Copyright © 1991, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

WIDEBAND DIGITAL PORTABLE COMMUNICATIONS: A SYSTEM DESIGN

by

Samuel Sheng

Memorandum No. UCB/ERL M91/108

3 December 1991

WIDEBAND DIGITAL PORTABLE COMMUNICATIONS: A SYSTEM DESIGN

by

Samuel Sheng

Memorandum No. UCB/ERL M91/108

3 December 1991

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

TITLE PAGE

Table of Contents

List of Figures

Acknowledgments

"100... 100... 1... S.O.S..."

- *Rush, Grace Under Pressure*

No project like this ever succeeds in being finished without the assistance of many, many people, most of whom have answered many questions with good cheer and casual aplomb. Likewise, many have gently prodded with the reminder, "if you can't answer it, then it'll probably make a good research area." There seem to have been many of these areas, at times, a seemingly infinite number; I only hope I have answered at least some of them to a degree of satisfaction.

First, this is for my parents, who have always supported me with unending amounts of love and encouragement, even if the technical details of the project were sometimes obscure. Secondly, to my friends and colleagues in the EECS department at U.C. Berkeley, many of whom have known me since I was a freshman- yes, this part of the project is finally finished. Also, the Fannie and John Hertz Foundation must be acknowledged, for their generous support over the past two years, and of which I have the honor of saying that I am a fellow.

To Jenny- for caring, for always understanding (well, at least most of the time), and for always being there.

And last, but certainly not least- to Professor Robert Brodersen, who has been patiently waiting an unconscionably long time for this thesis to materialize on his desk, and whose support and guidance have been invaluable, both in technical matters and otherwise.

Samuel Sheng
20 October, 1991

CHAPTER 1

INTRODUCTION

The recent efforts in the development of personal communication systems have prompted the need for high-speed, wireless digital transmission systems operating in the gigahertz band or higher. Defining the concept of personal communications is the ability for individual users to have instantaneous and portable access to fixed information networks via a lightweight graphics unit, capable of transferring data to other users and accessing fixed computing resources without any constraints on where or when such access takes place. This portable unit will thus serve as a multimedia terminal, supporting full-motion digital video and high-quality audio, and combining the functionality of today's analog mobile telephones, radio pagers, and laptop personal computers.

Since the portability requirement places severe constraints on the physical weight of the terminal, the amount of battery power available is quite limited. Thus, the terminal can only carry the bare minimum of computing resources to support its functionality; user computation will be performed by large, non-portable computing facilities, with the high-speed wireless link serving as the terminal's sole means of accessing the fixed computation serv-

ers and data networks. Direct point-to-point wireless communication is not allowed; the link only provides the final interface into the wired data network, much like a conventional telephone handset serves as the link into the telephony system (figure1.1). In essence, whereas the capability of moving massive amounts of digital data within networks already exists, the problem of easily getting data in and out of those networks is now addressed.

It is evident that the performance requirements on such a transmission system is quite severe. If the system is to support digital video, a minimum data rate of 2 MBps is required, even utilizing the best compression schemes available today [1]. Assuming a spectral efficiency of 1 Bps/Hz, a minimum of 2 MHz of bandwidth needs to be allocated to each user. Due to these high bandwidth requirements, carrier frequencies must be placed in or above the low microwave bands, owing to spectrum congestion at frequencies below 1.5 GHz. Likewise, since one major objective of personal communications lies in granting each person individual access, a large number of users need to be accommodated by the system simultaneously, densely packed into a small physical area; continuous network access by 1 user every 4 to 10 square meters is typical for an office environment.

Thus, the following constraints are imposed upon any transmission scheme that is to be used for such a personal communication system, which are not satisfied by any known transceiver system available at present:

- The analog circuitry must perform reliably at variable carrier frequencies above 1.5 GHz, while supporting a sustained user data rate in excess of 2 Mbps.
- The system must support simultaneous access by a large number of users, within a small area.
- Bit-error rates must be sufficiently small, *e.g.* below 10^{-6}.
- The transceiver cannot consume excessive amounts of power, as the portable terminal must be powered by batteries, nor can it be excessively large.

