Importance Sampling Simulation of Free-Space Optical APD Pulse Position Modulation Receivers

by

Kenneth R. Baker

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

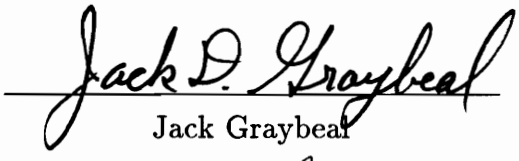
in

ELECTRICAL ENGINEERING

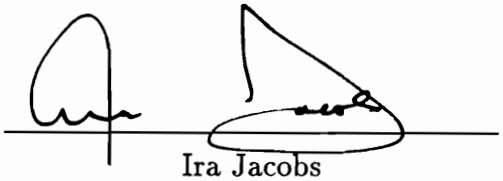
APPROVED:

Timothy Pratt, Chairman

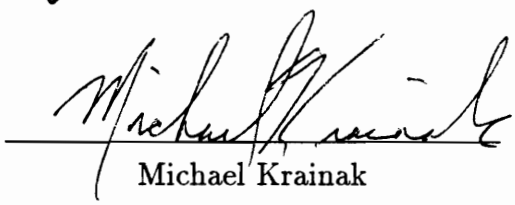
Charles Bostian



Jack Graybeal



Ira Jacobs



Michael Krainak

May, 1993
Blacksburg, VA 24061-0111

Importance Sampling Simulation of Free-Space Optical APD Pulse Position Modulation Receivers

by

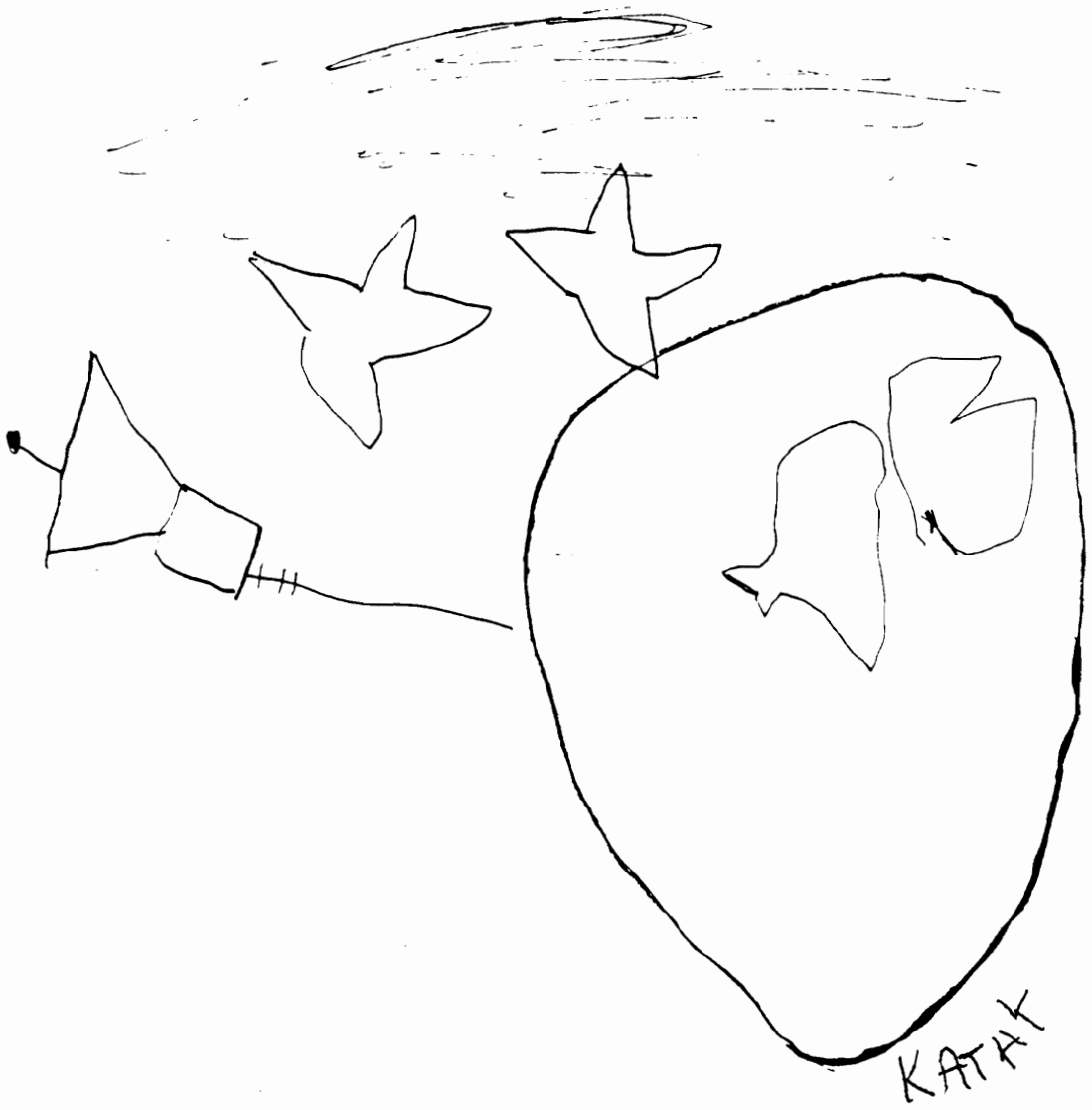
Kenneth R. Baker

A Dissertation in Electrical Engineering

ABSTRACT

Free-space optical communication technology has many advantages over RF/microwave in satellite and other spacecraft applications where reductions in size, weight and prime power requirements are combined with increased data transfer capability over long distances. Ultimately, the design and implementation of free-space optical communication systems is dependent on suitable analysis of the link. The analysis of these systems is difficult due to the complicated time-varying propagation of optical energy over the free-space channel. This difficulty is combined with a shortage of suitable analytical expressions for adequately determining the performance of free-space optical receivers. As the link must be modeled and analyzed, simulation of the free-space optical communication link can initiate the process of exploring the application of lightwave technology to the free-space channel. A prohibitive amount of time is required to simulate receiver bit error rate (BER) performance at the low error rates of interest. This dissertation presents the results achieved in reducing the amount of time required to simulate, to a given accuracy, the bit error rate performance of an APD based free-space optical receiver.

An improved technique for the importance sampling simulation of direct detection APD receivers has been developed. Two methods for efficiently simulating and biasing the probability distribution function of the APD process are presented and discussed. This is the first use the Webb, McIntyre, Conradi statistics in importance sampling simulation of an APD. The general procedure for applying importance sampling to the optical communication system simulation problem is presented in detail. The technique of importance sampling has been extended to include the simulation of maximum likelihood optical M-ary PPM receivers, an optical receiver relevant to free-space applications. The use of importance sampling is shown to reduce the time required to simulate M-PPM APD receivers by several orders of magnitude, from 9000 years to less than one hour in one example.



This dissertation is dedicated to

Nancy, Katheryn and Samantha,

my wife and daughters.

ACKNOWLEDGEMENTS

This work represents a labor of love and pain, both of which have been shared by my friend and wife, Nancy. She is acknowledged above all for this task would not have been completed without her dedication, her constant support throughout this endeavor and her belief in me even when I had little myself. Further, she should be acknowledged for the skillful creation of Figures 1.1, 1.2, 2.2, 4.1, 5.1, 5.6 and 5.8 from hand drawn sketches. This work is gratefully dedicated to her. I would also like to thank my daughter Kathy who provided a "drawing for Daddy's book," which appears on page *iii*.

I would like to express my deepest gratitude to Prof. Timothy Pratt, Ph.D. supervisor and committee chairman, for his guidance, both technical and otherwise, throughout my studies.

I wish to thank all of my Ph.D. advisory committee members: Prof. Charles W. Bostian, Prof. Ira Jacobs, Prof. Jack Graybeal (Chemistry Dept.) and Dr. Michael Krainak of NASA's Goddard Space Flight Center, for their time and effort throughout the course of this work. If, as I have found, education is a journey and not a destination then I have traveled much better as a result of my association with each of these gentleman and I have arrived at this turn in the road better off as a result of their company.

This effort has been greatly aided by many helpful discussions with Dr. Anthony Martino of NASA, Goddard, to whom I am grateful. He aided my initial BOSS simulation efforts by supplying me examples of QPPM receiver models from his own BOSS simulation studies.

I would like to acknowledge and thank the National Aeronautics and Space Administration, Goddard Space Flight Center and particularly Michael Krainak for financial support of this work.

TABLE OF CONTENTS

LIST OF FIGURES

Figures

CHAPTER 1
INTRODUCTION

The use of optical frequencies for spacecraft communication offers many advantages over traditional RF and microwave systems. These include increased bandwidth and data rate capability combined with a smaller size and weight for antennas. The realization of functional optical communication systems requires careful analysis and design. The analysis of free-space communication systems is difficult to perform analytically and must usually be performed with the liberal use of approximations. The task becomes exceedingly difficult when the effects of the atmospheric channel are considered because of the time varying nature of optical propagation through the atmosphere. Because of these difficulties, a simulation of the free-space optical communications system was chosen as the vehicle for analyzing link performance.

Simulation is a powerful and flexible tool for the solution of nonlinear communication systems problems. Its primary weakness is in the time required to achieve accurate results, especially when working with systems with low probability of error. It is desired to investigate the effects of the free-space channel on the reception of optical signals with low probability of error when a direct detection Avalanche Photodiode (APD) based receiver is used. This research has been directed at reducing the amount of time required to simulate, to a given accuracy, the bit error rate performance of a free-space optical receiver.

The simulator of a given communications system link can take different forms dependent on what aspect of the system is to be investigated. It is important then to consider at the beginning what the ultimate purpose of the simulator is to be, as this is a consideration during the design.

1.1 LEO SATELLITE COMMUNICATION

The original motivation for this study is rooted in a desire to improve the space to earth communications capability of small Low Earth Orbiting (LEO) spacecraft. Communication at optical frequencies should provide improved communications performance at a smaller size and weight. Such a system will operate as a burst-type communications link because of the time varying nature of the optical path through the earth's atmosphere.

The current trend in LEO satellites is a return to small size and weight with specific and individualized functionality. This is a result of several factors related to the economics of launch and construction and a desire to shorten the average ten year time span required for the design, construction and launch of a large spacecraft with multiple functions. Other factors are technical; for example some missions are best suited to lower earth orbits, such as some earth sensing missions. This "*smaller is better*" trend is aided by the concurrent advancement of general electronic technology which is permitting the design of spacecraft that are smaller and more powerful. The new smaller spacecraft are nearly as powerful in their ability to gather and handle information as their larger, primarily geostationary earth orbit (GEO), predecessors. The communications capacity of these smaller spacecraft needs to grow with their increased capability.

There are several issues that come into play when discussing communications systems for LEO spacecraft. Size and weight are always a consideration on any spacecraft but a smaller spacecraft bus implies less real estate on which to attach antennas. This creates a need for smaller antennas. As there is a direct relationship between the gain of an antenna and its size relative to a wavelength, one is immediately compelled to consider communication in the higher frequency portions of the electromagnetic spectrum. Additionally, the VHF/UHF and microwave spectra are crowded with space borne communications systems as well as terrestrial transmitters. The use of higher frequencies allows the use of smaller antennas and generally requires lower prime power while permitting a higher data bit rate. The penalty is that with increasing frequency the atmosphere begins to have a greater effect on the quality of the received signal. Another factor in any LEO satellite communication system is that due to their low altitude, LEO satellites are not continuously visible over a given earth location. If high data rate communication links can be achieved, even in bursts, then the utility of LEO spacecraft may be increased for some applications [1][2]. There is clearly a place for optical communication for LEO satellite to earth applications but the optical system must be designed to counter the specific difficulties of the atmospheric path.

Optical communication through the atmosphere has been under study and consideration for over 30 years. Its modern study came with the perfection of practical coherent optical sources. It was then that communications engineers began to truly appreciate something that had been painfully known to astronomers for decades: that optical propagation in the atmosphere is subject to turbulence, beam wander, scattering and absorption. Optical communication between earth and a moving spacecraft will be subject to all of these problems

and more. Whether inside or outside the atmospheric channel, free-space optical links often require high antenna gain. The resulting narrow beams impose severe pointing and tracking requirements in order to maintain each optical terminal pointed at the cooperating terminal. This requires a knowledge of the spacecraft vibration spectrum and the design and engineering of realizable control systems to compensate for this error source. The pointing and tracking problem is but one example of the multitude of problems and issues involved in the implementation of an optical earth-space link. The study of free-space optical communications is truly multi-disciplinary and the subject can easily fill whole volumes [3][4][5]. We have therefore taken one piece of this problem for in depth study. We have chosen the issue of pulse propagation through the free-space channel and its effect on the direct detection reception of optical M-ary Pulse Position Modulation (M-PPM).

1.2 OPTICAL COMMUNICATION SYSTEMS USING PPM

1.2.1 M-PPM

M-ary Pulse Position Modulation (M-PPM) has been shown to be superior to On-Off-Keying modulation over free-space optical channels[6]. M-PPM is a discrete type modulation scheme in which information is imparted onto the signal by the position in time that an optical pulse appears after the beginning of the PPM word. The M-PPM signal format and the optimum receiver architecture for its reception are described in detail in Chapter 2. Briefly, the M-PPM signal consists of a series of PPM words. A PPM word is a period of time in which a single pulse of light is made to reside in only one of M equal time slots within the word. This arrangement results in an alphabet of M different PPM words. For example, if a PPM word is designed to have four possible time slots in which a pulse might reside then the receiver function, after becoming synchronized to the beginning of the M-PPM word, is to determine which of the four time slots within the word period contained the laser pulse. With an alphabet of four words each word can be assigned two binary bits. That is, since there are four combinations of two binary bits each PPM word carries the information of two binary bits. When the slot is selected, the transmitted signal can be decoded into binary based on which time slot within the M-PPM word was found to contain the pulse.

1.2.2 The Free-Space Link

A block diagram of a complete optical communication link is shown in Fig. 1.1. Each block can be simulated on a digital computer and a sampled signal waveform can be transmitted

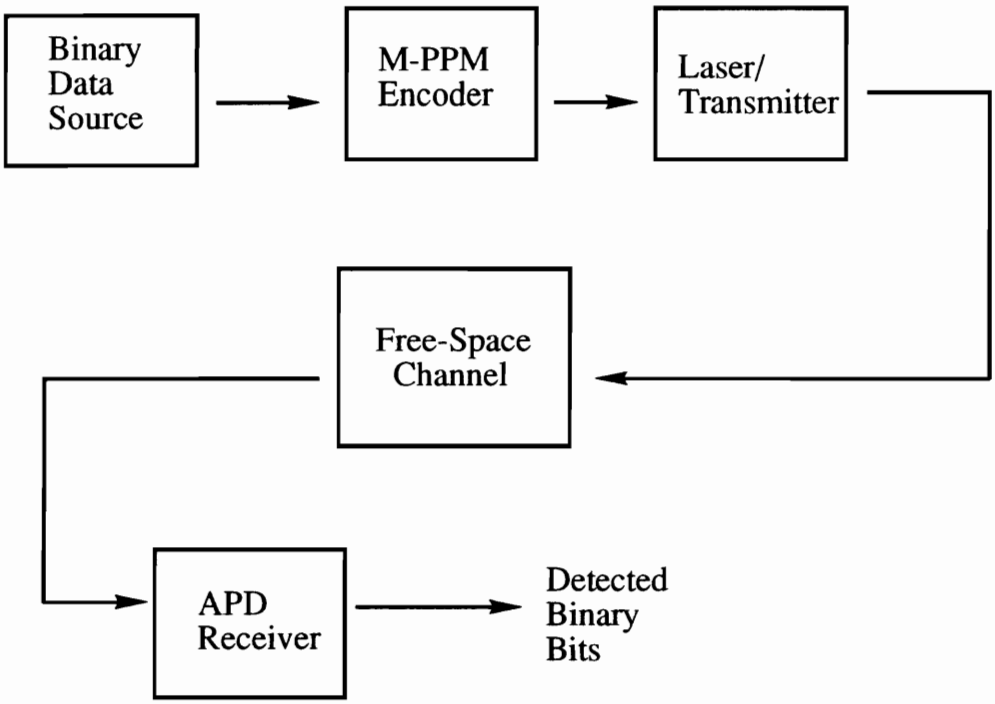


Fig. 1.1 Complete Communications Link Block Diagram Model

through each block. The binary data source generates random binary data, with an equal probability of ones and zeros. This binary data are converted into a M-PPM waveform by the M-PPM encoder. This M-PPM signal is applied to a model of the laser transmitter. An accurate simulation of a realistic optical communications link requires that the turn-on, turn-off characteristics of the laser transmitter be included in this model of a transmitter. Any aspect of the transmitting optics which affects the quality of the transmitted waveform should be modeled as part of the transmitter as well. This may include, for example, any pointing and tracking loop hardware which may cause changes in the amplitude of the received signal due to variations in pointing.

The transmitted waveform is applied to the model of the free-space optical channel. The channel model will vary depending on the link of interest and should model all channel effects which may result in the reception of an error. Such effects include any dispersive phenomena, multipath fading, attenuation due to channel scattering and absorption, etc. It may also include any background signal sources as well. After leaving the channel, the signal is applied to a model of the receiver. An accurate receiver model must include the effects of the pointing and tracking optics on the signal and any losses incurred by the receiver optics. The remainder of the receiver model includes models of the individual components; the APD, additive thermal noise of the receiver electronics, the preamplifier and filter and the PPM to binary decoding circuitry, for example. The output of the receiver is then the detected bit stream.

1.2.3 The Optical Channel and the Maximum Likelihood Receiver

In an optical channel, whether free-space or fiber guided, the carriers of information are the photons emitted from the source. Photons arrive in a random fashion that can be described by quantum theory. The photon arrival times are Poisson distributed so that the number of photons that arrive in any finite length of time, a PPM slot interval for example, is also Poisson distributed. This quantum nature of the photon counting channel results in a signal with noise-like properties even in the absence of external noise sources.

Pierce [7] showed that for the noiseless photon counting optical channel, PPM could obtain infinite channel capacity (in bits/photon) in the limit of infinite receiver bandwidth and infinite transmitted peak power. The optimum receiver in the maximum likelihood sense for a PPM signal counts detected photons in each slot [8][9][10][11]. The slot with the greatest count at the end of the PPM word is selected as the one which contained the transmitted pulse. It has

been shown that best performance of a PPM system in the presence of constant background radiation and a finite average power constraint is obtained by signaling with high peak power, short duration (impulsive) optical pulses [8] where the observable is the count of incident photons: the shorter the pulse the better the performance. This is not practical for at least two reasons. First, vanishingly short pulses would require an infinite bandwidth receiver, in the limit. Secondly, the finite average power constraint requires an ever increasing peak power which soon becomes impractical with real laser sources.

Over the free-space optical channel, the field of view of the receive telescope collects not only transmitted signal energy but also optical energy from extraneous sources. This background energy is transformed into output current along with the desired signal energy. Thus one of the slots of the PPM word has the transmitted signal energy in combination with background energy. The remaining M-1 slots have only background radiation. These are the *empty* slots. All previous literature dealing with PPM has considered that the background energy in all the *empty* slots is constant over the PPM word. As is discussed at length elsewhere in this dissertation, channel effects and bandwidth limitations will place more energy in slots adjacent to the slots with the signal. For example, dispersion through the atmospheric scatter channel will place energy in the first *empty* slot after the signal slot. The effect of this on communication system performance has not been studied except, perhaps, on a worst case basis. Because such atmospheric phenomena are difficult to analyze, simulation is an alternative method for study and evaluation.

As implied above, the optimal receiver for the PPM format can be shown to be a device which integrates the APD output over each pulse slot in the PPM word, compares the result in each slot and then selects the slot with the largest value as the slot most likely to contain the pulse of light [9][10]. No comparisons against a fixed threshold are required. The output from the integrator is proportional to the number of photons arriving during that time slot period. This proportionality is not deterministic due to the nature of the APD and the Johnson noise added by the preamplifier. The random nature of this proportionality is explored in Chapter 2.

1.2.4 The Atmospheric Channel

Communication through the atmosphere at optical wavelengths will be limited by attenuation due to absorption, attenuation due to scattering, and dispersion. Absorption refers to the processes of transferring energy into the atmospheric channel medium. Attenuation due to scattering refers to the loss of signal energy at the receiver due to the redirection of the energy

away from the receiver aperture as a result of interaction of the signal with particles in the channel. Dispersion is primarily a result of scattering by particles in the atmosphere. Turbulent variations in the index of refraction can also cause a small amount of dispersion. In this section, these processes are briefly reviewed.

If a collimated beam of light is incident on a scattering medium, a cloud or fog for example, it will in general become physically wider (spatial spreading), decollimated (angular spreading), and less intense as a result the scattering of energy out of the beam [12]. Attenuation due to scattering is not the same as attenuation due to absorption since the latter implies a transfer of energy into the medium [13]. As the light is scattered through the medium, different distances along the various possible scatter paths imply different transit times. Thus, light from a transmitted pulse will leave the scattering medium over some range of time intervals longer than the incident pulse. This variation in transit times effectively lengthens the incident pulse, producing an effect often referred to as multipath time spreading. The amount of spread is of significant interest to the designer of a pulsed communication system since it limits the maximum rate at which pulses can be resolved [12].

As stated, dispersion can result weakly from an interaction of light with the turbulent refractive index variations found in the atmospheric channel. The dispersion due to scattering by refractive index turbulence alone is very weak. During typical turbulence conditions the dispersion for a 6-m aperture receiver is on the order of 0.07ps for a vertical path through the entire atmosphere and 0.695 ps for a 15km horizontal path. It is significantly less for a point receiver aperture, less than 0.001 ps [14]. The primary source of dispersion is interaction of light with particulate matter to be found in the atmosphere. Naturally occurring scattering particles include H_2O molecules in the air in the form of vapor, as well as precipitation such as rain and snow. Man made aerosols, smog and pollutants for example, also form a scattering media in the atmospheric channel. Pulse stretching through clouds can be on the order of 70 μs [12][15].

The amount of dispersion observed is a function of the field of view of the receiving antenna [14][16]. As the antenna field of view is increased the amount of multiply scattered light able to reach the detector goes up. This must be accounted for in both modeling and experimental observations.

1.3 SIMULATION OF FREE-SPACE COMMUNICATION

1.3.1 Monte Carlo Simulation

As shown in Chapter 2, equations which describe the BER performance of a PPM receiver cannot easily be solved analytically. This is especially true if accurate statistics describing the output of the APD device are used. Monte Carlo simulation of the receiver, transmitter and the free-space channel will yield the desired BER performance bounds. Such a simulation enables such factors as pulse shape to be observed and treated as a variable through the analysis. Further, simulation permits analysis without defaulting to Gaussian statistics for the APD [17][18]. Simulation also permits the inclusion of system nonlinearities in the analysis. Thus simulation has been chosen to determine receiver performance.

Simulation of the system requires accurate models of the transfer properties for each block in Fig. 1.1. The requirement for a tidy analytic formula for each of these processes is not strict so long as the process can be accurately and efficiently modeled on a computer. An accurate implementation of the chain of events affecting the signal and noise as they flow through a model of the receiver system is the basis of Monte Carlo type simulation of the system. In Monte Carlo simulation of the BER, a large number of bits are sent through the system and detected bit stream is compared to the original bit stream. An error occurs when a detected bit is not the same as the transmitted bit. The number of errors is counted. The BER is the ratio of the number of errors to the total number of bits observed. The pertinent details of Monte Carlo simulation and the models used for simulation of the APD and other system components are presented in Chapter 3.

Since we are interested in the effect of channel phenomena on system performance, care should be taken in the design of the simulator to permit the simulation and study of received pulses with non-constant pulse shapes. Because this is a direct detection receiver, variations in the intensity of the received signal are important but changes in phase due to channel phenomena can be ignored. The desire to simulate non-constant, perhaps random, pulse shapes places certain requirements on the method of simulating the action of the APD. Without this requirement, look-up tables might be used to simulate the APD. Look-up tables for generating the APD statistics required for the simulator are valid but because of the signal dependent nature of the device, a different table is required for each input power level. When simulating a system without the channel, the shape of the input signal is well behaved and a finite number of look-up tables would be required and would suffice for the simulation. When channel effects are

included, the shape of the incoming pulse can vary greatly so that look-up tables become prohibitive. Thus, a method of simulating the APD has been pursued which does not rely on a look-up table.

1.3.2 Time Required to Perform Monte Carlo Simulation

The communications systems of interest are designed to carry information digitally with bit error rates in the range of 10^{-6} or less. While flexible and powerful, Monte Carlo simulation suffers from the amount of time required to generate the error statistics of a system operating at very low BER. Classical Monte Carlo simulation requires the simulation of a minimum of $\frac{10}{\text{BER}}$ information bits in order to predict the BER to within a 95% confidence interval. For example, to simulate a system with a BER of 10^{-6} requires 10^{7} bits to be sent through the system, corresponding roughly to the generation of 10 errors.

Suppose it is desired to simulate a system accurately to a BER level of 10^{-12}. Using the same 95% confidence level as above then it would be required to simulate 10^{13} bits. The time required for Monte Carlo simulation of a particular PPM system has been measured in our lab. The computing machine used in these studies is a SUN SPARC II. The simulation of five million binary bits required approximately 40 hours to complete. If these figures are extrapolated to the simulation of 10^{13} bits then it is determined that it would require approximately 9000 years to simulate 10^{13} bits. Note that this is approximately 9000 years for the generation of one data point on a BER versus input power curve.

This is the problem addressed by this dissertation. This research has been directed at reducing the amount of time required to simulate, to a given accuracy, the bit error rate performance of an APD based maximum likelihood M-PPM optical receiver.

1.3.3 Importance Sampling Simulation

Importance sampling is a modified form of Monte Carlo simulation. The purpose is to reduce the total number of bits which must be observed (simulated) to obtain a given accuracy in the result. This reduction is achieved by modifying the simulation – biasing it in such a way as to produce more errors. This biasing of the simulation must be performed in a controlled fashion so that its effects can be compensated for at the conclusion of the simulation procedure. If performed properly, importance sampling can yield reductions in the number of bits required in the simulation by orders of magnitude. For example, the simulation of the PPM BER data

point in the example of Section 1.3.2, which required 9000 years, has been completed through the use of importance sampling accurately in less than one hour. The entire BER versus signal power curve including probability of error as low as 10^{-12} can be produced in less than a day.

1.4 RESEARCH OBJECTIVES AND RESULTS

To date, importance sampling has not been applied to the simulation of the maximum likelihood PPM receiver. Indeed, there is no literature describing the application of importance sampling to the simulation of M-ary systems. This dissertation describes the application of importance sampling to the simulation of an M-PPM APD based optical receiver. A model of the exact system under consideration is shown in Fig. 1.2. This is a simplified version of the complete system shown in Fig. 1.1. Importance sampling has been initially applied to the receiver model only. The development of a channel model has been left for future research and development. The laser transmitter model has been simplified to its barest components for the sake of minimizing and controlling extraneous (not internal to the receiver) error sources to the receiver system during the investigation and implementation of importance sampling. The internal details of the receiver model are presented in the dissertation.

An improved technique for the importance sampling simulation of the direct detection APD receivers has been developed. Two methods for efficiently simulating and biasing the probability distribution function of the APD process are presented and discussed. These importance sampling APD models have application to the simulation of optical OOK threshold receivers as well. The general procedure for applying importance sampling to the optical communication system simulation problem is presented in detail. The science of importance sampling has been extended to include the simulation of maximum likelihood optical M-ary PPM receivers. This represents the first use of importance sampling in the simulation of optical PPM systems using an APD and additive Gaussian noise. The use of importance sampling is shown to reduce the time required to simulate M-PPM APD receivers by several orders of magnitude.

1.5 OTHER FREE-SPACE SYSTEMS

During the course of this study, the applicability of this research to other free-space optical communication problems became apparent. The natural immunity to background radiation and other properties of M-PPM are desirable in most free-space communications applications. Thus, this investigation of M-PPM APD receiver performance has many

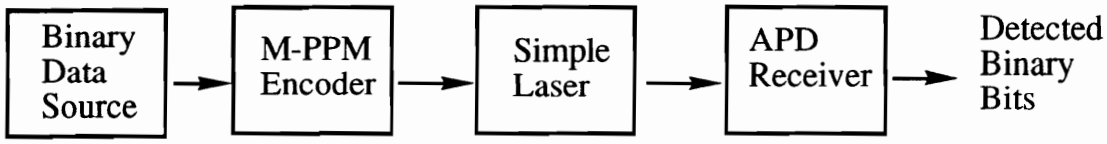


Fig. 1.2 Receiver Simulation Block Diagram Model

implications outside the issue of optical earth to LEO satellite communication links.

Links with a truly free-space channel, for example GEO to GEO, LEO to GEO, LEO to LEO intersatellite links, or earth orbit to deep space communication links, have all been considered candidates for optical frequency communication. These links have few channel propagation effects as the channel is a vacuum. Despite this, knowledge of PPM receiver performance to low levels of error rate may be desirable when considering such links. These systems will rely on complicated control systems in order to maintain correct pointing of each communications terminal. Characterization of the receiver over a wide range of input power levels can be combined with information concerning pointing and tracking capability to indicate overall system performance.

There are many optical communication links which have significant channel phenomena. Besides the space to earth link described here there are space to ocean applications, air to air applications, and undersea links to name a representative few. Many of these links generally involve great distances.

There is also interest in the performance of optical frequencies for use on the scatter channel communications link. These links make use of what is normally considered a hostile channel by using the scatter properties of the medium to direct energy into the receiver aperture in order to complete the link. Thus there is no line of sight path for the signal energy. Such links exist for the radio frequency portion of the spectrum but there is little mention concerning the optical scatter channel communications link in the literature. The question relative to this research concerns how would PPM with maximum likelihood detection well perform in this application.

Another area for application of free-space optical communication technology is indoor optical communication. Applications such as real time video and high data rate communication from the desk top will drive future network topology. It has been predicted that the desk of the not too distant future will need a 150 Mbps communications ability[19]. The personal communications revolution of today, referring here to the tremendous proliferation of cordless telephones, cellular telephones, pagers and even telephones on airplanes, is evidence that an ability to communicate without fiber and wire is strong in the user community. Research is currently underway at places like IBM and Bell Labs in the use of infrared systems to provide a wireless local area network capability[20][21][22][23]. Much of the current effort is directed at characterizing and compensating for the indoor optical channel. This channel has a large

multipath component which tends to broaden and distort the pulse shape. Simulation results have shown significant multipath dispersion contribution that broadens the pulse to over 40 nS [23]. Additionally the indoor channel will have a large background component due to the wide FOV antennas required for the mobile indoor system. The research research described in dissertation is directly applicable to a study of the use of PPM in this indoor multipath environment. It is interesting to note that this, admittedly brief, survey of the indoor optical literature indicates that PPM has not been considered for indoor optical applications although PPM has inherent multipath immunity.

A final form of communication, in the very broadest sense, is the remote sensing of objects in order to gain information pertaining to location, motion or composition. One example is radar, or lidar as it is known for the case of optical radiation. The transmission and reception of PPM from a single aperture has been suggested as having use in lidar. The concept is to form a Pseudo-random Noise (PN) coded lidar system that could resolve range to less than a single chip via the reception of and the synchronization to a PPM encoded PN sequence[24]. Reflected signals from uncooperating targets, such as clouds, will most likely contain a large multipath component. Again, it is hoped that the research reported here will improve the study of the feasibility of PPM in these applications through reduced simulation time.

Analysis of these potential PPM applications requires an understanding of the effect of the channel on the pulse using a PPM maximum likelihood receiver. Even the vacuum channel systems benefit from the consideration of bandwidth as it effects PPM performance. The effects of bandwidth on receiver performance is difficult to analyze but can be studied through simulation. Other researchers have considered Gaussian pulse shape on PPM due to dispersion in the optical fiber channel[25][26][27]. All of these studies have analyzed the threshold detection receiver. It is therefore pertinent to consider the "greatest of" or maximum likelihood receiver structure. As this receiver is difficult to analyze, we propose to explore the performance through simulation.

1.6 ORGANIZATION OF DISSERTATION

Chapter 2 presents a description of the M-PPM signal format, the maximum-likelihood M-PPM receiver and an in-depth discussion of the error mechanisms of the APD receiver to be modeled.

The fundamentals of Monte Carlo simulation are reviewed in Chapter 3. This permits

the introduction of principles for importance sampling simulation. Chapter 3 concludes with a discussion of the simulation package used to implement the techniques developed in this dissertation as well as a discussion of the models used for the APD and the PPM receiver system.

Chapter 4 reviews previous approaches to the simulation of communication systems. It is seen that clear guidelines have not been established for the application of importance sampling to even the simple threshold detection receiver system and that the M-PPM system has not been considered. There are few references in the existing literature to the application of importance sampling to simulation of APD optical systems in general. Yet careful review of the literature has produced concepts and techniques useful in the application of importance sampling simulation to improving the simulation of M-PPM APD optical receiver systems. These techniques are distributed among various investigators with no universal approach to the application of importance sampling to communication system simulation.

Chapter 5 presents the theory and practice of importance sampling as applied to the M-PPM maximum likelihood receiver. This includes a discussion of estimating the error in the simulation and techniques for determining the validity of the simulation operation. The optimum M-PPM bias function is derived and it shown to be unrealizable. Two sub-optimum alternatives are developed for experimentation and application.

In Chapter 6, examples of the use of importance sampling in the simulation of M-PPM systems are presented. Comparison of the number of trials required for Monte Carlo simulation to the number of samples required for importance sampling simulation shows improvement of several orders of magnitude. The amount of improvement is dependent on the exact system architecture, particularly the amount of memory inherent in the system.

Chapter 7 presents a summary of the results and concluding remarks.

CHAPTER 2
OPTICAL PPM RECEIVER OPERATION

2.0 INTRODUCTION

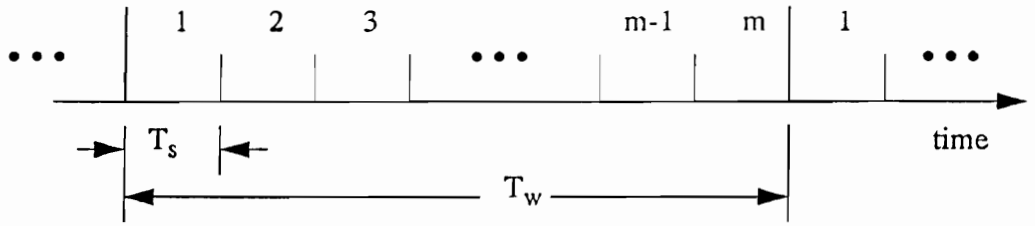
In this chapter we discuss in detail the optical PPM modulation format, the maximum likelihood receiver structure, and the processes which contribute to an optical PPM word error. With this mathematically based treatment of the optical receiver in place we then describe the analysis options available for such optical systems in Chapter 3.

2.1 OPTICAL M-ARY PULSE POSITION MODULATION

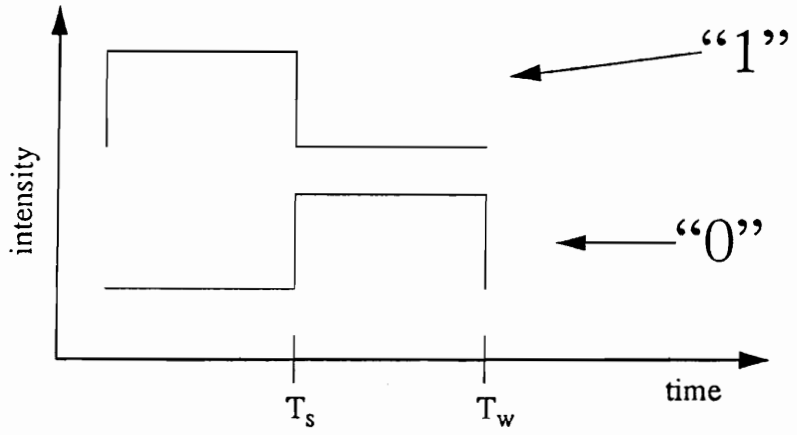
2.1.1 Description

PPM is a digital modulation format well suited to free space laser communication. It has low average power and provides some means of discrimination from background noise sources. The general PPM word structure is shown in Fig. 2.1(a). In M-ary PPM signaling, L binary bits are transmitted as a single pulse of light. This pulse can reside in only one of $M = 2^L$ possible time slots. We denote the time slot duration as T_S and the PPM word time as T_W where $T_W = M \cdot T_S$. Semiconductor laser diodes are both peak and average power limited. For a given average power, PPM signaling produces high peak power short duration pulses as M is increased. Such pulses are easier to distinguish from background and thermal noise. The value of M is constrained by the peak power output capability of the laser diode. (Peak power = M · average power) The peak power must be maintained below that level which produces facet damage in the laser. Peak power limitations of semiconductor lasers generally limit M to values of 8 or less [1][2].

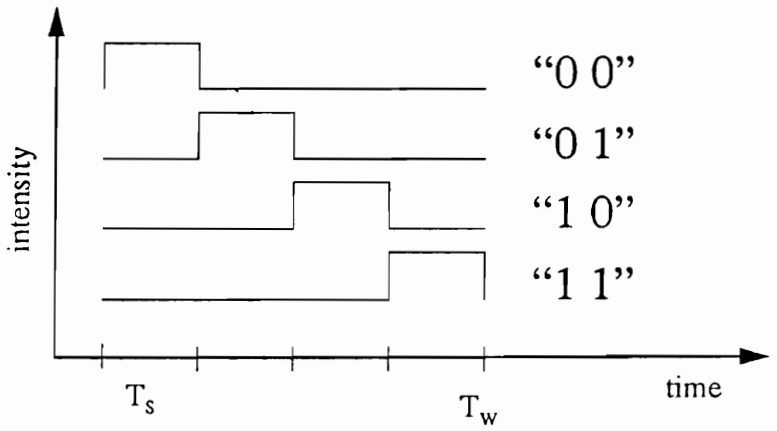
For example, in Binary PPM (BPPM) $M = 2$ and $L = 1$. In other words, each bit is mapped into one of two PPM words. This is illustrated in Fig. 2.1(b) along with one of two possible mappings of binary data into PPM symbols. For another example consider Quaternary PPM (QPPM). In this modulation $M = 4$ and $L = 2$ so that every two bits are mapped into one of four PPM words. QPPM is illustrated in Fig. 2.1(c). Shown for illustration is one of the four possible mappings of binary data into PPM symbols. QPPM has been shown to be more efficient over most free space optical communications channels[3]. It was proposed for use on the ESA SILEX Project and also on NASA's LCT communications projects[4].



a) General M-ary PPM Word Structure



b) BPPM structure and one possible mapping to binary



c) QPPM structure and one possible mapping to binary

Fig. 2.1 M-ary PPM Signal Format

where T_S designates the PPM slot time and T_W designates the PPM word time.

2.1.2 PPM Word Error and Bit Error with Maximum Likelihood Detection

The optimal receiver in the maximum likelihood sense for the PPM format can be shown to be a device which integrates the APD output over each pulse slot in the PPM word, compares the result from each slot and then selects the slot with the largest value as the slot that was most likely to have contained the pulse of light [5][6]. No comparisons against a fixed threshold are required. A generic block diagram of the optimum PPM receiver is shown in Fig. 2.2. The output from the integrator is proportional to the number of photons arriving during a slot time period. This proportionality is not deterministic due to the random nature of the APD and the Johnson noise added by the preamplifier. The statistics of this random process are explored in Section 2.2.1. In this section we derive an expression for the probability of a PPM word error and the relationship between the probability of bit error given that an incorrect slot choice (a PPM word error) has been made.

The probability of correct word reception, P_{cW}, is the probability that the integrator output for the slot containing the transmitted pulse is greater than the output from any of the other PPM slots. Alternatively, the probability of word error, P_{eW}, in this type of receiver is equal to the probability that the output of the integrator for a slot not containing the transmitted pulse is greater than the output for the slot which contained the pulse. It is possible to develop an equation for P_{eW} based on either alternative. We express the probability of word error as $1-\{P_{cW}\}$.

Consider the case that the pulse was transmitted in the q^{th} PPM slot where $1 \leq q \leq M$. Let r_q be the integrator output of the q^{th} slot and r_i represent the output of each of the other slots where $i = 1, 2, \ldots, M\text{-}1, M$ and $i \neq q$. The probability of the correct decision is

$$P_{cW} = \text{prob}(r_q > r_i) \qquad \text{for all } i = 1, 2, \ldots, M\text{-}1, M \text{ and } i \neq q$$

or equivalently

$$P_{cW} = \int_{\text{all } r} [\text{prob}(r_q = r)] \cdot [\text{prob}(r_{i=1} < r)] \cdot [\text{prob}(r_{i=2} < r)] \cdot \ \ldots \ \cdot [\text{prob}(r_{i=M} < r)], \quad i \neq q \tag{2.1}$$

The value of the integrator output, r, is a continuous random variable governed by probability density function discussed in Section 2.2. There are several mechanisms at work. Briefly, the randomness of the integrator output is the consequence of, first, the random Poisson distributed arrival times of the incident optical energy. These arriving photons are multiplied by the gain of the APD. This gain is a random quantity due to the physics of avalanche

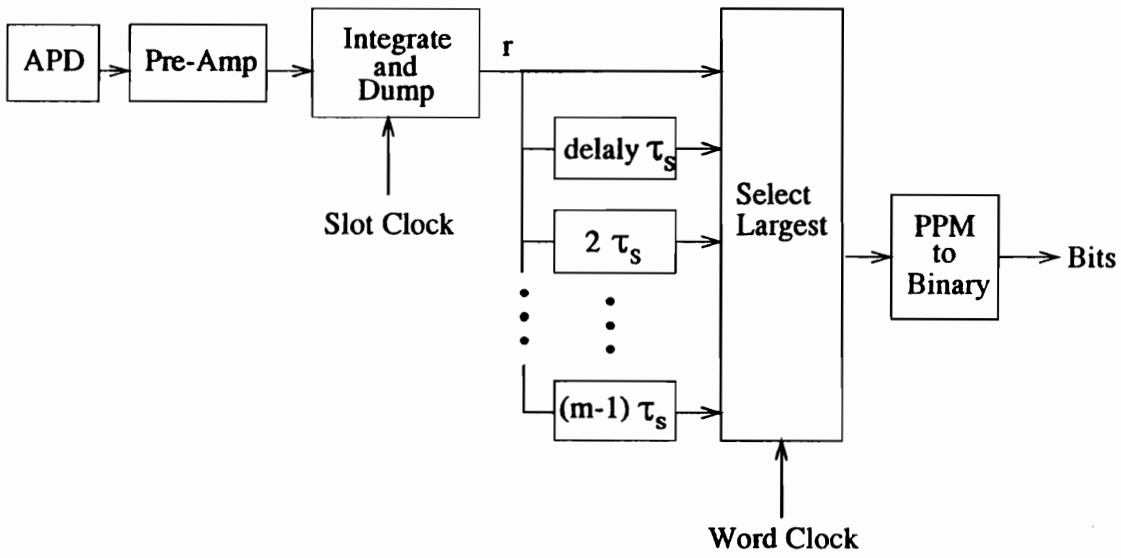


Fig. 2.2 Generic Maximum Likelihood Receiver for Optical M-ary PPM

where T_S designates the PPM slot time.

multiplication. Finally the signal out of the APD is a current which is amplified by a preamplifier stage. This stage further corrupts the signal with additive Gaussian noise. These mechanisms combine with Inter-Symbol Interference (ISI) and other effects of filtering to yield a probability density function (PDF) for r_q that we denote as $f_q(r)$ and a PDF for the r_i as $f_i(r)$, $i = 1, 2, \ldots, M\text{-}1, M$ and $i \neq q$. We assume for now that the probability densities for the $i \neq q$ time slots are equal. This corresponds to a constant average background noise level over the duration of the PPM word. With this notation the probability of correct detection is

$$P_{cW} = \int_{-\infty}^{\infty} f_q(r) \left[\int_{-\infty}^{r} f_i(r')dr' \right]^{M-1} dr \tag{2.2}$$

Eq.(2.2) is simply a restatement of eq.(2.1) using the PDF $f_i(r)$ to calculate the Prob(r_i<r). The probability of word error is then

$$P_{eW} = 1 - \int_{-\infty}^{\infty} f_q(r) \left[\int_{-\infty}^{r} f_i(r')\, dr' \right]^{M-1} dr \tag{2.3}$$

Eq.(2.3) forms the basis of much of the development found in this work.

The probability of a bit error is a function of P_{eW}. Recall that each PPM word of M slots contains L binary information bits where $L = \log_2(M)$. This yields an alphabet of $2^L = M$ PPM words each representing L binary bits. We assume the general condition that all PPM words are equally likely, which is equivalent to assuming that the ones and zeros that generated them are equally likely. Under the additional assumption that an incorrect word can, with equal likelihood, be any of the other $M-1$ words then a given bit in this incorrect word may equally likely be any of the bits in the same bit position of the $M-1$ other words. Hence there is a $\frac{1}{M-1}$ probability of each of the other bits occupying that position, some correct and some incorrect. In M equally likely patterns, a given bit position can be a one or a zero $\frac{M}{2}$ times. We immediately have that out of the $M-1$ other bits in that position, $\frac{M}{2}$ are different from the given bit and $(\frac{M}{2}-1)$ are the same. Thus given that a word is in error each bit has a probability $\left(\frac{M}{2}\right)\cdot\frac{1}{M-1}$ of being incorrect. This leads to the following expression for the probability of bit error, P_e, as

$$P_e = \text{BER} = \frac{1}{2}\left(\frac{M}{M-1}\right) \cdot P_{eW} \tag{2.4}$$

2.2 APD Receiver Operation

We have seen that PPM word errors occur when the photon count for a slot not containing the pulse is determined to be greater than the photon count of the slot which actually contained the pulse. Ignoring channel effects and bandwidth limitations, this error has a finite probability of occurring as a result of three error mechanisms in the APD based receiver. The random nature of the APD device is presented in the following sections as well as mathematical descriptions of thermal noise contributions to the APD based optical receiver.

2.2.1 APD Error Mechanisms

An APD is a photodiode with an internal gain mechanism. The existence of gain at the detector and before the preamplifier is advantageous for weak signal performance. This is the case in free-space optical systems as well as some long haul optical fiber systems. Consider the case of the reception of a weak signal by a device without internal gain, a p-i-n diode for example. The current out of the device is very small and must be amplified in order to make it large enough to measure. In many cases this small signal will be smaller than the thermal noise added by the receiver. Thus the limit on sensitivity becomes the noise level of the preamplifier. A device with gain, an APD or a photo-multiplier tube for example, at the front end of the optical receiver can extend the sensitivity by providing detected signals above the noise floor of the preamplifier. The sensitivity then becomes limited by the noise out of this detection device with multiplicative gain, including any noise inherent in the multiplication process.

The first APD error mechanism concerns the quantum nature of the signal itself. The signal is essentially a group of photons which are Poisson distributed in their time of arrival while the number of photons is proportional to the intensity of the incident radiation. As a result the signal at the receiver is noise-like. As the incident photons arrive at the detector they are absorbed and converted to primary electron-hole pairs inside the avalanche region of the APD. This conversion is not 100% efficient. It is a function of the wavelength and the material of the device. This gives rise to a factor known as the quantum efficiency, η, which describes the efficiency at which photons are converted to primary electrons and holes. Since we consider the output of the APD to be electrons we no longer discuss the holes.

The mean number of photons $\overline{n}$ that are absorbed from an incident optical field of intensity $P_0(t)$ watts in the time interval $\{t, t+dt\}$ is given by

$$\bar{n} = \frac{\eta}{hf}\int_{t}^{t+dt} P_0(\tau)\, d\tau \tag{2.5}$$

where: h = Plank's constant

f = frequency of the radiation

$P_0(t)$ = incident optical intensity at frequency f.

Recall that hf is the energy of a photon. $P_0(t)$ can include both signal and background power.

Since one electron is formed for every absorbed photon, $\bar{n}$ is equivalently the mean number of primary electrons found in the avalanche region of the APD over the time *dt*. The exact number is Poisson distributed about this mean. Hence the number of primary electrons, n, injected into the APD has a PDF $f_n(n)$ given by

$$f_n(n) = \frac{\bar{n}^n}{n!} \cdot \exp(-\bar{n}) \tag{2.6}$$

These primary electrons are accelerated through the avalanche region of the diode by the electric field from the reverse bias applied to the diode. This reverse bias controls the average gain of the device. As the electrons accelerate they collide with bound electrons to form new electron-hole pairs. Thus every primary electron results in g_i secondary electrons out of the APD device where g_i includes the primary electron and $i = \{0, 1, 2, \ldots, n\}$. This number g_i is a random quantity. It is a function of the detector's physical characteristics and the applied reverse voltage. The mean of $g_i = \langle g_i \rangle$ is the average gain of the detector and is denoted as G. Thus while an APD provides gain, that is (on average) more electrons are output from the APD than the number of primary photo-electrons detected at its input, this gain is random. This uncertainty in the number of output electrons for a given number of primary electrons is the second mechanism to influence PPM word errors. Let us explore the statistics of this process.

2.2.2 The Webb-McIntyre-Conradi (WMC) Density

The probability density function describing the random number of secondary electrons output by the APD in response to a single primary photon was established by Personick[7] and McIntyre[8] and confirmed by Conradi[9] to be

$$f_{g_i}(g_i) = \frac{(1-\kappa)^{g_i-1}\Gamma\left(\frac{g_i}{1-\kappa}\right)\left[\frac{1+\kappa(G-1)}{G}\right]^{\left[\frac{1+\kappa(G_i-1)}{1-\kappa}\right]}}{[1+\kappa(G_i-1)](G_i-1)!\,\Gamma\left(\frac{1+\kappa(g_i-1)}{1-\kappa}\right)}\left[\frac{G-1}{G}\right]^{g_i-1} \tag{2.7}$$

where: G = average gain = $<g_i>$

κ = ionization coefficient

$\Gamma(\cdot)$ = gamma function

The ionization coefficient κ is a device parameter proportional to the ratio of the hole and electron ionization rates and is generally a property of the material from which the device is made. Silicon APD's with $\kappa<0.01$ are now available commercially. As we shall see, the excess noise out of the APD device is proportional to κ so that smaller is better. Eq.(2.7) is sometimes referred to as the McIntyre density function.

Thus for every primary electron there are g_i electrons out of the APD device where g_i is a random integer described by the above McIntyre PDF. The total number of electrons out of the APD over the time interval $\{t, t+dt\}$ is a random number r where

$$r = \sum_{i=0}^{n} g_i. \tag{2.8}$$

Recall that n is a Poisson distributed random variable and that each g_i value is a random quantity under the McIntyre density function. Therefore r, the number of electrons out of an APD over the time interval $\{t, t+dt\}$, is a discrete random variable given by the summation of a random number of random variables. The range of r is $0 \leq r<\infty$ as there is often a significant probability of zero primary electrons if the incident energy is small or the time scale, dt, is small.

Of interest for both simulation and analysis reasons is the probability density function for the random variable r. The probability density function for r is, by definition,

$$\begin{aligned} f_r(r) &\equiv \text{Prob}\left[\sum_{i=0}^{n} g_i = r\right] \\ &= \sum_{n=0}^{\infty} f_{r|n}(r|n) \cdot f_n(n) \end{aligned} \tag{2.9}$$

where: $f_n(n)$ is the Poisson density function given in eq.(2.6) above

and $f_{r|n}(r|n) = \text{Prob}\left[\sum_{i=0}^{n} g_i = r \;\middle|\; n\right]$ = "the probability that the sum is equal to r given n."

By definition the probability of r conditioned on n is

$$f_{r|n}(r|n) = \text{prob}\left[\sum_{i=0}^{n} g_i = r \;\middle|\; n\right] \equiv f_g(g_0) * f_g(g_1) * f_g(g_2) * \ldots * f_g(g_{n-1}) * f_g(g_n) \tag{2.10}$$

In eq.(2.10) there are n convolutions of the McIntyre density function, eq.(2.7). Recall that n is a Poisson distributed random variable.

The convolutions are a result of the fact that the PDF of the sum of random variables is the convolution of the individual PDF's. As written, $f_{r|n}(r|n)$ is generally in an intractable form. As proposed by McIntyre[8], and later verified by Balaban, et al.[10], the conditional probability is given as

$$f_{r|n}(r|n) = \frac{n(1-\kappa)^{r-n}\,\Gamma\left(\frac{r}{1-\kappa}\right)\left[\frac{1+\kappa(G-1)}{G}\right]^{\left[\frac{n+\kappa(r-n)}{(1-\kappa)}\right]}}{[n+\kappa(r-n)](r-n)!\,\Gamma\left(\frac{n+\kappa(r-n)}{(1-\kappa)}\right)} \cdot \left[\frac{G-1}{G}\right]^{r-n} \tag{2.11}$$

If this expression is substituted into eq.(2.9) for the $f_r(r)$ then we arrive at what is occasionally referred to as the Conradi density function:

$$f_r(r) = \sum_{n=0}^{\infty} \frac{n(1-\kappa)^{r-n}\,\Gamma\left(\frac{r}{1-\kappa}\right)\left[\frac{1+\kappa(G-1)}{G}\right]^{\left[\frac{n+\kappa(r-n)}{1-\kappa}\right]}}{[n+\kappa(r-n)](r-n)!\,\Gamma\left(\frac{n+\kappa(r-n)}{1-\kappa}\right)} \left[\frac{G-1}{G}\right]^{r-n} \cdot \frac{\bar{n}^n}{n!} \cdot e^{-\bar{n}} \tag{2.12}$$

Eq.(2.12) is a function of κ, G and $\bar{n}$. The first two are APD parameters, assumed given and constant, while $\bar{n}$ is the mean number of primary photons and is a function of signal strength as given by eq.(2.5). The McIntyre and Conradi densities of eq.(2.11) and, particularly, eq.(2.12), constitute what is referred to as the *exact* APD statistics.

Parenthetically, note that the Conradi density function also appears in the literature in the following form [11][12]:

$$f_r(r) = \sum_{n=0}^{\infty} \frac{n\Gamma\left(\frac{r}{1-\kappa}+1\right)}{r\,(r-n)!\,\Gamma\left(\frac{\kappa r}{1-\kappa}+1+n\right)} \cdot \left[\frac{1+\kappa(G-1)}{G}\right]^{\left[\frac{n+\kappa r}{1-\kappa}\right]} \cdot \left[\frac{(1-\kappa)G-1}{G}\right]^{r-n} \cdot \frac{\bar{n}^n}{n!} e^{-\bar{n}},\ r \geq 1$$

It can be shown that this equation is the same as eq.(2.12) above. The limit $r \geq 1$ is given here but not in eq.(2.12) above. In both cases the $f_r(r = 0)$ is equal to the probability that $n = 0$ and is obtained from the Poisson density function of eq.(2.5). That is

$$f_r(r = 0) = f_n(n = 0) = \exp(-\bar{n})$$

Webb, McIntyre and Conradi (WMC) produced an approximation to the Conradi density function as[13]:

$$f_r(r) = \frac{1}{\sqrt{2\pi}\,\sigma} \cdot \frac{1}{\left[1+\left(\frac{r-\bar{n}G}{\sigma\lambda}\right)\right]^{\frac{3}{2}}} \cdot \exp\left[\frac{-(r-\bar{n}G)^2}{2\sigma^2\left[1+\left(\frac{r-\bar{n}G}{\sigma\lambda}\right)\right]}\right] \qquad \text{for } \bar{n}<r<\infty \tag{2.13}$$

where: G = the average gain of the APD

σ^2 = the variance of $r = \bar{n}G^2F$

$F = \kappa G + (2-\frac{1}{G})(1-\kappa)$ = the excess noise factor

and $\lambda = \frac{\sqrt{\bar{n}F}}{F-1}$.

The WMC density function is also a function of κ, G and $\bar{n}$. A new parameter appears in this expression known as the excess noise factor, F, a function of κ and the average gain G. The excess noise factor is the ratio of the actual noise out of the device to that which would exist if the multiplication process were noiseless. Noiseless multiplication implies that all primary electrons are multiplied by exactly the same factor. This is an ideal device such that the gain is a constant for each primary electron and not a random variable. Since F is proportional to κ, a smaller κ is better.

The form of eq.(2.13) illustrates that the output density function is approximately Gaussian for values of r in the neighborhood of its mean, specifically, when $|(r-\bar{n}G)| \ll \bar{n}G$. The range of the equation which is shown above, $\bar{n}<r<\infty$, is that which was originally published by WMC in [13]. This range somewhat restricts the usefulness for precise simulation or analysis of PPM receiver problems although it has been successfully used in both analysis [11] and simulation studies [14]. The latter author used the approximation without reference to any restrictions on the range of the random variable. This may be because the exact density falls off rapidly for values of $r<\bar{n}$. Physically the range of the random variable extends from zero to infinity. Many later authors have used this range when utilizing eq.(2.13). We also find it convenient to follow this trend and will assume that the approximation of eq.(2.13) is valid for $0<r<\infty$.

In conclusion, we find that there exist two mathematical formulations for describing the statistics of the output for an APD. The first is the family of exact formulas, generally eq.(2.7) through eq.(2.12). The second is the WMC approximation shown in eq.(2.13). In theory, at least, there are three ways that these relations might be used to generate the random variates needed for the simulation of an APD in a PPM receiver. The first method is the implementation of eq.(2.8) which utilizes eq.(2.6) and eq.(2.7). This requires an ability to

generate Poisson random variates according to eq.(2.6). Generating Poisson random variates is a standard capability of most computer systems. We then must generate random variates distributed via eq.(2.7), which is not a standard function. The second method begins with the implementation of eq.(2.12) for a given mean primary photocount. This might be accomplished through a cumulative distribution function (CDF) look-up table approach but it would require a new table for each new mean photo count. Thus a table is not practical for simulations in which channel effects may cause variation of the input power to the APD. Lastly there is the WMC approximation of eq.(2.13). An efficient method of generating random variates distributed via eq.(2.13) has been developed[15]. Due to the ease with which random variates can be generated, the speed of this generation method over implementing one of the *exact* APD statistical methods and the ability to generate APD random variates for a variable input optical intensity system, we have used the WMC PDF in our simulation studies. The method of generating random variates is described along with the APD model in Chapter 3, Section 3.3.2.

2.2.3 The WMC Density as an Inverse Gaussian Probability Density

In this section we show that the WMC PDF of eq.(2.13) is an inverse Gaussian density since they are related through a linear translation of variables. There are several consequences to this result. We immediately have a closed form cumulative distribution function (CDF) for the WMC PDF which was previously unavailable for application in APD detector analysis problems. There are now alternate methods for generating random variates governed by the WMC PDF which are useful for the simulation of optical systems. More generally, this finding makes available the extensive literature on the inverse Gaussian density for application to APD detector analysis problems. The inverse Gaussian density has been well studied and the reader new to the function is directed toward the excellent review found in [16] with many references, the discussion in [17] and [18], and the thorough analysis of the properties of the inverse Gaussian density function found in [19].

The inverse Gaussian density was first expressed by Schrödinger in 1915 to describe the first passage time in Brownian motion[16]. It was given the name inverse Gaussian by M. C. K. Tweedie[17] in 1945 who recognized that the cumulant generating function of this PDF and the Gaussian PDF are inverses. It is also known as Wald's density, particularly in the Russian literature[16]. In Tweedie's notation, the probability density function of a random variable X distributed as an inverse Gaussian with parameters μ and β is given by

$$f(x;\,\mu,\,\beta) = \begin{cases} \dfrac{\sqrt{\beta}}{\sqrt{2\pi}} \cdot [x]^{-\frac{3}{2}} \cdot \exp\left[\dfrac{-\beta(x-\mu)^2}{2\mu^2 x}\right] & x > 0 \\ 0 & \text{elsewhere} \end{cases} \tag{2.14}$$

where μ and β are positive[16]. The cumulative distribution function of X, denoted as F(x), is expressed in terms of the standard normal (Gaussian) cumulative distribution function, Φ, as

$$F(x) = \Phi\left[\sqrt{\frac{\beta}{x}}\left(-1+\frac{x}{\mu}\right)\right] + e^{\frac{2\beta}{\mu}}\Phi\left[-\sqrt{\frac{\beta}{x}}\left(1+\frac{x}{\mu}\right)\right] \tag{2.15}$$

The standard normal cumulative distribution function can be realized in terms of the Q function, the error function or the complementary error function as desired.

We now show that if a random variable distributed as the WMC PDF of eq.(2.13) is made the argument for a linear function and transformed into a new random variable then the PDF of the new variable will be the inverse Gaussian of eq.(2.14). Since the transformation is linear, the cumulative distributions of the original and transformed variables will be the same following a suitable change of variables.

Beginning with the WMC PDF as given in eq.(2.13), the transformation is as follows. We define a new variable, x, as

$$x = 1+\left(\frac{r-M}{\sigma\lambda}\right) \tag{2.16}$$

The derivative of x with respect to r is

$$\frac{dx}{dr} = \frac{1}{\sigma\lambda} \tag{2.17}$$

The PDF of the random variable x is determined by the fundamental transformation law of probabilities, which can be written[20]

$$f_X(X) = \frac{f_r[X(r)]}{\left|\frac{dx}{dr}\right|} \tag{2.18}$$

where $f_r[X(r)]$ is the original density written with the appropriate substitution of variables. Thus

$$f_X(X) = \frac{\frac{1}{\sqrt{2\pi}\,\sigma} \cdot \frac{1}{x^{\frac{3}{2}}} \cdot \exp\left[\frac{-\lambda^2(x-1)^2}{2x}\right]}{\left|\frac{1}{\sigma\lambda}\right|} \tag{2.19}$$

$$f_X(X) = \frac{\lambda}{\sqrt{2\pi}} \cdot [x]^{-\frac{3}{2}} \cdot \exp\left[\frac{-\lambda^2(x-1)^2}{2x}\right] \tag{2.20}$$

Comparison of eq.(2.20) with that of eq.(2.14) shows that the equations are identical if we assign $\beta = \lambda^2$ and $\mu = 1$.

The CDF of the WMC/inverse Gaussian density, eq.(2.15), can be written in terms of the APD parameters and the complimentary error function as

$$F(x) = \frac{1}{2}\left\{\text{erfc}\left[\frac{-\lambda(Z-1)}{\sqrt{2Z}}\right]\right\} + \frac{\exp(2\lambda^2)}{2}\left\{\text{erfc}\left[\frac{\lambda(Z+1)}{\sqrt{2Z}}\right]\right\} \tag{2.21}$$

where $Z = Z(x) = \frac{x}{\lambda\sigma} + \left[1 - \frac{M}{\lambda\sigma}\right]$ has been used for notational simplicity. Eq.(2.21) is derived from eq.(2.15) after making the appropriate substitution of variables, i.e. Z(x), and using the complimentary error function expression for the Gaussian CDF. Equivalent expressions can be obtained by using the Q function or the error function.

Having a closed form solution for the CDF of the WMC density brings new capability to the solution of some problems in communications using APD devices and avoids the need to devise numerical solutions or to approximate the density by a Gaussian.

The recognition of the WMC PDF as an inverse Gaussian density opens up new approaches for solution of the APD optical receiver sensitivity calculation problem, including efforts incorporating the additive Gaussian noise component into the result, as well as other APD based optical detection problems in the literature. It provides a ready tool for the simplified calculation of the threshold and approximate error probabilities of optical receivers in applications. Eq.(2.21) can be used to provide a quick proof of the inadequacy of the Gaussian PDF assumption for modeling the APD output statistics.

We have used eq.(2.13) and eq.(2.21) in our APD receiver simulation studies along with Ascheid's method of generating random variates. Eq.(2.21) is especially useful in the application of importance sampling techniques to the simulation of APD based optical receiver systems. This is discussed in Chapter 5.

2.2.4 Preamplifier Noise

An additional error mechanism to be considered is the additive thermal noise of the post detector electronics. The ISI contribution from the finite bandwidth of these electronics must also be considered and modeled in the simulation. The system bandwidth is typically dominated by the bandwidth of the preamplifier which follows the detector. Ample studies concerning the treatment of the optical preamplifier noise exist in the literature [21],[22],[23] so

it is not treated in detail here. With a noisy amplifier or a relatively noise free detector, for example an APD at low gain, the system noise temperature can be dominated by the noise temperature of the preamplifier. In systems with high APD gain and/or high excess noise ratio the APD statistics can dominate the receiver performance. The thermal noise power is a function of the thermal resistance presented by this amplifier. This noise temperature and resistance are functions of the amplifier design and controlled by the specific choice of a BJT or FET as the active device. For the simulation studies here, a model of a commercially available preamplifier in use in our lab was used. This model includes the thermal noise power and the filter bandwidth characteristics as specified on the data sheet.

In general, we note that the thermal noise contribution is a zero mean Gaussian distributed process. The variance is given by

$$\sigma_{th}^2 = \frac{4\mathrm{kTB}_w}{\mathrm{R}_{th}} \tag{2.22}$$

where B_w is the effective noise bandwidth of the equivalent filter of the system and R_{th} is the thermal resistance of the amplifier. This additive process serves to increase the variance of the output signals presented to the integrator and ultimately to the PPM slot comparator circuit.

2.2.5 Other Error Sources

There are other mechanisms which contribute to PPM word errors. These include timing jitter in the recovered slot and word clocks and bandwidth limitations which result in inter-symbol interference. When channel effects are included, pulse dispersion may contribute to misplacing pulse energy into neighboring empty slots. In the analysis of maximum likelihood PPM receivers, dispersion and ISI have traditionally been treated as an increase in the background noise level in the empty PPM time slots. One purpose of this investigation is to account more exactly for the effects on receiver performance of dispersion or ISI.

As for timing jitter, it has been has suggested that the effect of timing jitter can be investigated simultaneously with the same set of simulations[24]. The procedure is to collect an error histogram based on samples taken at the proper time within a slot and to also sample at times adjacent to this position. Statistics can be obtained as a function of variance from the proper slot sample time in this way and this permits an analysis of timing error performance while all other phenomena remain constant.

2.2.6 Concerning Extinction Ratio

The intent of this discussion is to put down in plain English and math what extinction ratio is and how to measure it as well as what it is not. We present an argument for redefining extinction ratio as the integral of the power in a slot so that dispersion can be looked at in a more realistic fashion.

We have made some references to extinction ratio in the above discussion. Extinction ratio refers to the non-zero intensity level emitted when the laser source is "off" relative to the "on" intensity. In practice, laser diodes are not switched completely on and completely off in a high speed data link. Rather, the output of the laser is switched from a high intensity to a low intensity. The level of light emitted in the "off" condition is a function of the laser diode threshold relative to the bias point, the duty cycle, and the amplitude of the input modulating signal. With PPM, the effect of light being emitted from the laser source during the *empty* slot periods is to raise the background noise level. The question pertinent to this research is: what is the effect of non-zero background level on PPM maximum likelihood receiver performance? To discuss this question in terms of extinction ratio we must make clear our definition of extinction ratio.

Confusion arises because there are two competing definitions of extinction ratio found in the literature. If we define the intensity of the signal during "on" pulses as I_1 and the intensity of the "off" pulses as I_0 then extinction ratio is first defined as the ratio of intensity of the "on" pulse to the intensity of the "off" pulse[11]. Let us denote extinction ratio defined this way as R_1 so that

$$R_1 \equiv \frac{I_1}{I_0}$$

Observe that extinction ratio defined in this way has a valid range from one to infinity, with an infinite extinction ratio implying a large signal to noise ratio.

The second definition of extinction ratio is the inverse of the first definition and is therefore the ratio of the intensity of the "off" pulse to the intensity of the "on" pulse [21][22][25]. Let us denote extinction ratio defined this way as R_2 so that

$$R_2 \equiv \frac{I_0}{I_1}.$$

Observe that extinction ratio defined in this way has a valid range from zero to one, with an extinction ratio of zero representing the ideal case, i.e. implying a large signal to noise ratio. In this work we adopt the second definition.

If BER performance penalty is to be characterized in terms of extinction ratio, then a precise definition of extinction ratio is required. When required we will adopt the second definition above for extinction ratio. This is to avoid the use of quantities which go to infinity in favor of quantities which range form zero to one. Where possible we will use an extended definition of extinction ratio because there is a need to readdress the definition of extinction ratio when referring to photon-counting, that is integrating, optical receiver structures.

Either of the above definitions yields a parameter that is convenient to measure in the lab because it is ostensibly easy to observe on an oscilloscope. The weakness in the above definitions, and the measurement, occur when one considers that the background intensity, that is the intensity in the *empty* slots, is typically not constant over the PPM word. This level becomes a subjective quantity on the part of the person making the measurement for one must make a judgment as to what the "off" level is. The time varying background level may result from channel influences on the background radiation or a time dispersive action from the channel which causes signal energy to spill into the next slot period. An action we might refer to as over-fill. Over-fill can also be a result of incorrect transmitter design, i.e. one with pulses that are too long in duration, or a result of incorrect frame and slot timing recovery in the receiver. The over-filling of a PPM slot will not be readily observed on an oscilloscope as contributing to error since the observed floor of the *empty* slots will appear constant. The value measured and used for extinction ratio will not reflect the true error causing properties of an incompletely extinguished slot period. In other words, over-filling of the PPM slot will not be perceived as contributing to a poor extinction ratio yet this placement of energy into an adjacent slot will contribute to an increase in BER. A better definition of extinction ratio, one which would account for the non-constant background level, would take the ratio of the *integral* of the energy in the PPM "on" slot to some combination of the energy integrated over each of the *empty* PPM slots. Many modern oscilloscopes are capable of making this measurement.

CHAPTER 3
MONTE CARLO SIMULATION

3.0 INTRODUCTION

The options available for the analysis of optical systems might be divided into two broad categories: analysis and simulation. A combination of simulation and analysis, often referred to as a semi-analytical treatment, is sometimes possible. In a semi-analytical simulation only part of the system is simulated with the results used in a computation of the final result. Simulation is often the only method available for the realistic treatment of optical systems owing to the nonlinear nature of the system function and the difficulty inherent in the use of accurate statistical functions for the avalanche photodiode. We therefore present this chapter on Monte Carlo simulation methods which includes a presentation of the basic models used and the fundamentals of importance sampling.

