL + mU) \cdot \{Itap \cdot [(6l + 3) \cdot Ltap - (9l + 1)] + Itap - 1\} \qquad \text{(A3-83)}$$

As for the ICI, we will consider the in-phase component of (3-13) and the quadrature component of (3-14) separately. In both cases, we see that we have l non-trivial multiplications per iteration due to the cos(·) and sin(·) of (3-15) and (3-16), respectively. In addition, some terms in (3-15) and (3-16) require one extra non-trivial multiplication therefore, to simplify the analysis, we can say that in each of these iterations, we can have up to one more non-trivial multiplication. Therefore, for the in-phase and quadrature components, there is a maximum of *l+1* non-trivial multiplications per iteration. The difference lies in the fact that for the in-phase component there are *(l+1)/2* additional iterations imposed by cos(·) of (3-15), and for the quadrature component there are *(l-1)/2* additional iterations imposed by sin(·) of (3-16). This is true because since we assume $l=3$ in the numerical analysis, we therefore look at the *l odd* component of (3-15) and (3-16). Through each n_j loop, for simplicity, we will denote these required additions by x. For the in-phase and quadrature components of (3-13) and (3-14), respectively, we go through the n_j iterations, similar to what was done for the ISI. For the n_0 loop where there is no decimation, there are $Ltap\cdot\{(l+2)(x+1)+3\}-1$ operations, while for all other loops where there is decimation, there are $(2Ltap-3)\cdot\{(l+2)(x+1)+3\}-1$ operations for each iterative loop. Therefore, the total number of operations resulting from the n_j loops is $Ltap\cdot\{(l+2)(x+1)+3\}-1+l\cdot\{(2Ltap-3)\cdot\{(l+2)(x+1)+3\}-1\}+l$. When we go through the *Itap* iterations for each and finally consider all of the $mL+mU+1$ interfering symbols from each of the interfering channels, we obtain for the total number of operations to compute the in-phase and quadrature components of the ICI of (3-13) and (3-14), respectively :

$$M_{I,ICI}=\left(2^l-1\right)\cdot\left(mL+mU+1\right)\cdot\left\{\begin{array}{l} Itap\cdot\left[\begin{array}{l} Ltap\cdot\left(l^3+\frac{11}{2}l^2+\frac{29}{2}l+6\right) \\ -\left(\frac{3}{2}l^3+\frac{15}{2}l^2+18l+1\right) \end{array}\right] \\ +Itap-1 \end{array}\right\} \qquad \text{(A3-84)}$$

$$M_{Q,ICI}=\left(2^l-1\right)\cdot\left(mL+mU+1\right)\cdot\left\{\begin{array}{l} Itap\cdot\left[\begin{array}{l} Ltap\cdot\left(l^3+\frac{7}{2}l^2+\frac{19}{2}l+4\right) \\ +\left(\frac{3}{2}l^3-\frac{15}{2}l^2-12l-1\right) \end{array}\right] \\ +Itap-1 \end{array}\right\} \qquad \text{(A3-85)}$$

If we sum together (A3-83), (A3-84) and (A3-85) and then simplify, we obtain for the computational complexity of the M-MCDD to compute the interference :

$$\begin{aligned} M_{op}=&\left[\left(mL+mU\right)\cdot(6l+3)+\left(2^l-1\right)\cdot\left(mL+mU+1\right)\cdot\left(2l^3+9l^2+24l+10\right)\right]\cdot Itap\cdot Ltap \\ &-l\cdot\left[9\cdot\left(mL+mU\right)+\left(2^l-1\right)\cdot\left(mL+mU+1\right)\cdot 15(l+2)\right]\cdot Itap \qquad \text{(A3-86)} \\ &-\left[\left(mL+mU\right)+2\cdot\left(2^l-1\right)\cdot\left(mL+mU+1\right)\right] \end{aligned}$$

If we thus substitute $mL=mU=10$ and $l=3$, as was assumed for the performance of the M-MCDD, we obtain the expression of (3-70).

A4 : Derivations From Chapter Four

A4.1 : Details of Quantization Noise Analysis for Probability of Bit Error

A4.1.1 : Output of the M-MCDD (Derivation of (4-7), (4-8), (4-9))

From the M-MCDD structure presented in Figure A-1, the output from the k^{th} desired channel has been determined and is indicated by (A3-9). It does not include the effect of the DAF prior to the M-MCDD or the substitution of the MF-TDMA input signal. From (A3-9), if $h(n_j \cdot 2^{J-1}T_s)$ are *quantized* to become $\hat{h}(n_j \cdot 2^{J-1}T_s)$, the resulting output from the M-MCDD is quantized and can be represented by :

$$\hat{y}_k(mT_c) = \begin{cases} (-1)^m \cdot \left\{ \begin{array}{l} \sum\limits_{n_1=-L}^{L} \hat{h}_1(n_1 T_s) \sum\limits_{n_2=-L}^{L} \hat{h}_2(n_2 2T_s) \sum\limits_{n_3=-L}^{L} \cdots \\ \cdots \sum\limits_{n_l=-L}^{L} \hat{h}_l(n_l 2^l T_s) \cdot x_{1,1}((2^l m - \sum\limits_{j=0}^{l} 2^j n_j)T_s) \cdot e^{j\pi(k+\frac{1}{2})\cdot\sum_{j=0}^{l} n_j} \end{array} \right\} & ; \begin{array}{l} k=0 \ , \\ k \ even \end{array} \\ (-1)^m \cdot \left\{ \begin{array}{l} \sum\limits_{n_1=-L}^{L} (-1)^{n_1 \cdot T_s} \cdot \hat{h}_1(n_1 T_s) \sum\limits_{n_2=-L}^{L} \hat{h}_2(n_2 2T_s) \sum\limits_{n_3=-L}^{L} \cdots \\ \cdots \sum\limits_{n_l=-L}^{L} \hat{h}_l(n_l 2^l T_s) \cdot x_{1,1}((2^l m - \sum\limits_{j=0}^{l} 2^j n_j)T_s) \cdot e^{j\pi(k+\frac{1}{2})\cdot\sum_{j=0}^{l} n_j} \end{array} \right\} & ; k \ odd \end{cases}$$

$$= \begin{cases} (-1)^m \sum\limits_{n_1=-L}^{L} \cdots \sum\limits_{n_l=-L}^{L} \left\{ \begin{array}{l} \prod\limits_{j=1}^{l} \left(\hat{h}_j(n_j \cdot 2^{J-1} T_s) \right) \\ \cdot x_{1,1}((2^l m - \sum\limits_{j=1}^{l} 2^{J-1} n_j)T_s) \cdot e^{j\pi(k+\frac{1}{2})\cdot\sum_{j=1}^{l} n_j} \end{array} \right\} & ; \begin{array}{l} k=0 \ , \\ k \ even \end{array} \\ (-1)^m \sum\limits_{n_1=-L}^{L} \cdots \sum\limits_{n_l=-L}^{L} \left\{ \begin{array}{l} (-1)^{n_1 \cdot T_s} \cdot \prod\limits_{j=1}^{l} \left(\hat{h}_j(n_j \cdot 2^{J-1} T_s) \right) \\ \cdot x_{1,1}((2^l m - \sum\limits_{j=1}^{l} 2^{J-1} n_j)T_s) \cdot e^{j\pi(k+\frac{1}{2})\cdot\sum_{j=1}^{l} n_j} \end{array} \right\} & ; k \ odd \end{cases} \qquad \text{(A4-1)}$$

If we may substitute (4-2), as described in section 4.1, into (A4-1) we obtain :

$$\hat{y}_k(mT_c)=\begin{cases} (-1)^m \sum\limits_{n_1=-L}^{L} \cdots\cdots \sum\limits_{n_l=-L}^{L} \left\{ \begin{array}{l} \prod\limits_{j=1}^{l}\left(h(n_j\cdot 2^{j-1}T_s)+\widetilde{h}_j(n_j\cdot 2^{j-1}T_s)\right) \\ \cdot x_{1,1}((2^l m-\sum\limits_{j=1}^{l}2^{j-1}n_j)T_s)\cdot e^{j\pi(k+\frac{1}{2})\cdot\sum_{j=1}^{l}n_j} \end{array} \right\} & ;\ \begin{array}{l} k=0\ , \\ k\ even \end{array} \\ (-1)^m \sum\limits_{n_1=-L}^{L} \cdots\cdots \sum\limits_{n_l=-L}^{L} \left\{ \begin{array}{l} (-1)^{n_1\cdot T_s}\cdot\prod\limits_{j=1}^{l}\left(h(n_j\cdot 2^{j-1}T_s)+\widetilde{h}_j(n_j\cdot 2^{j-1}T_s)\right) \\ \cdot x_{1,1}((2^l m-\sum\limits_{j=1}^{l}2^{j-1}n_j)T_s)\cdot e^{j\pi(k+\frac{1}{2})\cdot\sum_{j=1}^{l}n_j} \end{array} \right\} & ;\ k\ odd \end{cases} \tag{A4-2}$$

taking note that, from (4-2), the error processes $\widetilde{h}(n_j\cdot 2^{j-1}T_s)$ are represented by $\widetilde{h}_j(n_j\cdot 2^{j-1}T_s)$ because of the uniqueness of each error process from each stage. From (A4-2), we may expand the product :

$$\begin{aligned}\prod_{j=1}^{l}&\left(h(n_j\cdot 2^{j-1}T_s)+\widetilde{h}_j(n_j\cdot 2^{j-1}T_s)\right)= \\ &=\prod_{j=1}^{l}\left(h(n_j\cdot 2^{j-1}T_s)\right)+\prod_{j=1}^{l}\left(\widetilde{h}_j(n_j\cdot 2^{j-1}T_s)\right)+h(n_1T_s)\cdot\prod_{j=2}^{l}\left(\widetilde{h}_j(n_j\cdot 2^{j-1}T_s)\right) \\ &\quad+\prod_{j=1}^{2}\left(h(n_j\cdot 2^{j-1}T_s)\right)\cdot\prod_{j=3}^{l}\left(\widetilde{h}_j(n_j\cdot 2^{j-1}T_s)\right)+\cdots\cdots\end{aligned} \tag{A4-3}$$

Assuming that the filtering operations in every stage are identical, due to modularity, and that the error processes in each stage are also stochastically identical, we can conclude without loss of generality that [6-7] [31]:

$$\begin{aligned}\prod_{j=1}^{l}&\left(h(n_j\cdot 2^{j-1}T_s)+\widetilde{h}_j(n_j\cdot 2^{j-1}T_s)\right)= \\ &=\prod_{j=1}^{l}\left(\widetilde{h}_j(n_j\cdot 2^{j-1}T_s)\right)+\binom{l}{1}\cdot\prod_{j=1}^{1}\left(h(n_j\cdot 2^{j-1}T_s)\right)\cdot\prod_{j=2}^{l}\left(\widetilde{h}_j(n_j\cdot 2^{j-1}T_s)\right) \\ &\quad+\binom{l}{2}\cdot\prod_{j=1}^{2}\left(h(n_j\cdot 2^{j-1}T_s)\right)\cdot\prod_{j=3}^{l}\left(\widetilde{h}_j(n_j\cdot 2^{j-1}T_s)\right)+\cdots\cdots \\ &\quad+\binom{l}{l-1}\cdot\prod_{j=1}^{l-1}\left(h(n_j\cdot 2^{j-1}T_s)\right)\cdot\prod_{j=l}^{l}\left(\widetilde{h}_j(n_j\cdot 2^{j-1}T_s)\right)+\prod_{j=1}^{l}\left(h(n_j\cdot 2^{j-1}T_s)\right)\end{aligned} \tag{A4-4}$$

, thus :

$$\begin{aligned}\therefore\prod_{j=1}^{l}\left(h(n_j\cdot 2^{j-1}T_s)+\widetilde{h}_j(n_j\cdot 2^{j-1}T_s)\right)&=\sum_{a=1}^{l-1}\binom{l}{a}\cdot\prod_{j=1}^{a}\left(h(n_j\cdot 2^{j-1}T_s)\right)\cdot\prod_{j=a+1}^{l}\left(\widetilde{h}_j(n_j\cdot 2^{j-1}T_s)\right) \\ &\quad+\prod_{j=1}^{l}\left(\widetilde{h}_j(n_j\cdot 2^{j-1}T_s)\right)+\prod_{j=1}^{l}\left(h(n_j\cdot 2^{j-1}T_s)\right)\end{aligned} \tag{A4-5}$$

If we substitute (A4-5) into (A4-2), we see that (4-7) may be obtained where (3-6), (4-8) and (4-9) come directly from the substitution into (A4-2). Also, use is made of the fact that the DAF can be approximated by a halfband filter stage, similar to those used in the *l* subsequent stages of the M-MCDD [13].

