Université d'Ottawa · University of Ottawa

Study of a Spread Spectrum Subcarrier Multiplexed Optical Network

by

Tania Boutilier, B.Eng.

A thesis submitted to the School of Graduate Studies and Research

in partial fulfillment of the requirements for the degree of

Master of Applied Science

Ottawa-Carleton Institute for Electrical Engineering

Department of Electrical Engineering

Faculty of Engineering

University of Ottawa

September, 1996

Abstract

This thesis investigates the performance of a hybrid modulation scheme, combining subcarrier multiplexing and code division multiple access. This scheme may be useful in tomorrow's optical local area networks to take advantage of the capacity of optical fibers using commercially available optical technology.

The application of subcarrier multiplexing to optical local networks has generated much interest recently, however it is sensitive to nonlinear distortion generated by the laser diodes used to transmit data over the optical network. Earlier research has indicated that combining code division multiple access with subcarrier multiplexing may provide improved performance. However the lasers' nonlinearity has not been specifically included in earlier research, due to the complexity of the analysis.

In this thesis a simulation is used to evaluate the performance of the hybrid SCM-CDMA optical network. A custom program was written in the C language for this purpose. Preliminary results from the simulation program indicate that the performance of subcarrier multiplexed signals sent over an optical network may improve fairly substantially if code sequences are used to spread the bandwidth of information signals prior to optical transmission.

Acknowledgments

I would like to thank my supervisor Dr. Peter Galko for his help and encouragement over the course of the research and the writing of this thesis.

This work was partly supported by the Telecommunications Research Institute of Ontario (TRIO). Photonic Network Architectures project. I am grateful for the support offered by TRIO.

I would also like to thank my colleague Farideh Khaleghi at the University of Ottawa for several useful discussions I had with her during the course of this work.

Many thanks also to Bill Keays for his help with the workstations used to run earlier. SPW-based versions of the simulation.

Last but not least. I would like to express my heartfelt gratitude to my husband for his help and encouragement. Without his support this thesis would not have been completed.

Abbreviations

APD	Avalanche PhotoDiode
AWGN	Additive White Gaussian Noise
BPSK	Binary Phase Shift Keying
CDMA	Code Division Multiple Access
DEPSK	Differentially Encoded Phase Shift Keying
DPSK	Differential Phase Shift Keying
DS-SSMA	Direct Sequence Spread Spectrum Multiple Access
FDM	Frequency Division Multiplexing
FDMA	Frequency Division Multiple Access
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
IDT	InterDigital Transducer
IIR	Infinite Impulse Response
IMP	InterModulation Product
LAN	Local Area Network
LED	Light Emitting Diode

p-i-n diode	Positive-Intrinsic-Negative diode
p-n diode	Positive-Negative diode
PN-sequence	Pseudo-Noise sequence
RIN	Relative Intensity Noise
SAW	Surface Acoustic Wave
SCM	SubCarrier Multiplexing
SNR	Signal-to-Noise Ratio
SSMA	Spread Spectrum Multiple Access
TDM	Time Division Multiplexing
TDMA	Time Division Multiple Access
WDM	Wavelength Division Multiplexing
WDMA	Wavelength Division Multiple Access

Symbols

B	bandwidth
c	velocity of light (3*E*8 m/s)
e	electronic charge (1.6*E*–19 C)
E	energy
E_b	bit energy
f	frequency
f_s	simulation sampling rate
f_c	subcarrier frequency
$f_n \mid \omega_n$	natural undamped frequency
h	Planck's constant (6.626*E*–34 Js)
$H(f)$	frequency response of $h(t)$
I	current
K	Boltzmann's constant (1.38*E*–23 J/K)
λ	wavelength
P	power
$p_T(t)$	rectangular pulse waveform of duration T

N	number of users or pseudo-random sequence period
N_u	number of users in network
N_f	number of frequencies in network
N_c	number of spreading codes in network
N_s	number of clusters in network
$\frac{N_o}{2}$	AWGN spectral density
σ^2	noise variance
T	data symbol time interval (usually)
T_s	simulation sampling time
ω	radian frequency
ξ	damping ratio

Code Sequences:

$a(n)$	code sequence with values {0, 1}
$\alpha(n)$	code sequence with values {−1, +1}
$C_{x,y}(\cdot)$	discrete aperiodic cross-correlation function of sequences x and y
$h(x)$	generator polynomial of a code sequence
$R_{i,k}(\cdot)$ $\hat{R}_{i,k}(\cdot)$	continuous-time partial cross-correlation functions of signals i and k where τ is the delay between i and k

T_c	chip time interval
$\theta_{x,x}(\cdot)$	periodic autocorrelation function of sequence x
$\theta_{x,y}(\cdot)$	periodic cross-correlation function of sequences x and y
θ_c	peak cross-correlation magnitude
θ_a	peak out-of-phase autocorrelation magnitude
θ_{max}	maximum autocorrelation function

Laser:

g	optical gain coefficient
$I(t)$, I_A	current injected into the laser's active region
I_{DC}	constant bias current
I_{th}	threshold current for lasing
$N(t)$	carrier density
N_0	carrier density for transparency
m	optical modulation index
$\langle s \rangle$	average laser light intensity
$\langle \Delta s^2 \rangle$	mean square intensity of fluctuation of laser light output
$S(t)$	photon density

β	spontaneous emission factor
ε	gain compression factor
Γ	optical confinement factor
τ_p	photon lifetime
τ_{sp}	spontaneous recombination lifetime of carriers
V'	volume of laser's active region times electron charge

Optical Receiver and Photodetector:

C_D	internal capacitance of photodetector
f_{3dB}	cutoff frequency of optical receiver
F_n	noise factor of amplifier
i_A	noise current of optical receiver's operational amplifier
i_{TL}	thermal noise current of load resistance R_L
i_{TF}	thermal noise current of optical receiver feedback resistor R_F
I_D	photodetector's dark current
Ip	photodetector's initial primary photocurrent (before multiplication occurs in APDs).
i_s	photodetector shot noise current
$-G_o(f)$	open loop gain of optical receiver's operational amplifier

η	photodetector's quantum efficiency
P_o	optical power at input to photodetector
R	photodetector's responsivity
R_D	internal resistance of photodetector
R_F	optical receiver feedback resistor
R_L	load resistance of photodetector
T_c	effective temperature of amplifier
T_o	standard reference temperature (290 K)
T_R	temperature of R_L
v_A	noise source of optical receiver's operational amplifier

Table of Contents

List of Figures

Chapter 1: Introduction

1.1 Motives

The world of long distance communications has been revolutionized by the appearance of lightwave technology. The key to the great success of optical communications, lies with several advantages it offers over conventional electrical communication. Perhaps the most important of these are the enormous potential bandwidth (in the range of tens of Terahertz) and low transmission loss of optical fibers.

In recent years, the application of lightwave technology to local area networks (LANs) has emerged as an area of interest. It is thought that local area networks in the form of computer networks or "subscriber loops" in telephone networks, for example, represent the "ultimate market" for lightwave technology [1]. However, to date most optical LANs that have been demonstrated, have failed to take advantage of the huge transmission capacity of optical fibers, as is the case with point-to-point transmission links. At the present state of technology, optical LANs are limited by expensive optical components and lossy optical couplers among others.

In order to maximize the information transfer over any communication link, it is usual to multiplex several signals onto the transmission medium. Multiplexing schemes are related to multiple-access schemes, as they provide a way for several users to access a system. In the context of optical local area networks, various network configurations employing different multiplexing schemes have been proposed. Among these,

configurations using radio frequency or microwave frequency subcarrier multiplexing (SCM) have generated a lot of interest. SCM has emerged as a potentially important multiple access technique for future high-capacity lightwave networks.

In a system using SCM, the information signals are modulated onto different electrical carriers at radio or microwave frequencies and combined. The resulting signal is then used to intensity modulate an optical carrier. The so-called subcarriers refer to the radio (or microwave) frequency carriers as opposed to the optical carrier. At the receiver end, the optical signal is converted back to an electrical current by a photodetector. A particular signal can then be demultiplexed and demodulated using conventional methods. An example of a subcarrier multiplexed system is shown in Fig. 1.1.

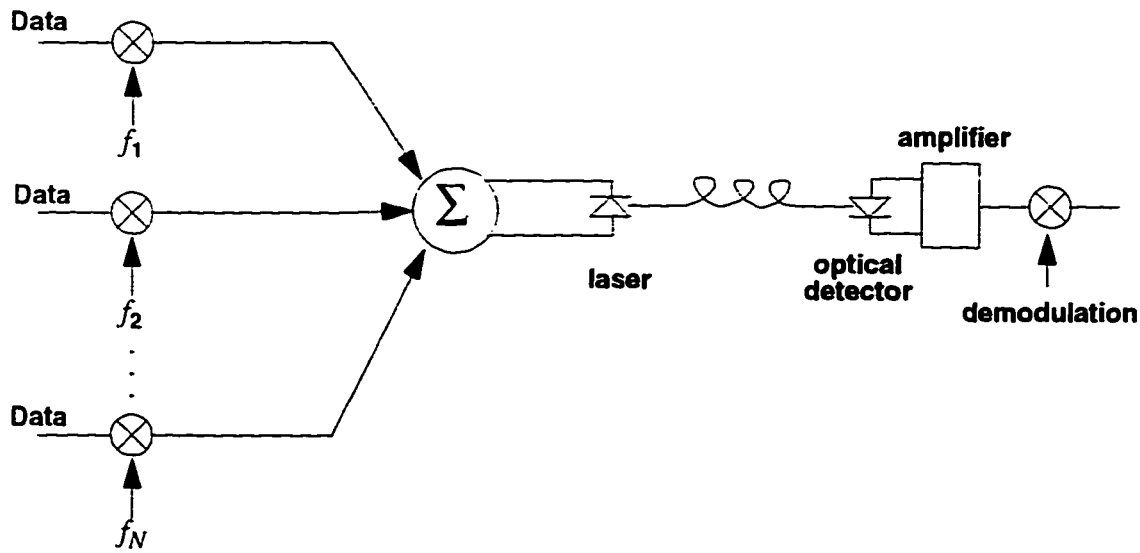


Figure 1.1 Subcarrier multiplexed system

One very attractive feature of subcarrier multiplexed systems is the independence of the different channels. This allows for great flexibility in the choice of modulation schemes, which may be digital or analog, and provides adaptability to future evolution.

There is also no need for synchronization between each channel and a master clock, as is the case for time division multiplexed systems. In addition to being flexible, SCM systems are also cost effective, as they provide a way to take advantage of the bandwidth potential of fiber optics using conventional well-established microwave techniques for which components are commercially available. The use of subcarrier multiplexing for analog and digital video distribution systems has been investigated extensively [2], [3], [4], [5].

The greatest weakness of SCM is that it is essentially an analog technique and as such requires the source output power to be a linear function of the input. Unfortunately, the laser, which is commonly used as an optical source, is an inherently nonlinear device. In multichannel applications where signals are modulated onto different subcarriers, the nonlinearity of the laser will result in interchannel interference, called intermodulation. The intermodulation distortion added to harmonic distortion also caused by the source nonlinearity and to noise in the system can significantly reduce the capacity of an SCM system.

Other multiple access techniques when applied to photonic networks have also run into problems. Time division multiple access (TDMA), for example, is used successfully on wideband point-to-point links. However in multiple access systems, TDMA is not as cost effective as SCM, because it makes inefficient use of the potential fiber bandwidth and supports too few users. In a TDMA system each receiver must receive all transmitted data

and select the appropriate bits. Since receiver sensitivity decreases with increasing bandwidth. these systems are often limited to low total data throughput rates or a small number of users [1].

WDMA or wavelength division multiple access is another technique with potential. In the optical communications literature (e.g.. [6]) it is often referred to as (optical) frequency division multiple access (FDMA) when the channel spacing is relatively close (comparable to the bit rate). Combining WDMA with coherent techniques. where the optical carrier is directly modulated with the data. opens up possibilities for networks with enormous throughputs. In a coherent WDMA system. data would be modulated onto the laser output via an external modulator. To maintain the high signal-to-noise ratio (SNR) required at all times. excellent carrier frequency control would be essential. At the receiver. the desired signal would be demodulated using an optical filter based on Fabry-Perot interferometric techniques. then detected. Most currently available lasers have a relatively wide linewidth whose center frequency tends to drift. These characteristics can cause unacceptable interference in coherent WDMA systems. Lasers capable of providing the rigid carrier frequency control necessary to successfully realize coherent WDMA networks have been difficult to achieve. They are neither inexpensive. nor readily available. and thus not suitable for use in networks. Other methods for achieving tight carrier frequency control. such as external modulation of the light output of a laser are beginning to be used in long haul communication applications. but are still much too expensive for local access applications.

For the moment, the technology necessary to achieve the high-capacity networks shown to be possible with single-mode optical fibers, is not yet available at a reasonable cost. However an intermediate solution has been put forward, to make the best possible use of the technology that is available. Vannucci has proposed the idea of a hybrid network where two multiple access techniques are combined in such a way that the strengths of each scheme mitigate the shortcomings of the other [7]. He has suggested a network where the use of FDMA is combined with code division multiple access (CDMA). The special properties of CDMA are used to offset the undesirable effects caused by the non-ideal lasers.

Code division multiple access is well-known for being resistant to interference. It has traditionally been used to fight interference in systems caused by jamming, interference from other users, and self-interference due to multipath propagation. The interference suppressing effect of CDMA can also be used in optical systems to help ease problems there.

In conventional CDMA systems, users share a common channel. The data they transmit is distinguished by a code unique to each user, which is superimposed on the data. In the most common form of CDMA, direct sequence spread spectrum multiple access (DS-SSMA), the code consists of a pseudo-random sequence (also called a pseudo-noise or PN-sequence), whose symbol rate is much greater than the data symbol rate. This has the effect of spreading the bandwidth of the data signal.

While CDMA is interference resistant, it is also spectrally inefficient due to the spreading of the signal bandwidths, resulting in low throughputs [7]. In Vannucci's hybrid network however, the combination of FDMA with CDMA allows for more effective use of the available bandwidth. Vannucci presents implementation details of the CDM/FDMA system and demonstrates feasibility with the current lightwave technology.

Farideh Khaleghi has taken up the idea of a hybrid network and applied it to SCM systems. In her thesis [8], she has proposed a network configuration where CDMA is added to a subcarrier multiplexed system. Khaleghi suggests that the interference suppressing feature of code division multiplexing can also be used to help combat the distortions caused by laser nonlinearity in SCM networks.

The combined SCM/CDMA network used by Khaleghi is shown in Fig. 1.2. In the network each user is assigned a particular code sequence, c_i, and subcarrier frequency, f_j, where the pair (c_i, f_j) is unique with respect to every other user. Data is combined with a code sequence then modulated onto a subcarrier. Various subcarriers in a cluster are combined and the resulting signal used to intensity modulate a light source. A cluster represents all the users sharing a particular optical source. The light emitted from the source is broadcast over a passive optical star coupler. At the receiving end of the network, only the intended receiver is able to correctly demodulate the detected signal. Every receiver is matched to a pair (c_i, f_j). The transmitter must generate the correct code sequence and the

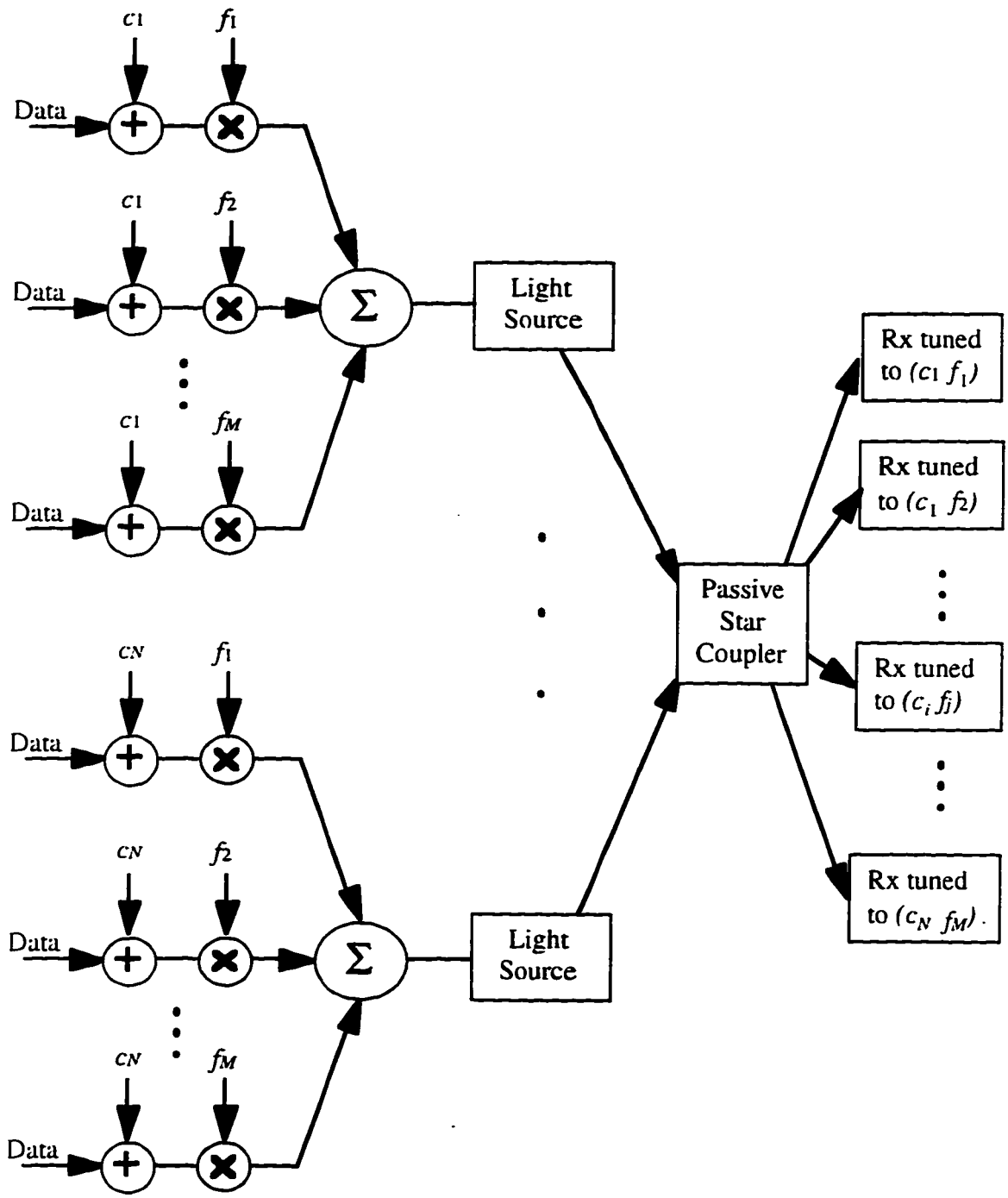


Figure 1.2 SCM/CDMA optical network

subcarrier frequency of the receiver to ensure that the data will reach its destination. More commonly, it is the transmitters that are preset to a code/frequency pair, and each receiver must tune to the correct frequency and sequence to receive the desired data.

Khaleghi provides a theoretical analysis (see also [9]) which demonstrates that, in the proposed hybrid network, the intermodulation products (IMPs) and harmonics produced by the lasers' nonlinearity, will have an effect on the network performance similar to the interference caused by non-matching sequences. She shows that this desirable effect is achieved by using spreading sequences for which the shift-and-add property holds.

To confirm the network performance predicted by the theoretical analysis, Khaleghi has set up the proposed network in the laboratory. She provides the results of the experiment in her thesis (see also [9]). However, the experiment does not consider the laser nonlinearity, it merely confirms that the hybrid configuration has potential for LAN applications. Indeed, in the experimental realization of the network, light emitting diodes (LEDs) are used rather than lasers. Since LEDs are basically linear devices, they would not cause the same degradation in the network performance as using lasers would.

Khaleghi also presents a theoretical performance study of the network. In it, she compares the use of different sequences and different methods for calculating the network performance. However, again she does not consider nonlinear distortion, arguing that, based on the theoretical analysis, the nonlinear effects would merely depend on the length of the sequence used.

In this thesis. the performance of the hybrid CDM/SCM network will be analyzed to determine the extent to which the use of spreading sequences can suppress the lasers' nonlinear distortion effects. Specifically we will compare the performance of the hybrid network to that of a subcarrier multiplexed optical network. To this effect, a simulation of the networks was created using the C programming language. The simulation takes into account most sources of performance degradation that would be present in a 'real' system. including of course the laser nonlinearity effects.

1.2 Contributions

In this thesis we attempt to verify the performance benefit of the hybrid SCM-CDMA optical network over optical networks using SCM alone. We have programmed a simulation to fulfil this objective.

The hybrid optical network was first designed based on the work by Khaleghi and a detailed block diagram developed. Each element in the network was modeled and its simulated performance verified. Particular attention was paid to the models of three key network elements. namely the laser. the optical receiver. and the SAW filter used to despread signals detected at the receiver.

The different network elements were combined into a single simulation of an entire local area network. The simulation program is able to handle networks with up to ten users. but could easily be modified for larger networks. The user of the simulation can also specify the length of the simulation in terms of the number of data bits used. the length and initial

state of the code sequences used to spread network users' data, the subcarrier frequencies, as well as the biasing of the signals at the input to the laser. This last parameter provides a way for the user to easily set the signal to noise ratio. The simulation produces the bit error rate of the simulated data at its output.

Preliminary results from our simulation indicate that a performance benefit may be obtained by using both SCM and CDMA in optical local area networks. This provides a tentative confirmation of Khaleghi's theoretical analysis in [8].

1.3 Outline of the Thesis

The network, its components and topology will be reviewed in Chapters 2 and 3. In Chapter 2, the optical core of the network is discussed. In particular, a review of the major optical components—laser and photodetector—is provided. System design choices, such as the LAN topology, are explained.

Chapter 3 consists of four sections. The first is a brief review of spread spectrum and CDMA. Next, popular spread spectrum sequences are presented. In the third and fourth sections, the performance of CDMA in linear and non-linear systems is analyzed.

Chapter 4 includes a detailed description of the implementation of the network. The models of the network components upon which the simulation program is based are explained.

In Chapter 5 the performance of the hybrid CDMA-SCM optical network is analyzed. In the first section Khaleghi's theoretical analysis is discussed. Simulation results are presented and discussed in the next section.

In Chapter 6, conclusions reached are outlined along with suggestions for further research.

Chapter 2: Optical LANs - Review of Components and Topologies

In order to better understand the hybrid SCM/CDMA network discussed in this thesis, this chapter provides a review of the general optical communications system and its components. All of the material in this chapter was taken from the literature [10], [11], [6], [12].

The function of any communications system is to convey an information signal from the information source over a transmission medium (the information channel) to its destination. In a fiber optic communications system of the kind we are studying, three components—an optical source, optical fiber, and an optical detector—are responsible for the transmission of the information from source to destination. Surrounding the optical components at each end is a layer of electrical signal processing. This "generic" system is shown in Figure 2.1.

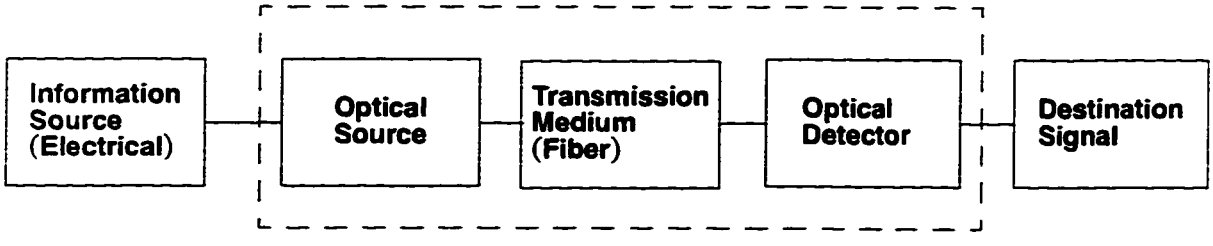


Figure 2.1 "Generic" optical communications system

For a communications system to be reliable and secure, it is necessary to chose all components, especially those responsible for transmission, with care to ensure that their individual performance as far as possible enhances rather than degrades the overall system performance.

2.1 Review of Optical Components

2.1.1 Transmission Medium

The transmission medium in a communications system refers to the path taken by the information signal from the transmitter to the receiver. The choice of a transmission medium is based on several characteristics. In the case of optical communications systems, although any medium which transmits light could be used in theory, optical fibers are the medium of choice.

For the purpose of this thesis, it will be assumed that optical fibers are ideal in all characteristics except attenuation. In other words it will be assumed that transmission of signals through fibers is linear and non-dispersive. For a brief review of the different types of optical fibers used in telecommunications networks and their characteristics, see Appendix B.

2.1.2 Light Sources

In fiber optic systems, the optical source generates the light beams which carry the information. It is often considered the active element in optical communications systems. The basic function of the optical source is to convert electrical energy in the form of a

current into optical energy (light) in an efficient manner which allows the light output to be effectively launched or coupled into the optical fiber. Three main types of optical light source are available. They are:

a) wideband continuous spectra sources (incandescent lamps)

b) monochromatic incoherent sources (light emitting diodes)

c) monochromatic coherent sources (lasers)

Historically, the most powerful coherent light sources were necessary to overcome the severe attenuation and dispersion of early fibers. For this reason, gas lasers (helium-neon) were initially used.

The optical sources most commonly used in optical fiber systems today are the light emitting diode (LED) and the semiconductor injection laser (i.e., the laser diode). These two devices largely fulfil the major requirements for optical fiber emitters. Their small size is compatible with the small diameters of fibers, and their solid-state structure and low power requirements are compatible with modern solid-state electronics.

LEDs are generally low power devices and are able to couple lower optical power into the fiber (of the order of a microwatt), and they have a smaller modulation bandwidth than lasers. Unlike LEDs, lasers have a high gain due to stimulated emission and more output (several mW) can be coupled into the fiber. The first generation of optical communications systems operated in the 0.8 μm transmission window of fibers. Both LEDs and lasers were utilized with multimode step-index fibers. Because of smaller bandwidth

and low coupling power, the LED is not suitable for long distance wideband transmission. With their superior performance (power, bandwidth and transmission distance) semiconductor lasers became more popular. After the development of single-mode fibers, which offer extremely low dispersion in the second transmission window around 1.3 μm, laser diodes became very important optical sources, especially in high speed long haul applications. From this point onward our discussion will focus on lasers, their operation and performance.

To understand how a laser diode generates light it is necessary to review some fundamental atomic concepts. The interaction of light with matter takes place in discrete energy packets or quanta, called photons. Quantum theory suggests that the electrons in atoms exist only in certain discrete energy states. They move from one state to another through the absorption or emission of light. The frequency of the absorbed or emitted radiation, f, is related to the difference in energy E between the higher energy state E_2 and the lower energy state E_1, by the expression

$$E = E_2 - E_1 = h \cdot f \quad , \tag{2.1}$$

where h = 6.626E–34 J*s is Planck's constant. Laser action is the result of three key processes: photon absorption, spontaneous photon emission, and stimulated photon emission. These three processes are illustrated by the two state energy diagrams of Figure 2.2. Normally an electron is in the ground state (here the lower energy state E_1). When a photon of energy $(E_2 - E_1)$ is incident on the electron it may be excited to the

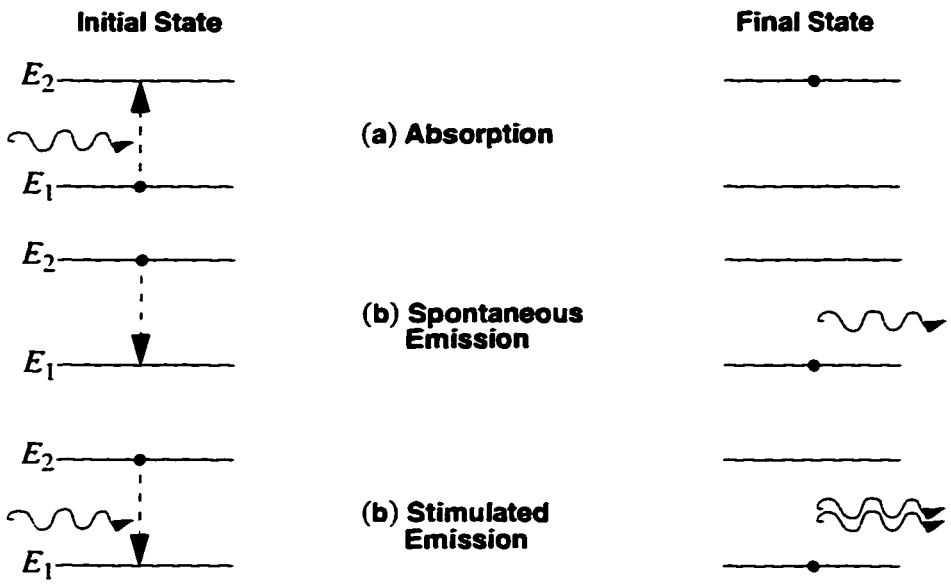


Figure 2.2 Energy state diagram showing: (a) absorption (b) spontaneous emission (c) stimulated emission [10]

higher energy state through the absorption of the photon. The electron may return to its ground state through the emission of a photon of energy hf. The transition to the lower energy state with the accompanying light emission may occur in one of two ways:

(i) the electron may randomly emit a photon and move to the lower energy state; this is called spontaneous emission.

(ii) the electron in the excited state may interact with a photon of energy $(E_2 - E_1)$ causing it fall to the lower energy state with the generation of a second photon; this is referred to as stimulated emission.

Spontaneous emission is the basic mechanism by which LEDs generate light. The random nature of the emitted photons results in incoherent radiation. Lasers obtain their special characteristics as optical sources from stimulated emission. Stimulated emission produces coherent radiation, as the released photons are all in phase, and it allows optical amplification to occur.

Returning to fundamental physics, it was shown by Einstein that the rates of the three transition processes of absorption, spontaneous emission, and stimulated emission are mathematically related. Under conditions of thermal equilibrium most electrons are at the ground state. Most photons are absorbed producing mostly spontaneous emission. Stimulated emission can only be achieved once the electrons in the excited or upper state exceed those in the lower state, a condition known as population inversion. Population inversion can be achieved by exciting electrons into the upper state using various "pumping" techniques. With semiconductor lasers, electrons are injected into the material at the device contacts via an electrical current to fill the lower states of the conduction band.

The light output of the laser in response to different current levels is shown in Figure 2.3. The solid line represents an ideal laser characteristic. The dashed line shows the laser's threshold current. Below the threshold current the laser emits little light and this light arises mainly from spontaneous emission. The laser in effect behaves like a LED in this region. However above the threshold the light power increases sharply even for small increases in current. Here the light is generated throughout stimulated emission with the laser acting as an amplifier of light.

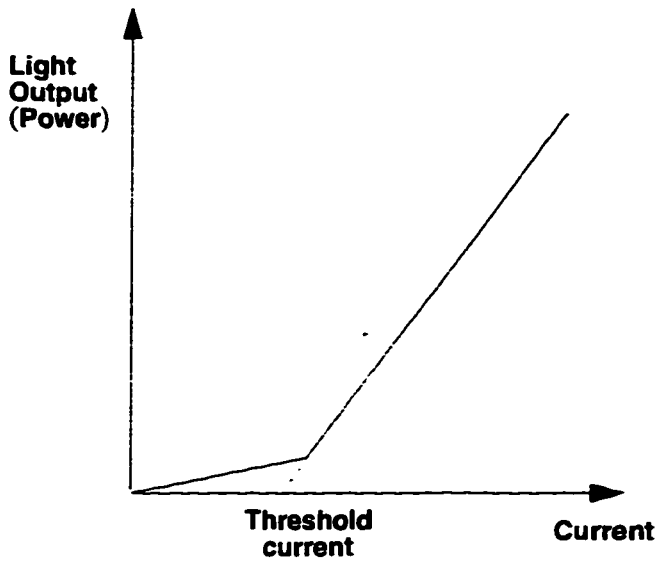


Figure 2.3 Ideal light output against current characteristic for an injection laser [10].

Hence, in addition to achieving population inversion, the necessary condition for stimulated emission, conditions must also be in place for optical amplification to occur. Light amplification occurs when photons colliding with atoms in the excited state cause the stimulated emission of photons, which in turn cause the release of more photons, generating a stream of coherent radiation. This process can only be sustained if some of the emitted light is retained within the laser medium and the conditions for coherence (stimulated emission) are maintained. Light can be retained in the laser by placing mirrors at each end of the laser cavity in which light is generated as shown in Figure 2.4. Positive feedback is created within the optical cavity by the reflection of photons at either end of the laser medium. By making one of the mirrors only partially reflecting, some of the generated light

can escape the laser medium and be launched into an optical fiber. Within a semiconductor laser, the optical cavity where the interactions between photons and electrons occur is referred to as the laser's active region; it is a subset of the total device.

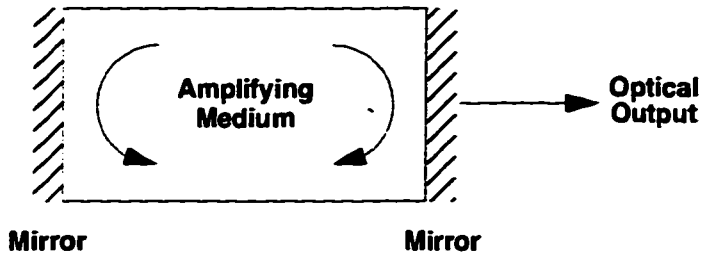


Figure 2.4 The semiconductor laser structure with reflecting mirrors at the ends of the cavity [10].

Lasers have been the subject of study since the beginnings of fiber optic communications. Due to their crucial role in this important type of communications system, researchers have sought to understand and predict their performance characteristics.

A set of rate equations for electron and photon density in the active layer of lasers have been developed, which describe the transient behaviour of semiconductor lasers. These equations for the electron (or carrier) density N and the photon density S are:

$$\frac{dN}{dt} = \frac{I_A}{V'} - \frac{N}{\tau_{sp}} - g\,(N - N_0)\,(1 - S)\,S \tag{2.2}$$

$$\frac{dS}{dt} = \Gamma g\,(N - N_0)\,(1 - \varepsilon S)\,S - \frac{S}{\tau_p} + \Gamma\beta\frac{N}{\tau_{sp}}\,, \tag{2.3}$$

where N is the electron density averaged over the volume of the laser's active region, I_A is the current injected into the active region, V' is the volume of the active region times electronic charge. τ_{sp} is the spontaneous recombination lifetime of the carriers, g is the

optical gain coefficient, S is the photon density. N_0 is the carrier density for transparency, Γ is the optical confinement factor. τ_p is the photon lifetime, β is the spontaneous emission factor, and ε is the gain damping coefficient.

These equations are valid under the following assumptions [12]:

a) the laser is operating in a single mode above threshold,

b) it has an ideal cavity with homogeneous population inversion, and

c) the gain coefficient is a linear function of in injected carrier density N above a minimal value N_0.

The terms of (2.2) and (2.3) may be balanced, and stable operation of the laser achieved by taking into account all factors which affect the number of electrons and photons in the laser's active region. These factors are represented by the different terms in the rate equations.

In (2.2), the quantity $\frac{dN}{dt}$ represents the rate of change of carrier density. The term $\frac{dN}{dt}$ is affected by the rate at which new electrons are injected into the active region ($\frac{I_A}{V}$) and the rate of loss of carriers through spontaneous emission ($\frac{N}{\tau_{sp}}$) and stimulated emission $g(N - N_0)(1 - \varepsilon S) S$. This last term of (2.2) represents the net stimulated emission. The term N_0 is the carrier density at which the stimulated emission is equal to stimulated absorption. Net stimulated emission only occurs for $N > N_0$. The optical-field-dependent gain compression, ε, damps the laser transient response.

The term $\frac{dS}{dt}$ in (2.3) for its part represents the rate of change in photon density. This value depends on the rate at which photons are generated and lost through stimulated emission and absorption respectively in the active region, $\Gamma g\,(N - N_0)\,(1 - \varepsilon S)\,S$. Only photons generated inside the active region contribute to the laser's light emission process. The quantity Γ, the optical confinement factor, represents the proportion of photons in the optical active region as compared to the volume of the electrical active region. The optical confinement factor arises from the difference in volume of these two regions. The photon density also changes when photons are spontaneously emitted into the lasing mode ($\Gamma\beta\frac{N}{\tau_{sp}}$). The spontaneous emission factor, β, is the probability that a spontaneously emitted photon happens to be coherent with stimulated emission. Finally, $\frac{dS}{dt}$ is decreased by the second term of (2.3), which largely represents the light output from the laser. The phenomena described by the laser rate equations can be represented pictorially as shown in Figure 2.5 [13].

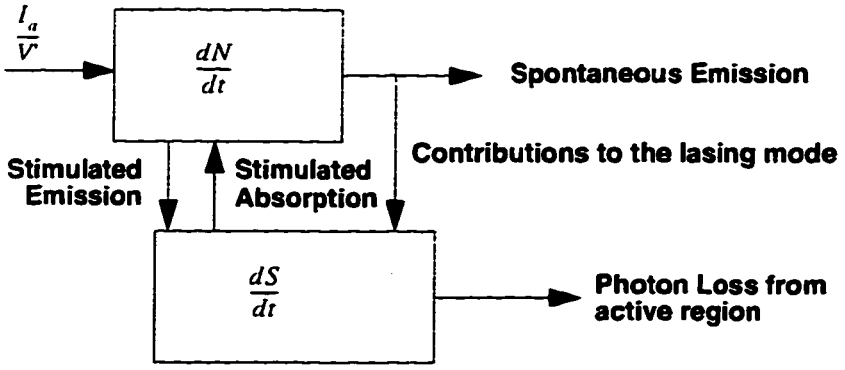


Figure 2.5 Changes in electron and photon population in the light generation of a laser diode as given by the rate equations [13].

Lasers as faithful transmitters of information are less than ideal: they are both nonlinear and noisy. The non-linear characteristics of laser have been investigated by several researchers, and two types of non-linearity are described in the literature.

The laser exhibits a memoryless non-linearity in its light output in response to injected current, as can be observed from its light-to-current characteristic (Figure 2.3).

Semiconductor lasers also have non-linear characteristics which are due to the stimulated emission process, which is formed by the product of the photon and electron densities.

Further non-linearity in the laser's output may be introduced in the laser's output through clipping. Clipping occurs when the injected signal current temporarily goes below the threshold current; at the points where this happens, the light output will be clipped at zero, no longer reflecting changes in the input current. However, contrary to the intrinsic nonlinearities, clipping can be avoided by proper biasing of the input.

In addition to its nonlinear characteristics, the laser also suffers from several noise sources: these are:

a) phase or frequency noise,

b) instabilities in operation such as kinks in the light output against current characteristic,

c) reflection of light back into the laser,

d) mode partition noise.

It is possible to reduce, if not eliminate the last three noise sources. However, phase noise is an intrinsic property of all laser types. It results from the discrete and random spontaneous or stimulated transitions which cause intensity fluctuations in the optical emission and are an inevitable aspect of laser operation.

It has been observed that the spectral density of this phase or frequency noise has a characteristic represented by $1/f$ to $1/f^2$ up to a frequency of around 1 MHz. Above 1 MHz the noise spectrum is flat or white. This white noise component is a problem in lasers, as over time the phase executes a random walk away from the value it would have had in the absence of spontaneous emission.

Fluctuations in the amplitude or intensity of the output of semiconductor lasers also leads to optical intensity noise. These fluctuations may be caused by variations in temperature or they result from spontaneous emission in the laser's output. The random intensity fluctuations create a noise source called relative intensity noise (RIN). RIN is described by the following equation:

$$\text{RIN} = \langle \Delta s^2 \rangle / \langle s \rangle^2, \tag{2.4}$$

where $\langle s \rangle$ is the average laser light intensity and $\langle \Delta s^2 \rangle$ is the mean-square intensity fluctuation of the light output.

As mentioned earlier, the laser's RIN is caused by the discrete nature of electrons. It manifests itself in each of the terms of the laser rate equations, which are not deterministic, but are actually the average of a stochastic process that is modelled as shot noise. The different components of the rate equations combine to give the relative intensity noise in the light output.

2.1.3 Optical Detectors

At the end of the transmission medium of any communications system is a receiver, whose function is to return the information signal to some desired format. Optical receivers have an optical detector at their front end followed by electrical amplification and signal processing. Optical detectors perform two roles in optical receivers: they detect the optical signal as their name suggests, and convert it from the optical domain into the electrical domain. The most commonly used photodetectors are semiconductor photodiodes.

