Bit Error Rate Simulation Enhancement and Outage Prediction in Mobile Communication Systems

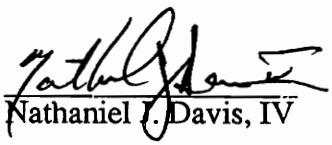
by

Berthold Thoma

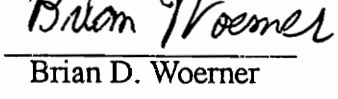
Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of
Master of Science
in
Electrical Engineering

APPROVED

Theodore S. Rappaport, chairman



Nathaniel J. Davis, IV



Brian D. Woerner

July 1992
Blacksburg, Virginia

Bit Error Rate Simulation Enhancement and Outage Prediction in Mobile Communication Systems

by

Berthold Thoma

Committee chairman:
Theodore S. Rappaport
Electrical Engineering

(ABSTRACT)

This thesis presents enhancements made to software designed for simulating digital mobile radio communication systems. Internally generated two-ray Rayleigh fading channel models as well as imported spatially varying complex channel impulse responses can be utilized in simulations. Implementation of receiver timing jitter and timing recovery models, as well as modifications of the channel generation (two-ray model) and conversion (imported channel), are among the enhancements that have been added.

The bit error rate and outage performance of the simulator is compared with analytical results. A study of the impact of the distribution of magnitude and phase components within a channel impulse response is performed. A study utilizing both timing recovery and timing jitter models is also included.

It has been found that for site specific BER and outage prediction, more descriptive channel parameters than narrowband fading statistics and statistical moments of a channel's power delay profile are needed. The studies also indicate that for simulations of channels with large delay spread timing recovery is essential.

Acknowledgments

I would like to thank Professor Theodore S. Rappaport for being my advisor and giving me the opportunity to work with the Mobile and Portable Radio Research Group. I also want to thank my committee members Professor Nathaniel J. Davis, IV and Professor Brian D. Woerner for their corrections and comments on my thesis.

I am very grateful to every member of the Mobile and Portable Radio Research Group, who assisted me throughout this thesis. In particular I want to thank Scott Seidel and Kurt Schaubach for reviewing my thesis as well as Linda Solowiej for her help in preparation of the manuscript. In addition, I want to thank Mike Keitz and Scott Seidel for providing some of the figures.

I would also like to acknowledge Senator John Fulbright for establishing the exchange program which gave me the opportunity to study at Virginia Tech and supported me during my stay in the United States.

Finally, special recognition is hereby expressed to my family for their commitment to education and guidance throughout my studies.

Table of Content

List of Figures

List of Tables

1. Introduction

1.1 Purpose of this Research

Lately, more and more people refer to our time as the information age [Luc89]. Never before have people been able to access or utilize such vast amounts of information. A recent growing demand is the ability to communicate with someone independent of their location. Cellular telephone is one of the key technologies that provides this service [Goo91a].

Since radio spectrum is a limited resource, the applied technology has to be as spectrally efficient as possible. Digital communication techniques, which promise better spectral efficiencies than the currently used analog systems, are being introduced throughout the world [Goo91b][Rap91b].

If these systems are to replace the current analog systems and grow to provide ubiquitous coverage, it is essential that the new systems provide voice quality transmission comparable to or superior to the current analog systems. This is especially true for the already introduced, (GSM[1], CT-2[2]) [GSM88] [Swa90] and the proposed (U.S. Digital Cellular, DECT[3]) [EIA89][DEC89] time division multiple access (TDMA) systems. In these new systems, sufficiently small bit error rate performance for each data frame has to be maintained within a specified coverage area. Simulation of such systems is a valuable alternative to field trials, since field trials involve high costs for equipment as well as for personnel, and because reliable measurement equipment that allows differentiation

1. Pan-European digital cellular system, also adopted by several countries outside Europe. Earlier: **G**roupe Special **M**obile, now: **G**lobal **S**ystem for **M**obile Communication
2. British Standard for second generation **C**ordless **T**elecommunications
3. **D**igital **E**uropean **C**ordless **T**elecommunications

between different bit error sources is not available. Furthermore, simulations produce reproducible results and allow for experimentation under well-defined conditions. Complex channel sounders and statistical channel models, as well as site-specific channel models based on ray tracing, will soon provide very good channel descriptions that can be used for accurate BER[1] computations [Rap92]. Since analytical methods are not yet available for wideband systems, the knowledge of the channel impulse response in combination with a simulator is the only viable way of studying both the bit error mechanisms and the temporal distribution of errors in mobile systems.

For this thesis, an existing bit error rate simulator developed in the MPRG[2] [Fun91] was modified for use as a research tool by other researchers as well as to provide insight into bit error rate mechanisms analysed in this thesis. The purpose of this thesis is to give a detailed understanding of how the current version of the **B**it **E**rror **R**ate **SIM**ulator, BERSIM, works as well as to explore the properties of a high data rate wireless mobile and portable communications link. In order to evaluate the performance of a system or to explore sources for received errors, several system parameters, such as velocity of the mobile unit, data rate, carrier frequency, type of modulation, and the channel characteristics are significant.

An important figure of merit analyzed in this thesis, besides the raw bit error rate, is the outage probability. In contrast to the bit error rate, which is clearly defined as the ratio of bits received in error to the total number of transmitted bits, the outage rate has been defined in several ways in the literature [Jak74][Sow88][Cha86]. Since this work deals with digital communications, the outage probability is always defined as the probability

1. **B**it **E**rror **R**ate
2. **M**obile and **P**ortable Radio **R**esearch **G**roup at Virginia Tech

that the number of errors within a block of successive bits exceeds a fixed threshold. If one considers the block as a word of coded bits, it can also be seen as the probability of receiving a word in error. In this case, the outage criterion represents the maximum number of errors a decoder is capable of correcting. In a real-world TDMA or packet radio system, an outage describes the condition under which a received frame cannot be used.

1.2 Previous Work

1.2.1 Previous Work Performed on the Program BERSIM.

The fundamental principles of this work are based on the Masters Thesis of Victor Fung [Fun91] [Rap91c]. He developed the original version of the bit error rate simulator BERSIM to predict bit error rates for outdoor flat and frequency selective fading channels as well as for indoor channels using measurement-based impulse response models generated by the indoor channel simulator SIRCIM[1] [Sei89][Rap91a]. Victor Fung's original version of BERSIM (from here on called BERSIM 1.0) was written in the PC specific language TURBO PASCAL with heavy emphasis on global variables. Because of the upcoming need for more powerful computing platforms, further development and maintenance of his original code was needed to develop a portable, modular, and maintainable version of BERSIM. These goals were to be accomplished by converting the program into the well standardized computer language, ANSI[2] C, which is an engineering industry standard programming language for all types of computer platforms. The ground layer for this work was provided by an almost one to one translation of Victor Fung's original code into ANSI C to support more powerful platforms such as UNIX-based workstations [Tho92a].

1. Simulation of Indoor Radio Channel Impulse response Models
2. American National Standards Institute

1.2.2 Previous Work on Bit Error Rate Prediction for Digital Mobile radio.

Three different fundamental approaches to predict bit error rates for digital mobile radio systems are found in literature. These methods are:

- Analytical (statistical techniques)
- Simulation of the communication system using measured channel characteristics
- Simulation of the communication system using a simulated channel model

Predicting bit error rates using analytical methods is the area where most research has focused over the last few decades. Pioneering work was performed by Bello and Nelin in their papers on binary error probabilities [Bel 62a] [Bel62b] [Bel63]. They were among the first to analyze the relationship between the fading rate (frequency of the occurrence of fades which is directly related to velocity of the mobile) and the bit error rate, as well as the impact of frequency selective fading on wireless digital data transmission.

However, Bello and Nelin made some errors in their calculations determining bit error probabilities for frequency selective radio channels. These errors where discovered by Bailey and Lindenlaub [Bai68] using methods similar to Bello and Nelin to study the performance of binary DPSK modulation in frequency-selective channels with matched filter reception. For their study, they used two different channel impulse responses as well as different signaling pulse shapes. It was concluded that the shape of the transmitted signaling pulse has a larger impact on the bit error rate than the shape of the channel impulse response. Both [Bai68] and [Bel63] considered only intersymbol interference from adjacent symbols.

Different channel impulse response shapes in combination with a timing recovery model were used by Glance and Greenstein [Gla83]. They concluded, that for rms delay spreads

[Jak74] less than 0.2 times the symbol period, the bit error rate for a frequency selective fading channel is independent of the shape of the impulse response. However, a dependency for larger rms delay spreads was discovered.

More recent analytical work was done by Liu and Feher [Liu89] [Liu90] [Liu91] as well as by Ohno and Adachi [Ohn90] [Ada92]. Those studies analysed the performance of the $\pi/4$ DQPSK modulation technique in frequency-selective Rayleigh fading channels. Similar results as for previously mentioned studies were found. Analytical expressions for outage probabilities for systems with cochannel interference were derived by Sowerby [Sow88] [Sow92].

With availability of more computing power as well as suitable simulation tools [Sha88], simulating a mobile radio system continuously gained popularity. Chuang was one of the first to simulate such a complete system [Chu87a] for his studies on the influence of delay spread on bit error rates as well as for outage prediction [Chu91] [Cha89a] [Cha89b]. Recently, commercially available software packages like BOSS[1] (Block Oriented System Simulator) have been used to implement mobile radio communication systems [Le92]. However, almost all of those systems use very simple channel models such as the two-ray channel model [Jak74], or they use a constant channel impulse response and then superimpose fading effects on the entire profile.

With the advent of complex channel sounders, which provide a complex channel impulse response, as proposed in [Lor91], real world channels can be used to study BER and outage rate, and their temporal distributions very realistically [Kad92]. Another approach can

1. Product of Comdisco Systems, Inc.

be taken by using realistic channel impulse response models as described in [Sei89] [Rap91a] [Hua91] instead of measured channels.

1.3 Outline of Thesis

The outline of this thesis is as follows: In Chapter 2, an introduction to the basic features and properties of a mobile radio propagation channel are given and frequently used terms describing this channel are explained. In Chapter 3, the new version of the bit error rate simulator, (termed BERSIM 2.0 throughout this thesis) is described in block diagram fashion, illustrating the flow of the information signal through the communication system. Major modifications and new features added to the original implementation of the bit error simulator will be described in detail in this chapter. Chapter 4 supplements Chapter 3 by giving a description of the software implementation of BERSIM 2.0. Chapter 5 describes simulations that were done and discusses their results. Chapter 6 is a summary with conclusions and recommendations for future work. References and appendices containing flow and structure charts are at the end of this thesis.

2. The Mobile Radio Communication Channel

The propagation channel is responsible for almost all bit error mechanisms existing in a digital mobile radio communication system. It is therefore of essential importance to have a detailed understanding of the channel in order to differentiate between different bit error sources.

Communication channels are generally characterized as narrowband or wideband depending on the symbol rate transmitted through the channel. The measure for determining whether the channel is either narrowband or wideband is the coherence bandwidth [Jak74] [Par89]. For a signal whose bandwidth is less than the coherence bandwidth, the channel is considered to be a narrowband channel, and vice versa. A definition of the coherence bandwidth will be given below together with a description of other important channel parameters.

Like any linear system, the propagation channel can be described in the time domain by its channel impulse response. In a typical mobile radio environment, the channel impulse response is a function of time as well as location. Since location of a mobile can be considered a function of time, the channel impulse response can be described as a time variant system. A typical notation for such a system is $h(t, \tau)$ where t indicates the time dependence and τ the channel delay. For computer simulations this channel can be modeled as a discrete time varying filter such as

$$h(t, \tau) = \sum_{i=0}^{N-1} a_i(t, \tau)\, e^{j\phi_i(t, \tau)}\, \delta(\tau - \tau_i) \tag{2.1}$$

where $a_i(t,\tau)$ is the amplitude of a multipath component with a delay of τ. The variable $\phi_i(t,\tau)$ is the corresponding phase. Time variance is evident even when the portable or mobile unit is not moving. This is because of the constant physical motion that exists in the world around the system. Such a channel is commonly termed a non-stationary channel. Figure 2.1 shows a power-delay profile, $p(\tau)$ which is the signal magnitude square of a channel impulse response, for a channel at a particular instant of time.

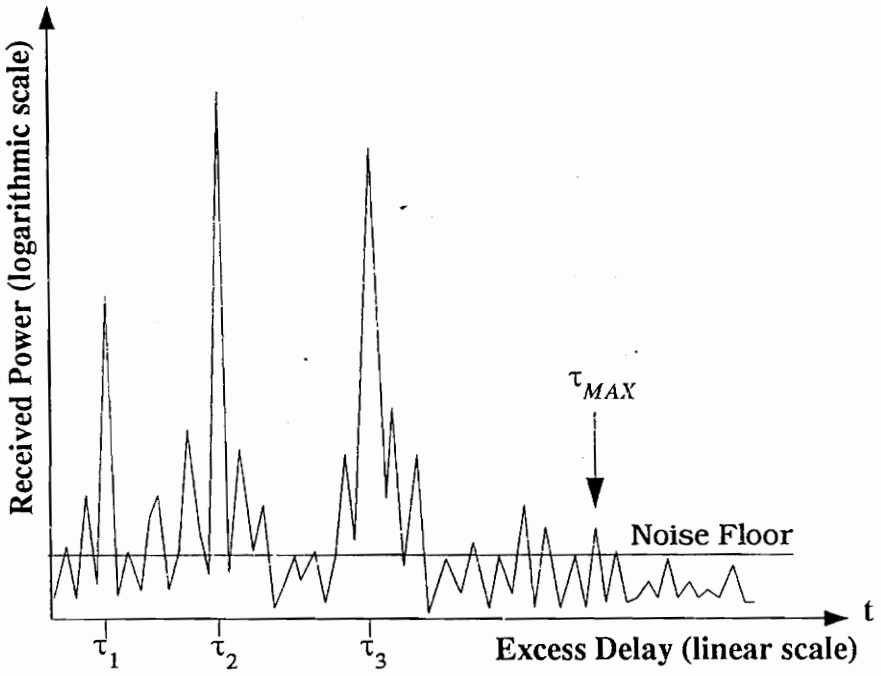


FIGURE 2.1: Power delay profile of mobile or portable radio channel

The power delay profile is frequently used to characterize the channel. This is because the power delay profile can be easily measured using wide band channel sounders, which can resolve delayed signal components. There are also several statistical moments of the power delay profile that are of practical interest and are commonly used to coarsely quantify the time dispersion of the channel. The two most important moments are the mean

excess delay $\bar{\tau}$ which is the normalized mean of $P(\tau)$ and the rms delay spread σ_{RMS} which is the variance of $P(\tau)$. They are defined as [Bel63] [Jak74] [Par92]:

$$\bar{\tau} = \frac{\int_{-\infty}^{\infty} \tau P(\tau)\, d\tau}{\int_{-\infty}^{\infty} P(\tau)\, d\tau} \tag{2.2}$$

and

$$\sigma_{RMS} = \sqrt{\frac{\int_{-\infty}^{\infty} (\tau - \bar{\tau})^2 P(\tau)\, d\tau}{\int_{-\infty}^{\infty} P(\tau)\, d\tau}} \tag{2.3}$$

For determining the coherence bandwidth, the maximum excess delay τ_{MAX} is also of significance. It is defined as the maximum delayed component that is within x dB of the strongest arriving multipath component, where x denotes the value of the signal level below the maximum signal level.

With knowledge of the channel impulse response, the coherence bandwidth can be defined as the inverse of the maximum excess delay τ_{MAX} [Par92]. As a result, for signals with a bandwidth much smaller than this inverse, all spectral components are affected in a nearly similar fashion resulting in strongly correlated signal components, where the propagation delay differences between different signals are negligible. This signals are termed narrow band signals and the channel is considered to be a narrowband channel.

Since the whole purpose of wireless communications is to provide tetherless communication service to subscribers, the systems are usually set up in or around heavily built-up

areas where people congregate, including inside buildings. In most cases, the received signal is a result of scattered signals from the surfaces of buildings and of diffraction over and around them. Figure 2.2 shows a two-dimensional model, where the three ellipses denote all possible reflector locations for signals with three different time delays.

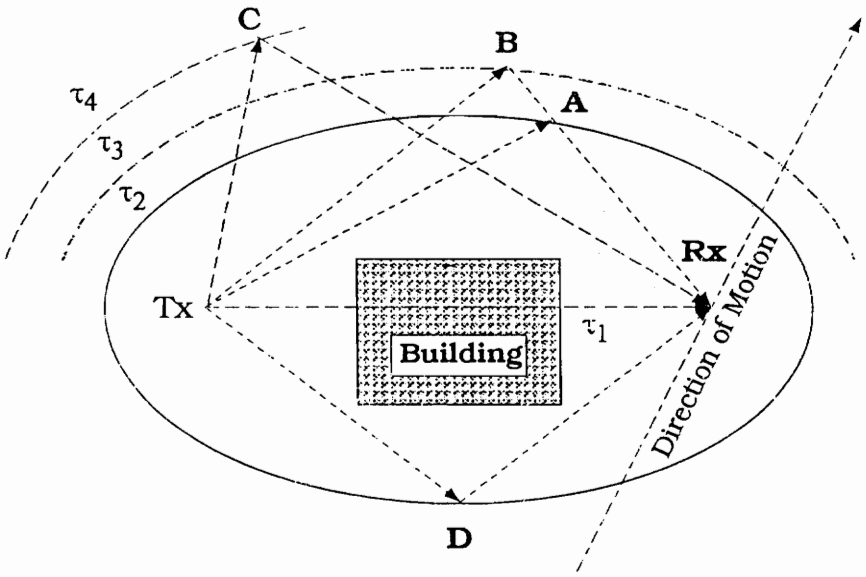


FIGURE 2.2: Two dimensional delay-multipath model

Each signal has a signal strength and phase which depend on time delays due to path length and polarization shifts due to reflections, scattering and/or diffraction. All simultaneously arriving multipath components add up vectorially at the receiver. Since each of the components will have a different amplitude and relative phase, these components can add up constructively or destructively. This variation is most prevalent as the receiver moves through the channel, resulting in a spatially varying signal strength. The angle of reception in relation to the direction of motion of the mobile unit is directly related to the doppler frequency of each multipath component. The constellation given in Figure 2.2, where a building blocks the line of sight and several other objects act as scatterers, can be

seen as a possible setup giving a power delay profile very similar to the one shown in Figure 2.1. The two inner ellipses model reflections resulting in the strong signal peaks at τ_2 and at τ_3. The weaker first component at τ_1 can be considered an attenuated line of sight signal.

2.1 Fading Properties of Narrowband Mobile Radio Channels

As mentioned before, the key condition for a mobile radio channel to be called a narrowband channel is the fact that the propagation delay differences between different signals are negligible with respect to the data symbol duration. For a narrowband signal, the received power is the phasor sum of the individual multipath components. No time delay information is available. The received power maxima, which is the sum of all arriving waves, has a period of half a wavelength. As a result, fades are observed over distances of about every half wavelength. This spatial fading, which occurs within a few wavelengths of traveled distance by the mobile, is called fast fading or small scale fading. Figure 2.3 shows a typical fast fading envelope representing a travel distance of about 50 wavelengths.

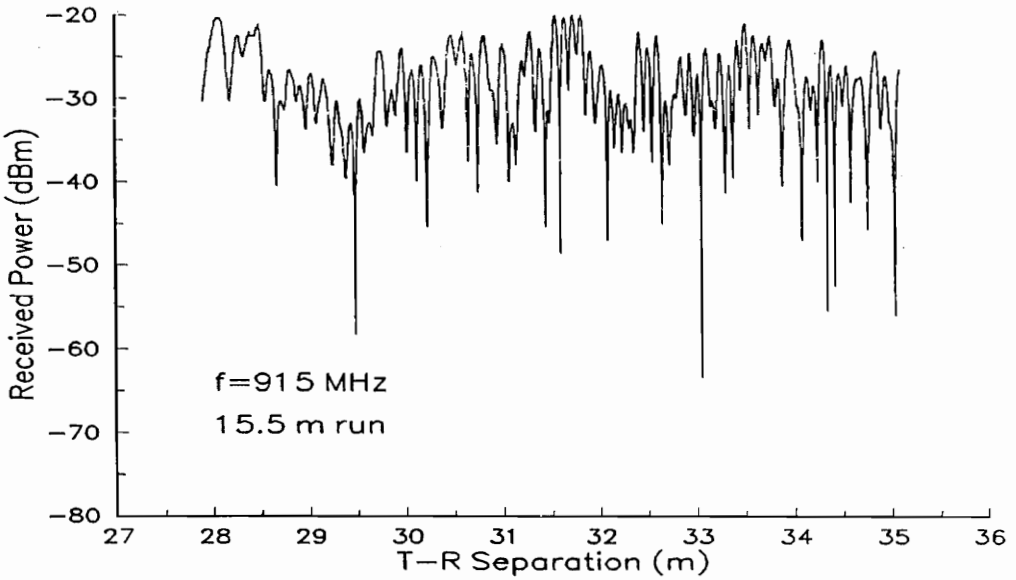


FIGURE 2.3: Typical Rayleigh fading signal power envelope

For urban areas without a line of sight path, the statistics of such a fading envelope usually follow a Rayleigh distribution [Jak74], whereas for less heavily built up areas or under

line of sight conditions, deep fades are less frequent and the PDF's of a received signal envelopes are more similar to Ricean or log-normal distributions [Par92][Rap89].

Another parameter that is highly dependent on the surroundings is shadowing. Shadowing is also known as slow fading or large scale fading. These fades usually have a period of several wavelengths. As mentioned before, slow fading originates from the blockage of significant signal components by obstacles. It has been found that slow fading is commonly log-normally distributed and can be seen as a longer-term variation of the local mean [Jak74] [Par92]. For the narrowband mobile radio channel, fading mechanisms are usually called flat fading because all frequency components are affected similarly.

2.1.1 Rayleigh Fading

Doppler shift is related to the spatial angle of arrival for a signal (due to the differential phase change of the signal) at the receiver as well as from the velocity of the mobile transmitter or receiver. Considering again the two-dimensional case and implying that only horizontally arriving waves contribute significantly to the arriving signal, the frequency deviation f_i of any arriving signal component from the carrier frequency f_c can be determined based on the angle of incidence α_i and the velocity of the mobile unit v.

$$f_i = \frac{f_c \times v}{c} \cos\alpha_i = \frac{v}{\lambda} \cos\alpha_i \tag{2.4}$$

The angle of incidence represents the angle between direction of motion and the arriving signal (Figure 2.2) and c is the velocity of light ($2.988 * 10^8$ m/s).

For N signals arriving at the receiving point simultaneously, the resulting field $E(t)$ can be expressed as the sum of all field components $E_n(t)$ [Cla68]

$$E(t) = \sum_n E_n(t) \tag{2.5}$$

The signal can be considered as a band pass signal with a carrier frequency f_c and represented in quadrature form such that

$$E(t) = \sum_n A_n \cos(2\pi f_n t + \phi_n) \cos 2\pi f_c t + \sum_n A_n \sin(2\pi f_n t + \phi_n) \sin 2\pi f_c t \tag{2.6}$$

which is equivalent to

$$E(t) = I(t) \cos 2\pi f_c t + Q(t) \sin 2\pi f_c t \tag{2.7}$$

If n is sufficiently large enough, the lowpass signals $I(t)$ and $Q(t)$ can be modeled as independent random Gaussian processes [Cla68]. Gaussian processes can be uniquely described by their mean value and their variance σ^2. For the Gaussian processes discussed here, σ^2 is equivalent to the mean power received. The mean of the Gaussian processes was considered to be zero since omnidirectional reception with uniform phase distribution is considered. The power envelope $P(t)$ can be obtained from the in-phase $I(t)$ and quadrature $Q(t)$ field components by

$$P(t) = I(t)^2 + Q(t)^2 \tag{2.8}$$

It can be shown that the probability density function of this power envelope is the Rayleigh distribution [Ric48]. The distribution is given as

$$P_r(r) = \frac{r}{\sigma^2} exp\left(-\frac{r^2}{2\sigma^2}\right) \tag{2.9}$$

where σ^2 is the mean power and $r^2/2$ is the short term (no shadowing effects) signal power.

The power spectral density of a stochastic signal can be determined by computing the Fourier transform of the signal's autocorrelation function $a_0(t)$. It can be shown [Cla68] that the autocorrelation function for the $I(t)$ and $Q(t)$ component is given as

$$a_0(\tau) = \frac{E_0}{2} \times J_0(2\pi f_m \tau) \tag{2.10}$$

where E_0 is the electric field amplitude. J_0 denotes the zero-order Bessel function of the first kind. Taking the Fourier transform, the power spectral density of $E(t)$ becomes

$$S(f) = \begin{cases} \dfrac{E_0^2}{2\pi f_m \sqrt{1-\left(\dfrac{f}{f_m}\right)^2}} & \text{if } |f| \le f_m \\ 0 & \text{elswhere} \end{cases} \tag{2.11}$$

Figure 2.4 shows a typical normalized narrowband power spectral density computed according to equation (2.11).

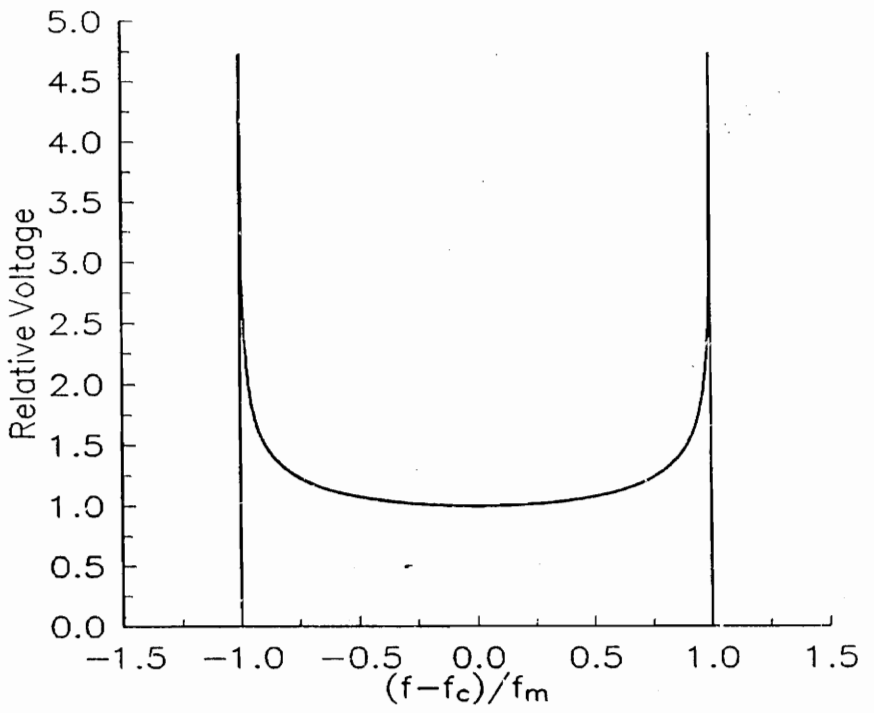


FIGURE 2.4: Power spectral density distribution of a Rayleigh faded signal

2.2 Fading Properties of the Wideband Mobile Radio Channel

The difference between a wideband and a narrowband channel is that major spectral signal components experience significantly different propagation delays with respect to the symbol duration [Par92]. As a result, the received signal is distorted. Unlike narrowband channels where the delay between all signal components is negligible with respect to the symbol duration, for wideband channels, the delayed signal components introduce echo effects. For high data rates, echoes of a symbol can arrive several symbol periods after the first arriving signal. This echo effect may cause overlapping of signals, a phenomenon which is known as intersymbol interference (ISI). The mechanism that causes this ISI is known as frequency selective fading.

Referring to Figure 2.2, the additional properties of a wideband channel can be explained. Unlike the narrowband model where almost simultaneous arrival of all signal components with respect to the symbol duration was considered, the delay between τ_1 and τ_{MAX} now becomes significant for a wideband channel since the signal is spread out over a relatively large delay. As can be seen in Figure 2.2, the paths TAR and TBR have the same angle of arrival (they will experience the same doppler shift), but have different delays with respect to symbol duration which will introduce intersymbol interference.

2.3 Summary

In this chapter, the properties of a mobile radio communication channel have been described. Frequently used terms to describe a channel have been explained. A analytical derivation for the occurrence of Rayleigh fading has been given. The reason for intersymbol interference in wideband channels has been explained.

3. The Bit Error Rate Simulator BERSIM

BERSIM consists, like any real world communication system, of three major components. Those system components are the transmitter, the channel, and the receiver. Since it is a simulation tool to predict bit error or outage rates, data is sent through the simulated system and then compared with uncorrupted reference data. Figure 3.1 shows the basic structure of the simulator.

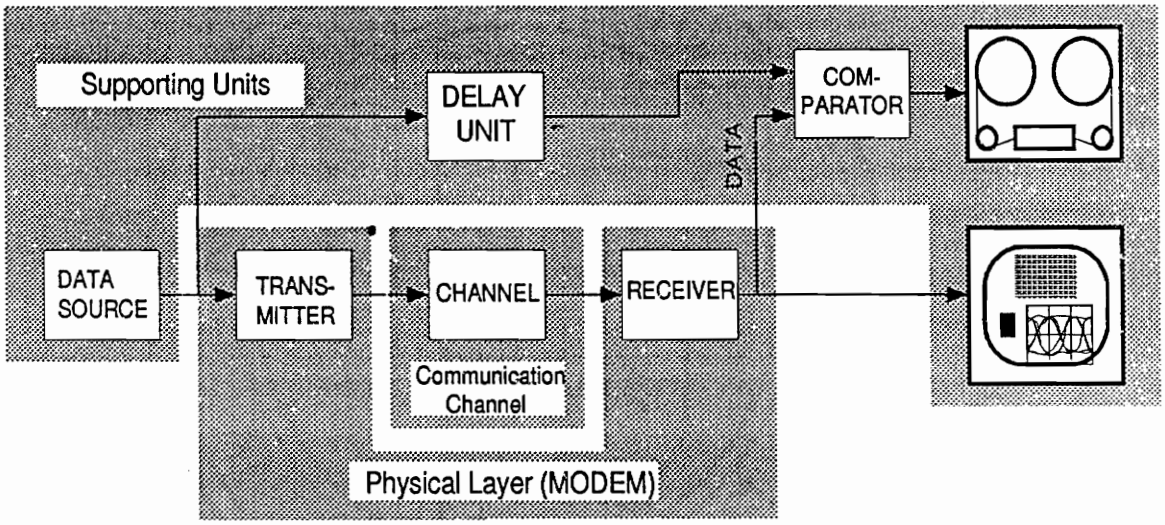


FIGURE 3.1: Basic structure of the bit error simulator BERSIM 2.0

All complex digital communication networks share common structures; therefore, it is possible to use existing network models to describe the structure of digital cellular radio and personal communication networks. Considering the recommended and well defined ISO[1] network model, BERSIM 2.0 simulates the actual channel and the physical layer (Layer 1) of a digital communication network. Below, the functional blocks of BERSIM are described based on this recommended structure.

1. International Standard Organization

The delay unit in the reference line compensates for the delay a signal experiences while traversing through the system. As described in [Fun91] [Rap91c], a lowpass representation of the signals is used to eliminate the need of simulating the high carrier frequency components. A brief description of the properties and the derivation of lowpass signals is given below. For more detailed derivations, refer to [Cou87].

3.1 Lowpass Signals

Many modulated signals can be conveniently expressed in the form

$$x(t) = a(t) \times \cos(\omega_0 t + \phi(t)) \tag{3.1}$$

where $x(t)$ is a real signal. Applying the trigonometric identity

$$\cos(\alpha + \beta) = (\cos\alpha \times \cos\beta) - (\sin\alpha \times \sin\beta) \tag{3.2}$$

and substituting $a(t)\sin\phi(t)$ with $x_s(t)$ as well as $a(t)\cos\phi(t)$ with $x_c(t)$ leads to the so called quadrature form of the signal given as

$$x(t) = x_c(t)\cos\omega_0 t - x_s(t)\sin\omega_0 t. \tag{3.3}$$

As long as the bandwidths of both $x_c(t)$ and $x_s(t)$ are less than $2\omega_o$ they are called low pass signals. The literature often refers to them as the in-phase and quadrature components of the message signal, denoted $I(t)$ and $Q(t)$ respectively.

With a lowpass signal representation, and with a system component given by its complex impulse response, one can apply standard system theory methods without considering the high frequency component (carrier frequency).

Besides describing a signal in terms of inphase and quadrature components, the complex envelope expression $\tilde{x}(t)$ of $x(t)$ is used. It is defined as

$$\tilde{x}(t) = x_c(t) + jx_s(t) = I(t) + jQ(t). \tag{3.4}$$

The ~ shall denote a signal given in complex envelope representation throughout this thesis. The complex envelope representation is used in this work to describe signal flow in block diagrams.

3.2 The Communication Channel

The term communication channel can be interpreted in two different ways. One interpretation was described in Chapter 2, considering the propagation characteristics of the mobile radio channel alone. The other, which is the more realistic approach, is to consider everything that impacts the channel as the signal travels from the transmitter to the receiver including all side effects such as attenuation, cochannel and adjacent channel interference, and impulsive noise. The transmission channel model, as seen between transmitter and receiver is shown in Figure 3.2.

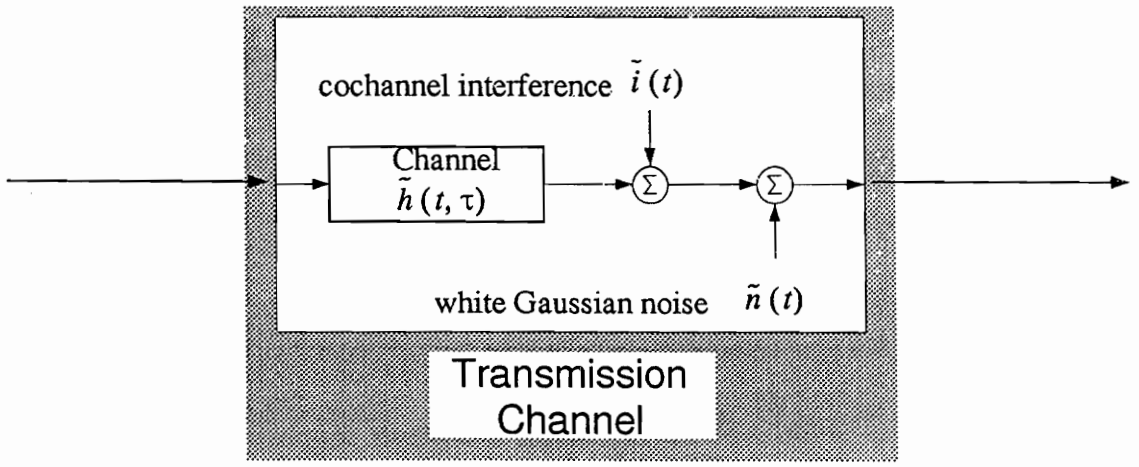


FIGURE 3.2: Implemented transmission channel model

In this model, only white Gaussian noise $\tilde{n}(t)$ and cochannel interference $\tilde{i}(t)$ are added to the signal. Note that for the model shown in Figure 3.2, all signals are given in complex

envelope representation. The order of adding the noise and interference is insignificant from a system theory standpoint.

3.2.1 The Radio Channel

BERSIM 2.0 gives the option of either simulating a one or two-ray complex Rayleigh fading channel, or importing any kind of a measured or simulated complex channel impulse response. The generation of a Rayleigh faded signal is briefly described below. The discussion here emphasizes the significant modifications that were done as part of this research. For further information regarding the Rayleigh fading envelope generation, see [Fun91][Smi75].

A detailed description of the implementation of external channel impulse responses follows the description of the internally generated channel model.

3.2.1.1 Rayleigh Fading Envelope Generation

Chapter 2 showed the relation between the lowpass signals I(t) and Q(t) and the resulting power spectrum for a narrowband fading envelope. The Rayleigh fading envelope generator realized in BERSIM was originally developed by Smith [Smi75]. In his simulation method, a fading envelope is generated from a given or generated power spectral density distribution. Figure 3.3 illustrates the method.

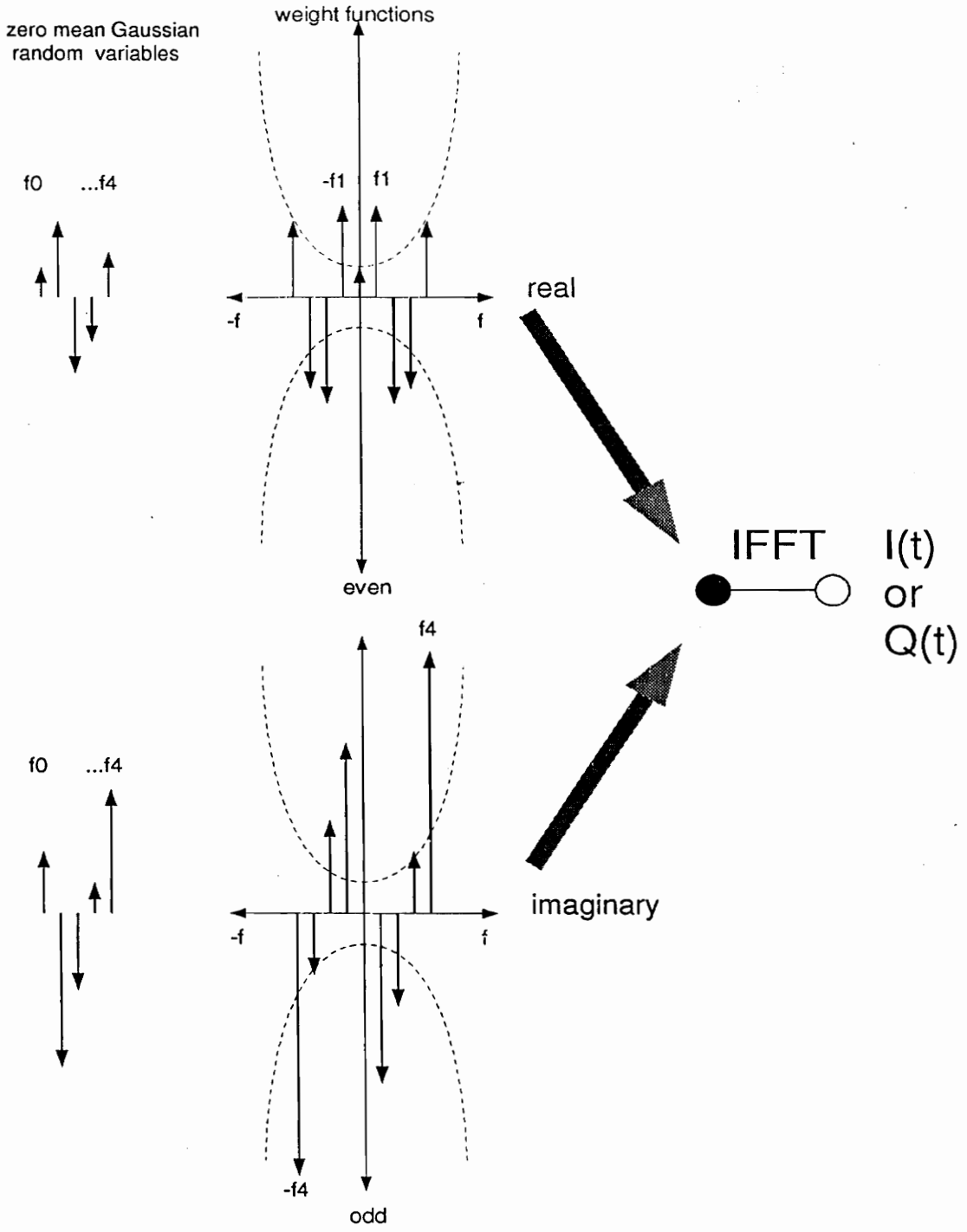


FIGURE 3.3: Generation of inphase I(t) and quadrature Q(t) components for a Rayleigh faded complex power envelope.

As seen in Figure 3.3, the Gaussian random variables are weighted according to a weighting function which approximates the power spectral density function of a Rayleigh fading power spectrum. An even (for the real part of the spectrum) and an odd (for the imaginary part of the spectrum) power spectrum function is generated in order to produce an all real signal in the time domain. This is a common property of any Fourier or inverse Fourier transformation. Two separate time domain signals are generated and used as the in-phase and quadrature components of the lowpass representation of a fading envelope. The weighting function used to weight the frequency components is represented by:

$$G(f) = \begin{cases} \dfrac{1}{\sqrt{\sqrt{1-\left(\frac{f}{f_m}\right)}}} & 0 \le f < f_m \\ \sqrt{f_m\left(\dfrac{\pi}{2} - \operatorname{atan}\dfrac{f_m - \Delta f}{\sqrt{2f_m - \Delta f}}\right)} & f = f_m \\ 0 & \text{elsewere} \end{cases} \tag{3.5}$$

where Δf is the spacing between adjacent frequency samples of the discrete representation of the power spectral density function.

A detailed study on the fading statistics produced by this simulator has been done in [Fun91]. Sample spacing in the frequency domain is directly related to the resolution in the time domain. These relations are governed by IFFT[1] and FFT[2] properties which are given as

$$\Delta f = \frac{1}{\text{time domain signal duration}} \tag{3.6}$$

1. Inverse Fast Fourier Transformation
2. Fast Fourier Transformation

and

$$\text{time domain sample spacing } \Delta t = \frac{1}{f_m}. \tag{3.7}$$

Since the number of samples representing a power spectrum is limited by the number of samples an IFFT can take, due to memory limitations, the maximum doppler shift has to be less than half the number of IFFT samples, M, times the sample spacing, Δf.

In BERSIM 1.0 the length of a fading envelope was determined by the speed of the mobile unit. According to the speed, a fixed envelope length in time was generated. BERSIM 1.0 generates fading envelopes of length 0.25 seconds for speeds between 50 and 150 km/h, 1.0 second for speeds between 10 and 50 km/h, and 10 seconds for speeds between 1 and 10 km/h. Each fading envelope has a resolution of 8092 samples (number of IFFT tabs) which are matched with 100 to 6075 samples to meet simulation constraints, which will be described later. The value of 6075 was chosen as a trade off between memory limitations, resolution, and calculation time for high symbol rates. A quarter of the data rate of 24300 (4*6075) sym/sec proposed for a future US digital cellular radio standard [EIA89] seemed to be the best choice. For some data rates different from this specific symbol rate, BER-SIM 1.0 has the tendency to choose a number of samples close to the lower bound of 100 samples. This results in very coarse envelope resolutions for these symbol rates. A worst case example is a simulation where a mobile speed of 40 km/h and a symbol rate of 10,000 symbols/sec is given, these parameters result in one fading envelope being generated for one second of time duration, but represented by only 100 samples. As a result, fades between two successive samples (spaced 10 ms apart), are lost because of poor resolution of the fading envelope.

For BERSIM 2.0, a different approach was adopted in order to achieve constant resolution for different speeds as well as to support variable carrier frequency. BERSIM 1.0 was designed around the carrier frequency of 850 MHz which is the center of the current US cellular band. BERSIM 2.0 uses the information about the actual maximum doppler shift to achieve a constant resolution over a certain range of frequencies and speeds.

According to [Par89][Jak74], the average fade duration $\tau(R)$ of a Rayleigh faded envelope is

$$\tau(R) = \frac{e^{(\rho^2)} - 1}{\rho f_m \sqrt{2\pi}} \tag{3.8}$$

where ρ is the ratio between the specified level, R, and the rms amplitude of the fading envelope. Knowing that the distance $s(R)$ traveled during a fade duration is

$$s(R) = v \times \tau(R), \tag{3.9}$$

where v is the vehicle velocity, one can calculate the traveled distance during a fade duration normalized by the wavelength λ as:

$$\frac{s(R)}{\lambda} = \frac{e^{(\rho^2)} - 1}{\rho\sqrt{2\pi}} \tag{3.10}$$

According to (3.10), in order to detect almost all fades at least 20 dB deep, with a average duration of 0.004λ, the sample spacing has to be less than 0.004λ. In BERSIM 2.0, the number of tabs, M_{IFFT}, for the IFFT is also limited to 8092 due to memory constraints. Therefore, for 8092 samples per envelope, the maximum envelope length is $8092*0.004\lambda$, which is about 32λ. To stay below this threshold, BERSIM 2.0 generates a fading envelope which is always 24λ long. The number of samples per envelope for the actual simulation is always between 3038 and 6075 due to a different algorithm for matching envelope

sample spacing with integer multiples of the symbol spacing as described below. Based on these considerations, the new duration of a fading envelope becomes a continuous function over various frequencies and speeds and is no longer just a constant. Note that a generated fading envelope always consists of M_{IFFT} samples spaced $\frac{24\lambda}{M_{IFFT}}$ apart, even though the number of sample points used for the simulation is lower.

3.2.1.2 Two-Ray Rayleigh Fading Model

For the two-ray Rayleigh fading model, two fading envelopes for each instant of time are generated as described in Section 3.2.1.1. Both are normalized envelopes such that the rms power level in each is set to unity. Before convolving the fading envelopes with the signal, the amplitude ratio between the two rays is adjusted according to the specified power ratio between ray one and ray two. This ratio is commonly known as C/D, or carrier-to-delayed signal ratio. Note that C/D is commonly expressed in decibels. BERSIM 2.0 also supports negative ratios to model a stronger delayed path which can be used to test timing recovery methods and to model obstructed topographies. The user-specified delay between the two simulated rays is matched to the correct sample spacing before convolution of the channel with the signal. (Figure 3.4)

Matching of the channel sample spacing with the signal sample spacing is necessary since discrete convolutions between the baseband signal and the channel impulse response are applied. The drawback of this method is that the delay can effectively only be varied in steps equivalent to the sample spacing.

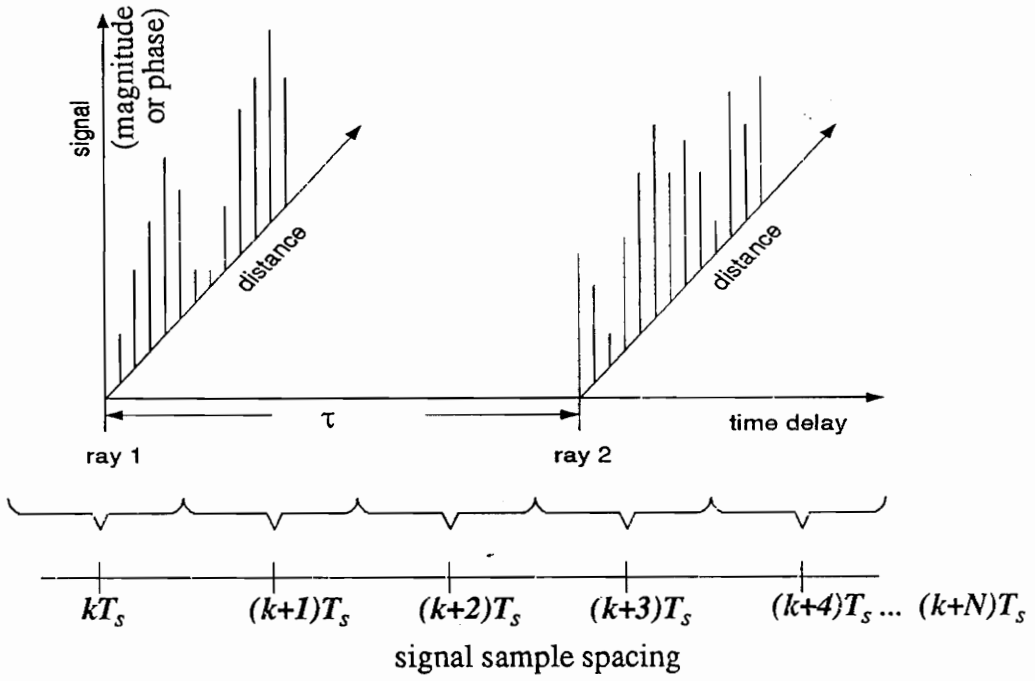


FIGURE 3.4: Matching of time delay between the rays with the discrete signal sample spacing

For a given number of samples per symbol N_s[1] and for a specified symbol rate, f_s, the minimum time increments for the delay $\Delta\tau_{MIN}$ become

$$\Delta\tau_{MIN} = \frac{1}{N_s \times f_s} \tag{3.11}$$

The Equation (3.11) shows that smaller delay variations can be achieved by increasing the number of samples per symbol, N_s. However, this would also increase the computation time, since the computation time is proportional to the number of samples per symbol.

1. in program: N_s = 13 for the implemented version of BERSIM

3.2.2 Externally Generated Channels

Unlike BERSIM 1.0, BERSIM 2.0 has been modified to support any kind of externally generated or measured channel impulse response. In order to provide this feature, a certain number of channel parameters, as well as information about the data format, must be input into the simulator. Table 3.1 lists all these parameters and their valid ranges.

TABLE 3.1: List of channel parameters that need to be provided together with an external file

Line number	Name	Meaning	Allowable values	Units
1	Data Type	Representation of complex channel impulse response. This allows future expansion.	"POLAR"	
2	Byte Order	Sets either 'low byte first' (used by PC's) or 'high byte first order' (used by Suns)	"LOW" or "HIGH"	
3	Precision	The number of bytes used to represent each floating-point value.	8 (only option presently supported)	
4	Frequency	The carrier frequency in MHz	> 0	MHz
5	Speed	The mobile speed (rate of change of path length)	> 0	Meters per second
6	Samples in profile	The number of polar-complex samples in each profile.	1-128	Samples (integer only)
7	Profile sample spacing	The time interval between samples in the profile.	1E-9 to 1E-3	Seconds
8	Number of profiles	The number of profiles in the channel file.	>= 12	Profiles (integer only)
9	Channel profile spacing	The distance (movement of the mobile) between consecutive profiles in the channel file	> 0, < 0.5	Wavelength at carrier frequency.

3.2.2.1 Dividing Large Channel Impulse Response Files into Sub-Channel Impulse Response Files.

Similar to the two-ray internally generated channel model, only a certain channel length (wavelength) can be supported. If an external channel is longer than the supported channel size, it has to be divided into smaller sub-channels. The bounds for a valid channel are

given by the needed resolution as well as memory limitations and execution speed considerations.

Based on the number of impulse responses per file and on spatial spacing between successive channel impulse responses, BERSIM 2.0 determines the optimum file size for a subfile (Figure 3.5).

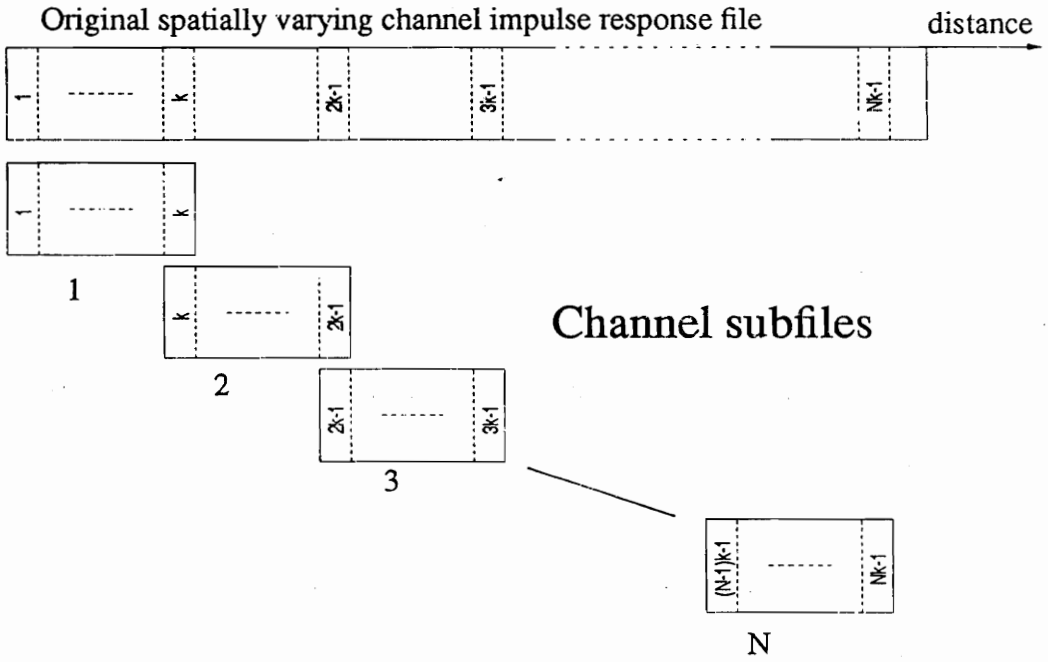


FIGURE 3.5: Principle of dividing long spatially varying impulse response files into subfiles

For later use in the actual simulation, channel files have to be all of the same file size. As one can see from Figure 3.5, each subfile starts with the last impulse response of the previous file. The maximum length of an impulse response subfile is limited to six wavelengths of travel distance. This limit was chosen to maintain a certain worst case resolution. Six wavelengths of channel information will provide exactly four times the worst case resolution than for a 24λ long channel file generated by the internal fading generator (Section 3.2.1.1). Increasing the resolution was considered reasonable since imported channels are

more likely to be used for system simulation where the mobile unit moves slowly or high data rates are simulated. A lower limit of 12 impulse responses for the length of each sub-channel was defined to limit the number of generated sub-channels. The number of generated sub-channels is directly related to computation time, since all manipulations described below are performed on each sub-channel.

3.2.2.2 Converting Sub-Channel Impulse Response Files into Simulation Format.

In order to use an imported channel impulse response file with the BERSIM 2.0 simulator, several conversions of the complex impulse response have to be performed. This conversion takes place after the external file has been approved to match the defined format for external files and after the splitting into subfiles.

Figure 3.6 shows a typical sub-channel impulse response file (power magnitude or phase) format. Let D be the number of multipath component samples and I be the number of impulse response profiles in a file. The number of multipath component samples can be considered also as the number of individual fading envelopes in a file. It can be seen that each sub-channel covers a track along which a mobile moves a distance of $(I-1)\,\Delta s_{ch}$, where Δs_{ch} is the spacing between two measured or simulated channel impulse responses.

The channel impulse response sample period $\Delta\tau_{ch}$ and the spatial spacing between adjacent channel impulse response profiles Δs_{ch} are known from information provided by the channel specification data file.

For real world systems, the number of symbols transmitted while moving along a track described by a spatial varying channel impulse response is much higher than the number of impulse responses available by currently used sounding and simulation tools.