A4.1.2 : Quantization Output of the Channel Detection Stage (Derivation of (4-10) to (4-12), (4-14) to (4-17))

In Appendix 3.1.2, we saw how $r_k(lT_b)$ was derived for the case of infinite quantization and for the lowpass outputs from the final stage, that is *k=0* and *k even*. Now, we will go through and show how (4-11) and (4-12), along with their respective components, were derived. For this derivation, we will also focus on the *k=0* and *k even* case since the same steps can be repeated to obtain the latter case, that is for the highpass outputs from the final stage (*k odd*). Therefore, following from (4-8) and (4-9), if we appropriately substitute the input MF-TDMA signal as described by (3-1) and (3-2) and isolate the desired term for the k^{th} channel similar to what was done for (3-7) in Appendix 3.1.2, we obtain for each quantization component of $\hat{y}_k(mT_c)$, respectively :

$$\begin{aligned}
\widetilde{y}_k(mT_c) &= A_k \cdot e^{j\phi_k} \cdot \sum_i e^{j\alpha_{k,i}} \sum_{n_0} \cdots\cdots \sum_{n_l} \left\{ \begin{array}{l} \prod_{j=0}^{l} \left(\widetilde{h}_j(n_j \cdot 2^j T_s) \right) \\ \cdot h_s[(2^l m - \sum_j 2^j n_j) \cdot T_s - iT_b - \gamma_k] \end{array} \right\} \\
&+ \sum_{q \neq k} A_q \cdot e^{j\phi_q} \cdot \sum_i e^{j\alpha_{q,i}} \sum_{n_0} \cdots\cdots \sum_{n_l} \left\{ \begin{array}{l} \prod_{j=0}^{l} \left(\widetilde{h}_j(n_j \cdot 2^j T_s) \right) \\ \cdot h_s[(2^l m - \sum_j 2^j n_j) \cdot T_s - iT_b - \gamma_q] \cdot e^{j\pi(k-q) \cdot \sum_j n_j} \end{array} \right\} \qquad \text{(A4-6)} \\
&+ (-1)^m \cdot \sum_{n_0} \cdots\cdots \sum_{n_l} \prod_{j=0}^{l} \left(\widetilde{h}_j(n_j \cdot 2^j T_s) \right) \cdot z[(2^l m - \sum_j 2^j n_j) \cdot T_s] \cdot e^{j\pi(k+\frac{1}{2}) \cdot \sum_j n_j}
\end{aligned}$$

$$
\begin{aligned}
y_k'(mT_c) = {} & A_k\cdot e^{j\phi_k}\cdot\sum_i e^{j\alpha_{k,i}}\sum_{n_0}\cdots\sum_{n_l}\left\{\begin{array}{l}\left\{\sum_{a=1}^{l}\binom{l+1}{a}\prod_{j=0}^{a-1}\left(h(n_j\cdot 2^jT_s)\right)\cdot\prod_{j=a}^{l}\left(\tilde{h}_j(n_j\cdot 2^jT_s)\right)\right\}\\ \cdot h_s[(2^lm-\sum_j 2^jn_j)\cdot T_s-iT_b-\gamma_k]\end{array}\right\}\\
& +\sum_{q\neq k}A_q\cdot e^{j\phi_q}\cdot\sum_i e^{j\alpha_{q,i}}\sum_{n_0}\cdots\sum_{n_l}\left\{\begin{array}{l}\left\{\sum_{a=1}^{l}\binom{l+1}{a}\prod_{j=0}^{a-1}\left(h(n_j\cdot 2^jT_s)\right)\cdot\prod_{j=a}^{l}\left(\tilde{h}_j(n_j\cdot 2^jT_s)\right)\right\}\\ \cdot h_s[(2^lm-\sum_j 2^jn_j)\cdot T_s-iT_b-\gamma_q]\end{array}\right\} \qquad \textbf{(A4-7)}\\
& +(-1)^m\cdot\sum_{n_0}\cdots\sum_{n_l}\left\{\begin{array}{l}\left\{\sum_{a=1}^{l}\binom{l+1}{a}\prod_{j=0}^{a-1}\left(h(n_j\cdot 2^jT_s)\right)\cdot\prod_{j=a}^{l}\left(\tilde{h}_j(n_j\cdot 2^jT_s)\right)\right\}\\ \cdot z[(2^lm-\sum_j 2^jn_j)\cdot T_s]\cdot e^{j\pi(k+\frac{1}{2})\cdot\sum_j n_j}\end{array}\right\}
\end{aligned}
$$

From (4-5) and (4-7), we see easily how (4-10) can be obtained. Thus, each of the quantization components of (4-10) comes out as :

$$\tilde{r}_k(lT_b)=\sum_{\xi=l_1}^{l_2}\tilde{y}_k[(\beta_l-\xi)T_c]\cdot g[(\xi+\mu_l)T_c] \qquad \textbf{(A4-8)}$$

$$r_k'(lT_b)=\sum_{\xi=l_1}^{l_2}y_k'[(\beta_l-\xi)T_c]\cdot g[(\xi+\mu_l)T_c] \qquad \textbf{(A4-9)}$$

, respectively. (4-11) and (4-12) may both be obtained in a similar manner, that is, by substituting (A4-6) and (A4-7) into (A4-8) and (A4-9), respectively, and using appropriate indices. Thus, if we substitute the appropriate indices according to those defined in (A4-8) and (A4-9), we obtain:

$$
\begin{aligned}
\tilde{y}_k[(\beta_l-\xi)T_c] = {} & A_k\cdot e^{j\phi_k}\cdot e^{j\alpha_{k,\alpha}}\sum_{n_0}\cdots\cdots\sum_{n_l}\left\{\begin{array}{l}\prod_{j=0}^{l}\left(\tilde{h}_j(n_j\cdot 2^jT_s)\right)\\ \cdot h_s[(2^l(\beta_l-\xi)-\sum_j 2^jn_j)\cdot T_s-\alpha T_b-\gamma_k]\end{array}\right\}\\
& +A_k\cdot e^{j\phi_k}\cdot\sum_{i\neq\alpha}e^{j\alpha_{k,i}}\sum_{n_0}\cdots\cdots\sum_{n_l}\left\{\begin{array}{l}\prod_{j=0}^{l}\left(\tilde{h}_j(n_j\cdot 2^jT_s)\right)\\ \cdot h_s[(2^l(\beta_l-\xi)-\sum_j 2^jn_j)\cdot T_s-iT_b-\gamma_k]\end{array}\right\}\\
& +\sum_{q\neq k}A_q\cdot e^{j\phi_q}\cdot\sum_i e^{j\alpha_{q,i}}\sum_{n_0}\cdots\cdots\sum_{n_l}\left\{\begin{array}{l}\prod_{j=0}^{l}\left(\tilde{h}_j(n_j\cdot 2^jT_s)\right)\\ \cdot h_s[(2^l(\beta_l-\xi)-\sum_j 2^jn_j)\cdot T_s-iT_b-\gamma_q]\\ \cdot e^{j\pi(k-q)\cdot\sum_j n_j}\end{array}\right\} \qquad \textbf{(A4-10)}\\
& +(-1)^{(\beta_l-\xi)}\cdot\sum_{n_0}\cdots\cdots\sum_{n_l}\left\{\begin{array}{l}\prod_{j=0}^{l}\left(\tilde{h}_j(n_j\cdot 2^jT_s)\right)\\ \cdot z[(2^l(\beta_l-\xi)-\sum_j 2^jn_j)\cdot T_s]\cdot e^{j\pi(k+\frac{1}{2})\cdot\sum_j n_j}\end{array}\right\}.
\end{aligned}
$$

$$
\begin{aligned}
&y_k'[(\beta_l-\xi)T_c]= \\
&=A_k\cdot e^{j\phi_k}\cdot e^{j\alpha_{k,\alpha}}\sum_{n_0}\cdots\sum_{n_l}\left\{\begin{array}{l}\left\{\sum_{a=1}^{l}\binom{l+1}{a}\prod_{j=0}^{a-1}\left(h(n_j\cdot 2^jT_s)\right)\cdot\prod_{j=a}^{l}\left(\widetilde{h}_j(n_j\cdot 2^jT_s)\right)\right\}\\ \cdot h_s[(2^l(\beta_l-\xi)-\sum_j 2^jn_j)\cdot T_s-\alpha T_b-\gamma_k]\end{array}\right\}\\
&+A_k\cdot e^{j\phi_k}\cdot\sum_{i\neq\alpha}e^{j\alpha_{k,i}}\sum_{n_0}\cdots\sum_{n_l}\left\{\begin{array}{l}\left\{\sum_{a=1}^{l}\binom{l+1}{a}\prod_{j=0}^{a-1}\left(h(n_j\cdot 2^jT_s)\right)\cdot\prod_{j=a}^{l}\left(\widetilde{h}_j(n_j\cdot 2^jT_s)\right)\right\}\\ \cdot h_s[(2^l(\beta_l-\xi)-\sum_j 2^jn_j)\cdot T_s-iT_b-\gamma_k]\end{array}\right\}\\
&+\sum_{q\neq k}A_q\cdot e^{j\phi_q}\cdot\sum_i e^{j\alpha_{q,i}}\sum_{n_0}\cdots\sum_{n_l}\left\{\begin{array}{l}\left\{\sum_{a=1}^{l}\binom{l+1}{a}\prod_{j=0}^{a-1}\left(h(n_j\cdot 2^jT_s)\right)\cdot\prod_{j=a}^{l}\left(\widetilde{h}_j(n_j\cdot 2^jT_s)\right)\right\}\\ \cdot h_s[(2^l(\beta_l-\xi)-\sum_j 2^jn_j)\cdot T_s-iT_b-\gamma_q]\\ \cdot e^{j\pi(k-q)\cdot\sum_j n_j}\end{array}\right\}\\
&+(-1)^{(\beta_l-\xi)}\cdot\sum_{n_0}\cdots\sum_{n_l}\left\{\begin{array}{l}\left\{\sum_{a=1}^{l}\binom{l+1}{a}\prod_{j=0}^{a-1}\left(h(n_j\cdot 2^jT_s)\right)\cdot\prod_{j=a}^{l}\left(\widetilde{h}_j(n_j\cdot 2^jT_s)\right)\right\}\\ \cdot z[(2^l(\beta_l-\xi)-\sum_j 2^jn_j)\cdot T_s]\cdot e^{j\pi(k+\frac{1}{2})\cdot\sum_j n_j}\end{array}\right\}
\end{aligned}
\qquad\text{(A4-11)}
$$