3.1 Monte Carlo Methods

In this chapter we review two Monte Carlo techniques available for the computer simulation of link performance as characterized by the BER of a digital system. A tutorial description of each simulation technique is given beginning with conventional Monte Carlo, what we will refer to as classical Monte Carlo simulation. The presentation defines the notation used in this work and presents the basic principles of this widely used simulation method. Following the discussion of classical Monte Carlo simulation is a tutorial description of the modified Monte Carlo technique known as Importance Sampling. We present these Monte Carlo techniques using as an example the analysis of a simple baseband binary Non-Return to Zero (NRZ) communications system.

For the interested reader, there are at least five techniques for the simulation of communications systems. These techniques are:

1) Monte Carlo simulation
2) Importance Sampling (or modified Monte Carlo simulation)
3) Extreme-Value theory
4) Tail Extrapolation
5) Quasi-Analytical (noiseless simulation combined with analytical representation of noise)

A ready introduction to these techniques can be found in [1] where the application of each to the computer simulation of BER is presented in a tutorial fashion along with pertinent references. An excellent description of importance sampling can be found in [2] which references some of the original work. The *IEEE Journal on Selected Areas in Communications* has published two special issues entitled "Computer-Aided Modeling, Analysis, and Design of Communication Systems" in January 1984 and January 1988. A third issue is scheduled for publication in early 1993.

3.1.1 A Binary NRZ Example

At some level, many binary digital communications receivers can be modeled by the baseband on-off keyed (OOK) system shown in Fig. 3.1. NRZ pulses can be assumed without loss of generality. In this model a pulse of V volts lasting for the whole of one bit period is sent if a "one" is to be transmitted. If a "zero" is to be communicated then no pulse is sent. The received signal is corrupted by noise. This noise may be additive, which is often assumed in the literature, or it may be signal dependent as is the case for optical systems. We have allowed for the possibility of signal dependent noise in a general mathematical sense by denoting the combination of signal and noise through the function $Q(\cdot)$, with the result a signal χ. The signal and noise is then processed by the receiver and equalizer system function $g(\cdot)$. The voltages at the output of the sampling device form a randomly distributed sequence of voltage values $\{\nu_i\}$. The value of ν_i is randomized by the noise processes affecting the signal through the system. The values of ν_i will follow one of two probability densities depending on which signal was sent; i.e. whether the signal is a 'one' or a 'zero'. A representative diagram of these PDF's is shown in Fig. 3.2. In the figure, $f_0(\nu)$ is the probability density function of the voltage level to be examined by the threshold logic device given that a 'zero' was sent; $f_1(\nu)$ is the PDF given that a 'one' was sent. Note that in the signal dependent noise case the probability densities $f_0(\nu)$ and $f_1(\nu)$ may not be of the same shape. Signal dependent noise is the case for optical receivers with APD detectors as discussed earlier.

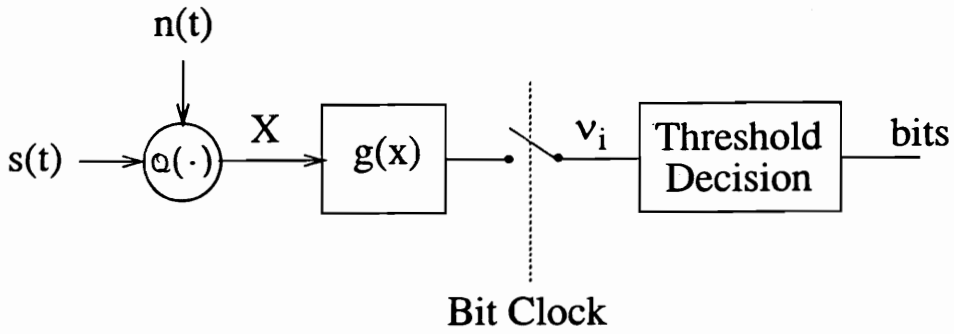


Fig. 3.1 Baseband Binary Communication System Model

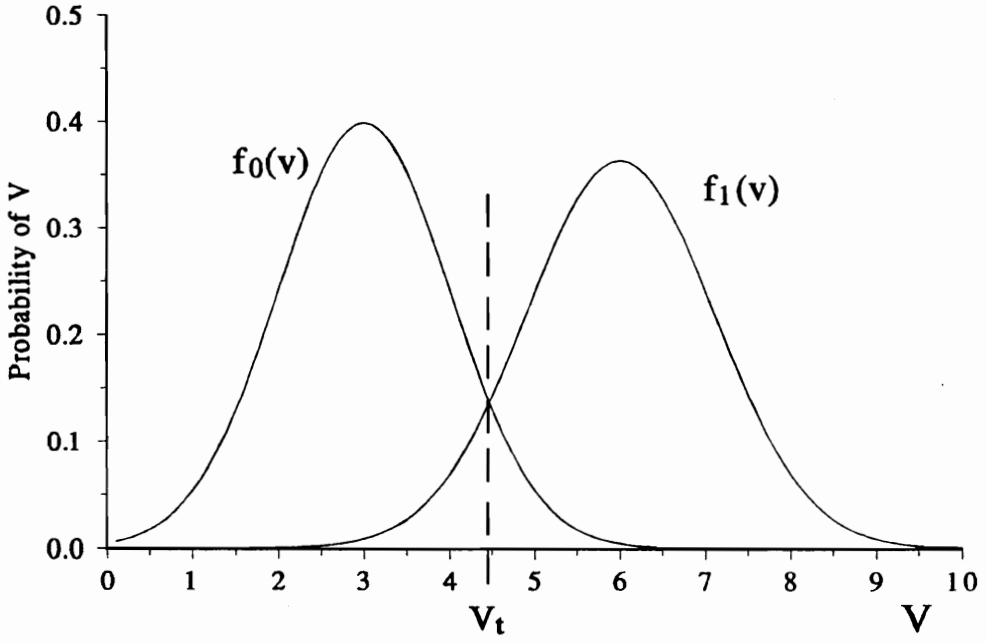


Fig. 3.2 Probability Distribution Functions of the Detected Voltage

where V is the detected voltage, V_T is the threshold voltage, $f_0(v)$ and $f_1(v)$ are the PDF's of the detected voltage given that the signal was a 0 or a 1 respectively.

To summarize, the output random variable ν can be expressed mathematically as a function of the input variables and the system function as:

The hypothesis that s_0 was sent, H_0: $\nu = g(\chi_0) = g[Q(s_0,n)]$ (3.1 a)

The hypothesis that s_1 was sent, H_1: $\nu = g(\chi_1) = g[Q(s_1,n)]$ (3.1 b)

where: s_0 = the 'zero' signal

s_1 = the 'one' signal

n = the noise process

χ = $Q(s,n)$ = the input random process (signal and noise)

and $g(\cdot)$ = the transfer function of the system being simulated

This is a statement of the detection problem in terms of Neyman-Pearson hypothesis testing. The variable χ represents the random signal at the input to the transfer function. We will denote the PDF of this random process as $f(\chi)$.

When simulating such a system on a digital computer the time variable is by necessity discretized and the signals throughout the system are sampled. In truth the signal values are quantized owing to the finite word size of the machine. We are permitted to ignore this fact in most instances due to the great precision available on modern computing machines and will treat signal values as if they were continuous valued yet discretized in time. In the mathematics of the simulation then the sampled value at the threshold decision device is:

$$\begin{aligned} \nu_i &= g[\chi_i, \chi_{i-1}, \chi_{i-2}, \cdots, \chi_{i-K}] \\ &= g[Q(s(i),n_i), Q(s(i\text{-}1),n_{i-1}), Q(s(i\text{-}2),n_{i-2}), \ldots, Q(s(i\text{-}K),n_{i-K})] \end{aligned} \tag{3.2}$$

Expressed as above we have explicitly acknowledged the fact that the detected voltage at time i will, in general, have a contribution from the previous signals input into the system. This is a system with memory. A memory of length K as described by the above equations. This is often the actual condition, a result of filter ringing and other forms of electronic memory inside the receiver. We require this situation to be addressed by a simulation procedure as well. Note that in the importance sampling literature system memory as defined above is often called the system dimension. We will continue to use the word memory in the context above in order to reserve the word dimension to refer to the value of M in M-ary PPM systems.

The conditional PDF's of the detected voltage ν_i will also be a function of the previous samples and can be denoted as:

$$H_0: \quad f_0(\nu_i) = g[f(\chi_{0i}, \chi_{i-1}, \chi_{i-2}, \cdots, \chi_{i-K})]$$
$$= g[f\{Q(s_0(i),n_i), Q(s(i\text{-}1),n_{i-1}), Q(s(i\text{-}2),n_{i-2}), \ldots, Q(s(i\text{-}m),n_{i-K})\}] \quad (3.3\,a)$$

$$H_1: \quad f_1(\nu_i) = g[f(\chi_{1i}, \chi_{i-1}, \chi_{i-2}, \cdots, \chi_{i-K})]$$
$$= g[f\{Q(s_1(i),n_i), Q(s(i\text{-}1),n_{i-1}), Q(s(i\text{-}2),n_{i-2}), \ldots, Q(s(i\text{-}m),n_{i-K})\}]. \quad (3.3\,b)$$

The density $f(\chi_i, \chi_{i-1}, \chi_{i-2}, \cdots, \chi_{i-K})$ is the joint probability density function of the K past signal and noise input samples.

The purpose of the threshold device in the receiver shown in Fig. 3.1 is to determine whether bit_i was a 'one' or 'zero' based upon the examination of the voltage sample ν_i. The technique is to decide that a 'one' was sent if the voltage is above (in this example) a predetermined threshold voltage, V_T, (see Fig. 3.2) and to decide that a 'zero' was sent if the voltage is below this threshold. An error will occur if the voltage is below threshold when a 'one' was sent. An error will also occur if the voltage is above the threshold when a 'zero' was sent. Mathematically the probability of error under each hypothesis is

$$H_0: \quad \text{Prob[error|'zero']} \equiv P_0 = \int_{V_T}^{\infty} f_0(\nu)d\nu = 1-F_0(V_T) \quad (3.4\,a)$$

$$H_1: \quad \text{Prob[error|'one']} \equiv P_1 = \int_{-\infty}^{V_T} f_1(\nu)\, d\nu = F_1(V_T) \quad (3.4\,b)$$

where $F_1(\cdot)$ and $F_0(\cdot)$ are the corresponding cumulative distribution functions. The total probability of error, P_e, is a weighted combination of the above probabilities:

$$P_e = \text{Prob}(0) \cdot P_0 + \text{Prob}(1) \cdot P_1 \quad (3.5)$$

where Prob(0) and Prob(1) are the probabilities of sending a 'zero' and a 'one' respectively. In practice, all data links use techniques that ensure equal numbers of ones and zeros are transmitted. Hence it is usually appropriate to assume Prob(0) = Prob(1) = 0.5. The quantity 'P_e' is otherwise known as the BER.

A computer simulation of such a system can assist in either the determination of the optimum value of V_T or to predict system performance in the form of BER, or both. In our work we are only concerned with the analysis of the BER performance of a system through simulation. In the next section we describe the Monte Carlo simulation of a baseband binary OOK system in which the goal is to determine the BER performance.

3.1.2 Monte Carlo Simulation

This section describes the theory and implementation of Monte Carlo simulation for determination of bit error probability. The emphasis is toward the introduction of a modified form of Monte Carlo simulation known as importance sampling which is presented in Section 3.2.

The use of Monte Carlo simulation in determining the BER is straightforward, albeit time consuming for determination of low probability of bit error. The method is to implement mathematical models of the component parts of a communication system and to form them into a computer program through which sampled signals can be processed. To determine the BER, one must count the number of information bits found to be in error at the detector output, knowing which bits were sent. The BER is simply the ratio of the number of bits in error to the total number of bits transmitted. In other words:

$$\hat{P}_e = \frac{n}{N} \tag{3.6}$$

where n=the number of ν_i observed to cause an error, N=the total number of ν_i observed and $\hat{P}_e$ is the *estimate* of the error probability, P_e. Note that this is simply the fundamental definition of probability and that the estimate will approach the true probability of error as $N \rightarrow \infty$.

In practice the question which needs to be answered is how large must N be given that it must be finite. It can be shown that for a 95% confidence interval, the total number of bits transmitted and examined through the system, N, should be (at a minimum) ten times greater than $\frac{1}{P_e}$ [1]. That is

$$N \geq \frac{10}{P_e} \tag{3.7}$$

where P_e is the probability of error. A 95% confidence interval means that the true probability of error is bounded as: $0.5\hat{P}_e < P_e < 2\hat{P}_e$ with a probability of 0.95. For example, if a BER ($= P_e$) of 10^{-9} is the error rate of the system, then at least 10^{10} bits must be run through the model of the system to have 95% confidence in the result to within $0.5 \times 10^{-9} < P_e < 2 \times 10^{-9}$. The derivation of this result can be found in [1] where the confidence intervals associated with error counting are derived.

The requirement to simulate 10^{10} bits or more through a system to determine one point on a BER curve illustrates the time consuming nature of this type of BER estimation. Its strength is in its ease of implementation and ability to analyze any system, linear or nonlinear,

that can be modeled in a computer program. It is the purpose of importance sampling to reduce the number of bits which must be simulated through the system and yet still achieve an estimate that lies within a desired confidence interval. The generation of a BER curve to include probability of error down to 10^{-6} via a BOSS simulation of a free space optical communications system using QPPM modulation can take between 1 and 2 days on a SUN SPARC workstation. BOSS is a communication system computer simulation package described in Section 3.3.1 below.

In order to introduce importance sampling into the Monte Carlo simulation procedure, we describe the procedure again in a more mathematical fashion. Consider the probability of error given a zero is sent, the H_0 hypothesis. From eq.(3.4 a), we write

$$P_0 = \int_{V_T}^{\infty} f_0(\nu)d\nu \tag{3.8}$$

We concentrate on the probability of error given a zero is sent but the following development applies equally to P_1, the probability of error given a one is sent. Eq.(3.8) can also be written as:

$$P_0 = \int_{-\infty}^{\infty} h_0(\nu) \cdot f_0(\nu)d\nu \tag{3.9}$$

where we define:

$$h_0(\nu) \equiv \begin{cases} 1 & \nu \geq V_T \\ 0 & \nu < V_T \end{cases} \tag{3.10}$$

The function $h_0(\nu)$ is a deterministic function of the random variable ν. Cast in this form, we see from eq.(3.9) that P_0 is the expected value of the function $h_0(\nu)$.

Monte Carlo simulation is simply a discretized version of eq.(3.9) over a finite limit, to the extent that the integral sign is replaced with a summation. Performing this substitution results in a loss of equality which in turn initiates an estimation to the true probability of error as in eq.(3.6). The objective of Monte Carlo estimation is then to generate an estimator of P_0 (the true probability of error) which we denote as $\hat{P}_0$. This estimator is defined as:

$$\hat{P}_0 = \frac{1}{N}\sum_{i=1}^{N} h_0(\nu_i) \tag{3.11}$$

Eq.(3.11), along with a dual statement relating to $\hat{P}_1$, describes the Monte Carlo estimation procedure. It is important to note that the expected value (the mean) of the estimator $\hat{P}_0$ can

be shown to be P_0, ie.

$$E[\hat{P}_0] = P_0. \tag{3.12}$$

Thus the estimator $\hat{P}_0$ is said to be an unbiased estimator. Further, assuming independent errors (this may not be the case for some forms of signal coding) the variance of $\hat{P}_0$ is:

$$\sigma_0^2 = \frac{P_0(1-P_0)}{N} \tag{3.13}$$

where we note that the variance decreases with increasing N. This is actually a restatement of the property observed from eq.(3.6) above. As the variance decreases the estimator is contained closer to the mean. Since the estimator is unbiased, the mean is the desired probability of error, P_0. In other words the estimate can be made to be as close to the actual value as we like simply by increasing N. The penalty is increased computing time.

Eq.(3.13) might also be recast as:

$$\sigma_0^2 = \frac{1}{N}\int_{V_T}^{\infty} f_0(\nu)(1\text{-}P_0)d\nu. \tag{3.14}$$

This form illustrates that the variance of $\hat{P}_0$ might also be decreased if we could find an alternate PDF for the voltage at the threshold device, $f_0(\nu)$. This is the principle behind the technique of importance sampling. Importance sampling is a procedure to decrease the variance of the Monte Carlo estimator via the use of an artificial PDF, say $f_0^*(\nu)$, so that the variance can be made sufficiently small with fewer samples through the system.

3.2 Importance Sampling

3.2.1 Basic Concepts

As introduced at the end of Section 3.1.2, the modified simulation technique called importance sampling is the process of introducing a modified probability density function into eq.(3.9), and ultimately eq.(3.11), to reduce the computation time of the simulation process. The reduced run time is obtained by causing the variance of the estimator to become smaller for a given number of simulation samples, N. In this section we present the theory and practice of importance sampling as applied to a binary NRZ system.

We examine the theory of importance sampling by postulating that we know of a new output PDF which will reduce the variance and thus investigate its effect on the mathematics of

Monte Carlo simulation. If we call this new PDF $f_0^*(\nu)$ (note that the '$*$' does not signify a complex conjugate, only a modified function) then we form directly from eq.(3.9):

$$P_0 = \int_{-\infty}^{\infty} h_0(\nu) \cdot \frac{f_0(\nu)}{f_0^*(\nu)} \cdot f_0^*(\nu) d\nu$$

or

$$P_0 = \int_{-\infty}^{\infty} h_0^*(\nu) \cdot f_0^*(\nu) d\nu \tag{3.15}$$

where:

$$h_0^*(\nu) \equiv h_0(\nu) \cdot \frac{f_0(\nu)}{f_0^*(\nu)} \tag{3.16}$$

The probability of error is now estimated by:

$$\hat{P}_0 = \frac{1}{N} \sum_{i=1}^{N} h_0^*(\nu_i) \tag{3.17}$$

Examination of eq.(3.17) as compared to eq.(3.11) shows that the output sequence $\{\nu_i\}$ is now distributed by the PDF $f_0^*(\nu)$ and that this alteration has been compensated for by an alteration of the counting function $h_0(\nu)$. The counting function is now $h_0^*(\nu)$ which is $h_0(\nu)$ weighted by the ratio of $f_0(\nu)$ to $f_0^*(\nu)$.

This completes the importance sampling procedure. It remains only to show that taking the above actions reduces the variance of the estimator, that the estimator is unbiased (In the sense that the expected value of the estimator is the true value.) and to describe how one might choose a bias function for the input random process. It has been shown repeatedly elsewhere that the importance sampling estimator is unbiased and that following the above procedure will reduce the variance of the estimator, see for example [1] and [3]. As for choosing a bias function, this has been the subject of much of the literature on importance sampling as can be seen in Chapter 4. We discuss some of the basic principles of bias function selection in the remainder of this section and in the next section.

There is some important distinctions which should be made at this time concerning the bias function. The goal is to bias the *output* PDF's in such a way as to favor the ν_i's that cause errors to occur. The PDF at the output is unknown. If it were known there would be no need in performing the simulation since the BER could be calculated directly using eq.(3.4) and eq.(3.5). The principle method of achieving a biased PDF $f_0^*(\nu)$ at the output is by biasing the input samples to the system, that is by altering the PDF of the input random variable, $f(\chi)$. This PDF is known (or assumed) by the user. It forms the input to the simulation. In order to

implement importance sampling we must know (or guess) how the random variables at the input map into the random variables at the output through the system transfer function $g(\chi)$. One must also know which portions of the output PDF to bias in order to promote the occurrence of errors. Stated another way, if the regions of the input variable, χ, which contribute to an important event are known, then the distribution of the input random variable is modified in such a way that more samples are taken from these important regions.

A simple example is shown in Fig. 3.3 in which the PDF $f(\chi)$ is modified to form $f^*(\chi)$. In this example, if $f^*(\chi)$ is used rather than $f(\chi)$, then more samples will come from regions $[x_1, x_2]$ and $[x_3, x_4]$. Notice that this leads to fewer samples from other regions. If an input sample χ is selected from $f^*(\chi)$ then its probability of selection is biased relative to the true probability as described by $f(\chi)$. We define the bias $B(\chi)$ as:

$$B(\chi) \equiv \frac{f^*(\chi)}{f(\chi)} \tag{3.18}$$

Since the output random variable ν is determined from χ, the probability that $\nu=\nu_i$, where $\nu=g(\chi)$, is increased (or decreased) by the bias $B(\chi)$. Therefore the weight, or the significance, of the output sample ν_i is altered by the bias to χ at the input. If the input samples are selected according to the bias $B(\chi)$, then the weight, $W(x)$, of each *output* sample is decreased (or increased) by the weight of the input sample. This weight is

$$W(\chi) \equiv \frac{f(\chi)}{f^*(\chi)} = \frac{1}{B(\chi)}. \tag{3.19}$$

In classical Monte Carlo simulation each input and output has a weight of 1 for all χ. This change in weighting must be accounted for in interpreting the results of the modified simulation.

3.2.2 On Biasing and Unbiasing

The unbiasing of the output random variable is performed as the errors are counted and collected. Eq.(3.17) is a description of the process of unbiasing. Recalling the definition of $h_0^*(\nu)$ from eq.(3.16) we see that since the biased output PDF, $f_0^*(\nu)$, is greater than the unbiased output PDF, $f_0(\nu)$, in the region of $\nu \geq V_T$ then we sum numbers less than one when counting the errors produced in our simulation. Knowing that errors are more likely to occur by design, then they do not count as much. This can be seen mathematically if we write out $h_0^*(\nu)$ as

$$h_0^*(\nu) \equiv \begin{cases} \dfrac{f_0(\nu)}{f_0^*(\nu)} & \nu \geq V_T \\ 0 & \nu < V_T \end{cases} \tag{3.20}$$

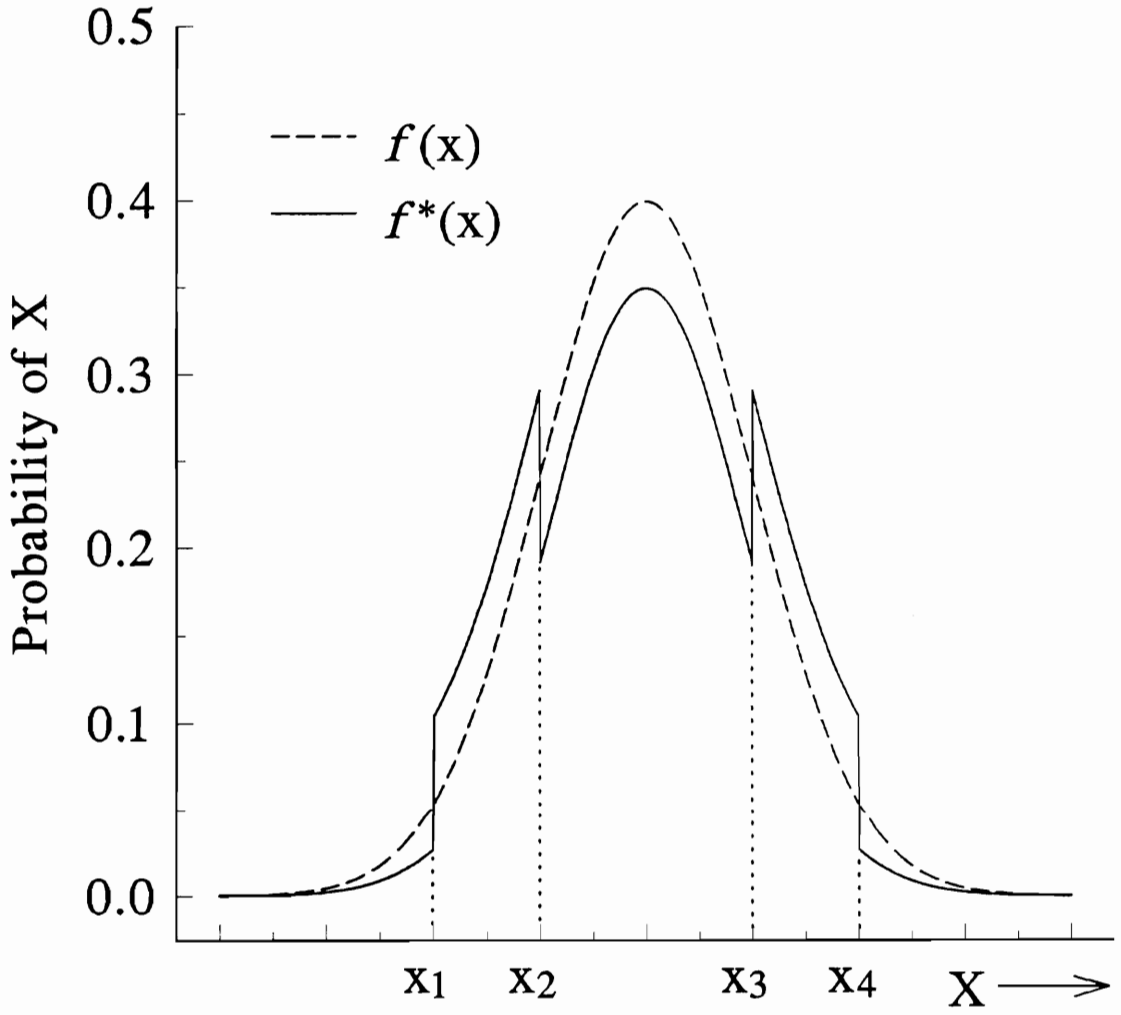


Fig. 3.3 Biasing of a Probability Distribution Function

where the dashed line is the original PDF, $f(x)$, and the solid line is the biased PDF, $f^*(x)$. The probability of producing random variates in the regions $[x_1, x_2]$ and $[x_3, x_4]$ has been increased under the density $f^*(x)$.

What is needed is a method of biasing the PDF of the output random variable ν to insure improvement over classical Monte Carlo. An intuitive feel for why importance sampling works is helpful in this pursuit. Consider the following description.

Insight into the biasing and unbiasing procedure can be found by reconsidering the process for simulating the baseband binary NRZ communications system shown in Fig. 3.1 using Monte Carlo techniques without importance sampling. Recall that in this example n(t) is a Gaussian distributed random signal added to the input signal to form an input random process χ. The goal is to generate a BER curve which describes the BER performance of the system as a function of input signal to noise power. The procedure begins by selecting an input signal to noise power level and running the simulation until the BER performance at that level is known. This provides one point on the BER curve. This procedure is repeated at various input signal to noise levels until there are enough points to determine the curve of BER performance as a function of input signal to noise level.

Let us consider a single simulation process for obtaining an individual point on the BER curve. During the simulation we generate an independent sequence of random numbers representing the sampled version of our noisy input signal $\chi = n(t)+s(t)$, assuming here additive noise for simplicity. This sequence is distributed according to the PDF of the input random process which we have called $f(\chi)$. These samples are processed through the receiver system to generate a new random sequence that we have called ν which has a PDF $f(\nu)$. Thus the receiver processing performs a mapping of the random variable χ into the random variable ν. Using these output samples we could construct an estimate of the PDF of ν in a histogram form. In order to predict performance we must generate enough ν_i's to accurately depict the histogram for that PDF. When we have generated a suitable histogram, we can estimate the probability of error by integrating the PDF above or below threshold as the case may be. This integration is performed by counting the number of samples above (or below) the threshold voltage and dividing by the total number of samples. This process yields an estimate of the integral of the tail of the PDF since the total area under the PDF is one by definition. This is an interpretation of eq.(3.6) or eq.(3.11) that is helpful in understanding the unbiasing process and why importance sampling works as a whole.

Considering Monte Carlo simulation as the generation of a histogram estimate of the output PDF, $f(\nu)$, provides insight into the roll of importance sampling and the technique for biasing and unbiasing of the random variables. For example, traditional Monte Carlo simulation requires that sufficient output samples be generated to estimate the entire PDF $f(\nu)$

to within a desired level of confidence, but in reality we are only interested in estimating the area of $f(\nu)$ under the tails, above and below threshold of the appropriate output density. Importance sampling is a technique for causing more output samples to fall into the tail areas of the PDF histogram. When we know the shape of the tail $f(\nu \mid 0)$ above threshold or the shape of the tail $f(\nu \mid 1)$ below threshold then the generation of output samples can stop.

Consider that the output histogram is generated by dividing the ν axis into a finite number of intervals called bins. Each bin represents a portion of the total probability that we denote as p_j. That is, the probability that the output random variable ν will be in the j^{th} histogram bin $[\nu_{j-1}, \nu_j]$ is p_j. An estimate of this probability, $\hat{P}_j$, is obtained as

$$\hat{P}_j = \lim_{N \to \infty} \frac{n_j}{N} \tag{3.21}$$

where: n_j = the number of samples in the interval $[\nu_{j-1}, \nu_j]$

N = the total number of samples.

In this discussion, the interval $[V_T, \infty)$ in the output histogram can either be thought of as some finite set of bins $[\nu_{j-1}, \nu_j]$ or equivalently that the j^{th} bin spans the entire interval $[V_T, \infty)$. We will generally refer to the interval as the j^{th} bin as either interpretation by the reader will serve.

When constructing the histogram of the output random variable ν for the case of classical Monte Carlo simulation, the weight of all of the n_j that fall into the j^{th} bin $[\nu_{j-1}, \nu_j]$ is one. This is intrinsic to eq.(3.11) above, but when constructing a histogram of the output random variable ν for the case of importance sampling, then the weight of all the n_j that fall into the j^{th} bin $[\nu_{j-1}, \nu_j]$ is given by

$$W_{av[j]} = \frac{1}{n_j} \sum_{i=1}^{n_j} \frac{1}{B(\chi_{ij})} \tag{3.22}$$

where χ_{ij} is the i^{th} input sample that produces an output sample in the j^{th} bin $[\nu_{j-1}, \nu_j]$.

If the biased estimate of the probability in the j^{th} bin is $\hat{P}^*$ then

$$\hat{P}_j^* = \lim_{N \to \infty} \frac{n_j^*}{N} \tag{3.23}$$

where: n_j^* = the weighted (biased) number of samples in the interval $[\nu_{j-1}, \nu_j]$

N = the total number of samples.

The unbiased output for the j^{th} bin will be obtained by multiplying the biased estimate by the

average weight of the bin. That is

$$\hat{P}_j = W_{av[j]} \frac{n_j^*}{N} = \frac{1}{N} \sum_{i=1}^{n_j^*} \frac{1}{B(x_{ij})}$$

or

$$\hat{P}_j = W_{av[j]} \cdot \hat{P}_j^* \tag{3.24}$$

Hence, in order to unbias the histogram we need to sum the weights of each of the ν that caused an error at the output and then divide by N, the total number of ν.

3.2.3 Sample Size Reduction Factor

Let the estimator of the j^{th} bin obtained through modified Monte Carlo simulation be called $\hat{P}_{mj}$ and let the estimator of the j^{th} bin obtained through classical Monte Carlo simulation be called $\hat{P}_{cj}$. Comparing the variance of these estimators, σ_{mj}^2 and σ_{cj}^2 provides an indication of the amount of improvement that can be obtained through the application of importance sampling. We show that the variance of $\hat{P}_{mj}$ can be significantly smaller for a given value of N if the biasing is chosen such that $B(\chi)>1$ for every $x \in I_j$. The interval I_j, possibly disjoint, is the set of all χ such that $\chi \in \{\chi_j |\ \nu_j = g(\chi_j)\}$ where $\nu_j \in [\nu_{j-1}, \nu_j]$.

Recalling eq.(3.13), we have

$$\sigma_{cj}^2 = \frac{p_j(1-p_j)}{N} \approx \frac{p_j}{N} \qquad \text{(approximation for } p_j \ll 1) \tag{3.25}$$

This can be written as

$$\sigma_{cj}^2 = \frac{1}{N_c} \int_{I_j} f(\chi)\, dx \tag{3.26}$$

where N_c is the total sample size for classical Monte Carlo simulation. Similarly the variance of the modified Monte Carlo estimator is

$$\sigma_{mj}^2 \approx \frac{1}{N_m} \int_{I_j} W_{av[j]}(\chi) \cdot f(\chi)\, dx \tag{3.27}$$

where N_m is the total sample size for classical Monte Carlo simulation and $W(\chi)$ is the average weight of the elements in the j^{th} bin.

The variance of the two estimators will be equal for a given confidence in the estimator. That is, for a given error in the estimators

$$\sigma_{cj}^2 = \sigma_{mj}^2$$

$$\frac{1}{N_c}\int_{I_j} f(\chi)\,dx \approx \frac{1}{N_m}\int_{I_j} W_{av[j]}(\chi)\cdot f(\chi)\,dx$$

Hence
$$R = \frac{N_c}{N_m} \approx \frac{\int_{I_j} f(\chi)\,dx}{\int_{I_j} W_{av[j]}(\chi)\cdot f(\chi)\,dx} \tag{3.28}$$

where: R = the sample size reduction factor.

Eq.(3.28) shows that for an equivalent amount of error in the estimation, the ratio of sample sizes will be greater than one; that is, the N_m will be less than N_c, if the x in the interval I_j are biased greater than one: $B(\chi) \gg 1$ or equivalently $W(\chi) \ll 1$.

An example of the use of the modified Monte Carlo procedure as applied to a classic communications system such as the NRZ case used here is given in Section 4 of [2]. The reader is referred there for a numerical example illustrating the procedure as well as numerical comparisons of the reduction in sample size. The authors establish that the reduction in sample size increases with lower values of p, the probability of error. It is shown in [2] that a 10^{-7} probability of error can be estimated using the importance sampling technique with a sample size of $N_m = 10^4$. The accuracy of the estimate is equivalent to the simulation of $N_c = 10^8$ samples taken from an unbiased sample set. A non-zero memory length will inhibit the reduction in sample size but the savings is still significant.

3.3 Optical Receiver Simulation Model

3.3.1 BOSS

The Block Oriented System Simulator, BOSS[4], is a general purpose simulation package designed to run on a DEC VAXStation or a SUN SPARC workstation. The simulation work presented here has been performed using a SUN SPARC II platform running BOSS version 2.7. (Some of the initial work was performed at Goddard Space Flight Center on a SPARC I using version 2.6.) BOSS may be viewed as an operating system for simulation-based analysis and design of communication systems. It oversees all software functions that are necessary to build simulation models, to execute simulations, to view the results and to perform design iterations. BOSS provides an interactive framework for simulation-based analysis and design

with capabilities to develop simulation models in a hierarchical fashion using graphical block diagram representations. It is an example of a software package that is designed to take advantage of the integrated hardware/software environment offered by workstations. Using this block diagram approach the user can configure and execute waveform level simulations, review the results and then perform design iterations.[5]

BOSS is written in LISP and implemented on a workstation to take advantage of the processing power and the integrated hardware/software environment. BOSS assumes that fundamental devices and modules, subsystems and systems can be represented in a hierarchical block diagram form. The topology of a system or subsystem is specified using a graphical block diagram editor. A high resolution graphics terminal is used to display, edit, and interact with block diagrams and to enter parameter values. To construct a new module, the user selects from a set of modules which exist in the module library, places them on the screen and connects them together using the block diagram editor. Library modules are represented on the graphics terminal by icons of blocks with labels. Moving blocks on the screen and connecting them is performed using a mouse as a pointing device. Thus, BOSS eliminates the need for an intermediate language for describing the topology of the system. BOSS encourages a hierarchical approach to model building whereby the user starts with a set of simple building blocks and synthesizes increasingly complex models rapidly. Once a model definition is complete, it is easily added to the model library and can be recalled as a singled functional block at a higher level in the system design. The user can also enter FORTRAN subroutines as models into the BOSS library[5]. This capability has been used in the implementation of much of the importance sampling related building blocks since these do not exist in the BOSS library.

Each module in the BOSS library is in actuality a FORTRAN subroutine. BOSS generates FORTRAN simulation code from the block diagram representation and the simulation is executed with no user intervention. In general, the user needs no knowledge of FORTRAN nor has any indication that a FORTRAN program is being executed. The block diagram representation of the system, run-time parameter values, and the simulation results are all kept together by BOSS in one or more databases. The user can view the simulation results along with block diagrams of the system and run-time parameters in a window environment[5]. Many of the figures in this document are printed BOSS block diagrams and BOSS windows.

3.3.2 APD Model

The physics and the statistics which describe the signal dependent gain process of an

APD have been described in Chapter 2. We now describe the implementation of a complete model of an APD device suitable for implementation on a digital computer.

There are two contributions to the output current of an avalanche photodiode that are modeled. The first is the dark current and the second is the current due to the signal. The dark current is current which flows out of the device even in the absence of incident light. This current consists of two components: the dark current due to surface leakage, i_{DS} and the bulk dark current, i_{DB}. The surface leakage current depends on surface defects in the semiconductor, cleanliness, bias voltage and surface area. A common method of reducing surface dark current is through the use of a guard ring structure during the construction of the device which shunts the surface current away from the load resistor. Since the avalanche multiplication process takes place inside the semiconductor material the surface current is not affected by avalanche gain. The bulk current, on the other hand, arises from electrons and/or holes which are thermally generated in the p-n junction region of the diode. In an APD these carriers get accelerated by the electric field present in the gain region of the device and are thus multiplied by the avalanche gain mechanism[6].

While surface leakage current is a result of the voltage bias across the device, it is relatively insensitive to it. Most importantly it does not experience any multiplicative effects from the gain of the device since it is effectively outside of the device. Surface current is a small contribution to the total output current and is essentially constant. In the modeling of the APD, we have not implemented a surface current term due to its small magnitude and its constant nature. In an integrating receiver such as this, a constant would only provide an offset without contributing to the error performance.

Our APD model is placed in the system so that the input is optical power in watts. This power must be converted to a mean number of incident photons per unit time in order to make use of eq.(2.13). As discussed in the previous section, time is discretized during simulation and we shall denote this unit of time as *dt*. The conversion from optical power to number of photons is obtained through multiplication by a constant. The constant is obtained through manipulation of eq.(2.5) where it is assumed that the input optical power is constant over the time period *dt* so that the integration disappears. Let this constant be called the power constant defined as

$$\text{power constant} = \frac{\lambda \eta \, dt}{\mathrm{h\,C}} \tag{3.29}$$

where C is the speed of light and λ is the wavelength of the incident radiation. The mean

number of photo-electrons per dt is calculated at each simulation sample step from

$$\bar{n} = (\text{power constant}) \cdot \text{power} \tag{3.30}$$

Note that the value of the power constant is constant throughout a given simulation and needs only be calculated once at the beginning. Note also that this implementation does not restrict the optical power to be one of a finite number of values, predetermined before the simulation, so that we are free to introduce channel effects into the simulation.

After obtaining a mean number of photons due to the input optical intensity, the dark current of the device must be added prior to the gain mechanism module. This involves the addition of a constant, the value of which is determined via the specification of the parameters for the APD to be simulated. Specifically the dark constant is

$$\text{dark constant} = \frac{\text{dark current}}{\text{unit charge}} \cdot dt \tag{3.31}$$

This expression gives the number of electrons per unit time dt. Again this value is constant throughout the simulation and need only be calculated once. This value is summed with the mean number of primary photo-electrons from above and the result becomes the parameter $\bar{n}$ used in eq.(2.13). A module to generate random variates as per the WMC density has been constructed inside BOSS using the primitive definition facility.

The block diagram of our APD model is shown in Fig. 3.4. The block titled WMC RANGEN generates random variates as per the WMC density given the input number of mean photo-electrons and parameters for the gain and ionization coefficient. The original topology for this module was provided by Dr. Tony Martino of Goddard Space Flight Center. His model was similar but used an implementation of the exact statistics of eq.(2.7) through eq.(2.12) in place of the WMC random generator. Initially, the input optical power goes through some circuitry to guarantee that the input optical power is positive. This was implemented in order to safeguard against slightly negative values which can occur after the implementation of an FFT prior to the APD. The optical power is multiplied by the power constant and then the dark current term is added. The sum is then input to the random variate generator module WMC RANGEN.

The WMC density given in eq.(2.13) describes the distribution of the number of electrons out of the APD, denoted here and in Chapter 2 as r. For simulation purposes the input is $\bar{n}$, the mean number of primary electrons as found from eq.(2.5) summed with the dark current contribution as described above. The parameters of the expression are κ and G. These

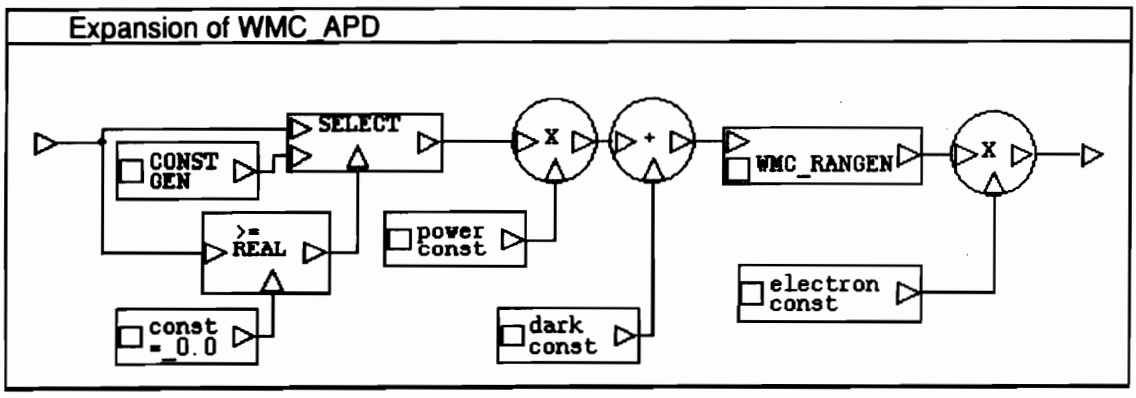


Fig. 3.4 Block Diagram of an Avalanche Photodiode

parameters are described in Section 2.2.2 and are constant throughout the simulation.

An efficient method of generating the random variates according to the WMC density is given by Ascheid[7]. The process is based on the acceptance-rejection method. It is easily implemented on the computer and has been used for the studies here. Briefly, the method begins with a zero mean, unit variance Gaussian random variable, x. The generation of Gaussian random variates is a well established procedure. It is shown in [7] that if a new random variable, z, is defined as a function of x as:

$$z = h(x) = x\left[\frac{x}{2\lambda} + \sqrt{\left(\frac{x}{2\lambda}\right)^2 + 1}\right] \tag{3.32}$$

then this new random variable has a PDF:

$$f_z(z) = \frac{1}{\sqrt{2\pi}} \frac{1+\frac{z}{2\lambda}}{\left(1+\frac{z}{\lambda}\right)^{\frac{3}{2}}} \exp\left\{-\frac{z^2}{2\left(1+\frac{z}{\lambda}\right)}\right\}. \tag{3.33}$$

The WMC PDF is obtained through a simple algebraic maneuver such that:

$$f_{WMC}(y) = 2\,\frac{1}{2+\frac{y}{\lambda}}\, f_z(y) = 2\; g(y)\; f_z(y) \tag{3.34}$$

where y is related to r through the transformation

$$y = \frac{r - \bar{n}\,G}{\sigma} \tag{3.35}$$

The procedure for generating a WMC random variate is as follows:

1) Generate a uniform random number $U \in [0,1]$
2) Generate a normal random variate x as N[0,1]
3) Compute $z = h(x)$ using eq.(3.32)
4) Test for $g(z) \geq U$: if true then accept $y = z$, if not go to 1).

Note that, probably due to a typographical error, the procedure as originally given in [7] incorrectly rejects the variate if g(z) as greater than or equal to U. Otherwise, the procedure is correct as given in [7] and is both simple to implement and efficient. The module WMC RANGEN was created and added to the BOSS library of modules using the BOSS primative module creation procedure.

To summarize, the complete APD model is implemented as follows. First, the value of the mean input, $\bar{n}$, is calculated from the input optical power and the dark current at each

simulation instant *dt*. Each $\bar{n}$ value is applied to the WMC RANGEN module block which generates a WMC distributed random variate via Ascheid's method. The resultant random variable represents the number of electrons out of the APD. This value is converted to a current through multiplication by the unit charge and division by the unit time, *dt*. This conversion to current from electrons is the purpose of the final multiplication shown in the APD model of Fig. 3.4. In our implementation, we convert the number of electrons to microamperes for convenience as follows. The random variate out of the WMC RANGEN block, r, is a number representing the number of electrons out of the APD device over a time *dt* with incident energy P(t). This value is converted to a current, I, by multiplying by the unit charge and dividing by the time period *dt*.

$$I = \frac{q\,r}{dt} \qquad \text{where: } q = 1.6021917\text{x}10^{-19}\ \text{C} \tag{3.36}$$

The variable I represents the amount of current out of the APD over each time interval *dt*.

A comparison test was performed between the APD which uses the WMC generator and the APD originally in use at GSFC which implemented the exact statistics of eq.(2.7) through eq.(2.12). No difference was found in the statistics of the output. A strictly controlled time comparison was not made but it appears that the APD based on the WMC RANGEN using the Ascheid algorithm is faster. This seems reasonable since repeated generations of a random variable for each APD output value r is not performed under Ascheid's technique. The APD using the WMC generator is generally more robust to differing input light levels since the exact implementation becomes increasingly slower as the optical input level becomes greater. At larger input optical powers the number of random variables to be summed, $\bar{n}$, is larger and hence it is slower. The WMC density is held to adequately describe the APD multiplication statistics for large and small values of $\bar{n}$. The speed is also insensitive to the values used for parameters G and κ.

3.3.3 PPM System

Referring to the block diagram of an optical PPM system in Fig. 3.5, a complete description of the simulation procedure follows. The simulation begins with a binary data source: a generator of random one's and zero's. This source feeds a PPM encoder which forms an M-ary PPM word from every L bits of input binary data. The output of the PPM source is a logical level from which optical pulse shapes representing the PPM words are derived.

The samples representing the optical pulse shape are produced by a model of the laser

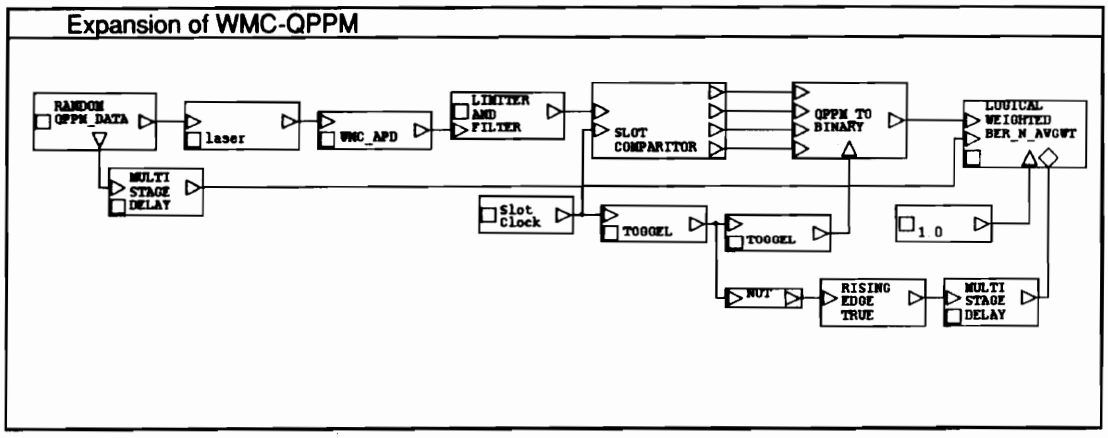


Fig. 3.5 Block Diagram of an Optical PPM System

source. The laser source model generates optical power levels corresponding to the input logical levels. In other words, it maps the input logical levels (*on* or *off*) of the input PPM stream into optical power levels. Parameters for the laser model are the output power level, the extinction ratio of the laser transmitter and the background radiation level. The last two, along with the APD bulk dark current, cause APD output to exist during the empty slot periods. These quantities are predetermined inputs to the simulation. The output pulse shape can easily include the turn on and turn off characteristics of the laser diode[8] but this has not been implemented in the work reported here.

In order to incorporate bandwidth constraints and pulse shape phenomena into the simulation and yet perform the simulation on a discrete time machine, each PPM word is sampled. As a convenience we have ensured that each PPM slot is divided into an integer number of samples. This number is selectable by the user. As before, let the time space between samples be denoted as *dt*. The Nyquist sampling rate must be satisfied in the selection of *dt* if pulse dispersion through the receiver is to be emulated. For square pulses the bandwidth is infinite and therefore cannot be sampled without introducing aliasing. Obviously approximations must be made for the case of pulses with sharp edges. It has been shown that 8 to 16 samples during the period of a rectangular pulse with zero rise time is sufficient to characterize the waveform accurately[9]. The goal is to preserve the amplitude information along the simulation path.

The samples produced by the laser source provide input to a model of the APD. At each time interval *dt*, the APD module generates a number representing the amount of output current (in μA) based on the input optical power for that time interval, *dt*, and the APD parameters as described above. The parameters of the APD model are the ionization ratio, κ, and the average gain, G. The current from the APD is then summed with a zero mean Gaussian distributed random variate with a variance equal to the noise power of the receiver noise temperature. This emulates the additive noise of the preamplifier circuit.

The noisy current samples are passed through a low pass filter that emulates the filter characteristics of the receiver and preamplifier. In the case of an integrating matched filter, the output of the filter is then summed over a slot period, T_s, with the output dumped after each slot. These slot sums are compared in order to determine which PPM word was received. The received word is then converted to binary bits. These detected bits are compared to the transmitted bits (properly delayed) to find transmission errors which are identified and counted. Alternatively, if pulse shaping is used for bandwidth reduction the noisy current samples from

the output of the low pass filter can be passed through a matched filter (square root raised cosine, for example) and the output sampled at the appropriate point in the slot period. In this latter case the two filters can be combined into a single filter element.

3.4 The PPM Problem

The application of importance sampling to M-ary PPM reception utilizing the optimum receiver of Chapter 2 has not previously been attempted. All previous applications of importance sampling have dealt only with threshold detection receivers. The comparison which takes place between received slot energy requires a new and extended definition of bias function realization in order achieve a reduction in run time. The implementation of importance sampling in the simulation of the maximum likelihood optical M-ary PPM receiver is different because the objective of the simulation is more than just the estimation of a tail probability but is the estimation of the integral in M-space inside a closed surface. This closed region represents the region of error probability in much the same way that the tail of a Gaussian represents the region of error probability in our NRZ receiver example above. Further, one of the input processes of this optical system is not additive in nature but multiplicative and signal dependent. Others have dealt with the simulation of APD based receivers but have improved run time only through quasi-analytic techniques and some measure of improved APD simulation.

The successful implementation of IS to M-ary PPM simulation provides an improved analytical tool for the investigation of any number of PPM system phenomena. This simulator can be expanded to explore pointing and tracking effects and other channel phenomena such as turbulence. This should provide a powerful tool to NASA and others working with PPM to explore system issues through simulation. Improved and efficient Monte Carlo simulation of the receiver, transmitter and the dispersive channel will permit the analysis of BER performance bounds not currently possible. Such a simulation can enable such factors as pulse shape to be observed and treated as a variable through the analysis. Further, simulation will permit such analysis as these without defaulting to Gaussian statistics for the APD.

In Chapter 4 we review the previous work in the area of importance sampling and use the results from this previous work dealing with threshold detection to point the way to implementation for the multi-dimensional bias function necessary for M-ary PPM.

CHAPTER 4

APPROACHES TO IMPORTANCE SAMPLING AND OPTICAL SIMULATION

4.0 INTRODUCTION

Importance Sampling (IS) is a modification to Monte Carlo simulation developed by the general statistics and mathematics community to increase the efficiency of Monte Carlo simulation[1][47]. It reduces the number of samples required for a desired accuracy in the estimate and thus reduces the amount of time required to make a thorough study. Interestingly, one of the earliest references to its use in the analysis of a communication system concerned the simulation of an optical system[2]. While our concern here is in the use of IS for optical communications, the majority of work using IS in the communications area has been in contexts other than optical communications. The next importance sampling work in the communications area was four years later[3], co-authored by the author of [2]. Shortly afterwards importance sampling was adopted by the radar community where it was used to speed the simulation of false alarm thresholds in radar systems[4][5]. It has been applied to many areas of communication engineering since then. Increased computing power and greater availability have fueled an increased interest in simulation methods for solving nonlinear communication system problems.

The literature concerning the use of IS in problems of communications engineering can be divided broadly into two main categories. The first is devoted to the derivation or discussion of optimum bias functions, $f^{*}_{opt}(\cdot)$. It should be noted that much of the IS literature devotes some discussion to the choice of bias function. It has been felt, and not without justification, that the choice of bias function is of great significance when implementing the technique. It is seen that the optimum bias function results in a zero variance of the estimator and hence, in theory, only one sample need be simulated. But the optimum bias is unrealizable since its implementation requires foreknowledge of the solution sought through simulation. However, much insight into IS can be obtained through the study of the optimum bias function, so we have chosen to group these works together. This permits the introduction of the concept of an optimum bias function and enables it to be contrasted with the optimum bias function for the

M-ary PPM maximum likelihood receiver found in Chapter 5.

The second category of IS literature consists of works whose primary focus was not the optimum bias function. These works fall into several subcategories. Most of them develop a particular approach to a suboptimum solution to the bias function. There are several of these and each has been developed and used in the simulation of a specific type of communication system problem. Although there are differences in the communication systems considered, they are predominantly systems with additive Gaussian noise and the detection method has been that of threshold detection exclusively. While additive Gaussian noise and threshold detection have been the rule, each of these papers has made some contribution to the understanding of IS in communications system simulation or has developed a unique approach to applying IS to communication system analysis and is therefore of interest to our work.

Additionally, there are some works which have addressed system memory issues. The ability to reduce the number of bits required to be simulated using IS is directly related to the memory of the system. Here, memory refers to the number of prior samples which have influence on the current bit decision. In the literature, this memory is sometimes referred to as the dimensionality of the system in order to prevent confusion with the other concepts of memory. We retain the use of the term memory in order to apply the term dimension to the factor M in our M-ary PPM discussions. Several techniques have been developed to mitigate the effects of memory. Most rely on a linear system model or one that can be linearized with some level of confidence that the resulting linearized system is a suitable approximation to the original nonlinear system. We treat these systems together in order to gain insight into methods of dealing with system memory. Especially as M gets larger, PPM simulation tends to require a large number of samples to represent the PPM waveform. Hence the effects of memory and methods to mitigate its effects are of concern here.

Related to system memory is the issue of multiple noise sources. The optical systems under study here have at least two distinct noise sources in the receiver. These are the non-additive noise of the APD device and the additive Gaussian noise of the preamplifier. If the entire link is considered, random atmospheric effects can be considered as an additional noise source. Thus the ability to apply IS to simulations in which more than one random input exists is of importance. There has been a relatively small amount of work considering more than one random input. It has been principally associated with the simulation of satellite links in which thermal noise is added at the satellite transponder and also at the downlink receiver.

The technique of importance sampling has found much use in other fields of engineering and non-engineering alike where a casual search of the literature will find it applied to problems of quantum physics, simulation of bio-medical systems, and almost any area in which simulation is used to analyze a problem. The literature identified here is that concerned with electronic and communication systems. This is sufficient since the electrical engineering literature is seen to contain all contributions relevant to this study. The purpose of this chapter is to review the literature available concerning importance sampling and to present it in a fashion that permits the reader to quickly understand the current state of the art.

To accomplish our purpose we have divided the literature as described above and now summarize this partitioning. Section 4.1 reviews the derivation of the optimum bias function and related works. This is, by necessity, oriented toward additive Gaussian noise and threshold detection since the maximum likelihood receiver structure has not been considered. The notation used in the summary presented here is general enough to permit the non-additive noise case to follow easily. In Section 4.2, we review each of the versions of importance sampling that occurs in the literature and summarize what makes it distinctive. In most cases, they are distinctive in the approach taken to obtain a suboptimum bias function. Section 4.3 groups those works which make a contribution to two ancillary issues relevant to the implementation of IS. These concern methods of dealing with the deleterious effects of system memory on IS variance reduction and the consequences of multiple input noise sources in the simulation. As an aside, the reader may notice that there is some overlap in each of these groups, as might be expected, since several researchers addressed not only bias function selection but system memory as well. Our groupings, while convenient to this cause, are not recognized as boundaries in the world of importance sampling.

The next section, Section 4.4, is necessary for completeness in this coverage of the IS literature. It groups the various types of systems which have been simulated using IS. It illustrates that the IS approach used is very dependent on the nature of the system being studied. The final section, Section 4.5, is a review of pertinent optical systems simulation efforts. As noted above, most of the body of work concerning importance sampling has been concerned with other than optical systems. Yet some literature exists. None has brought the application of importance sampling to the simulation of APD optical receivers without resorting to Gaussian approximations for the APD model or the use of a look up table for the generation of APD shot noise random variates. Also in this section is a review of optical system simulation in which importance sampling was not used. These works are relevant in that they can be

studied for applicable techniques and models for the simulation of optical components and channels.

4.1 OPTIMUM BIAS FUNCTION

Monte Carlo simulation, both unmodified and modified (i.e. IS), is an estimator of the expected value of the function of a random variable or variables. In the communications system context, the function is often nonlinear. The PDF's of the input random variables are considered the only thing known to the investigator other than a model for the system function. The expected value is estimated by counting the number of occurrences of the output of interest and dividing by the number of trials. The modification to Monte Carlo analysis which is known as IS is simply to skew the input PDF's so that there are more than a *natural* number of occurrences of the outcome of interest, i.e. bit errors in a communications system are the important events. The counting process, in this case, is incremented by a number less than one (the weight) in order that the result be an unbiased estimate of the true (unmodified) output probability. It is the variance of this estimate which is of concern when trying to determine the accuracy of the estimate using either of the Monte Carlo methods.

The variance of the unmodified Monte Carlo estimate is well known as[20][29]

$$\sigma^2_{MC}[\hat{P}_e] = \frac{P_e[1 - P_e]}{N} \tag{4.1}$$

where $\hat{P}_e$ is the estimate of the true expected value P_e, the probability of bit error, and N is the number of trials. As can be seen the variance decreases with order N. It is the purpose of IS to cause the variance to decrease at a rate faster than N^{-1}. When this is achieved then fewer input simulation samples need to be examined for a given error in the estimate. Or conversely, the variance will be smaller and hence the estimate better for the same number of input samples N.

In discussing the optimum biasing function we need to define what is meant by *optimum*. The interpretation has been slightly different for different authors. If we could find a bias function which caused the variance of our IS estimator to be zero then this would seem ideal, if not simply optimum. Imagine if the variance could be made zero, perhaps made zero regardless of the number of trials N, then only one experiment would be required in order to determine $\hat{P}_e$. An estimate with zero variance means that $\hat{P}_e = P_e$ if the estimator is unbiased. We find that such an optimum bias function does exist mathematically but that it is

unrealizable. It is unrealizable because it requires one to know the answer being sought in order to construct the bias function. Yet study of this optimum bias function can indicate some the the pertinent properties to look for in a suboptimum choice of bias function. This is the subject of Section 4.1.1 below.

If we can't have the optimum bias function described in the above paragraph then a realizable alternative needs to be sought. The unrealizable optimum solution is derived in an unconstrained sense. The derivation is unconstrained in that there are no mechanisms built into the derivation which will eliminate the unrealizable optimum from the set of possible functions derived as *optimum*. To eliminate the possibility of the unrealizable optimum function from the derivation is to constrain the derivation. Two methods of constraining the derivation in order to achieve a suboptimum bias function are discussed in Section 4.1.2.

The optimum bias function is approached in a different light in Section 4.1.3. Here the bias function, and IS in general, is treated in a sequence-dependent fashion. That is, the simulation is carried out for each input bit sequence that is possible under the system memory. For example, if the current bit decision is influenced by not only the current signal but also the signals from the five previous bits then then there are $2^5 = 32$ different bit sequences to be tested since each sequence will produce a different result, in general. The BER of the entire system is then found by determining the BER for each possible input sequence and averaging the result. IS and bias function determination are treated in this context in Section 4.1.3 and we show that the entire concept is of importance when analyzing systems with ISI contributions to the output error rate.

The final section, Section 4.1.4, presents some interesting and useful phenomena that have been developed as researchers have worked through the mathematics of IS.

4.1.1 Unconstrained Optimum Bias Derivation

The optimum bias function sought is one which will minimize the variance of the IS estimator. We first present a mathematical description of the IS BER estimation which is both general and rigorous. Similar derivations abound in the literature[21][39][29]. The purpose in repeating it here is to provide a vehicle for the presentation of the optimum bias function and to do it using notation which will carry over to the non-additive noise case which is typical for optical communications. Most of the derivations in the literature are not general since they inherently assume additive noise. This presentation is important not only for understanding

what has been done but also for what has been added to the understanding by this work.

In M-ary PPM the receiver must decide between M hypotheses. Let us consider the signal detection problem in which we wish to decide between two hypotheses, M = 2,

$$H_0\text{: } \mathbf{R} = g[\mathcal{Q}(\mathbf{s}_0, \mathbf{n})]$$

and

$$H_1\text{: } \mathbf{R} = g[\mathcal{Q}(\mathbf{s}_1, \mathbf{n})]. \tag{4.2}$$

where $\mathbf{R}$ is the K-dimensional receive vector, $\mathbf{s}_0$ and $\mathbf{s}_1$ are the respective K-dimensional signal vectors and $\mathbf{n}$ is the K-dimensional input noise vector. K represents the memory of the system since it is seen that it takes K input samples to produce an output vector. The decision process is based on some manipulation or attribute of the receive vector $\mathbf{R}$. $\mathcal{Q}$ is the function which combines the input vectors in order to generate the input to the system transfer function $g(\cdot)$. Often $\mathcal{Q}(\cdot)$ is assumed additive. The system transfer function may or may not be linear. This system arrangement is portrayed symbolically in Fig. 4.1.

Define $\Omega_0 \subseteq \mathbf{R}^K$ to be the region of $\mathbf{R}^K$ space where hypothesis H_0 is chosen. Similarly, $\Omega_1 \subseteq \mathbf{R}^K$ to be the region where hypothesis H_1 is chosen. The average probability of a wrong decision can be written as

$$P_e = \Pi_0 \int_{\Omega_1} f_{\boldsymbol{R}\,|\,H_0}(\mathbf{r})\, d\mathbf{r} + \Pi_1 \int_{\Omega_0} f_{\boldsymbol{R}\,|\,H_1}(\mathbf{r})\, d\mathbf{r} \tag{4.3}$$

where Π_0 and Π_1 are the *a priori* probabilities that H_0 and H_1 are true. The probability density functions $f_{\boldsymbol{R}\,|\,H_0}(\mathbf{r})$ and $f_{\boldsymbol{R}\,|\,H_1}(\mathbf{r})$ are the density functions of the received vector conditioned on the hypotheses H_0 and H_1. In general the detection system in direct detection optical communications receivers is asymmetric so that the PDF's of the receive vectors are asymmetric.

To simplify notation, we proceed with only one half of the the RHS of eq.(4.3), that is for only one hypothesis. The developments are easily applied to the other hypothesis.

$$P_{e\,|\,k} = \Pi_k \int_{\Omega_k} f_{\boldsymbol{R}\,|\,k}(\mathbf{r})\, d\mathbf{r} \qquad \text{for } k = 0 \text{ or } 1 \tag{4.4}$$

At this point we normalize by the probability weighting factor Π_k and make it clear that in the remainder of this development we are discussing the probability of error for one symbol, 0 or 1 in this binary case. This is the symbol probability of error, i.e. for a single hypothesis. This permits us to drop the conditional subscript, realizing that it is simply part of the total system error probability. We do this with no loss of generality since each part can be developed

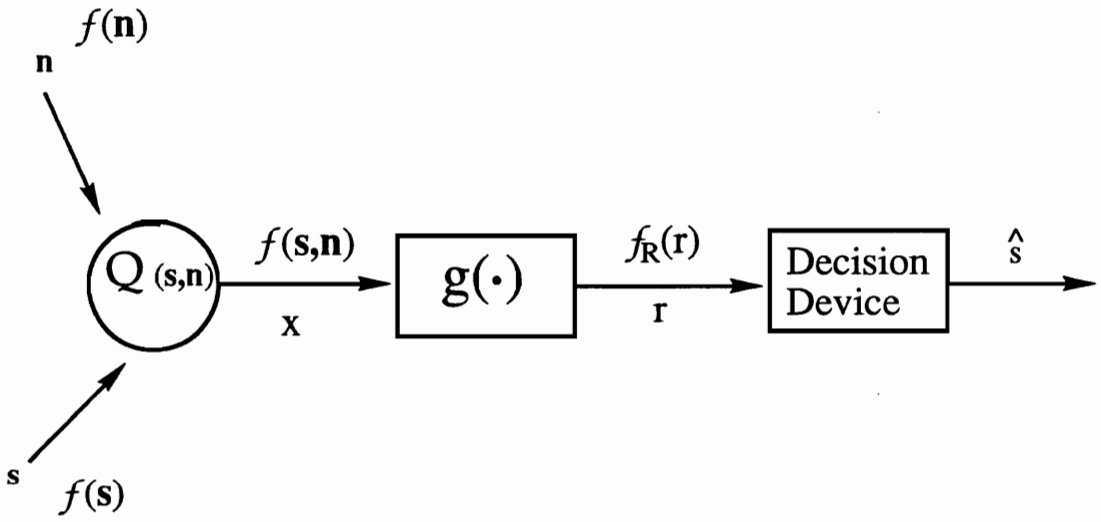


Fig. 4.1 Symbolic System Model

where n is the input noise process which is distributed according to the density $f(n)$ and s is the input signal process which is distributed according to the density $f(s)$. These are combined at the input of the receiver through the function $Q(s, n)$ to form the random variable x. The system transfer function is $g(\cdot)$ so that $g(x) = r$. The random variable r is distributed according to the probability density function $f_R(r)$ and is the random variable examined by the decision device in order to decide which signal was sent. This detected signal is $\hat{s}$.

separately from this point forward.

The probability of error under a single hypothesis, the symbol error, is then

$$P_e = \int_{\Omega} f_R(\mathbf{r})\, d\mathbf{r} \tag{4.5}$$

where Ω is the subspace of $\mathbf{R}^K$ defined by all possible values of $\mathbf{r}$ that result in a symbol error. As each output vector $\mathbf{r}$ is a direct consequence of input vectors $\mathbf{s}$ and $\mathbf{n}$ and since we will be biasing only the noise vectors and not the signal vectors, let us rewrite the probability of error in terms of the input vectors and the joint probability density function of $\mathbf{s}$ and $\mathbf{n}$.

$$P_e = \int_{\Omega} f(\mathbf{s}, \mathbf{n})\, d\mathbf{s}\, d\mathbf{n} \tag{4.6}$$

It is important to distinguish the densities and their implications here. Referring to Fig. 4.1, the density function $f(\mathbf{s}, \mathbf{n})$ is the joint probability density function of the signal and noise vectors after they have been combined via the process $\mathcal{Q}$. In the additive noise case, this is generally a simple translation of the input noise PDF. In the optical case this is not so since the noise is signal dependent as discussed in Chapter 2. The region $\tilde{\Omega}$ is that region of the $\mathbf{X}^K$ system *input* space that results in a bit error. The PDF of the received signal vector, $f_R(\mathbf{r})$, is the PDF at the system output before the detector, for example at the threshold decision device input. The region Ω is the region of the $\mathbf{R}^K$ system output space, the region of all possible vectors $\mathbf{r}$, in which a bit error occurs. Mathematically, the receive vector $\mathbf{r} = g[\mathcal{Q}(\mathbf{s}, \mathbf{n})]$ where g is the system transfer function.

Define an indicator function

$$H(\mathbf{s}, \mathbf{n}) = \begin{cases} 1 & (\mathbf{s}, \mathbf{n}) \in \Omega \\ 0 & (\mathbf{s}, \mathbf{n}) \notin \Omega \end{cases} \tag{4.7}$$

Eq.(4.6) can now be written as

$$P_e = \int_{\mathbf{X}^K} H(\mathbf{s}, \mathbf{n}) f(\mathbf{s}, \mathbf{n})\, d\mathbf{s}\, d\mathbf{n} \tag{4.8}$$

Eq.(4.8) is an expression defining P_e as the mean, the expected value, of the function $H(\mathbf{s}, \mathbf{n})$. This is the basis for unmodified (classical) Monte Carlo simulation since the estimator for this mean is

$$\hat{P}_e = \frac{1}{N} \sum_{i=1}^{N} H(\mathbf{s}_i, \mathbf{n}_i) \tag{4.9}$$

where N is the number of trials of the simulation. The vectors $\mathbf{s}_i$ and $\mathbf{n}_i$ are the sampled input vectors which are independent and identically distributed. The estimate $\hat{P}_e$ can easily be shown to be an unbiased estimator, i.e. $E[\hat{P}_e] = P_e$, with a variance

$$\sigma^2_{MC}[\hat{P}_e] = \frac{P_e[1-P_e]}{N} \tag{4.10}$$

which is true for independent input samples. For a discussion of the case when the inputs cannot be considered independent see [24].