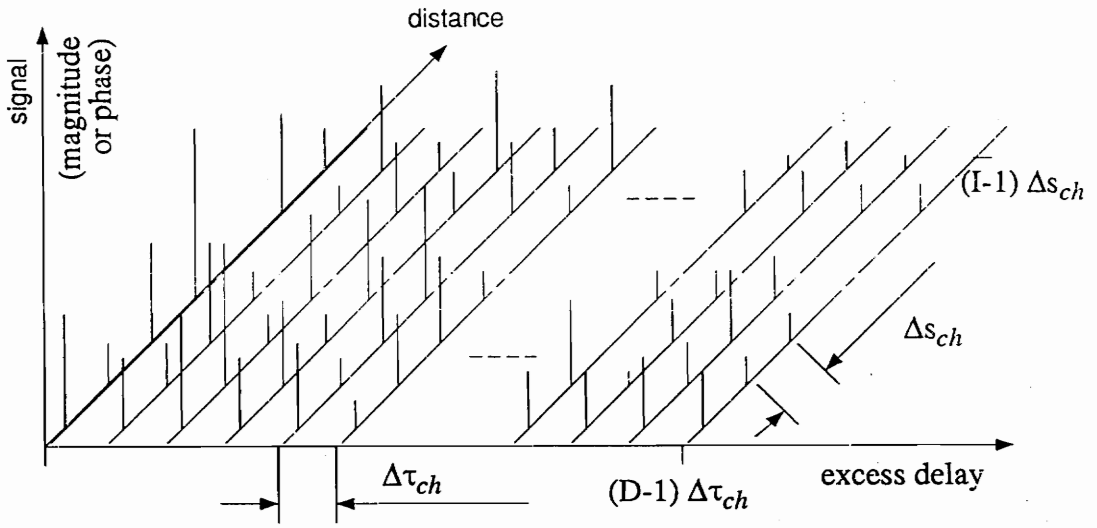


FIGURE 3.6: Format of typical sub-channel impulse response file

Since the spacing of responses is too large (e.g., a quarter wavelength with SIRCIM [Rap91]) with respect to symbol rate and velocity, interpolation techniques are used to achieve higher resolution. However, for computational convenience it is desirable to have a time invariant channel (i.e. static impulse response) for a complete symbol duration. For this reason, the transmission of a single symbol is simulated as a transmission for a fixed receiver location. For high data rates and low velocities, several symbols are transmitted at the same instantaneous location. This simplification is valid since the number of symbols per sub-channel is a function of the speed (as shown in equations (3.12) and (3.14)). As a result, several symbols are transmitted for one discrete time duration when instantaneous channel changes can be neglected [Cha87a]. The space difference between successive symbol transmission locations Δs_{symb} is a function of symbol rate and speed, and is given by

$$\Delta s_{symb} = \frac{(I-1)\,\Delta s_{ch}}{\left(\frac{\text{symbols per subchannel}}{\text{symbols per impulse response}}\right)} \tag{3.12}$$

with

$$\text{symbols per subchannel} = \frac{(I-1)\,\Delta s_{ch} \times f_s}{\text{velocity}}. \tag{3.13}$$

Equation (3.13) substituted into equation (3.12) results in

$$\Delta s_{symb} = \frac{\text{velocity} \times \text{symbols per impulse response}}{f_s} \tag{3.14}$$

To obtain the corresponding channel impulse response for a particular symbol transmission location, cubic spline interpolation is applied on each of the D fading envelopes. The fading envelopes shown in Figure 3.6 represent conventional log scaled envelopes. However, the cubic spline is done on a linear unscaled envelope. Each fading envelope is then sampled at the determined symbol transmission locations s_{symb}.

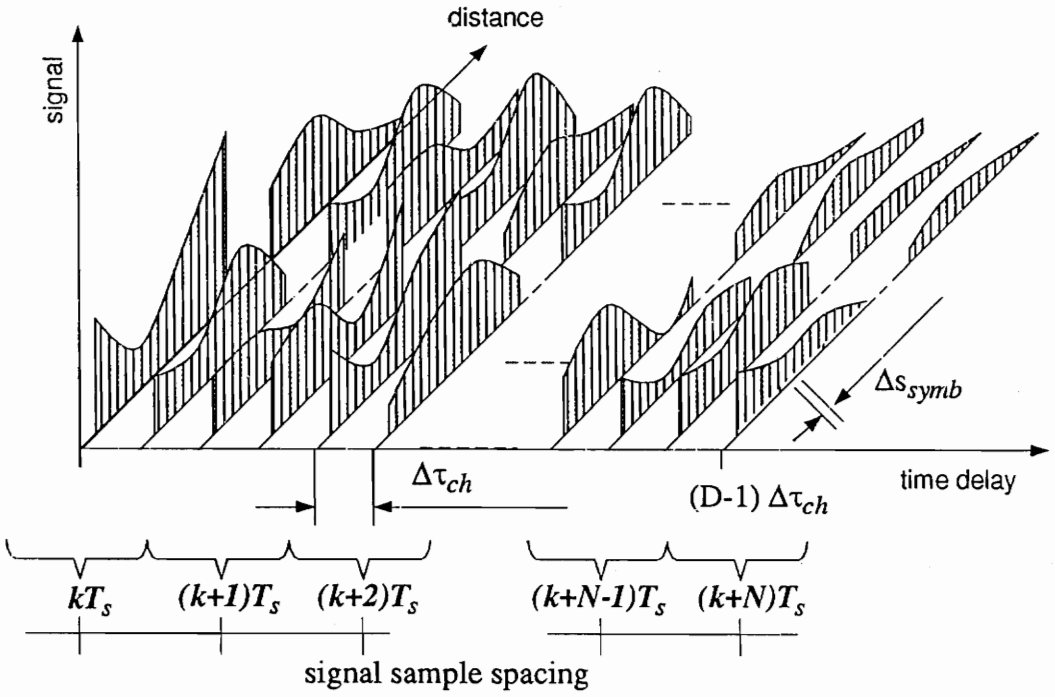


FIGURE 3.7: Format of typical sub-channel impulse response file after applying cubic spline and sampling at the symbol transmission locations

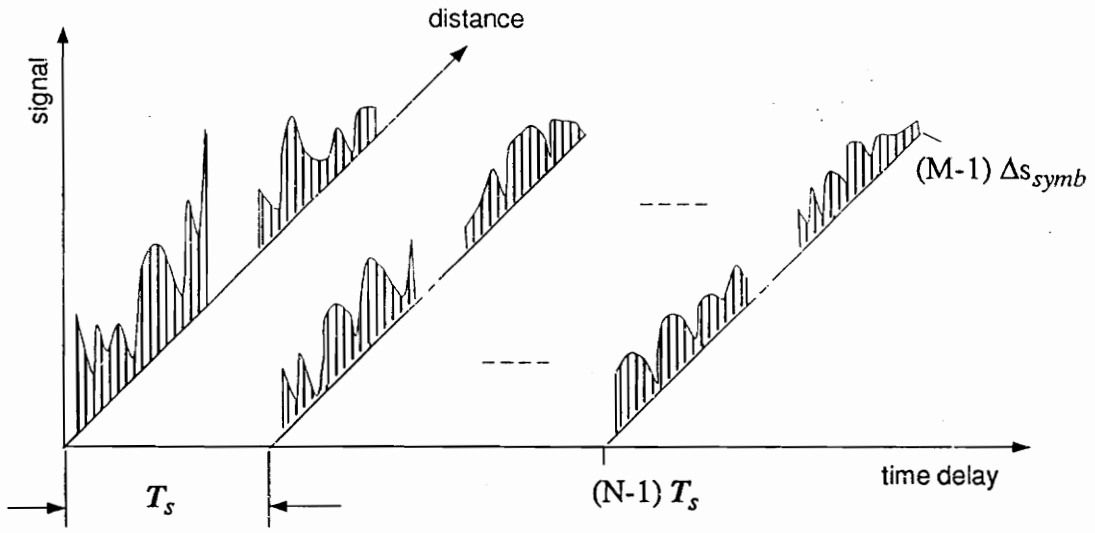


FIGURE 3.8: Format of typical sub-channel impulse response file after matching to the discrete signal sample spacing

After this procedure, a series of channel impulse responses with correct spatial spacing are available. In the next step, the channel impulse response sample period is matched to the symbol sampling period T_S. Therefore, all channel delay profile samples that are within the range of half a symbol duration from a symbol sample point are added using complex addition (Figure 3.7). The resulting channel impulse response is given as

$$h(s, kT_s) = \sum_{d \geq \frac{T_s}{2}}^{d < \frac{T_s}{2}} h_{sub}(kT_s + d) \tag{3.15}$$

BERSIM 1.0 performed similar impulse response manipulations on imported SIRCIM channels. The key difference for BERSIM 2.0 is the order in which the manipulations are performed. BERSIM 1.0 first performed the symbol sampling matching procedure before applying spline interpolation. As a result, deep magnitude fades and rapid phase changes were filtered out for wideband as well as for narrow band signals. This is due to the fact

that a phasor summation of the multipath components before applying spline interpolation results in a loss of channel information. Figure 3.9 shows two typical narrowband fading envelopes generated by BERSIM1.0 and BERSIM2.0. Both were generated using the same wideband channel impulse response generated by the indoor channel simulator SIR-CIM. Besides the significant difference of the fading envelope shapes, it was found that the mean of a fading envelope generated by BERSIM1.0 was off by several dB. For the fading envelopes shown in Figure 3.9 a study of the envelope statistics gave a mean of -2 dB for the envelope generated with BERSIM1.0, where as for the other envelope generated by BERSIM2.0 the mean is equivalent to the desired 0 dB.

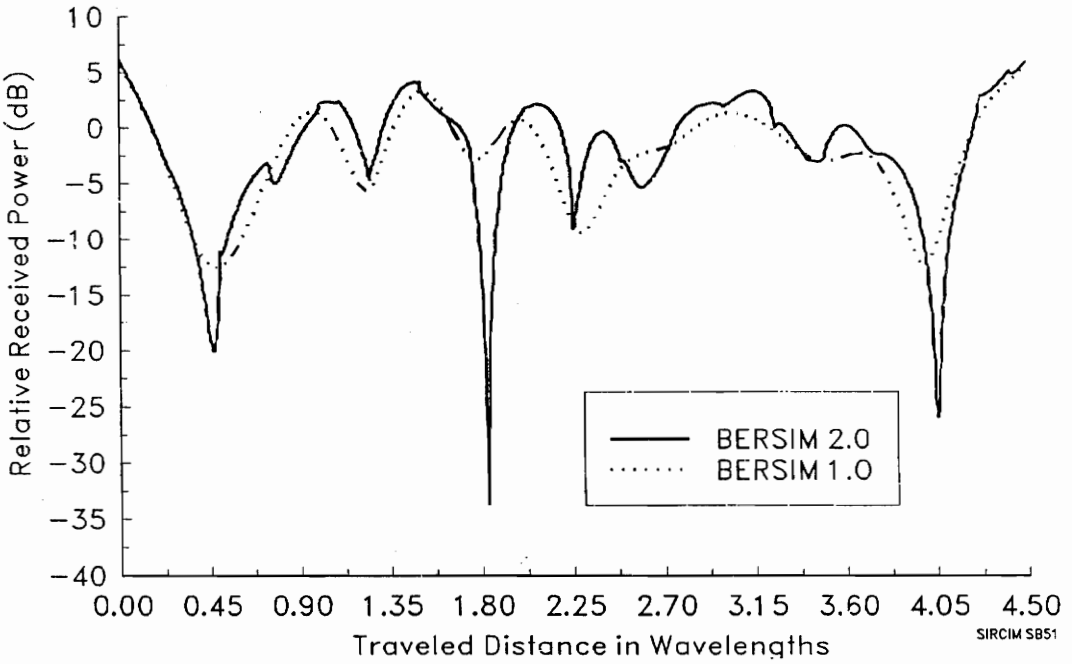


FIGURE 3.9: Narrowband fading envelope generated from a wideband channel impulse response, using BERSIM 1.0 and BERSIM 2.0 channel impulse response conversion method.

The simulation section of this thesis (Chapter 5) shows the significant difference of these two methods in terms of bit error rate estimation.

3.2.2.3 The Unity Gain Channel

A unity gain channel is a channel that has no average attenuation or amplification impact on the transmitted signal. It can be also described as a channel where the average power that is transmitted is equivalent to the received average power even though the received power varies with time. Unlike BERSIM 1.0, BERSIM 2.0 determines exactly the number of impulse responses, N, needed for a particular simulation. All power magnitudes of the channel impulse responses are added up and divided by their number. The channel gain factor G_{ch} is calculated as

$$G_{ch} = \sqrt{\frac{N \times M}{\sum_{N}\sum_{M} P(N,M)}} \tag{3.16}$$

where M is the number of discrete multipath components in a single impulse response and P(N,M) is the power magnitude of a multipath component. Each signal magnitude component of all channel impulse response components is then multiplied by the channel gain factor to convert the current, non-normalized, channel to a unity gain channel.

3.2.3 Adding White Gaussian Noise

The mean signal level coming out of the transmitter is always constant and for convenience, is normalized to unity. For this reason, the amount of noise that gets added is used to simulate both attenuation of the channel as well as variation of the transmitter signal level. For example, a low signal-to-noise ratio at the receiver can be caused by two reasons. One reason is that the transmitted signal power was rather low. The other reason could be large transmitter to receiver separation resulting in a low signal to noise ratio due to high path loss the signal experiences. The noise source is considered to have a uniform distribution over all frequencies (white noise). The amplitude distribution of the added

noise is Gaussian with zero mean such as any thermal noise process. The standard deviation σ at the distribution is varied depending on the simulated energy-per-bit to noise spectral density (E_b/N_0) ratio. An exact derivation of the relevant noise bandwidth and the resulting variance σ^2 for the noise distribution is given in [Fun91].

3.2.4 Adding Cochannel Interference

A cochannel interference signal $\tilde{i}(t)$ is added according to the user specified signal to cochannel interference ratio. In order to have a realistic interference signal, a cochannel interference signal is generated in a separate, preceding run of the simulator. With this simulation prior to the desired simulation, a cochannel interference signal is recorded, which is later added on to the desired signal during the actual simulation (Figure 3.10). Thus, cochannel interference is modeled as interference that would exist in a real world channel. The same simulation parameters such as modulation, frequency, data rate, roll-off factor, velocity, and delay for the two-ray case are applied.

For this separate run, a random data stream is modulated and pulse shaped with a scaled square root raised cosine filter, before it gets sent through a different channel. The scaling of the filter is done according to the desired cochannel interference ratio.

For a M-tab digital filter and a cochannel interference ratio given in dB, the scaled filter coefficients $\tilde{h}(m)$ are given by equation (3.17) for all $m \geq 0$ and $m < M$, where M represents the number of filter tabs. The variables $h_c(m)$ represent the filter impulse response magnitudes of the filter used on the desired signal. The scaled filter tabs $h_i(m)$ for the interference signal are calculated as

$$h_i(m) = \sqrt{10^{-\frac{C/I}{10}}} \times h_c(m) \tag{3.17}$$

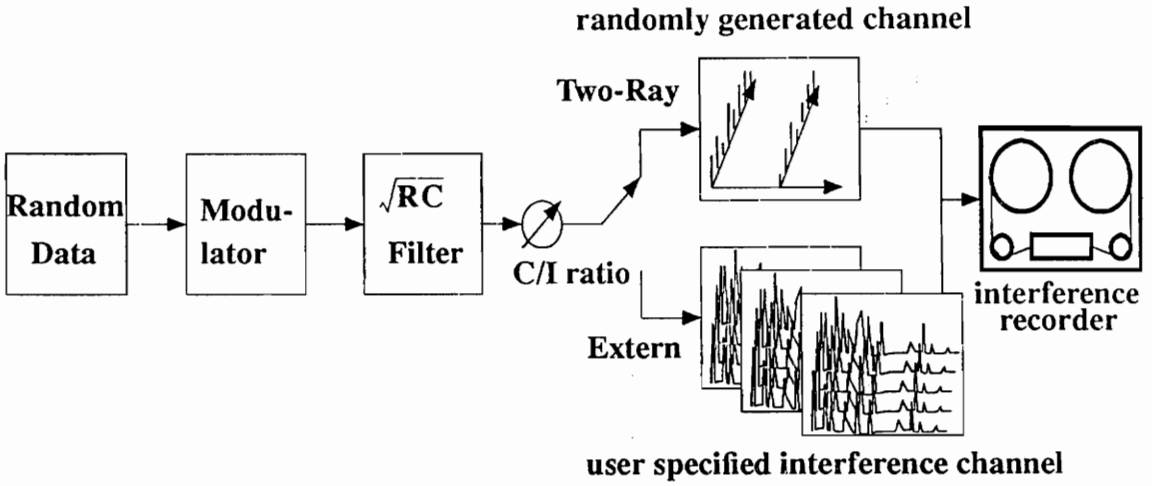


FIGURE 3.10: BERSIM cochannel interference generation

For two-ray Rayleigh or flat fading Rayleigh fading simulations, the interference channel is always a new internally generated Rayleigh fading channel. Whenever external channel impulse response files are imported, a user selectable channel is used to serve as the cochannel. The resulting signal $\tilde{i}(t)$ gets recorded. During the actual simulation, this signal is added to the desired signal $\tilde{c}(t)$. A constant random time shift, rT_s, ($0 < r < N_s$, where N_s = number of samples per symbol) is added to insert an additional random delay into the interfering random data stream. The resulting interfered signal $\tilde{c}_i$ is

$$\tilde{c}_i(t) = \begin{cases} c_c(kT_s) + i_c(kT_s + rT_s) \\ +j(c_s(kT_s) + i_s(kT_s + rT_s)) \end{cases} \tag{3.18}$$

The delay shift prevents general synchronism of the desired and the interfering signal. Thus the interfering signal is only for one simulation out of N_s simulations synchronized on the average.

3.3 The Physical Layer (Modem)

According to the ISO-definition, the physical layer encompasses everything that is concerned with bit transmission and recovery. This includes the modulation scheme used as well as filtering etc. BERSIM 2.0 supports three modulation/demodulation methods that incorporate variable raised cosine filters for symbol shaping, timing recovery units, sample and hold units, and threshold compare decision units (Figure 3.11, Figure 3.12, Figure 3.13). In this section, the implementations of different modulation techniques are explained first, followed by a discussion of the units which are common to all or nearly all of the different physical layer components.

3.3.1 π/4 DQPSK Modulator/Demodulator

The implemented π/4 DQPSK modulator/demodulator or modem is the baseband realization of a quadrature differential modulator and of an IF differential detector [Liu89] [Anv91] [Rap91c]. Figure 3.11 shows the block diagram of the complete implemented π/4- DQPSK system.

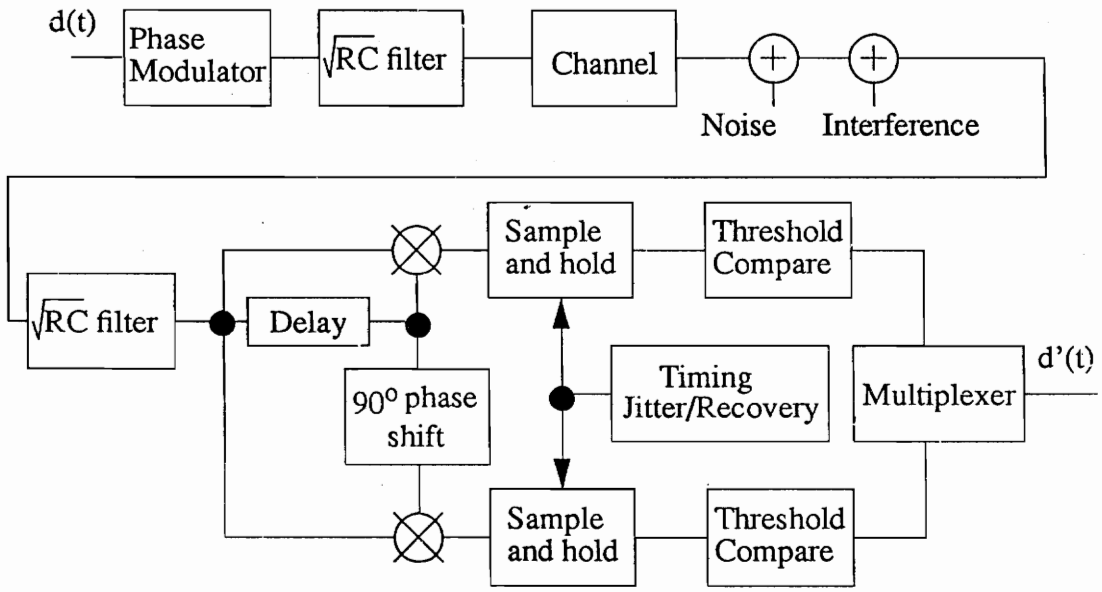


FIGURE 3.11: Block diagram of the π/4 DQPSK system

As for any differential phase modulation technique, the phase difference between the current and the previous symbol is associated with a particular information word. For quadrature modulation techniques, such a word carries 2 bits. Consequently, four different words (00, 01, 10, 11) are represented by different phase shifts [Pro89]. The words and their corresponding phase shifts, used in BERSIM, are listed in Table 3.2 .

TABLE 3.2: List of words a symbol can represent and their corresponding phase shifts.

Word	Phase shift
00	$\pi/4$
01	$3\pi/4$
10	$5\pi/4$
11	$7\pi/4$

At initialization and at the beginning of each new channel file, the phase of the baseband signal, ϕ, is arbitrarily set to $\pi/4$. For each succeeding bit pair, the corresponding phase difference is then added. The resulting signal for the nth symbol at the output of the modulator can be described as follows [Fun91] [Rap91c].

$$x(t) = \begin{cases} x_c(t) = \sum_n A\cos(\phi_n)\,\delta(t-nT) & I(t) \\ x_s(t) = \sum_n A\sin(\phi_n)\,\delta(t-nT) & Q(t) \end{cases} \tag{3.19}$$

where T represents a symbol duration. The amplitude, A, is set to unity. At this point, both baseband signals are a sequence of delta functions spaced by one symbol duration. Based on the sampling period T_s for each symbol, the discrete signal description for the k th sample becomes

$$x(kT_s) = \begin{cases} x_c(kT_s) = \sum_{k=0,N_s,2N_s,\ldots} A\cos(\phi_k)\,\delta(t-kT_s) & I(kT_s) \\ x_s(kT_s) = \sum_{k=0,N_s,2N_s,\ldots} A\sin(\phi_k)\,\delta(t-kT_s) & Q(kT_s) \end{cases} \tag{3.20}$$

where N_s denotes the number of samples per symbol. The symbols get their real shape after convolution with the square root raised cosine filter. At the receiver side, the signal is again convolved with a square root raised cosine filter to realize a matched filter receiver that maximizes the instantaneous signal power [Cou90].

In order to determine the phase shift between two successive symbols, the signal is buffered for one symbol duration in the delay unit. To model the IF differential decoder, the complex conjugate of the buffered signal, denoted as

$$r^*[k-N_s] = \begin{cases} r_r[k-N_s] & I(kT_s) \\ -r_s[k-N_s] & Q(kT_s) \end{cases} \tag{3.21}$$

is multiplied by the not delayed signal, $\tilde{r}[k]$. Two decoder branches are used to decode the first and the second bit of a symbol separately. To decode the second bit in a symbol the signal $\tilde{r}^*[k-1]$ is phase shifted by 90°. Both delayed signals are then multiplied by the current signal such that the signal at the input of the sample and hold unit, for the first (even) bit, becomes

$$y_{even}[k] = Re(\tilde{r}[k] \times \tilde{r}^*[k-N_s]) \tag{3.22}$$

and the signal for the second (odd) bit, becomes

$$y_{odd}[k] = Re(\tilde{r}[k] \times \tilde{r}^*_{90}[k-N]) \tag{3.23}$$

After sampling at a determined sampling instant and after the following threshold compare, a multiplexer puts both bits into the correct order. The order of the received output bit stream $d'(t)$ is determined by the relation (3.24) [Fun92].

$$d'(t) = \sum_{l=0,2,4,\ldots} R_l \mathrm{rect}\left(\frac{t - lT_b - \frac{T_b}{2}}{T_b}\right) + R_{l+1}\mathrm{rect}\left(\frac{t-(l+1)T_b - \frac{T_b}{2}}{T_b}\right) \tag{3.24}$$

where l represents the bit index and rect() denotes the rectangular pulse function of bit duration T_b. It can be easily seen that T_b has to be $T_{sym}/2$. Further note that R_l and R_{l+1} are either integer values 1 or 0, depending on the output of the threshold compare unit.

3.3.2 BPSK Modulator/Demodulator

Figure 3.12 shows a block diagram of the BPSK system implemented in BERSIM 2.0.

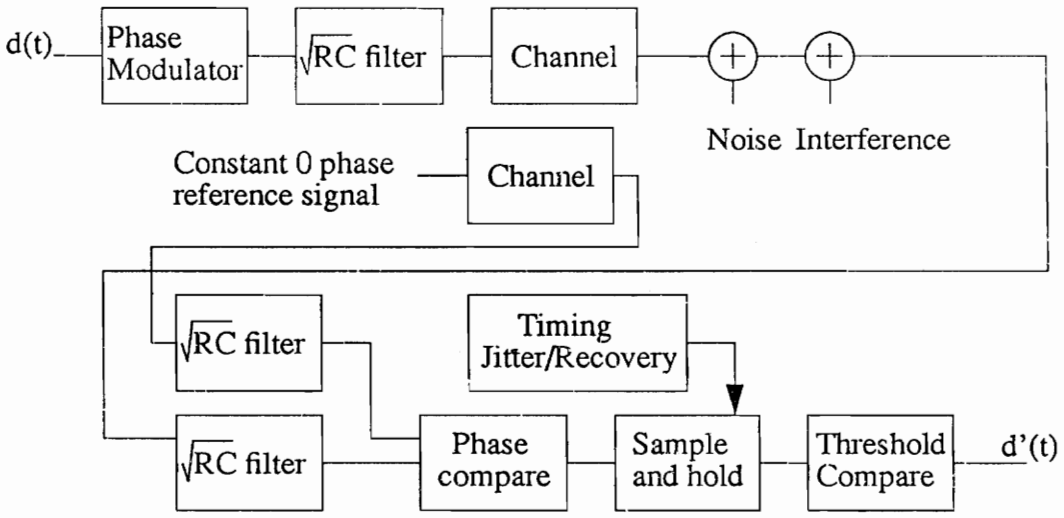


FIGURE 3.12: Block diagram of the BPSK system

Since the BPSK system in BERSIM 1.0 only supported flat fading channels, a new BPSK receiver was implemented for BERSIM 2.0. This new receiver models a coherent detection like the narrowband receiver implemented in BERSIM 1.0. Perfect phase recovery is assumed and is achieved by sending an information-free constant phase reference signal along with the actual signal.

The reference signal is given as

$$x_{ref}(t) = \begin{cases} x_{cref}(t) = \sum_n A(\delta(t-nT)) & I(t) \\ x_{sref}(t) = 0 & Q(t) \end{cases} \tag{3.25}$$

which indicates that the reference phase was set to zero. The variable A, can be any non-zero number, however, it is considered here to be one. Given the reference phase of 0^0 and considering a phase shift of 180^0 for optimum performance, the desired signal at the modulator output becomes an all real signal with the polarity depending on the information. The equations below describe the signal $x(t)$ for transmitted "high" and "low" bits.

$$x_1(t) = \begin{cases} x_c(t) = \sum_n A\delta(t-nT) & I(t) \\ x_s(t) = 0 & Q(t) \end{cases} \tag{3.26}$$

$$x_0(t) = \begin{cases} x_c(t) = \sum_n (-A)\delta(t-nT) & I(t) \\ x_s(t) = 0 & Q(t) \end{cases} \tag{3.27}$$

The reference signal experiences the same phase shifts by the channel as the desired signal. No noise or cochannel interference is added to the reference signal. Therefore, at the input of the matched filter, the discrete representations of both signals are given as

$$\tilde{y}(kT) = \tilde{x}(kT_s) * \tilde{h}(kT_s) \tag{3.28}$$

and

$$\tilde{y}_{ref}(kT) = \tilde{x}_{ref}(kT_s) * \tilde{h}(kT_s) \tag{3.29}$$

respectively.

The signals $y(kT_s)$ and $y_{ref}(kT_s)$ get convolved with the matched filter. The reference phase is given by

$$\phi_{ref}(kT_s) = \frac{r_{sref}(kT_s)}{r_{cref}(kT_s)} \tag{3.30}$$

where $r_{sref}(kT_s)$ denotes the inphase part and $r_{cref}(kTs)$ denotes the quadrature part of the signal after the convolution with the matched filter. The desired signal gets shifted by the reference phase such that the shifted signal $y_{ps}(kT_s)$ becomes

$$y_{ps}(kT_s) = \begin{cases} y_{cps}(kT_s) & I(kT_s) \\ y_{sps}(kT_s) & Q(kT_s) \end{cases} \tag{3.31}$$

$$y_{cps}(kT_s) = r_c(kT_s)\cos\phi_{ref}(kT_s) + r_s(kT_s)\sin\phi_{ref}(kT_s) \tag{3.32}$$

$$y_{sps}(kT_s) = -r_c(kT_s)\sin\phi_{ref}(kT_s) + r_s(kT_s)\cos\phi_{ref}(kT_s). \tag{3.33}$$

The in-phase component passes into the sample and hold unit where it is sampled at the time instant supplied by the timing unit. The sampled signal is then compared with the threshold. The resulting bit stream can be given by [Fun91].

$$d'(t) = \sum_n R'_n \mathrm{rect}\left(\frac{t - nT - \frac{T}{2}}{T}\right) \tag{3.34}$$

where

$$R'_n = \begin{cases} 0 & y_{cps}(kT_s) > 0 \\ 1 & y_{sps}(kT_s) \le 0 \end{cases} \tag{3.35}$$

and

$$k = nN + (N-1) + \Delta k. \tag{3.36}$$

The function rect() denotes the rectangular pulse function of a symbol duration T_{sym} and the parameter Δk represents the deviation of the sampling point introduced by timing jitter or timing recovery.

3.3.3 FSK Modulator/Demodulator

The FSK system shown in Figure 3.13 represents the implementation of the non-coherent binary FSK system.

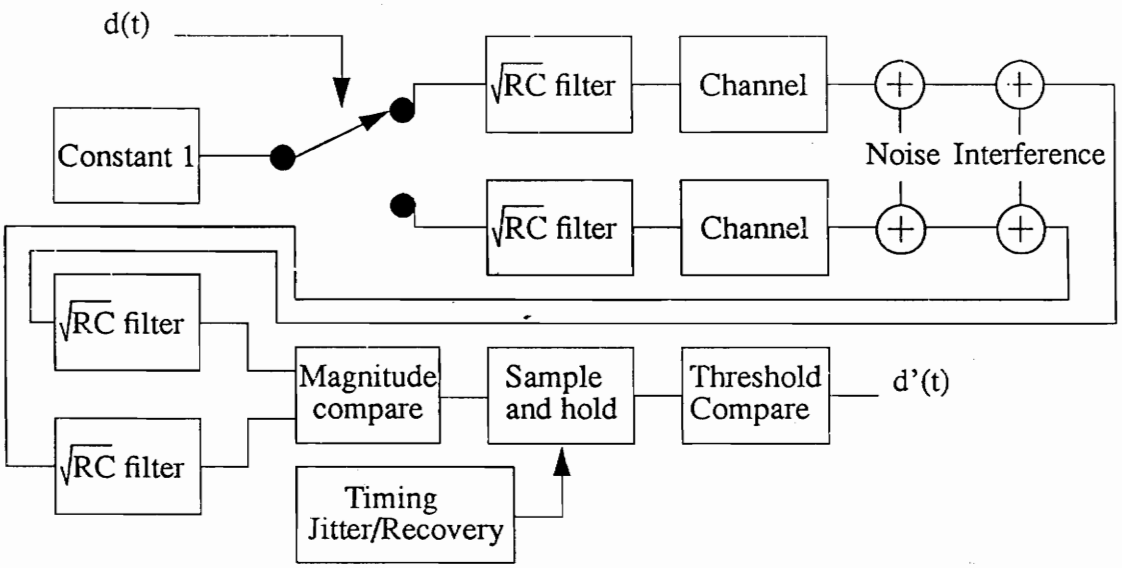


FIGURE 3.13: Block diagram of the FSK system

The system works like two non-coherent On-Off Keying (OOK) systems working with complementary data on different frequencies. The frequencies f_1 and f_2 are considered far enough apart such that no interaction between them exists. On the other hand, they are considered to be close enough such that the channel properties can be considered the same for both frequencies. Before filtering with the square root raised cosine filter, the signal can be represented as two baseband signals ($x_{f1}(t)$ and $x_{f2}(t)$), each given as a sequence of real delta functions spaced by the symbol duration such as

$$x_{f_1}(t) = \begin{cases} x_{f1_C}(t) = \sum_n AR_1 \delta(t-nT) & I(t) \\ x_{t1_S}(t) = 0 & Q(t) \end{cases} \tag{3.37}$$

$$x_{f_2}(t) = \begin{cases} x_{f2_C}(t) = \sum_n AR_2 \delta(t-nT) & I(t) \\ x_{t2_S}(t) = 0 & Q(t) \end{cases} \tag{3.38}$$

The possible values for the parameter R_x are listed in the table below.

TABLE 3.3: Binary words and the corresponding R_x values.

word	R_1	R_2
0	0	1
1	1	0

Similar to the implemented BPSK system, a complex convolution is performed with the channel. At the receiver, envelope detectors are used to detect the envelope of each on-off keying branch. Therefore, the magnitude of each signal is used as these model the output of envelope detectors. The resulting signals are subtracted from each other and the difference is passed on to the sample and hold unit. After holding the signal, a threshold and compare unit rebuilds the transmitted bit stream.

3.3.4 Raised Cosine Filter

Shaping the waveform prior to transmission is a common technique to compress the spectrum occupied by the signal. A smaller signal bandwidth also reduces the sensitivity of a wideband signal to frequency selective fading. BERSIM uses raised cosine filters to shape the transmitted waveform. Each filter is split into two square root raised cosine filters, one at the transmitter side and one at the receiver side. With this method, pulse shaping and matched filtering at the receiver side are achieved. An important descriptor for a square root raised cosine filter is the roll-off-factor, α. Figure 3.14 shows a typical normalized

magnitude frequency response of a square root raised cosine filter for different roll-off-factors.

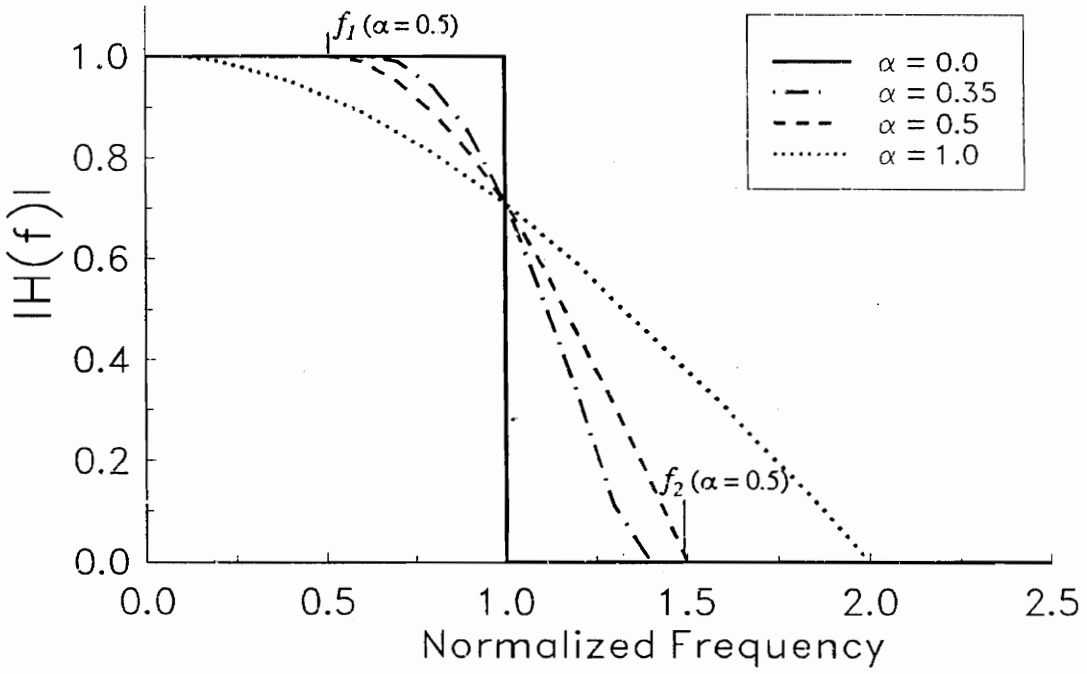


FIGURE 3.14: Transmission function of a square root raised cosine filter for different roll-off-factors.

As can be seen in the figure, the roll-off-factor is defined as

$$\alpha = \frac{f_2 - f_1}{f_2 + f_1} \tag{3.39}$$

where f_2 and f_1 are the frequencies between which the spectrum is shaped. For a detailed description of how a square root raised cosine filter wave form in the time domain is generated such that it can be used in BERSIM, refer to Victor Fung's thesis [Fun91] as no major modifications were made.

3.3.5 Timing Recovery

Timing recovery is implemented in real world systems wherever the timing for the data symbol clock needs to be recovered and the optimum sampling instant for the sample and hold unit needs to be determined. Since BERSIM is a software simulation tool whose transmitter and receiver are implemented within one subprogram, a timing recovery loop in order to recover the symbol clock at the receiver side is not essential. This is because the transmitter and receiver are synchronized automatically due to the nature of the program. Nevertheless, timing recovery becomes very important if channels with delay (Figure 3.15) or frequency selective fading channels (Figure 3.16) are used.

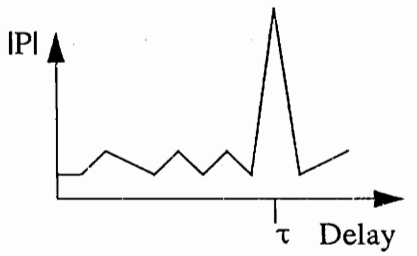


FIGURE 3.15: Single channel impulse response of a channel with delay τ.

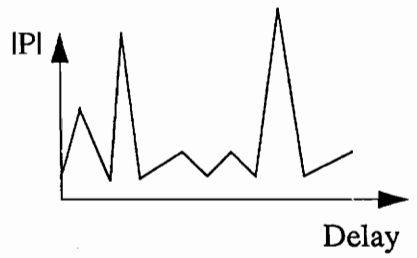


FIGURE 3.16: Single channel impulse response of a channel with significant delay spread.

BERSIM 1.0 did not apply any kind of sampling timing adjustment at the receiver side. As a result, it always sampled at the fixed sampling point for an ideal delay free channel. Such a channel is shown in Figure 3.17.

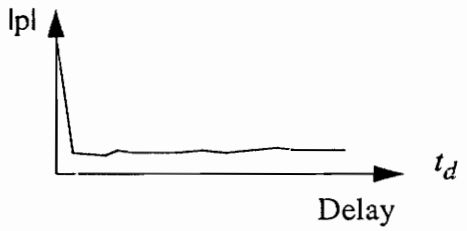


FIGURE 3.17: Single channel impulse response for an ideal delay free channel.

For this work, three additional methods for determining the sampling point were implemented and tested. One of them was the approach of modeling an ideal sampling unit, whereas the two other methods model a squaring bit synchronizer as described in [Hol80]. For results and comparison of these techniques, refer Chapter 5. Figures 3.18, 3.20, and 3.21 try to show the differences among the currently implemented sample point determination methods (timing recovery models).

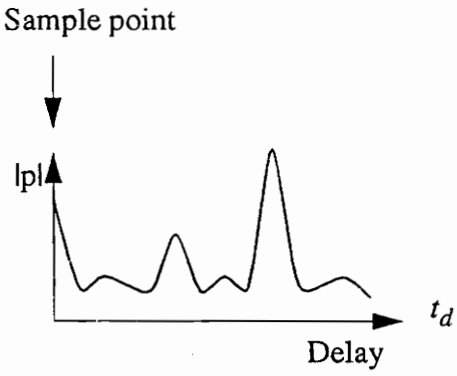


FIGURE 3.18: Sampling location as always given in BERSIM 1.0 and optional in BERSIM 2.0.

3.3.5.1 Squaring Bit Synchronizer Model

The squaring bit synchronizer is a timing recovery circuit that can be used for coherent as well as for non-coherent receiver implementations. Figure 3.19 shows the typical structure for such a squaring bit synchronizer.

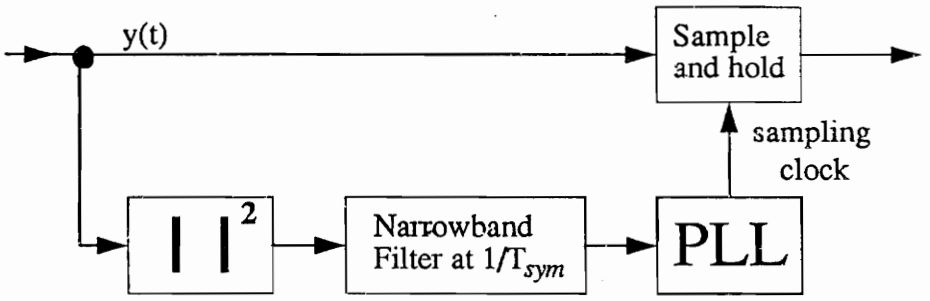


FIGURE 3.19: Block diagram of a squaring timing loop.

A squaring bit synchronizer is implemented directly in front of the sample and hold unit. Squaring the signal removes the information content of the signal. A narrow bandpass filter and phase locked loop (both center frequencies are set to $1/T_{sym}$) are used to synchronize the sample clock to the incoming signal.

Chuang has shown that if the loop bandwidth of the PLL (B) is much higher than the fading rate (F), the recovered timing t_d can be approximated with the knowledge of the power delay profile p (t) [Chu86]. He found that

$$t_d \approx \text{centroid of } p\,(t) \tag{3.40}$$

because, for B>>F, the channel can considered to be stationary with respect to the loop bandwidth. The centroid is equivalent to the normalized mean excess delay which has been defined in Chapter 2. Considering that the loop bandwidth has to be much less than

the symbol rate in order to recover the symbol clock, and a fading period of $\lambda/2$, this model can be used for symbol rates >> v/λ.

The other extreme case is given when the bandwidth of the phase locked loop is much smaller than the fading rate. For this extreme, Chuang found that the timing recovery loop of a digital receiver tracks the average delay of the channel. He defined the average delay of the channel as the average of t_d over the ensemble of impulse responses. For simulations modeling this extreme case, the average delay over all channel impulse responses used is calculated beforehand and kept fixed during the simulation. This model is very pessimistic in terms of predicted bit error rates, because a infinitely slow timing recovery loop is modeled. This is especially true for long simulations using complex imported channels, where large variations of the mean excess delay are possible [Dev87]. On the other hand, this method defines a constant sample point throughout the simulation, which is still more realistic than sampling at $t_d = 0$ constantly. Figure 3.20 shows an approximated sampling location if the squaring bit synchronizer is modeled

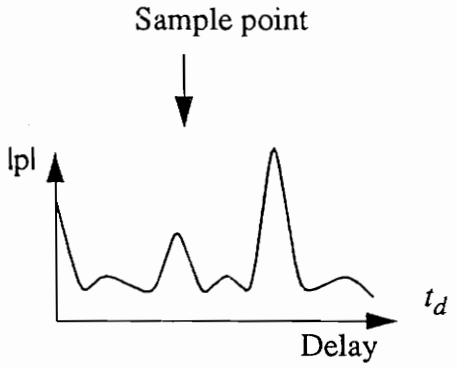


FIGURE 3.20: Sampling location if the squaring bit synchronizer is modeled.

If a fast timing recovery loop is modeled, the sample point is given by the centroid of a single channel impulse response. If a slow timing recovery loop is modeled, the sample point is given by the ensemble of impulse responses used for the simulation.

3.3.5.2 Ideal Sampling Recovery

In order to evaluate the performance of the squaring synchronizer models, a third additional tracking method was implemented. For this tracking method, an ideal tracking circuit was considered, which samples at a delay equivalent to the delay of the strongest multipath component in a channel profile (Figure 3.21). At this particular instant of time, the eye opening of the received signal was considered at its best.

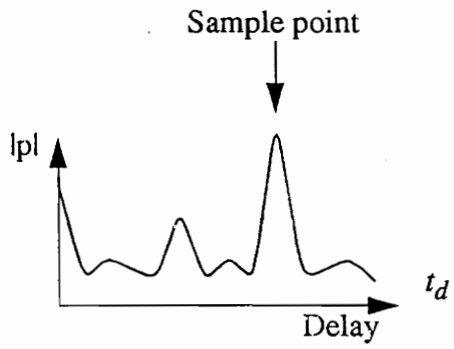


FIGURE 3.21: Sampling location if the maximum of the signal components is tracked.

Results and findings about the implemented recovery circuits, are discussed in Chapter 5.

3.3.6 Timing Jitter

Timing jitter is defined in several ways [Oer88] [Gar80] [Mar90]. In this work, timing jitter is the variation of the ideal sampling point [Gar80]. Timing jitter has been implemented to simulate the impact of timing imperfection. Sources that can introduce timing jitter in real world receivers are

- noise sources with spectral components close to the desired clock frequency in timing recovery loops,
- pattern noise [Gar80], which is data dependent and is caused by the intersymbol interference, and
- imperfect timing recovery due to non-ideal timing recovery circuits.

The timing jitter is modeled as a Gaussian-distribution about the ideal sample point. It is known that timing jitter increases with decreasing signal to noise ratio [Cou90]; nevertheless, the exact relation depends on the timing recovery circuit implementation. For this reason, timing jitter can be varied independently from the signal to noise ratio in BERSIM 2.0. The standard deviation, σ, of the timing jitter can be specified in terms of the symbol duration. This means, for example, that for a chosen σ of $0.5T_{sym}$, 68.27% of the samples are within the correct symbol duration, but only $Q(\frac{1}{N_s \times \sigma})$ of the time the received signal gets sampled at the optimum point. Thus, greater intersymbol interference effects and more bit errors are introduced by a larger timing jitter.

$Q()$ denotes the normalized error function of a Gaussian distribution.

3.3.7 Sample and Hold Unit

The sample and hold units samples the signal $\tilde{y}(t)$ at the sampling point determined by the timing circuit and passes it on to the threshold compare unit.

3.3.8 Threshold Compare Unit

In the threshold compare unit, the held value of the sampled signal is compared to a threshold and a decision is made of whether the received bit was a "high" or a "low" bit. The thresholds are different for the three implemented modulation techniques depending on the kind of detection (coherent or non-coherent) that is used [Fun91].

3.4 Supporting Units

3.4.1 Data Source

BERSIM 2.0 supports two methods to provide data to be sent through the simulator. The two methods are:

- RANDOM Data
- CUSTOM Data

As the names imply, the RANDOM data mode generates a random binary data stream with maximum entropy. This means the probability, p_1, of transmitting a "1" and the probability, p_0, of transmitting a "0" is 0.5 for both cases and the source is considered to be memory free such that there is no relation between successive bits.

In the CUSTOM data mode, any kind of user-provided data can be sent through the simulator. This option gives the opportunity of simulating application specific data patterns, such as speech data, video signals or video compression data.

A detailed description of the way user specified data is read and processed is given in the next chapter, describing the software implementation.

3.4.2 Delay Unit

The delay unit is a simple data buffer that stores the data for a certain amount of processing cycles. This is necessary because of the delay a signal experiences while transversing through the simulator. The sources for the delay are the implementation of the square root raised cosine filters and the channel. The resulting simulator delay, D_{sim}, is given as

$$D_{sim} = 2D_{Hrc} + N_s \tag{3.41}$$

where $(N_s\text{-}1)\ T_s$ is the discrete channel delay as defined in Figure 3.6 and D_{hrc} is the amount of time the square root raised cosine filters need to be shifted in order to make them causal filters [Fun91] [Kam87]. The complete channel delay is needed for timing recovery and timing jitter purposes. This is described in more detail in Chapter 4.

3.4.3 Eye Pattern Display

Eye diagrams are a very effective aid to qualitatively evaluate the performance of digital communication systems. Figure 3.22 shows a typical eye diagram for a $\pi/4$ DQPSK signal. Usually, eye diagrams are measured using an oscilloscope triggered by the symbol clock and hooked up to a receiver circuit before its sample and hold unit [Cho90]. In BERSIM 2.0, eye patterns such as Figure 3.22 are generated by dynamically overlapping 128 consecutive symbols in blocks of four symbols (one symbol not visible in diagram).

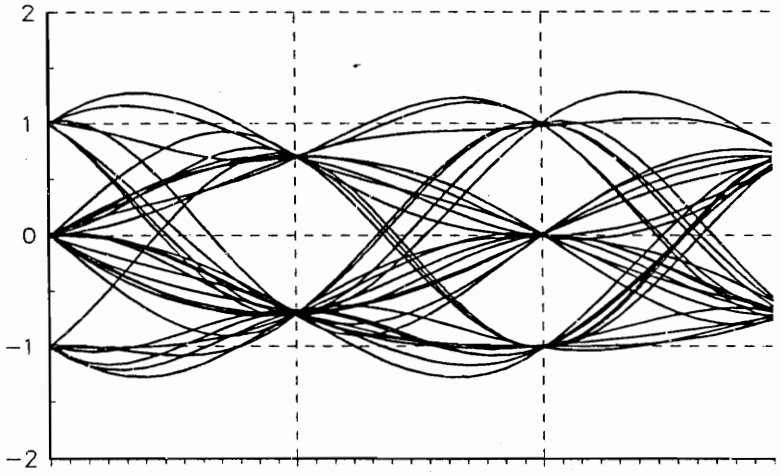


FIGURE 3.22: Eye diagram of the in-phase part of a $\pi/4$ DQPSK modulated signal, in a AWGN channel with 35dB Eb/No. The symbols are shaped with a raised cosine filter with a roll-off factor, α of 0.35.

Eye patterns are generated by writing several signals on top of each other. For distortion free transmission, the "eye" is wide open, horizontally as well as vertically. It can be easily seen that the optimum sampling point is exactly in the center of the eye. Several distortion mechanisms experienced in mobile digital communications have a major impact on the shape of the eye pattern. Two of these mechanisms are signal fading and inter-symbol interference.

Figure 3.23 shows the typical eye diagram of a BPSK signal that suffers from fast fading. The vertical eye opening, which is directly related to the received signal strength, shows very rapid changes. Those changes can even result in a vertical eye closure causing errors in detection.

I

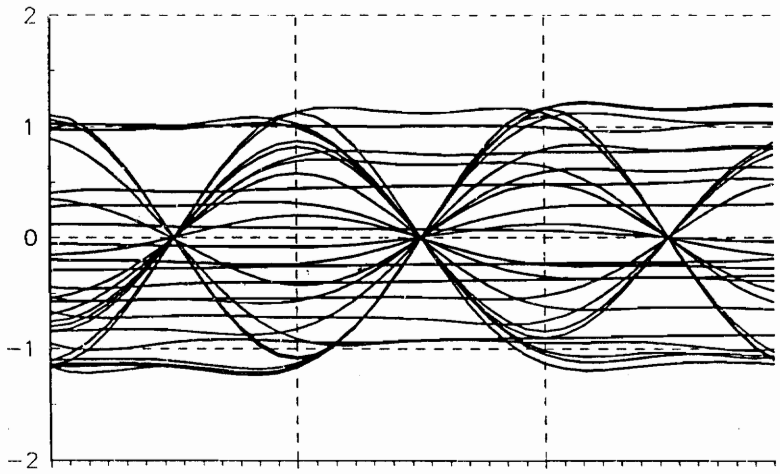


FIGURE 3.23: Eye diagram of the inphase part of a BPSK modulated signal, in a fast, flat, Rayleigh fading channel with 100dB Eb/No. The symbols are shaped with a raised cosine filter with a roll-off factor, α of 1.0.

Intersymbol interference is generated by multiple delayed receptions of the same symbol. It therefore causes horizontal eye closure. Figure 3.24 shows a typical eye diagram for this kind of distortion where the sampling point varies from bit to bit. Again, if the eye closes horizontally, errors will occur due to sampling errors.

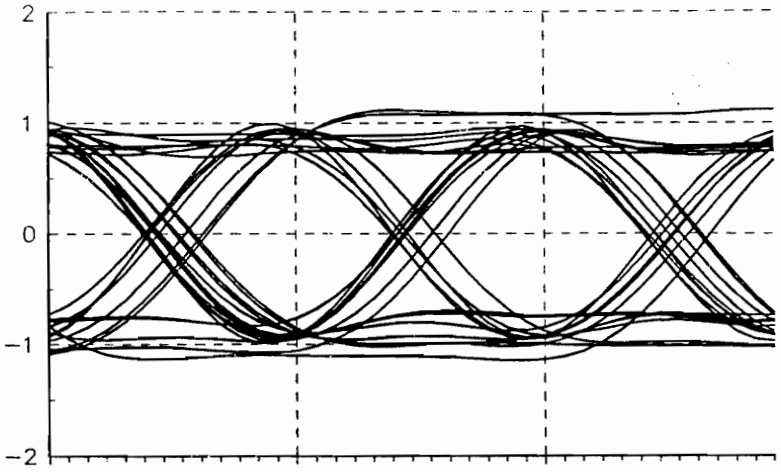


FIGURE 3.24: Eye diagram of the in-phase part of a BPSK modulated signal, in a two-ray frequency selective fading channel with 100dB Eb/No and a delay of 12μs between ray 1 and ray 2. The symbols are shaped with a raised cosine filter with a roll-off factor, α of 1.0.

Another source introducing signal and therefore eye pattern distortion are is noise received at the receiver. Figure 3.25 shows a typical noise distorted eye pattern.

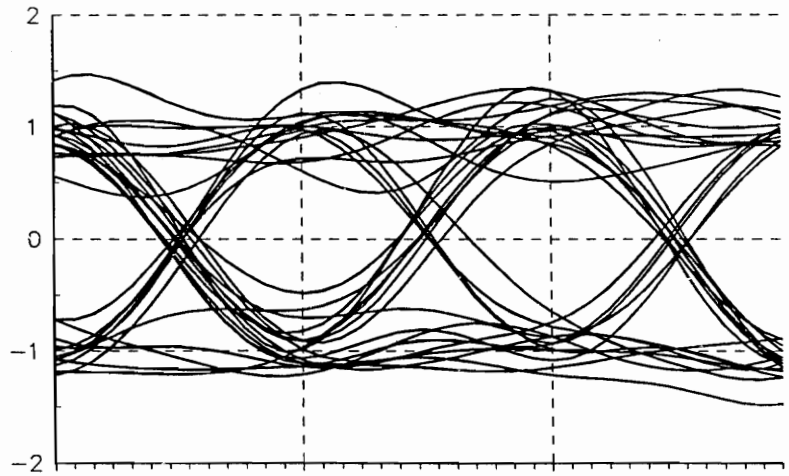


FIGURE 3.25: Eye diagram of the inphase part of a BPSK modulated signal, in a AWGN channel with 12dB Eb/No. The symbols are shaped with a raised cosine filter with a roll-off factor, α of 1.0.

In a real-world system, and in simulations of such real-world systems, all three mechanisms are present. Nevertheless, watching the eye pattern as well as the bit error rate displayed onscreen gives a good understanding about the source of the bit errors.

In BERSIM 2.0, the sampling point of each of the displayed symbols is marked as well. This way, how the simulated timing recovery methods work with respect to the optimum eye opening point can be observed. This implementation proved to be very valuable in analyzing results found in this thesis.

The eye pattern signals displayed by the bit error rate simulator BERSIM 2.0 are, as in real world receivers, taken right before the sample and hold unit. For the $\pi/4$ DQPSK system, the first or even bit branch in the receiver was selected randomly. Since sampling is performed on the inphase part of the signal, the I(t) signal is displayed.

For BPSK simulations, as for $\pi/4$ DQPSK simulations, the inphase part of the signal is displayed, since this signal is passed into the sample and hold unit.