Substituting (A4-10) and (A4-11) into (A4-8) and (A4-9), respectively, gives us for the output of the channel detection filter resulting from finite quantization :

$$
\begin{aligned}
\widetilde{r}_k(lT_b)&=A_k\cdot e^{j\phi_k}\cdot e^{j\alpha_{k,\alpha}}\sum_{\xi=I_1}^{I_2}\sum_{n_0}\cdots\cdots\sum_{n_l}\left\{\begin{array}{l}\prod_{j=0}^{l}\left(\widetilde{h}_j(n_j\cdot 2^jT_s)\right)\\ \cdot h_s[(2^l(\beta_l-\xi)-\sum_j 2^jn_j)\cdot T_s-\alpha T_b-\gamma_k]\\ \cdot g[(\xi+\mu_l)T_c]\end{array}\right\}\\
&+A_k\cdot e^{j\phi_k}\cdot\sum_{i\neq\alpha}e^{j\alpha_{k,i}}\sum_{\xi=I_1}^{I_2}\sum_{n_0}\cdots\cdots\sum_{n_l}\left\{\begin{array}{l}\prod_{j=0}^{l}\left(\widetilde{h}_j(n_j\cdot 2^jT_s)\right)\\ \cdot h_s[(2^l(\beta_l-\xi)-\sum_j 2^jn_j)\cdot T_s-iT_b-\gamma_k]\\ \cdot g[(\xi+\mu_l)T_c]\end{array}\right\}\\
&+\sum_{q\neq k}A_q\cdot e^{j\phi_q}\cdot\sum_i e^{j\alpha_{q,i}}\sum_{\xi=I_1}^{I_2}\sum_{n_0}\cdots\cdots\sum_{n_l}\left\{\begin{array}{l}\prod_{j=0}^{l}\left(\widetilde{h}_j(n_j\cdot 2^jT_s)\right)\\ \cdot h_s[(2^l(\beta_l-\xi)-\sum_j 2^jn_j)\cdot T_s-iT_b-\gamma_q]\\ \cdot g[(\xi+\mu_l)T_c]\cdot e^{j\pi(k-q)\cdot\sum_j n_j}\end{array}\right\}\\
&+\sum_{\xi=I_1}^{I_2}(-1)^{(\beta_l-\xi)}\cdot\sum_{n_0}\cdots\cdots\sum_{n_l}\left\{\begin{array}{l}\prod_{j=0}^{l}\left(\widetilde{h}_j(n_j\cdot 2^jT_s)\right)\cdot z[(2^l(\beta_l-\xi)-\sum_j 2^jn_j)\cdot T_s]\\ \cdot g[(\xi+\mu_l)T_c]\cdot e^{j\pi(k+\frac{1}{2})\cdot\sum_j n_j}\end{array}\right\}
\end{aligned}
\qquad\text{(A4-12)}
$$

$$
\begin{aligned}
r_k'(lT_b) = {} & A_k \cdot e^{j\phi_k} \cdot e^{ja_{k,\alpha}} \sum_{\xi=I_1}^{I_2} \sum_{n_0} \cdots \sum_{n_l} \left\{ \begin{array}{l} \left\{ \sum_{a=1}^{l} \binom{l+1}{a} \prod_{j=0}^{a-1} \left(h(n_j \cdot 2^j T_s) \right) \cdot \prod_{j=a}^{l} \left(\tilde{h}_j(n_j \cdot 2^j T_s) \right) \right\} \\ \cdot h_s[(2^l(\beta_l - \xi) - \sum_j 2^j n_j) \cdot T_s - \alpha T_b - \gamma_k] \\ \cdot g[(\xi + \mu_l)T_c] \end{array} \right\} \\
& + A_k \cdot e^{j\phi_k} \cdot \sum_{\iota \neq \alpha} e^{ja_{k,\iota}} \sum_{\xi=I_1}^{I_2} \sum_{n_0} \cdots \sum_{n_l} \left\{ \begin{array}{l} \left\{ \sum_{a=1}^{l} \binom{l+1}{a} \prod_{j=0}^{a-1} \left(h(n_j \cdot 2^j T_s) \right) \cdot \prod_{j=a}^{l} \left(\tilde{h}_j(n_j \cdot 2^j T_s) \right) \right\} \\ \cdot h_s[(2^l(\beta_l - \xi) - \sum_j 2^j n_j) \cdot T_s - iT_b - \gamma_k] \\ \cdot g[(\xi + \mu_l)T_c] \end{array} \right\} \\
& + \sum_{q \neq k} A_q \cdot e^{j\phi_q} \cdot \sum_{\iota} e^{ja_{q,\iota}} \sum_{\xi=I_1}^{I_2} \sum_{n_0} \cdots \sum_{n_l} \left\{ \begin{array}{l} \left\{ \sum_{a=1}^{l} \binom{l+1}{a} \prod_{j=0}^{a-1} \left(h(n_j \cdot 2^j T_s) \right) \cdot \prod_{j=a}^{l} \left(\tilde{h}_j(n_j \cdot 2^j T_s) \right) \right\} \\ \cdot h_s[(2^l(\beta_l - \xi) - \sum_j 2^j n_j) \cdot T_s - iT_b - \gamma_q] \\ \cdot g[(\xi + \mu_l)T_c] \cdot e^{j\pi(k-q) \cdot \sum_j n_j} \end{array} \right\} \\
& + \sum_{\xi=I_1}^{I_2} (-1)^{(\beta_l - \xi)} \cdot \sum_{n_0} \cdots \sum_{n_l} \left\{ \begin{array}{l} \left\{ \sum_{a=1}^{l} \binom{l+1}{a} \prod_{j=0}^{a-1} \left(h(n_j \cdot 2^j T_s) \right) \cdot \prod_{j=a}^{l} \left(\tilde{h}_j(n_j \cdot 2^j T_s) \right) \right\} \\ \cdot z[(2^l(\beta_l - \xi) - \sum_j 2^j n_j) \cdot T_s] \cdot g[(\xi + \mu_l)T_c] \cdot e^{j\pi(k + \frac{1}{2}) \cdot \sum_j n_j} \end{array} \right\}
\end{aligned}
\qquad \text{(A4-13)}
$$

Once we define (4-14) through (4-17), including the case for the highpass outputs (*k odd*), the in-phase components of (A4-12) and (A4-13) become (4-11) and (4-12), respectively. Please note that from (4-11) and (4-12) the quantities $\tilde{u}(\alpha T_b - iT_b)$ and $u'(\alpha T_b - iT_b)$ have been combined to represent (4-14) and the quantities $\tilde{v}(\alpha T_b - iT_b)$ and $v'(\alpha T_b - iT_b)$ have been combined to represent (4-15) and (4-16) in their in-phase and quadrature components, respectively.

A4.1.3 : Derivation of the Probability of Bit Error (Derivation of (4-24))

To derive the probability of bit error expression of (4-24), we follow the same steps as was done to derive (3-20). The difference in this derivation is the presence of the quantization noise variable, η_q . We note for the derivation that the variable η_q is random with respect to the carrier phase ϕ_Q , the timing phases γ_Q and γ_k , the quantization noise $\tilde{h}(n_j \cdot 2^j T_s)$ and the Gaussian noise, $z(\cdot)$ [*]. We also note that since the distribution for η and η_q is approximately Gaussian (see Appendix 4.1.4 for details), the p.d.f.'s $f_\eta(\eta)$ and $f_{\eta q}(\eta_q)$ are even functions. Therefore, both η and η_q may be positive or negative quantities with a probability of $\frac{1}{2}$ [31]. Throughout the derivation, we must assume a fixed value for η and η_q . Then to obtain the unconditional probability of bit error, we must average over the joint p.d.f. between the interference η and the quantization noise η_q , $f(\eta, \eta_q)$ [31]. If we assume that we have a fixed timing offset γ_k , the unconditional probability of bit error will become the expression that is shown by (4-24).

A4.1.4 : Explanation for η_q and η being approximately Gaussian Distributed

First of all, for η, we see easily from the form of (3-18) that it is a sum of the ISI and ICI components therefore according to the Central limit Theorem, we see how η can be approximately Gaussian distributed [31]. It is more difficult to see how η_q can be Gaussian distributed since there are products between the uniformly distributed random variables, $\hat{h}_j(n_j \cdot 2^J T_s)$, representing the quantization noise from each halfband filter. To show that η_q is approximately Gaussian distributed, we must show that $\tilde{r}_k'^I(lT_b)$ and $r_k'^I(lT_b)$ are approximately Gaussian. This may be done by looking into the random components of each of (4-11) and (4-12), respectively. Assuming that the random variables in the set, $\hat{h}_j(n_j \cdot 2^J T_s)\,;\forall j$, are statistically independent and identically distributed (i.i.d), we can solve for the p.d.f's of the ISI and ICI of the quantization noise, represented by (4-14) through (4-16), respectively [31]. These p.d.f's may be obtained by solving the following transformation :

$$Y = X + X^2 + X^3 + \cdots\cdots + X^l + X^{l+1} \qquad \text{(A4-14)}$$

, with $X \sim Uniform\left(0, \frac{\Delta^2}{12}\right)$. First, for the transformation $Y' = X^i$, we get [34]:

$$f_{y'}(y') = \frac{\Delta}{2i} \cdot y'^{(1/i)-1} \quad ; \quad \left(\frac{-1}{\Delta}\right)^i \le y' \le \left(\frac{1}{\Delta}\right)^i \qquad \text{(A4-15)}$$

Through convolution of (A4-15) $l+1$ times according to the form of (A4-14), we obtain for the p.d.f.'s of $\tilde{u}'(aT_b - iT_b), \tilde{v}'^I(aT_b - iT_b)$ and $\tilde{v}'^Q(aT_b - iT_b)$ [31] [34]:

[*] Please note that when the context of discussion involves quantization, we use q to represent the quantization variable. Thus, so as not to confuse our choice of variables from previous chapters, the channel representation is given as Q when not discussing the k^{th} channel. Also, quadrature components are shown as $\boldsymbol{Q}$.

$$f_y(y) = \frac{2^{(l+1)B}}{(l+1)!} \cdot y^{\sum_{i=1}^{l+1} \frac{1}{i} - l} \quad ; \quad |y| \leq \sum_{i=1}^{l+1} 2 \cdot 2^{-i(B+1)} \tag{A4-16}$$

According to the Central Limit Theorem, we know that $\tilde{u}'(\alpha T_b - iT_b), \tilde{v}'^{I}(\alpha T_b - iT_b)$ and $\tilde{v}'^{Q}(\alpha T_b - iT_b)$ are approximately Gaussian distributed with zero mean, judging from the form of (4-14) through (4-16), respectively [31]. This fact is also supported by the form of (A4-16). As for the distribution of $\tilde{z}_k'^{I}(lT_b)$ from (4-17), we see that there is a product between a stationary Gaussian random variable and the sum of quantization random variables. This sum has already been determined to be approximately Gaussian with zero mean from the distributions of (4-14) through (4-16). According to the Central Limit Theorem for products, the product between the two zero mean Gaussian random variables is approximately lognormally distributed, assuming the Gaussian noise and the quantization noise are statistically independent [31]. In both cases for $\tilde{r}_k^{I}(lT_b)$ and $r_k'^{I}(lT_b)$, represented by (4-11) and (4-12), respectively, we must try to determine their approximate distributions given that we have a Gaussian random variable summed with a lognormal random variable. Although difficult to obtain, we can use the Central Limit Theorem to say their distributions are approximately Gaussian with non-zero means [31].

Since we have shown that (4-11) and (4-12) can be approximately Gaussian distributed, we can say that η_q is also approximately Gaussian with non-zero mean since it is the sum of both $\tilde{r}_k^{I}(lT_b)$ and $r_k'^{I}(lT_b)$ [31].

A4.2 : Convergence of the Gram-Charlier Series (Proof of (4-38))

If we expand the exponential terms in both expectations of (4-38) through a Taylor Series expansion, the conditions set in (4-38) may be simplified by considering [20] [34] :

$$E\left\{e^{\frac{\eta^2}{4}}\right\} = E\left\{\sum_{i=0}^{\infty} \frac{\eta^{2i}}{4^i \cdot i!}\right\} = \sum_{i=0}^{\infty} \frac{1}{4^i \cdot i!} \cdot E\left\{\eta^{2i}\right\} \tag{A4-17}$$

$$E\left\{e^{\frac{\eta_q^2}{4}}\right\} = E\left\{\sum_{i=0}^{\infty} \frac{\eta_q^{2i}}{4^i \cdot i!}\right\} = \sum_{i=0}^{\infty} \frac{1}{4^i \cdot i!} \cdot E\left\{\eta_q^{2i}\right\} \tag{A4-18}$$

so that the conditions for the convergence of the Gram-Charlier Series representations of $f_{\eta}(\eta)$ and $f_{\eta q}(\eta_q)$ become [20]:

$$\sum_{i=0}^{\infty} \frac{1}{4^i \cdot i!} \cdot E\left\{\eta^{2i}\right\} < \infty \tag{A4-19}$$

and :

$$\sum_{i=0}^{\infty} \frac{1}{4^i \cdot i!} \cdot E\left\{\eta_q^{2i}\right\} < \infty \tag{A4-20}$$

, respectively.

Since the range of values for both variables includes sufficiently small quantities (ie. $\xi_{\eta,MAX}$ and $\xi_{\eta_q,MAX} \ll 1$), it is sufficient to show [34]:

$$\lim_{i \to \infty} E\left\{\eta^{2i}\right\} \to 0 \tag{A4-21}$$

$$\lim_{i \to \infty} E\left\{\eta_q^{2i}\right\} \to 0 \tag{A4-22}$$

for (A4-19) and (A4-20) to hold, respectively.