For importance sampling, or modified Monte Carlo, simulation we write, from eq.(4.8)

$$P^*_e = \int_{\mathbf{X}^K} H(\mathbf{s}, \mathbf{n})\, W(\mathbf{s}, \mathbf{n})\, f^*(\mathbf{s}, \mathbf{n})\, d\mathbf{s}d\mathbf{n} \tag{4.11}$$

where $f^*(\mathbf{s}, \mathbf{n})$ is a joint PDF over the space $\mathbf{X}^K$ and

$$W(\mathbf{s}, \mathbf{n}) = \frac{f(\mathbf{s}, \mathbf{n})}{f^*(\mathbf{s}, \mathbf{n})} \tag{4.12}$$

is the weight. If the simulation is built so that $f^*(\mathbf{s}, \mathbf{n})$ governs the input random process, the estimator for P_e becomes

$$\hat{P}^*_e = \frac{1}{N^*} \sum_{i=1}^{N^*} H(\mathbf{s}_i, \mathbf{n}_i)\, W(\mathbf{s}_i, \mathbf{n}_i) \tag{4.13}$$

where N^* the number of simulation trials. This estimator is unbiased because

$$E_*[\hat{P}^*_e] = \int_{\mathbf{X}^K} \left(\hat{P}^*_e\right) f^*(\mathbf{s}, \mathbf{n})\, d\mathbf{s}d\mathbf{n} \tag{4.14}$$

$$= \int_{\mathbf{X}^K} \left[\frac{1}{N^*} \sum_{i=1}^{N^*} H(\mathbf{s}_i, \mathbf{n}_i)\, W(\mathbf{s}_i, \mathbf{n}_i) \right] f^*(\mathbf{s}, \mathbf{n})\, d\mathbf{s}d\mathbf{n} \tag{4.15}$$

$$= \frac{1}{N^*} \sum_{i=1}^{N^*} \int_{\mathbf{X}^K} H(\mathbf{s}_i, \mathbf{n}_i)\, W(\mathbf{s}_i, \mathbf{n}_i)\, f^*(\mathbf{s}, \mathbf{n})\, d\mathbf{s}d\mathbf{n} \tag{4.16}$$

$$= \frac{1}{N^*} \sum_{i=1}^{N^*} \int_{\mathbf{X}^K} H(\mathbf{s}_i, \mathbf{n}_i)\, f(\mathbf{s}, \mathbf{n})\, d\mathbf{s}d\mathbf{n} \tag{4.17}$$

$$= \frac{1}{N^*} \sum_{i=1}^{N^*} P_e = P_e \tag{4.18}$$

The variance of the IS estimator is found from

$$\sigma_{IS}^2[\hat{P}_e^*] = E_*\left\{(\hat{P}_e^*)^2\right\} - E^2\left\{\hat{P}_e^*\right\} \tag{4.19}$$

The first term on the RHS can be simplified as

$$E_*\left\{(\hat{P}_e^*)^2\right\} = E_*\left\{\left[\frac{1}{N^*}\sum_{i=1}^{N^*} H(\mathbf{s}_i, \mathbf{n}_i)\, W(\mathbf{s}_i, \mathbf{n}_i)\right]^2\right\} \tag{4.20}$$

$$= \left[\frac{1}{N^*}\right]^2 \left[\sum_{i=1}^{N^*} E_*\left\{H^2(\mathbf{s}_i, \mathbf{n}_i)\, W^2(\mathbf{s}_i, \mathbf{n}_i)\right\} + \sum_{\substack{i,j=1 \\ i \neq j}}^{N^*} E_*\left\{H(\mathbf{s}_i, \mathbf{n}_i)\, W(\mathbf{s}_i, \mathbf{n}_i)\, H(\mathbf{s}_j, \mathbf{n}_j)\, W(\mathbf{s}_j, \mathbf{n}_j)\right\}\right] \tag{4.21}$$

$$= \left[\frac{1}{N^*}\right]^2 \left[N^*\, E_*\left\{H^2(\mathbf{s}_i, \mathbf{n}_i)\, W^2(\mathbf{s}_i, \mathbf{n}_i)\right\} + \sum_{\substack{i,j=1 \\ i \neq j}}^{N^*} E_*\left\{H(\mathbf{s}_i, \mathbf{n}_i)\, W(\mathbf{s}_i, \mathbf{n}_i)\, H(\mathbf{s}_j, \mathbf{n}_j)\, W(\mathbf{s}_j, \mathbf{n}_j)\right\}\right] \tag{4.22}$$

$$= \left[\frac{1}{N^*}\right]^2 \left[N^* \int_{\mathbf{X}^K} H^2(\mathbf{s}_i, \mathbf{n}_i)\, W^2(\mathbf{s}_i, \mathbf{n}_i) f^*(\mathbf{s}, \mathbf{n})\, d\mathbf{s}d\mathbf{n}\right] +$$

$$\left[\frac{1}{N^*}\right]^2 \left[N^*[N^* - 1]\left(P_e\right)^2 + \sum_{\substack{i,j=1 \\ i \neq j}}^{N^*} \mathrm{COV}_*\left\{H(\mathbf{s}_i, \mathbf{n}_i)\, W(\mathbf{s}_i, \mathbf{n}_i)\, H(\mathbf{s}_j, \mathbf{n}_j)\, W(\mathbf{s}_j, \mathbf{n}_j)\right\}\right] \tag{4.23}$$

where we have used $E[XY] = E[X]E[Y] + COV[XY]$ and COV means covariance. Noting that $H^2(\mathbf{s}_i, \mathbf{n}_i) = H(\mathbf{s}_i, \mathbf{n}_i)$ we continue as

$$= \left[\frac{1}{N^*}\right]^2 \left[N^* \int_{\mathbf{X}^K} H(\mathbf{s}_i, \mathbf{n}_i)\, W(\mathbf{s}_i, \mathbf{n}_i) f(\mathbf{s}, \mathbf{n})\, d\mathbf{s}d\mathbf{n}\right] +$$

$$\left[\frac{1}{N^*}\right]^2 \left[N^*[N^*\text{-}1]\left(P_e\right)^2 + \sum_{\substack{i,j=1 \\ i \neq j}}^{N^*} \mathrm{COV}_*\left\{H(\mathbf{s}_i, \mathbf{n}_i)\, W(\mathbf{s}_i, \mathbf{n}_i)\, H(\mathbf{s}_j, \mathbf{n}_j)\, W(\mathbf{s}_j, \mathbf{n}_j)\right\}\right] \tag{4.24}$$

$$= \frac{1}{N^*} \int_{\mathbf{X}^K} H(\mathbf{s}_i, \mathbf{n}_i)\, W(\mathbf{s}_i, \mathbf{n}_i) f(\mathbf{s}, \mathbf{n})\, d\mathbf{s}\, d\mathbf{n} + \left[1 - \frac{1}{N^*}\right] \left(P_e\right)^2 +$$

$$\left[\frac{1}{N^*}\right]^2 \sum_{\substack{i,j=1 \\ i \neq j}}^{N^*} \mathrm{COV}_* \left\{ H(\mathbf{s}_i, \mathbf{n}_i)\, W(\mathbf{s}_i, \mathbf{n}_i)\, H(\mathbf{s}_j, \mathbf{n}_j)\, W(\mathbf{s}_j, \mathbf{n}_j) \right\} \quad (4.25)$$

Substituting the above expression into eq.(4.19) for the variance and noting that

$$-\frac{\left(P_e\right)^2}{N^*} = -\frac{P_e}{N^*} \int_{\mathbf{X}^K} H(\mathbf{s}_i, \mathbf{n}_i) f(\mathbf{s}, \mathbf{n})\, d\mathbf{s}\, d\mathbf{n} \quad (4.26)$$

then the variance of the IS estimator is

$$\sigma_{IS}^2[\hat{P}_e^*] = E_* \left\{ (\hat{P}_e^*)^2 \right\} - \left\{ P_e \right\}^2 \quad (4.27)$$

$$= \frac{1}{N^*} \int_{\mathbf{X}^K} H(\mathbf{s}_i, \mathbf{n}_i) \left[W(\mathbf{s}_i, \mathbf{n}_i) - P_e^* \right] f(\mathbf{s}, \mathbf{n})\, d\mathbf{s}\, d\mathbf{n} +$$

$$\left(\frac{1}{N^*}\right)^2 \sum_{\substack{i,j=1 \\ i \neq j}}^{N^*} \mathrm{COV}_* \left\{ H(\mathbf{s}_i, \mathbf{n}_i)\, W(\mathbf{s}_i, \mathbf{n}_i)\, H(\mathbf{s}_j, \mathbf{n}_j)\, W(\mathbf{s}_j, \mathbf{n}_j) \right\} \quad (4.28)$$

If $H(\mathbf{s}_i, \mathbf{n}_i)\, W(\mathbf{s}_i, \mathbf{n}_i)$ and $H(\mathbf{s}_j, \mathbf{n}_j)\, W(\mathbf{s}_j, \mathbf{n}_j)$ are independent for $i \neq j$ then

$$\sigma_{IS}^2[\hat{P}_e^*] = \frac{1}{N^*} \int_{\mathbf{X}^K} H(\mathbf{s}_i, \mathbf{n}_i) \left[W(\mathbf{s}_i, \mathbf{n}_i) - P_e \right] f(\mathbf{s}, \mathbf{n})\, d\mathbf{s}\, d\mathbf{n} \quad (4.29)$$

In the simulation of optical communications systems the input samples can usually be considered independent. Some forms of coding can create a condition in which the input samples are not independent from bit to bit.

Eq.(4.29) is an expression for the variance of the IS estimator and as such has been the starting point for various analytical journeys into IS methodology by many researchers. They have considered the subtleties of this equation and found a variety of ways to interpret and utilize its message. We explore many of these things in the following paragraphs.

Without further ado, the optimum bias function can be found by inspection of eq.(4.29). If the optimum bias is to be found from minimizing the variance then it can be seen that the variance can be made zero if

$$f_{opt}^*(\mathbf{s}, \mathbf{n}) = \frac{H(\mathbf{s}_i, \mathbf{n}_i) f(\mathbf{s}, \mathbf{n})}{P_e} \quad (4.30)$$

Eq.(4.30) might be rewritten as

$$\overset{*}{f}_{opt}(\mathbf{s},\,\mathbf{n}) = \begin{cases} \dfrac{f(\mathbf{s},\,\mathbf{n})}{P_e} & (\mathbf{s},\,\mathbf{n}) \in \tilde{\Omega} \\ 0 & \text{otherwise} \end{cases} \tag{4.31}$$

This latter form enables one to better examine the salient features of the optimum solution.

This result was presented in [21] where it was presumably arrived at by inspection just as it was here, in [29] where it was arrived at through minimization of the variance using the calculus of variations, in [39] by applying the Lagrange multiplier method and in [36] where it was arrived at through the use of Jensen's inequality on the first term on the RHS of eq.(4.23). The result briefly appears in many works, particularly those discussing bias function issues, but the above references represent some of the more noteworthy.

First let us discuss the obvious paradoxes which make this the trivial solution to the optimum bias problem. To construct the optimum PDF we must know the region $\tilde{\Omega}$ and the area in that region under the joint PDF $f(\mathbf{s},\,\mathbf{n})$. This is P_e. The value of P_e is precisely the value which is to be estimated! The true importance region $\tilde{\Omega}$, or its counterpart Ω, is generally unknown at the start of simulation process as well. It is often known at the output of the system, in terms of the threshold, for example, but at the output the PDF(s) of the signal(s) involved are not known. If they were known there would be no need in simulating the system.

The notable features of the optimum bias function are its the shape and its region of existence. It seems significant that $\overset{*}{f}_{opt}(\mathbf{s},\,\mathbf{n})$ should have the same shape as the unmodified PDF in the region of importance and zero elsewhere. It has the same shape but is scaled by a constant. The result is that the weight for any $(\mathbf{s},\,\mathbf{n})$ pair will be a constant – the desired P_e it turns out. With a zero variance, then only one trial is required but the weight used in the error count of that trial will be the desired answer. Further, we are guaranteed that if $\overset{*}{f}_{opt}(\mathbf{s},\,\mathbf{n})$ is made to govern the simulation effort then every trial will result in a significant event, a bit error in this case, since only $(\mathbf{s},\,\mathbf{n})$ pairs in the error region of the sample space $\mathbf{X}^k$ will be generated using this joint PDF. This brings up the general point, that every simulation trial which does not produce an important event, a bit error, is a wasted effort because no real information is added to the knowledge of the system's error behavior.

In summary, the following things have been identified by researchers in the literature as the important aspects of the optimum bias function. First, it should resemble the nonbiased

PDF as much as possible in the significant region. To do so means that the weights generated are constant and small, i.e. close to the desired probability of error value. Further the optimum PDF should be as small as possible, if not zero, outside the important region. This will promote the generation of output errors and decrease the simulation time. These features of the optimum bias have been noticed by previous researchers and various means have been employed in order to arrive at, or approximate, this function. These are presented in Section 4.2 below.

4.1.2 Constrained Optimum Bias Derivation

As seen in Section 4.1.1, an unconstrained mathematical derivation for the optimum IS bias function yields a solution that while ideal is unrealizable. Recognizing this, some attempts have appeared in the literature to derive an optimum bias function while constraining the possible outcomes to not include the unrealizable optimum. The results are uninspiring. The first uses Large Deviation Theory to find a function or class of functions which are asymptotically efficient[36]. The second attempts to minimize the variance of the IS estimator by trying to find the density function which is closest to the unconstrained optimal solution in the Ali-Silvey sense[30]. A discussion of and the results from each of these works is presented in this section.

An asymptotically efficient probability density function for biasing an IS simulation would be one for which the computational burden of estimation grows less than exponentially fast. If you consider that the accuracy of Monte Carlo estimation improves as $\frac{1}{N}$ then the goal, or the constraint to be used in the derivation of a suboptimum IS bias function, would be to cause the accuracy to improve at a faster rate. To obtain an asymptotically efficient simulation a class of "exponentially twisted" (otherwise known as "tilted") densities from large deviations theory were considered in [36]. Of practical interest, they consider the optimum bias derivation considering the multidimensional input case. They arrive at an *asymptotically efficient exponentially twisted* density which is, in general, made of convex combinations of exponentially twisted densities. The form of the combination is governed by the dominating point of the system to be simulated. In the linear ISI case this solution agrees with the shifting method of Lu and Yao[25] who describe a method of forming a biased input PDF by "shifting" the original input density. This is often referred to as *improved importance sampling* and is presented in Section 4.2.3 below. The important point is that shifting of the input PDF's can be asymptotically efficient. They demonstrate that Scaling Importance Sampling (SIS), i.e. scaling to increase the variance, and Efficient Importance Sampling (EIS) are NOT asymptotically

efficient. The methods SIS and EIS are discussed in detail in Sections 4.2.2 and 4.2.7 respectively.

While this work presents a method of forming suboptimum solutions which are optimum in the asymptotically efficient sense, a deeper study of Large Deviation Theory is required for proficient utilization of these results to the biasing of signal dependent APD noise in the M-PPM problem of concern here. They discuss two particular types of problems which are helpful. The first is independent identically distributed (i.i.d.) sums or Markov sums. The second is the simulation of systems with small Gaussian noise inputs. An example considering the simulation of a digital communications channel with nonlinear ISI is presented.

A general framework for determining a constrained optimal solution from any constraint set is presented in[30]. The technique requires one to find a class of functions which is least favorable in terms of Bayes risk from sets of functions that are closest to the optimum solution in the Ali-Silvey sense. While this method is the most general found it is too complicated to produce any practical results in the span of one person's life time.

4.1.3 Sequence-Dependent Derivation

The optimum bias function can be approached in yet another manner. In a digital system with ISI and a memory greater than one symbol, the analysis and derivation of the optimum (and suboptimum) bias function and the simulation itself can be considered separately for each symbol sequence through the system[40]. Let m be the memory of the system in number of symbols. This can be related to K, the memory of the system in terms of simulation samples, by

$$\mathrm{K} = \frac{m\mathrm{T}_s}{dt} \tag{4.32}$$

where T_s is the symbol duration and dt is the sampling interval. If M is the number of different symbols, then given that the first symbol is fixed, this is the conditioning symbol in the subsequent analysis, there are $\mathrm{L} = \mathrm{M}^{m-1}$ possible realizations of **s**. For clarity, let us denote **s** explicitly as

$$\mathbf{s} = \{\mathrm{s(t)}, \mathrm{s(t} - dt), \mathrm{s(t} - 2dt), \ldots, \mathrm{s(t} - [\mathrm{K} - 1]dt)\} \tag{4.33}$$

While the work in [40] does not lay to rest the method of optimum bias for IS simulation any better than any of the other papers discussed in this chapter, it does present some noteworthy concepts for consideration.

First is a concept that the authors refer to as the "System Threshold Characteristic" (STC). The STC is a function created out of the analysis of a system which contains the ISI characteristics of the system. It can be looked upon as a conditional probability density function which describes the error probabilities of the system around threshold conditioned on the noise vector **n**. It is actually first introduced and used by these authors in [17] wherein more effort is placed in defining STC than in [40]. If one considers the distribution of voltages at the input to the decision device, even in the absence of noise in the system the voltage is still random due to ISI effects - the output being different for each randomly occurring signal sequence. When the distribution of the noise at the decision device input is considered as well then the result is the voltage distribution normally considered at that point in the system. The STC is a distribution conditioned on a particular noise voltage at a particular point in the system. It can be understood by considering the usual error analysis of a communications system as follows.

We begin by considering the probability of error, P_e, much as before.

$$P_e = \int_{-\infty}^{\infty} H[g(\mathbf{s}, \mathbf{n})] f(\mathbf{s}) f(\mathbf{n}) \, d\mathbf{s} \, d\mathbf{n} \tag{4.34}$$

where

$$\mathbf{s} = \{s(t_i), s(t_i - dt), s(t_i - 2dt), \ldots, s(t_i - [K-1]dt)\} \tag{4.35}$$

$$\mathbf{n} = \{n(t_i), n(t_i - dt), n(t_i - 2dt), \ldots, n(t_i - [K-1]dt)\} \tag{4.36}$$

$H(\cdot)$ is the indicator function as before and $g(\cdot)$ is the system function in which we have included the signal and noise combining function $\mathcal{Q}$ for notational simplicity. Note also that we have implicitly assumed that the signal and noise are independent in the above formulation.

The STC is simply the integral over **s** in eq.(4.34), denoted $T(\mathbf{n})$

$$T(\mathbf{n}) = \int_{-\infty}^{\infty} H[g(\mathbf{s}, \mathbf{n})] f(\mathbf{s}) \, d\mathbf{s} \tag{4.37}$$

It thus embodies the ISI characteristics of the system. We might now write eq.(4.34) as

$$P_e = \int_{-\infty}^{\infty} T(\mathbf{n}) f(\mathbf{n}) \, d\mathbf{n} \tag{4.38}$$

While $T(\mathbf{n})$ is never found explicitly it provides a useful tool for considering the behavior of these systems. By conjecturing about the properties of $T(\mathbf{n})$, we gain understanding. For

example it must be that $0 < T(\mathbf{n}) < 1$ and that for some region of noise vectors $\mathbf{n}$ there will always be a symbol error. For other regions of $\mathbf{n}$ there will never be an error but in between there must be a region wherein some values of noise $\mathbf{n}$ will result in an error for some combinations of symbol inputs and not for others. In utilizing this concept of $T(\mathbf{n})$ the authors have treated this region as a straight line from 0 to 1 between some unknown noise voltage points α and β on the possible voltages around threshold. For a fuller discussion of this concept, see [17] and [40], but the fundamental benefit is to acknowledge that the characterization of ISI needs to be a part of the simulation effort. This concept is also used to investigate the effect of memory on IS performance as discussed in Section 4.3.1.

In [40] a sequence-dependent version of the optimum bias function is formed using the STC concept. It is

$$f_i^*(\mathbf{n}) = \begin{cases} \frac{\lambda_i}{\sqrt{L}} f(\mathbf{n}) & \mathbf{n} \in \Omega_i \\ 0 & \text{otherwise} \end{cases} \qquad i=1,2,\ldots,L \tag{4.39}$$

The constant λ_i is defined in [40] as a normalizing factor based on the STC and ultimately on the probability of error sought. This result is simply a sequence-dependent version of eq.(4.30), the unrealizable optimum bias function in the non-sequence-dependent case. It suffers from the same inconsistencies in that construction of such a function still requires prior knowledge of the error probability and precise knowledge of the important region, specifically the error regions for each of the L sequences.

There is a final result of interest from this approach. It is shown that, for a linear system at least, the biasing scheme should have a time varying mean. This presumes that shifting of the mean is the method of biasing used. The time variance is proportional to the impulse response of the system. The theory developed in [40] asserts that the optimum direction to bias the input is the direction of the impulse response in K-dimensional space. It is pointed out that this optimum degenerates to the shifting method (IIS) of Lu and Yao[25] in the single dimension case.

To summarize, the sequence-dependent approach to IS points out that there are several things to consider. First, consider the optimum bias function of eq.(4.39). If it could be synthesized and used to govern the simulation effort, the variance of the output for that sequence would be zero. Hence, that sequence would only need to be tried once. If there are L sequences, each m bits long, then mL bits is the number of bits that would need to be simulated. This represents the maximum improvement possible under these ideal conditions. In

other words, this mL limit represents the minimum number of bits which must be simulated under any optimum or suboptimum bias condition if ISI is to be fully accounted for in system performance. Secondly, this concept of shifting the input bias in the direction of the system impulse response, i.e. the time varying bias function, will apply to any system and any bias method. In most IS implementations in the literature, the bias function is a constant in time. Most analyses of the IS procedure assume this. It is shown in [40] that keeping the bias constant places some limits on the possible improvement from IS. A constant bias is only optimum in the case of an integrate and dump system response. Even when such a device is used in the system, the overall system response is other than constant.

4.1.4 Additional Phenomena from the Optimum Bias Derivation

As part of the derivation, analysis and discussion of IS and biasing for IS in the literature, several useful properties of interest to this work have been developed. We collect them in this section for review and discussion. This permits their use in later discussion. These phenomena are all a part of and associated with the mathematics of IS as found in this chapter. The items are not formally named but are denoted here for later reference. To begin, we present the concept of the average weight along with some indication of its usefulness. Next, we relate a property of the weighting function and its relation to the probability of error value that is being estimated. Specifically we show that the maximum weight must be greater than the probability of error. Finally we present expressions for the upper and lower bounds on the IS estimator variance. These expressions are handy for bounding the estimator error. They can also be used to estimate the amount of improvement in simulation size over classical Monte Carlo. While this section is somewhat of a conglomeration, it represents some of the more useful ideas from the attempts in the literature to reduce IS to a universally applicable technique.

The Average Weight

The concept of average weight is very useful for monitoring the status of the IS simulation. With some development it may provide a mechanism for applying feedback to the IS simulation procedure so that the biasing and the simulation itself becomes adaptive and more automatic. Observe that the variance of the IS estimator, eq.(4.29), can also be written as

$$\sigma_{IS}^2[\hat{P}_e^*] = \frac{1}{N^*}\left[\overline{W} - P_e^2\right] \tag{4.40}$$

where $\overline{W}$ is defined as

$$\overline{W} = \int_{\mathbf{X}^K} \frac{f(\mathbf{s}, \mathbf{n})}{f^*(\mathbf{s}, \mathbf{n})} f(\mathbf{s}, \mathbf{n})\, d\mathbf{s}\, d\mathbf{n} = E\left[H(\mathbf{s}, \mathbf{n})\, W(\mathbf{s}, \mathbf{n})\right] \tag{4.41}$$

Note that $\overline{W}$ is the expected value of the weight function over the input region $\mathbf{X}^K$ and that this expectation is taken with respect to the non-biased PDF.

It can be shown that for importance sampling to be effective we require $\overline{W} \leq P_e$[29]. This can easily be seen by comparing the variance of the estimate using classical Monte Carlo, which is

$$\sigma^2_{MC}[\hat{P}_e] = \frac{P_e[1 - P_e]}{N}, \tag{4.42}$$

to the IS variance from eq.(4.40), written as

$$\sigma^2_{IS}[\hat{P}^*_e] = \frac{\left[\frac{\overline{W}}{P_e} - P_e\right] P_e}{N^*}. \tag{4.43}$$

The definition of $\overline{W}$, which we refer to as the average weight, has been used to develop a suboptimum bias function[29]. Specifically, it has been used to contrive the optimum amount of shift.

The more useful role for the average weight is in monitoring the variance of the IS estimate as shown below. The IS improvement ratio is defined as $r = \frac{N}{N^*}$ and is obtained by equating the variance of the Monte Carlo estimator and the variance of the IS estimator as described in Section 3.2.3. If this is done using the above expressions we find that

$$r = \frac{N}{N^*} = \frac{\left[P_e - P_e^2\right]}{\left[\overline{W} - P_e^2\right]}. \tag{4.44}$$

If $\overline{W}$ could be evaluated analytically (and this is not an impossible occurrence) then the variance of the IS estimator could be evaluated exactly, and the improvement ratio would be known and the optimum bias as a function could be established.

Often $\overline{W}$ cannot be found analytically. A method has been established to estimate $\overline{W}$ concurrent with the IS simulation[29]. Doing this provides some indication of how well importance sampling works, or has worked. In other words, $\overline{W}$, and with it r and the variance, can be estimated along with the estimation of P_e. This can be seen by rewriting $\overline{W}$ in eq.(4.41) as

$$\overline{W} = \int_{\mathbf{X}^K} \left[\frac{f(\mathbf{s}, \mathbf{n})}{f^*(\mathbf{s}, \mathbf{n})}\right]^2 f^*(\mathbf{s}, \mathbf{n})\, d\mathbf{s}\, d\mathbf{n} = \int_{\mathbf{X}^K} W^2(\mathbf{s}, \mathbf{n}) f(\mathbf{s}, \mathbf{n})\, d\mathbf{s}\, d\mathbf{n} = E_*\left[H(\mathbf{s}, \mathbf{n})\, W^2(\mathbf{s}, \mathbf{n})\right] \tag{4.45}$$

Where $\overline{W}$ is now written as the expected value over $f^*(\mathbf{s}, \mathbf{n})$, the density function which governs the IS simulation. From this equation, we form an estimate of $\overline{W}$, which we denoted as $\hat{\overline{W}}$, as

$$\hat{\overline{W}} \equiv \frac{1}{N^*} \sum_{i=1}^{N^*} H(\mathbf{s}_i, \mathbf{n}_i)\, W^2(\mathbf{s}_i, \mathbf{n}_i) \tag{4.46}$$

Given an estimate of $\overline{W}$ we can estimate the improvement ratio and the variance of the IS estimator. This technique has been tried and shown to work in the simulation of one-dimensional Laplacian densities and 10-dimensional Gaussian densities[29]. It provides a significant tool for obtaining confidence in the IS estimate.

Max[W(s,n)] must be $\geq$ *Pe*

It has been shown that the estimated P_e is bounded by the largest weight value which is produced in the summation[14]. The result is applicable for *any* bias method so that the proof is in terms of a general IS parameter α. For example, α may be the amount of shift or represent the amount of variance increase. In the general case the weight, as a function of α and the random variable y, might be expressed as

$$W(y,\alpha) = \frac{f(y)}{f^*(y)} = \frac{f(y)}{f(y,\alpha)} \tag{4.47}$$

Let H(y) be the system indicator function. Since $f(y,\alpha) > 0 \ \forall \ (y,\alpha)$ and $1 \geq H(y) \geq 0 \ \forall \ y$ then

$$\hat{P}_e^* = \frac{1}{N^*} \sum_{i=1}^{N^*} H(y)\, W(y,\alpha) \leq \frac{1}{N^*} \sum_{i=1}^{N^*} \max\left[W(y,\alpha)\right] = \max\left[W(y,\alpha)\right] \tag{4.48}$$

This proves that the largest weight used in IS must be at least as great as the value trying to be estimated. That is, we must maintain

$$\max\left[W(y,\alpha)\right] \geq P_e \tag{4.49}$$

This means that there is a limit to the amount of bias that should be applied. If the weights are made too small, i.e. the bias made too great, then the resulting estimator will be biased. The above property has been used to show that an over translation of the input bias function results in an underestimation of P_e and then exploited as a means of determining the optimum shift to use for IS through an iterative procedure[14].

Upper Bound and Lower Bound on IS Variance

An upper and lower bound to the importance sampling estimation variance has been derived[28][26]. Consider a generalized weight function

$$W(y,\alpha) = \frac{f(y)}{f^*(y)} = \frac{f(y)}{f(y,\alpha)} \tag{4.50}$$

and a decision space Ω, where perhaps $\Omega = [y: y \geq T]$ for example. Then the estimation variance has been shown to be upper and lower bounded as

$$\sigma^2_{IS}(\alpha) \leq \sigma^2_{UB}(\alpha) = \frac{P_e}{N^*}\left[\max[W(y,\alpha)] - P_e\right] \tag{4.51}$$

$$\sigma^2_{IS}(\alpha) \geq \sigma^2_{LB}(\alpha) = \frac{P_e^2}{N^*}\left[\frac{1}{P_1(\alpha)} - 1\right] \tag{4.52}$$

where $P_1(\alpha)$ is the error probability using the modified PDF $f^*(y)$, i.e. $P_1(\alpha) \equiv \int_\Omega f^*(\mathbf{s},\mathbf{n},\alpha)\, d\mathbf{s}\, d\mathbf{n}$ which can be estimated in the usual fashion as $\hat{P}_1(\alpha) \equiv$ (number of errors)/N^*.

The improvement ratio $r = \frac{N}{N^*}$ is then upper and lower bounded by

$$r(\alpha) \leq r_{UB}(\alpha) = \frac{P_1(\alpha)[1 - P_e]}{P_e[1 - P_1(\alpha)]} \tag{4.53}$$

$$r(\alpha) \geq r_{LB}(\alpha) = \frac{[1 - P_e]}{\left[\max[W(y,\alpha)] - P_e\right]} \approx \frac{1}{\left[\max[W(y,\alpha)] - P_e\right]} \tag{4.54}$$

The proofs for these results can be found in [28] and [26].

The bounds above provide direct approaches for estimating the simulation variance and improvement ratio. Particularly useful in this regard is the upper bound on the variance since that indicates how well the simulation is proceeding and the lower bound of the improvement ratio since that indicates a conservative estimate of the number of trials that must be performed. Note that these expressions rely only on knowing the maximum weight value in a the decision space. One further point: equalities in the above expressions are obtained if $W(y,\alpha) = W$ is any constant in the decision region Ω. This suggests that a constant or nearly constant weight value is desirable. The optimum bias function also has a constant weight. Some of the sub-optimum versions of importance sampling presented in Section 4.2 also suggest that a nearly constant weight value is desirable.

4.2 VARIOUS VERSIONS OF IMPORTANCE SAMPLING

While the acronyms used to denote each of the various IS methods in this section have been adopted and expanded from those found in the literature, it will be quickly evident to the reader that other names and acronyms exist for many of these methods. There is no standard. Indeed, many titles in the literature contain the words "efficient importance sampling" or "improved importance sampling." Some of these refer to the techniques defined here; many do not. The names find some standard usage but primarily only with the authors who originally cooked up the technique and the name. We have attempted to adopt names and notation which come close to the set of acronyms which finds some measure of universal usage in the literature. Some compromise and extension has been made in order to make the acronyms unique. Without this there would be four CIS acronyms, one each for Classical Importance Sampling, Conventional Importance Sampling, Conditional Importance Sampling and Composite Importance Sampling. Each of these occur in the literature. It is regrettable that the names "efficient importance sampling" or "improved importance sampling", for example, do not give any indication of the type of IS that is being discussed. More descriptive names could be given to these types of IS but it was decided that to do so would only compound the problem and leave the reader with no familiarity with what is actually found in the literature. This discussion is the first attempt to catalog in one place the various IS methods found in the literature.

4.2.1 Classical Monte Carlo (MC)

An explanation and presentation of the mathematical principles of Monte Carlo simulation has been given in Chapter 3. The literature concerning unmodified Monte Carlo simulation methods, which we refer to as classical Monte Carlo, can be found in abundance in the scientific and engineering literature of many fields as well the purely mathematical and statistical literature. This includes many textbooks devoted to the subject. A complete accounting of this literature is not practical or necessary for this work. It is prudent to cite some of the more readable works on the subject for the interested reader, particularly those works oriented towards utilizing Monte Carlo methods in the analysis of communications system problems.

The review by Jeruchim[24] of several simulation methods, including classical Monte Carlo and importance sampling, is very good in that it presents MC and IS in the context of communication system analysis and because it presents the two together along with other

techniques so that they may be contrasted and compared. It suffers only in that the presentations are succinct and thus may not be fully beneficial to the reader new to these techniques and their nomenclature.

For more in depth study, there are several texts which discuss and describe importance sampling as well as other methods of simulations and their principles[41-46]. Excellent descriptions of simulation techniques and more recent developments presented in a more modern context can be found in the January 1984 and the January 1988 issues of the *IEEE Journal on Selected Areas in Communications*, Vol. 2, No. 1 and Vol. 6, No. 1 respectively. These issues of the journal have concentrated on computer simulation of communication systems. A third issue devoted to computer aided analysis and design of communications systems is expected in the early part of 1993.

4.2.2 Scaling Importance Sampling (SIS)

The scaling version of IS refers to forming a bias function by increasing the variance of the input noise density function. Typically this has been a Gaussian density function. Increasing the variance of the input noise process will cause more samples to be taken from the tails of the distribution. This can be performed by creating a zero mean unit variance Gaussian random generator whose output is a value X, for example. A new Gaussian random variate, Y, can be created from X by applying it to the following function: $Y = \alpha X + \beta$. The random variable Y will then have a mean of β and a variance scaled to α. The mean is not varied for SIS as we have defined it here, i.e. β is always zero. An example of the creation of an SIS biased PDF is shown in Fig. 4.2 where an increase in variance, $\alpha > 1$, is depicted.

SIS is often referred to as CIS, for Classical Importance Sampling, particularly by the authors Lu and Yao[25] or as Conventional Importance Sampling, again CIS, as in [22] and other works by these authors. This title of *classical* or *conventional* attests to its implementation as the first widely used bias technique beginning with its presentation by Shanmugam and Balaban[3] in one of the first IS works in the communications literature. The only previous work had discussed methods of modifying the look-up table used to generate random variates[2]. It has been referred to as *scaling* only by more recent authors. It was presented in [3] simply as a means of increasing the number of samples from the tails of the input Gaussian density function. The system analyzed was a simple baseband communications system with additive Gaussian noise and a memoryless nonlinearity followed by threshold detection. The authors of [3] were demonstrating the use of IS on a very simple communications

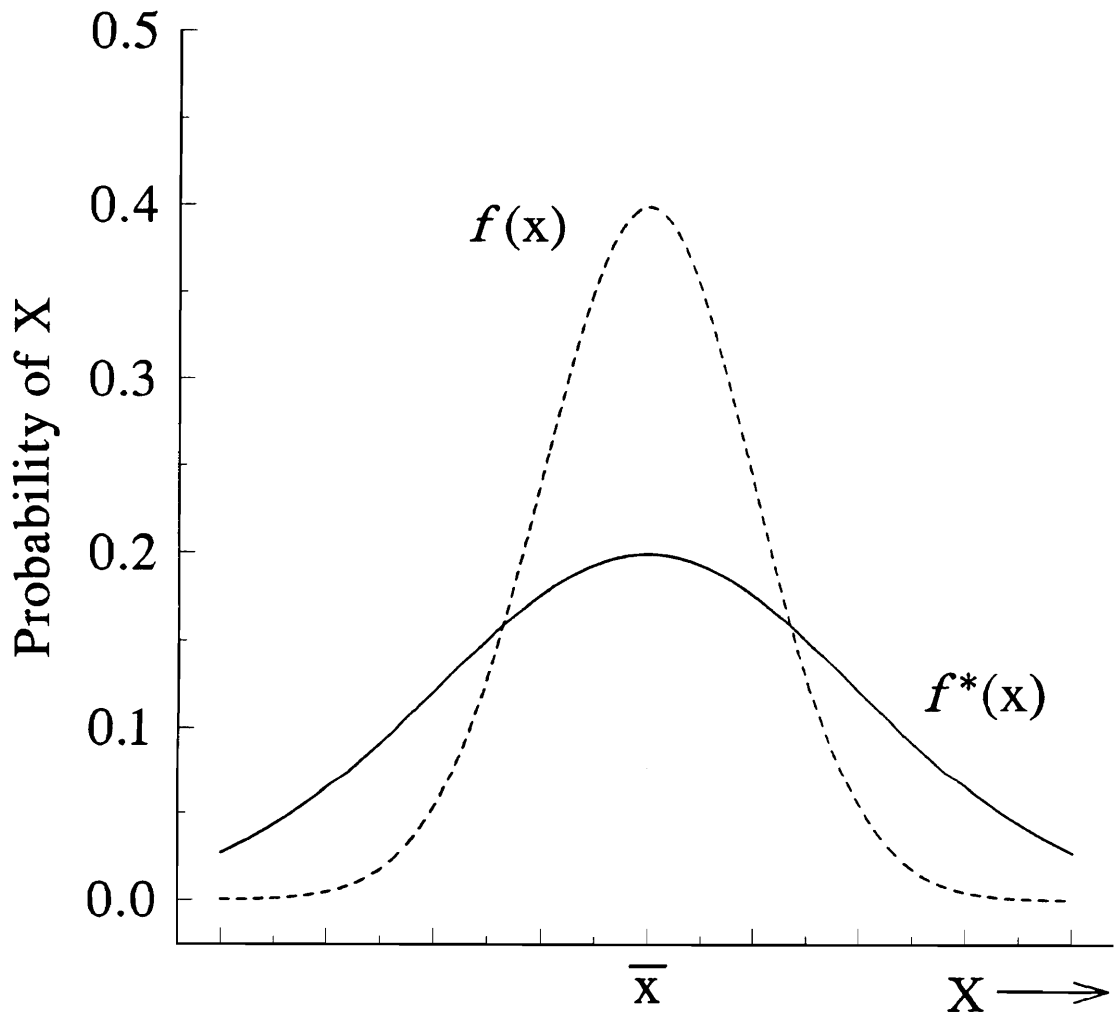


Fig. 4.2 Scaling Importance Sampling (SIS) Bias Function Example
where $f(x)$ is the original input PDF, $f^*(x)$ is the biased PDF obtained by increasing the variance of $f(x)$, $\bar{x}$ is the mean of $f(x)$ and $f^*(x)$.

system problem, that is, they were as much introducing IS as they were analyzing a communications problem.

A method of scaling was presented in [3] which was adopted by many of the researchers who followed. A bias function B(x), where

$$B(x) = \frac{f_X^*(x)}{f_X(x)}, \tag{4.55}$$

was formed as

$$B(x) = \frac{c}{[f_X(x)]^\alpha} \tag{4.56}$$

where c is chosen so that area under the new PDF function, $f_X^*(x)$, is equal to one and and α chosen for maximum improvement in sample size reduction. It can be shown that choosing c such that

$$c = \sqrt{\frac{(1-\alpha)}{(2\pi)^\alpha}} \tag{4.57}$$

will maintain the area under the PDF equal to one and that the variance of the new PDF is increased by $\frac{1}{1-\alpha}$ where α is in the interval (0, 1). This bias method was frequently adopted and is referred to often in the IS literature as classical or conventional IS.

As discussed in [3], the amount of IS improvement is a function of the choice of α and that the optimum α; α_{opt}, defined as that value of α providing the most IS improvement, is a function of system memory and the probability of error to be estimated. An expression for α_{opt} is given in [3, eq.(20 b)]. It is a function of the unknown tail threshold (if it were known there would be no need in performing the simulation) and the system memory. The authors suggest that "a conservative value of $\alpha = 0.5$ can be used in most cases without lowering the value of r significantly" where r is the sample size reduction to be obtained from IS. This inability to determine the optimum IS parameters (α in this case) to use in the simulation without first knowing the result of the simulation is typical of the IS problem and recurs throughout the literature. Indeed, this problem has been the focus of much of the IS research to be found in the literature.

Scaling remains a popular method of biasing the input noise density in a simulation. Its popularity diminished somewhat when it was proven that shifting the mean of the input noise density function, what we have called here Improved Importance Sampling (IIS) for historical reasons, was a better method of biasing. It has been shown that IIS is fundamentally able to yield larger improvement ratios over scaling. This is discussed in Section 4.2.3.

It is interesting to note that increasing the variance of the input noise PDF is equivalent to increasing the noise power and hence lowering the input C/N ratio. This change in input C/N ratio is then adjusted for at the output by the weighted counting of errors.

Comparisons of SIS to later methods of bias function generation are plentiful. They can be found in [25][27][28] where SIS and IIS techniques are applied to an AWGN threshold detection system similar to that discussed above in [3]. IIS is consistently shown to outperform SIS. SIS has been compared to EIS in an extensive set of experiments[22]. More is said concerning this comparison when EIS is discussed in Section 4.2.7 below. It was the bias method used in a study of satellite multihop links where two input noise sources existed in the simulation model[23], one at the terrestrial receiver and one noise source at the satellite transponder.

Derivations of the optimum bias functions for Viterbi decoders generate a multidimensional bias function[20][19]. These have been shown to reduce to the SIS bias method in one dimension. In these cases, the bias technique has been denoted as Modified SIS.

The generation of a bias function by increasing the variance maintains the same shape, the same general properties, as the original non-biased PDF function for the input noise in the simulation. It has been conjectured that this is a desirable property, in that replacing the original function with a function that bears little resemblance to the original may result in deficient simulation results owing to the simulation of signals which do not resemble realistic signals in the system.

4.2.3 Improved Importance Sampling (IIS)

Improved Importance Sampling (IIS) is simply a shifting of the mean of the input PDF in such a way as to create more input random variates in the tail of the PDF which causes more errors. In the binary communication receiver case with input signals of +A volts and $-$A volts, threshold detection and AWGN, for example, the mean of the input noise density function is shifted up (positive) when the input signal is $-$A and shifted down (more negative) when the incoming signal is a +A volts. An example of a representative IIS biased PDF, $f_X^*(x)$, is shown in Fig. 4.3 along with the original PDF $f_X(x)$, for comparison. This causes more samples to be taken from the appropriate tail region of the original noise PDF. The reader may notice that this arrangement effectively lowers the input C/N ratio in a manner similar to that mentioned in the discussion concerning SIS. This lowering of the input C/N ratio is, again, a controlled

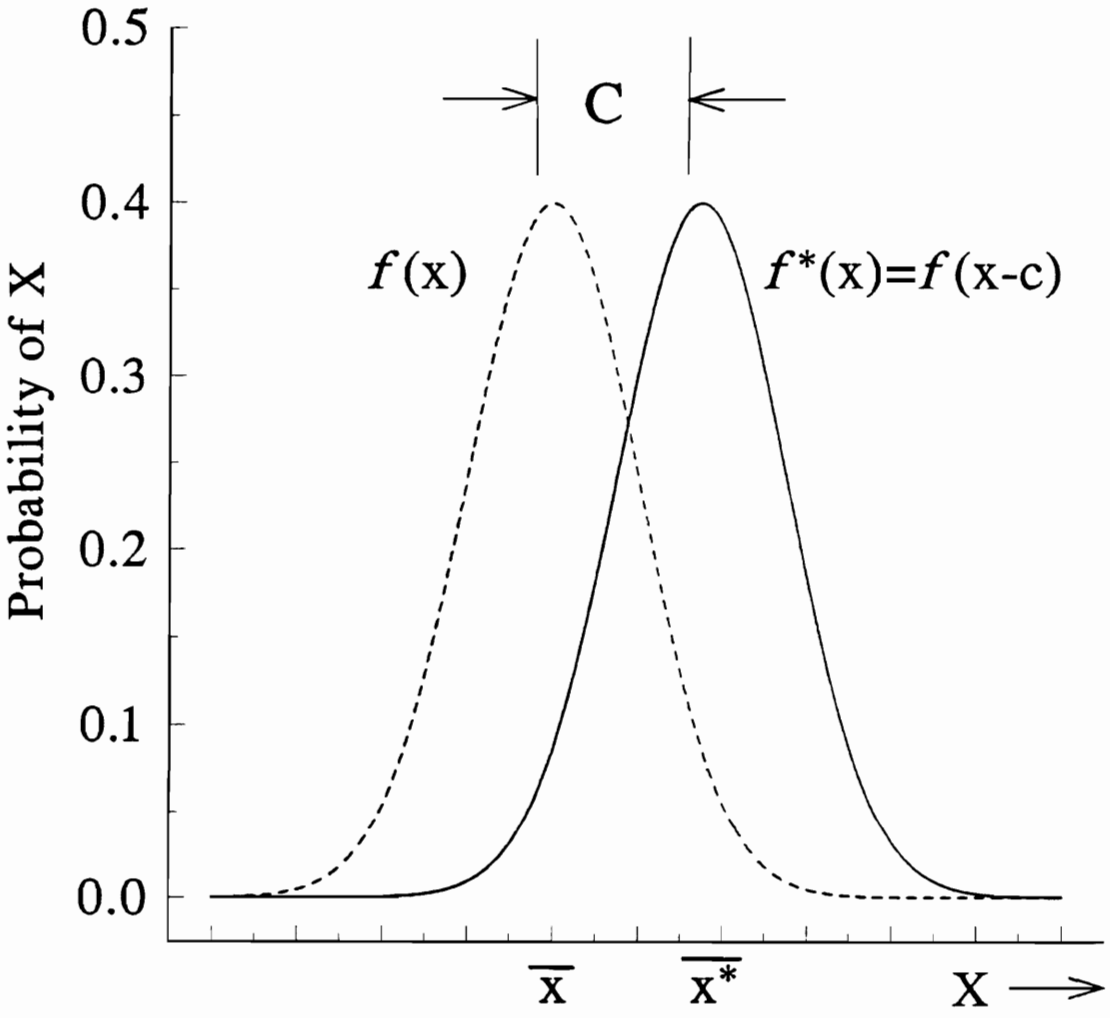


Fig. 4.3 Improved Importance Sampling (IIS) Bias Function Example
where $f(x)$ is the original input PDF, $f^*(x)$ is the biased PDF obtained by shifting the mean of $f(x)$ by C, $\bar{x}$ is the mean of $f(x)$ and $\bar{x}^*$ is the mean of $f^*(x)$.

process which is compensated for by the way in which errors are counted.

IIS first appeared in January of 1988[25]. It was shown to perform better and yield greater savings in sample size than increasing the variance using SIS. This result was conjectured to stem from the fact that the input signal variance is smaller with IIS than with SIS. Both methods lower the input C/N ratio but IIS lowers the C/N ratio and yet maintains an input signal with the same variance. With a smaller variance at the input, the variance at the output will be smaller by necessity. As is typical, the system used for comparison in [25] utilized threshold detection with AWGN. The authors derive the optimum shift, C_{opt}, and variance improvement for both SIS and IIS. This derivation is good for comparison of the methods but the resulting formulas are in terms of the unknown thresholds and, moreover, the derivation assumes a linear system. Again, here is the classic IS problem wherein the optimum IS parameter depends on the answer sought.

IIS is discussed again and compared with SIS in [26],[27] and [28]. Two of these works discuss IIS in the context of digital communications systems and one in the context of estimating radar false alarm probabilities. They are also notable in that they develop an upper and lower bound to the IS estimator variance as presented in Section 4.1.4. A technique is presented in which the upper bound is used to derive a suboptimum shift, C, of the input density function[28]. The method is valid in principle, but on closer inspection one is still left with needing at least some idea of the desired probability of error in order to make a reasonable choice. Nevertheless, the technique is useful.

In an effort to simplify the optimum shift selection problem, an iterative method was developed to find C_{opt} through a series of short simulation runs[14]. It was observed that if the mean of the input PDF was not shifted enough, $C<<C_{opt}$, then very few errors, often none, would be generated in a very short simulation run of perhaps 100 bits. The exact number to use being somewhat arbitrary. A small shift in the input PDF means that the PDF used in the simulation is not yet very different from the original unmodified PDF and having no errors generated in a short simulation run is a natural occurrence if the probability of bit error is small. More importantly, it was observed that if the mean of the input PDF was shifted beyond the optimum, $C>>C_{opt}$, then the observed BER decreased with increasing shift for a constant set of input parameters. This latter property can be shown mathematically to occur[14]. The BER measured even for a small number of sample bits was constant for values of C, the amount of shift, in the region around the C_{opt}. The procedure suggested by these observations is to perform a series of short simulation runs, keeping all things constant except for the amount of

shift. If the BER results are plotted versus the amount of shift then it is possible to observe a region of shift in which the BER is generally constant, somewhere before the region in which it decreases quickly. In this way one can select a near optimum value of shift parameter for a given system.

The simulation of systems with several different types of input PDF's have been studied experimentally using the iterative method described in the proceeding paragraph[14]. All of the systems have a threshold detection decision device. The PDF's examined are the Gaussian, Rayleigh, asymmetrical exponential and the APD noise density functions. All of these are additive except for the APD shot noise. The mathematics used to develop their results is general enough to consider both additive and non-additive input noise. They report improvement factors over MC simulation ranging from three to eight orders of magnitude. This is the only work which discusses shifting of APD shot noise statistics. The authors report using the WMC approximation for the shot noise in the APD model.

Further, the authors in [14] acknowledge that shifting one-sided PDF's is not straight forward. This is because when shifting down one-sided statistics it is conceivable that you will arrive at a PDF which is non-zero for values less than zero. In other words, by shifting you can create a $f_X^*(x)$ from which negative value random variates can be drawn. In the case of an APD this corresponds physically to simulating an APD in which current flows in the opposite direction for certain values of input optical power. To quote, "For single-sided PDF's like the Rayleigh and the WMC, a modification of the standard, translation-based biasing technique had to be used for the IS estimator ... to be correct."[14] They *do not* say what this modification should be or how they produced the results reported for the Rayleigh and WMC densities. In Table 1 of [14] they report improvement factors from 10^3 to 10^5 for systems using the WMC density function.

One of the best derivations for the optimum IS bias density function is given in [29]. After deriving the unrealizable optimum the authors go on to derive IIS as a suitable suboptimum bias function. With the mathematical approach taken they are able to show that the best choice of C, i.e. the amount of shift, is specific to every noise density function and signal to noise ratio. Once again the optimum shift value cannot be determined. They go on to claim that shifting one half of the distance between the binary signals is a robust choice. While their derivations are general, they chose to apply the results to systems utilizing the generalized Gaussian family of density functions. A later work by these authors utilized these results and applied them to the simulation of the DS-SSMA problem using various non-Gaussian detector

structures and background noise distributions[31]. One significant result from these works is that it was shown that the gain in the number of simulation trials required for a fixed accuracy over standard Monte Carlo simulation was approximately inversely proportional to the error probability. As the number of simulation samples required for given accuracy in standard Monte Carlo is proportional to the BER, this means that in theory a properly applied IS method should require a constant number of samples throughout the generation of a BER curve.

4.2.4 Pseudo-Optimum Importance Sampling (PIS)

The literature examples discussed in this section and the next present a fundamentally similar technique for generating a bias function which attempts to approximate the optimum IS bias function of eq.(4.30) or eq.(4.31). Observing the salient features of the optimum bias function and without knowledge of the true importance region and of P_e it was decided to select a suboptimum PDF as

$$\overset{*}{f}_{\delta}(\mathbf{s},\,\mathbf{n}) = \begin{cases} \dfrac{f(\mathbf{s},\,\mathbf{n})}{\epsilon} & (\mathbf{s},\,\mathbf{n}) \in \tilde{\Omega}_{\delta} \\ \\ 0 & \text{otherwise} \end{cases} \tag{4.58}$$

where an *assumed importance region* $\tilde{\Omega}_{\delta}$ has been created. This suboptimum bias function can also be expressed in a form reminiscent of eq.(4.30) as

$$\overset{*}{f}_{\delta}(\mathbf{s},\,\mathbf{n}) = \frac{\mathrm{H}_{\delta}(\mathbf{s},\,\mathbf{n})\,f(\mathbf{s},\,\mathbf{n})}{\epsilon} \tag{4.59}$$

$\mathrm{H}_{\delta}(\mathbf{s},\,\mathbf{n})$ is *not* an indicator function. Symbol decisions are still made according the decision rule dictated by the system model, $\mathrm{H}(\mathbf{s},\,\mathbf{n})$, as always. The constant ϵ is a small number conjectured to be close to P_e and held to a relationship with $\tilde{\Omega}_{\delta}$ as it is the value which normalizes the integral of $\overset{*}{f}_{\delta}(\mathbf{s},\,\mathbf{n})$ to one, i.e.

$$\epsilon = \int_{-\infty}^{\infty} \mathrm{H}_{\delta}(\mathbf{s},\,\mathbf{n})\,f(\mathbf{s},\,\mathbf{n})\,d\mathbf{s}\,d\mathbf{n} \tag{4.60}$$

Eq.(4.60) forms the link between the selection of ϵ and the selection of $\tilde{\Omega}_{\delta}$. Select one and you select the other. Implicit in the adoption of this technique is the ability to solve the integral in eq.(4.60). Hence the PIS technique is limited to use with PDF's for which this integral can be solved or, at least, approximated with high accuracy. A one dimensional example of the Pseudo-Optimum IS bias function is shown in Fig. 4.4 along with the original PDF function. Also shown is the unknown threshold value, V_T, and the assumed cutoff for the bias function at

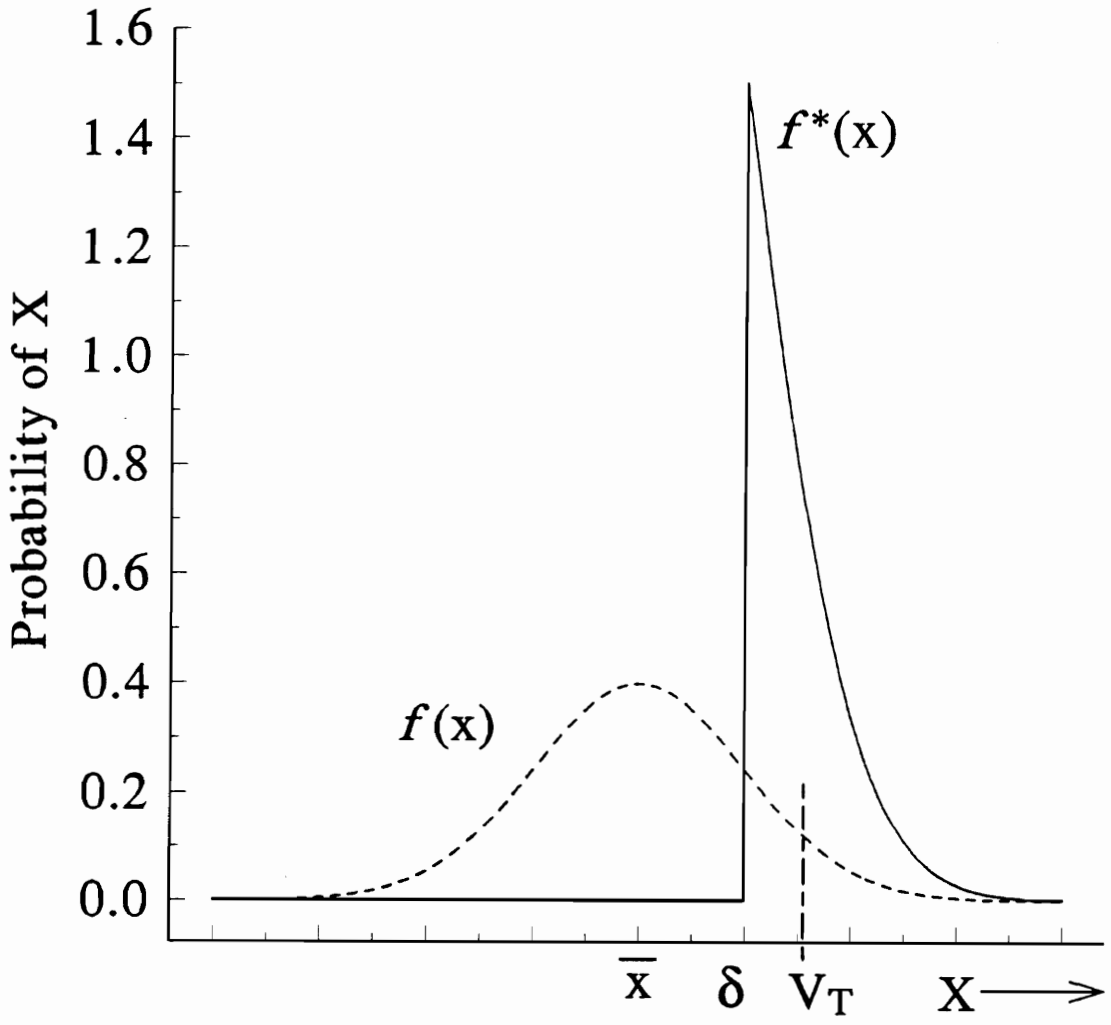


Fig. 4.4 Pseudo-Optimum Importance Sampling (PIS) Bias Function Example
where $f(x)$ is the original input PDF, $f^*(x)$ is the biased PDF, V_T is the true system threshold, $x \geq V_T$ is the true important region, δ is the assumed error threshold, $x \geq \delta$ is the assumed important region, $\bar{x}$ is the mean of $f(x)$.

δ. The assumed important region in this example is for those x values greater than δ. The true important region consists of those x values greater than V_T.

This bias scheme was first suggested in [21] following a derivation of the optimum bias function. (Aside: The technique was referred to as OIS in [21]. OIS was an acronym for Optimum Importance Sampling. We have chosen not to call it optimum importance sampling because it is not.) A slightly modified form of PIS was presented in [39]. A comparison between using the Gaussian density function and the Rayleigh density function for $f(\mathbf{s}, \mathbf{n})$ in the above expressions was studied for [9]. In the next few paragraphs we discuss some pertinent features of the above method and describe how it might be implemented. In Section 4.2.5 we present a similar method whose objective was not necessarily to approximate the optimum bias function but which shares some striking similarities to this method.

Concerning the selection of ϵ and $\tilde{\Omega}_\delta$, let us consider two cases[21]. In this discussion we express our functions in terms of one random vector $\mathbf{x}$ without loss of generality.

Case 1: Assume that $H_\delta(\mathbf{x}) \geq H(\mathbf{x})$ for all $\mathbf{x}$. That is, the true importance region is a subset of the assumed one for all $\mathbf{x}$. Recalling the variance for the IS estimator from eq.(4.29)

$$\sigma^2_{IS}[\hat{P}^*_e] = \frac{1}{N^*} \int_{\forall \mathbf{x}} H(\mathbf{x}) \left[W(\mathbf{x}) - P_e \right] f(\mathbf{x}) \, d\mathbf{x} \tag{4.61}$$

Using the above PIS bias method we have

$$W(\mathbf{x}) = \frac{f(\mathbf{x})}{f^*(\mathbf{x})} = \frac{\epsilon}{H_\delta(\mathbf{x})} \tag{4.62}$$

Under the conditions of Case 1

$$\begin{aligned} \sigma^2_{IS}[\hat{P}^*_e] &= \frac{1}{N^*} \int_{\forall \mathbf{x}} H(\mathbf{x}) \left[\frac{\epsilon}{H_\delta(\mathbf{x})} - P_e \right] f(\mathbf{x}) \, d\mathbf{x} \\ &= \frac{1}{N^*} P_e(\epsilon - P_e) \end{aligned} \tag{4.63}$$

Setting the variances equal for the IS and non-IS estimator it is possible to express the sample size savings ratio as

$$\frac{N}{N^*} = \frac{(1 - P_e)}{(\epsilon - P_e)} \approx \frac{1}{(\epsilon - P_e)} \tag{4.64}$$

In other words the amount of improvement is directly dependent on how closely our assumed importance region is the true importance region or, similarly, how close our ϵ is to P_e. Consider that $\frac{N}{N^*} \approx \frac{1}{\epsilon}$ for $\epsilon \gg P_e$ and that $\frac{N}{N^*} \approx \frac{1}{\kappa P_e}$ for $\epsilon \approx (1+\kappa)P_e$ which means that the improvement

ratio can be made to go as P_e^{-1} if κ can be made as small as desired. This ability has been referred to as asymptotically efficient by other researchers[36] and is desirable in that it means that the number of samples required for simulation would remain constant regardless of how small P_e became.

Case 2: Assume that $H_\delta(\mathbf{x}) \leq H(\mathbf{x})\ \forall\ \mathbf{x}$. That is, the true importance region is actually larger than the assumed one which means that the assumed region does not contain all of the true one. (This condition corresponds to the case the $\delta > V_T$ in Fig. 4.4.) In this case

$$\sigma_{IS}^2[\hat{P}_e^*] = \frac{1}{N^*}\int_{\forall \mathbf{x}} H(\mathbf{x})\left[\frac{\epsilon}{H_\delta(\mathbf{x})} - P_e\right] f(\mathbf{x})\, d\mathbf{x} \rightarrow \infty \tag{4.65}$$

This result is due to the existence of a range of $\mathbf{x}$ in which $H(\mathbf{x}) = 1$ while $H_\delta(\mathbf{x}) = 0$. This result is reported in [21] and [39]. It is not precisely correct since under the conditions of case 2 the estimator $\hat{P}_e^*$ becomes biased. The following explanation is in order.

If the assumed importance region is smaller than the actual one then, since input samples are drawn from this region only, an error will occur at every trial. Every weight that is used in the summation is $\epsilon < P_e$ so that

$$\hat{P}_e = \frac{1}{N^*}\sum_{i=1}^{N^*} \epsilon_i = \epsilon < P_e \tag{4.66}$$

With an error on every trial the output will be constant and the variance will be zero. But the estimator is now biased so that the expected value of the estimator is ϵ which is less than the desired P_e. The trouble with eq.(4.65) is that the integration is carried out over all $\mathbf{x}$. This is the correct definition and method of calculating the variance. It does not precisely predict the simulation variance since the simulation is governed by the (incorrectly) biased PDF so that none of the $\mathbf{x}$ variates which result in the $H(\mathbf{x}) = 1$ while $H_\delta(\mathbf{x}) = 0$ condition will be generated in the simulation. Hence under the conditions of case 2, the output will be constant, the variance will be zero but the estimator is now biased so that the result obtained by the simulation will be wrong. This result was first pointed out in [9] where it is stated that under the conditions of case 2, the "the estimator is not a meaningful measure."

We see that PIS has the potential to be a very powerful technique if it can be implemented. An adaptive method of implementation has been suggested for the case of additive Gaussian noise[21]. It requires one to first make a conservative conjecture as to the values of $\tilde{\Omega}_\delta$ and ϵ. It should be large enough to contain the true importance region yet small enough to keep ϵ small. After implementing the bias function of eq.(4.58) one should have a

sample size reduction in the range of (ϵ^{-1}, P_e^{-1}). By counting the number of errors and maintaining a large percentage for a given number of trials, the accuracy and the goodness of the chosen parameters can be judged. Three numerical examples of simulating satellite links are reported in [21] with good agreement to published values down to the 10^{-10} level using 512 bits.

A modified form of PIS is reported in [39]. Here, an effort is made to ease the selection of the optimum importance region through the blurring of the cut-off between the region of the PDF that should be zero and the region that is not. This concept originated by rethinking what the form of the optimum bias solution is telling us. It might be interpreted that the PDF which governs the simulation should have a region that is zero, a region that is shaped like that of the original PDF and a transition region in the middle which moves the value of the PDF from zero to non-zero in a continuous manner. If the unknown threshold can be assured of existing in the transition region then the bias function will produce correct results. This places less stringent requirements on the selection of the assumed importance region at the cost of some loss in sample size reduction performance. Note that the PIS technique from [21] described above can be considered a special case of this modified form, one with an infinitesimally small transition region.

One might notice that the important region is often the tail of the density function. Hence the choice of an assumed importance region generally amounts to the selection of an assumed threshold that, as pointed out above, should include the actual unknown threshold. For the additive white Gaussian noise case, an investigation was made comparing the use of the Gaussian tail (GT) to the Rayleigh tail (RT) in the type of IS biasing we have described here and denoted as PIS[9]. The purpose is to investigate how each performs in the context that the true threshold is not known and an estimate must be used. The analysis shows that the RT gives better performance than the GT in the case that the value V_T is not known. If V_T is known (there is no point in doing the simulation) the GT is best. Indeed, as we can see from examination of eq.(4.63) the GT yields a zero variance in the case of V_T known. The RT variance is small for V_T known and goes to zero as $V_T \rightarrow \infty$[9].

The strikingly similar AIS method, discussed in the next section, has also been compared to the RT and GT bias functions[9]. The AIS technique is found to be more robust than the RT when the V_T is unknown. As V_T becomes better known, the RT performs better than AIS. For a specific system example, it is shown that if the threshold is known within 10% at $P_e = 10^{-3}$ or within 5% at $P_e = 10^{-10}$ then the RT method yields a greater sample size

reduction than the AIS method. Thus the comparison work of [9] presents an interesting lesson. When beginning the investigation of a completely unknown system perhaps the techniques of AIS or IIS should be utilized in the beginning. As the system becomes known and for verification purposes the PIS technique using a RT as the biased input distribution forms the most robust combination.

4.2.5 Absolute Value Importance Sampling (AIS)

What we have chosen to call here the Absolute value Importance Sampling (AIS) method forms a bias function in the following manner. If the system has an input process which generates random variates X then the new random variates, X′, are related to X as

$$X' = |X| + C \tag{4.67}$$

This is a translation of the absolute value of the original random variable. Assuming for a moment that it is the positive tail of the density function which is of interest, this technique increases the probability of random variates in the positive half of the density function by two. The shift increases the tendency for random variates to be generated only in the important region. Just as with the PIS method of Section 4.2.4, the shift cannot be made greater than the (unknown) threshold lest the IS estimator become biased and unusable. This technique is equally applicable to studies concerning the lower tail of the density function through use of a negative sign before the absolute value bars in eq.(4.67) along with an appropriate change in direction of the shift.

The PDF of X′ is

$$f^*(\mathbf{x}) = \begin{cases} 2f(\mathbf{x} - C) & \mathbf{x} \geq C \\ 0 & \mathbf{x} < C \end{cases} \tag{4.68}$$

assuming that it is the positive tail of the PDF that is to be promoted in the simulation. An example is shown in Fig. 4.5. It should be pointed out that this technique seems to require a symmetric PDF for its implementation. All reported cases of its use have involved the use of Gaussian PDF's in the additive noise environment[8][37]. The effect of "folding" the PDF via the absolute value operator of an asymmetric PDF would need to be studied before attempting this method.

While this technique shares some resemblance to IIS through the act of shifting, the similarity with PIS is striking. Many of the same comments concerning the selection of the proper assumed threshold or assumed importance region apply here as well. Here, these

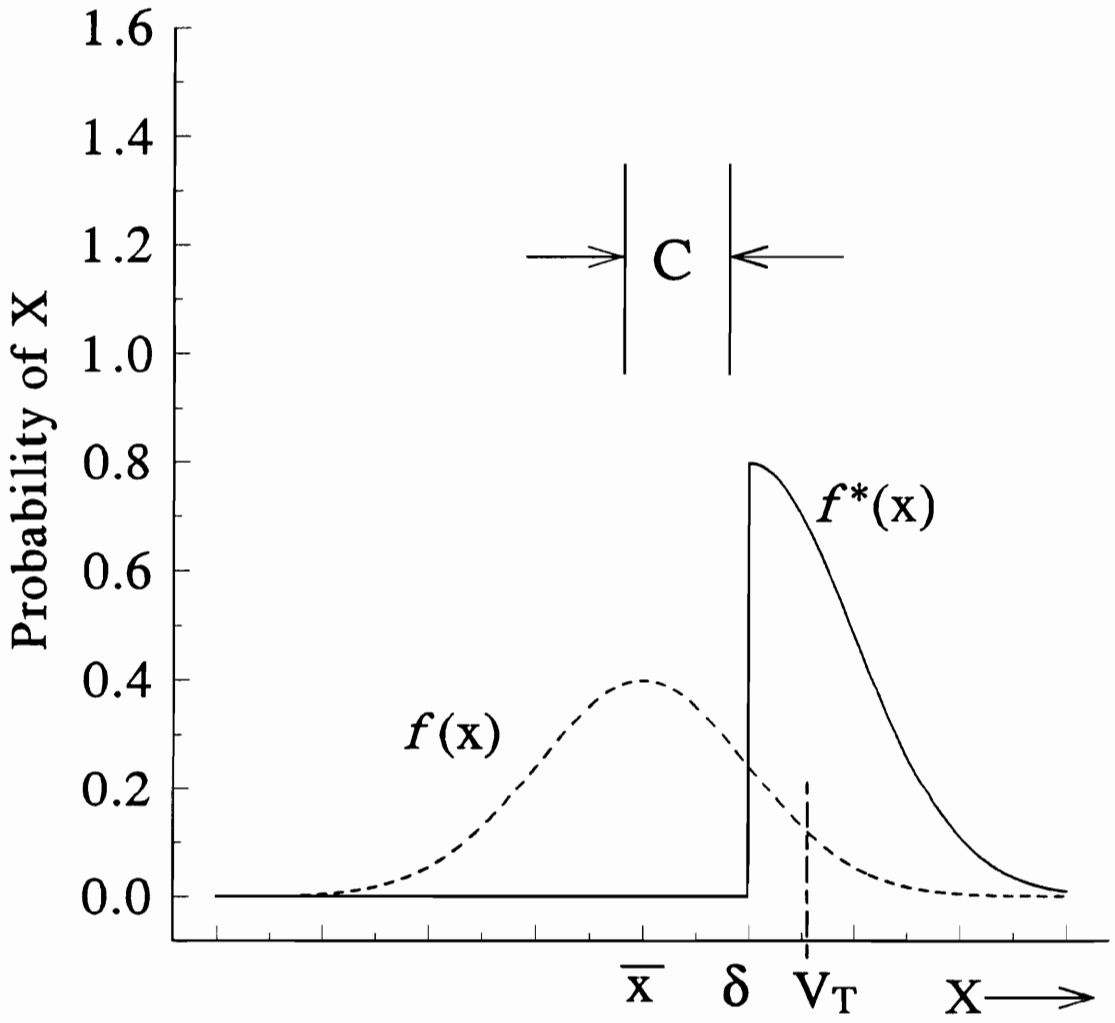


Fig. 4.5 Absolute Value Importance Sampling (AIS) Bias Function Example
where $f(x)$ is the original input PDF, $f^*(x) = 2|f(x - C)|$ is the biased PDF, V_T is the true system threshold, $x \geq V_T$ is the true important region, δ is the assumed error threshold, $x \geq \delta$ is the assumed important region, $\bar{x}$ is the mean of $f(x)$, $\delta = \bar{x} + C$.

comments apply to the amount of shift. For example, the performance of this bias technique improves as C is made closer to the unknown threshold and as already mentioned the technique degenerates to a biased estimator if the shift is made beyond the unknown threshold, i.e. the assumed importance region is made too small. The mechanism and explanation for this is the same as already discussed in the PIS case.

Interestingly, and to add to the IS name confusion, this method was first described as a "composite" bias method[8]. It was also presented and published independently at approximately the same time where it was called *Shifting the Absolute Value*[37]. Analysis shows that this scheme is better than simple translation by approximately a factor of 2.5[8]. This makes sense since in the additive noise case there is a higher probability of generating a random variable X in the range which causes an error. The work also draws parallels with Devetsikiotis and Townsend[14] in the shifting and the discussion of the use of non-Gaussian functions for the biased PDF.