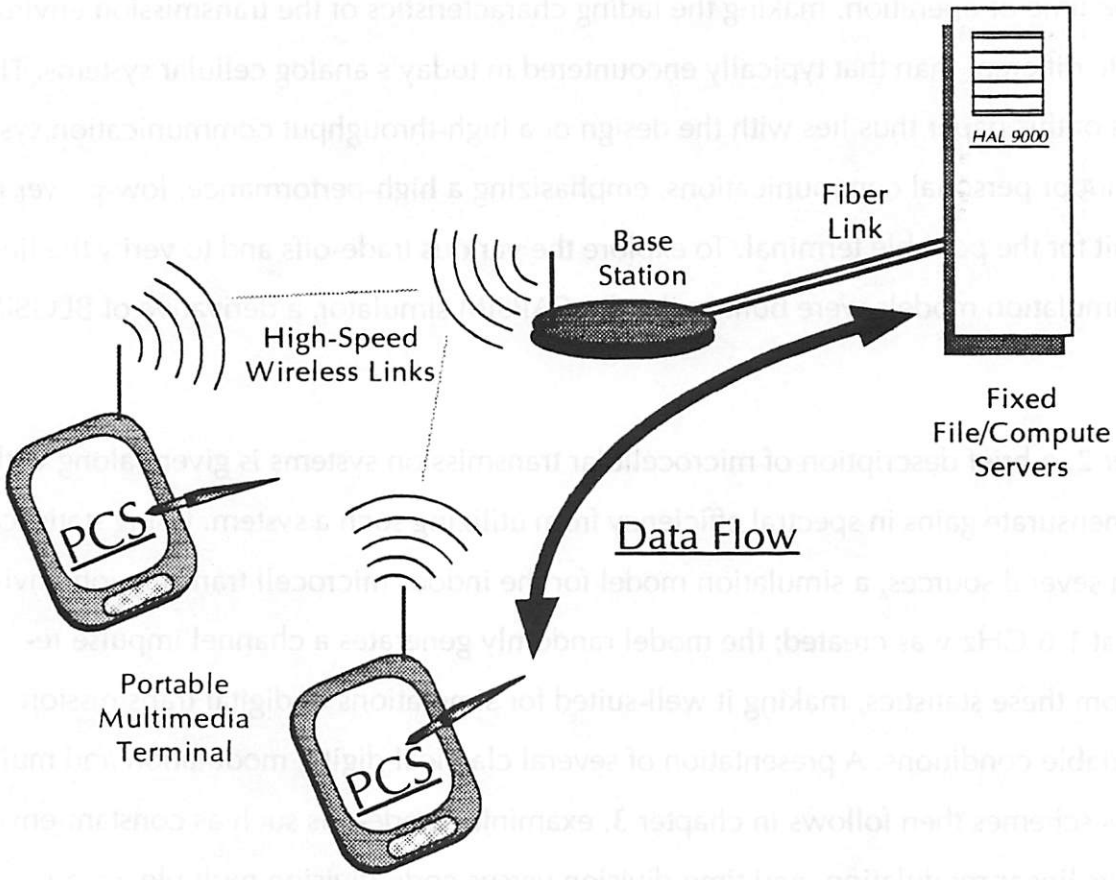

Figure 1.1: PCS System Overview

Goals and Objectives

In this paper, design strategies and techniques for implementing such a system will be explored, and a complete system specification proposed. Since such a system will first be utilized as a step beyond conventional wireless LAN's, an indoor microcellular environment will be of primary concern, although many of the techniques are applicable to both the outdoor and indoor environments. The issues discussed range from low-level analog design, where a receiver structure using passband sampling is presented, up to the multiple access schemes utilized and its impact on the transceiver system. Since only indoor operation is considered, the receiver is assumed to be moving slowly or completely stationary

during the time of operation, making the fading characteristics of the transmission environment quite different than that typically encountered in today's analog cellular systems. The emphasis of this paper thus lies with the design of a high-throughput communication system for indoor personal communications, emphasizing a high-performance, low-power receiver unit for the portable terminal. To explore the various trade-offs and to verify the final design, simulation models were built under the CAPSIM simulator, a derivative of BLOSIM [2, 3].

In chapter 2, a brief description of microcellular transmission systems is given, along with the commensurate gains in spectral efficiency from utilizing such a system. Using statistical data from several sources, a simulation model for the indoor microcell transmission environment at 1.6 GHz was created; the model randomly generates a channel impulse response from these statistics, making it well-suited for simulations of digital transmission under variable conditions. A presentation of several classical digital modulation and multiple access schemes then follows in chapter 3, examining trade-offs such as constant-envelope versus linear modulation, and time-division versus code-division multiple access. Given the transmission environment in question, it turns out that linear modulation utilizing spread-spectrum multiple access possesses many desirable properties. Using this as our target system design specification, chapter 4 addresses the inherent hardware design issues, given the constraints of current technology and emphasizing the simplification and integration of the analog circuitry as much as possible. Lastly, the simulation of the complete system is analyzed, verifying that the modulation system is sufficiently robust to meet the above performance requirements. Conclusions and a description of future work are then summarized at the end of chapter 5.

CHAPTER 2

INDOOR PICOCELLULAR COMMUNICATIONS

The advantages in improved spectral efficiency afforded by cellular systems have long been known; having been employed extensively in present-day analog mobile radiotelephony, the large-scale cells utilized only exploit these advantages to a limited extent. By scaling down cell sizes, tremendous increases in spectral efficiency can be achieved. In this chapter, an analysis of these gains is presented, along with a description of the cellular concept as realized in an indoor environment. A statistical characterization of the indoor picocellular transmission environment and a corresponding model are then described.

Cellular Systems

As shown in figure 2.1, a simple cellular scheme consists of dividing the entire service area for the personal communication system into "cells" of radius R, with a single base station serving all mobile users within that cell; each cell utilizes its own distinct set of frequencies, different from its surrounding neighbors. As users move from cell to cell, their transac-