To show that (A4-21) is true for the case of the interference variable, η, we will refer to our actual numerical observations. We note from our observations that the filter coefficients of $h(\cdot)$, $h_s(\cdot)$ and $g(\cdot)$ are very small in value. They are sufficiently small to the point where we can say, for the ISI and ICI coefficients :

$$\left|u(\alpha T_b - iT_b)\right| \ll 1 \quad , \quad \left|v(\alpha T_b - iT_b)\right| \ll 1 \tag{A4-23}$$

, respectively.

Therefore, from the interference moment expressions of (3-40) and (3-41), we have higher powers of $u(\alpha T_b - iT_b)$ and $v(\alpha T_b - iT_b)$, respectively, which will become exponentially small as $\rho \to \infty$, where ρ is the order of the moment associated with (3-40) and (3-41). Thus, for (3-40) and (3-41), this implies :

$$\lim_{\rho \to \infty} E\left\{\eta_{1,j}^{\rho}\right\} \to 0 \tag{A4-24}$$

and :

$$\lim_{\rho \to \infty} E\left\{\eta_{2,j}^{\rho}\right\} \to 0 \tag{A4-25}$$

, respectively. Thus, from (A4-24) and (A4-25), we see that (A4-21) is true.

To show that (A4-22) is true for the case of the quantization noise variable, η_q, we must consider the quantization error component which does not exist in η. We must show that the quantization noise moments of (4-43), (4-44) and (4-52) become exponentially small as $\rho \rightarrow \infty$, where ρ is the order of the moment associated with (4-43), (4-44) and (4-52). To do so, we will consider the quantization error component for the lowpass outputs from the final stage (*k=0* and *k even*) inherent in each of these three components of the quantization noise moments. The same can be said for the highpass outputs from the final stage (*k odd*). We can say that (A4-22) holds iff :

$$\lim_{\rho\rightarrow\infty}\left\{\int_{-2^{-B-1}}^{2^{-B-1}}\cdots\int_{-2^{-B-1}}^{2^{-B-1}}\left\{\sum_{n_0}\cdots\sum_{n_l}\left(\tilde{h}_j^{kp}(n_j\cdot 2^J T_s)\right)^2\right\}^{\rho/2} d\tilde{h}\cdots d\tilde{h}\right\}\rightarrow 0 \qquad \text{(A4-26)}$$

where we are considering only the even order moments of the quantization noise. For $B\geq 1$, we see that the range of values for $\tilde{h}_j(n_j\cdot 2^J T_s)$ with j fixed is sufficiently small (ie. $\left|\tilde{h}_j(n_j\cdot 2^J T_s)\right|\leq 2^{-B-1}\ll 1$). Thus, higher powers of $\tilde{h}_j(n_j\cdot 2^J T_s)$ will become exponentially small. Since the finite sum of these values is also exponentially small, (A4-26) must hold. Therefore, (A4-22) is true.

Having shown (A4-21) and (A4-22) are true, it thus follows that (A4-19) and (A4-20) are true, respectively, and the expression presented in (4-38) is satisfied [20] [34]. This subsequently implies that the Gram-Charlier Series representations for $f_\eta(\eta)$ and $f_{\eta q}(\eta_q)$ converge numerically [20].

A4.3 : Expansion of the Upper Bound on Q(·) (Derivation of (4-40))

From (4-39), we see that the Q-function may be upper bounded by an exponential function. We can expand this exponential function straightforwardly as [23] :

$$\exp\left\{-\frac{1}{2}\cdot\left(\frac{A_k u(0)-\eta-\eta_q}{\sigma_o}\right)^2\right\}=\exp\left\{-\frac{1}{2}\cdot\left[\begin{array}{l}\left(\frac{\frac{A_k}{2}u(0)-\eta}{\sigma_o}\right)^2+\left(\frac{\frac{A_k}{2}u(0)-\eta_q}{\sigma_o}\right)^2\\ +2\cdot\left(\frac{\frac{A_k}{2}u(0)-\eta}{\sigma_o}\right)\cdot\left(\frac{\frac{A_k}{2}u(0)-\eta_q}{\sigma_o}\right)\end{array}\right]\right\}$$

$$=\left\{\begin{array}{l}\exp\left\{-\frac{1}{2}\cdot\left(\frac{\frac{A_k}{2}u(0)-\eta}{\sigma_o}\right)^2\right\}\cdot\exp\left\{-\frac{1}{2}\cdot\left(\frac{\frac{A_k}{2}u(0)-\eta_q}{\sigma_o}\right)^2\right\}\\ \cdot\exp\left\{-\left(\frac{\frac{A_k}{2}u(0)-\eta}{\sigma_o}\right)\cdot\left(\frac{\frac{A_k}{2}u(0)-\eta_q}{\sigma_o}\right)\right\}\end{array}\right\} \qquad \text{(A4-27)}$$

From the first term of (A4-27), we have a function of η. Likewise, the middle term is a function of η_q. However, the third term is a joint function of both η and η_q and is not separable by straightforward means. We must expand this joint function in the form of a Taylor Series. Its series expansion yields [34] :

$$\exp\left\{-\left(\frac{\frac{A_k}{2}u(0)-\eta}{\sigma_o}\right)\cdot\left(\frac{\frac{A_k}{2}u(0)-\eta_q}{\sigma_o}\right)\right\}=\sum_{m=0}^{\infty}\frac{(-1)^m}{m!}\cdot\left(\frac{\frac{A_k}{2}u(0)-\eta}{\sigma_o}\right)^m\cdot\left(\frac{\frac{A_k}{2}u(0)-\eta_q}{\sigma_o}\right)^m \quad \text{(A4-28)}$$

Once (4-39), (A4-27) and (A4-28) are combined, we see very easily that the expression we have is identical to (4-40).

A4.4 : Derivation of the Moments of the Quantization Noise (Derivation of (4-43), (4-44), (4-52))

For this derivation, only the lowpass outputs from the final stage (*k=0* and *k even*) will be considered since the same derivation can be done for the highpass outputs from the final stage (*k odd*). So, from (4-42), we have the truncated quantization noise variable. If we look at (4-42) on a symbol by symbol basis, that is for fixed *i* , we have :

$$\eta_{q,i}^{\bullet}=\eta_{q,1,i}+\eta_{q,2,Q,i}+\tilde{z}_k'^{I}(lT_b)=\eta_q^{\bullet}\Big|_{i\ fixed} \quad \text{(A4-29)}$$

Now, we shall proceed to derive the ISI moment of the quantization noise, specified by (4-43). The ν^{th} moment of the ISI for fixed interfering symbol comes from (4-42). It is expressed as :

$$E[\eta_{q,1,i}^{\nu}]=\tfrac{1}{2}\cdot A_k^{\nu}\cdot 2^{\Psi+1}\sum_{a_{k,i}}\int_{\substack{-2^{-B-1}\\ \tilde{h}}}^{2^{-B-1}}\cdots\int_{\substack{-2^{-B-1}\\ \tilde{h}}}^{2^{-B-1}}\int_{\substack{-T/2\\ \gamma_k}}^{T/2}\left(a_{k,i}\cdot\tilde{u}'(\alpha T_b-iT_b)\right)^{\nu}f(\gamma_k)\cdot d\gamma_k\cdot d\tilde{h}\cdots d\tilde{h} \quad \text{(A4-30)}$$

In the analysis, it is assumed that the input quantization noise, timing phase, carrier phase and Gaussian noise are independent. Using the same argument as in section 3.2.2 and knowing that the input quantization noise is evenly distributed, the odd order moments of $\eta_{q,1,i}$ will be zero [31]. We can thus consider the even order moments only. (A4-30) may be simplified to become :

$$E[\eta_{q,1,i}^{\nu}]=A_k^{\nu}\cdot 2^{\Psi}\sum_{a_{k,i}}a_{k,i}^{\nu}\cdot\int_{\substack{-2^{-B-1}\\ \tilde{h}}}^{2^{-B-1}}\cdots\int_{\substack{-2^{-B-1}\\ \tilde{h}}}^{2^{-B-1}}\int_{\substack{-T/2\\ \gamma_k}}^{T/2}\left(\tilde{u}'(\alpha T_b-iT_b)\right)^{\nu}f(\gamma_k)\cdot d\gamma_k\cdot d\tilde{h}\cdots d\tilde{h} \quad \text{(A4-31)}$$

The integral expression of (A4-31) is a joint function of timing phase and quantization noise. In order to make the expression of (A4-31) computable, we must separate these variables. Schwartz's Inequality may be used to separate these variables [34]. First of all, we substitute (4-14) into the integral expression of (A4-31) to give :

$$\int_{\substack{-2^{-B-1}\\ \tilde{h}}}^{2^{-B-1}}\cdots\int_{\substack{-2^{-B-1}\\ \tilde{h}}}^{2^{-B-1}}\int_{\substack{-T/2\\ \gamma_k}}^{T/2}\left(\sum_{\xi=I_1}^{I_2}\sum_{n_0}\cdots\sum_{n_l}\begin{Bmatrix}h_s[(2^l(\beta_l-\xi)-\sum_j 2^j n_j)\cdot T_s-iT_b-\gamma_k]\\ \cdot g[(\xi+\mu_l)T_c]\cdot \hbar_j^{lp}(n_j\cdot 2^j T_s)\end{Bmatrix}\right)^v \begin{matrix}f(\gamma_k)\cdot d\gamma_k\\ \cdot d\tilde{h}\cdots d\tilde{h}\end{matrix} \qquad \text{(A4-32)}$$

According to Schwartz's Inequality, we can separate two variables using the following upper bound [34] :

$$\sum(a\cdot b)\leq\sqrt{\sum a^2}\cdot\sqrt{\sum b^2} \qquad \text{(A4-33)}$$

Thus, applying (A4-33) to (A4-32) gives us the following upper bound for (A4-32) :

$$\int_{\substack{-2^{-B-1}\\ \tilde{h}}}^{2^{-B-1}}\cdots\int_{\substack{-2^{-B-1}\\ \tilde{h}}}^{2^{-B-1}}\int_{\substack{-T/2\\ \gamma_k}}^{T/2}\left(\begin{matrix}\sum_{\xi=I_1}^{I_2}\sum_{n_0}\cdots\cdots\sum_{n_l}\begin{Bmatrix}h_s^2[(2^l(\beta_l-\xi)-\sum_j 2^j n_j)\cdot T_s-iT_b-\gamma_k]\\ \cdot g^2[(\xi+\mu_l)T_c]\end{Bmatrix}\\ \cdot\sum_{n_0}\cdots\cdots\sum_{n_l}\left(\hbar_j^{lp}(n_j\cdot 2^j T_s)\right)^2\end{matrix}\right)^{v/2} \begin{matrix}f(\gamma_k)\cdot d\gamma_k\\ \cdot d\tilde{h}\cdots d\tilde{h}\end{matrix} \qquad \text{(A4-34)}$$

We can now separate the variables in (A4-34) to give us two distinct parts to the equation; one for the timing phase and one for the quantization error. Thus :

$$\begin{aligned}&\int_{-T/2}^{T/2}\left(\sum_{\xi=I_1}^{I_2}\sum_{n_0}\cdots\cdots\sum_{n_l}\begin{Bmatrix}h_s^2[(2^l(\beta_l-\xi)-\sum_j 2^j n_j)\cdot T_s-iT_b-\gamma_k]\\ \cdot g^2[(\xi+\mu_l)T_c]\end{Bmatrix}\right)^{v/2} f(\gamma_k)\cdot d\gamma_k\\ &\cdot\int_{-2^{-B-1}}^{2^{-B-1}}\cdots\int_{-2^{-B-1}}^{2^{-B-1}}\left\{\sum_{n_0}\cdots\cdots\sum_{n_l}\left(\hbar_j^{lp}(n_j\cdot 2^j T_s)\right)^2\right\}^{v/2}\cdot d\tilde{h}\cdots d\tilde{h}\end{aligned} \qquad \text{(A4-35)}$$