Semiconductor photodiodes use intrinsic absorption to convert optical radiation into electrical current. In effect they act as LEDs in reverse. Photons incident on the photodiode excite electrons from the valence to the conduction band in the semiconductor generating an electrical current. Three types of photodiode are the p-n, the p-i-n, and the avalanche photodiode.

In p-n photodiodes, a reverse bias at the junction of positively and negatively doped semiconductor regions creates a potential barrier. Current can only flow when, through the absorption of photons, electrons are raised across the bandgap into the conduction band and are thus freed to move.

The p-i-n diodes are very similar to p-n diodes except that they have a wide intrinsic semiconductor layer between the p and n regions. The intrinsic layer improves the efficiency and speed of the p-i-n's relative to the p-n diodes, making them by far the more popular of the two devices.

The other most commonly used photodiode is the avalanche photodiode (APD). It is a semiconductor junction detector with internal gain. The gain gives APDs an advantage over p-i-n's in detecting very faint incoming light. APDs have more sophisticated structures than p-i-n's in order to create an extremely high electrical field region. The primary carriers, generated in the depletion region by the absorption of photons, can acquire sufficient energy in this high field region to excite new carriers. This process is known as impact ionization. When the new carriers themselves can acquire enough energy to excite more carriers and so on, avalanche multiplication has been initiated. Historically, APDs have required much higher reverse bias voltages than p-i-n diodes for avalanche multiplication to occur. The multiplication factor M is a measure of the internal gain provided by an APD; it is defined as:

$$M = \frac{I}{I_p}, \tag{2.5}$$

where I is the total output current at the operating voltage, and I_p is the initial primary photocurrent before carrier multiplication occurs.

In addition to requiring higher bias voltages, other drawbacks APDs suffer from are:

(i) the random nature of the gain mechanism which leads to an additional noise contribution

(ii) fabrication difficulties due to their more complex structure

(iii)variation of the multiplication factor with temperature.

Important properties of all detectors are responsivity, spectral response, and rise time. The responsivity R is the ratio of the output current of the detector to its optical input power:

$$R = \frac{I}{P} . \tag{2.6}$$

The spectral response refers to the curve of the detector responsivity as a function of wavelength. Because of the rapid change in responsivity with wavelength, different detectors must be used in the windows of the optical spectrum where fiber losses are low.

The rise time is the time for the detector output current to change from 10 to 90% of its final value when the optical input power is a step function. The detector's bandwidth is related to the rise time. It is desirable for the detector to have as small a rise time as possible.

A quantity related to a photodetector's responsivity, is its quantum efficiency η. The quantum efficiency is defined as the fraction of incident photons which are absorbed by the photodetector and generate electrons which are collected at the detector terminals. The incident optical power and the output current are related through the relationship:

$$I = \eta Pe/hf . \tag{2.7}$$

where e is the charge on an electron and $h \cdot f$ is the photon energy. The responsivity is related to the quantum efficiency as follows:

$$R = \eta e/hf \ . \tag{2.8}$$

Photodiodes suffer from a number of noise sources. Even when no light is incident on them, photodiodes generate some current due to background radiation. This is referred to as the dark current. In many photodiodes the dark current can be reduced to such an extent that is insignificant compared to other noise sources.

In common with other optical devices, the current generated by photodiodes exhibits fluctuations resulting from the statistical nature of the quantum detection process. The result of these fluctuations is shot noise.

As mentioned earlier, APDs suffers from extra noise due to the random nature of their gain mechanism.

2.2 Optical Local Area Network Topologies

The key devices used in optical communications systems have been reviewed, now it is time to examine systems which use these components.

The most basic communications system is a unidirectional point-to-point link between two users using an optical transmitter at one end, an optical receiver at the other end, and optical fiber as the transmission medium. Generally it is desirable to connect many more than just two users. To accommodate multiple users over a single point-to-point fiber

connection a variety of different multiplexing methods can be used. In today's fiber optic communications network time division multiplexing is most frequently used. Wavelength division multiplexing is now beginning to be introduced. Beyond the basic point-to-point link, it is necessary to be able to connect not only many users, but also many users at diverse locations. The ability to distribute optical signals among multiple locations is thus essential.

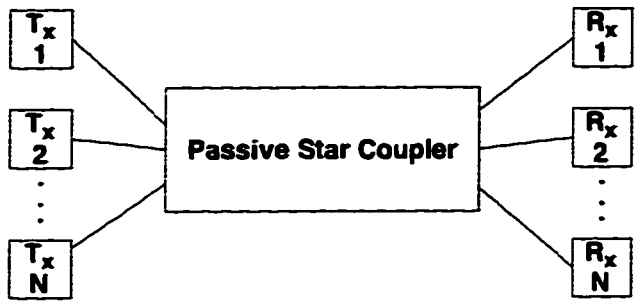
Local area networks represent a possibly significant application for fiber optic technology due to the huge potential market they represent. A LAN may be characterized as a (data) network within a local environment (typically between 50 m and 10 km according to [10]). It may provide interconnection on either a random access or a switched circuit basis. Many services may eventually be provided over photonic LANs, including high-capacity interconnection of computers, video conferencing, multimedia services, as well as connection to the Internet.

A typical optical fiber has a potential capacity of about 50 THz in the second and third transmission windows. A challenge facing telecommunications services providers ever since the beginnings of fiber optic communications has been how to best take advantage of the enormous bandwidth of fibers. This problem has been particularly difficult to resolve in local access networks and LANs. It has been found that fiber cannot simply replace copper to increase the capacity of a wire-based LAN. While fiber-based systems offer readily available transmission bandwidth, their traffic capacity may be limited by fundamental power considerations.

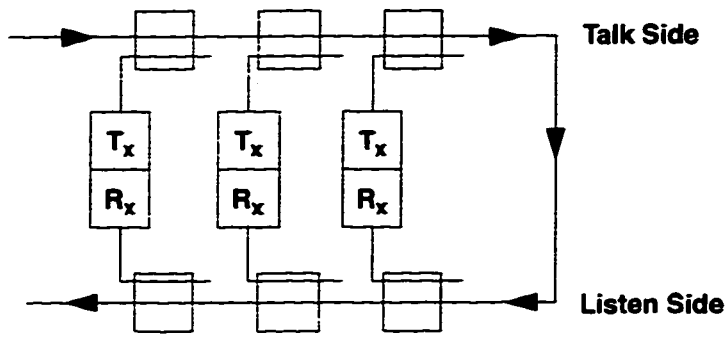
An ideal optical receiver on a point-to-point link, limited only by shot noise will require a minimum energy E to detect a transmitted bit. Operation of the link at a rate B requires a power level EB at the receiver. For a transmitted power P, the maximum power-limited transmission rate is P/E. Similar limitations appear when these energy considerations are applied to LANs, however the severity of the power limitation is dependent on the topology used. The three basic LAN topologies used are the star, the bus, and the ring, shown in Figure 2.6.

The optical star forms a central hub in a star network, which may be either active or passive. In passive operation, a star coupler accepts input signals from all transmitters and broadcasts them to all receivers. Each receiver selects the signal intended for it. A passive star divides the input power from all transmitters equally to all outputs. Assuming an N-user network with an ideal coupler and lossless fiber, if a transmitter has power P, the recipient node will receive a signal of power P/N.

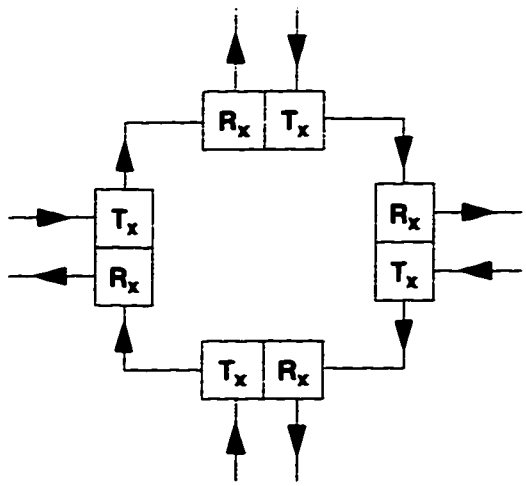
In a unidirectional bus topology, each user's traffic is input on the "talk side" of the bus and removed on the "listen side" via two sets of directional couplers. The bus is popular for use in LANs using coaxial cable, however it is impractical with fiber because it wastes power. In an N-user network with ideal couplers, only P/N^2 useful power is delivered to each receiver. The poor power efficiency of the bus stems from losses associated with coupling power into the "talk side" of the bus; the fraction of power coupled to each coupler is less than one. Since this signal must then be distributed among N users, it is attenuated by $1/N$, resulting in an overall transmitter-to-receiver attenuation of $1/N^2$. In addition to



a) Passive star



b) Unidirectional bus [14]



c) Ring [14]

Figure 2.6 LAN topologies

poor power efficiency, buses also suffer from an uneven power distribution between connected receivers. The main drawback of this is that the optical receivers must then have a large dynamic range.

A ring is usually a closed loop of point-to-point links connected in series. Each ring node contains an optical transmitter and a receiver. The nodes' function is that of active regenerators. In a ring network, the power travels from any one transmitter to a single receiver. There is no power sharing as in the case of bus and star networks. However in rings the bandwidth is electronically processed at each node. This limits the bandwidth to the electronic processing capability of a single node. Currently electronic processing is limited to rates of 10 Gb/s or less. This limitation imposed by the electronics at the ring's nodes is often referred to as the electronic bottleneck. At 10 Gb/s, it may be argued that the electronic bottleneck will not slow down most users, however requiring 10 Gb/s electronic processing capability at every node is very costly. Cost is a critical concern in LANs and local access networks, where relatively few users must share the cost of the network.

The disadvantages of rings and buses can be overcome. The electronic bottleneck of rings can be avoided through solutions such as multihop network architectures or the use of wavelength division multiplexing. Even with WDM which has recently been gaining popularity and where the requirement that all nodes have the same electronic processing capability is removed, rings remain a costly network architecture. The power division problem of buses may be compensated for with optical amplifiers. However the usefulness of optical amplifiers in LANs is limited by two factors: maximum output power and

internal noise generation. These considerations combined with simplicity of implementation make the passive star topology a popular choice for optical LAN applications. It is also the topology considered in the work done for this thesis.

Chapter 3: Description of CDMA Networks and their Performance

The objective of this chapter is to provide a theoretical analysis of the performance of CDMA networks in the presence of non-linearities. This analysis follows the approach taken by Khaleghi [8],[9]. The background material introducing CDMA and its performance characteristics is taken from the literature [15], [16], [17], [18].

The performance of many real world communications channels can be accurately determined using a stationary additive white Gaussian noise (AWGN) model. For these channels, transmitters are selected to be relatively efficient in their user of power and bandwidth, and demodulators are designed to yield a minimum bit error probability for the given transmitted signal in AWGN.

However, some important types of communication channels do not fit the stationary AWGN model. One example is a transmission channel where there are multiple propagation paths between the transmitter and receiver. In this case the receiver interferes with itself via a delayed reception of its own signal.

Subcarrier multiplexed systems also experience degradations which cannot be modelled as stationary AWGN. The performance of SCM systems is very sensitive to non-linearities in the systems' elements such as those inherent to semiconductor lasers, since

SCM is basically an analog technique. Laser diode non-linearities appear primarily as intermodulation distortion. As was discussed in the previous chapter, lasers generate noise signals at multiple frequencies which may interfere with the message signals.

Spread spectrum is a modulation and demodulation technique which is commonly used for the purpose of combatting or suppressing the detrimental effects of interference due to jamming, interference arising from other users of the channel, and self-interference due to multipath propagation. In the same vein, it can also be used to mitigate the harmful effects of laser intermodulation distortion. In addition, spread spectrum signals are also used to achieve privacy in the presence of other listeners and to transmit signals at very low power making their detection by unintended listeners unlikely.

The performance of a generic multiple access spread spectrum system is discussed in Section 3.3 followed by the analysis of how spread spectrum can alleviate the effects of non-linearities in Section 3.4. First however, an introduction to spread spectrum and code division multiple access is provided in Section 3.1 and a review of commonly used pseudo-random codes or sequences is presented in Section 3.2.

3.1 Introduction to Spread Spectrum and CDMA

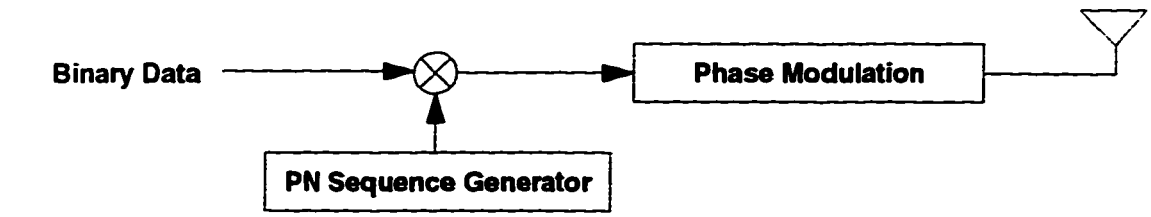
In most networks, multiple-access techniques are used to transmit the signals of many users over a shared transmission medium. In Chapter 1 some of the techniques were briefly compared. Code division multiple access techniques rely primarily on superimposing different sequences or codes on transmitted data to provide the multiple-

access capability and—unlike traditional time- and frequency division multiple access—don't require precise time or frequency coordination between transmitters in the system. In a CDMA system, a user superimposes a pseudo-random sequence onto his message signal. The desired receiver can recover the transmitted information intended for it by knowing the code used by the corresponding transmitter. CDMA is an application of spread spectrum. In fact it is sometimes referred to as spread spectrum multiple access (SSMA).

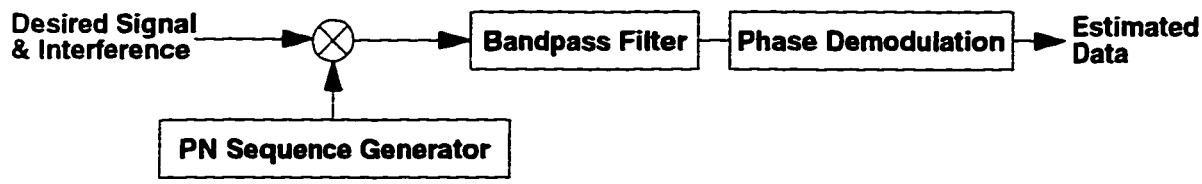
Spread spectrum is a modulation and demodulation technique which is distinguished by the fact that the transmission bandwidth is much greater than the minimum bandwidth required to transmit the information.

One method for spreading the spectrum of an information signal is to modulate the signal a second time using a very wideband spreading signal. When the bandwidth is spread by direct modulation of a data modulated carrier with a spreading signal or code, this is called direct-sequence (DS) spread spectrum. With another common form of spread spectrum, the spreading code is used to control the frequency of transmission of the data modulated carrier, thus indirectly modulating the information signal by the spreading code. The material in this thesis only presents information on direct-sequence spread spectrum, as it is probably the most commonly used type of spread spectrum.

The simplest form of DS spread spectrum uses binary phase shift keying (BPSK) as the spreading modulation. This is shown in Figure 3.1. At the transmitter the information signal in the form of a bit stream is multiplied by the spreading function and the combined signal is phase modulated and sent over the combined transmission medium. At the



a) Transmitter



b) Receiver

Figure 3.1 DS-spread spectrum system

receiver. the signal is "despread" and demodulated. Despreading is achieved by correlation of the received signal with the spreading function. The spreading signal is chosen to have properties which facilitate the demodulation of the transmitted signal by the intended receiver. and which make demodulation by an unintended receiver as difficult as possible. In [15] it is shown that this may be achieved by choosing sets of spreading sequences with the properties such that.

(i) for each sequence x. the magnitude of the autocorrelation is small for $1 \leq l \leq N-1$

(ii) for each pair of sequences x and y, the magnitude of the cross-correlation $\theta_{xy}(l)$ is small for all l.

For a pair of sequences x and y. with period N. the periodic cross-correlation is defined as follows:

$$\theta_{x,y}(l) = \sum_{i=0}^{N-1} x_i y_{i+l} \quad (3.1)$$

The periodic autocorrelation of a sequence x is simply $\theta_{x,x}(\cdot)$.

The properties of spreading codes are described in greater detail in Appendix C.

3.2 Binary Maximal Length and Gold Sequences

This section examines the properties of a few popular types of binary sequences. The sequences that have received the most attention in the literature are the binary maximal-length linear feedback shift register sequences, commonly referred to as m-sequences. As their name suggests, m-sequences are those sequences with the maximum possible period $N = 2^n - 1$ from an n-stage binary shift register with linear feedback. Also considered later in the section are Gold sequences, non-maximal length sequences which nevertheless have very good autocorrelation and cross-correlation properties. Other popular binary pn-sequences are reviewed in Appendix C. In this discussion, we shall restrict ourselves to binary sequences.

Let $h(x) = h_0x^n + h_1x^{n-1} + ... + h_{n-1}x + h_n$ denote a binary polynomial of degree n where $h_0 = h_n = 1$ and the other h's take on values 0 and 1. A binary sequence u is said to be generated by $h(x)$ if for all integers j,

$$h_0u_j \oplus h_1u_{j-1} \oplus h_2u_{j-1} \oplus \ ... \ \oplus h_nu_{j-n} = 0. \quad (3.2)$$

Here, $\oplus$ denotes addition modulo-2 (i.e., EXCLUSIVE-OR function). Replacing j by $j + n$ in (3.2) and using the fact that $h_0 = 1$, we obtain

$$u_{j+n} = h_n u_j \oplus h_{n-1} u_{j+1} \oplus \ldots \oplus h_1 u_{j+n-1}. \quad (3.3)$$

From this it follows that the sequence u can be generated by an n-stage binary linear feedback shift register which has a feedback tap connected to the ith cell if $h = 1$, $0 < i < N$. Since $h_n = 1$, there is always such a connection for the nth cell. For example, the shift register in Figure 3.2 corresponds to $h(x) = x^5 + x^2 + 1$.

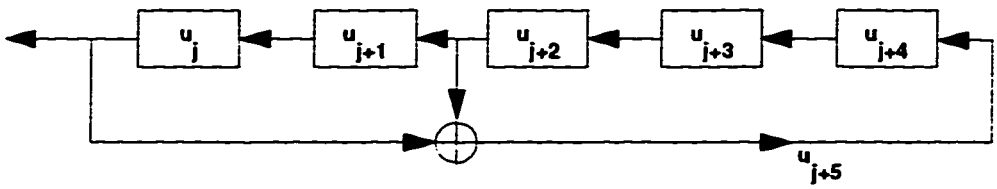


Figure 3.2 Maximal-length linear feedback register with polynomial x^5+x^2+1 [15]

A shift register can generate several different sequences, however their period cannot exceed $2^n - 1$ where n is the degree of the generator polynomial $h(x)$. If $h(x)$ is an irreducible polynomial of degree n, a sequence u generated by $h(x)$ will have this maximum period $N = 2^n - 1$ and is called an m-sequence. $h(x)$ in this case is called a primitive binary polynomial of degree n.

The m-sequences have several well-known properties: those which are useful to spread-spectrum communications are as follows:

Property I: The period of an m-sequence u is $N = 2^n - 1$

Property II: There are exactly N non-zero sequences generated by $h(x)$ and they are just the N different phases of u: namely u, Tu, T^2u, ..., $T^{N-1}u$.

Property III: Given distinct integers i and j. $0 < i, j < N$ there is a unique integer, distinct from both i and j such that $0 < k < N$ and

$$T^i u \oplus T^j u = T^k u \,. \tag{3.4}$$

A sequence of period N is an m-sequence if and only if it has Property III, which is commonly known as the shift-and-add property.

Property IV: The number of "ones" in an m-sequence is greater by one than the number of "zeros". The Hamming weight of a sequence which is really the number of ones in the sequence is as follows for an m-sequence u:

$$wt(u) = \frac{1}{2}(N+1) = 2^{n-1}. \tag{3.5}$$

Property V: Let q denote a positive integer, and consider the sequence v formed by taking every qth bit of an m-sequence u. The sequence v is said to be a decimation by q of u and will be denoted by $u[q]$. Let $gcd(a,b)$ denote the greatest common divisor of the integers a and b. If $u[q]$ is not identically zero, then $u[q]$ has period $N/gcd(N,q)$ and is generated by polynomial $\hat{h}(x)$ whose roots are the qth powers of the roots of $h(x)$.

Let u and v denote m-sequences of period $N = 2^n - 1$ and let $v = u[q]$. Many properties of the cross-correlation function follow directly from results presented earlier. We have that $\theta_{uv}(l) = \theta_{uv}(l+N)$ and $|\theta_{uv}(l)| < N$ for all l. Other useful results are

$$\sum_{l=0}^{N-1} \theta_{u,v}(l) = +1 \tag{3.6}$$

and

$$\sum_{l=0}^{N-1} [\theta_{u,v}(l)]^2 = N^2 + N - 1. \tag{3.7}$$

It is easy to find pairs of sequences with good cross-correlation properties (i.e.. for which the periodic cross-correlation function is relatively small in magnitude). However for the application that is under consideration here. large sets of sequences are often needed.

3.2.1 Gold sequences

One important class of periodic sequences which provides larger sets of sequences with good periodic cross-correlation is the class of Gold sequences. A set of Gold sequences of period $N = 2^n - 1$. consists of $N + 2$ sequences for which $\theta_c = \theta_a = t(n)$. Gold sequences have a three-valued cross-correlation with $\{-1, -t(n), t(n) - 2\}$. A set of Gold sequences can be constructed from appropriately selected m-sequences as described below.

Suppose a shift register polynomial $f(x)$ factors into $h(x)\hat{h}(x)$ where $h(x)$ and $\hat{h}(x)$ have no factors in common. Then the set of all sequences generated by $f(x)$ is just the set of sequences of the form $a \oplus b$ where a is some sequence generated by $h(x)$ and b is some sequence generated $\hat{h}(x)$. Now suppose that $h(x)$ and $\hat{h}(x)$ are two different primitive binary polynomials of degree n that generate the sequences u and v respectively. of period $N = 2^n - 1$. If y denotes a non-zero sequence generated by $f(x) = h(x)\hat{h}(x)$. then from the above and Property II of m-sequences. we find that either

$$y = T^l u \tag{3.8}$$

or

$$y = T^j v \tag{3.9}$$

or

$$y = T^i u \oplus T^j v \ , \tag{3.10}$$

where $0 < i, j < N-1$ and where $T^i u \oplus T^j v$ denotes the sequence whose kth element is $u_{i+k} \oplus v_{i+k}$. From this it follows that y is some phase of a sequence in the set $G(u, v)$ defined by

$$G(u, v) = \{u, v, u \oplus v, u \oplus Tv, u \oplus T^2 v, \ \ldots \ , u \oplus T^{N-1} v\} \ . \tag{3.11}$$

Note that $G(u, v)$ contains $N + 2 = 2^n + 1$ sequences of period N.

Considering the cross-correlation function of two distinct sequences y, $z \in G(u, v)$, it can be shown that $T^i y \oplus T^j z$ is some phase of a sequence in this set. Let $y = u \oplus T^{l_1} v$ and $z = u \oplus T^{l_2} v$. Then $T^i y \oplus T^j z$ is equal to

$$\begin{aligned} T^i\left(u \oplus T^{l_1} v\right) \oplus T^j\left(u \oplus T^{l_2} v\right) &= \left(T^i u \oplus T^{i+l_1} v\right) \oplus \left(T^j u \oplus T^{j+l_2} v\right) \\ &= (T^i u \oplus T^j u) \oplus \left(T^{i+l_1} v \oplus T^{j+l_2} v\right). \end{aligned} \tag{3.12}$$

According to Property III of m-sequences

$$\begin{aligned} T^i u \oplus T^j u &= T^k u, \\ T^{i+l_1} v \oplus T^{j+l_2} v &= T^{k'} v. \end{aligned} \tag{3.13}$$

Hence $T^i y \oplus T^j z = T^k u \oplus T^{k'} v = u \oplus T^{k+k'} v \in G(u, v)$, which is simply a phase of another sequence of the set $G(u,v)$. The above calculations show that the sum modulo-2 of two arbitrarily shifted Gold sequences is simply another Gold sequence. Therefore the shift-and-add property holds for Gold sequences.

It is of interest to note that a sequence $a = (a_0, a_1, \ldots a_{N-1})$ with values $\{0, 1\}$ can be mapped onto an equivalent sequence $\alpha = (\alpha_0, \alpha_1, \ldots, \alpha_{N-1})$ with values $\{+1, -1\}$ using $\alpha = 1 - 2a$ for $0 < i < N - 1$. For the sequence α multiplication is the equivalent operation to summation modulo-2 for sequence a. Therefore taking the product of two sequences of values $\{+1, -1\}$ will produce another Gold sequence.

3.3 Performance Evaluation of a CDMA System

Having identified and described the key parameters which determine the performance of a CDMA system, we will now provide an analysis of the performance of such a system. The approach used will seek to determine the signal-to-noise ratio at the ith receiver (SNR_i) in a K-user system. The SNR is one of the most important measures of performance that can be obtained with a reasonable amount of calculation. Phase shifts and time delays in the system as well as user data symbols will be treated as mutually independent random variables. It should be pointed out that the signal-to-noise ratio will be computed by means of probabilistic averages (expectations) with respect to the phase shifts, time delays and data symbols.

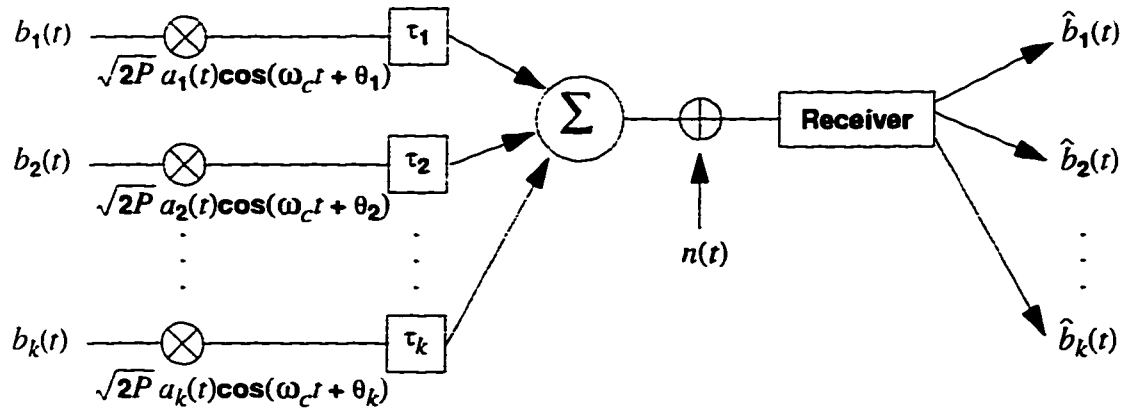


Figure 3.3 Phase-coded spread spectrum multiple access system model [19]

The model of a K-user CDMA system is shown in Figure 3.3.The kth user's data signal $b_k(t)$ is a sequence of unit amplitude, positive and negative rectangular pulses of duration T. If $(b_l^{(k)})$ is the corresponding sequence of elements of $\{+1,-1\}$,

$$b_k(t) = \sum_{l=-\infty}^{\infty} b_l^{(k)} P_T(t-lT), \tag{3.14}$$

where

$$p_T(t) = \begin{cases} 1, 0<t<T \\ 0, \text{ otherwise.} \end{cases}$$

This signal represents the kth user's binary information sequence. The kth user is assigned a code waveform $a_k(t)$, which consists of a periodic sequence unit amplitude positive and negative, rectangular pulses of duration T_c. The signal $a_k(t)$ can be expressed similarly to $b_k(t)$:

$$a_k(t) = \sum_{j=-\infty}^{\infty} a_j^{(k)} P_{T_c}(t - lT_c) . \quad (3.15)$$

The data signal $b_k(t)$ is multiplied with the code waveform $a_k(t)$ and the combined signal is then modulated onto the carrier $c_k(t)$ given by

$$c_k(t) = \sqrt{2P}\cos(\omega_c t + \theta_k) . \quad (3.16)$$

The transmitted signal is thus

$$s_k(t) = \sqrt{2P} a_k(t) b_k(t) \cos(\omega_c t + \theta_k) . \quad (3.17)$$

where θ_k represents the phase of the kth carrier. ω_c represents the common center frequency. and P represents the common signal power.

Assuming an asynchronous system. the received signal $r(t)$ in Figure 3.3 is given by

$$r(t) = n(t) + \sum_{k=1}^{K} \sqrt{2P} a_k(t - \tau_k) b_k(t - \tau_k) \cos(\omega_c t + \phi_k) . \quad (3.18)$$

where $\phi_k = \theta_k - \omega_c \tau_k$ and $n(t)$ is the channel noise process which we will assume to be a white Gaussian process with a two-sided spectral density $N_0/2$. Since we are concerned with relative phase shifts modulo 2π and relative time delays modulo T. there is no loss in generality in assuming $\theta_i = 0$ and $\tau_i = 0$. and considering only $0 < \tau_k < T$ and $0 < \theta_k < 2\pi$ for $k \neq i$.

If the received signal $r(t)$ is the input to a correlation receiver matched to $s_i(t)$. the output is:

$$Z_i = \int_0^T r(t)\, a_i(t) \cos\omega_c t dt \quad . \tag{3.19}$$

In all that follows we assume that $\omega_c \gg T^{-1}$ since the frequency response of a realistic hardware implementation of the correlation receiver is such that we can then ignore the double frequency component of $r(t)\cos\omega_c t$. The condition $\omega_c \gg T^{-1}$ is always satisfied in practical SSMA communications systems.

The output of the correlation receiver at $t = T$ is given by

$$Z_i = \sqrt{P/2}\left\{ b_0^{(i)} T + \sum_{\substack{k=1 \\ k \neq i}}^{K} [\, b_{-1}^{(k)} R_{k,i}(\tau_k) + b_0^{(k)} \hat{R}_{k,i}(\tau_k)\,] \cdot \cos\phi_k \right\} \tag{3.20}$$

$$+ \int_0^T n(t)\, a_i(t) \cos\omega_c t dt$$

where $R_{k,i}$ and $\hat{R}_{k,i}$ are the continuous-time partial cross-correlation functions defined by

$$R_{k,i}(\tau) = \int_0^{\tau} a_k(t-\tau)\, a_i(t)\, dt$$
$$\hat{R}_{k,i}(\tau) = \int_{\tau}^{T} a_k(t-\tau)\, a_i(t)\, dt \qquad (3.21)$$

for $0 < \tau < T$. If τ takes on a value between lT_c and $(l+1)T_c$, these two cross-correlation functions can be written as

$$R_{k,i}(\tau) = C_{k,i}(l-N)\,T_c + [C_{k,i}(l+1-N) - C_{k,i}(l-N)]\,(\tau - lT_c) \qquad (3.22)$$

$$\hat{R}_{k,i}(\tau) = C_{k,i}(l)\,T_c + [C_{k,i}(l+1) - C(l)]\,(\tau - lT_c) \qquad (3.23)$$

where $C_{k,i}(l)$ is the aperiodic cross-correlation function defined in (C.20) of Appendix C.

The desired signal component of Z_i is $\sqrt{P/2}Tb_0^{(i)}$, where only $b_0^{(i)} = +1$ is considered here because of the symmetry involved. The variance of the interference and noise component of Z_i meanwhile is

$$\begin{aligned}\mathrm{Var}\{Z_i\} &= \left(\frac{P}{4T}\right)\sum_{\substack{k=1\\k\neq i}}^{K}\int_0^T \left(R_{k,i}^2(\tau) + \hat{R}_{k,i}^{\,2}(\tau)\right)d\tau + \frac{1}{4}N_0T \\ &= \left(\frac{P}{4T}\right)\sum_{\substack{k=1\\k\neq i}}^{K}\sum_{l=0}^{N-1}\int_{lT_c}^{(l+1)T_c} \left(R_{k,i}^2(\tau) + \hat{R}_{k,i}^{\,2}(\tau)\right)d\tau + \frac{1}{4}N_0T\end{aligned} \qquad (3.24)$$

where the expectation has been computed with respect to the mutually independent random variables ϕ_k, τ_k, $b_{-1}^{(k)}$, and $b_0^{(k)}$ for $0 < k < K$ and $k \neq i$. Note that the interference terms from other user signals in are also random and have been included as additional noise components. The interference terms due to other users are of particular interest in our analysis. consequently we define the parameter V as follows:

$$V = \frac{1}{T} \sum_{\substack{k=1 \\ k \neq i}}^{K} \sum_{l=0}^{N-1} \int_{lT_c}^{(l+1)T_c} R_{k,i}^2(\tau) + \hat{R}_{k,i}^{\ 2}(\tau)\, d\tau \tag{3.25}$$

Substituting for $R_{k,i}(\tau)$ and $\hat{R}_{k,i}(\tau)$ from (3.22) and (3.23) into (3.24) and evaluating the resulting integral. the following expression is obtained for Var $\{Z_i\}$

$$\text{Var}\{Z_i\} = \frac{PT^2}{12N^3} \left(\sum_{\substack{k=1 \\ k \neq i}}^{K} r_{k,i} \right) + \frac{1}{4} N_0 T \tag{3.26}$$

where

$$r_{k,i} = \sum_{l=0}^{N-1} \{ C_{k,i}^2(l-N) + C_{k,i}(l-N)\, C_{k,i}(l-N+1) + C_{k,i}^2(l-N+1) + C_{k,i}^2(l) + C_{k,i}(l)\, C_{k,i}(l+1) + C_{k,i}^2(l+1) \} \tag{3.27}$$

The definition for $C_{k,i}(l)$ is given in (C.20) of Appendix C.

This expression can be written in terms of a new set of correlation parameters $u_{k,i}(n)$ which are defined by

$$u_{k,i}(n) = \sum_{l=1-N}^{N-1} C_{k,i}(l)\, C_{k,i}(l+N) \quad . \tag{3.28}$$

Equation (3.27) now becomes: [19]

$$r_{k,i} = 2u_{k,i}(0) + u_{k,i}(1) \tag{3.29}$$

For the signal-to-noise ratio. $\sqrt{\frac{P}{2}}T$ is divided by the rms noise $\sqrt{\text{Var}\{Z_i\}}$. This is

$$SNR_i = \left\{ (6N^3)^{-1} \sum_{\substack{k=1 \\ k \neq i}}^{K} [2u_{k,i}(0) + u_{k,i}(1)] + \frac{N_0}{2E} \right\}^{-1/2} \quad . \tag{3.30}$$

Pursley[19]. [20] gives the following approximation which can be used for a preliminary system design.

$$(6N^3)^{-1} \sum_{\substack{k=1 \\ k \neq i}}^{K} r_{k,1} \equiv \frac{(K-1)}{3N} . \tag{3.31}$$

which yields

$$SNR_i \equiv \{ \frac{K-1}{3N} + \frac{N_0}{2E} \}^{-1/2} . \tag{3.32}$$

The variance of Z_i from (3.26) can also be written as

$$\text{Var}\{Z_i\} \equiv \frac{P}{4}\left[\frac{2T^2}{3N}(K-1) \right] + \frac{1}{4}N_0 T \quad . \tag{3.33}$$

Finally the parameter V defined in (3.25) is now given by

$$V = \frac{2T^2}{3N}(K-1). \tag{3.34}$$

The value of the SNR at the ith receiver obtained in (3.32) and the estimate of the variance of the noise terms in (3.33) show that both SNR_i and $\mathrm{Var}\{Z_i\}$ depend to some degree on the number of users in the system and on the length of the sequence. This dependence is also reflected in the parameter V. In the next section we will investigate the effect of a non-linear system on the parameter V (which represents in effect the interference from other users). This analysis will follow the approach used by Khaleghi [8], [9].

3.4 Performance Evaluation of a CDMA System with Non-Linearities

Let use now consider a system with three users (i, j, k) arranged in the same manner as the system shown in Figure 3.3. Code and data waveforms for each user are defined in the same manner as in (3.14) and (3.15). Note that the code waveforms all have the same period T. This system however in addition to suffering from AWGN in the transmission channel is also non-linear. The non-linearity is such that the signals of users j and k are multiplied during transmission. One form that non-linearities commonly take is the multiplication of data signals from other users to form higher order interference terms. The non-linearity modelled here is a simplified version of the type of non-linearity that would occur in a SCM system using semiconductor lasers. The system is also assumed to be asynchronous such that the signal of user j is delayed by τ_j with respect to user i and user k's signal is delayed by τ_k compared to user j's signal, where $0 < \tau_j, \tau_k < T$.

The product of the code waveforms of users j and k may be given by a third waveform $a_{jk}(t)$. From Figure 3.4 it is apparent that $a_{jk}(t)$ is also periodic with period T.

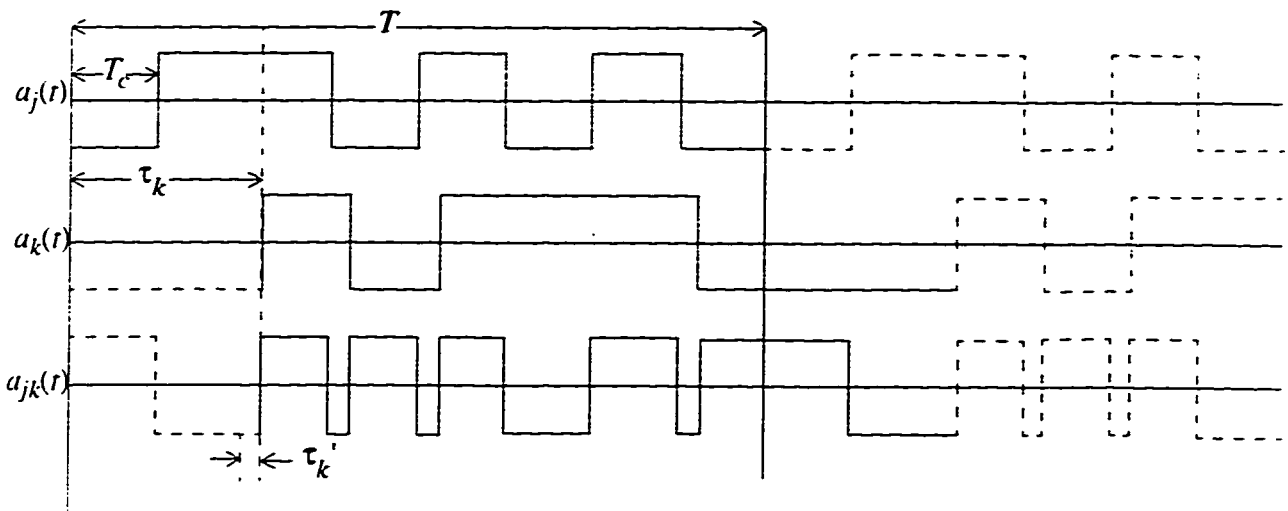


Figure 3.4 Product of two sequence waveforms

It can be seen that

$$a_{jk}(t) = \begin{cases} T^{l+1} a_j(t)\, a_k(t) & mT_c \le t \le mT_c + \tau_k' \\ T^{l} a_j(t)\, a_k(t) & mT_c + \tau_k' \le t \le (m+1)\, T_c \end{cases} \tag{3.35}$$

where $0 < m < N$ is an integer. The reader will recall that T stands for a circular shift to the left and τ_k' is defined such that $\tau_k = lT_c + \tau_k'$. where $0 < \tau_k' < T_c$ and l is an integer. We can also write the above in terms of sequences rather than waveforms as follows:

$$a_n^{(jk)} = \begin{cases} (T^{l+1} a_n^{(j)})\, a_n^{(k)} & m \le n \le m + (\tau_k'/T_c) \\ (T^{l} a_n^{(j)})\, a_n^{(k)} & m + \tau_k'/T_c \le n \le m+1. \end{cases} \tag{3.36}$$

In Section 3.2 it was explained that for certain sets of sequences, specifically Gold, Kasami, and maximally connected sets of m-sequences, the shift-and-add-property holds. Since the product of sequences of elements $\{+1, -1\}$ corresponds to modulo-2 addition of sequences of elements $\{0, 1\}$, $(T^{l+l}a_n^{\,j})a_n^{\,k}$ and $(T^l a_n^{\,j})a_n^{\,k}$ are thus two sequences belonging to the same set of sequences as $a_n^{\,j}$ and $a_n^{\,k}$ ($a_n^{\,j}$ and $a_n^{\,k}$ being sequences for which shift-and-add holds). For simplicity we will denote

$$\begin{aligned} (T^l a_n^{(j)})\, a_n^{(k)} &= a_n^{(l)}. \\ (T^{l+1} a_n^{(j)})\, a_n^{(k)} &= a_n^{(l+1)}. \qquad n = 0, 1, \ldots, N-1. \end{aligned} \tag{3.37}$$

The corresponding waveforms are

$$\begin{aligned} [T^l a_j(t)]\, a_k(t) &= a_l(t). \\ [T^{l+1} a_j(t)]\, a_k(t) &= a_{l+1}(t). \end{aligned} \tag{3.38}$$

Assuming that the receiver of the non-linear system under study is tuned to code wavelength $a_i(t)$, the output of the correlation receiver is given by

$$\begin{aligned} Z_i = \sqrt{P/2}\,\{b_0^{(i)} T + [b_{-1}^{jk} R_{jk,\,i}(\tau_j) + b_0^{jk} \hat{R}_{jk,\,i}(\tau_j)]\cos\phi_{jk}\} \\ + \int_0^T n(t)\, a_i(t) \cos(w_c t)\, dt. \end{aligned} \tag{3.39}$$

The calculations that follow focus on the term ξ:

$$\xi = \sqrt{P/2}\,[b_{-1}^{jk} R_{jk,\,i}(\tau_j) + b_0^{jk} \hat{R}_{jk,\,i}(\tau_j)]\cos\phi_{jk}\,. \tag{3.40}$$

which represents the interference from the combined sequences of users j and k.