The signal displayed for FSK simulations is not the raw I(t) or Q(t) signal of a demodulated signal like for the two previously mentioned modulation techniques. Here, the differential signal of the two demodulated OOK signals is displayed as it gets passed into the sample and hold unit.

3.4.4 Error Detection Unit With Error Recorder

The error detection unit performs an exclusive OR comparison of the received bit stream with the delayed bit stream coming out of the delay unit. The intermediate bit error rate is updated and calculated in periodic cycles depending on the data block length. In addition, optionally for every single transmitted bit, a corresponding error bit ("high" for error; "low" for no error) is recorded in a bit error pattern file in the order shown in Figure 3.26.

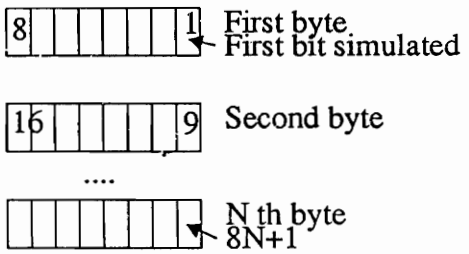


FIGURE 3.26: Order of the recorded bits in the bit error pattern file.

This file can be used for further data processing such as for determining error distributions or it can be fed into the BERSIM hardware simulator developed by MPRG [Rap91d] [BER92].

3.5 Summary

A description of the program BERSIM has been given. The description was based on the ISO network model. Therefore, the simulator is described in terms of channel, modem, and supporting units implementation. A detailed description of the radio channel modeling for internally generated Rayleigh fading channels and of conversions applied to imported channels was presented. For the modem, each functional block has been described from a communication theoretical standpoint.

4. BERSIM Software Realization

BERSIM 2.0 is written in the computer programming language C. To provide portability between different platforms, the well known and well defined dialect ANSI C has been used. Using ANSI C guarantees portability since this standard has strict, well defined requirements regarding program structure and syntax.

In this chapter, the software realization of the simulator described in Chapter 3 will be explained. A top-to-bottom approach is taken, which means that the program description occurs on four hierarchical levels:

1. Global description of the major modules in BERSIM and their interaction using a state diagram.
2. Description of the data flow between the major modules.
3. Description of the function of each major module, structure, and internal signal flows.
4. Description of the actual (low level) code implementation.

The low level code description is realized by well documented code which is supplied as a MPRG technical report [Tho92b]. Since the code is extensively commented, only particular code implementations which require additional attention will be discussed in this chapter. Graphical aids will be used in these descriptions whenever useful and applicable.

4.1 Global Description

BERSIM 2.0 consists of five almost independent executable programs. They are almost independent because each program can run individually; however, some of the programs need to have some initial data provided. The names of those programs are:

- *intface*
- *extern*
- *tworay*
- *tsimul*
- *prntdat*

The modules were realized in the form of separate executable programs for several reasons. First, some modules need to exchange large amounts of data (several Mbyte) on a file level using the hard drive as a buffer device. This necessitated that the modules operate independently so that sufficient memory would be available. Second, memory management features were more tractable to support individual executable modules for the program running on a PC. Last, using separate executable programs facilitated testing of each module. This way, it is not necessary to have test programs which would also need to be updated with program enhancements.

Each of these programs has an individual task to perform. Their basic functions and interactions are described based on the state diagram given in Figure 4.1.

The program ***intface*** can be considered the part of BERSIM 2.0 which interacts with the operator as well as controls the operations of other programs. Every part of a simulation (job) is initiated by this program and returns to this program after execution.

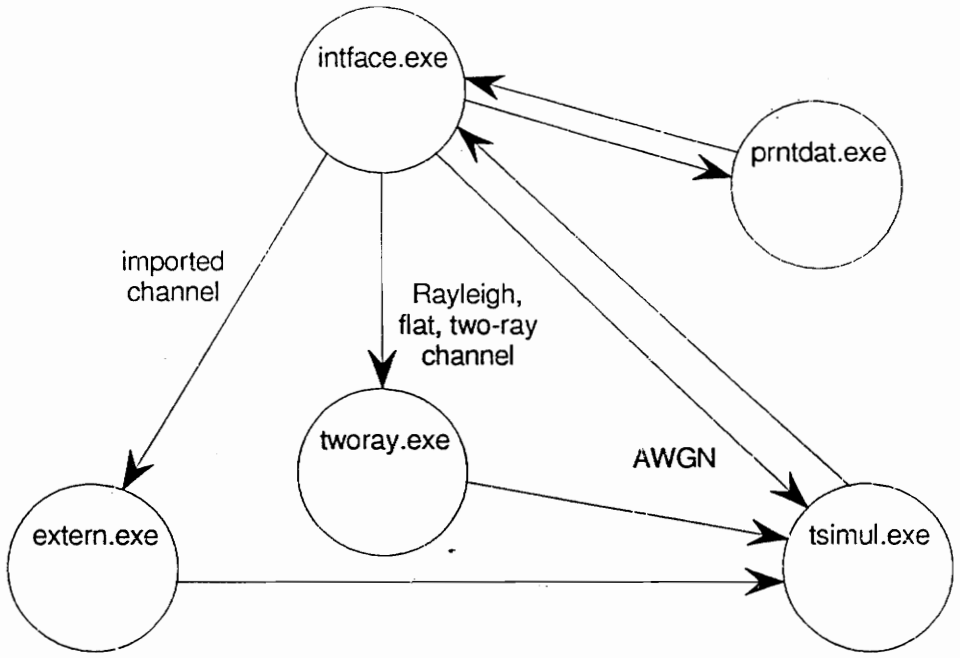


FIGURE 4.1: State diagram of the bit error simulator BERSIM

The program ***extern*** converts imported complex channel impulse responses into the correct format needed for the simulation as described in Section 3.2.2. The program ***tworay*** generates the flat fading or two-ray Rayleigh fading impulse responses, which are provided as internal channel models. The actual bit error simulation for the mobile radio system is conducted by the program ***tsimul***. The program ***prntdat*** is a supporting program that prints simulation results in a formatted fashion.

A simulation is initiated and performed in the following way. After the user specifies the simulation parameters and starts a simulation, the program ***intface*** calls one of the channel conversion or generation programs depending on the selected type of simulation. Only for simulations consisting of white Gaussian noise channels and for simulations with stationary receivers are the channel conversion or generation programs not called. After the channels have been generated or converted, this channel data is fed to the program ***tsimul***

which initiates the bit error simulation. The program ***prntdat*** can be called by the user anytime after a simulation, in order to print a simulation record.

4.2 Global Data Flow in BERSIM

Since the five major modules in BERSIM are executable programs, parameter exchange between programs is accomplished by using shared data files. Figure 4.2 is a data flow illustrating how the interfacing between the programs occurs.

Referring to Figure 4.2, the program BERSIM maintains two files which are used by almost all the programs within BERSIM. The files are named **menu13.inf** and ***simname*.inf**. The file ***simname*.inf** contains all the parameters related to one particular simulation, called ***simname***. The file **menu13.inf** contains the name of the currently active simulation. There is only one file with the name **menu13.inf** but there can be any number of parameter files of the type ***simname*.inf** whose name ***simname*** can be alphanumerical combination of letters other than the word **menu13**. Whenever the controlling program ***intface*** calls one of the other programs (child processes), the controller (mother process) writes first the name of the new active simulation into the **menu13.inf** file. The set of ***simname.inf*** files can be seen as a library of previously specified simulations by the user. Each child process that is called by the mother process first checks the **menu13.inf** file for the corresponding simulation parameter file name (***simname***) and then checks the corresponding simulation parameter file out of the library. All processing is done based on the simulation parameters stored in this file (***simname*.inf**). The programs ***extern*** and ***tworay*** modify some parameters before updating the file prior to their termination.

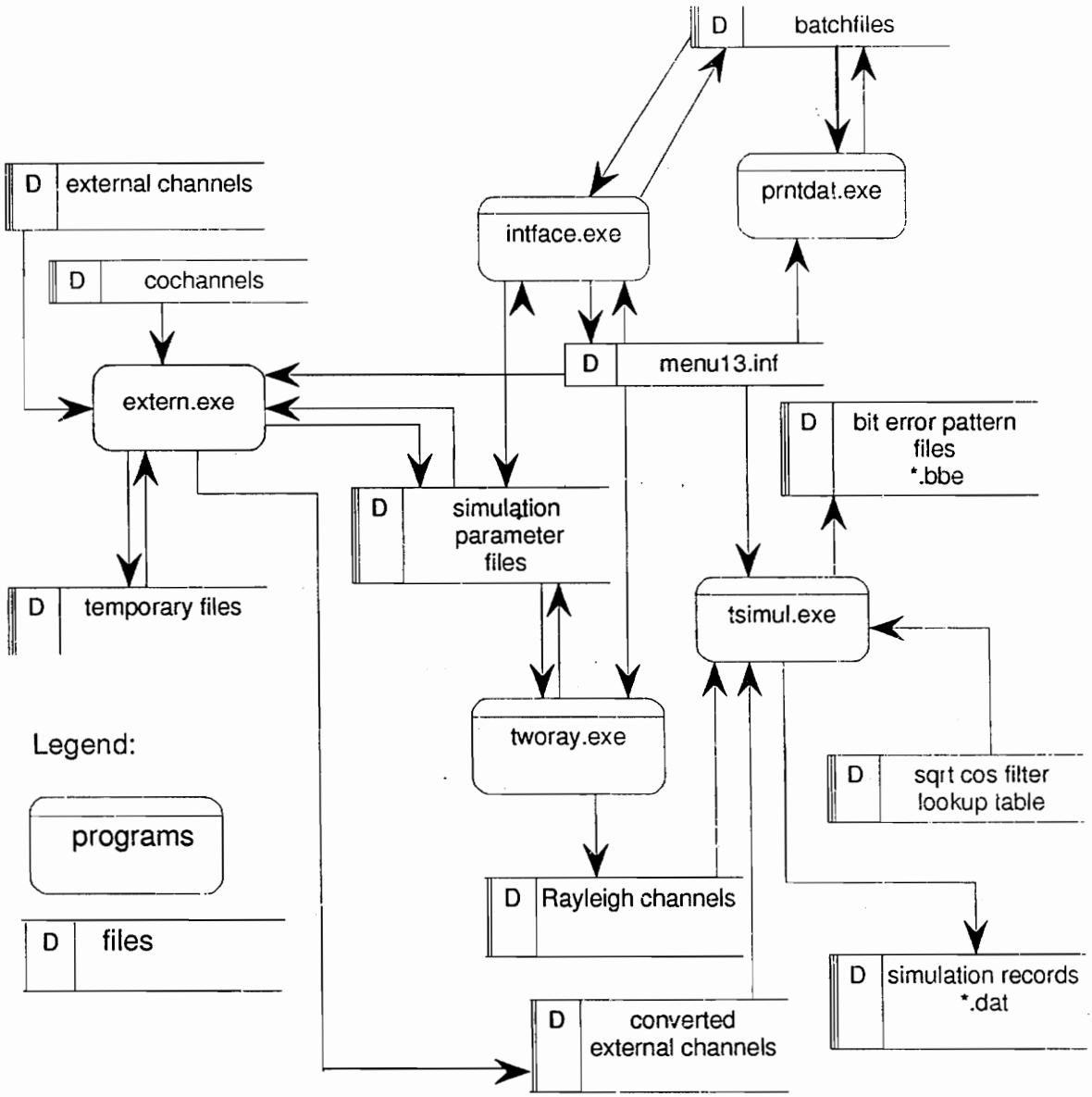


FIGURE 4.2: Data flow between the five independent executable programs in BERSIM.

The files with the extension **.bth** hold several parameter file names. These files are used for batch simulations where several simulations are performed in a sequential, automated fashion. The batch file simulations will be described in more detail later on in this work. At this point, it is important to know that only the controller (***intface***) and the print program (***prntdat***) have access to ***batchname*.bth** files. If a batch simulation is in process, the

controller reads a file name out of a specified batch file and puts it into the **menu13.inf** file. After placing it in the **menu13.inf** file, it calls the needed simulation programs. This process is performed in sequential order for each filename listed in a batch file. The print program makes use of a batch file in a similar way. If it is called and finds the name in the **menu13.inf** file to be a batch file, it automatically prints out all simulation records of a batch of simulations.

The program ***extern*** has, in addition to the two files ***simname*.inf** and **menu13.inf**, access to channel files which provide the input data of the desired channel and the interfering channel. For each complex channel impulse response, three files are provided. The files are required to have a **.bin** extension for the power magnitude file, a **.phs** extension for the phase file, and a **.ch** extension for the "channel info file". The "channel info file" is just an ordinary ASCII file that holds the channel parameters which have to be provided with external channel files (see Chapter 3.2.2, and Table 3.1, "List of channel parameters that need to be provided together with an external file," on page 29). The program ***extern*** reads the name of the channel file to be used in the simulation as part of the simulation parameters, whereas, the name of the cochannel interference file is specified in the program setup. While converting an external channel file into BERSIM format, the program ***extern*** generates temporary files. These temporary files can be rather large. Sufficient disk space (about 15 Mbyte) needs to be provided in order to permit the conversion of an external channel file.

Unlike the program ***extern***, the program ***tworay*** requires no input other than the two **.inf** files. To save memory, the two complex fading envelopes (twom1_*.dat, twop1_*.dat, twom2_*.dat, and twop2_*.dat) are saved in a different format than the channel impulse responses provided by ***extern***. Whereas ***extern***'s output files represent a sequence of chan-

nel impulse responses, ***tworay's*** output files represent two single envelopes, each given as signal (not power) magnitude and phase.

Depending on the type of simulation requested, (imported channel or internally generated two-ray model) the actual simulation module ***tsimul*** expects to find the corresponding channel files which are provided by either ***tworay*** or by ***extern***. The same statement can be made for simulations using user specified data. The simulation parameter file will tell ***tsimul*** if the simulation with user defined data is desired and, if it is, where to find this data.

If recording the bit error pattern was selected by the operator, ***tsimul*** records this pattern in a file which has the same name as the actual running simulation, but with the extension **.bbe** (e.g., ***simname*.bbe**).

Before the end of a simulation, a simulation record containing all simulation parameters and simulation results is generated. This simulation record has the same base name as the simulation parameter file, but with the extension **.dat** (e.g., ***simname*.dat**).

Recording the signal before the sample and hold unit (eye pattern signal) is also supported in BERSIM 2.0. However, the recording of this signal cannot be enabled from the interface, because accidently recording the eye-signal would require an excessive amount of disk space. Recording of the eye-signal is included for future research endeavors. It is presently commented out in the working code.

4.3 Description of Each Program Within BERSIM

4.3.1 Dynamic Memory Allocation

In designing the core of the BERSIM modules, available memory was always a major concern. In order to run the program under MS-DOS on personal computers, which are limited to about 600k of memory with this operating system, dynamic memory allocation is used frequently. Dynamic allocation for one dimensional arrays is straight forward, so even an unexperienced C programmer should have little difficulty understanding the process by examining the BERSIM 2.0 code. However, the method used for allocating two dimensional arrays is more complex, so a description of the procedure is given here.

The *ANSI C calloc* command is used to first allocate the entire amount of memory needed by the array (A' in Figure 4.3). Then, the memory for a one dimensional array of pointers is reserved (A in Figure 4.3). The array of pointers has the same dimension as the desired y-dimension of the two dimensional array (Figure 4.3).

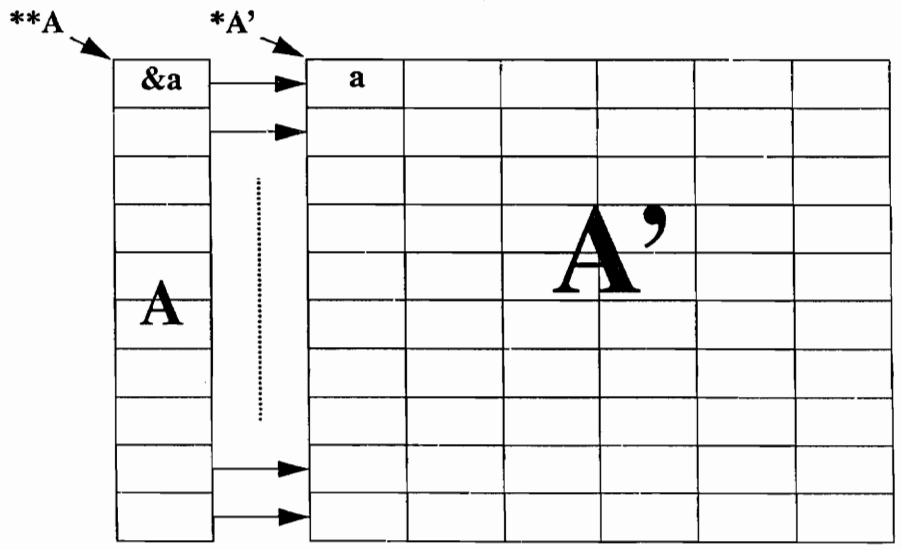


FIGURE 4.3: Dynamic memory allocation of two-dimensional arrays.

Next, the array of pointers gets filled with the addresses of the first elements of the x plane. With this method the conventional way of addressing array locations such as A[i][j] could be still used, but the array was allocated at the memory heap and not on the more size limited memory stack.

4.3.2 The Interface and Process Controlling Program *intface.exe*

The functions of the program *intface* can be divided into two groups. The first is the group of functions in charge of parameter input and output between user and program. The second is the group of functions which controls the program flow.

Structure charts and flowcharts for the program *intface* are given in Appendix A.1. Following a brief introduction in the use of *intface*, a description of the input parameters and the dependencies of these parameters upon each other are given. Next, the control portion of the program will be addressed. Finally, the file structure and compiling issues will be discussed.

If one starts the program *intface*, which is equivalent to starting the program BERSIM, one gets a machine independent (SUN, PC) input window (Figure 4.4) which gives five choices to the operator.

```
        Mobile Radio Bit Error Simulator (BERSIM)
Version 2.0  (C Version)                Copyright 1991
Mobile and Portable Radio Research Group, Virginia Tech



1. Set/Retrive Parameters
   (Modify simulation and communication system parameters)
2. Set up batch run
   (Create and save batch run file)
3. Run Simulation
   (Run BER simulation (single or batch))
4. Print simulation results
   (Prints simulation result files on printer)
0. Quit

   Select task: □
```

FIGURE 4.4: Main menu of program intface.

Selecting choice 1 brings up the parameter I/O window (Figure 4.5).

```
        Mobile Radio Bit Error Simulator (BERSIM)
Version 2.0  (C Version)                Copyright 1991
Mobile and Portable Radio Research Group, Virginia Tech
     *******   Set/Retrive Parameters   *******

              File: noname00
          1. Duration of run (sec)        :   1.000
          2. Generate BER pattern file    :   NO
          3. Select simulation data source:  RANDOM
          4. Mode                         : Free run
          5. Calculate Outage probability :  YES
               501. Outage block length   : 324
               502. Outage BER threshold  : 1.00e-02

   Transmitter                 Channel                  Receiver
11.Modulation:   Pi/4 DQPSK  21.Type :   Two_Ray     31.Speed    : 10.0 km/h
12.Symbolrate:  24300 sym/s  22.Delay:  3.1656 us    32.Eb/No    :  35 dB
13.Rolloff   :   0.35        23.C/D  :   0 dB        33.STiming  :  0
14.Frequency :  850.0 MHz    24.C/I  : 100 dB        34.Jitter   : 0.00
                             25.File :     N/A
          98. Retrive file                99. Save File
                          0. Quit
      Select parameter number (Enter): □
```

FIGURE 4.5: Select/Retrieve Parameters window

Using this window, simulation parameters can be modified by first selecting the corresponding parameter number and entering the new value for this parameter. Saving the current file adds it to the library of simulation parameter files, which must be done in order to use the selected parameters for simulations. Retrieving a file allows the operator to look at a file in the library and to perform any desired modifications on it. Except for "Quit", all other parameters are simulation parameters and these values are stored in a *simname*.inf file. The Tables 4.2, 4.2, and 4.3 give a brief description of each variable and the limits that apply. Other than the given echo of the program (Figure 4.5), they are grouped as simulation set up parameters, modem parameters, and channel parameters.

Selecting choice 2 in the main window puts the user in the batch file generation mode. After entering a file name for the batch file, one can select between two ways of setting up a batch file. One is called a random input batch file, where the user can set up several simulations with completely different parameters by manually generating the required simulation parameter files and adding their names to the batch file. The other is called a range batch generation. In this mode, one parameter can be selected to vary over a specified range. In order to generate such a sequence of parameter input files, where only one parameter varies over a desired range, one has to define a "seed" simulation parameter file. The variable which should be varied has to be selected and its maximum desired value, as well as the step size of the increments, has to be defined. The generated simulation parameter files are named by using the first 5 characters of the seed file name, then appending a sequential number from 00 to 99. For example, if the seed file is "TESTBER.INF", the sequential files will be TESTB00.INF, TESTB01.INF...)

TABLE 4.1: Simulation set up parameters

Parameter	**Description**	Limitations
Duration of run (sec)	Selected simulation duration	Available disk space; Max. available channel file length for imported channel files; Max. available amount of data for CUSTOM data simulation.
Data source (RANDOM / CUSTOM)	Selected data source Random data or data from a user specified file.	none
Data source file	Name of the data file to be used for simulation.	
Mode (Free Run, Bounded)	Select if brute force "free run" method or "bounded binomial sampling" method described in [Cro76][Fun91].	
Maxbits	Maximum number of bits to be transmitted if "bounded mode" is used.	Same as for duration.
Minbits	Minimum number of bits to be transmitted in the "bounded mode"	none
Minerror	Error threshold after which a simulation is terminated if the bounded binomial sampling method is used.	none
Bit error pattern? (YES/NO)	Switches the bit error pattern recording on or off.	
Calculate outage probability? (YES/NO)	Switches outage probability calculation on or off.	
Outage block length	Block length in bits used to calculate outage probability.	
Outage threshold	Sets the BER criteria in percent of errors per block.	

TABLE 4.2: Modem parameters

Parameter	Description	Limitations
Modulation ($\pi/4$ DQPSK, BPSK, FSK)	Type of modulation used for simulation	
Roll off factor	Roll off factor of the square root raised cosine filter	0.0 - 1.0 in steps of 0.05
Frequency (MHz)	Carrier frequency	max. supported doppler shift of 150 Hz for the internally generated Rayleigh channels. Frequency can not be varied for imported channels.
Sample Timing recovery	0 => no recovery 1 => max. delay profile component 2 => centroid of delay profile 3 => centroid over all delay profiles	
Timing jitter	Standard deviation in σ of the jitter of the sampling unit normalized by the symbol duration	0.0 - 2.0 σ

TABLE 4.3: Channel parameters

Parameter	Description	Limitations
Eb/No (dB)	Bit energy to noise spectral density ratio	-100 to 100 dB
C/I (dB)	Carrier to interference ratio, which is the ratio of the desired signal to the interference signal.	-100 to 100 dB
Channel type	"Two-ray"or "External" Implemented channel model.	
Delay	Delay between the first and the second ray if the internal two-ray model is used.	0 to $\frac{2}{\text{symbol rate}}$

TABLE 4.3: Channel parameters

Parameter	Description	Limitations
C/D (dB)	Carrier-to-delayed signal ratio. Power ratio between the first and the second ray if the internal two-ray model is used.	-100 to 100 dB
Receiver speed (km/h) or (m/s)	Velocity of the mobile or portable unit.	max. supported doppler shift of 150 Hz for the internal generated Rayleigh channels. No limits apply for imported channels.

If option 3 in the main menu (run simulation) is selected, the operator can select between running a batch of simulations or a single simulation. After entering in the name of the desired file to simulate (either single or batch simulation), the program *intface* takes control over the entire simulation run. It first provides the correct simulation parameter file name to the other programs (menu13.inf), before calling them in the order needed for the selected simulation.

Figure 4.6 shows the order in which the other executable programs are called by the program *intface* for a single simulation. The program *intface* first "checks out" the channel parameter file from the library. If a cochannel interference signal needs to be generated (depending to the selected cochannel interference ratio), it sets the cochannel flag which is one of the simulation parameters and initiates a simulation run. For a cochannel simulation, the same operations/simulations as for the primary channel are done. However, the signal after the convolution with the channel is only recorded for later use as the interfering signal, and is not demodulated in the receiver. After the cochannel interference signal is generated, the program *intface* checks out the parameter file again and resets the cochannel interference flag. Now the other modules are called again for the desired simulation run.

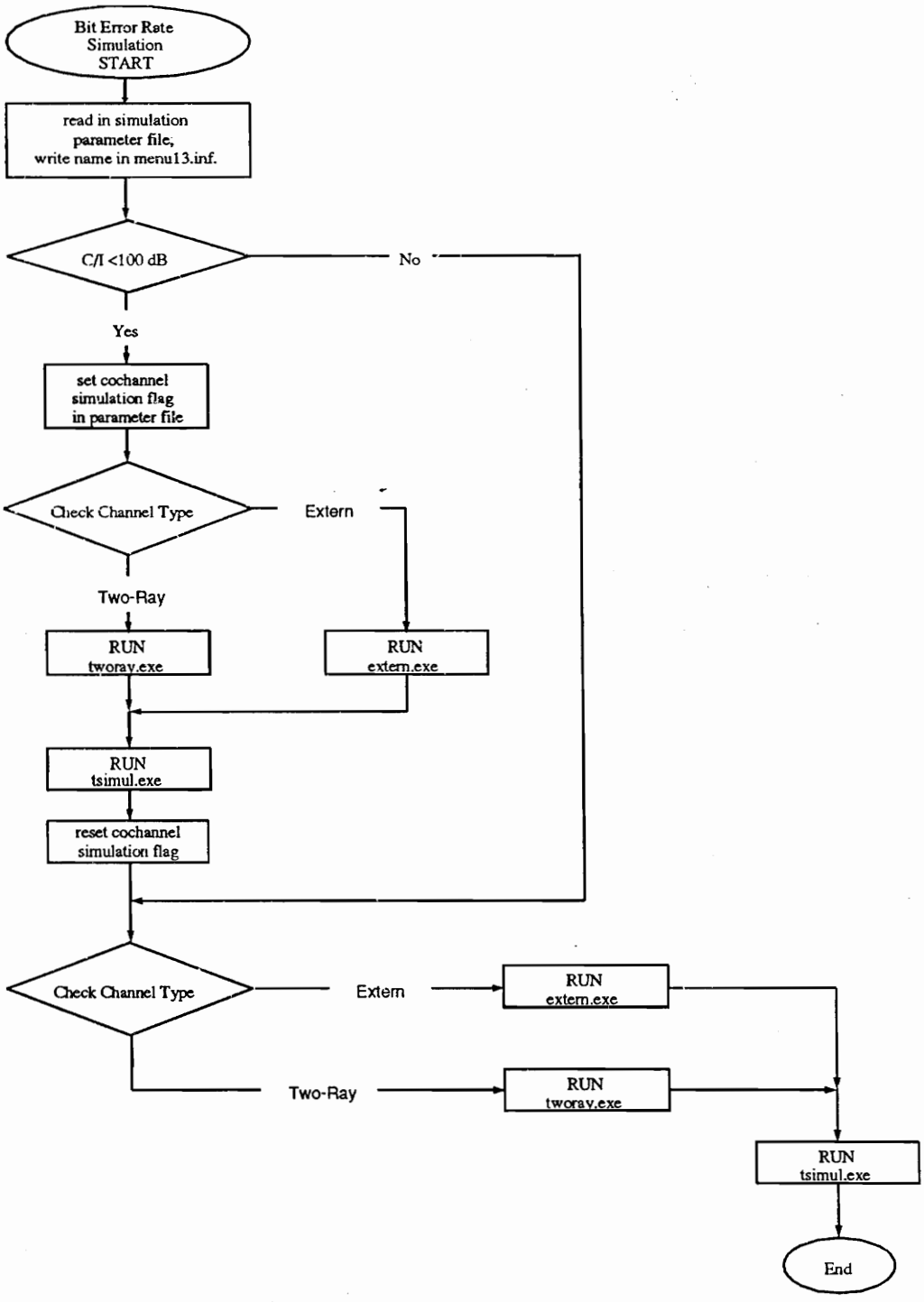


FIGURE 4.6: Order in which intface.exe calls other executable BERSIM modules for simulations where a channel impulse response is needed.

4.3.2.1 File Structure and Compiling Issues

In order to generate an executable program *intface*, the following files are needed for compilation:

- intface.c (file containing the core code of intface)
- paraio.c (file containing the simulation parameter routines)
- struct.h (file containing the structure and constant declarations used in BERSIM)

If the program is compiled on a UNIX system, one must also link the libraries *termcap* and *curses* to the program in order to have screen-oriented cursor motions [SUN90].

4.3.3 The Program *extern.exe*

The program *extern* is the program which performs the file conversion as described in Section 3.2.2. Structure charts and flow charts of this program are given in Appendix Section A.3. Referring to Appendix A.3, the program reads first the name of the current simulation file and then the corresponding simulation parameters.

The function *Splitextfiles* checks the size and the format of the external channel file. If needed, the function splits the file into subfiles. The major constraint concerning the maximum file size is given by the maximum array size for the DOS operating system. For all the manipulations to be performed on a channel impulse response, it is desirable to have the set of channel impulse responses, representing a certain travel distance, in the memory at once. Since a DOS operating system supports only array sizes of 64K with the use of ANSI C commands, an upper bound for the number of channel impulse responses per channel subfile is given. Considering a maximum supported resolution of 128 multipath components per channel impulse response, each given as an 8-byte floating point number, the number of channel impulse responses per subfile is limited to 32 impulse responses.

Often, this number of impulse responses cannot be put into one channel subfile because the subfile length must be less then 6λ in order to maintain the desired resolution (see Section 3.2.2). For the simulation program *tsimul*, each subfile has to be of the same size. To satisfy all these constraints, the function *Splitextfiles* might need to dispose some of the original imported channel impulse responses at the end of the original imported channel impulse response file. For example, if an imported impulse response file contains 38 impulse responses with a given impulse response spacing of λ/4, the last impulse response gets disposed. The program will generate 2 subfiles each holding 19 impulse responses and the last impulse response of the first subfile will be the first impulse response of the second subfile (see Figure 3.5). The last impulse response in the original channel file is disposed in order to have two files of the same size.

The function *Findresamppts* finds the best matching number of sample points for a cubic splined fading envelope as described in Section 3.2.2. All cubic splined fading envelopes have the length of the traveled distance covered by a channel subfile. Considering N symbols to be transmitted per channel sub-impulse response, the function finds the best matching combination of sample points (*resamppts*) and symbols to be simulated per sample point (*no-of-sym*) that provides the highest resolution.

4.3.3.1 Splining of the Signals

The cubic spline methodology used to perform a spline interpolation on the magnitude and phase of the fading envelopes is similar to algorithms used in [Joh82] and [Pre90].

To rebuild a continuous channel impulse response, the very last point of the cubic spline is not saved for further processing. This was done because this point is equivalent to the very first point of the next file (Figure 4.7)

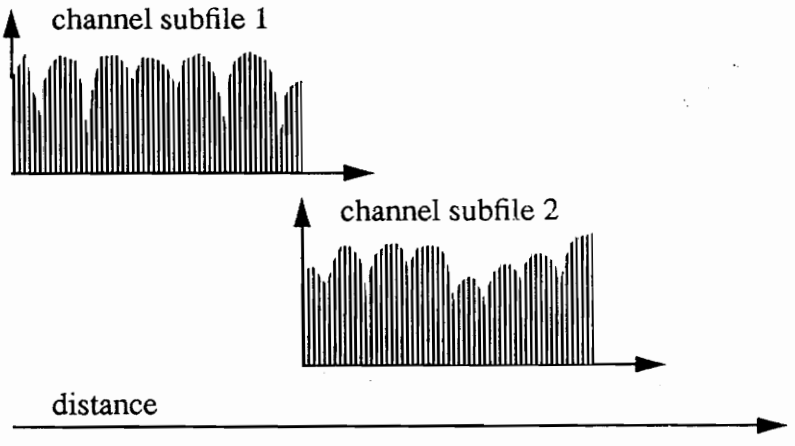


FIGURE 4.7: Overlapping channel subfiles.

The rms delay spread calculation (which is the reported rms delay spread in the simulation record) is performed after the cubic spline interpolation. A spatially averaged rms delay spread is computed from the spatially averaged channel impulse response. The centroid over all channel impulse responses used for the timing recovery simulation is calculated after the splining as well. As a last step, all splined fading envelopes are added vectorially into bins as described in Section 3.2.2.2.

The converted external channel files have the names:

- /extern/**inm#.dat** for the channel impulse response magnitude files
- /extern/**inp#.dat** for the channel impulse response phase files
- /cochannel/**cochani1.mag** for the cochannel impulse response magnitude file
- /cochannel/**cochani1.phs** for the cochannel impulse response phase file.

4.3.3.2 File Structure and compiling issues

For compilation of extern, the following files are needed:

- extern.c (main program)

- externcv.c (file conversion functions)
- externfu.c (general functions)
- externio.c (input/output functions)
- paraio.c (parameter file input/output functions)
- struct.h (structure and constant declarations)

4.3.4 The Program *tworay.exe*

Flow charts and structure charts of the program *tworay* are given in Appendix A.2. Appendix A.2 gives an overview of the structure of the program *tworay*. The Rayleigh fading envelopes are generated by the function *Genrayleigh*. As mentioned in Section 3.2.1.1 the length of a fading envelope generated by the program *tworay* is fixed to 24λ in order to have constant resolution regardless of frequency and speed.

Since a discrete inverse fast Fourier transformation is used to generate the fading envelopes, the constant resolution in terms of samples per wavelength has its trade off in the frequency domain. For the given application, the resolution in the frequency domain for a discrete IFFT is given as

$$\Delta f = \frac{1}{T_{envelope}}. \tag{4.1}$$

where $T_{envelope}$ is the duration of the generated fading envelope.

The time needed to travel 24λ is given by

$$t_{24\lambda} = \frac{24\lambda}{v} \tag{4.2}$$

one can substitute the velocity using the doppler frequency equation

$$fd = \frac{v}{\lambda} \tag{4.3}$$

to get

$$t_{24\lambda} = \frac{24}{fd} \tag{4.4}$$

Substituting this equation into (4.1) indicates that

$$\Delta f = \frac{fd}{24}. \tag{4.5}$$

always. Thus, the discrete doppler spectrum in *Genrayleigh* always has only 24 frequency components, regardless of mobile speed and frequency. For doppler shifts larger than 200 Hz, the relatively large spacing of the frequency components results in generated fading envelopes whose PDFs[1] differ from the PDF of a Rayleigh distribution. The difference is significant at the tail ends of the distribution, which means deep fades are less likely to occur in the simulated channel. For this reason, the impact of random FM decreases on bit error rate simulations performed for doppler shifts much larger than this threshold. This results in overly optimistic bit error rate predictions. The only way to increase the maximum supported doppler frequency and not lose resolution in the time domain is to implement an IFFT that uses more samples than the 8092 tab IFFT implemented for this research. However, since the maximum array size on a personal computer running MS-DOS is limited to 64K, no IFFT with more than 8092 tabs can be implemented. Although this larger IFFT size can be achieved on the workstation, for this work it was preferred to have the same resolution on both platforms.

1. Probability Density Function

The resampling routine in *tworay* operates differently than the cubic spline routine used in *extern*. A linear interpolation method is applied to find the envelope values at the locations, for which a symbol transmission is simulated. Similar to the program *extern*, the locations are determined by a routine called *Findresamppts* that determines the symbol transmission locations based on the symbol rate. A more detailed description of the routine can be found in its function header in the code listing. For a description of the other routines in *tworay*, see the function headers in the code as well as to corresponding flow charts.

4.3.4.1 File Structure and Compiling Issues

To compile the program *tworay*, the following files are needed:

- tworay.c (two-ray program file)
- paraio.c (simulation parameter input/output routines)
- struct.h (structure definitions and constant declarations).

To compile it on a UNIX system, the libraries

- termcap.h
- curses.h

must also be linked to the program in order to have screen-oriented cursor motions [SUN90].

4.3.5 The Simulation Program *tsimul.exe*

The program *tsimul* is the core of the bit error rate simulator. Its structure diagrams and several program flow charts can be found in Appendix A.4.

The program *tsimul* is a data block oriented simulator. Block oriented simulation means, that a block of data is passed into the simulated system and passed through all stages before the next block of data is pasted into the simulator. Since a continuous signal is simulated in combination with a system that has memory (any kind of time dispersive filter), interaction between successive blocks is always present.

In the current version of BERSIM, a signal block is 40 symbols long. However, the size of a signal block can be varied rather easily by changing a "*#define*" statement in the constant declaration file **struct.h**. Larger block sizes can speed up the simulation time. However, the drawback is a larger memory requirement which is already very limited on a personal computer running MS-DOS. Each of these 40 symbols is sampled 13 times, resulting in a signal block 520 samples long.

The number 13 has its origin in BERSIM 1.0 where it represents the maximum possible number of samples per symbol for simulations in combination with SIRCIM channels [Sei89] [Rap91a]. The reason for this selection was that the minimum sample spacing of the signal samples needed to be greater than the spacing between multipath samples of a SIRCIM channel. Since SIRCIM's multipath samples are spaced 7.8125 ns apart, the maximum signal sample spacing could only be 7.8125 ns as well. In order to support symbol rates up to almost 10 Msym/sec, 13 samples per symbol is the maximum possible number which still meets this constraints.

In BERSIM 2.0 where any external channel can be imported, this constraint is no longer a consideration. However, since the number of samples per symbol has to be kept rather low to save computation time, an odd number of samples per symbol proved to be very useful. This way a symbol is symmetric with a sample point right in the center.

Although the symbol block length of 40 symbols can be varied easily, the number of samples per symbol *can not* be varied *without* changes to the program. The part of the program which needs to be changed is the implementation of the square root raised cosine filters. Currently, the weights of the filter tabs are stored in the form of look-up tables for each valid roll-off factor. The stored filter impulse responses are 12 symbol periods long with each symbol represented by 13 samples (tabs) [Fun91]. Thus, for a variation of the number of samples per symbol, the look-up tables need to be modified as well, or a module which generates the needed filter impulse response dynamically needs to be implemented.

To account for the memory nature of the system, each signal block carries the tail end of the previous block as a preamble. The length of this preamble is determined by the system component which has an impulse response with the largest temporal spread. In BERSIM 2.0, the square root raised cosine filters have the largest temporal spread. According to [Fun91], a square root raised cosine filter impulse response can be described with sufficient accuracy within 12 symbol durations. As a result, the preamble has to be 155 samples long. Figure 4.8 shows the structure of a complete signal block.

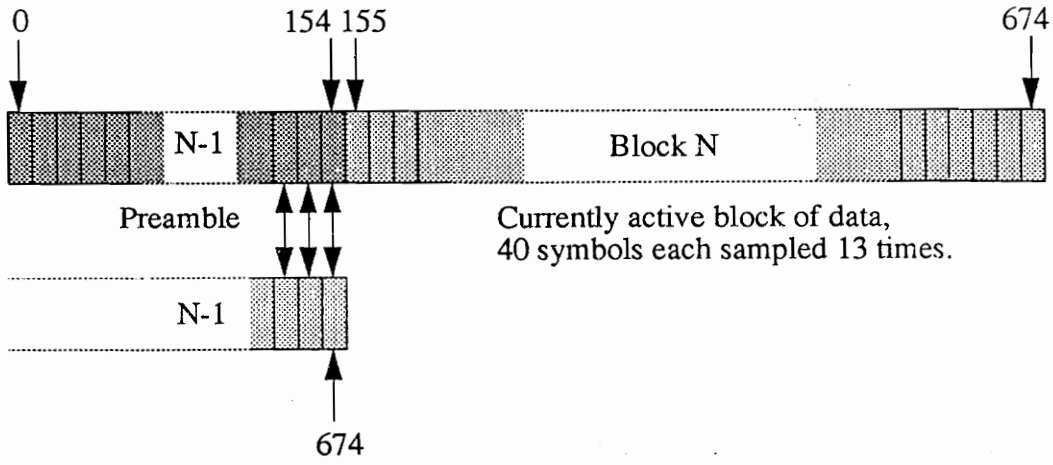


FIGURE 4.8: Simulation block structure

BERSIM uses baseband representation of signals; therefore, each signal is represented by two signal blocks (I(t) and Q(t) channel). The current version of *tsimul* has 4 channels going through the system (e.g., for FSK, I(t) and Q(t) channels for the frequencies 1 and 2, or for BPSK, I(t) and Q(t) channels for the reference and for the simulation signal). Those 4 channels are processed in a sort of parallel fashion. They represent two pairs of complex baseband signals.

All signal-processing in the module *tsimul* is performed in the time domain using discrete complex algebra and discrete complex convolutions. Before each convolution of a signal block with a filter, the preamble of a signal block is filled with the tail of the previous block. Out of several methods to perform this task, including the use of a signal block twice as long as the one implemented to have a wrap around array (no memory copying needed), copying a block of memory holding the tail end of the previous signal block into the memory location allocated for the preamble was the most efficient.

As discussed in Section 4.3.3 and in Section 4.3.4, a channel file is often not long enough for an entire simulation run. As a result, channel files need to be switched during a simula-

tion. Figure 4.9 shows that if the maximum number of symbols to be transmitted per channel is reached, the channel file is switched. This switching can occur in the middle of a block of data.

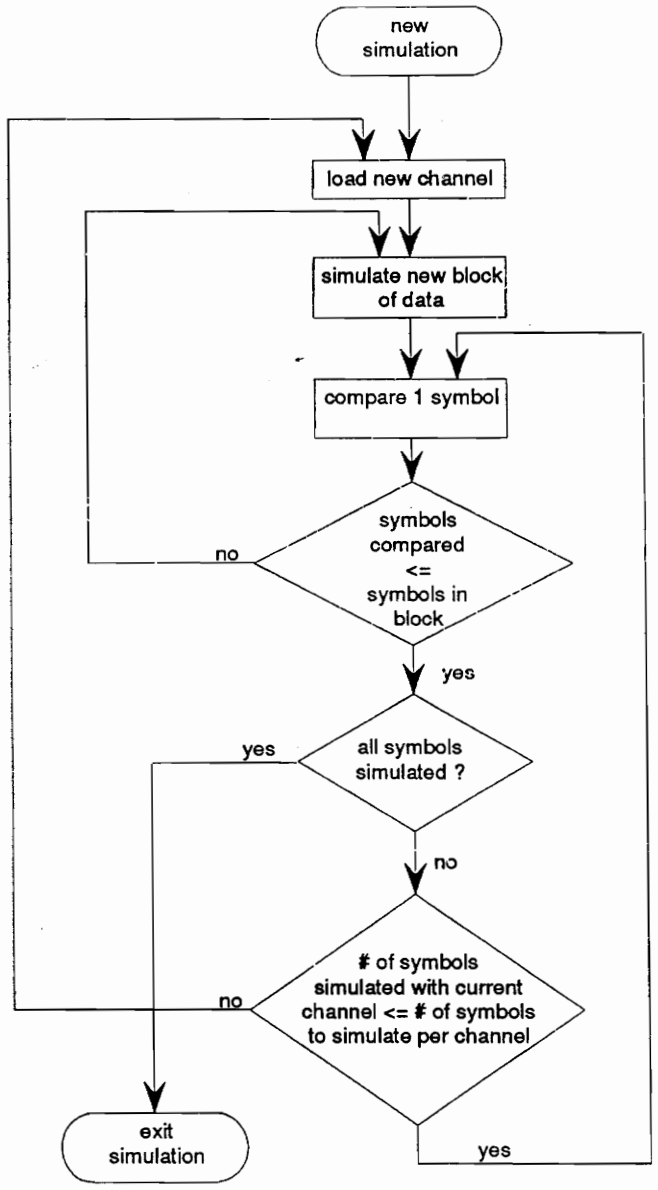


FIGURE 4.9: Switching of channel files

Switching at the end of a complete data block would put additional constraints on the sample spacing of the fading envelope samples, resulting in poorer resolution. With this method, the channel file would need to end at the end of a data block. The resolution would become a function of the block length which is not preferable for possible future variations of the block length.

Another factor which must be carefully considered is the fact that at the beginning, and between successive channel impulse responses for the two-ray model, discontinuities of the channel's spatially varying impulse responses are present. Random data is sent through the channel at the beginning of each new channel impulse response until the steady state is reached. The data bits received in this time period are not considered or recorded for any further processing. The number of random bits sent is equal to the delay a signal experiences in the channel. Additionally, no mobile movement is simulated for this period, which means that the first channel impulse response of a channel file is kept until the steady state is reached and relevant data is transmitted. With this method, the channel transition between two successive imported channels is kept continuous for imported channels.

For simulations involving the internal Rayleigh fading generator, the beginning of a new channel can be considered as the start of a new simulation for a different location with the same parameters. If user defined data is sent, the last block which is simulated with a given channel file is only filled with user specified data, up to the symbol where the channels get switched. A continuous flow of user specified data over file boundaries is guaranteed.

It has to be emphasized that even though the channel is continuous over channel transitions for imported channels, the actual signal can not be considered continuous because

the signal at the end of the previous file (usually not the end of a block!) is not the preamble of the first block of the succeeding file. However, this has no significant impact on the BER as well as on the recorded error pattern. Modifications of the program to achieve continuity of the signal (e.g., for soft decision decoding across file boundaries) over file boundaries would require a very complicated bookkeeping process over all system blocks a signal is sent through.

4.3.5.1 Convolution of the Signal With the Channel

A convolution of the signal with the channel is only performed when the mobile is moving and spatially varying channel impulse responses are provided. Two different convolution routines (*Convolveexternal* and *Convolvetworay*) are provided for several reasons. One reason is that the channel data provided by the modules *extern.exe* and *tworay.exe* is saved in different formats to conserve disk space. Another reason is that different convolution methods are applied to save computation time.

The two-ray channel model allows modifications of the convolution that can speed up the convolution process significantly for large delays. Considering the discrete convolution equation which can be expressed as

$$y(t) = \sum_{k=0}^{K-1} x(k) \times h(t-k) \tag{4.6}$$

and knowing that *h(t)* is given as

$$h(t) = a_1 \delta(t) + a_2 \delta(t-(K-1)) \tag{4.7}$$

equation (4.6) can be expressed as

$$y(t) = x(0)\, a_1 \delta(0) + x(K-1)\, a_2 \delta(0). \tag{4.8}$$

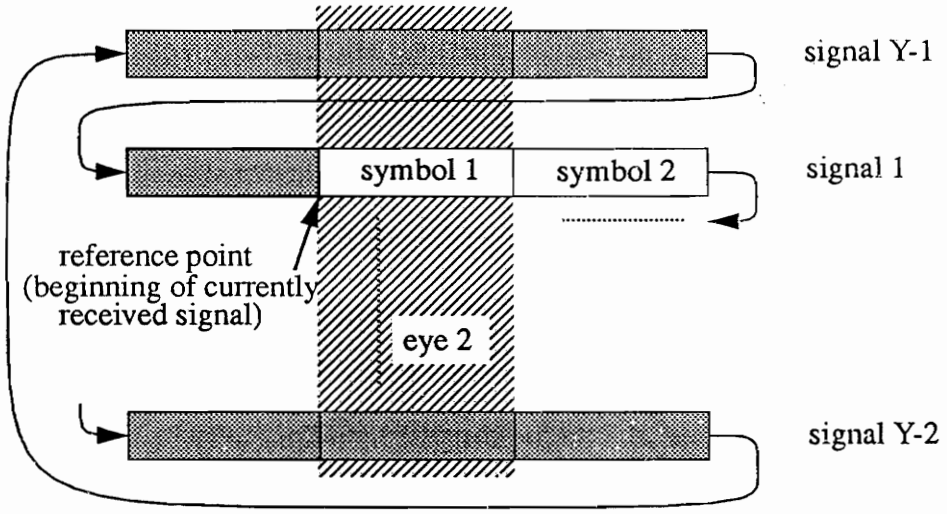
Equation (4.8) is the new convolution equation, which takes into account that all tabs other than tab 0 and tab K-1 are zero in a two-ray channel impulse response. In imported channel impulse response files, all discrete delay tabs from tab 0 to (K-1) can be filled. Thus a complete discrete convolution needs to be performed.

4.3.5.2 Matched Filtering at the Receivers

Each of the three receivers is implemented twice in the corresponding receiver function (*Pi4DQPSKrec*, *BPSKrec*, *FSKrec*). The reason for this double implementation is to save computation time. Whenever no timing recovery, no timing jitter, and no eye pattern display or recording have been selected, the signal level after the matched filter is determined only at the end of the integration (convolution with matched filter) for each symbol. It can be also said that the received signal is only calculated for a fixed sample point. This method saves, in BERSIM 2.0, 12 full convolutions per symbol, because the signal sample points in between those endpoints are not calculated. However, no complete received signal is available and the sample point is always this particular sample throughout the entire simulation run. In all other cases, a full convolution of the signal with the matched filter is performed. As a result, a fully continuous received signal allowing variation of the sample point in the sample and hold unit as well as monitoring the signal in form of its eye pattern is simulated.

4.3.5.3 Eye Pattern Display

To monitor the eye pattern, the complete received signal stream at the input of the sample and hold unit is copied into an array which is organized as an endless wrap around array (Figure 4.10).



Y is the number of signals displayed in eye pattern window

FIGURE 4.10: Organization of eye pattern array

The size of the eye pattern array is a function of the number of signals that are displayed as well as the number of eyes in the window. As a result, for most cases the array size will not be a multiple of a signal block size. Therefore, the beginning of the most recent signal in the array varies during a simulation. A pointer is passed into the graphing routine *Grapheye* which points to the beginning of the most recent signal block. The graphing routine uses this reference point to distinguish between previous and current signals (Figure 4.10). If valid, old signals from previous signal blocks are present, they are graphed, in grey, in the background. Then the current signals are drawn on top of the old signals in yellow. This way, the signals which relate to the most recent simulated block can be always distinguished from signals of previous blocks. At the beginning of each channel file, no valid old signals are present. Therefore, no old signals are monitored whenever channel files are switched.

4.4 Summary

the software implementation of BERSIM has been explained in this chapter. First, an overview describing the interaction of the four independent executable programs *intface*, *tworay*, *extern*, and *tsimul* was given. Second, the frequently used memory allocation technique for two-dimensional arrays was introduced. Finally, the implementation of each program was explained and limitations for certain variables were shown. Most limitations result from the limited available memory on a personal computer in combination with the operating system MS-DOS.

5. Simulations and Results

The simulations performed for this work can be divided into two groups. One group is the set of simulations that were performed to verify that the current version of BERSIM produces results which match analytically derived bit error rates. The other group of simulations were performed to analyze bit error rate mechanisms as well as the impact of the significant program modifications on the performance of the simulator.

5.1 Verification of Program Operation

In order to ensure that the program is producing correct results, several simulations performed in [Fun91] were repeated and compared with the literature. This way, a good comparison between BERSIM 1.0 and BERSIM 2.0 with respect to published work could be made. Additional simulations were performed to verify that the simulator was functioning correctly.

5.1.1 Additive White Gaussian Noise Channel

To test for correct functioning of the transmitter and receiver modules, a simulation using a pure white Gaussian noise channel was performed. Figure 5.1 shows that the simulation results follow the theoretical results given in [Cou90] [Pro89] very closely. It has been shown in [Fun91] that for small roll-off factors, a discrepancy at high Eb/No occurs. This is caused the implementation of the square root raised cosine filters.

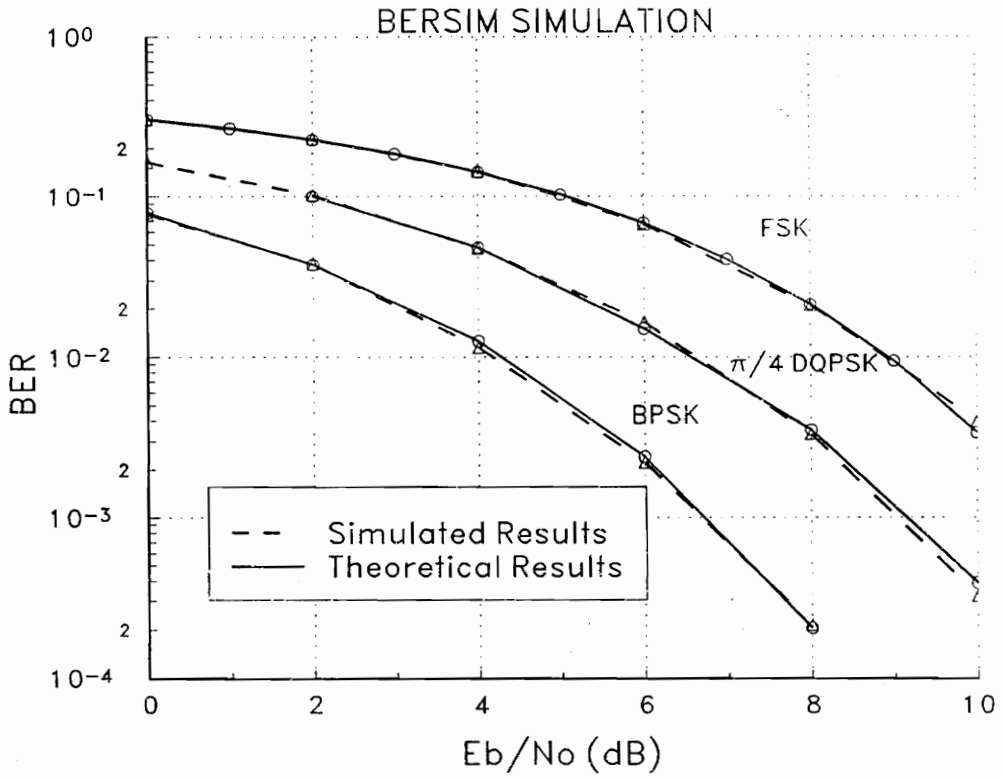


FIGURE 5.1: BER vs. Eb/No for a AWGN channel for π/4 DQPSK, BPSK, and FSK compared with theoretical results.
The symbols are shaped using a raised cosine filter with a roll-off factor α = 1.0.

5.1.2 Rayleigh Flat Fading Channel

In order to check the simulator in combination with flat fading channels, the predicted bit error rates were compared to analytical results calculated by Liu and Feher [Liu 89] [Liu 91]. Their analytical results seemed to be very suitable for comparison because they analyze a system that is very similar to the π/4 DQPSK system implemented in BERSIM. The only significant difference is that Liu considered a system which used a raised cosine filter at the transmitter side and a brick-wall filter with bandwidth equal to $(1+\alpha)f_N + f_d$ at the receiver end, where f_d is the maximum doppler frequency and f_N is the Nyquist frequency of the signal (symbol rate/2).