If we substitute (A4-35) into (A4-31), we now have an upper bound for the ISI moment of the quantization noise which is exactly (4-43). As for the moments of the ICI of the quantization noise, the same idea is used. As was the case for $\eta_{q,1,i}$, since the input quantization noise is evenly distributed, the odd order moments of $\eta_{q,2,Q,i}$ will also be zero. We can thus consider the even order moments only [31]. To determine the ICI moments of the quantization noise, we refer to the ICI portion of (4-42) for the Q^{th} channel ($Q\neq k$). It can thus be expressed as :

$$E[\eta^{\nu}_{q,2,Q,J}]=\frac{1}{2\pi}\cdot 2^{\Psi-1}\cdot A_Q^{\nu}\cdot\sum_{\substack{a_{Q,J}\\ b_{Q,J}}}\int_{\substack{0\\ \phi_Q}}^{2\pi}\int_{\substack{-T/2\\ \gamma_Q}}^{T/2}\int_{\substack{-2^{-B-1}\\ \tilde{h}}}^{2^{-B-1}}\cdots\int_{\substack{-2^{-B-1}\\ \tilde{h}}}^{2^{-B-1}}\left\{\begin{array}{l} a_{Q,J}\cdot\begin{bmatrix}\tilde{v}'^{I}(\alpha T_b-iT_b)\cdot\cos\phi_Q\\ -\tilde{v}'^{Q}(\alpha T_b-iT_b)\cdot\sin\phi_Q\end{bmatrix}\\ -b_{Q,J}\cdot\begin{bmatrix}\tilde{v}'^{I}(\alpha T_b-iT_b)\cdot\sin\phi_Q\\ +\tilde{v}'^{Q}(\alpha T_b-iT_b)\cdot\cos\phi_Q\end{bmatrix}\end{array}\right\}^{\nu}\begin{array}{l}\cdot f(\gamma_Q)\cdot d\gamma_Q\\ \cdot d\phi_Q\cdot d\tilde{h}\cdots d\tilde{h}\end{array}\quad\text{(A4-36)}$$

For the sake of simplicity of the analysis, we will assume that the carrier phase is a constant value. We will set its non-random value to $\frac{1}{4}\cdot\pi$ to maximize the upper bound [34]. In this case, (A4-36) simplifies to become :

$$E[\eta^{\nu}_{q,2,Q,J}]=2^{\Psi-1}\cdot\left(\frac{1}{\sqrt{2}}\right)^{\nu}\cdot A_Q^{\nu}\cdot\sum_{\substack{a_{Q,J}\\ b_{Q,J}}}\int_{\substack{-T/2\\ \gamma_Q}}^{T/2}\int_{\substack{-2^{-B-1}\\ \tilde{h}}}^{2^{-B-1}}\cdots\int_{\substack{-2^{-B-1}\\ \tilde{h}}}^{2^{-B-1}}\left\{\begin{array}{l} a_{Q,J}\cdot\begin{bmatrix}\tilde{v}'^{I}(\alpha T_b-iT_b)\\ -\tilde{v}'^{Q}(\alpha T_b-iT_b)\end{bmatrix}\\ -b_{Q,J}\cdot\begin{bmatrix}\tilde{v}'^{I}(\alpha T_b-iT_b)\\ +\tilde{v}'^{Q}(\alpha T_b-iT_b)\end{bmatrix}\end{array}\right\}^{\nu}\begin{array}{l}\cdot f(\gamma_Q)\cdot d\gamma_Q\\ \cdot d\tilde{h}\cdots d\tilde{h}\end{array}\quad\text{(A4-37)}$$

Now, similar to the ISI moment derivation, we must try to express the integral expression of (A4-37) in two separate entities : one for the timing phase and the other for the quantization error. The first step is to substitute (4-15) and (4-16) into the integral expression of (A4-37). Since (4-15) and (4-16) are the in-phase and quadrature components of the ICI of the quantization error, respectively, they only differ in the complex components contained by $\tilde{v}'(\alpha T_b - iT_b)$, that is the cos(·) and sin(·) terms. The real components are common between (4-15) and (4-16). We can use this fact to simplify the expression to become :

$$\int_{\substack{-T/2\\ \gamma_Q}}^{T/2}\int_{\substack{-2^{-B-1}\\ \tilde{h}}}^{2^{-B-1}}\cdots\int_{\substack{-2^{-B-1}\\ \tilde{h}}}^{2^{-B-1}}\left\{\begin{array}{l} a_{Q,J}\cdot\left(\begin{array}{c}\displaystyle\sum_{\xi=-I_1}^{I_1}\sum_{n_1}\cdots\sum_{n_J}\left\{\begin{array}{l}h_s[(2^{l}(\beta_l-\xi)-\sum_j 2^{j}n_j)\cdot T_s-iT_b-\gamma_Q]\\ \cdot g[(\xi+\mu_l)T_c]\cdot\hbar_j^{\psi}(n_j\cdot 2^{j}T_s)\end{array}\right\}\\ \cdot\begin{pmatrix}\cos(\pi(k-Q)\cdot\sum_j n_j)\\ -\sin(\pi(k-Q)\cdot\sum_j n_j)\end{pmatrix}\end{array}\right)\\ -b_{Q,J}\cdot\left(\begin{array}{c}\displaystyle\sum_{\xi=-I_1}^{I_1}\sum_{n_1}\cdots\sum_{n_J}\left\{\begin{array}{l}h_s[(2^{l}(\beta_l-\xi)-\sum_j 2^{j}n_j)\cdot T_s-iT_b-\gamma_Q]\\ \cdot g[(\xi+\mu_l)T_c]\cdot\hbar_j^{\psi}(n_j\cdot 2^{j}T_s)\end{array}\right\}\\ \cdot\begin{pmatrix}\cos(\pi(k-Q)\cdot\sum_j n_j)\\ +\sin(\pi(k-Q)\cdot\sum_j n_j)\end{pmatrix}\end{array}\right)\end{array}\right\}^{\nu}\begin{array}{l}\cdot f(\gamma_Q)\cdot d\gamma_Q\\ \cdot d\tilde{h}\cdots d\tilde{h}\end{array}\quad\text{(A4-38)}$$

From (A4-38), we can simplify by grouping under the common summation, thus :

$$\int_{-T/2}^{T/2}\int_{-2^{-B-1}}^{2^{-B-1}}\cdots\int_{-2^{-B-1}}^{2^{-B-1}}\left\{\begin{array}{l}\left[\displaystyle\sum_{\xi=I_1}^{I_2}\sum_{n_0}\cdots\cdots\sum_{n_l}\left\{\begin{array}{l}h_s[(2^l(\beta_l-\xi)-\sum_j 2^j n_j)\cdot T_s-iT_b-\gamma_Q]\\ \cdot g[(\xi+\mu_l)T_c]\cdot\hbar_j^{b}(n_j\cdot 2^j T_s)\end{array}\right\}\right]^{v}\\ \cdot\left(\begin{array}{l}a_{Q,l}\cdot\left(\begin{array}{l}\cos(\pi(k-Q)\cdot\sum_j n_j)\\ -\sin(\pi(k-Q)\cdot\sum_j n_j)\end{array}\right)\\ -b_{Q,l}\cdot\left(\begin{array}{l}\cos(\pi(k-Q)\cdot\sum_j n_j)\\ +\sin(\pi(k-Q)\cdot\sum_j n_j)\end{array}\right)\end{array}\right)\end{array}\right\}\begin{array}{l}\cdot f(\gamma_Q)\cdot d\gamma_Q\\ \cdot d\tilde{h}\cdots d\tilde{h}\end{array}\tag{A4-39}$$

Now, as the previous case, we use Schwartz's Inequality, expressed in (A4-33), to separate the variables. (A4-39) thus becomes upper bounded by [34] :

$$\int_{-T/2}^{T/2}\int_{-2^{-B-1}}^{2^{-B-1}}\cdots\int_{-2^{-B-1}}^{2^{-B-1}}\left\{\left[\begin{array}{l}\left(\begin{array}{l}\displaystyle\sum_{\xi=I_1}^{I_2}\sum_{n_0}\cdots\cdots\sum_{n_l}\left(\begin{array}{l}h_s^2[(2^l(\beta_l-\xi)-\sum_j 2^j n_j)\cdot T_s-iT_b-\gamma_Q]\\ \cdot g^2[(\xi+\mu_l)T_c]\end{array}\right)\\ \cdot\left(\begin{array}{l}a_{Q,l}\cdot\left(\begin{array}{l}\cos(\pi(k-Q)\cdot\sum_j n_j)\\ -\sin(\pi(k-Q)\cdot\sum_j n_j)\end{array}\right)\\ -b_{Q,l}\cdot\left(\begin{array}{l}\cos(\pi(k-Q)\cdot\sum_j n_j)\\ +\sin(\pi(k-Q)\cdot\sum_j n_j)\end{array}\right)\end{array}\right)^2\end{array}\right)\\ \cdot\displaystyle\sum_{n_0}\cdots\cdots\sum_{n_l}\left(\cdot\hbar_j^{b}(n_j\cdot 2^j T_s)\right)^2\end{array}\right]^{v/2}\right\}\begin{array}{l}\cdot f(\gamma_Q)\cdot d\gamma_Q\\ \cdot d\tilde{h}\cdots d\tilde{h}\end{array}\tag{A4-40}$$

We shall look at the portion of (A4-40) containing the cos(·) and sin(·) terms in order to simplify it. If we expand the square of this binomial, we get :

$$\left(\begin{array}{l}a_{Q,l}\cdot\left(\begin{array}{l}\cos(\pi(k-Q)\cdot\sum_j n_j)\\ -\sin(\pi(k-Q)\cdot\sum_j n_j)\end{array}\right)\\ -b_{Q,l}\cdot\left(\begin{array}{l}\cos(\pi(k-Q)\cdot\sum_j n_j)\\ +\sin(\pi(k-Q)\cdot\sum_j n_j)\end{array}\right)\end{array}\right)^2=\left(\begin{array}{l}a_{Q,l}^2\cdot\left(\begin{array}{l}\cos(\pi(k-Q)\cdot\sum_j n_j)\\ -\sin(\pi(k-Q)\cdot\sum_j n_j)\end{array}\right)^2\\ +b_{Q,l}^2\cdot\left(\begin{array}{l}\cos(\pi(k-Q)\cdot\sum_j n_j)\\ +\sin(\pi(k-Q)\cdot\sum_j n_j)\end{array}\right)^2\\ -2\cdot a_{Q,l}\cdot b_{Q,l}\cdot\left(\begin{array}{l}\cos(\pi(k-Q)\cdot\sum_j n_j)\\ -\sin(\pi(k-Q)\cdot\sum_j n_j)\end{array}\right)\\ \qquad\cdot\left(\begin{array}{l}\cos(\pi(k-Q)\cdot\sum_j n_j)\\ +\sin(\pi(k-Q)\cdot\sum_j n_j)\end{array}\right)\end{array}\right)\tag{A4-41}$$

We shall expand each of the terms of (A4-41) as was done for (A4-40). The expansion term by term yields for the right-hand side of (A4-41) :

$$\left(\begin{array}{l}
a_{Q,J}^2 \cdot \left(\begin{array}{l} \cos(\pi(k-Q)\cdot\sum\limits_J n_J) \\ -\sin(\pi(k-Q)\cdot\sum\limits_J n_J) \end{array}\right)^2 \\
+b_{Q,J}^2 \cdot \left(\begin{array}{l} \cos(\pi(k-Q)\cdot\sum\limits_J n_J) \\ +\sin(\pi(k-Q)\cdot\sum\limits_J n_J) \end{array}\right)^2 \\
-2\cdot a_{Q,J}\cdot b_{Q,J}\cdot \left(\begin{array}{l} \cos(\pi(k-Q)\cdot\sum\limits_J n_J) \\ -\sin(\pi(k-Q)\cdot\sum\limits_J n_J) \end{array}\right) \\
\qquad\qquad\cdot \left(\begin{array}{l} \cos(\pi(k-Q)\cdot\sum\limits_J n_J) \\ +\sin(\pi(k-Q)\cdot\sum\limits_J n_J) \end{array}\right)
\end{array}\right)
=
\begin{array}{l}
a_{Q,J}^2 \cdot \left(\begin{array}{l} \cos^2(\pi(k-Q)\cdot\sum\limits_J n_J) \\ +\sin^2(\pi(k-Q)\cdot\sum\limits_J n_J) \\ -2\cdot\left(\begin{array}{l} \cos(\pi(k-Q)\cdot\sum\limits_J n_J) \\ \cdot\sin(\pi(k-Q)\cdot\sum\limits_J n_J) \end{array}\right) \end{array}\right) \\
+b_{Q,J}^2 \cdot \left(\begin{array}{l} \cos^2(\pi(k-Q)\cdot\sum\limits_J n_J) \\ +\sin^2(\pi(k-Q)\cdot\sum\limits_J n_J) \\ +2\cdot\left(\begin{array}{l} \cos(\pi(k-Q)\cdot\sum\limits_J n_J) \\ \cdot\sin(\pi(k-Q)\cdot\sum\limits_J n_J) \end{array}\right) \end{array}\right) \\
-2\cdot a_{Q,J}\cdot b_{Q,J}\cdot \left(\begin{array}{l} \cos^2(\pi(k-Q)\cdot\sum\limits_J n_J) \\ -\sin^2(\pi(k-Q)\cdot\sum\limits_J n_J) \end{array}\right)
\end{array}
\tag{A4-42}$$