The term ξ depends on the partial continuous-time cross-correlation functions which (as defined in (3.21)) are given by

$$R_{jk,i}(\tau_j) = \int_0^{\tau_j} a_{jk}(t-\tau_j)\, a_i(t)\, dt.$$
$$\hat{R}_{jk,i}(\tau_j) = \int_{\tau_j}^{T} a_{jk}(t-\tau_j)\, a_i(t)\, dt. \qquad (3.41)$$

The value ξ of is also dependent on the sign of b_{-1}^{jk} and b_0^{jk} . Both the cross-correlation functions and b_0^{jk} or b_{-1}^{jk} are affected by the fact that $a_k(t)$ is delayed by τ_k compared to $a_j(t)$. More specifically the sign b_0^{jk} or b_{-1}^{jk} may differ before and after τ_k.

Two possible relationships between $\tau_j.\tau_k$ and the period T of the sequence waveforms need to be considered.

a) $\tau_j + \tau_k =< T$ as shown in Figure 3.5a

b) $\tau_j + \tau_k > T$ as shown in Figure 3.5b

In case a) s_1 is the sign of b_{-1}^{jk} for $0 < t < \tau_j$. s_2 is the sign of b_0^{jk} for $\tau_j < t < \tau_j + \tau_k$. and s_3 is the sign of b_0^{jk} for $\tau_j + \tau_k < t < T$. Substituting (3.35) into (3.41) and taking into account s_1. s_2. and s_3. the following expressions are obtained of $R_{jk,i}(\tau_j)$ and $\hat{R}_{jk,i}(\tau_j)$:

where $\alpha = \tau'_k/T_c$.

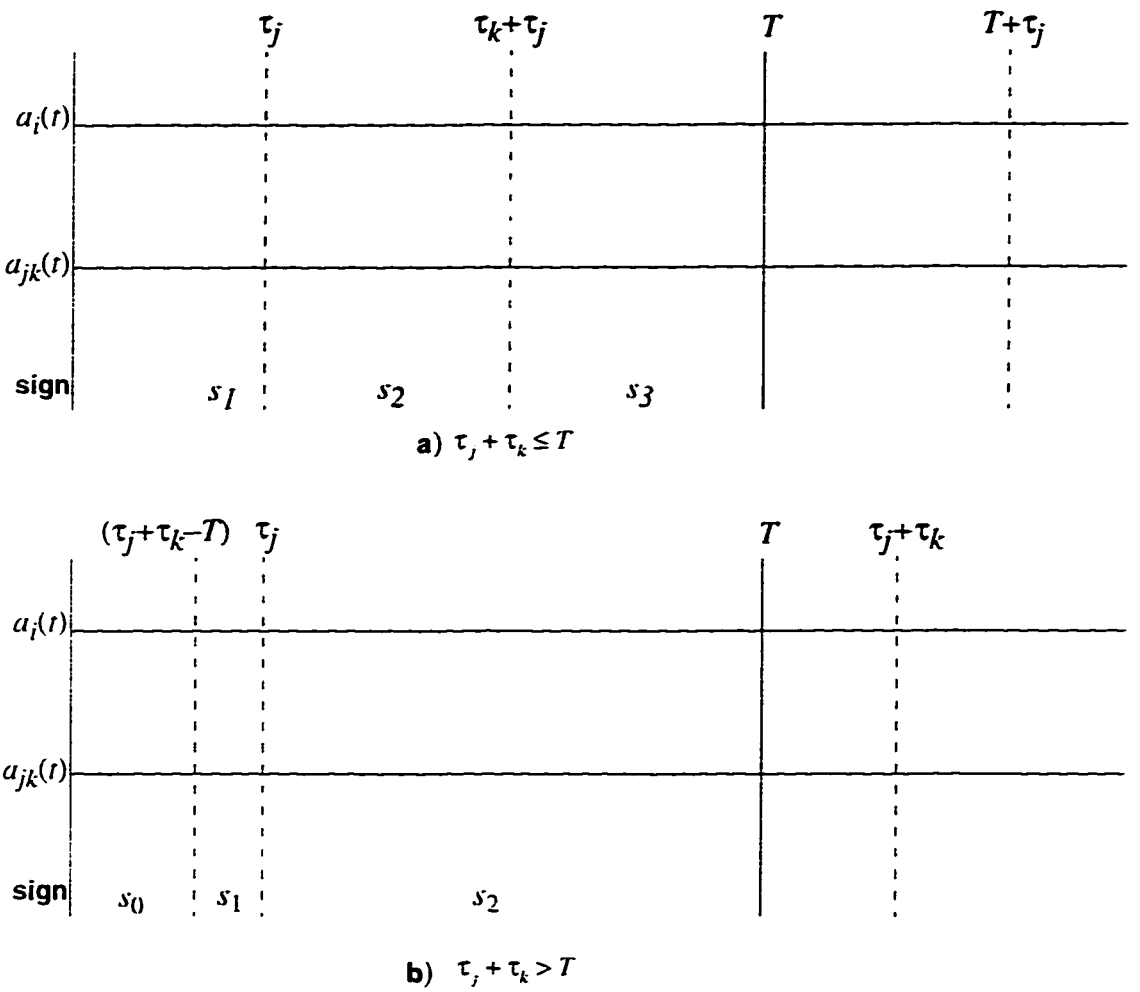


Figure 3.5 Comparison of $a_i(t)$ and $a_{jk}(t)$

In case b). s_0 is the sign of b_{-1}^{jk} for $0 < t < \tau_j + \tau_k - T$. s_1 is the sign of b_{-1}^{jk} for $\tau_j + \tau_k - T \le t < \tau_j$. and s_2 is the sign of b_0^{jk} for $\tau_j < t < T$. In this case the expressions for $R_{jk,i}(\tau_j)$ and $\hat{R}_{jk,i}(\tau_j)$ are

(3.43)

The terms s_0, s_1, s_2 and s_3 can be treated as independent random variables. Also note that $E[s_0^2] = E[s_1^2] = E[s_2^2] = E[s_3^2] = 1$.

$$R_{jk,i}(\tau_j) = s_1\left[\int_0^{\tau_j} \alpha a_{l+1}(t-\tau_j)\, a_i(t)\, dt + \int_0^{\tau_j} (1-\alpha)\, a_l(t-\tau_j)\, a_i(t)\, dt\right], \qquad (3.42)$$

$$\hat{R}_{jk,i}(\tau_j) = s_2\left[\int_{\tau_j}^{\tau_j+\tau_k} \alpha a_{l+1}(t-\tau_j)\, a_i(t)\, dt + \int_{\tau_j}^{\tau_j+\tau_k} (1-\alpha)\, a_l(t-\tau_j)\, a_i(t)\, dt\right]$$
$$+ s_3\left[\int_{\tau_j+\tau_k}^{T} \alpha a_{l+1}(t-\tau_j)\, a_i(t)\, dt + \int_{\tau_j+\tau_k}^{T} (1-\alpha)\, a_l(t-\tau_j)\, a_i(t)\, dt\right].$$

$$R_{jk,i}(\tau_j) = s_0\left[\int_0^{\tau_j+\tau_k-T} \alpha a_{l+1}(t-\tau_j)\, a_i(t)\, dt + \int_0^{\tau_j+\tau_k-T} (1-\alpha)\, a_l(t-\tau_j)\, a_i(t)\, dt\right]$$
$$+ s_1\left[\int_{\tau_j+\tau_k-T}^{\tau_j} \alpha a_{l+1}(t-\tau_j)\, a_i(t)\, dt + \int_{\tau_j+\tau_k-T}^{\tau_j} (1-\alpha)\, a_l(t-\tau_j)\, a_i(t)\, dt\right].$$

$$\hat{R}_{jk,i}(\tau_j) = s_2\left[\int_{\tau_j}^{T} \alpha a_{l+1}(t-\tau_j)\, a_i(t)\, dt + \int_{\tau_j}^{T} (1-\alpha)\, a_l(t-\tau_j)\, a_i(t)\, dt\right].$$

Returning to the parameter ξ: its variance is given by

$$\mathrm{Var}(\xi) = \frac{P}{4T}\int_0^T \left(R^2_{jk,\,i}(\tau_j) + \hat{R}^2_{jk,\,i}(\tau_j) \right) d\tau_j \tag{3.44}$$

$$= \frac{P}{4T}\int_0^T \left\{ \left[\int_0^{\tau_j} \alpha a_{l+1}(t-\tau_j)\, a_i(t)\, dt \right]^2 + \left[\int_0^{\tau_j} (1-\alpha)\, a_l(t-\tau_j)\, a_i(t)\, dt \right]^2 \right.$$

$$+ \left[\int_{\tau_j}^{T} \alpha a_{l+1}(t-\tau_j)\, a_i(t)\, dt \right]^2 + \left[\int_{\tau_j}^{T} (1-\alpha)\, a_l(t-\tau_j)\, a_i(t)\, dt \right]^2$$

$$\left. + 2\left[\int_0^T \alpha a_{l+1}(t-\tau_j)\, a_i(t)\, dt \right] \cdot \left[\int_0^T (1-\alpha)\, a_l(t-\tau_j)\, a_i(t)\, dt \right] \right\} d\tau_j .$$

Using (C.10) of Appendix C. the following expression can easily be obtained

$$\int_0^T a_l(t-\tau_j)\, a_i(t)\, dt = T_c [\, (1-\alpha')\, \theta_{l,\,i}(n) + \alpha' \theta_{l,\,i}(n+1) \,] \,. \tag{3.45}$$

where $nT_c \le \tau_j < (n+1)\,T_c$ and $\alpha' = \frac{1}{T_c}(\tau_j - nT_c)$. The average of the right hand side of (3.45) over α' is uniformly distributed over [0. 1]. Therefore. the average value of the last term of (3.44) over n is at most equal to $2\left(\frac{T_c}{N}\right)\left(\frac{T_c}{N}\right) = \frac{2T^2}{N^4}$ based on the result of (3.6) in the section on properties of m-sequences. It is worth mentioning that (3.6) is true not only for m-sequences. but also for Gold and Kasami sequences. The first four terms of (3.44) are equal to

$$(1-\alpha)^2 R^2_{l,\,i}(\tau_j) + \alpha^2 R^2_{l+1,\,i}(\tau_j) + (1-\alpha)^2 \hat{R}^2_{l,\,i}(\tau_j) + \alpha^2 \hat{R}^2_{l+1,\,i}(\tau_j) \,. \tag{3.46}$$

The average of these terms over α is equal to

$$\frac{1}{3}\left[R^2_{l,i}(\tau_j)+\hat{R}^2_{l,i}(\tau_j)\right]+\frac{1}{3}\left[R^2_{l+1,i}(\tau_j)+\hat{R}^2_{l+1,i}(\tau_j)\right]. \tag{3.47}$$

Let us define the term V' as equal to the variance of ξ (given by (3.69)) divided by $P/4$. V' can also be written as

$$\begin{aligned} V' &= \frac{1}{T}\int_0^T \left[R^2_{jk,i}(\tau_j)+\hat{R}^2_{jk,i}(\tau_j)\right]d\tau_j \\ &= \frac{1}{T}\int_0^T \{\frac{1}{3}\left[R^2_{l,i}(\tau_j)+\hat{R}^2_{l,i}(\tau_j)\right]+\frac{1}{3}\left[R^2_{l+1,i}(\tau_j)+\hat{R}^2_{l+1,i}(\tau_j)\right]\}d\tau_j \\ &= \frac{1}{3}\left[\frac{1}{T}\int_0^T\left(R^2_{l,i}(\tau_j)+\hat{R}^2_{l,i}(\tau_j)\right)d\tau_j\right]+\frac{1}{3}\left[\frac{1}{T}\int_0^T\left(R^2_{l,i}(\tau_j)+\hat{R}^2_{l,i}(\tau_j)\right)d\tau_j\right]+\frac{2T^2}{N^4}. \end{aligned} \tag{3.48}$$

Substituting (3.34) into each of the terms in square brackets above we obtain

$$V' = \frac{1}{3}\left[\frac{2}{3}\frac{T^2}{N}\right]+\frac{1}{3}\left[\frac{2}{3}\frac{T^2}{N}\right]+\frac{2T^2}{N^4} \tag{3.49}$$

where for large N, the last term can be considered negligible, hence

$$V' \equiv \frac{2}{3}\left[\frac{2}{3}\frac{T^2}{N}\right] = \frac{2}{3}(V). \tag{3.50}$$

As an extension the product of three or more waveforms of period T is also a periodic waveform of the same period T, consisting of three or more sequences expressible by an equation such as (3.35). The continuous-time partial cross-correlation functions from this product waveform and an arbitrary sequence waveform $a_i(t)$ are defined in a similar

manner to (3.41) and the definition of the term V' can be extended for such partial cross-correlation functions. The new V' is linearly proportional to the number of multiplying waveforms, resulting in

$$V'(n) \equiv \frac{n}{3}\left(\frac{2T^2}{3N}\right). \tag{3.51}$$

where n is the number of multiplying waveforms. It is worth repeating that n also represents the order of the non-linear interference term.

Chapter 4: CDMA/SCM Optical LAN Simulation

In the preceding chapter it was shown how CDMA can be used to mitigate the performance degradations which occur in non-linear systems. However, as mentioned earlier CDMA used alone is spectrally inefficient because of the bandwidth spreading which makes it so resistant against interference. The idea of a hybrid network which would combine the benefits of CDMA with the strengths of frequency division multiplexing was proposed by Vanucci [7]. This idea was taken up by Khaleghi who has studied a network using CDMA and subcarrier multiplexing. Khaleghi has provided a theoretical analysis[8]. [9] which shows that combining CDMA with SCM results in a network where the intermodulation products and harmonics produced by the non-linear laser diodes have a similar effect compared to that caused by other users' non-matching sequences. Khaleghi also set up an experiment to confirm the results of the theoretical analysis[8], [9]. The experiment however did not include lasers and so did not consider laser non-linearity.

Performing a theoretical analysis of the hybrid network, which includes laser induced distortion, to verify Khaleghi's results would be extremely difficult. The difficulty would lie in predicting the actual levels of intermodulation distortion produced by the network's lasers. The intermodulation distortion in a multichannel (or multi-subcarrier) system is dependent on the phases of all the carriers. Since these phases are assumed to be independent, the resulting distortion level is random. It is practically not feasible to determine the distortion level analytically.

In light of the difficulty of the analytical approach, a simulation of the hybrid CDMA/SCM network was constructed. The network configuration which was used for the simulation is shown in Figure 1.2.

The network in Figure 1.2 is configured as a star. The advantages of the star topology compared to other configurations used in LAN's were described in Chapter 2. Each user is assigned a unique pair of coordinates consisting of a code sequence c_i and a subcarrier frequency f_j. The intended receiver of a particular transmission decodes and demodulates the received signal through knowledge of the code, frequency pair. Note that it is possible to combine fixed sources with tunable receivers or tunable sources with fixed receivers. Users may be grouped in clusters around a particular light source. The number of users per cluster will depend on the number of available frequencies. As many clusters as there are codes can be set up depending only on the performance quality required in the network.

As shown in Figure 1.2 the data of each user is added to a sequence of chips modulo-2. The resulting spread data signals modulate the FDM subcarriers. The subcarriers of all users are summed and used to intensity modulate a laser diode. Light emitted from the laser is guided thought the network via optical fibers and broadcast to all receivers over a passive star coupler. The section that follows presents a little of the history of our simulation and addresses some issues specific to computer simulations. Section 4.2 describes the models which form the basis for the simulation.

4.1 Computer Simulation Issues

The goal of any simulation is to reproduce some aspect of a "real life" situation or system, which cannot easily be analyzed theoretically and where performing an experiment may be costly or impractical. In our case, both criteria apply to some degree. Conducting an experiment involving an entire optical network would be hard to arrange and certain components of the network, such as the lasers are relatively expensive. A simulation can be a convenient, low cost alternative.

4.1.1 Simulation Construction Approach

When this work was begun, two options were available to approach the construction of a simulation: writing a custom simulation program or constructing a simulation using a commercially available software package BOSS (Block Oriented System Simulator) or its newer sister product, the Signal Processing Worksystem (SPW). Due to their graphical user interface and transparency to actual coding (which would normally allow the user to focus on the simulation issues rather than coding issues), BOSS and SPW were initially chosen to build the network simulation. The post-processing functions of BOSS, such as the ability to plot various signals were especially useful in verifying the characteristics of the laser module. However, for this project, BOSS and SPW both suffered from critical limitations, and a custom program was written in C. With BOSS, an important module in the simulation caused a segmentation fault, which was not resolved even with help from the BOSS hotline staff. With SPW, simulations were limited in size to a relatively small number of data

symbols (about ten) due to the very small sampling period required (explained in the next section) and the finite memory space of the work stations running the simulations. Some special features of the custom C program will now be described.

Simulations of the hybrid optical network were expected to eventually handle thousands and even millions of samples, generating potentially enormous files. The simulation program was therefore written with several features put in place to limit the size of the files that must be handled during the simulation. In the custom simulation program, larger simulations are run by looping through a series of short simulations (referred to as loops). The state of network components at the end of each loop is recorded and used to set the components' state at the beginning of the next loop. For convenience the size the data block processed during a single loop was set to 32 data symbols.

Memory space required for intermediate steps of the simulation is only allocated for the time that it is used then freed up again. User signal information is thus passed along a sort of "bucket brigade" of arrays as it is processed by the network.

A further memory space saving measure was applied to all subroutines where data processed is in the form of ones and zeros. For these subroutines, the data bits are stored in blocks of 32, which is the length of a long integer on most computers. Data in the subroutines is processed making extensive use of the bit manipulating functions in the C language. This measure can provide significant space savings compared to allocating an integer variable for each data bit, although the program code for manipulating the bits is a little more complex.

A final feature to make the simulation program more manageable is the use of interpolation and decimation at intermediate stages in the network. This feature allows the size of some files required during the simulation to be considerably reduced. We will now explain why this feature is required for our simulation.

A computer simulation is fundamentally a discrete process in that the simulation produces a finite set of output values over a finite set of time samples (in the case of a time-domain simulation). This is illustrated in Figure 4.1. If the set of possible output values on the vertical axis of Figure 4.1 is large (tends toward infinity) and the value of the sampling period, T_s is very small (approaches zero), then the simulation output can approximate the behaviour of most systems including analog continuous-time systems. All simulation results depend to some degree on the choice of the range of values the simulation functions can take on and on the value of T_s. The parameter T_s is defined as the time interval between any two adjacent time samples.

Ideally T_s should be very small, approaching zero not only to provide good quality simulation results, but also to ensure that all subsystems of the simulation are stable. As will be discussed in Section 4.1.2, the stability of the laser module depends on having a small simulation time step.

There is however an important constraint to setting T_s very small. Given a fixed data rate B, where $T_s << T = 1/B$, then many samples of length T_s will be required at the simulation output to describe even one complete data bit. In our case the performance of the hybrid network will be determined by comparing its bit error rate to that of a SCM

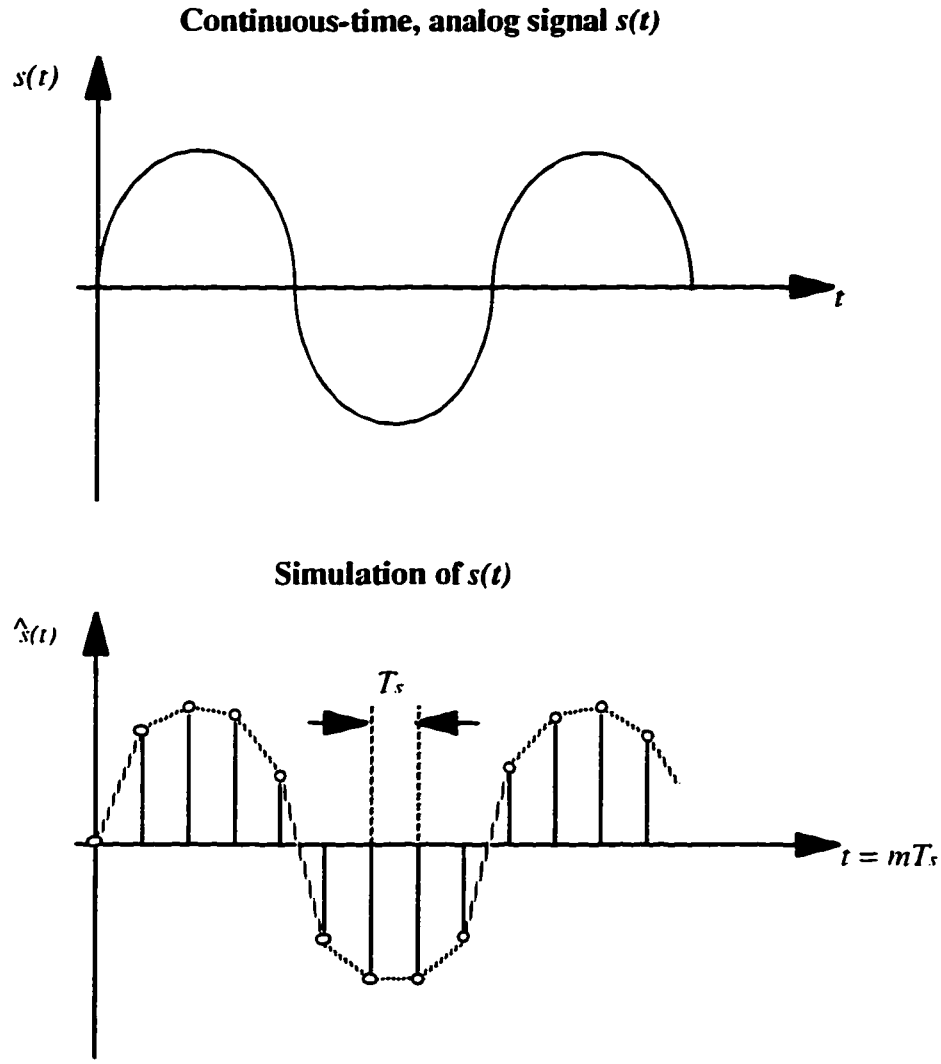


Figure 4.1 Comparison of a signal and its simulation

optical network. This requires simulations of thousands even millions of data bits. If 1E+6 bits must be simulated and $T/T_S = N$, then NE+6 samples must be simulated. It follows that if N is large then the quantity of data handled by the simulation may become hard to manage.

Simulating a network which requires T_s to be very small might seem to be an onerous task, except for the fact that only some components in the network under study require a small value of T_s. More specifically at the optical core of the network a small value of T_s is needed, but not for the components preceding the phase modulation of the users' signals or the modules following the received signal's phase demodulation. These components are discrete-time components and their performance is less affected by a larger value of T_s, provided that the simulation rate f_s ($= 1/T_s$) is equal to or larger than the highest data rate of the component being simulated. If the fundamental sampling rate can be lowered for some modules, the number of samples to be processed at these modules will be decreased, reducing overall amount of data processed by the simulation.

However having a fundamental sampling rate that is not the same for all simulation modules, necessitates the introduction of new modules into the simulation program. These new modules perform the conversion between different sampling rates at some points in the network. The conversion from a lower to a higher sampling rate is referred to as interpolation, while a reduction in the sampling rate is called decimation. The operations of interpolation and decimation will be described further in Section 4.2, where all the components of the hybrid network are presented in detail.

4.1.2 Requirements for Simulation Stability

One would expect the simulation of a laser to be stable, since real lasers are stable physical systems. However for reasons which will be explained, the simulated laser can become unstable if the simulation time step used is too large. This problem was encountered and described by Neusy [21]. Our simulation of the laser agrees with his results.

A laser diode is basically a continuous-time system represented by a set of differential equations (2.2) and (2.3). In the simulation of the laser, the following relation is used to approximate the derivative of the carrier density, N and the photon density, S:

$$\frac{d}{dt}N(t) \cong \frac{N(t+T_s) - N(t)}{T_s}. \tag{4.1}$$

where T_s is the simulation time step. Substitution of the above into the rate equations, gives a set equations of the form:

$$V(t+T_s) = N(t) + T_s f[I(t), N(t), S(t)]. \tag{4.2}$$

where $I(t)$ is the current injected into the laser's active region. This current was defined as I_A in (2.2)

It can be seen that this approximation of the rate equations is the model of a feedback system, where each simulation output sample depends on earlier values of the simulation. In systems with feedback, it is possible for the feedback to be such that the

system becomes unstable. The approximation of a continuous system with a discrete system in (4.2) is an example of Euler forward integration. which is of the general form[22]:

$$x_i(nT_s) = x_i(nT_s - T) + Tg_i(nT_s - T_s). \tag{4.3}$$

This mapping. from continuous to discrete. is stable as long as the poles of the z-transform are within the unit circle as indicated in Figure 4.2 and Figure 4.3. In Figure 4.3

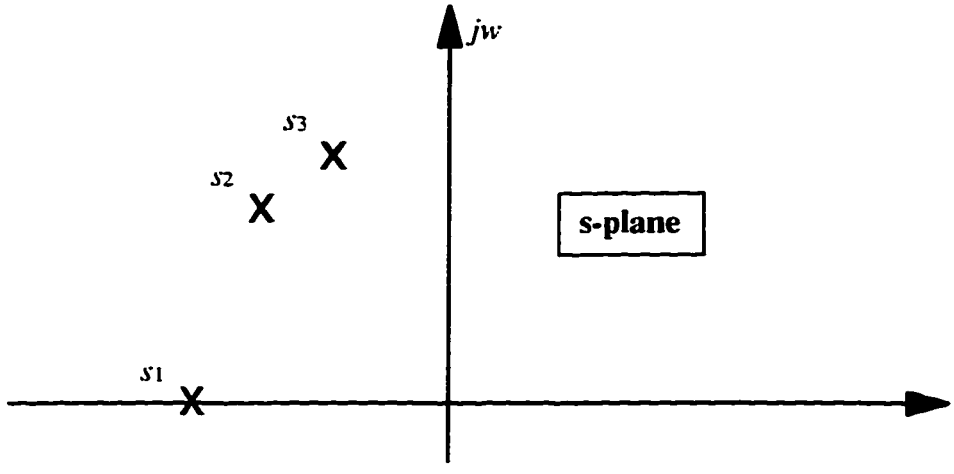


Figure 4.2 S-plane poles

it can be seen that the imaginary axis is mapped onto a line that is tangential to the unit circle. The transformation from the s-plane to the z-plane is given by

$$z = sT_s + 1. \tag{4.4}$$

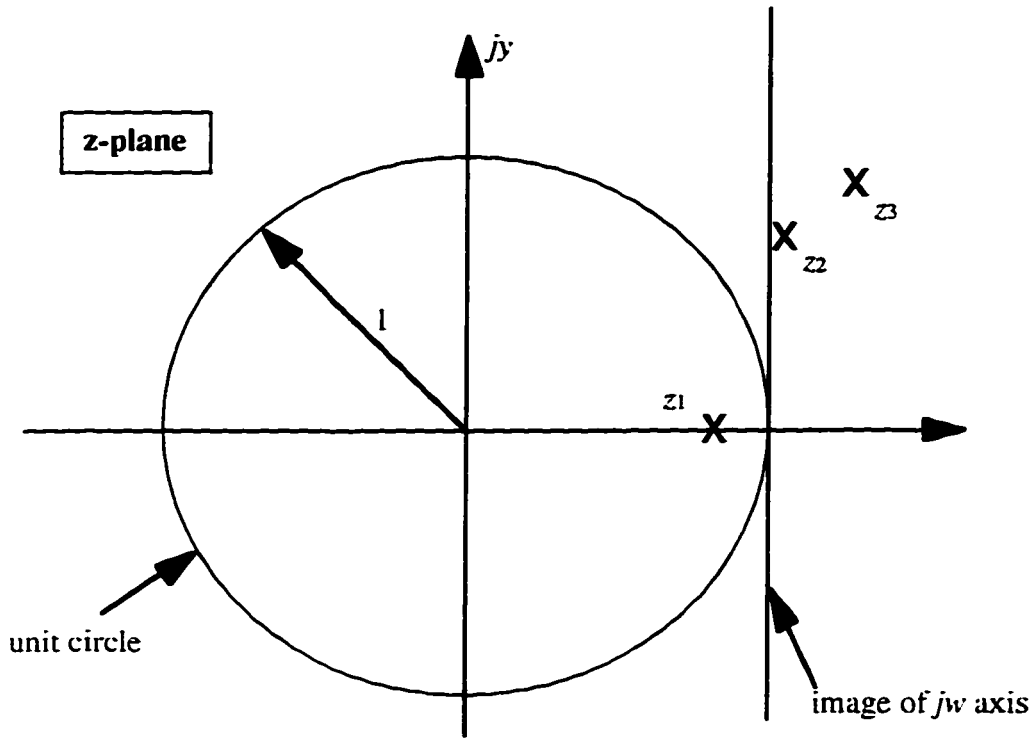


Figure 4.3 Z-plane poles

As with any discrete system, stable performance is obtained if the poles of the system are within the unit circle. Therefore, for stability $\|z_p\| < 1$. i.e., when $\|sT_s + 1\| < 1$. With poles at $s = \alpha + j\omega$, it is possible to determine the value of T_s at which the system is on the verge of instability by solving for T_s in (4.4). We obtain the following:

$$T_s = \frac{2\alpha}{\alpha^2 + \omega^2}. \tag{4.5}$$

In [21] Neusy shows that for a laser biased at 75 mA, α= 3.82E10 and ω = 2.589E10, giving a critical T_s = 3.587E–11.

In Figure 4.2 and Figure 4.3 one can clearly see the mapping of the imaginary axis in the s-domain onto a line tangential to the unit circle in the z-domain. This implies that for a good representation of the frequency characteristics of the laser, T_S must be chosen to be small enough, such that $\|\omega T_s\| \ll 1$. In our simulations T_S was set to 2.5E-12. Assuming that ω = 2.589E10, this gives $\|\omega T_s\| = 0.065$.

Maintaining T_S small enough provides stable laser simulations, but it would be of interest to find a mapping from the s-plane to the z-domain for which the laser would always be stable. Backward Euler integration conserves stability of a system, but the transformation is of the form[22]:

$$x_i(nT_s) = x_i(nT_s - T_s) + T_s g_i(nT_s) \tag{4.6}$$

In our case since g_i is a function of x_i, we cannot make this substitution. Hence, we conclude that by choosing a sufficiently small time step (i.e., such that $\|\omega T_s\| \ll 1$) for our simulations of the laser, we will have a stable simulation and accurate results.

4.2 The CDMA/SCM Network and its Components

In this section, all functions performed by various components of the hybrid network will be presented in greater detail than before. A more detailed illustration of the hybrid network's components and how they are interconnected is shown in Figure 4.4. Components are numbered in the order that they will be presented below. Some components are presented in pairs, such as the differential encoder and decoder, since they represent an operation which is performed at the input to the network and later "undone" at

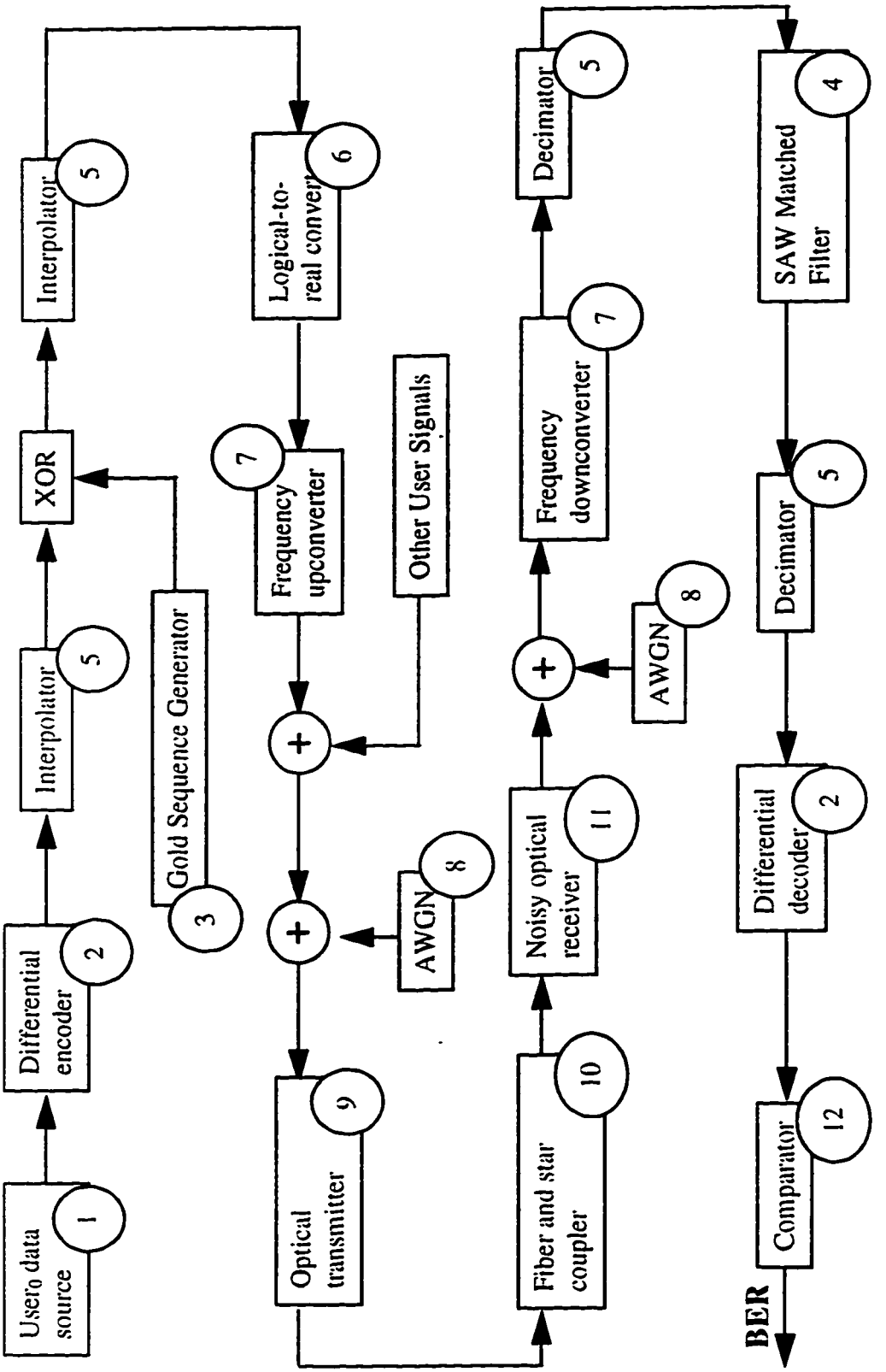


Figure 4.4 Block diagram of the SCM/CDMA optical LAN

the receiver. Note that signals are encoded, spread and modulated by a series of subsystems at the transmitter end of the network in a certain order. The order is then reversed for the equivalent demodulating, despreading and decoding modules. Also note that certain components are specific to a simulation environment.

The simulation program, written in the C language, is provided in Appendix A. For the purpose of better managing the simulation, the network was divided into two parts. The portion from the user data generation up to the adder which sums all the users' signals prior to the light source can be seen as the signal generation or input part of the network, these operations are all run in subroutine hscm_in.

The part of the network starting at the optical transmitter and ending with the final output of the network is loosely referred to as the "signal processing" portion of the LAN. Each of the components in this part of the network, from the AWGN source to the final demodulation of the data modifies the transmitted signal. This segment of the network is included in subroutine hscm_sp. All other subroutines are named in a fairly obvious fashion.

4.2.1 User Data Source

The user data source produces a sequence of bits (in blocks of 32) taking on values {0, 1} where each value is equally probable. The block diagram of the user data source is shown in Figure 4.5. It can be seen that the data bits are generated by taking the modulus of the integer output of a random number generator divided by two.

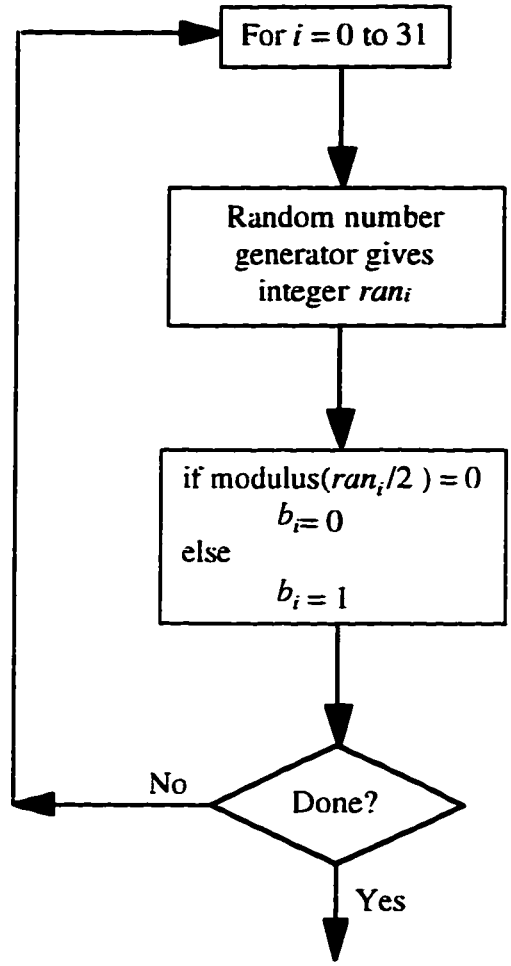


Figure 4.5 Block diagram of the user data source module

The random number generator produces uniformly distributed deviates. It is used not only for the user data source. but also as a basis for most of the components with noise. For example. in other network components. transformations are applied to uniform random values to give random values with a Poisson or a normal distribution. It is thus necessary to ensure that the values produced by the random number generator are truly random and have the desired uniform distribution. To this effect the random number generator is based on three linear congruential generators as provided in [23].

4.2.2 Differential Encoder and Decoder

A system in which binary data directly modulates the phase of a carrier requires that some knowledge of the carrier phase of the recovered signal be available to the demodulator. The carrier recovery in such systems needs to be synchronized with the transmitter and the operation to extract the carrier phase from the received signal may introduce a phase ambiguity. Both these issues can be avoided by encoding the information in phase differences between successive data transmissions (or bits) as opposed to absolute phase encoding. The signals resulting from the encoding process are said to be differentially encoded, and phase modulation of a carrier using a differentially encoded signal is referred to as differentially encoded phase-shift keying (DEPSK).

The operation to differentially encode data bits is shown in Figure 4.6. It can be seen that each bit at the output of the differential encoder is the sum modulo-2 of the user's data bit and the previous differentially encoded bit. The delay in the feedback loop is set to the duration T of one user symbol.