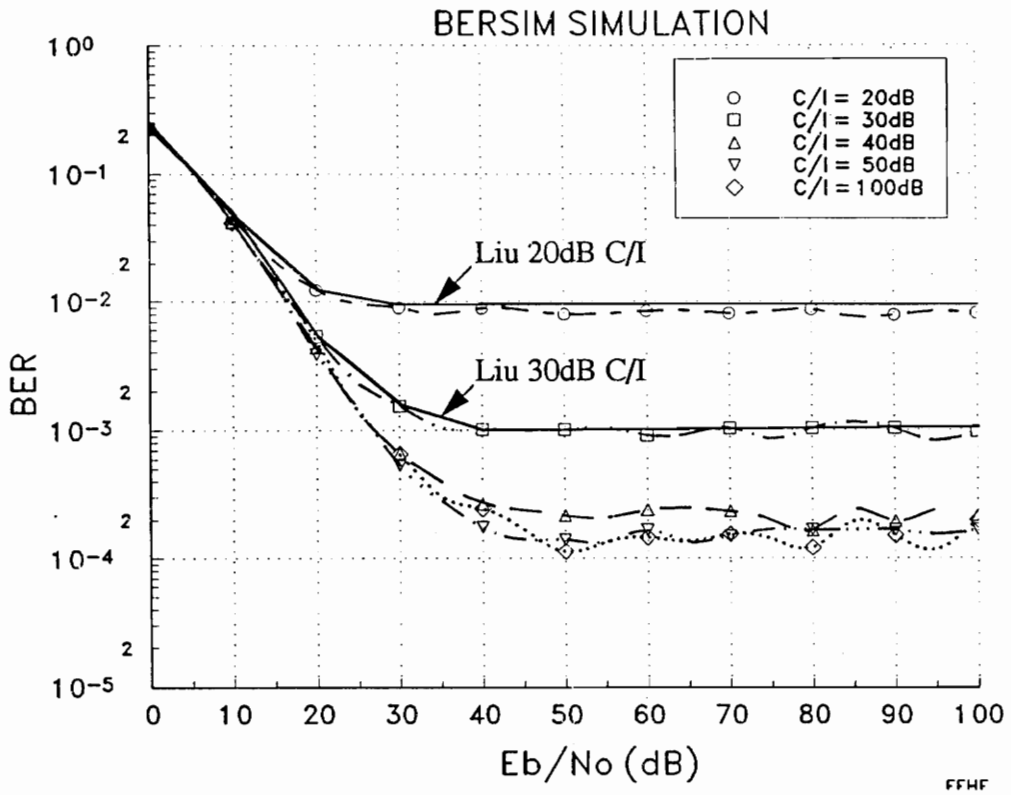


FIGURE 5.2: BER vs. Eb/No of $\pi/4$ DQPSK in a flat Rayleigh fading channel corrupted by AWGN and cochannel interference. fc = 850 MHz, fs=24.300 kBd, α = 0.2, v = 50 km/h.

Figure 5.2 and Figure 5.3 compare their analytical results for a flat fading channel with results obtained by the simulator BERSIM 2.0. Both figures illustrate the impact of interference on the bit error rate. It can be seen that BERSIM predicts the same bit error rates as calculated when the cochannel interference is kept constant and the signal-to-noise ratio (Eb/No) is varied (Figure 5.2). For the simulation where the signal to noise ratio is kept constant, a slightly better bit error rate than determined by Liu is predicted. However, the reason for this small difference could be the different filter implementation at the receiver in Liu's model.

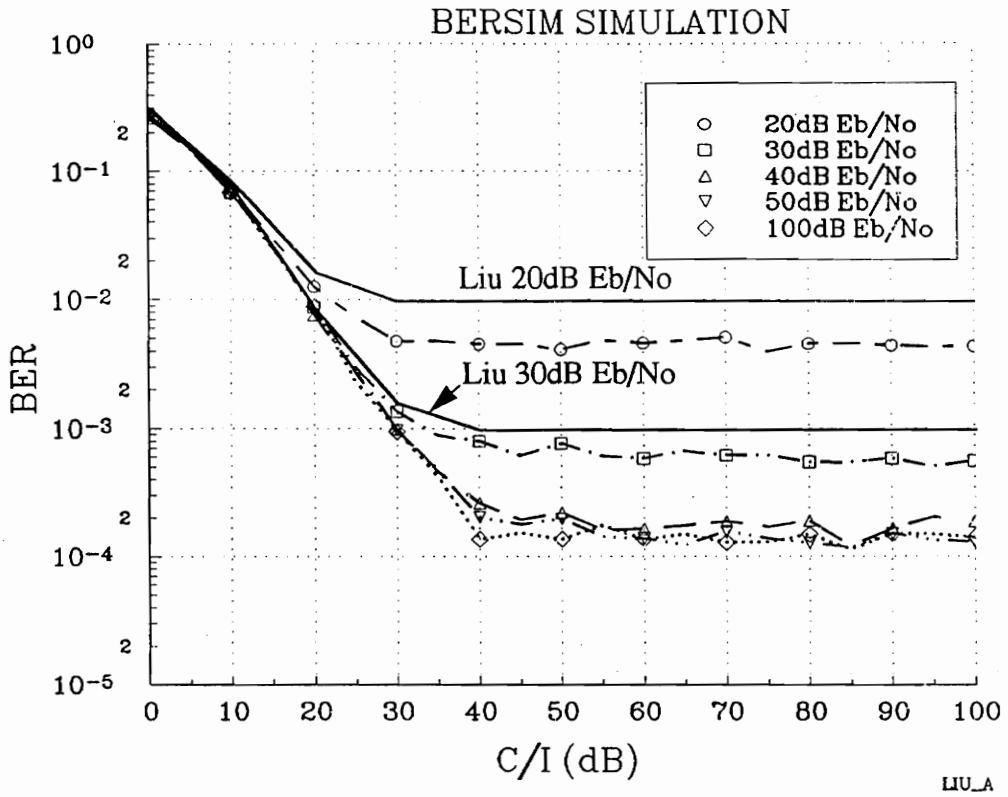


FIGURE 5.3: BER vs. C/I of $\pi/4$ DQPSK in a flat Rayleigh fading channel corrupted by AWGN and cochannel interference. fc = 850 MHz, fs = 24.300 kBd, α = 0.2, v = 50 km/h.

The bit error floor exhibited in Figure 5.2 and Figure 5.3 is a result of the mobile speed of 50 km/h that has been simulated for these two figures.

A simulation of a flat fading channel for various mobile speeds (Figure 5.4), has been done to explore this mobile speed dependent bit error floor. This error floor is due to random FM which is time derivative of the received phase [Lee82][Jak74]. If the mobile moves fast, the phase between successive symbol detections changes rapidly and a false detection occurs. Therefore, the error floor due to random FM is a function of the maximum doppler frequency and the symbol rate.

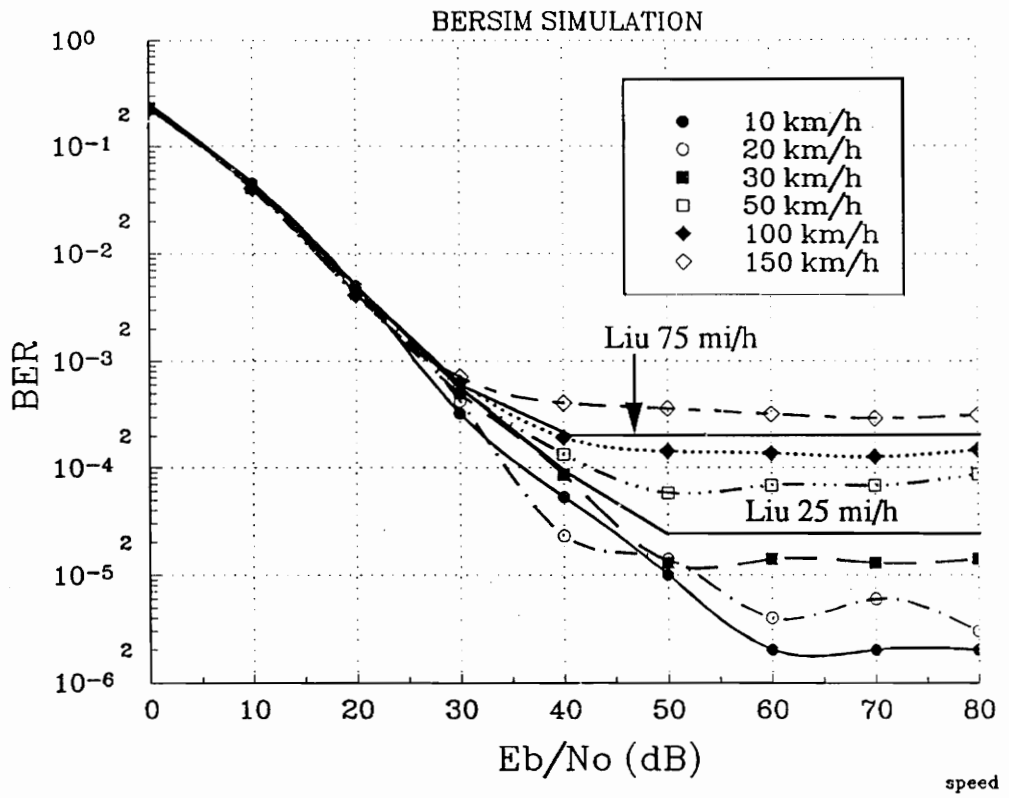


FIGURE 5.4: BER vs. Eb/No in a flat Rayleigh fading channel for various mobile speeds. fc = 850 MHz, fs = 24.300 kBd, α = 0.2, C/I = 100 dB.

Figure 5.4 shows the influence of random FM and compares with results calculated by [Liu91]. Again, it can be seen that calculated and simulated results match well. Figure 5.4 also shows that, up to the point where a constant error floor due to random FM is experienced, the bit error rate is almost independent of the speed of the mobile. For low Eb/No ratios, the bit errors are dominated by noise, whereas for high Eb/No ratios the bit errors rate are dominated by random FM and therefore depend on the speed of the mobile.

5.1.3 Frequency Selective Fading Channels

To verify BERSIM 2.0 is generating correct results for frequency selective fading channels, a simulation similar to an analytically calculated frequency selective fading model by [Liu91] was performed. For this simulation, a two-ray Rayleigh fading envelope was used and the power ratio between the first and second ray was varied over a range of 80 dB (Figure 5.5, Figure 5.6).

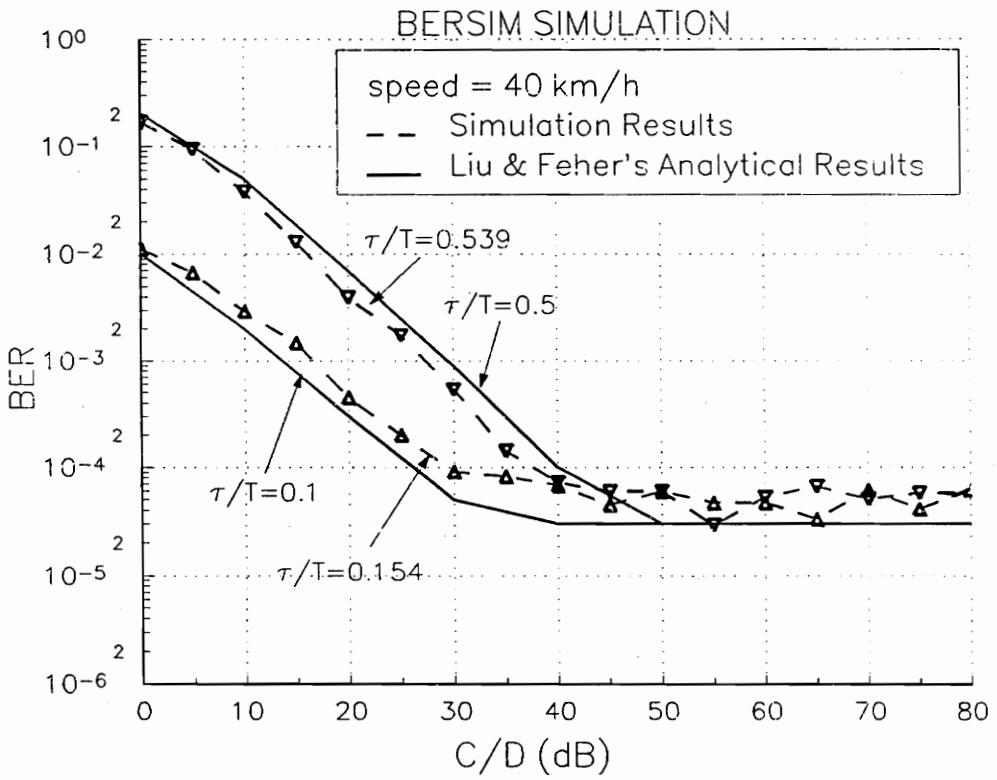


FIGURE 5.5: BER vs. C/D of π/4 DQPSK in a frequency selective two-ray Rayleigh fading channel for two different signal delays.
Eb/No = 100 dB, C/I = 100 dB, fc = 850 MHz, fs = 24.300 kBd, α = 0.2, v = 40 km/h.

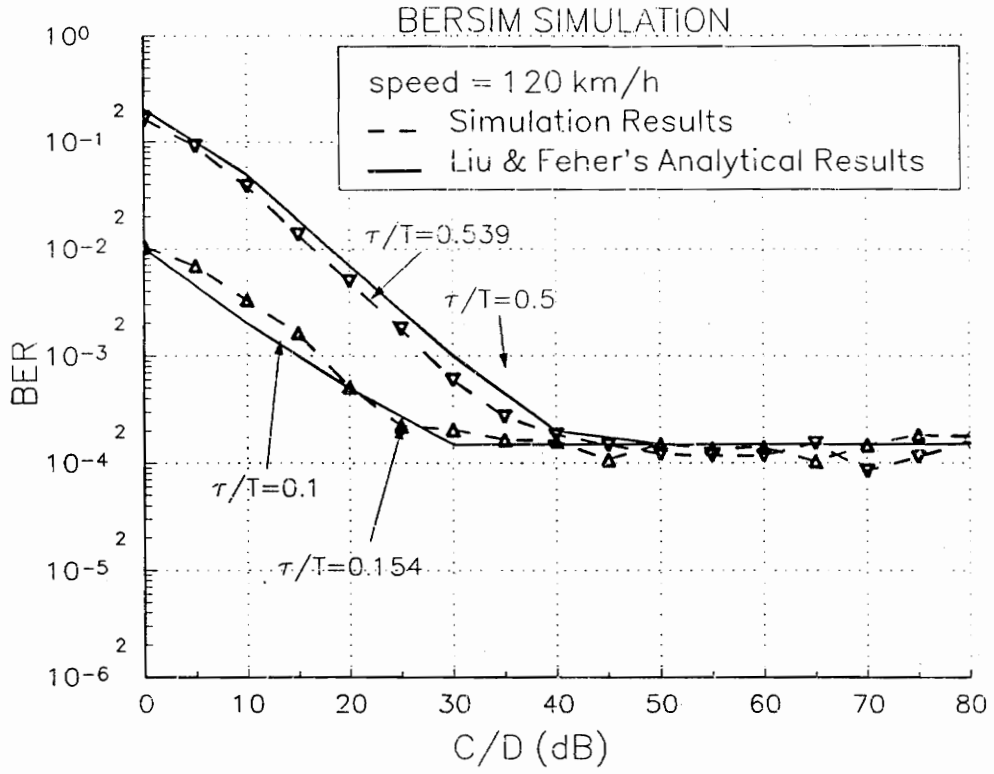


FIGURE 5.6: BER vs. C/D of π/4 DQPSK in a frequency selective two-ray Rayleigh fading channel for two different signal delays.
Eb/No = 100 dB, C/I = 100 dB, fc = 850 MHz, fs = 24.300 kBd, α = 0.2, v = 120 km/h.

Two different delays between the rays were selected similar to the delays used in [Liu] and two sets of simulations for two different doppler shifts were performed.

For the simulation of a mobile speed of 120 km/h the simulated and calculated results match well, whereas for 40 km/h the BERSIM results predict a slightly lower bit error floor than determined by Liu. It can be seen that for small C/D ratios ISI dominates the bit error rate. For larger C/D ratios, the channel becomes a flat fading channel and speed dependent error floors are observed. As mentioned before, the error floors are a result of random FM which, in a flat fading channel, sets a lower bound on the bit error rate.

An additional verification of the simulated results was obtained by comparing the normalized rms delay spread, d, of a noise free channel defined as

$$d = \frac{\sigma_{RMS}}{T_{symbol}} \tag{5.1}$$

with results published by [Ohn90]. In this work, as in [Chu87a], the impact of rms delay spread using two-ray Rayleigh faded channel models was studied. As was shown in Chapter 3.2.1.2, BERSIM does not allow variation of the delay between the two rays to occur in sufficiently small increments. A fixed delay of one symbol duration was chosen and the d ratio was adjusted by varying the power ratio between the first and the second ray. Figure 5.7 compares these simulated results with analytical results found by Ohno and simulation results found by Chuang [Chu87a] using a Gaussian shaped delay profile and QPSK modulation with coherent detection.

Based on the previous results that the Rayleigh envelope based "two-ray" part of the program BERSIM 2.0 functions correctly, a channel generated by the module *tworay* was modified, so that it could be read like a converted external channel. Running the module *tsimul* in the "external" mode using this converted channel file produced exactly equivalent results. This verified that the simulator works also for imported channels, provided that the channel conversion is done correctly.

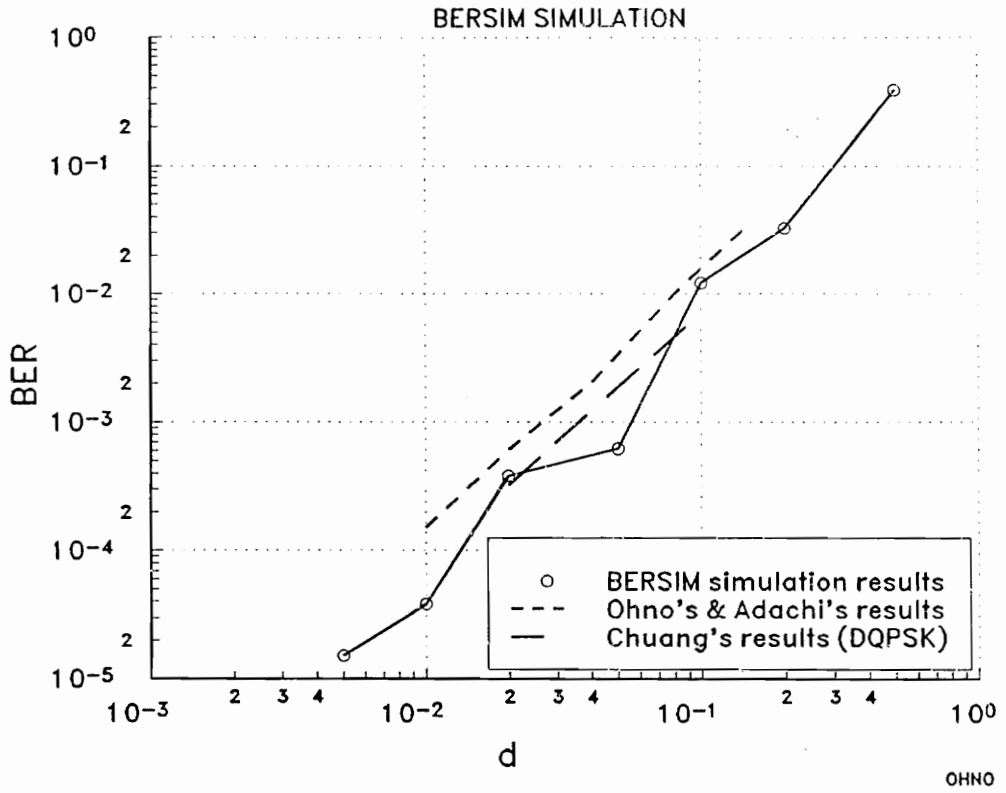


FIGURE 5.7: BER vs. d for $\pi/4$ DQPSK in a frequency selective two-ray Rayleigh fading channel. Eb/No = 100 dB, C/I = 100 dB, τ = 41 µs, fc = 850 MHz, fs = 24.300 kBd, α = 0.35, v = 1 m/s.

5.1.4 Conclusion

The bit error rate simulator has been shown to work correctly for internally generated Rayleigh fading channels. It has also been also proven, that if two Rayleigh fading envelopes are converted into a file format such as provided by the module *extern*, equivalent results to the two-ray simulation are obtained.

5.2 Timing Jitter Impact

Figure 5.8 shows the impact of simulated timing jitter for a nearly ideal channel (a stationary AWGN channel where the signal to noise ratio was set to 100 dB) on all implemented modulation techniques.

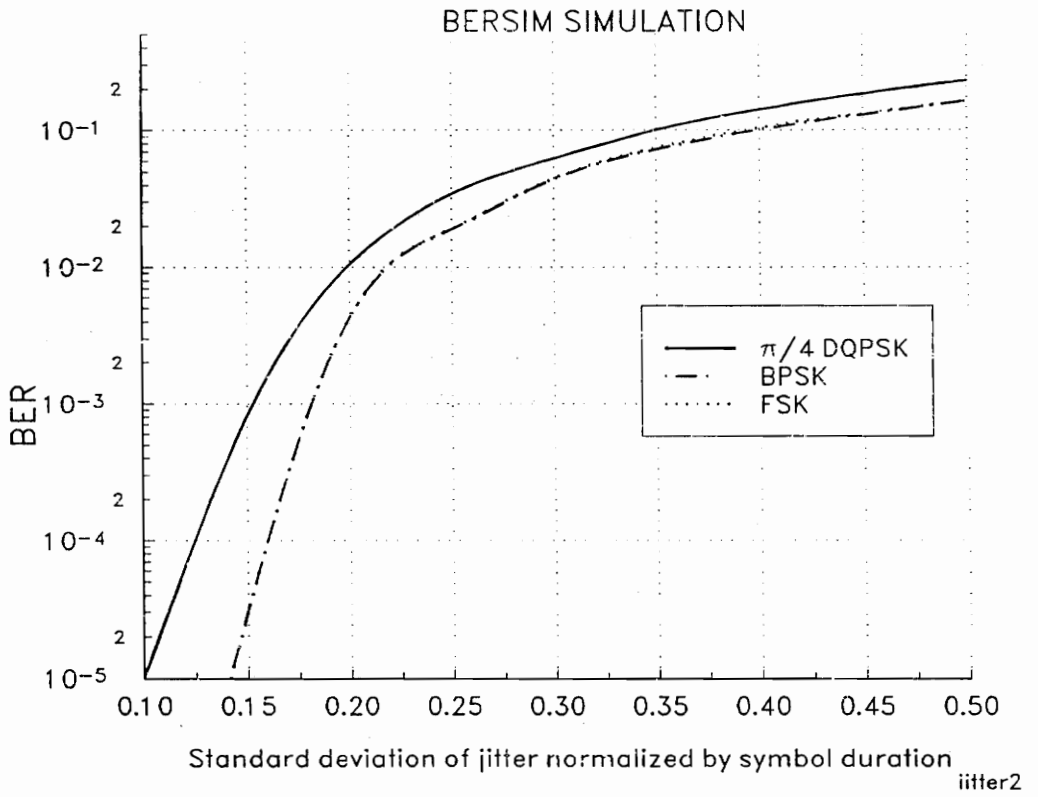


FIGURE 5.8: BER vs. σ_{jitter} for π/4 DQPSK, BPSK, and FSK in an ideal AWGN channel. Eb/No = 100 dB, C/I = 100 dB, α = 1.0.

With this method, it was ensured that the only error source is sampling errors at the receiver. Figure 5.8 can be used as a reference for a minimum error floor that will be introduced for any simulation if timing jitter is introduced.

To analyze the sensitivity of the three implemented modulation techniques to timing jitter, a set of simulations was performed for an AWGN channel with an Eb/No ratio of 20 dB.

As can be seen from Figure 5.8 and Figure 5.9, the BPSK and the FSK signal show identical sensitivity to timing jitter. Note that in Figure 5.1 it is shown that FSK performs poorer in a white Gaussian noise channel which accounts for the slight difference of FSK and BPSK for small σ. The modulation technique π/4 DQPSK seems to be most sensitive to timing jitter.

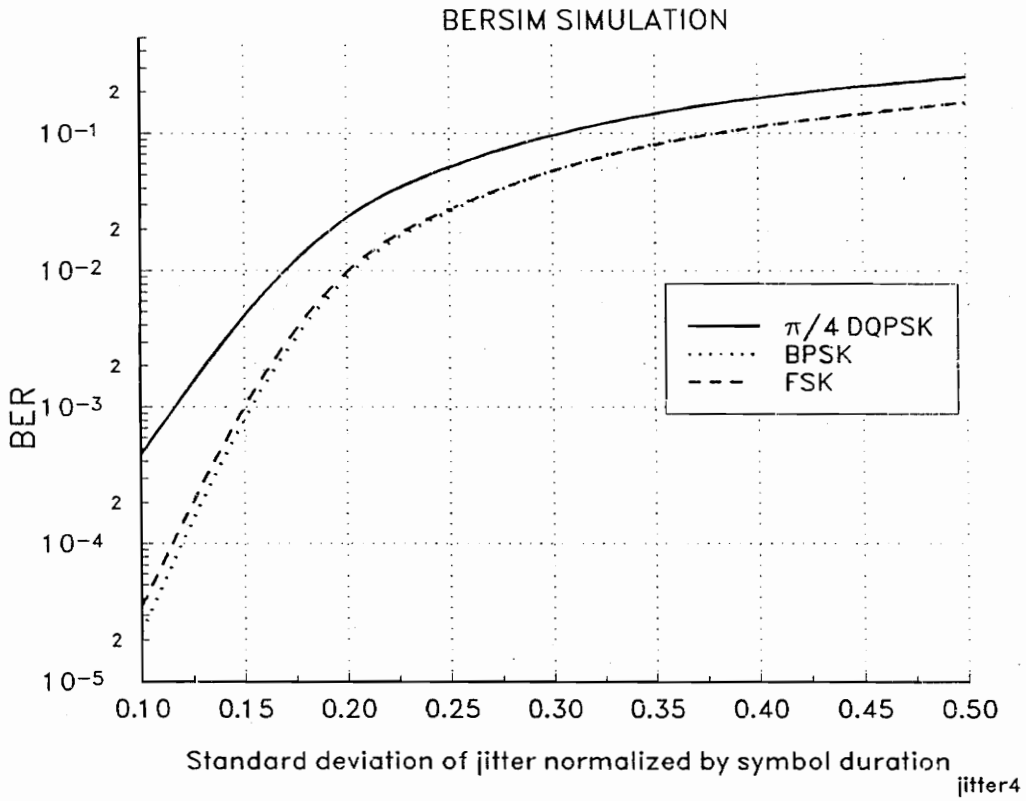


FIGURE 5.9: BER vs. σ_{jitter} for π/4 DQPSK, BPSK, FSK in an AWGN channel. Eb/No = 20 dB, C/I = 100 dB, α = 0.35.

5.2.1 Conclusion

The impact of timing jitter on the performance of the implemented receivers has been shown. It can be seen that π/4 DQPSK systems with an IF differential detector need to have accurate timing recovery because of their high sensitivity to incorrect sampling.

5.3 Simulation of Timing Recovery Methods

In Chapter 5.2, we discussed the impact of sampling at the wrong time instant by varying the sampling point. One of the factors that can cause incorrect sampling is multipath time delay jitter [Dev87]. In this chapter, we analyze the different timing recovery methods implemented in BERSIM 2.0. In order to analyze how each of the four timing recovery methods performs for a channel which introduces intersymbol interference, a set of simulations using a two-ray channel model was performed. For the simulations, the proposed US digital cellular modem specifications were used. The delay between the first and the second ray was set to 1/2 symbol duration. With this setup the power ratio between the two rays (C/D) was varied from -80 to 80 dB resulting in a variation of the rms delay spread. The maximum average rms delay spread of 1/4 of a symbol period is given for a C/D of 0 dB. This ensured that the simulation was performed within bounds, within which the squaring recovery loop can be modeled as suggested by [Chu87b].

Again, notice that Figure 5.10 exhibits error floors due to random FM at the tails of the function. It can be seen that in situations where the second ray is stronger than the first ray (large C/D), timing recovery is absolutely necessary. If no sample timing adjustment is used, incorrect bit error rates are predicted because the sample point is off by the delay between the two rays.

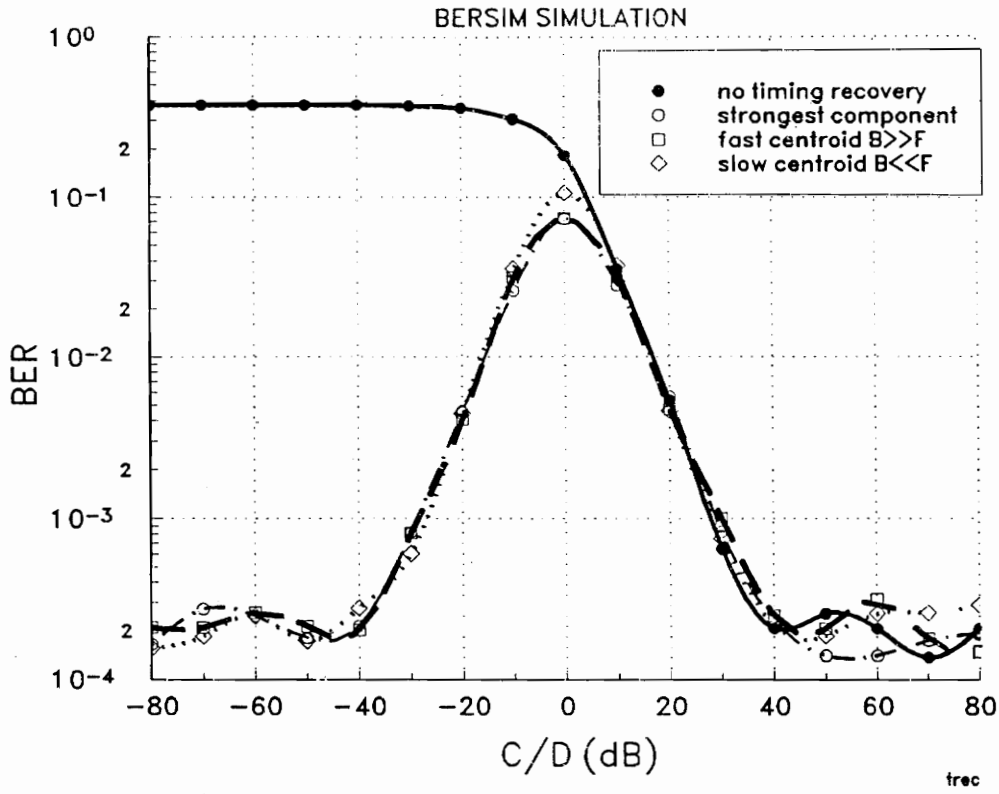


FIGURE 5.10: BER vs. C/D of various timing recovery models for π/4 DQPSK in a frequency selective two-ray Rayleigh fading.
Eb/No = 100 dB, C/I = 100 dB, fc = 850 MHz, fs = 24.300 kBd, α=0.35, v = 120 km/h.

A closer look at the simulated results for large rms delay spreads (Figure 5.11) shows that the impact of timing recovery can be neglected for simulations where the C/D ratio is larger that 15 dB. For the given two-ray model with a spacing of 1/4T, the normalized rms delay spread d becomes 0.08. This value matches results reported by Chuang [Chu87b], who found that the simulated timing recovery methods differ in performance only above this threshold. However, it has to be emphasized that whenever the first ray in a delay profile is not the strongest ray, timing recovery should be used with the simulator BERSIM. It can also be seen from Figure 5.11 that the fast squaring recovery loop performs almost as well as a system that would be able to track the strongest signal compo-

ponent. The slow squaring timing loop results can be considered the worst case tracking performance of a real-world timing recovery circuit (Chapter 3.3.5 [Chu87b]).

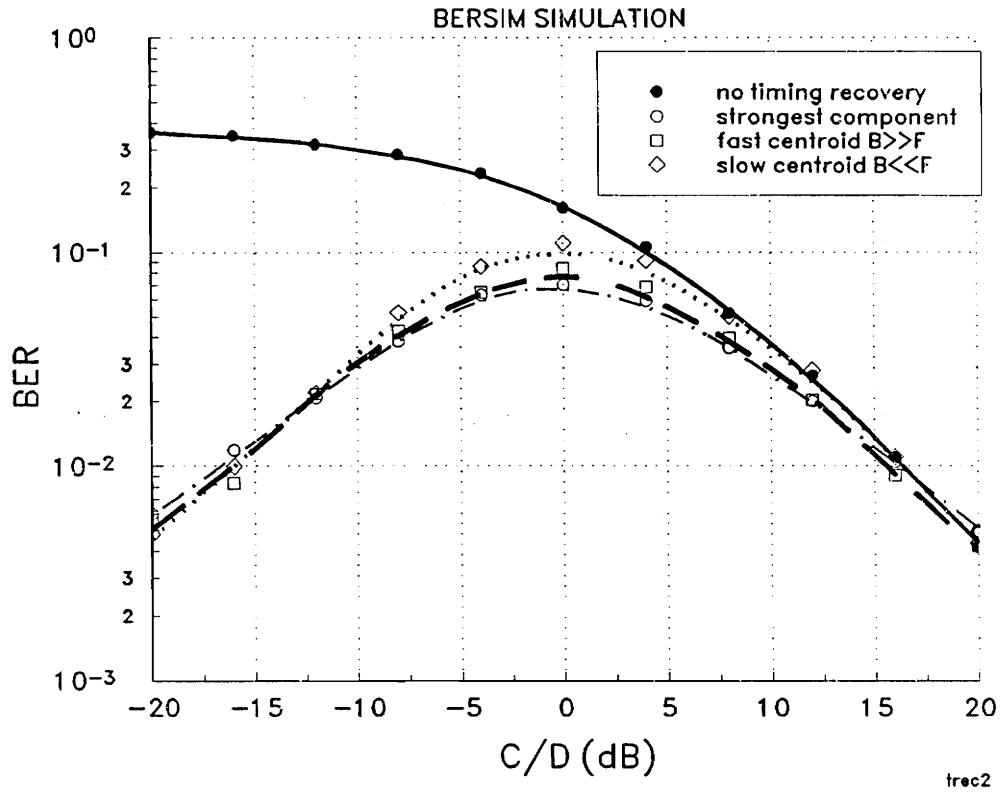


FIGURE 5.11: BER vs. C/D of various timing recovery models for π/4 DQPSK in a frequency selective two-ray Rayleigh fading.
Eb/No = 100 dB, C/I = 100 dB, fc = 850 MHz, fs=24.300 kBd, α = 0.35, v = 120 km/h.

5.3.1 Comparison of Timing Recovery Results for Simulated Real-World Channels

In order to investigate how the timing recovery schemes influence the predicted bit error rate for more realistic multi-ray channel impulse responses, the same channels and data rates used by Fung in [Fun91] [Fun92] are used here. The data rates simulated are based on proposed data rates for indoor microcell systems. The simulated data rates were 450 kbps as proposed by Bellcore [Cox91] and 1.152 Mbps as proposed for the European DECT [DEC89] system. A data rate of 2.000 Mbps is simulated to determine the perfor-

mance for even higher data rates. The modulation technique $\pi/4$ DQPSK is used for all simulated data rates for better comparison, even though other modulation techniques are proposed. The two channels used are channel models for an obstructed topography. They were generated with SIRCIM [Sei89][Rap91a]. SIRCIM simulates impulse responses based on statistical models which are developed based on extensive empirical research of the properties of indoor channel impulse responses. Figure 5.12 and Figure 5.13 describe the two used channels.

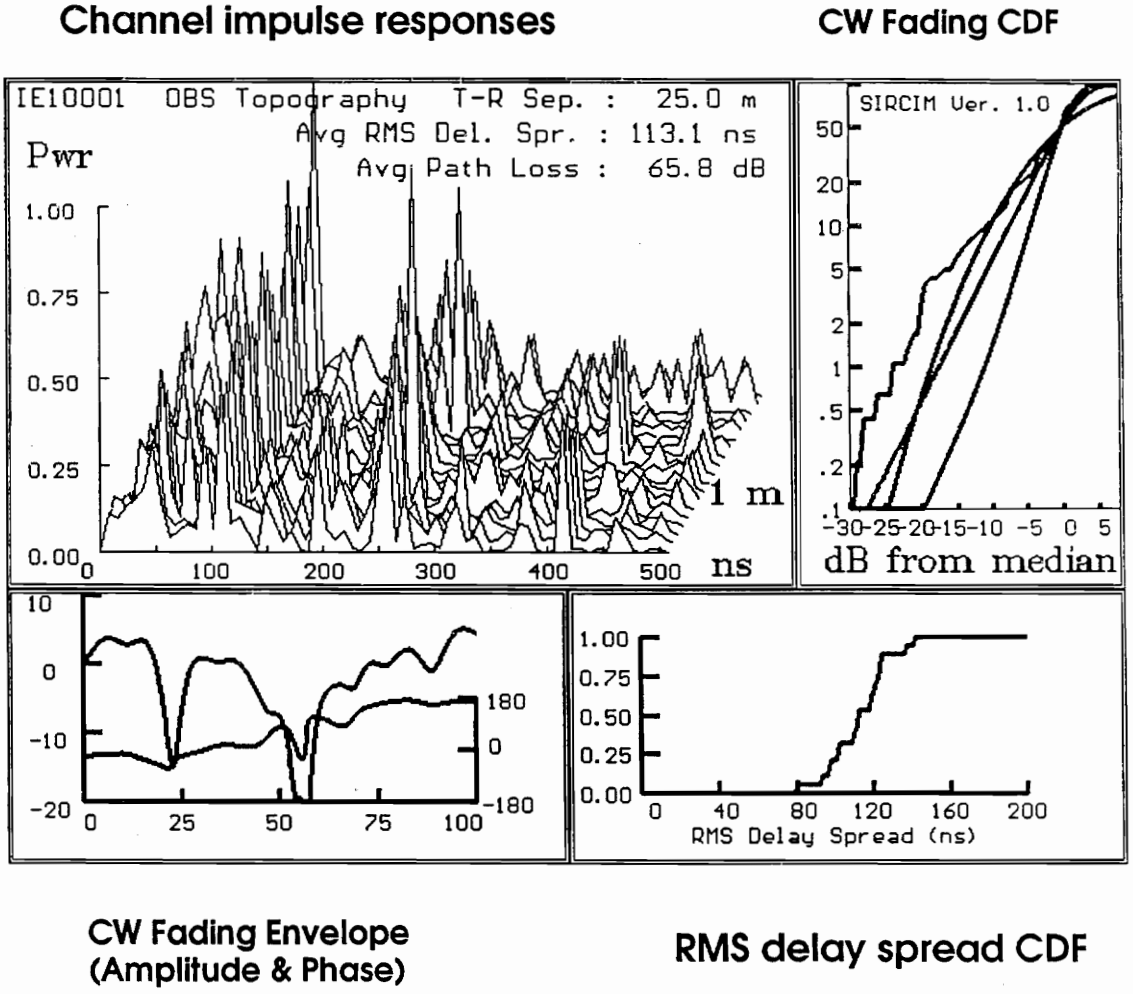


FIGURE 5.12: Statistical characteristics of the impulse response profile for channel 1 generated by SIRCIM.

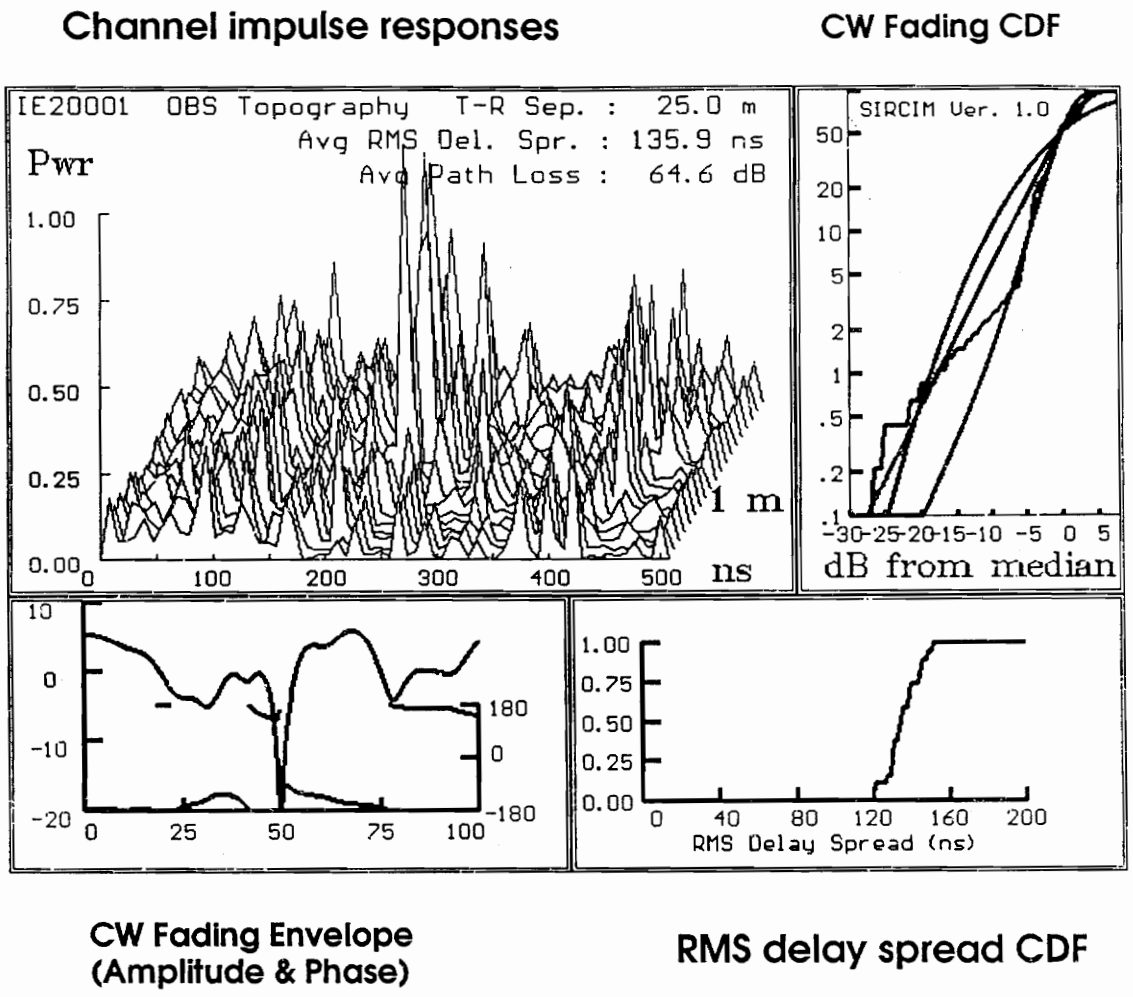


FIGURE 5.13: Statistical characteristics of the impulse response profile for channel 2 generated by SIRCIM.

A comparison of the simulation results obtained by using BERSIM 2.0 (Figure 5.15), with results obtained by BERSIM 1.0 (Figure 5.14) described in [Fun91], shows significant differences of the predicted bit error rates, even if no timing recovery is used (which is always the case in BERSIM 1.0). For example, the irreducible BER floor experienced for a data rate of 450kBps in both channels for BERSIM 1.0 is about 4 to 5 times higher than determined by BERSIM 2.0. On the other hand, the bit error rates predicted by BERSIM 1.0 for high data rates are lower than the bit error rates predicted by BERSIM 2.0. This

averaging effect, noticeable for BERSIM 1.0 results, is due to the incorrect file conversion methods used by BERSIM 1.0 as well as due to the incorrect channel gain normalization.

By studying the channel impulse responses given in Figure 5.14 and Figure 5.15 it becomes obvious that for high data rates, sampling the first arriving signal component does not model real-world receiver behavior. It can be expected that especially channel 2 introduces significant signal delay. For this reason, sample point adjustment by using timing recovery methods becomes necessary.

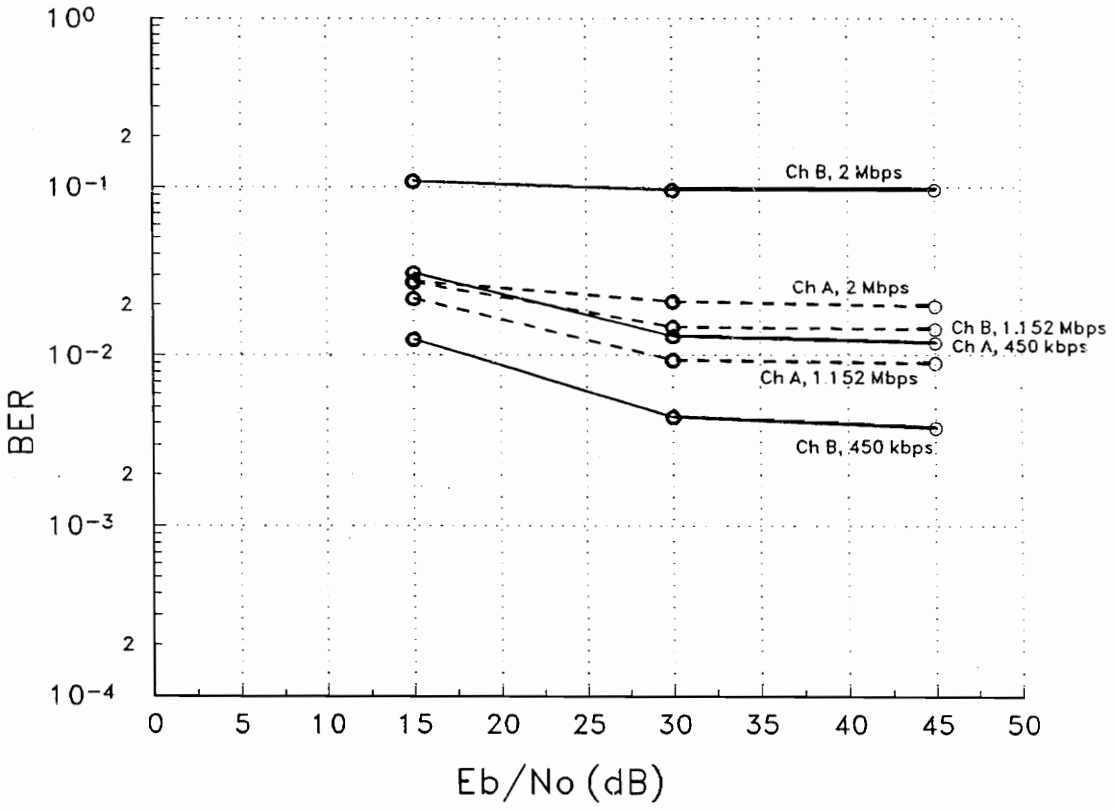


FIGURE 5.14: BER vs. Eb/No in Channel 1&2 for data rates of 450 kbps, 1,152 kbps, 2,000 kbps; Modulation $\pi/4$ DQPSK, v = 1 m/s, fc = 1300 MHz, α = 0.35, C/I = 100 dB; NO TIMING RECOVERY, BERSIM 1.0

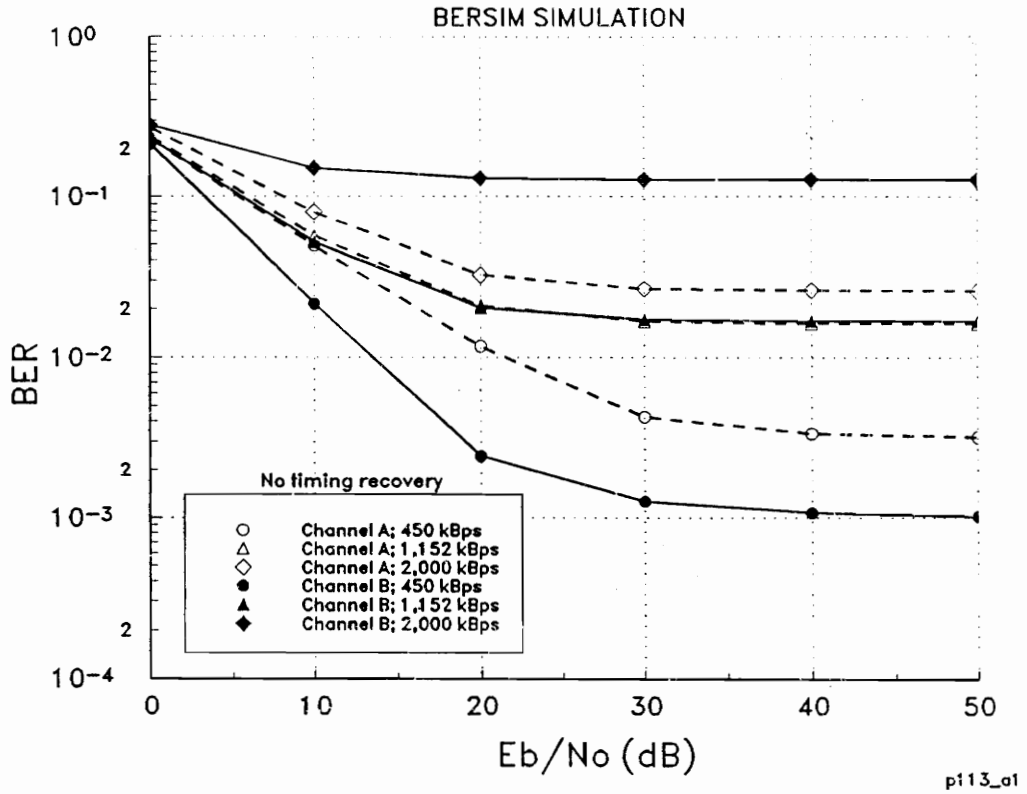


FIGURE 5.15: BER vs. Eb/No in Channel 1&2 for data rates of 450 kbps, 1,152 kbps, 2,000 kbps; Modulation $\pi/4$ DQPSK, v = 1 m/s, fc = 1300 MHz, α = 0.35, C/I = 100 dB; NO TIMING RECOVERY, BERSIM 2.0

Figure 5.16 shows that for the model which samples constantly at the centroid (determined over all channel impulse responses), the predicted bit error rates for channel 2 decrease significantly for high data rates. The predicted bit error rates for channel 1 decrease only slightly for high data rates in combination with the modeled slow squaring timing recovery loop.

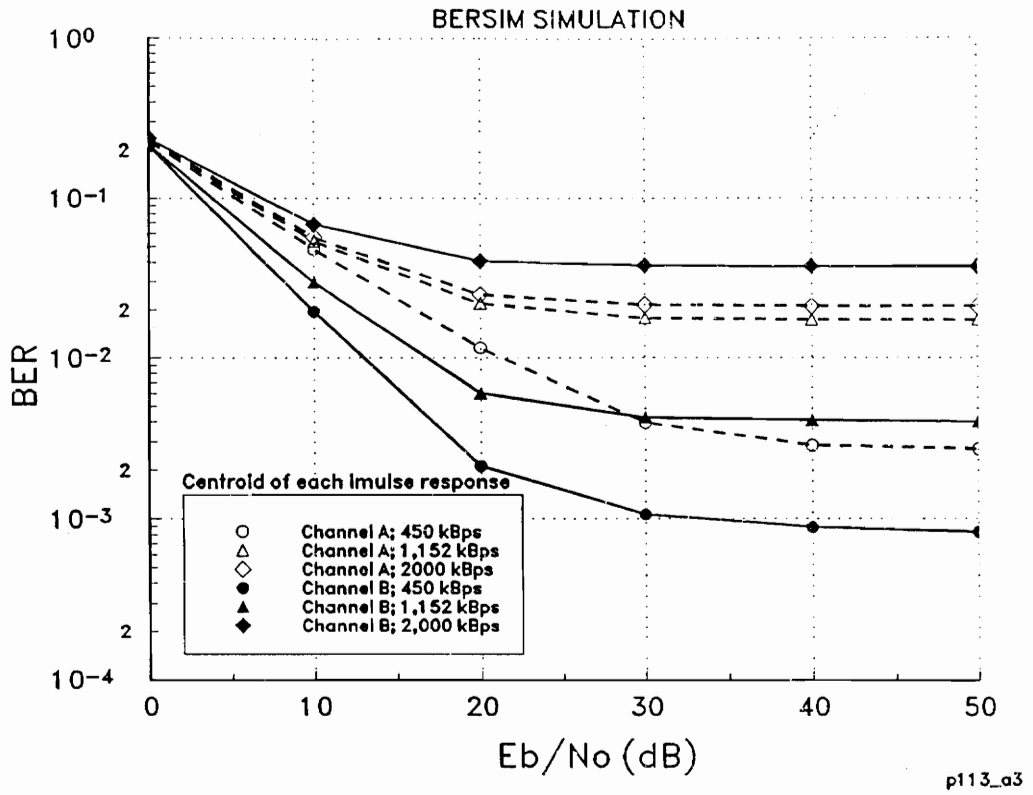


FIGURE 5.16: BER vs. Eb/No in Channel 1&2 for data rates of 450 kbps, 1,152 kbps, 2,000 kbps; Modulation $\pi/4$ DQPSK, v = 1m/s, fc = 1300 MHz, α = 0.35, C/I = 100 dB; SLOW TIMING RECOVERY LOOP (B << F).

If the model for the fast squaring timing recovery loop is used, an additional decrease of estimated bit error rates is experienced (Figure 5.17). Again the additional gain of transmission performance is higher for channel 2 than for channel 1. This time the timing recovery improves even the performance for the lower data rates. For these data rates, the channels can be considered almost as narrowband channels. After the file conversion, they are only modeled by two rays. As a result, timing recovery does not offer much performance improvement.

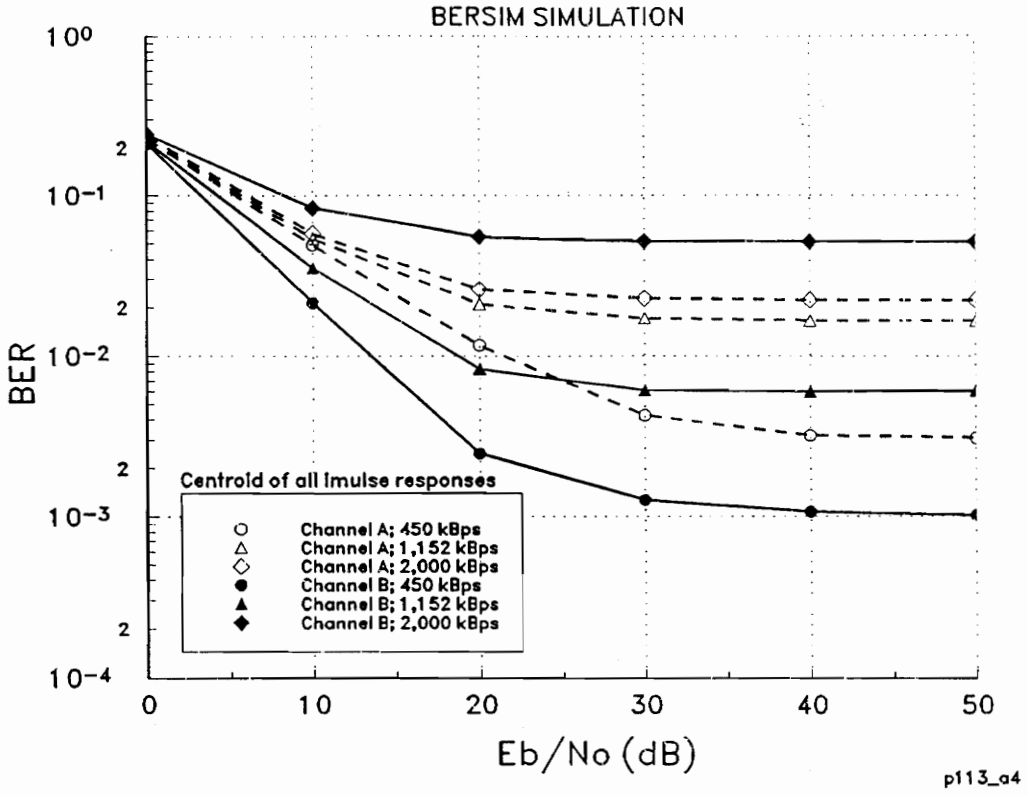


FIGURE 5.17: BER vs. Eb/No in Channel 1&2 for data rates of 450 kbps, 1,152 kbps, 2,000 kbps; Modulation π/4 DQPSK, v = 1 m/s, fc = 1300 MHz, α = 0.35, C/I = 100 dB; FAST TIMING RECOVERY LOOP (B >> F).