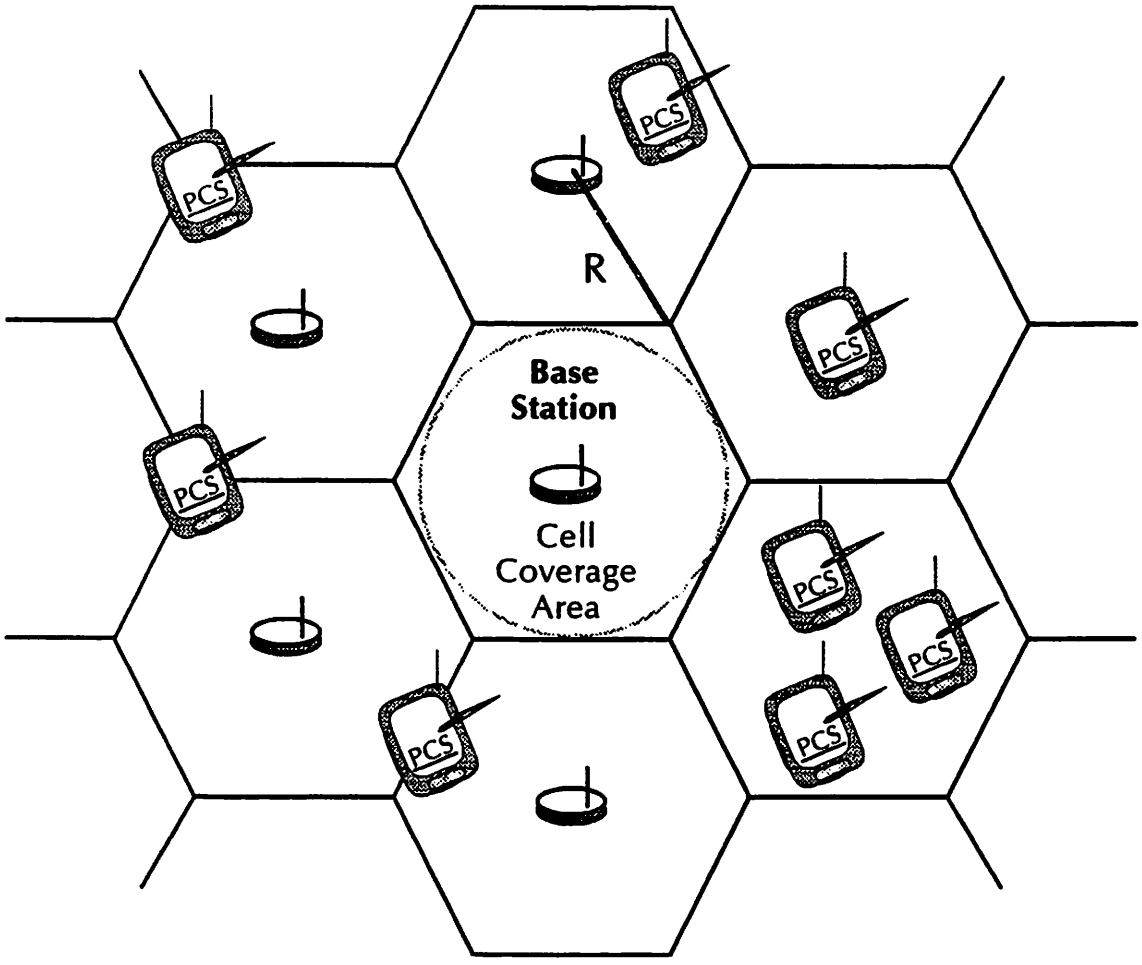

Figure 2.1: Cellular Communications System

tions with the network are "handed off" from base station to base station, reconfiguring the network dynamically as the need arises. Since the base stations themselves are responsible for determining where and when handoffs occur by monitoring the received power from the portable terminals, such reconfiguration is thus accomplished locally. Clearly, the complexity involved with cellular network control is much greater than that required for a classical "umbrella" scheme, with one base antenna for the entire service area, and much of the "intelligence" of the network must be deployed within the base station.

The key advantage to cellular systems is that they allow the network to achieve spatial multiple access of the users. If two cells are separated by sufficient distance, each can use the

same frequency bands at the same time without resulting in disastrous cochannel interference. Thus, frequency reuse becomes possible, as opposed to an umbrella scheme where every user must be assigned a different frequency slot. Figure 2.2 shows several classical reuse patterns [4]; such patterns are typically characterized by a frequency reuse factor K, which represents the number of distinct frequency sets that need to be used to cover the entire service area.

Thus, instead of one user per frequency band, the network can now support N users per band, where N is the number of cells utilizing that band. From the point of view of spectrum usage, each user effectively consumes only B/N Hz of bandwidth, where B is the physical bandwidth needed to support transmission. Hence, cellular systems are said to be spectrally efficient. Clearly, minimizing the distance D between cells utilizing the same frequency yields the greatest frequency reuse, since the number of cells that can use the same frequency band in the service area is maximized, and hence the greatest gain in spectral efficiency is achieved. This frequency reuse distance is geometrically related to K and R by $D=R\sqrt{3k}$. In terms of system capacity, supposing that the total service area serving the N users is given by $\pi\rho^2$, then the number of cells is given by $(\rho/R)^2$ and hence the total bandwidth for the system has been reduced by a factor equal to $(R/\rho)^2K$ relative to an equivalent non-cellular system. As shown in figure 2.3, the normalized bandwidth required by a cellular system is plotted as a function of R and K; since it is quadratically dependent on R, it is of greatest benefit that the cell radius be reduced. Clearly, when $R=\rho$ and $K=1$, the original "umbrella" scheme is yielded.

Ostensibly, the ultimate limits on minimizing R (and hence D) lie in how much cochannel interference the system can tolerate, and the required complexity in network control. An important fact is that the level of cochannel interference is independent of the scaling of R, since the transmit power in each cell scales with R and hence the relative interference stays constant. Thus, cochannel interference is only a function of K. Conversely, complexity in network control is only a function of R, since more handoffs will necessarily occur as R de-

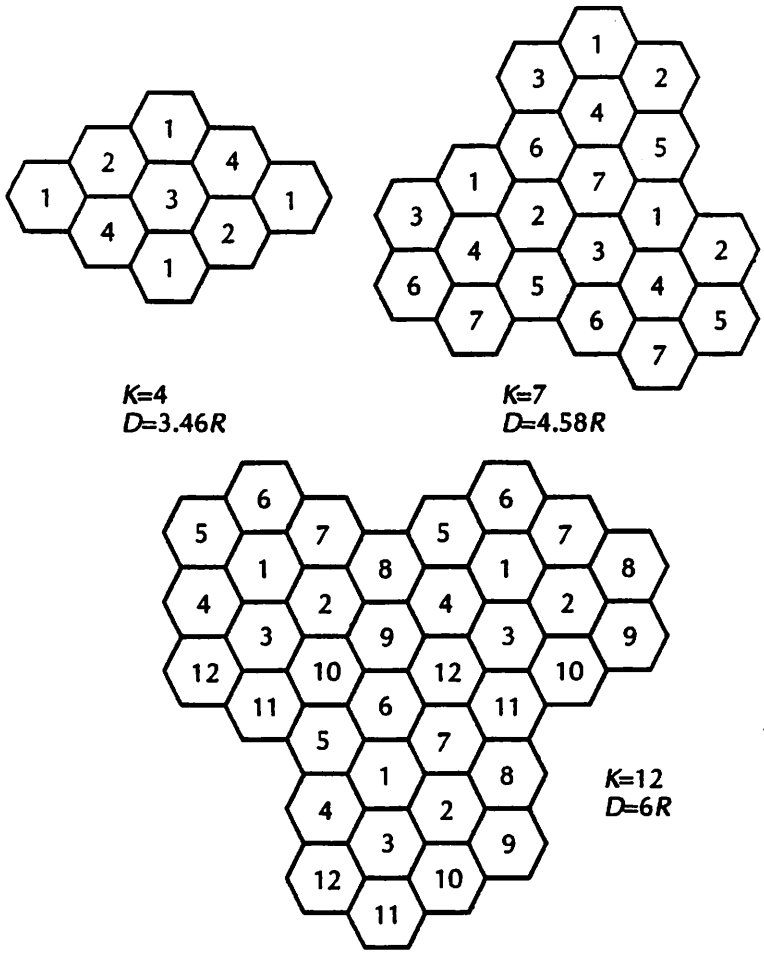

Figure 2.2: Frequency Reuse Patterns

creases. Both K and R are highly dependent on the transmission environment; however, it is clear that these massive gains in spectral efficiency that will be needed to support the high-speed, high-bandwidth requirements that a personal communications system stipulates. A detailed analysis of the optimum choice of K and R for a general cellular system lies beyond the scope of this thesis; however, for an indoor system, the choice of R and K is already dictated to a great extent by the environment, as shown below.