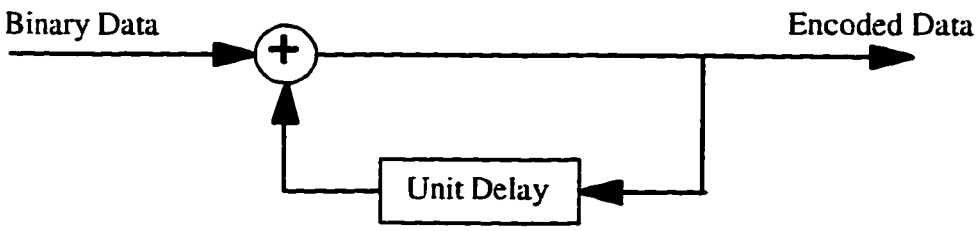


Figure 4.6 Block diagram of differential encoder

The block diagram of the module which decodes differentially encoded bits is shown in Figure 4.7. The decoding process simply involves XORing each encoded bit with a delayed version of itself. The delay is also set to T in this case. It will be shown in Section

4.2.4 that the operations described above will successfully encode and decode a sequence of bits. It may be noted that the block diagram does not include any filtering or decision submodules. These operations were included in the comparator module, which is described in Section 4.2.12

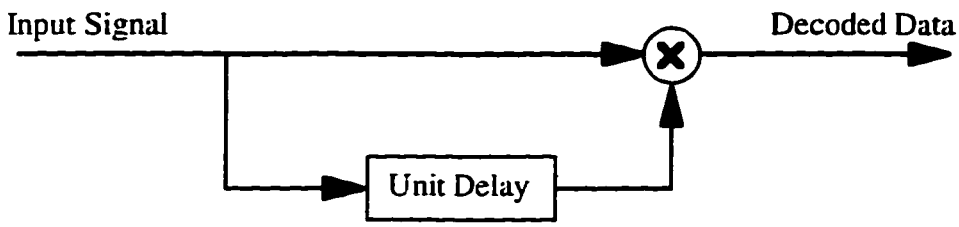


Figure 4.7 Block diagram of differential decoder

4.2.3 Gold Sequence Generator

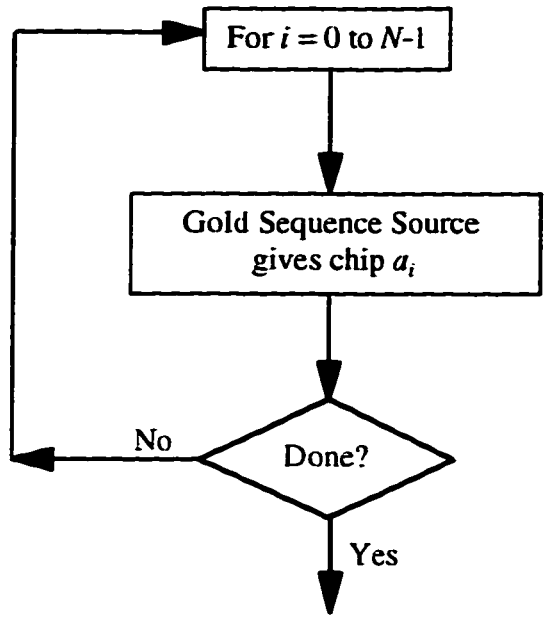
The differentially encoded bits are combined with a sequence of chips through the XOR operation. Chips are the bits of length T_C produced by the PN-sequence generator. The length of a data bit is $T = NT_C$, where N is the PN-sequence period. For our simulation we have chosen Gold sequences to spread the data bits due to their relative ease of generation and their good correlation properties (see Section 3.2).

We will now explain how to generate a series of chips, which form a Gold sequence. As was discussed in Chapter 3, Gold sequences are a class of linear binary shift register sequences, which possess good cross-correlation properties and for which the shift-and-add property holds, while having reasonably large numbers of sequences in a particular set. Gold sequences can be obtained from the sum modulo-2 of a preferred pair of m-sequences.

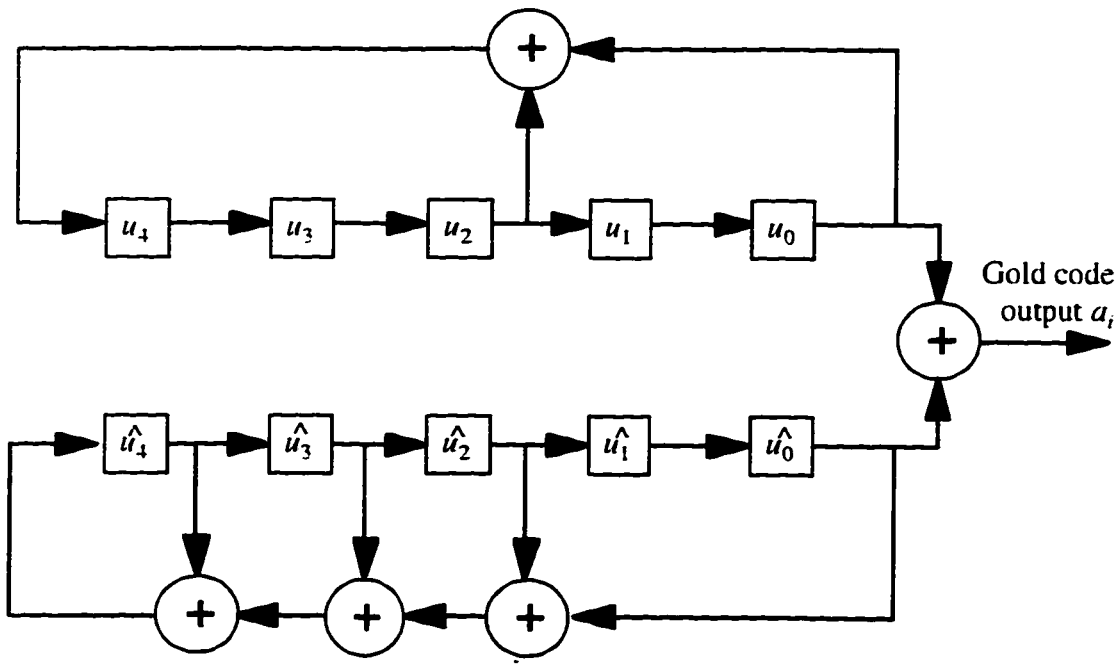
Recall that if a sequence a is generated from the polynomial $h(x)$ and a sequence b is generated from $\hat{h}(x)$. then $f(x) = h(x)\hat{h}(x)$ is the generator polynomial of a Gold sequence $a \oplus b$.

A set of sequences of order $n = 5$ and period $N = 31$ was used for many simulations. It was the result of combining two m-sequences whose generator polynomials are $h(x) = x^5 + x^2 + 1$ and $\hat{h}(x) = x^5 + x^4 + x^3 + x^2 + 1$ respectively. The polynomial $h(x)$. which generates sequence a was obtained from a table of irreducible polynomials given in [24]. Sequence b produced by $\hat{h}(x)$ is a decimation of a such that $b = a[3]$. This decimation meets the requirements to be a proper decimation. and so (a, b) form a preferred pair. Polynomials are often specified by an octal number (this is the case in [24]). The octal number represents the value taken on by the coefficients of the generator polynomial. It can be seen that 45_8 represents $h(x)$ and 75_8 represents $\hat{h}(x)$. The equivalent decimals that were used in the simulation are *37* for $h(x)$ and *61* for $\hat{h}(x)$.

The block diagram of the Gold sequence generator is shown in Figure 4.8. It consists of the module "Gold Sequence Source" set inside a loop of length N where N is the period of the Gold sequence. Every time it is accessed Gold Sequence Source generates a chip of value 0 or 1. It can be seen that the generator polynomials. $h(x)$ and $\hat{h}(x)$ specify the connections in the feedback paths of the two shift registers of Gold Sequence Source. The output of the two shift registers is XORed to give the sequence of chips corresponding to the Gold sequence $a \oplus b$. Different Gold sequences can be obtained by varying the initial condition of the cells of the shift registers. Different initial states of the shift registers



a) Gold sequence generator



b) Gold sequence source

Figure 4.8 Block diagram of Gold sequence generator

will cause sequences that are delayed with respect to each other to be generated. For the reference transmitter the initial condition of both shift registers was set to 1 (i.e., only $u_0 = 1$ initially; all other u_i's set to zero). It should be noted that the initial setting of the shift registers should not be zero, as this will produce an all-zero sequence.

4.2.4 SAW Matched Filter

In order to recover the initial information signal sent out by a user (say user$_0$), the intended receiver in a CDMA system must, among other demodulation and decoding operations, remove the spreading sequence that was superimposed on the data bit sequence. This function is usually performed by a filter matched to the sequence used by user$_0$. Conceptually the matched filter in this case is a transversal filter, where the signal is repetitively delayed and added to itself. One design technique used for such a filter is to use a Fourier series approach to make a correspondence between the impulse response of the filter and the taps in the delay line. In this case, where the filter is matched to a PN-sequence, the taps take on the values of the sequence. In practical systems the matched filter is often implemented using a surface acoustic wave (SAW) filter. The model of the SAW matched filter used in the network simulation is illustrated in Figure 4.9.

We now present an analysis to show how DPSK data spread with a PN-sequence, (α_n), can be recovered using the SAW filter in Figure 4.9 matched to the PN-sequence, (α_n), and a differential decoder as depicted in Figure 4.7. The system under analysis is shown in Figure 4.10

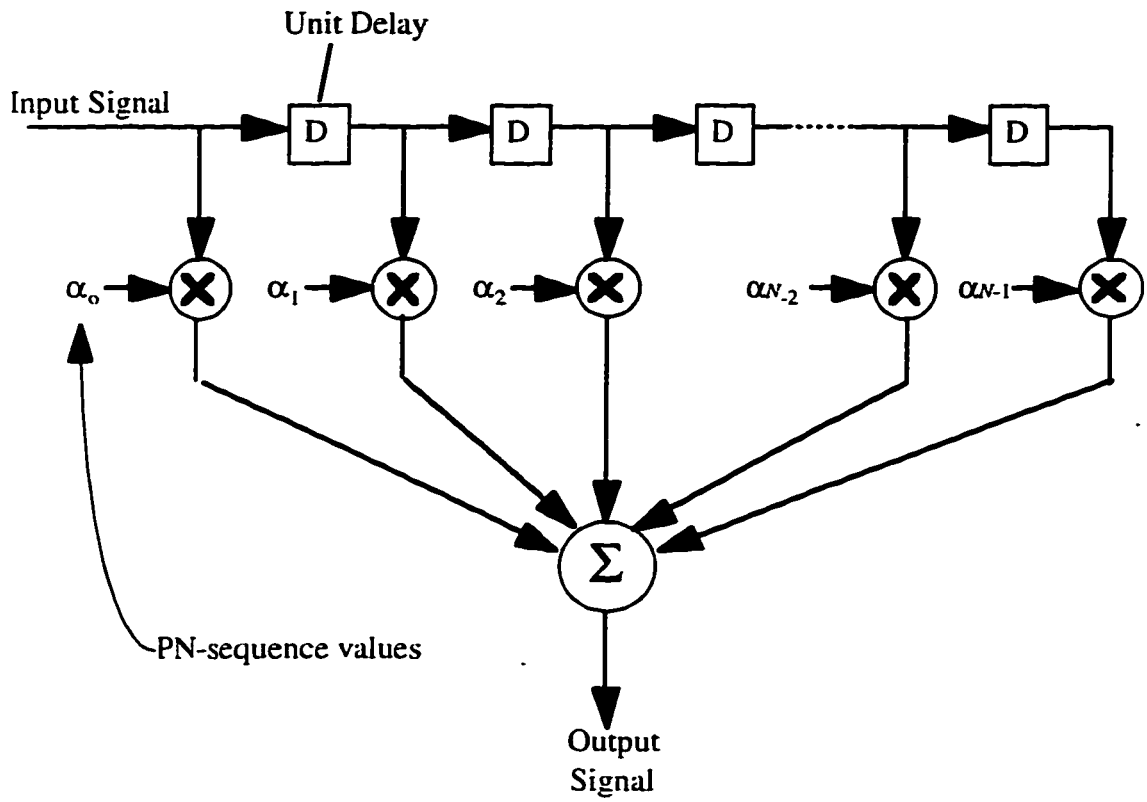


Figure 4.9 Model of the SAW matched filter

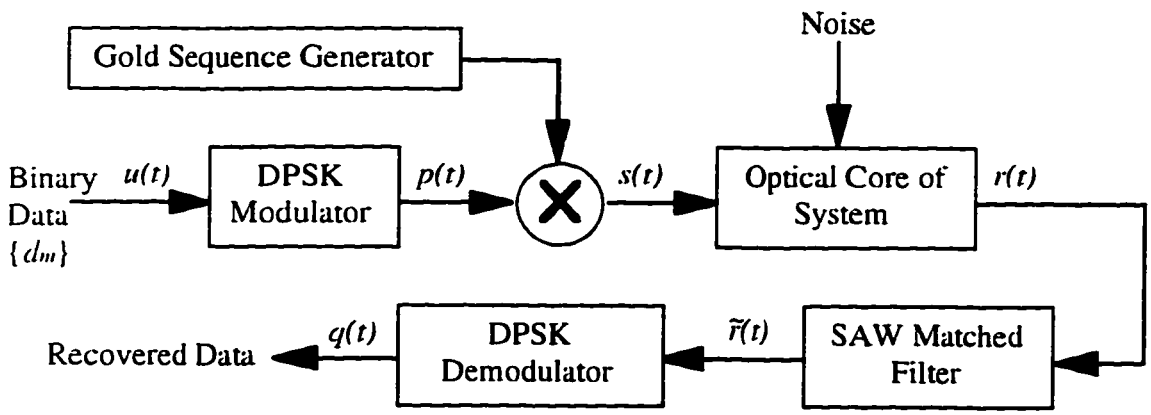


Figure 4.10 Block diagram of system using DPSK and spread-spectrum

To begin with, the binary data (d_m) is on a baseband waveform of NRZ format. The baseband waveform, $u(t)$ can be written as

$$u(t) = \sum_m d_m p_{NRZ}(t - mT) \tag{4.7}$$

where $p_{NRZ}(t)$ is a rectangular pulse of length T starting at $t = 0$. It may be recalled that for binary DPSK, the information bits are transmitted by shifting the phase of the carrier by 180^o degrees (binary 1) or by 0^o degrees (binary 0) relative to the phase in the previous signalling interval, T. The signal waveform $p(t)$ at the output of the DPSK modulator is given by

$$p(t) = Re\left\{K\left[\sum_m \hat{d}_m p_{NRZ}(t - mT)\right] e^{j\omega_0 t}\right\} \tag{4.8}$$

where $\hat{d}_m = d_m \hat{d}_{m-1}$.

The signal is next multiplied by a Gold sequence. The Gold sequence generator produces the chips (α_n), which have a duration of T_c. The period of the Gold sequence is N. The output of the Gold sequence generator, $v(t)$ takes on the same form as (4.7) except that the pulse waveform now has a duration of only T_c.

The signal $s(t)$ can be expressed as

$$s(t) = p(t) \cdot v(t) = Re\left\{K \sum_m \hat{d}_m \left[\sum_{n=mN}^{(m+1)N-1} \alpha_n \tilde{p}_{NRZ}(t - nT_c)\right] e^{j\omega_0 t}\right\} \tag{4.9}$$

where N is also assumed to be the number of chips per data symbol, and where $\tilde{p}_{NRZ}(t)$ is the combined pulse waveform.

At the receiver end, the detected signal $r(t)$ is to be despread using a SAW device matched to the particular sequence being used at the transmitter. The transfer function $H(\omega)$ of the SAW matched filter can be obtained as follows.

The impulse response of a filter matched to an input $\tilde{s}(t)$ is:

$$h(t) = \tilde{s}(T-t) \qquad 0 \le t \le T \tag{4.10}$$

Taking the Fourier transform of $h(t)$, we get:

$$H(\omega) = \tilde{S}^*(\omega)\, e^{-j\omega T} \tag{4.11}$$

where $\hat{S}^*(\omega)$ is the complex conjugate of the Fourier transform of $\tilde{s}(t)$.

The input signal $s(t)$ to which the filter is matched, is a PSK signal multiplied by a PN-sequence. It is represented by the part of (4.9) in square brackets.

The Fourier transform of $\tilde{s}(t)$ can be calculated as follows:

$$\tilde{S}(\omega) = \int_{-\infty}^{\infty} \sum_{n=0}^{N-1} \alpha_n \tilde{p}_{NRZ}(t - nT_c) \cos\omega_0 t e^{-j\omega T} dt \ . \tag{4.12}$$

After some calculations, one can obtain $\tilde{S}(w)$:

$$\tilde{S}(\omega) = \tilde{P}_{NRZ}(\omega) \sum_{n=0}^{N-1} \alpha_n e^{-j\omega T_c} \cdot \frac{T}{2} \left\{ \operatorname{sinc}\left[\frac{(\omega-\omega_0)T}{2\pi}\right] + \operatorname{sinc}\left[\frac{(\omega+\omega_0)T}{2\pi}\right] \right\} \tag{4.13}$$

where $\operatorname{sinc} x = \sin\pi x/(\pi x)$. Therefore

$$\tilde{S}^*(\omega) = \tilde{P}^*_{NRZ}(\omega) \sum_{n=0}^{N-1} \alpha_n e^{-j\omega T_c} \cdot H_{\tilde{s}}(\omega), \tag{4.14}$$

where assuming that $\tilde{p}_{NRZ}(t)$ is real. $\tilde{P}_{NRZ}(\omega) = \tilde{P}_{NRZ}^{*}(-\omega)$. Thus the transfer function of the SAW matched filter is:

$$H(\omega) = \left\{ \tilde{P}_{NRZ}(\omega) H_{\tilde{s}}(\omega) \sum_{n=0}^{N-1} \alpha_n e^{j\omega n T_c} \right\} e^{-j\omega T} \quad . \tag{4.15}$$

Since $T = NT_c$.

$$H(\omega) = \tilde{P}_{NRZ}(\omega) H_{\tilde{s}}(\omega) \sum_{n=0}^{N-1} \alpha_n e^{-j\omega (N-n) T_c} \quad . \tag{4.16}$$

One can see that the above equation represents a tapped delay line such as the one drawn in Figure 4.9. followed by a filter with transfer function $\tilde{P}_{NRZ}(\omega) H\tilde{s}(\omega)$.

Once the signal has been despread. it is sampled and passed through the DPSK demodulator as shown in Figure 4.10. It is assumed that the sampling is done at every $t = mT$.

The signal at the output of the SAW matched filter. $\tilde{r}(t)$. can be calculated as follows. At the output of the SAW matched filter. $\tilde{R}(\omega) = R(\omega) H(\omega)$. where $H(\omega)$ is the frequency response of the SAW matched filter. $R(\omega)$ is the Fourier Transform of the detected signal. Assuming that the signal was transmitted and received perfectly by the system. we will set $r(t) = s(t)$ for the purpose of this analysis.

$S(\omega)$ can be calculated from (4.9) as follows:

$$
\begin{aligned}
S(\omega) &= \int_{-\infty}^{\infty} s(t)\, e^{-j\omega t} dt \\
&= \int_{-\infty}^{\infty} K \sum_{m} \hat{d}_m \left[\sum_{n=mN}^{(m+1)N-1} \alpha_n \tilde{p}_{NRZ}(t - nT_c) \right] \cos\omega_0 t e^{-j\omega t} dt
\end{aligned}
\tag{4.17}
$$

$S(\omega)$ takes on the same form as $\tilde{S}(\omega)$ in (4.12). Similarily to (4.14) it may be written

$$
S(\omega) = K\tilde{P}_{NRZ}(\omega)\, H_s(\omega) \sum_{m} \hat{d}_m \left[\sum_{n=mN}^{(m+1)N-1} \alpha_n \right] e^{-j\omega n T_c} . \tag{4.18}
$$

Thus $\tilde{R}(\omega)$ can written as follows

$$
\begin{aligned}
\tilde{R}(\omega) = &\left[K\tilde{P}_{NRZ}(\omega)\, H_s(\omega) \sum_{m} \hat{d}_m \left[\sum_{n=mN}^{(m+1)N-1} \alpha_n \right] e^{-j\omega n T_c} \right] \\
&\left[\tilde{P}_{NRZ}(\omega)\, H_{\tilde{s}}(\omega) \sum_{n=0}^{N-1} \alpha_n e^{-j\omega (N-n) T_c} \right].
\end{aligned}
\tag{4.19}
$$

Comparing $H_s(\omega)$ and $H_{\tilde{s}}(\omega)$, one can see that they are equal. Setting $K = (N-1) - n$, $\tilde{R}(\omega)$ now becomes:

$$\tilde{R}(\omega) = K[\tilde{P}_{NRZ}(\omega) H_s(\omega)]^2 \left(\sum_m \hat{d}_m \sum_{n=mN}^{(m+1)N-1} \alpha_n e^{-j\omega n T_c} \right) \cdot \sum_{n=0}^{N-1} \alpha_n e^{-j\omega(k+1)T_c} \tag{4.20}$$

The last term of the above equation represents the tapped delay line of the SAW matched filter. The last two terms combined represent the response of the tapped delay line to data multiplied by a PN-sequence (to which the tapped delay line is matched).

Looking at the diagram of the tapped delay line (Figure 4.9), if the input is a PN-sequence with terms (α_0, α_1, ..., α_{N-1}), one can see that after N delays the terms of the sequence and the taps of the delay line will match perfectly. At this point the output of the tapped delay line will be N no matter what the sequence, assuming that the (α_n) take on values $\{-1, +1\}$. With only the matching PN-sequence one would then expect look like the waveform shown in Figure 4.11.

This output can be seen as a cross-correlation of two identical sequences, where one of the sequences is truncated to N values with the rest set to zero. Specifically at every $(m+1)NT_c$, the output of the tapped delay line corresponds to the $\theta(0)$ of the matching sequence. $\theta_{x,y}(l)$, the cross-correlation function of a pair of PN-sequences was defined in (3.1). The autocorrelation of a sequence (x_n) is simply $\theta_{x,x}(l) = \theta(l)$.

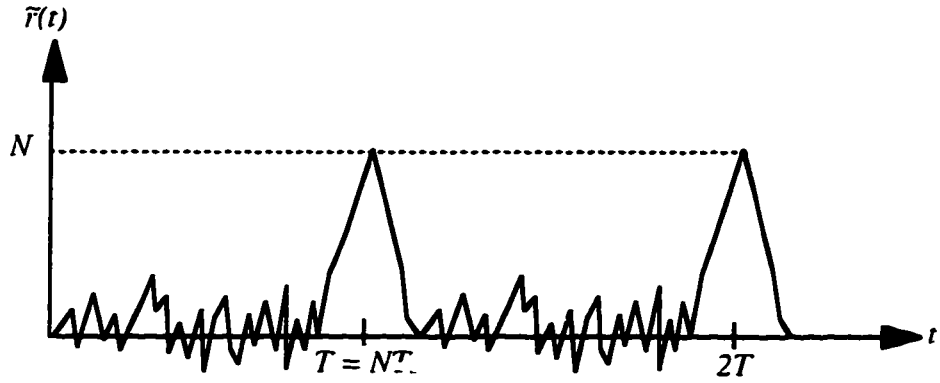


Figure 4.11 Expected output of SAW matched filter

The output terms of the tapped delay line in the SAW matched filter over several data bits can be calculated explicitly based on the last two terms of (4.20) . Considering the ith. and $(i+1)$th data bits and assuming the sequence period $N = 3$. the total output of the tapped delay line can be shown to be given by:

(4.21)

$$
\begin{aligned}
&[\hat{d}_{i-1}(\alpha_0^2+\alpha_1^2+\alpha_2^2)]\,e^{-j\omega 3iT_c} + [\hat{d}_{i-1}(\alpha_1\alpha_0+\alpha_2\alpha_1)+\hat{d}_i(\alpha_0\alpha_2)]\,e^{-j\omega(3i+1)T_c} \\
&+ [\hat{d}_{i-1}(\alpha_2\alpha_0)+\hat{d}_i(\alpha_0\alpha_1+\alpha_1\alpha_2)]\,e^{-j\omega(3i+2)T_c} + [\hat{d}_i(\alpha_0^2+\alpha_1^2+\alpha_2^2)]\,e^{-j\omega[3(i+1)]T_c} \\
&+ [\hat{d}_i(\alpha_0\alpha_1+\alpha_1\alpha_2)+\hat{d}_{i+1}(\alpha_0\alpha_2)]\,e^{-j\omega[3(i+1)+1]T_c} \\
&+ [\hat{d}_i(\alpha_0\alpha_2)+\hat{d}_{i+1}(\alpha_0\alpha_1+\alpha_1\alpha_2)]\,e^{-j\omega[3(i+1)+2]T_c} \\
&+ \left[\hat{d}_{i+1}\left(\alpha_0^2+\alpha_1^2+\alpha_2^2\right)\right]e^{-j\omega[3(i+2)]T_c}
\end{aligned}
$$

The expression in (4.21) is in the frequency domain. The exponential terms thus represent a delay of a certain length. One can see that (very conveniently) once every NT_c, just when the taps of the delay line line up with the coefficients of the PN-sequence, it should be possible to recover a data bit without it being corrupted by a previous bit. This occurs when the coefficient of the delay in the above equation is a multiple of $N = 3$. More specifically one can see that by sampling at the appropriate times (or by choosing terms with a specific delay)—here every $t = (m+1)T$—undesirable terms can be eliminated.

Through sampling, only terms with a delay which is a multiple of N are retained. The output of the tapped delay line at $t = (m+1)T$ can now be written as

$$\sum_{m} \hat{d}_m \theta(0)\, e^{-j\omega(m+1)T} \quad . \tag{4.22}$$

where $\theta(0) = \sum_{n=0}^{N-1} \alpha_n^2$.

The output of the SAW filter $\tilde{r}(t)$ is

$$\tilde{r}(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \tilde{R}(\omega)\, e^{j\omega t} d\omega \tag{4.23}$$

At $t = (m+1)T$, it is given by:

(4.24)

$$\tilde{r}([m+1]T) = \frac{1}{2\pi}\int_{-\infty}^{\infty} K[\tilde{P}_{NRZ}(\omega)H_s(\omega)]^2\left[\sum_m \hat{d}_m\theta(0)e^{-j\omega(m+1)T}\right]e^{j\omega(m+1)T}d\omega$$

$$= \sum_m \hat{d}_m\theta(0)K\left(\frac{1}{2\pi}\int_{-\infty}^{\infty}[\tilde{P}_{NRZ}(\omega)H_s(\omega)]^2 d\omega\right)$$

The last term of the above equation represents the combined energy of the signals $\tilde{p}_{NRZ}(t)$ and $h_s(t)$, denoted by E. Thus (4.24) becomes

$$\tilde{r}([m+1]T) = KE\theta(0)\sum_m \hat{d}_m \,. \tag{4.25}$$

Passing the signal $\tilde{r}(t)$ through the differential decoder, shown in Figure 4.7, one obtains the output signal $q(t)$ given by:

$$q([m+1]T) = KE\theta(0)\sum_m \hat{q}_m \,. \tag{4.26}$$

where $\hat{q}_m = \hat{d}_m\hat{d}_{m-1}$. Recall that $\hat{d}_m = d_m\hat{d}_{m-1}$. So, $\hat{q}_m = d_m\hat{d}_{m-1}\hat{d}_{m-1}$. Since $\hat{d}_m$ takes on the values $\{-1, +1\}$, one can see that $\hat{d}_m^2 = 1$. thus $\hat{q}_m = d_m$. and

$$q([m+1]T) = KE\theta(0)\sum_m d_m \,. \tag{4.27}$$

The above analysis therefore shows that it is possible to correctly recover data. which is differentially encoded. spread by a PN-sequence and modulated onto a carrier by using the SAW filter in Figure 4.9 matched to the input PN-sequence followed by a DPSK demodulator.

4.2.5 Interpolation and Decimation

As is shown in Figure 4.4. interpolation and decimation are used on several occasions in the hybrid network simulation. As was explained earlier in Section 4.1.1. the interpolation and decimation modules are required to provide the conversion between modules using different sampling rates. The decimation module is also used to provide the appropriate sampling following the SAW filter to correctly recover the received data.

Interpolation is defined as an operation performed on a discrete-time signal to increase its sampling rate. Suppose that we have a sequence of samples as follows:

$$x[n] = x_c[nT] \tag{4.28}$$

The sampling rate of this signal is $B = 1/T$. It can be increased to $B' = 1/T'$ by replacing T with $T' = T/L$. where L is the interpolation or upsampling factor. In effect each sample of the original sequence $x[n]$ is repeated L times in the new sequence

$$x'[n] = x_c(nT') \tag{4.29}$$

Decimation was mentioned earlier in Chapter 3 of this thesis in the context of m-sequences. Contrary to interpolation, the sampling rate of a signal is decreased with decimation by a factor M where $T'' = T \cdot M$, where T is the sampling time period of the input signal to the decimation module. Here one of every M samples of the original sequences appears in the new sequence, which can be written as:

$$x''[n] = x_c(nT'') \tag{4.30}$$

Since the hybrid network is asynchronous, it may be assumed that the signals being sampled at the output of the Butterworth filters and the output of the SAW matched filters will be delayed with respect to the original data sequence. This delay must be taken into account in the decimation module in order to produce a recovered signal that is as error-free as possible. We have found that it was to set this delay correctly was difficult, but that it was an important element to obtain low bit error rates at the output of the network.

4.2.6 Logical to Real Conversion

In a "real life" network, the differential encoding of the user's information bits and the spreading of the data would have of course involved real values and equipment. For example, the user's data would probably be represented by a series of positive and negative pulses, as would the output of the Gold sequence generator.

In the case of a simulation, convenience and memory space savings favour performing logical operations on the user data and PN-sequence, which take on values $\{0, 1\}$ rather than $\{+1, -1\}$. The equipment which generates and processes the data is

assumed to be ideal except for producing Gaussian white noise. which is added to the signal just prior to the optical transmitter (the AWGN is discussed in Section 4.2.8). It is however necessary to move from logical to real values prior to the phase modulator. as this is a continuous-time and analog component of the network.

The logical-to-real conversion module performs a double role. It converts the set of samples taking on values {0. 1} at its input into samples taking on values {+1. −1}. It also takes the sample values. which at its input are contained in a set of blocks with 32 samples per block. and places them into an array of real numbers. At the output of the logical-to-real conversion. each sample takes on the value +1.0 or −1.0 and is represented by a variable in an array of type double float.

4.2.7 Frequency Upconverter and Downconverter

The frequency upconverter uses the user's binary data to modulate the phase of a subcarrier. The subcarrier is a sinusoid of frequency f_C. which in the case of the reference user was set to 3.0*E*9 Hz. This value was chosen to meet several requirements. The laser used for the simulation was found to have a bandwidth of about 6 GHz: this limits f_C to less than this value. However due to the very small value of the sampling period T_s. it was necessary to set f_C as well as the data and chip rates as high as possible. It can be seen that with the chosen f_C. about 133 samples per carrier period will be needed. Recall that the sampling period was set to 2.5*E*-12 s. A reasonably high frequency carrier is also desirable

to meet the condition that $f_c \gg 1/T$, where T is the data bit duration. Satisfying this condition makes downconversion at the receiver easier. Figure 4.12 represents the model of the frequency upconverter.

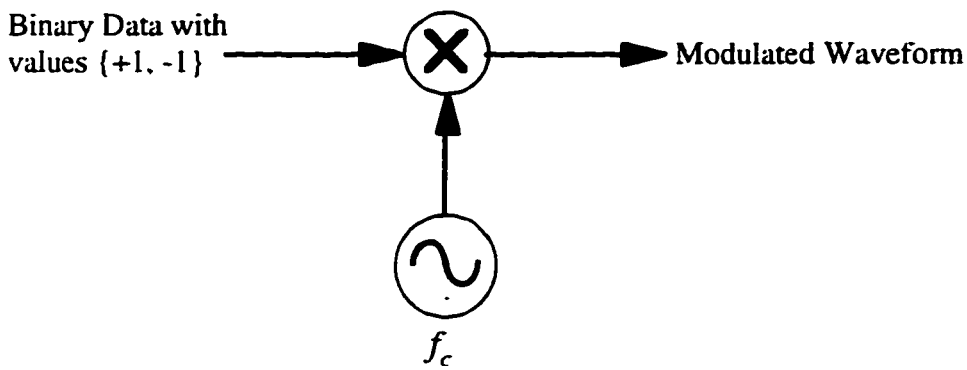


Figure 4.12 Block diagram of frequency upconverter

The block diagram of the downconvertor is shown in Figure 4.13. A received signal is first bandpass filtered to remove extraneous noise and a constant component created by the optical receiver. Next it is multiplied by a sinusoid whose frequency is matched to that of the carrier of the reference user. A pass through a lowpass filter completes the downconversion operation.

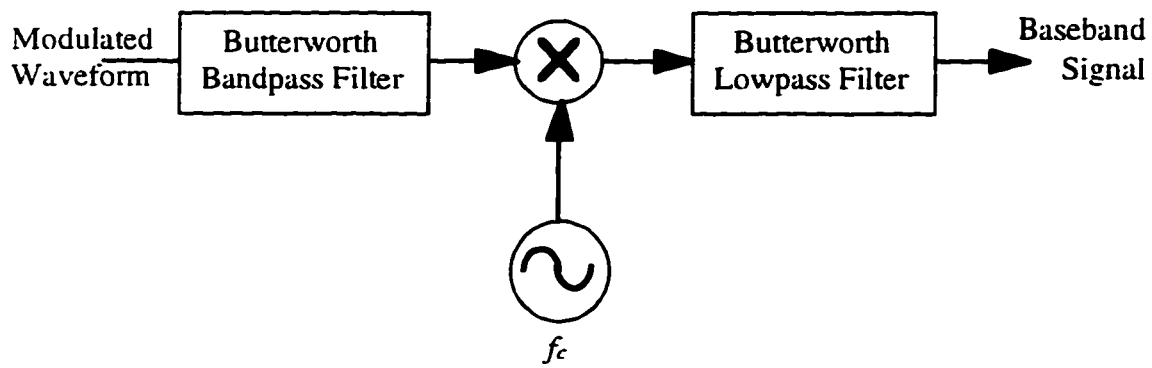


Figure 4.13 Block diagram of frequency downconverter

Both the lowpass and bandpass filters are second order Butterworth filters. The filtering operations were performed in the time domain. Although using a FFT is usually recommended[23], the time-domain approach was preferred in our case due to the large amount of data to be processed.

The most general linear filter takes a sequence x_k of input points and produces a sequence y_n of output points by the formula

$$y_n = \sum_{k=0}^{M} b_k x_{n-k} + \sum_{j=1}^{N} a_j y_{n-j} \tag{4.31}$$

Here the $M+1$ coefficients b and the N coefficients a are fixed and define the filter response. The filter described by (4.31) produces each new output value from the current and M previous input values, and from its own N previous output values. If $N = 0$, so that there is no second summation in (4.31), then the filter is nonrecursive or finite impulse response (FIR). If $N \neq 0$, then it is called recursive or infinite impulse response (IIR). The response of an IIR filter is not necessarily infinitely long; typically its response will drop off exponentially at late times, rapidly becoming negligible. Note that Butterworth filters are IIR filters.

The relation between the b_k's and a_j's and the filter response function $H(f)$ is

$$H(f) = \frac{\sum_{k=0}^{M} b_k e^{-2\pi i k (fT_s)}}{1 - \sum_{j=1}^{N} a_j e^{-2\pi i j (fT_s)}} \tag{4.32}$$

where T_s as usual is the sampling interval. To design a filter. it is necessary to obtain a suitable but preferably small set of values for the b_k's and a_j's from the desired $H(f)$.

The frequency response of a continuous-time analog lowpass Butterworth filter is

$$H(f) = \frac{1}{1 + j\left(\frac{f}{f_{3dB}}\right)^N} \tag{4.33}$$

where N and f_{3dB} respectively represent the order and the cutoff or 3dB frequency of the filter.

Several techniques are available to map the continuous-time frequency response $H(f)$ onto $H(z)$. the discrete-time frequency response. or in other words to obtain values for the b_k's and a_j's of the filter. We simply used values for the filter coefficients as computed by the software package Matlab for a second order Butterworth. The block diagram of a general 2nd order Butterworth filter is shown in Figure 4.14. The values of the coefficients used are those given below.

In Matlab the command [B, A] = BUTTER(N, w_n) designs an N'th order lowpass digital Butterworth filter and returns the filter coefficients in length N+1 vectors B and A. The cut-off frequency ω_n must be within the range $0.0 < \omega_n < 1.0$ with 1.0 corresponding to half the sampling rate. In our case the cutoff frequency was set to f_{3dB} = 7.5E8 Hz and the sampling rate was f_s = 4.0E11. This gave ω_n = 0.00375. With N = 2. Matlab returned the following filter coefficients:

B = 1.0E–4 * {0.3441. 0.6882. 0.3441}: A = {1.0000. –1.9833. 0.9835}

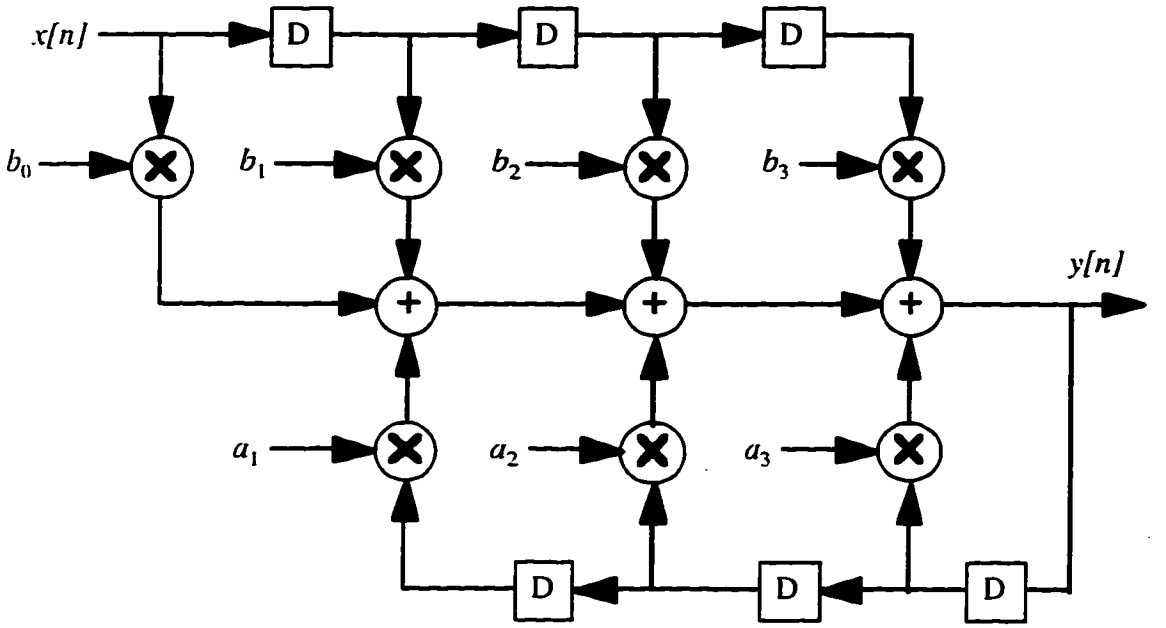


Figure 4.14 Model of general 2nd order Butterworth filter

If ω_n is a two-element vector. $\omega_n = [\omega_1, \omega_2]$. Matlab returns a bandpass filter of order $2N$ with passband $\omega_1 < \omega < \omega_2$. For the reference user whose carrier frequency is 3.0E9 Hz. ω_1 was set to 0.01125 and ω_2 to 0.01875. Setting $N = 1$. Matlab gave the following:

$$B = \{0.0116, 0.0000, -0.0116\}: A = \{1.0000, -1.9747, 0.9767\}$$

The lowpass filter produced the frequency response shown in Figure 4.15. Figure 4.16 shows the frequency response of the bandpass filter. These responses while adequate do not have a particularly steep roll-off beyond the cut-off frequency. This is primarily due to the low order of the filters. However in practice higher order filters are difficult to build and the use of 2nd order filters is not uncommon.