A comparison of the fast centroid timing recovery results with the results given by the strongest signal component tracking model (Figure 5.18) shows that the strongest signal component tracking model preforms well for channel 1. However, the performance is slightly poorer for high data rates in combination with channel 2. The reason for this could be that fast variations of the sampling point introduce artificial jitter into the differential phase detection. It can therefore be concluded that it is not necessarily desirable to develop a timing recovery scheme that is capable of tracking the strongest signal component.

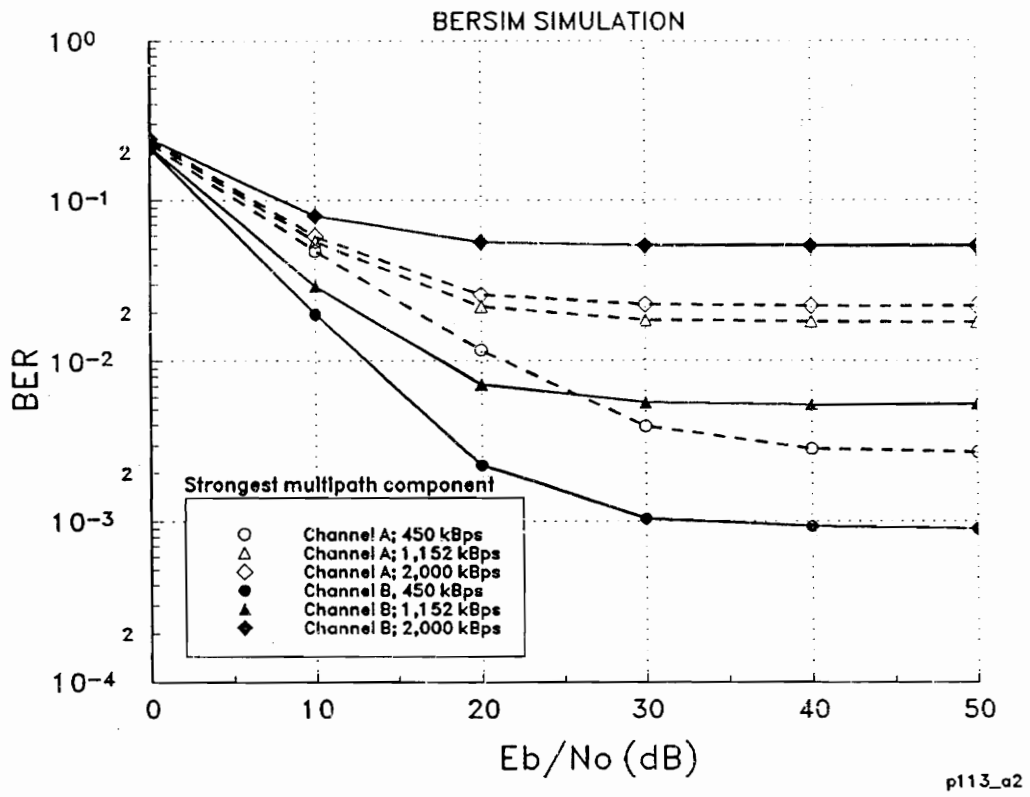


FIGURE 5.18: BER vs. Eb/No in Channel 1&2 for data rates of 450 kbps, 1,152 kbps, 2,000 kbps; Modulation $\pi/4$ DQPSK, v = 1 m/s, fc = 1300 MHz, α = 0.35, C/I = 100 dB; RECEIVER TRACKS STRONGEST SIGNAL COMPONENT.

5.3.2 Conclusion

It has been found that the timing recovery schemes can improve the prediction of bit error rates for d' larger than 0.05. The model of a slow squaring timing loop can be used to model a worst case timing recovery, whereas the maximum component and the fast centroid method can be used to model a effective timing recovery. No timing recovery, as is done in BERSIM 1.0, should only be used for simulations where the first arriving signal component is the strongest and the normalized rms delay spread, d' is less than 0.05.

5.4 Simulation of Outage in Interference Dominated and White Gaussian Noise Dominated Systems

For modern mobile radio systems, outage probability is a more important figure of merit than the raw bit error rate. Figure 5.19 shows outage rates predicted for several mobile speeds determined for a flat fading channel. The outage rates were simulated under exactly the same conditions as the bit error rates shown in Figure 5.4. The outage frame size was set to 324 bits which is the proposed frame size for the US digital cellular system [EIA89]. An outage criterion of 1% is selected (3 bits in error/per frame is tolerable).

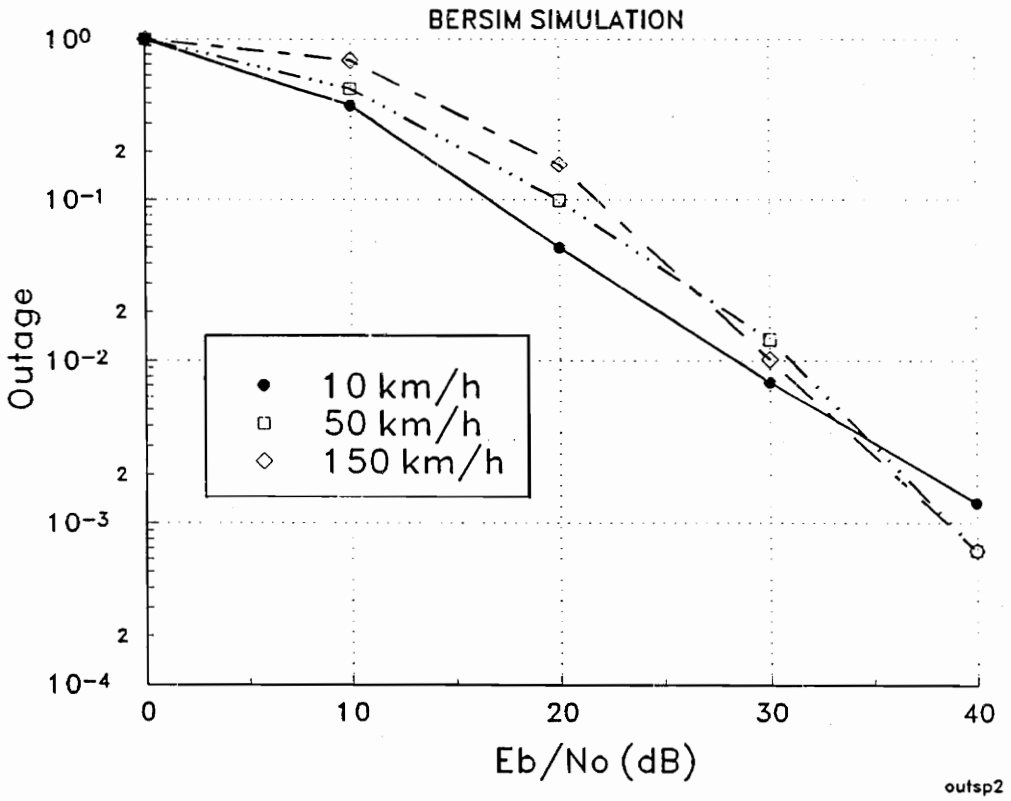


FIGURE 5.19: Outage vs. Eb/No for π/4 DQPSK in a flat Rayleigh fading channel for various mobile speeds. fc = 850 MHz, fs = 24.300 kBd, α = 0.35, C/I = 100 dB, Outage frame size 324 bit, Outage criterion 1%.

For the noise limited channel (Figure 5.19), outage probability proved to be almost independent of the mobile speed. However, note that the outage rate for slow vehicle speeds and small signal to noise ratios is slightly lower than for high vehicle speeds. If the signal to noise ratio is increased, a crossover point can be found after which a faster mobile experiences less outage for the same given signal to noise ratio. This is due to the fact that the time needed to pass through a fade becomes shorter when the mobile moves faster. Error bursts become shorter but occur more frequently.

For signal to noise ratios larger than 40 dB, fades that reach the noise level are generally not long enough (within the considered range of velocities) to cause enough errors within a frame period. For high speeds, bit errors introduced by random FM also occur only in very short bursts, which are usually not long enough to exceed the selected error threshold. Successive bursts do usually occur in different frames. Therefore, a system that can tolerate up to 3 errors per frame in combination with a flat fading channel is more noise limited than limited by other error mechanisms. The signal to noise ratio, however, can be controlled by the system using power control. The big disadvantage of increasing the signal to noise ratio by increasing the signal power is an increase of cochannel interference experienced by other cellular radio cells reusing the same frequency.

Figure 5.20 shows the relationship between cochannel interference and outage probability for the situation where cochannel interference dominates the system. The only significant difference when compared with a white Gaussian noise dominated system is that outage decreases more rapidly with speed for cochannel interference ratios around 30 dB. However, a system cannot be designed to work only above a certain vehicle speed, which means cochannel interference limits the overall performance of a system.

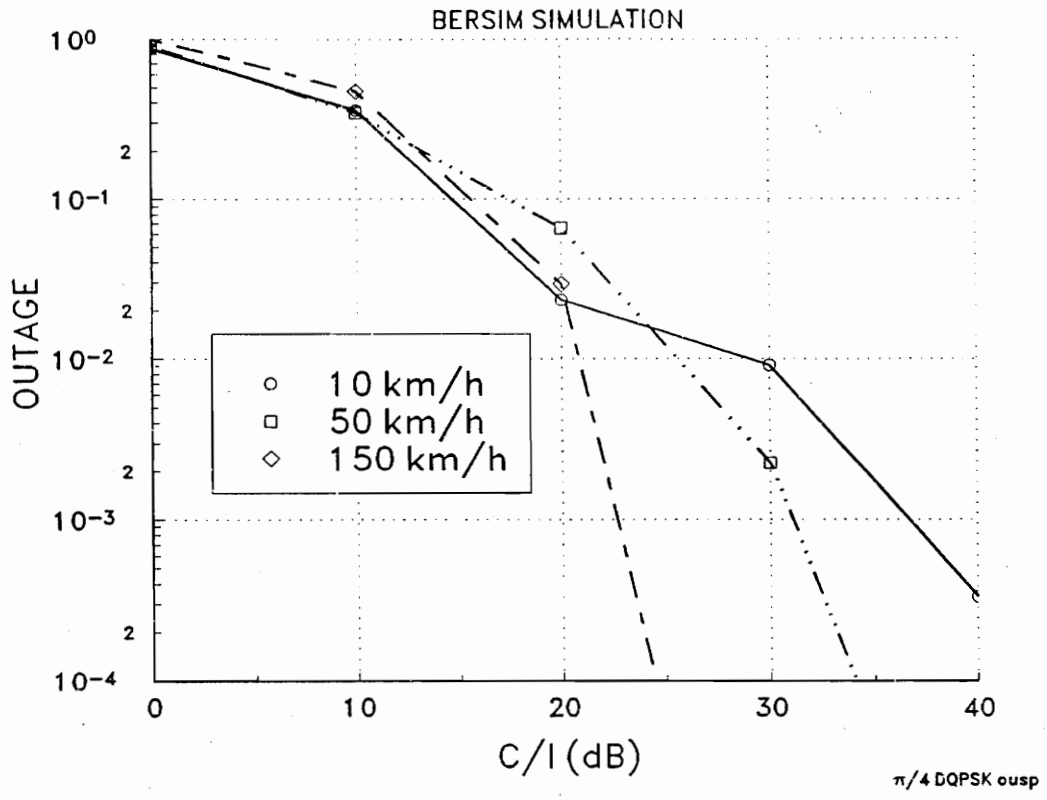


FIGURE 5.20: Outage vs. C/I for π/4 DQPSK in a flat Rayleigh fading channel for various mobile speeds. fc = 850MHz, fs = 24.300 kBd, α = 0.35, Eb/No = 100 dB, Outage frame size 324 bit, Outage criterion 1%.

The maximum tolerable cochannel interference automatically puts constraints on the maximum achievable signal to noise/interference ratio. Systems that are designed under those constraints are called interference limited systems.

5.4.1 Conclusion

The outage determined for a noise dominated system varies only significantly for higher speeds from the outage determined for a interference dominated system. For the outage criterion of 1%, outage is caused almost always by noise or interference. The outage rate does not have a fixed floor due to random FM as the bit error rate.

5.5 Simulation to Determine the Importance of the Shape of the Channel Impulse Response and the Influence of the Phase Distribution.

5.5.1 Comparison of Two Channels with Similar Spatially Averaged RMS Delay Spread

The BER and outage simulation presented in this section utilize two indoor channel impulse response models produced by the statistical indoor propagation simulator SIR-CIM[Sei89]. Figure 5.21 and Figure 5.22 describe the channel impulse response. The upper left portion of the figures show typical spatially-varying LOS channel multipath delay profiles (square magnitude impulse responses) which are realistic representations of field measured multipath profiles reported in [Rap91a].

As previously addressed in the function verification section of this thesis (Chapter 5.1), results for BERs and outage simulations of frequency selective fading channels where already presented in [Chu87a][Ohn89]. However, those studies simulated portable communication systems with fixed channel impulse responses that did not change shape, or had a fixed average shape.

Since this section focuses on the impact of multipath and frequency selective fading mechanisms, timing jitter is not considered. No timing recovery at the receiver is assumed, since LOS channels are used. The purpose of the simulations described in this section is not to predict absolute bit error rates, but to analyze bit error and outage sources.

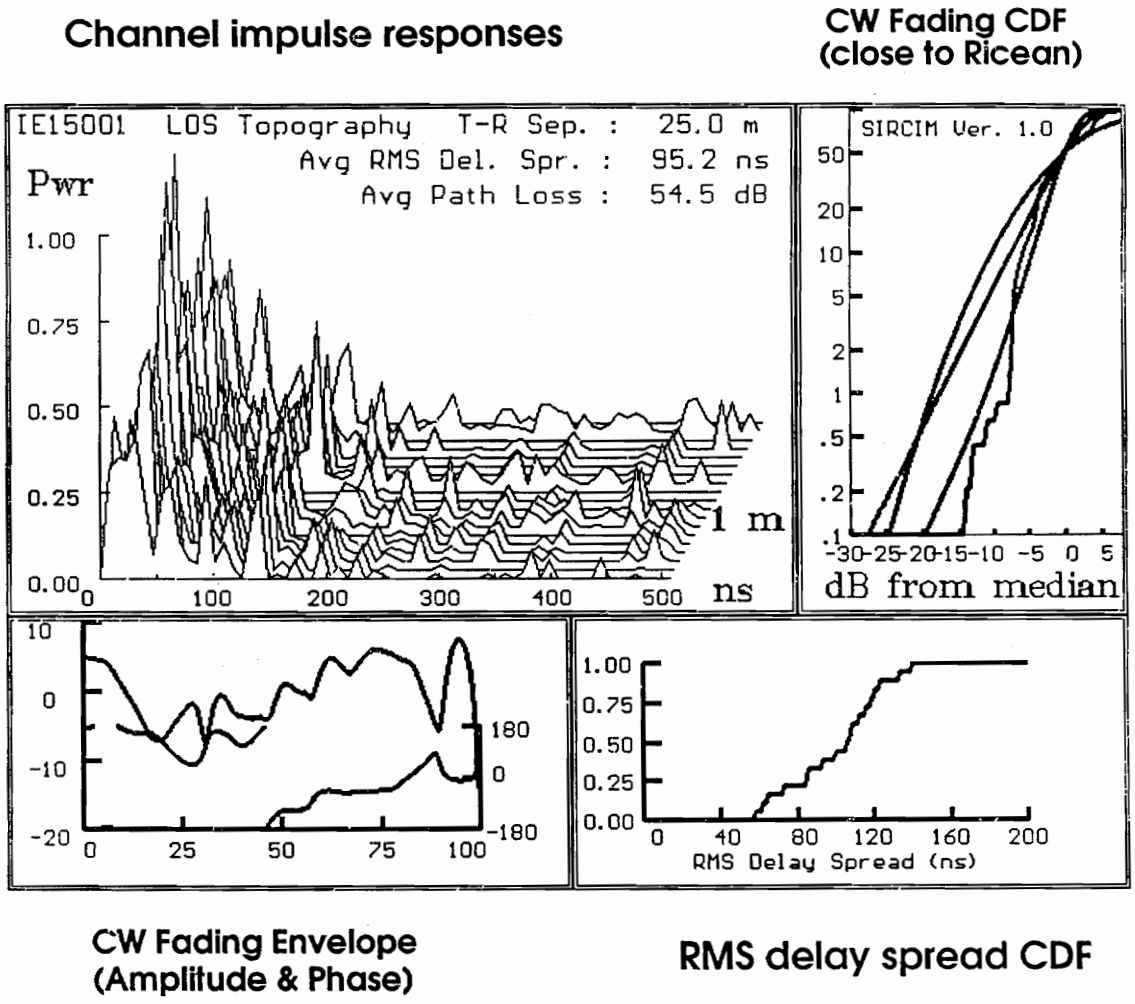


FIGURE 5.21: Statistical characteristics of the impulse response profile for channel A generated by SIRCIM.

The performance of a frequency-selective fading channel can be evaluated as a function of the normalized delay spread parameter d'. The normalized delay spread parameter d' is defined as:

$$d' = \frac{\sigma_{rms}}{T_b} \tag{5.2}$$

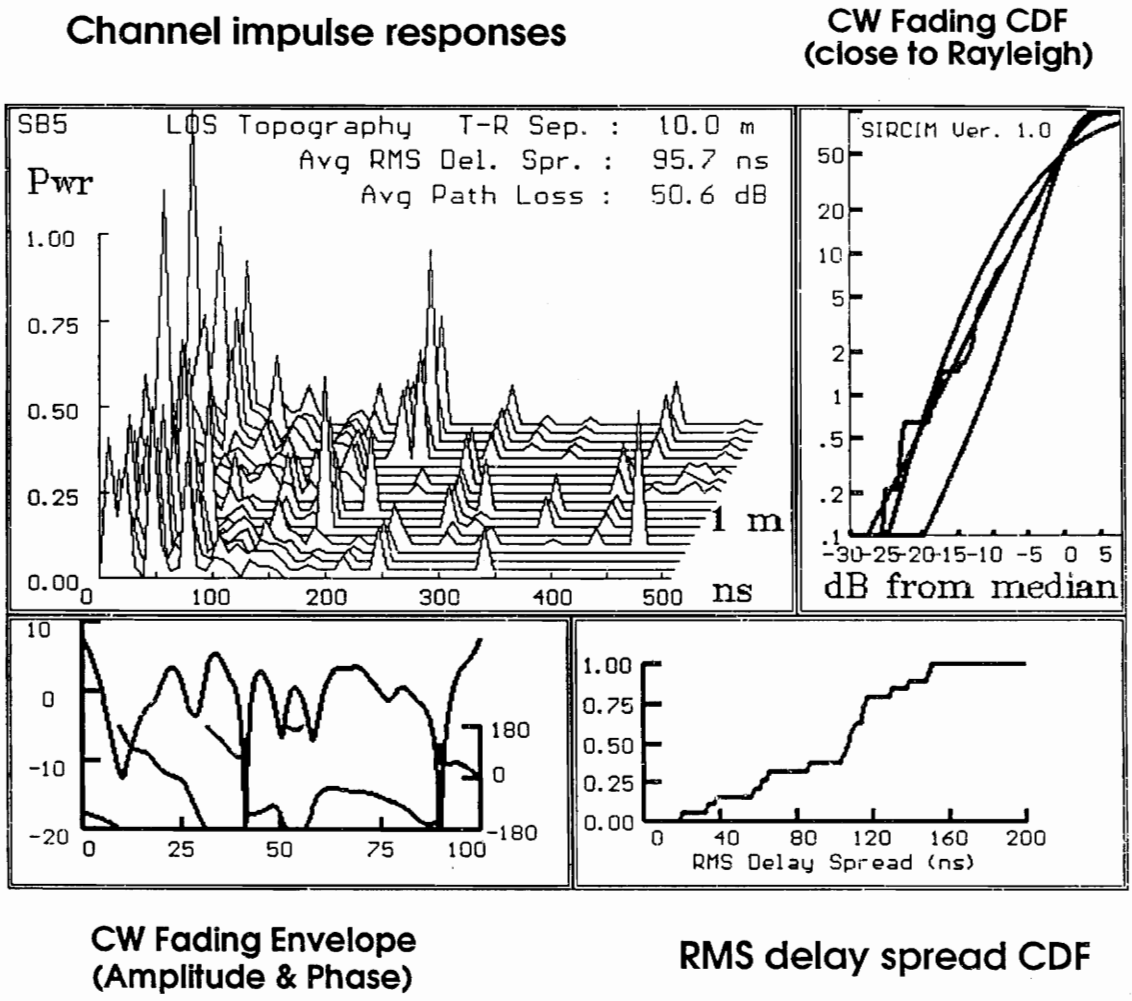


FIGURE 5.22: Statistical characteristics of the impulse response profile for channel B generated by SIRCIM.

Since SIRCIM models measured channels, the channel impulse response varies while the portable unit is moving. As a result, the statistics of the channel impulse response changes dynamically over a local area.

Both channels were initially selected under the following considerations:

- One channel should have Rayleigh narrowband fading statistics and the other should have Ricean narrowband fading statistics.
- Both channels should have similar rms delay spread distributions.
- Both channels should have almost the same average rms delay spread.

The average rms delay spread $\langle\sigma_{rms}\rangle$ is determined by the channel conversion program *extern* as the ensemble average (spatial average) of all power delay profiles used in the simulation. It was calculated as

$$\langle\sigma_{rms}\rangle = \frac{\sum_{n=1}^{N} \tau_n \sum_{m=1}^{M} p_m(\tau_n)}{\sum_{n=1}^{N} \sum_{m=1}^{M} p_m(\tau_n)} \tag{5.3}$$

where $p_m(\tau_n)$ represents the power magnitude of the multipath component with the delay τ_n of the power delay profile p_m. A carrier frequency of 1.3 GHz was selected for the simulated communication system. As a result, the 19 generated channel impulse responses spaced by a quarter of a wavelength represent exactly a distance of 1.04 m. At each location, the mobile is assumed to move along this 1.04 m track at a velocity of 1.04 m/s, which roughly corresponds to the walking speed of a portable radio user.

The transmitter-receiver (T-R) separation which is indicated in Figure 5.21 and Figure 5.22 is irrelevant for the simulations because the channel gains are set to unity as described in Chapter 3.2.2.3.

The two selected channels have a nearly identical spatially-averaged rms delay spreads of about 100 ns. Channel A (Figure 5.21) has a rms delay spread of 95.2 ns whereas channel

B (Figure 5.22) has a rms delay spread of 95.7 ns. As desired, the significant difference between the two channels lies in their narrowband (CW) fading characteristics, which results from the vectorial sum of all multipath components in a profile (Chapter 3.2.2). As seen in the upper right parts of Figure 5.21 and Figure 5.22, the flat-fading characteristic of channel A follows almost a Ricean distribution, whereas the flat-fading characteristic of channel B follows a more typical Rayleigh distribution.

Several simulations, each simulating 1 second of data transmission, were run to simulate outage and bit error rate for a portable unit moving along the two 1m tracks. The symbol rate was selected according to the channel's absolute rms delay spread and the selected normalized rms delay spreads d'. The outage frame length was set to 1024 bits, which is considered a reasonable TDMA frame length for high capacity indoor systems. The outage criterion was set to 1%, equivalent to 11 bit errors or more per frame. The roll-off factor of the implemented signal shaping filters was set to 0.35, which is the same as for the proposed US digital cellular standard [EIA89].

About 6000 channel impulse responses were interpolated over space from the original 19 channel impulse responses generated by SIRCIM, which means between 5 and 500 symbols were transmitted during each channel impulse response sample over space. The Eb/No ratio was varied as the channel noise parameter in the simulations. BER and outage performances of these two channels for $\pi/4$ DQPSK are shown in Figure 5.23 to Figure 5.26. For BPSK they are shown from Figure 5.27 to Figure 5.30, and for FSK they are shown from Figure 5.31 to Figure 5.34.

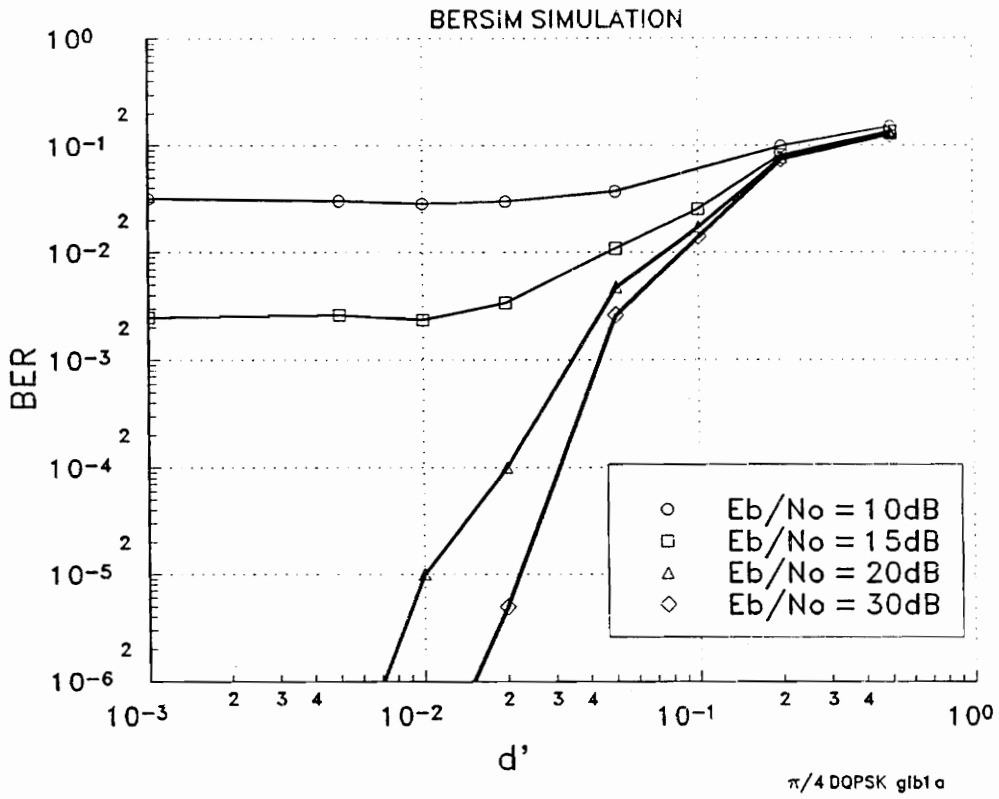


FIGURE 5.23: BER vs. d' for π/4 DQPSK in the frequency-selective fading channel A with almost Ricean narrowband fading statistics.
fc = 1300 MHz, Eb/No = 10-30 dB, α = 0.35, v = 1 m/s, C/I = 100 dB

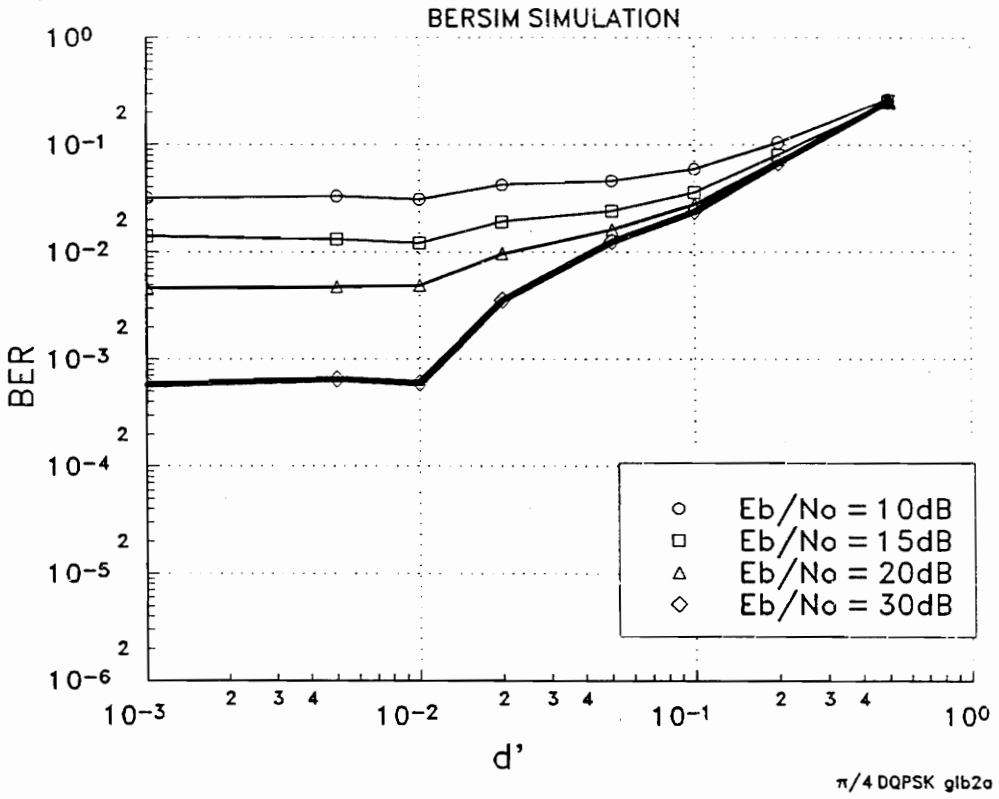


FIGURE 5.24: BER vs. d' for π/4 DQPSK in the frequency-selective fading channel B with almost Rayleigh narrowband fading statistics.
fc = 1300 MHz, Eb/No = 10-30 dB, α = 0.35, v = 1 m/s, C/I = 100 dB

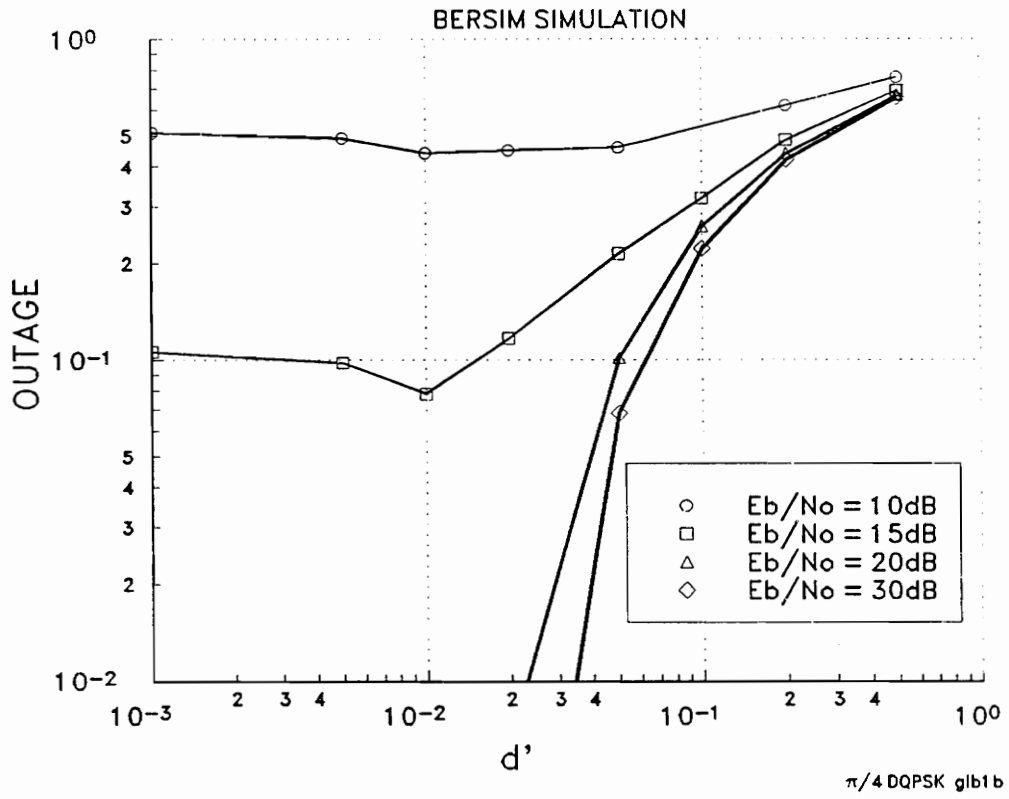


FIGURE 5.25: Outage vs. d' for π/4 DQPSK in the frequency-selective fading channel A with almost Ricean narrowband fading statistics.
Outage block size 1024 bit, Outage criterion 1%,
fc = 1300 MHz, Eb/No = 10-30 dB, α = 0.35, v = 1 m/s, C/I = 100 dB

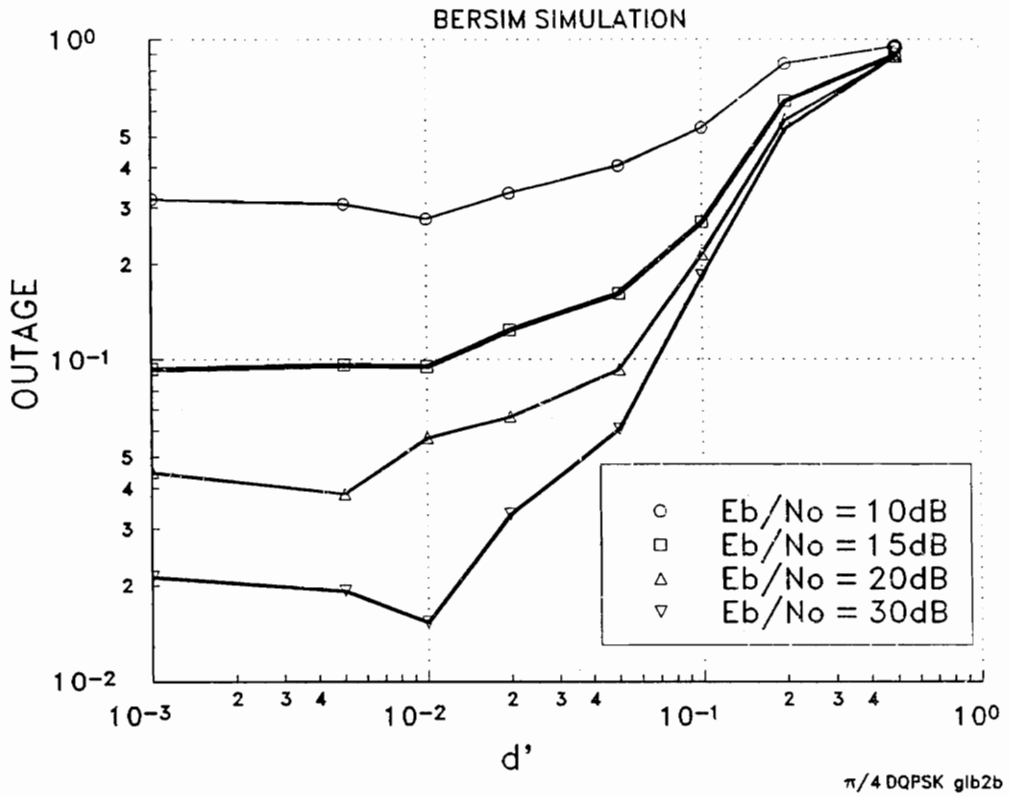


FIGURE 5.26: Outage vs. d' for π/4 DQPSK in the frequency-selective fading channel B with almost Rayleigh narrowband fading statistics.
Outage block size 1024 bit, Outage criterion 1%,
fc = 1300 MHz, Eb/No = 10-30 dB, α = 0.35, v = 1 m/s, C/I = 100 dB

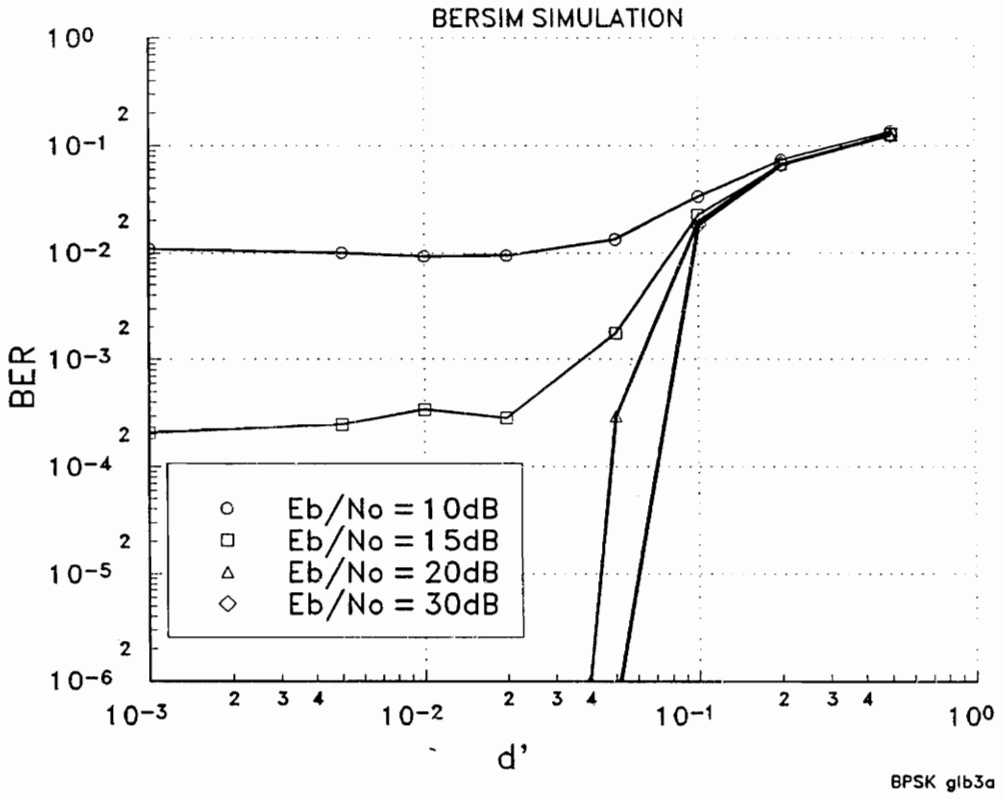


FIGURE 5.27: BER vs. d' for BPSK in the frequency-selective fading channel A with almost Ricean narrowband fading statistics.
fc = 1300 MHz, Eb/No = 10-30 dB, α = 0.35, v = 1 m/s, C/I = 100 dB

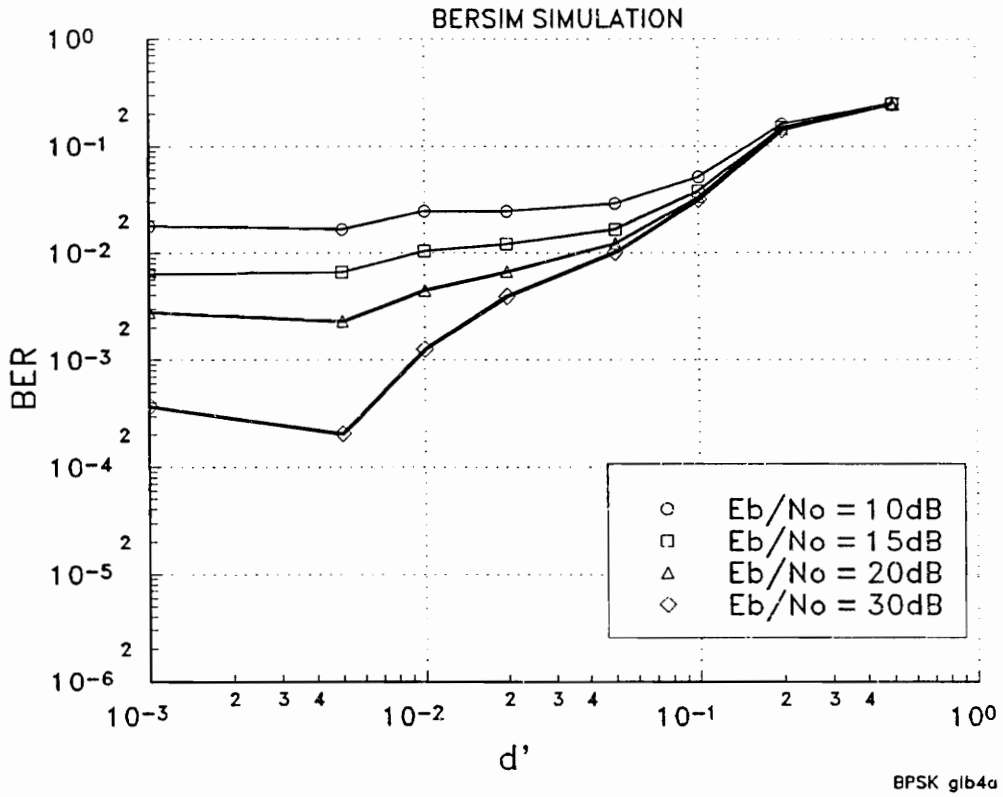


FIGURE 5.28: BER vs. d' for BPSK in the frequency-selective fading channel B with almost Rayleigh narrowband fading statistics.
fc = 1300 MHz, Eb/No = 10-30 dB, $\alpha = 0.35$, v = 1 m/s, C/I = 100 dB

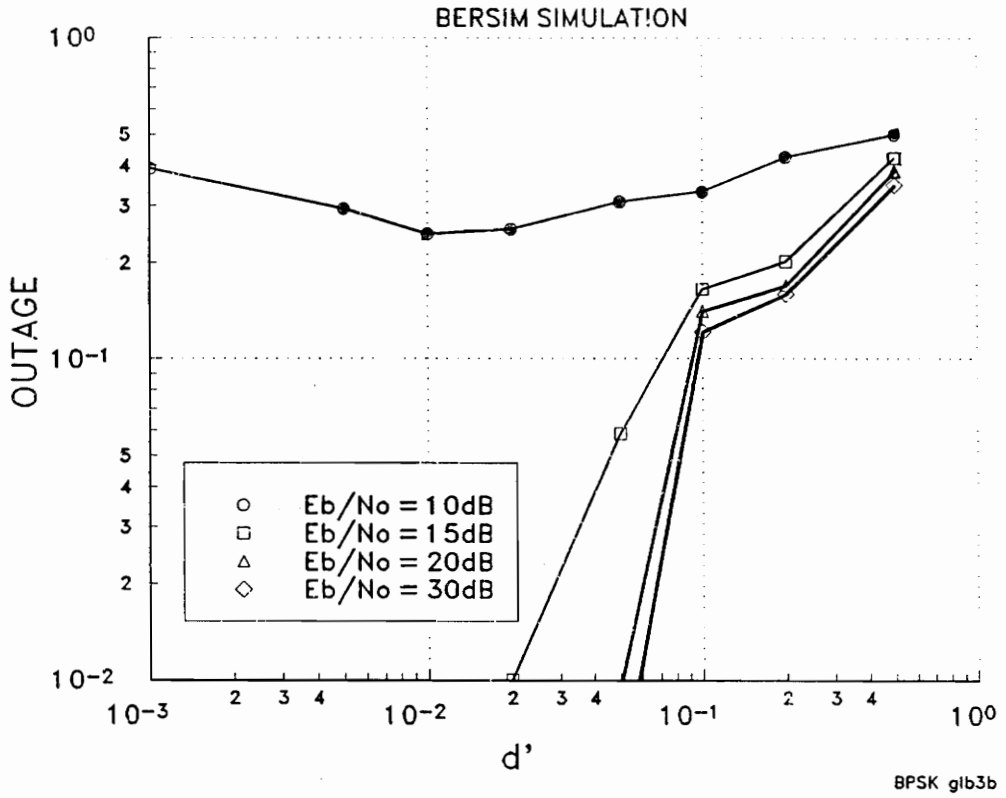


FIGURE 5.29: Outage vs. d' for BPSK in the frequency-selective fading channel A with almost Ricean narrowband fading statistics.
Outage block size 1024 bit, Outage criterion 1%,
fc = 1300 MHz, Eb/No = 10-30 dB, α = 0.35, v = 1 m/s, C/I = 100 dB

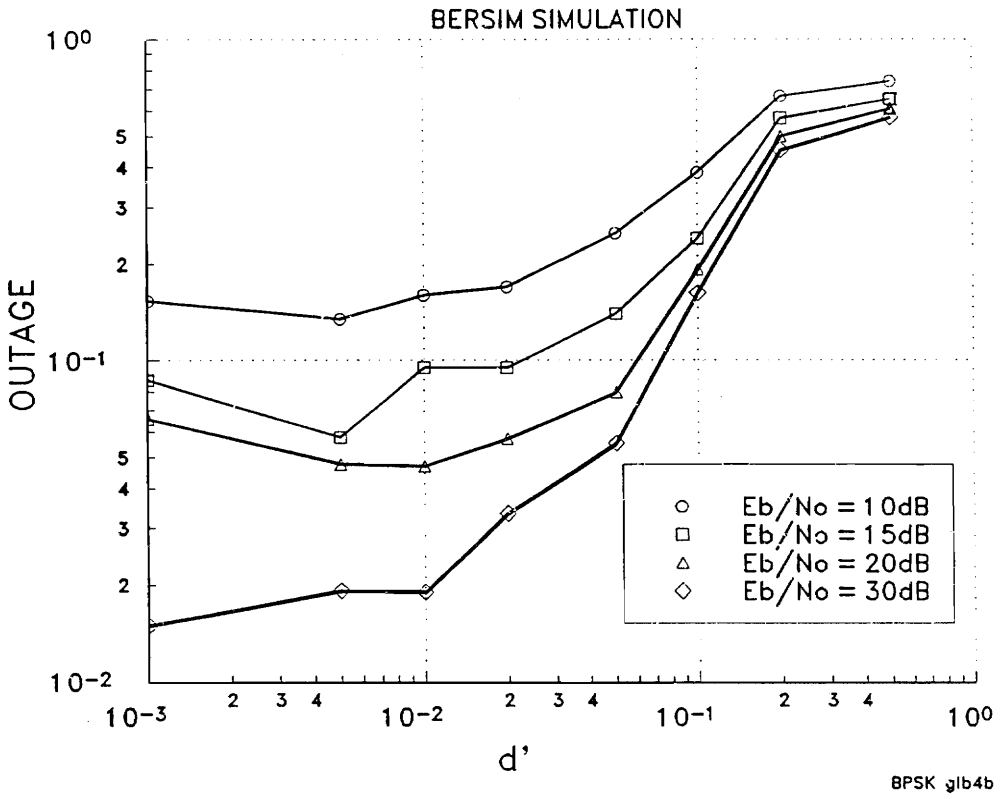


FIGURE 5.30: Outage vs. d' for BPSK in the frequency-selective fading channel B with almost Rayleigh narrowband fading statistics.
Outage block size 1024 bit, Outage criterion 1%,
fc = 1300 MHz, Eb/No = 10-30 dB, α = 0.35, v = 1 m/s, C/I = 100 dB

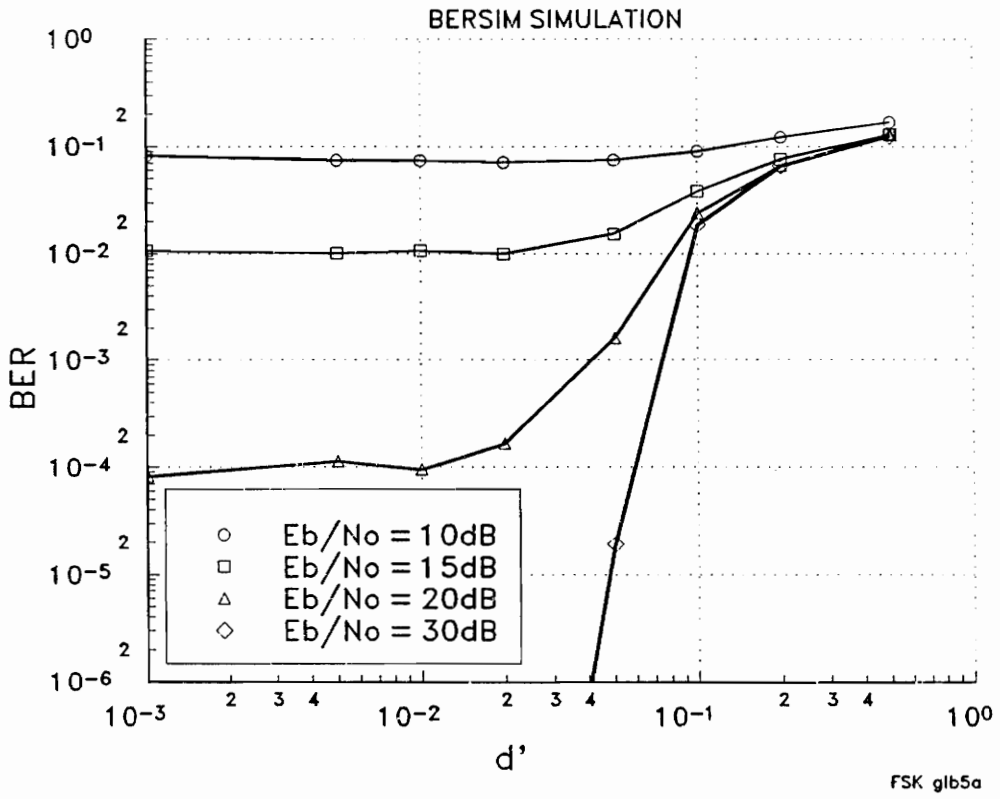


FIGURE 5.31: BER vs. d' for FSK in the frequency-selective fading channel A with almost Ricean narrowband fading statistics.
fc = 1300 MHz, Eb/No = 10-30 dB, α = 0.35, v = 1 m/s, C/I = 100 dB

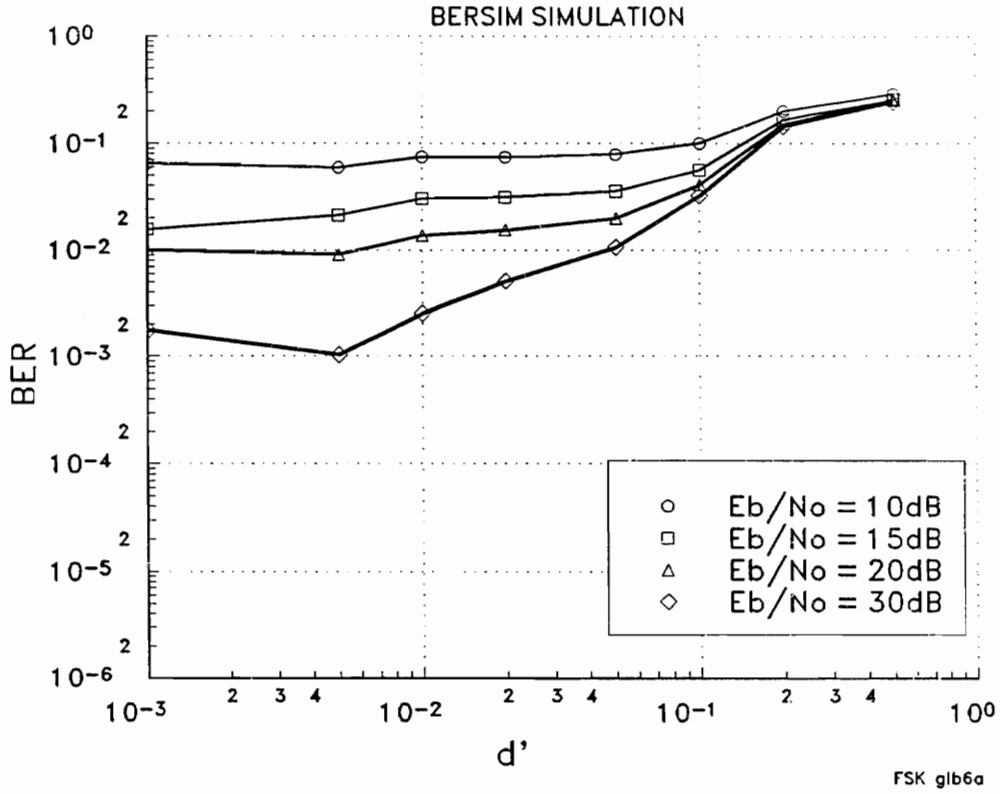


FIGURE 5.32: BER vs. d' for FSK in the frequency-selective fading channel B with almost Rayleigh narrowband fading statistics.
fc = 1300 MHz, Eb/No = 10-30 dB, α = 0.35, v = 1 m/s, C/I = 100 dB

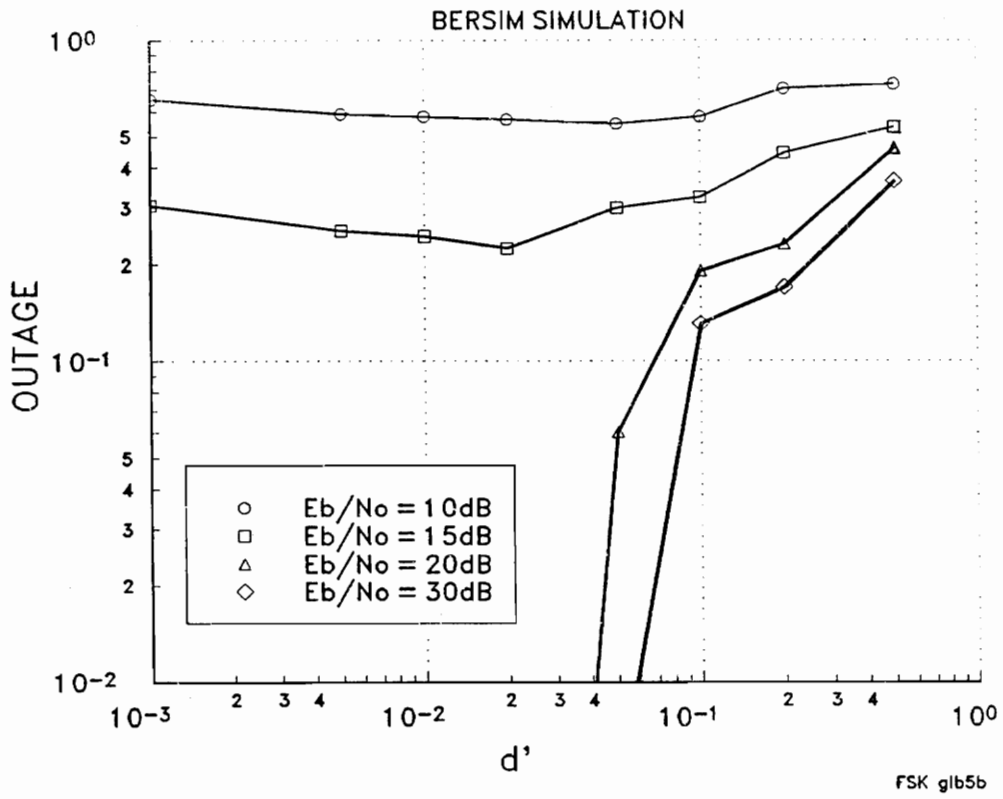


FIGURE 5.33: Outage vs. d' for FSK in the frequency-selective fading channel A with almost Ricean narrowband fading statistics.
Outage block size 1024 bit, Outage criterion 1%,
fc = 1300 MHz, Eb/No = 10-30 dB, α = 0.35, v = 1 m/s, C/I = 100 dB

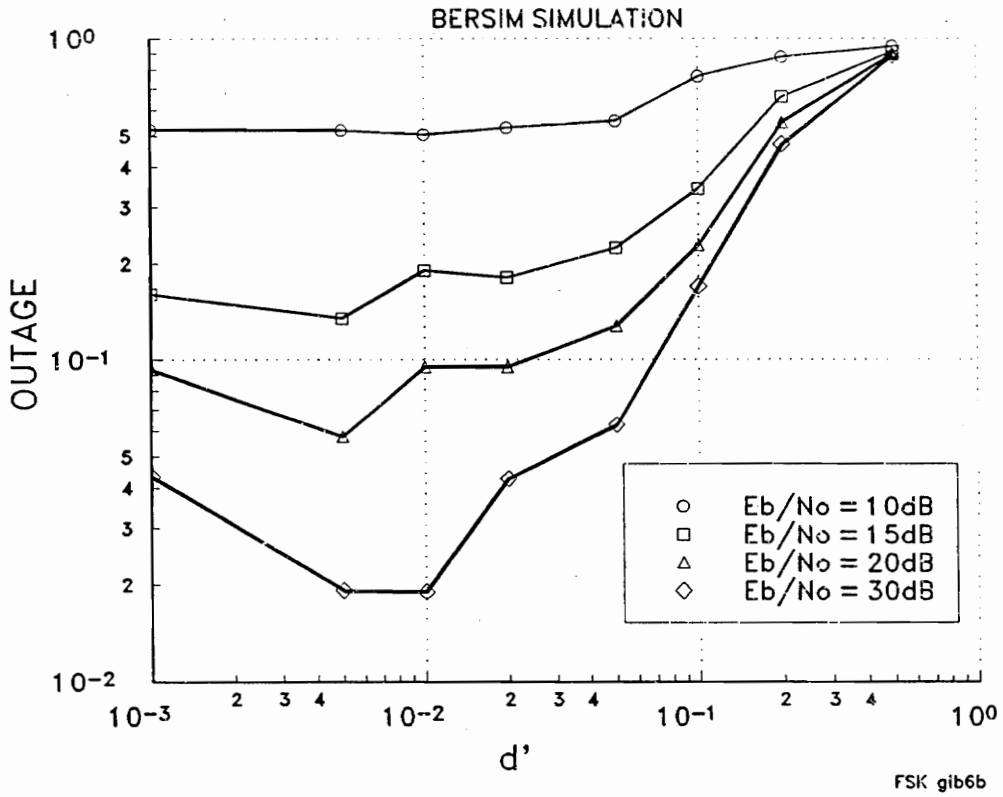


FIGURE 5.34: Outage vs. d' for FSK in the frequency-selective fading channel B with almost Rayleigh narrowband fading statistics.
Outage block size 1024 bit, Outage criterion 1%,
fc = 1300 MHz, Eb/No = 10-30 dB, α = 0.35, v = 1 m/s, C/I = 100 dB

A comparison of the bit error rate experienced in channels A and B (Figures 5.23, 5.24, 5.27, 5.28, 5.31, and 5.32) shows that for d' less than 10^{-2} and for low signal to noise ratios, a bit error floor exists independent from the modulation technique. For high signal to noise ratios, negligible bit error rates occur at low delay spreads. Specifically, when d' is less than 0.02, flat fading dominates the BER (the channel is not frequency-selective). The channel with the Rayleigh distributed narrowband fading characteristics (channel B) exhibits a larger BER than the channel with the Ricean narrowband fading (channel A) as the signal to noise ratio increases. The deeper fades in the Rayleigh fading channel are responsible for this result.