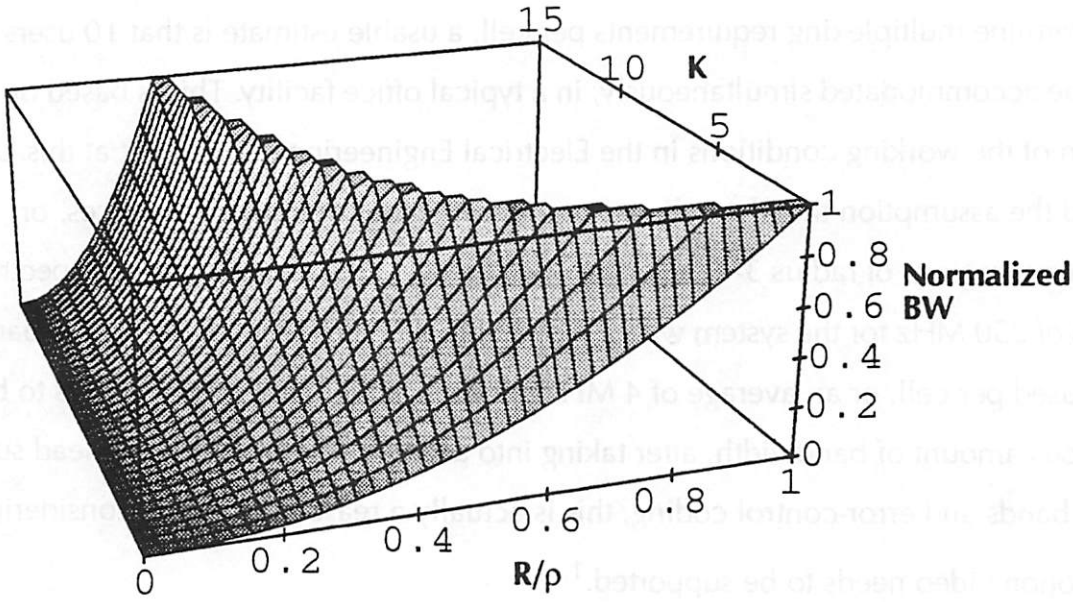

Figure 2.3: Normalized BW of Cellular System

Normalized with respect to the non-cellular case, assuming a constant number of users and data rate.

The Indoor Picocellular Environment

With an indoor environment, it is actually no longer feasible to have only a single network transceiver station serving all of the terminals in the building. Due to the 5 to 15 dB attenuation through walls [5], the total microwave output power from all of the transmitters would have to be inordinately (and dangerously) high. However, this attenuation can be taken advantage of by a cellular network. Each room then naturally becomes its own cell; likewise, the cellular scheme now moves into three dimensions, since the floors also provide RF isolation. Since the cells are now extremely small, on the order of a two to five meters, this cellular strategy is commonly referred to as microcellular or picocellular networking. Hence, R is usually dictated by the size of the room, and K can be as low as 3 to 4, depending on how much attenuation is provided by the walls. If K is increased to 6 or 7,

the assumption that cochannel interference is negligible becomes reasonable for most indoor office environments.

To determine multiplexing requirements per cell, a usable estimate is that 10 users per cell must be accommodated simultaneously, in a typical office facility. This is based on a examination of the working conditions in the Electrical Engineering department at this University, and the assumption that the cells are comprised either of individual offices, or hexagonal regions of radius 3-4 meters in open areas. Thus, assuming a total spectrum allocation of 250 MHz for the system with a K of 6, this allocates around 40 MHz of bandwidth to be used per cell, or an average of 4 MHz per user. Although 4 MHz appears to be a tremendous amount of bandwidth, after taking into account transmission overhead such as guard bands and error-control coding, this is actually a reasonable value considering that full-motion video needs to be supported.[1]

Likewise, mention needs to be made about the antennas to be used in such a personal communications system. Due to portability requirements, they must be necessarily small, and also must be omnidirectional, since a mobile transceiver would require constant adjustment if used with an directive antenna array like a Yagi. For a stripline omnidirectional dipole, antenna gains from 0-6 dBi can be reasonably expected.

Lastly, the allowed transmission power needs to be considered. Assuming that the condition of no cochannel interference is required, and a K of 7 with cells of 4 meter radius, the frequency reuse distance is thus 18.3 meters. Given that the 1 meter reference path loss is approximately 30 dB, a conservative power loss through 5 walls of 20 dB, and using the median path loss coefficient of 2.6, the power in the nearest cochannel cell is given by the link-power equation

1. If full-motion video were supported using existing low-resolution NTSC analog transmission, it would already require 6 MHz per band per independent user.

$$P_{cochannel} = P_{transmit} - 30\ dB - 20\ dB - 2.6 \cdot 10 \log (18.3/1)$$

Assuming a transmitter-receiver separation of 5 meters, the power of the received signal is equal to $P_{transmit}$ - 30dB - 2.6·10 log (4/1), or 34.6 dB above the cochannel interference power. Evidently, the isolation provided by the walls is significant in allowing such heavy frequency reuse; without the wall isolation, the cochannel interference power would be only 16 dB below the signal level. The design condition chosen is intended to make the cochannel interference power equal to the thermal noise power at the receiver, which as described below is limited by the performance of the input amplifier. In this way, neither the cochannel interference nor the receiver noise is allowed to dominate, yielding a maximum SNR utilizing the minimum transmission power. For a high-performance, low-noise amplifier, a noise figure as low as 5 dB can be realized, relative to a 50Ω load, resulting in a noise power of -87.5 dBm over a bandwidth of 1 MHz. Using the link power equation above, this implies that $P_{transmit}$ is optimal at -4.1 dBm, or 0.33 mW. Even accounting for a ±10 dB variation owing to the crudity of the wall-attenuation assumption, this is significantly smaller than the +30 dBm used in conventional analog cellular, and yields a significant reduction in power consumption in the transceivers used in the system.

Statistical Characterization and Multipath Propagation

Unfortunately, the indoor environment also presents several additional transmission difficulties, the dominant one being the numerous reflections of the radio signal off of walls, furniture, and even people. A large number of paths exist between transmitter and receiver; the received signal is thus corrupted by severe multipath distortion and inter-symbol interference, since these reflection can still be significant several symbols later. Also, in a picocellular net, it is likely that there is no direct line-of-sight propagation path due to shadowing, so we become critically dependent on the reflected waves. All of these effects,

of course, vary slowly with time; even with the remote unit fixed in place during operation, the motion of people can cause significant variations in the environment. The critical statistical parameters to be determined are: the number of paths, the excess time delay for a single path, the path loss, and the time variation of the received signal, assuming a cell size of 2 to 10 meters. Using measured data collated from various sources [5, 6, 8], such a characterization of the indoor propagation environment at 1.6 GHz is determined.