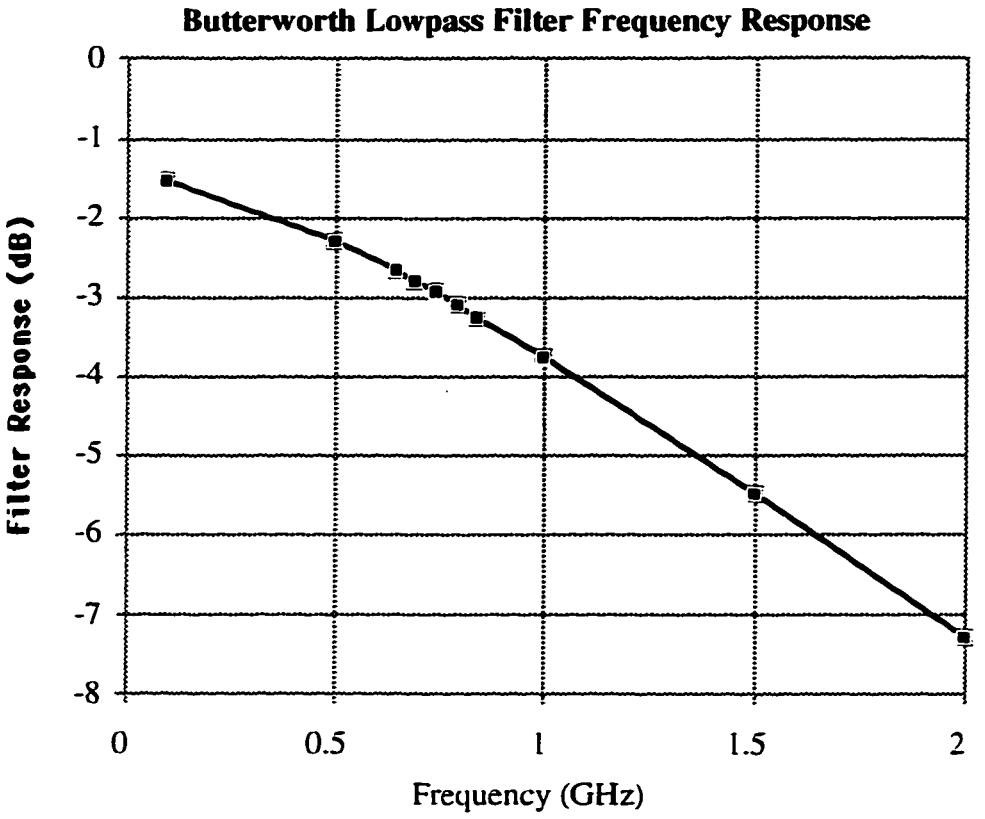


Figure 4.15 Frequency response of Butterworth lowpass filter

It was found in running early simulations that the Butterworth filters introduce both a substantial delay and some distortion in the signal waveform. The delay was largely compensated for during the first decimation of the received signal. but the distortion may have caused the network's bit error rate performance to be degraded. It was also found that there was some discrepancy between the frequency response of the lowpass and bandpass filters. However to the best of our knowledge. the program for the modules was correct and the filter parameters were correctly specified.

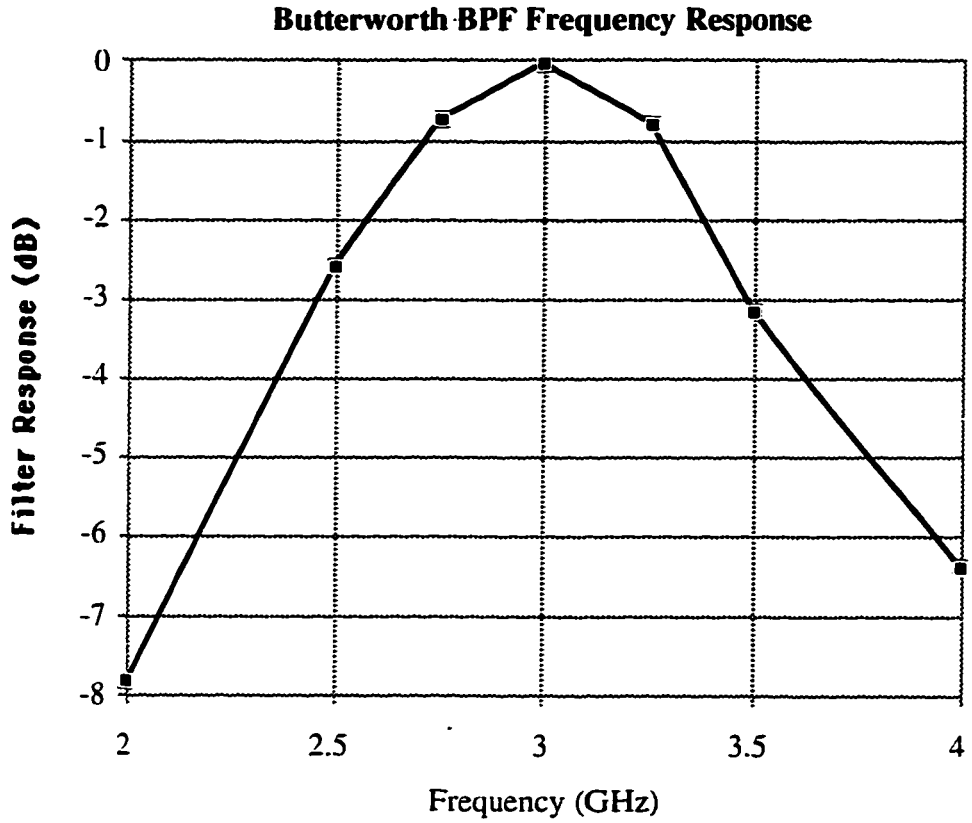


Figure 4.16 Frequency response of Butterworth bandpass filter

4.2.8 Additive White Gaussian Noise

The phase modulated signal in Figure 4.4 is combined with similar signals from other users in the network and white Gaussian noise is added to this combined signal. This AWGN represents the thermal noise that would have been introduced by the electronic equipment used to generate the users' data signals and process them prior to optical transmission. AWGN is also added to the received signal at the optical receiver, as thermal noise is also generated during the detection, despreading, and demodulation processes.

The AWGN module outputs a sequence of real random values with a Gaussian distribution given a desired variance as input. The values produced by the AWGN function are computed by applying a transformation method on a pair of uniform random numbers[23] produced by the same random number generator described in Section 4.2.1

Given a pair of random deviates x_1 and x_2 with a uniform probability distribution. it is possible to modify x_1 and x_2 in such a way to produce a pair of random deviates y_1 and y_2 with a normal distribution. The required transformation. called the Box-Muller transformation is given by

$$\begin{aligned} y_1 &= \sqrt{-2\sigma^2 \ln x_1}\cos 2\pi x_2 \\ y_2 &= \sqrt{-2\sigma^2 \ln x_1}\sin 2\pi x_2 \end{aligned} \tag{4.34}$$

where σ^2 is the variance of the Gaussian probability distribution function. An alternate way to calculate y_1 and y_2 avoids the use of trigonometric functions. In this case instead of using two uniform deviates (x_1, x_2) to represent a random point inside a unit square. we use two values (v_1, v_2) to represent a random point inside a unit circle. v_1 and v_2 can be obtained from (x_1, x_2) by

$$\begin{aligned} v_1 &= 2x_1 - 1. \\ v_2 &= 2x_2 - 1. \end{aligned} \tag{4.35}$$

The pair of Gaussian random deviates. y_1 and y_2 for their part are given by,

$$\begin{aligned} y_1 &= \sqrt{-2\sigma^2 \ln R}\frac{v_1}{\sqrt{R}}. \\ y_2 &= \sqrt{-2\sigma^2 \ln R}\frac{v_2}{\sqrt{R}}. \end{aligned} \tag{4.36}$$

where $R = v_1^2 + v_2^2$ represents the distance of a random point from the center of the unit circle. R must be between 0 and 1 for the transformation of (4.36) to be valid. In the program code for the AWGN module, values for v_1 and v_2 were always selected so that this condition on R was respected.

For the purpose of the simulation σ^2 was set to 6.624E–11. The origin of this value is explained in Section 4.2.11 which describes the optical receiver which also suffers from thermal noise.

4.2.9 Optical Transmitter

In the hybrid network, the optical transmitter shown in Figure 4.4, consists of a semiconductor laser preceded by a couple of operations to bias the input current to the laser appropriately. Figure 4.17 shows the block diagram of the optical transmitter. In Figure 4.17 m' is the optical modulation index, here it is given by

$$m' = \frac{I_{sp}}{I_{DC}} \tag{4.37}$$

where the current at the input of the laser is $I_{in} = I_{DC} + I_{sp}\sin\omega t$. The term I_{DC} is a constant bias current. Generally, the optical modulation index, m is defined in terms of the emitted signal power. It can be related to m'. In the network under study, the laser is intensity modulated. The input current, through a judicious choice of m' and I_{DC} is such that it is always above the threshold current.

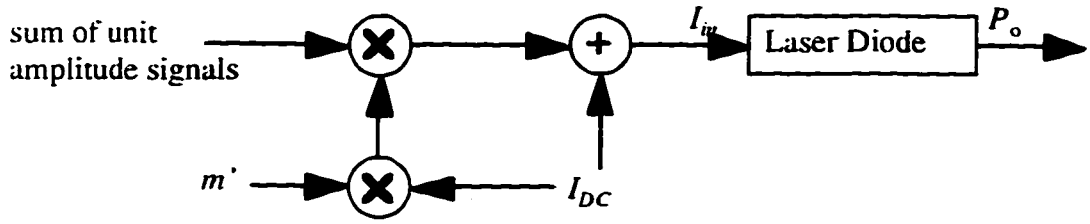


Figure 4.17 Block diagram of optical transmitter

The laser diode in a fiber optic system can represent a major source of degradation, as it is both non-linear and noisy. In this section we will examine key laser parameters to choose values for some of these parameters and to determine how well the simulated laser compares to the theoretical model or to other results in the literature. We will present a linearized transfer function for the laser diode, determine the steady-state values of the carrier and photon densities (N and S respectively), examine the lasers' nonlinear distortion, and look into its noise (RIN) characteristics. A similar investigation was done by Neusy[21] who also simulated a laser diode. We compare many of our results with his.

A) Linearized Laser Transfer Function

Semiconductor lasers were reviewed in Chapter 2. They can be modelled using the rate equations given in (2.2) and (2.3). These equations can be linearized using a perturbation analysis given by Lau[25] and also used by Neusy. In the analysis, the current, carrier density, and photon density are expressed as the sum of a steady-state and a term for small signal fluctuations from the steady-state. These expressions for the three variables are

$$\begin{aligned} N(t) &= \bar{N} + \Delta N \\ S(t) &= \bar{S} + \Delta S \\ I(t) &= \bar{I} + \Delta I \end{aligned} \tag{4.38}$$

Substituting the above into the rate equations and using the fact that $\frac{I}{V} - g(\bar{N} - N_0)[1 - \varepsilon\bar{S}]\bar{S} - \frac{\bar{N}}{\tau_{sp}} = 0$ and $\Gamma g(\bar{N} - N_0)[1 - \varepsilon\bar{S}]\bar{S} - \frac{\bar{S}}{\tau_p} + \Gamma\beta\frac{1\bar{N}}{\tau_{sp}} = 0$. the new expression for the rate equations can be reduced to

$$\begin{aligned} \frac{d}{dt}\Delta N &= A_1\Delta N + A_2\Delta S + \frac{\Delta I}{V} \\ \frac{d}{dt}\Delta S &= A_3\Delta N + A_4\Delta S \end{aligned} \tag{4.39}$$

where

$$\begin{aligned} A_1 &= -g[1 - \varepsilon\bar{S}]\bar{S} - \frac{1}{\tau_{sp}} \\ A_2 &= -g(\bar{N} - N_0)[1 - 2\varepsilon\bar{S}] \\ A_3 &= \Gamma g[1 - \varepsilon\bar{S}]\bar{S} + \Gamma\beta\frac{1}{\tau_{sp}} \\ A_4 &= \Gamma g(\bar{N} - N_0)[1 - 2\varepsilon\bar{S}] - \frac{1}{\tau_p} \end{aligned} \tag{4.40}$$

The system described by (4.39) and (4.40) is linear and it is now possible to apply linear systems theory for second order systems by taking the Fourier transform of the expressions in (4.39). The following new expression can be obtained:

$$\frac{\Delta S}{\Delta I} = \frac{A3}{V}\frac{1}{-\omega^2 + A_4A_1 - A_2A_3 + j\omega(-A_1 - A_4)} \tag{4.41}$$

This second order transfer function has two poles. From the location of these poles, some of the laser characteristics, such as its stability, bandwidth and response type, will be discussed. For any linear second order system, we define ω_n to be the natural undamped frequency and ξ as the damping ratio. In terms of the coefficients of the transfer function, these two parameters are given by

$$\omega_n^2 = A_1A_4 - A_2A_3$$
$$\xi = \frac{(-A_1 - A_4)}{2\sqrt{A_1A_4 - A_2A_3}} \tag{4.42}$$

Poles will be given by $-\alpha \pm j\omega$ where $\alpha = \xi\omega_n$ and $\omega = \omega_n\sqrt{1-\xi^2}$. The significance of ω_n is that it gives use the distance of the poles from the origin. The damping ratio ξ, gives an indication of the type transient response that may be expected from the laser. For $\xi < 0$, we have an unstable system. For $0 < \xi < 1$, we have an underdamped system with some overshoot in the transient response, and for $\xi > 1$ we have an overdamped system with no overshoot in the transient response. In addition if $\xi < 0.707$, resonance will occur at $\omega_p = \omega_n\sqrt{1-\xi^2}$. If there is resonance in the linear transfer function, it will also appear in the distortion spectrum and could affect the information signal transmitted by the laser.

The expressions for the A_i's in (4.40) show that the laser response is dependent on intrinsic laser parameters, such as ε, β, Γ, and g. The response is also dependent on the steady-state values of carrier and photon densities, $\bar{N}$ and $\bar{S}$. Both densities are heavily related to the bias current, especially the photon density, which is essentially proportional to the bias current minus the threshold. The bias current therefore affect the transfer function as well as the nonlinear distortion spectrum and the RIN spectrum of the laser.

The intrinsic laser parameters used for the simulation are given in Table 4.1. They are the parameters for a high-speed GaAlAs single-mode laser diode (Ortel LS-620) as given by Way [26]. Following Neusy, we modified the gain compression, so that $\varepsilon = 2.6E\text{-}23$ rather than $3.8E\text{-}23$. With the new value of ε the laser has a higher bandwidth of 6 GHz, which is common for high-speed lasers.

Table 1: Laser Parameters

Parameter	Description	Value
V	Volume of active region times electron charge	$1.44E\text{–}35\ m^3c$
τ_p	photon lifetime	2.0 ps
τ_{sp}	carrier lifetime	3.72 ns
N_0	transparent carrier density	$4.6E24\ m^{-3}$
g	optical gain constant	$1.0E\text{–}12\ s^{-1}m^{-3}$
Γ	optical confinement factor	0.646
β	spontaneous emission factor	0.001
ε	gain compression factor	$2.6E\text{–}23\ m^{-3}$
I_{th}	threshold current	21 mA

We obtained the transfer function of the laser diode both theoretically and through simulation. The theoretical transfer function can be obtained by taking the magnitude of (4.41). With the simulation, the simplest method to obtain the transfer function is to input a single carrier at each frequency and record the response. We have plotted the laser's transfer function for a bias current of 75 mA. In Figure 4.18. It can be seen that our simulation of the laser produces a comparable transfer function compared to the theoretical

curve and Neusy's simulation[21]. It may be observed from Figure 4.18 that the laser does indeed have a bandwidth of 6 GHz and that with a 75 mA bias the laser experiences little resonance, which is desirable.

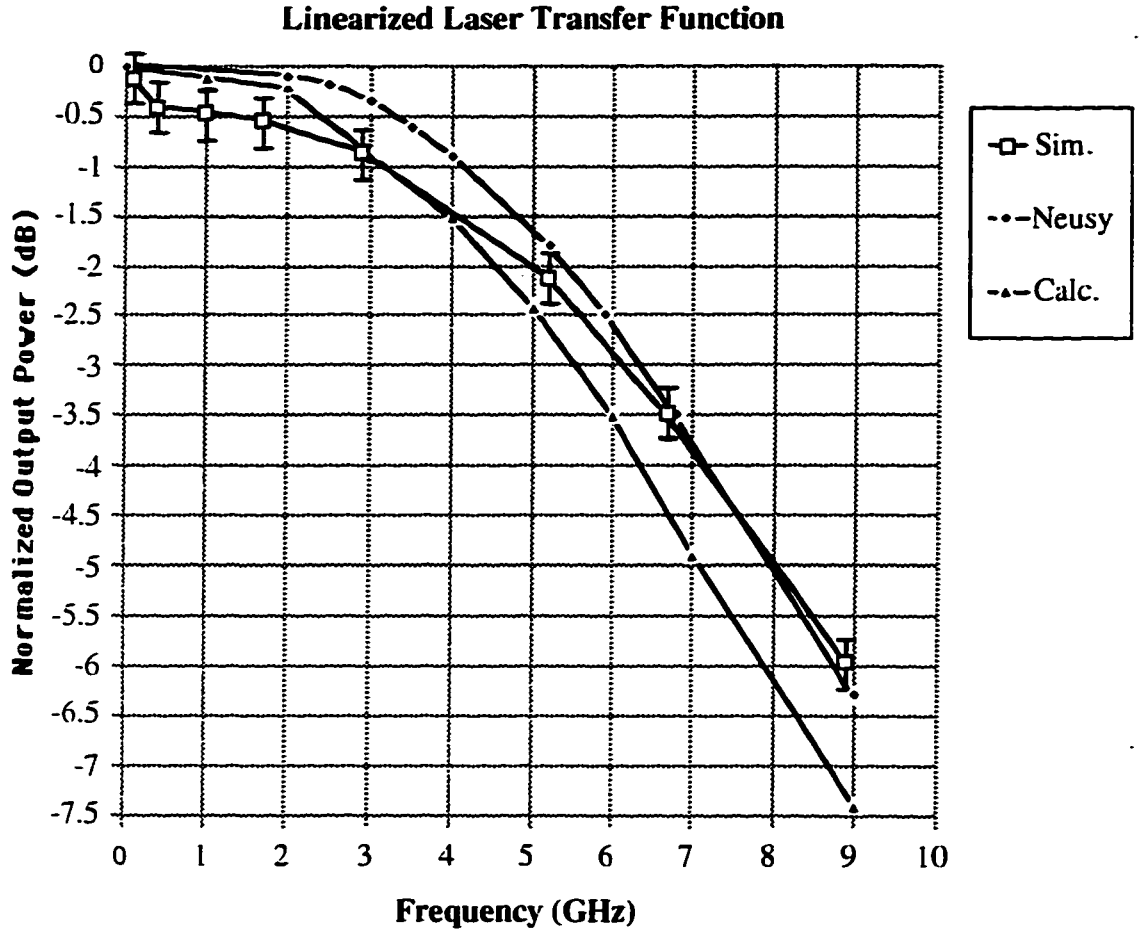


Figure 4.18 Laser diode transfer function

The stability of the laser can also be commented on using the laser parameters. A linear system will be stable as long as the real part of its poles is negative, i.e., it is necessary that $\xi > 0$. Recalling the definition of ξ, it can be seen that the condition for stability is met

when $-A_1 > A_4$. which implies that $g(1-\varepsilon\bar{S})\bar{S} + \frac{1}{\tau_{sp}} + \frac{1}{\tau_p} - \Gamma g(\bar{N}-N_0)(1-2\varepsilon\bar{S}) > 0$. The last two terms relate to the rate of photons entering and leaving the active region. For any physical laser $\xi > 0$ and therefore lasers are stable.

B) Steady-State Condition of the Laser

Neusy used the phase plane analysis technique to further examine the laser's response for different bias currents. This technique can be applied to certain second-order differential equations. It eliminates the independent variable. In the case of the laser rate equations. the independent variable *t* can be removed by dividing one rate equation by the other to obtain

$$\frac{dN}{dS} = \frac{\frac{I_A}{V} - \frac{N}{\tau_{sp}} - g_0(N-N_0)[1-\varepsilon S]S}{\Gamma g(N-N_0)[1-\varepsilon S]S - \frac{S}{\tau_p} + \Gamma\beta\frac{N}{\tau_{sp}}} . \qquad (4.43)$$

The phase plane is formed by plotting the slopes of the carrier density against the photon density. We used this technique in SPW. taking advantage of its graphical capabilities to determine the steady-state values of the laser. The plot of the carrier to photon density. with the parameters of the laser set for stable operation. is in the general shape of a spiral. The steady-state values of *N* and S are given by the point at the center of the spiral. For a bias current of 75 mA. they were found to be as follows:

$N = 5.48470E24$ $\qquad$ $S = 4.82581E21$

These values were used as the initial conditions of the carrier and photon densities (i.e., $N(t = 0)$ and $S(t = 0)$) for the network simulations as they would minimize the time taken for the laser to settle into its steady-state operation.

C) Non-linear Distortion

In the analysis to determine the laser's transfer function, the laser was treated as a linear device, however a non-linear coupling of the carrier and photon densities in the stimulated emission term laser rate equations indicates that the laser is not linear. The distortion spectrum of the laser can be obtained analytically, through perturbation analysis, in much the same way as the linear transfer function was obtained. With this technique, every type of distortion can be determined with respect to the linear term, e.g., $S^{2\omega_1}/S^{\omega_1}$, $S^{\omega_1-\omega_2}/S^{\omega_1}$. The drawback of this method is that it becomes very tedious, due to the complexity of the rate equations, for higher order terms. Also having generated expressions for the various distortion terms, the question remains as to how to calculate the total distortion from all the distortion terms. It is nonetheless instructive to see how the second order distortion spectrum varies with bias current and laser parameters for example.

Proceeding with the perturbation analysis [26], the approach is to write down for the current, and carrier and photon densities a series with components at all the original frequencies as well as at the beat frequencies generated by the laser non-linearities. Substituting these expressions into the rate equations, applying trigonometric identities to

deal with the products of sinusoids, and disregarding all higher order terms with respect to lower order terms. The carrier and photon densities, and the current are now written as follows:

$$\begin{aligned} N(t) &= N^0 + \Re\left(N^{\omega_1} e^{j\omega_1 t}\right) + \Re\left(N^{2\omega_1} e^{j2\omega_1 t}\right) \\ S(t) &= S^0 + \Re\left(S^{\omega_1} e^{j\omega_1 t}\right) + \Re\left(S^{2\omega_1} e^{j2\omega_1 t}\right) \\ I(t) &= I^0 + \Re\left(I^{\omega_1} e^{j\omega_1 t}\right) \end{aligned} \tag{4.44}$$

Substituting into the rate equations, we obtain the following pair of second order differential equations:

$$\begin{aligned} \frac{dN^{2\omega_1}}{dt} &= B_1 N^{2\omega_1} + B_2 S^{2\omega_1} + B_3 N^{\omega_1} S^{\omega_1} + B_4 S^{\omega_1} S^{\omega_1} \\ \frac{dS^{2\omega_1}}{dt} &= B_5 N^{2\omega_1} + B_6 S^{2\omega_1} + B_7 N^{\omega_1} S^{\omega_1} + B_8 S^{\omega_1} S^{\omega_1}. \end{aligned} \tag{4.45}$$

where

$$
\begin{aligned}
B_1 &= -g\,[1-\varepsilon\bar{S}]\,\bar{S}-\frac{1}{\tau_{sp}} \\
B_2 &= -g\,(\bar{N}-N_0)\,[1-2\varepsilon\bar{S}] \\
B_3 &= -\frac{1}{2}g\,[1-2\varepsilon\bar{S}] \\
B_4 &= \frac{1}{2}g\,(\bar{N}-N_0)\,\varepsilon \\
B_5 &= \Gamma g\,[1-\varepsilon\bar{S}]\,\bar{S}+\Gamma\beta\frac{1}{\tau_{sp}} \\
B_6 &= \Gamma g\,(\bar{N}-N_0)\,[1-2\varepsilon\bar{S}]-\frac{1}{\tau_p} \\
B_7 &= \frac{1}{2}\Gamma g\,[1-2\varepsilon\bar{S}] \\
B_8 &= -\frac{1}{2}\Gamma g\,(\bar{N}-N_0)\,\varepsilon
\end{aligned}
\tag{4.46}
$$

All terms in the differential equations have frequency $2\omega_1$ The coefficients B_3, B_4, B_5, and B_8 refer to the components resulting from the products $S^{\omega_1}\cdot S^{\omega_1}$ and $S^{\omega_1}\cdot N^{\omega_1}$ that occur at $2\omega_1$. The components at d.c. have been ignored. Note that in these equations, it is the linear terms. $S^{\omega_1}\cdot S^{\omega_1}$ and $S^{\omega_1}\cdot N^{\omega_1}$ that drive these second order distortions. Since the equations in (4.44) are linear, their Fourier transform can be taken to obtain the spectrum. This has been done by Darcie [27] for the expression of (4.44), and is an extension of work by Lau[25]. Darcie has obtained the following expression for the distortion:

$$
\frac{S^{2\omega_1}}{S^{\omega_1}} = m\frac{\omega^2}{\omega_n^2 g\,(2\omega)} \tag{4.47}
$$

where m is the optical modulation index in this case. The term $g(\omega)$ is the small signal frequency response derived earlier and given by:

$$g(\omega) = \frac{\omega_n^2}{-\omega^2 + 2j\xi\omega_n + \omega_n^2} \tag{4.48}$$

Neusy has plotted this function for bias currents of 50 mA and 75 mA, and has noted the impact of the bias current on the distortion level. With a 50 mA bias the distortion peaks at approximately 3 GHz, whereas the distortion reaches a maximum around 8 GHz when the bias current is 75 mA. This indicates that the choice of a 75 mA bias current is desirable in order to avoid excessive distortion in the laser's 6 GHz signal band. The above analysis can be extended to third order distortion [27].

Neusy has presented several results, obtained via simulations, on the total distortion level caused by the laser. During test simulations of the laser alone, we compared the distortion level of our laser simulation to Neusy's and found reasonably good agreement considering the uncertainty inherent in the results. Our results assuming a bias current of 75 mA, with Neusy's superimposed are shown in Figure 4.19.

D) Relative Intensity Noise

The relative intensity noise of a laser is an undesirable effect in laser diodes caused by the stochastic nature of light as was discussed in Chapter 2. As was mentioned then, each term of the laser rate equations is represented by a shot noise process which can be modelled as a Poisson random process. The constant values taken on by the terms of the

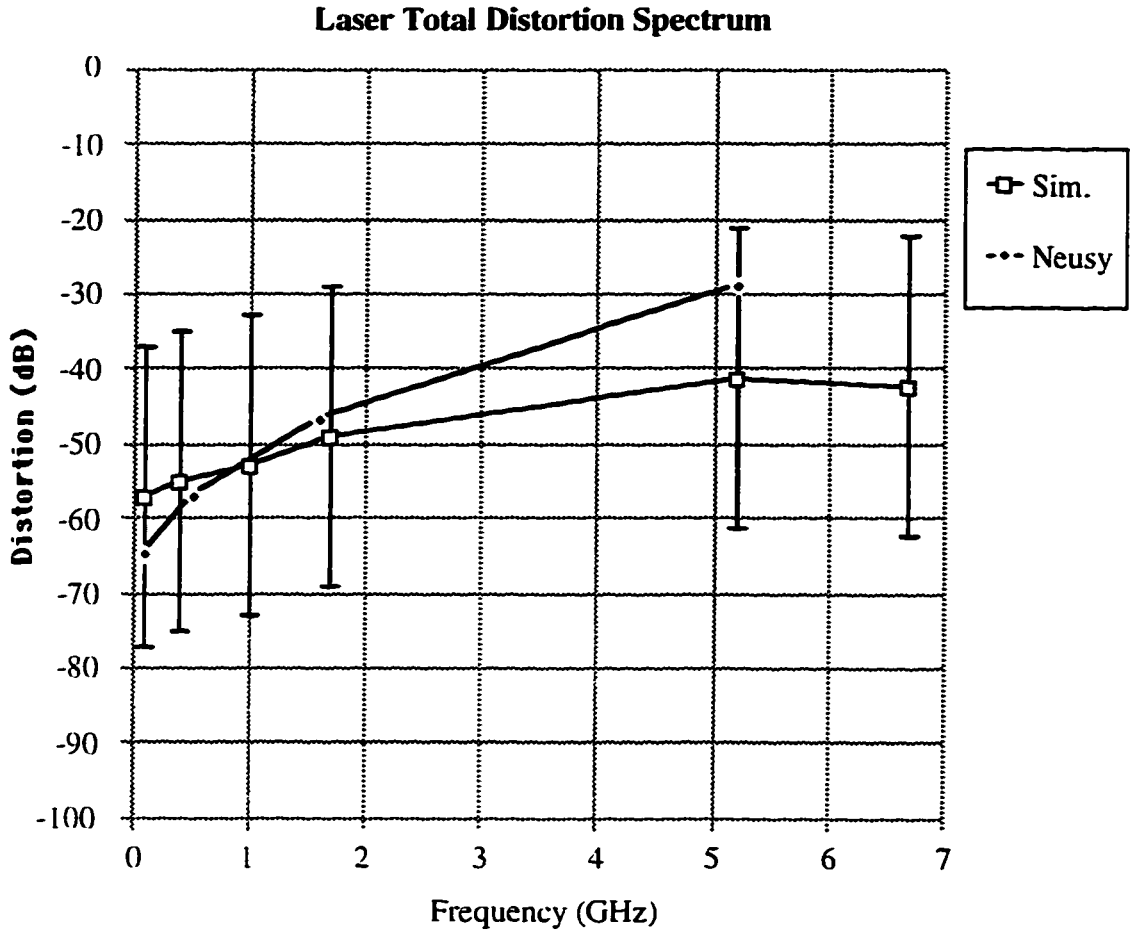


Figure 4.19 Laser diode non-linear distortion spectrum

rate equations without RIN are the mean of this Poisson process. The fluctuations of each term of the rate equations can thus be obtained in a manner very similar to method used to generate the shot noise in the receiver. This method is explained in detail in Section 4.2.11.

This method of obtaining the noisy laser's output is a little different from the way Neusy constructed his noisy laser simulation. Neusy built a noiseless laser module and added noise with a uniform distribution to it. In our simulation of the laser, a submodule generates the shot noise with a Poisson distribution that is the cause of RIN. The output of this submodule is added directly to all components of the laser function.

An analytical expression for RIN in terms of the laser's intrinsic parameters can be developed. Following Neusy's approach all the fluctuations from the terms of each laser rate equation can be lumped together. A pair of modified rate equations can be written as follows:

$$\begin{aligned} \frac{d}{dt}N'(t) &= \frac{I_A}{q} - \frac{N'}{\tau_{sp}} - g'(N' - N'_0)\,[1 - \varepsilon S']\,S' + F_n(t) \\ \frac{d}{dt}S'(t) &= \Gamma g'(N' - N'_0)\,[1 - \varepsilon S']\,S' - \frac{S'}{\tau_p} + \Gamma\beta\frac{N'}{\tau_{sp}} + F_n(t), \end{aligned} \tag{4.49}$$

where all the terms have the same meanings as in the previous rate equations except that here N' is the number of carriers. S' is the number of photons. q is the electron charge. $F_n(t)$ is the carrier random fluctuations, and $F_s(t)$ represents the photon random fluctuations. $F_n(t)$ and $F_s(t)$ are Langevin noise terms. N'_0 is the electron number for transparency and g' . is the optical gain with respect to photon number.

The spectral densities can be written as follows.

$$\langle F_n^2(f)\rangle = \frac{I_A}{q} - \frac{N'}{\tau_{sp}} - g'N'[1-\varepsilon S']S' + g'N'_0[1-\varepsilon S']S'$$

$$\langle F_s^2(f)\rangle = \Gamma g'N'[1-\varepsilon S']S' + \Gamma g'N'_0[1-\varepsilon S']S' + \frac{S'}{\tau_p} + \Gamma\beta\frac{N'}{\tau_{sp}} \qquad (4.50)$$

$$\langle F_s(f)F_n(f)\rangle = \Gamma g'N'_0[1-\varepsilon S']S' + \Gamma g'N'[1-\varepsilon S']S' + \Gamma\beta\frac{N'}{\tau_{sp}}.$$

We would like to find the RIN spectrum from the modified rate equations. The RIN spectrum is given by

$$RIN(\omega) = \frac{\langle \Delta S(\omega)\rangle^2}{\langle S(\omega)\rangle^2} \qquad (4.51)$$

which is a slightly modified version of (2.4).

To obtain the RIN spectrum. the modified rate equations were linearized by applying a perturbation analysis. The spectrum was then obtained by taking the Fourier transform. The approach will now be explained further. Substituting $N(t) = \bar{N}' + \Delta N$ and $S(t) = \bar{S}' + \Delta S$ into (4.48). the following set of differential equations is obtained

$$\frac{d}{dt}\Delta N = C_1\Delta N + C_2\Delta S + F_n(t) \qquad (4.52)$$

$$\frac{d}{dt}\Delta S = C_3\Delta N + C_4\Delta S + F_s(t).$$

where

$$
\begin{aligned}
C_1 &= -g'[1-\varepsilon \bar{S}']\,\bar{S}' - \frac{1}{\tau_{sp}} \\
C_2 &= -g'(\bar{N}' - N'_0)\,[1-2\varepsilon \bar{S}'] \\
C_3 &= \Gamma g'[1-\varepsilon \bar{S}']\,\bar{S}' + \Gamma\beta\frac{1}{\tau_{sp}} \\
C_4 &= \Gamma g'(\bar{N}' - N'_0)\,[1-2\varepsilon \bar{S}'] - \frac{1}{\tau_p}
\end{aligned}
\tag{4.53}
$$

Note that the C_i's are not the same as the A_i's derived for the linear transfer function of the laser. since N' and S' refer to the steady state carrier number and photon number respectively.

The expressions in (4.52) are for a linear system. Applying the Fourier transform. the spectrum of the photon fluctuations is given by:

$$
\Delta S(\omega) = \frac{C_3 F_n(\omega) + (j\omega - C_1) F_s(\omega)}{(C_1 C_4 - C_2 C_3 - \omega^2) - j\omega(C_1 + C_4)} \,. \tag{4.54}
$$

Applying the definition for RIN spectrum. one obtains the following[28]:

$$
\mathrm{RIN}(\omega) = \frac{1}{\bar{S}'^2} \frac{C_3^2 \langle F_n(\omega)^2 \rangle + (\omega^2 + C_1^2)\langle F_s(\omega)^2 \rangle - 2C_1 C_3 \langle F_s(\omega) F_n(\omega) \rangle}{(C_1 C_4 - C_2 C_3 - \omega^2)^2 - \omega^2 (C_1 + C_4)^2} \tag{4.55}
$$

The RIN spectrum was calculated according to (4.54) for a bias current of 75 mA and plotted along with the RIN spectrum of our simulation obtained on SPW and Neusy's simulation result in Figure 4.20. It can be seen that some differences exist between the different approaches. but that the results agree for the working bandwidth of 6 GHz of the

laser. It is possible that a different value for the gain compression factor, ε was used in Neusy's simulation than was used in our case as well as for the theoretical curve. A smaller value of ε would account for the peak in Neusy's RIN curve.

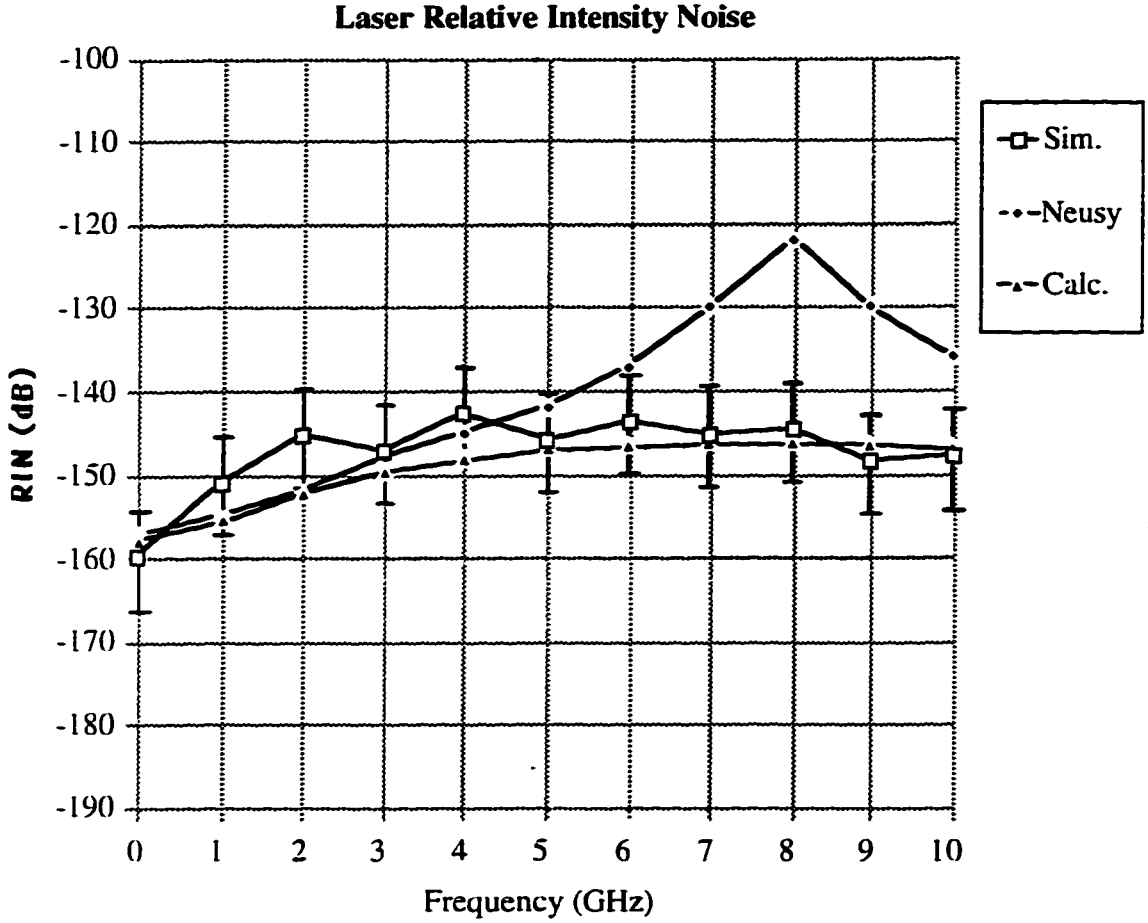


Figure 4.20 Laser RIN spectrum

4.2.10 Optical Fiber and Passive Star Coupler

For the purposes of the simulation, the single mode glass fiber used for the simulation is assumed to be ideal, i.e., linear and non-dispersive. The attenuation of the fiber is assumed to be 3.0 dB per km which is considered normal for fiber operating in the 800 nm window. A laser-to-fiber coupling loss of 5 dB is also assumed. The average distance between users in the network is taken to be about 2 km.

The passive star coupler is a glass device which broadcasts light from several input ports to multiple output ports. It is shown in Figure 4.21. Assuming that a coupler connects N users, it has $2N$ ports. As was explained in Chapter 2, the star coupler distributes power equally to each of its output ports from any one of its input ports. We have assumed that the coupler has an excess loss of L_E. The transmission efficiency for each port of the coupler is $\frac{1}{N}$ and the total distribution loss, including the loss due to two couplers each having loss L_C, is given by

$$L = -10\log\left(\frac{1}{N}\right) + L_E + 2L_C$$

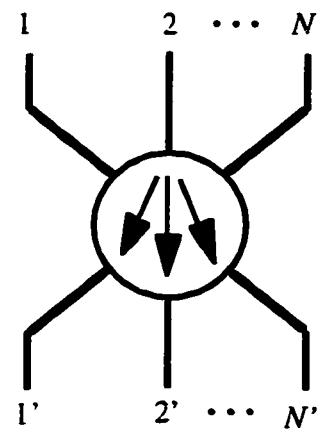


Figure 4.21 Star coupler model

4.2.11 Optical Receiver

In this section the optical receiver will be described in detail. The block diagram of a general direct detection lightwave receiver is shown in Figure 4.22. This configuration was used for the simulation: the choice of a photodetector, an amplifier and a filter will be explained along with values for key parameters.