Note that for $\pi/4$ DQPSK, the flat fading error floors experienced for the channel with the Rayleigh distributed narrowband fading characteristics (channel B, Figure 5.24) match exactly the bit error rates found for the flat Rayleigh fading channel for various mobile speeds (Figure 5.4, range 0 to 30 dB). From this, it can be concluded that the current file conversion used to convert imported channel files works correctly. This is because an internally generated flat fading channel and an imported channel, whose narrowband statistics were externally determined to be Rayleigh, produced exactly the same bit error rates. For BERSIM 1.0 those results were always quite different, because of the different file conversion.

From Figures 5.24, 5.28, and 5.32 it can be found that for an Eb/No of 20 dB, only FSK experiences an error floor in channel B. A general comparison of FSK with the two other modulation techniques proves that FSK exhibits the poorest performance for slow fading conditions and low Eb/No. This is manly due to the higher noise sensitivity of FSK, which was experienced for AWGN noise channels and shown in Figure 5.1.

For values of d' greater than 0.05, intersymbol interference dominates both types of channels independent of Eb/No and the modulation technique (Figures 5.23, 5.24, 5.27, 5.28, 5.31, and 5.32). Note that as d' increases above 10^{-2}, the slopes of the BER are quite different for the two channels. At very high Eb/No, the BER for $\pi/4$ DQPSK in combination with channel B (Figure 5.24) increases by the square of d'. This was also found in [Chu87a] for a channel model with a fixed power delay profile that experienced Rayleigh fading.

A comparison of the outage probabilities for $\pi/4$ DQPSK in Figure 5.25 and Figure 5.26 shows almost equal outage probabilities of channels A and B can occur for the normalized rms delay spread d' grater than or equal to $5*10^{-2}$. However, no similar result could be found for the other modulation techniques.

It is significant that for almost all simulations and for all modulation techniques, the outage rate decreases with increasing normalized rms delay spread. It reaches its minimum in most cases at d' of about 0.02. The reason for this observation is most likely the combination of two error mechanisms. One is that individual multipath components do not fade as deep as the phasor sum of all multipath components due to frequency diversity [TTL91]. Therefore, deep signal fades and burst errors are less frequent. However, the increasing signal spread introduces intersymbol interference which has a random nature. The outage rate, in combination with the bit error rate, gives an indication of the burstiness of the channel. As a result, the outage probability can go down significantly while the bit error rate decreases only slightly.

In the flat fading range, the Rayleigh channel shown in Figure 5.26 experiences a slightly better outage performance than the Ricean channel shown in Figure 5.25 for signal to noise ratios of 10 dB, even though the raw bit error rates indicate that channel A is better.

Therefore, errors in channel B with the Rayleigh narrowband fading envelope appear to be more clustered than in channel A for small signal to noise ratios.

In general, when the rms delay spread is normalized to the bit period, BPSK offers similar performance to $\pi/4$ DQPSK (Figures 5.23, 5.24, 5.27, and 5.28). This contradicts findings by Chuang [Chu87a] who found that 4-level modulations (QPSK, OQPSK, and MSK) are more resistant to delay spread than BPSK for constant information throughput. The reason for the different results is that Chuang considered coherent detection for all modulation schemes, which performs better than the differential detection for slow fading channels [Chu90], whereas in BERSIM, only BPSK is coherently detected. FSK offers the poorest performance for flat fading as well as for frequency selective fading conditions.

5.5.2 Comparison of Channels with the Same Spatially Varying Channel Impulse Response.

To get a further understanding about the importance of the delay profile for bit error rate prediction, the exact same magnitude impulse response as for channel B was taken and new phase information was synthesized. This was possible by using SIRCIM in the mode which generates a narrowband fading envelope from a previously existing magnitude impulse response. Several SIRCIM simulations were performed to find two very different narrowband fading envelopes based on the same magnitude impulse response.

Figure 5.35 and Figure 5.36 describe these new channel impulse responses based on channel B. It can be seen that the narrowband characteristics of the channel described in Figure 5.35 (Channel B.1) indicate that a narrowband signal in this channel suffers from almost no fades. The other extreme is the channel described in Figure 5.36 (Channel B.2) where the narrowband fading envelope has properties worse than a Rayleigh fading envelope down to 1% probability.

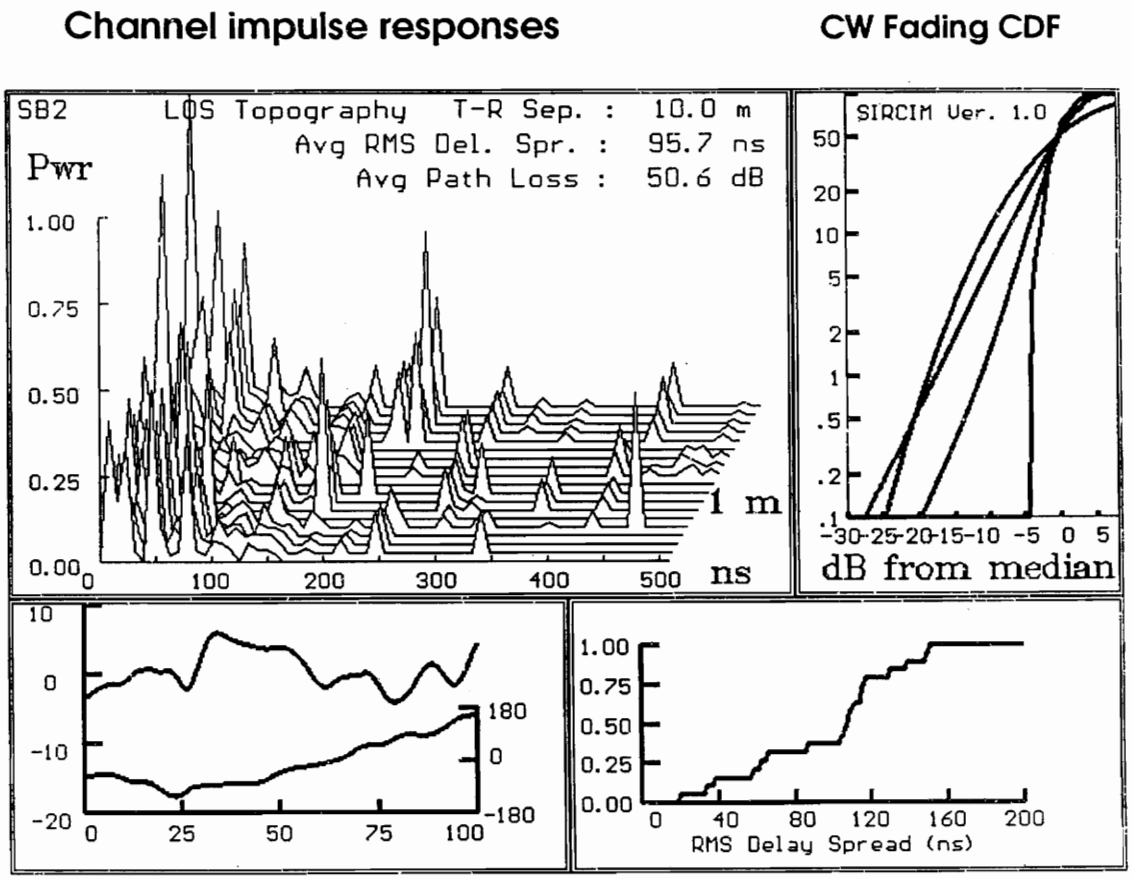


FIGURE 5.35: Statistical characteristics of impulse response profile for channel B.1 generated by SIRCIM.

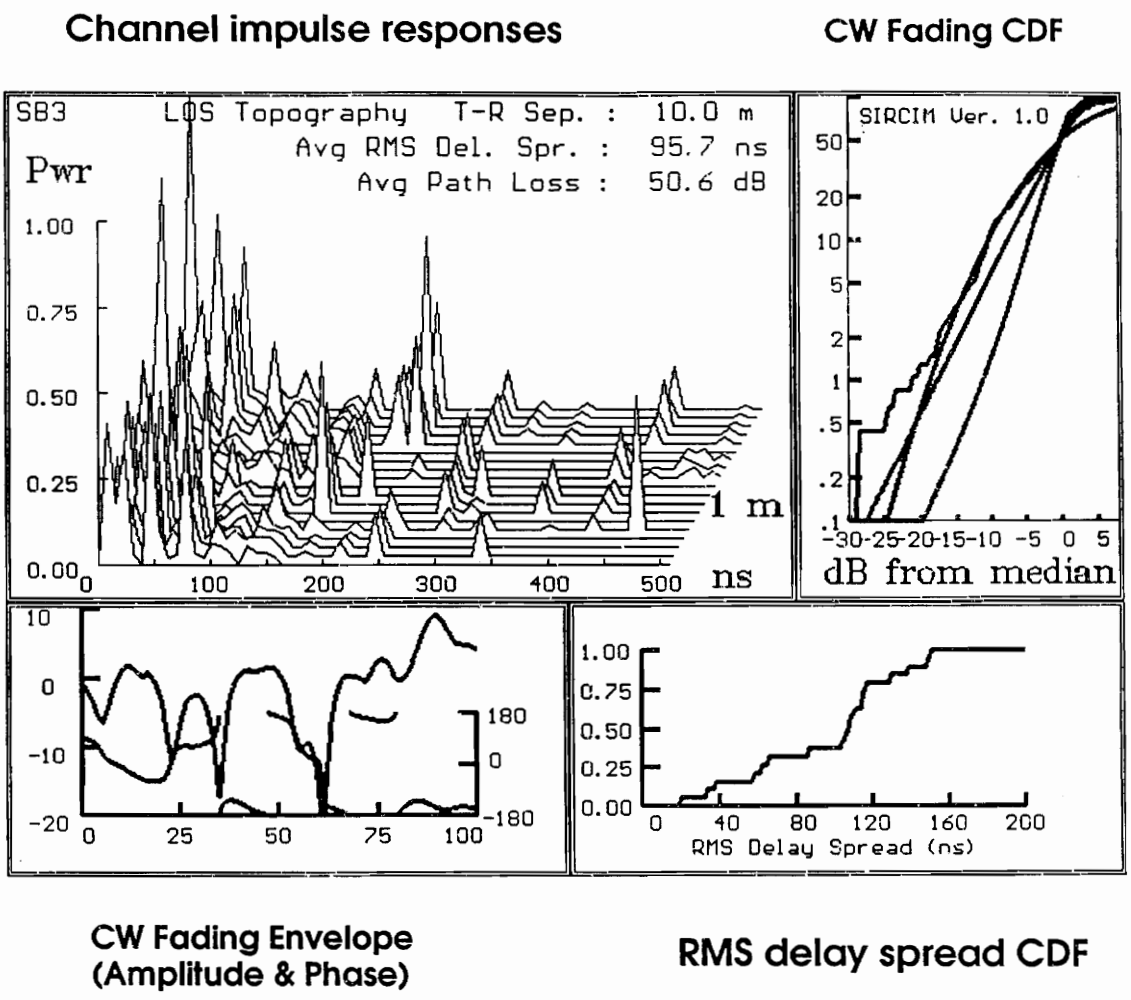


FIGURE 5.36: Statistical characteristics of impulse response profile for channel B.2 generated by SIRCIM.

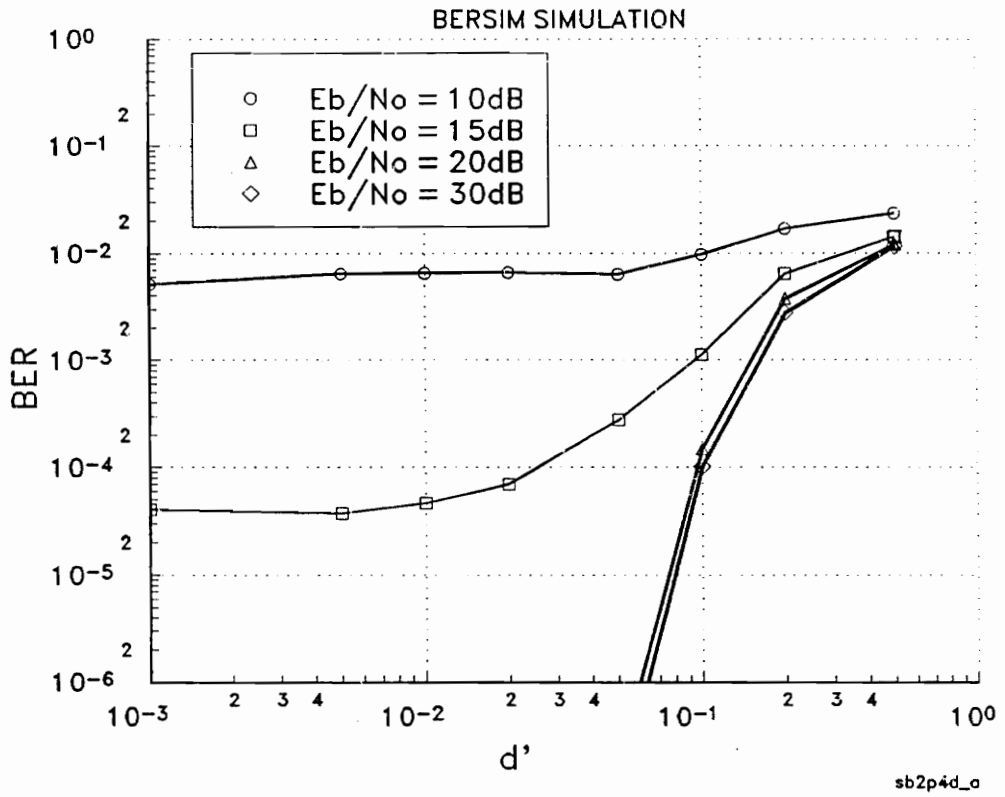


FIGURE 5.37: BER vs. d' for $\pi/4$ DQPSK in the frequency-selective fading channel B.1 with almost no fades of the narrowband fading envelope.
fc = 1300 MHz, Eb/No = 10-30 dB, α = 0.35, v = 1 m/s, C/I = 100 dB

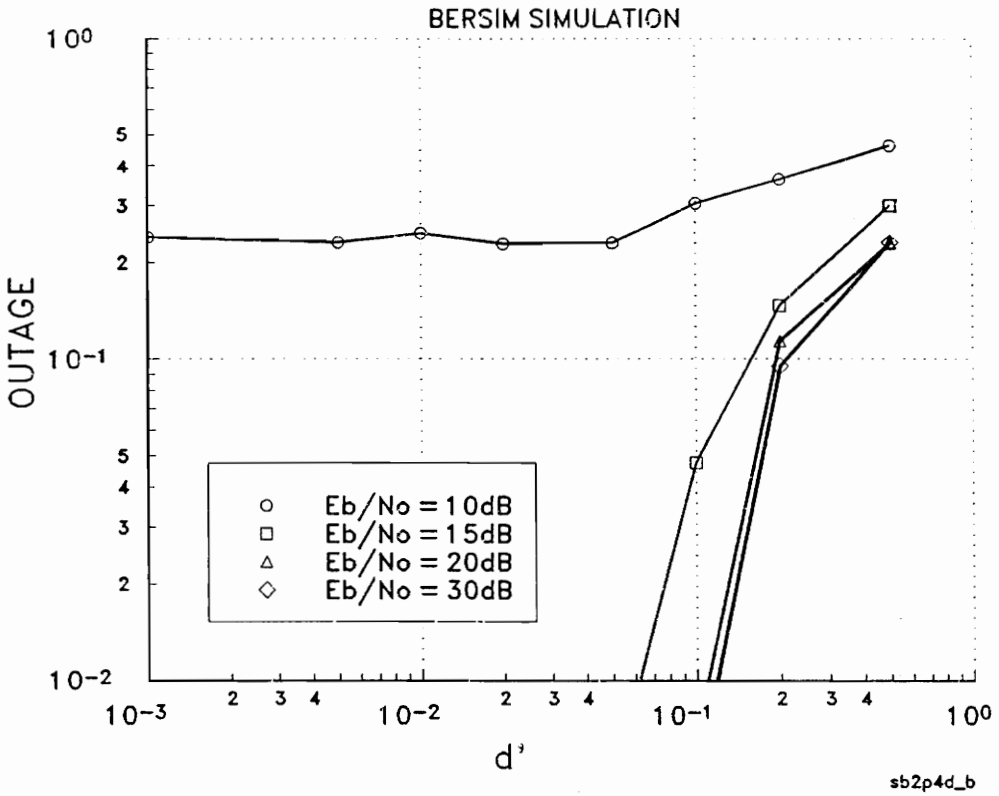


FIGURE 5.38: Outage vs. d' for π/4 DQPSK in the frequency-selective fading channel B.1 with almost no fades of the narrowband fading envelope.
Outage block size 1024 bit, Outage criterion 1%,
fc = 1300 MHz, Eb/No = 10-30 dB, α = 0.35, v = 1 m/s, C/I = 100 dB

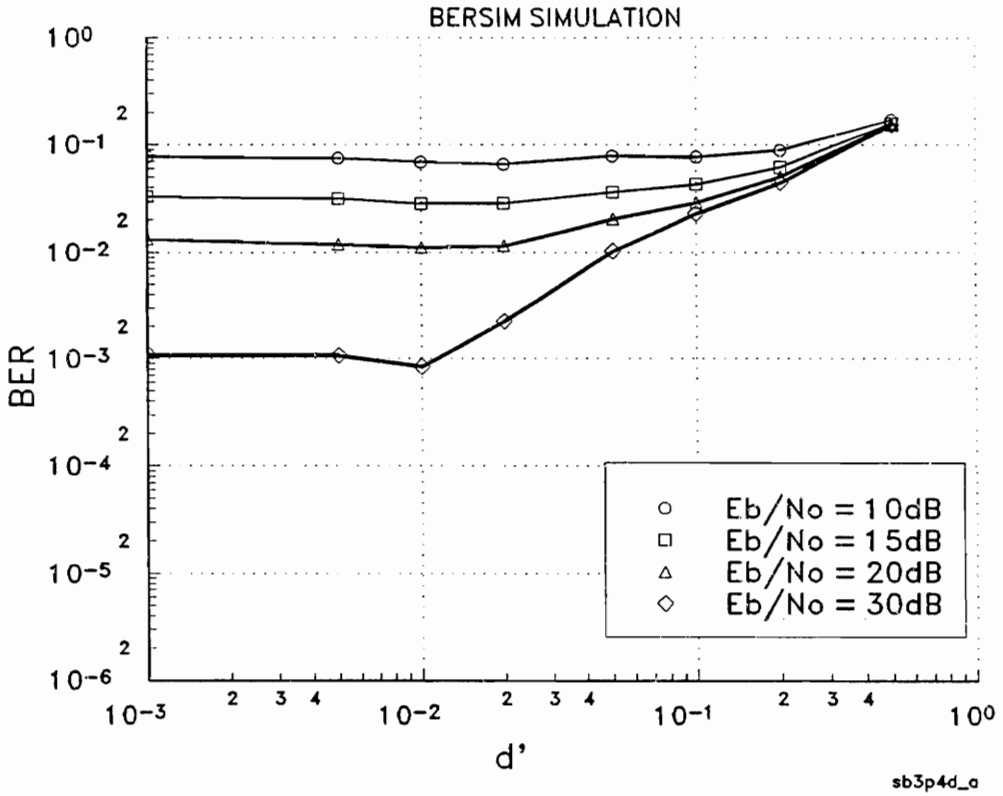


FIGURE 5.39: BER vs. d' for $\pi/4$ DQPSK in the frequency-selective fading channel B.2 with extreme deep fades of the narrowband fading envelope.
fc = 1300 MHz, Eb/No = 10-30 dB, α = 0.35, v = 1 m/s, C/I = 100 dB

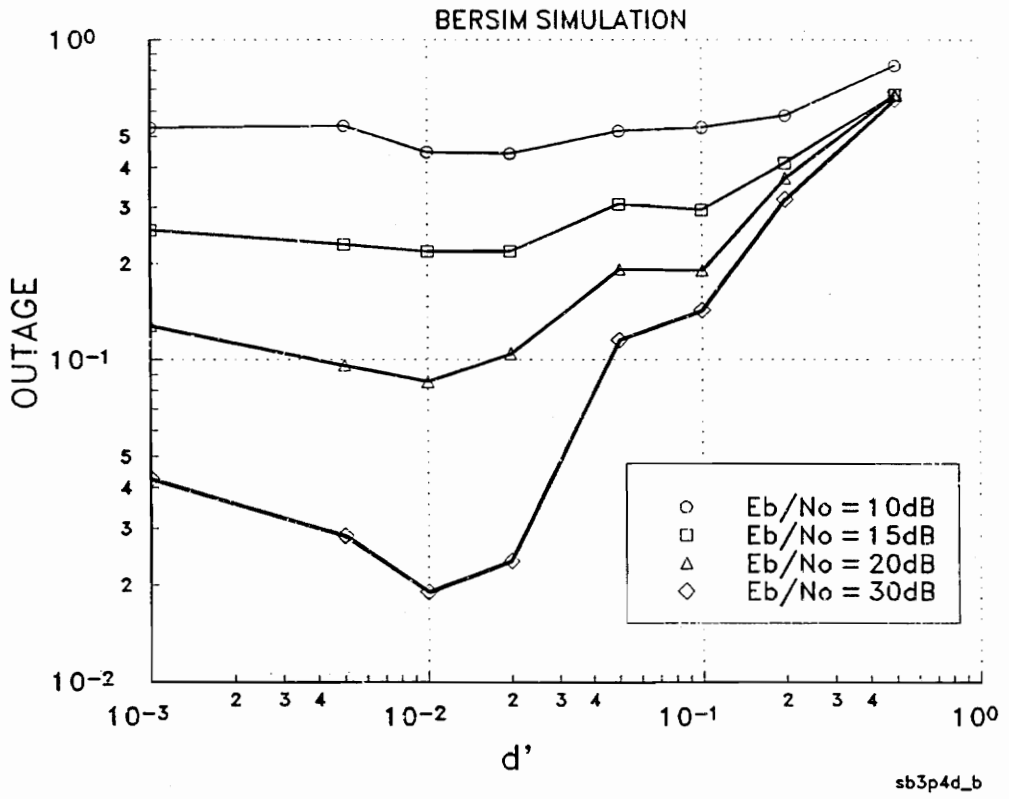


FIGURE 5.40: Outage vs. d' for π/4 DQPSK in the frequency-selective fading channel B.2 with extreme deep fades of the narrowband fading envelope.
Outage block size 1024 bit, Outage criterion 1%,
fc = 1300 MHz, Eb/No = 10-30 dB, α = 0.35, v = 1 m/s, C/I = 100 dB

For the following analysis of the error mechanisms based on these additional channels, $\pi/4$ DQPSK modulation is considered. Figure 5.37 and Figure 5.39 show that as expected, the bit error rate varies significantly between both channels under flat fading conditions. For d' below 0.1 channel B.2 performs for all signal to noise ratios poorer than the before analyzed channel B (Rayleigh, Figure 5.24). However, for a normalized delay spread of more than 0.1 this channel begins to perform better than the previously analyzed channel with Rayleigh narrowband statistics. This indicates that the phase distribution of the multipath components impacts the BER significantly, even at large normalized delay spreads. From the outage results (Figure 5.38 and Figure 5.40) it can be also seen that this channel becomes less bursty than channel B for d' above 0.1. The significance of the phase becomes evident if channel B.1 (Figure 5.35) is considered. Figure 5.37 shows that for large normalized delay spreads, the predicted bit error rates are of more than an order of magnitude smaller than the bit error rates predicted based on the channels B and B.2 (Figures 5.24 and 5.39). This indicates that the phase distribution of the multipath components is at least of equal importance as the magnitude distribution. As can be seen from Figure 5.37, favorable phase distributions can provide good system performance even for high normalized delay spreads.

5.5.3 Comparison of Channels with Almost Same Spatially Varied RMS Delay Spread and Similar Narrowband Fading Statistics.

Based upon these experiences, simulations were run to determine if knowledge of the narrowband fading statistics, in combination with the averaged rms delay spread, could be used to make an assumption about the expected bit error rate. Numerous SIRCIM simulations were performed using the same magnitude channel impulse response (as in Section 5.5.2) of channel B to find narrowband fading envelopes whose statistics were similar to that of channel A (Ricean). With this method two channels with nearly the same spatial

averaged rms delay spread and narrowband fading characteristics could be compared. The only significant difference between channel A and channel B.3 (Figure 5.41) are their different magnitude and phases for the spatially varying complex impulse responses.

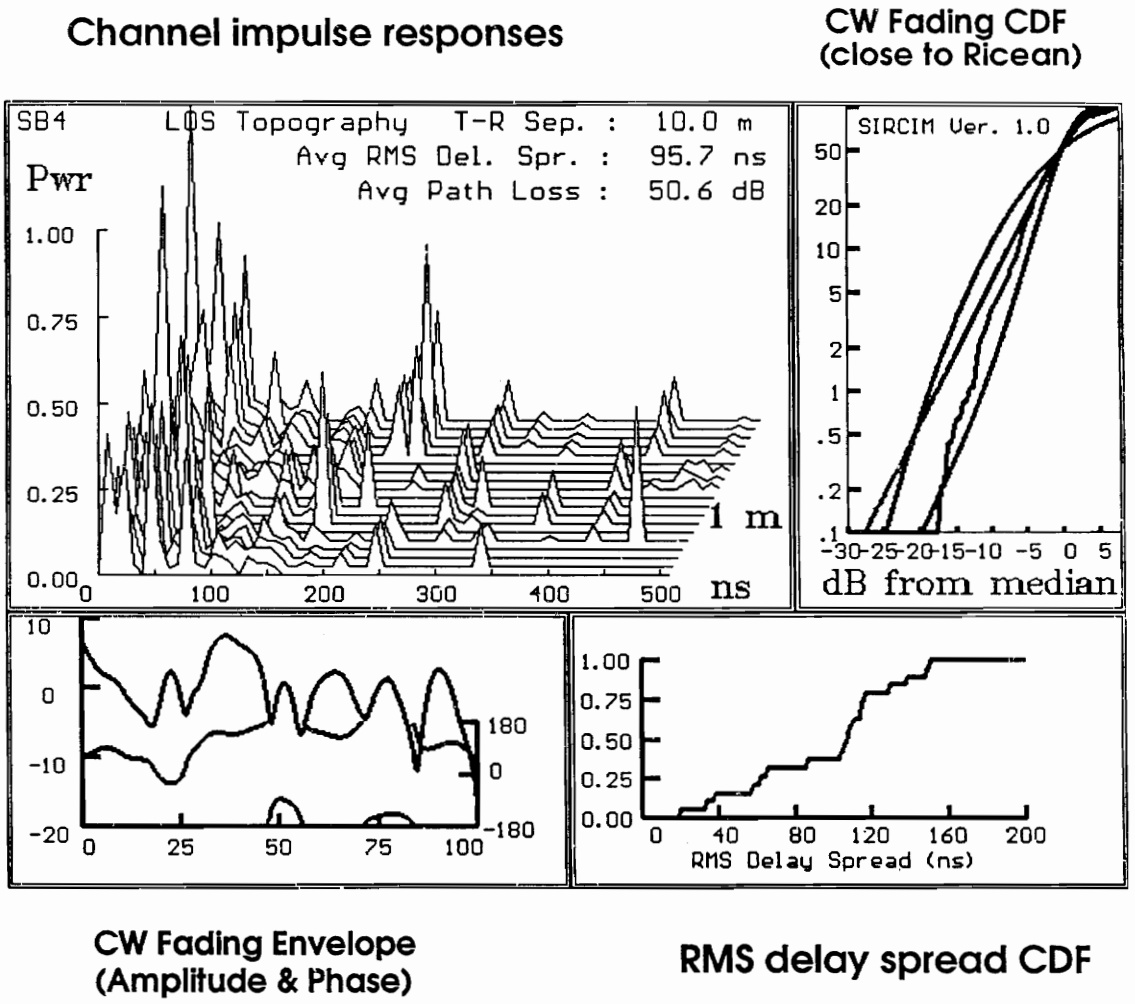


FIGURE 5.41: Statistical characteristics of impulse response profile for channel B.3 generated by SIRCIM.

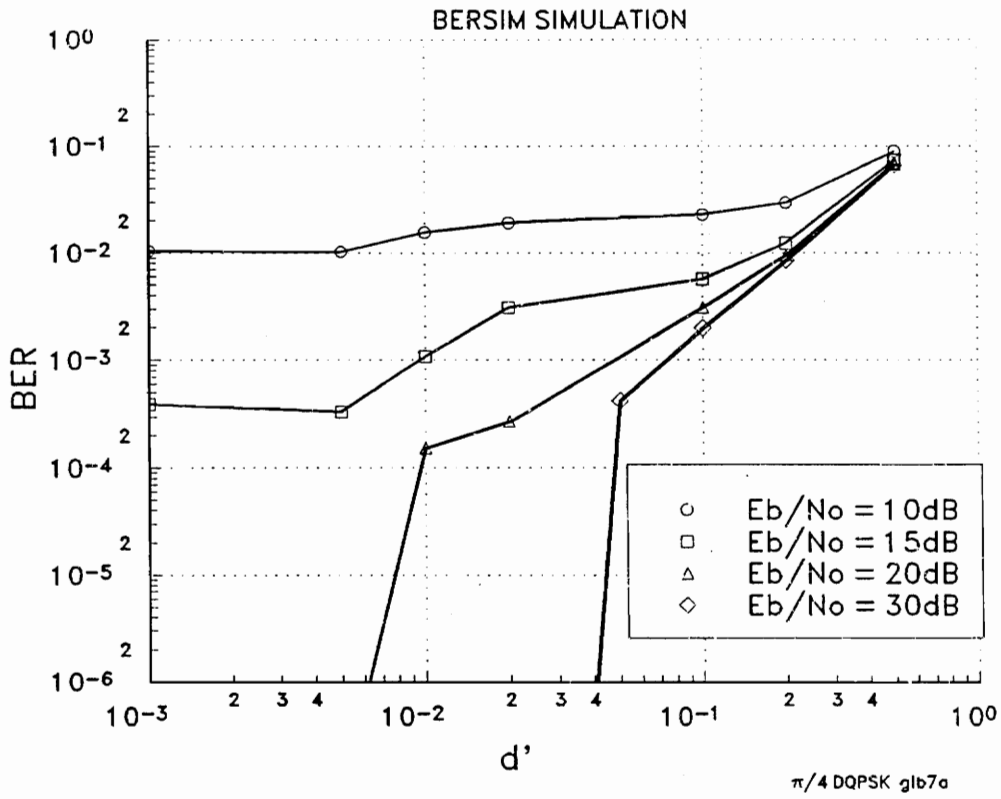


FIGURE 5.42: BER vs. d' for π/4 DQPSK in the frequency-selective fading channel B.3 with almost Ricean narrowband fading statistics.
fc = 1300 MHz, Eb/No = 10-30 dB, α = 0.35, v = 1 m/s, C/I = 100 dB

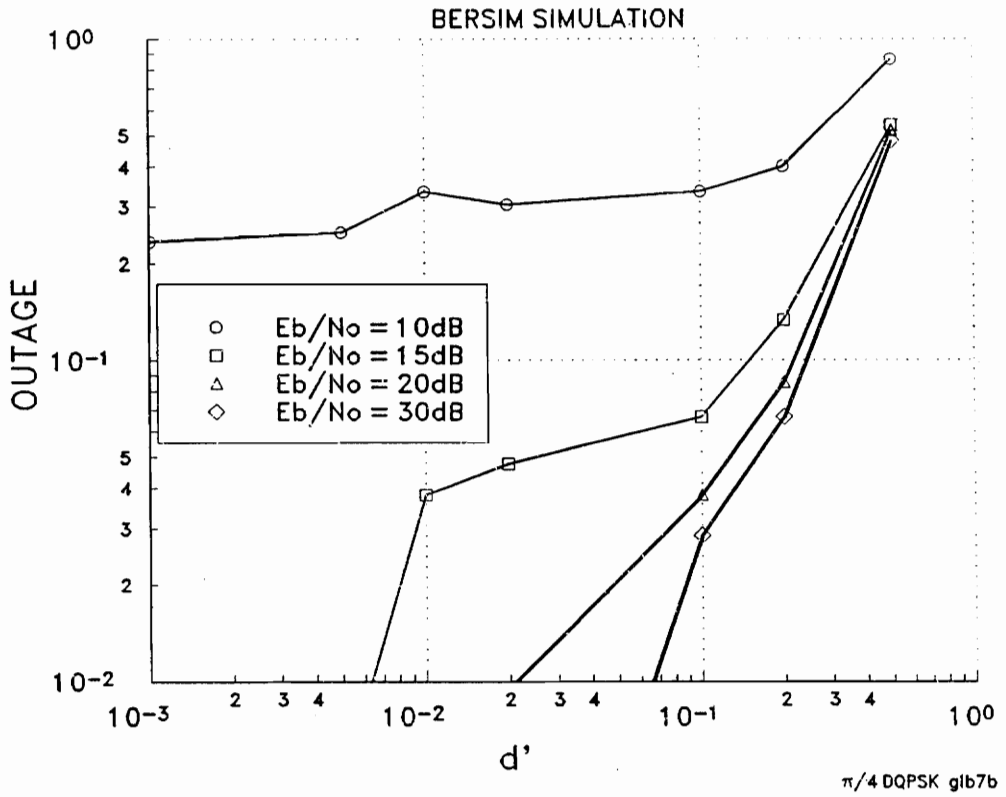


FIGURE 5.43: Outage vs. d' for π/4 DQPSK in the frequency-selective fading channel B.3 with almost Ricean narrowband fading statistics.
Outage block size 1024 bit, Outage criterion 1%,
fc = 1300 MHz, Eb/No = 10-30 dB, α = 0.35, v = 1 m/s, C/I = 100 dB

The comparison of the results (Figure 5.23, Figure 5.25, Figure 5.42, and Figure 5.43) shows that even though both channels have very similar narrowband fading statistics, the predicted bit error rates and outage rates for narrowband signals are different. The bit error floor experienced in combination with channel B.3 is about 1/3 lower than the error floor experienced using channel A. More significant, however, are the different bit error rates for large normalized rms delay spreads. These differences are even more evident for the outage rates shown in Figure 5.25 and Figure 5.43. As a result, even for similar spatially averaged rms delay spread and similar narrowband fading statistics, the corresponding bit error rate and outage rate can differ significantly for large normalized rms delay spreads. Similar findings were made by Turkmani [Tur92] who found that the distribution of the error pattern, and therefore the burstiness of the errors, is highly dependent on the shape of the power delay profile.

5.5.4 Comparison with Two-Ray Model

To compare the results obtained from the analyzed channels (A, B, B.1, B.2, B.3) with the IS-54 [EIA89] two-ray Rayleigh fading model, similar simulations as described above were performed using BERSIM's two-ray model. The delay τ between the two rays was varied for d' larger than 0.1 and the average power of the rays was set equal. The spatially averaged rms delay spread for the two-ray model can be derived from equation (2.2) and (2.3). It has a closed form, for two rays with equal average power, given by

$$\langle \sigma_{rms} \rangle = \frac{\tau}{2} \tag{5.4}$$

Below this threshold, the rms delay spread of the two-ray model was varied by adjusting the average power ratio between ray 1 and ray 2.

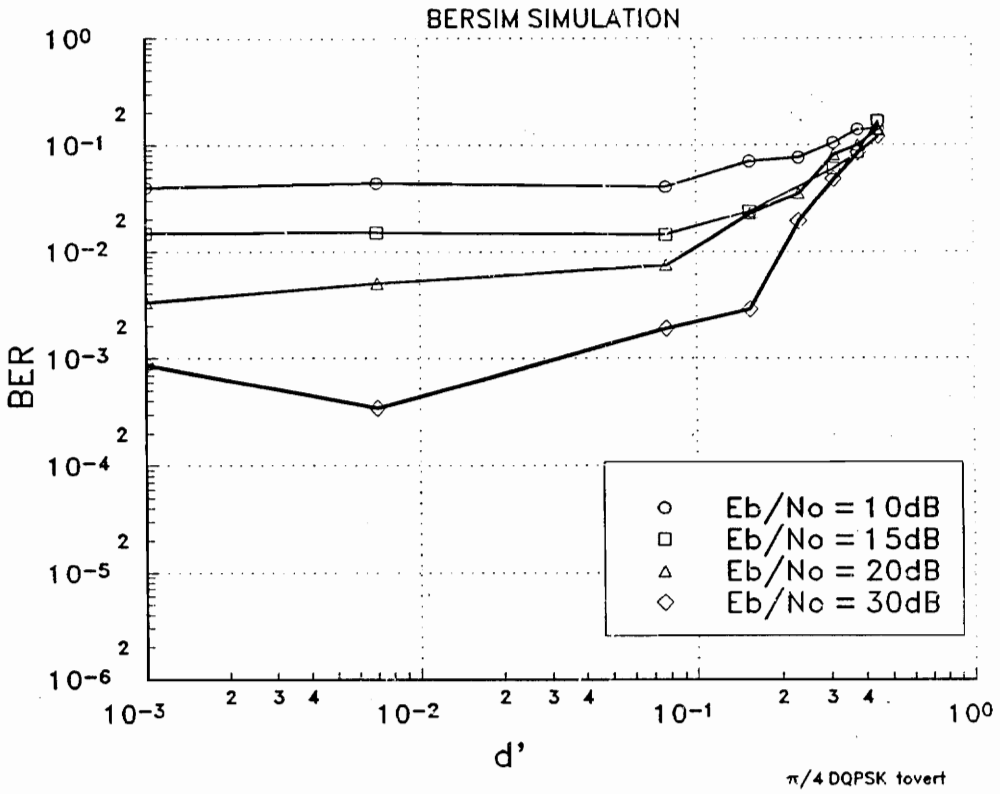


FIGURE 5.44: BER vs. d' for π/4 DQPSK in a two-ray Rayleigh frequency-selective fading channel.
fc = 1300 MHz, Eb/No = 10-30 dB, α = 0.35, v = 1 m/s, C/I = 100 dB

Figure 5.23 shows that for the signal to noise ratios considered, the bit error rate is almost constant for normalized delay spreads below this threshold. However, for delay spreads above this threshold the bit error rate increases rapidly. A comparison with the previously discussed channels shows that the two-ray Rayleigh model definitely predicts bit error rates which are among the worst found for small delay spreads (flat fading region). For normalized rms delay spreads around 0.1, though, the two ray model predicts lower bit error rates than simulated SIRCIM channels with similar narrowband fading statistics and error floors. In this region the two-ray Rayleigh model **cannot** be considered a worst case model.

5.5.5 Conclusion

The influence of multipath component distributions was analyzed. It was found that coherently detected BPSK with perfect phase recovery performed best among the three compared modulation techniques for flat fading as well as for frequency selective fading channels. The significant finding of this chapter is that even if the power delay profile of a channel is known, no reliable bit error rate prediction for narrow or wideband signals can be made. It has been found that the knowledge of the phase distribution among the multipath components is of equal importance as the knowledge of the magnitude distributions. In order to make reliable bit error rate predictions, a complex channel impulse response is essential. It has also been found that without equalization in many cases, reliable data transmission with outage rates of less than 10% is not possible when the normalized rms delay spread exceeds 0.2. This or even lower thresholds have been proposed in several publications [Jak74] [Bel63] [Chu87a]. However, this threshold can only be considered a "rule of thumb". A comparison of the simulated real-world channels with a two-ray Rayleigh channel illustrated that the two-ray Rayleigh channel can be considered a extreme channel model. However, it has been also found that real-world channels can exhibit higher BERs when compared against two-ray Rayleigh channel results.

This study shows how delay spread parameters or narrowband fading characteristics alone are not sufficient to predict the BER performance and outage probabilities in frequency selective fading channels. This important result suggests that true impulse response models that retain *both the amplitude and phase* of individual multipath components should be used to simulate actual performance.

6. Conclusion

6.1 Summary of Work

A software simulator for digital mobile and portable communication systems was rewritten and enhanced. The simulator (BERSIM 2.0) is based on a program originally developed by Victor Fung as part of his masters thesis (BERSIM 1.0). BERSIM models a digital mobile radio communication system using complex transfer functions in combination with lowpass signal representations. It can simulate the modulation schemes $\pi/4$ DQPSK (IF Differential detection), BPSK (coherent detection with perfect phase recovery), and FSK (differential non-coherent detection).

The major enhancements were:

- An identical version of BERSIM can now run on both MS-DOS and UNIX platforms.
- A new method to determine the optimum fading envelope length for constant resolution over a wide range of mobile speeds has been implemented.
- A variable carrier frequency for the internally generated two-ray channel model can now be selected.
- User specified channel files, that hold the spatially varying complex impulse response of the channel, for any size of channel can be used.
- Outage for packet based data transmission can be evaluated
- Timing recovery methods to determine the correct sampling instant can be modeled.
- A new BPSK receiver has been implemented.
- A new method of transforming imported channel impulse responses has been implemented.
- Simulation data can now be user specified.
- Timing jitter can be simulated.

- Cochannel interference in combination with imported channels can be simulated.
- A graphical user interface for the PC version, with dynamically changing eye pattern display and sample point indication, has been developed.
- Optional AWGN channel simulation is possible.

The function of BERSIM 2.0 was verified by comparison of simulation results with analytical and simulation results found in the literature. Simulations were performed to determine the sensitivity of the implemented modulation techniques for timing jitter. It has been found that $\pi/4$ DQPSK is most sensitive to timing jitter.

The outage rate for noise dominated systems was compared with the outage experienced in interference dominated (one interferer) systems. It has been found that interference and noise equally effect system performance.

The performance of the four timing recovery schemes has been evaluated. It has been found that for channels with rms delay spread larger than $0.05T_{sym}$, and for channels that have a channel impulse response which introduces signal delay, timing adjustment is essential.

A study of the importance of the distribution of multipath component within a delay profile and their corresponding phase distribution was made. It was found that the knowledge of the channel impulse response does not provide enough information to reliably estimate bit error rates or outage probabilities. It has also been found that the phase distribution of the channel impulse response components is of equal importance as the amplitude distribution. It is concluded that an additional figure of merit that describes this information in the channel needs to be found in order to estimate bit error rates for frequency selective fading channels. It is also concluded that a side specific propagation prediction tool needs

to provide complex channel impulse response data in order to allow site specific bit error rate estimation.

6.2 Recommendations for Extension of This Research

The first recommendation for future enhancements on the program BERSIM is to shift the personal computer version from the MS-DOS operating system to the OS2 operating system. OS2 is a full 32-bit operating system that can take advantage of the power of modern 32-bit microprocessors as well as utilize the full quantity of memory available on the PC (i.e., there is not a 64k memory segmentation problem). So far, BERSIM 2.0 suffers from so many constraints imposed by the MS-DOS operating system that further extensions might not be possible for unexperienced programmers without significant difficulties. New compilers available for the OS2 operating system, in combination with program modifications that use the larger available memory, should decrease the computation time by a factor of three. With these modifications, the computation time on the workstation will be decreased, as well. This is because the same code is used on both platforms for better maintainability.

With more available memory, the program modifications could include larger simulation block sizes and/or higher resolution by increasing the number if samples per symbol. If a fully continuous received signal is needed, one source of speeding up the processing time should be the implementation of the matched filter in the frequency domain. With more memory, a higher maximum doppler shift for simulations based on the Rayleigh fading envelope model could be also supported since larger IFFTs (needed for the fading signal generation) could be used. The current version of BERSIM 2.0 is already set up for variation of the IFFT size. This is especially important to support simulations of future personal communication networks operating at frequencies of more than 2GHz.

For simulations of a stationary terminal as well as for line of sight microcell simulations, a channel model that follows a Ricean distribution could be incorporated. Furthermore, multi-ray channel models which are suggested for testing of equipment for the European GSM[1] standard can be easily implemented. Even the implementation of a constant amplitude multi-ray channel model (uniform distributed phase) could be of advantage for further studies of bit error mechanisms as well as for preliminary evaluation of equalizers.

Another future software modification could be the implementation of new modulation techniques such as GSMK[2], 16APSK[3], and others which are of current research interest. Consequently, the filter modules should be changed in a way that various filter impulse responses can be generated dynamically.

For direct-sequence spread-spectrum work, the perfect phase recovery for BPSK should be modified to reflect the real world. One way to do this would be to corrupt the ideal reference signal in BPSK.

More complex timing recovery modules in combination with channel equalizers are also useful future extensions. In combination with an equalizer could be a module incorporated which frames the data for according to the frame specifications of the proposed U.S. digital cellular standard or any other standard. This would be another step towards simulating a complete implemented system.

One recommendation in terms of research on bit error rate mechanisms is to analyze the importance of multipath time delay jitter in real world channels [Dev87][Rap88] on the bit

1. earlier: Groupe Special Mobile, now: Global System for Mobile Communication
2. Gaussian Minimum Shift Keying
3. 2 Level 8 Phase Shift Keying

error rates. It has been found that this variation puts thresholds on the performance of timing recovery loops. BERSIM 2.0 already provides the feature to collect data about the average delay of each channel impulse response. SIRCIM and SMRCIM[1] provide the ideal channel models for this kind of research.

Simulation of systems utilizing antenna diversity can also be one future research area. Several branch selection methods are suitable to be simulated with BERSIM. With upcoming parallel processing computers, the implementation of antenna diversity techniques should be easier and faster than could be achieved with single processor machines. It is, therefore, recommended that a future researcher not perform these extensions on the actual program until multi-processor machines are available. Otherwise, the study of antenna diversity techniques should be performed off-line by first recording signal and the corresponding data stream for each branch on the hard drive. An independent program should be used for further processing.

BERSIM 2.0 is patent pending, and is available from Virginia Tech.

1. Simulation of Mobile Radio Channel Impulse Models developed by MPRG, available from VTIP Inc.

7. References

[Ada92] Adachi, F., Ohno, K., and Ikura, M., "Postdetection Selection Diversity Reception with Correlated, Unequal Average Power Rayleigh Fading Signals for π/4 Shift QDPSK Mobile Radio," RF Design., vol. 14, no. 2, pp. 49 - 56, Feb. 1991.

[Anv91] Anvari, K., Woo, D., "Susceptibility of π/4 DQPSK TDMA Channel to Receiver Impairments," IEEE Trans. Veh. Tech., vol. VT-41, no. 2, pp. 199 - 211, May 1992.

[Bai68] Bailey, C. C., and Lindenlaub, J. C., "Further Results Concerning the Effect of Frequency-Selective Fading on Differential Coherent Matched Filter Receivers," IEEE Trans. Comm., pp. 749 - 751, Oct. 1968.

[Bel62a] Bello, P. A., and Nelin, B. D., "Predetection Diversity Combining with Selectively Fading Channels," IRE Trans. Comm. Sys., vol. CS-10, pp. 32 - 42, March 1962.

[Bel62b] Bello, P. A., and Nelin, B. D., "The Influence of Fading Spectrum on the Binary Error Probabilities of Incoherent and Differentially-Coherent Matched Filter Receivers," IRE Trans. Comm. Sys., vol. CS-10, pp. 160 - 168, June 1962.

[Bel63] Bello, P. A., and Nelin, B. D., "The Effect of Frequency Selective Fading on the Binary Error Probabilities of Incoherent and Differentially Coherent Matched Filter Receivers," IEEE Trans. Comm., pp. 170 - 186, June 1963.

[BER92] VTIP Inc., *Users Manual for BERSIM 2.0,* to appear in fall 1992.

[Cha86] Chang L. F., Arnold, R. C., Bernhardt, R. C., and Porter, P. T., "Coding as Means to Implement Diversity in a Frequency Reusing Portable Radio System," Second Nordic Seminar on Digital Land Mobile Radio Commun. (DMR II), Stockholm, Sweden, pp. 90 - 93, Oct. 1986.

[Cha89a] Chang L. F., and Chuang, J. C-I., "Diversity Selection Using Coding in a Portable Radio Communications Channel with Frequency-Selective Fading," IEEE Journal on Selected Areas in Communications, vol. SAC-7, no. 1, pp. 89 - 98, January 1989.

[Cha89b] Chang L. F., and Chuang, J. C-I., "Outage Probability for a Frequency-Selective Fading Digital Portable Radio Channel with Selection Diversity Using Coding," Proc. IEEE Globecom '89, pp. 176 - 181, Dec., 1989.

[Chu86] Chuang, C-I. J., "The Effects of Multipath Delay Spread on Timing Recovery," Proc. IEEE Int. Conf. on Comm., pp. 55 - 59., Toronto, Canada, June 1986.

[Chu87a] Chuang, C-I. J., "The Effects of Time Delay Spread on Portable Radio Communications channels with Digital Modulation," IEEE Journal on Sel. Areas in Comm., vol. SAC-5, pp. 879 - 889, June 1987.

[Chu87b] Chuang, C-I. J., "The Effects of Multipath Delay Spread on Timing Recovery," IEEE Trans. Veh. Tech., vol. VT-35, no. 3, pp. 135 - 140, Aug. 1987.

[Chu90] Chuang, C-I. J.,"Comparison of coherent and Differential Detection of BPSK and QPSK in a Quasistatic Fading Channel," IEEE Trans. Comm., vol. 38, no.5, pp. 565 - 567, July 1991.

[Chu91] Chuang, C-I. J., and Sollenberger, N. R.,"Burst Coherent Demodulation with Combined Symbol Timing, Frequency Offset Estimation, and Diversity Selection," IEEE Trans. Comm., vol. 39, no.7, pp. 1157 - 1164, July 1991.

[Cla68] Clarke, R. H., "A Statistical Theory of Mobile-Radio Reception," The Bell System Tech. Journal, pp. 957 - 1000, July-August, 1968.

[Cou90] Couch II, L. W., *Digital and Analog Communication Systems*, Macmillan, 1990.

[Cox91] Cox, D. C., "A Radio System Proposal for Widespread Low-Power Tetherless Communications," IEEE Trans. Comm., vol. 39, no.2, pp. 324 -3 35, Feb. 1991.

[Cro76] Crow, E. L., and Miles, M. J., "A Minimum Cost, Accurate Statistical Method to Measure Bit Error Rates," Int. Conf. Computer Comm. Rec., pp. 631 - 635, 1976.

[DEC89] DECT: "Digital European Cordless Telecommunication System - Common Interface Specifications," Code: RES-3(89), DECT, European Telecommunications Standarts Institute, Nice, France, 1989.

[Dev87] Devasirvatham, D. M. J., "Multipath Time Delay Jitter Measured at 850 MHz in the Portable Radio Environment," IEEE Journal on Sel. areas in Comm., vol. SAC-5, no.5, pp. 855 - 861, June 1987.

[EIA89] Electronic Industries Association, "Dual-Mode Subscriber Equipment-Network Equipment Compatibility Specification," Interim Standard 54, Dec. 1989.

[Fun91] Fung, V., "Simulation of BER Performance of FSK, BPSK, /4 DQPSK in Flat and Frequency-Selective Fading Channels," Masters Thesis in Electrical Engineering, Virginia Polytechnic Institute and State University, Blacksburg VA, August, 1991.

[Fun92] Fung, V., Rappaport T. S., and Thoma, B., "Bit Error Simulation for $\pi/4$ DQPSK Mobile Radio Communications using Two-Ray and Measurement-Based Impulse Response Models,"accepted for publication in IEEE Journal on Selected Areas in Communications, MPRG at Virginia Tech, June 1992.

[Gar80] Gardner, F. M., "Self-Noise in Synchronizers," IEEE Trans. Comm., vol. COM - 28, no.8, pp. 1159 - 1163, August 1980.