Using trigonometric identities, we can simplify (A4-42) further [34] :

$$\begin{array}{l}
a_{Q,J}^2 \cdot \left(\begin{array}{l} \cos^2(\pi(k-Q)\cdot\sum\limits_J n_J) \\ +\sin^2(\pi(k-Q)\cdot\sum\limits_J n_J) \\ -2\cdot\left(\begin{array}{l} \cos(\pi(k-Q)\cdot\sum\limits_J n_J) \\ \cdot\sin(\pi(k-Q)\cdot\sum\limits_J n_J) \end{array}\right) \end{array}\right) \\
+b_{Q,J}^2 \cdot \left(\begin{array}{l} \cos^2(\pi(k-Q)\cdot\sum\limits_J n_J) \\ +\sin^2(\pi(k-Q)\cdot\sum\limits_J n_J) \\ +2\cdot\left(\begin{array}{l} \cos(\pi(k-Q)\cdot\sum\limits_J n_J) \\ \cdot\sin(\pi(k-Q)\cdot\sum\limits_J n_J) \end{array}\right) \end{array}\right) \\
-2\cdot a_{Q,J}\cdot b_{Q,J}\cdot \left(\begin{array}{l} \cos^2(\pi(k-Q)\cdot\sum\limits_J n_J) \\ -\sin^2(\pi(k-Q)\cdot\sum\limits_J n_J) \end{array}\right)
\end{array}
=
\left(\begin{array}{l}
a_{Q,J}^2 + b_{Q,J}^2 \\
-2\cdot\left(a_{Q,J}^2 - b_{Q,J}^2\right)\cdot\left(\begin{array}{l} \cos(\pi(k-Q)\cdot\sum\limits_J n_J) \\ \cdot\sin(\pi(k-Q)\cdot\sum\limits_J n_J) \end{array}\right) \\
-2\cdot a_{Q,J}\cdot b_{Q,J}\cdot \left(\begin{array}{l} \cos^2(\pi(k-Q)\cdot\sum\limits_J n_J) \\ -\sin^2(\pi(k-Q)\cdot\sum\limits_J n_J) \end{array}\right)
\end{array}\right)
\tag{A4-43}$$

But we know that both $a_{Q,i}$ and $b_{Q,i}$ are chosen from the binary set *{±1}* with equal probability thus the right-hand side of (A4-43) is simplified to its final form :

$$\left(\begin{array}{l} a_{Q,i}^2+b_{Q,i}^2 \\ -2\cdot\left(a_{Q,i}^2-b_{Q,i}^2\right)\cdot\left(\begin{array}{l}\cos(\pi(k-Q)\cdot\sum_j n_j) \\ \cdot\sin(\pi(k-Q)\cdot\sum_j n_j)\end{array}\right) \\ -2\cdot a_{Q,i}\cdot b_{Q,i}\cdot\left(\begin{array}{l}\cos^2(\pi(k-Q)\cdot\sum_j n_j) \\ -\sin^2(\pi(k-Q)\cdot\sum_j n_j)\end{array}\right)\end{array}\right)=2-2\cdot a_{Q,i}\cdot b_{Q,i}\cdot\left(\begin{array}{l}\cos^2(\pi(k-Q)\cdot\sum_j n_j) \\ -\sin^2(\pi(k-Q)\cdot\sum_j n_j)\end{array}\right) \tag{A4-44}$$

$$=2\cdot\left(1-a_{Q,i}\cdot b_{Q,i}\cdot\left(\begin{array}{l}\cos^2(\pi(k-Q)\cdot\sum_j n_j) \\ -\sin^2(\pi(k-Q)\cdot\sum_j n_j)\end{array}\right)\right)$$

If we substitute (A4-44) into (A4-40) and separate the integrations according to their variables, we get :

$$\left(\sqrt{2}\right)^v\cdot\int\limits_{-T/2}^{T/2}\left\{\begin{array}{l}\sum\limits_{\xi=I_1}^{I_2}\sum\limits_{n_0}\cdots\sum\limits_{n_l}\left\{\begin{array}{l}h_s^2[(2^l(\beta_l-\xi)-\sum_j 2^j n_j)\cdot T_s-iT_b-\gamma_Q] \\ \cdot g^2[(\xi+\mu_l)T_c]\end{array}\right\} \\ +a_{Q,i}\cdot b_{Q,i} \\ \cdot\left(\begin{array}{l}\sum\limits_{\xi=I_1}^{I_2}\sum\limits_{n_0}\cdots\sum\limits_{n_l}\left\{\begin{array}{l}h_s^2[(2^l(\beta_l-\xi)-\sum_j 2^j n_j)\cdot T_s-iT_b-\gamma_Q] \\ \cdot g^2[(\xi+\mu_l)T_c]\cdot\sin^2(\pi(k-Q)\cdot\sum_j n_j)\end{array}\right\} \\ -\sum\limits_{\xi=I_1}^{I_2}\sum\limits_{n_0}\cdots\sum\limits_{n_l}\left\{\begin{array}{l}h_s^2[(2^l(\beta_l-\xi)-\sum_j 2^j n_j)\cdot T_s-iT_b-\gamma_Q] \\ \cdot g^2[(\xi+\mu_l)T_c]\cdot\cos^2(\pi(k-Q)\cdot\sum_j n_j)\end{array}\right\}\end{array}\right)\end{array}\right\}^{v/2}\begin{array}{l}\cdot f(\gamma_Q) \\ \cdot d\gamma_Q\end{array} \tag{A4-45}$$

$$\cdot\int\limits_{\underset{\tilde{h}}{-2^{-B-1}}}^{2^{-B-1}}\cdots\int\limits_{\underset{\tilde{h}}{-2^{-B-1}}}^{2^{-B-1}}\left\{\sum_{n_0}\cdots\cdots\sum_{n_l}\left(\hbar_j^{b}(n_j\cdot 2^j T_s)\right)^2\right\}^{v/2}\cdot d\tilde{h}\cdots d\tilde{h}$$

Finally, substituting (A4-45) into (A4-37) will give an upper bound for the ICI moment of the quantization noise which corresponds exactly to (4-44), using one further trigonometric identity not stated in (A4-44)[♦] [34]. To derive the moments for the Gaussian noise aggregated with the quantization noise, we proceed in a manner similar to what was done previously. Thus, to find the moments of $\tilde{z}_k'^{\,l}(lT_b)$, we first use (4-17) :

$$E\left\{\left(\tilde{z}_k'^{\,l}(lT_b)\right)^{v}\right\}=E\left\{\left(\sum_{\xi=I_1}^{I_2}(-1)^{\beta_l-\xi}\sum_{n_0}\cdots\cdots\sum_{n_l}\left\{\begin{array}{l} z[(2^l(\beta_l-\xi)-\sum_j 2^j n_j)\cdot T_s] \\ \cdot g[(\xi+\mu_l)T_c]\cdot\cos\left(\pi(k+\frac{1}{2})\cdot\sum_j n_j\right) \\ \cdot\hbar_j^{lp}(n_j\cdot 2^j T_s) \end{array}\right\}\right)^{v}\right\} \quad \text{(A4-46)}$$

Again, we use Schwartz's Inequality from (A4-33) and it gives us [34] :

$$E\left\{\left(\tilde{z}_k'^{\,l}(lT_b)\right)^{v}\right\}\le E\left\{\left(\begin{array}{l}\displaystyle\sum_{\xi=I_1}^{I_2}\sum_{n_0}\cdots\cdots\sum_{n_l}\left\{\begin{array}{l} z^2[(2^l(\beta_l-\xi)-\sum_j 2^j n_j)\cdot T_s] \\ \cdot g^2[(\xi+\mu_l)T_c]\cdot\cos^2\left(\pi(k+\frac{1}{2})\cdot\sum_j n_j\right) \end{array}\right\} \\ \displaystyle\cdot\sum_{n_0}\cdots\cdots\sum_{n_l}\left(\hbar_j^{lp}(n_j\cdot 2^j T_s)\right)^2 \end{array}\right)^{v/2}\right\} \quad \text{(A4-47)}$$

Since we initially assumed that the Gaussian noise and the quantization noise were independent, we can separate the expectations in (A4-47), thus [31]:

$$\begin{array}{l} E\left\{\left(\tilde{z}_k'^{\,l}(lT_b)\right)^{v}\right\}\le E\left\{\left(\displaystyle\sum_{\xi=I_1}^{I_2}\sum_{n_0}\cdots\cdots\sum_{n_l}\left\{\begin{array}{l} z^2[(2^l(\beta_l-\xi)-\sum_j 2^j n_j)\cdot T_s] \\ \cdot g^2[(\xi+\mu_l)T_c]\cdot\cos^2\left(\pi(k+\frac{1}{2})\cdot\sum_j n_j\right) \end{array}\right\}\right)^{v/2}\right\} \\ \qquad\qquad\cdot E\left\{\left(\displaystyle\sum_{n_0}\cdots\cdots\sum_{n_l}\left(\hbar_j^{lp}(n_j\cdot 2^j T_s)\right)^2\right)^{v/2}\right\} \end{array} \quad \text{(A4-48)}$$

From (A4-48), we see that for the Gaussian noise component we have exactly $(2L+1)(l+1)(I_2-I_1+1)$ random variables. For this portion, we must average the expression over $(2L+1)(l+1)(I_2-I_1+1)$ normal distributions of zero mean and variance σ^2 since $z(t)\sim N(0,\sigma^2)$ and $z(t)$ is a stationary and white random process. In the quantization noise component, we must average the expression over $(2L+1)(l+1)$ uniform distributions with zero mean and variance $2^{-2B}/12$ since $\tilde{h}(\cdot)$ $\sim Uniform(0,2^{-2B}/12)$ and $\tilde{h}(\cdot)$ is also stationary. Since the p.d.f. of a uniform distribution is simply a constant over the range of the random variable, for the joint p.d.f. of the expression we must average over is $2^{B(2L+1)(l+1)}$, since process $\tilde{h}(\cdot)$ was assumed to be white [31].

[♦] The last trigonometric identity needed to complete the derivation is [34] :

$$\cos^2(x)-\sin^2(x)=\cos(2x)$$

Following these final steps, we obtain the upper bounded moments of the Gaussian noise aggregated with quantization noise, shown in (4-52).

A4.5 : Derivation of the Correlation Between the Quantization and Interference (Derivation of (4-53), (4-54))

Similar to the previous derivation, only the lowpass outputs from the final stage (*k=0* and *k even*) will be considered. Thus, to prove the expressions for the correlation between the interference and quantization noise, as expressed in (4-53) and (4-54), we will first look at the interference and quantization noise on a symbol by symbol basis (ie. for fixed i) [2]. In both cases, we shall refer to (3-36) and (4-42), respectively. For the interference, we may express the ISI as $\eta_{1,i}$ and the ICI as $\eta_{2,Q,i}$, as is indicated by (A3-31) through (A3-33), respectively. For the quantization noise, we can express the ISI portion by $\eta_{q,1,i}$ and the ICI portion by $\eta_{q,2,Q,i}$, thus :

$$\eta_q^{\bullet} = \eta_{q,1,i} + \eta_{q,2,Q,i} + \tilde{z}_k'^{I}(lT_b) \qquad \text{(A4-49)}$$

where :

$$\eta_{q,1,i} = A_k \cdot a_{k,i} \cdot \tilde{u}'(\alpha T_b - iT_b) \qquad \text{(A4-50)}$$

and :

$$\eta_{q,2,Q,i} = A_Q \cdot \begin{Bmatrix} a_{Q,i} \cdot [\tilde{v}'^{I}(\alpha T_b - iT_b) \cdot \cos\phi_Q - \tilde{v}'^{Q}(\alpha T_b - iT_b) \cdot \sin\phi_Q] \\ -b_{Q,i} \cdot [\tilde{v}'^{I}(\alpha T_b - iT_b) \cdot \sin\phi_Q + \tilde{v}'^{Q}(\alpha T_b - iT_b) \cdot \cos\phi_Q] \end{Bmatrix} \qquad \text{(A4-51)}$$

and $\tilde{z}_k'^{I}(lT_b)$ is given by (4-17).