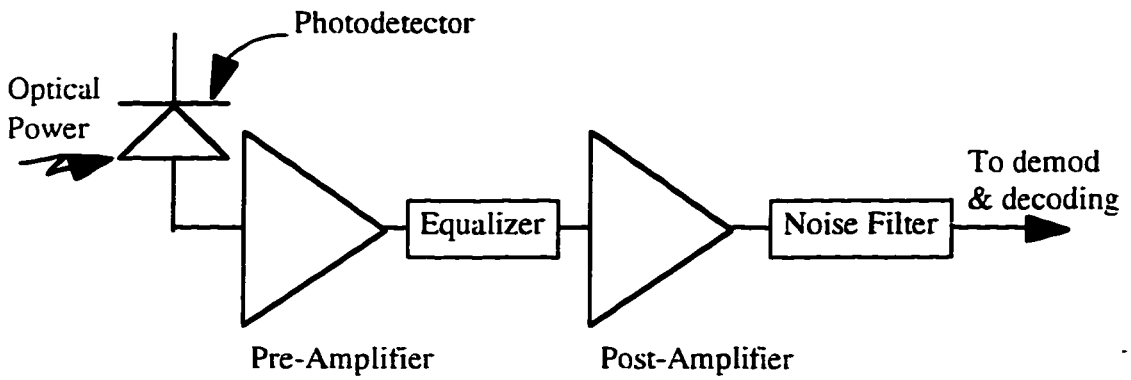


Figure 4.22 Block diagram of general optical receiver

In a lightwave receiver the choice of the photodetector and the pre-amplifier is particularly important, as these two components essentially determine how well an optical system is able to perform.

The two most popular types of photodetector are the p-i-n photodiode and the avalanche photodiode. As was discussed in Chapter 2, important characteristics of photodetectors include responsivity, dark current, transit time and bandwidth. APD's, due to their internal gain have a greater sensitivity. However their transit times are longer and they suffer from additional noise due to the avalanche multiplication factor, as compared to p-i-n diodes. The p-i-n diodes also offer the advantages of lower cost and greater linearity.

The p-i-n diode was chosen as the photodetector in the network's optical receiver for these reasons, and because the application under consideration is not a long haul link where maximum sensitivity would be essential.

The measure of conversion efficiency of a photodetector is its responsivity as defined in equations (2.6) and (2.8) of Chapter 2. Responsivities for p-i-n diodes operating around 800 nm are in the range of 0.5 to 0.7 A/W.

Two major sources of signal degradation that occur during detection are thermal noise and shot noise. Thermal noise is caused by the random motion of thermally agitated free electrons in a conductor. It is prevalent in resistors at room temperature. In optical receivers it originates in the load resistance R_L of the p-i-n diode and in the resistors in the pre-amplifier. Thermal noise in the load resistor, for example, can be described in terms of the mean square value of the thermal noise current i_T:

$$\overline{i_T^2} = \frac{4KT_R B}{R_L} \quad . \tag{4.56}$$

where K is Boltzmann's constant. T_R is the temperature of the resistance, and B is the electrical bandwidth of the receiver.

In semiconductor photodiodes, shot noise is a set of random fluctuations superimposed on the output current of the detector. It arises from the random recombination of free electrons and holes. The generation of an electron-hole pair results from the

absorption of a photon, and the signal emerging from the photodetector is dictated by the statistics of photon arrivals. The probability of detecting z photons in time τ when it is expected on average to detect z_m obeys the Poisson distribution, i.e.,

$$P(z) = \frac{z_m^z e^{-z_m}}{z!} \tag{4.57}$$

where z_m is at once the mean and the variance of the distribution. The average number of electrons generated in time τ is equal to the average number of photons detected this time period, z_m, where $z_m = r_e\tau = \left(\frac{\eta\lambda P_0}{h \cdot c}\right)\tau$ and r_e is the electron generation rate and P_0 is the optical power.

Shot noise can also be described from another perspective, in terms of its current power. Since shot noise has a white frequency distribution, hence its power spectral density is constant. For a receiver with bandwidth B, the shot noise current power is,

$$\overline{i_s^2} = 2eI_pB \tag{4.58}$$

Equation (4.58) assumes that there is no dark current, otherwise $(I_p + I_D)$ would be used.

The operation of a p-i-n diode with only a dark current and shot noise can be simulated following the block diagram in Figure 4.23. In the block diagram, an ideal p-i-n diode first converts the received optical power P_0 into a current I_p according to (2.7) of Chapter 2. This current represents the expected value for any p-i-n diode. The expected number of electrons generated over time τ, z_m, can be obtained from I_p as follows:

$$z_m = \frac{1}{e} \int_{(n-1)\tau}^{n\tau} I_P(t)\, dt \equiv \frac{\tau}{e} I_P(n\tau) \tag{4.59}$$

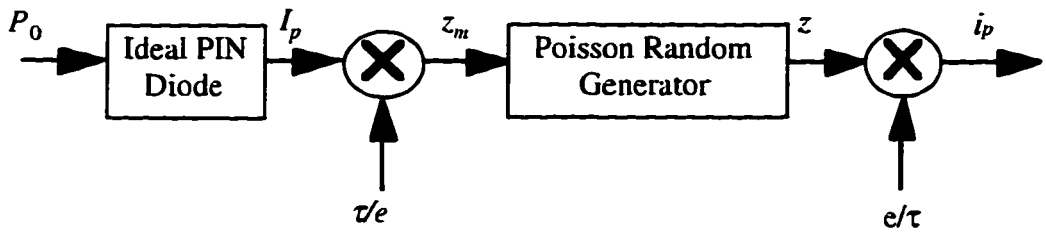


Figure 4.23 Model of p-i-n diodes with shot noise

Thus z_m can be obtained from I_p by dividing by the electronic charge e and multiplying by a time interval τ. provided that τ is chosen sufficiently small so that I_p can indeed be considered approximately constant. One can also see that consecutive values of z_m are independent of each other. since "mutually exclusive" time intervals are considered. The quantity z_m is used as the input to a Poisson random generator. This block. given a mean value z_m. produces an output z whose probability corresponds to (4.57). The actual number of electrons generated over a time τ is z. A value for the current corresponding to z is obtained by reversing the procedure by which z_m was obtained. i.e., by dividing z by τ and multiplying by e. The value thus obtained is the output current of a photodiode with shot noise. The value of τ was found such that the variance of i_p is indeed equal to (4.58). Recall that the variance σ_z^2 of the Poisson process is equal to z_m and can be written.

$$\sigma_z^2 = z_m = \left(\frac{\eta\lambda P_0}{hc}\right)\tau. \tag{4.60}$$

From this the variance of i_p, σ_P^2 . can be obtained as follows:

$$\sigma_P^2 = \left(\frac{e}{\tau}\right)\sigma_z^2 = \frac{e^2}{\tau^2}\left(\frac{\eta\lambda P_0}{hc}\right)\tau. \tag{4.61}$$

Recalling that $I_p = \frac{\eta e\lambda P_0}{h \cdot c}$, the preceding equation can be rewritten

$$\sigma_P^2 = \frac{e}{\tau} I_P \,. \tag{4.62}$$

Comparing the above equation with (4.58), one can see that τ must be set equal to $\frac{1}{2B}$, where assuming a receiver with no bandwidth limitation, $B = \frac{f_s}{2}$. Recall that f_s is the simulation sampling rate.

Following the photodetector is a first amplifier. It is often called the pre-amplifier, as it may precede other amplifiers and filters in the optical receiver. Its configuration must be chosen with care. Of the three pre-amplifier topologies most commonly used—low-impedance, high-impedance and transimpedance—the low-impedance configuration was initially chosen primarily for its simplicity. This topology also offered a good dynamic range and required no equalization. However its noise is somewhat high, its bandwidth is somewhat small, and it is only possible to improve one of these two characteristics at the expense of the other. For this reason, the low-impedance amplifier was replaced by the transimpedance amplifier, which offers a significantly wider bandwidth and better noise characteristics. The transimpedance configuration also does not require the use of equalization, however it is not as simple as the low-impedance amplifier.

An equivalent circuit of an optical receiver using a p-i-n diode and a transimpedance amplifier configuration is shown in Figure 4.24. In Figure 4.24 I_p is the noiseless photocurrent generated by the photodetector, I_D is detector's dark current. C_D and R_D are the detector's internal capacitance and resistance respectively. R_D is small enough to be considered negligible. R_L is the detector's load resistance and i_{TL} is the thermal noise current it generates. i_S is the detector's shot noise current. The basic operational amplifier (op-amp) has an open loop gain of $-G_0(f)$. and i_A and v_A are the equivalent noise sources associated with it. The feedback resistor R_F has a thermal noise current source i_{TF} associated with it.

The optical receiver's block diagram used in the simulation can be split up into three sections:

a) the generation of the detector's photocurrent and the shot noise which have already been described.

b) the addition of the thermal noise from the resistors in the receiver and the amplifier's excess noise. and

c) the receiver's overall transfer function.

We will consider first c) then discuss b).

To obtain the receiver's overall transfer function. the equivalent circuit of Figure 4.24 is redrawn in Figure 4.25. where i_{IN} is the total input current to the amplifier Also $R_{TL} = R_L \| R_A + R_D$and $C_T = C_A + C_D$.

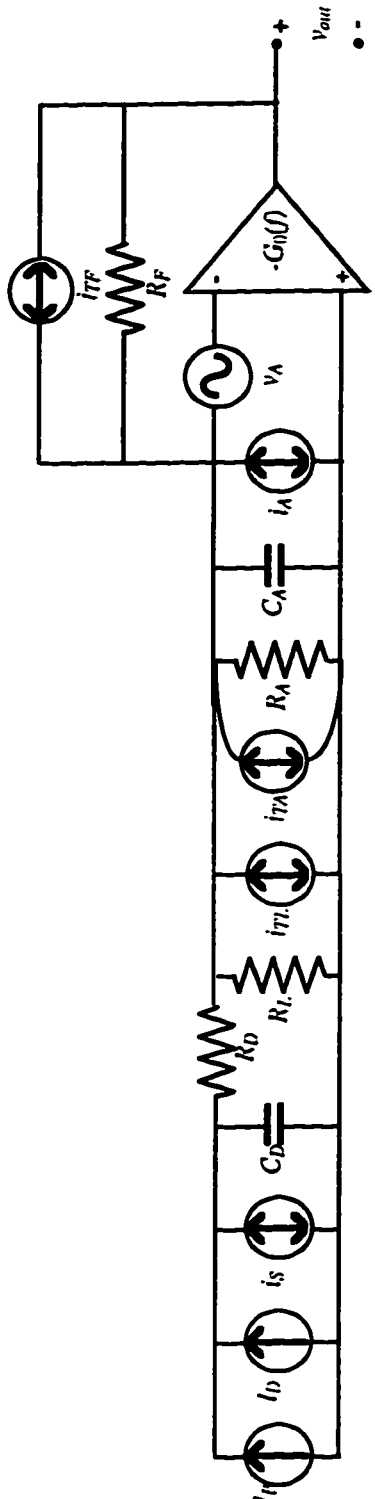


Figure 4.24 Equivalent circuit of noisy optical receiver

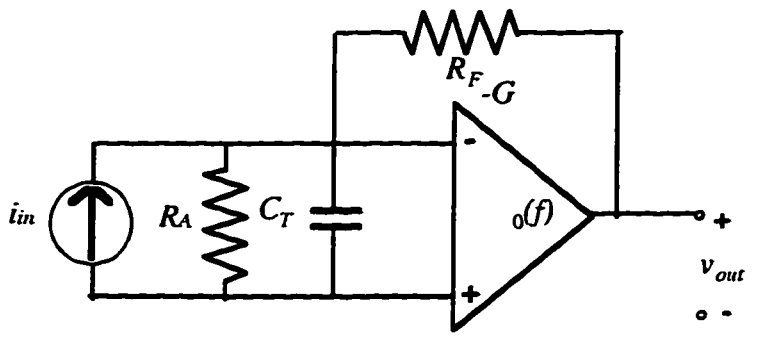


Figure 4.25 Equivalent circuit of optical receiver's amplifier

The transfer function of a transimpedance amplifier can in general be written as

$$A_f(f) = \frac{A_0(f)}{1+\beta(f)A_0(f)} = \frac{V_{out}(f)}{I_{in}(f)} \tag{4.63}$$

where $A_0(f)$ is the open-loop transfer function. $A_0(f)$ can be found by breaking the feedback loop and finding the transfer function of the resulting equivalent circuit. Replacing the amplifier by its equivalent circuit. Figure 4.25 can be redrawn as shown in Figure 4.26.

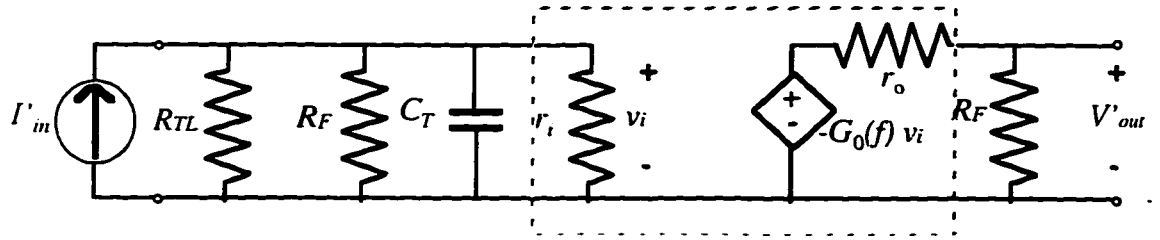


Figure 4.26 Noisy optical receiver with equivalent circuit of amplifier

From the circuit in Figure 4.26 one can obtain $A_0(f)$ after a little calculation. Assuming that $r_0 << R_{TL} << (R_{TL}$ and $r_i)$. one finds

$$A_0(f) = \frac{V'_{out}(f)}{I'_{in}(f)} = \frac{-R_F G_0(f)}{1+j2\pi f C_T R_F} \tag{4.64}$$

The value of $\beta(f)$ of (4.63) is obtained by examining the feedback circuit (in this case R_F) with the input short circuited. We find that $\beta(f)$ is given by

$$\beta(f) = \left.\frac{I_F}{V_0}\right|_{V_1 = 0} = -\frac{1}{R_F} \tag{4.65}$$

Now the closed-loop response of the amplifier can be obtained:

$$A_f(f) = \frac{-R_F G_0(f)}{[1 + j2\pi f C_T R_F] + G_0(f)} \tag{4.66}$$

If we consider $G_0(f) = G_0$. then we obtain the following transfer function for the amplifier:

$$A_f(f) = \frac{-R_F}{\frac{1}{G_0} + j2\pi f \frac{C_T R_F}{G_0} + 1} \equiv \frac{-R_F}{1 + \frac{jf}{f_1}} \tag{4.67}$$

where $f_1 = \dfrac{G_0}{2\pi R_F C_T}$.

$A_f(f)$ in this case can be simulated by a gain block having a value $-R_F$ followed by a block representing a first order Butterworth low-pass filter with 3 dB frequency $f_1 = \dfrac{G_0}{2\pi R_F C_T}$.

However in reality most op-amps do not have an infinite bandwidth; one can thus represent the op-amp as being yet another low-pass filter with the following transfer function

$$G_0(f) = \frac{G_0}{1 + \frac{jf}{f_2}} \tag{4.68}$$

The closed-loop transfer function is now calculated as follows. Recall (4.63) where.

$$A_0(f) = \frac{-R_F}{1 + j2\pi f C_T R_F} \cdot \left(\frac{G_0}{1 + \frac{jf}{f_2}} \right). \tag{4.69}$$

and let $f_0 = \frac{1}{2\pi R_F C_T}$. $A_f(f)$ can now be written as

$$A_f(f) = \frac{\frac{-R_F G_0}{[(1 + jf/f_0)(1 + jf/f_2)]}}{1 + \left\{ \frac{G_0}{[(1 + jf/f_0)(1 + jf/f_2)]} \right\}} \tag{4.70}$$

After some manipulation one can obtain.

$$A_f(f) = \frac{A_{f_0} f_n^2}{-f^2 + j_2 \xi f_n f + f_n^2} = \frac{A_{f_0}}{1 + \frac{j_2 \xi f}{f_n} - \frac{f^2}{f_n^2}} \tag{4.71}$$

where

$$\begin{aligned} f_n^2 &= \alpha f_1^2 (G_0 + 1) \\ A_{f_0} &= -R_F G_0 / (G_0 + 1) \\ \xi^2 &= (\alpha + 1)^2 / 4\alpha (G_0 + 1) \\ \alpha &= f_2 / f_0 \end{aligned} \tag{4.72}$$

In (4.71). f_n represents the natural frequency, ξ is the damping factor and α is the pole separation factor.

If one sets $\xi = \frac{1}{\sqrt{2}}$. and assuming that $G_0 \gg 1$ then the following simplifications can be made:

$\alpha \equiv 2G_0$:

$f_2 = \alpha f_0 = 2G_0 f_0$;

$A_{f_0} \equiv -R_F$;

$f_n^2 \equiv \alpha f_0^2 G_0 = (2G_0) f_0^2 G_0$.

This last relation gives $f_n = \sqrt{2} f_0 G_0$. $A_f(f)$ can now be written as

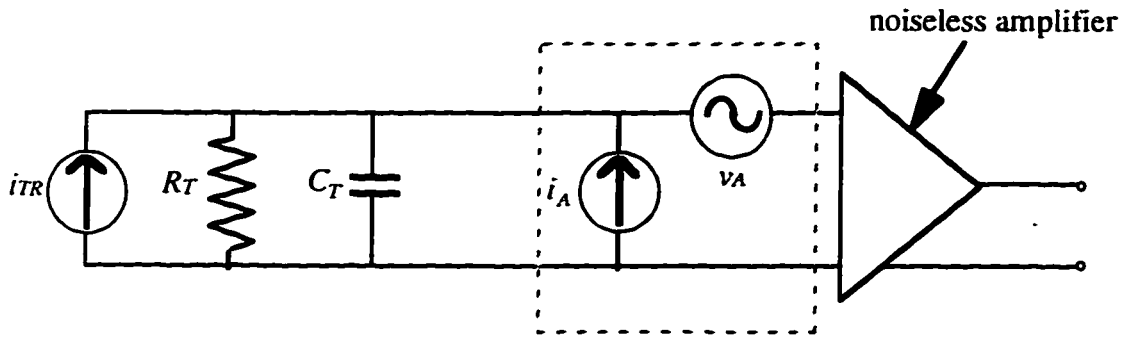
$$A_f(f) = \frac{-R_F}{1 + j\frac{\sqrt{2}f}{f_n} - \frac{f^2}{f_n}} . \tag{4.73}$$

where

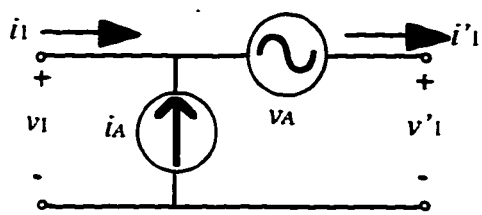
$$f_n = \sqrt{2} G_0 f_0 = \frac{\sqrt{2} G_0}{2\pi R_F C_T} . \tag{4.74}$$

The above transfer function can thus be simulated using a gain block with value $(-R_F)$ followed by a block representing a 2nd order Butterworth low-pass filter with cutoff frequency $f_n = \frac{\sqrt{2} G_0}{2\pi R_F C_T}$.

For the calculation of the thermal noise contributed by the receiver to its output signal, we go back to Figure 4.19. Considering only the noise sources now we can redraw the equivalent circuit as illustrated in Figure 4.27a, where $i_{TR} = i_{TL} + i_{TA} + i_{TF}$ and $R_T = R_L \| R_A \| R_F$.



a) Optical receiver thermal noise sources



b) Amplifier thermal noise sources

Figure 4.27 Equivalent circuit of optical receiver with thermal noise sources

The noise sources associated with the basic amplifier (i_A, v_A) are usually correlated to some degree. They can be represented in terms of two uncorrelated current sources. This procedure is necessary to be able to directly add the mean squared values of the noise sources and determine the total noise factor of the basic amplifier. Figure 4.27b shows the part of Figure 4.27a between the dashed lines.

In the circuit of Figure 4.27b. i_A will be split into two parts; one which is entirely uncorrelated and the other which is completely correlated. The current i_A can then be written as

$$i_A = i_u + i(v) . \tag{4.75}$$

where i_u represents the uncorrelated portion of i_A and $i(v)$ represents the correlated part. Since $i(v)$ is fully correlated with v_A it must be proportional to v_A. The term i_A is thus written

$$i_A = i_u + Y_C v_A \,. \tag{4.76}$$

where $Y_C = G_C + jB_C$ is called the correlation admittance. Figure 4.27b can be redrawn as shown in Figure 4.28. From this figure, one can write

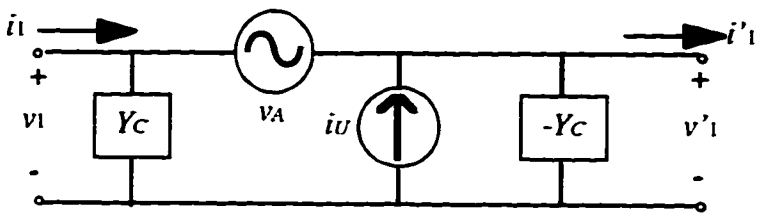


Figure 4.28 Equivalent circuit of amplifier noise sources

$$v_1 = v_A + v_1' \qquad \text{and} \qquad i_1' = i_1 + i_A \,. \tag{4.77}$$

where the relation for i_1' was obtained from Figure 4.27b. Substituting (4.69) into the above, i_1' becomes

$$i_1' = i_1 + (i_u + Y_C v_A) = i_1 + i_u + Y_C (v_1 - v_1') \,. \tag{4.78}$$

The complete circuit at the input of the noiseless amplifier can now be represented by Figure 4.29, where $i_C' = (Y_C + Y_T)\, v_A$.

Assuming maximum power transfer (we are assuming that the source—represented by the detector and its load resistance—and the amplifier have matched impedances), the noise available at the input of the noiseless amplifier originating in the conductance G_T is.

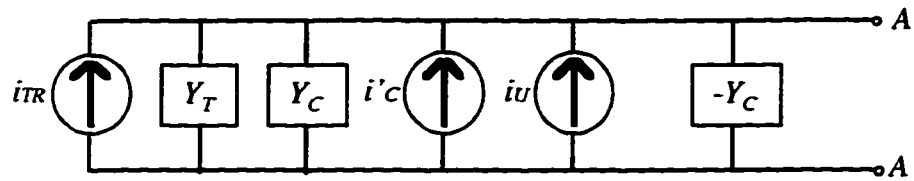


Figure 4.29 Equivalent circuit of amplifier noise sources redrawn

$$N_i = \frac{\overline{i_{TR}^2}}{4(G_T + Y_C - Y_C)} \tag{4.79}$$

The noise contributed by the noisy portion of the amplifier is:

$$N_{ni} = \frac{\overline{i'^2_C} + \overline{i_u^2}}{4(G_T + Y_C - Y_C)} = \frac{|Y_C + Y_T|^2 \overline{v_A^2} + \overline{i_u^2}}{4(G_T + Y_C - Y_C)} . \tag{4.80}$$

The noise factor of the amplifier F_n is defined as

$$F_n = \frac{GN_i + GN_{ni}}{GN_i} = 1 + \frac{N_{ni}}{N_i} . \tag{4.81}$$

where G is the available power gain of the amplifier. The equation for F_n can also be given by:

$$F_n = 1 + \frac{\overline{i_u^2} + |Y_C + Y_T|^2 \overline{v_A^2}}{\overline{i_{TR}^2}} . \tag{4.82}$$

where for the purpose of defining F_n, all the noise power values were measured at the standard reference temperature $T_0 = 290^\circ$ K. Thus

$$\begin{aligned} \overline{i_u^2} &= 4kT_0G_uB \\ \overline{v_A^2} &= 4kT_0R_A'B \\ \overline{i_{TR}^2} &= 4kT_0G_TB \end{aligned} \tag{4.83}$$

where $R_A' \neq R_A$. Substituting the above into (4.82), we get

$$F_n = 1 + \frac{4KT_0G_uB + |Y_C + Y_T|^2 4kT_0R_A'B}{4kT_0G_TB}$$
$$= 1 + \frac{G_u}{G_T} + \frac{R_A'}{G_T}[(G_C + G_T)^2 + (B_C + B_T)^2]. \quad (4.84)$$

Since T_e, the effective temperature of the amplifier can be given by $T_e = (F_n - 1)T_0$, the mean square value of the thermal noise current source at the input of the noiseless amplifier, which represents the noise contributed by the noisy amplifier, can be found. It is written as

$$\overline{i_{AN}^2} = \frac{4KT_eB}{R_F} \quad (4.85)$$

So, finally the block diagram of the optical receiver in the simulated network is shown in Figure 4.30. The values for various parameters in the simulation block diagram,

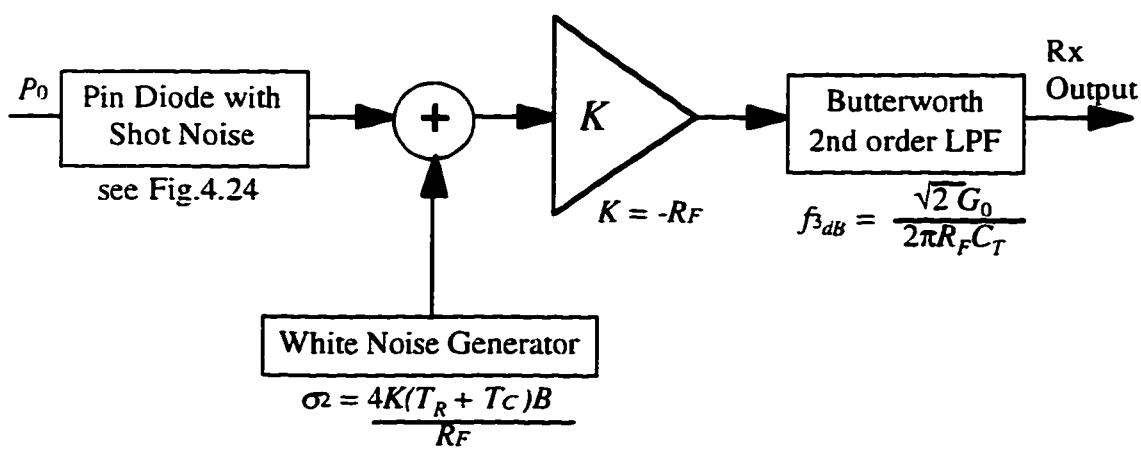


Figure 4.30 Block diagram of optical receiver module used for simulation

to be set by the user are the wavelength λ at which the p-i-n operates as well as its quantum

efficiency η, the noise bandwidth B, which is set equal to the simulation sampling frequency, the amplifier feedback resistor R_F, the amplifier effective temperature T_e, and the amplifier cutoff frequency f_{3dB}. These values were chosen as follows:

1) λ = 835 nm. The wavelength was chosen so as to be the same as the operating wavelength of the laser diode.

2) η = 0.9. This choice of gives a responsivity of R = 0.6049, which is in the middle of the range for p-i-n diodes operating around 800 nm.

3) f_s = 4.0E11 Hz. This value of f_s is required for stable operation of the laser diode as was discussed in Section 4.1.2.

4) f_{3dB} = 6 GHz. The cutoff frequency for the receiver was chosen to be as large as the bandwidth of the laser. This way the receiver does not accidentally filter out signals in the range passed by the laser.

5) R_F = 75 Ω. Since

$$R_F = \frac{\sqrt{2}G_0}{2\pi f_{3dB} C_T} \tag{4.86}$$

and assuming that f = 6 GHz, $C_T = C_A + C_D$ = 10 pF (C_D and C_A are usually found to be about 5 pF each), and $G_0 \equiv 20$. Setting G_0 = 20 (which is somewhat small) gives R_F = 75 Ω. R_F was kept small, because of the assumption that it is much smaller than $R_L \| R_A$.

6) $T_e = 865^o$ K. F_n was chosen to be 6 dB which gives $T_e = 865$ K. The choice for F_n was somewhat arbitrary, since F_n depends on such parameters as the source admittance Y_T and the correlation admittance Y_C, for which it is difficult to obtain values. This choice is considered to be about twice as bad as today's state-of-the-art optical receivers (hence it is a conservative choice).

4.2.12 Comparator

The comparator in Figure 4.31 is used to compare the reference user's initial data to the signal recovered at the output of the network. In the comparator, the user's data is first converted from logical to real values using the converter described in Section 4.2.6. It is then multiplied by the recovered signal. If this product is positive then the recovered signal matches the original data, if it is negative an error occurred in the recovery of the signal and a variable representing the number of errors is incremented by one. It has been observed that with DPSK data errors often occur in pairs as each differentially encoded data bit depends on the value of the previous bit. As a final step the number of errors is divided by the total number of bits that were simulated to obtain the simulation bit error rate.

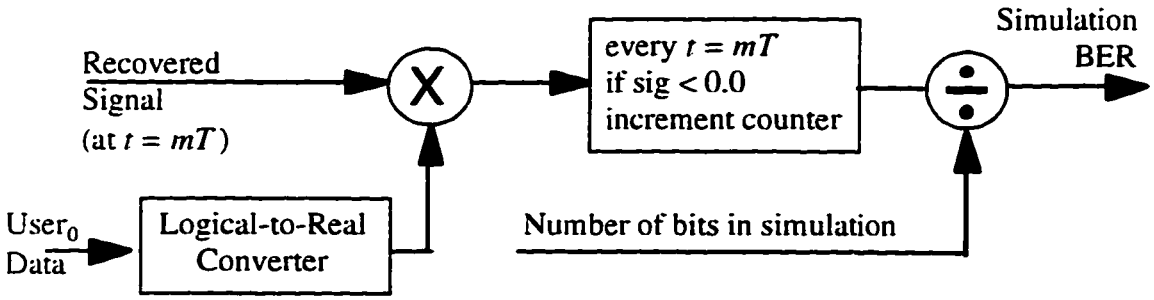


Figure 4.31 Block diagram of comparator module

Chapter 5: Performance of the CDMA/ SCM Optical LAN

In this chapter we will present evaluations of the performance of the hybrid CDMA/SCM optical local area network under study. In Section 5.1 a theoretical analysis, following the approach taken by Khaleghi in [8] is described and some preliminary results of our simulation program are presented in Section 5.2.

5.1 Theoretical Performance Analysis

The model of the system under analysis was given in Figure 1.2. During each symbol interval of duration T, the ith transmitting laser is intensity modulated by the combined signals of different users connected to that laser. The signal of each user consists of BPSK modulated data spread by a Gold sequence. Sequences are specified by the values they take on in N equal length intervals of duration T_c, where these intervals are called chips. The users' data and sequence waveforms are defined in the same manner as in (3.14) and (3.15). The signal waveform of the kth user is also similar to (3.17); it is given by

$$s_k(t) = a_k(t)\, b_k(t) \cos(\omega_c t + \theta_k) \tag{5.1}$$

In this analysis, we will assume the following without loss of generality: the quantum efficiency of the photodetector is unity, the fiber and passive star coupler are ideal and lossless. We will define M as the number of users in each cluster, where a cluster

includes only those users sharing the same light source. The number of clusters is N_s, the number of spreading codes is N_c, the number of frequencies N_f, and the number of users is N_u.

The instantaneous optical power of each laser is:

$$P(t) = \bar{P}_T\left[1 + m\sum_{k=1}^{M} S_k(t)\right] \tag{5.2}$$

where m is the optical modulation index. It will be necessary to keep m within the range $0 < m < 1/M$, assuming that the signal levels in (5.1) are normalized to unity, to avoid such effects as clipping at the laser output.

By assuming a lossless communications system, the RF photocurrent corresponding to the desired signal at each receiver is

$$A = \frac{R\bar{P}_T m}{N_s} , \tag{5.3}$$

where R is the photodector's responsivity. If $R = 1$ and $m = 1/M$ then $A = \dfrac{\bar{P}_T}{M \cdot N_s} = \dfrac{\bar{P}_T}{N_u}$

We can write the bit energy E_b as

$$E_b = \frac{A^2 T}{2} = \frac{\bar{P}_T^2}{N_u^2} \cdot \frac{T}{2} \tag{5.4}$$

The expression for received signal can be written as follows:

$$r(t) = \sum_{k=1}^{N_c} S_k(t-\tau_k) + \sum_{l=1}^{L} S_l'(t-\tau_l) + n(t) . \tag{5.5}$$

where the τ's are uniformly distributed over $[0, T]$, because users are independently sending their signals. The first term of (5.5) includes different user signals. $n(t)$ is additive white gaussian noise with a double-sided spectral density of height $N_0/2$, and the second term represents the sum of the interference signals. The latter include harmonics and intermodulation products (IMP) falling around the center frequency f_c. It is assumed here that the number of interference signals is equal to the number of codes used regardless of the number of users per cluster, since each receiver is using one frequency and rejecting all others.

Equation (5.5) can be written in greater detail as follows:

$$\begin{aligned} r(t) &= A \sum_{k=1}^{N_c} a_k(t-\tau_k)\, b_k(t-\tau_k) \cos(\omega_c t + \Theta_k) + n(t) \\ &\quad + A \sum_{l=1}^{L} w_l a_l'(t-\tau_l) \cos(\omega_c t + \Theta_l) \end{aligned} \tag{5.6}$$

where $\Theta_k = -\omega\tau_k + \theta_k$ and θ_1 is assumed to be zero without loss in generality. The parameter w_l in the last term represents the weight of each harmonic or IMP. The weights of harmonics are 6 dB less than the weight of IMPs of the same order. Only third or lower order IMPs and harmonics will be considered, as the higher order distortion is small enough to be considered negligible. For second order IMPs $a_l'(t)$ is a waveform like (3.35). Equation (3.35) can be extended to consider higher order non-linearities. Using Gold, Kasami, or m-sequences to spread the user data, results in the IMPs and harmonics having

the same effect on the system performance. For these sets of sequences, cross-correlation and autocorrelation functions have the same sidelobes and take on values from the same set[1].

If we consider the signals transmitted by the *j*th and *k*th users, we can see that the amplitude of the second order IMP is proportional to

$$a_k(t-\tau_k)\,b_k(t-\tau_k)\,a_j(t-\tau_j)\,b_j(t-\tau_j) \tag{5.7}$$

The AWGN in the system, $n(t)$ has a double-sided spectral density of height $N_0/2$ where

$$N_0 = \frac{4KTF}{R_L} + 2eI_D + (RIN)\,I_D^2 \tag{5.8}$$

In (5.8), the reader may recognize Boltzmann's constant K, T the absolute temperature, which is set to 290° K, the electron charge e, the electronic receiver's amplifier noise figure F, and R_L the photodiode load resistor with a 50 ohm nominal value, from earlier analyses in the thesis.

After detection at the optical receiver, despreading and demodulation collapse the wideband coded signal centered at frequency f_c into a narrowband baseband signal. A signal sample at the receiver's low-pass filter output can be expressed as:

$$\xi = \int_0^T r(t)\,a_1(t)\cos(\omega_c t)\,dt \quad . \tag{5.9}$$

where it is assumed that the receiver is tuned to the signal of user$_1$.

Substituting (5.6) into the above, the expression for ξ can be written as follows:

$$\begin{aligned}\xi = & A\sum_{k=1}^{N_c}\int_0^T a_k(t-\tau_k)\,b_k(t-\tau_k)\,a_1(t)\cos(\omega_c t+\Theta_k)\cos\omega_c t dt \\ & + A\sum_{l=1}^{L} w_l\int_0^T a_l'(t-\tau_l)\,a_1(t)\cos(\omega_c t+\Theta_l)\cos\omega_c t dt \\ & + \int_0^T n(t)\,a_1(t)\cos(\omega_c t)\,dt.\end{aligned} \tag{5.10}$$

Following the same procedure as in Section 3.3. of Chapter 3 (see (3.19) and (3.20)), we can rewrite (5.10) as follows:

$$\begin{aligned}\xi = & \frac{AT}{2}b_0^1 + A\sum_{k=2}^{N_c}[b_{-1}^{(k)}R_{k,1}(\tau_k) + b_0^{(k)}\hat{R}_{k,1}(\tau_k)]\cos\Theta_k \\ & + \frac{A}{2}\sum_{l=1}^{L} w_l\int_0^T a_l'(t-\tau_l)\,a_1(t)\cos(\Theta_l)\,dt + \eta,\end{aligned} \tag{5.11}$$

where η is equal to the last term of (5.10). Also recall that b_0^1 is the users$_1$'s information bit which is being detected and b_{-1}^1 is the previous bit. In (5.11), the first term represents the desired signal, while all other terms represent noise or interference. The second term represents interference from other users' signals, the third term is the interference introduced by the lasers' IMPs and harmonics, and the last term is the white noise present in the network.

A probability expression for the system under analysis can be calculated in a fairly straightforward manner if it is assumed that all the multiuser interference terms, harmonics, and IMPs are Gaussian distributed. Under the Gaussian assumption, the power of the interference terms as well as the harmonic and IMP terms is equal to their variance. The noise and interference terms of (5.11) are mutually independent.

For binary PSK modulation, the error probability is given by

$$P(e) = \frac{1}{2}\text{erfc}\left(\sqrt{\gamma_b}\right) \tag{5.12}$$

where γ_b is the bit energy-to-noise ratio. The term γ_b is half the signal-to-noise plus interference ratio. The latter is found by dividing the average signal power by the combined power of noise and interference. The average signal power is given by $\{\frac{AT}{2}\}^2$, which is the first term of (5.11) squared.

The power of the inference from other users' signals can be found following the approach of Section 3.3. The variance of the second term of (5.11) is given by

$$\begin{aligned}\text{Interference Power} &= \frac{A^2}{8T}\sum_{k=2}^{N_c}\int_0^T R_{k,1}^2(\tau) + \hat{R}_{k,1}^2(\tau)\,d\tau \\ &= \frac{A^2}{8}V = \frac{A^2}{8}\frac{2T^2}{3N}(N_c-1) = \left(\frac{AT}{2}\right)^2\frac{1}{3N}(N_c-1).\end{aligned} \tag{5.13}$$

where V was defined in (3.25) and its value given in (3.34). The power of the IMP and harmonics is given by

$$\text{Nonlinearity term Power} = \frac{A^2}{8}\sum_{l=1}^{L}\frac{n_l}{3}w_l^2\left(\frac{2T^2}{3N}\right) = \left(\frac{AT}{2}\right)^2\sum_{l=1}^{L}n_l w_l^2\left(\frac{1}{9N}\right). \quad (5.14)$$

based on the result obtained in (3.51). The power of the white noise is written as

$$\text{Noise Power} = N_0\frac{T}{4} \quad (5.15)$$

Hence the signal-to-noise plus interference ration can be found using (5.13). (5.14) and (5.15) and substituting (5.3) for A

$$SNR = 2\gamma_b = \frac{P_T^2}{(N_c-1)\frac{1}{3N}P_T^2 + \left(\sum_{l=1}^{L}n_l w_l^2\right)\frac{1}{9N}P_T^2 + N_0\frac{1}{T}N_u^2} \quad (5.16)$$

Using (5.4). the expression for SNR becomes:

$$SNR = \frac{E_b}{(N_c-1)\frac{1}{3N}E_b + \left(\sum_{l=1}^{L}n_l w_l^2\right)\frac{1}{9N}E_b + \frac{N_0}{2}} \quad (5.17)$$

It should be noted that the probability of error given in (5.12) is that obtained for a coherent system with BPSK modulated signals. In our simulation. we considered DPSK modulated data for which the error probability is given by

$$P(e) = \frac{1}{2}e^{-\gamma_b}. \quad (5.18)$$

where γ_b is the SNR of (5.17).

5.2 Hybrid Network Simulation Results

In Section 4.1.1, we briefly referred to some of the difficulties associated with the creation of the simulation program. The latest simulation program is in effect the third attempt at simulating the hybrid network. Chapter 4 hopefully provides an appreciation for the magnitude of the task involved. The difficulties with the creation of the simulation program have taken up much of the time devoted to this thesis. The simulations while functional have produced only very preliminary results.

The simulations run to date have involved a simple version of the hybrid network. Only one light source with two users is assumed and the star coupler was not included. While this represents a very limited networking scenario, it nevertheless should provide an indication of whether the hybrid network's performance is superior to that of a network using only SCM. The spreading sequences used were a pair of Gold codes of period $N = 31$. Also the simulations so far have involved relatively small numbers of samples (hundreds rather than thousands or millions) and so are not statistically extremely reliable. However they give some indication of what may be expected from longer simulations and more complex network configurations.