[Gla83] Glance, B., and Greenstein, L. J., "Frequency-Selective Fading Effects in Digital Mobile Radio with Diversity Combining," IEEE Trans. Comm., vol. COM-31, no.9, pp.1085-1094, Sept. 1983.

[Goo91a] Goodman, D. J., "Trends in Cellular and Cordless Communications," IEEE Communications Magazine, vol. 29, no.6, June 1991.

[Goo91b] Goodman, K., "Second Generation Wireless Information Networks," IEEE Trans. Veh. Tech., vol. VT-40, no. 2, pp. 366 - 374, Aug. 1991.

[GSM88] GSM, "Physical Layer on the Radio-Path," GSM 05.04, Release GSM/PN, European Telecommunications Standards Institute, Nice, France, July 1988.

[Hol80] Holmes, J. K., "Tracking Performance of the Filter and Square Bit Synchronizer," IEEE Trans. Comm., COM-28, pp. 1154 - 1158, 1980.

[Hua91] Huang, W. H., "Simulation of Adaptive Equalization in Two-Ray, SIRCIM, and SMRCIM Mobile Radio Channels," Masters Thesis in Electrical Engineering, Virginia Polytechnic Institute and State University, Blacksburg VA, December, 1991.

[Jak74] Jakes, W. C., *Microwave Mobile Communications*, John Wiley & Sons, 1974.

[Joh82] Johnson, L. W., and Riess, R. D., *Numerical Analysis*, second edition, Addison Wesley, 1982.

[Kad92] Kadel, G., "Determination of the GSM-System Performance from Wideband Propagation Measurements," IEEE Veh. Tech. Conf., pp. 540 - 541, Denver 1992.

[Kam87] Kamen, E. W., *Introduction to signals and Systems*, Macmillan Publ. Company, New York 1987.

[Le92] Le, M. T., and Sheikh, A. U., "Performance of $\pi/4$-DQPSK in a Rayleigh/Log-normal/Delay Spread/AWGN/Cochannel Interference Environment," IEEE Veh. Tech. Conf., pp. 147 - 150, Denver 1992.

[Lee82] Lee, W. C. Y., *Mobile Communications Engineering*, McGraw-Hill, 1982.

[Lee89] Lee, W. C. Y., *Mobile Cellular Telecommunications Systems*, McGraw-Hill, 1989.

[Liu89] Liu, C. L., and Feher, K., "Noncoherent Detection of $\pi/4$-DQPSK Systems in a CCI-AWGN Combined Interference Environment," Proceedings of the IEEE Veh. Tech. Conf., pp. 83 - 94, May, 1989.

[Liu90] Liu, C. L., and Feher, K., "Performance of Non-Coherent $\pi/4$-QPSK in a Frequency-Selective Fast Rayleigh Fading Channel," IEEE Int. Conf. on Comm., pp. 335.7.1. - 335.7.5., Atlanta, Apr. 1990.

[Liu91] Liu, C. L., and Feher, K., "Bit Error Rate Performance of $\pi/4$-QPSK in a Frequency-Selective Fast Rayleigh Fading Channel," IEEE Trans. Veh. Tech., vol. VT-40, no. 3, pp. 558 - 568, Aug. 1991.

[Lor91] Lorenz, R. W., and Kadel, G., "Propagation Measurement Using a Digital Channel Sounder Matched to the GSM-System Bandwidth," IEEE Int. Conf. on Comm., pp. 548 - 552, Denver, June 1991.

[Luc89] Lucky, R. W., *Silicon Dreams: Information, Man, and Machine*, St. Martin's Press New York, 1989.

[Mar90] Marcos, S., Macchi, O., "Statistical Properties of Timing Jitter Due to Data Echo in Digital Modem Receivers," IEEE Trans. Comm., vol. 38, no.4, pp. 546 - 550, April 1990.

[Ohn90] Ohno, K., and Adachi, F., "Postdetection Diversity Reception of QDPSK Signals Under Frequency Selective Rayleigh Fading," IEEE Veh. Technol. Conf., pp. 431 - 436, Orlando, 1990.

[Oer88] Oerder, M., Meyr, H., "Digital Filter and Square Timing Recovery," IEEE Trans. Comm., vol. 36, no.5, pp. 605 - 612, May 1988.

[Par92] Parsons, J. D., *The Mobile Radio Propagation Channel*, Halsted Press, 1992.

[Par89] Parsons, J. D., and Gardiner., J. G., *Mobile Communication Systems*, Blackie, 1989.

[Pre90] Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T., *Numerical Recipes in C*, Cambridge University Press, 1989.

[Pro89] Proakis, J. G., *Digital Communications*, McGraw-Hill, 1989.

[Rap88] Rappaport, T. S., "Delay spread and time delay jitter for the UHF factory multipath channel," IEEE Veh. Tech. Conf., pp. 186 - 189, Philadelphia 1988.

[Rap89] Rappaport, T. S., "Indoor Radio Communications for Factories of the Future," IEEE Comm. Magazine, vol. 27, no.5, pp. 15 - 24, May 1989.

[Rap91a] Rappaport, T. S., Seidel, S. Y., and Takamizawa, K., "Statistical Channel Impulse Response Models for Factory and Open Plan Building Radio Communication System Design," IEEE Trans. Comm., vol. 39, no.5, pp. 794 - 807, May 1991.

[Rap91b] Rappaport, T. S., "The Wireless Revolution," IEEE Comm. Magazine, vol. 29, no.11, pp. 52 - 71, Nov. 1991.

[Rap91c] Rappaport, T. S., and Fung, V., "Simulation of Bit Error Performance of FSK, BPSK, and /4 DQPSK in Flat Fading Indoor Radio Channels Using a Measurement-Based Channel Model," IEEE Trans. Vech Tech.,vol. VT-40, no. 4, pp. 731 - 741, Nov. 1991.

[Rap91d] Rappaport, T. S., Fung, V., and Keitz M. D.," Computer-Based Bit Error Simulation for Digital Wireless Communication," Application for United States Letters Patent, Docket CIT.036, May 1991.

[Rap92] Rappaport, T. S., Seidel, S. Y., and Schaubach, K. R., "Site-Specific Propagation Prediction for PCS System Design," 2nd Symposium on Wireless Personal Communications., pp. 16-1 - 16-27, Blacksburg 1992.

[Ric48] Rice, S. O., "Statistical Properties of a Sine Wave Plus Random Noise," Bell System Tech. Journal, vol. 27, pp. 109 - 157, Jan. 1948.

[Sei89] Seidel, S. Y., "UHF Indoor Radio Channel Models for Manufacturing Environments," Masters Thesis in Electrical Engineering, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, Aug. 1989.

[Sha88] Shanmugan, K. S., "An Update on Software Packages for Simulation of Communication Systems (Links)," IEEE Journal on Sel. areas in Comm., vol. 6, no.1, pp. 5 - 12, Jan. 1988.

[Sow88] Sowerby, K. W., and Williamson, A. G., "Outage Probability in Mobile Radio Systems with Multiple Interferers," Electron. Letters, vol. 24, no. 17, Aug. 1988, pp. 1073 - 1075.

[Sow92] Sowerby, K. W., and Williamson, A. G., "Estimating Reception Reliability in Cellular Mobile Radio Systems," IEEE Veh. Tech. Conf , pp. 151 - 154, Denver 1992.

[Swa90] Swain, R. S., and Holmes, D. W. J., "The Digital Cordless Telecommunications Common Air Interface," British Telecom Technol. J., vol. 8, no. 1, pp. 12 - 18, Jan. 1990.

[SUN90] SUN Microsystems, *Utilities & Libraries,* Revision A, March 1990.

[Tho92a] Thoma, B., "BERSIM V2.0, Bit Error Simulator for Mobile and Portable Radio Communication Systems," MPRG Technical Report MPRG-TR-92-06, Bradley Department of Electrical Engineering at Virginia Tech, February 1992.

[Tho92b] Thoma, B., "BERSIM V2.0, program listing," MPRG Technical Report MPRG-TR-92-13, Bradley Department of Electrical Engineering at Virginia Tech, July 1992.

[TTL91] Telesis Technologies Laboratory, Experimental License Report to the FCC, August 1991.

[Tur92] Turkmani, A. M. D., and Ladki, M. "A Study of The Error Distributions in the Wideband Mobile Radio Channel," IEEE Veh. Tech. Conf., pp. 355 - 358, Denver 1992.

Appendix A.
Flow Charts and Structure Charts for BERSIM 2.0

Appendix A.1 Flow Charts and Structure Charts for program *intface*

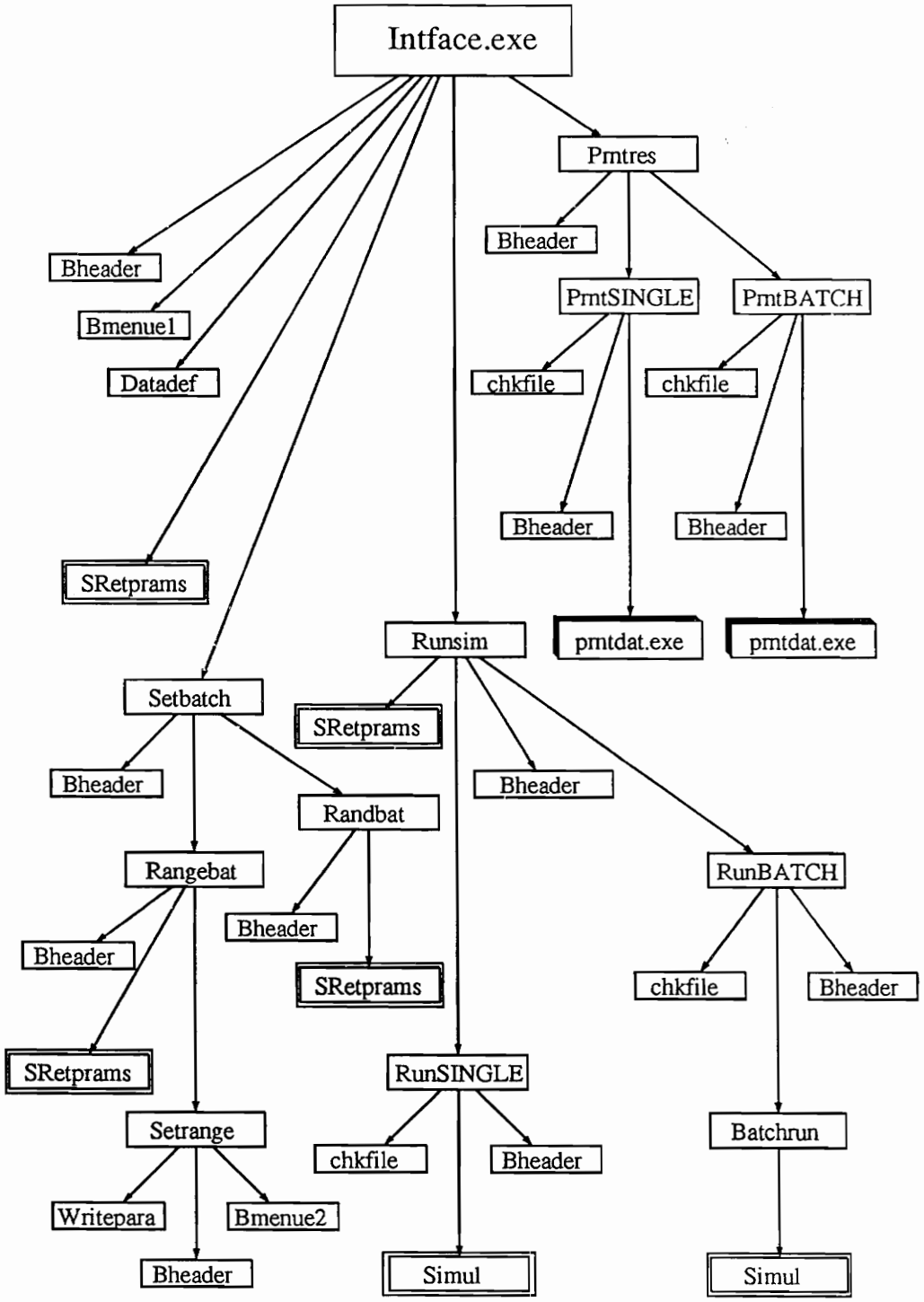


FIGURE A.1: Structure chart for program intface

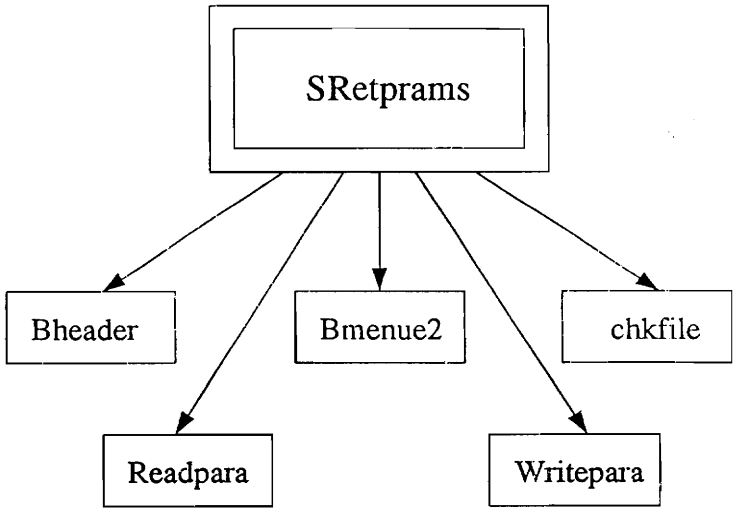


FIGURE A.2: Structure chart for function SRetprams in program intface

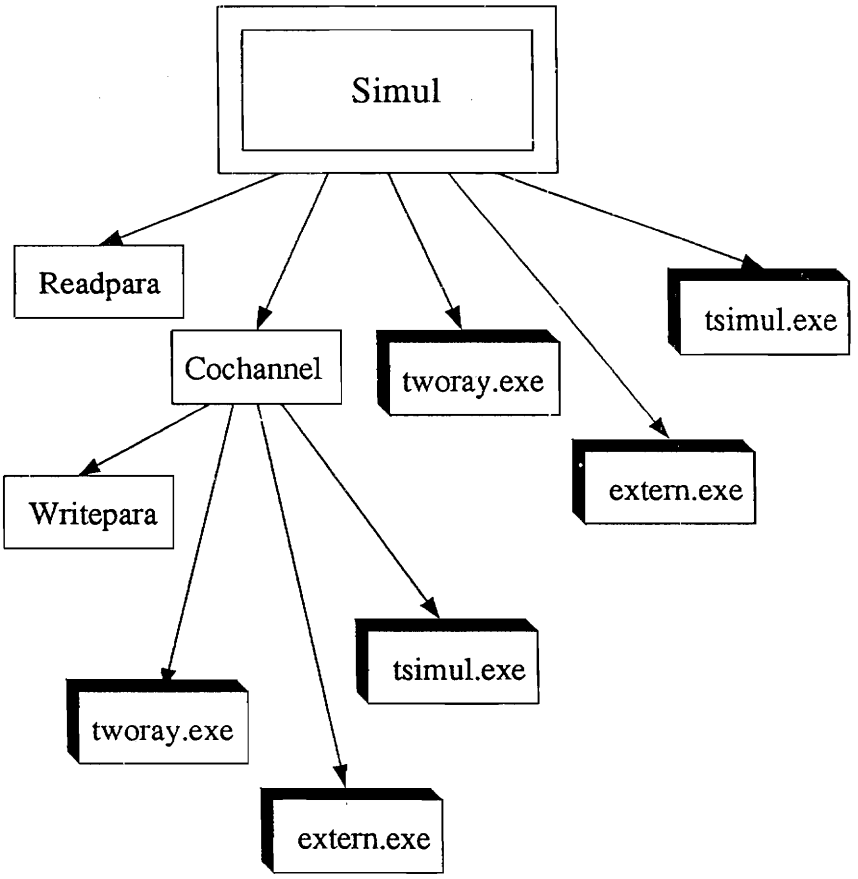


FIGURE A.3: Structure chart for function Simul in program intface

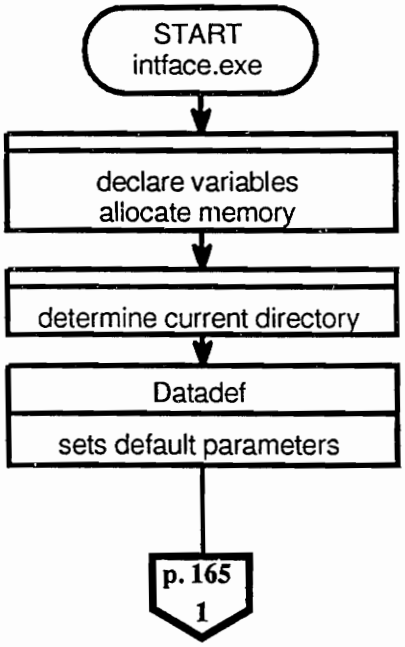


FIGURE A.4: Flow chart 1 for program intface

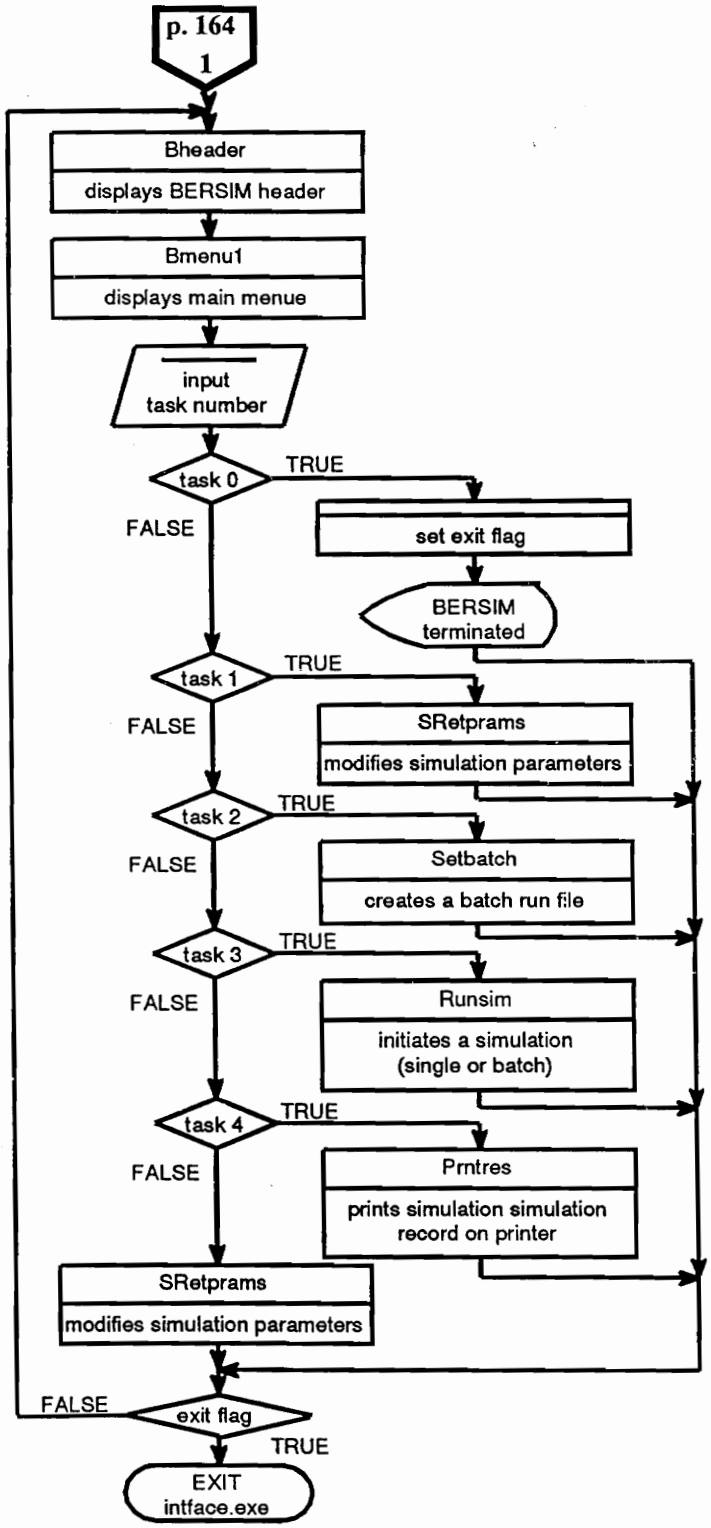


FIGURE A.5: Flow chart 1 for program intface (continued)

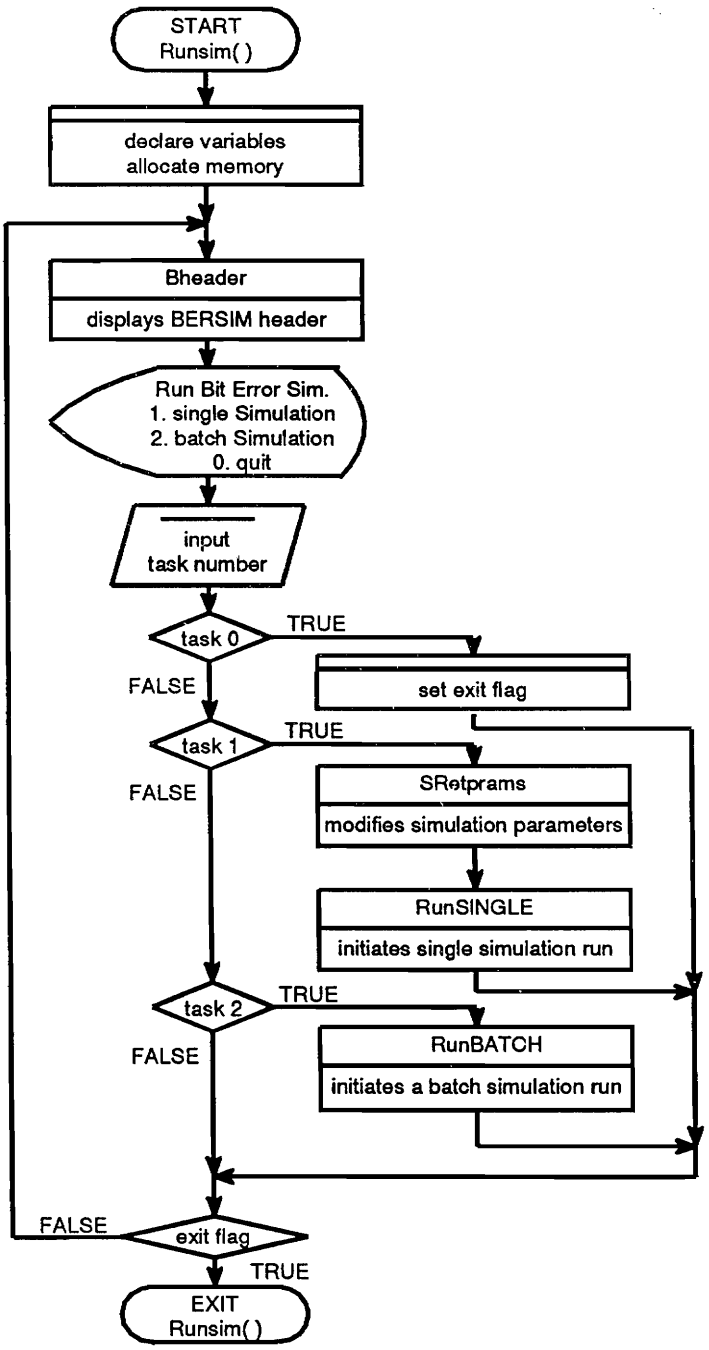


FIGURE A.6: Flow chart 2 for program intface

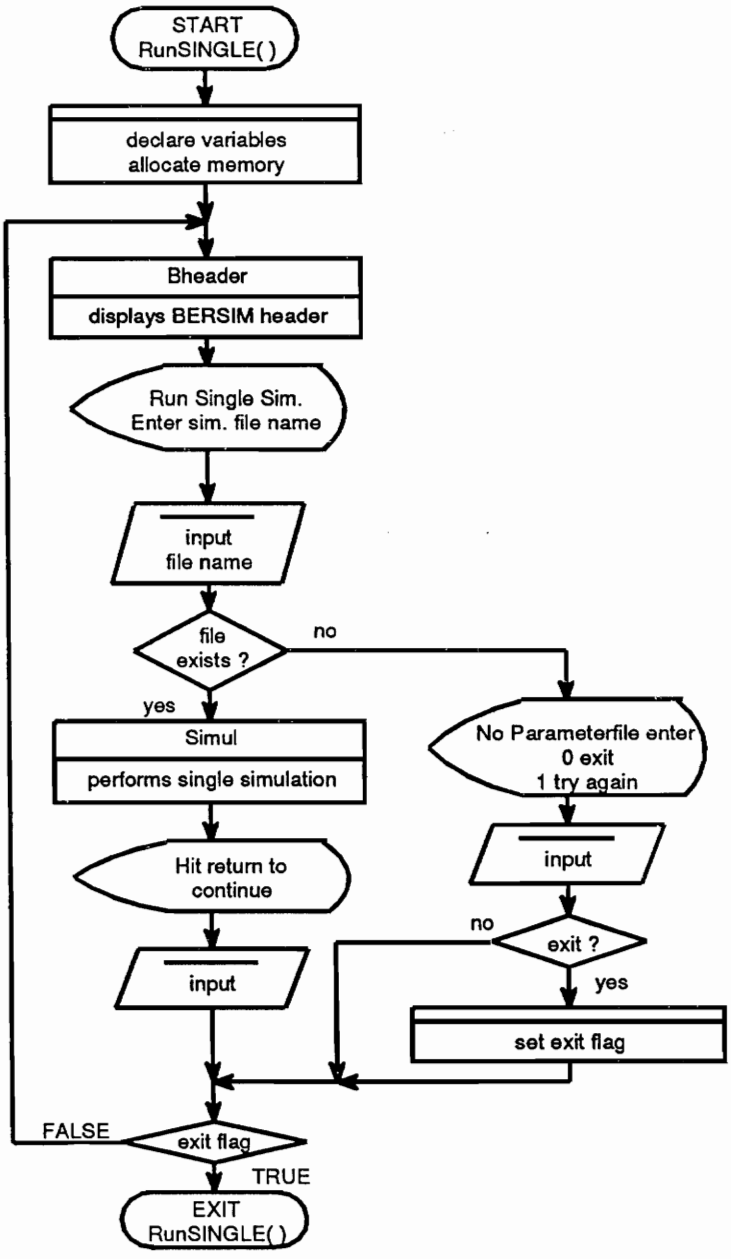


FIGURE A.7: Flow chart 3 for program intface

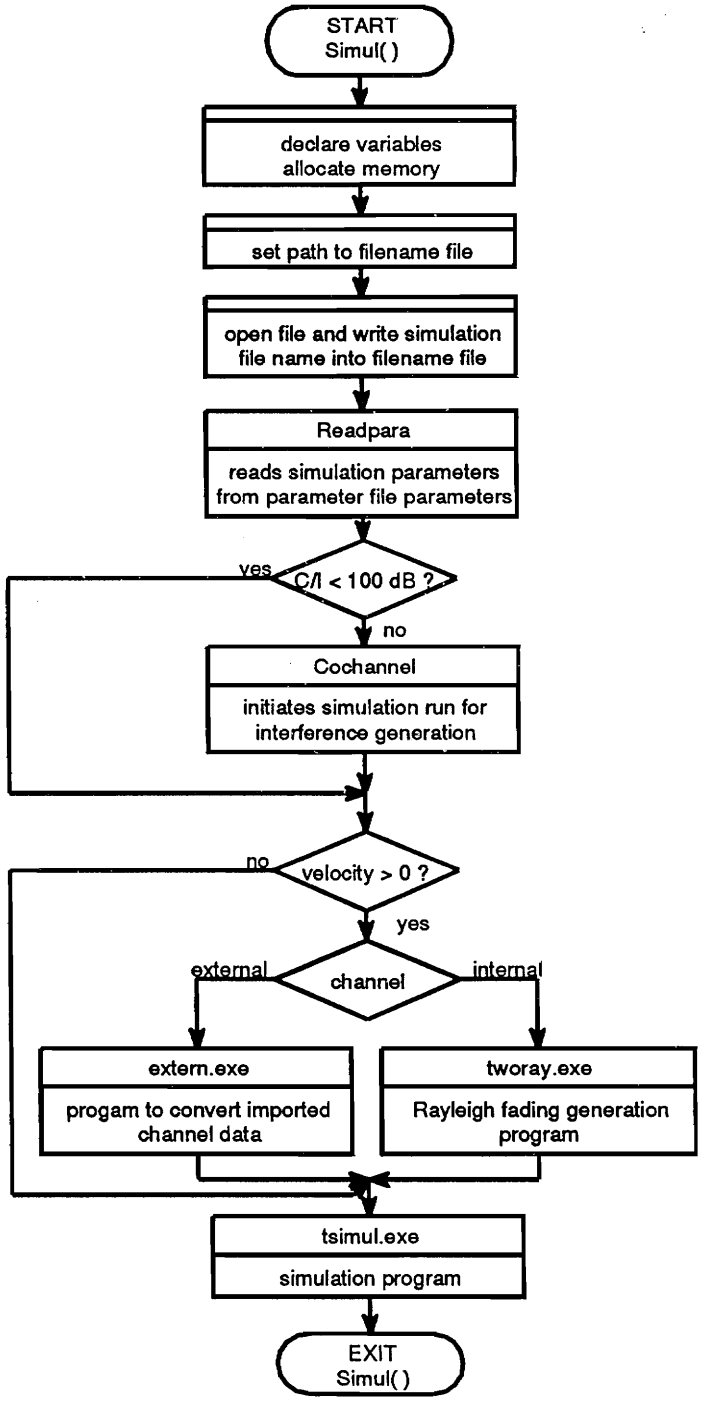


FIGURE A.8: Flow chart4 for program intface

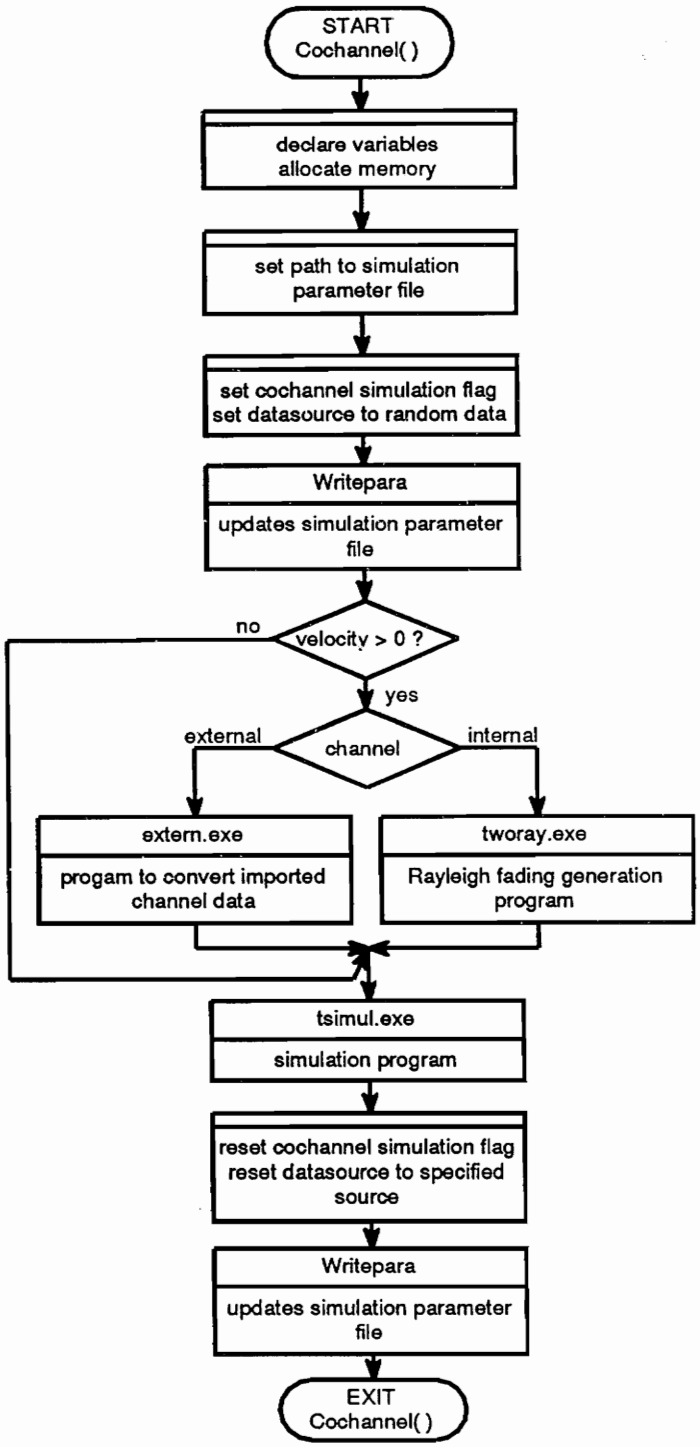


FIGURE A.9: Flow chart 5 for program intface

Appendix A.2 Flow Charts and Structure Charts for program *tworay*

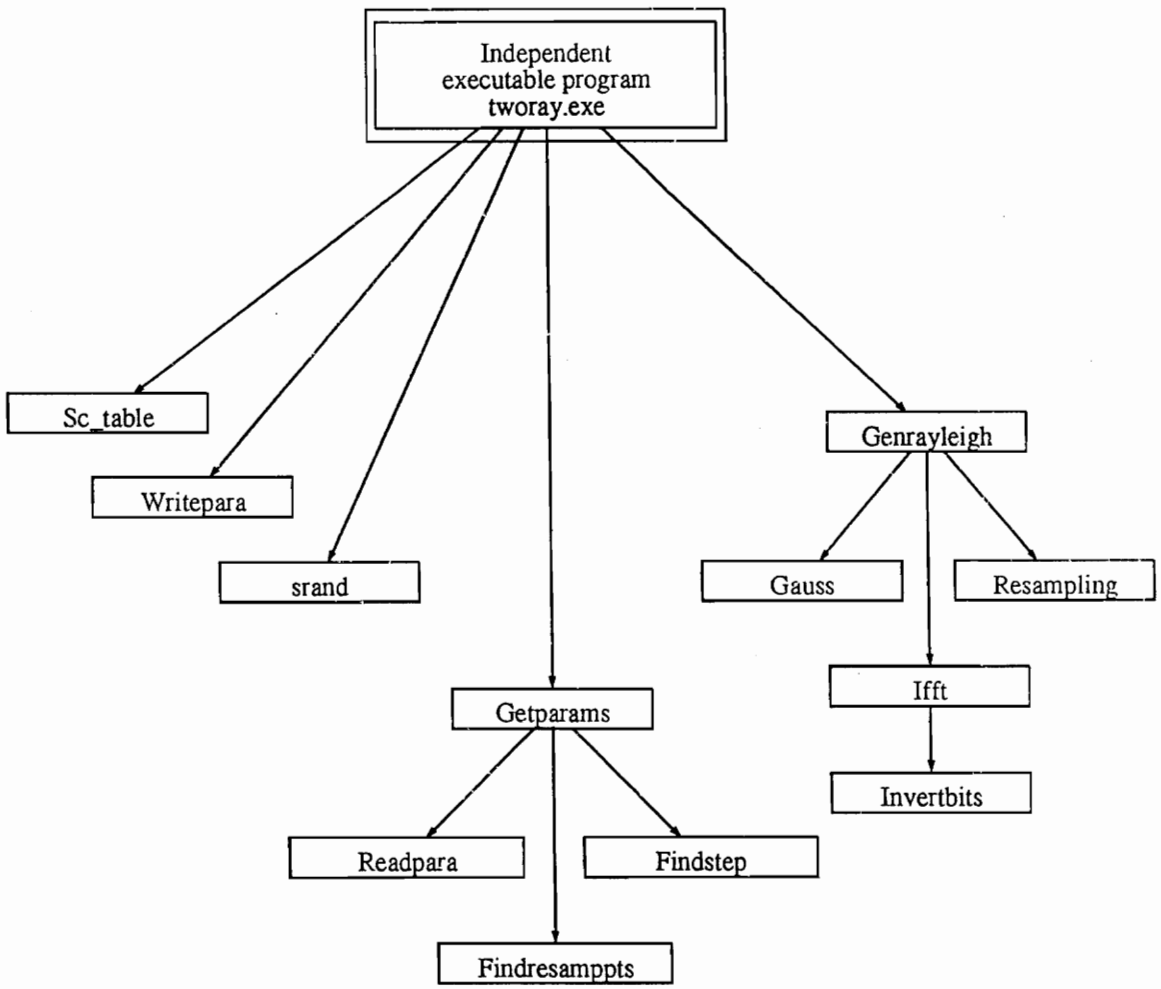


FIGURE A.10: Structure chart for program tworay

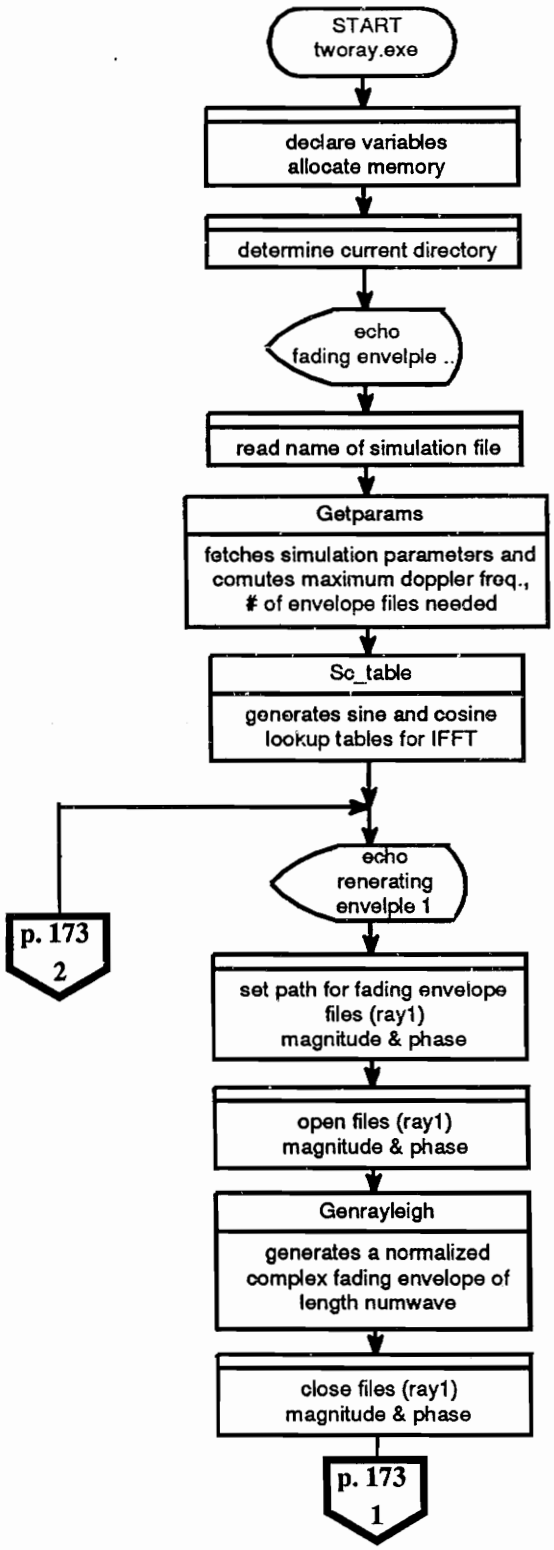


FIGURE A.11: Flow chart 1 for program tworay

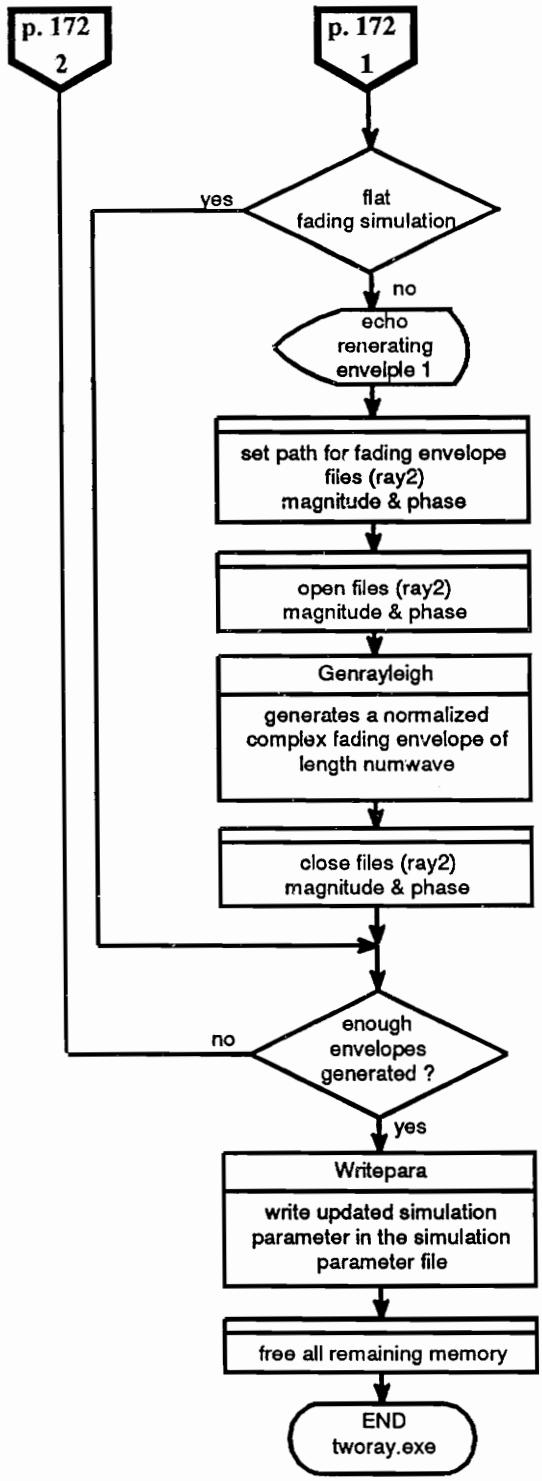


FIGURE A.12: Flow chart 1 for program tworay (continued)

Appendix A.3 Flow Charts and Structure Charts for program *extern*

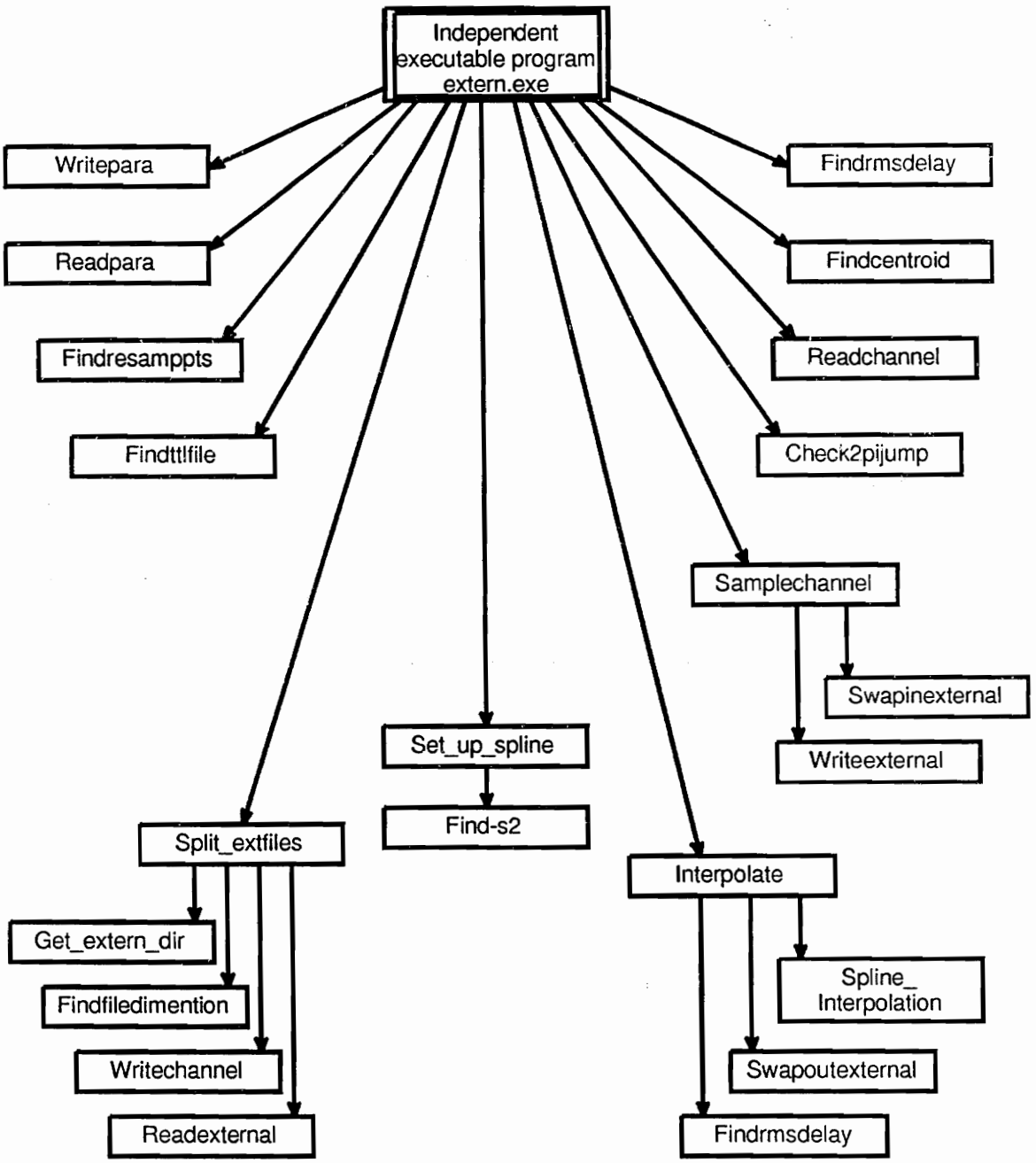


FIGURE A.13: Structure chart for program extern

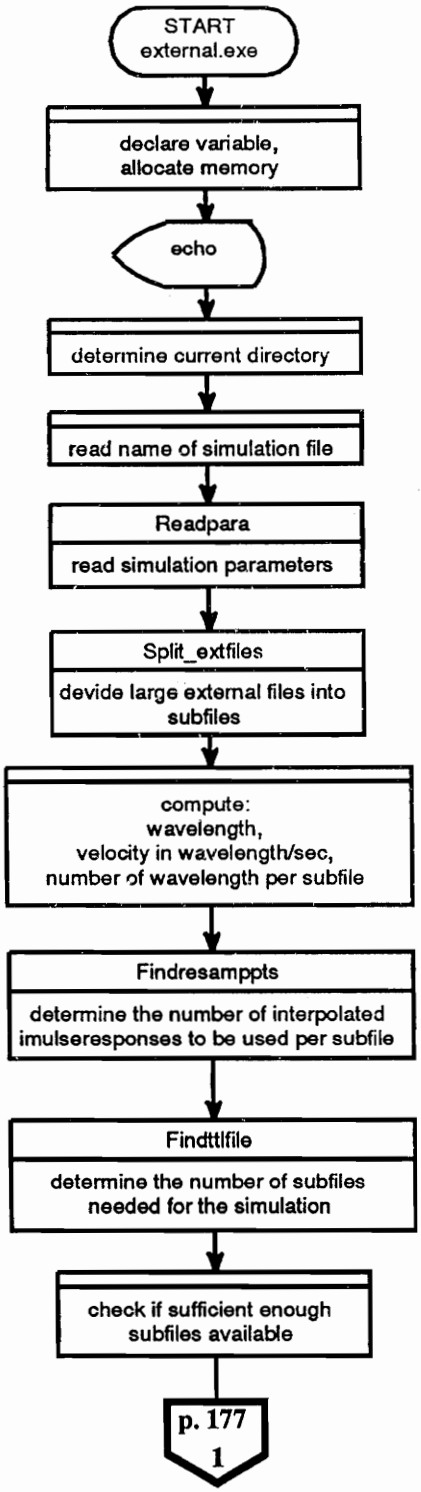


FIGURE A.14: Flow chart 1 for program extern

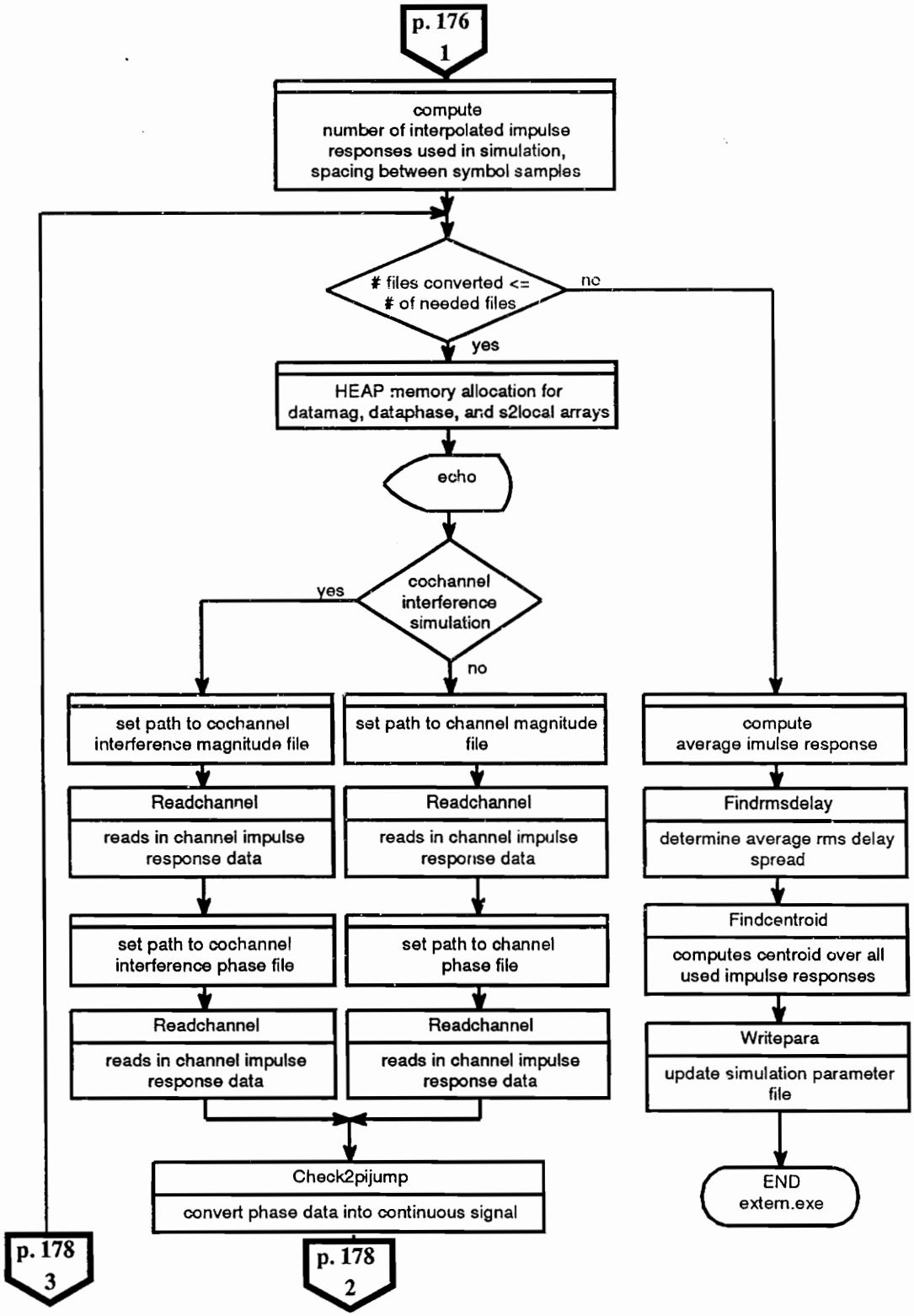


FIGURE A.15: Flow chart 1 for program extern (continued)

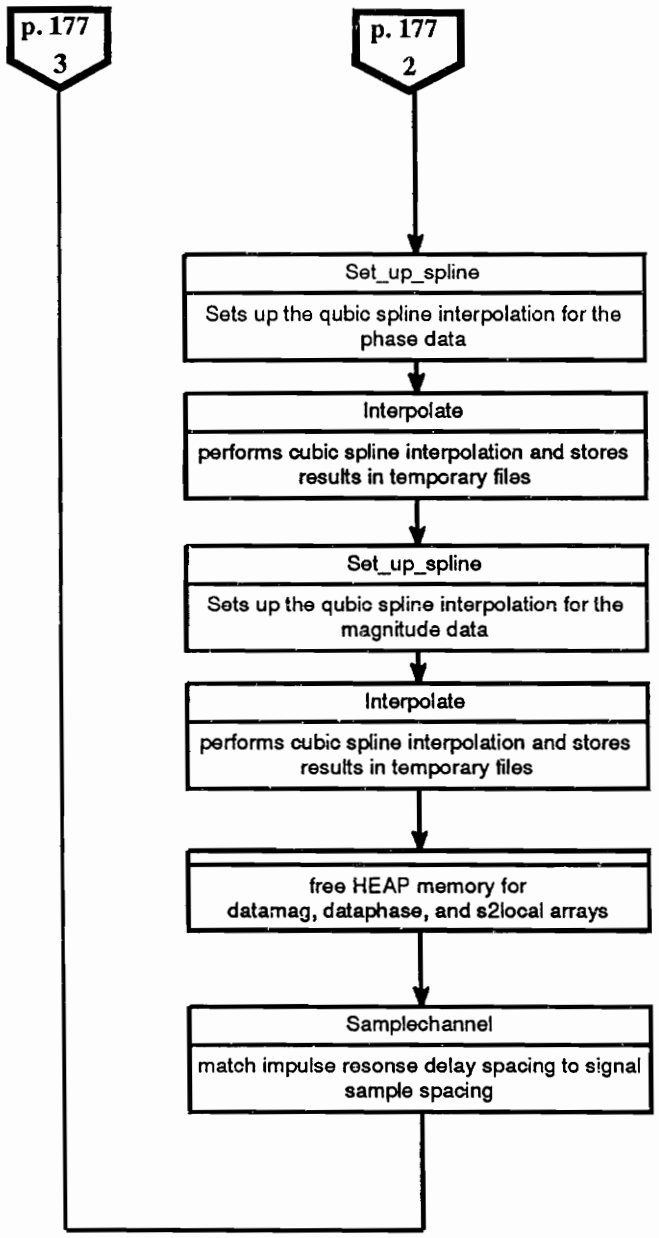


FIGURE A.16: Flow chart 1 for program extern (continued)