First of all, to find the correlation between η' and $\eta_q^{\bullet}$, we first look at the Gaussian noise term in (A4-49). Because $\tilde{z}_k'^{I}(lT_b)$ has zero mean, it has no effect on the overall correlation thus it can be discarded. Secondly, we assume that the data symbols from one channel to the next are independently distributed (ie. $E\{a_{q,,i} \cdot a_{kj}\}=0$, $E\{b_{q,i} \cdot b_{kj}\}=0$, $E\{a_{q,i} \cdot b_{kj}\}=0$ and $E\{b_{q,i} \cdot a_{kj}\}=0$), thus the correlation is simplified to [31]:

$$E\left\{\eta'\eta_q^{\bullet}\right\} = E\left\{\eta_{1,i} \cdot \eta_{q,1,j}\right\} + E\left\{\eta_{2,Q,i} \cdot \eta_{q,2,Q,j}\right\} \qquad \text{(A4-52)}$$

In other words, correlation exists only among the ISI components and ICI components of the interference and quantization noise but not between both components. This is due to

First, we will try to derive an upper bound for (A4-58). If we substitute (4-14) into (A4-58), we get :

$$E\left\{\eta_{1,J}\cdot\eta_{q,1,J}\right\}=E\left\{\begin{matrix}A_k^2\cdot u(\alpha T_b-iT_b)\\ \cdot\sum_{\xi=I_1}^{I_2}\sum_{n_0}\cdots\cdots\sum_{n_l}\left\{\begin{matrix}h_s[(2^l(\beta_l-\xi)-\sum_J 2^J n_J)\cdot T_s-iT_b-\gamma_k]\\ \cdot g[(\xi+\mu_l)T_c]\cdot\hbar_J^{lp}(n_J\cdot 2^J T_s)\end{matrix}\right\}\end{matrix}\right\} \quad \text{(A4-60)}$$

If we use Schwartz's Inequality as expressed in (A4-33) on the summation of (A4-60), as was done previously, we obtain for an upper bound of (A4-60) [34] :

$$E\left\{\eta_{1,J}\cdot\eta_{q,1,J}\right\}\leq E\left\{\begin{matrix}A_k^2\cdot u(\alpha T_b-iT_b)\\ \cdot\sqrt{\sum_{\xi=I_1}^{I_2}\sum_{n_0}\cdots\cdots\sum_{n_l}\left\{\begin{matrix}h_s^2[(2^l(\beta_l-\xi)-\sum_J 2^J n_J)\cdot T_s-iT_b-\gamma_k]\\ \cdot g^2[(\xi+\mu_l)T_c]\end{matrix}\right\}}\\ \cdot\sqrt{\sum_{n_0}\cdots\cdots\sum_{n_l}\left(\hbar_J^{lp}(n_J\cdot 2^J T_s)\right)^2}\end{matrix}\right\} \quad \text{(A4-61)}$$

Since the timing phase and the quantization error are assumed to be statistically independent, (A4-61) may be expressed as [31]:

$$E\left\{\eta_{1,J}\cdot\eta_{q,1,J}\right\}\leq E\left\{\begin{matrix}A_k^2\cdot u(\alpha T_b-iT_b)\\ \cdot\sqrt{\sum_{\xi=I_1}^{I_2}\sum_{n_0}\cdots\cdots\sum_{n_l}\left\{\begin{matrix}h_s^2[(2^l(\beta_l-\xi)-\sum_J 2^J n_J)\cdot T_s-iT_b-\gamma_k]\\ \cdot g^2[(\xi+\mu_l)T_c]\end{matrix}\right\}}\end{matrix}\right\} \quad \text{(A4-62)}$$
$$\cdot E\left\{\sqrt{\sum_{n_0}\cdots\cdots\sum_{n_l}\left(\hbar_J^{lp}(n_J\cdot 2^J T_s)\right)^2}\right\}$$

From (A4-62), if we average the first expectation over the timing phase and the second expectation over all the quantization error variables, we will obtain (4-53) exactly. In the second expectation, we average over *(2L+1)(l+1)* uniform random variables thus the joint p.d.f. over which we must average is $2^{B(2L+1)(l+1)}$ since the error process, $\tilde{h}_J(n_J\cdot 2^J T_s)$, is white and stationary [31].

Now, we will try to derive an upper bound for (A4-59). First of all, we will expand the multiplications in (A4-59). This expansion yields :

$$E\left\{\eta_{2,Q,i}\cdot\eta_{q,2,Q,i}\right\}=E\left\{A_Q^2\cdot\left\{\begin{array}{l}\left(\begin{array}{l}v^I(\alpha T_b-iT_b)\cdot\tilde{v}'^I(\alpha T_b-iT_b)\cdot\cos^2\phi_Q\\+v^Q(\alpha T_b-iT_b)\cdot\tilde{v}'^Q(\alpha T_b-iT_b)\cdot\sin^2\phi_Q\\-\left(\begin{array}{l}v^I(\alpha T_b-iT_b)\cdot\tilde{v}'^Q(\alpha T_b-iT_b)\\+v^Q(\alpha T_b-iT_b)\cdot\tilde{v}'^I(\alpha T_b-iT_b)\end{array}\right)\cdot\sin\phi_Q\cdot\cos\phi_Q\end{array}\right)\\+\left(\begin{array}{l}v^I(\alpha T_b-iT_b)\cdot\tilde{v}'^I(\alpha T_b-iT_b)\cdot\sin^2\phi_Q\\+v^Q(\alpha T_b-iT_b)\cdot\tilde{v}'^Q(\alpha T_b-iT_b)\cdot\cos^2\phi_Q\\+\left(\begin{array}{l}v^I(\alpha T_b-iT_b)\cdot\tilde{v}'^Q(\alpha T_b-iT_b)\\+v^Q(\alpha T_b-iT_b)\cdot\tilde{v}'^I(\alpha T_b-iT_b)\end{array}\right)\cdot\sin\phi_Q\cdot\cos\phi_Q\end{array}\right)\end{array}\right\}\right\}\qquad\text{(A4-63)}$$

We can simplify (A4-63) using trigonometric identities, thus [34] :

$$E\left\{\eta_{2,Q,i}\cdot\eta_{q,2,Q,i}\right\}=E\left\{A_Q^2\cdot\left(\begin{array}{l}v^I(\alpha T_b-iT_b)\cdot\tilde{v}'^I(\alpha T_b-iT_b)\\+v^Q(\alpha T_b-iT_b)\cdot\tilde{v}'^Q(\alpha T_b-iT_b)\end{array}\right)\right\}\qquad\text{(A4-64)}$$

If we substitute (4-15) and (4-16) into (A4-64), we obtain :

$$E\left\{\eta_{2,Q,i}\cdot\eta_{q,2,Q,i}\right\}=E\left\{A_Q^2\cdot\left(\begin{array}{l}v^I(\alpha T_b-iT_b)\\\qquad\cdot\sum_{\xi=I_1}^{I_2}\sum_{n_0}\cdots\cdots\sum_{n_l}\left\{\begin{array}{l}h_s[(2^l(\beta_l-\xi)-\sum_j 2^j n_j)\cdot T_s-iT_b-\gamma_Q]\\\cdot g[(\xi+\mu_l)T_c]\cdot\cos(\pi(k-Q)\cdot\sum_j n_j)\\\cdot\hbar_j^{bp}(n_j\cdot 2^j T_s)\end{array}\right\}\\+v^Q(\alpha T_b-iT_b)\\\qquad\cdot\sum_{\xi=I_1}^{I_2}\sum_{n_0}\cdots\cdots\sum_{n_l}\left\{\begin{array}{l}h_s[(2^l(\beta_l-\xi)-\sum_j 2^j n_j)\cdot T_s-iT_b-\gamma_Q]\\\cdot g[(\xi+\mu_l)T_c]\cdot\sin(\pi(k-Q)\cdot\sum_j n_j)\\\cdot\hbar_j^{bp}(n_j\cdot 2^j T_s)\end{array}\right\}\end{array}\right)\right\}\qquad\text{(A4-65)}$$

From (A4-65), we split the expectation into two parts, each one corresponding to one term in the summation. Next, we apply Schwartz's Inequality of (A4-33) to each expectation giving us an upper bound for (A4-65) [34] :

$$E\left\{\eta_{2,Q,J}\cdot\eta_{q,2,Q,J}\right\}\le E\left\{A_Q^2\cdot\left(\begin{array}{l} v^I(\alpha T_b-iT_b) \\ \cdot\sqrt{\sum_{\xi=I_1}^{I_2}\sum_{n_0}\cdots\sum_{n_l}\left\{\begin{array}{l} h_s^2[(2^I(\beta_l-\xi)-\sum_j 2^J n_j)\cdot T_s-iT_b-\gamma_Q] \\ \cdot g^2[(\xi+\mu_l)T_c]\cdot\cos^2(\pi(k-Q)\cdot\sum_j n_j) \end{array}\right\}} \\ \cdot\sqrt{\sum_{n_0}\cdots\sum_{n_l}\left(\hbar_j^{lp}(n_j\cdot 2^J T_s)\right)^2} \end{array}\right)\right\}$$

$$+E\left\{A_Q^2\cdot\left(\begin{array}{l} v^Q(\alpha T_b-iT_b) \\ \cdot\sqrt{\sum_{\xi=I_1}^{I_2}\sum_{n_0}\cdots\sum_{n_l}\left\{\begin{array}{l} h_s^2[(2^I(\beta_l-\xi)-\sum_j 2^J n_j)\cdot T_s-iT_b-\gamma_Q] \\ \cdot g^2[(\xi+\mu_l)T_c]\cdot\sin^2(\pi(k-Q)\cdot\sum_j n_j) \end{array}\right\}} \\ \cdot\sqrt{\sum_{n_0}\cdots\sum_{n_l}\left(\hbar_j^{lp}(n_j\cdot 2^J T_s)\right)^2} \end{array}\right)\right\} \quad \text{(A4-66)}$$

From (A4-66), we can split each expectation into two parts due to the independence between the timing phase and the quantization noise [31]. Furthermore, we can take the quantization noise as a common factor between both summations and regroup to obtain only two summations multiplied by eachother. Thus, (A4-66) reduces to :

$$E\left\{\eta_{2,Q,J}\cdot\eta_{q,2,Q,J}\right\}\le E\left\{A_Q^2\cdot\left(\begin{array}{l} v^I(\alpha T_b-iT_b) \\ \quad\cdot\sqrt{\sum_{\xi=I_1}^{I_2}\sum_{n_0}\cdots\sum_{n_l}\left\{\begin{array}{l} h_s^2[(2^I(\beta_l-\xi)-\sum_j 2^J n_j)\cdot T_s-iT_b-\gamma_Q] \\ \cdot g^2[(\xi+\mu_l)T_c]\cdot\cos^2(\pi(k-Q)\cdot\sum_j n_j) \end{array}\right\}} \\ +v^Q(\alpha T_b-iT_b) \\ \quad\cdot\sqrt{\sum_{\xi=I_1}^{I_2}\sum_{n_0}\cdots\sum_{n_l}\left\{\begin{array}{l} h_s^2[(2^I(\beta_l-\xi)-\sum_j 2^J n_j)\cdot T_s-iT_b-\gamma_Q] \\ \cdot g^2[(\xi+\mu_l)T_c]\cdot\sin^2(\pi(k-Q)\cdot\sum_j n_j) \end{array}\right\}} \end{array}\right)\right\} \quad \text{(A4-67)}$$

$$\cdot E\left\{\sqrt{\sum_{n_0}\cdots\cdots\sum_{n_l}\left(\hbar_j^{lp}(n_j\cdot 2^J T_s)\right)^2}\right\}$$

If we average both expectations in (A4-67) over the timing phase and the quantization error variables, respectively, as was done previously, then the final result will be (4-54). In this case, however, we must take a finite sum of the first expectation in (A4-67) over all the interfering channels, that is, all channels except for the k^{th} . Also, we use two trigonometric identities to obtain an expression in terms of uu(·) and vv(·)[*] [34]. Once that is done, the expression will be identical to (4-54).