A comparison between several linear and nonlinear biasing methods, including AIS has been performed[37]. As defined in [37], the linear methods are SIS, which is called Variance Modification (VM), and IIS, referred to as Mean value Modification (MM). The analysis in [37] assumes linear systems with additive Gaussian noise throughout with the hope of contributing to the understanding of nonlinear non-Gaussian systems. The combination of SIS and IIS, that is shifting with increased variance, is also analyzed. Computation is presented to show that a combination of SIS and IIS degenerates to just IIS when optimized. In something of a contradiction, the combination of SIS and IIS is developed as suboptimum in [40]. The nonlinear options to PDF biasing explored in [37] are: 1) AIS, which this author calls Shifting the Absolute Value (SAV), and 2) PIS, which this author calls Sample Elimination (SE). As discussed in Section 4.2.4, PIS requires some knowledge of the threshold and implies an ability to evaluate the area under the PDF. Some investigation of interval (rather than tail) estimation using these approaches is presented.

Analysis and experimental results using linear systems with AWGN and threshold detection show that AIS is more efficient than either of the linear techniques and more robust in the selection of parameters[37]. The PIS method is seen to approach the optimum but is very sensitive to parameter choice. For example, when the threshold uncertainty is within 1% PIS provides a greater increase in sample size savings than AIS. When the threshold is known only to 10% AIS provides consistently better improvements in sample size reduction.

4.2.6 Conditional Importance Sampling (CIS)

Conditional Importance Sampling (CIS) is an adaptive variation of IS that changes the bias applied to some input PDF conditioned on the realizations of some other random sources. This permits the shaping of a biased joint input PDF to match the optimum solution[11]. The principles of applying this method to nonlinear communications systems are presented in [11] which reports the improvement over unconditional (non-adaptive) IS to be significant.

We adopt the example used in [11] in order to explain the concept. Consider a baseband binary system with phase noise and AWGN. Let the random variable for the phase be ϕ. The phase noise has a symmetric PDF assumed to be

$$f(\phi) = \frac{1}{2\pi I_o(\alpha)} e^{\alpha cos(\phi)} \tag{4.69}$$

The signal is $\pm A$. The signal, the phase noise and the additive thermal noise are assumed independent. The threshold device chooses $\pm A$ relative to a threshold set at zero volts. Without additive noise, the signal component at the decision device is $\pm A \cdot \cos(\phi)$ depending on whether a $+A$ or $-A$ signal was sent. Let us concentrate on the hypothesis that a $+A$ signal was sent, hence the voltage at the decision is $A \cdot \cos(\phi)$, a random voltage. For each value of ϕ there are a range of possible additive noise values which when added to $A \cdot \cos(\phi)$ will produce a negative value. This becomes a received voltage below threshold when it should be above. The concept of CIS is to generate a random variate ϕ based on the PDF in eq.(4.69) and then, conditioned on the ϕ generated, to limit the possible random noise variates to those guaranteed to cause an error. The error region in this case would be those values of $N < -\cos(\phi)$. In other words, the PDF used to generate the random variates for the noise is conditioned on the value of ϕ generated. This combination presents an error condition to the detector at every trial.

In summary, CIS describes a technique which is applicable to the case when there is more than one error (noise) input PDF to the system. The first random variate is drawn from the undistorted PDF. The second random variate is drawn from a PDF which has been conditioned, usually by controlling the range of possible output values, by the value of the first random variate so that the combination of the two will occur in the space guaranteed to cause an error. This is fine in principle and seems to work great in the examples used by the authors in [11] but it still relies on knowing what the error region is. That is, one must know the conditional dependence and, just as important, how the conditional dependence is transferred through the system to the decision device if the adaptive technique is to be programed and

implemented.

The CIS technique was compared to SIS and IIS where it was shown that the performance of CIS is best[11]. It is also shown that IIS (shifting) can lead to performance that is worse than classical Monte Carlo. This side effect of shifting too far was identified in [14] and used in an iterative method of determining the optimum shift.

4.2.7 Efficient Importance Sampling (EIS)

Efficient Importance Sampling (EIS) is a technique developed to mitigate the effects of system memory. "The basic idea behind EIS is that if we have a linear system and L additive Gaussian noise sources, the sampled noise value at the decision point is still a Gaussian random variable. If we can characterize that Gaussian random variable directly, the deleterious effects of memory disappear because in the weight formula each of the densities will be one dimensional"[18]. But, of course, a linear system with Gaussian inputs can be analyzed mathematically, i.e. without simulation. The extension of the EIS principle to nonlinear non-Gaussian systems is given much treatment in [17], [18], and [22].

EIS is based on the following development. Let V be the output of the receiver at the point of the decision device. This is a random voltage with a PDF $f_V(\mathrm{v})$. The probability of error (BER) is given by

$$\mathrm{P}_e = \int_{\nu \in \Omega} f_V(\mathrm{v})\, \mathrm{dv} \tag{4.70}$$

where Ω is the region of the output variable that corresponds to an error. In the fashion typical of IS we define a function H(v) so that

$$\mathrm{H}(\mathrm{v}) = \begin{cases} 1 & \mathrm{v} \in \Omega \\ 0 & \mathrm{v} \notin \Omega \end{cases} \tag{4.71}$$

We now have

$$\mathrm{P}_e = \int_{-\infty}^{\infty} \mathrm{H}(\mathrm{v}) f_V(\mathrm{v})\, \mathrm{dv} \tag{4.72}$$

The remainder of the development follows the typical pattern in that we construct a new bias function from which to draw the random variates for our simulation, call it $f_V^*(\mathrm{v})$. Mathematically this is

$$\mathrm{P}_e = \int_{-\infty}^{\infty} \mathrm{H}(\mathrm{v}) \frac{f_V(\mathrm{v})}{f_V^*(\mathrm{v})} f_V^*(\mathrm{v})\, \mathrm{dv} \tag{4.73}$$

Eq.(4.73) is the same as eq.(4.72) as long as $f_V^*(v)$ is well behaved. As is usual with IS derivations the probability of error is the expectation of the weighted function H(v)

$$P_e = E_*\left[H(v)\,W(v)\right]\,dv \tag{4.74}$$

where E_* is the expectation taken with respect to $f_V^*(v)$ and W(v) is the weight. In the simulation process an estimation of this expectation is made through counting weighted errors.

The difference in this development and the IS development as it usually appears, for example as in Section 4.1.1, is that typically the development is related back to the input PDF, the one that is known to the person performing the simulation. It is not usually left in terms of the output PDF as seen in eq.(4.70) through eq.(4.74) above. This is known as the *output* version of importance sampling[17].

It is shown in [17] that the variance of the *output* importance sampling estimator is a function of only the single weight of this biased output random variable v. This is equivalent to a memory length of exactly one in contrast to the *input* version of importance sample normally described. When the weight used for the unbiasing of the IS error count is actually a multiplication of K independent weights from the input of the system, as it is for the input version of IS, then the variance of this estimator will be greater than if the weight is derived from a single term at the output as above. Of course the trouble with this approach is that we do not know *a priori* what the output PDF $f_V(v)$ is. If we did then there would be no need for simulation.

This approach/method is developed from the start assuming a linear system with L additive Gaussian inputs. In this case it is known that the output noise is Gaussian since the inputs are. The full development can be examined in Section V of [17]. The incongruity evident from our brief explanation above is treated. It is shown that the *output* weight function

$$W(v) = \frac{f_V(v)}{f_V^*(v)} \tag{4.75}$$

can be formed using the fact that for a linear system the output can be written as

$$\nu = \sum_{j=0}^{K-1} h_j\, x_j \tag{4.76}$$

Here, the $\{h_j\}$ is the (perhaps estimated) impulse response and the x_j form the input noise sequence. Thus the K-dimensional noise vector that affects the output is still under the control

of the simulation user but knowledge of $\{h_j\}$ permits it to be combined into a single equivalent noise sample at the output.

This approach does indeed mitigate the effects of memory. It is limited to linear systems or as the authors put forth in [17] "linear statistically equivalent systems". This amounts to a linearization of the input response of the system. This equivalent or pseudo impulse response can be be estimated within the simulation itself through a statistical regression. Examples and a discussion of fitting to a linear regression model by a pre-simulation 'sounding' procedure is given in [17]. EIS and SIS are compared thoroughly in [22] where the method is seen to work and perform well against SIS for the examples given. It is demonstrated to give reasonable results for systems that could not be considered linear. The reader may note that the only reported use of EIS is by the authors of [22][18][17].

In summary, EIS is a technique whose primary purpose is to mitigate the effects of memory. It relies on a linear system, or at least a "linearized" one, in order to process the input noise samples through the system, as in eq.(4.76). Note that the determination of the impulse response coefficients $\{h_j\}$ has the added side effect of determining precisely what the system memory is. Knowing K is required for performing IS since it indicates how the weights should be calculated. This is true for EIS but also for the other *input* versions of IS as well since the weight used in the error counting process is the product of the last K input weights, assuming they are independent.

4.2.8 Combination of EIS and PIS (MEIS)

There have been only two references to applying IS to M-ary communications[10][33]. In [10] simulation of 16-QAM and 64-QAM systems by combining EIS and PIS is discussed. In using EIS the authors assumed a Gaussian noise density at the input to the decision device. An equivalent noise source was developed by sounding out the system prior to simulation. They develop an equivalent noise source by simulating a Gaussian input through a linear system to obtain the variance at the output. After determining the equivalent noise source, it is biased by forming a PIS type PDF from the Gaussian. The non-Gaussian PIS noise generator is a simple read from a pre-stored file. Even though this is technically an M-ary simulation, each symbol is treated individually so that in the final analysis this is simply another case of AWGN with threshold detection.

Many of the developments are incomplete or seem flawed. For example, a method for

choosing δ, the assumed cut off point for the non-Gaussian PDF (see Section 4.2.4), is contradictory. An expression is given for determining δ given the percent number of errors AND V_T. If you know V_T then you know where to put δ! BER plots and time comparisons of MEIS to EIS and classical MC are given.

4.2.9 Householder Importance Sampling (HIS)

The deleterious effects of system memory on IS simulations are well known. A method to mitigate the effects of system memory through the controlled correlation of input samples has been established[13]. The technique is applicable to linear systems with i.i.d. zero-mean Gaussian input noise. The transformation is accomplished by processing the noise through a suitably constructed orthogonal Householder transform. Hence the name we have chosen for it here, Householder Importance Sampling (HIS). The transformation of the system results in a sample size reduction equivalent to that obtainable for the IS simulation of a system with a memory of 1 without changing its statistical properties.

The construction of the Householder matrix requires some knowledge of the system transfer function which can be obtained through a sounding procedure, given in [13]. This system function, h, is also mentioned in [40] where it is used to determine a time varying mean shift to be applied to the noise distribution. The sounding of a system to obtain its impulse response is of importance for EIS as well[17][18][22]. HIS is very similar to EIS in its operation. A detailed comparison would be required in order to determine the extent of the similarity. The relationship, if any, between these results would be an interesting analysis to make but has not been performed here since these linear system results are of marginal interest to the optical PPM problem.

This technique has been applied to a nonlinear digital satellite channel where both up-link and down-link noise are present[33]. The system studied transmitted 4-PSK and 8-PSK modulated signals. As in [10], the M-ary modulation detection reduced to a threshold detection problem in the final analysis and in the application of IS.

4.3 SYSTEM MEMORY AND MULTIPLE NOISE SOURCES

4.3.1 System Memory

The memory of a system will decrease the improvement, i.e. the reduction in required sample size, available through importance sampling. For example, the computation of the ratio

of $\frac{N}{N^*}$ using the variance increasing IS method (SIS) showed that, for a P_e of 10^{-6}, the sample size reduction for a unit memory simulation is on the order of 25,000 and for a memory equal to five samples, it is 1000[3].

System memory, in the context of this report at least, refers to the number of past input samples which have some influence on the current bit decision. In the optical systems of interest here this is generally a result of filtering effects. Digital filter modules implemented with IIR structures in reality have an infinite memory, although the influence of past input samples on the current output diminishes with time. The simulation of signal synchronizers and implementations of Viterbi decoders result in systems with extremely long memories as well. These devices are not currently of interest in this work. In M-ary PPM, as M increases the number of simulation samples required to represent the waveform of the symbol generally increases. This is because a PPM word is made of a pulse and $M-1$ non-pulse time slots. If it is decided that it requires 8 samples to represent the pulse, then in order to maintain uniform sampling throughout, there will be $8 \cdot M$ samples required to represent the entire word. Our work to date indicates that all of the samples in the PPM word which resulted in a bit error must be considered in the IS weight calculation procedure. System devices with long memory, filters for example, may necessitate the consideration of more samples but, at a minimum, the number of samples required to represent the PPM word must always be considered. Hence the affects of memory are of importance to this work. A review of existing methods of dealing with system memory is the subject of this section.

The discussion above concerning the number of bits in a M-PPM word increasing with M brings up an interesting point that should be developed at the start of any consideration of memory. Subject to certain constraints, the amount of memory in terms of samples is under the direct control of the system designer. An example will illustrate. If the system memory, due to filters and equalizers is m bits long then the memory in terms of samples is $K = mT_s/dt$ samples long where T_s is the time length of a bit and dt is the time between samples. The simulation designer has direct control of the sample time dt. If one chooses to simulate a system using 8 samples per bit versus 16 samples per bit then one has cut the memory in half. Of course there are limits to how few bits can be used to represent a waveform and maintain some level of confidence that the pertinent properties of the waveform have been preserved. These limits are dictated by Nyquist's sampling theorem and a judgment of how much bandwidth is required to be maintained throughout the simulation in order to preserve the authenticity of the measurement. Digital signals present an interesting problem in that pure square waves have an

infinite bandwidth. In practice, 8 to 20 samples per bit has been suggested as sufficient for such signals.

The effect of a system memory greater than one on IS performance, and methods of mitigating its effects, have been amply studied and analyzed[3][17][18][22][13][33][12][19][20]. A very enlightening analysis which graphically illustrates the effects of memory on sample size reduction is found in [17] and also in [18]. Here, using the concept of the system threshold characteristic, an expression is derived which can be used to calculate the sample size improvement ratio of IS over classical Monte Carlo as a function of increasing standard deviation bias, i.e. SIS. While the results presented in [17] and [18] are calculated in terms of the IS parameter governing the amount of variance increase for SIS, we have observed similar results as a function of C where C is the amount of shift applied to input PDF's. Indeed the effects illustrated in these graphs, that of the optimum IS parameter being a function of memory and specifically that at larger memories significant IS improvement occurs over a relatively limited range of the shift parameter, was used as a method of finding the optimum parameter through repeated short simulation runs[14]. The analysis, displayed on a graph in Fig. 4.6, shows a marked decline in sample size savings with an increase of memory assuming a constant P_e estimation of 10^{-8}. The improvement ratio decreases from over 10^6 for K=1 to ~ 5000 for K=10 to less than 10 for K=60. Further, and of equal interest, the optimum IS bias parameter decreases with increasing memory. In other words as the memory of the system increases, the amount of shift or amount of increase in the variance which is optimum becomes smaller and so does the improvement over classical Monte Carlo.

To compound the situation, the exact memory length is often not known prior to simulation. At other times it is known but known to be effectively infinite. This can happen with some signal synchronizers, Viterbi decoders and, of particular interest here, with IIR implementation of filters within the system simulation model. While the effect of prior samples on the current output diminishes in most of these cases as the samples get older, a finite value must be selected as a cut-off for use in calculating the weight in a practical simulation. To explain the effect of using the incorrect memory on IS simulation, let us consider an output variable Y which is a function of K independent input variables X_1, X_2, X_3, ..., X_K, that is

$$Y = g(X_1, X_2, X_3, \ldots, X_K) \tag{4.77}$$

where $g(\cdot)$ represents the system transfer function and is an ordinary function of K variables which may be linear or nonlinear. Following an analysis first found in [3], let us compute the

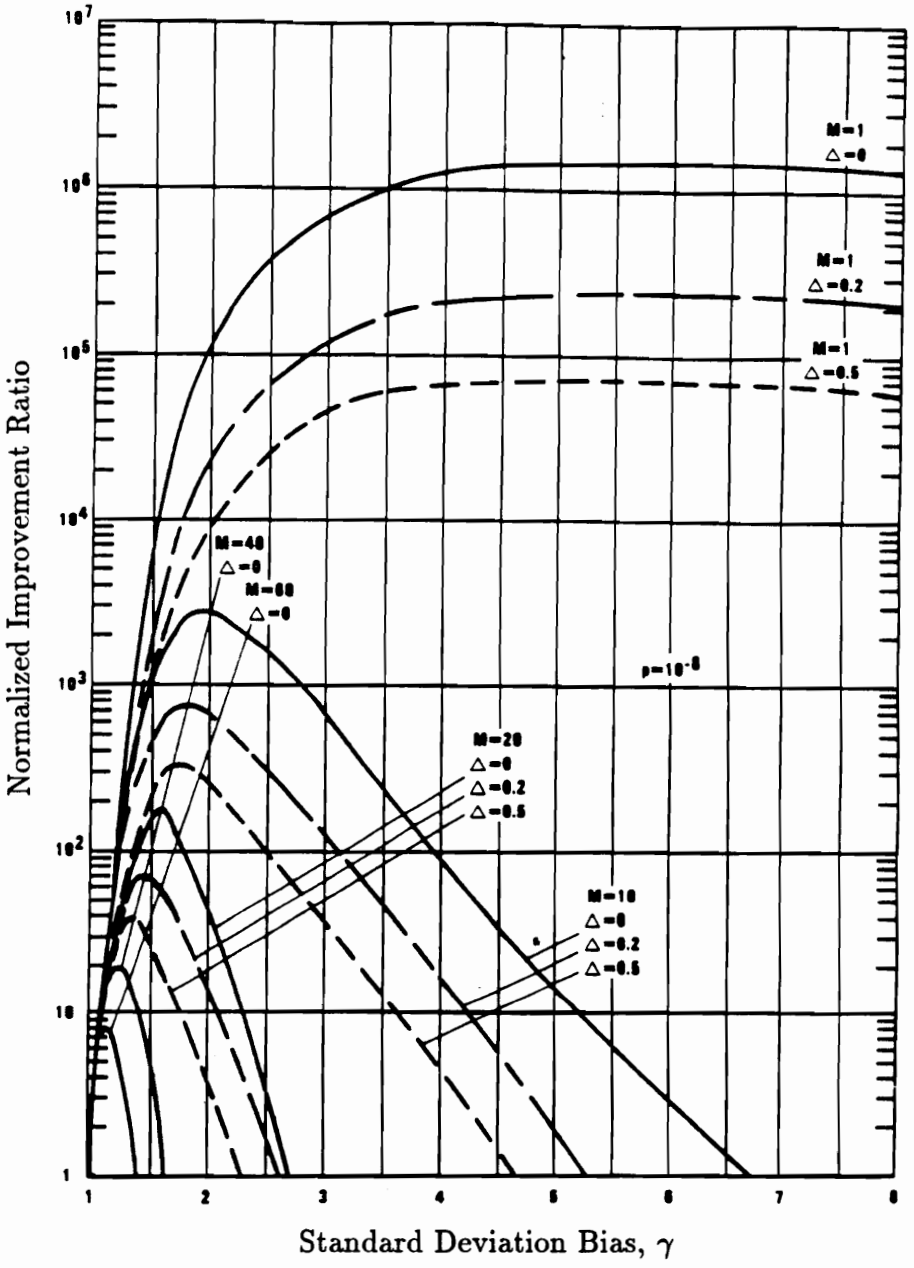


Fig. 4.6 IS Improvement as a Function of IS Bias Parameter for Various Values of Memory (M) and Relative ISI (Δ), for $P_e = 10^{-8}$ where the ordinate is in units of normalized improvement ratio and the abscissa, γ, is the increase in standard deviation of input bias function. (Fig. 4 from Hahn and Jeruchim (1987)[17])

bias of Y and examine the effect of memory on the IS estimator's performance.

Let $$y_i = g(x_{1i}, x_{2i}, x_{3i}, \ldots, x_{mi}, \ldots, x_{Ki}) \tag{4.78}$$

Each input sample x_{mi} is selected from the biased PDF $\overset{*}{f}(x)$ so that the weight of each input sample relative to the unbiased PDF $f(x)$ is

$$W(x_{mi}) = \frac{f(x_{mi})}{\overset{*}{f}(x_{mi})} = W_{mi} \tag{4.79}$$

Since the i^{th} output sample y_i is determined by the value of K independent variables, the probability of y_i is increased or decreased by the product of the increased or decreased probability of occurrence of each of the K input samples. The weight of y_i is the inverse of this increased or decreased probability. Mathematically we have

$$W(y_i) = \prod_{m=1}^{K} W_{mi} = W_i \tag{4.80}$$

Recall that the unbiased estimator from an IS based Monte Carlo simulation is

$$\hat{P}_e = \frac{1}{N^*} \sum_{i=1}^{N^*} W_i \tag{4.81}$$

which was an approximation to the expected value

$$E_*[H(y)W(y)] = \int_{-\infty}^{\infty} H(y)W(y)\overset{*}{f}(y)\, dy \tag{4.82}$$

Suppose that the memory length of the system is not known *a priori*. This is not an unusual occurrence. Let us investigate the effect of calculating the weight W_i from eq.(4.80) using a value of K which is greater than the actual system memory. If the weight computation is carried out using a memory of K+1 instead of the correct memory of K then the computed weight of y_i is

$$W^+(y_i) = \prod_{m=1}^{K+1} W(x_{i-m+1}) = W_i \cdot W_{i-K} \tag{4.83}$$

where W_i is the correct weight for y_i. Notice that the expected value of W(x) over all x is

$$E_*[W(x)] = \int_{-\infty}^{\infty} W(x)\overset{*}{f}(x)\, dx = 1 \tag{4.84}$$

so that the value of $W^+(y_i)$ averaged over all possible values of W_{i-K} will be equal to W_i. Therefore the estimator in eq.(4.81) and eq.(4.82) will be unaffected, that is, unbiased. However the variance of the estimator will be enlarged due to the required averaging of the excess weight(s).

On the other hand, if the weight computation of eq.(4.80) is made using a memory length of K − 1 instead of the correct length K then the computed weight of y_i is

$$W^-(y_i) = \prod_{m=1}^{K-1} W(x_{i-m+1}) = \frac{W_i}{W_{i-K}} \tag{4.85}$$

where, again, W_i is the correct weight for y_i. The expected value of $\frac{1}{W(x)}$ over all x is

$$E_*\left[\frac{1}{W(x)}\right] = \int_{-\infty}^{\infty} \frac{1}{W(x)} f^*(x)\, dx = \int_{-\infty}^{\infty} \frac{f^*(x)}{f(x)} f^*(x)\, dx \neq 1 \tag{4.86}$$

since $f^*(x)$ is generally greater than $f(x)$. The value of $W^-(y_i)$ averaged over all possible values of $(W_{i-K})^{-1}$ will be not be equal to W_i and the estimate of the tail probability will be skewed, i.e. biased. Hence, we have shown that if the memory is selected too large, the result will be correct but the efficiency of the IS simulation will be compromised. If the memory used to calculate the weight is too small then the answer will be biased.

Two importance sampling methods have been developed with the specific intent of mitigating the effects of memory. These are the EIS method and the HIS method described in Sections 4.2.7 and 4.2.9 respectively of this report. The objective of each is to effectively make the memory length of the system equal to one, where the maximum improvement through IS is obtained. EIS does this by "pushing" the bias through a linear system, or a statistically linearized equivalent, and applying the bias at the output, i.e. at the input to the decision device. An experimental comparison of SIS and EIS which illustrates, first, the improvement due to use of EIS and, secondly, the deleterious effects of memory on IS simulation has been performed[22]. The EIS method and PIS were combined to simulate a satellite transponder[10]. As for HIS, the method processes the noise through a Householder transformation based on the system response so that the noise is biased at the input but that the effect is still that of a system with memory of one. The reader is referred to the sections of this report describing these methods for further details and references.

Since memory has such a large effect in the simulation of Viterbi decoders, there has been much activity by those simulating these devices to develop methods of mitigating the

effects of system memory on the benefits of IS[12][19][20][34][35]. A rule is given in [20] for determining the amount of memory to use, i.e. what a suitable memory truncation point is for minimal estimator bias balanced against increased estimator variance, for the IS simulation of (n,k,m) convolutional codes through a Viterbi decoder. Truncation of long or infinite memory is also addressed in [17], see Section 3 B. In general, two approaches to the simulation of Viterbi decoders have appeared, one is the *block method*[19][20] and the other is known as an *error event simulation method*[34][35]. Signal synchronizers, i.e. phase locked loops, are often called out as systems with long memory in the IS literature but there have been no works addressing the IS simulation of these devices.

4.3.2 Multiple Noise Sources

Multiple noise sources often occur in the simulation of realistic systems. This has often been associated with satellite communications links and multi-hop communication links wherein noise is added at the satellite transponder and again at the ground terminal receiver. This scenario has seen treatment in the literature[23][18][22]. The noise in radar system simulation is often the result of multiple input noise sources and has been studied[4][32]. Other treatments of multiple inputs have arisen in the context of general considerations concerning the implementation of IS[17][40]. Our interest in multiple noise inputs concerns both the receiver model and the simulation of a channel and channel effects. The optical receiver has two noise sources. The first is the signal dependent shot noise of the APD device. The second is the additive Gaussian noise of the pre-amp following the APD. In the simulation of a free-space system through the atmospheric channel, the channel causes a random time varying change in the amplitude, angle of arrival and phase of the received signal. This then would constitute yet another noise source in the simulation.

Several questions arise in applying importance sampling to the simulation of systems with multiple noise inputs, some of which have been addressed in the literature. For example, do all the noise sources in the system need to be biased? Are they all biased the same? The answers to these questions have been examined in the literature. We briefly report some results.

It has been shown that the variance of the IS estimator increases as the number of independent sources increases[17]. It is logical and has been shown that the memory of the system goes up as the number of independent sources increases[17]. Assume for a moment that all the noise sources are equally biased; it has also been shown that the amount of biasing applied to the noise sources needs to decrease as the number of independent noise inputs

increases[32]. These three statements are in harmony with a similar finding concerning increasing memory where it has been shown that increasing memory causes the optimum amount of IS bias to decrease and the potential amount of IS improvement to decrease[17][18]. Hence it seems that increasing the number of input noise sources produces the same side effects on IS performance as increasing system memory. We invite the reader to reconsider Fig. 4.6 in the light of these statements concerning multiple noise sources.

The question exists as to whether all of the noise sources need to be biased. This has been investigated and the following observations and guidelines result. The first observation seems obvious yet needs to be stated; if an input is not biased its memory will not affect the estimator's variance[17]. It should be noted that each noise input generally has a different memory associated with it as a result of physical separation in the system model. For example, in the one-hop satellite link the noise source added at the transponder has a memory determined by the filters in the transponder and satellite HPA models combined with the filtering in the ground terminal receiver model. The noise source added at the ground terminal "sees" only the filtering of the ground terminal receiver. Hence, in general, this noise source will have a different memory than the noise source at the transponder.

In a system with multiple noise sources, each source has a different level of influence on the generation of an error, in general. If a noise source is a significant contributor to the overall system error performance then the variance of the estimator will not be reduced significantly by applying IS only biasing to the other, less significant, input sources[17]. This seems reasonable when you consider that the act of not biasing the primary error producing input is akin to performing Monte Carlo simulation without importance sampling. The difficulty lies in determining the relative importance of each input noise source on error generation within the model. No guidelines for making such a determination can be found in the literature. Perhaps this is because it is too system specific to address.

An interesting experiment was carried out to explore this relationship between the amount of biasing and the level of contribution by the noise source to output errors[17]. It proceeded as follows: suppose we have two zero-mean Gaussian noise sources x and y that combine additively to produce a composite noise. Consider

$$n_d = \alpha x + \beta y \tag{4.87}$$

where we set $\alpha^2 + \beta^2 = 1$. If the $\mathrm{E}[x^2] = \mathrm{E}[y^2] = \sigma^2$ then the relationship between α and β will cause the $\mathrm{E}[n_d^2] = \sigma^2$ as well. The ratio of α/β is thus a measure of the relative importance of

the noise sources in causing errors. A simple integrate and dump filter was assumed for the system and the relative power levels were set to ensure $P_e = 10^{-6}$. Assuming that only x was biased and using equations developed in [17], the normalized variance ($N \cdot \sigma^2[\hat{P}_e]$) versus the relative ratio of β/α, i.e. the relative importance of y to x to the constant total power of n_d, was plotted. This plot can be seen in Fig. 4.7. For reference, the normalized variance for the case of no bias, that is no IS, is plotted and also the normalized variance when both x and y were biased equally. These are the dashed lines at the top and bottom of the graph respectively. It is seen that the act of not biasing y when it is an important contributor to the output greatly increases the variance of the estimator, by approximately three orders of magnitude over the equal biasing of both processes. The interesting phenomenon seen from this experiment is that when y is not a significant contributor, less than approximately $0.2\,x$, it is better to not bias y! It is conjectured that when the ignored variable has little effect on the outcome of the experiment, its inclusion in the weighting merely produces additional variance in the estimator[17]. This latter point is well taken for APD based optical receivers since the additive noise from today's low noise preamp devices is often dominated by the APD shot noise, especially when the APD is used at high gain.

Mitchell[4] was among the first to address multiple input sources in the radar and communications system context. In his classic work, he illustrated that IS improvement was possible when more than one source was involved through the use of several interesting and pertinent examples. Jeruchim[23] again showed the validity of IS in the multiple noise source environment and suggests the semi-analytic treatment of the last noise source in satellite multi-hop simulation if the final receiver can be made linear. Hahn, et al.[18] first pointed out that neglecting an important noise source can greatly reduce the IS improvement ratio. It is important to note that the use of EIS, when it can be applied, effectively combines all sources at the end of the system and eliminates the need to determine the optimum relative bias between multiple input noise sources. Parhi and Berkowitz[32] derived the criteria for optimally biasing an arbitrarily weighted sum of independent exponential variates. They show that the PDF biasing needs to decrease as the number of independent noise sources increases. Hahn and Jeruchim[17] present probably the most complete treatment of multiple input noise sources to date. We have quoted most of their results above. Jeruchim, et al.[22] considered a satellite link with one hop and two input noise sources. They almost immediately assumed that the noise sources were equally biased and carried on with the analysis. The only justification given was for the convenience of simplifying the mathematics. Results from [17] and [40] seem to

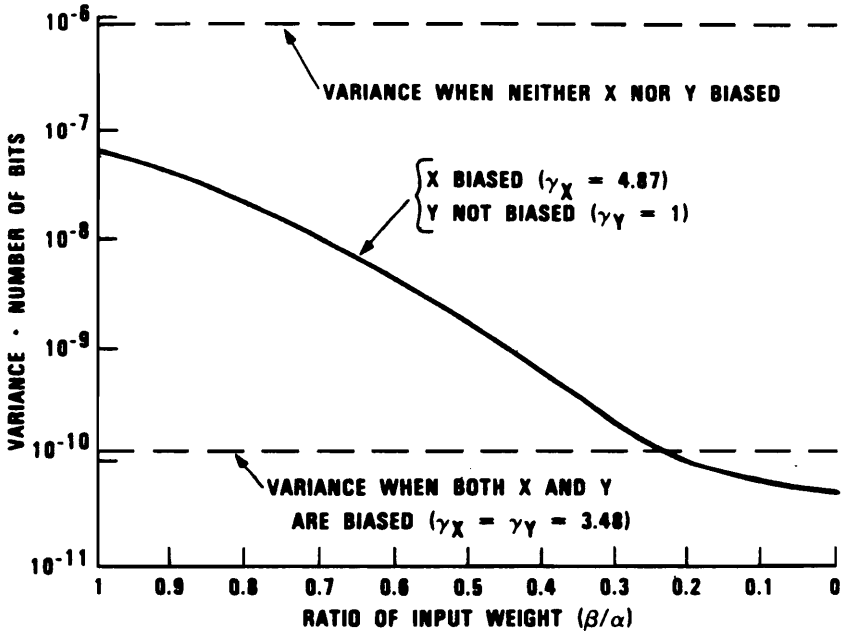


Fig. 4.7 **Variance of the BER Estimate When One of Two Error Contribution Inputs Remains Unbiased,** where the ordinate is in units of normalized improvement ratio and the abscissa is the ratio of the relative importance of noise source y to noise source x. (Fig. 5 from Hahn and Jeruchim (1987)[17])

indicate that unless the relationship between the sources is known, equal biasing is at least safe. Wolfe, et al.[40] devoted one paragraph to the multiple input system as part of their sequence-dependent IS analysis. Having shown that one of the best strategies of biasing through shifting was to make the shift time varying in accordance with the impulse response of the system, they extend this result to say that each noise source should have a time varying shift in accordance with the impulse response "seen" at that particular point in the system by that noise source input.

4.4 VARIOUS SYSTEMS

This section provides a measure of completeness to this review of the importance sampling literature. Its purpose is simply to group the IS works according to the type of system to which it has been applied. It is intended to permit the reader to quickly determine which references are concerned with a system of interest. Some of the works discussed below are not discussed elsewhere in this report. A number of them contain some original and unique approaches to the simulation of certain systems with unique problems. It is important to note that maximum likelihood detection of the type of interest to this research has not been considered elsewhere.

4.4.1 Digital Detection

Digital threshold detection has been the primary focus of most of the works referenced in this chapter and throughout this work. This category dominates the literature on IS by far. In some sense the categories in the remainder of Section 4.4 simply represent some of the easily identifiable subcategories of digital detection which may require special attention or unique consideration. Many of the digital receiver simulations were oriented toward the ground to satellite transponder to ground link[21][22][23][33]. A general threshold receiver architecture without specific application was the object of analysis in many cases [8][9][10][14][25][28][3]. In other cases the system was modeled as a general decision making device over an unspecified region Ω of the decision variable[17][18][26][29][37][39][40]. One unique example is the work which considered both phase noise and threshold detection in a generalized receiver application[11]. The papers dealing with fiber optic communication systems all assumed threshold detection[2][16][38].

4.4.2 Viterbi Decoders

Viterbi decoders represent systems with long memory. The memory is infinite in theory, that is, the current bit decision is based on the input of all the previous bits through the decoder. But the effect on the current decision is not equal for all of these bits, the most recent bits affect the current bit decision more than the earlier. What then should be the relative weighting of the input bias weights to the error counting process? The problems associated with simulating Viterbi decoders using IS are addressed in [19] and [20]. Here, the IS simulation of (n,k,m) convolutional codes through a Viterbi decoder is studied and guidelines for truncating the memory used for weight generation is given.

A Viterbi decoder study can also be found in [34] which develops an *error event simulation method* which is more robust than the *block method* developed in [19] and [20]. This method involves the use of a linear translation based on the distance properties of the code and knowledge of the Viterbi decoder error event phenomena. See also [35] for discussion related to the simulation of these systems.

The IS simulation of Viterbi decoders and long memory effects are also studied in [12]. This work describes three bias schemes applicable to Viterbi coded systems, i.e. systems with long memory. It presents a partial weighting scheme in which the biased input is applied in bursts for a number of samples less than K, the memory. This is a modification of conventional IS to give improvement due to long memory. They also present, in the context of Viterbi decoders, two simple extensions to SIS biasing (increasing the variance). The first is to vary the mean (shift) as well as scale and, second, to use a non-Gaussian bias density. As they study a complex valued system, they apply these methods separately to each of the quadrature channels. The result is that they consider three bias routines. The first, called *conventional*, is to simply increase the variance of both the I and Q channel signals. In the second, *star*, the variance is increased in one dimension and the mean is shifted in the second. The third, *cross*, constitutes the use of a non-Gaussian density function in which the density is uniform with zero mean in one dimension and a mean of $\pm\frac{1}{\sqrt{2}}$ in the other dimension. The *cross* bias method was found to cause estimator bias. No explanation was offered.

4.4.3 Radar - False Alarm Rates

The IS simulation of radar false alarm rates and associated radar performance parameters was one of the first applications of IS considered in the communications engineering

literature[5][4]. The realistic simulation of radar systems involves the use of non-Gaussian PDF's and often the system is one in which multiple noise inputs exist. IIS was discussed in the context of radar systems simulation in [27].

False alarm thresholds and MTI delay line cancelers are studied in [32]. In this work, the criteria for optimal IS simulation of an arbitrary weighted sum of independent exponential variates is derived. The method involves the decorrelation of the clutter in order to perform IS. This is required in order make the input clutter samples independent. This is a multiple input study as well. It demonstrates that the amount of PDF biasing needs to be reduced as the number of input variates increases for a constant threshold.

4.4.4 DS-SSMA

Direct Sequence-Spread Spectrum Multiple Access systems constitute systems with non-Gaussian input noise statistics. The focus of most of the simulation work in this area has been in predicting the effect of a random number of uncorrelated and random phased background signals on the performance of an individual user. Analytical methods exist for placing bounds on this problem but simulation has been considered as a means of analysis as well. IS has been applied to the simulation of DS-SSMA systems in [31], [34] and [35]. Each of these papers provides insight to and references for the problem of simulating DS-SSMA systems.

4.4.5 Large Networks

In this age of ISDN and global communication, networks and in particular large networks are increasingly under study. The simulation of the signal routing behavior of these networks and the switching systems inside them is becoming a standard tool for analysis. The application of IS simulation to these networks has been considered in [15] and the references therein.

4.4.6 Adaptive Equalizers

The IS simulation of an adaptive system presents a unique challenge. It is the intent of IS to artificially but realistically and in a controlled fashion create more errors than would normally occur under a given set of conditions. It is the intent of an adaptive equalizer to compensate for, to adapt to, the incoming signal and to reduce these errors. A unique solution to this quandary is presented in [6]. The adaptive system is simulated via the use of 'twin

systems,' each running in parallel with the same input signal. One system, the IS system, is fed biased noise samples in order to create additional errors. But the equalizer tap values of the IS system are updated from the non-IS system in order to prevent the IS system from compensating to the unrealistic errors.

4.5 REVIEW OF OPTICAL SYSTEMS SIMULATION

4.5.1 Optical Systems Simulation Using Importance Sampling

The first reported use of importance sampling in the simulation of optical communications systems is reported by Balaban[2]. This paper is directed toward traditional fiber optic communications systems and used optical non-return to zero (NRZ) pulses with threshold detection as an example. The approximation due to Webb, McIntyre and Conradi[48] was used for the APD statistics. As an aside, this work seems to have originated the name "WMC PDF" for this approximation. (As pointed out in Chapter 2, this PDF is actually an Inverse Gaussian density function.) In Section 3 of [2], they perform a numerical solution for the BER of a system using the WMC PDF and compare this to an analysis using a Gaussian PDF for the APD statistics.

As for Monte Carlo analysis of the system, they simulated a system with a very defined and finite memory length filter. The biasing of the WMC PDF is performed by altering the values in the WMC PDF table which is used to generate WMC random variates. They describe two biasing methods, one in which the biased PDF is made quasi-uniform over a range of the input variable and the second where the bias is a constant multiplier, and hence a constant weight, over each interval of the input variable. They used the second method to bias the tails of the densities as this was threshold detection. The amount of biasing was by trial and error. Both the APD and the additive Gaussian noise were biased using generally the same method. The output was taken in a histogram format and smoothing was performed using asymptotic functions to find the best fit of a known function to the tails of the output density function.

This is the first work which simulated an optical system using importance sampling. As such it is the first work to consider biasing of the APD statistics. They did not resort to a Gaussian approximation for these statistics as was typical at the time. The work presented in this dissertation is an improvement over that found in [2] in that the biasing of the APD is performed without table modification and is thus faster and more convenient to use. Further, the resulting estimator is accurate enough to represent the answer without the added

complication of curve fitting to a histogram. The error associated with this process is not discussed in [2].

The shifting of the APD density function was presented in [14] with very few details concerning its implementation. The reader is referred to Section 4.2.3 of this thesis for a detailed discussion of this investigation. The primary purpose of the work was to present their iterative method of determining the optimum shift for IIS simulation. An in depth discussion of APD biasing was not presented.

4.5.2 Semi-Analytical Optical Systems Simulation

There are two recent works in the literature discussing the simulation of optical systems which do not promote IS but do use some form of semi-analytic treatment in order to minimize simulation time. These are Elrefaie, Townsend, Romeiser, Shanmugan [16] and Townsend and Shanmugam [38]. Each of these is reviewed in this section.

The work by Elrefaie, Townsend, Romeiser, Shanmugan[16] presents what the authors call a "hybrid" approach, which means that it combines simulation with analysis. Analysis is used to account for thermal noise. There are essentially three simulation topic areas addressed in this work 1) baseband signal generation, 2) optical waveform generation and shaping, and 3) application of noise statistics to optical signals. Each of these topics is also of interest to this research. The system under study is a single-mode fiber system. Excellent models for optical source waveforms, fiber attenuation, fiber dispersion and receiver noise are presented. Results are presented for both LED and laser sources, using both p-i-n and APD detectors. According to the authors, the advantage of simulation over purely mathematical analysis is that simulation permits the use of complicated models for system elements, permits waveform comparison with measured data and that, if desired, measured data can be used in the simulation. In their "hybrid" approach, what is commonly referred to as a semi-analytic approach, analysis is used to decrease computation time for certain statistically based phenomena. In this case, it is thermal noise.

The simulations in [16] were performed using SYSTID[49][50]. Receiver thermal noise is handled analytically since the receiver is assumed linear prior to the detector. Shot noise is nonlinear and is therefore simulated. As discussed earlier, in order to account for ISI the input bit pattern must contain all possible combinations of K bits where K is the memory of the system in bits. These authors claim that a K in the range of 6 to 10 is sufficient to account for

the ISI phenomenon found in single mode fibers and optical filters.

An interesting and detailed method of modeling the source waveform is presented in [16]. It is noted that laser sources require modeling of mode partition noise, chirping and reflection. The optical fiber model accounts for dispersion in the signal. The references concerning fiber dispersion have some applicability to free-space dispersion. The simulation results show that fiber dispersion causes a greater penalty on the APD receiver than the p-i-n receiver. This is believed to be due to the higher excess noise generated out of the APD when the power is increased to compensate for ISI.

The models used for detector simulation in [16] are of interest to our research. The authors in [16] claim that p-i-n shot noise is often neglected since the thermal noise of the pre-amp which follows dominates the error mechanism while with an APD the "shot noise becomes comparable to or greater than the thermal noise." In their p-i-n model, they implemented the shot noise anyway since they wanted to use it in their APD model. The APD model used in [16] begins by calculating the average number of input electron hole pairs per unit time. This value was used as the mean value in a Poisson distributed random number generator. The Poisson distributed random variate was then used to determine the number of McIntyre distributed random variates to be summed. This is the model that has been referred to as the *exact* approach, see Chapter 2 for a discussion. Both the Poisson and McIntyre random variates are generated from look-up table CDF's generated by summations of the PDF's. On the order of 10-1000 calls to the McIntyre CDF are made for each simulation sample. This time consuming processes is somewhat mitigated after the first run by storing the APD value for use during subsequent runs in which the APD parameters are the same.

It was found that when the variance of shot noise is equal to the variance of the thermal noise, 2048 bits are enough to estimate BER down to 10^{-10}. If shot noise variance is greater than thermal variance then more bits are required but the authors are not specific. For the shot noise limited receiver with ISI, they recommend the use of their APD model along with tail extrapolation techniques. An alternate approach is suggested but not examined. This would be to use a Gaussian shot noise model, and then determine the first and second moments at the output using simulation (as in [51] or [52]).

In the semi-analytic approach used in [16], the mean and variance of the Gaussian noise at the threshold detector input is determined via a calibration run using only the thermal noise as input. During the BER simulation, each sample in an output bit is treated individually. At

each sample in the pulse the probability of error is determined by the distance of the output sample from the threshold. This makes use of the Gaussian noise density at the detector which was determined from the calibration run. In other words, a Q-function look-up is perform for each sample in the simulated bit. Each of these is averaged with the probability of bit error at that particular sample for each bit throughout the simulation to obtain a final BER average. Doing this for each sample in the bit enables the study of timing jitter. For example, if there are 16 samples in a bit there will be 16 probability of bit error outputs, one for each time position along the bit. The minimum BER of the 16 shows the optimum sample time and its associated BER.

The work by Townsend and Shanmugam[38] represents the most recent work addressing the simulation of optical systems. The work concentrates on a method of improving the efficiency of digital lightwave communication simulation via 1) an acceptance-rejection method of implementing the APD WMC statistics and 2) by utilizing tail extrapolation to reduce run time. They also implement a semi-analytical treatment of additive thermal noise which is similar to that already discussed in [16] and is not discussed further here. They do not mention the consideration of timing jitter in their analysis. As an aside, the introduction provides a good review of optical link analysis options that have been seen in the literature. As such it provides a reference list for a review of numerical methods as well as simulation approaches.

This work uses BOSS for it simulation efforts and therefore, in some sense, represents a standard to be met or beaten for improving optical simulation run time. They report simulation results down to the 10^{-12} level using 30,000 bits ten times with an acceptable confidence interval using only the semi-analytical treatment. The confidence interval is determined by use of Student's t-distribution. The tail extrapolation analysis is performed with 3000 bit runs. The errors associated with tail extrapolation can be greater than the simulation approaches such as the semi-analytic techniques or the method presented here. In fact the method of performing a tail extrapolation is described in some detail in [38]. A derivation of the validity of performing a tail extrapolation is given which adds new insight into the process to the literature. Tail extrapolation can be performed by using several simulations at pseudo-thresholds; short simulations of 3000 bits in this case. The trouble is that errors in this process can be magnified by the log-log scale used in the analysis and by the extrapolation process itself.

The acceptance-rejection method of Ascheid[7] is more efficient and easier to implement than the clumsy method presented in [38]. Yet, [38] provides the first attempt at efficiently generating the random variates based on WMC statistics. They utilize two comparisons for

judging their generator. First, they compare the run time of their generator to a generator using the exact method, used for example in[16]. Second, they display figures overlaying histograms of 100,000 points generated at a constant power via their generator and the exact approach for a visual comparison.

4.6 SUMMARY

In this chapter we have reviewed all of the pertinent literature concerning the application of importance sampling to communications systems. We have seen that much of the literature is concerned with threshold detection receivers, additive noise and the development of suboptimum bias functions for particular system simulation efforts. Yet this review has provided tools for the application of importance sampling to the non-threshold detection, non-additive noise simulation of optical systems which we can draw upon.

We began with a description of the optimum importance sampling bias function. We have seen that it is formed from a weighted subsection of the original input noise density, the subsection being that portion of the input density which is responsible for errors and the weight being that constant which causes the area under the new density to become equal to one. This optimum bias function is unrealizable since its construction requires knowledge of the the very quantities sought through simulation, namely the probability of error and equivalently the portion of the input density which is responsible for bit errors. The optimum bias is developed from attempting to minimize the variance associated with the Monte Carlo estimator. We have reviewed both constrained and unconstrained derivations of the optimum bias function. Simulation using the optimum bias function, if it could be done, would result in a variance of zero for the estimator.

Through examination of the sequence-dependent derivation of the the optimum bias function we have seen that the memory of the system will limit the minimum number of bits required to be simulated. The minimum number of bits required for simulation is determined from the number of possible signal sequences which in turn is determined by the memory of the system in number of bits. If there are L sequences, each m bits long, then mL bits is the number of bits that would need to be simulated in order to of account for ISI in the system.

We collected a set of properties associated with the mathematics of the importance sampling. These have been developed at different times and in different forms by various researchers. The first of these is the average weight, $\overline{W}$. The average weight has several useful

properties and can be estimated along with the simulation. It is shown that $\overline{W}$ must be less than or equal to probability of error, P_e. It is shown that the variance of the estimator is directly related to the average weight as well as the sample size improvement ratio. The average weight can therefore provide an indication of how well the IS simulation is performing.

A property concerning the maximum weight value is developed which states that the maximum possible weight must be greater than the probability of error to be estimated. When forming a bias function through shifting or increasing the variance, the maximum weight will in general decrease with increasing shift or increasing variance. Hence there is, in general, a maximum amount of shift or variance increase for a given P_e estimate. Beyond this maximum the estimated probability of error will be biased smaller than the true probability of error. This property was identified and used in [14] as a means of determining the optimum shift to be applied to bias the input density through a series of short simulations. The significance of the maximum weight is seen again in the application of PIS. The assumption of an importance region and the maintenance of the same shape of the original input noise density results in a weight value which is a constant. This is the parameter ϵ in eq.(3.58) or eq.(3.59). We know from the properties of the maximum weight that this ϵ must be at least as great as the probability of error to be estimated. This is seen again in the discussion of PIS in Section 4.2.4 where two cases, one in which the assumed importance region is smaller than the true region and one in which the assumed importance region is at least as great as the true region. It was shown in the analysis of case two that if the region is too small, which means that $\epsilon < P_e$, then the estimate will be biased and smaller than the true probability of error.

Finally, expressions for the upper and lower bound of the variance of the importance sampling estimate were presented. These expressions can be used to indicate the validity of the importance sampling simulation and to predict the sample size improvement over classical Monte Carlo. It is interesting that the importance sampling estimator variance is upper bounded by an expression involving the maximum possible weight. From this and the previous properties we see that keeping track of the statistics of the weights generated will provide information concerning the status of the importance sampling simulation.

We reviewed various versions of the importance sampling that have been developed. Most are distinguished by the method of forming a bias function. Scaling Importance Sampling (SIS) refers to forming the bias function by increasing the variance of the input noise density. Improved Importance Sampling (IIS) biases the input density by shifting the mean. Pseudo-Optimum Importance Sampling (PIS) forms an approximation to the optimum bias function by

assuming an importance region. It is shown that this assumed region must completely contain the true importance region. Absolute-Value Importance Sampling (AIS) forms a bias function by folding, through the absolute value operation, the input noise density. The folded density is also shifted to promote the generation of random variates from the tail region of the input density. This method shares some similarity with the PIS method. Conditional Importance Sampling (CIS) is applicable to systems with more than one input noise density. In CIS random variates from one density are conditioned on the values generated from the first random input noise density. The conditioning is performed in a manner guaranteed to promote communication errors.

Finally two versions of importance sampling simulation for combating the effects of system memory on the amount of improvement are developed. The first, Efficient Importance Sampling (EIS), describes a method of applying the bias to the output probability density by finding a statistically equivalent linear model of the system under study. The coefficients of a linear model are found by performing an analysis procedure prior to the simulation. The second method also requires a linear or at least a linearized system model. With this model the input noise samples are processed through a Householder transfer function based on the system model prior to simulation through the system. Therefore we have denoted this method as Householder Importance Sampling (HIS). Both of these methods, when they can be applied, result in an importance sampling simulation with the equivalent of a system memory of one.

We have also reviewed the results found in the literature concerning the affects of system memory on the improvement in sample size possible from importance sampling. We have seen that as the system memory goes up, the potential reduction in the number of simulation samples required over classical Monte Carlo for a given estimator variance goes down. We have also seen that as the amount of memory goes up, the amount of bias to be applied to the input density needs to decrease; all other parameters being constant. Knowledge of system memory issues is important in any application of importance sampling but it is particularly important in M-ary PPM since the number of samples required for representing a PPM word is large.

It is shown that the variance of the IS estimator increases as the number of independent noise sources increases. The effect of multiple noise sources on importance sampling simulation is particularly relevant to optical systems simulation since we are essentially guaranteed to have more than one input noise source. In optical systems, the probability of error will be influenced by the shot noise of the detector and the additive thermal noise of the electronic circuits after

the detector. The shot noise of the detector is a non-additive signal-dependent noise source. Very little has been published on the importance sampling simulation of signal-dependent noise sources. This includes a general lack of discussion on biasing the statistics of APD devices. We have seen that the amount of bias applied to each of the noise sources in a multiple noise source system should be proportional to the amount of influence the noise source has on bit errors. The determination of the level of influence on communication error for each input noise source is difficult. When in doubt we are advised to bias all sources equally. At the same time we find that if a noise source has little influence on the probability of error it is best to not bias this source as to do so will increase the estimator variance rather than reduce it.

In Section 4.4 we grouped the importance sampling references according to the system to which it was applied. Besides the general digital communications system, importance sampling has been applied to Viterbi decoder simulation, radar simulation, DS-SSMA systems, large networks and adaptive equalizers.

In the last section we reviewed the simulation of optical systems both with and without importance sampling. There has only been one reference to optical system simulation using importance sampling. This was a fiber optic system using NRZ modulation, an APD detector and threshold detection. The APD statistics were biased through manipulation of the look-up table used to generated the random variates. The use of a look-up table limits the input signal to the detector to a finite number of power levels. This would not be suitable in the simulation of systems which included channel effects.

The other optical simulation references utilized semi-analytical or tail extrapolation techniques in order to decrease the time required to obtain results through simulation. Tail extrapolation techniques are prone to error. The semi-analytical methods rely on a linear system between the detector and the decision device.

CHAPTER 5

M-PPM IMPORTANCE SAMPLING SIMULATION

5.0 Introduction

In this chapter the mathematical foundation of importance sampling as applied to the M-PPM maximum likelihood receiver is established. The derivation begins with a mathematical development of classical Monte Carlo as applied to the M-PPM receiver. This description introduces the concept of a multi-dimensional probability space in which the error region of the receiver can be said to exist. The concept of a conditional indicator function is also introduced. With classical Monte Carlo and its associated concepts in place, the modification to Monte Carlo known as importance sampling is derived for the M-PPM receiver in Section 5.1.2. This includes a derivation of the IS estimator, a proof that the estimator is unbiased and a derivation of the estimator variance. The optimum bias function for the M-PPM receiver is developed in Section 5.1.3. From this expression, various bias methods and requirements on the bias function for reducing the sample size are developed in the remainder of Section 5.1.

In Section 5.2, the various bias function alternatives are explored in detail. This includes a brief description of bias methods which are not desirable. It is shown that shifting the mean of the input densities is a viable method of biasing the input statistics of the M-PPM APD optical receiver. A conditional bias method is also proposed based on the unique character of the M-PPM receiver decision mechanism. There are several possible methods to implement each of these techniques. Two versions are selected and the details of implementing each are presented in Sections 5.4 and 5.5 respectively. The complete simulation model and associated issues are presented in Section 5.3.

5.1 Application of Importance Sampling to M-PPM Maximum Likelihood Detection

As indicated in Chapter 4, the technique of importance sampling to improve the time required for Monte Carlo simulation is well documented for the threshold detection receiver. The application of IS to the simulation of APD based optical receivers has received very little

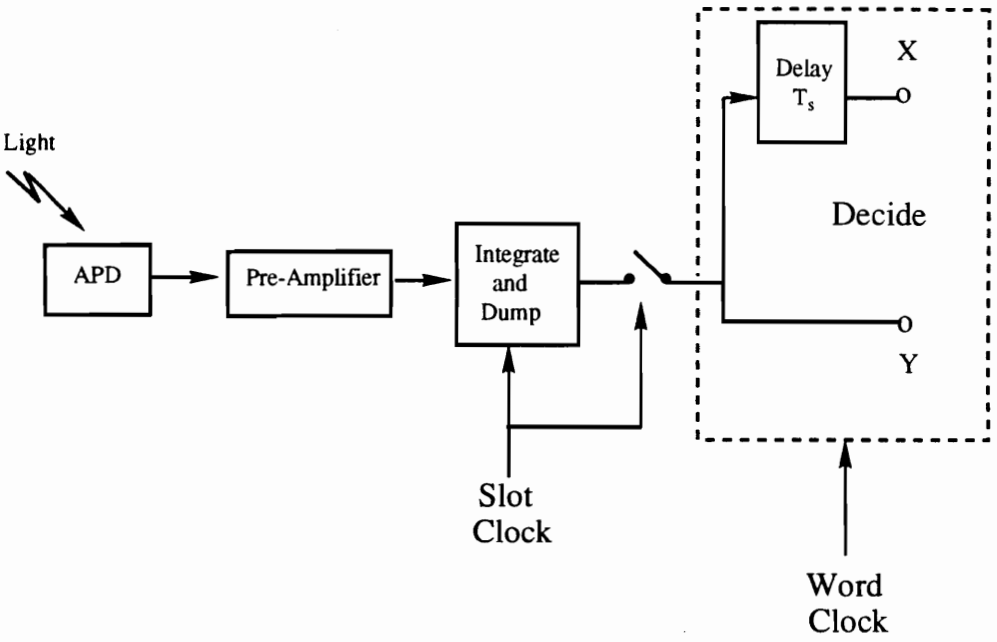
attention while the maximum likelihood optical PPM APD receiver has not been addressed at all. We begin Section 5.1 with a discussion of Monte Carlo simulation as applied to the M-PPM receiver. Section 5.1.1, establishes the mathematics of Monte Carlo simulation, wherein the estimator is developed and shown to be unbiased and an expression for the variance of the Monte Carlo estimator is also presented. This discussion introduces the concept of a multi-dimensional probability space and a conditional indicator function. In Section 5.1.2, importance sampling is derived for the PPM receiver. We develop the M-PPM IS estimator and show that it is unbiased and also present the variance for the M-PPM IS estimator. In Section 5.1.3 we present the optimum bias function for the M-PPM receiver. Finally, in Section 5.1.4 we present some useful expressions for the ratio of sample size improvement of M-PPM IS over M-PPM Classical Monte Carlo simulation.

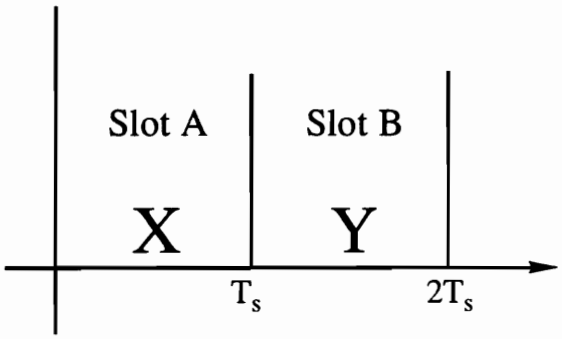
5.1.1 Monte Carlo Simulation of M-PPM

In this section we discuss the mathematics of classical Monte Carlo analysis for M-PPM signaling. In order to simplify some of the ensuing mathematics and to permit the use of figures and diagrams, we use Binary Pulse Position Modulation (BPPM) for illustration. Its extension to higher order PPM words is straight forward. The important results and concepts are expressed in terms of the general M-PPM case for completeness.

Binary pulse position modulation was introduced in Chapter 2 and perhaps is the simplest form of PPM modulation. Recall that in BPPM the PPM word time frame is divided into two time slots. For illustration purposes, we define a binary *one* to be encoded by a laser pulse in the first time slot with the second slot empty. A binary *zero* will be encoded with a pulse in the second time slot and the first slot empty. For clarity, we will refer to the first time slot as slot A and to the second time slot as slot B. A BPPM word was illustrated in Fig. 2.1(b). For computer simulation, a sampled version of this signal is generated.

Unlike a threshold detection receiver, where the hypothesis decision is based on a comparison of the measured output voltage against a fixed threshold, the maximum likelihood receiver compares the photon count from each time slot and selects the slot with the largest count as discussed in Chapter 2. This is a relative measure as the decision of slot A is made relative to the value in slot B and vice versa. Thus a comparison is made between two random variables, instead of the comparison of a random variable against a fixed threshold as is made in a traditional threshold detection receiver. We reflect this new understanding with a small modification to the PPM receiver model. In Fig. 5.1, we present a model of the BPPM receiver





BPPM Signal Components

Fig. 5.1 Two Random Variable Model of a BPPM Receiver

where we have expressly shown the output to be two random variables. The random variable X represents the measurement of slot A while Y is the random variable representing the measurement of slot B. These two random variables are made available for examination at the end of each PPM word frame. Slot A is selected as containing the transmitted laser pulse if X is greater than Y and slot B is selected as containing the transmitted laser pulse if Y is greater than X.

For analysis, we initially assume that the random variables representing each slot measurement are independent. In our BPPM example this means that X and Y are independent. This assumption is reasonable under normal receiver operating conditions. Some fraction of independence will be lost due to pulse spreading or slot timing errors, both of which result in some transmitted laser energy arriving in an adjacent empty slot. In this event the measured value for an empty slot can be argued to be dependent on the value of the slot containing the pulse. In proper receiver operation there is no dependence. This is because the operation of the receiver considers each slot measurement independently so that any additional light in an adjacent slot is perceived only as additional background light. We will find that the results presented are valid regardless and that slot independence is a convenience rather than a requirement.

Continuing our definition of BPPM modulation encoding, let us denote the probability of sending a pulse in slot A as P(A). P(A), for example, might be the probability of sending a *one*. We denote the probability of sending a pulse in slot B as P(B) and this might be the probability of sending a *zero*. The actual probability of sending a *one* or a *zero* is a function of the source statistics and does not play a part in the development that follows. Let the overall word error probability be denoted as P_{eW}, defined as

$$P_{eW} \equiv (P_{eW} \mid A) \cdot P(A) + (P_{eW} \mid B) \cdot P(B) \tag{5.1}$$

where $P_{eW} \mid A$ is the probability of word error given that the pulse was sent in the A slot and, similarly, $P_{eW} \mid B$ is the probability of word error given that the pulse was sent in the B slot. $P_{eW} \mid A$ and $P_{eW} \mid B$ are the symbol error probabilities for the slot A symbol and the slot B symbol respectively. Let us denote each of these symbol error probabilities as P_{eA} and P_{eB} so that $P_{eA} \equiv P_{eW} \mid A$ and $P_{eB} \equiv P_{eW} \mid B$. The P_{eA} is simply the probability that Y is greater than X given that the pulse was sent in the A slot. Similarly, the P_{eB} is the probability that X is greater than Y given that the pulse was sent in the B slot. Hence, eq.(5.1) can be rewritten as

$$P_{eW} \equiv \text{prob}(Y>X \mid A) \cdot P(A) + \text{prob}(X>Y \mid B) \cdot P(B) \tag{5.2}$$

Let us concentrate on the error probability for a single symbol as the total word error probability is simply the weighted sum of the individual symbol error probabilities as shown in eq.(5.2). We arbitrarily consider the case that the laser pulse was sent in slot A. P_{eA} is a function of two probability densities $f(X \mid A)$ and $f(Y \mid A)$. Representations of these densities are shown in Fig. 5.2. [We drop the conditional notation, i.e. $(\ldots \mid A)$, in the remainder of this discussion as redundant.] These output PDF's describe the distribution of the two voltage random variables observed at the decision device. They are a result of processing the APD output current through the receiver system. As discussed in Chapter 2, the output current of the APD is summed with the Gaussian distributed thermal noise of the preamplifier. This noisy signal current is filtered through the remainder of the receiver system and finally integrated to produce a test sample for each PPM time slot. The computer simulation is a discrete time model of this processing. Under the hypothesis of symbol A, the laser pulse resides in slot A and therefore, in general, the mean of $f(X)$ will be larger than the mean of $f(Y)$ as shown in Fig. 5.2.

Bear in mind that all of the statements in this section have a dual case in which the laser pulse was sent in slot B. This includes the existence of two other probability density functions $f(X \mid B)$ and $f(Y \mid B)$ in which the mean of $f(Y)$, i.e. $f(Y \mid B)$, is in general greater than the mean of $f(X)$, i.e. $f(X \mid B)$. In fact, all of these arguments apply directly to the M-PPM case since the only change is that we then have M slots, {A, B, C, ...} and would therefore compare M random variables at the decision point of the receiver. In this discussion we are simply considering the probability of error for an arbitrarily selected symbol. The results will be the same whether we have two symbols, as in BPPM, or M symbols.

Considering the case of the laser pulse sent in slot A, i.e. symbol A, we can write the P_{eA} as

$$P_{eA} \equiv \text{prob}(Y>X) = \int_{-\infty}^{\infty} f(X) \int_{X}^{\infty} f(Y)\, dy\, dx \tag{5.3}$$

The double integral is a mathematical expression for prob(Y>X) utilizing the definition of PDF's. If we define a function $H(y \mid x)$ in the same spirit as traditionally defined for Monte Carlo and IS simulation mathematics (see for example eq.(3.10), §3.1.2) then we write:

$$H(y \mid x) \equiv \begin{cases} 1 & y > x \\ 0 & y \leq x \end{cases} \tag{5.4}$$

Eq.(5.3) can now be written as

$$P_{eA} \equiv \text{prob}(Y>X) = \int_{-\infty}^{\infty} f(X) \int_{-\infty}^{\infty} H(y \mid x) f(Y)\, dy\, dx \tag{5.5}$$

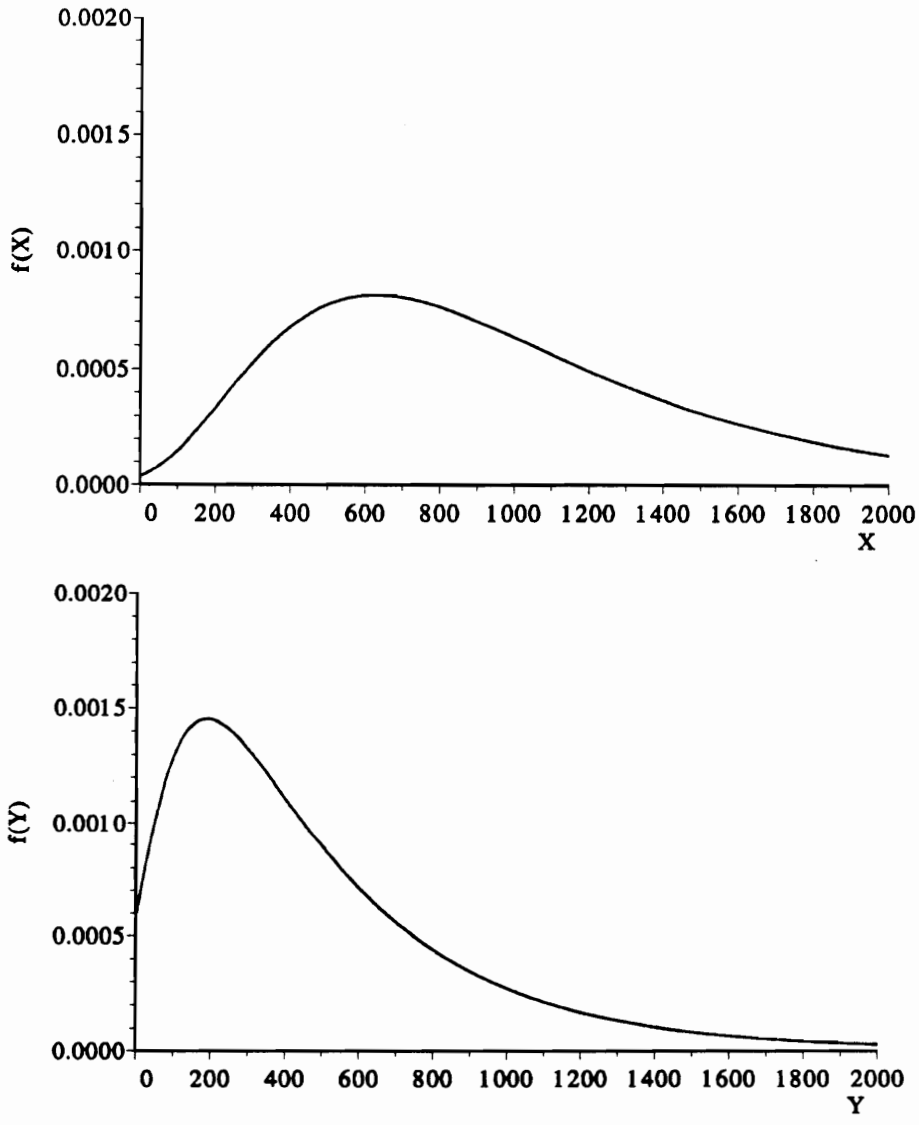


Fig. 5.2 Slot Measurement Output Probability Density Functions: f(X) and f(Y)

(These distributions have been simulated using the WMC PDF with parameters: G = 100, $\kappa = 0.02$, $\overline{n} = 5$ and $\overline{n} = 10$.)

$$= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} H(y \mid x) f(X) f(Y) \, dy \, dx \tag{5.6}$$

$$= E[H(y \mid x)] \tag{5.7}$$

$$\approx \frac{1}{N} \sum_{i=1}^{N} H(y_i \mid x_i) \tag{5.8}$$

where it can be seen that we have defined P_{eA} as the two dimensional expectation of our *conditional* indicator function $H(y \mid x)$.

The last step, eq.(5.8), is an estimation of this expected value as a summation. Let us denote the estimator for P_{eA} as $\hat{P}_{eA}$. Hence let

$$\hat{P}_{eA} \equiv \frac{1}{N} \sum_{i=1}^{N} H(y_i \mid x_i) \tag{5.9}$$

Note that the Monte Carlo simulation process is an estimation of the expected value of an indicator function just as it is in the case concerning the simulation of binary systems. Under the operation of the M-PPM maximum likelihood receiver, the indicator function is now a conditional operator. This marks one of the fundamental differences between Monte Carlo simulation for the threshold receiver and Monte Carlo simulation of M-PPM receivers.

Let us take a moment to reflect on the results to this point and attempt to provide illustrations for some of the concepts that have been expressed mathematically. If $f(X)$ and $f(Y)$ are independent then the product $f(X)f(Y)$ in eq.(5.6) is the joint probability density of the pair of random variables X and Y, denoted generally as $f(X, Y)$. If we consider these two random variables as coordinates on a plane, they would be plotted as a two dimensional probability density function $f(X, Y)$ on the coordinate system shown in Fig. 5.3. Fig. 5.3 displays the the joint probability space showing the two orthogonal axes over which the random variables X and Y are distributed. Note that X and Y have non-zero probability over the first quadrant of the 2-D probability space only. Fig. 5.4 is an orthographic view of the same probability space showing that the joint probability density forms a surface in probability space. The extension to higher order PPM formats is straightforward but cannot be illustrated as in Fig. 5.3 or Fig. 5.4. For example, QPPM requires a five dimensional space to depict the probability *surface* formed from the PDF's of the four slots in the PPM word. As can be seen in Fig. 5.4, BPPM forms a three dimensional image which can be illustrated and from which principles can be presented. Hence, we have concentrated on BPPM to develop and present our ideas and concepts.