First, the path loss in a multipath situation must be considered. It has been found experimentally [8] that the total received power at a particular distance d of a multipath profile can be modeled as a log-normal (normal in dB) distribution about a local mean path-loss law given by d^n, valid for d approximately larger than λ, where λ is the carrier wavelength at 1.6 GHz = c/f = 0.1875m. Values of n ranging from 1.5 to 6 have been reported, although typical values range from 1.5 to 3. Figure 2.4 shows a measured path loss characteristic [8], with the best-fit curve varying as $d^{-2.6}$. It has also been found that for a single path component the received power has a log-normal distribution about an exponential path-loss law $d^{n(\tau)}$, where τ is the excess delay time defined as $t_{arr} - t_0$ and t_0 is the delay of the line-of-sight path, with an obstructed topology exhibiting greater attenuation than a line-of-sight one as a result of the extra path distances incurred by shadowing. Generally, the standard deviation of the log-normal distribution is insensitive to τ, and is approximately 4 dB for line-of-sight topologies, and 5 dB for obstructed ones.

The problem of the statistics of the number of paths and their arrival times at the receiver also needs to be considered. Clearly, receiver sensitivity here is critical, since more paths that can be resolved by a receiver with higher sensitivity. For low-sensitivity receivers the number of path has been measured to be approximately Poisson, with a mean of 4.4 paths and standard deviation of 2.1, for a receiver threshold of 30 dB below a10λ received power reference. As receiver sensitivity increases, a Poisson distribution becomes a poorer model for the number of paths, and a normal distribution with a mean of 22.4 paths and standard

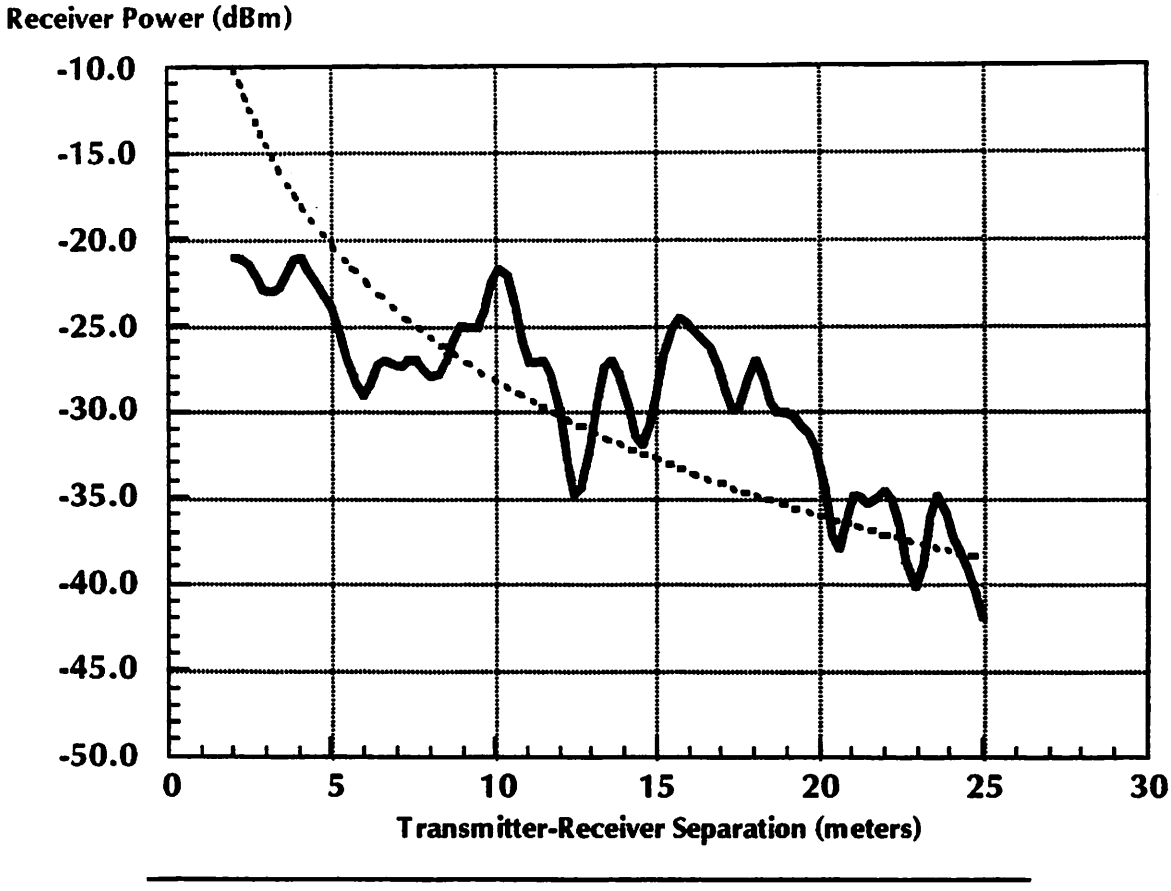

Figure 2.4: Measured Path Loss Characteristic
(from [8])

deviation of 8.6 paths provides a good statistical model for a 48 dB threshold. The other path-dependent factor is the distribution of arrival times; having determined the behaviour of the number of paths, the probability of a path having an excess arrival time τ needs to be characterized. From several measurements [6, 9], the excess arrival times follow an exponential distribution, with a mean time of 30 nsec for transmitter-receiver separations below 10 meters.

Unfortunately, the difficulties presented by the multipath environment are further compounded by the fact that the environment is also time-varying, and hence results in fading, where the environment varies such that destructive interference occurs at the receiver and causes much more attenuation than predicted by path loss alone. Since it is assumed that

the remote device is relatively stationary, the typical Rayleigh-distributed fading patterns found in analog mobile environments are not encountered. Instead, the primary cause of time-variation is the movement of people in the propagation path. For only a few people, the distribution has been found to be Rician more than Rayleigh; however, as the number of persons increases, the distribution becomes increasingly Rayleigh. From the measurements taken, a typical fluctuation results in a signal deviation 10% of the time in excess of 4 dB away from the median, and a deviation in excess of 8 dB 1% of the time [8].

Lastly, noise effects within the channel must be considered. From measurements taken of the 1-2 GHz frequency band, the additive white Gaussian noise present on the channel is essentially at the thermal noise floor (-110 dBm at 300°K over 1 MHz); hence, as far as Gaussian noise is concerned, the noise contribution is dominated by the noise at the input of the receiver. However, the low-microwave band also exhibits significant impulsive noise, resulting from RF interference from such sources as computer systems, power microwave cavities in cooking ovens, and common relay switches. It is extremely difficult to characterize these sources, since the presence and characteristics of such generators varies so widely, even within the same building. From some recent measurements, it has been shown that the average noise factor above the thermal noise floor for impulsive sources can be as high as 50dB [7].