Appendix A.4 Flow Charts and Structure Charts for program *tsimul*

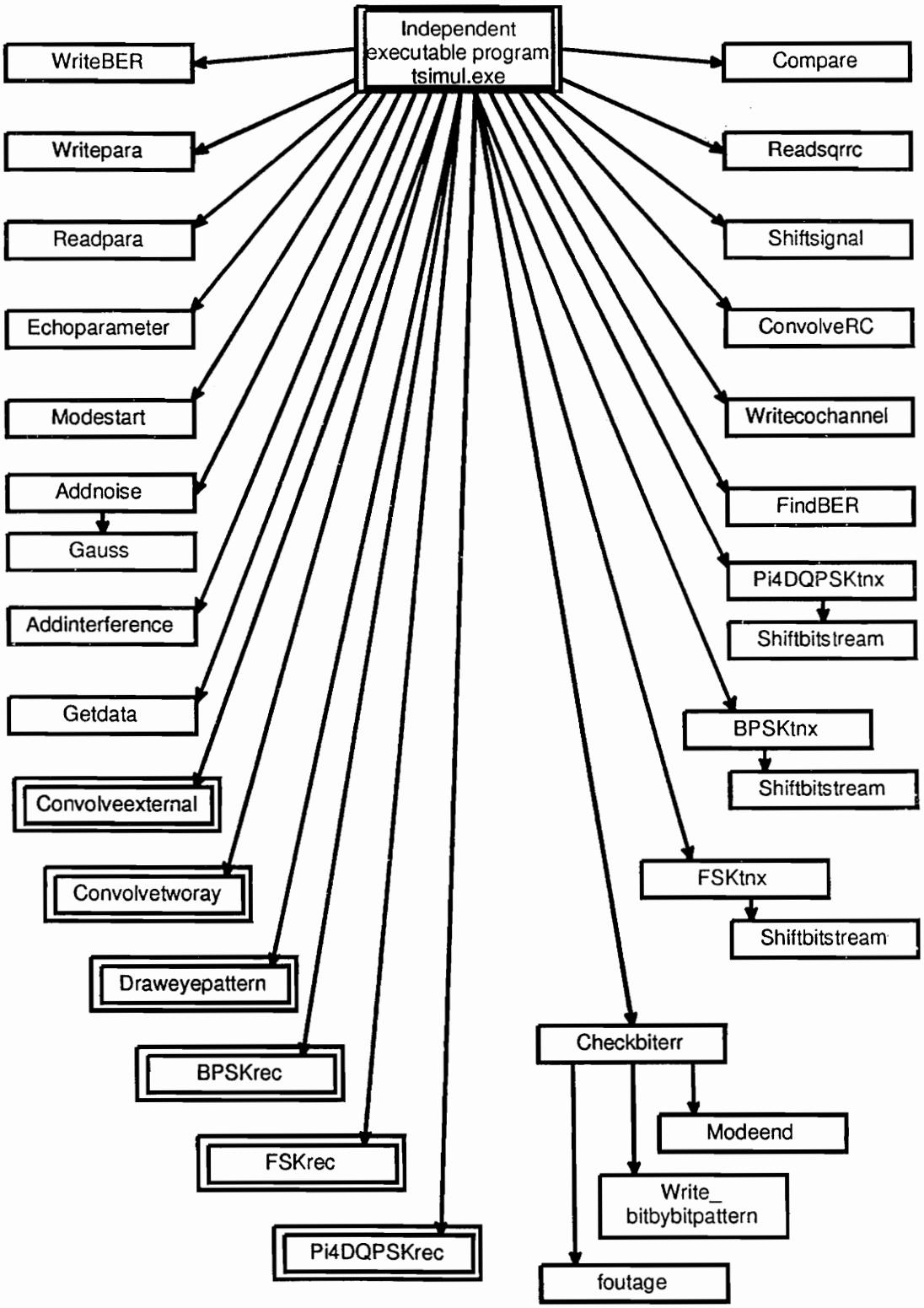


FIGURE A.17: Structure chart for program tsimul

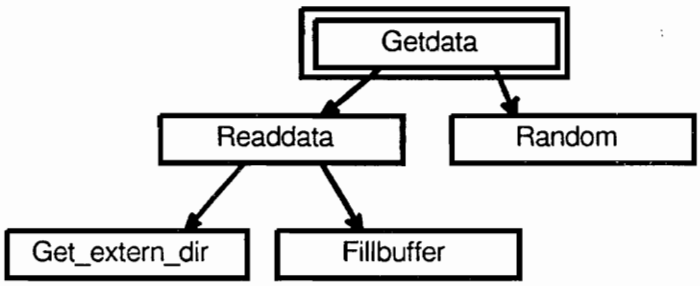


FIGURE A.18: Structure chart for function Getdata in program tsimul

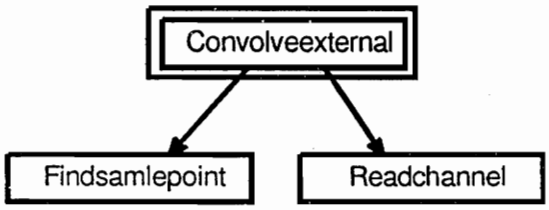


FIGURE A.19: Structure chart for function Convolveexternal in program tsimul

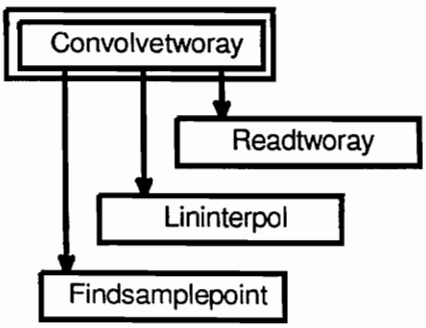


FIGURE A.20: Structure chart for function Convolvetworay in program tsimul

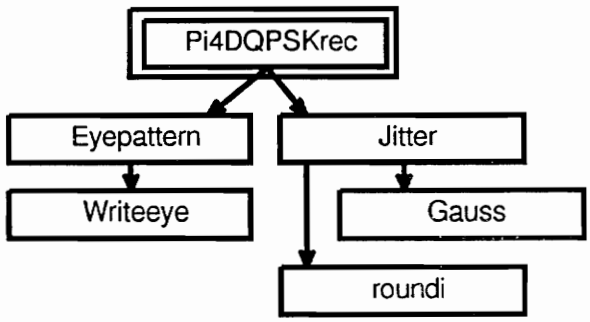


FIGURE A.21: Structure chart for $\pi/4$ DQPSK receiver function in program tsimul

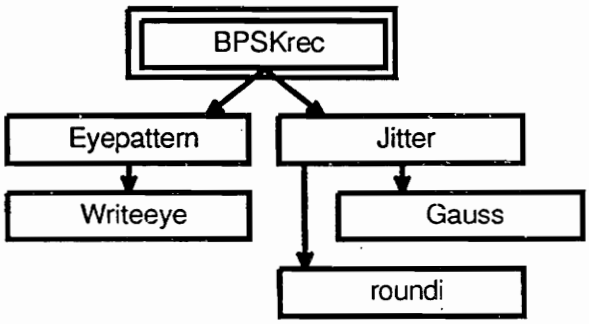


FIGURE A.22: Structure chart for BPSK receiver function in program tsimul

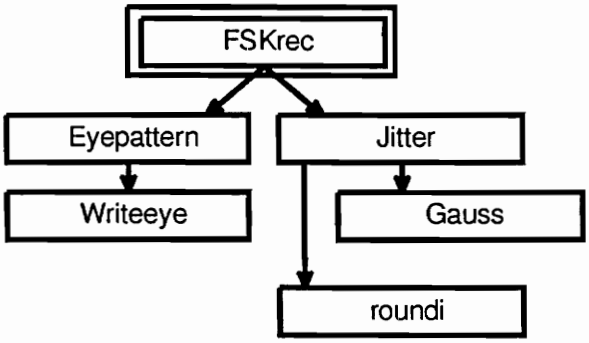


FIGURE A.23: Structure chart for FSK receiver function in program tsimul

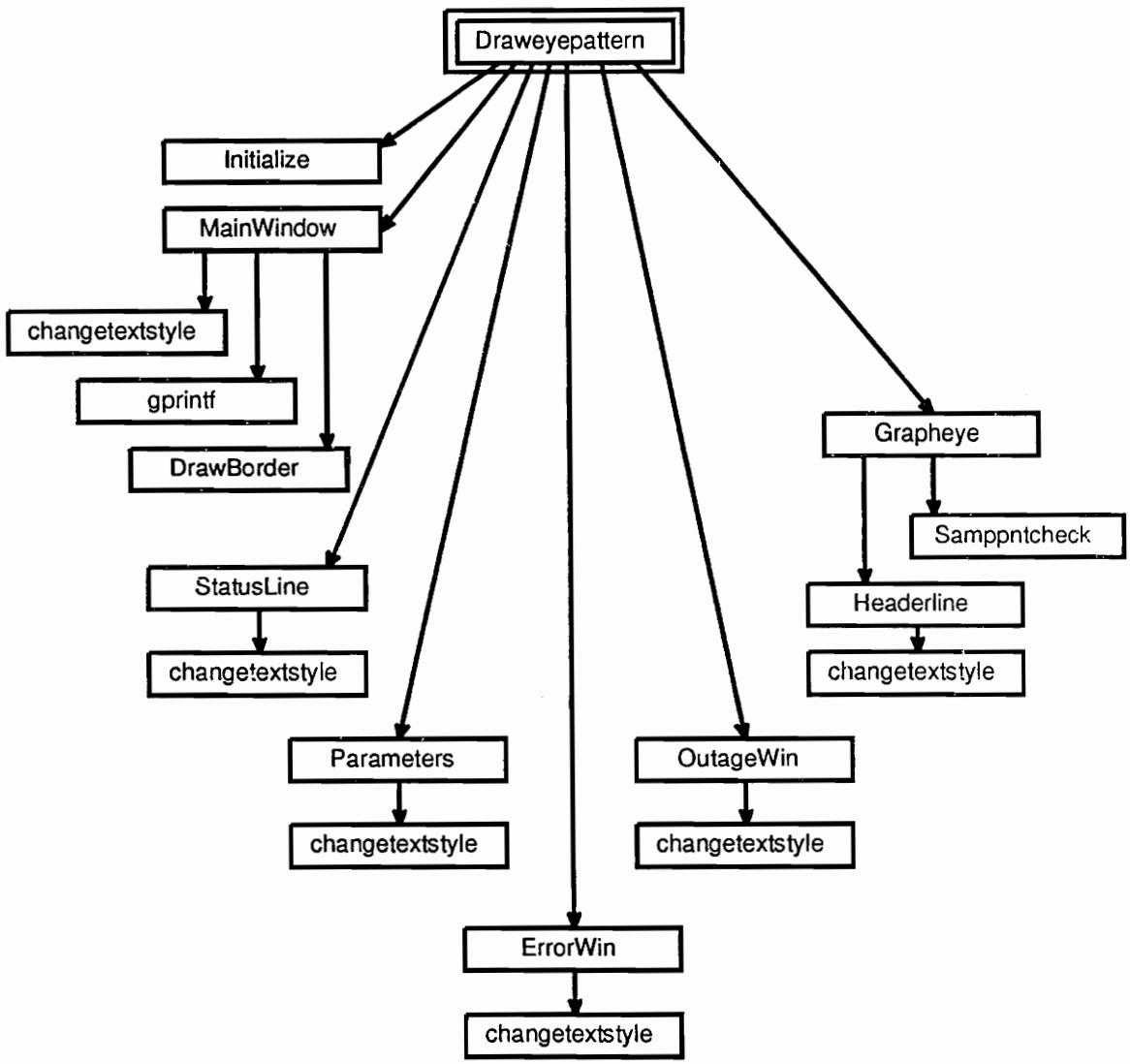


FIGURE A.24: Structure chart for function Draweyepattern in program tsimul

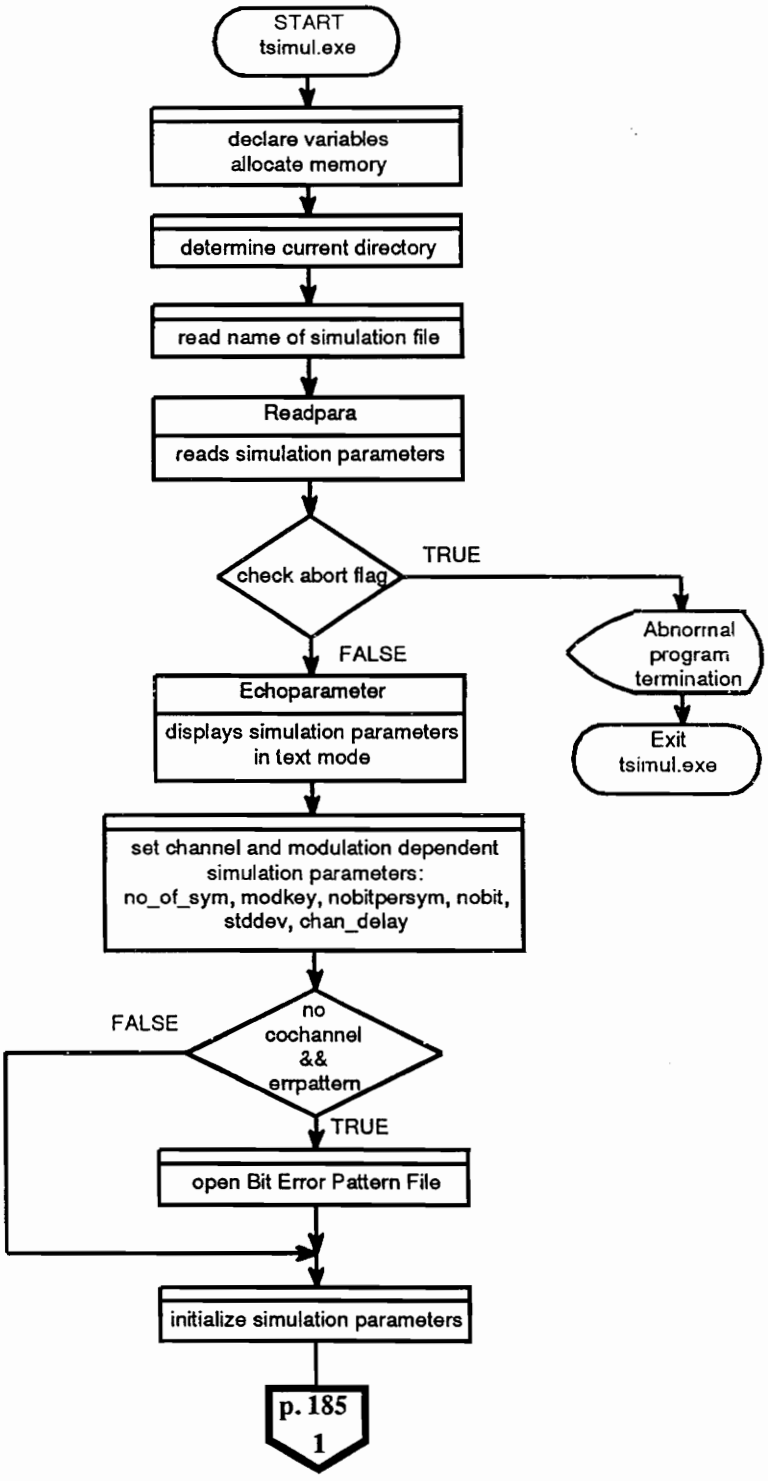


FIGURE A.25: Flow chart 1 for program tsimul

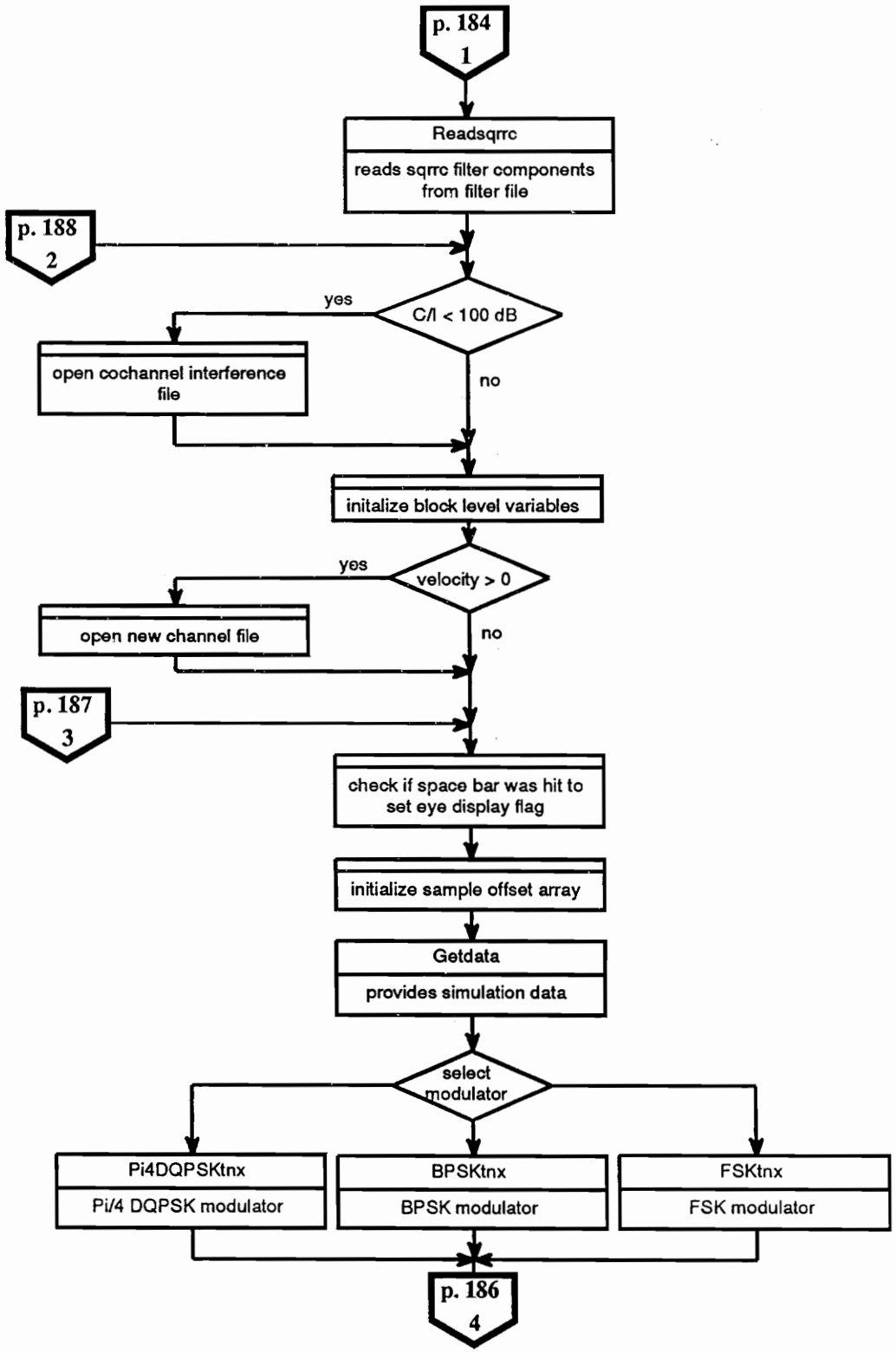


FIGURE A.26: Flow chart 1 for program tsimul (continued)

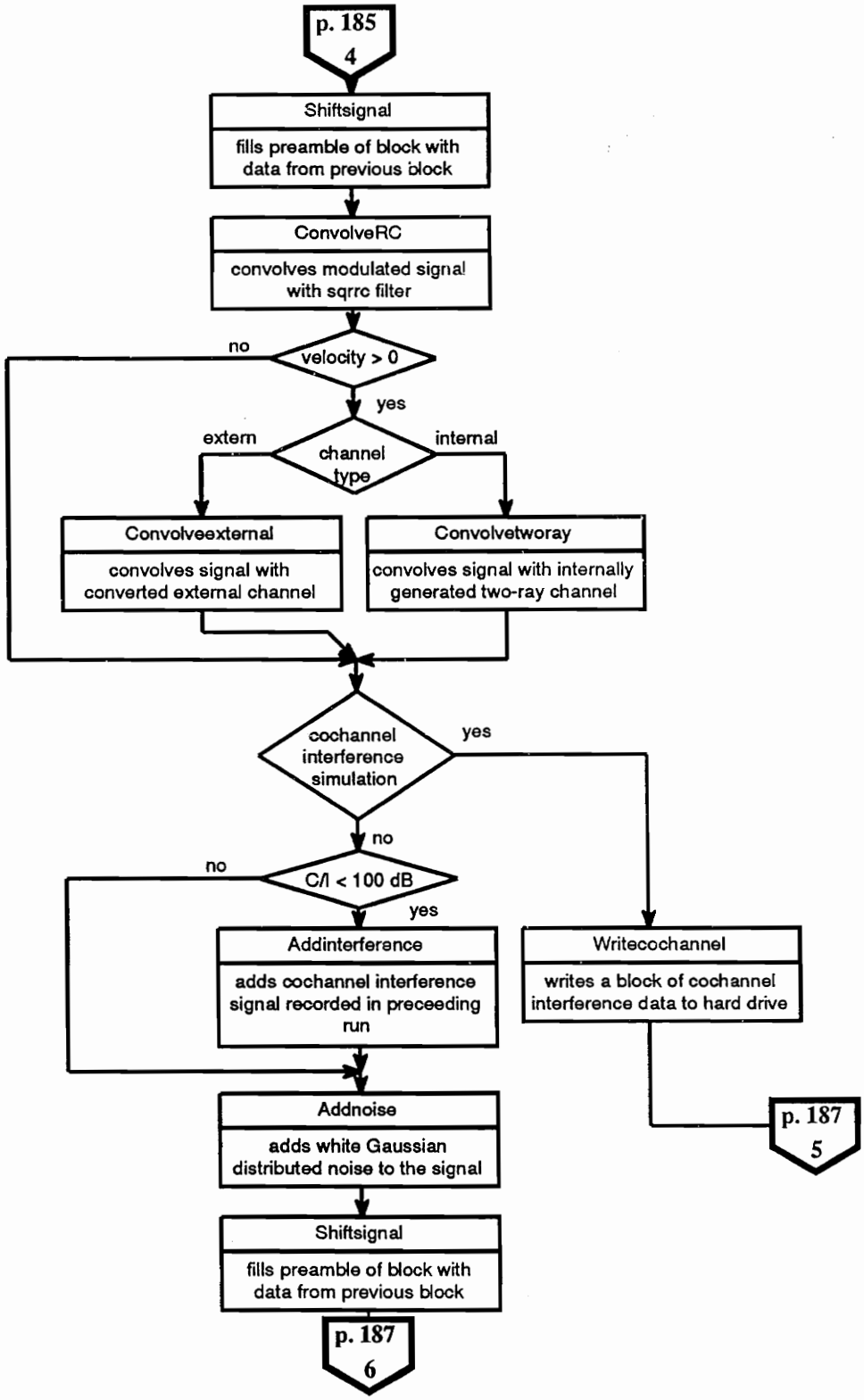


FIGURE A.27: Flow chart 1 for program tsimul (continued)

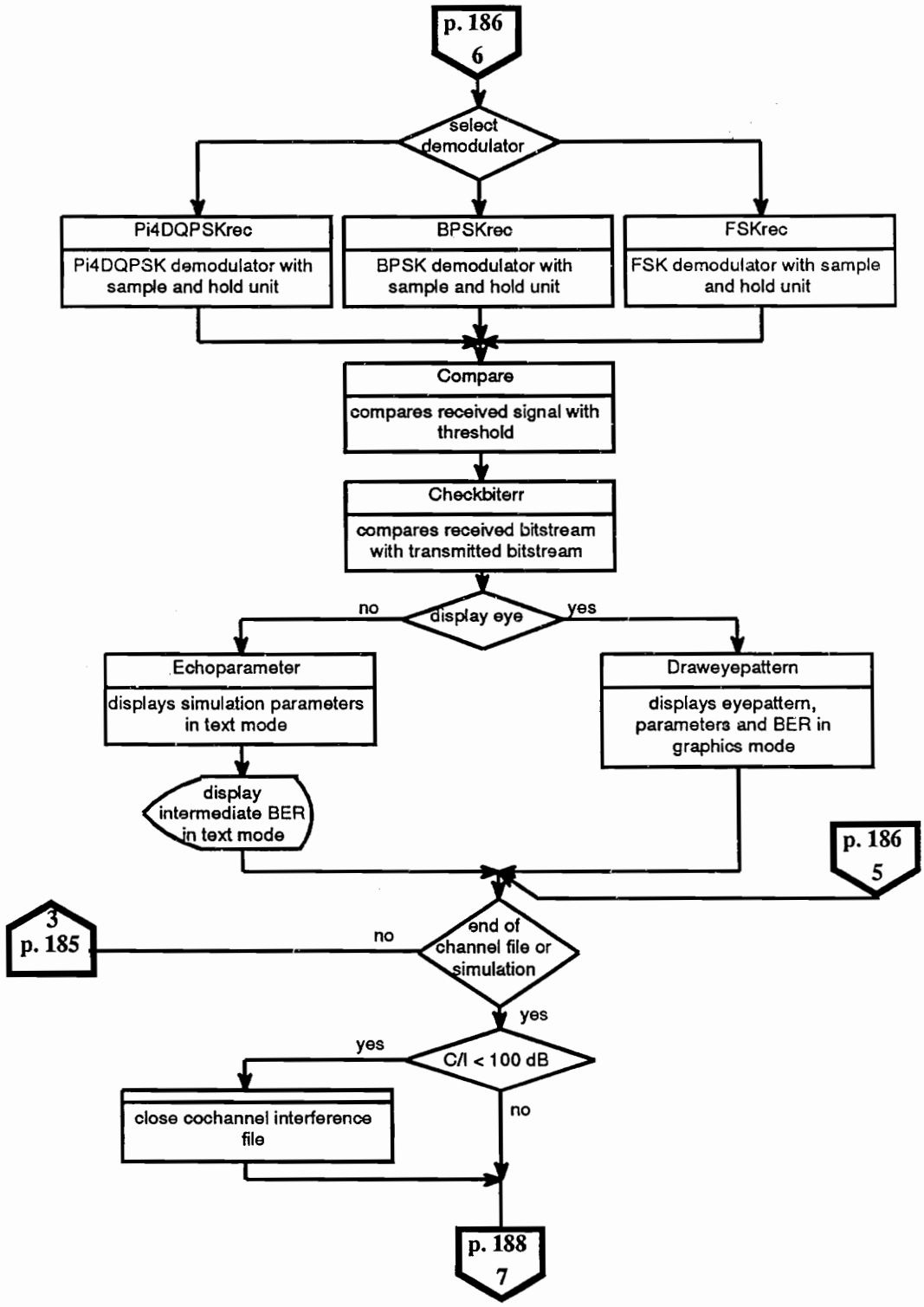


FIGURE A.28: Flow chart 1 for program tsimul (continued)

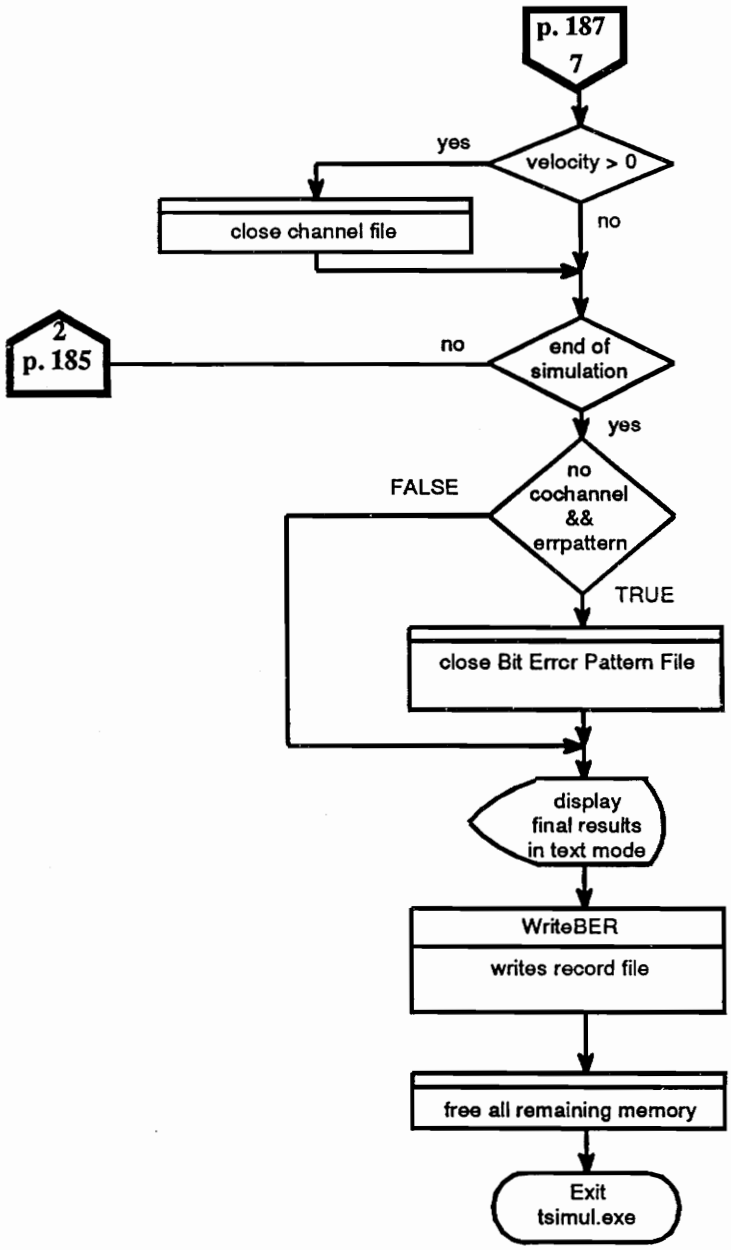


FIGURE A.29: Flow chart 1 for program tsimul (continued)

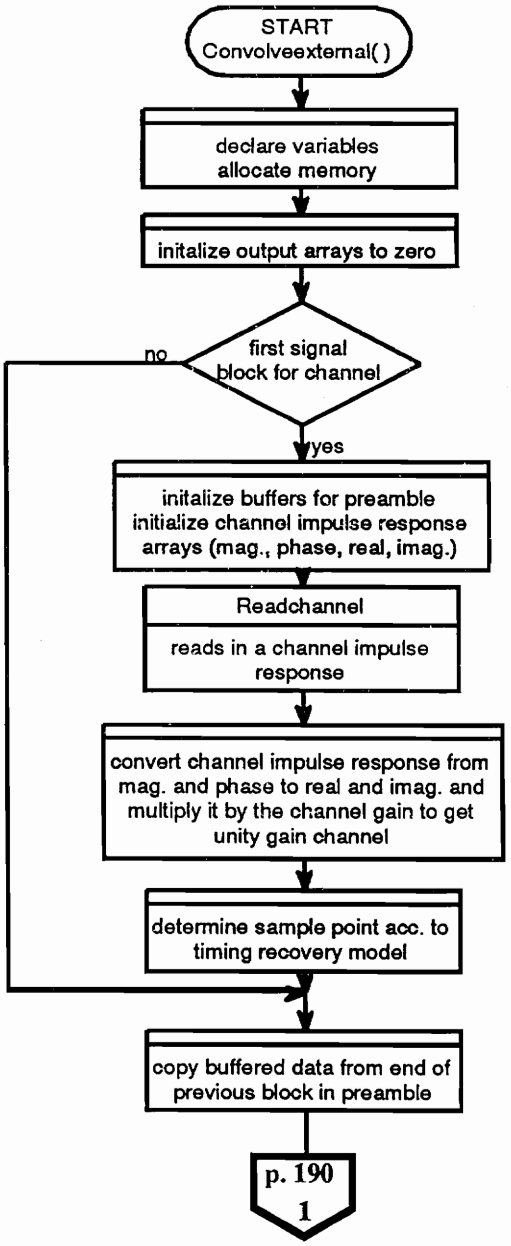


FIGURE A.30: Flow chart 2 for program tsimul

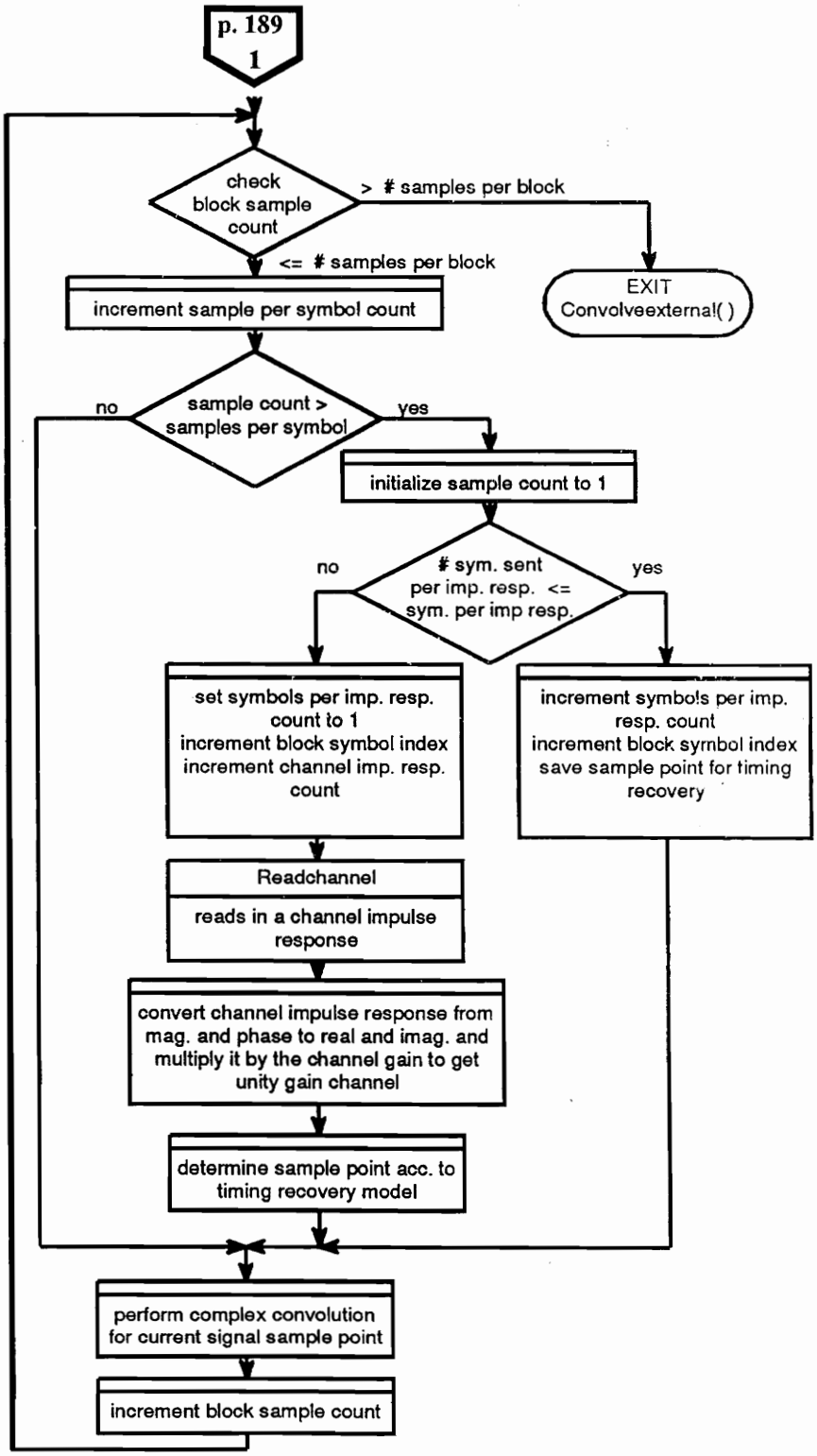


FIGURE A.31: Flow chart 2 for program tsimul (continued)

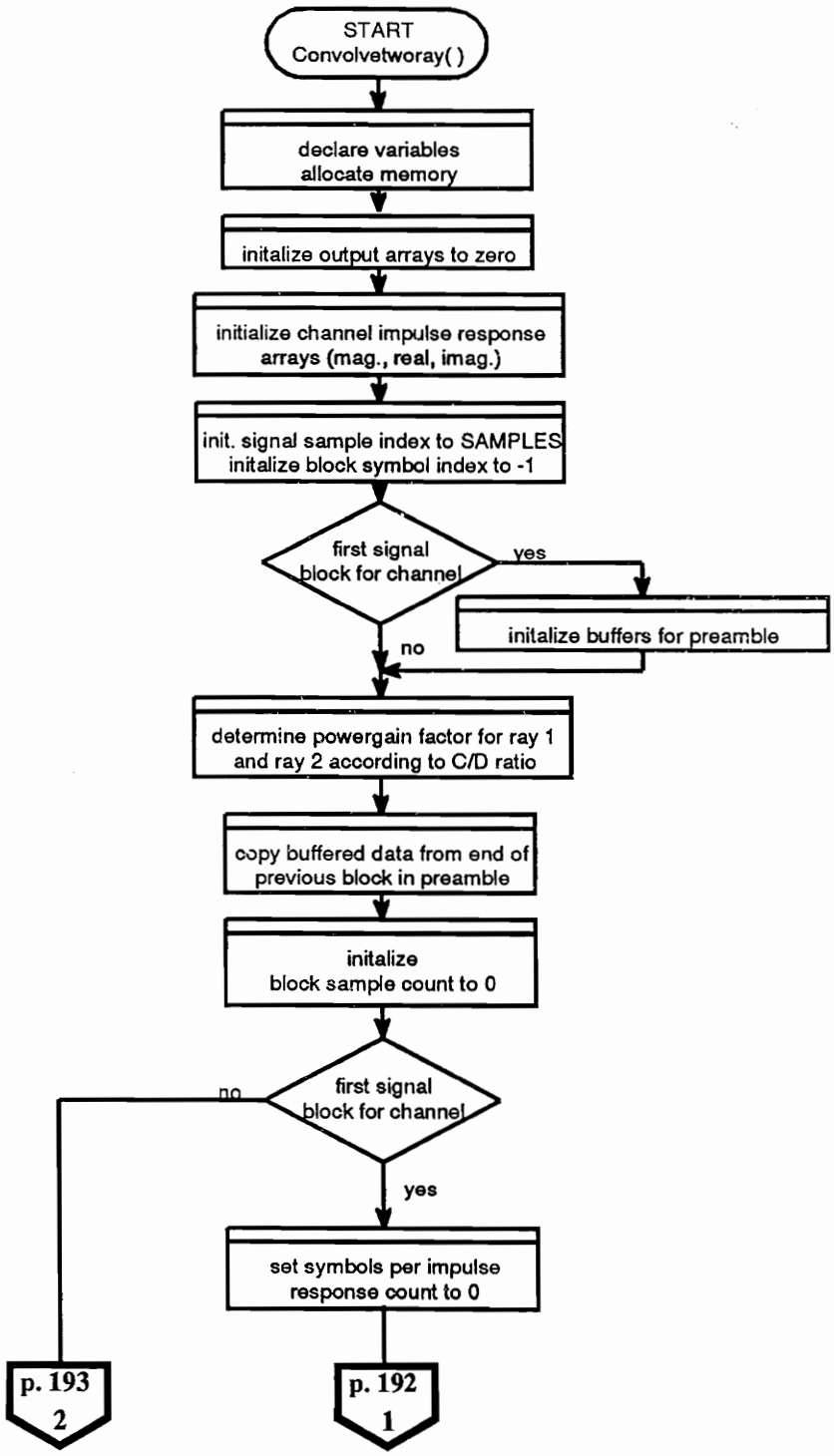


FIGURE A.32: Flow chart 3 for program tsimul

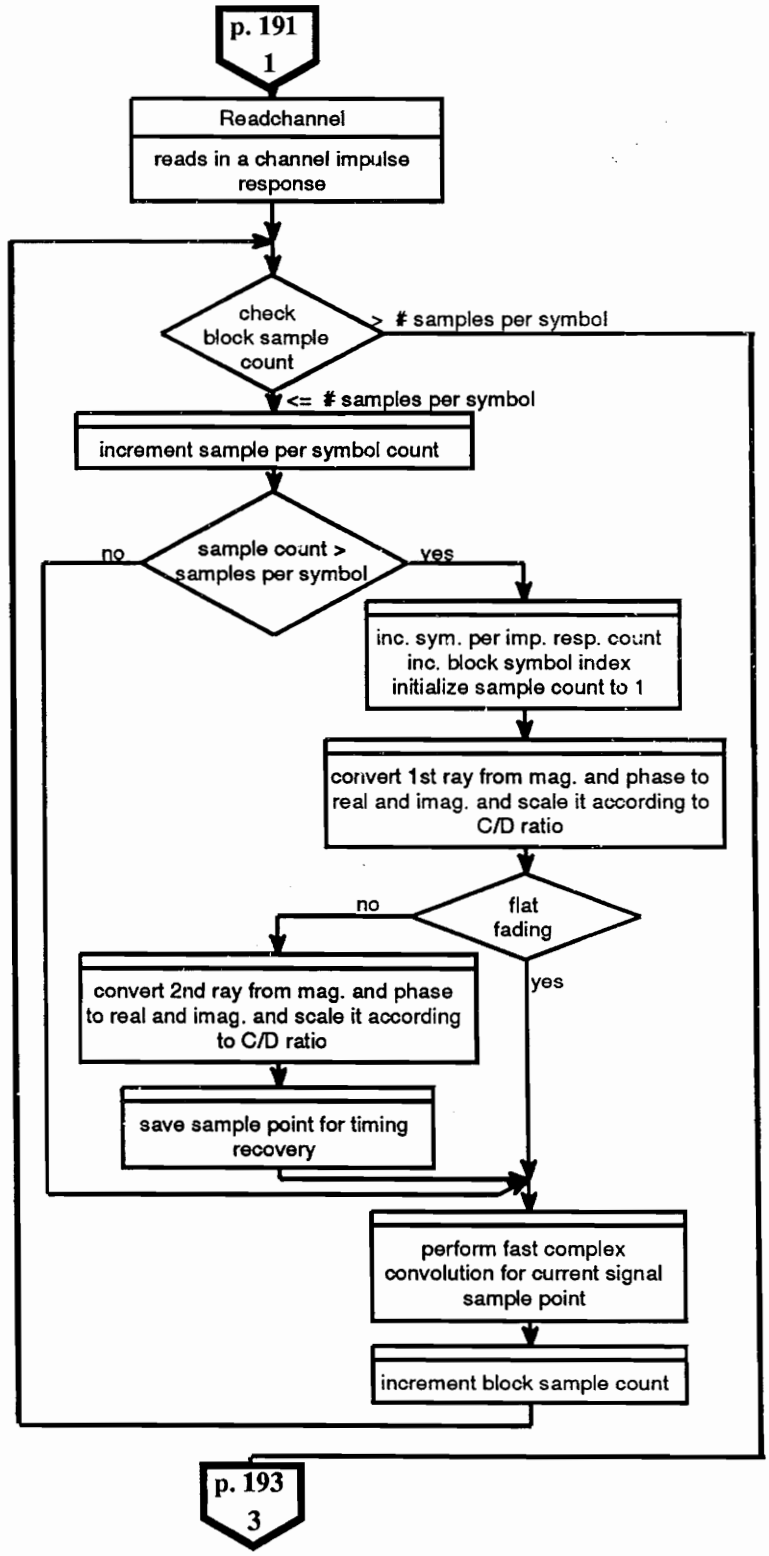


FIGURE A.33: Flow chart 3 for program tsimul (continued)

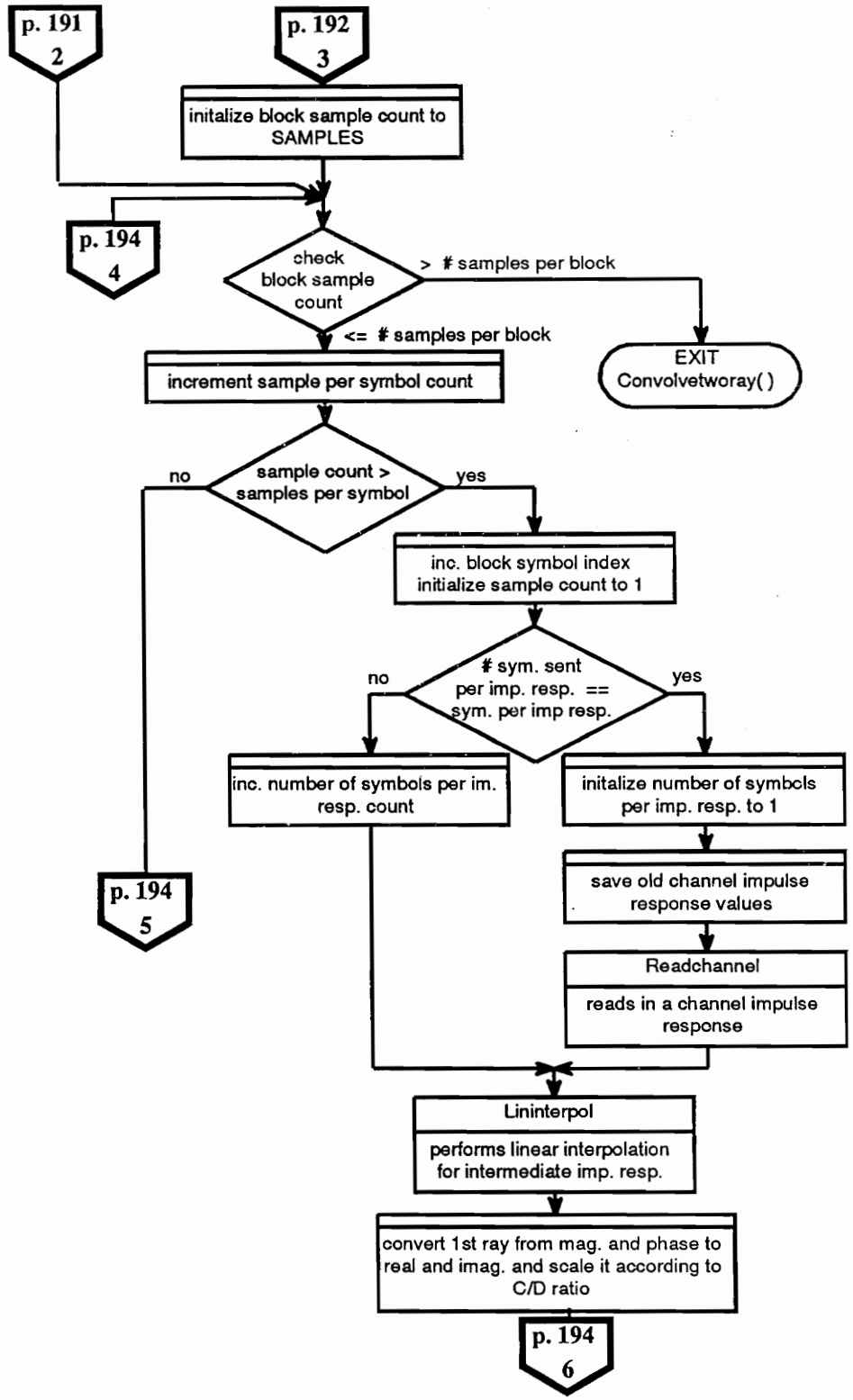


FIGURE A.34: Flow chart 3 for program tsimul (continued)

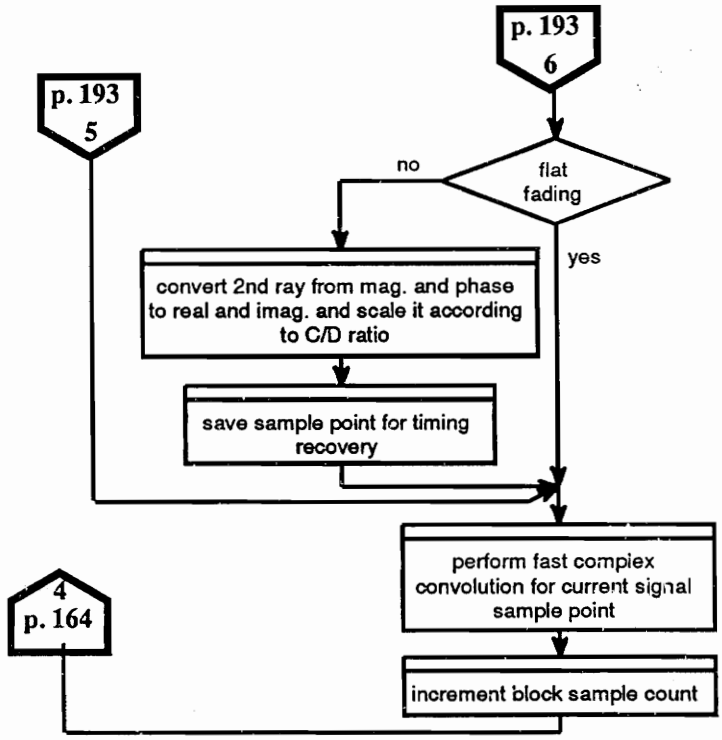


FIGURE A.35: Flow chart 3 for program tsimul (continued)

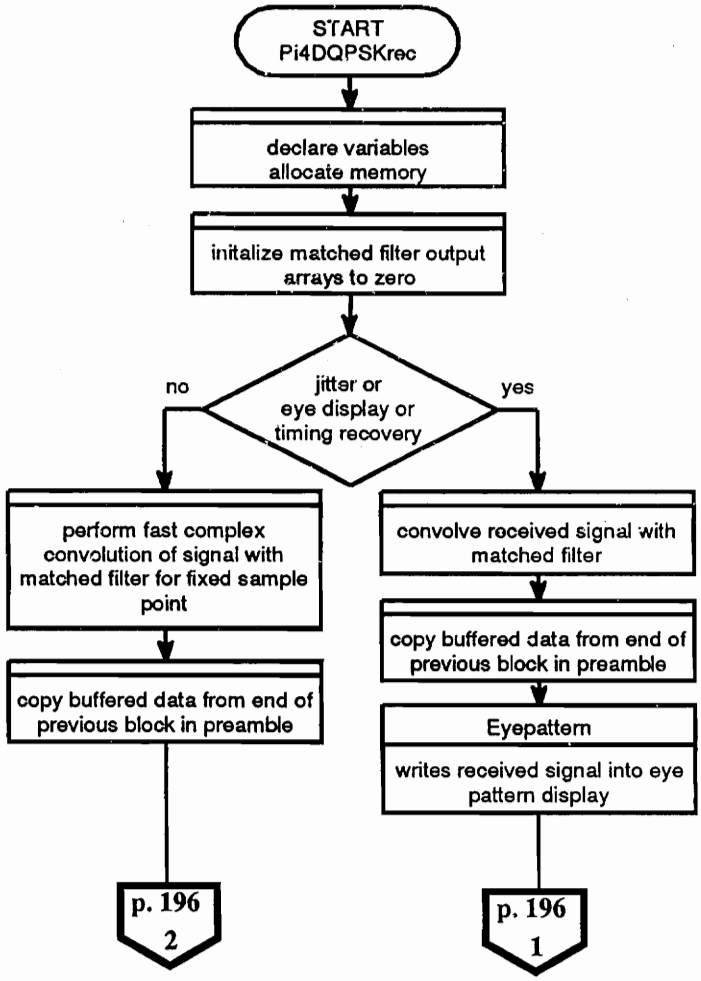


FIGURE A.36: Flow chart 4 for program tsimul

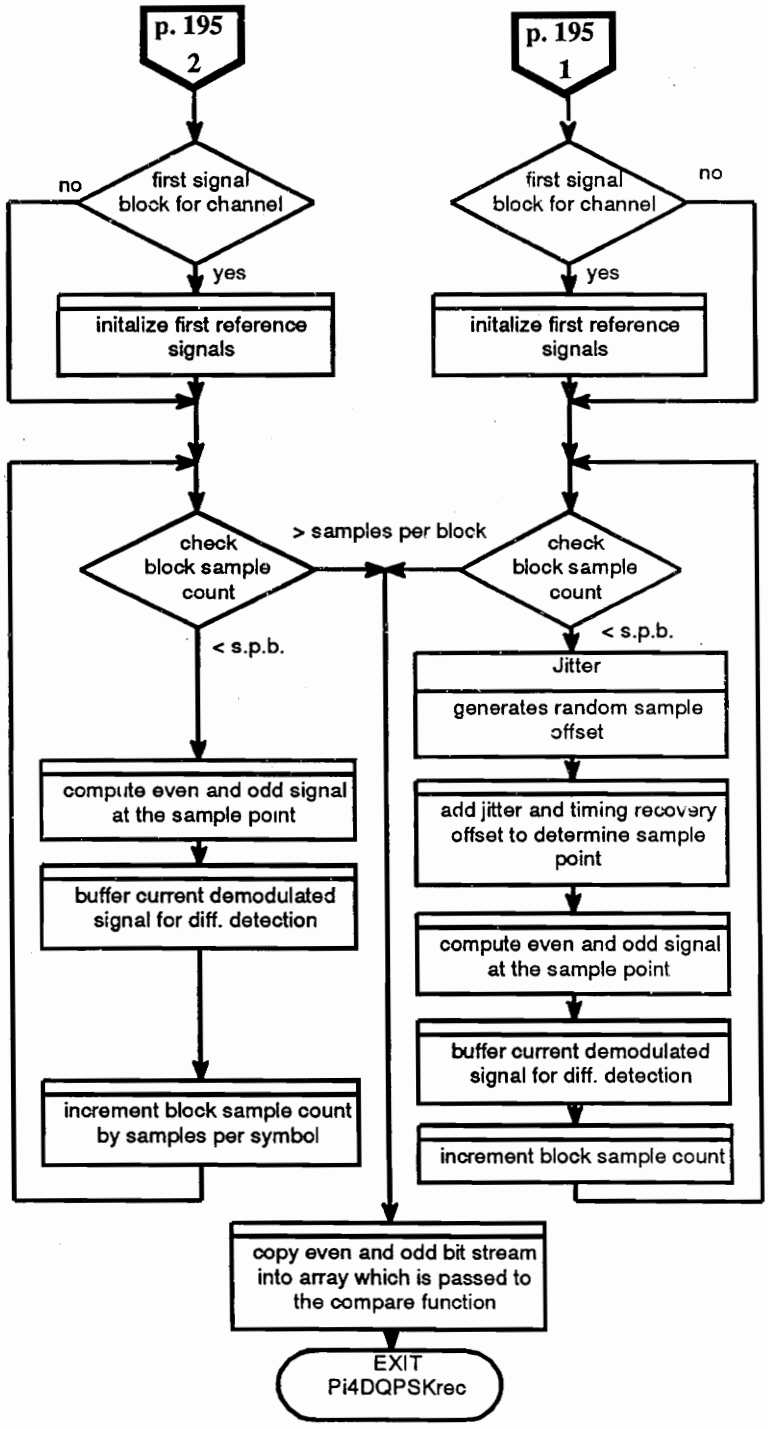


FIGURE A.37: Flow chart 4 for program tsimul (continued)

Vita

Berthold Thoma was born in Wenkheim, Germany on June 29, 1963. He received his Diplom Ingenieur (FH) degree from the Fachhochschule Heilbronn, Germany in 1990. Before entering the Fachhochschule Heilbronn in 1985, Mr. Thoma was on a 3 1/2 year apprenticeship with the A.W.d.H. Neckarzimmern, Germany to qualify as a trained radio engineer. During his studies in Germany, he received a Carl Duisberg scholarship to work for 6 months as a research assistant at Middlesex Polytechnic, London in the robotics laboratory. For his thesis at the Fachhochschule Heilbronn he joined S-TEAM Elektronik Untereisesheim where he developed a direct-sequence spread-spectrum based LAN transceiver. In 1990, he entered Virginia Tech as a Fulbright scholar where he joined the Mobile and Portable Radio Research Group. His graduate research focused on simulations of mobile radio communication systems to analyze bit error rate mechanisms in those systems.

Mr. Thoma is a member of the IEEE.