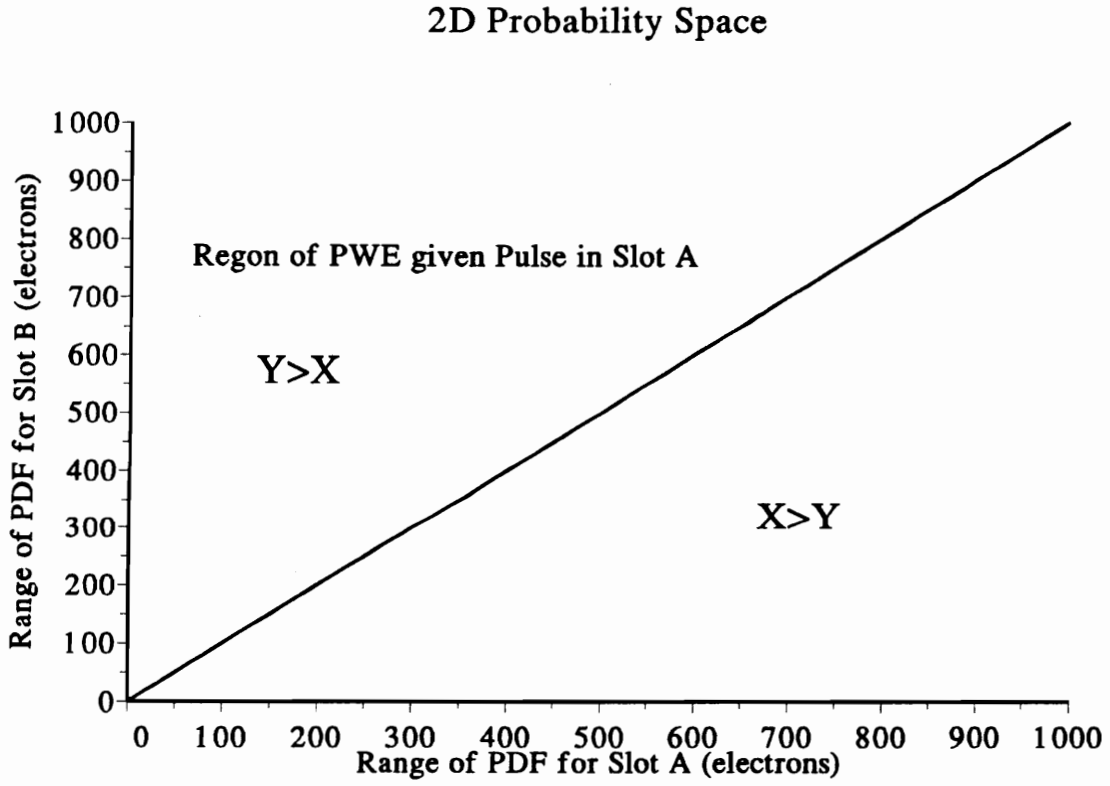


Fig. 5.3 Coordinate System for the Two-Dimensional Probability Space

The RV X is plotted along the horizontal axis and the vertical axis is for RV Y.

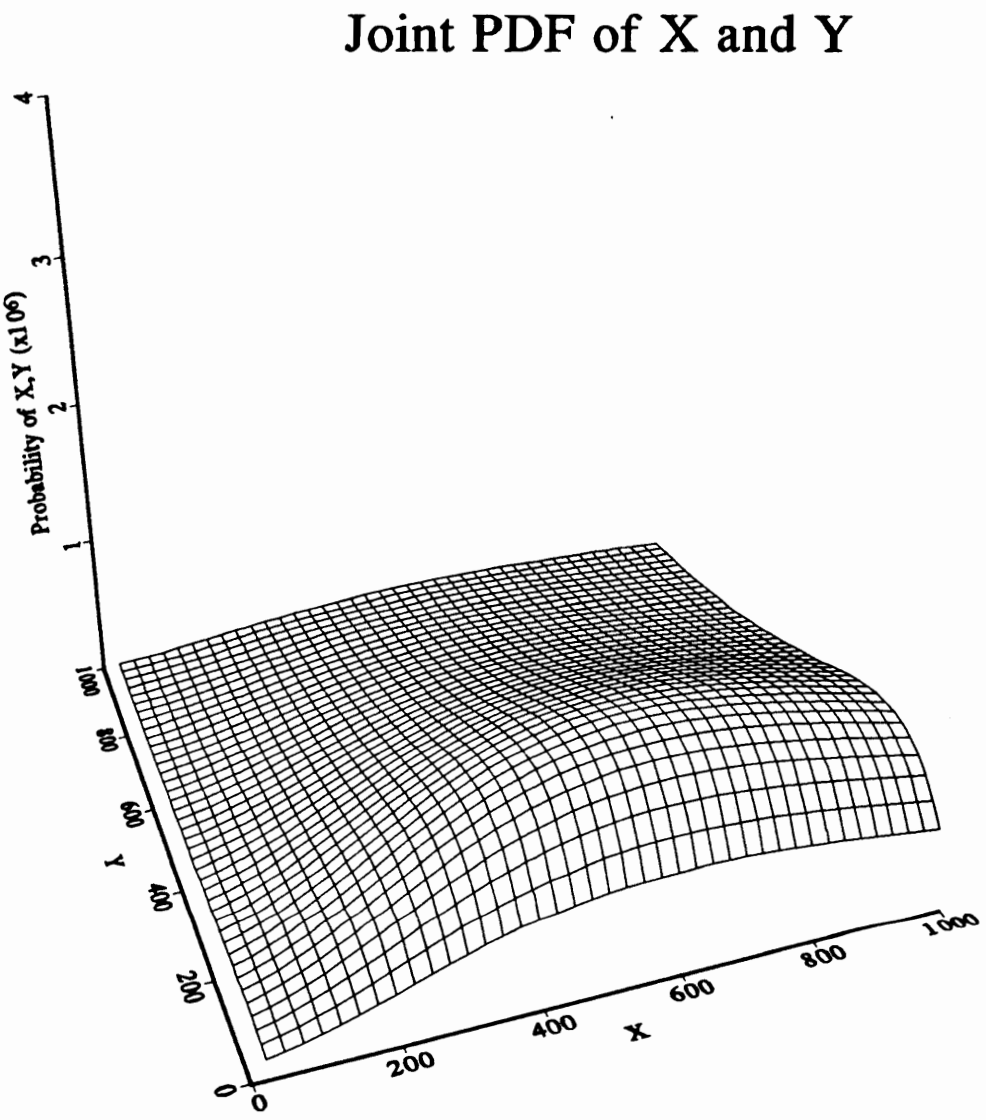


Fig. 5.4 Orthographic View of the Two-Dimensional Joint Probability Surface

X is the output random variable for the slot A, Y is the output random variable for the slot B.

(This joint output density was simulated using the same PDF's shown in Fig. 5.2.)

The line $X = Y$ shown in Fig. 5.3 marks the error boundary for a BPPM word error. Given that the laser pulse was transmitted in slot A then an incorrect choice will be made if the X and Y values produced at the end of the BPPM word measurement fall into the space above the line $X = Y$. This is the area in which $Y > X$ so that the receiver will incorrectly decide that the pulse was in slot B. In terms of importance sampling, this is the important region, the region where errors occur. By application of the definition of joint probability we note that the P_{eA} is equal to the area under the surface of the joint PDF above the $X = Y$ line. This is actually a restatement of eq.(5.3) or, equivalently, eq.(5.6) in words.

The conditional indicator function $H(y \mid x)$ from eq.(5.4) is shown in Fig. 5.5. It is a surface with the value of one over the region above the $X = Y$ line (the area in which $Y > X$) and zero elsewhere. This view of $H(y \mid x)$ can be applied to a physical interpretation of eq.(5.6) where it is seen that the function $H(y \mid x)$ is multiplied with the joint PDF and the limits of integration are extended to the entire range of the joint PDF. In other words, this multiplication selects the portion of the joint PDF (with a weight equal to one) that represents the P_{eA}. The extension of the limits to $\pm\infty$ permits the surface integral to be interpreted as the expected value of the function $H(y \mid x)$ as in eq.(5.7). All of these statements are directly analogous to the development of Monte Carlo simulation as presented in Chapter 3 or in the literature.

The estimation of the expected value of $H(y \mid x)$ is represented in eq.(5.8) or eq.(5.9). Monte Carlo simulation is the process of generating the random variates (x_i, y_i) and charting them into the probability space of Fig. 5.3, Fig. 5.4 or Fig. 5.5. The function $H(y \mid x)$ acts to accumulate the points which fall into the $Y>X$ portion of the plane, thus this becomes the important region. Each point that falls into the area where $Y>X$ increments the error count summation by one, i.e. each point has a weight of one, while (x_i, y_i) points outside this area increment the summation by zero. Since the (x_i, y_i) variates are generated according to the joint PDF, this summation and its subsequent division by the total number of simulated bits as per eq.(5.9), is an estimate of the area under the joint PDF surface in the region above the $X = Y$ line. To reiterate, this area represents P_{eA}.

A completely analogous set of statements and expressions exists for the case that the laser pulse was transmitted in slot B. Here we deal with two different PDF's which might be called $f(X \mid B)$ and $f(Y \mid B)$ if we follow the notation used above. We can write

$$P_{eB} \equiv P_{eW} \mid B = \text{prob}(X>Y) = \int_{-\infty}^{\infty} f(Y) \int_{y}^{\infty} f(X)\, dx\, dy \tag{5.10}$$

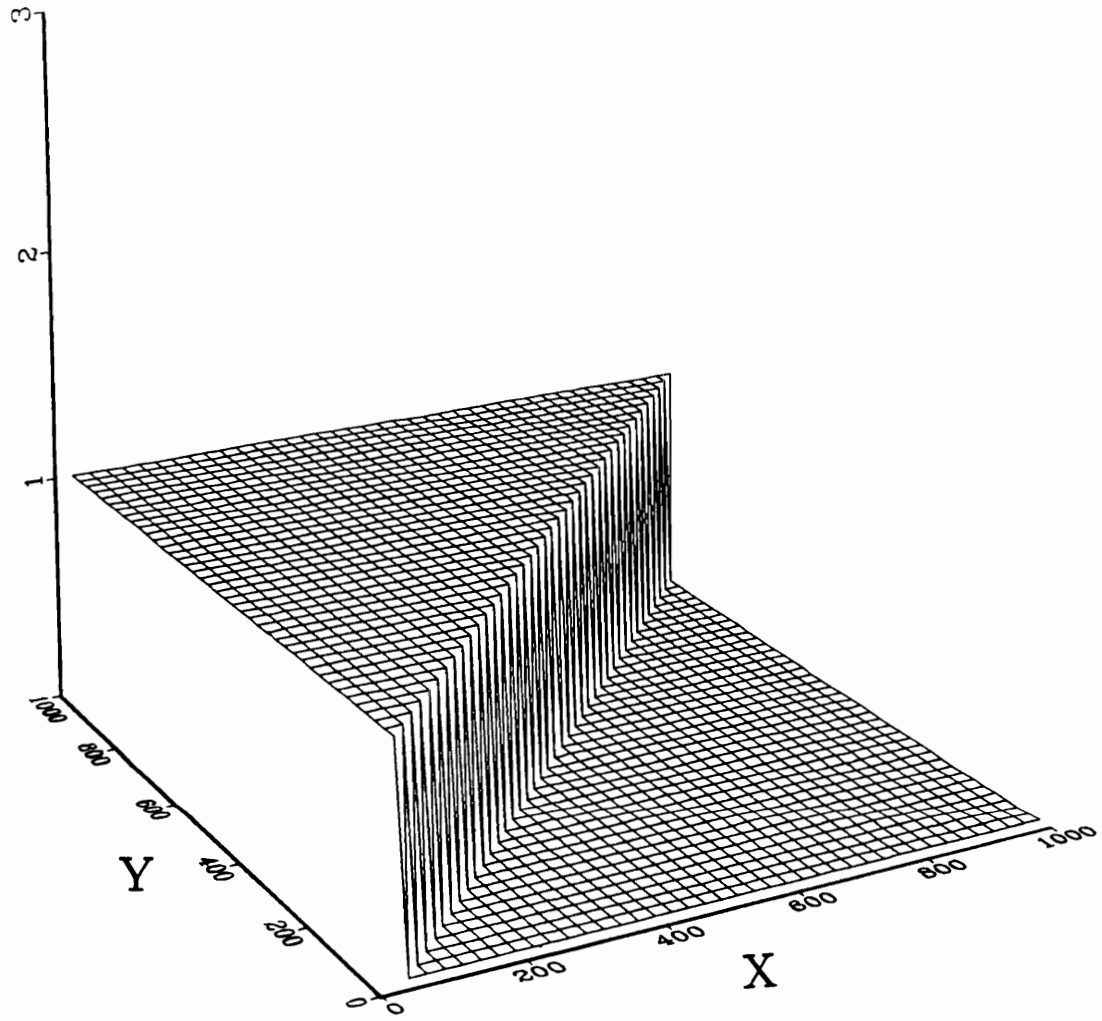


Fig. 5.5 Orthographic View of the Two-Dimensional Conditional Indicator Function H(y | x).

X is the output random variable for the slot A, Y is the output random variable for the slot B.

where we have again dropped the conditional notation of the PDF's. We must also define an appropriate indicator function which will yield the area under the joint PDF surface below the X = Y line. In this case we define

$$H(x \mid y) \equiv \begin{cases} 1 & x > y \\ 0 & x \leq y \end{cases} \tag{5.11}$$

This permits us to write

$$P_{eB} = \int_{-\infty}^{\infty} f(Y) \int_{-\infty}^{\infty} H(x \mid y) f(X)\, dx\, dy \tag{5.12}$$

$$= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} H(x \mid y) f(X) f(Y)\, dy\, dx \tag{5.13}$$

$$= E\big[H(x \mid y)\big] \tag{5.14}$$

$$\approx \frac{1}{N} \sum_{i=1}^{N} H(x_i \mid y_i) \equiv \hat{P}_{eB} \tag{5.15}$$

where we have defined $P_{eW} \mid B$ as the two dimensional expectation of a function $H(x \mid y)$. The function $H(x \mid y)$ is similar to our earlier function $H(y \mid x)$. The function $H(x \mid y)$ is simply a mirror image of the function $H(y \mid x)$ since it is one where $H(y \mid x)$ is zero and zero where $H(y \mid x)$ is one.

We now wish to extend the expressions for the Monte Carlo estimator of the BPPM case developed above to the general M-PPM signal format. We will adopt again the notation introduced in Chapter 2 and assume that the pulse was transmitted in the q^{th} PPM slot where $1 \leq q \leq M$. Let r_q be the integrator output of the q^{th} slot and r_i represent the output of each of the other slots where $i = \{1, 2, \ldots, M\text{-}1, M\}$ and $i \neq q$. The probability density function for r_q we denote as $f_q(r)$ and a PDF for each slot variable r_i as $f_i(r)$, where $i = \{1, 2, \ldots, M\}$ and $i \neq q$. With this notation, the probability of error for the q^{th} symbol is

$$P_{eQ} = \int_{-\infty}^{\infty} f_q(r_q) \left[\int_{r_q}^{\infty} f_1(r_1)\, dr_1 \int_{r_q}^{\infty} f_2(r_2)\, dr_2 \ldots \int_{r_q}^{\infty} f_i(r_i)\, dr_i \ldots \int_{r_q}^{\infty} f_M(r_M)\, dr_M \right] dr_q \tag{5.16}$$

where $i \neq q$ for any of the integrals inside the bracket. In the M-PPM case we define a conditional indicator function as

$$H(r_1, r_2, \ldots, r_i, \ldots, r_M : i \neq q \mid r_q) \equiv \begin{cases} 1 & \text{for any } r_i > r_q\,,\ i \neq q \\ 0 & \text{elsewhere} \end{cases} \tag{5.17}$$

To simplify notation, let us define the following vector notation for the M-ary conditional indicator function in eq.(5.17) as

$$H(\mathbf{r}_{i \neq q} \mid r_q) \equiv H(r_1, r_2, \ldots, r_i, \ldots, r_M : i \neq q \mid r_q) \tag{5.18}$$

Eq.(5.16) is now rewritten as

$$P_{eQ} \equiv \text{prob}(r_i > r_q) = \int_{-\infty}^{\infty} f(r_q) \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} H(\mathbf{r}_{i \neq q} \mid r_q) \Big[f(r_1) \ldots f(r_{i \neq q}) \ldots f(r_M)\, dr_i \Big] dr_q \tag{5.19}$$

$$= \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} H(\mathbf{r}_{i \neq q} \mid r_q) f(r_q) \Big[f(r_1) \ldots f(r_{i \neq q}) \ldots f(r_M)\, dr_i \Big] dr_q \tag{5.20}$$

$$= \int_{-\infty}^{\infty} H(\mathbf{r}_{i \neq q} \mid r_q) \left[\prod_{i=1}^{M} f(r_i)\, dr_i \right] \tag{5.21}$$

where, for simplicity, we have notated the multiple integrals of eq.(5.20) in the form of multi-dimensional integral as shown in eq.(5.21). This is again the expected value of the indicator function and can be written as

$$P_{eQ} = E\Big[H(\mathbf{r}_{i \neq q} \mid r_q) \Big] \tag{5.22}$$

This expectation can be approximated by

$$P_{eQ} \approx \frac{1}{N} \sum_{j=1}^{N} H(\mathbf{r}_{ij \neq qj} \mid r_{qj}) \equiv \hat{P}_{eQ} \tag{5.23}$$

We see that the Monte Carlo estimator for the general M-PPM symbol Q is of the same form as before in eq.(5.9) for the BPPM case. We have only extended the definition of our conditional indicator function in order to incorporate the additional slots in M-PPM.

Concerning the assumption of independence between each of the slot counts, this can easily be relaxed as it was useful only in the explanation and development of eq.(5.21). One could have also started with the general case and defined the joint probability density function for the set of slot random variables r_i as $f(r_1, r_2, \ldots, r_i, \ldots r_M)$. Beginning with a joint probability density one can immediately write the general version of eq.(5.20) as

$$P_{eQ} \equiv \text{prob}(r_i > r_q) = \int_{-\infty}^{\infty} H(\mathbf{r}_{i \neq q} \mid r_q) f(r_1, r_2, \ldots, r_i, \ldots r_M)\, d\mathbf{r} \tag{5.24}$$

where $d\mathbf{r} = dr_1\, dr_2 \ldots dr_i \ldots dr_M$

Again for notational simplicity, let us write the joint probability function of the slot random variables in terms of the **r** vector so that

$$f(r_1, r_2, \ldots, r_i, \ldots r_M) \equiv f(\mathbf{r}) \tag{5.25}$$

where $\mathbf{r} \equiv r_1, r_2, \ldots, r_i, \ldots r_M$. With this notation eq.(5.24) becomes

$$P_{eQ} \equiv \text{prob}(r_i > r_q) = \int_{-\infty}^{\infty} H(\mathbf{r}_{i \neq q} \mid r_q) f(\mathbf{r})\, d\mathbf{r} \tag{5.26}$$

Eq.(5.26) is a general expression of the probability of error for symbol Q. It assumes no independence between slots or any particular value for M. Note that this generalization has no effect on the form of eq.(5.22) or the estimator in eq.(5.23) as the expected value of the function remains the same.

When the slots are independent the joint probability density function is the product of the individual probability densities so that

$$f(\mathbf{r}) = f(r_1, r_2, \ldots, r_i, \ldots r_M) = \prod_{i=1}^{M} f(r_i) \tag{5.27}$$

If eq.(5.27) is substituted into eq.(5.26) then we obtain eq.(5.21).

It is important to establish that the estimator we have developed in eq.(5.23) is unbiased, that is, that the expected value of the estimator is the quantity sought. We will again default to the BPPM case for ease of notation and explanation. In this case, we would like to establish that the expected value of the summation in eq.(5.9) is P_{eA}.

$$E\left[\hat{P}_{eA}\right] = E\left[\frac{1}{N}\sum_{i=1}^{N} H(y_i \mid x_i)\right] \tag{5.28}$$

$$= \frac{1}{N} E\left[\sum_{i=1}^{N} H(y_i \mid x_i)\right] \tag{5.29}$$

$$= \frac{1}{N}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\sum_{i=1}^{N} H(y_i \mid x_i) f(X) f(Y)\, dy\, dx \tag{5.30}$$

$$= \frac{1}{N}\sum_{i=1}^{N}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} H(y_i \mid x_i) f(X) f(Y)\, dy\, dx \tag{5.31}$$

$$= \frac{1}{N}\sum_{i=1}^{N} P_{eA} \tag{5.32}$$

$$= P_{eA} \tag{5.33}$$

Therefore we have established that the estimator of eq.(5.9) is an unbiased estimator. A similar development can be made for the $\hat{P}_{eB}$ or, indeed, any of the symbols in the general M-ary PPM case so that we have that the estimator of eq.(5.23) is an unbiased estimator.

We would like to establish the variance of the estimator of eq.(5.23) for future reference. This will permit comparison with the importance sampling estimator variance and analysis of the improvement ratio. We again default to the BPPM case where the variance of the estimator of P_{eA} in eq.(5.9), $\sigma^2(\hat{P}_{eA})$, is found from

$$\sigma^2(\hat{P}_{eA}) = E\left[(\hat{P}_{eA})^2\right] - E^2\left[\hat{P}_{eA}\right] \tag{5.34}$$

which, using eq.(5.33), immediately becomes

$$\sigma^2(\hat{P}_{eA}) = E\left[(\hat{P}_{eA})^2\right] - \left[P_{eA}\right]^2 \tag{5.35}$$

To simplify the first term on the Right Hand Side (RHS), we begin as

$$E\left[(\hat{P}_{eA})^2\right] = E\left[\left(\frac{1}{N}\sum_{i=1}^{N} H(y_i \mid x_i)\right)^2\right] \tag{5.36}$$

$$= \frac{1}{N^2} E\left[\sum_{i=1}^{N} H^2(y_i \mid x_i) + \sum_{\substack{i,j=1 \\ i \neq j}}^{N} H(y_i \mid x_i) H(y_j \mid x_j)\right] \tag{5.37}$$

$$= \frac{1}{N^2}\left[\sum_{i=1}^{N} E\left\{H^2(y_i \mid x_i)\right\} + \sum_{\substack{i,j=1 \\ i \neq j}}^{N} E\left\{H(y_i \mid x_i) H(y_j \mid x_j)\right\}\right] \tag{5.38}$$

$$= \frac{1}{N^2}\left[N \cdot E\left\{H^2(y_i \mid x_i)\right\} + \sum_{\substack{i,j=1 \\ i \neq j}}^{N}\left[E\left\{H(y_i \mid x_i)\right\} E\left\{H(y_j \mid x_j)\right\} + COV\left\{H(y_i \mid x_i) H(y_j \mid x_j)\right\}\right]\right] \tag{5.39}$$

$$= \frac{1}{N^2}\left[N \cdot E\left\{H^2(y_i \mid x_i)\right\} + N(N-1) E^2\left\{H(y_i \mid x_i)\right\} + \sum_{\substack{i,j=1 \\ i \neq j}}^{N} COV\left\{H(y_i \mid x_i) H(y_j \mid x_j)\right\}\right] \tag{5.40}$$

$$= \frac{1}{N} P_{eA} + \left(1 - \frac{1}{N}\right) P_{eA}^2 + \frac{1}{N^2} \sum_{\substack{i,j=1 \\ i \neq j}}^{N} COV\left\{H(y_i \mid x_i) H(y_j \mid x_j)\right\} \tag{5.41}$$

where we have used the property that $E[XY] = E[X]E[Y] + COV[XY]$ and $COV[\cdot]$ refers to the covariance.

If $H(y_i \mid x_i)$ and $H(y_j \mid x_j)$ are independent for $i \neq j$ then the covariance term is zero.

The $H(y_i \mid x_i)$ and $H(y_j \mid x_j)$ are independent for each $i \neq j$ if each (x_i, y_i) pair is independent. In the simulation process, care is taken to insure that the generated samples are independent. In this case the variance of the estimator becomes

$$\sigma^2(\hat{P}_{eA}) = E\left[(\hat{P}_{eA})^2\right] - \left[P_{eA}\right]^2 \tag{5.42}$$

$$= \frac{1}{N}P_{eA} + (1 - \frac{1}{N})P_{eA}^2 - P_{eA}^2 \tag{5.43}$$

$$= \frac{P_{eA}(1 - P_{eA})}{N} \tag{5.44}$$

which is the usual form for the variance of the Monte Carlo estimator. Compare, for example, eq.(5.44) to eq.(3.13). The variance of each symbol in the general M-PPM case would have the same form as eq.(5.44).

At this point we have developed the basic expressions describing the Monte Carlo simulation of the maximum likelihood optical M-PPM receiver. We have written the expression for the estimator in eq.(5.23). We have shown that the estimator is unbiased in eq.(5.28) through eq.(5.33). Finally, we have generated an expression for the variance of the estimator in eq.(5.44) where it is seen to resemble the variance expression for threshold detection Monte Carlo estimator. In the next section we develop a similar set of equations describing the application of importance sampling to M-PPM.

5.1.2 Multi-Dimensional Importance Sampling

The application of importance sampling to the type of two or more dimensional probability space introduced in Section 5.1.1 follows a procedure similar to that described in Chapter 3 for the one dimensional baseband communications system. Here, importance sampling simulation is developed for the general M-PPM signal format assuming nothing concerning independence between slots. We return to the notation of Section 5.1.1 where the pulse is assumed to have been transmitted in the q^{th} PPM slot where $1 \leq q \leq M$. The development of importance sampling for the general M-PPM format begins with the general expression for the probability of error for the general symbol Q in eq.(5.26). We proceed by multiplying and dividing by an as yet undefined PDF $f^*(\mathbf{r})$ so that

(Note that the superscript asterisk [*] denotes a modified function, not complex conjugation.)

$$P_{eQ} \equiv \text{prob}(r_i > r_q) = \int_{-\infty}^{\infty} H(\mathbf{r}_{i \neq q} \mid r_q) f(\mathbf{r})\, d\mathbf{r} \tag{5.45}$$

$$= \int_{-\infty}^{\infty} H(\mathbf{r}_{i \neq q} \mid r_q) \frac{f(\mathbf{r})}{f^*(\mathbf{r})} f^*(\mathbf{r})\, d\mathbf{r} \tag{5.46}$$

$$= \int_{-\infty}^{\infty} H^*(\mathbf{r}_{i \neq q} \mid r_q) f^*(\mathbf{r})\, d\mathbf{r} \tag{5.47}$$

$$= E_*\left[H^*(\mathbf{r}_{i \neq q} \mid r_q)\right] \tag{5.48}$$

where $$H^*(\mathbf{r}_{i \neq q} \mid r_q) \equiv H(\mathbf{r}_{i \neq q} \mid r_q) \frac{f(\mathbf{r})}{f^*(\mathbf{r})} = H(\mathbf{r}_{i \neq q} \mid r_q) W(\mathbf{r}) \tag{5.49}$$

or equivalently

$$H^*(\mathbf{r}_{i \neq q} \mid r_q) \equiv \begin{cases} W(\mathbf{r}) & \text{for any } r_i > r_q\,,\ i \neq q \\ 0 & \text{elsewhere} \end{cases} \tag{5.50}$$

We have defined $$W(\mathbf{r}) \equiv \frac{f(\mathbf{r})}{f^*(\mathbf{r})} \tag{5.51}$$

as the weight of the conditional indicator function. Note that it is a general function of the vector $\mathbf{r}$, the set of slot random variables.

As in previous developments of importance sampling, our estimator is the expected value of a modified indicator function, $H^*(\mathbf{r}_{i \neq q} \mid r_q)$ of eq.(5.49) or eq.(5.50). The expected value is taken with respect to the biased PDF's used in the simulation so that it is marked with a subscript '$*$'. From eq.(5.48) we now write the importance sampling estimator for the general M-PPM slot Q as

$$\hat{P}^*_{eQ} \equiv \frac{1}{N^*} \sum_{j=1}^{N^*} H^*(\mathbf{r}_{ij \neq qj} \mid r_{qj}) \tag{5.52}$$

where N^* is the number of M-PPM slot Q words that are examined in our importance sampling estimate of P_{eQ}.

Nest, it is shown that the general M-PPM estimator is an unbiased estimator and then a general expression for the variance of the M-PPM importance sampling estimator is developed.

To be an unbiased estimator, the expected value of the right hand side of eq.(5.52) should be P_{eQ}. Note that this is the expected value taken relative to the modified PDF's. The expected value of the right hand side of eq.(5.52) is then

$$E_*\left[\hat{P}^*_{eQ}\right] = E_*\left[\frac{1}{N^*}\sum_{j=1}^{N^*} H^*(\mathbf{r}_{ij\neq qj} \mid \mathbf{r}_{qj})\right] \tag{5.53}$$

$$= \frac{1}{N^*} E_*\left[\sum_{j=1}^{N^*} H^*(\mathbf{r}_{ij\neq qj} \mid \mathbf{r}_{qj})\right] \tag{5.54}$$

$$= \frac{1}{N^*} \sum_{j=1}^{N^*} E_*\left[H^*(\mathbf{r}_{ij\neq qj} \mid \mathbf{r}_{qj})\right] \tag{5.55}$$

$$= \frac{1}{N^*} \sum_{j=1}^{N^*} \int_{-\infty}^{\infty} H^*(\mathbf{r}_{i\neq q} \mid \mathbf{r}_q) f^*(\mathbf{r})\, d\mathbf{r} \tag{5.56}$$

$$= \frac{1}{N^*} \sum_{j=1}^{N^*} \int_{-\infty}^{\infty} H(\mathbf{r}_{i\neq q} \mid \mathbf{r}_q) f(\mathbf{r})\, d\mathbf{r} \tag{5.57}$$

$$= \frac{1}{N^*} \sum_{j=1}^{N^*} P_{eQ} = P_{eQ} \tag{5.58}$$

Thus the importance sampling estimator is unbiased.

Just as for the classical Monte Carlo M-PPM error estimator, we would now like to establish the variance of the estimator shown in eq.(5.52). We proceed as before and write

$$\sigma^2(\hat{P}^*_{eQ}) = E_*\left[(\hat{P}^*_{eQ})^2\right] - E^2_*\left[\hat{P}^*_{eQ}\right] \tag{5.59}$$

which immediately becomes

$$\sigma^2(\hat{P}^*_{eQ}) = E_*\left[(\hat{P}^*_{eQ})^2\right] - P^2_{eQ} \tag{5.60}$$

after application of eq.(5.58).

To simplify the first term on the RHS, we write

$$E_*\left[(\hat{P}^*_{eQ})^2\right] = E_*\left[\left(\frac{1}{N^*}\sum_{j=1}^{N^*} H^*(\mathbf{r}_{ij\neq qj} \mid \mathbf{r}_{qj})\right)^2\right] \tag{5.61}$$

$$= \frac{1}{N^{*2}} E_* \left[\sum_{j=1}^{N^*} H^{*2}(\mathbf{r}_{ij \neq qj} \mid \mathbf{r}_{qj}) + \sum_{\substack{j,k=1 \\ j \neq k}}^{N^*} H^*(\mathbf{r}_{ij \neq qj} \mid \mathbf{r}_{qj}) H^*(\mathbf{r}_{ik \neq qk} \mid \mathbf{r}_{qk}) \right] \tag{5.62}$$

$$= \frac{1}{N^{*2}} \left[\sum_{j=1}^{N^*} E_* \left\{ H^{*2}(\mathbf{r}_{ij \neq qj} \mid \mathbf{r}_{qj}) \right\} + \sum_{\substack{j,k=1 \\ j \neq k}}^{N^*} E_* \left\{ H^*(\mathbf{r}_{ij \neq qj} \mid \mathbf{r}_{qj}) H^*(\mathbf{r}_{ik \neq qk} \mid \mathbf{r}_{qk}) \right\} \right] \tag{5.63}$$

$$= \frac{1}{N^{*2}} \left[\sum_{j=1}^{N^*} E_* \left\{ H^2(\mathbf{r}_{ij \neq qj} \mid \mathbf{r}_{qj}) W^2(\mathbf{r}_j) \right\} + \sum_{\substack{j,k=1 \\ j \neq k}}^{N^*} E_* \left\{ H(\mathbf{r}_{ij \neq qj} \mid \mathbf{r}_{qj}) W(\mathbf{r}_j) H(\mathbf{r}_{ik \neq qk} \mid \mathbf{r}_{qk}) W(\mathbf{r}_k) \right\} \right] \tag{5.64}$$

$$= \frac{1}{N^{*2}} \left[N^* \cdot E_* \left\{ H^2(\mathbf{r}_{i \neq q} \mid \mathbf{r}_q) W^2(\mathbf{r}) \right\} \right] +$$

$$+ \frac{1}{N^{*2}} \left[\sum_{\substack{j,k=1 \\ j \neq k}}^{N^*} E_* \left\{ H(\mathbf{r}_{ij \neq qj} \mid \mathbf{r}_{qj}) W(\mathbf{r}_j) \right\} E_* \left\{ H(\mathbf{r}_{ik \neq qk} \mid \mathbf{r}_{qk}) W(\mathbf{r}_k) \right\} \right]$$

$$+ \frac{1}{N^{*2}} \left[\sum_{\substack{j,k=1 \\ j \neq k}}^{N^*} COV_* \left\{ H(\mathbf{r}_{ij \neq qj} \mid \mathbf{r}_{qj}) W(\mathbf{r}_j) H(\mathbf{r}_{ik \neq qk} \mid \mathbf{r}_{qk}) W(\mathbf{r}_k) \right\} \right] \tag{5.65}$$

again, we have used the property that $E[XY] = E[X]E[Y] + COV[XY]$ where $COV[\cdot]$ refers to the covariance. Continuing ...

$$= \frac{1}{N^{*2}} \left[N^* \cdot \int_{-\infty}^{\infty} H^2(\mathbf{r}_{i \neq q} \mid \mathbf{r}_q) W^2(\mathbf{r}) f^*(\mathbf{r}) \, d\mathbf{r} \right] +$$

$$\frac{1}{N^{*2}} \left[N^*(N^* - 1) E_*^2 \left\{ H(\mathbf{r}_{i \neq q} \mid \mathbf{r}_q) W(\mathbf{r}) \right\} + \sum_{\substack{i,j=1 \\ i \neq j}}^{N^*} COV_* \left\{ H^*(\mathbf{r}_{ij \neq qj} \mid \mathbf{r}_{qj}) H^*(\mathbf{r}_{ik \neq qk} \mid \mathbf{r}_{qk}) \right\} \right] \tag{5.66}$$

$$= \frac{1}{N^{*2}} \left[N^* \cdot \int_{-\infty}^{\infty} H(\mathbf{r}_{i \neq q} \mid \mathbf{r}_q) W(\mathbf{r}) f(\mathbf{r}) \, d\mathbf{r} \right] +$$

$$\frac{1}{N^{*2}} \left[N^*(N^* - 1) P_{eQ}^2 + \sum_{\substack{i,j=1 \\ i \neq j}}^{N^*} COV_* \left\{ H^*(\mathbf{r}_{ij \neq qj} \mid \mathbf{r}_{qj}) H^*(\mathbf{r}_{ik \neq qk} \mid \mathbf{r}_{qk}) \right\} \right] \tag{5.67}$$

$$= \frac{1}{N^*}\int_{-\infty}^{\infty} H(\mathbf{r}_{i \neq q} \mid \mathbf{r}_q) W(\mathbf{r}) f(\mathbf{r})\, d\mathbf{r} + P_{eQ}^2 - \frac{1}{N^*} P_{eQ}^2 + \frac{1}{N^{*2}} \sum_{\substack{i,j=1 \\ i \neq j}}^{N^*} COV_* \left\{ H^*(\mathbf{r}_{ij \neq qj} \mid \mathbf{r}_{qj}) H^*(\mathbf{r}_{ik \neq qk} \mid \mathbf{r}_{qk}) \right\} \quad (5.68)$$

Eq.(5.68) is an expression for the first term of the RHS of eq.(5.60). Substituting eq.(5.68) into eq.(5.60) and making use of the fact that

$$P_{eQ}^2 = \int_{-\infty}^{\infty} H(\mathbf{r}_{i \neq q} \mid \mathbf{r}_q) P_{eQ} f(\mathbf{r})\, d\mathbf{r} \quad (5.69)$$

we obtain

$$\sigma^2(\hat{P}_{eQ}^*) = E_* \left[(\hat{P}_{eQ}^*)^2 \right] - P_{eQ}^2 \quad (5.70)$$

$$= \frac{1}{N^*} \left[\int_{-\infty}^{\infty} H(\mathbf{r}_{i \neq q} \mid \mathbf{r}_q) W(\mathbf{r}) f(\mathbf{r})\, d\mathbf{r} - P_{eQ}^2 \right] + \frac{1}{N^{*2}} \sum_{\substack{i,j=1 \\ i \neq j}}^{N^*} COV_* \left\{ H^*(\mathbf{r}_{ij \neq qj} \mid \mathbf{r}_{qj}) H^*(\mathbf{r}_{ik \neq qk} \mid \mathbf{r}_{qk}) \right\} \quad (5.71)$$

$$= \frac{1}{N^*} \int_{-\infty}^{\infty} H(\mathbf{r}_{i \neq q} \mid \mathbf{r}_q) \left[W(\mathbf{r}) - P_{eQ} \right] f(\mathbf{r})\, d\mathbf{r} + \frac{1}{N^{*2}} \sum_{\substack{i,j=1 \\ i \neq j}}^{N^*} COV_* \left\{ H^*(\mathbf{r}_{ij \neq qj} \mid \mathbf{r}_{qj}) H^*(\mathbf{r}_{ik \neq qk} \mid \mathbf{r}_{qk}) \right\} \quad (5.72)$$

If $H^*(\mathbf{r}_{ij \neq qj} \mid \mathbf{r}_{qj})$ and $H^*(\mathbf{r}_{ik \neq qk} \mid \mathbf{r}_{qk})$ are independent for $j \neq k$ then the covariance term is zero. In this case the variance of the importance sampling estimator can be written as

$$\sigma^2(\hat{P}_{eQ}^*) = \frac{1}{N^*} \int_{-\infty}^{\infty} H(\mathbf{r}_{i \neq q} \mid \mathbf{r}_q) \left[W(\mathbf{r}) - P_{eQ} \right] f(\mathbf{r})\, d\mathbf{r} \quad (5.73)$$

The reader is encouraged to compare eq.(5.73) to eq.(4.29) in order to note the similarities and differences between this expression and the expression for the variance of the importance sampling estimate for the threshold detection receiver.

The basic expressions describing the application of importance sampling to the simulation of the maximum likelihood optical PPM receiver have been developed. We have written the expression for the estimator in eq.(5.52). We have shown that the estimator is unbiased in eq.(5.53) through eq.(5.58). Finally, we have generated an expression for the variance of the estimator in eq.(5.73). In developing these equations we have assumed nothing

concerning the independence of the measured slot random variables. Neither have we assumed or determined any properties required by the biased function $f^*(\mathbf{r})$ in order to insure that the importance sampling simulation will require fewer samples than a classical Monte Carlo simulation for an equivalent error in the estimate. The form of the biased function $f^*(\mathbf{r})$ determines the value(s) of the weight function $W(\mathbf{r})$. The desired form of the weight function will be the subject of the next section. How to generate it is the subject of the remainder of this chapter.

5.1.3 Optimum Bias Function for M-PPM

The optimum bias function for the threshold detection receiver was presented in Chapter 4 as that PDF which minimizes the variance of the importance sampling estimator. It was seen that an unrealizable bias function existed which actually made the variance equal to zero. This function was obtained through various means by various authors, the most straight forward of which was by inspection.

Through inspection of the variance of the importance sampling estimator for the PPM receiver, a similar optimum bias function can be obtained. Consider the expression for the importance sampling estimator variance for the general M-PPM slot Q in eq.(5.73). Modeled after the threshold detection receiver optimum bias function, we postulate

$$f^*_{opt}(\mathbf{r}) = \frac{H(\mathbf{r}_{i \neq q} \mid \mathbf{r}_q) f(\mathbf{r})}{P_{eQ}} \tag{5.74}$$

as the optimum bias function to minimize the variance of the estimator. Adopting eq.(5.74) as the bias function in importance sampling simulation, we obtain the following expression

$$W_{opt}(\mathbf{r}) = \frac{f(\mathbf{r})}{f^*_{opt}(\mathbf{r})} = \frac{P_{eQ}}{H(\mathbf{r}_{i \neq q} \mid \mathbf{r}_q)} \tag{5.75}$$

which describes the weights generated under the optimum PDF bias function. Importance sampling simulation of the M-PPM receiver using the optimum bias function in eq.(5.74) leads to a zero estimator variance much as it did in the threshold detection case discussed in Chapter 4. By inserting eq.(5.75) into eq.(5.73) we find that the variance is indeed zero.

$$\sigma^2_{opt}(\hat{P}^*_{eQ}) = \frac{1}{N^*} \int_{-\infty}^{\infty} H(\mathbf{r}_{i \neq q} \mid \mathbf{r}_q) \left[W_{opt}(\mathbf{r}) - P_{eQ} \right] f(\mathbf{r})\, d\mathbf{r} = 0 \tag{5.76}$$

We see that we are left with a situation completely analogous to the case of the threshold receiver. In eq.(5.74) we have an expression which causes the variance of the estimator to be zero but which is as unrealizable as its threshold detection counterpart since the construction of the optimum bias function requires *a priori* knowledge of the quantity sought, P_{eQ}, and *a priori* knowledge of the region of the input density over which errors will occur, the important region.

5.1.4 Sample Size Reduction

Through the use of importance sampling, we wish to reduce the number of simulation samples required in order to obtain a valid estimate of the probability of error. In order to ascertain the desired properties of the weight function W(r) let us examine and compare the normalized error of the M-PPM classical Monte Carlo estimate and the normalized error of the M-PPM importance sampling estimate. The normalized error of the estimates given by eq.(5.23) and eq.(5.52) are proportional to the variance of the estimators[1][2]. It follows that a comparison of the variance of each estimator will indicate the relative strength of each estimate. Let the variance for the counting estimator of eq.(5.23) be σ_C^2, where the subscript C designates the classical Monte Carlo estimation. This variance was produced in eq.(5.44) as

$$\sigma_C^2 = \frac{P_{eQ}(1-P_{eQ})}{N} \tag{5.77}$$

$$= \frac{1}{N}\int_{-\infty}^{\infty} H(\mathbf{r}_{i\neq q}\,|\,\mathbf{r}_q)\big[1-P_{eQ}\big]f(\mathbf{r})\,d\mathbf{r} \tag{5.78}$$

$$= \frac{1}{N}\int_{\Omega_q} \big[1-P_{eQ}\big]f(\mathbf{r})\,d\mathbf{r} \tag{5.79}$$

where the area of integration, Ω_q, in the last expression is the region such that $\mathbf{r}_i > \mathbf{r}_q$, $i \neq q$.

Similarly, let the variance for the counting estimator of eq.(5.52) be σ_{IS}^2, where the subscript *IS* designates Monte Carlo estimation with importance sampling. This variance is given in eq.(5.73) as

$$\sigma_{IS}^2 = \frac{1}{N^*}\int_{-\infty}^{\infty} H(\mathbf{r}_{i\neq q}\,|\,\mathbf{r}_q)\big[W(\mathbf{r})-P_{eQ}\big]f(\mathbf{r})\,d\mathbf{r} \tag{5.80}$$

$$= \frac{1}{N^*}\int_{\Omega_q} \big[W(\mathbf{r})-P_{eQ}\big]f(\mathbf{r})\,d\mathbf{r} \tag{5.81}$$

where again the area of integration in the last expression is the region such that $r_i > r_q$, $i \neq q$.

For a given normalized error, the variances for each estimator will be equal. Therefore we can derive a sample size reduction factor, R, by setting the variances equal and solving as follows

$$\sigma_C^2 = \sigma_{IS}^2 \tag{5.82}$$

$$\frac{1}{N}\int_{\Omega_q}\left[1 - P_{eQ}\right]f(\mathbf{r})\,d\mathbf{r} = \frac{1}{N^*}\int_{\Omega_q}\left[W(\mathbf{r}) - P_{eQ}\right]f(\mathbf{r})\,d\mathbf{r} \tag{5.83}$$

$$R \equiv \frac{N}{N^*} = \frac{\int_{\Omega_q}\left[1 - P_{eQ}\right]f(\mathbf{r})\,d\mathbf{r}}{\int_{\Omega_q}\left[W(\mathbf{r}) - P_{eQ}\right]f(\mathbf{r})\,d\mathbf{r}} \tag{5.84}$$

If we design the weighting function $W(\mathbf{r})$ to be less than 1 in the region Ω_q then eq.(5.84) shows that the ratio R of the number of samples used in the classical Monte Carlo simulation to the number of samples used in the modified Monte Carlo simulation will be greater than one. This indicates a reduction in the number of samples which must be simulated in order to achieve the same level of error in the estimate. The amount of this reduction will be directly proportional to the amount of bias applied to the simulation statistics. In the remainder of this chapter, methods are explored for biasing the APD parameters in order to cause more errors to occur with a weight function that is less than one.

For importance sampling in the M-PPM case, we have devised a conditional indicator function, eq.(5.49), which we repeat below

$$H^*(\mathbf{r}_{i \neq q} \mid \mathbf{r}_q) \equiv H(\mathbf{r}_{i \neq q} \mid \mathbf{r}_q)\frac{f(\mathbf{r})}{f^*(\mathbf{r})} \equiv H(\mathbf{r}_{i \neq q} \mid \mathbf{r}_q)\,W(\mathbf{r}) \tag{5.85}$$

or equivalently

$$H^*(\mathbf{r}_{i \neq q} \mid \mathbf{r}_q) \equiv \begin{cases} W(\mathbf{r}) & \text{for any } r_i > r_q\,,\ i \neq q \\ 0 & \text{elsewhere} \end{cases} \tag{5.86}$$

We defined the weight function $W(\mathbf{r})$ as

$$W(\mathbf{r}) \equiv \frac{f(\mathbf{r})}{f^*(\mathbf{r})} = \frac{f(r_1, r_2, \ldots, r_i, \ldots, r_M)}{f^*(r_1, r_2, \ldots, r_i, \ldots, r_M)} \tag{5.87}$$

We have seen the role that the weight function plays in importance sampling simulation through our examination of threshold detection receiver simulation. We know, therefore, that we control the efficiency of the importance sampling process via our control of the weight function. The weight function, in turn, is determined by our choice of biased sampling function $f^*(\mathbf{r})$. We have not, until now, discussed how to bias the input densities of the M-PPM simulation.

We consider the case that the random variables representing the measurement over each slot in the M-PPM word are independent. Assuming independence, the joint probability density of the set of slot variables becomes equal to the product of individual slot variable probability densities. That is

$$f(\mathbf{r}) \equiv f(r_1, r_2, \ldots, r_i, \ldots r_M) = \prod_{i=1}^{M} f(r_i) \tag{5.88}$$

Under these conditions the weight function becomes

$$W(\mathbf{r}) = \frac{\prod_{i=1}^{M} f(r_i)}{\prod_{i=1}^{M} f^*(r_i)} \tag{5.89}$$

$$= \frac{f(r_1)}{f^*(r_1)} \cdot \frac{f(r_2)}{f^*(r_2)} \cdot \ldots \cdot \frac{f(r_M)}{f^*(r_M)} \tag{5.90}$$

$$= W(r_1) \cdot W(r_2) \cdot \ldots \cdot W(r_M) \tag{5.91}$$

In other words, under the assumption of independent slots one can control the overall value of the weight function by controlling the weights of each individual slot. We might, for example, bias only one of the slots while the other slots remain unbiased with a weight of one. The mathematical development of M-PPM IS has not placed any restrictions on which slot is biased. One might bias the slot which contains the pulse: somehow lower its value relative to the empty slots in order to cause more errors. One might artificially raise the value of one or more of the empty slots. This will increase the probability that an empty slot value will be found to be greater than the value of the slot which contained the pulse, thus causing more errors to occur. Alternatively, one might raise the empty slot values and simultaneously lower the pulse slot value in order to increase the probability of errors occurring. Note that all of these options are akin to lowering the signal to noise ratio at the detector in order to cause more errors, reminiscent of the IS procedure in the threshold detection receiver. In SIS (§4.2.2), the variance of the input noise source was increased which is equivalent to increasing the noise power. In IIS (§4.2.3), the mean of the input noise source is shifted in such a way as to lower

the signal to noise ratio and to create more errors.

If we bias the input statistics to just one of the slots, either raising the average value of an empty slot or lowering the average value of the pulse slot, then we create more errors. We also cause the weight for that slot to be less than one. Examination of eq.(5.91) shows that even if only one slot has a weight less than one, the overall weight will be less than one. As pointed out in eq.(5.84), if the weight term is less than one then fewer simulation samples will be required for a given estimation error margin.

We have not addressed whether it is better to bias all of the slots or just one. Eq.(5.91) and eq.(5.84) give no insight nor indicate any preference. Let us examine specific methods of how to create more PPM word errors, causing $W(\mathbf{r}) \leq 1.0$ and reducing the required number of simulation samples, by examining the BPPM case in detail. As in Section 5.1.1, we again concentrate on the case that the laser pulse was sent in slot A. For continuity, let us quickly re-derive importance sampling for the case of BPPM. We begin with eq.(5.6) where we multiply and divide each of the PDF's by as yet undefined PDF's $f^*(X)$ and $f^*(Y)$ so that

$$P_{eA} \equiv \text{prob}(Y>X) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} H(y \mid x) f(X) f(Y)\, dy\, dx \tag{5.92}$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} H(y \mid x) \frac{f(X)}{f^*(X)} f^*(X) \frac{f(Y)}{f^*(Y)} f^*(Y)\, dy\, dx \tag{5.93}$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} H^*(y \mid x) f^*(X) f^*(Y)\, dy\, dx \tag{5.94}$$

$$= E_*\left[H^*(y \mid x)\right] \tag{5.95}$$

where $$H^*(y \mid x) \equiv H(y \mid x) \frac{f(X)}{f^*(X)} \frac{f(Y)}{f^*(Y)} = H(y \mid x) W(X) W(Y) \tag{5.96}$$

The expected value of $H^*(y \mid x)$ is estimated through our simulation process by

$$\hat{P}^*_{eA} = \frac{1}{N^*} \sum_{j=1}^{N^*} H^*(y_j \mid x_j) \tag{5.97}$$

For notational convenience, we have defined the weight over the PDF of X and the weight over the PDF of Y to be

$$W(X) \equiv \frac{f(X)}{f^*(X)} \tag{5.98}$$

$$\text{and} \quad W(Y) \equiv \frac{f(Y)}{f^*(Y)} \tag{5.99}$$

respectively. We can now write

$$H^*(y_j \mid x_j) \equiv \begin{cases} W(x_j)\,W(y_j) & y_j > x_j \\ 0 & y_j \leq x_j \end{cases} \tag{5.100}$$

We desire that the probability of producing the random variates (x_i, y_i) in the region above the $X = Y$ line be greater than it is in the unbiased case. Thus we want $f^*(X) \geq f(X)$ and/or $f^*(Y) \geq f(Y)$ in the region that $X > Y$. If we can achieve this, examination of the definition of $H^*(y \mid x)$ in eq.(5.96) shows that each term in the summation will either be zero or a value less than or equal to one since each non-zero term is weighted by the ratio $\frac{f(X)}{f^*(X)}$ and $\frac{f(Y)}{f^*(Y)}$.

Let us consider the above equations while examining the probability regions for X and Y and the probability densities $f(X)$ and $f(Y)$ depicted in Fig. 5.2, Fig. 5.3 and Fig. 5.4. We would like to increase the probability of generating (x, y) pairs in the region $Y > X$. We have several options for accomplishing this. They are:

1) we might raise the mean of $f(Y)$

2) we might lower the mean of $f(X)$

3) we might both raise the mean of $f(Y)$ and lower the mean of $f(X)$

4) we might generate an X value, say x_i, and then generate a Y value conditioned on this value x_i such that the possible Y values are limited to values of $Y \geq x_i$. The probability density used would be the tail of $f(Y)$ above the value $Y = x_i$

This is by no means an exhaustive list as other methods or combinations might be conceived. For example, one might increase the variance of one or both of the densities $f(Y)$ or $f(X)$. Increasing the variance will increase the probability of random variates from both the upper and lower tail regions of the density at the expense of values near the mean. This action may not necessarily increase the occurrence of (x, y) pairs in the region $Y \geq X$. Given the difficulties inherent in increasing the variance of the WMC PDF, described in detail in Section 5.2.2 below, this method of biasing was abandoned early in this work.

Case 1) Raise the Mean of $f(Y)$: If we raise, that is shift up, the mean of $f(Y)$ to form a new PDF $f^*(Y)$ then $f^*(Y)$ will generally be greater than $f(Y)$ in the upper region the Y axis of Fig. 5.3 or Fig. 5.4. This means that there will be a greater probability of generating Y random

variates in the upper region of the Y-axis where $f^*(Y) > f(Y)$. Considering the (x, y) probability space shown in Fig. 5.3, we see that this means that more (x, y) pairs will be generated in the area $Y \geq X$. This corresponds to the generation of more BPPM word errors.

In the region that $f^*(Y) \geq f(Y)$ then the weight $W(Y)$ will be less than one. Under the conditions of Case 1), the $f(X)$ density used to generate X random variates is unmodified so that $W(X) = 1.0$ for all X. With $W(Y)$ generally less than one in the important region, the modified sampling function $H^*(y \mid x)$ will generally be less than one in the summation of eq.(5.97) used to generate the estimate of P_{eA}.

Case 2) Lower the Mean of $f(X)$: If we lower, that is shift down, the mean of $f(X)$ to form a new PDF $f^*(X)$ then $f^*(X)$ will generally be greater than $f(X)$ in the lower region the X axis of Fig. 5.3 or Fig. 5.4. This means that there will be a greater probability of generating X random variates in the lower region of the X-axis where $f^*(X) > f(X)$. Considering the (x, y) probability space shown in Fig. 5.3, we see that this means that more (x, y) pairs will be generated in the area $Y \geq X$ which again corresponds to the generation of more BPPM word errors.

In the region that $f^*(X) \geq f(X)$ then the weight $W(X)$ will be less than one. Under the conditions of Case 2), the $f(Y)$ density used to generate Y random variates is unmodified so that $W(Y) = 1.0$ for all Y. Under these conditions, the modified sampling function $H^*(y \mid x)$ will generally be less than one in the summation of eq.(5.97) used to generate the estimate of P_{eA}.

Case 3) Both Raise the Mean of $f(Y)$ and Lower the Mean of $f(X)$: If we both lower, that is shift down, the mean of $f(X)$ to form a new PDF $f^*(X)$ and raise, that is shift up, the mean of $f(Y)$ to form a new PDF $f^*(Y)$ then we will enhance the probability that the (x, y) pair generated will be in the region $Y \geq X$. Both of the weight functions, $W(X)$ and $W(Y)$, will be less than one in general. Therefore the modified indicator function will generally be less than one in the error region. Thus we have increased the probability of an important event occurring and have caused the weight used to increment the error count to be less than one.

Case 4) Generate a Y Value Conditioned On the X Value Such That $Y \geq X$: If we condition the Y value on the value generated for the X random variate so that $Y \geq X$ is guaranteed, then we are guaranteed to generate an (x, y) pair in the error region. The modified joint probability function for conditional M-PPM IS will be $f(X)f(Y)$, modified by a constant, in the region $y \geq x$ and zero elsewhere. The constant insures that the area under the PDF is equal to one. This might be written as

$$\hat{f}(X,Y) = \begin{cases} \dfrac{f(X)f(Y)}{\epsilon} & y > x \\ 0 & x < y \end{cases} \tag{5.101}$$

where

$$\epsilon = \int_0^\infty \int_x^\infty f(X)f(Y)\,dy\,dx \tag{5.102}$$

$$= \int_0^\infty f(X)[1 - F_Y(X)]\,dx \tag{5.103}$$

$$= \int_0^\infty f(X)dx - \int_0^\infty f(X)F_Y(X)\,dx \tag{5.104}$$

$$= 1 - \int_0^\infty f(X)F_Y(X)\,dx \tag{5.105}$$

The expression for ϵ in eq.(5.105) has not yet been solved analytically. It can be seen that the value of ϵ will be less than one when one observes that the integral in eq.(5.105) is bounded by one. Consider that both $f(X)$ and $F_Y(X)$ are positive for all X and that $0 \leq F_Y(X) \leq 1.0$ for all X so that

$$\int_0^\infty f(X)F_Y(X)\,dx \leq \int_0^\infty f(X)dx = 1.0 \tag{5.106}$$

Thus ϵ is positive and less than one. The weight function under the conditions of case 4) can be expressed as

$$W(X,Y) \equiv \frac{f(X,Y)}{\hat{f}(X,Y)} = \begin{cases} \epsilon & y > x \\ 0 & y \leq x \end{cases} \tag{5.107}$$

Note that the weight is the constant ϵ regardless of which (x, y) pair is generated.

The properties desired of the densities at the output of the receiver have now been presented. Basically, we desire the weight function W(r) to be less than one, at least in the error region. It has been shown that this will cause a reduction in the number simulation samples required for a given error in the estimate. In the next section, specific methods to generate output densities with the desired properties through manipulation of the input are explored.

5.2 M-PPM Bias Function

5.2.1 On Input Versus Output Importance Sampling

Throughout Section 5.1, all of the M-PPM importance sampling development and expressions have been given in terms of the random variable measured at the output of the slot

integrator before the decision device. This is the *output* version of importances sampling, using the term introduced in Chapter 4. We have ascertained some desired properties for the densities of the output variables. We must now begin a discussion concerning how to cause the random variables at the output to have these properties via control of the input processes of the simulation.

Our PPM receiver model has two input densities, the signal dependent shot noise of the APD detector and the additive thermal noise of the amplifier and receiver electronics. The signal dependent noise can be biased but perhaps not in the same fashion as the additive noise densities. To consider the APD device model as a noise source, it might be thought of as a device which performs a mapping of a deterministic input value to a random output. The deterministic value is the amount of incident light, which results in a specific number of incident primary photo-electrons. The APD maps the number of incident primary photo-electrons to a random number of output electrons according to a specific probability density function. For the APD, the PDF which describes the mapping is not constant since it is a function of the number of primaries. In these terms, there are essentially two approaches by which to skew or bias the output of the APD in some desired fashion and yet preserve, as much as possible, the basic properties of the APD during a simulation. One might alter the input to the mapping, one might alter the mapping function or perhaps some combination of the two.

The reader may notice that little discussion has been offered concerning the biasing of the additive thermal noise of the receiver. This is because there is a wealth of information concerning the biasing of Gaussian input densities already available in the literature and presented in Chapter 4. We also find that often the thermal noise component of the receiver has a very small effect on the generation of PPM errors owing to the high gain of the APD under typical free-space optical communication system conditions. In other words, more often than not the receiver is shot noise limited rather than thermal noise limited and, as discussed in Section 4.3.2, under these conditions it is not always beneficial to bias the thermal noise component of the receiver.

5.2.2 On Increasing the Variance of the WMC PDF

Increasing the variance of the APD output is one method of modifying the mapping between the number of input primary photo-electrons and the output. Consider the WMC PDF of eq.(2.13) used to generate the output of the APD, repeated below for convenience.

$$f_{WMC}(r) = \frac{1}{\sqrt{2\pi}\,\sigma} \cdot \frac{1}{\left[1+\left(\frac{r-\bar{n}G}{\sigma\lambda}\right)\right]^{\frac{3}{2}}} \cdot \exp\left[\frac{-(r-\bar{n}G)^2}{2\sigma^2\left[1+\left(\frac{r-\bar{n}G}{\sigma\lambda}\right)\right]}\right] \tag{5.108}$$

where: r = the number of output electrons from the APD

$\bar{n}$ = the mean number of primary photo-electrons

G = the average gain of the APD

$\sigma^2 = \bar{n}\,G^2\,F$ = the variance

$F = \kappa G + (2-\frac{1}{G})(1-\kappa)$ = the excess noise factor

κ = ionization ratio

and $\lambda = \frac{\sqrt{\bar{n}\,F}}{F-1}$.

The variance is seen to be proportional to the mean number of primary photo-electrons, $\bar{n}$, the average gain, G, and the excess noise factor. What is not obvious from examination of eq.(5.108) is that an increase in the variance of the PDF causes a shift in the maximum value of the density. This is particularly evident when $\bar{n}$ is small. When $\bar{n}$ is large the WMC PDF appears approximately Gaussian, although it is not, and increasing the variance within the function, within eq.(5.108), begins to have the desired effect. The shift in the mean, or more precisely a change in the general shape of the function, causes the sample values to be generated in a fashion not intended by the increase in the variance. It was found that on average the generated weight values are not less than one dependent on the value of $\bar{n}$. Various methods of compensating for the change in mean as a function of the increase in variance were considered without success.

This method of biasing the WMC PDF was abandoned since it was not a particularly efficient method of generating more samples in the error region of the M-PPM receiver. Increasing the variance will cause more samples to be generated in the tails of the WMC PDF distribution but this action alone will not necessarily cause an increase in the number of output sample sets in the error region of the M-PPM receiver. This is because the M-PPM receiver is dealing with sets of random variable. While it is desired to generate random variates from the upper tail of the density during the empty slot periods, it is desired to generate random variates from the lower tail of the density during the pulse slot period. Simply increasing the variance will generate more variates of both ends of the density simultaneously and does not efficiently result in more M-PPM errors.

5.2.3 On Shifting the Mean of the WMC PDF

As introduced in Section 5.1.4, shifting the output of the APD in such a way as to increase the probability of occurrence in one or the other of the tails of the APD output distribution is a direct way in which to increase the error probability in the M-PPM receiver. For example, increasing the mean in one or more of the empty PPM slots will increase the probability of error. Similarly, the mean of the APD output during the slot containing the pulse could be shifted down in order to create more PPM word errors. Note that the ability to shift the APD output density is also of interest in the simulation of optical receivers using threshold detection. Shifting the APD density could be used to cause errors in an optical threshold receiver in exactly the same way as shifting the mean of the Gaussian input density of a threshold receiver subject to additive noise.

Shifting down, i.e. making smaller, the mean of a one-sided density such as the WMC PDF must be performed carefully. This is because physically an APD does not absorb electrons at the output in response to incident light; it creates current. As the mean of a one-sided density is shifted in a negative direction, parts of the lower portion of the density begin to exist and have a non-zero probability in the region of the variable below zero. This means that the model for an APD based on such a distribution has a finite probability of producing negative values for the number of output electrons. This would violate physical reality, of course, as it corresponds to the absorption of current at the output of the APD. There are two alternatives for preventing this problem.

Any solution to the problem of shifting the mean of a one-sided PDF down demands that the new PDF be truncated at zero and made to have a zero value of the probability density for all values less than zero. The two solutions offered here differ only in how the integral of the resulting probability function is made to maintain a value of one, as required by a legitimate PDF. When the original PDF is truncated below zero some of the area is thrown away. In the first solution, the truncated probability is spread evenly throughout the remainder of the PDF above zero, in the second solution the probability mass in the truncated portion can be concentrated at the $X = 0$ point of the output variable. This second solution increases the probability of generating zero output electrons from the APD. The increase in the probability of zero output electrons from the APD may or may not be the most desirable of the two alternatives offered here. The choice is dependent on the system and the desires of the simulator.

It might at first seem conceivable that the mean of the WMC PDF could be shifted, up or down, by shifting the input number of primary photo-electrons. This is equivalent to shifting the signal power levels up or down. This is not a viable alternative since the variance and the shape of the WMC PDF are also functions of the mean number of input primaries. In other words shifting the input does more than shift the output and will not in general produce the desired results. This is a property of the signal dependent nature of the APD noise source. Therefore, the shifting of the APD output is best performed by adding or subtracting a constant from the random variate generated by the WMC random generator in order to control the results better. In the case of shifting down, if the result is negative it is either thrown away and a new variate is generated or the result is set to zero. It is in this way that the two methods of shifting down, described above, are realized.

As discussed, the desired M-PPM word errors can be generated by shifting the APD output down in the slots containing the pulse, shifting the APD output up in the slots that are empty or a combination of both. We have implemented IS in the M-PPM receiver by shifting up the mean of the APD output during the empty slots and leaving the APD output values during the slot containing the pulse unchanged. This was implemented because of the ease in which the APD output can be shifted up versus the complications inherent in shifting a one-sided density down. The capability of shifting down the APD WMC density has been greatly enhanced through the discovery of a CDF for WMC density during the course of this research. This discovery came late in this work and has not yet found application in this aspect of the study. The implementation of an APD model able to shift both up and down is of greater concern to the simulation of threshold detection optical receivers and has been left for development during future research activities in this area.

5.2.4 On Conditional Importance Sampling and the M-PPM Receiver

The concept of conditional importance sampling in the context of M-PPM receiver importance sampling simulation was introduced in Section 5.1.4. It was described there in terms of the BPPM system variables X and Y but can be extended to the M-ary case without modification. In terms of the general M-PPM format, the generation of the random variates for all empty slots is conditioned on the full slot random variable. It should also be possible to condition the value of the random variates for the pulse slot on the empty slot values. The decision to consider here the conditioning the empty slot values on the pulse slot was made somewhat arbitrarily.

It should be pointed out that the concept of a conditional bias function for the M-PPM receiver has as much or more in common with the Pseudo-Optimum Importance Sampling (PIS) method of §4.2.4 as it does with the Conditional IS (CIS) bias function method discussed in §4.2.6. This is because the purpose of the conditioning during the importance sampling simulation of the M-PPM receiver is to restrict the input values to those which will cause an error. Recall that in PIS, a pseudo-threshold was chosen somewhere in the range of the input variable. This pseudo-threshold, δ, was to be selected so that the true (unknown) threshold was enclosed in the assumed importance region demarcated by δ. Having selected δ, random variates were selected only from that portion of the PDF which remained in the assumed importance region. The conditional importance sampling proposed here for the M-PPM receiver delivers the same effect.

Consider again the BPPM case, by conditioning the empty slot variable on the pulse slot variable only that portion of the PDF surface on the side of the X = Y line guaranteed to cause errors are generated. Thus we have set up a pseudo-threshold in the joint probability surface of Fig. 5.4. While this pseudo-threshold appears as a line in the BPPM case, and is thus easily visualized, it can be made to exist in any order (any value of M) M-PPM modulation. In a fashion analogous to the one dimensional PIS bias function method, the conditioning operation proposed here forms a demarcation in M-dimensional probability space. This demarcation results in the formation of an assumed importance region.

Considered another way, the conditional importance sampling bias function for M-PPM simulation is a natural consequence of the way in which the M-PPM receiver operates. Recall that the M-PPM receiver under study here required a conditional indicator function to describe the Monte Carlo simulation of the receiver. This indicator function selected that portion of the joint probability surface wherein some $\mathbf{r}_{i \neq q} \geq \mathrm{r}_q$. As discussed elsewhere, this is a conditional operation. The function $\mathrm{H}(\mathbf{r}_{i \neq q} \mid \mathrm{r}_q)$ assumes the value of 1.0 for a particular set of $\mathbf{r}_{i \neq q}$ conditioned on the particular realization of r_q. There is a direct analogy in the binary threshold receiver case. There, the indicator function selected a particular range of the output (or equivalently the input) random variable based on the threshold, where the threshold is a constant on the range of possible values. In a similar fashion the PIS bias function was based around the selection of a constant pseudo-threshold δ, a constant on the range of possible values.

All of the lessons learned from the study of PIS in Chapter 4 will apply to the conditional importance sampling bias function proposed here. These include the observed benefits as well as the observed precautions. One precaution concerned the guarantee of an error

in the probability of error estimate if the pseudo-threshold δ was chosen in such a way as not to include the true threshold. Under these conditions, every simulation sample resulted in an error. The sum of the weights in this case could not approach the true probability of error in the limit because each individual weight was less than the true probability of error. In the M-PPM case, the error region and thus the position to place our pseudo-threshold boundary would seem clear. This is not necessarily the case.

Examination of the conditional indicator function $H(\mathbf{r}_{i \neq q} \mid \mathbf{r}_q)$ seems to indicate exactly where the pseudo-boundary should be, at the point where $\mathbf{r}_{i \neq q} \geq \mathbf{r}_q$. This is true at the *output* density. We must condition the variables and processes at the *input* to the receiver simulation. In the BPPM case this might appear to be exactly at the $X = Y$ line where the variables X and Y are now the APD output random variable. This may not be the case due to ISI and filtering effects through the receiver. Therefore a guard region, an offset, should be incorporated into the generation of the conditional APD random variates. If we call this offset Θ then the APD output random variates Y would be generated so that they are greater than or equal to $(X - \Theta)$. This offset is positive and can be made closer and closer to zero as required. In this way, we gain some control of the conditioning operation during the simulation and thus some control on the average weight value.

Of practical concern is the exact relation of the individual samples for the output random variates of the empty slot APD values to the individual samples of the pulse slot APD values. The sampled M-PPM signal can be made to have a constant integer number of samples for each slot. This can be assured by careful design of the simulation through judicious choice of the sampling interval, dt, and the time length of a M-PPM word. The output formulation of conditional importance sampling introduced in Section 5.1.4 was in terms of the single random variable produced at the output of the integrator. When implementing this bias method, one must work with the output of the APD, ignoring for the moment the additive thermal noise source. Thus we must condition the group of random variates representing the output of the APD during an empty slot against the group of random variates representing a pulse slot. There are several possibilities for doing this and some consideration of the general relationship of the group of slot random variates to the single integrator output random variable $\mathbf{r}$ is in order.

The integrator random variate $\mathbf{r}_i$ can be simplistically thought of as the sum of the group of APD random variables which form a particular slot i. This group of random variables is subject to processing through the receiver filter which blurs this concept but this simplification will permit the development of several possible conditioning methods. Given that $\mathbf{r}_q$ is the sum

of the group of APD output random variates for the slot Q, assumed to contain the pulse, how do we generate the group of APD output random variates for a slot $r_{i \neq q}$ so that we condition $r_{i \neq q}$ to be greater than or equal to r_q, or more precisely $r_q - \Theta$ where Θ is the desired offset. Some of the possible options are:

(1) First generate the pulse slot APD output samples and form an average. This average would be used as the threshold for the generation of the empty slot APD output random variates. This method effectively finds the mean value of the pulse slot and then ensures that each APD output random variate generated in the empty slots is at least a great as this mean, or more precisely the mean minus Θ where Θ is the desired offset.