Channel Model

For multipath transmission, an early time-varying, wideband model proposed by Turin is given as

$$\sum_{k=0}^{n} a_k e^{jq_k} s(t - t_k) + n(t)$$

where s(t) is the complex-valued, lowpass representation of the transmitted pulse, n(t) is the

noise component, q_k is the carrier phase shift, and a_k are the amplitudes of each arrival at time t_k in the multipath profile. Since along each propagation path, the physical media is essentially "ideal" (i.e., represented by a simple attenuation constant), the impulse response of such a channel can thus be characterized by an expression of the form

$$h(t) = \sum_{k=0}^{n} a_k \delta(t - t_k) e^{jq_k}$$

By using the statistical data described above, a model of this impulse response suitable for simulation will be developed.

The path strength coefficients a_k are set to the local mean value given by d^n, where n=2.6, multiplied by the variation which is normally distributed in dB. Also, since of primary interest is a simulation model suitable for verifying the functionality and performance of a transceiver for PCS, the time-variation of the channel is sufficiently slow that we can assume it is constant over the simulation time. Thus, for our model the path attenuations are fixed at the beginning of the simulation and remains invariant. Conversely, if long-term error performance simulations are required, such as Monte-Carlo estimation of the bit-error rate, the model can be easily adapted for such purposes, by simply incorporating time variability into the model. The only consideration that needs to be made is that over time, the amplitude and phase of each multipath component must be continuously related to the original randomly generated profile – the profile cannot be "randomly" updated. Likewise, the number of paths and individual path delays are treated similarly. At the start of the simulation, the number of paths is randomly determined, which is normally distributed as described in the previous section. For each path, an exponentially distributed random value is assigned, corresponding to the path delay.

A subroutine capable of generating such a profile is listed in appendix A; incorporated as a CAPSIM routine, any simulation block (or "star", in CAPSIM parlance) can simply call GenerateMPath() to create a multipath profile corresponding to these statistics. Two CAPSIM

stars which do so are also listed; one of these stars, MPathImpulse.s, models the channel as a real-valued FIR filter, whose impulse response is simply given by $\{a_k\}$; k=0..n. This star is useful for testing analog modulator and demodulator structures, since it gives the true passband response of the channel. The other star, multipath.s, gives an equivalent baseband representation of the channel in terms of real and complex components. This star is useful for examining the baseband digital modulation and demodulation strategies; essentially, it encapsulates the analog passband modulation, channel effects, and passband demodulation into a single model. The key difference lies in the number of data points that the system must process in order to simulate a single transmitted symbol; in the case of mpathImpulse.s, the passband modulation is modeled, requiring an extremely fine timestep to capture the gigahertz band carrier, whereas with multipath.s the timestep needs to only capture the baseband phenomena, which is bandlimited to several megahertz. Clearly, both will be useful in exploring the modulator and demodulator structures for our PCS transceiver.

The impulse response of such a randomly generated channel is shown in figure 2.5, which is quite similar to impulse responses measured in the field [9]. In figure 2.6 is the magnitude frequency response of the channel, $H(j\omega)$, plotted in a 200 MHz bandwidth about 1.5 GHz. Evidently, there are several severe fading nulls, some as deep as 25 dB below the peak. If such a null happens to occur within the transmission bandwidth, signal degradation and possibly complete link breakdown will result, since the signal-to-noise ratio has been compromised by the null.

As an important application of this model, the effects of the multipath on an actual transmitted signal will be simulated, by generating an eye diagram of the received signal. Using a simple 4-PSK QAM (see chapter 3) modulation scheme with white input data, the effects of the multipath distortion is examined for a data throughput of 2 Mbps at a carrier of 1.5 GHz. As described above, the multipath star.s is used heavily in this type of simulation. A 50% excess-bandwidth raised-cosine filter response is assumed for the combined transmit

and receive bandlimiting filters, with the filter partitioned equally between the transmitter and the receiver, *i.e*, the transmit filter has a frequency response equal to $\sqrt{H(j\omega)}$, where $H(j\omega)$ is the transfer function of the raised-cosine lowpass filter, and similarly for the receive filter. Also, the amplitude of the transmitted signal is normalized to one, since only the relative loss is of real interest.

In figure 2.7, the ideal (no channel) eye diagram of the in-phase received signal is shown after a transmission of 100 symbols; as expected, the "eyes" are fully open, and the signal value there is precisely contained in the set {-1,0,1}, indicating zero distortion and perfect recoverability. The waveform in between the sampling points is the response of the raised - cosine filter; this also verifies that the filter responses were designed correctly, as the Nyquist zero-forcing criterion is clearly achieved. In figure 2.8, the channel is now inserted