A4.6 : Derivation of the Maxima and Minima of the Quantization Noise (Derivation of (4-63), (4-64))

$\xi_{\eta_q,MAX}$ was needed in order to maximize the upper bounds of the truncation errors given by (4-72) and (4-74). To derive $\xi_{\eta_q,MAX}$, we must analyze the maxima of $\eta_{q,1}$, $\eta_{q,2}$ and $\widetilde{z}_k^{\prime I}(lT_b)$, as was done for the interference counterparts in Appendix 3.3. To do so, we must make use of the coefficient rounding error as explained by Liu, to upper bound the quantization error from each halfband filter [28]. If we assume we have B bits of quantization and if ξ_h is a fixed value in the range of the filter coefficient quantization error, then the rounding error can be upper bounded by [28] :

$$\left|\xi_{\widetilde{h}}\right| \leq 2^{-B-1} \tag{A4-68}$$

Thus, the maximum values for $\eta_{q,1}$ and $\eta_{q,2}$ are :

$$max\ \eta_{q,1} = \max_{\gamma_k} \sum_{i=\alpha-m_l}^{\alpha+m_u} \left|A_k \cdot \widetilde{u}'_{\max}(\alpha T_b - iT_b)\right| \tag{A4-69}$$

and :

$$max\ \eta_{q,2} = \sum_{Q\neq k} A_Q \max_{\gamma_Q, \phi_Q} \sum_{i=\alpha-m_{l,Q}}^{\alpha+m_{u,Q}} \left| \begin{array}{l} \widetilde{v}_{\max}^{\prime I}(\alpha T_b - iT_b)\cdot\left(\cos\phi_Q + \sin\phi_Q\right) \\ +\widetilde{v}_{\max}^{\prime Q}(\alpha T_b - iT_b)\cdot\left(\cos\phi_Q - \sin\phi_Q\right) \end{array} \right| \tag{A4-70}$$

, respectively. To maximize with respect to ϕ_Q in (A4-70), we can set ϕ_Q to a constant non-random value of $\frac{1}{4}\cdot\pi$, thus (A4-70) becomes :

$$max\ \eta_{q,2} = \sqrt{2}\sum_{Q\neq k} A_Q \max_{\gamma_Q} \sum_{i=\alpha-m_{l,Q}}^{\alpha+m_{u,Q}} \left|\widetilde{v}_{\max}^{\prime I}(\alpha T_b - iT_b)\right| + \left|\widetilde{v}_{\max}^{\prime Q}(\alpha T_b - iT_b)\right| \tag{A4-71}$$

, where $\widetilde{u}'_{\max}(\alpha T_b - iT_b)$, $\widetilde{v}_{\max}^{\prime I}(\alpha T_b - iT_b)$ and $\widetilde{v}_{\max}^{\prime Q}(\alpha T_b - iT_b)$ are the *maximum* values of the ISI and ICI, respectively, with respect to the halfband filter coefficient quantization error, $\widetilde{h}_j(n_j \cdot 2^j T_s)$. These quantities are formed by first observing (4-14), (4-15) and (4-16), respectively.

[*] The two trigonometric identities needed to complete the derivation are [34]:
cos²(x)=½·{1+cos(2x)} *and* sin²(x)= ½·{1-cos(2x)}

With the maximum rounding error in (A4-68), the maximum values of the ISI and ICI are as expressed in (4-65), (4-66) and (4-67), respectively [28]. For the maximum value of $\tilde{z}_k'^{I}(lT_b)$, we assume that $z(t) \sim N(0, \sigma^2)$ is bounded by [31]:

$$|\xi_z| \leq 3.291 \cdot \sigma \qquad \text{(A4-72)}$$

where ξ_z is a fixed value in the range of $z(\cdot)$. (A4-72) holds iff ξ_z remains within the 99.95% confidence interval of $z(\cdot)$ [31]. Thus, using the maximum rounding error in (A4-68), the maximum value of $\tilde{z}_k'^{I}(lT_b)$ for the lowpass outputs from the final stage (*k=0* and *k even*) is [28]:

$$max\, \tilde{z}_k'^{I}(lT_b) =$$

$$= 3.291\sigma \cdot \left(\begin{array}{l} \left|2^{-B-1}\right| \cdot \sum_{\xi=l_1}^{l_2} (-1)^{\beta_l - \xi} \cdot \sum_{n_0} \cdots \sum_{n_l} g[(\xi+\mu_l)T_c] \cdot \cos\left(\pi(k+\frac{1}{2}) \cdot \sum_j n_j\right) \\ + \sum_{a=1}^{l} \binom{l+1}{a} \cdot \sum_{\xi=l_1}^{l_2} \sum_{n_0} \cdots \sum_{n_{a-1}} \left\{ \begin{array}{l} \prod_{j=0}^{a-1} \left(h(n_j \cdot 2^j T_s)\right) \\ \left|2^{-B-1}\right| \cdot \left(\sum_{n_a} \cdots\cdots \sum_{n_l} \left\{ \begin{array}{l} g[(\xi+\mu_l)T_c] \\ \cdot \cos\left(\pi(k+\frac{1}{2}) \cdot \sum_j n_j\right) \end{array} \right\} \right) \end{array} \right\} \\ \\ \end{array} \right) \qquad \text{(A4-73)}$$

A similar result is obtained for the latter case, that is for *k odd*. Thus, if we add (A4-69), (A4-71) and (A4-73), we will have (4-63). As for the minimum value of the quantization noise, $\xi_{\eta_q,MIN}$, it may be derived if we are looking for the minimum values of $\eta_{q,1}$ and $\eta_{q,2}$. Thus, to minimize these quantities, we must minimize $\eta_{q,1}$ and $\eta_{q,2}$ with respect to the timing phases, set the ISI and ICI components to their most negative values, respectively, and choose $\tilde{u}'_{\min}(aT_b - iT_b)$, $\tilde{v}'^{I}_{\min}(aT_b - iT_b)$ and $\tilde{v}'^{Q}_{\min}(aT_b - iT_b)$ which are the *minimum* values of the ISI and ICI, respectively, with respect to the halfband filter coefficient quantization error, $\tilde{h}_j(n_j \cdot 2^j T_s)$. These minimum values can be seen from the corresponding maximum values of (4-65), (4-66) and (4-67). If we set the rounding error of (A4-68) to its most negative value, in this case $\xi_{\tilde{h},\min} = -2^{-B-1}$, it is easily seen that these minimum values are simply the negative of (4-65), (4-66) and (4-67), these being the corresponding maximum values. Therefore, the minimum values for $\eta_{q,1}$ and $\eta_{q,2}$ are [28]:

$$min\ \eta_{q,1} = -\min_{\gamma_k} \sum_{i=\alpha-m_l}^{\alpha+m_u} \left| A_k \cdot \widetilde{u}'_{\min}(\alpha T_b - iT_b) \right| \qquad \text{(A4-74)}$$

and :

$$min\ \eta_{q,2} = -\sum_{Q\neq k} A_Q \min_{\gamma_Q, \phi_Q} \sum_{i=\alpha-m_{l,Q}}^{\alpha+m_{u,Q}} \left| \begin{array}{l} \widetilde{v}'^{I}_{\min}(\alpha T_b - iT_b)\cdot\left(\cos\phi_Q + \sin\phi_Q\right) \\ +\widetilde{v}'^{Q}_{\min}(\alpha T_b - iT_b)\cdot\left(\cos\phi_Q - \sin\phi_Q\right) \end{array} \right| \qquad \text{(A4-75)}$$

, respectively. By setting $|\phi_q| = \pi/4$, we set the ICI to its most negative value by maximizing its magnitude. Therefore, (A4-75) becomes :

$$min\ \eta_{q,2} = -\sqrt{2}\sum_{Q\neq k} A_Q \min_{\gamma_Q} \sum_{i=\alpha-m_{l,Q}}^{\alpha+m_{u,Q}} \left| \widetilde{v}'^{I}_{\min}(\alpha T_b - iT_b) \right| + \left| \widetilde{v}'^{Q}_{\min}(\alpha T_b - iT_b) \right| \qquad \text{(A4-76)}$$

To minimize $\widetilde{z}'^{I}_k(lT_b)$, we set ξ_z to its minimum value and in this case, its most negative value is $\xi_z = -3.291\cdot\sigma$ [31]. Setting this value in (A4-73) gives us the minimum value of $\widetilde{z}'^{I}_k(lT_b)$ for the lowpass outputs from the final stage (*k=0* and *k even*) and if we add it to (A4-74) and (A4-76), we get the first half of (4-64). Similarly, for *k odd*, we can repeat the above and obtain the second half of (4-64).

A4.7 : Derivation of the Truncation Error for the Finite Taylor Series of Q(·) (Derivation of (4-74))

(4-71) and (4-72) express the truncation errors for only using the first $2K_i$ terms in each Gram-Charlier Series for the marginal distributions of the interference and quantization, respectively [35]. The truncation error for considering only the first $2K_3$ terms in the orthogonal polynomial expansion of the joint probability density function has been considered negligible compared to (4-71) and (4-72). This is due to the fact that the coefficient of (4-73) for this particular truncation error would involve the joint interference-quantization moments and for purposes of analysis, the higher order joint moments are much smaller than the marginal moments thus this truncation error would be considered negligible [18].

To derive the truncation error for using only the first *(M+1)* terms in the Taylor Series for Q(·), we consider the integral form of R_M [30, pg. 880] :

$$
\begin{aligned}
R_M &= \int_0^{\psi_\eta \cdot \psi_{\eta_q}} \frac{d^{M+1}}{dt^{M+1}} \left(e^t \cdot \frac{(-1)^M}{M!} \cdot \psi_\eta^M \cdot \psi_{\eta_q}^M \right) \cdot dt \\
&= \frac{(-1)^{M+1}}{(M+1)!} \cdot \psi_\eta^{M+1} \cdot \psi_{\eta_q}^{M+1}
\end{aligned}
\qquad \text{(A4-77)}
$$

, where :

$$
\psi_\eta = \frac{\frac{A_t}{2} u(0) - \xi_\eta}{\sigma_o} \qquad \text{(A4-78)}
$$

$$
\psi_{\eta_q} = \frac{\frac{A_t}{2} u(0) - \breve{\xi}_{\eta_q}}{\sigma_o} \qquad \text{(A4-79)}
$$

, and ξ_η and $\xi_{\eta q}$ represent fixed values in the range of the interference and quantization, respectively. Thus, combining (A4-77) through (A4-79), we may arrive at a final expression for R_M as :

$$
\begin{aligned}
|R_M| &= \left| \frac{(-1)^{M+1}}{(M+1)!} \cdot \left(\frac{\frac{A_t}{2} u(0) - \xi_\eta}{\sigma_o} \right)^{M+1} \cdot \left(\frac{\frac{A_t}{2} u(0) - \xi_{\eta_q}}{\sigma_o} \right)^{M+1} \right| \\
&= \frac{1}{(M+1)!} \cdot \left(\frac{\frac{A_t}{2} u(0) - \xi_\eta}{\sigma_o} \right)^{M+1} \cdot \left(\frac{\frac{A_t}{2} u(0) - \xi_{\eta_q}}{\sigma_o} \right)^{M+1} \\
&= \frac{1}{(M+1)!} \cdot \left(\frac{-\left(\frac{A_t}{2} u(0) - \xi_\eta\right)}{\sigma_o} \right)^{M+1} \cdot \left(\frac{-\left(\frac{A_t}{2} u(0) - \xi_{\eta_q}\right)}{\sigma_o} \right)^{M+1}
\end{aligned}
\qquad \text{(A4-80)}
$$

If we choose the values of ξ_η and $\xi_{\eta q}$ to be maximum, as is indicated by (3-44) and (4-63), respectively , then we will maximize the right-hand side of (A4-80) and we will have the exact same expression as (4-74).