The preliminary performance curve for the two user network is drawn in Figure 5.1 for both a hybrid and an SCM system. The performance curve of a theoretical single-user system is also included. The curves seem to indicate that a fairly important performance improvement is obtained with the hybrid configuration. The difference between the theoretical curve and those of the simulated systems indicates that components in the

network simulation cause fairly serious degradations which are not taken into account in the theoretical case. It is not entirely clear however whether this gap could be decreased by refining the simulation program, or whether the difference in performance is simply an indication of the magnitude of the performance impairments present in a fairly realistic network. At the writing of this thesis however, it appears that the simulation program may be at least part of the problem. In particular the Butterworth filter modules have been singled out as a potential source of performance degradation. As was mentioned in Section 4.2.7, the Butterworth filters introduce both distortion and delay into the simulated signal waveform. The delay was compensated for during the decimation of the signal following the frequency downconversion. However the filters' distortion and the gentle roll-off of their frequency response may have caused the network's performance to be degraded.

An examination of the signal at the output of the downconverter revealed that higher frequency components were still present though attenuated. The high frequency components caused the amplitude of the signal to vary above and below the decision threshold very quickly (within a few sampling periods). The distortion of the filters appeared to exacerbate these variations. It is possible that a number of errored samples may thus have been present at the output of the frequency converter and then at the output of the decimation module that follows it.

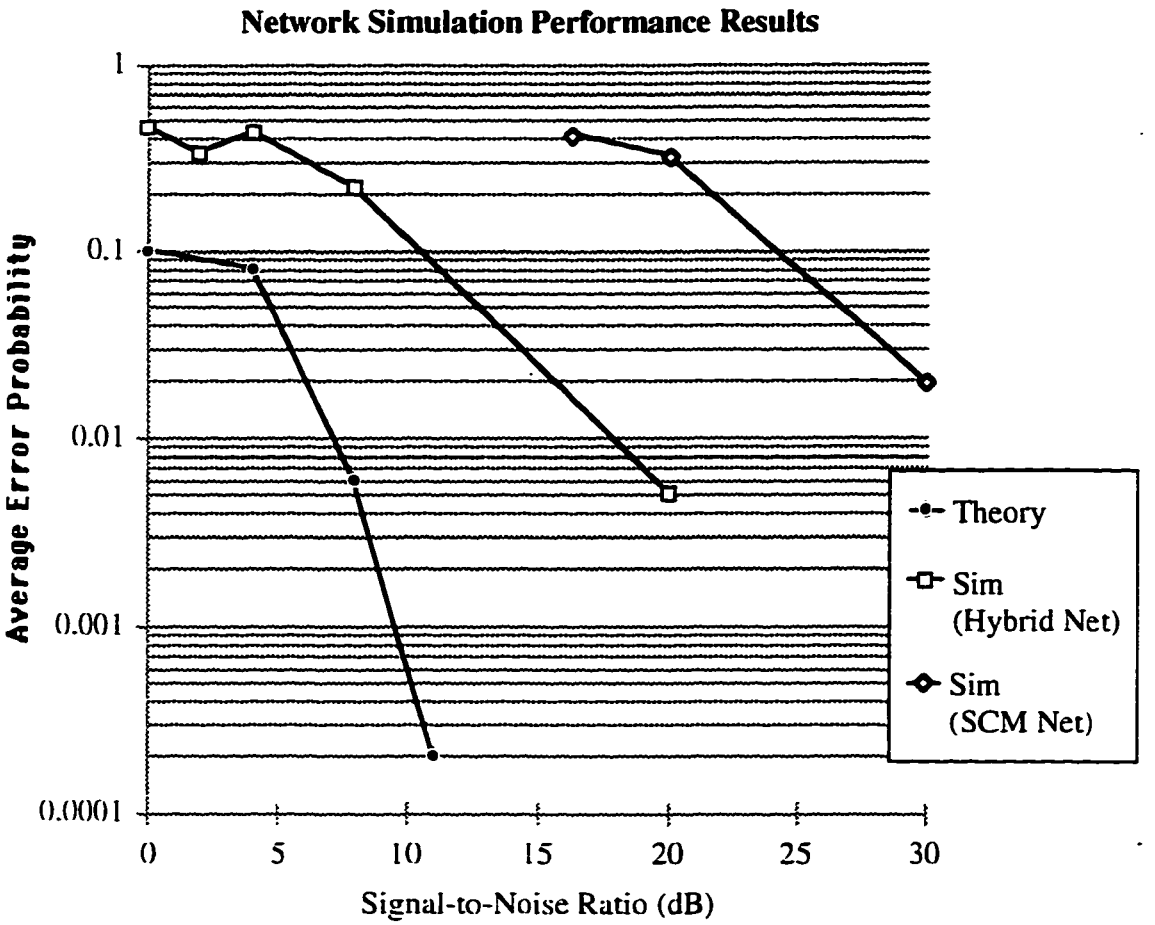


Figure 5.1 Network simulation results

Chapter 6: Conclusions

6.1 Summary

In this thesis, we have investigated the performance benefits that might be obtained by combining subcarrier multiplexing with code division multiple access in optical local area networks. Earlier research by Khaleghi indicated that combining CDMA with SCM might alleviate problems caused by the non-linearity of lasers in SCM systems. However Khaleghi's theoretical analysis and the experiments she conducted did not specifically deal with the laser non-linearity. Our objective was to provide an investigation which demonstrates that a hybrid SCM-CDMA optical network has an improved performance despite the distortion introduced by the optical sources.

Due to the difficulties involved with an analytical approach, we decided to construct a simulation of the local area network using SCM with and without CDMA. Once the simulation program was built, the performance of the two approaches was compared.

The first step in building the simulation program was to design the network. Each user's data was encoded using an assigned Gold sequence and DPSK modulated. The summed inputs of all users were passed over the optical portion of the network. Following the optical receiver, the detected signal were demodulated, despread and compared with the original data of a reference user to calculate the bit error rate of the network.

As a second step in constructing the simulation, a model was developed for each element in the network. The models of three key elements—the laser, the optical receiver, and the SAW filter—were described in detail and their performance verified either through simulation or theoretical analysis.

The laser is the most complex element in the optical LAN: it was modeled by applying a Euler forward integration to the classic laser rate equations. We verified the simulated performance of the laser by comparing it against the results reported by Neusy and others. We found that although our approach to programming the laser simulation module was slightly different than that of Neusy, the laser's transfer function, its distortion level, and the RIN spectrum within the laser's 6 GHz bandwidth agreed reasonably well with Neusy's results.

A SAW filter matched to the reference user's code sequence was used to despread received signals. The SAW matched filter can be modeled as a transversal filter with its coefficients set equal to the code sequence values. We provided an analysis which demonstrated that a transversal filter will correctly recover the original data sent by the reference user.

A third complex element in the simulation is the optical receiver. The optical receiver used for the simulation is composed of a photodetector followed by an amplifier and a filter. The photodetector was assumed to suffer from shot noise and the amplifier from thermal noise. We found through a detailed analysis that the optical receiver can be modeled by a noiseless photodiode followed by a shot noise module, a linear gain block to

which white Gaussian noise is added, and a lowpass filter. The photodetector shot noise was modeled as a Poisson random process. The parameter governing the optical receiver submodules were also determined.

All elements of the hybrid optical network were combined into one simulation program. Special care was taken in designing the data flow through the program to avoid issues such as requiring excessive amounts of computer memory space during the simulations or experiencing unstable behaviour from the laser module. The first issue was dealt with by handling only relatively small quantities of data at one time, in effect looping through several small simulations in the process of running a longer one. The laser instability was avoided by setting the simulation time step sufficiently small.

While the simulation program was functional at the writing of this thesis, only preliminary results have been obtained to date. The simulations that were run involved networks with only one light source and a very small number of users. They indicate that combining CDMA with subcarrier multiplexing does indeed seems to improve the performance of the optical LAN over using SCM alone. The results are not statistically extremely reliable, as only relatively small numbers of bits were simulated, but they give an indication of what may be expected from longer simulations and more complex network configurations. The simulations also seem to point out that the performance of both the hybrid and the SCM optical LANs fall short of the theoretical single-user performance. This difference in performance may be due to limitations in the simulation program, or it may be caused by performance impairments present in realistic networks, but not apparent

in a theoretical analysis. At the writing of this thesis. it appears that the simulation program may still require some improvement. In particular the Butterworth filter modules appear to cause degradation in the network's performance.

6.2 Suggestions for Further Research

Our simulation of the hybrid optical LAN has so far provided only preliminary results. It would be of interest to further refine the simulation program and obtain more results from it. For example the frequency downconversion module used a Butterworth second order lowpass filter. We used a real-time simulation of the filter, however filters are more effectively simulated using FFT's. A simulation redesigned to accommodate an FFT-based simulation of the lowpass and bandpass filters might provide better results. Other areas which deserve more investigation are as follows.

Our network simulations used Gold sequence for spreading of users' data. Other code sequences might also have been used for non-linearity suppression. More specifically multilevel sequences may be worthy of investigation as they are more bandwidth efficient than the binary ones.

Our simulation focused on an optical LAN which used SCM and CDMA. However wavelength division multiplexing is gaining much attention in the telecommunications industry for interoffice networks. The use WDM in local access networks is also being investigated. In a similar vein as Vanucci's original proposal to combine FDMA with CDMA. it may also be worthwhile to examine combining SCM with CDMA in

multiwavelength local access network. The performance improvement of such a network could be verified with a simulation program similar to the one that was described in this thesis.

Appendix A

Included is the complete C code of the simulation program.

```
/* Hybrid SCM-CDMA Optical Network Simulation */
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#define MAX_USERS 2
#define PI 3.141592654
#define ELECTRON 1.6e-19
#define HTIMESC 1.9878e-25
#define EE 2.718281828
#define BOLTZMANN 1.38e-23
/* Global Variables */
int word_size;
long seed;        /* user input */
double sfreq = 4.0e11;
double input_noise_var = 6.624e-11;
int samples_per_bit = 24800;        /* dpsk */
double bit_rate = 1.612903e7;
int samples_per_chip = 800;
double chip_rate = 5.0e8;
int pnseq_config1 = 37;
int pnseq_config2 = 61;
int shift_register_order = 5;
int shift_register_init;    /* ui */
double carrier_freq;                /* demod */ /* ui user0 freq */
double lpf_bandwidth;       /* ui */
double bpf_bandwidth;       /* ui */
double bpf_center_freq;     /* ui same as carrier freq */
double volume_times_charge = 1.44e-35;     /* laser */
double electron_lifetime = 3.72e-9;
double trans_carrier_density = 4.6e24;
double optical_gain = 1.0e-12;
double gain_compression = 2.6e-23;
double photon_lifetime = 2.0e-12;
double coherent_sp_emission = 0.001;
double optical_confinement = 0.646;
double initial_carrier_density = 5.48470e24;
double initial_photon_density = 4.82581e21;
double laser_efficiency = 0.1;
double opt_mod_index;       /* ui */
double laser_dc_bias = 75.0e-3;
double wavelength = 8.35e-7;            /* pin */
double pin_quantum_efficiency = 0.9;
double receiver_gain = -75.0;
double amp_noise_fig = 3.981072;
double amp_feedback_res = 75.0;
double rx_passband_edge = 6.0e9;
double fiber_atten = -3.0;  /* must be negative; in dB/km */    /* fiber */
double fiber_length = 2.0;    /* in km */
double lf_coupling_loss = -5.0;    /* must be negative; in dB */
long transmission_delay1 = 715;
long transmission_delay2 = 30;
long ito_delay = 0;
/* System Subroutines */
void hscm_in(long, int, double *, long *, double *);
void hscm_sp(double *, double , long *, long *, long);
```

```
/* Building Block Subroutines */
long signal_source(void);
long ran2(long);
double ran1(long);
void diff_encoder(long *, long *, int, int);
void gold_seq_src(long *, int, int, int);
void interpolate(long *, long *, int, int);
void xor(long *, int, long *, int, long *);
void logical_to_real(long *, double *, long);
void modulator(double *, double *, double, long, long);
void white_noise(double *, double, long);
void laser2n_complete(double *, double, double, long, double *, long);
void laser2_n(double, double, double, double *, double *, double *, long);
double laser_sn_gen(double, long);
double poisson(double, long);
double gammln(double);
void fiber_n(double *, double *, long);
double pin_diode(double *);
void photodet_n(double *, double *, long, double, long);
void butterworth_rx(double *, double *, long, long);
void butterworth2(double *, double *, double, long, long);
void butter_bandpass2(double *, double *, double, double, long, long);
void decimate(double *, double *, long, int, long);
void saw_filt_tdl2(double *, double *, long, int);
void diff_decoder(double *, double *, int, long);

void main(void)
{
    int num_users, user;        /* ui */
    char sim_id[6];
    unsigned long num_bits;     /* ui */
    unsigned long num_loops, loop, inloop;
    long space;
    double *freq;      /* ui */
    double raninit;
    long *data0;            /* input for user0 for each loop */
    double *data_temp;
    long num_err, *final_err;
    long total_err = 0;
    double ber;

/* initialize global variable word_size */
    word_size = 8 * sizeof(long);

/* obtaining values from user */
    printf("Welcome to a simulation of a hybrid SCM-CDM optical
network!!\n\n");
    printf("First, enter a value for the number of users in the network:\n");
    scanf("%d", &num_users);
    printf("Enter a simulation id\n");
    scanf("%s", &sim_id);
    printf("Enter the total simulation length in data bits\n");
    scanf("%u", &num_bits);

    if(num_bits % (word_size) == 0)
        num_loops = (long) (num_bits / word_size);
    else
        num_loops = (long) (num_bits / word_size) + 1;
    printf("This simulation will go through %ld loops\n", num_loops);

    freq = (double *)calloc(num_users, sizeof(double));
    printf("Enter the carrier frequency assigned to each user");
    freq += num_users;
    for(user = 0; user < num_users; user++)
```

```
    {
        --freq;
        printf("\nFor User%d:  ", user);
        scanf("%le", freq);
        if(user == 0)
        {
            carrier_freq = *freq;
            bpf_center_freq = *freq;
        }
    }
    printf("\n");

    for(user = 0; user < num_users; user++)
        printf("Freq[%d]= %le \n", user, *(freq+num_users-1-user));

    printf("Enter an integer value (>0) to initialize Gold sequence generator
shift registers\n");
    scanf("%d", &shift_register_init);
    printf("Enter a value for the optical modulation index\n");
    scanf("%lf", &opt_mod_index);
    printf("Please enter output filter values.\n");
    printf("The bandpass filter bandwidth: \n");
    scanf("%le", &bpf_bandwidth);
    printf("The lowpass filter bandwidth:\n");
    scanf("%le", &lpf_bandwidth);
    printf("Finally enter a negative seed value for the random generator\n");
    scanf("%ld", &seed);
    printf("Thank you. The values you have entered were:\n");
    printf("num=%d ** sri=%d ** omi=%lf \nbbw=%le ** lbw=%le ** seed=%ld",
    num_users,
    shift_register_init, opt_mod_index, bpf_bandwidth, lpf_bandwidth, seed);
    printf("\n\nLet the games begin...\n\n");

/* Initialize random number generator */
    raninit = ran1(seed);
    printf("Initial value of ran1 is %lf\n", raninit);

/* Reserve space for pointers */
    space = samples_per_bit * word_size;
    data_temp = (double *)calloc(space, sizeof(double));
    data0 = (long *)calloc(num_loops, sizeof(long));
    final_err = (long *)calloc(num_loops, sizeof(long));
    if((data_temp || data0 || final_err) == NULL)
    {
        printf("No Memory. Big Problem!!\n");
        exit(1);
    }

/* Start the simulation run */
    data0 += num_loops;
    final_err += num_loops;
    for(loop = 0; loop < num_loops; loop++)
    {
        for(inloop = 0; inloop < space; inloop++)
            *(data_temp+space-1-inloop) = 0.0;
        --data0;
        --final_err;
        hscm_in(loop, num_users, freq, data0, data_temp);
        printf("hscm_in: loop %ld complete\n", loop);
        hscm_sp(data_temp, carrier_freq, data0, &num_err, loop);
        printf("hscm_sp: loop %ld complete\n", loop);
        total_err += num_err;
        *final_err = num_err;
    }
```

```
    printf("The simulation has completed successfully!!!\n");

    printf("The total error count was %ld\n", total_err);
    ber = (double) total_err / num_bits;
    printf("The bit error rate was %g\n", ber);

/* printing out error counts for each loop */
/*    printf("Loop\t Number of Errors\n");
    for(loop = 0; loop < num_loops; loop++)
        printf("%d\t %ld\n", loop, *(final_err+num_loops-1-loop));  */

    printf("**** That's all folks!! ***\n");
}

void hscm_in(long simloop, int user_num, double *ufreq, long *input0,
                        double *out)
{
    long *input, *diff_out;
    int u, v, w;
    int init_bits;
    long seq_length, seq_longs, spd_length, mod_length;
    long *interpol, *gold_src;
    long *spread, *spd_intpl;
    double *spd_real, *mod_sig;
    int reginit;
    int t;

/* Calculating space requirements */
    init_bits = word_size;
    seq_length = (samples_per_bit / samples_per_chip);
    seq_longs = (seq_length + 1) / word_size;
    spd_length = seq_length * word_size;
    mod_length = samples_per_bit * word_size;

/* Reserve space for input */
    input = (long *)calloc(user_num, sizeof(long));
    if(input == NULL)
    {
       printf("%ld SOURCE: No Memory. Big Problem!!\n", simloop);
       fflush(NULL);
       exit(1);
    }

/* running signal_source */
    input += user_num;
    for(u = 0; u < user_num; u++)
    {
        --input;
        *input = signal_source();
        if(u == 0)
            *input0 = *input;
    }
    printf("%ld SOURCE: So far, so good...\n", simloop);
    fflush(NULL);

/* reserve space for diff_out */
    diff_out = (long *)calloc(user_num, sizeof(long));
    if(diff_out == NULL)
    {
        printf("%ld DIFF: No Memory. Big Problem!!\n", simloop);
        fflush(NULL);
        exit(1);
    }
```

```
/* pass data through diff_encoder */
    diff_encoder(input, diff_out, simloop, user_num);
    printf("%ld DIFF: So far, so good...\n", simloop);
    fflush(NULL);

/* free up old space */
    free(input);

/* start on large loop through the remaining input system one user at a time */
    reginit = shift_register_init;
    diff_out += user_num;
    ufreq += user_num;
    for(v = 0; v < user_num; v++)
    {
/*  interpolate differentially encoded data */
        interpol = (long *)calloc(seq_length, sizeof(long));
        if(interpol == NULL)
        {
            printf("%ld.%d INTPL(1): No Memory. Big Problem!!\n", simloop, v);
            fflush(NULL);
            exit(1);
        }
        --diff_out;
        interpolate(diff_out, interpol, init_bits, seq_length);
        printf("%ld.%d INTPL(1): So far, so good...\n", simloop, v);
        fflush(NULL);

/* generate gold sequence - one for each user */
        gold_src = (long *)calloc(seq_longs, sizeof(long));
        if(gold_src == NULL)
        {
            printf("%ld.%d GOLD: No Memory. Big Problem!!\n", simloop, v);
            fflush(NULL);
            exit(1);
        }
        gold_seq_src(gold_src, seq_length, seq_longs, reginit);
        reginit += 2;
        printf("%ld.%d GOLD: So far, so good...\n", simloop, v);
        fflush(NULL);

/* combining interpolated diff data & gold sequence */
        spread = (long *)calloc(seq_length, sizeof(long));
        if(spread == NULL)
        {
            printf("%ld.%d XOR: No Memory. Big Problem!!\n", simloop, v);
            fflush(NULL);
            exit(1);
        }
        xor(interpol, spd_length, gold_src, seq_length, spread);
        printf("%ld.%d XOR: So far, so good...\n", simloop, v);
        fflush(NULL);

/* free up old space */
      free(interpol);
      free(gold_src);

/* interpolate spread data */
      spd_intpl = (long *)calloc(samples_per_bit, sizeof(long));
      if(spd_intpl == NULL)
      {
          printf("%ld.%d INTPL(2): No Memory. Big Problem!!\n", simloop, v);
          fflush(NULL);
          exit(1);
```

```
        }
          interpolate(spread, spd_intpl, spd_length, samples_per_chip);
          printf("%ld.%d INTPL(2): So far, so good...\n", simloop, v);
          fflush(NULL);

/* free old space */
        free(spread);

/* turn logical into real */
          spd_real = (double *)calloc(mod_length, sizeof(double));
          if(spd_real == NULL)
          {
              printf("%ld.%d LOGREAL: No Memory. Big Problem!!\n", simloop, v);
              fflush(NULL);
              exit(1);
          }
          logical_to_real(spd_intpl, spd_real, mod_length);
          printf("%ld.%d LOGREAL: So far, so good...\n", simloop, v);
          fflush(NULL);

/* free up old space */
          free(spd_intpl);

/* modulate data onto carrier */
          --ufreq;
          mod_sig = (double *)calloc(mod_length, sizeof(double));
          if(mod_sig == NULL)
          {
              printf("%ld.%d MOD: No Memory. Big Problem!!\n", simloop, v);
              fflush(NULL);
              free(spd_real);
              exit(1);
          }
          modulator(spd_real, mod_sig, *ufreq, simloop, mod_length);
          printf("%ld.%d MOD: So far, so good...\n", simloop, v);
          fflush(NULL);

/* free old space */
          free(spd_real);

/* add modulated signal to output sum */
          out += mod_length;
          mod_sig += mod_length;
          for(w = 0; w < mod_length; w++)
          {
              --out;
              --mod_sig;
              *out += *mod_sig;
          }
/* free up mod_sig space */
          free(mod_sig);
      }
/* End of v-loop */

/* free up old space */
      free(diff_out);

      printf("End of Simulation Part 1\n");
}

void hscm_sp(double *data_temp, double carrier_freq, long *in0,
                        long *out_err, long loopsim)
{
      int ww, x, y, z;
```

```
    long seq_length2, seq_longs2, spd_length2, mod_length2, final_length;
    double *awgn, awgn_var;
    double *noisy_sig, *laser_out, *fiber_out, *det_sig, *blt_sig;
    double *demod_sig, *flt_sig, *dec_flt, *saw_sig;
    double *dec_saw, *decoded_sig;
    double *indbl0;
    unsigned long decoded_data;
    int mark_bit, loop1;
    double amp_noise_var, lpf_flt_bw;
    int t;

/* Calculating space requirements */
    seq_length2 = (samples_per_bit / samples_per_chip);
/*    seq_longs2 = (seq_length2 + 1) / word_size;  */
    spd_length2 = seq_length2 * word_size;
    mod_length2 = samples_per_bit * word_size;
    final_length = word_size;

 * adding white noise to input signal */
    noisy_sig = (double *)calloc(mod_length2, sizeof(double));
    awgn = (double *)calloc(mod_length2, sizeof(double));
    if((noisy_sig == NULL) || (awgn == NULL))
    {
        printf("%ld NOISE: No Memory. Big Problem!!\n", loopsim);
        fflush(NULL);
        exit(1);
    }
    awgn_var = input_noise_var;
    white_noise(awgn, awgn_var, mod_length2);
    printf("%ld NOISE: So far, so good...\n", loopsim);
    fflush(NULL);

    awgn += mod_length2;
    data_temp += mod_length2;
    noisy_sig += mod_length2;
    for(ww = 0; ww < mod_length2; ww++)
    {
        --awgn;
        --data_temp;
        --noisy_sig;
        *noisy_sig = *data_temp + *awgn;
    }
    printf("%ld ADD NOISE: So far, so good...\n", loopsim);
    fflush(NULL);

/* biasing input signal for laser */
   noisy_sig += mod_length2;
   for(x = 0; x < mod_length2; x++)
   {
       --noisy_sig;
       *noisy_sig *= (laser_dc_bias * opt_mod_index);
       *noisy_sig += laser_dc_bias;
   }

/* free up old space */
    free(awgn);

/* transmitting signal over optical part of network */
    laser_out = (double *)calloc(mod_length2, sizeof(double));
    if(laser_out == NULL)
    {
        printf("%ld LASER: No Memory. Big Problem!!\n", loopsim);
        fflush(NULL);
        exit(1);
```

```
    }

/* passing through laser */
    laser2n_complete(noisy_sig, initial_carrier_density,
                initial_photon_density, mod_length2, laser_out, loopsim);
    printf("%ld LASER: So far, so good...\n", loopsim);
    fflush(NULL);

/* free up old space */
    free(noisy_sig);

/* allocating memory for fiber output */
    fiber_out = (double *)calloc(mod_length2, sizeof(double));
    if(fiber_out == NULL)
    {
        printf("%ld FIBER: No Memory. Big Problem!!\n", loopsim);
        fflush(NULL);
        exit(1);
    }

/* passing over fiber */
    fiber_n(laser_out, fiber_out, mod_length2);
    printf("%ld FIBER: So far, so good...\n", loopsim);
    fflush(NULL);

/* free up old space & reserve new */
    free(laser_out);
    det_sig = (double *)calloc(mod_length2, sizeof(double));
    if(det_sig == NULL)
    {
        printf("%ld DET: No Memory. Big Problem!!\n", loopsim);
        fflush(NULL);
        exit(1);
    }

/* calculate noise variance from amplifier */
    amp_noise_var = 4.0 * BOLTZMANN * (sfreq / 2.0) *
                (300.0 + 290.0 * (amp_noise_fig - 1.0)) / amp_feedback_res;
/*    printf("Note that amp noise is %e \n", amp_noise_var);        */

/* getting detected at photodetector */
    photodet_n(fiber_out, det_sig, mod_length2, amp_noise_var, loopsim);
    printf("%ld DET: So far, so good...\n", loopsim);
    fflush(NULL);

/* free up old space */
    free(fiber_out);

/* preliminary filtering of signal */
    blt_sig = (double *)calloc(mod_length2, sizeof(double));
    if(blt_sig == NULL)
    {
        printf("%ld BPF: No Memory. Big Problem!!\n", loopsim);
        fflush(NULL);
        exit(1);
    }
    butter_bandpass2(det_sig, blt_sig, bpf_center_freq, bpf_bandwidth,
                        loopsim, mod_length2);
    printf("%ld BPF: So far, so good...\n", loopsim);
    fflush(NULL);

/* free up old space */
    free(det_sig);
```

```
/* beginning demodulation */
    demod_sig = (double *)calloc(mod_length2, sizeof(double));
    if(demod_sig == NULL)
    {
        printf("%ld DEMOD: No Memory. Big Problem!!\n", loopsim);
        fflush(NULL);
        exit(1);
    }
    modulator(blt_sig, demod_sig, carrier_freq, loopsim, mod_length2);
    printf("%ld DEMOD: So far, so good...\n",  loopsim);
    fflush(NULL);

/* free up old space */
    free(blt_sig);

/* filter signal */
    flt_sig = (double *)calloc(mod_length2, sizeof(double));
    if(flt_sig == NULL)
    {
        printf("%ld LPF: No Memory. Big Problem!!\n", loopsim);
        fflush(NULL);
        exit(1);
        }
    butterworth2(demod_sig, flt_sig, lpf_bandwidth, loopsim, mod_length2);
    printf("%ld LPF: So far, so good...\n", loopsim);
    fflush(NULL);

/* free up old space */
    free(demod_sig);

/* decimate filtered signal (1st time) */
    dec_flt = (double *)calloc(spd_length2, sizeof(double));
    if(dec_flt == NULL)
    {
        printf("%ld DEC(1): No Memory. Big Problem!!\n", loopsim);
        fflush(NULL);
        exit(1);
    }
    decimate(flt_sig, dec_flt, mod_length2, samples_per_chip,
                        transmission_delay1);
    printf("%ld DEC(1): So far, so good...\n", loopsim);
    fflush(NULL);

/* free up old space */
    free(flt_sig);

/* despread signal: passing through saw_filter */
    saw_sig = (double *)calloc(spd_length2, sizeof(double));
    if(saw_sig == NULL)
    {
        printf("%ld SAW: No Memory. Big Problem!!\n", loopsim);
        fflush(NULL);
        exit(1);
    }
/* run saw filter through a loop */
    dec_flt += spd_length2;
    saw_sig += spd_length2;
    for(z = 0; z < word_size; z++)
    {
        dec_flt -= seq_length2;
        saw_sig -= seq_length2;
        if((loopsim ==0) && (z == 0))
            loop1 = 0;
        else
```

```
            loop1 = 1;
/*      printf("Loop1=%d\n", loop1);        */
        saw_filt_td12(dec_flt, saw_sig, seq_length2, loop1);
    }
    printf("%ld SAW: So far, so good...\n", loopsim);
    fflush(NULL);

/* free up old space */
    free(dec_flt);

/* decimate despread signal (2nd time) */
    dec_saw = (double *)calloc(final_length, sizeof(double));
    if(dec_saw == NULL)
    {
        printf("%ld DEC(2): No Memory. Big Problem!!\n", loopsim);
        fflush(NULL);
        exit(1);
    }
    decimate(saw_sig, dec_saw, spd_length2, seq_length2, transmission_delay2);
    printf("%ld DEC(2): So far, so good...\n", loopsim);
    fflush(NULL);

/* free up old space */
    free(saw_sig);

/* decode signal */
    decoded_sig = (double *)calloc(final_length, sizeof(double));
    if(decoded_sig == NULL)
    {
        printf("%ld DECODE: No Memory. Big Problem!!\n", loopsim);
        fflush(NULL);
        exit(1);
    }
    diff_decoder(dec_saw, decoded_sig, loopsim, final_length);
    printf("%ld DECODE: So far, so good...\n", loopsim);
    fflush(NULL);

/* free up old space */
    free(dec_saw);

/* convert in0 to an array of doubles */
    indbl0 = (double *)calloc(final_length, sizeof(double));
    if(indbl0 == NULL)
    {
        printf("%ld IN0: No Memory. big Problem!!\n", loopsim);
        fflush(NULL);
        exit(1);
    }
    logical_to_real(in0, indbl0, final_length);
    printf("%ld IN0: So far, so good...\n", loopsim);
    fflush(NULL);

/* compare values of indbl0 and decoded_sig */
    *out_err = 0;
    indbl0 += final_length;
    decoded_sig += (final_length - ito_delay);
    for(y = 0; y < (final_length - ito_delay); y++)
    {
        --indbl0;
        --decoded_sig;
        if(((*indbl0) * (*decoded_sig)) < 0.0)
            ++(*out_err);
    }
    printf("\n");
```

```
/* convert signal from array of doubles back to bits in a long */
    decoded_sig += final_length;
    decoded_data = 0L;
    for(y = 0; y < final_length; y++)
    {
        --decoded_sig;
        decoded_data <<= 1;
        if(*decoded_sig < 0.0)
            mark_bit = 1;
        else
            mark_bit = 0;
        decoded_data = (decoded_data | mark_bit);
    }

/* free up old space */
    free(indbl0);
    free(decoded_sig);

    printf("in = %lx, out = %lx \n", *in0, decoded_data);
    printf("End of Simulation Part 2\n");
    fflush(NULL);
}


long signal_source(void)
{
    long count;
    long ranbit, ransig = 0;
    long ranno;

/* Run random number gen through a loop to generate the random bits */
    for(count = 0; count < word_size; count++)
    {
        ransig <<= 1;
        ranno = ran2(count);
        ranbit = ranno % 2;
        ransig = ransig | ranbit;
    }
    return(ransig);
}

long ran2(long idum2)
{
    double randbl;
    long out1;

    randbl = ran1(idum2);

    out1 = (long)(randbl * 1000000);
    return(out1);
}

double ran1(long idum)
{
    static long int Ran1Ix1, Ran1Ix2, Ran1Ix3;
    static double Ran1R[97];

    long int m1 = 259200, ia1 = 7141, ic1 = 54773;
    double rm1 = 3.8580247e-6;
    long int m2 = 134456, ia2 = 8121, ic2 = 28411;
    double rm2 = 7.4373773e-6;
    long int m3 = 243000, ia3 = 4561, ic3 = 51349;
    int j;
```

```
    double out;

    if(idum < 0)
    {
        Ran1Ix1 = (ic1-idum) % m1;
        Ran1Ix1 = (ia1*Ran1Ix1 + ic1) % m1;
        Ran1Ix2 = Ran1Ix1 % m2;
        Ran1Ix1 = (ia1*Ran1Ix1 + ic1) % m1;
        Ran1Ix3 = Ran1Ix1 % m3;
        for(j = 0; j < 97; j++)
        {
            Ran1Ix1 =  (ia1*Ran1Ix1 + ic1) % m1;
            Ran1Ix2 =  (ia2*Ran1Ix2 + ic2) % m2;
            Ran1R[j] = (Ran1Ix1 + Ran1Ix2 * rm2) * rm1;
        }
        idum = 1;
    }
    Ran1Ix1 =  (ia1 * Ran1Ix1 + ic1) % m1;
    Ran1Ix2 =  (ia2 * Ran1Ix2 + ic2) % m2;
    Ran1Ix3 =  (ia3 * Ran1Ix3 + ic3) % m3;
    j = 1 + (97 * Ran1Ix3) / m3;
    if((j > 97) || (j < 1))
        printf("ran1: Big Problem.....\n");
    out = Ran1R[j-1];
    Ran1R[j-1] = (Ran1Ix1 + Ran1Ix2 * rm2) * rm1;
    return(out);
}

void diff_encoder(long *data, long *diff, int init, int users)
{
    static long *prev_diff;
    unsigned long temp, mask;
    int lsize, count, num;

/* initialize prev_diff */
    if(init == 0)
    {
        prev_diff = (long *)calloc(users, sizeof(long));
        if(prev_diff == NULL)
        {
            printf("diff_encoder: No Memory. Big Problem!!\n");
            exit(1);
        }
        prev_diff += users;
        for(num = 0; num < users; num++)
        {
            --prev_diff;
            *prev_diff = 0L;
        }
    }

    prev_diff += users;
    data += users;
    diff += users;
    lsize = word_size;
    for(num = 0; num < users; num++)
    {
        --prev_diff;
        -- data;
        --diff;
        mask = 1L;
        mask <<= (lsize - 1);
        for(count = 0; count < lsize; count++)
        {
```

```
            temp = *data & mask;
            *diff = (temp ^ *prev_diff) | *diff;
            *prev_diff = (mask & *diff) >> 1;
            mask >>= 1;
        }
/* prev_diff value for next run */
        *prev_diff = (*diff & 01) << (lsize - 1);
    }
}

void gold_seq_src(long *output, int seqlen, int outlen, int shift_init)
{
    int iloop, jloop;
    int n;
    unsigned int reg1, reg2;
    unsigned int feedback1 = 0, feedback2 = 0;
    int onebit = 1, topbit = 1;
    int out1, out2;
    int rtemp1, rtemp2, gtemp1, gtemp2;

    n = outlen - 1;
    *output = 0;
    reg1 = shift_init;
    reg2 = shift_init;

    for(iloop = 0; iloop < seqlen; iloop++)
    {
        out1 = (reg1 & topbit) ? 1 : 0;
        out2 = (reg2 & topbit) ? 1 : 0;
        if(((iloop % word_size) == 0) && (iloop != 0))
            --n;
        *(output + n) = (*(output + n)) | ((out1 + out2) % 2);
        if(iloop != ((seqlen+1)/outlen-1))
            *(output + n) <<= 1;

        for(jloop = 0; jloop < shift_register_order; jloop++)
        {
            rtemp1 = (reg1 & onebit) ? 1 : 0;
            rtemp2 = (reg2 & onebit) ? 1 : 0;
            gtemp1 = (pnseq_config1 & onebit) ? 1 : 0;
            gtemp2 = (pnseq_config2 & onebit) ? 1 : 0;
            feedback1 += (rtemp1 * gtemp1);
            feedback2 += (rtemp2 * gtemp2);
            onebit <<= 1;
        }
        feedback1 %= 2;
        feedback2 %= 2;

        reg1 >>= 1;
        feedback1 <<= (shift_register_order - 1);
        reg1 = reg1 | feedback1;
        reg2 >>= 1;
        feedback2 <<= (shift_register_order - 1);
        reg2 = reg2 | feedback2;

        feedback1 = 0;
        feedback2 = 0;
        onebit = 1;
    }
}

void interpolate(long *input, long *output, int inlen, int num)
{
    int i, j;
```

```
    int blocks;
    unsigned long one_bit = 1L;
    unsigned int input_bit;
    int bit_num = 0;

    blocks = inlen / word_size;
    input += blocks;
    output += (blocks * num - 1);
    for(i = 0; i < inlen; i++)
    {
        if((i % word_size) == 0)
        {
            --input;

            one_bit = 1L;
            one_bit <<= (word_size - 1);
        }
        if((*input & one_bit) == 0)
            input_bit = 0;
        else
            input_bit = 1;
        one_bit >>= 1;
        for(j = 0; j < num; j++)
        {
            *output <<= 1;
            *output = input_bit | (*output);
            ++bit_num;
            if(bit_num == word_size)
            {
          --output;
          bit_num = 0;
            }
        }
    }
}

void xor(long *data, int datalen, long *sequence, int seqlen, long *output)
{
    int i;
    int datanum = 0, seqnum = 0;
    int units;
    unsigned long databit = 1L, seqbit = 1L;

    units = datalen / word_size;
    data += (units - 1);
    output += (units - 1);
    databit <<= (word_size - 1);
    seqbit <<= (word_size - 1);

    for(i = 0; i < datalen; i++)
    {
        if(datanum == word_size)
        {
            data--;
            output--;
            databit = 1L;
            databit <<= (word_size - 1);
            datanum = 0;
        }
        if(seqnum == seqlen)
        {
            seqbit = 1L;
            seqbit <<= (word_size - 1);
            seqnum = 0;
```

```
        }

        if((((*data & databit) == 0) && ((*sequence & seqbit) == 0)) ||
            (((*data & databit) != 0) && ((*sequence & seqbit) != 0)))
            *output = *output | 0;
        else
            *output = *output | 1;
        if(datanum != (word_size - 1))
            *output <<= 1;
        databit >>= 1;
        datanum++;
        seqbit >>= 1;
        seqnum ++;
    }
}

void logical_to_real(long *logic, double *realnum, long samples)
{
    unsigned long onebit = 1L;
    long blocks;
    long i;

    blocks = samples / word_size;
    logic += blocks;
    realnum += samples;
    for(i = 0; i < samples; i++)
    {
        if((i % word_size) == 0)
        {
            --logic;
            onebit = 1L;
            onebit <<= (word_size - 1);
        }
        --realnum;
        if((*logic & onebit) == 0)
            *realnum = 1.0;
        else
            *realnum = -1.0;
        onebit >>= 1;
    }
}

void modulator(double *data, double *mod, double ufreq, long loopnum,
                        long samples)
{
    double carrier, interval;
    double echant;
    long i, j;

    interval = ufreq / sfreq;
    echant = loopnum * samples * interval;

    data += samples;
    mod += samples;
    for(i = 0; i < samples; i++)
    {
        --data;
        --mod;
        *mod = *data * sin(2 * PI * echant);
        echant += interval;
    }
}

void white_noise(double *outnoise, double var, long idumg)
```

```
{
    static int GasdevIset = 0;
    static double GasdevGset;

    double fac, r = 0.0, v1, v2;
    double gasdev;
    long i;

    outnoise += idumg;
    for(i = 0; i < idumg; i++)
    {
        --outnoise;

        if(GasdevIset == 0)
        {
            while((r >= 1.0) || (r <= 0.0))
            {
          v1 = 2.0 * ran1(idumg) - 1.0;
          v2 = 2.0 * ran1(idumg) - 1.0;
           r = v1 * v1 + v2 * v2;
            }
            fac = sqrt(-2.0 * var * log(r) / r);
            GasdevGset = v1 * fac;
            *outnoise = v2 * fac;
            GasdevIset = 1;
        }
        else
        {
            GasdevIset = 0;
            *outnoise = GasdevGset;
        }
        r = 0.0;
    }
}

void laser2n_complete(double *ic, double icd, double ipd, long num,
                      double  *lpo, long first)
{
    static double prev_cd;
    static double prev_pd;
    double carrier_density_change;
    double photon_density_change;
    double light_output;
    long i;


    if(first == 0)
    {
        prev_cd = icd;
        prev_pd = ipd;
    }

/*  printf("   i     laser_output\n");  */
    ic += num;
    lpo += num;
    for(i = 0; i < num; i++)
    {
        --ic;
        --lpo;
        laser2_n(*ic, prev_cd, prev_pd, &carrier_density_change,
                &photon_density_change, &light_output, i);

/* convert light in photons to light power - formerly opt_convert */
        *lpo = (light_output * laser_efficiency * volume_times_charge *
```

```
                        HTIMESC) / (ELECTRON * wavelength);

            prev_cd += carrier_density_change;
            prev_pd += photon_density_change;
        }
}

void laser2_n(double ic, double cd, double pd, double *cdc, double *pdc,
              double *lo, long nseed)
{
    double cir, spe, ste, sta;
    double cir_n, spe_n,ste_n, sta_n;
    double cohspe_n, conf_ste_n, conf_sta_n;
    double radp, radp_n;

/* calculate carrier injection rate - formerly carrier_rate() */
    cir = ic / volume_times_charge;
    cir_n = laser_sn_gen(cir, nseed);

/* calculate spontaneous emission - formerly sp_emission() */
    spe = cd / electron_lifetime;
    spe_n = laser_sn_gen(spe, nseed);

/* calculate stimulated emission - formerly stim_emit() */
    ste = optical_gain * cd * (1.0 - gain_compression * pd) * pd;
    ste_n = laser_sn_gen(ste, nseed);

/* calculate stimulated absorption - formerly stim_absorb() */
    sta = optical_gain * trans_carrier_density
                * (1.0 - gain_compression * pd) * pd;
    sta_n = laser_sn_gen(sta, nseed);

/* calculate radiated photons - formerly photon_output() */
    radp = pd / photon_lifetime;
    radp_n = laser_sn_gen(radp, nseed);

    cohspe_n = laser_sn_gen(coherent_sp_emission * spe * optical_confinement,
nseed);
    conf_ste_n = laser_sn_gen(optical_confinement * ste, nseed);
    conf_sta_n = laser_sn_gen(optical_confinement * sta, nseed);

    *cdc = (cir_n - spe_n - (ste_n - sta_n)) / sfreq;
    *pdc = ((conf_ste_n - conf_sta_n) - radp_n + cohspe_n) / sfreq;
    *lo = radp_n;
}

double laser_sn_gen(double mean, long rseed)
{
    extern double sfreq, volume_times_charge;
    double tmp1, tmp2, shot_noise;

    tmp1 = (volume_times_charge / ELECTRON) / sfreq;
    tmp2 = poisson((mean*tmp1), rseed);
    shot_noise = tmp2 / tmp1;

    return(shot_noise);
}

double poisson(double xm, long idump)
{
    static double poidevoldm = -1.0, poidevsq, poidevalxm, poidevg;

    double em, t, y, emi;
```

```
    if(xm < 12.0)
    {
        if(xm != poidevoldm)    /* use direct method */
        {
            poidevoldm = xm;
            poidevg = exp(-xm); /* if xm is new, compute exponential */
        }
        em = -1.0;
        t = 1.0;
        while(t > poidevg)
        {
            em += 1.0;
            t *= ran1(idump);

        }
    }
    else
    {
        if(xm != poidevoldm)
        {
            poidevoldm = xm;
            poidevsq = sqrt(2.0 * xm);
            poidevalxm = log(xm);
            poidevg = xm * poidevalxm - gammln(xm + 1.0);
        }
        t = 0.0;
        em = -1.0;
        while(ran1(idump) > t)
        {
            while(em < 0.0)
            {
               y = PI * ran1(idump);
               y = sin(y) / cos(y);
               em = poidevsq * y + xm;
            }
            emi = floor(em);
            t = 0.9 * (1.0 + y * y) * exp(emi * poidevalxm - gammln(emi + 1.0) -
poidevg);
            em = -1.0;
        }
    }
    return(emi);
}

double gammln(double xx)
{
    double stp = 2.50662827465;
    double x, tmp, ser;
    double gammln_out;

    x = xx - 1.0;
    tmp = x + 5.5;
    tmp = (x + 0.5) * log(tmp) - tmp;
    ser = 1.0 + 76.18009173 / (x + 1.0) - 86.50532033 / (x + 2.0)
    + 24.01409822 / (x + 3.0) - 1.231739516 / (x + 4.0)
    + 0.120858003e-2 / (x + 5.0) - 0.536382e-5 / (x + 6.0);
    gammln_out = tmp + log(stp * ser);

    return(gammln_out);
}

void fiber_n(double *opt_in, double *opt_out, long nums)
{
    long i;
```

```
    double elog, tmp1, tmple, tmp2;

    elog = log10(EE);
    tmp1 = (lf_coupling_loss + fiber_length * fiber_atten) /10;
    tmple = tmp1 / elog;
    tmp2 = exp(tmple);

    opt_in += nums;
    opt_out += nums;
    for(i = 0; i < nums; i++)
    {
        --opt_in;
        --opt_out;
        *opt_out = *opt_in * tmp2;
    }
}

void photodet_n(double *rx_opt_sig, double *flt_detsig, long num,
                double noise_var, long start)
{
    double pin_out, pin_shot, pin_noise, *rx_noise;
    double *det_sig;
    long i;

/* reserving space for internal signals */
    det_sig = (double *)calloc(num, sizeof(double));
    rx_noise = (double *)calloc(num, sizeof(double));
    if((det_sig == NULL)|| (rx_noise == NULL))
    {
        printf("photodet_n: No memory. Big Problem!!\n");
        exit(1);
    }

/* obtaining rx_noise values */
    white_noise(rx_noise, noise_var, num);

/* running through receiver - filter */
    rx_opt_sig += num;
    det_sig += num;
    rx_noise += num;

    for(i = 0; i < num; i++)
    {
        --rx_opt_sig;
        --det_sig;
        --rx_noise;
        pin_out = pin_diode(rx_opt_sig) / (sfreq * ELECTRON);
        pin_shot = poisson(pin_out, i) * sfreq  * ELECTRON;
        *det_sig = receiver_gain * (pin_shot + *rx_noise);
    }

    butterworth_rx(det_sig, flt_detsig, start, num);

    free(rx_noise);
    free(det_sig);
}

double pin_diode(double *rx_sig)
{
    double det_sig;

    det_sig = (*rx_sig) * (pin_quantum_efficiency * wavelength * ELECTRON) /
HTIMESC;
```

```
    return(det_sig);
}

void butterworth_rx(double *in_data, double *out_flt, long first,
                    long interval)
{
    double a_coeff[3] = {0.0, -1.8669, 0.8752};
    double b_coeff[3] = {0.0021, 0.0042, 0.0021};
    static double in_array[3], out_array[3];
    double tmpout;
    int i, j, k, ii;
    long h;

/* First time butterworth_rx is called, initialize input/output arrays */
    if(first == 0)
    {
        for(i = 0; i < 3; i++)
        {
            in_array[i] = 0.00;
            out_array[i] = 0.00;
        }
    }

    in_data += interval;
    out_flt += interval;
    for(h = 0; h < interval; h++)
    {
        --in_data;
        --out_flt;

/* Calculating new filter output */
        in_array[0] = *in_data;
        out_array[0] = b_coeff[0] * in_array[0];
        for(j = 1; j <= 2; j++)
            out_array[0] += b_coeff[j] * in_array[j] - a_coeff[j] * out_array[j];

        *out_flt = out_array[0];

/* Calculating new filter state values */
        for(k = 2; k >=1; k--)
        {
        in_array[k] = in_array[k-1];
        out_array[k] = out_array[k-1];
        }
    }
}

void butterworth2(double *in_data, double *out_flt, double cutoff_freq,
                  long first, long interval)
{
    double a2_coeff[3] = {1.0000, -1.9833, 0.9835};
    double b2_coeff[3] = {0.3441e-4, 0.6882e-4, 0.3441e-4};
    double a5_coeff[3] = {1.0000, -1.9889, 0.9890};
    double b5_coeff[3] = {0.1534e-4, 0.3067e-4, 0.1534e-4};
    double a10_coeff[3] = {1.0000, -1.9944, 0.9945};
    double b10_coeff[3] = {0.3845e-5, 0.7689e-5, 0.3845e-5};
    double a_coeff[3], b_coeff[3];
    static double in_array[3], out_array[3];
    double tmpout;
    int i, j, k, ii;
    long h;

    for(ii = 0; ii < 3; ii++)
    {
```

```
        if(cutoff_freq == 7.5e8)
        {
             a_coeff[ii] = a2_coeff[ii];
             b_coeff[ii] = b2_coeff[ii];
        }
        else if(cutoff_freq == 5.0e8)
        {
            a_coeff[ii] = a5_coeff[ii];
            b_coeff[ii] = b5_coeff[ii];
        }
        else if(cutoff_freq == 2.5e8)
        {
            a_coeff[ii] = a10_coeff[ii];
            b_coeff[ii] = b10_coeff[ii];
        }
        else
        {
            printf("Error! Invalid cutoff frequency.\n");
            fflush(NULL);
            exit(1);
        }
    }

/* First time butterworth2 is called, initialize input/output arrays */
    if(first == 0)
    {
        for(i = 0; i < 3; i++)
        {
            in_array[i] = 0.00;
            out_array[i] = 0.00;
        }
    }

    in_data += interval;
    out_flt += interval;
    for(h = 0; h < interval; h++)
    {
        --in_data;
        --out_flt;

/* Calculating new filter output */
        in_array[0] = *in_data;
        out_array[0] = b_coeff[0] * in_array[0];
        for(j = 1; j <= 2; j++)
            out_array[0] += b_coeff[j] * in_array[j] - a_coeff[j] * out_array[j];
        *out_flt = out_array[0];

/* Calculating new filter state values */
        for(k = 2; k >=1; k--)
        {
        in_array[k] = in_array[k-1];
        out_array[k] = out_array[k-1];
        }
    }
}

void butter_bandpass2(double *input_data, double *output_flt,
         double center_freq, double bpf_bw, long premier, long duration)
{
    double a2_coeff[3] = {1.0000, -1.9747, 0.9767};
    double b2_coeff[3] = {0.0116, 0.0000, -0.0116};
    double a5_coeff[3] = {1.0000, -1.9823, 0.9844};
    double b5_coeff[3] = {0.0078, 0.0000, -0.0078};
    double a10_coeff[3] = {1.0000, -1.9900, 0.9922};
```

```
    double b10_coeff[3] = {0.0039, 0.0000, -0.0039};
    double a_coeff[3], b_coeff[3];
    static double in_array[3], out_array[3];
    double tmpout;
    int i, j, k, ii;
    long h;

    for(ii = 0; ii < 3; ii++)
    {
        if((center_freq == 3.0e9) && (bpf_bw == 1.5e9))
        {
            a_coeff[ii] = a2_coeff[ii];
            b_coeff[ii] = b2_coeff[ii];
        }
        else if((center_freq == 3.0e9) && (bpf_bw == 1.0e9))
        {
            a_coeff[ii] = a5_coeff[ii];
            b_coeff[ii] = b5_coeff[ii];
        }
        else if((center_freq == 3.0e9) && (bpf_bw == 0.5e9))
        {
            a_coeff[ii] = a10_coeff[ii];
            b_coeff[ii] = b10_coeff[ii];
        }
        else
        {
            printf("Error! Invalid frequency parameters.\n");
            fflush(NULL);
            exit(1);
        }
    }

/* First time butter_bandpass is called, initialize input/output arrays */
    if(premier == 0)
    {
        for(i = 0; i < 3; i++)
        {
            in_array[i] = 0.00;
            out_array[i] = 0.00;
        }
    }

    input_data += duration;
    output_flt += duration;
    for(h = 0; h < duration; h++)
    {
        --input_data;
        --output_flt;

/* Calculating new filter output */
        in_array[0] = *input_data;
        out_array[0] = b_coeff[0] * in_array[0];
        for(j = 1; j <= 2; j++)
            out_array[0] += b_coeff[j] * in_array[j] - a_coeff[j] * out_array[j];
        *output_flt = out_array[0];

/* Calculating new filter state values */
        for(k = 2; k >=1; k--)
        {
        in_array[k] = in_array[k-1];
        out_array[k] = out_array[k-1];
        }
    }
}
```

```
void decimate(double *in_sig, double *out_sig, long len_in, int blip,
                        long sam_delay)
{
    long i;

    in_sig += (len_in - sam_delay);
    out_sig += (len_in / blip);
    for(i = (0+sam_delay); i < len_in; i++)
    {
        --in_sig;
        if(((i-sam_delay) % blip) == 0)
        {
            --out_sig;
            *out_sig = *in_sig;
        }
    }
}

void saw_filt_tdl2(double *input, double *tdl_out, long period, int first)
{
/* Gold sequence variables */
    int hloop, iloop, jloop;
    unsigned int reg1, reg2;
    unsigned int feedback1 = 0, feedback2 = 0;
    int onebit = 1, topbit = 1;
    int out1, out2;
    int rtemp1, rtemp2, gtemp1, gtemp2;
    static double tap[255];
/* TDL variables */
    long gloop;
    double tmp_output;
    static double *delay_head, *delay_zero, *delay_tmp;
    static long delay_len;
    int b;

    tdl_out += period;
    input += period;
    for(gloop = 0; gloop < period; gloop++)
    {
        --tdl_out;
        --input;
/* First time saw_filt_tap2 is called, do the following operations */
/* 1) calculate Gold sequence output and place into array tap */
        if(first == 0)
        {
            reg1 = shift_register_init;
            reg2 = shift_register_init;

            for(iloop = 0; iloop < period; iloop++)
            {
          out1 = (reg1 & topbit) ? 1 : 0;
          out2 = (reg2 & topbit) ? 1 : 0;
          tap[period-1-iloop] = (double)(1 - 2 * ((out1 + out2) % 2));

          for(jloop = 0; jloop < shift_register_order; jloop++)
          {
              rtemp1 = (reg1 & onebit) ? 1 : 0;
              rtemp2 = (reg2 & onebit) ? 1 : 0;
              gtemp1 = (pnseq_config1 & onebit) ? 1 : 0;
              gtemp2 = (pnseq_config2 & onebit) ? 1 : 0;
              feedback1 += (rtemp1 * gtemp1);
              feedback2 += (rtemp2 * gtemp2);
              onebit <<= 1;
```

```
            }
            feedback1 %= 2;
            feedback2 %= 2;

            reg1 >>= 1;
            feedback1 <<= (shift_register_order - 1);
            reg1 = reg1 | feedback1;
            reg2 >>= 1;
            feedback2 <<= (shift_register_order - 1);
            reg2 = reg2 | feedback2;

            feedback1 = 0;
            feedback2 = 0;
            onebit = 1;
              }

/* 2) set up the delay array */
              delay_len = period;
              delay_zero = (double *)calloc(delay_len, sizeof(double));
              delay_head = delay_zero;
              if(delay_zero == NULL)
            printf("Not enough memory...");
          }
          first = 1;

/* calculating output of tdl */
          *delay_zero = *input;
          tmp_output = 0.0;
          for(b = 1; b <= period; b++)
          {
              if((delay_zero + b - 1) < (delay_head + delay_len))
            delay_tmp = delay_zero + b - 1;
              else
            delay_tmp = delay_zero - delay_len + b - 1;

              tmp_output += ((*delay_tmp) * tap[b-1]);
          }
          *tdl_out = tmp_output;

          if((delay_zero + delay_len - 1) < (delay_head + delay_len))
              delay_zero += delay_len - 1;
          else
              delay_zero -= 1;
    }
}

void diff_decoder(double *insig, double *outsig, int init, long sig_len)
{
    static double prev_sig;
    long i;


/* initialize prev_data */
    if(init == 0)
        prev_sig = 1.0;

    insig += sig_len;
    outsig += sig_len;
    for(i = 0; i <  sig_len; i++)
    {
        --insig;
        --outsig;
        *outsig = (*insig) * prev_sig;
* set value of prev_sig for next run */
```

```
        prev_sig = *insig;
    }
}
```