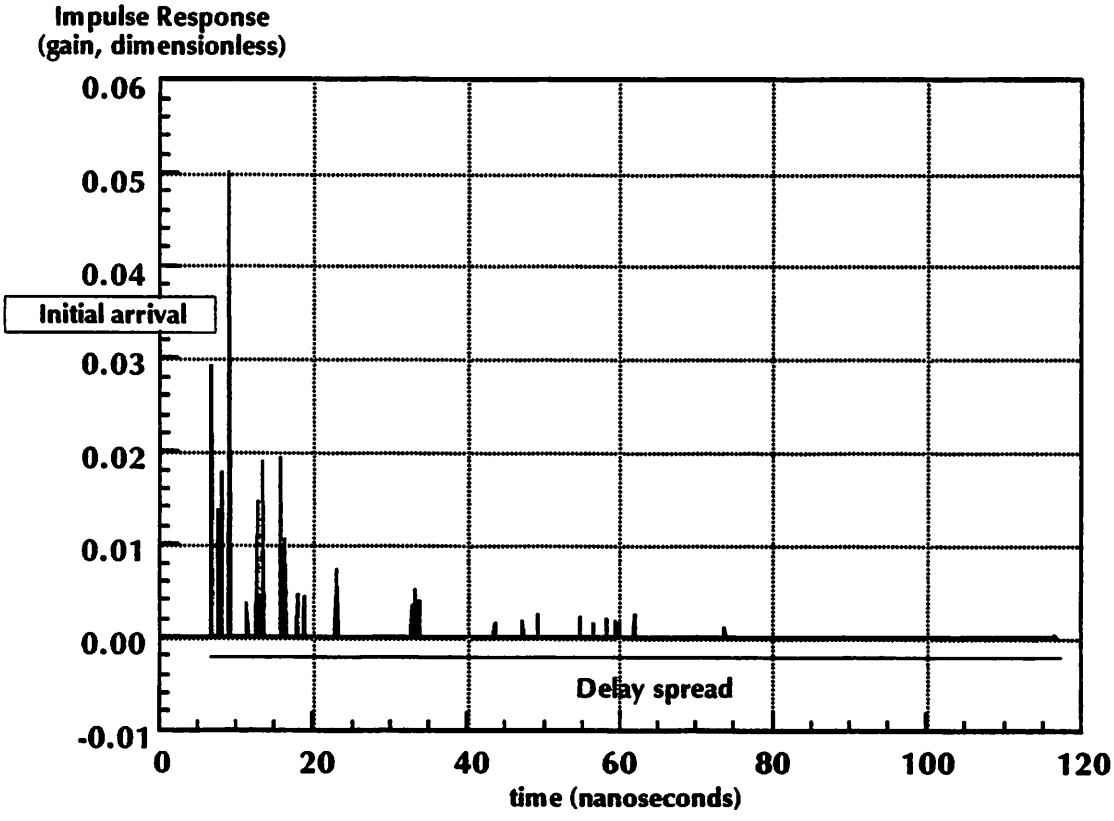

Figure 2.5: Simulated Channel Impulse Response

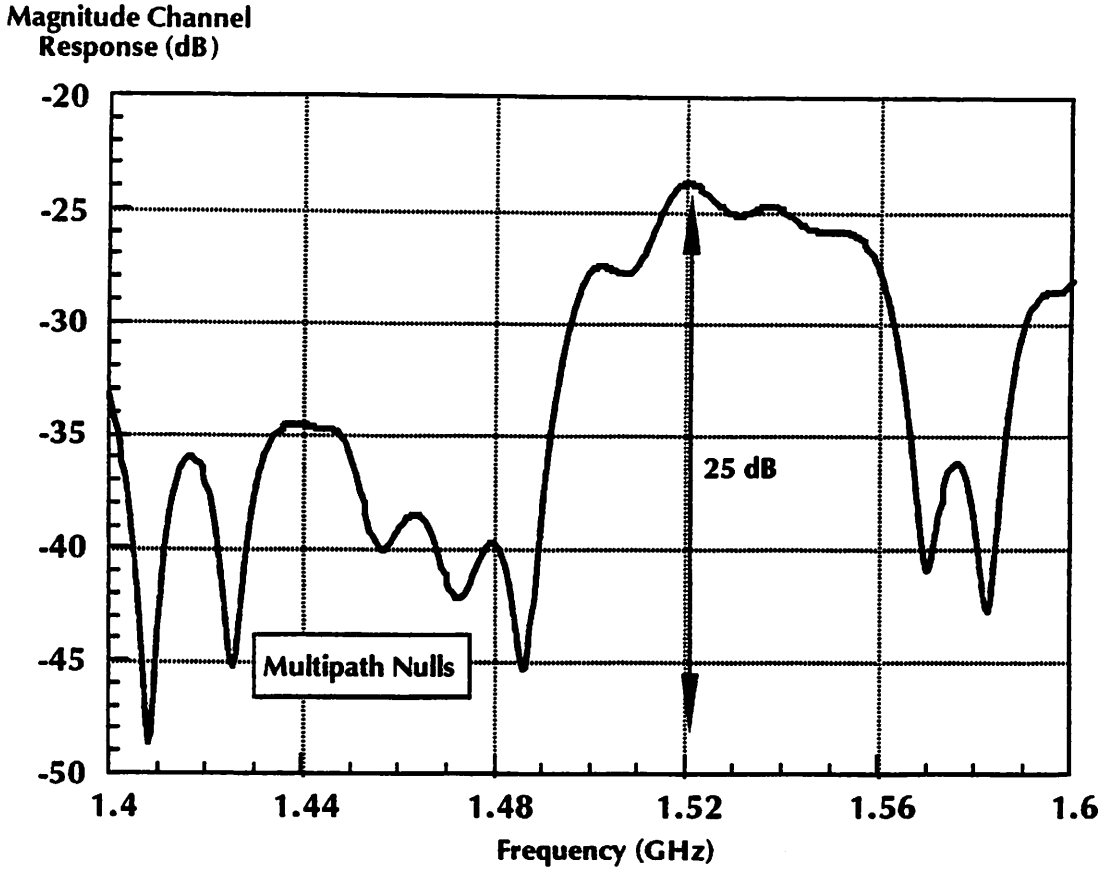

Figure 2.6: Simulated Channel Frequency Response

between the transmitter and receiver in the system, and the resulting eye diagram is shown. Demonstrating the key difficulty in transmitting within such an environment, the eye openings are now relatively closed due to the multipath distortion. The "zero" level has clearly split into two; this is to be expected, since when the in-phase symbol is zero, the quadrature signal is one. Due to the random phase shift caused by the channel, the in-phase signal is corrupted by crossover distortion from the quadrature phase.

Lastly, a simulation was done including the effects of thermal noise at the input of the receiver, with an SNR of 41 dB, and the resulting eye diagram plotted in figure 2.8. Comparing this against figure 2.7, it is clear that the primary interference is caused by multipath distortion for such low noise levels.