(2) First generate the pulse slot APD output samples and save each. Each individual sample is then used as a threshold, after taking account of the desired offset Θ, for the corresponding empty slot APD output random variate.

(3) This method is a variation of method 2) above. Once again we must first generate the pulse slot APD output samples and save them. This time the order of the samples would be mixed. The mixed up set of the *ON* slot APD samples would be used as the threshold (after being adjusted by Θ) for conditioning the range of each empty slot APD output random variable.

In all of these methods, the sum of the APD output random variates of an empty slot are conditioned so as to be greater than or equal to the sum of the APD output random variates from the *ON* slot, minus any offset.

To check this and to provide an example, let us apply some mathematics to method two. Let there be K samples per M-PPM slot and let each APD output current random variate be $E_1, E_2, \ldots, E_K$ during an empty slot. Let $O_1, O_2, \ldots, O_K$ be the set of APD output current random variables for the M-PPM *ON* slot, slot Q. Under the simplification that r_q is equal to the sum of the APD output samples we write

$$r_q = \sum_{k=1}^{K} O_k \tag{5.109}$$

Each empty slot APD output current random variable, E_i, is generated according to the upper tail of the WMC PDF, limited to values greater than or equal to $(O_i - \Theta)$. Hence each r_i, where $i \neq q$, is related to r_q as

$$r_{i \neq q} = \sum_{k=1}^{K} E_k \geq \sum_{k=1}^{K} (O_k - \Theta) = \sum_{k=1}^{K} O_k - K\Theta = r_q - K\Theta \tag{5.110}$$

We see that under the conditions of this simplification we are guaranteed to generate an r_i greater than or equal to r_q only if $\Theta = 0$. As discussed, the simplification used here does not incorporate the filtering effects on the APD output current random variates. The purpose of the offset Θ is to provide control and some guard against the effects of filter ISI on the summation. That is, it is a guard against making the assumed importance region too small. If the processing of the APD output random variates, including the effects of additive noise, causes the magnitude of any of the E_k to decrease or the magnitude of any of the O_k to increase in the summations of eq.(5.110) then we cannot comfortably insure that we will have included the entire importance region in our assumed one without the use of an offset Θ. As discussed in relation with the PIS bias method in §4.2.4, if the assumed importance region is smaller than the true one then the resulting estimation will be biased.

The inclusion of an offset in the conditional importance sampling method with cause the bias function of eq.(5.101) to become

$$f^*(X,Y) = \begin{cases} \dfrac{f(X)f(Y)}{\epsilon} & y > x - \Theta \\ 0 & \Theta - x < y \end{cases} \tag{5.111}$$

With the offset included, the constant becomes

$$\epsilon = \int_0^\infty \int_{x-\Theta}^\infty f(X)f(Y)\,dy\,dx \tag{5.112}$$

$$= \int_0^\infty f(X)[1 - F_Y(X - \Theta)]\,dx \tag{5.113}$$

$$= 1 - \int_0^\infty f(X)F_Y(X - \Theta)\,dx \tag{5.114}$$

since $F_Y(\cdot)$ is zero for values of X less than zero eq.(5.114) can be rewritten as

$$= 1 - \int_0^\infty f(X)F_Y(X)\,dx \tag{5.115}$$

From eq.(5.115) we see that ϵ is the expected value relative to the X density of the CDF of the Y random variable, that is

$$\epsilon = 1 - E_X[F_Y(X)] \tag{5.116}$$

This formulation suggests that value of ϵ might be estimated, perhaps along with the simulation procedure itself. The expected value in eq.(5.116) would be estimated as

$$E_X[F_Y(X)] \doteq \frac{1}{L}\sum_{i}^{L} F_Y(x_i) \tag{5.117}$$

where L is the number x_i random variates drawn during the course of the simulation. This suggests that the conditional IS simulation might proceed in the following manner. One would vary the parameter Θ until approximately 100% errors are made to occur. This provides an estimation of the importance region. At the same time, the $E_X[F_Y(X)]$ would be estimated for that particular Θ. With Θ, and knowledge of ϵ then the BER estimate is complete. It is ϵ. The trouble with this approach is that the estimate of ϵ can not be made accurately without as many trials as would be required in classical Monte Carlo simulation since importance sampling has not been applied to the estimation of ϵ! One has traded one estimation problem for another and once again the "optimum" bias function is beyond reach. It seems that the best option if one had to commit to using conditional IS for the M-PPM system would be to develop a strictly analytical method of determining, or estimating, ϵ after simulating the system to find Θ.

Unrelated to the inability to determine ϵ, it is possible to generate random variates according to the distribution of eq.(5.111). A method for doing so is presented in Section 5.4 where the implementation of conditional M-PPM importance sampling is described. The ability to demonstrate conditional IS for the M-PPM receiver rests on being able to solve, or at least approximate, the integral expression in eq.(5.114). To date, no suitable solution is known to exist. In preparation for this eventuality, and because parts of the technique have spin-off benefit to the PIS type simulation of APD based optical *threshold* receivers, the conditional M-PPM importance sampling method will remain a part of this discussion. In Section 5.4 we describe the implementation of this method ignoring the fact that the solution to eq.(5.114) does not yet exist. The exercise is worthwhile if only for its benefit to the APD based threshold detection receiver simulation problem.

5.3 M-PPM IS System Model

5.3.1 Performing IS Simulation

In the previous sections, the theory of applying IS to the simulation of M-PPM APD receivers has been developed. In the remainder of this chapter, the implementation of IS simulation of the receiver is described. This will necessitate the discussion of the system as a whole and the introduction of some auxiliary modules required for the BER determination of the receiver system.

Let us first discuss the simulation of a system without the added complication of importance sampling. This would be the system used for classical Monte Carlo simulation. Fig. 5.6 shows a complete system diagram of the M-PPM system that is to be simulated as it would appear for classical Monte Carlo simulation. The construction of such a system block diagram would be the first step in performing such a simulation using BOSS or any of simulation package. BOSS modules or combinations of BOSS modules will have to be constructed to perform each of the functions shown in the block diagram of Fig. 5.6.

The progression of waveform samples is for left to right in the system of Fig. 5.6. The binary bit source is a standard BOSS module. The user specifies only the bit rate, the relative probability of *ones* and *zeros* and a seed for the random generator. The seed is used in a pseudo-random bit generator and provides some minimal control over the resulting bit stream since the use of the same seed will spawn the generation of the same bit stream. The ability to repeatedly use the same bit sequence can be of a benefit on occasion. The output is a stream of logical *true* and *false* values which form a sampled binary bit stream. The sampling interval is set by the user at the start of the simulation run.

Progressing through the system, the bits are fed into a M-PPM encoder which uses the bit stream to form a M-PPM wave form. There is no standard BOSS module for encoding M-PPM from binary. This was constructed from modules in the BOSS library. The output waveform is still a stream of logical *true* and *false* values. These values are fed to a model of a laser which converts the logical *true* and *false* values into values of optical power. These values of optical power enter the receiver when they are applied to a model of the APD. The action of the APD has been described in detail elsewhere and need not be repeated here. The output of the APD is a value of current in microamperes which is immediately summed with a random value representing the thermal noise of the receiver system. These samples are then applied to a preamplifier/filter model. This block simulates the bandwidth limiting properties of the optical receiver preamplifier. As the type of filter and its properties have a great effect on the receiver simulation and on the application of importance sampling, it is discussed in more detail in Section 5.3.2 below. It might be pointed out that the model of the APD has no bandwidth limiting properties. These have been assumed to be much greater than the preamplifier so that the system bandwidth is set by the bandwidth of the preamplifier filter.

Following the preamplifier, is the slot integrator. This integrator has been implemented using a convolutional filter. The convolutional filter is a standard BOSS module in which the impulse response is specified by the user and is, by necessity, finite. As the signals are

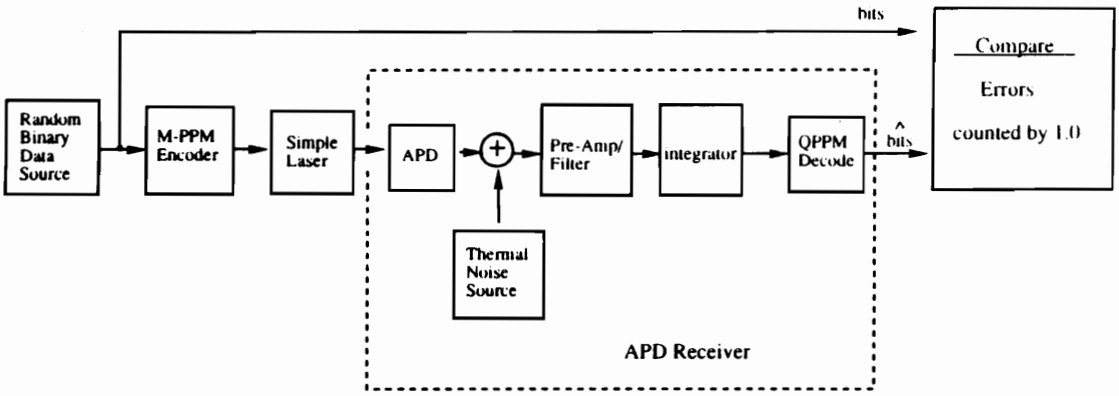


Fig. 5.6 Block Diagram for the Classical Monte Carlo Simulation of BER

processed, the module performs a continual convolution of the impulse response with the incoming signal.

The filter/integrator impulse response used in many of these studies is shown in Fig. 5.7. The impulse response is equal to one for the first half of the response thus providing the integrating action of a matched filter. The number of samples is made to correspond to the number of samples used in a PPM slot over the time period of a PPM slot. The second half of the impulse response is at a constant value of -0.05, again for the number of samples in one PPM slot. The integrator is made slightly imperfect with the addition of the section at -0.05. This section subtracts from the previous integrator value and adds a memory to the system of two slot lengths. Using this filter in an M-PPM IS simulation means the system length for each bit error is the number of samples in one word plus one more slot's worth of samples. This was felt to be realistic for actual system performance while providing a measure of control of the system memory for experimentation purposes.

Implied in the above integrator design is a perfect slot clock. The slot, bit and M-PPM word clocks used in the APD receiver simulation are assumed given and in perfect synchronization with the incoming signal. This simplification was necessary for the development of importance sampling and can be relaxed with future development. There is nothing about the importance sampling procedures developed here that require such a clock. An improvement to the system simulation would be to incorporate the circuitry required to derive the time clocks from the incoming signal. The effect of memory on these types of circuits will have to be explored and accounted for in the simulation process. An alternative method to account for timing jitter on the received signal has been suggested by Elrefaie, Townsend, Romeiser and Shanmugan[3]. This method was discussed in Section 4.5.2 and uses a perfect timing clock at the receiver and considers the BER at various sampling points along the bit.

The output of the integrator is applied to the M-PPM decoding module. Here, each of the samples representing the output of the slot integration is compared. The largest is selected as the slot containing the pulse. Based on the slot selection, a binary word is determined and is produced at the output of this module. This output is a waveform of logical *true* and *false* samples which represents the detected, i.e. the received, binary bit stream. It is this bit stream that is to be compared with original binary bit stream in order to detect the occurrence of errors.

The last block performs the comparison between the original bit stream and the

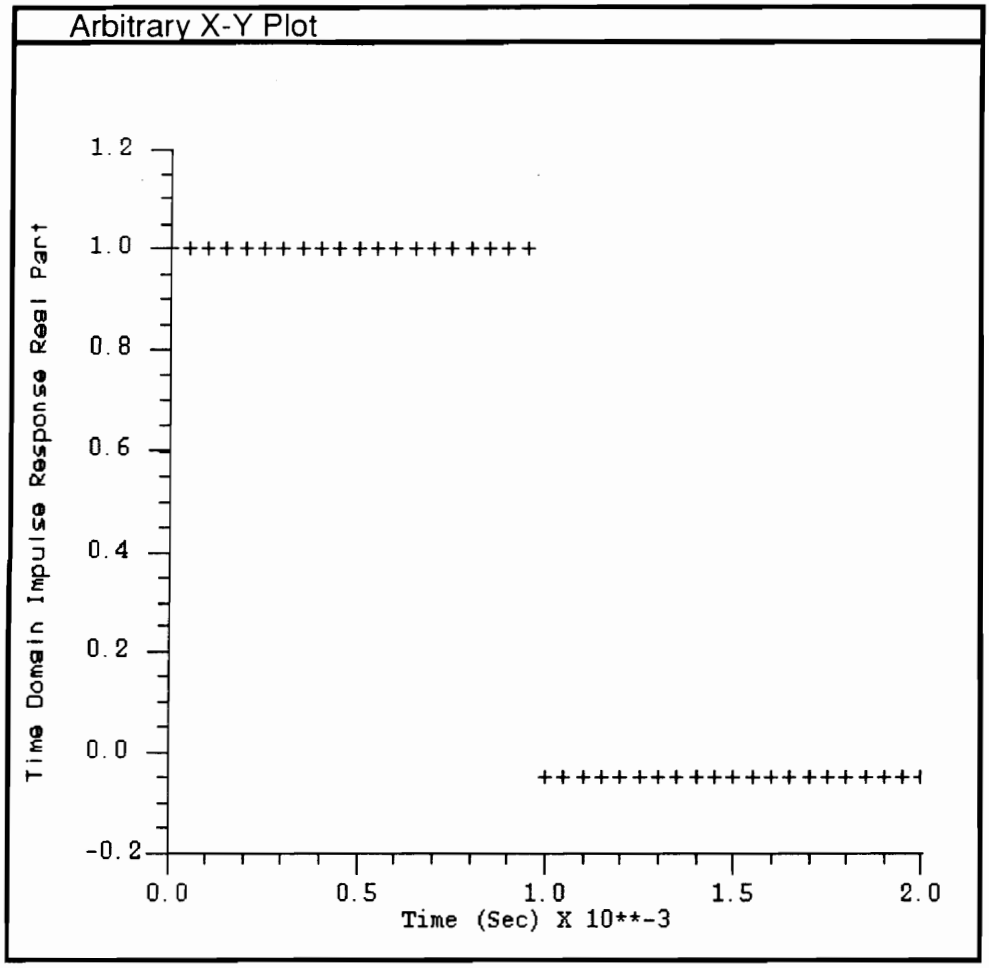


Fig. 5.7 Convolutional Filter Impulse Response

detected binary bit stream. This block must be implemented from BOSS library modules. In classical Monte Carlo simulation, there is primarily one output from this block which is a count of the total number of errors found. This can be augmented by a count of the total number of bits observed and, since the numbers are in the computer anyway, why not have the machine take the ratio of the number of errors to the number of bits to provide the estimate to the BER.

This completes the system description and simulation method for classical Monte Carlo. Several of the modules described above must be modified in order to implement importance sampling into the simulation. These changes can be observed through comparison of Fig. 5.6, the block diagram for the classical Monte Carlo simulation system, and Fig. 5.8 which shows the block diagram of a system similar to that of Fig. 5.6 but which incorporates importance sampling. First, notice that the error source inputs of the receiver, the APD and generator of thermal noise samples, have been modified. There is both an internal modification to the function of the block, represented by the * on the module diagram, as well as the addition of a second output. The additional output port provides the weight value of each random sample generated by these blocks. These weights must be collected and combined to form the appropriate weight value for each symbol. The symbol weight is applied to the module for detecting and counting bit errors.

The comparison module is also modified under importance sampling. In classical Monte Carlo simulation, each detected bit error increments the error count by one. In importance sampling simulation, each detected bit error increments the error count by the weight of the symbol in error. It is shown that additional outputs are desirable from the IS simulation of the system in order to monitor the validity of and thus control the procedure.

Thus for importance sampling simulation we require modification to the way the APD operates, (possibly) the way the thermal noise is generated and finally modification to the way that errors are counted. It also requires the addition of circuitry which will handle the weight values and prepare them for the counting procedure. The modification to the APD function is perhaps the biggest task. Two methods for performing this have been selected out of the possibilities outlined in Section 5.2 above. The first concerns the shifting of the APD output. This is discussed in Section 5.4 below. The second concerns the generation of conditional APD random variates and the calculation of the CDF of the WMC probability density. This is described in Section 5.5 below. As for the other aspects of importance sampling implementation, there is the determination of the symbol weight. This is presented in Section 5.3.3 below. The modifications to the calculation of the BER and the other outputs from the

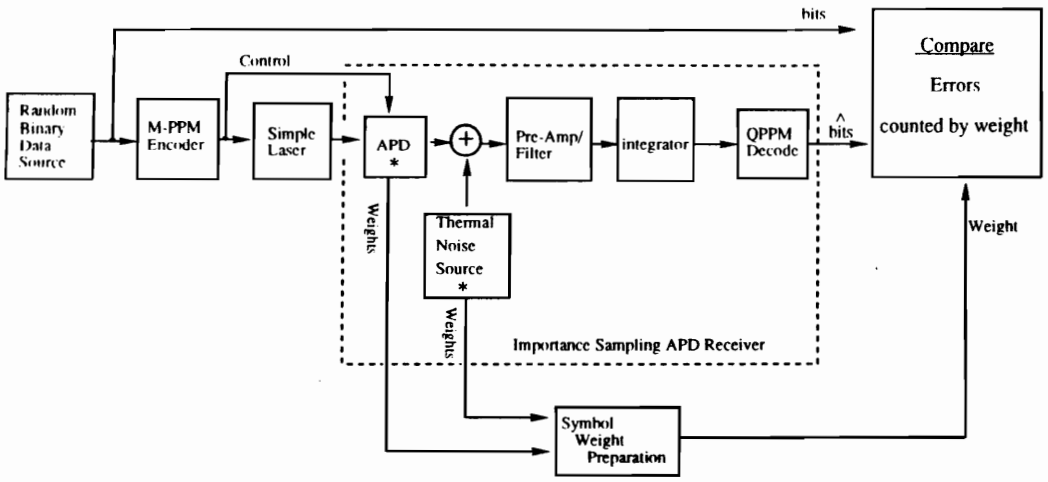


Fig. 5.8 Block Diagram for the Importance Sampling Simulation of BER

comparison module pertinent to the implementation of the IS is discussed in Sections 5.3.4 and 5.3.5 below.

5.3.2 On Filtering and System Memory

An infinite impulse response (IIR) filter, as the name implies, has an infinite memory. The incorporation of an IIR filter into an importance sampling simulation model requires that the system memory used for weight calculation to be an approximation of the actual system memory. The appropriate value to use for the system memory depends on the particular filter. Examination of the magnitude of the impulse response usually indicates a point at which the magnitude is insignificant. Choosing the system memory length to be at this value will generally introduce a negligible amount of bias into the simulation results. As discussed in Chapter 4, when in doubt of the system memory it is better to assume a longer value than one too short.

In order to gain a measure of control over the length of the system memory many of the simulation results were carried out using only the convolutional filter, leaving out any preamplifier filter, particularly any IIR filter. With the convolutional filter, the filter impulse response is specified by the user. Since the impulse response is specified by the user, the length of the memory associated with it is known.

5.3.3 On Calculating PPM Word Weight

For the implementation of importance sampling in M-PPM, the calculation of bit error rate must be altered in the method of counting word errors. Instead of counting each error and incrementing the count by one, the bit error count must be incremented by the weight of the PPM word found to be in error. Note that we have specified incrementing the *bit* error count by the M-PPM *word* weight. This works because of the inherent ability of randomly generated word errors to generate random bit errors, assuming random words are being transmitted. The random condition means that the error count will naturally sort out the relationship of bit errors to word errors established in Chapter 2. For example, there are two binary bits per QPPM word. If a word error is detected then one bit may be in error or both bits may be in error after decoding the incorrect word. The number of bits found in error depends on which word is assumed correct and how many bits in this word are in the same position as in the correct word. One bit at least will always be in error for a given word error, sometimes more than one. When

more than one bit is in error then the weight of that word is counted more than once. If all is random, as specified above, then the result will reflect the correct relationship between word errors and bit errors. The weight of the PPM word is found as follows.

Recall that each PPM slot is sampled. Each sample is processed through the laser and then the APD individually. Regardless of which method of importance sampling is used, at the output of the APD a random value has been generated according to a modified PDF. An integer number of these samples represent the PPM slot. The PPM slot is processed through an integrator or other equalizing filter in order to form a value representing the photon count for that slot. Since each of the slot samples was generated independently, the photon count is derived from the summation or other combination of a set of independent random variables. The probability of a particular value of slot count is thus the product of the probability of the individual samples. Similarly, the weight of a PPM slot is thus the product of the weights of each of the samples. Since the receiver treats the PPM word as if composed of independent PPM slots, the weight of a PPM word is the product of the weights of the individual slots. Putting these things together, the word weight is the product of the weights of the individual samples.

Given that each bit error is incremented by the word weight, the minimum memory for a particular bit error is one word length worth of samples. As discussed elsewhere, filtering and other processing may cause a particular bit decision to be influenced by a particular number of previous samples that we have referred to as the system memory. It can be seen then that each bit and thus each bit error is the result of processing the set of samples that make up the word from which it came. Each of these samples has had its impact on the final result.

In the simulation studies here, the weight of each individual APD sample has been produced at the output of all of our IS APD models concurrently with each individual output current sample. When these individual weights leave the APD, it may be necessary to apply them to a first-in first-out delay module in order to delay them by the same delay experienced by the corresponding current sample as it travels through the system processes. This is because the bit error counting is performed at the end of the system, naturally, and it is required that the weights arrive at the bit error counter at the same time as its corresponding sample. After delay, if any, the weights are applied to a module constructed to continually calculate the product of the K previous weights where K is the number of samples in the memory of the system. The module used is called the SLIDING SAMPLE PRODUCT and an example of its use can be seen in Fig. 5.9. The SLIDING SAMPLE PRODUCT might be described as a sliding

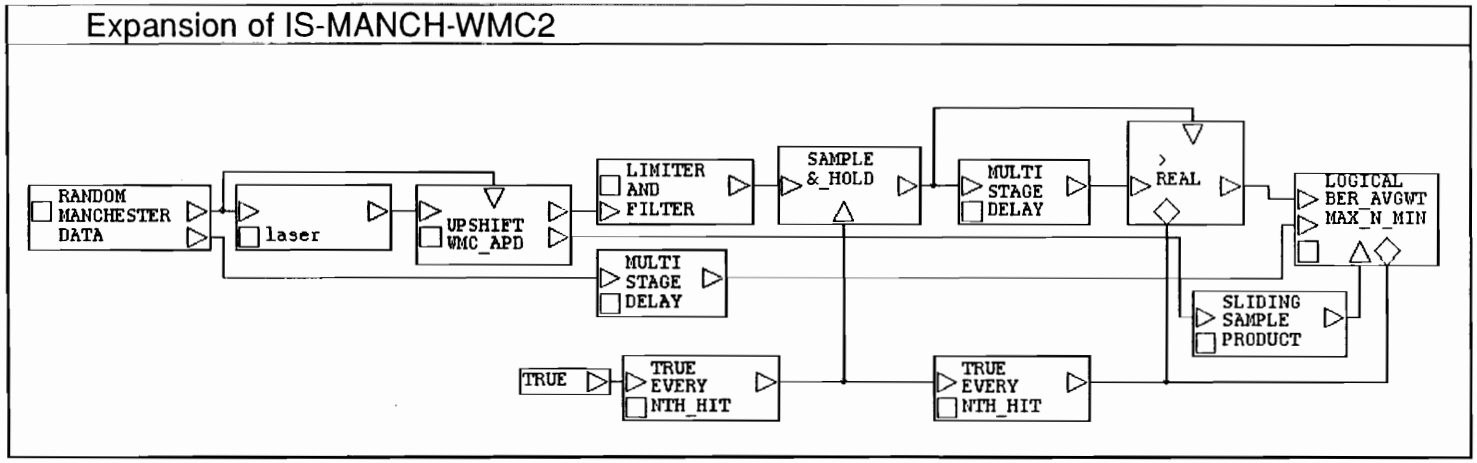


Fig. 5.9 A BPPM Importance Sampling Simulation System Block Diagram

product window since the individual weights are provided as input and the output is continually updated as the product of the last K of them. The number of weight samples used in the product is adjustable, being specified by the user at the beginning of the simulation. Fig. 5.9 is a completed simulation block diagram for the importance sampling simulation of a PPM system. The example shown in Fig. 5.9 is that of a BPPM importance sampling system using shifting of the APD density in order to bias the input.

5.3.4 On Calculating System Bit Error Probability

At this point we have developed expressions for estimating, both with and without importance sampling, the probability of error for a particular M-PPM symbol. The total error probability for a M-PPM link, given that we have the probability of error for each individual symbol, should be stated for completeness. Similarly, we have an expression for the variance of the estimator for an individual symbol but have said nothing concerning the variance of the overall bit error rate estimate.

As implied by eq.(5.1), the total system word error probability is the weighted sum of the individual error probabilities. Therefore

$$P_{eW} = P_{eA} \cdot P(A) + P_{eB} \cdot P(B) + \ldots + P_{eQ} \cdot P(Q) + \ldots + P_{eM} P(M) \tag{5.118}$$

where we have M different symbols and have assigned letter names to each symbol, beginning with A. If through a simulation process we generate an estimate for each of the symbol error probabilities then we form an estimate for the word error probability in the same manner

$$\hat{P}_{eW} = \hat{P}_{eA} \cdot P(A) + \hat{P}_{eB} \cdot P(B) + \ldots + \hat{P}_{eQ} \cdot P(Q) + \ldots e + \hat{P}_{eM} \cdot P(M) \tag{5.119}$$

The symbol error estimates are simply the classical Monte Carlo estimates or the importance sampling estimates of eq.(5.23) or eq.(5.52) respectively. Proper simulation of a communications link would require that the symbols be generated with the actual probability of occurrence. Alternatively, each symbol could be simulated and the estimation of each symbol error probability combined as per eq.(5.119). In the simulation work performed here, all symbols are assumed equally likely and a generated in that fashion. Errors are counted at the binary level and enough bits are counted to ensure that all symbols and all combinations of symbols have been adequately represented.

If the symbols are equally likely then the estimate for the word error probability becomes

$$\hat{P}_{eW} = \frac{1}{M}\sum_{i=1}^{M} \hat{P}_{eQi} \tag{5.120}$$

The summation becomes equal to $\hat{P}_{eQ}$ only if all of the symbol error probabilities are equal. This condition constitutes a symmetric error channel.

The variance of the total word error probability estimate is given by a weighted sum of the variances for each symbol estimate. This is because the variance of a sum of random variables is equal to the sum of the variances as long as they are uncorrelated. The probability of error for each symbol should be independent and uncorrelated under normal operating conditions. Therefore, assuming equal probabilities of all M symbols, the variance of the word probability estimate, $\sigma^2(\hat{P}_{eW})$, is

$$\sigma^2(\hat{P}_{eW}) = \frac{1}{M}\sum_{i=1}^{M} \sigma_i^2(\hat{P}_{eQ}) \tag{5.121}$$

The summation will equal $\sigma^2(\hat{P}_{eQ})$ only for a symmetric error channel.

5.3.5 On Calculating Average Weight

There are basically two possible reasons to simulate a communications system. A simulation can be used to examine waveforms through the system. It can also be used to estimate error rates through the production and counting of errors. It is the latter quantity that is of primary interest to this research. Therefore, the primary output from the simulation experiments performed here is a numerical value for the bit error rate. A number, or group of numbers, is generated at the culmination of a simulation run. A description of the module which produces this output is in order for better interpretation of the results.

A module was designed and constructed out of modules in the existing BOSS library to measure, calculate and to print at the end of a simulation run the necessary output quantities. There are actually several quantities of interest. These are

1) the number of bits found in error
2) the maximum weight value used in the calculation
3) the minimum weight value used in the calculation
4) the estimate of the average weight
5) the estimate of the BER
6) the total number of bits examined

The module constructed to produce and print these quantities is shown in Fig. 5.10 and is named: LOGICAL BER_AVGWT MAX_N_MIN. This module is installed into a system with an enable signal option and is designed to be enabled and executed only once per binary bit. The module works using the print signal modules seen on the right hand side of the block diagram of Fig. 5.10. These blocks print the value of the input when the print module is executed. This blocks are executed only once, at the end of the simulation, as they are controlled by the enable signal produced by the MODULO_N IMPULSE TRAIN block. This block is set to output a true only at the end of the simulation run.

Many of the above quantities, one through six, are self explanatory and some are related. The number of bits found to be in error is simply the result of a running count of the bit errors. Examination of the number of errors produced provides information concerning the amount of bias that has been applied. If all of the bits are in error then most likely the results are wrong as discussed in the Chapter 4 concerning PIS. If very few errors were generated then the results are weak at best since this means that not enough *important events* have been examined in order to make a good estimate. This means that either more bits must be simulated or more bias must be applied to the input densities. Similarly, the total number of bits examined, number 6) in the list above, is simply a running count of the number of times that the LOGICAL BER_AVGWT MAX_N_MIN module is executed. As the module is executed once per bit this is the total number of bits simulated. This number is used in the calculation of the BER and the average weight. Also, printing it for examination provides a check that the simulation was performed as intended.

Each weight value used in the calculation of the BER is examined by the LOGICAL BER_AVGWT MAX_N_MIN module and the value of the maximum and minimum weight is stored and printed. Examination of these quantities provides some indication of the status of the importance sampling procedure. Recalling some of the IS properties found in the literature and discussed in Chapter 4, it was shown that the maximum weight needed to be greater than or equal to the probability of error under study. If it was not, then the results could not be valid. It was also shown, through examination of the optimum bias function solution, that it is desirable to generate weight values that are constant or nearly so. Comparison of the maximum and minimum weight provides some indication of the variance in the weights generated. The ability to examine the minimum weight has provided vital information when trying to identify and eliminate underflow errors during the development of the importance sampling techniques described here.

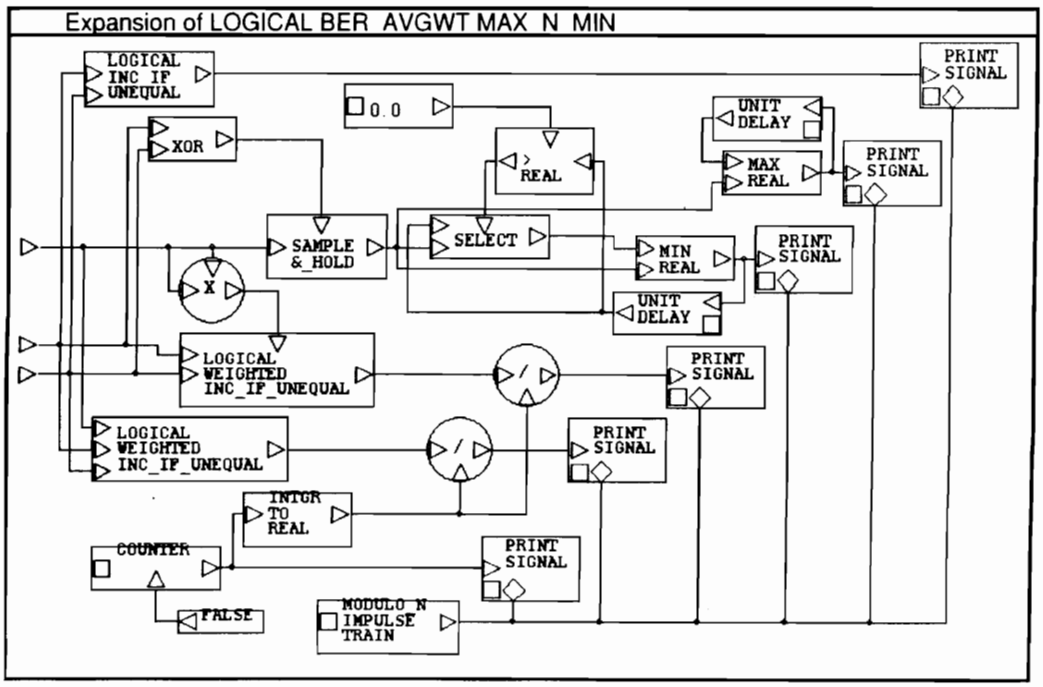


Fig. 5.10 Importance Sampling BER Estimation Module Block Diagram

Recall from Chapter 4 that the average weight was identified as a useful tool for gauging the status of the importance sampling procedure. It can be estimated along with the simulation and this estimate used to make a running estimate of the variance of the BER estimate. The ability to estimate the estimator variance is seen as a possible mechanism for providing feed-back into the importance sampling simulation procedure. This feed-back would be used to adjust the amount of bias applied to the input densities and to indicate and automatically stop the simulation when a user specified error in the estimate had been achieved. These principles and the procedure to estimate the average weight were described in Chapter 4 and will not be repeated here.

For clarity and completeness, the definition of the average weight for the M-PPM estimator and its relation to the M-PPM estimator variance are as follows. Retaining the concept of the average weight as defined for single dimensional importance sampling, the multi-dimensional average weight, $\overline{W}$, is defined as

$$\overline{W} \equiv \int_{-\infty}^{\infty} H(\mathbf{r}_{i \neq q} \mid \mathbf{r}_q) W(\mathbf{r}) f(\mathbf{r})\, d\mathbf{r} \tag{5.122}$$

Beginning with eq.(5.73), the variance of the importance sampling estimator is expressed in terms of the average weight as

$$\sigma_{IS}^2 = \frac{1}{N^*} \int_{-\infty}^{\infty} H(\mathbf{r}_{i \neq q} \mid \mathbf{r}_q) \left[W(\mathbf{r}) - P_{eQ} \right] f(\mathbf{r})\, d\mathbf{r} \tag{5.123}$$

$$= \frac{1}{N^*} \left[\int_{-\infty}^{\infty} H(\mathbf{r}_{i \neq q} \mid \mathbf{r}_q) W(\mathbf{r}) f(\mathbf{r})\, d\mathbf{r} - \int_{-\infty}^{\infty} H(\mathbf{r}_{i \neq q} \mid \mathbf{r}_q) P_{eQ} f(\mathbf{r})\, d\mathbf{r} \right] \tag{5.124}$$

$$= \frac{1}{N^*} \left[\overline{W} - P_{eQ}^2 \right] \tag{5.125}$$

$$= \frac{1}{N^*} \left[\frac{\overline{W}}{P_{eQ}} - P_{eQ} \right] P_{eQ} \tag{5.126}$$

Just as in the one dimensional case discussed in Chapter 4, we see that in order to obtain a variance smaller than the variance of the classical Monte Carlo estimator, $\overline{W}$ must be less than P_{eQ}. This is seen clearly if we equate the variance for the classical Monte Carlo estimator, eq.(5.44), to the variance of the importance sampling estimator above and form an improvement ratio in a manner similar to that in eq.(5.84).

$$\sigma_C^2 = \sigma_{IS}^2 \tag{5.127}$$

$$\frac{P_{eQ}(1-P_{eQ})}{N} = \frac{1}{N^*}\left[\frac{\overline{W}}{P_{eQ}} - P_{eQ}\right]P_{eQ} \tag{5.128}$$

$$R \equiv \frac{N}{N^*} = \frac{\left[1-P_{eQ}\right]}{\left[\frac{\overline{W}}{P_{eQ}} - P_{eQ}\right]} \tag{5.129}$$

The average weight is estimated during the simulation of a M-PPM simulation in the same manner described in Chapter 4 during the single dimensional importance sampling simulation. From the definition of the average weight in eq.(5.122), we notice that the weight function is average over the entire probability space and that is is averaged with respect to the unmodified probability density function $f(\mathbf{r})$. Using a similar manipulation as in the single dimensional case, it can be written that

$$\overline{W} \equiv \int_{-\infty}^{\infty} H(\mathbf{r}_{i\neq q} \mid \mathbf{r}_q) W(\mathbf{r}) f(\mathbf{r})\, d\mathbf{r} \tag{5.130}$$

$$= \int_{-\infty}^{\infty} H(\mathbf{r}_{i\neq q} \mid \mathbf{r}_q) W^2(\mathbf{r}) f^*(\mathbf{r})\, d\mathbf{r} \tag{5.131}$$

$$= E_*\left[H(\mathbf{r}_{i\neq q} \mid \mathbf{r}_q) W^2(\mathbf{r})\right] \tag{5.132}$$

which is the expected value with respect to the biased density $f^*(\mathbf{r})$. This expected value can be estimated in the usual fashion as

$$\hat{\overline{W}} \equiv \frac{1}{N^*}\sum_{j=1}^{N^*}\left[H(\mathbf{r}_{ij\neq qj} \mid \mathbf{r}_{qj}) W^2(\mathbf{r}_j)\right] \tag{5.133}$$

where $\hat{\overline{W}}$ is the estimate of the average weight. The accumulation of the $W^2(\mathbf{r}_j)$ terms and the division of the total by N^* is performed by the LOGICAL BER_AVGWT MAX_N_MIN module. The estimate of the average weight is printed at the end of the simulation along with the other quantities listed in 1) through 6) above.

5.4 M-PPM Shift Up IS APD Model

5.4.1 Generation of Shifted Random Variates

Shifting the mean of the output density of the APD random variate generator is accomplished be simply adding (or subtracting) a constant for each random variate that is generated. This is more desirable than shifting the input because it provides better control of the weight values generated and the amount of bias applied. The shift up IS APD block diagram is shown in Fig. 5.11. This block diagram might be contrasted with the block diagram of the unmodified APD module shown in Fig. 3.4 so that the modifications which cause the output to be shifted can be seen clearly. As shown, the output of the WMC random variate generator, the block labeled WMC_RANGEN, is summed with a constant. This summing operation shifts the mean of the output random variates. The value of the user supplied constant is the shift parameter and is provided through the block labeled CONST SELECT. The details of the operation of this block along with entire shift up APD model is presented in Section 5.4.2 below.

5.4.2 Weight Calculation

The weight is a simple matter to calculate when shifting the output, especially in the context of a block diagram simulation package such as BOSS. Recall that in the APD model, a value for the number of input primaries in determined from the incident input light at each simulation step. This $\bar{n}$ value is applied to a WMC random variate generator module which produces an output r based on $\bar{n}$ and the other APD parameters. The random variate r is then summed with the shift constant C to form the shifted output variate r′. The weight of the output variable r′ is found, as per the definition of weight, as

$$W(r') \equiv \frac{f(r')}{f^*(r')} = \frac{f(r')}{f(r)} \tag{5.134}$$

This is valid because the value of the shifted probability density $f^*(\cdot)$ at r′ is the same at the probability density of the unshifted density $f(\cdot)$ at r.

This realization simplifies the construction of an APD with a shiftable output which also calculates the weights of the modified output variables. This is because only one type of module which calculates the value of the WMC PDF need be designed and constructed. As no block previously existed in the BOSS simulation package which would calculate the value of the

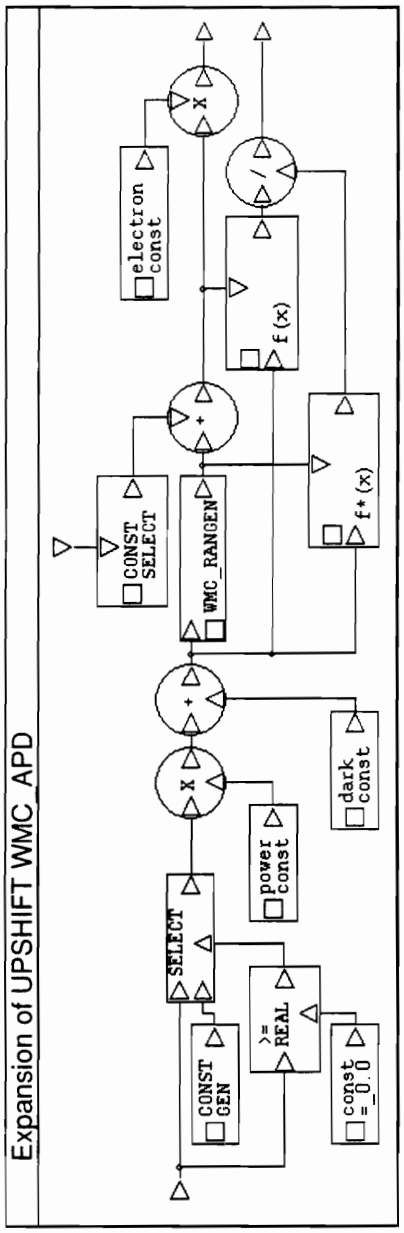


Fig. 5.11 Shift Up APD Block Diagram

WMC PDF a module was constructed using the primary module definition capability of BOSS. This is accomplished by first writing the FORTRAN subroutine code that will perform the desired operation of the block, in this case, the calculation of the function $f(r)$. This subroutine is incorporated into the BOSS operating structure and made to look like any of the other block diagram modules in the BOSS library. The completed module for calculating the value of the PDF function $f(r)$ is called PDF_CALC (WMC). This module has two inputs, $\bar{n}$ and r. The block has one output which is $f(r)$. It also requires as input the APD parameters for the average gain G and the ionization ratio κ. The subroutine code which forms the basis for this module is given in Appendix 5.1.

5.4.3 Shift Up IS APD Model

One of the major changes to the shift up APD model of Fig. 5.11 from that of the normal APD model of Fig. 3.4 is that at the top of the block diagram there is an input to control the shifting operation. This input controls the value of the shift parameter that is produced at the output of the CONST SELECT block. This block provides one of two user supplied constants depending on whether the logical valued control signal is *true* or *false*. For this application the input control signal is that of the logical M-PPM signal. When the control signal is *true*, during the *ON* periods of the input signal (which need not be PPM), the CONST SELECT block outputs one of two constant values it holds. When the control signal is *false*, during the OFF periods of the input signal, the CONST SELECT block outputs the other value. The values of these constants are provided by the user as part of the initialization of the simulation run. Thus for an importance sampling APD there is an additional input required in order to control the biasing operation.

For example, in the current applications of this APD we only want to shift up. Therefore the constants are provide as a positive constant for use when the control is *false* and a value of zero for use when the control is *true*. Thus during the empty slot periods, the logical M-PPM control signal is *false* and each of the empty slot APD output current samples will be modified by the addition of a positive constant. During the slot which contains the pulse, the control signal is *true* and thus each of the *ON* slot APD output current samples will not be modified since the additive constant is zero.

The two blocks required for calculating $f(r)$ and $f^{*}(r)$ also appear in the shift up APD block diagram of Fig. 5.11. These are the PDF_CALC (WMC) modules described in Section 5.4.2 which have been relabeled for clarity of function. The ratio of the output values from each

block is taken at each sample interval and provided as output from the APD block. Thus the weight of each output current random variate is produced along with its corresponding output current sample.

5.5 M-PPM Conditional IS APD Model

If conditional IS for the M-PPM APD receiver, as introduced in Section 5.1.4 and Section 5.2.4, is to be implemented a method of generating random variates according to the probability density of eq.(5.111) is required. As discussed in §5.2.4, it is possible to generate conditional random variates but not possible to know their weight. The generation of conditional random variates requires the design and implementation of a new module not found in any simulation library. This module will implement an efficient method of generating random variates according to the upper tail of a WMC probability density. The module would take as input a given threshold, the quantity $[x-\Theta]$ in eq.(5.101), and generate a WMC distributed random variate r according to that portion of the WMC density greater than or equal to $[x-\Theta]$. The generation of the random variate would be based on the input number of primary photo-electrons $\bar{n}$ in the usual fashion as well as the APD parameters of gain G and ionization ratio κ. The design and implementation of a module to generate variates from only the upper portion of a WMC PDF is presented in Section 5.5.1.

A second module has been implemented which performs a calculation of the CDF of the WMC PDF at the input threshold. This is $F_Y(\cdot)$ in eq.(5.105). The module to calculate the WMC CDF will be the subject of Section 5.5.2.

The remainder of Section 5.5 describes how these new modules are incorporated into a block diagram which will enable the implementation of conditional importance sampling for the M-PPM receiver. Section 5.5.3 describes the implementation of a conditional APD model. Section 5.5.4 explores a higher level module which completes the conditional importance sampling task. This module enables the formation of the M-PPM words in which the empty slots are conditioned on the pulse slot.

5.5.1 Generation of Conditional WMC Random Variates

Although it is not (yet) possible to calculate the value of ϵ in eq.(5.105), or more precisely eq.(5.114), it is fairly straightforward to generate WMC random variates conditioned to exist only in a given probability region as described by eq.(5.101). Continuing to concentrate

on the BPPM case, we will assume for discussion that it is desired to generate Y random variates conditioned on the X random variates. Recall that the X variates are the values of the APD output for the pulse containing the slot. The choice of conditioning the generation of the Y variates on the X variates was arbitrary. It is just as easy to generate X random variates conditioned on the Y (empty slot) random variates.

To generate random variates conditioned as per eq.(5.101), there is no reason to modify the way in which the X variate is generated. Hence the weight of the X random variates, W(X), will equal 1.0 for all X. After X is generated, it is used to condition Y by limiting the range of possible Y values that can be generated. Our modified probability density function for the Y random variates will look like the tail of the $f(Y)$ for $Y \geq T$, where the threshold $(x - \Theta)$ as been generalized to some value T. It will have the same shape but will be modified by a constant in order to cause the area under the PDF to be equal to one. That is, under the conditions of case 4) in Section 5.1.4, $f^*(Y)$ will be defined as

$$f^*(y_i) = \begin{cases} \dfrac{f(Y)}{1.0 - F_Y(T)} & y_i > T \\ 0 & y_i \leq T \end{cases} \tag{5.135}$$

where $$F_Y(T) = \int_{\infty}^{T} f(y)\, dy$$

$F_Y(T)$ is the cumulative distribution function of the PDF $f(Y)$ evaluated at T. Therefore $1.0 - F_Y(T)$ is the probability that $y_i \geq T$. Thus the area under the biased PDF $f^*(Y)$ is equal to unity.

Note that the weight function for optical PIS simulation will be constant, less than one, and can be expressed as

$$W(y_i) = \begin{cases} 1.0 - F_Y(T) & y_i > T \\ 0 & y_i \leq T \end{cases} \tag{5.136}$$

where T is the assumed threshold. Note also that the weight for each y_i is a constant regardless of which y_i variate is generated. The weight depends only on the value of the threshold.

The design and construction of a partial WMC random variate generator will also permit the pseudo-optimum type importance sampling to be performed on a APD based optical threshold receiver. An optical PIS simulation would be performed in a fashion directly analogous to the procedure described in §4.2.4 for the additive noise receiver. Thus for both M-PPM importance sampling as well as threshold receiver IS simulation, it is desired to design a

WMC random variate generator which generates variates from only the upper portion of the WMC PDF, above a given threshold, in an efficient manner. Ascheid's algorithm[4] for generating WMC random variates is both efficient and easy to implement. It is desired to maintain these properties in the partial WMC random variate generator to be constructed. Therefore consideration is given to incorporating this algorithm into the design.

A possible method of generating random variates above a given threshold would be to use a simple acceptance-rejection routine. In this method, a random variate is generated according to the Ascheid method in the normal fashion. If it is greater than or equal the desired threshold, we keep and use it. If it is less than the threshold, we reject it and generate a new one. This procedure would be repeated until a suitable random variate is obtained. The problem with this method is that we have lost all of the efficiency of the random number generation method. If T is the threshold and $F_Y(\cdot)$ is the CDF of the WMC PDF then, on average, we would obtain an acceptable random variate only $\left[1 - F_Y(T)\right] \cdot 100\%$ of the time. This might be an acceptable percentage for small values of T, perhaps in the neighborhood of the mean, but to generate random variates at only the extreme upper tail of the distribution would most likely require several cycles of the Ascheid random number generator before an acceptable variate is obtained and would become very time consuming. Let us consider a modification to Ascheid's algorithm.

It is desired to modify Ascheid's algorithm in order to generate random variates only from the upper portion of the WMC PDF, above some given threshold T. To understand how this might be done, let us discuss and examine how the method works. Ascheid's algorithm for generating WMC distributed random variates was introduced in §3.3.2. The reader may wish to refer to this before proceeding here. In truth, it is quite straight forward and can be explained easily with the aid of Fig. 5.12. Fig. 5.12 is a plot of the transformation function h(x). Recall that

$$z \equiv h(x) = x\left[\frac{x}{2\lambda} + \sqrt{\left(\frac{x}{2\lambda}\right)^2 + 1}\right] \tag{5.137}$$

Ascheid's method works by first generating an X random variate (Not to be confused with the slot A random variate X.) distributed according to a zero mean unit variance Gaussian PDF. This corresponds to choosing values along the X axis of Fig. 5.12 according to a Gaussian probability density. Each X value is mapped onto the Z axis through the transformation of eq.(5.137). Thus, Z random variates are generated and it is found that these variates are distributed according the the following PDF.

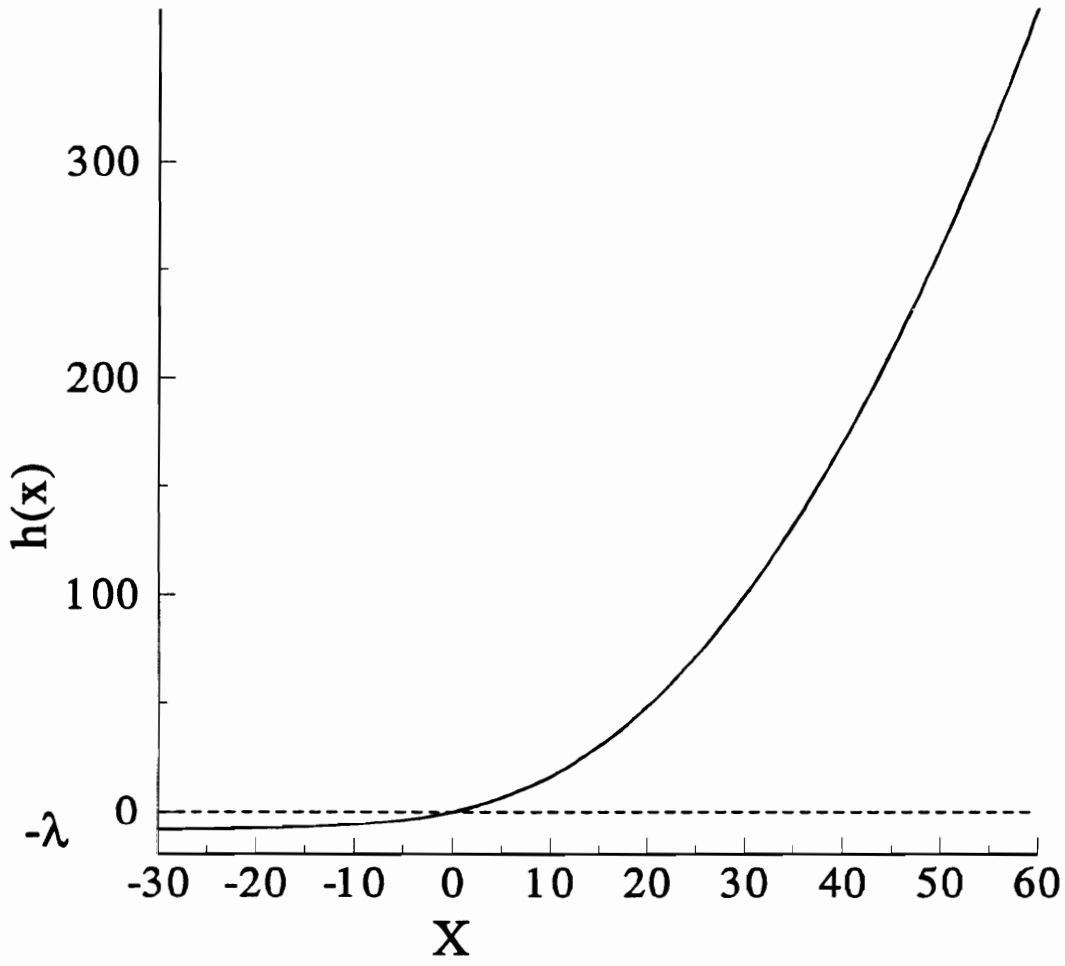


Fig. 5.12 Transformation Function h(x) Used in WMC Random Variate Generation Method by Ascheid[4].

The solid line is a plot of $Z \equiv h(x) = x\left[\frac{x}{2\lambda} + \sqrt{\left(\frac{x}{2\lambda}\right)^2 + 1}\right]$ with λ chosen arbitrarily. The abscissa variable, X, is a unit variance, zero mean Gaussian random variable used to generate random variates Z through the transformation h(x).

$$f_Z(z) = \frac{1}{\sqrt{2\pi}} \frac{1+\frac{z}{2\lambda}}{\left(1+\frac{z}{\lambda}\right)^{\frac{3}{2}}} \exp\left\{ -\frac{z^2}{2\left(1+\frac{z}{\lambda}\right)} \right\} \tag{5.138}$$

The PDF of eq.(5.138) is almost the desired WMC PDF in normalized form. A normalized WMC PDF is obtain through a simple acceptance-rejection procedure performed on the random variate Z. On average, less than two Z variates need be generated before a legitimate WMC random variate is found. The reader is referred again to §3.3.2 and to Ascheid[4] for an explanation of the remainder of the procedure. We will not address it here for it is at this point that we can modify the algorithm in order to generate WMC random variates above a desired threshold.

Suppose it is desired to generate WMC random variates only from the upper portion of the WMC density greater than or equal to some threshold T. The value T is some number of output electrons. The Z random variates are normalized so that the value T corresponds to a Z value, Z_T, as

$$Z_T = \frac{T - \bar{n}G}{\sigma} \tag{5.139}$$

Except for rejecting those Z random variates that do not meet the acceptance-rejection test, it is as if the WMC random variates were distributed along the Z axis of Fig. 5.12. Since the transformation of eq.(5.137) is monotonicly increasing, then one can easily generate random variates that are greater than or equal to some Z_T by limiting the Gaussian random variates to only those greater than or equal to some X_z. The value X_z is obtained as that X value which corresponds to the value Z_T. It is found by using the inverse of h(x) in eq.(5.137), which is

$$X_z = h^{-1}(Z_T) \equiv \frac{Z_T}{\sqrt{1+\frac{Z_T}{\lambda}}} \tag{5.140}$$

If we only generate Gaussian random variates greater than of equal to X_z then we will only generate WMC random variates greater than or equal to Z_T. Once they are de-normalized, Z random variates greater than or equal to Z_T correspond to WMC random variates greater than or equal to T. Thus, if an efficient method of generating Gaussian random variates above a given threshold X_z can be obtained, then we can generate WMC random variates above a desired threshold, T.

A method to generate Gaussian random variates limited to those above a given threshold is given by Knuth[5]. It is repeated here without explanation.

Step 1) Generate two random variates U and V which are uniformly distributed [0, 1]

Step 2) Form $X = \left[X_z^2 - 2\ln(V)\right]$

Step 3) IF $U \cdot X \geq X_z$ then GO TO Step 1)

else X is a Gaussian distributed random variate greater than or equal to X_z.

This method, which we will refer to as the *tail method*, is not valid for X_z values less than zero. The method was tested and investigated for efficiency. Its efficiency was found to be a function of the threshold X_z. As X_z approaches zero from a positive direction the efficiency becomes worse until it appears to not work at all for $X_z = 0$. Because this method is not valid for all possible X_z of interest to our simulation, alternate methods were devised in order to efficiently generate Gaussian random variates from the upper tail given any possible threshold X_z.

Through experimentation, the number of trials required to generate a valid Gaussian random variate for a given threshold using the tail method was determined. Using the error function to solve for the area under the Gaussian PDF above a given threshold, the average number of trials required by an acceptance-rejection procedure to generate Gaussian random variates above a threshold was also determined. In performing acceptance-rejection for Gaussian random variates in the positive portion of the PDF, it is assumed that each random variate is generated using the absolute value operation. That is, only positive Gaussian random variates are generated by the procedure. The average number of trials was determined for each method and are tabulated in Table 5.1 below for comparison.

TABLE 5.1 Efficiency of Random Variate Generators

Threshold	Tail Method number of trials	acceptance-rejection number of trials
2.0	1.2	21.9
1.0	1.5	3.15
0.7	1.86	2.066
0.66	1.877	1.96
0.6	2.12	1.82
0.5	2.3	1.62
0.1	8.8	1.08

It is seen that as the threshold gets larger the tail method becomes more efficient and the acceptance-rejection becomes less efficient. Conversely, as the threshold gets closer to zero the tail method begins to require an increasingly larger number of trials in order to generate a valid Gaussian random variate while the acceptance-rejection method becomes more efficient. This suggests that alternate methods of generating Gaussian random variates above a given threshold should be chosen based on the value of the threshold

A procedure for generating Gaussian random variates above a given threshold was implemented which uses alternate methods based on the value of the threshold. If the threshold is less than zero, acceptance-rejection is used with an unmodified Gaussian random number generator. If the threshold is greater than or equal to 0.66 then the tail method is used to generate random variates. If the threshold is between 0.0 and 0.66 then acceptance-rejection is used with a Gaussian random number generator made to generate only positive random variates. Using this variable technique, the average number of trials required to generate a suitable Gaussian random variate above any given threshold in the range of $\pm\infty$ should be less than two.

After the threshold limited Gaussian random variate is generated, it is applied to Ascheid's method in the usual fashion. A FORTRAN subroutine for implementing this conditional WMC random variate generator was written and incorporated into BOSS in the form of a primitive BOSS module called CIS_UP WMC_RANGEN. The code used to implement this function can be found in Appendix 5.2.

Note that a similar procedure can be used to generate the conditional WMC random variates below a given threshold. In fact only small modification to the code found in Appendix 5.2 would be required. The partial Gaussian random variate generator could remain essentially unchanged, only modifying the sign of the generator at particular points in order to generate Gaussian random variates below a given threshold instead of above. Otherwise the technique would remain the same. The construction of a partial WMC random variate generator for both above and below a given threshold will permit the importance sampling simulation of optical APD threshold receivers using a technique denoted as PIS in §4.2.4. This Pseudo-Importance Sampling technique will also require the ability to calculate the CDF of the WMC PDF. The Construction of a BOSS module to perform this calculation is described in Section 5.5.2 below.

5.5.2 Calculation of Cumulative Probability

The design of a second BOSS module which will perform the calculation the CDF will be the subject of this section. This is the calculation of $F_Y(T)$ in eq.(5.101) or eq.(5.135) where $F_Y(\cdot)$ is the CDF of the WMC PDF. The solution presented here is not to be considered the optimum numerical method to overcome the overflow and underflow conditions which can occur under certain simulation conditions utilizing this function. The determination of such an optimum solution was not the subject of this research. What is presented here is a discussion of the problems associated with the computer calculation of the WMC CDF and the method used in this study to overcome them.

The expression for the cumulative distribution function of the WMC PDF seems to be of manageable complexity when considered for manual solution. The computer implementation of the WMC CDF in eq.(2.21), repeated below for convenience, requires some study in order to prevent overflow and underflow errors from occurring during the calculation. The WMC CDF contains the APD parameters and can be written in a variety of forms depending on which expressions are chosen for the CDF of a Gaussian probability density which form part of the formula. It is given in eq.(5.141) below in a form which uses both the regular and the complimentary error function.

$$F_Y(x) = \frac{1}{2}\left\{\text{erfc}\left[\frac{-\lambda(Z-1)}{\sqrt{2Z}}\right]\right\} + \frac{\exp(2\lambda^2)}{2}\left\{1.0 - \text{erf}\left[\frac{\lambda(Z+1)}{\sqrt{2Z}}\right]\right\} \qquad (5.141)$$

where $Z = Z(x) \equiv \frac{x}{\lambda\sigma} + \left[1 - \frac{M}{\lambda\sigma}\right]$

$$\lambda = \frac{\sqrt{\bar{n}\,F}}{F - 1.0}$$

$$M = \bar{n}G$$

$$\sigma = \sqrt{\bar{n}\,F}G$$

F = excess noise factor of the APD

This form was found convenient for computer solution as explained below.

There are effectively two parts of eq.(5.141) which can cause error during the machine calculation of the function. Under some realistic input conditions, the $2\lambda^2$ term in the exponential can be very large. For example, simulation conditions used in the studies reported here sometimes caused $2\lambda^2$ to became equal to 8913.2 which is a problem because the exponential of 8913.2 is bigger than the largest number available on the SUN Workstation.

This is an overflow condition. It can be shown that the term

$$\left\{1.0 - \text{erf}\left[\frac{\lambda(Z+1)}{\sqrt{2Z}}\right]\right\}$$

is small under the conditions that result in a large $2\lambda^2$ term. That the above term is small can be seen if one considers that the product of the large exponential and the term in brackets, summed with a complimentary error function term (which is between zero and one) and divided by two forms a valid cumulative distribution function. A CDF is always between zero and one. The value of the error function approaches one in the limit for increasingly positive values of the error function argument. Therefore the difference, shown in the bracketed expression above, of the error function term subtracted from 1.0 becomes exceedingly small. It is as small as the $\exp(2\lambda^2)$ term is large. This is the underflow condition.

Using double precision arithmetic, the largest number possible on the SUN Workstation is approximately 2.2×10^{308}. It can be shown that

$$2.2 \times 10^{308} \doteq \exp(709.98) \tag{5.142}$$

We will use this fact to solve both the underflow and overflow conditions simultaneously by repeatedly factoring the right hand term of eq.(5.141) in the following manner

First, $2\lambda^2$ is tested. If it is less than 703 then nothing need be done and eq.(5.141) can be solved as it is written. The value 703 was chosen arbitrarily as being large but slightly less than the maximum 709.98 value. If $2\lambda^2$ is greater than or equal to 703 then we make the following algebraic manipulation to the right hand side of eq.(5.141)

$$\exp(2\lambda^2)\left\{1.0 - \text{erf}\left[\frac{\lambda(Z+1)}{\sqrt{2Z}}\right]\right\} = \exp(D)\left\{\exp(702) - \exp(702)\cdot\text{erf}\left[\frac{\lambda(Z+1)}{\sqrt{2Z}}\right]\right\} \tag{5.143}$$

where $D = 2\lambda^2 - 702$

When the manipulation is performed, each of the terms inside the brackets $\{\cdot\}$ on the RHS is calculated using quadruple precision mathematics. When completed, the term D is tested to determine if it is or is not still greater than of equal to 703. Assuming that it is, then D is decremented by 702 and the result from the calculation of the terms in the brackets is multiplied by $\exp(702)$. This process is repeated until D becomes less than 703 at which time the calculation is completed by multiplying the term in the brackets by $\exp(D)$. This manipulation might be written as

$$\exp(2\lambda^2)\left\{1.0 - \mathrm{erf}\left[\frac{\lambda(Z+1)}{\sqrt{2Z}}\right]\right\} =$$

$$= \exp(D)\left\{\exp(702)\left\{\exp(702)\left\{\ldots\exp(702)\left\{\exp(702) - \exp(702)\cdot\mathrm{erf}\left[\frac{\lambda(Z+1)}{\sqrt{2Z}}\right]\right\}\right\}\right\}\right\} \quad (5.144)$$

where $D = 2\lambda^2 - n(702)$ and n = the largest integer such that D is positive and less than 703. The calculations are performed from the inner most brackets to the outer. The error function is determined using double precision as this is the highest precision available for this built-in function. The subtraction and all of the multiplications are performed using quadruple precision mathematics. The result is combined with the first term on the RHS of eq.(5.141) in double precision with the final result a single precision real number in the range of zero to one.

There is one other underflow condition which can possibly occur. If the CDF function, eq.(5.141), is evaluated at values of x much much less than the mean then the function assumes very small values. The smallest single precision number possible on the SUN Workstation is approximately 1.175×10^{-38}. A test was devised to determine if the given x value would cause the function $F_Y(x)$ to be less than the smallest permissible real value. If it is determined that $F_Y(x)$ will be less than 1.175×10^{-38} then $F_Y(x)$ is assigned the value of zero and no further calculation is made. Under anticipated operating conditions it is not clear that the WMC CDF will ever be evaluated at x values which would result in this anticipated underflow condition. It was decided to implement the underflow test in order to make the CDF calculation code more robust.

The test is derived by first reformulating the CDF of eq.(5.141) in terms of the Q-function and then to use a bounding function to simplify and form an upper limit to $F_Y(x)$. Using the Q-function, eq.(5.141) can be written as

$$F_Y(x) = \frac{1}{2}\left[\mathrm{erfc}\left[\frac{-\lambda(Z-1)}{\sqrt{2Z}}\right] + \exp(2\lambda^2)\left\{Q\left[\frac{\lambda(Z+1)}{\sqrt{Z}}\right]\right\}\right] \quad (5.145)$$

Study of the behavior of each of the terms in the expression above determined that the right hand term was the site of the underflow for small values of x. The complimentary error function term on the left hand side, $\mathrm{erfc}(\cdot)$, was tested with many large values of the argument and not found to underflow. It is suspected that the built-in complimentary error function has underflow protection although documentation concerning the internal workings of the built-in functions was not available. Regardless of the reason, the development of a bounding function

for the right-hand term provides a suitable test for underflow values of the cumulative distribution function and was found to prevent the occurrence of underflow errors when tested.

The Q-function is bounded as

$$Q(\mathrm{v}) \leq \frac{1}{\sqrt{2\pi}\,\mathrm{v}} \exp\left\{\frac{-\mathrm{v}^2}{2}\right\} \quad \text{for } \mathrm{v} \geq 3 \tag{5.146}$$

This is a valid bound for the Q-function in eq.(5.145) as the argument of the Q-function is always larger than 3 in the underflow condition of concern. Using this bound, the following manipulation is made.

$$\exp(2\lambda^2)\left\{Q\left[\frac{\lambda(\mathrm{Z}+1)}{\sqrt{\mathrm{Z}}}\right]\right\} \leq \exp(2\lambda^2)\frac{\sqrt{\mathrm{Z}}}{\sqrt{2\pi}\,\lambda(\mathrm{Z}+1)}\exp\left[\frac{-\lambda^2(\mathrm{Z}+1)^2}{2\mathrm{Z}}\right] = \tag{5.147}$$

$$= \frac{\sqrt{\mathrm{Z}}}{\sqrt{2\pi}\,\lambda(\mathrm{Z}+1)}\exp(2\lambda^2)\exp\left[\frac{-2\lambda^2(\mathrm{Z}+1)^2}{4\mathrm{Z}}\right] \tag{5.148}$$

$$= \frac{\sqrt{\mathrm{Z}}}{\sqrt{2\pi}\,\lambda(\mathrm{Z}+1)}\exp\left[2\lambda^2\left(1-\frac{(\mathrm{Z}+1)^2}{4\mathrm{Z}}\right)\right] \tag{5.149}$$

$$= \frac{\sqrt{\mathrm{Z}}}{\sqrt{2\pi}\,\lambda(\mathrm{Z}+1)}\exp\left[-\frac{\lambda^2(\mathrm{Z}-1)^2}{2\mathrm{Z}}\right] \tag{5.150}$$

Hence, it is shown that

$$\exp(2\lambda^2)\left\{Q\left[\frac{\lambda(\mathrm{Z}+1)}{\sqrt{\mathrm{Z}}}\right]\right\} \leq \frac{\sqrt{\mathrm{Z}}}{\sqrt{2\pi}\,\lambda(\mathrm{Z}+1)}\exp\left[-\frac{\lambda^2(\mathrm{Z}-1)^2}{2\mathrm{Z}}\right] \tag{5.151}$$

The natural logarithm of the minimum single precision real number, 1.175×10^{-38}, shows that

$$1.175 \times 10^{-38} \doteq \exp(-87.3365) \tag{5.152}$$

To test for the underflow condition the following relationship is examined:

$$\text{Is } [-87] \text{ greater than or less than } \left[ln\left\{\frac{\sqrt{\mathrm{Z}}}{\sqrt{2\pi}\,\lambda(\mathrm{Z}+1)}\right\} - \frac{\lambda^2(\mathrm{Z}-1)^2}{2\mathrm{Z}}\right]? \tag{5.153}$$

where $ln\{\cdot\}$ is the natural logarithm. If the quantity on the right hand side is less than -87 then the CDF value, if evaluated, will be less than 1.175×10^{-38}. Therefore, we consider a CDF value this small to be zero and assign it that value without further computation. This test

prevents the underflow condition due to the attempted evaluation of incredibly small values of the CDF.

The FORTRAN subroutine which forms the basis of the primitive BOSS module developed to evaluate the CDF of the WMC density is given in Appendix 5.3.

5.5.3 Conditional IS APD Model

With the design and construction of the two new primary BOSS modules above, one for generating conditional WMC random variates and the other to calculate the CDF of the WMC density, we are able to partially implement a conditional importance sampling APD model. The APD model described here can be used to implement the conditional importance sampling simulation for the M-PPM receiver as described in Section 5.2.4. There are two block units which are needed. The first is a complete APD unit, similar to the normal APD model described in Section 3.3.2, but modified to take a threshold value. The threshold value is used with the conditional WMC random variate generator described in Section 5.5.1 so that the output of the APD is conditioned to be greater than or equal to the input threshold. The second block unit required is one which will generate APD values for the slot containing the pulse and then to use these values as the conditional values for the generation of the APD values of the empty slots. This is a tricky proposition when one considers that the pulse slot may be the last slot in the M-PPM word and that these values are needed before the other slots are generated. The solution requires a more complicated APD model than is normally used. It uses the conditional APD model inside. We first present the conditional APD model and then, in the next section, the system used to form conditional importance sampling M-PPM words.

The conditional APD model is shown in Fig. 5.13. It is called CIS_UP APD which is an abbreviation for Conditional Importance Sampling – Upper PDF tail APD. The CIS_UP APD is the same as the normal APD model, see Section 3.3.2 and Fig. 3.4, except for two changes. The first is the threshold input shown at the top. The threshold is assumed to be in units of the output current in microamperes. The input threshold value is therefore divided by the same electron constant used to convert the number of output electrons from the random number generator to current. The result at the output of the division block is a threshold value in units of number of electrons. This number is provided to two blocks: the CIS_UP WMC_RANGEN which is the conditional random number generator that was described in Section 5.5.1, and the WMC cumulative distribution calculation module that was described in Section 5.5.2.