Appendix B

This appendix gives a brief review on optical fibers and some of their characteristics.

The optical fiber is basically a cylindrical waveguide usually made of glass or plastic, which uses the principle of total internal reflection to guide light. In its simplest form an optical fiber consists of a central core surrounded by another layer called the cladding, whose refractive index is slightly lower than that of the core. Such a fiber is generally referred to as a step-index fiber. A second kind of fiber is the graded-index fiber. It is characterized by an index of refraction in the core which decreases gradually from the center to the core boundary.

The desirable characteristics of an optical channel include low attenuation, efficient light collection, and low signal distortion. An important source of signal distortion in optical fibers is modal dispersion or pulse spreading. This is caused by the difference in the propagation times of light rays on different paths down the fiber. The number of modes (or different paths) supported by a fiber is determined by the normalized frequency, υ, defined by:

$$\upsilon = k_0 a (n_1^2 - n_2^2)^{1/2}, \tag{B.1}$$

where $k_0 = 2\pi/\lambda$, a is the core radius, and λ is the wavelength of the light. The index of refraction of the core and the cladding are represented by n_1 and n_2 respectively. It can be shown that if $\upsilon < 2.405$ in a step-index fiber, then only a single mode is supported. Fibers designed to satisfy this condition are called single-mode fibers. The main physical

difference between single-mode and multimode fibers is the much smaller core radius of the single-mode fibers. So, while modal dispersion is eliminated in single-mode fibers, their smaller core radius limits their light collection efficiency. Graded-index fibers represent a compromise between good light acceptance and low modal dispersion. Although they support more than one mode, pulse spreading is reduced in graded-index fibers, because the propagation time light on different paths are approximately equal due to the non-uniform refractive index of the core.

Of the three fiber types discussed, single mode fibers have become the dominant and most widely used fiber type in the telecommunications industry. They tend to be preferred for several reasons. Among the most important of these are the facts that single-mode fibers exhibit the highest transmission bandwidths and lowest losses of all fiber types and that they have superior transmission quality due to the absence of modal dispersion. Multimode fibers are however still used in more localized communications, for example local area networks.

Appendix C

This appendix provides an introduction to spreading sequences and their properties followed by a review of some commonly used pseudo-random sequences other than those covered in Section 3.2.

In most networks, multiple-access techniques are used to transmit the signals of many users over a shared transmission medium. Code division multiple access techniques rely primarily on superimposing different sequences or codes on transmitted data to provide the multiple-access capability and—unlike traditional time- and frequency division multiple access—don't require precise time or frequency coordination between transmitters in the system. In a CDMA system, a user superimposes a pseudo-random sequence onto his message signal. The desired receiver can recover the transmitted information intended for it by knowing the code used by the corresponding transmitter. CDMA is an application of spread spectrum. In fact it is sometimes referred to as spread spectrum multiple access (SSMA).

Spread spectrum is a modulation and demodulation technique which is distinguished by the fact that the transmission bandwidth is much greater than the minimum bandwidth required to transmit the information.

One method for spreading the spectrum of an information signal is to modulate the signal a second time using a very wideband spreading signal. When the bandwidth is spread by direct modulation of a data modulated carrier with a spreading signal or code, this is

called direct-sequence (DS) spread spectrum. With another common form of spread spectrum, the spreading code is used to control the frequency of transmission of the data modulated carrier, thus indirectly modulating the information signal by the spreading code. The material in this thesis only presents information on direct-sequence spread spectrum, as it is probably the most commonly used type of spread spectrum.

The simplest form of DS spread spectrum uses binary phase shift keying (BPSK) as the spreading modulation. For our purposes a code will be defined as a periodic sequence of ones and zeros. Multivalued sequences also exist, but we will focus on binary sequences as they are more commonly used in communications applications. The spreading codes used are often referred to as pseudo-random or pseudo-noise (PN) sequences due to their noise-like properties. Specific PN-sequences include m-sequences and Gold codes among others.

In most practical applications, a binary sequence is actually transmitted as a sequence of unit amplitude, positive and negative pulses. This pulse sequence is obtained from the original sequence of zeros and ones, by replacing each 1 with $\{-1\}$ and each 0 by $\{+1\}$. The term spreading code is often used to refer to the sequence obtained from a binary shift register, whereas the term spreading waveform is usually employed to refer to the sequence of positive and negative pulses that is used as input to the spreading or despreading modulator in a communications system.

As stated earlier, sequences used in spread spectrum applications must have certain desirable characteristics. The tools to measure the effectiveness of sequences will now be described.

In systems where spread spectrum is frequently used, sequences must have one or both of the following properties [15]:

(i) each signal in the set is easy to distinguish from a time-shifted version of itself;

(ii) each signal in the set is easy to distinguish from (a possibly time-shifted version of) every other signal in the set.

The second property is particularly important for CDMA systems, where a receiver must detect the signal intended for it among signals from several transmitters.

In most systems, periodic signals are used because of the simplifications in system implementation that result. Consequently this discussion will consider only sets of signals with the property that for some T, $x(t) = x(t+T)$ for all t and every signal x in the set.

One of the most common and most useful measures of distinguishability is the mean squared difference. For our purposes, two signals are easy to distinguish if and only if the mean squared difference between them is large, and we will require not only that $x(t)$ is easy to distinguish from $y(t)$, but also that $-x(t)$ is easy to distinguish from $y(t)$. Both $+x(t)$ and $-x(t)$ must be considered whenever modulation processes are involved, such as when binary data is modulated onto $x(t)$ or when $x(t)$ is modulated onto a carrier signal. Thus the measure of distinguishability is the quantity

$$\frac{1}{T}\int_0^T [y(t) \pm x(t)]^2 dt = \frac{1}{T}\left\{\int_0^T [y^2(t) + x^2(t)] \pm 2\int_0^T x(t)\, y(t)\, dt\right\} . \qquad \text{(C.1)}$$

The first integral on the right hand side of (C.1) is the energy in $x(t)$, $0 < t < T$, plus the energy in $y(t)$, $0 < t < T$. Thus for fixed signal energy, $y(t)$ is easy to distinguish from both $+x(t)$ and $-x(t)$ if and only if the magnitude of the quantity r as defined below is small.

$$r = \int_0^T x(t)\, y(t)\, dt \qquad \text{(C.2)}$$

In a communications system using correlation receivers or matched filters, r represents the output of the filter matched to $y(t)$ when the input is $x(t)$. In a multi-user system $x(t)$ and $y(t)$ may represent the signals assigned to two different transmitters, in which case the parameter r is a measure of the crosstalk interference between the two signals.

Properties (i) and (ii) require distinguishability of $x(t)$ and $y(t+\tau)$ for all $[0, T]$ where $x(t)$ and $y(t)$ may be the same or different signals and where τ represents a delay imposed on $y(t)$. Consequently it is the magnitude of the cross-correlation function

$$r_{x,y}(\tau) = \int_0^T x(t)\, y(t+\tau)\, dt \qquad \text{(C.3)}$$

that is of interest.

Due in part to the relative simplicity of their generation, the signals of interest for most applications are periodic signals which consist of sequences of elemental time pulses, also called chips. These pulses are all of the same shape so that a signal can be written as

$$x(t) = \sum_{n=-\infty}^{\infty} x_n \phi(t - nT_c) , \tag{C.4}$$

where $\phi(t)$ is the basic pulse waveform and T_c is its time duration. If $x(t)$ has period T for all t, then T must be a multiple of T_c and the sequence (x_n) must be periodic with a period which is a divisor of $N = T/T_c$.

Similarly $y(t)$ can be written as

$$y(t) = \sum_{n=-\infty}^{\infty} y_n \phi(t - nT_c) . \tag{C.5}$$

It is easy to see, then, that the parameter r of (C.2) is given by

$$r = \lambda \sum_{n=0}^{N-1} x_n y_n , \tag{C.6}$$

where the constant λ is

$$\lambda = \int_0^{T_c} \phi^2(t)\, dt . \tag{C.7}$$

For example, if $\phi(t) = p_{T_c}(t)$, a unit amplitude rectangular pulse of duration T_c which starts at $t = 0$, then $\lambda = T_c$. Assuming that $y(t)$ is delayed by $\tau = lT_c$ relative to $x(t)$, then 3.6 generalizes to

$$r_{x,y}(\tau) = \lambda \sum_{n=0}^{N-1} x_n y_{n+l} \,. \tag{C.8}$$

In this case (C.8) can be rewritten as

$$r_{x,y}(\tau) = \lambda \theta_{x,y}(l) \,. \tag{C.9}$$

where $\theta_{x,y}(l)$ is the periodic cross-correlation of the sequences x and y. For arbitrary values of τ (not an exact multiple of T_c as defined above), $r_{x,y}(\tau)$ can be determined from the periodic cross-correlation function. For instance, if $\phi(t) = p_{T_c}(t)$ then for $0 \le \tau < T$.

$$r_{x,y}(\tau) = T_c \theta_{x,y}(l') + (\tau - l'T_c)\,[\theta_{x,y}(l'+1) - \theta_{x,y}(l')] \,. \tag{C.10}$$

where l' is the largest integer such that $l'T_c < \tau$. It is also worth noting that for any choice of elemental pulse $\phi(t)$.

$$\max\{|r_{x,y}(\tau)| : 0 \le \tau \le T\} = \lambda \max\{|\theta_{x,y}(l)| : 0 \le l \le N-1\} \,. \tag{C.11}$$

Since the periodic cross-correlation parameters of the continuous time signals $x(t)$ and $y(t)$ of (C.4) and (C.5) are completely determined by the cross-correlation function, the system design problem described earlier reduces to the problem of finding sets of periodic sequences with the following properties:

(i) for each sequence x, the magnitude of the autocorrelation is small for $1 \le l \le N-1$

(ii) for each pair of sequences x and y, the magnitude of the cross-correlation $\theta_{xy}(l)$ is small for all l.

We will now proceed to define the cross-correlation function and related properties of sequences in more detail. Assume $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$ denote vectors of length N, where $\boldsymbol{x} = (x_0, x_1, \ldots, x_{N-1})$. The inner product $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ of two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ is defined by $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = x_0 y_0 + x_0 y_0 + \ldots + x_{N-1} y_{N-1}$. Note that $\langle \boldsymbol{x}, \boldsymbol{x} \rangle$ is a positive number for all non-zero vectors. The norm $\|\boldsymbol{x}\|$ of x is the positive square root of $\langle \boldsymbol{x}, \boldsymbol{x} \rangle$ and $\sum x$ denotes $x_0 + x_1 + \ldots + x_{N-1}$. Let T denote the operator which shifts vectors cyclically to the left by one place, that is $T\boldsymbol{x} = (x_1, x_2, \ldots, x_{N-1}, x_0)$. If T is applied k times to $\boldsymbol{x}$, the result is $T^k \boldsymbol{x}$. We see that $T^k \boldsymbol{x} = (x_k, x_{k+1}, \ldots, x_{N-1}, x_0, x_1, \ldots, x_{k-1})$ for $0 < k < N$ while $T^N \boldsymbol{x} = \boldsymbol{x}$. For larger values of k, $T^k \boldsymbol{x} = T^{k'} \boldsymbol{x}$ where $k' = k \bmod N$. Similarly the operator T^{-1} shifts vectors cyclically to the right by one place, and it is easy to see that $T^{-k} \boldsymbol{x} = T^{N-k} \boldsymbol{x}$ for $0 < k < N$. We also have that $\|T^k \boldsymbol{x}\| = \|\boldsymbol{x}\|$ and $\sum (T^k \boldsymbol{x}) = \sum \boldsymbol{x}$. The period of $\boldsymbol{x}$ is defined to be the least positive integer M such that $T^M \boldsymbol{x} = \boldsymbol{x}$. For $T^i \boldsymbol{x} \neq T^j \boldsymbol{x}$ for $0 < i < j < M$, the vectors $\boldsymbol{x}$, $T\boldsymbol{x}$, $T^2\boldsymbol{x}$,..., $T^M\boldsymbol{x}$ are cyclically equivalent; that is, they are cyclic shifts of each other. Generally M can be any divisor of N, but in most cases of interest we have $M = N$. Given $\boldsymbol{x}$ a vector of length N, we can generate an (infinitely long) periodic sequence x by repeating the vector $\boldsymbol{x}$ over and over again. More formally given $\boldsymbol{x} = (x_0, x_1, \ldots, x_{N-1})$, the periodic sequence x is generated from $\boldsymbol{x}$ by:

$$x = \ldots x_{-2}, x_{-1}, x_0, x_1, x_2, \ldots x_{N-1}, x_N \ldots . \tag{C.12}$$

For vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ of length N we define the periodic cross-correlation function $\theta_{x,y}(\cdot)$ by

$$\theta_{x,y}(l) = \langle x, T^l y \rangle, \; l \in Z \; . \tag{C.13}$$

where Z is the set of all integers. If x and y are the sequences generated from $\boldsymbol{x}$ and $\boldsymbol{y}$ respectively the (C.13) is equivalent to:

$$\theta_{x,y}(l) = \sum_{i=0}^{N-1} x_i y_{i+l} \quad . \tag{C.14}$$

It is easy to verify that for each $l \in Z$

$$\theta_{x,y}(l) = \theta_{x,y}(l+N) \tag{C.15}$$

and

$$\theta_{x,y}(-l) = \theta_{x,y}(l) \, . \tag{C.16}$$

The periodic autocorrelation function $\theta_x(l)$ for the sequence x is just $\theta_{xx}(.)$.

It is often of interest to provide bounds on the maximum values of the correlation functions. For a set $\Re$ of periodic sequences, the peak cross-correlation magnitude is given by

$$\theta_c = \max\{|\theta_{x,y}(l)| : 0 \le l \le N-1, x \in \Re, y \in \Re, x \ne y\} \tag{C.17}$$

and θ_a, the peak out-of-phase autocorrelation magnitude is given by

$$\theta_a = \max\{|\theta_{x,x}(l)| : 1 \le l \le N-1, x \in \Re\} \, . \tag{C.18}$$

θ_{max} is the maximum correlation function defined by

$$\theta_{max} = \max\{\theta_a, \theta_c\} \, . \tag{C.19}$$

So far only periodic correlation functions have been described. In a CDMA system characterized by potentially arbitrary delays between signals, it is also necessary to be familiar with the aperiodic cross-correlation properties of the sequences involved. Given a pair of sequences x, y with period N, the discrete aperiodic cross-correlation function $C_{x,y}(l)$ is defined by:

$$C_{x,y}(l) \begin{cases} \sum_{i=0}^{N-1-l} x_i y_{i+l}, & 0 \le l \le N-1; \\ \sum_{i=0}^{N-1+l} x_{i-l} y_i, & 1-N \le l \le 0; \\ 0, & |l| \ge N. \end{cases} \tag{C.20}$$

Notice that the periodic cross-correlation function can be expressed as a function of $C_{x,y}(l)$ as follows:

$$\theta_{x,y}(l) = C_{x,y}(l) + C_{x,y}(l-N) \quad \text{for } 0 \le l < N . \tag{C.21}$$

We also define the odd cross-correlation function by

$$\hat{\theta}_{x,y}(l) = C_{x,y}(l) - C_{x,y}(l-N) \quad \text{for } 0 \le l < N . \tag{C.22}$$

The properties of some common types of pn-sequences, namely m-sequences and Gold sequences, were described in Section 3.2. For multiuser systems, it is desirable to find sets of sequences with good cross-correlation properties. We will now consider other sets of periodic sequences which have good periodic correlation as measured by the peak periodic correlation parameters and θ_c and θ_a as defined in (C.17) and (C.18) respectively.

C.1 Preferred pairs of m-sequences

Gold and Kasami proved that certain pairs of m-sequences exhibit a three-valued cross-correlation function with values $\{-1, -t(n), t(n)-2\}$ where $t(n)$ is defined as

$$t(n) = 1 + 2^{\lfloor (n+2)/2 \rfloor}, \tag{C.23}$$

where $\lfloor \alpha \rfloor$ denotes the integer part of the real number α. Two m-sequences of length N with the above three-valued cross-correlation function are referred to as a preferred pair of m-sequences. Larger sets of sequences with better cross-correlation properties than m-sequences are derived from preferred pairs of m-sequences

C.2 Maximal connected sets of m-sequences

A connected set of m-sequences is collection of m-sequences which has the property that each pair in the collection is preferred pair. A largest possible connected set is called a maximal connected set.

While sets of maximal-connected m-sequences are larger that just pairs, they are still relatively small. They are useful for applications which require only a few sequences with excellent cross-correlation and autocorrelation properties. However most applications (such as CDMA) require much larger sets of sequences. Unfortunately, large sets of m-sequences generally have quite poor cross-correlation properties, and thus are inadequate for such applications. It is therefore desirable to obtain larger sets of sequences with period $N = 2^n - 1$ which have a bound $\theta_c \le t(n)$ on peak periodic cross-correlation

as for maximal connected sets. Since these larger sets, such as Gold or Kasami sequences must contain some non-maximal length sequences, then their peak periodic autocorrelation θ_a must exceed 1.

C.3 Kasami sequences

Kasami sequences are generated using a procedure similar to that used for Gold sequences. Before going any further let us define the parameter $s(n)$ as follows:

$$s(n) = \frac{1}{2}[t(n) + 1] = 1 + 2^{n/2}. \tag{C.24}$$

Let n be even and let u denote an m-sequence of period $N = 2^n - 1$ generated by $h(x)$. Consider the sequence $w = u[s(n)] = u[2^{n/2} + 1]$. It can be shown that w is a sequence of period $2^{n/2} - 1$. It is generated by the polynomial $h'(x)$ of degree $n/2$ and is thus an m-sequence. Now, consider the sequences generated by the polynomial $h(x)h'(x)$ of degree $3n/2$. Clearly any such sequence must be one of the forms $T^i u$, $T^j w$, $T^i u \oplus T^j w$, $0 < i < 2^n - 1$, $0 < j < 2^{n/2} - 1$. Thus any sequence y of period $2^n - 1$ generated by $h(x)h'(x)$ is some phase of a sequence in the set $K_s(u)$ defined by

$$K_s(u) \equiv \{u, u \oplus w, u \oplus Tw, \ldots, u \oplus T^{2^{n/2}-2}w\}. \tag{C.25}$$

This set is called the small set of Kasami sequences. The correlation function for sequences belonging to $K_s(u)$ take on values in the set $\{-1, -s(n), s(n) - 2\}$. Consequently for the set $K_s(u)$,

$$\theta_{max} = S(n) = 1 + 2^{n/2}. \tag{C.26}$$

Notice that for the set $K_S(u)$ is approximately half of the value of achieved for sets of Gold sequences. On the other hand, $K_S(u)$ contains only $2^{n/2} = (N+1)^{1/2}$ sequences while gold sequence sets contain $N+2$ sequences.

Now let $\hat{h}(x)$ denote the polynomial which generates the sequence $u[t(n)]$. The polynomial $h(x)\hat{h}(x)$ generates the set of Gold sequences $G(u, u[t(n)])$ if $n \neq 0 \mod 4$. Now all the sequences generated by $f(x) = h(x)\hat{h}(x)h'(x)$ are of the form $a \oplus b \oplus c$ where a, b, c are generated by $h(x)$, $\hat{h}(x)$ and $h'(x)$ respectively. The set of sequences of period N generated by $h(x)\hat{h}(x)h'(x)$ is called the large set of Kasami sequences and is denoted by $K_L(u)$. This set contains both a small set of Kasami sequences and a set of Gold sequences as subsets. More interestingly, the correlation bound $\theta_{max} = t(n)$ is the same as that of the second subset.

It is not difficult to see that the shift-and-add property holds for the small and the large set of Kasami sequences.

Bibliography

[1]Darcie, T. E., "Subcarrier Multiplexing for Multiple-Access Networks", Journal of Lightwave Technology, Vol. LT-5, No. 8, Aug 1987, Pp. 1103-1110

[2]Darcie, Thomas E., "Subcarrier Multiplexing for Lightwave Networks and Video Distribution Systems", IEEE Journal on Selected Areas in Communications, Vol. 8, No. 7, Sept 1990, Pp. 1240-48

[3]Darcie, T. E., and G. E. Bodeep, "Lightwave Multi-Channel Analog AM Video Distribution Systems", GLOBECOM'89 (1989), Pp. 1004-1007

[4]Olshansky, Robert, V. A Lanzisera, and P. M. Hill, "Subcarrier Multiplexed Lightwave Systems for Broadband Distribution", Journal of Lightwave Technology, Vol. 7, No. 9, Sept 1989, Pp. 1329-1341

[5]Way, Winston I., "Subcarrier Multiplexed Lightwave System Design Considerations for Subscriber Loop Applications", Journal of Lightwave Technology, Vol. 7, No. 11, Nov 1989, Pp. 1806-1818

[6]Palais, Joseph C., Fiber Optic Communications, Prentice-Hall, Englewood Cliffs, N.J., 1992

[7]Vanucci, Giovanni, "Combining Frequency Division and Code Division Multiplexing in a High-Capacity Optical Network", IEEE Network, Mar 1989, Pp. 21-30

[8]Khaleghi, Farideh, "Spread-spectrum Fiber LANs", M.A.Sc. Thesis, University of Ottawa, 1992

[9]Khaleghi, F., and M. Kavehrad, "A Subcarrier Multiplexed CDM Optical Local Area

Network Theory and Experiment". IEEE Transaction on Communications. Vol 43. No. 1. Jan 1995. Pp. 75-87

[10]Senior. John M.. Optical Fiber Communications: Principles and Practice. Prentice-Hall International Inc.. London. 1992

[11]Miller. Stewart E. and Ivan P. Kaminow (Ed.). Optical Fiber Telecommunications II. Academic Press. San Diego. Ca.. 1988

[12]Kressel. Howard (Ed.). Semiconductor Devices for Optical Communication. Springer Verlag. Heidelberg. 1982

[13]Harder. Christoph. et al.. "Noise Equivalent Circuit of a Semiconductor Laser Diode". IEEE Journal of Quantum Electronics. Vol. QE-18. No. 3. Mar 1982. Pp. 333-337

[14]Henry. Paul S.. "High-Capacity Lightwave Local Area Networks". IEEE Communications Magazine. Oct 1989. Pp. 20-26

[15]Sarwate. D. V.. and M. B. Pursley. "Cross-correlation Properties of Pseudorandom and Related Sequences". Proceedings of the IEEE. Vol. 68. No. 5. May 1980. Pp. 593-619

[16]Ziemer. Rodger E.. and R. L. Peterson. Digital Communications and Spread Spectrum Systems. MacMillan Publishing Company. New York. N.Y.. 1985

[17]Holmes. Jack K.. Coherent Spread Spectrum Systems. Robert E. Krieger Publishing Company. Malabar. Fa. 1982

[18]Proakis. John G.. Digital Communications. McGraw-Hill Book Company. New York. N.Y.. 1989

[19]Pursley. Michael B.. "Performance Evaluation for Phase-Coded Spread-Spectrum Multiple-Access Communication - Part I: System Analysis". IEEE Transactions on

Communications. Vol. COM-25. No. 8. Aug 19777. Pp. 795-799

[20]Pursley. Michael B. and D. V. Sarwate. "Performance Evaluation for Phase-Coded Spread-Spectrum Multiple-Access Communication - Part II: Code Sequence Analysis". IEEE Transactions on Communications. Vol. COM-25. No. 8. Aug 1977. Pp. 800-803

[21]Neusy. Philippe C.. "Subcarrier Multiplexing for Fiber-Based Video Distribution: A Performance Study". M. A. Sc. Thesis. University of Ottawa. 1991

[22]Gregorian. R.. and G. C. Temes. Analog MOS Integrated Circuits for Signal Processing. John Wiley and Sons. New York. N.Y.. 1986Press. William H..

[23]Numerical Recipes. Cambridge University Press. Cambridge. N.Y.. 1986

[24]Weldon. P. A.. W. W. Peterson. and E. J. MacLane. A Survey of Modern Algebra. Macmillan. New York. N.Y.. 1965

[25]Lau. K. Y.. A. Yariv. "Intermodulation Distortion In a Directly Modulted Semiconductor Injection Laser". Applied Physics Letters. Vol. 45. No. 10. Nov 1984 Pp. 1034-1036

[26]Way. Winston I.. "Large Signal Nonlinear Distortion Prediction for a Single-Mode Laser Diode under Microwave Intensity Modulation". Journal of Lightwave Technology. Vol. LT-5. No. 3. Mar 1987 Pp 305-315

[27]Darcie. T. E.. R. S. Tucker and G. J. Sullivan. "Intermodulation and Harmonic Distortion in InGaAsP Lasers". Electronic Letters. Vol 21.. No. 16. June 1985. Pp 665-666

[28]Yamamoto. Y.. "AM and FM Quantum Noise in Semiconductor Laser - Part I: Theoretical Analysis". IEEE Journal of Quantum Electronics. Vol. QE-19. No. 1. Jan 1983. Pp. 34-46.