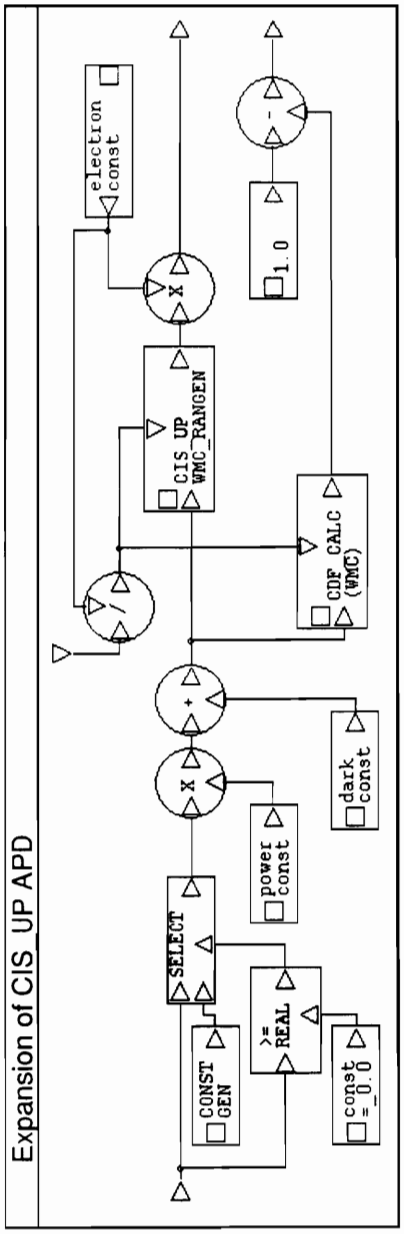


Fig. 5.13 Conditional APD Model: CIS_UP APD

The optical input portion of the CIS_UP APD functions in the same manner as the normal APD model so that the value at the output of the addition block is $\bar{n}$, the average number of primaries. The value $\bar{n}$ is provided to both the CIS_UP WMC_RANGEN module and the CDF_CALC (WMC) module. The conditional random number generator uses the $\bar{n}$ value and the threshold number to generate a WMC distributed random variate greater than or equal to the threshold number. This is the number of secondary electrons, conditioned on the threshold number, and it is multiplied by the electron constant in order to convert it to output current in microamperes. The $\bar{n}$ value is used by the CDF calculation block to calculate $F_Y(\cdot)$ at the threshold. This is the $F_Y(T)$ value shown in eq.(5.135) or eq.(5.136). The $F_Y(T)$ value calculated by the block is subtracted from 1.0 and output from the CIS_UP APD module as the weight value for the APD output current random variate.

This completes the conditional APD block. In order to perform conditional M-PPM importance sampling, it is necessary to adopt a method of generating conditional M-PPM words in which the samples in the *empty* M-PPM slots are conditioned on the samples generated for the M-PPM slot which contains the pulse. This is the subject of the next section.

5.5.4 Conditional IS M-PPM Word Generator

In order to demonstrate that the conditional importance sampling method proposed for use in the M-PPM receiver could be implemented, lacking only knowledge of ϵ as discussed previously, it was decided to construct the remaining module needed for the implementation of the bias function method introduced as case 4) in Section 5.1.4. The result is the M_ARY CIS APD block as shown in Fig. 5.14. Its function is to generate APD values for the slot containing the pulse and then to use these values as the conditional values for the generation of the APD values of the empty slots. This requires some storage capability since the pulse slot may not be the first slot in the M-PPM word and the pulse slot APD output values are needed in order to provide the conditioning information necessary for the generation of APD output values for the other slots.

The conditional importance sampling method chosen for implementation was case 2) in Section 5.2.4, that of saving the individual APD output samples from the *ON* slot and using them as the threshold values for each corresponding sample of the *OFF* slots. The M_ARY CIS APD block diagram shown in Fig. 5.14 performs this function and its operation is described in the following paragraphs. Please note that the M_ARY CIS APD block has been designed to be used with any value of M. All of the components which require knowledge of the word length

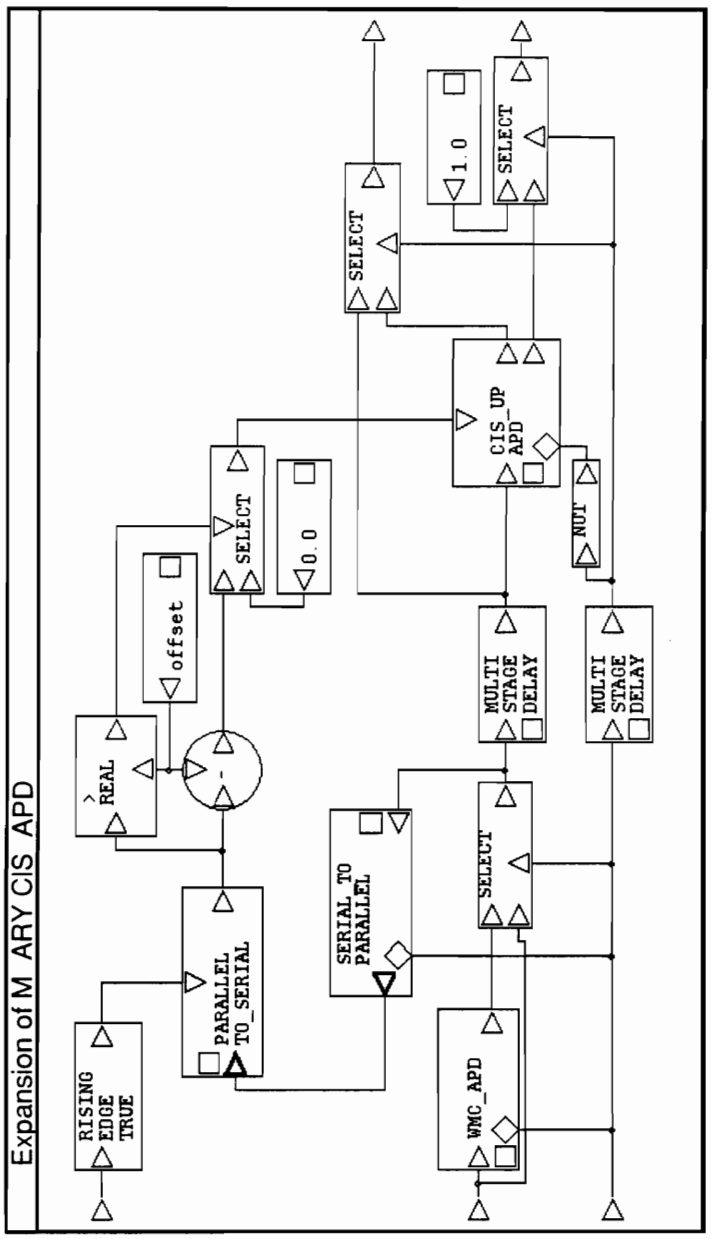


Fig. 5.14 Conditional IS M-PPM Word Generator: M_ARY CIS_APD

or the number of samples in a word have been keyed to an input parameter "M". The user specifies this constant as part of the simulation set-up so that all of the vector lengths and word delays are appropriate for the chosen M-PPM format. The only restriction is that there be an integer number of samples per PPM slot. Even this mild requirement could be lifted through careful (and more complicated) design of the individual components. The description concerning the internal workings of this module has been left to Appendix 5.4.

5.6 SUMMARY

In this chapter we have derived the application of importance sampling for the maximum likelihood M-PPM APD based optical receiver. This has never been explored before. This has included a derivation of the importance sampling estimator and the variance of this estimator. We have introduced the concept of a multi-dimensional probability of error region required for describing the M-PPM receiver. We have introduced the concept of a conditional importance sampling estimator and seen that such a conditional estimator is required in order to describe the operation of this receiver adequately. The conditional estimator is indispensable for describing receiver operation in terms of importance sampling.

We have derived the optimum M-PPM importance sampling bias function and shown that it is, in many ways, similar to the single dimensional optimum importance sampling bias function. One primary difference is that the importance region is multi-dimensional and is expressed in terms of a conditional operator function. This conditional nature of the importance region has suggested a conditional method of bias function implementation. The implementation of this conditional bias function method requires the solution of an integral of the area of a multidimensional WMC probability function over the area of the assumed importance region. This integral has not yet been solved. The solution of this integral would permit the solution of the probability of error for the M-PPM receiver analytically, assuming no additive noise component. Hence, while a solution to this integral must exist, it has not been determined as yet.

The mathematical treatment of the importance sampling for the M-PPM receiver allows for a variety of possible bias methods. The number of possibilities is large. Only two methods of biasing the M-PPM simulation statistics have been explored in detail here. The first is a shifting of the APD densities, a version of which has been implemented, as described in Section 5.3, and will be used to demonstrate IS for the M-PPM receiver. The results are presented in Chapter 6. The second is the fore mentioned conditional method. Modules for its

implementation have been constructed, especially those which have dual use in the simulation of APD threshold receivers.

We have discussed several methods of biasing the WMC APD statistics. It was shown that increasing the variance of the signal dependent WMC distribution was not a practical method of generating the desired statistics at the output of the M-PPM receiver. Shifting the mean, both positively and negatively, has been discussed and positive shifting of the APD density has been used to implement importance sampling for the M-PPM receiver. The conditional bias function method was also presented along with various options for specifying the relationship between the samples of the *ON* slot and those of the *OFF* slots.

Specific techniques for implementing the shifting and conditional importance sampling bias methods have been selected. BOSS modules designed to perform these methods of importance sampling have been designed and described here. These include the generation of conditional WMC random variates, the calculation of the cumulative distribution of the WMC PDF, the design of an APD model with biased output densities which are efficient and do not rely on table look-up methods in order to produce and bias the outputs. The generation of conditional WMC random variates has never been attempted or described before. A cumulative distribution function for the WMC probability density function has not been realized before. The intricacies of its machine calculation in the APD simulation setting of interest here have been reviewed and a BOSS primitive module has been designed which permits its calculation without error. Until this work, all reported simulations of an APD using importance sampling have utilized a look-up table. The utilization of the Ascheid's efficient WMC random variate generation technique in an importance sampling context has not been reported previously.

In the next chapter, the ideas and concepts developed here are confirmed through experimentation.

APPENDIX 5.1
Subroutine WMCPROBCALC

Subroutine Code for WMC PDF Calculation

This FORTRAN subroutine is used to form a new BOSS library module which has been labeled: PDF_CALC (WMC). This module accepts a value for the mean number of primary photo-electrons, $\bar{n}$, and a value for the number of secondary electrons, assigned the variable x in the code below. The module PDF_CALC (WMC) then calculates the value of the WMC probability density at the value x, $f_{WMC}(x)$, using this subroutine. The BOSS module PDF_CALC (WMC) uses the APD parameters of average gain and κ in the calculation.

```
      subroutine wmcprobcalc (x,meanx,avgain,k,alpha,probx)
c
c     x = the number of secondaries out of apd whose probability is desired.
c     meanx = the average number of primaries incident to the apd.
c     avgain = the average gain, G
c     k = the ionization ratio
c     alpha = the bias constant, 0<alpha<=1, the variance is increased by
c     1/alpha, set alpha=1 for unbiased WMC pdf calculation.
c     probx = the output, = the probability of x secondaries with WMC pdf.
c
      implicit real*8 (a-h,o-z)
      include '/bossdir/system/BOSSFORTRAN.INC'
c
      real x,meanx,avgain,k,alpha,probx
c
      cnst = 1.0/sqrt(2.0 * 3.141592653589793)
      fe = k*avgain+(2. - (1./avgain))*(1. - k)
      root = sqrt(fe * meanx)
      sig = avgain * root
c
      sig = sig/sqrt(alpha)
c
      delta = root/(fe - 1.)
      delmean = x - (meanx * avgain)
      denom = 1.0 + (delmean/(sig * delta))
c
      probx=cnst/(sig*(denom**1.5))*exp( - (delmean**2)/(2.*(sig**2)*denom))
c
      return
      end
```

APPENDIX 5.2
Subroutine CISWMCRAN

Subroutine Code for Generating Conditional WMC Random Variates

This FORTRAN subroutine is used to form a new BOSS library module which has been labeled: CIS_UP WMC_RANGEN. This module accepts a value for the mean number of primary photo-electrons, $\bar{n}$, and a threshold value in units of number of secondary electrons. The module CIS_UP WMC_RANGEN then produces a random variate distributed as per the WMC probability density function limited to those values above the threshold value. This random variate is used as a value for the number of secondary electrons. The BOSS module CIS_UP WMC_RANGEN uses the APD parameters of average gain and κ in the calculation.

The WMC random variates are generated by various methods depending on the value of the threshold. All of them are modifications to the method of generating WMC random variates first given by Ascheid[3]. The methods differ only in how the Gaussian distributed random variate used as a seed for the WMC random variate is generated. Depending on the value of the threshold, the proper Gaussian random variate is generated by acceptance-rejection or by an algorithm for generating Gaussian random variates from only the tail of the Gaussian density[4].

```
      subroutine ciswmcran(avgprinum,avggain,elecnum,s1,s2,fe,sseed,isfirst,pseed,k,thresh)
c
      implicit none
      include '/bossdir/system/BOSSFORTRAN.INC'
c
      integer sseed,pseed
      real avgprinum, avggain,elecnum,s1,s2,fe,k
      real ran,root,u,g,x
      real delta,sigma,y
      real v1,v2,r,fac,gstore
      real thresh,gt,tp,T
      integer isfirst,arej,ab
c
      if(isfirst.eq.0) then
          sseed=pseed
          fe=k*avggain+(2.-1./avggain)*(1.-k)
          s1=sqrt(fe)/(fe-1.)
          s2=avggain*sqrt(fe)
          isfirst=1
      end if
c
      root = sqrt(avgprinum)
```

```
        delta = s1*root
        sigma = s2*root
c
        T=(thresh-(avgprinum*avggain))/sigma
        gt=delta/((2*delta)+T)
        tp=T/sqrt(1+(T/delta))
c
        if (tp.le.0.0) then
            ab=0
            arej=7
        else
            if (tp.ge.0.66) then
                ab=7
                arej=0
            else
                ab=7
                arej=7
            endif
        endif
c
2       u = ran(sseed)*gt
c
c   generate gaussian random number
c
        if (arej.gt.0) then
1           v1=2.0*ran(sseed)-1.0
            v2=2.0*ran(sseed)-1.0
            r=v1**2+v2**2
            if(r.ge.1.0) go to 1
                fac=sqrt(-2.0*log(r)/r)
                if (ab.gt.0) then
                    gstore=abs(v1*fac)
                    g=abs(v2*fac)
                else
                    gstore=v1*fac
                    g=v2*fac
                endif
c
                if (g.lt.tp) then
                    if (gstore.lt.tp) then
                    goto 1
                else
                    g=gstore
                endif
            endif
c
            else
3           v1=ran(sseed)
            v2=ran(sseed)
            g=sqrt(tp**2-(2*log(v1)))
            if ((v2*g).gt.tp) goto 3
```

```
        endif
c
c    transform variable
c
        x= g/2.0/delta
        y = g*(x+sqrt(1.0+x**2))
        x = 1.0/(2.0+y/delta)
c
c   Test below changed from .gt. to .le. KRB 2/17/92
        if (x .lt. u) go to 2
c
        elecnum = sigma*y + avgprinum*avggain
c
c   Test elecnum >= 0.0 else reject
        if (elecnum .lt. 0.0) go to 2
c
        return
        end
```

APPENDIX 5.3
Subroutine WMCDFCALC

Subroutine Code for Evaluating the Cumulative Distribution Function of the WMC Probability Density

This FORTRAN subroutine is used to form a new BOSS library module which has been labeled: CDF_CALC (WMC). This module accepts a value for the mean number of primary photo-electrons, $\bar{n}$, and a threshold value, x, in units of number of secondary electrons. The module CDF_CALC (WMC) then calculates the value of the WMC cumulative distribution function at the value x, $F_{WMC}(x)$, using this subroutine. The BOSS module CDF_CALC (WMC) uses the APD parameters of average gain and κ in the calculation. The calculation of the CDF is based on the CDF of the Inverse Gaussian density.

```
      subroutine wmcdfcalc (rxbar,rthr,rprob,rgain,rk)
c
c     rxbar = the average number of primaries incident to the apd.
c     rgain = the average gain, G
c     rk = the ionization ratio
c     rthr=rthreshold
c     rprob = the output, = the probability of x<rthr
c
      implicit none
      real*16 qq702,qdd,qqe,qlhs
      real*8 dfe,dlam,dzthr,drt2z,dx1,dx2,da,dc,dprb,d_erf,d_erfc
      real rxbar,rthr,rprob,rgain,rk
c
      dfe=(rk*rgain)+(2.0 - (1.0/rgain))*(1.0 - rk)
      dlam=dsqrt(dfe*rxbar)/(dfe - 1.0)
c
      dzthr=(1.0/dfe)+(rthr*(dfe - 1.0)/(rxbar*dfe*rgain))
      drt2z=dsqrt(2.0*dzthr)
c
      dx2=dlam*(dzthr+1.0)/drt2z
c
      da= - (dlam**2)*((dzthr - 1)**2)/(2*dzthr)
      dc=da - DLOG(dx2)
c
      if (dc .lt. - 86.0) then
        rprob=0.0
      else
        dx1=( - dlam*(dzthr - 1.0))/drt2z
        da=2.0*(dlam**2)
        if (da .gt. 703.0) then
```

```
          dc=da - 702.0
          qq702=QEXP(QEXTD(702.0))
          qdd=qq702*d_erf(dx2)
          qqe=qq702 - qdd
10         if (dc .gt. 703.0) then
             dc=dc - 702.0
             qqe=qq702*qqe
             go to 10
           else
            qlhs=QEXP(QEXTD(dc))*qqe
           end if
        else
          qqe=QEXP(QEXTD(da))
          qdd=qqe*d_erf(dx2)
          qlhs=qqe - qdd
        end if
        dprb=0.5*(d_erfc(dx1)+DBLEQ(qlhs))
        rprob=sngl(dprb)
      end if
c
      return
      end
```

APPENDIX 5.4
The M-ARY CIS APD Module

The M_ARY CIS APD is shown in Fig. 5.9 and has three inputs. These are, from bottom left and proceeding clockwise, the logical M-PPM signal, the optical M-PPM signal and the M-PPM word clock. The logical M-PPM signal is the same signal used to drive the laser. It is simply a logical *true* or *false* assignment to each of the samples which make up the M-PPM word. The optical M-PPM signal is the received optical signal and represents the optical power incident on the APD. The M-PPM word clock is a logical 50% duty cycle square wave whose rising edge corresponds to the beginning of each PPM word. This signal is necessary for timing purposes.

The M_ARY CIS APD has two outputs. These are, from top right and proceeding clockwise, the APD output current in micro-Amps and the weight of each output current sample. It may be noticed that even though this module contains two complete APD modules the inputs and the outputs make it seem as though it is a single importance sampling APD with two extra inputs, the M-PPM word clock and the logical M-PPM signal. What is not obvious from looking only at the inputs and outputs is that there is a delay of one complete M-PPM word from the time that the optical energy arrives until the time that the corresponding output current is available. Let us explore the reason for this.

The M_ARY CIS APD shown in Fig. 5.9 works in the following manner. As the input optical M-PPM signal arrives at the APD so also does a logical M-PPM signal which serves, sometimes in combination with the M-PPM word clock, as a control signal for the internal processing of the optical signal. The optical signal is immediately applied to a WMC_APD module *OR* it is not, depending on whether the logical M-PPM signal is *true* or *false*. The WMC_APD block is a normal unaltered and non-importance sampling APD model except for the addition of the small diamond shaped input at the bottom of the WMC_APD block which is an enable control. When the input to this control is *true* then the WMC_APD functions normally. When the input to the enable control is *false* then the module is effectively turned off and the output value is frozen at the last value prior to being disabled. The WMC_APD enable control is tied to the logical M-PPM signal so that during the empty slots the WMC_APD is turned off and only when the *ON* slot arrives does the WMC_APD take the input optical power and generate an output current sample.

The SELECT block following the WMC_APD is also controlled by the logical M-PPM signal. When the control signal is *true* then the output of the SELECT block is the upper signal which corresponds to the output of the WMC_APD. When the control signal is *false*, during the M-PPM *OFF* slots, then the output of the SELECT is the lower signal input which is the optical power values for the *OFF* slots. The output of the SELECT block goes to two different modules. The first is a MULTI STAGE DELAY and the second is a SERIAL_TO PARALLEL converter.

The SERIAL_TO PARALLEL converter is also controlled via an enable input connected to the logical M-PPM input signal. The enable signal turns the SERIAL_TO PARALLEL converter on during the M-PPM *ON* slots. When the converter is on, the input values, which in this case are the output current samples from the WMC_APD, are read into a vector. The number of elements in the vector is the same as the number of samples in an M-PPM slot. The action of the SERIAL_TO PARALLEL converter is thus to collect the output current samples of the M-PPM *ON* slot into a vector. The action of this block will be the same regardless of the position of the M-PPM *ON* slot in the M-PPM word since it is controlled by the logical M-PPM signal. At the end of the *ON* slot the output of the SERIAL_TO PARALLEL converter is a vector containing the output current samples of the M-PPM *ON* slot and as a result of the action of the enable control these samples will remain there until the SERIAL_TO PARALLEL converter is activated again. Since there is only one *ON* slot per M-PPM word, the set of APD output current values have been collected and saved and will be used as the threshold values during the generation of the *OFF* slot samples.

At the beginning of the next M-PPM word, the input M-PPM word clock will rise *true*. The M-PPM word clock is sampled by a module called the RISING EDGE TRUE which looks for the *false* to *true* transition. When the transition occurs the module output is a single *true* pulse. This pulse commands the PARALLEL TO_SERIAL converter to latch and store the contents of the vector at the output of the SERIAL_TO PARALLEL converter. While the PARALLEL TO_SERIAL converter control input is false, as it is throughout the remainder of the M-PPM word, the stored vector is read out serially. When the end of the vector is reached, the output begins again at the first element of the vector. The PARALLEL TO_SERIAL converter is programed to act as a circular shift register under these conditions. Thus the output current samples from the M-PPM *ON* slot are created and saved and made available in a cyclical fashion beginning at the next M-PPM word.

Since all of the M-PPM slots must be examined in order to be sure that the M-PPM *ON*

slot has been processed and made available at the output of the PARALLEL TO_SERIAL converter, there is a one M-PPM word delay following the WMC_APD and SELECT block. In BOSS, the module which performs this sort of delay is called a MULTI STAGE DELAY. A one word delay is installed following the the output of the SELECT. Recall that the output of the SELECT at this point is the optical background power during the M-PPM *OFF* slots as the APD current has not been generated as yet for the M-PPM *OFF* slots. In order to keep the logical M-PPM control signal in sync with the optical signal, this control signal is delayed for one M-PPM word as well using the second, lower, MULTI STAGE DELAY module.

As each WMC_APD output sample is read out from the PARALLEL TO_SERIAL block, the offset constant is subtracted from each. In order to insure that negative threshold values are not generated, each sample is tested to insure that it is greater than the offset constant. If it is not then the threshold value is selected as zero. Otherwise, each output current sample is diminished by the offset and sent to the CIS_UP APD for use as a threshold.

The CIS_UP APD takes the delayed optical signal as input, as well as the threshold values. The CIS_UP APD is controlled by the logical M-PPM signal in much the same way as the WMC_APD block was. When the logical M-PPM signal is *false*, the CIS_UP APD is turned on, the input from the MULTI STAGE DELAY is the optical power corresponding to the M-PPM *OFF* slots, and the threshold generated during the M-PPM *ON* slots are used to condition the random variates generated by the CIS_UP APD. During the M-PPM *OFF* slots, when the CIS_UP APD is on, the SELECT module which controls the APD current output selects the values generated by the CIS_UP APD module to be the output. During the M-PPM *ON* slots, the CIS_UP APD is disabled and the SELECT block at the output selects the output of the MULTI STAGE DELAY to be the output. Recall that it is during these times that the previously generated WMC_APD output current samples are at the output of the MULTI STAGE DELAY.

As for the weights, they are also applied to a SELECT block at the weight output which is controlled by the logical M-PPM signal. When the logical level is *true*, during the M-PPM *ON* slot, the output weight is selected as the constant value 1.0. This is the correct weight value because the output current samples at this time are the current samples generated by the WMC_APD which was not modified. When the logical level of the M-PPM control signal is *false*, during the M-PPM *OFF* slots, the weights are selected as those generated by the CIS_UP APD module. This is valid since the CIS_UP APD is enabled during this time and the output current samples have been conditioned on the non-modified WMC_APD output current values.

CHAPTER 6
SIMULATION STUDIES

6.0 Introduction

The theoretical aspects of importance sampling as applied to the simulation of the M-PPM receiver were presented in Chapter 5. Chapter 5 also included a description of the design of two different M-PPM receiver importance sampling simulation systems. The first was a system in which the mean of the APD output density is shifted up during the M-PPM *OFF* slots. This will be referred to simply as "shift up importance sampling." The second was the M-PPM conditional importance sampling (CIS) system. CIS has been partially implemented, lacking only an analytical solution for the (constant) weight of each PPM word.

In this chapter, results and observations are presented which make use of the theory from Chapter 5. Results are presented from the simulation of a BPPM system using shift up importance sampling. As part of the presentation, a procedure is provided for performing importance sampling simulation of the M-PPM receiver. The results demonstrate that the theory of importance sampling developed for the M-PPM receiver is accurate. Shift up importance sampling has been used to produce BER curves down to the 10^{-12} BER level of error probability. A demonstration that the importance sampling estimator is unbiased is also provided. It is observed that the effect of increased memory of QPPM over BPPM diminishes the amount of improvement available from the procedure. Finally, we show that the CIS system does generate errors in the predicted and necessary fashion.

6.1 Shift Up M-PPM Importance Sampling

The M-PPM receiver system for the shift up importance sampling simulation of BPPM can be seen in BOSS block diagram form in Fig. 5.9. The shift up importance sampling system for QPPM is very similar. In the QPPM system, the input data source generates a QPPM waveform, naturally, and the decoding from QPPM to binary is more complicated than the decoding of BPPM to binary but otherwise the two block diagrams are the same.

At high values of bit error rate there is no need to use importance sampling in the generation of a BER versus input power curve. This is because the generation of errors is very probable when the BER is high and therefore very few bits need to be observed. Thus to begin

the production of a BER curve it is convenient to use classical Monte Carlo. Classical Monte Carlo is used until the BER becomes so low that the generation of data points begins to require the observation of a large number of bits so that the amount of time required becomes excessive. In the shift up systems, importance sampling can be effectively turned off by setting the value of the shift constant to zero. Alternatively, a copy of the importance sampling system can be generated within BOSS and the shift up APD replaced with an unmodified APD model. In this case the input to the weight counter would be made equal to a constant value of one. Using either technique, one is able to simulate system BER performance using classical Monte Carlo simulation with exactly the same system as used to perform IS simulation.

6.1.1 MC BER Results

BER curves generated using classical Monte Carlo simulation of a BPPM and a QPPM optical system are shown in Fig. 6.1. Each of these systems contains a specialized/simplified version of the BPPM and QPPM receivers. These receivers have no additive Gaussian noise source and there is no post APD filtering aside from the convolutional filter integrator of the type described in §5.3.2 of Chapter 5. This simplification eliminates the second noise source so that the issues surrounding importance sampling simulation of systems with multiple noise sources are circumvented. This also results in a system with effectively infinite bandwidth.

The lack of a thermal noise source in the system was found to be irrelevant to the resulting BER performance. This was confirmed by placing an additive Gaussian noise source after the APD with a noise power equivalent to that of a commercially available preamplifier. The BER performance of this system with a thermal noise source was the same as the system without the noise source. This is believed to indicate that the APD receiver is shot noise limited under the conditions of this simulation. Shot noise limited meaning that the error performance is dominated by the excess noise out of the APD so that the additive thermal noise is insignificant.

The error bars for each point in Fig. 6.1 are the one sigma standard deviation of the estimate. Recall that the standard deviation for Monte Carlo estimation is given by

$$\sigma_{MC} = \sqrt{\frac{P_e(P_e - 1)}{N}} \tag{6.1}$$

The system parameters used to produce the BER curves of Fig. 6.1, as well as many of the other results presented here, are displayed in the Table 6.1 below. These parameters were

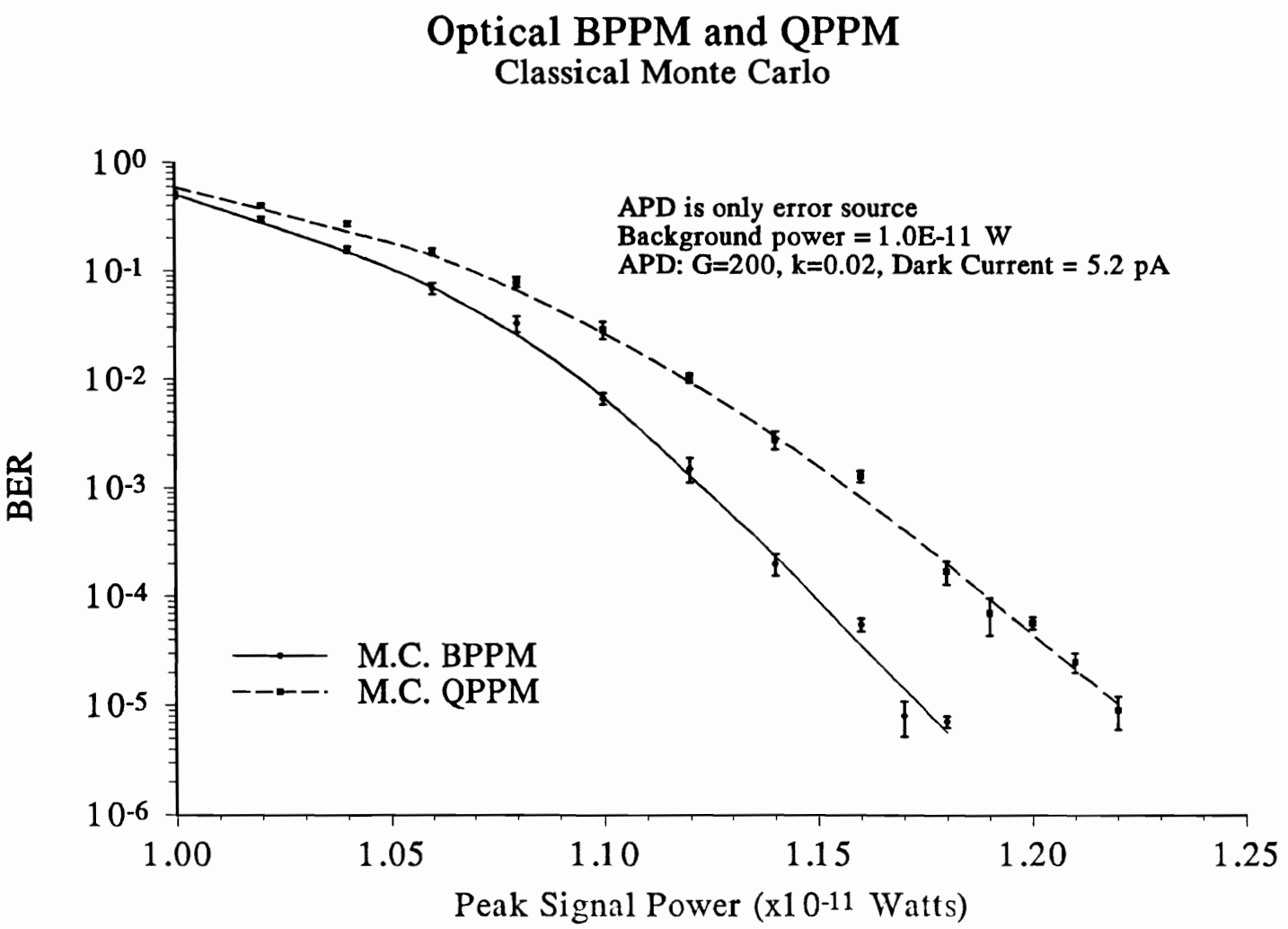


Fig. 6.1 BPPM and QPPM BER Curves from Classical Monte Carlo Simulation

used in the simulation of both the BPPM and QPPM systems. Note that the data bit rate is only 1 kbps. It will be shown that, as far as importance sampling is concerned, the actual bit rate used is irrelevant and that a more significant parameter is the number of samples per PPM slot. With a data bit rate of 1000 bits per second and dt, the sampling interval, at 5×10^{-5} seconds then the number of samples per PPM slot is 10 for both the BPPM and QPPM systems.

TABLE 6.1 Classical Monte Carlo Simulation Parameters

dt (sampling interval)	5.0E-5	seconds
data bit rate	1000	bits per second
signal power	variable	W
background power	1.0E-11	W
wavelength	0.81	microns
APD gain	200	ratio
quantum efficiency	0.75	ratio
κ (APD ionization ratio)	0.02	ratio
dark current	0.0052	nanoamperes
shift constant	0.0	number of electrons

The background power level appears at the APD input during the *OFF* slots and it was arbitrarily selected to be 1×10^{-11} W. The wavelength is a common semi-conductor laser wavelength. The quantum efficiency, the ionization ratio and the dark current were selected as typical of the RCA C30921E silicon avalanche photodiode. The average gain was set at 200 which is typical for free-space receiver applications. For Monte Carlo simulation the amount of the shift, i.e. the shift constant, is set to zero. Note that setting the amount of shift to zero will cause the shift up APD to automatically generate a weight value of one as indicative of Monte Carlo simulation.

With the importance sampling bias turned off for the BPPM and QPPM systems, BER data points were generated down to a level of approximately 10^{-6} in Fig. 6.1. This required the observation of more than 10^7 bits for some of the lower BER values, a process which

required several days to complete. As a result of this research, such time is no longer required but a Monte Carlo test was performed here in order to have data points generated using MC simulation for comparison to those generated by IS simulation.

6.1.2 IS BER Results

A BER curve for BPPM is shown in Fig. 6.2 which has points generated using both classical Monte Carlo and importance sampling. The points connected by a solid line are from classical Monte Carlo and this is the same BPPM BER curve presented in Fig. 6.1.

The remaining points, connected by a dashed line, show the results of importance sampling simulation using $N^* = 1000$ for all input power levels except those at $P_s = 1.25 \times 10^{-11}$ W where $N^* = 10^4$ was used. At each power level five simulations were performed using different bit streams and different seeds for the noise source random generators. The BER values plotted in Fig. 6.2 are the average of the five simulation runs. As can be seen, the importance sampling portion of the curve is in good agreement with the Monte Carlo portion of the curve. Further, the variation over each of the five importance sampling runs is minimal. This can be seen in Fig. 6.3 which shows the individual BER points used to construct Fig. 6.2.

The data points at $P_s = 1.25 \times 10^{-11}$ W are at the 10^{-12} level. To generate these points using classical Monte Carlo would require the observation of approximately 10^{13} bits. The time required for the Monte Carlo simulation of five million binary bits through this BPPM system required approximately 40 hours to complete when performed on a SUN SPARC II. (The simulation code is a FORTRAN program generated through the BOSS simulation package.) Extrapolating this rate to the simulation of 10^{13} bits, it is determined that it would require approximately 9000 years to simulate 10^{13} bits and thus obtain one data point on the BER versus input power curve. As mentioned, these points were generated using $N^* = 10^4$ bits which required less than one hour for the generation of the five points used in Fig. 6.2 and shown in Fig. 6.3.

6.1.3 Procedure

When generating the importance sampling results of Fig. 6.2 and Fig. 6.3, the following procedure was developed and used.

Recalling Fig. 4.6, it is observed that there is an optimum value for the importance

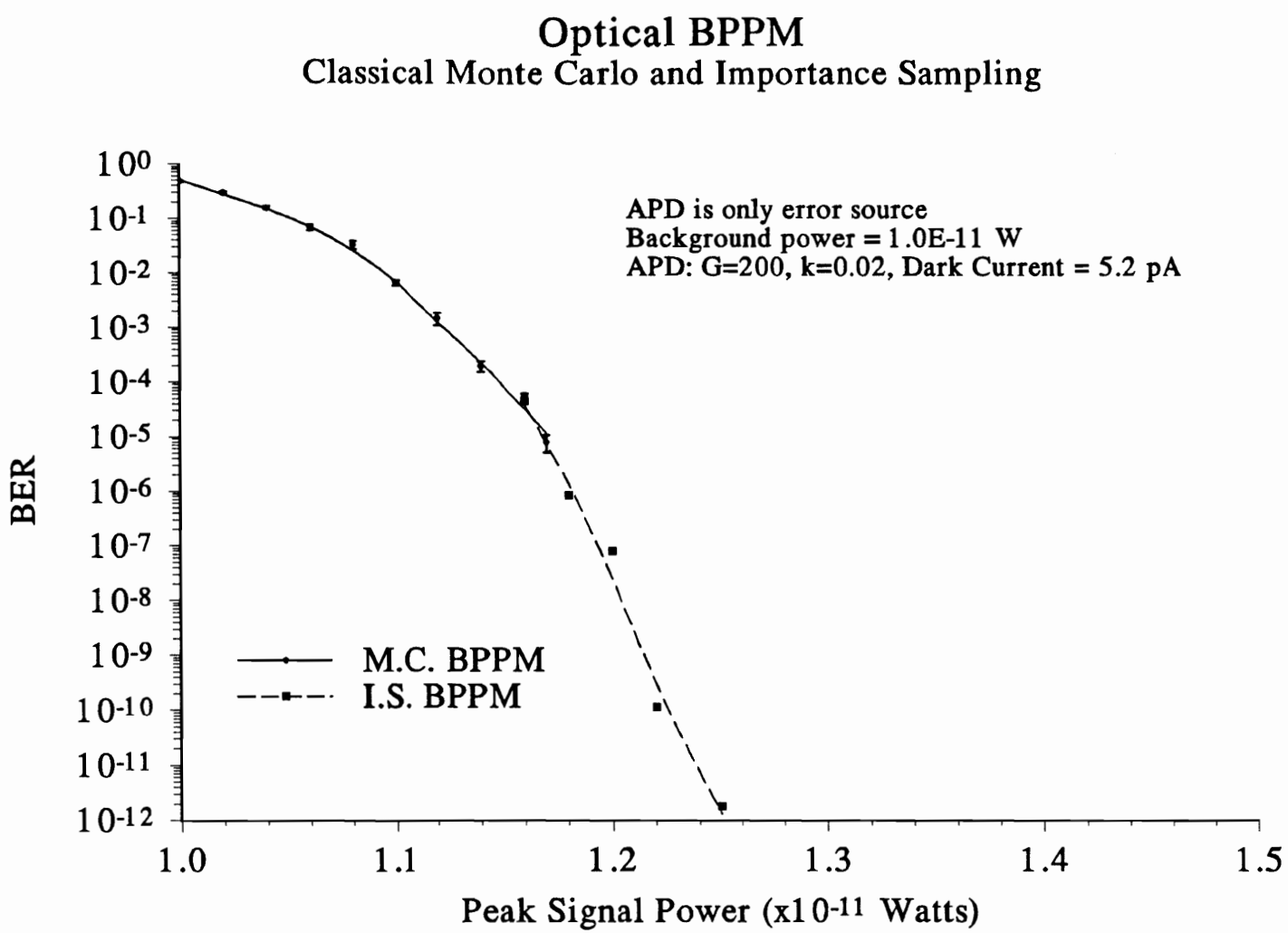


Fig. 6.2 BPPM BER Curve (averaged IS points)

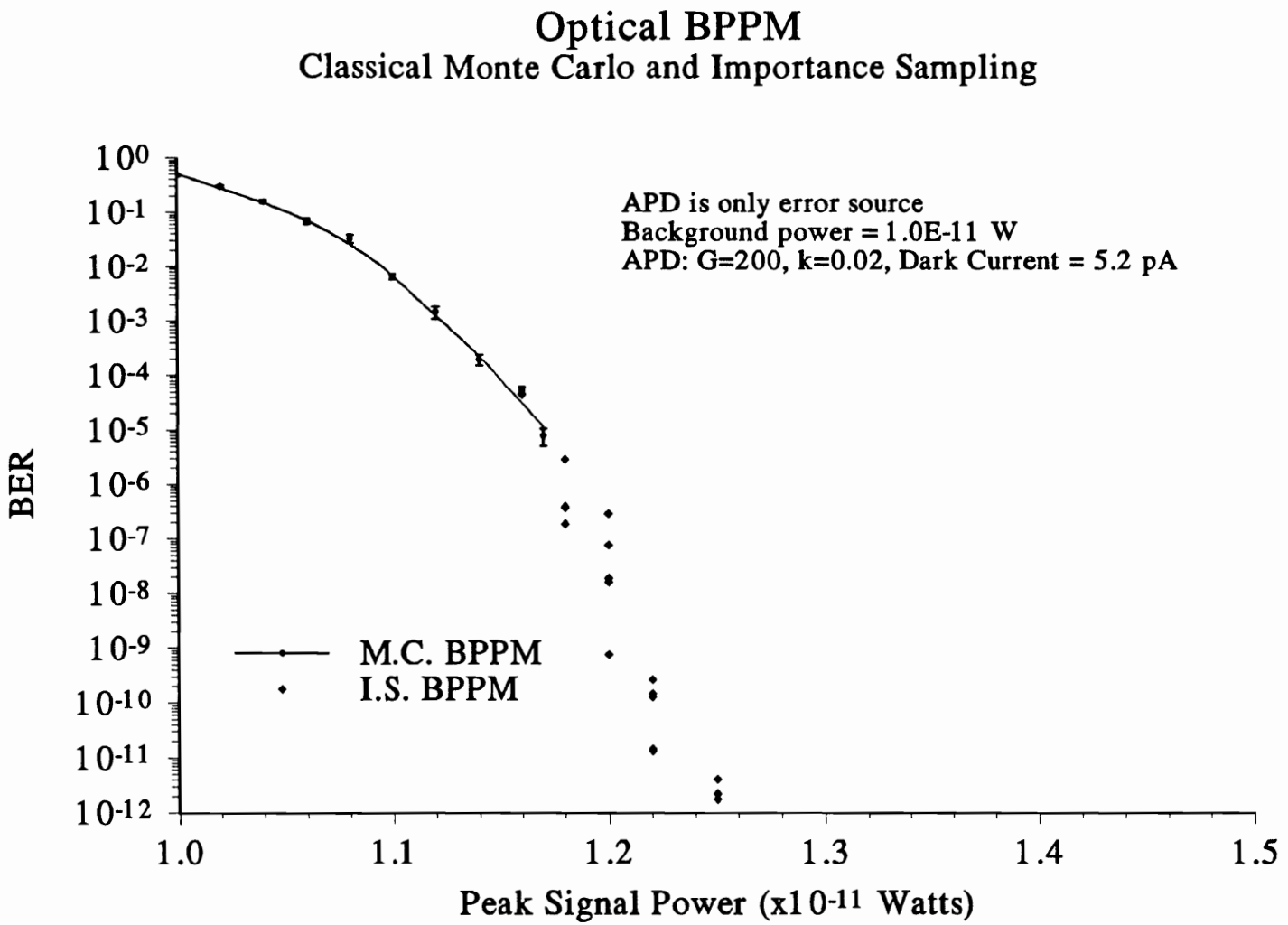


Fig. 6.3 BPPM BER Curve (five individual IS points)

sampling bias parameter. The bias parameter in the case of the shift up systems is the amount of shift and the optimum value is that shift which yields the largest possible reduction in the number of observed bits. The primary part of the procedure is to determine the appropriate amount of shift. In total three parameters must be determined, besides the selection of a suitable shift value, one must determine a suitable number of bits to be observed, N^*, and a value to use for the memory of the system when calculating the weight of a PPM symbol. All three of these parameters are interrelated and the procedure below will produce a selection for all three.

The memory of these simplified systems is well known due to the use of the convolutional filter. This was the reason for selecting it for use, so that system memory would be well controlled during the development of IS. Recall from §5.3.2 that the impulse response of the convolutional filter integrator is two slot periods long. The value of the impulse response is a constant 1.0 during the first half of the filter and a constant -0.05 during the second half, as displayed in Fig. 5.7. The use of this filter makes the system memory equal to that of a PPM symbol plus one more slot's worth of samples. As discussed in Chapter 5, the PPM decision is based on the entire PPM word regardless of the filter. In this case the trailing part of the convolutional filter causes the last slot of the previous PPM word to also affect the current PPM word. Because the convolution process adds one more sample at the end of the impulse response, the system memory is equal to the number of samples in a PPM word plus the number of samples in a slot plus one. Thus the BPPM system with ten samples per slot has a memory of 31 while the QPPM system with 10 samples per slot has a memory of 51.

Concerning the shift, one can safely shift too little and the results will be valid provided that enough bits have been observed. This is the primary penalty for using a shift amount smaller than the optimum shift value, that a larger number of bits must be observed in order to obtain valid results. That is, the value of N^* will need to be larger since the improvement ratio is smaller than that which is obtained if one uses a value of shift closer to the optimum. To see that a small shift is safe, consider that one need not shift at all. This is a default to the classical Monte Carlo simulation case and the improvement ratio becomes one.

It can be seen in Fig. 4.6 that shift amounts greater than the optimum are still valid for a finite range but incorrect results will be obtained for values of shift that are far greater than the optimum. As the shift is increased the maximum weight value decreases as will be observed in the plots which follow. It was pointed out in Chapter 4 that if the maximum weight value used in the weighted error summation becomes less than the true BER value then the results

will be wrong. That the maximum weight value will decrease with increasing shift is a property that was proven for any PDF satisfying some very mild conditions by Devetsikiotis and Townsend[1]. The technique developed here is an enhanced version of the method proposed in [1] using this property.

The optimum shift value will be different for each input power level. In order to determine the best shift value, the input signal power is held constant and several simulations are performed starting with a small shift and then incrementing the amount of shift to larger and larger values. At the beginning, a value for N^* needs to be assumed as well as a system memory value. The N^* assumption is based on a rough estimate of the BER level to be determined and an estimate of the improvement ratio to be expected. This assumption is based partly on experience but also on observation of the results from the MC simulations or the previous IS simulation. For this reason, it is recommended to pursue BER information from high levels of BER to low levels of BER.

That the N^* and system memory choices are valid can be confirmed, as described below, through observation of the following quantities. At each step in the shift five items are recorded at the end of the simulation: 1) the maximum weight, 2) the minimum weight, 3) the estimated average weight, 4) the number of errors observed and 5) the BER. These quantities are then plotted versus the amount of shift as shown in the example of Fig. 6.4. Searching for the trends described below on a plot similar to Fig. 6.4 will indicate the optimum shift, the validity of the choice for N^* and the validity of the value used for the system memory.

A plot similar to Fig. 6.4 needs to be generated for each signal power level on the curve where it is desired to determine the BER level. The plot in Fig. 6.4 was used in the generation of the BER plot of Fig. 6.2 for the BER data point at a signal power of 1.18×10^{-11} watts. Although busy, a single plot of all the quantities is instructive since a judgment must be made based on the magnitudes of each quantity relative to the others. In other words, it is necessary to consider several items at once when determining the validity of the IS simulation parameters. Therefore, it is ultimately more convenient to observe the plots on one graph. The busy nature of Fig. 6.4 makes it a somewhat inconvenient vehicle for explanation. Therefore, individual components of Fig. 6.4 are reproduced and displayed separately for use in the following discussion.

There is a value of shift that must be obtained before errors begin to occur. As the shift is increased the number of errors continues to increase. This is shown in Fig. 6.5 where the

number of errors has been normalized by dividing by the total number of bits (1000 in this case). The BER versus shift curve also appears in Fig. 6.5 for reference. The number of errors generated relative to the total number of bits observed is an important consideration. Just as with classical Monte Carlo, the variance of the estimate of the BER is smaller with an increasing number of errors observed. Although the variance of the IS estimator, eq.(5.73), does not appear to depend directly on the number of errors generated, the weight function W(**r**) does. As W(**r**) becomes smaller the variance of the estimator becomes smaller. We have already seen that the weight value generally becomes smaller with increasing shift and that the number of errors becomes greater with increasing shift so that one might infer that the two are correlated. Further, the lower bound to the variance of the estimator becomes smaller with increasing numbers of errors as seen in eq.(4.52). Similarly, the upper bound to the improvement ratio becomes larger with increasing numbers of errors generated as per eq.(4.53). Finally, our estimate of the average weight becomes better with an increased number of errors as seen in eq.(5.133). Thus, we have strong indication that producing a high percentage of errors is desirable.

Fig. 6.6 is a plot of the minimum weight, the maximum weight and the measured BER as a function of the amount of shift. As the shift is increased, the minimum weight value observed during the course of the simulation run begins to fall rapidly. The maximum weight value maintains a somewhat steady value until at a certain point it begins to drop rapidly with increasing shift. This drop indicates the shift is becoming too large and occurs at a shift values greater than approximately 40000 in Fig. 6.6 or equivalently Fig. 6.4. It is observed that the BER maintains a similarly flat or relatively constant value until it also begins to drop. The existence of an identifiable flat region in the BER curve indicates that the N^* value is large enough and that the memory value used is sufficient. As seen in Fig. 6.6, the BER curve also begins to drop at shift values greater than 40000. One last point, as the maximum weight falls it becomes, at some point, less than the average value of the BER through the flat region. When the maximum weight becomes less than the BER through the flat region, the shift is too great as per the theorem developed in [1]. Thus the optimum shift must be somewhere less than 40000 or 45000 or so in Fig. 6.4.

Note also the point at which that minimum weight becomes less than the measured BER level. They cross at a shift of approximately 23000 in Fig. 6.6. After observing various plots of this type it has been noted that this point, where the BER and the minimum weight cross, always occurs at low values of the shift in the region where the BER curve is generally

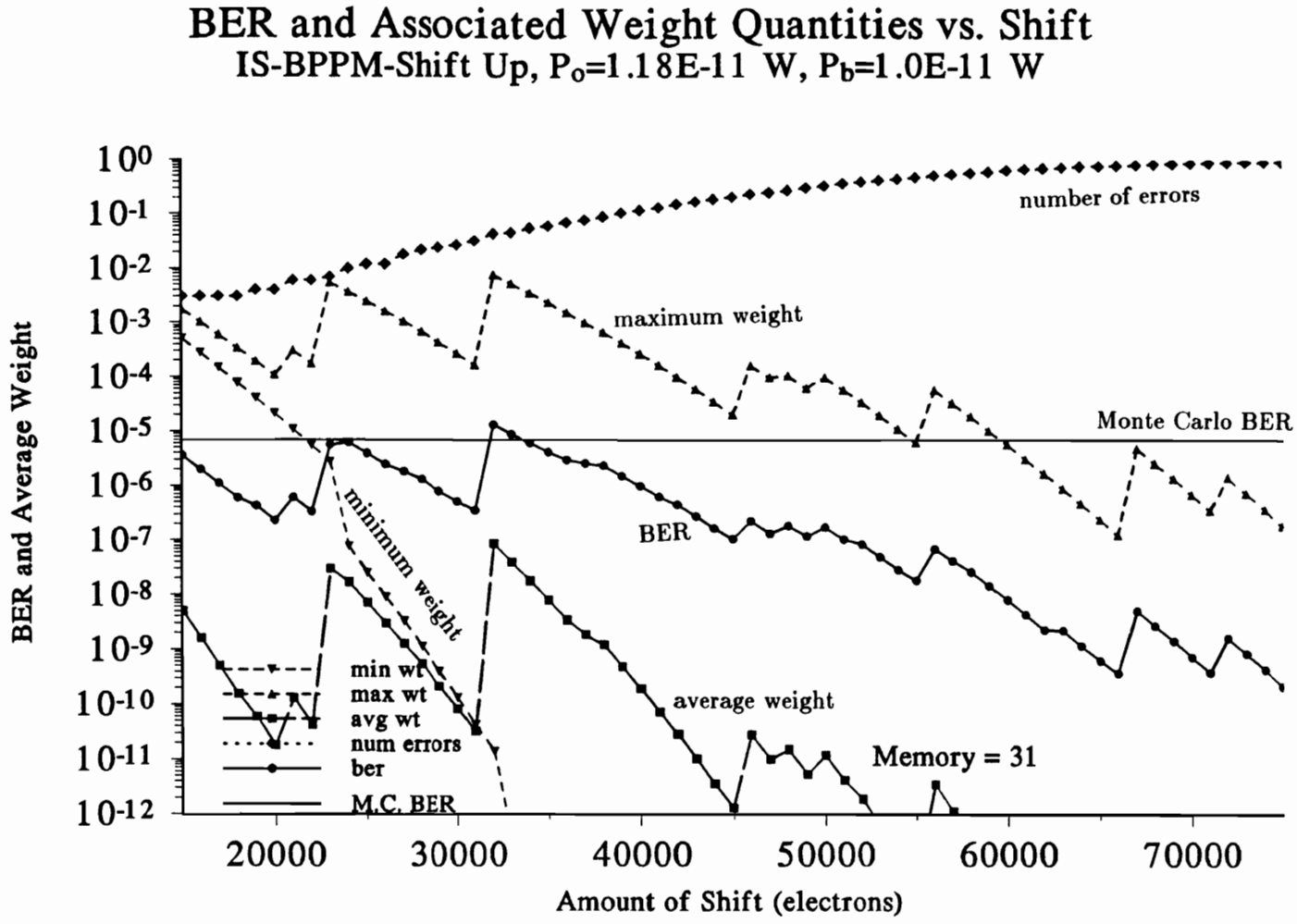


Fig. 6.4 BPPM BER and Associated Weight Quantities versus Shift, P = 11.8 pW

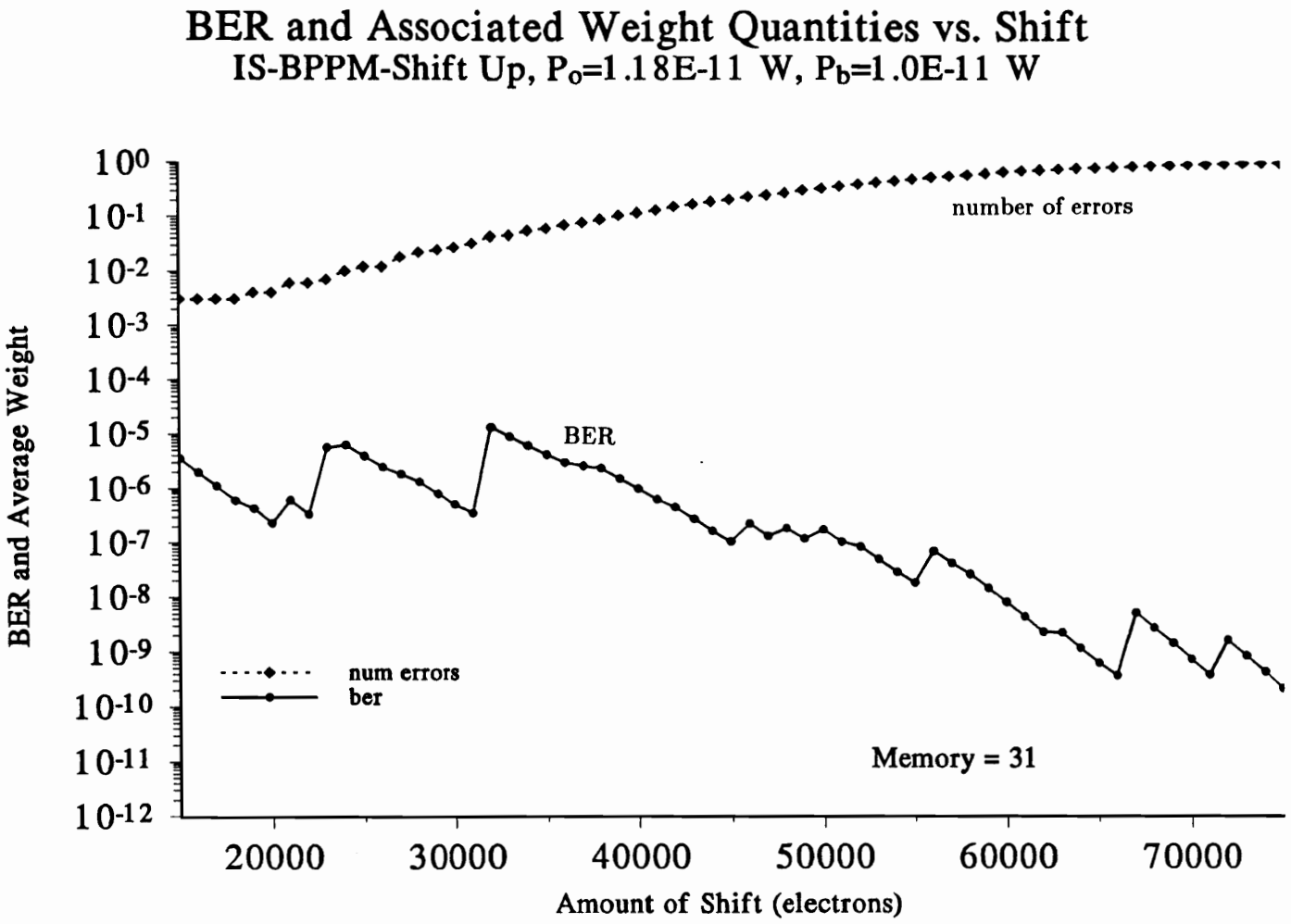


Fig. 6.5 BPPM BER and Number of Errors Generated versus Shift, P = 11.8 pW

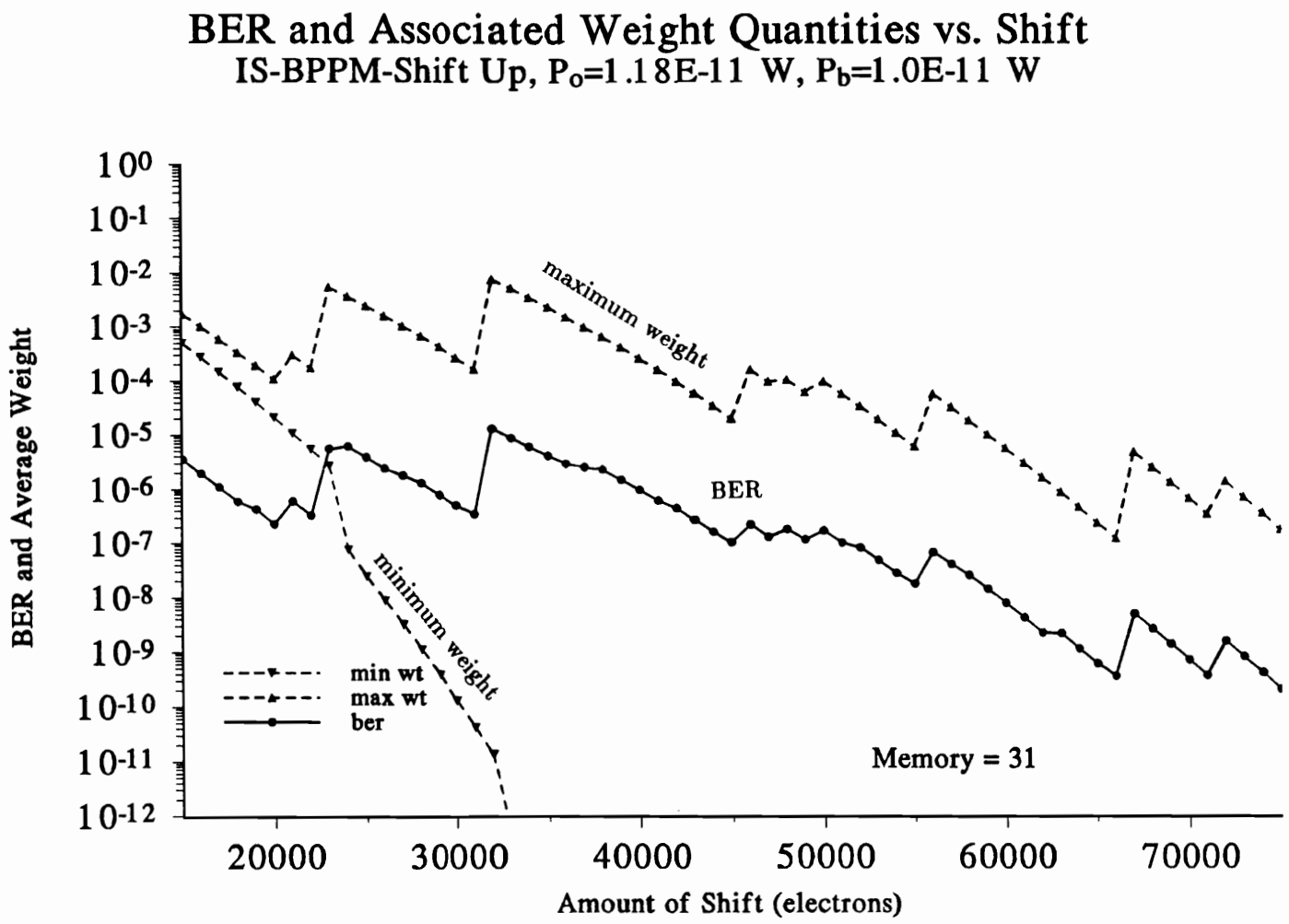


Fig. 6.6 BPPM BER, Minimum Weight, and Maximum Weight versus Shift, P = 11.8 pW

flat. Also note that the average weight becomes greater than the minimum weight. This can be observed in Fig. 6.4 where it is seen that the average weight and the minimum weight cross at about the 30000 shift value. These trends concerning the minimum weight and the average weight have been observed repeatedly over many importance sampling trials when the parameters are correct.

It becomes plain that there is currently no firm method to determine the optimum shift to use. Yet the example of Fig. 6.4 illustrates all of the necessary features. If the number of observed bits, N^*, is large enough and the memory used in the calculation of the weight is sufficient for the system memory then the generally flat region in the BER curve of Fig. 6.4, and equivalently Fig. 6.5 or Fig. 6.6, will be observed. The flat region in the BER curve provides perhaps the strongest clue but needs to be considered along with the general trend of the maximum weight (Does it have a flat region as well before falling off?), and with the number of errors (Has there been a significant number of errors produced?) and also considering the status of the average weight. (Is the average weight generally between the maximum weight and the minimum weight?) From the plot of the weight quantities of Fig. 6.4 it can be seen that there is a generally flat region of the BER curve which exists between values of shift from 15000 to about 40000 to 45000 or so.

The BER value measured at the input power of 1.18×10^{-11} W was determined using classical Monte Carlo and has been plotted in Fig. 6.4 also. It is the solid line at a BER level of 7×10^{-6}. This information is not available, of course, when determining the optimum shift but was included in Fig. 6.4 to provide some reference for the concepts above.

For additional examples, all of the BER versus shift plots used in the generation of the IS BER points of Fig. 6.2 are shown in Fig. 6.7 through Fig. 6.10. Fig. 6.7 is the shift plot for a constant power level of 1.16×10^{-11} W. It also includes a line at the MC determined BER level of 5.5×10^{-5} for reference. Note that the generally flat region of the BER and maximum weight versus shift curves extends over a large range of the shift parameter. This attests to the large improvement ratio at this BER level. Fig. 6.8 is a shift plot at the 1.2×10^{-11} W power level. Note that there no MC derived BER line since Monte Carlo simulations at this BER level would require days to produce. Note also that the generally flat region of the BER and maximum weight versus shift curves is followed by the region of decreasing values. From Fig. 6.8 it is seen that valid shift values would be less than approximately 50 000. Fig. 6.9 is a shift plot for the constant power level of 1.22×10^{-11} W. The same classic features can be observed in this plot as well.

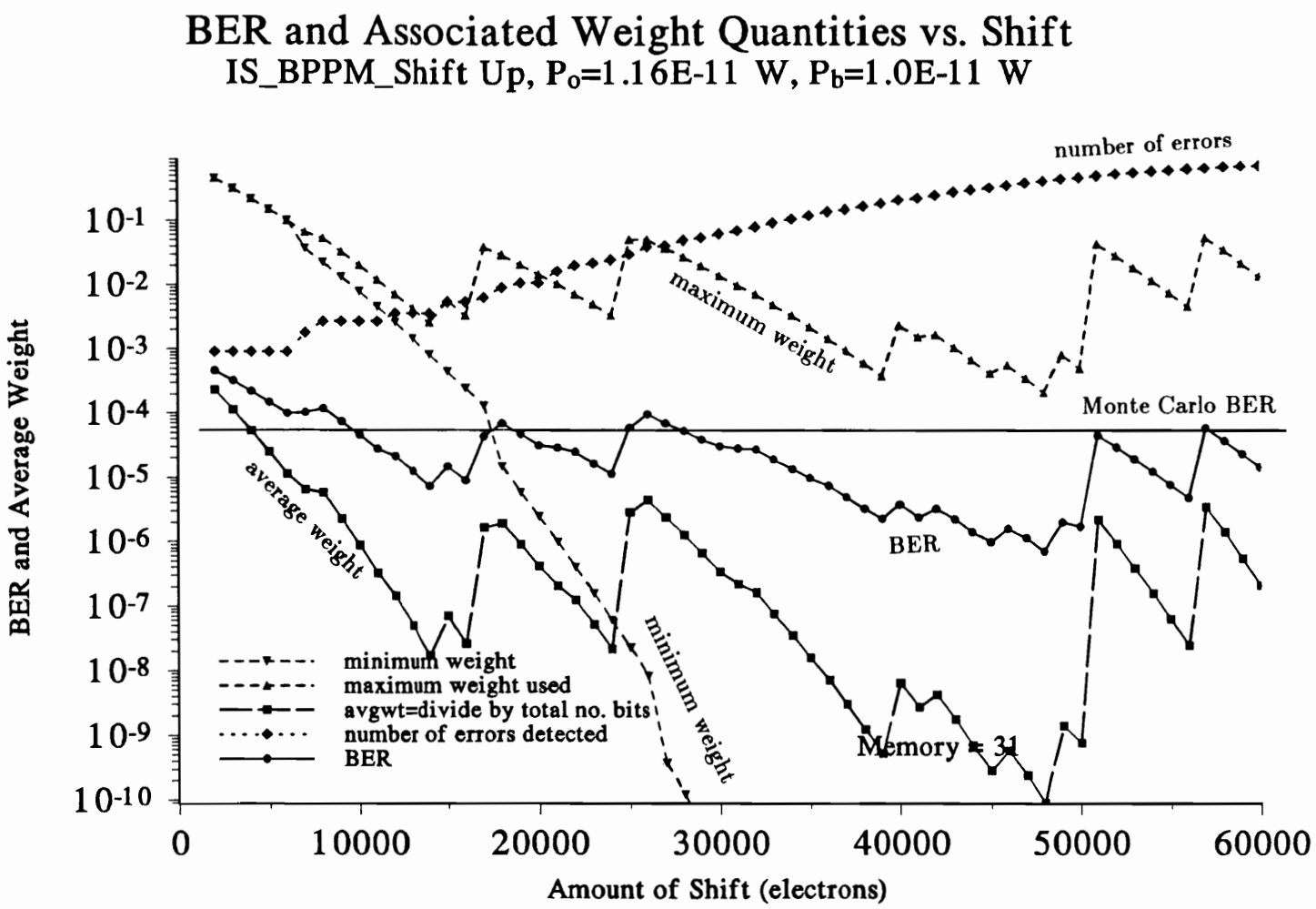


Fig. 6.7 BPPM BER and Associated Weight Quantities versus Shift, P = 11.6 pW

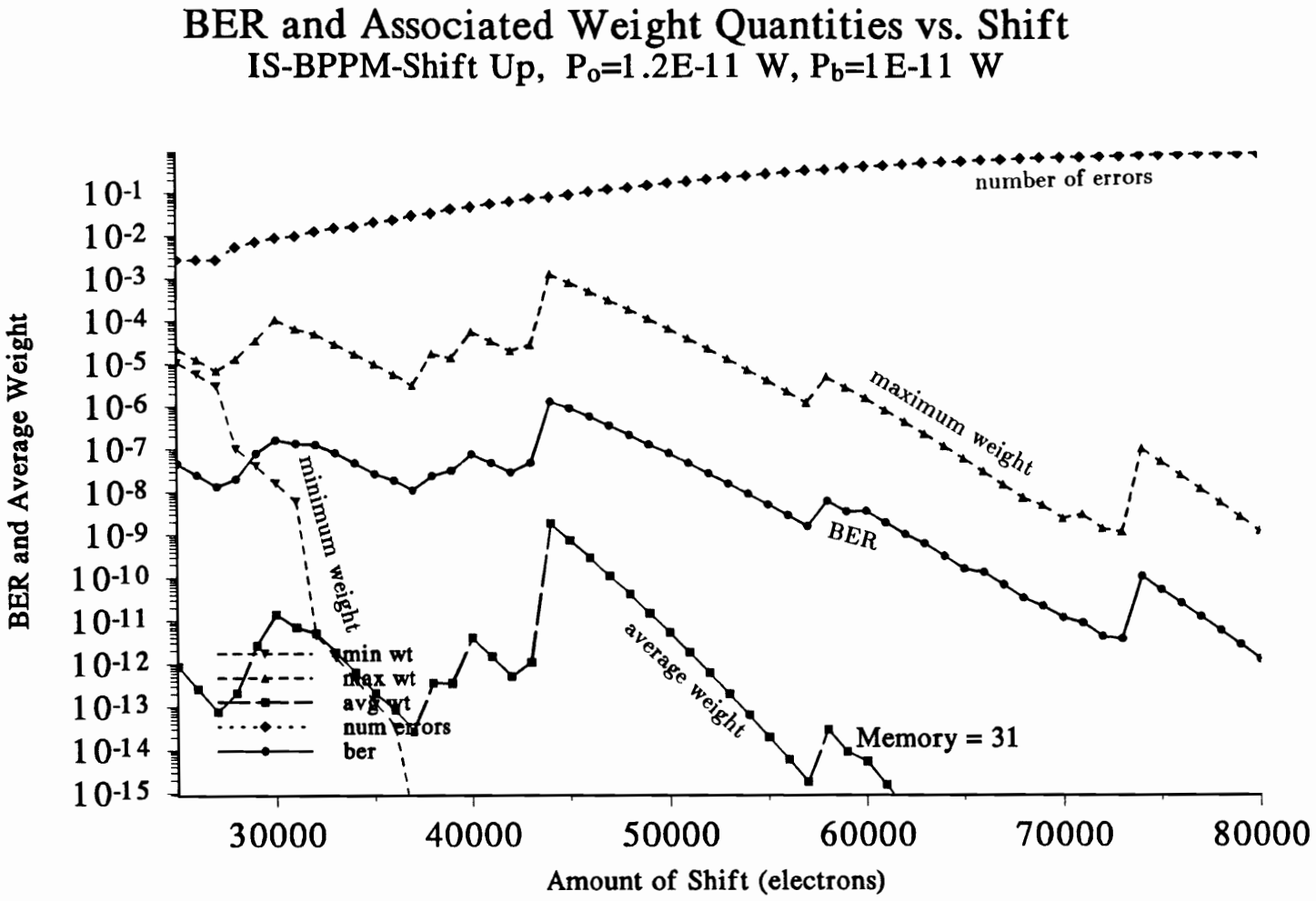


Fig. 6.8 BPPM BER and Associated Weight Quantities versus Shift, P = 12.0 pW

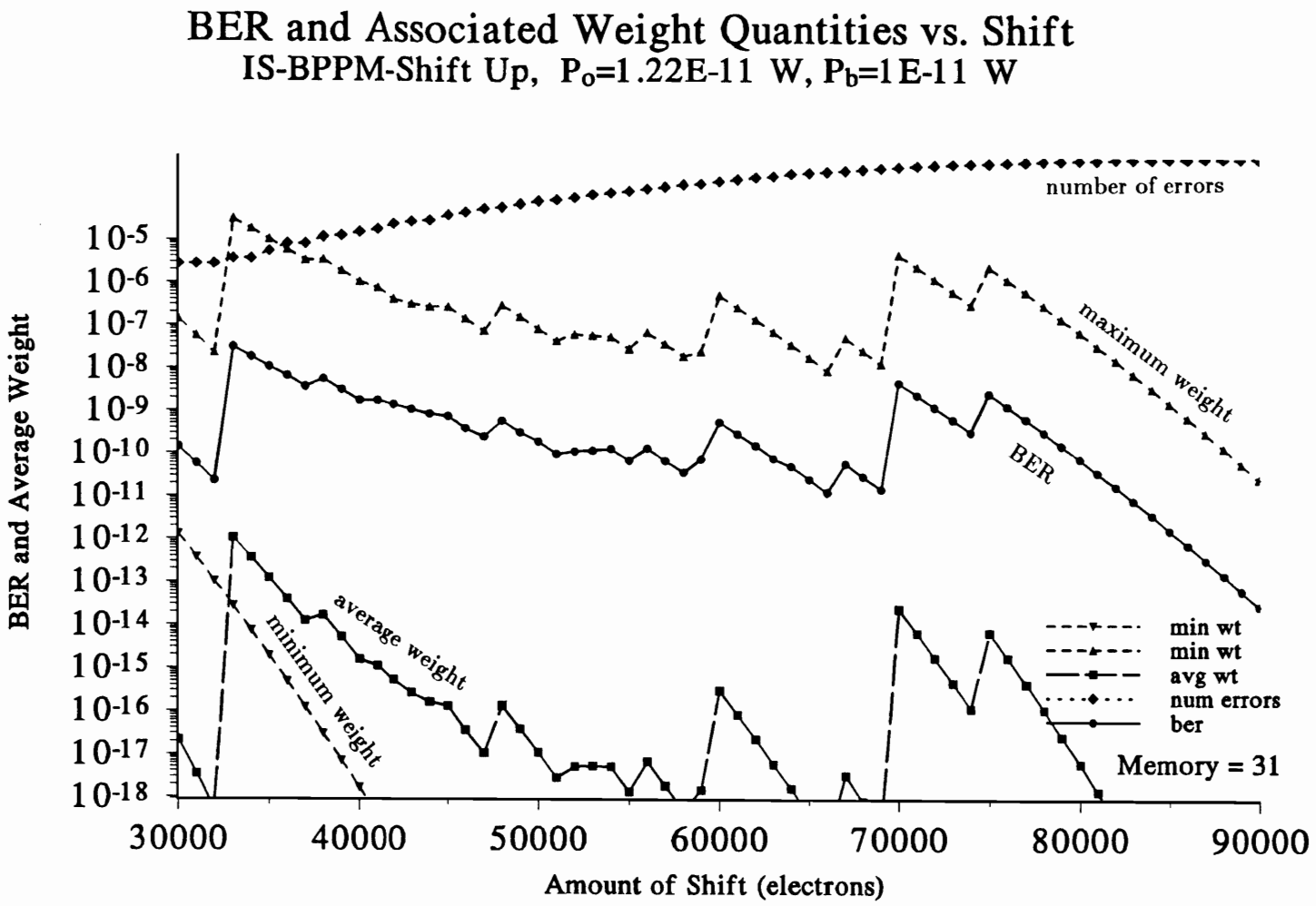


Fig. 6.9 BPPM BER and Associated Weight Quantities versus Shift, P = 12.2 pW

In Fig. 6.10 is shown the relative number of errors, the BER for $N^* = 1000$, and the BER for $N^* = 10K$ versus shift for the signal power level of 1.25×10^{-11} W. The other weight parameters have been eliminated from this plot for clarity. Consider that the flat region of of the BER curve for $N^* = 1000$ is relatively small and not as clearly defined as in the previous plots. To confirm the valid range of shift parameter, the sequence of simulation runs was repeated at $N^* = 10k$ for shift values in the range of 50 000 to 80 000. It is seen that the stable region of the BER curve is in the suspected range. Further, it is seen that increasing N^* brings the stable region of BER curve into focus and thus indicates that the N^* needs to be increased from the $N^* = 1000$ value.

6.1.4 A Counter Example

Fig. 6.4 presents an example of a valid shift plot when all of the IS parameters are valid. Let us consider a counter example. When importance sampling was applied to the QPPM system using the parameters of Table 6.1, all was not as it might be. Fig. 6.11 shows the result of applying the technique above to the IS simulation of the QPPM system whose classical Monte Carlo BER performance results were shown in Fig. 6.1. The number of bits observed at each value of shift was 5000 as compared to the $N^* = 1000$ used for most of the BER points in the BPPM system. The memory of the QPPM system using the parameters of Table 6.1 is 51. Despite the increase in N^*, the BER and the maximum weight are seen to fall continuously from very small values of shift onwards. The number of errors at these shift values is small, less than 10 out of 5000 bits, indicating that N^* will have to increase in order to produce valid results.

In other words, importance sampling is not very efficient in this case. For large values of system memory, importance sampling becomes less efficient at reducing the number of bits that need to be observed. This can be seen in Fig. 4.6 where it is observed that for large memory values, the curves do not reach large values of improvement ratio. The memory of the QPPM system is dominated by the number of samples per slot, which in turn is set by the parameters of *dt* and bit rate as specified in Table 6.1.

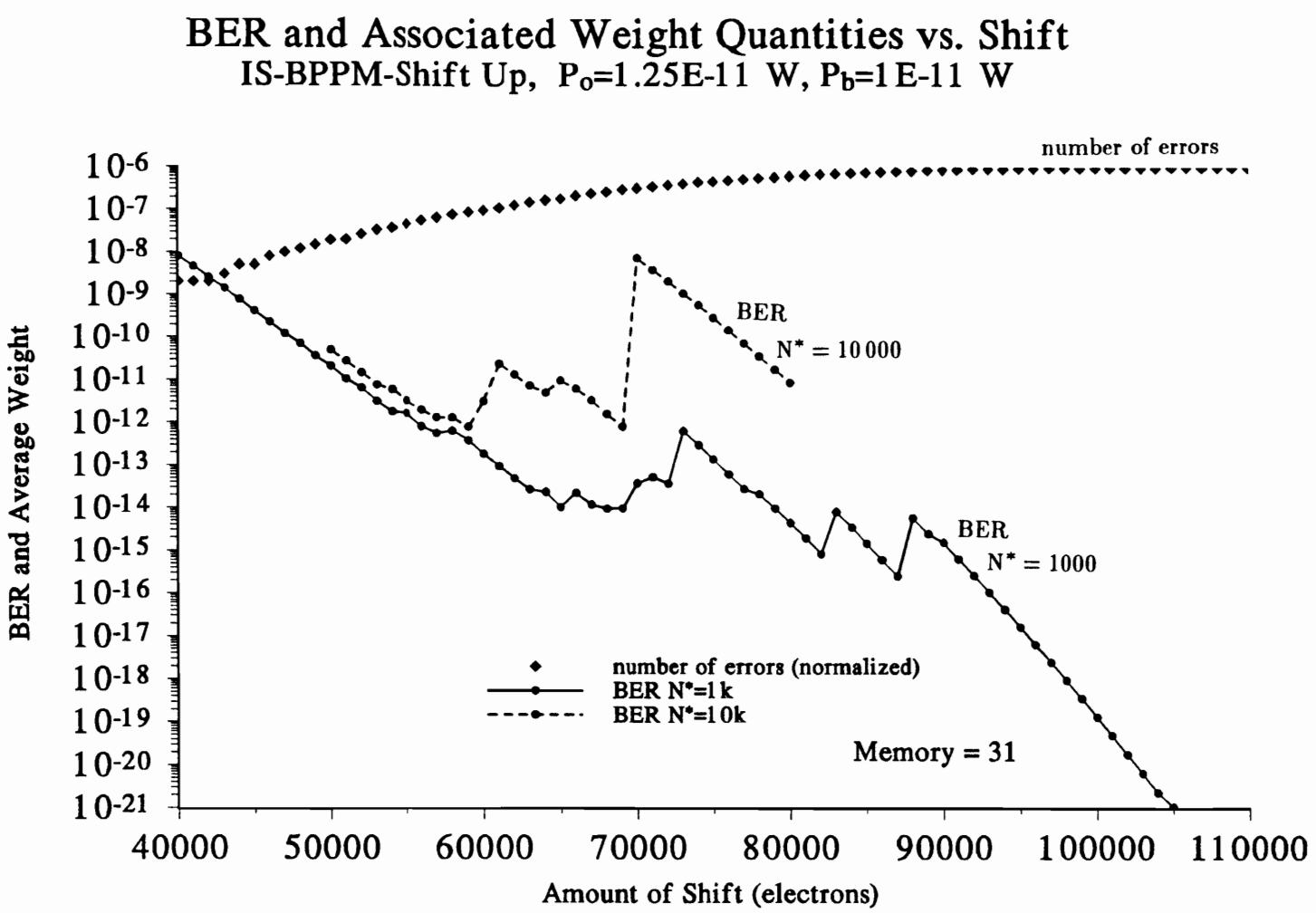


Fig. 6.10 BPPM BER and Number of Errors versus Shift, P = 12.5 pW

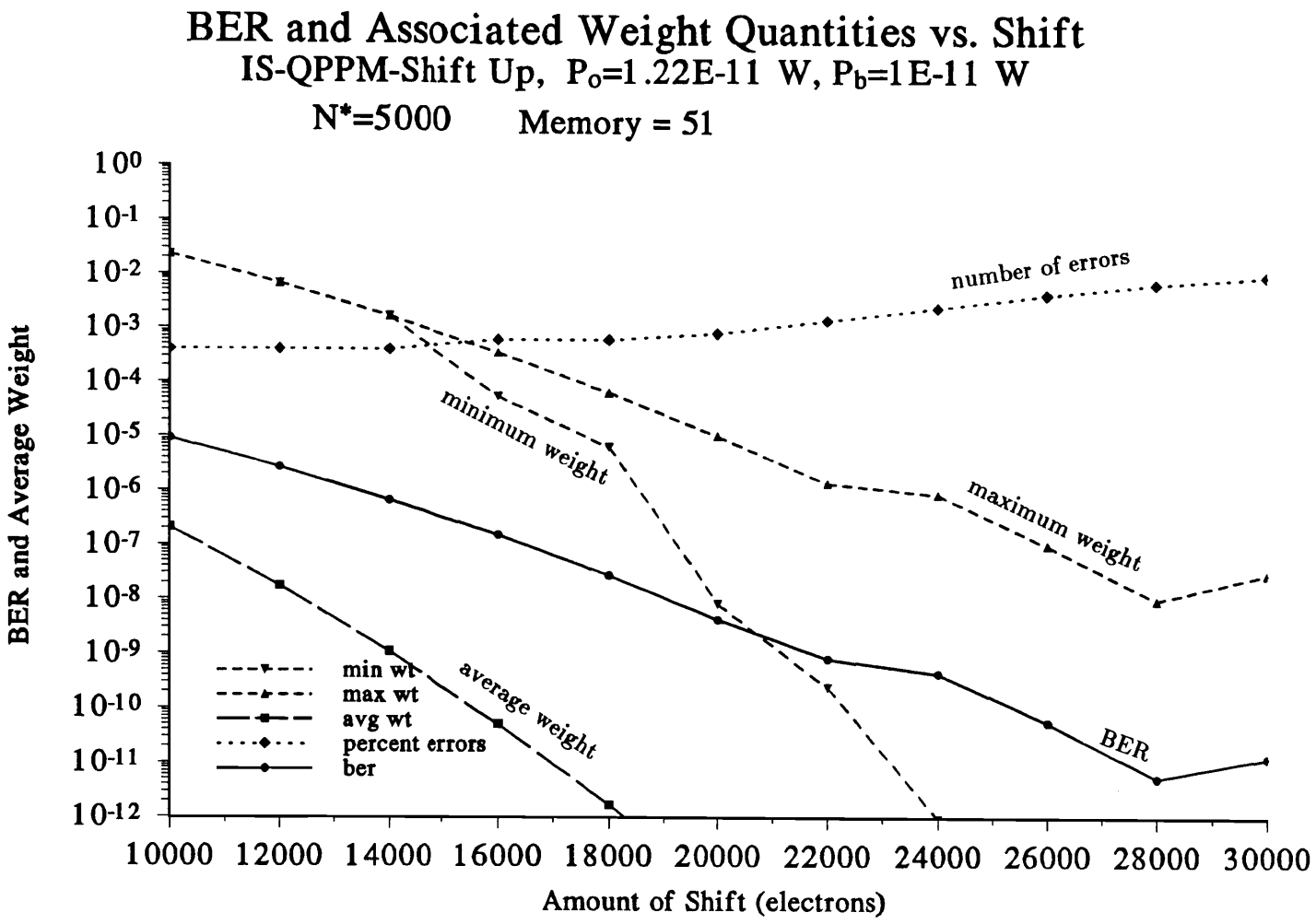


Fig. 6.11 QPPM BER and Associated Weight Quantities vs. Amount of Shift, Large System Memory (10 samples per slot)

6.2 Observations

6.2.1 Estimation Error Bounds

After performing the IS simulation it is important to be able to generate a confidence interval for the estimate produced. In Chapter 4, upper and lower bounds to the importance sampling variance were provided from the literature[2][3]. Recalling eq.(4.51) through eq.(4.54), the variance has been shown to be upper and lower bounded as

$$\sigma_{IS}^2 \leq \sigma_{UB}^2 = \frac{P_e}{N^*}\left[\max\left[W(r)\right] - P_e\right] \tag{6.2}$$

$$\sigma_{IS}^2 \geq \sigma_{LB}^2 = \frac{P_e^2}{N^*}\left[\frac{1}{P_1(r)} - 1\right] \tag{6.3}$$

where $P_1(r)$ can be estimated as $\hat{P}_1(r) \equiv$ (number of errors)/N^*.

The improvement ratio $R = \frac{N}{N^*}$ can be upper and lower bounded by

$$R \leq R_{UB} = \frac{P_1(r)\left[1 - P_e\right]}{P_e\left[1 - P_1(r)\right]} \tag{6.4}$$

$$R \geq R_{LB} = \frac{\left[1 - P_e\right]}{\left[\max\left[W(r)\right] - P_e\right]} \approx \frac{1}{\left[\max\left[W(r)\right] - P_e\right]} \tag{6.5}$$

These bounds can be used to estimate the simulation variance and improvement ratio of the M-PPM IS simulation.

The estimate of the average weight that is taken during the simulation can also be used to provide an estimate of the variance of the estimate. Recalling eq.(5.125)

$$\hat{\sigma}_{IS}^2 = \frac{1}{N^*}\left[\hat{\overline{W}} - P_e^2\right] \tag{6.6}$$

This expression can be used with the variance expression for the classical Monte Carlo estimate, see eq.(6.1), to form an improvement ratio estimate based on the estimated average weight.

$$\hat{R} \equiv \frac{N}{N^*} = \frac{P_e(1 - P_e)}{\hat{\overline{W}} - P_e} \tag{6.7}$$

As an example of the utility of these expressions the following example is offered. The

upper and lower bounds to the variance and the standard deviation will be calculated using the values produced by a representative importance sampling simulation. The simulation used will be one of the simulation examples used in the plot of the BER based on IS generation of 30% errors. The following output quantities were produced during the IS simulation of the BPPM system when the input signal power was 1.18×10^{-11} W:

$$\text{number of bits} = 1000$$
$$\text{number of errors} = 316$$
$$\text{maximum weight observed} = 1.894 \times 10^{-4}$$
$$\text{average weight observed} = 5.1232 \times 10^{-11}$$
$$\text{P} = 3.9147 \times 10^{-7}$$

Using these values in eq(6.2) through eq.(6.7), the following bounds and estimates are obtained:

$$3.137 \times 10^{-16} = \sigma^2_{LB} < \sigma^2_{IS} < \sigma^2_{UB} = 7.4 \times 10^{-14}$$

which leads to bounds on the standard deviation of

$$1.8213 \times 10^{-8} = \sigma_{LB} < \sigma_{IS} < \sigma_{UB} = 2.72 \times 10^{-7}$$

for BER of 3.9147×10^{-7}

Using the average weight the variance and the standard deviation are estimated to be

$$\sigma^2_{IS} = 5.1078 \times 10^{-14}$$

$$\sigma_{IS} = 2.26 \times 10^{-7}$$

which is seen to fall between the upper and lower bounds of the variance calculated above.

Similarly, the improvement ratio is bounded as

$$5.29 \times 10^3 = \text{R}_{LB} < \text{R} < \text{R}_{UB} = 1.8 \times 10^6$$

and the improvement ratio obtained using the estimated average weight is

$$\hat{\text{R}} \equiv \frac{\text{N}}{\text{N}^*} = 7.664 \times 10^3$$

which is between the upper and lower bounds of the improvement ratio.

Thus the quality of the IS estimate can be determined.

6.2.2 An Unbiased Estimator

In order to confirm that the importance sampling estimator is truly an unbiased

estimator, a simple experiment was performed. Using the parameters as per Table 6.1, the BER value of the BPPM system at a signal power of 1.16×10^{-11} W was estimated using importance sampling and using Monte Carlo simulation. A large number of bits were observed in each case, i.e. both N and N* were large. The large number of samples in the Monte Carlo simulation was used in order to provide a BER data point with great accuracy. The BPPM BER at 1.16×10^{-11} W was determined using MC to be 5.06×10^{-5} with a standard deviation of 2.123×10^{-5}. This result is the average of five simulation runs, one with $N = 10^6$ and the other four with $N = 5.0 \times 10^5$.

An importance sampling simulation of the same BPPM system was performed using a shift value of 39000, corresponding to approximately 20% error generation, and $N^* = 5.0 \times 10^6$. The BER using this IS simulation was 4.562×10^{-5}, well within experimental error and in very good agreement with the Monte Carlo simulation result. The standard deviation of the IS result was determined through use of the estimated average weight to be 1.321×10^{-5}. The average weight is exceedingly reliable in this case since it was estimated based on the observation of over 1 million errors. Both of these points are plotted at the 1.16×10^{-11} W power level in Fig. 6.2 and Fig. 6.3. It is seen that they are almost indistinguishable from each other.

If the IS estimator were a biased estimator it should have become evident after the simulation of five million bits: therefore it is not.

6.2.3 Percent Number of Errors

From the discussion of the procedure to determine the amount of shift to use, it is seen that there is a large range of shift values which might be used. It has been observed empirically that shift values in the range where the minimum weight crosses the measured BER are safe and have been used. This region corresponds to the generation of about 100 errors out of 1000 bits for the system of Fig. 6.4, that is about 10% of the bits are in error. The guidelines given above for the selection of the amount of shift included consideration the percent number of bits in error as well as observations of the slope of the BER versus shift curve, the maximum weight versus shift curve and the relative values of the average weight and minimum weight to the measured BER. All of these together indicate suitable values of shift to use. To specify that the generation of 10% errors as the only criteria may be the quickest indicator of a suitable shift but will not always be accurate by itself. Consider Fig. 6.11 as a counter example, at the shift where 10% errors are made to occur the BER value is far from correct. One must also observe the trend of the maximum weight versus the amount of shift, for example, to make sure that

one is not operating in a region where the maximum weight is falling.

The question remains concerning what effect the percent number of errors has on the BER results. To test this, the BPPM system was simulated using importance sampling at five different shift values. The shift values used caused 10%, 30% and 50% of the bits observed to be in error during the simulation at each signal power. The results of this experiment were plotted together and shown on the BER plot of Fig. 6.12. As can be seen there is no perceived difference in using 10% versus 30% versus 50% generated errors. The BER curves of Fig. 6.2 and Fig. 6.3 were produced using the 30% generated error BER values shown in Fig. 6.12 except for the point at $P = 1.16 \times 10^{-11}$ W which was at a 20% error rate shift.

6.2.4 CIS Error Generation

In order to determine the validity of the CIS M-PPM importance sampling concept, the system was built and simulated. As there is not yet a valid weight constant to use in the simulation, one can only observe the generation of errors. Fig. 6.13 is a plot of number of errors produced using conditional importance sampling versus the amount of offset. As predicted by the theory of Chapter 5, as the amount of offset goes to zero the number of errors generated goes to 100%.

6.3 Summary

The application of importance sampling to the M-PPM receiver has been demonstrated. It has provided an ability to characterize the BER performance of optical receiver systems to very low probability of error levels, levels that were heretofore unobtainable. A procedure for performing importance sampling simulation of these receivers has been specified which includes an ability to specify the accuracy of the BER estimate through bounds on the estimation variance. Issues concerning importance sampling have been explored. For example, it has been demonstrated that the importance sampling estimation system used here is unbiased. The choice of bias point, referring here to the amount of shift, has been explored. It was demonstrated that accurate results are obtained over a range of shift values which result in the generation of different percentage amounts of errors during the same simulation. Finally, the validity of the CIS technique to generate errors in a controlled fashion has been demonstrated.

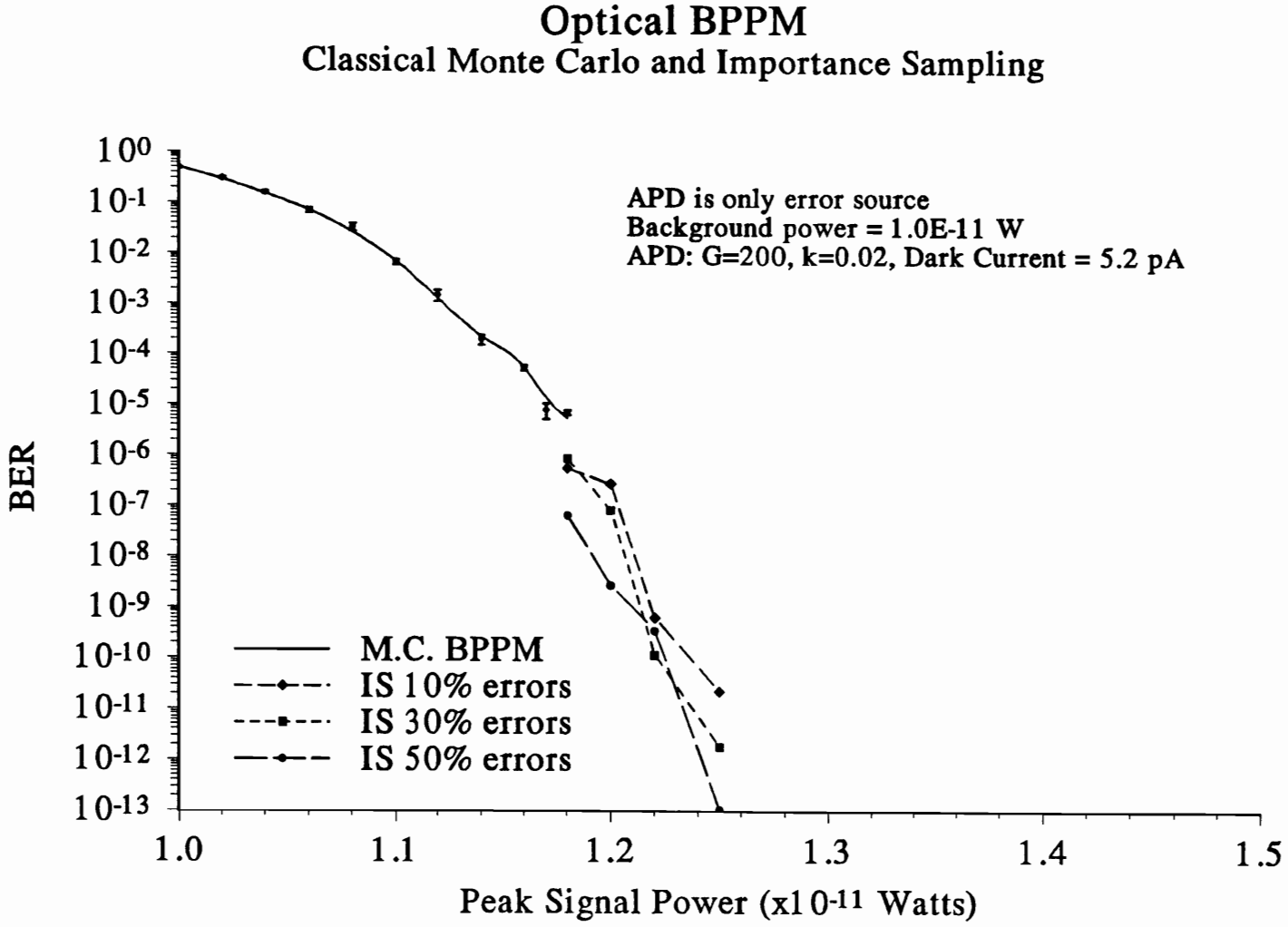


Fig. 6.12 BPPM BER Curves Using 10%, 30% and 50% Generated Errors

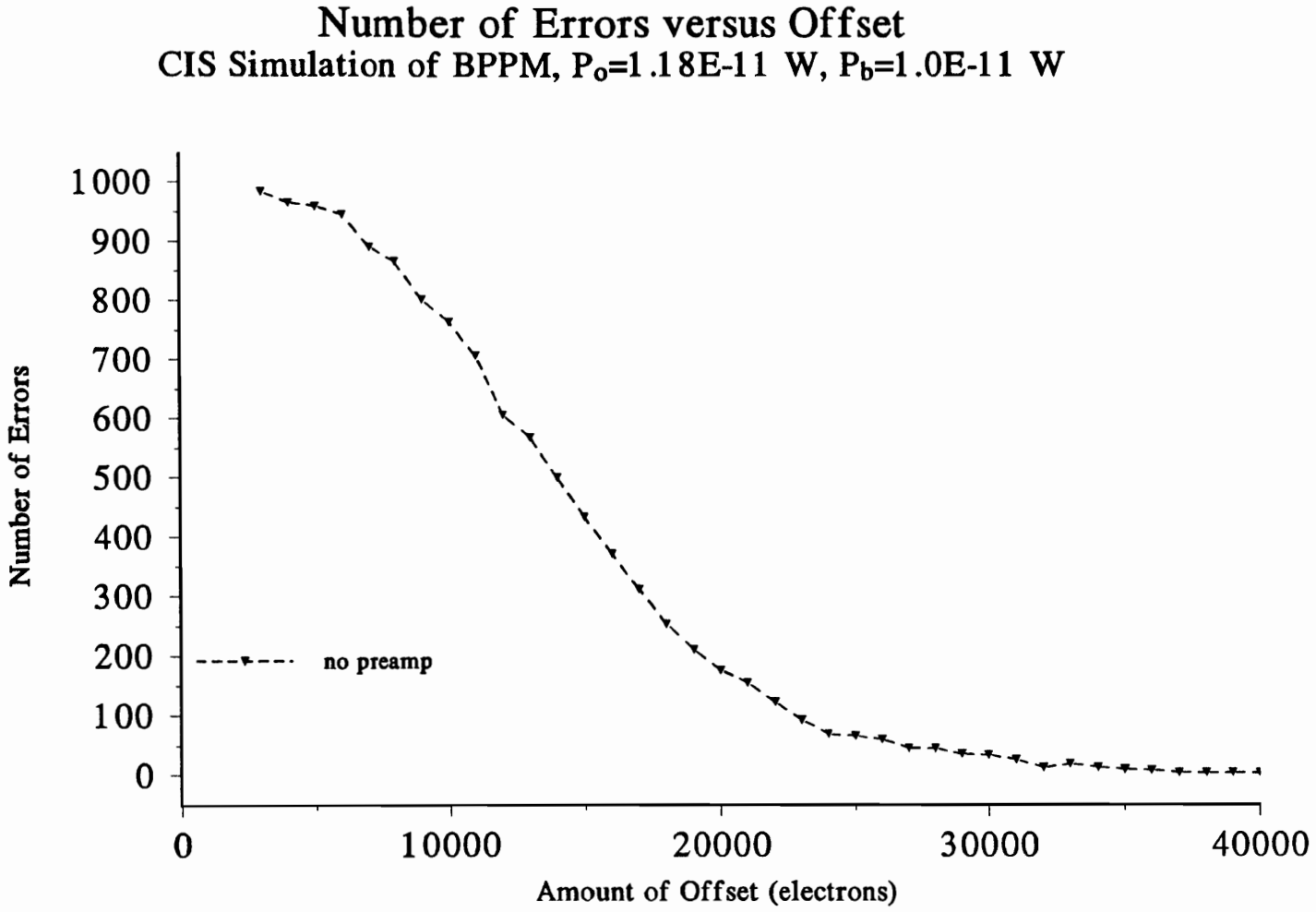


Fig. 6.13 Number of Errors Generated versus Amount of Offset Using BPPM CIS

CHAPTER 7
SUMMARY AND CONCLUSION

This concludes the derivation and demonstration of the use of importance sampling to reduce the amount of time required to simulate a M-PPM APD optical receiver. The ability to apply importance sampling to the simulation of these receivers increases the utility of this tool for the investigation of free-space optical communication links utilizing this technology. It is now possible to characterize receiver performance to BER levels heretofore unobtainable. This will enhance the ability to investigate free-space optical communication links as the simulation procedure was developed to include non-constant input optical pulse shapes which can result from channel influences.

In Chapter 2, the fundamental operation of the M-PPM maximum likelihood receiver was reviewed and presented. This included a description and explanation of the APD output statistics. The approximation to the output PDF known as the WMC PDF was introduced. It was seen that an APD model based on this approximation was both valid and carried the ability to accept any input power level as input. The use of this PDF in the simulation results in an efficient model, in terms of computer time, due the existence of a WMC PDF random variate generation method given by Ascheid. A CDF for the WMC PDF was discovered in the course of this research and presented in chapter 2 as well. The existence of a CDF for the WMC PDF greatly enhances the ability to perform certain importance sampling operations using an APD model based on the WMC statistics. This CDF, or more likely the link established here between the WMC PDF and the inverse Gaussian probability density, should aid in the analytical treatment of APD receivers as well. Chapter 2 concludes with a discussion of the remaining error sources in an APD optical receiver and a description of extinction ratio.

Chapter 3 presented the theory and practice of classical Monte Carlo simulation and its modified form known as importance sampling. This discussion introduced the basic concepts of biasing the input noise sources in a simulation and the subsequent unbiasing required to obtain the correct results from the simulation. Monte Carlo simulation and importance sampling simulation were introduced using a threshold detector receiver example. This is referred to as the one dimensional case, as opposed to the multi-dimensional case which characterizes the M-PPM receiver, in this dissertation. Chapter 3 concludes with a discussion of the simulation

package BOSS used to demonstrate some of the concepts developed in this dissertation. The model used for the unmodified APD and the basic PPM receiver system simulation model are given in Chapter 3 as well.

The available literature on the subject of importance sampling is reviewed in Chapter 4. The concept of an optimum bias function is introduced and examples from the literature for biasing the input error densities to a simulation are discussed in detail. It is seen that there is no universal approach to the application of importance sampling and that approximately nine methods to bias the input error densities of a simulation can be identified from the literature. Some of these are attempts to approximate the optimum bias function. The pseudo-optimum importance sampling (PIS) method is the prime example of this with absolute value importance sampling (AIS) and conditional importance sampling (CIS) falling into this category as well. Several methods have been developed with the purpose of mitigating the effects of system memory, as in efficient importance sampling (EIS) and Household importance sampling (HIS). The staples of importance sampling biasing must be considered to be the techniques of increasing the variance of the input noise densities, scaling importance sampling (SIS), and the technique of shifting the mean of the input densities known in the literature as improved importance sampling (IIS). It is shown that IIS leads to a more efficient importance sampling operation than SIS. Chapter 4 also contains a discussion of system memory as it effects the ability of importance sampling to reduce the number of samples that must be observed in the simulation procedure.

Chapter 4 also groups the importance sampling literature by the type of system that is addressed. Chapter 4 concludes with a presentation of optical system simulation literature that has addressed the need to increase the speed of the estimation. This includes one discussion of the use of importance sampling in optical APD threshold receiver simulation. The APD model used in that work was based on a look-up table method for generating the APD output random variates. Other works describe the use of semi-analytical techniques to increase the speed of the calculation, including the use of tail-extrapolation.

Chapter 5 presents the theory and practice of importance sampling as applied to the M-PPM maximum likelihood receiver. This discussion introduces the concepts of a conditional indicator function and a multi-dimensional probability of error region. An optimum bias function for the M-PPM receiver is developed. It is shown to hold properties similar to its one-dimensional analog in that it is effectively unrealizable. The variance of the M-PPM importance sampling estimator is established and is used to determine the desired form of the biased input

noise densities. The desired form of the biased input density is, of course, that which will decrease the number of samples required to be observed in the simulation process over the number in the classical Monte Carlo case.

At this point, Chapter 5 begins the development of two of the many possible bias methods which can be applied to the M-PPM APD optical receiver. The first of the two methods is to shift the mean of the APD density up during the periods that the PPM slot is empty. The second is a variation on the PIS importance sampling technique as it is applied to the M-PPM receiver architecture. We have designated this as conditional importance sampling for in this method the values of the M-PPM *OFF* slots are conditioned on the values of the M-PPM *ON* slots in such a way to guarantee the generation of an M-PPM symbol error. This M-PPM conditional IS method is seen to be only partially implementable at this time. The technique requires the solution of an integral of the M-PPM joint probability density function over an assumed importance region of the M-dimensional probability space. This integral has yet to be solved.

The design and construction of modules to implement each of the two M-PPM importance sampling methods is described in Chapter 5. The first APD model is one in which the mean of the output statistics are shifted up, i.e. made more positive. This will cause the generation of more PPM errors if used during the M-PPM *OFF* slots. The second APD module is one in which the output is limited to values greater than a given threshold. The resulting output random variates of this conditional APD are distributed according to the tail of the WMC PDF above the threshold. The design algorithm for generating these variates is given. This same design can be modified to produce an APD which generates APD output values which are *less* than a given threshold. The combination of these two conditional type APD's could be used in the PIS type simulation of optical APD threshold receivers. Further, a technique for implementing the machine calculation of the WMC cumulative distribution function is also presented. These models are put together to form a technique to generate M-PPM words in which the *OFF* values are conditioned on the PPM *ON* slot value. The construction of a simulation block diagram to perform this function is described. The complete system model for the importance sampling simulation, using either the shifting method or the conditional method, of the M-PPM APD receiver is given and discussed in Chapter 5. This includes a discussion of the modified method of counting errors which is required to *unbias* the importance sampling simulation and a description of how to calculate the M-PPM symbol weight.

In Chapter 6, some of the principles developed in this dissertation are demonstrated.

The use of importance sampling on the simulation of BPPM and QPPM receiver systems is demonstrated. Reduction by several orders of magnitude in the number of bits required to be observed is demonstrated through the production of a BER curve for the BPPM receiver down to a level of 10^{-12}. We also demonstrate that the importance sampling estimator is unbiased through a simple test. It is found that the importance sampling simulation of QPPM is not as efficient as the IS simulation of BPPM. This is due to the increased memory inherent in the QPPM word. The technique for applying importance sampling is described in detail, including a description of how to determine that the procedure is working correctly. Finally, conditional importance sampling for the M-PPM receiver is implemented and shown to function as predicted lacking only the solution of the joint PDF integral as discussed above.

This research has accomplished the goal of providing the basis of an improved simulation tool for the investigation of free-space optical communication systems. The work described here has also opened up several paths for future work in the area of importance sampling, in the areas of optical receiver investigation and optical receiver simulation. Let us discuss them individually.

The investigation of importance sampling as applied to the M-PPM receiver can continue at great length. For example, the effects of word length on the improvement ratio performance of the M-PPM importance sampling receiver needs to be investigated. After characterizing the magnitude of the problem, appropriate methods to mitigate the deleterious effects of increased memory can be investigated. For example, the extent of the problem may necessitate the somewhat drastic action of implementing some memory reducing from of IS, like EIS, to the simulation of M-PPM receivers or it may only require careful attention to sample interval selection.

The consequences relative to the choice of memory length used in the IS simulation of the M-PPM receiver should be characterized. Assuming an incorrect memory length or the assumption of a finite memory length when using a IIR filter in the system can lead to errors in the simulation. The memory length choice to use in the simulation of any system, M-PPM or otherwise, using an IIR filter is an open issue in the science of importance sampling.

Other methods of implementing importance sampling simulation of the M-PPM receiver should be investigated. The methods selected here, shifting up during the *OFF* slots and conditional IS in which the *OFF* slots are conditioned against the *ON* slot, were chosen as representative examples only. Investigation should be made as to which of the many possible

bias methods presented in Chapter 5, if any, might yield better performance over another. For example, is it just as good to shift up only during the *OFF* slots as it would be to shift down during the *ON* slot? Perhaps it results in a more efficient simulation if one shifts both up and down during the appropriate slot in the PPM word. Each of these methods should be implemented and compared for efficiency and accuracy.

As for optical receiver simulation, it is now possible to apply importance sampling to the simulation of APD optical receivers utilizing threshold detection. Two methods are now available for investigation. Shifting, both up and down, of the APD output can be applied to the simulation of such a receiver. The ability to do this is now greatly enhanced by to the discovery of a CDF for the WMC PDF and with it the ability to form an accurate APD model in which the output density is shifted down. The application of importance sampling to this sort of receiver simulation should be investigated and implemented. As a result of this research, it is also possible to apply pseudo-optimum type importance sampling to the optical APD receiver. This is possible due to the development of a conditional random variate generator of the WMC probability density in the course of this work, the discovery of a CDF for the WMC density and the development of a method for the machine calculation of the CDF of the WMC density. This latter ability is necessary for the calculation of the weight value of the conditional WMC random variate. Each of these methods should be implemented, explored and compared for efficiency. This will provide an improved tool for the characterization of fiber optic receiver performance.

In the general area of importance sampling, the average weight is seen to provide indication of the status of the importance sampling simulation. It is proposed that the average weight estimate, in combination with the maximum and minimum weight and the number of errors generated proportional to the number bits observed, can be used to determine in some automatic fashion the optimum shift value for the bias function. It might also be used to determine when the simulation has reached a specified accuracy in order to automatically terminate the simulation. In other words, it is suspected that the average weight, along with the other IS system indicators above, could be used to provide feed-back to the simulation process in order to automatically control the selection of the optimum bias parameter and to terminate the simulation when complete. Further research is required to determine if this is possible or practical.

It is now possible to carry on with investigation of free-space optical communication links through the atmosphere between earth and LEO satellites.

REFERENCES

CHAPTER 1

[1] J. H. Shapiro and Robert C. Harney, "Burst-Mode Atmospheric Optical Communication," IEEE 1980 National Telecommunications Conference, pp. 27.5.1-27.5.7, 1980.

[2] Sherman Karp, "An Optical Burst Communications System Through Clouds," *Digital Communications*, E. Biglieri and G. Prati, Ed., pp. 171-178, Proceedings of the Second Tirrenia International Workshop on Digital Communications, Sept. 2-6, (North-Holland), 1985.

[3] Morris Katzman, ed., *Laser Satellite Communications*, Printice-Hall, 1987.

[4] William K. Pratt, *Laser Communication Systems*, John Wiley & Sons, 1969.

[5] David L. Begley, ed., *Selected Papers on Free-Space Laser Communications*, SPIE Milestone Series, Vol. MS-30, SPIE, Bellingham WA, 1990.

[6] James B. Abshire, "Performance of OOK and Low-Order PPM Modulations in Optical Communications When Using APD-Based Receivers," IEEE Trans. Comm., Vol. COM-32, No. 10, p.1140-1143, October (1984).

[7] Robert M. Gagliardi and Sherman Karp, "M-ary Poisson Detection and Optical Communications," IEEE Trans. Comm. Tech., Vol. COM-17, No. 2, pp. 208-216, April 1969.

[8] D. L. Snyder, *Random Point Processes*, John Wiley & Sons, New York, 1975, Chpt. 2.

[9] R. M. Gagliardi and S. Karp, *Optical Communications*, John Wiley & Sons, New York, 1976, Chpt. 7.

[10] N. Sorensen and R. Gagliardi, "Performance of Optical Receivers with Avalanche Photodetection," IEEE Trans. Communications, Vol. COM-27, No. 9, p.1315-1321, Sept. (1979).

[11] Edward A. Bucher, "Computer Simulation of Light Pulse Propagation for Communication Through Thick Clouds," Applied Optics, Vol. 12, No. 10, pp. 2391-2400, October 1973.

[12] Earl J. McCartney, *Optics of the Atmosphere*, John Wiley and Sons, New York, 1976.

[13] Eli Brookner, "Multipath Dispersion for the Atmosphere Laser Channel," Proc. IEEE (Letters), Vol. 58, No. 10, pp. 1767-1769, Oct. 1970.

[14] Edward A. Bucher and Robert M. Lerner, "Experiments on Light Pulse Communication and Propagation Through Atmospheric Clouds," Applied Optics, Vol. 12, No. 10, pp. 23401-2414, October 1973.

[15] J. R. Gilder and K. Yao, "Experimental Study of the Transient Response of Multiple Scattered Laser Radiation," Proc. IEEE, Vol. 58, No. 10, pp. 1764-1766, Oct. 1970.

[16] F. M. Davidson and Xiaoli Sun, "Gaussian Approximation Versus Nearly Exact Performance Analysis of Optical Communication Systems with PPM Signaling and APD Receivers," IEEE Trans. Comm., Vol. 36, No. 11, p.1185-1192, Nov. 1988.

[17] R. Gagliardi and G. Prati, "On Gaussian Error Probabilities in Optical Receivers," IEEE Trans. Comm., Vol. COM-28, No. 9, pp. 1742-1747, Sept. 1980.

[18] J. Bryan Lyles and Daniel C. Swinehart, "The Emerging Gigabit Environment and the Role of the Local ATM," IEEE Communications Magazine, Vol. 30, No. 4, pp. 52-58, April 1992.

[19] Fritz R. Gfeller and Urs Bapst, "Wireless In-house Data Communication via Diffuse Infrared Radiation," Proc. IEEE, Vol. 67, No. 11, pp. 1474-1486, Nov. 1979.

[20] T. S. Chu and M. J. Gans, "High Speed Infrared Local Wireless Communication," IEEE Communications Magazine, Vol. 25, No. 8, pp. 4-10, Aug. 1987.

[21] K.S. Natarajan, K.C. Chen, P.D. Hortensius, "Considerations for the Design of High Speed Wireless Optical Networks," *Proceedings* IEEE Workshop on Wireless Local Area Networks, pp. 140-143, May 9-10, Worcester MA, 1991.

[22] John R. Barry, et al., "Simulation of Multipath Impulse Response for Indoor Diffuse Optical Channels," *Proceedings* IEEE Workshop on Wireless Local Area Networks, pp. 81-87, May 9-10, Worcester MA, 1991.

[23] Kamran Kiasaleh, "PN Code-Aided Ranging for Optical Satellite Communication Systems," IEEE Trans. Comm., Vol. COM-39, No. 12, pp. 1832-1844, Dec. 1991.

[24] R. A. Cryan, et al., "Optical Fibre Digital Pulse Position Modulation Assuming a Gaussian Received Pulse Shape," IEE Proceedings, Vol. 137, Pt. J, No. 2, pp. 89-96, April 1990.

[25] Ian Garrett, "Pulse Position Modulation for Transmission Over Optical Fibers with Direct or Heterodyne Detection," IEEE Trans. Comm., Vol. COM-31, No. 4, pp. 518-527, April 1983.

[26] J.J.O. Pires, "Digital Pulse Position Modulation Over Optical Fibres with Avalanche Photodiode Receivers," IEE Proceedings, Vol. 133, Pt. J, No. 5, pp. 309-313, Oct. 1986.

[27] J. R. Pierce, "Optical Channels: Practical Limits with Photon Counting," IEEE Trans. Comm., Vol. COM-26, No. 12, pp. 1819-1821, Dec. 1978.

CHAPTER 2

[1] J. Katz, "Average Power Constraints in AlGaAs Semiconductor Lasers Under Pulse-Position-Modulation Conditions," Opt. Commun., Vol. 56, pp.330-333, January (1986).

[2] G. S. Mecherle, "Impact of Laser Diode Performance on Data Rate Capability of PPM Optical Communication," Proc. IEEE Military Commun. Conference (MILCOM'85), Boston, MA, Oct. 20-23, 1985, pp. 115-121.

[3] James B. Abshire, "Performance of OOK and Low-Order PPM Modulations in Optical Communications When Using APD-Based Receivers," IEEE Trans. Comm., Vol. COM-32, No. 10, p.1140-1143, October (1984).

[4] M. Fitzmaurice, R. Bruno, "Laser Communication Transceiver for Space Station Freedom," *Free-Space Laser Communication Technologies* II, David L. Begley, Bernard D. Seery, Editors, Proc. SPIE 1218, p.439-448 (1990).

[5] D. L. Snyder, *Random Point Processes*, John Wiley & Sons, New York, 1975, Chpt. 2.

[6] R. M. Gagliardi and S. Karp, *Optical Communications*, John Wiley & Sons, New York, 1976, Chpt. 7.

[7] S. D. Personick, "Statistics of a General Class of Avalanche Detectors With Applications to Optical Communication," BSTJ, Vol. 50, No. 10, pp. 3075-3095, Dec. 1971.

[8] R. J. McIntyre, "The Distribution of Gains in Uniformly Multiplying Avalanche Photodiodes: Theory," IEEE Trans. Electron Devices, Vol. ED-19, No. 6, pp. 701-713, June 1972.

[9] J. Conradi, "The Distribution of Gains in Uniformly Multiplying Avalanche Photodiodes: Experimental," IEEE Trans. Electron Devices, Vol. ED-19, No. 6, pp. 713-718, June 1972.

[10] P. Balaban, P. E. Fleischer and H. Zucher, "The Probability Distribution of Gains in Avalanche Photodiodes," IEEE Trans. Electron Devices, Vol. ED-23, No. 10, pp. 1189-1190, Oct. 1976.

[11] F. M. Davidson and Xiaoli Sun, "Gaussian Approximation Versus Nearly Exact Performance Analysis of Optical Communication Systems with PPM Signaling and APD Receivers," IEEE Trans. Comm., Vol. 36, No. 11, p.1185-1192, Nov. 1988.

[12] N. Sorensen and R. Gagliardi, "Performance of Optical Receivers with Avalanche Photodetection," IEEE Trans. Communications, Vol. COM-27, No. 9, p.1315-1321, September (1979).

[13] P. P. Webb, R. J. McIntyre and J. Conradi, "Properties of Avalanche Photodiodes," RCA Review, Vol. 35, p. 234-278, June 1974

[14] P. Balaban, "Statistical Evaluation of the Error Rate of the Fiberguide Repeater Using Importance Sampling," BSTJ, Vol. 55, No. 6, pp. 745-766, July- August 1976.

CHAPTER 2 (cont.)

[15] Gerd Ascheid, "On the Generation of WMC - Distributed Random Numbers," IEEE Trans. Comm., Vol. COM-38, No. 12, p.2117-2118, Dec. 1990.

[16] J. Leroy Folks and R. S. Chhikara, "The Inverse Gaussian Distribution and its Statistical Application – A Review," J. R. Statist. Soc. B, Vol. 40, No. 3, pp. 263-289, 1978.

[17] M. C. K. Tweedie, "Inverse Statistical Variates," Nature, Vol. 155, p.453, 1945.

[18] N. L. Johnson and S. Kotz, "Inverse Gaussian (Wald) Distributions," Chpt. 15 in *Distributions in Statistics: Continuous Univariate Distributions-1.* Houghton Mifflon Co., Boston, 1970.

[19] M. T. Wasan, "On an Inverse Gaussian Process," Skand. AktuarTidskr., Vol. 51, pp.69-96, 1968.

[20] Athanasios Papoulis, *Probability, Random Variables, and Stochastic Processes* (New York, NY: McGraw-Hill) 1984, p. 95.

[21] S. D. Personick, P. Balaban, J. H. Bobsin and P. R. Kumar, "A Detailed Comparison of Four Approaches to the Calculation of the Sensitivity of Optical Fiber System Receivers," IEEE Trans. Comm., Vol. COM-25, No. 5, pp. 541-548, May 1977.

[22] S. D. Personick, "Receiver Design for Digital Fiber Optic Communication Systems," BSTJ, Vol. 52, No. 6, July-August, 1973, pp. 843-866.

[23] T. Muoi, "Receiver Design for High Speed Optical Fiber Systems," IEEE J. Lightwave Technology, Vol. LT-2, pp. 243-267, June 1984.

[24] A. F. Elrefaie, J. K. Townsend, M. B. Romeiser and K. S. Shanmugan, "Computer Simulation of Digital Lightwave Links," IEEE J. Selected Areas in Comm, Vol. 6, No. 1, p.94-105, Jan. 1988.

[25] R. G. Smith and S. D. Personick, "Receiver Design for Optical Fiber Communication Systems," in Semiconductor Devices for Optical Communication, H. Kressel, Ed., New York: Spring-Verlag, 1980, Ch. 4.

CHAPTER 3

[1] Michael C. Jeruchim, "Techniques for Estimating the Bit Error Rate in the Simulation of Digital Communication Systems," IEEE J. Selected Areas in Comm, Vol. SAC-2, No. 1, p.153-170, Jan. 1984.

[2] K. S. Shanmugan and P. Balaban, "A Modified Monte-Carlo Simulation Technique for the Evaluation of Error Rate in Digital Communication Systems," IEEE Trans. Comm., Vol. COM-28, No. 11, p.1916-1924, Nov. 1980.

[3] P. Balaban, "Statistical Evaluation of the Error Rate of the Fiberguide Repeater Using Importance Sampling," BSTJ, Vol. 55, No. 6, pp. 745-766, July- August 1976.

CHAPTER 3 (cont.)

[4] K. S. Shanmugan, et al., "Block-Oriented Systems Simulator (BOSS)," in Proc. MILCOM'86, Oct. 1986, paper 36.1.

[5] K. Sam Shanmugan, "An Update of Software Packages for Simulation of Communication Systems (Links), IEEE J. Selected Areas in Comm, Vol. SAC-6, No. 1, p.5-12, Jan. 1988.

[6] Gerd Keiser, *Optical Fiber Communications*, New York: McGraw-Hill, 1983, Chapter 6.

[7] Gerd Ascheid, "On the Generation of WMC - Distributed Random Numbers," IEEE Trans. Comm., Vol. COM-38, No. 12, p.2117-2118, Dec. 1990.

[8] A. F. Elrefaie, J. K. Townsend, M. B. Romeiser and K. S. Shanmugan, "Computer Simulation of Digital Lightwave Links," IEEE J. Selected Areas in Comm, Vol. 6, No. 1, p.94-105, Jan. 1988.

[9] *Block Oriented Systems Simulator (BOSS) User's Guide*, Ver. 2.7, Simulation Parameters, p. 2-31, COMDISCO Systems, Inc., June 1991.

CHAPTER 4

[1] H. Kahn, "Use of Different Monte Carlo Sampling Techniques," Symposium on Monte Carlo Methods, edited by H. A. Mayer, New York: John Wiley, 1956, pp. 146-190.

[2] P. Balaban, "Statistical Evaluation of the Error Rate of the Fiberguide Repeater Using Importance Sampling," BSTJ, Vol. 55, No. 6, pp. 745-766, July-August 1976.

[3] K. S. Shanmugan and P. Balaban, "A Modified Monte-Carlo Simulation Technique for the Evaluation of Error Rate in Digital Communication Systems," IEEE Trans. Comm., Vol. COM-28, No. 11, p.1916-1924, Nov. 1980.

[4] R. L. Mitchell, "Importance Sampling Applied to Simulation of False Alarm Statistics," IEEE Trans. Aerosp. Electron. Syst., Vol. AES-17, pp. 15-24, Jan. 1981.

[5] Gerald W. Lank, "Theoretical Aspects of Importance Sampling Applied to False Alarms," IEEE Trans. of Info. Theory, Vol. IT-29, No. 1, pp. 73-82, Jan. 1983.

[6] Wael A. Al-Qaq, Michael Devetsikiotis and J. Keith Townsend, "Importance Sampling Methodologies for Simulation of Communication Systems with Adaptive Equalizers and Time-Varying Channels," in *Conf. Record*, IEEE GLOBECOM '92, Vol. 3, Orlando FL, pp. 1830-1834, Dec. 6-9, 1992.

[7] Gerd Ascheid, "On the Generation of WMC - Distributed Random Numbers," IEEE Trans. Comm., Vol. COM-38, No. 12, p.2117-2118, Dec. 1990.

[8] Norman C. Beaulieu, "A Composite Importance Sampling Technique for Digital Communication System Simulation," IEEE Trans. Comm., Vol. COM-38, No. 4, pp. 393-396, April 1990.

[9] Norman C. Beaulieu, "An Investigation of Gaussian Tail and Rayleigh Tail Density Functions for Importance Sampling Digital Communication System Simulation," IEEE Trans. Comm., Vol. COM-38, No. 9, pp.1288-1292, Sept. 1990.

[10] Cao shigang, Lin jingdong and Feng chongxi, "Modified Efficient Importance Sampling and its Application in the Simulation of MQAM Systems," Proceedings of IEEE Globecom'90, pp. 704.5.1-704.5.5, 1990.

[11] Tao Chen and Charles L. Weber, "Nonlinear Communication System Simulation via Conditional Importance Sampling," MILCOM'90, Conf. Rec., Monterey, p.62.1.1-62.1.5, Sept.30-Oct.3, 1990.

[12] S.X. Chen, P.J. Smith, M. Shafi and D. Vere-Jones, "Some Improvements to Conventional Importance Sampling Techniques for Coded Systems Using Viterbi Decoding," Electronics Letters, Vol. 26, No. 12, pp.802-804, 7 June 1990.

[13] Bruce R. Davis, "An Improved Importance Sampling Method for Digital Communication System Simulations," IEEE Trans. Comm., Vol. COM-34, No. 7, p.715-719, July 1986.

[14] Michael Devetsikiotis and J. Keith Townsend, "A Useful and General Technique for Improving the Efficiency of Monte Carlo Simulation of Digital Communication Systems," Proceedings of IEEE Globecom'90, pp. 704.4.1-704.4.7, 1990.

[15] Michael Devetsikiotis and J. Keith Townsend, "On the Efficient Simulation of Large Communication Networks Using Importance Sampling," in *Conf. Record*, IEEE GLOBECOM'92, Vol. 3, Orlando FL, pp. 1455-1859, Dec. 6-9, 1992.

[16] A. F. Elrefaie, J. K. Townsend, M. B. Romeiser and K. S. Shanmugan, "Computer Simulation of Digital Lightwave Links," IEEE J. Selected Areas in Comm, Vol. 6, No. 1, p.94-105, Jan. 1988.

[17] Peter M. Hahn and Michel C. Jeruchim, "Developments in the Theory and Application of Importance Sampling," IEEE Trans. Comm., Vol. COM-35, No. 7, p.706-714, July 1987.

[18] P. M. Hahn, M. C. Jeruchim and T. J. Klandrud, "Implementation of Importance Sampling in Multi-Hop Communication Simulation," in Proc. IEEE GLOBECOM '86, Vol. 1, Houston TX, pp. 4.1.1-4.1.5, Dec. 1986.

[19] M. A. Herro and J. M. Nowack, "The use of importance sampling in coded digital communication simulation," 5th Int. Conf. Syst. Eng., Fairborn OH, pp. 477-480, Sept. 1987.

[20] M. A. Herro and J. M. Nowack, "Simulated Viterbi decoding using importance sampling," IEE Proceedings, Vol. 135, Pt. F, No. 2, pp. 133-142, April 1988.

[21] C.-L. I and B. B. Lusignan, "The Optimum BER Estimator for Digital Satellite Communication Systems," in Collect. Tech. Papers AIAA 11th Commun. Satellite Syst. Conf., San Diego, CA, pp. 144-151, March 1986.

[22] Michel C. Jeruchim, Peter M. Hahn, Kevin P. Smyntek and Robert T. Ray, "An Experimental Investigation of Conventional and Efficient Importance Sampling," IEEE Trans. Comm., Vol. COM-37, No. 6, p.578-587, June 1989.

[23] Michel C. Jeruchim, "On the Application of Importance Sampling to the Simulation of Digital Satellite and Multihop Links," IEEE Trans. Comm., Vol. COM-32, No. 10, p.1088-1092, Oct. 1984.

[24] Michel C. Jeruchim, "Techniques for Estimating the Bit Error Rate in the Simulation of Digital Communication Systems," IEEE J. Selected Areas in Comm, Vol. SAC-2, No. 1, p.153-170, Jan. 1984.

[25] Dingqing Lu and Kung Yao, "Improved Importance Sampling Technique for Efficient Simulation of Digital Communication Systems," IEEE J. Selected Areas in Comm, Vol. 6, No. 1, p.67-75, Jan. 1988.

[26] D. Lu and K. Yao, "On Some New Importance Sampling Results for Simulation of Non-Linear Digital Communication Systems," in *Proceedings of the 1988 Conference on Information Sciences and Systems*, pp. 336-340, 1988.

[27] D. Lu and K. Yao, "Importance Sampling Simulation Techniques Applied to Estimating False Alarm Probabilities," 1989 IEEE International Symposium on Circuits and Systems (ISCAS'89), Vol. 1 of 3, p.598-601, May 8-11, Portland OR, 1989.

[28] D. Lu and K. Yao, "Estimation Variance Bounds of Importance Sampling Simulations in Digital Communication Systems," IEEE Trans. Comm., Vol. COM-39, No. 10, p.1413-1417, Oct. 1991.

[29] Geoffrey Orsak and Behnaam Aazhang, "On the Theory of Importance Sampling Applied to the Analysis of Detection Systems," IEEE Trans. Comm., Vol. COM-37, No. 4, p.332-339, April 1989.

[30] Geoffrey Orsak and Behnaam Aazhang, "Constrained Solutions in Importance Sampling via Robust Statistics," IEEE Trans. of Info. Theory, Vol. 37, No. 2, pp. 307-316, March 1991.

[31] Geoffrey Orsak and Behnaam Aazhang, "Efficient Importance Sampling Techniques for Simulation of Multiuser Communication Systems," IEEE Trans. Comm., Vol. COM-40, No. 6, p.1111-1118, June 1992.

[32] Keshab K. Parhi and Raymond S. Berkowitz, "On Optimizing Importance Sampling Simulations," IEEE Trans. Circuits and Systems, Vol. CAS-34, No. 12, pp. 1558-1563, Dec. 1987.

[33] Zoran R. Petrovic and Dragan O. Cuberovic, "Application of Efficient Importance Sampling Method to the Simulation of Digital Satellite Links," MELECON'89: Mediterranean Electrotechnical Conference Proceedings, A.M. Barbosa, Ed., pp. 485-8, Lisbon, 11-13 April 1989.

[34] John S. Sadowsky, "A New Method of Viterbi Decoder Simulation Using Importance Sampling," IEEE Trans. Comm., Vol. COM-38, No. 9, p.1341-1351, Sept. 1990.

[35] John S. Sadowsky and Randall K. Bahr, "Direct-Sequence Spread-Spectrum Multiple-Access Communications with Random Signature Sequences: A Large Deviations Analysis," IEEE Trans. Info. Theory, Vol 37, No. 3, pp. 514-527, May 1991.

[36] John S. Sadowsky and James A Bucklew, "On Large Deviation Theory and Asymptotically Efficient Monte Carlo Estimation," IEEE Trans. Info. Theory, Vol 36, No. 3, pp. 579-588, May 1990.

[37] Heinz-Josef Schlebusch, "Nonlinear Importance Sampling Techniques for Efficient Simulation of Communication Systems," IEEE International Conference on Communications ICC'90, Conference Record, p. 312.5.1-312.5.5, 16-19 April 1990.

[38] J. Keith Townsend and K. Sam Shanmugan, "On Improving the Computational Efficiency of Digital Lightwave Link Simulation," IEEE Trans. Comm., Vol. COM-38, No. 11, p.2040-2048, Nov. 1990.

[39] Q. Wang and V. K. Bhargava, "On the Application of Importance Sampling to BER Estimation in the Simulation of Digital Communication Systems," IEEE Trans. Commun., Vol. COM-35, No. 11, pp. 1231-1233, Nov. 1987.

[40] Robert J. Wolfe, Michel C. Jeruchim and Peter M. Hahn, "On Optimum and Suboptimum Biasing Procedures for Importance Sampling in Communication Simulation," IEEE Trans. Comm., Vol. COM-38, No. 5, p.639-647, May 1990.

[41] P. Bratley, B. L. Fox, L. E. Schrage, *A Guide to Simulation*, 2nd ed. Springer-Verlag, 1987.

[42] J. M. Hammersley and D. C. Handscomb, *Monte Carlo Methods*, New York: Methuen, 1964.

[43] R. Y. Rubinstein, *Simulation and the Monte Carlo Method*, New York: Wiley, 1981.

[44] J. P. C. Kleijnen, *Statistical Techniques in Simulation*, Part 1, Chapter III, New York: Marcel Decker, 1974, Part 1.

[45] A. M. Law and W. D. Kelton, *Simulation Modeling and Analysis*, New York: McGraw-Hill, 1991.

[46] V. G. Hansen, "Importance Sampling in computer simulation of signal processors," *Computers and Electrical Engineering*, Vol. 1, pp.545-550, 1974.

[47] H. Kahn and A. W. Marshall, "Methods of Reducing Sample Size in Monte Carlo Computations," Journal of Operational Research Society of America, Vol. 1, pp. 263-278, 1953.

CHAPTER 4 (cont.)

[48] P. P. Webb, R. J. McIntyre and J. Conradi, "Properties of Avalanche Photodiodes," RCA Review, Vol. 35, p. 234-278, June 1974

[49] M. Fashano and A. L. Strodtbeck, "Communications Systems Simulation and Analysis with SYSTID," IEEE J. Selected Areas in Comm, Vol. SAC-2, No. 1, p.8-28, Jan. 1984.

[50] K. S. Shanmugan, et al., "Simulation of Lightwave Communication Links Using SYSTID," presented at Globecom'85, New Orleans, LA, Dec. 1985.

[51] Donald G. Duff, "Computer-Aided Design of Digital Lightwave Systems," IEEE J. Selected Areas in Comm, Vol. SAC-2, No. 1, p.171-185, Jan. 1984.

[52] S. D. Personick, P. Balaban, J. H. Bobsin and P. R. Kumar, "A Detailed Comparison of Four Approaches to the Calculation of the Sensitivity of Optical Fiber System Receivers," IEEE Trans. Comm., Vol. COM-25, No. 5, pp. 541-548, May 1977.

CHAPTER 5

[1] Michael C. Jeruchim, "Techniques for Estimating the Bit Error Rate in the Simulation of Digital Communication Systems," IEEE J. Selected Areas in Comm, Vol. SAC-2, No. 1, p.153-170, Jan. 1984.

[2] K. S. Shanmugan and P. Balaban, "A Modified Monte-Carlo Simulation Technique for the Evaluation of Error Rate in Digital Communication Systems," IEEE Trans. Comm., Vol. COM-28, No. 11, p.1916-1924, Nov. 1980.

[3] A. F. Elrefaie, J. K. Townsend, M. B. Romeiser and K. S. Shanmugan, "Computer Simulation of Digital Lightwave Links," IEEE J. Selected Areas in Comm, Vol. 6, No. 1, p.94-105, Jan. 1988.

[4] Gerd Ascheid, "On the Generation of WMC - Distributed Random Numbers," IEEE Trans. Comm., Vol. COM-38, No. 12, p.2117-2118, Dec. 1990.

[5] Donald E. Knuth, *The Art of Computer Programing*, Vol. 2: Semi-Numerical Algorithms, 2^{nd} Ed., Addison-Wesley: Reading MA, p.123 (1981).

CHAPTER 6

[1] Michael Devetsikiotis and J. Keith Townsend, "A Useful and General Technique for Improving the Efficiency of Monte Carlo Simulation of Digital Communication Systems," Proceedings of IEEE Globecom'90, pp. 704.4.1-704.4.7, 1990.

[2] D. Lu and K. Yao, "On Some New Importance Sampling Results for Simulation of Non-Linear Digital Communication Systems," in *Proceedings of the 1988 Conference on Information Sciences and Systems*, pp. 336-340, 1988.

[3] D. Lu and K. Yao, "Estimation Variance Bounds of Importance Sampling Simulations in Digital Communication Systems," IEEE Trans. Comm., Vol. COM-39, No. 10, p.1413-1417, Oct. 1991.

GLOSSARY

AIS Absolute value Importance Sampling, in which the input noise density is biased by taking the absolute value and shifting the mean.

APD Avalanche Photodiode

AWGN Average White Gaussian Noise

BER Bit Error Rate, also known as the Bit Error Ratio

BJT Bipolar Junction Transistor

BPPM Binary Pulse Position Modulation = 2 slot pulse position modulation

BSTJ Bell System Technical Journal

CDF Cumulative Distribution Function

CIS Conditional Importance Sampling, in which one or more of the input noise densities is conditioned on the value of another input noise density in such a way as to generate errors.

COV covariance

DS-SSMA Direct Sequence - Spread Spectrum Multiple Access

dt simulation sampling interval in seconds

EIS Efficient Importance Sampling, an importance sampling technique developed to mitigate the effects of system memory by transferring the input noise densities to the output through the use of a linear or linearized system transfer function.

erf error function

erfc complimentary error function

ESA European Space Agency

$f(\cdot)$ unmodified probability density function

$f^*(\cdot)$ modified (biased) probability density function

FOV Field of View

Gbps	Giga-bits per second
GEO	Geostationary Earth Orbit
GHz	Giga-Hertz
GSFC	Goddard Space Flight Center, NASA
HIS	Householder Importance Sampling, an importance sampling technique developed to mitigate the effects of system memory by transferring the input noise densities to the output through a Householder transformation based on system transfer function.
i.i.d.	independent and identically distributed
IIR	infinite impulse response
IIS	Improved Importance Sampling, in which the input noise density is biased by shifting the mean.
ISDN	Integrated Services Digital Network
ISI	inter-symbol interference
κ	ionization ratio of the APD, the ratio of the ionization rate of holes to electrons
LEO	Low Earth Orbit
Lidar	lightwave version of Radar
M-PPM	M-ary Pulse Position Modulation
Mbps	mega-bits per second
MC	Monte Carlo, usually referring to classical Monte Carlo simulation
MEIS	Modified Efficient Importance Sampling, combined use of EIS and PIS
N	the number of trials (number of bits observed) in classical Monte Carlo simulation
N*	the number of trials (number of bits observed) in importance sampling simulation
NASA	National Aeronautics and Space Administration
NRZ	Non-Return to Zero
OOK	On-Off Keying
PDF	Probability Density Function

PIS Pseudo-optimum Importance Sampling, in which a biased input noise density is obtained by assuming an importance region and truncating the true input density outside this assumed region.

PPM Pulse Position Modulation

PSK Phase Shift Keying

QPPM Quaternary Pulse Position Modulation = 4 slot pulse position modulation

RF Radio Frequency

RHS Right Hand Side

SIS Scaling Importance Sampling, in which the input noise density is biased by increasing the variance.

WMC Webb, McIntyre and Conradi; more properly known as P. P. Webb, Robert J. McIntyre and Jan Conradi.

VITA

Kenneth Baker received the B.S. degree in Electrical Engineering from the University of Maryland, College Park in May 1983. While an undergraduate, he was a co-operative education intern at the National Bureau of Standards (now NIST) where he worked in the Electrical Measurements and Standards Division. After graduation, from 1983 to 1985, he work in the Communications Systems Division of the Harris Corporation, Melbourne Florida as a Senior RF and Microwave Engineer. While there he worked on microwave hardware including the design of gallium arsenide microwave monolithic integrated circuits for high frequency analog applications. In September 1985, he came to Virginia Tech to pursue advanced degrees in Electrical Engineering. Mr. Baker was awarded a Master of Science in Electrical Engineering in 1987. His master's topic concerned the automated prediction of VHF transmitter coverage area using digital terrain maps. As a member of Virginia Tech's Satellite Communications Group he has worked on propagation prediction of terrestrial UHF/VHF links, microstrip filter design for FSK receivers and participated in the ground terminal design for 12.5, 20 and 30 GHz propagation experiments using ESA's OLYMPUS satellite. He has also studied continuous phase modulation and coding methods for VHF/UHF LEOSAT communication applications. From 1989 to 1991, he was a member of the research staff of the Virginia Tech chemistry department. There, he developed methods of measuring the complex dielectric constant of liquids and solids at microwave frequencies using a vector network analyzer. Interest in optical satellite communications systems began when he interned at the Japanese National Space Development Association (NASDA) as a Fellow of the National Science Foundation Summer Institute in Japan, 1990.

Mr. Baker is a member of the IEEE and a licensed amateur radio operator. He has been the recipient of Pratt Presidential Fellowship 1987-1988, NSF Fellowship for the Study of the Japanese Language 1989-1990, NSF Summer Institute in Japan 1990 and a NASA Graduate Student Researcher Fellowship. Mr. Baker has studied the spoken and written Japanese language for two years while in graduate school. His research interests include electromagnetic wave propagation in turbulent media, receiver design and free-space optical communications technology.