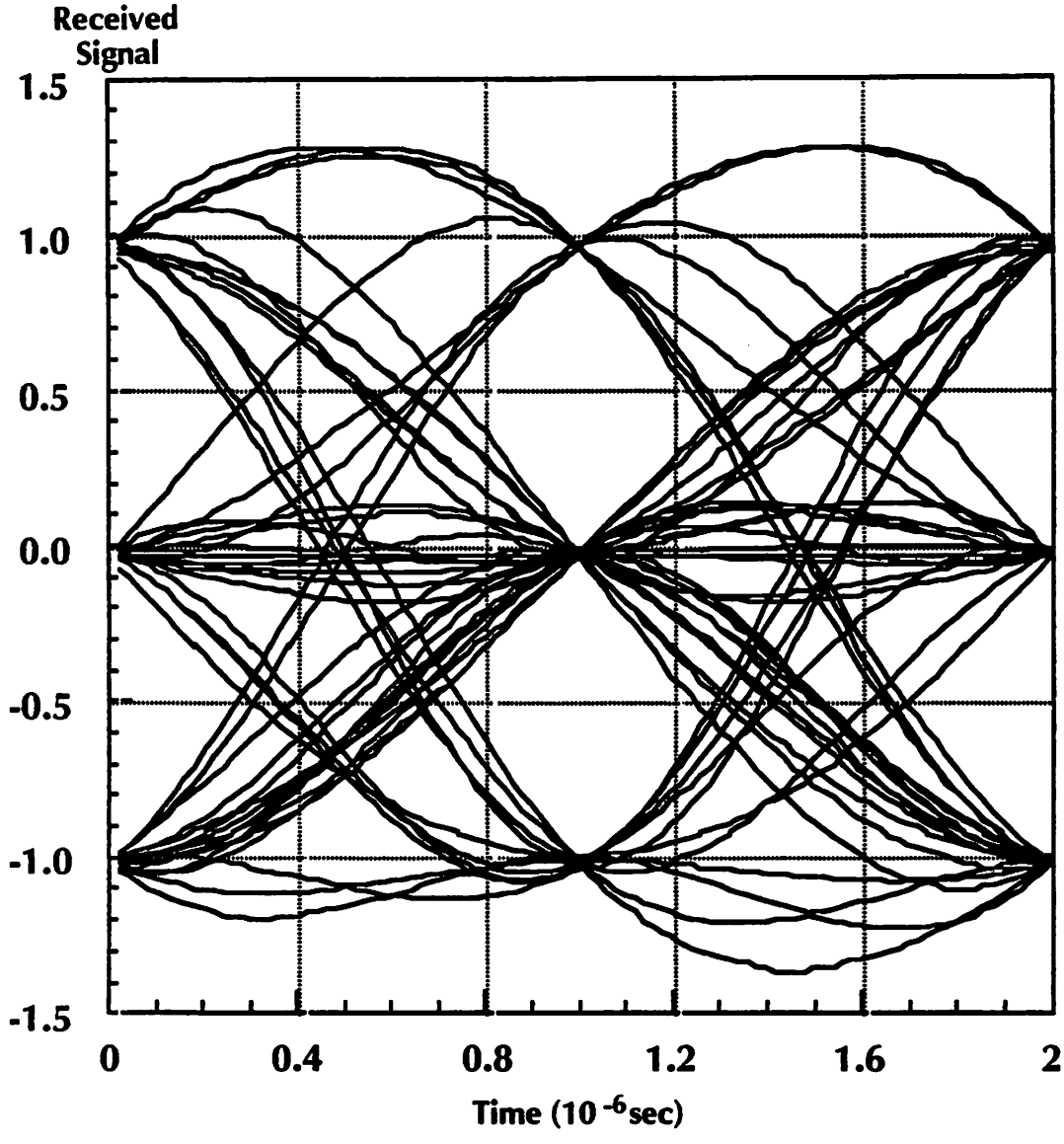

Figure 2.7: Ideal Eye Diagram for QPSK, no ISI

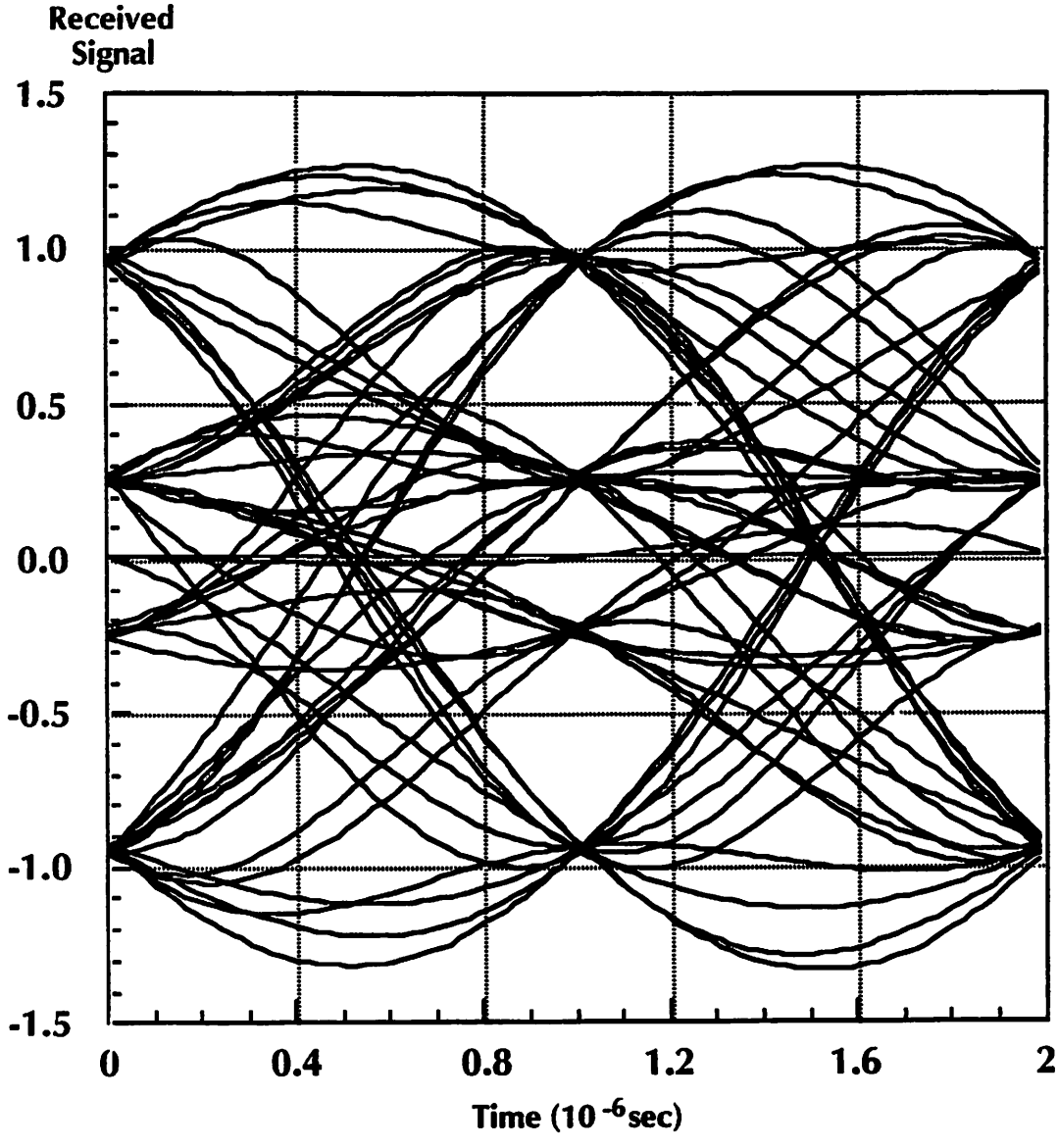

Figure 2.8: Eye Diagram for QPSK, channel ISI included

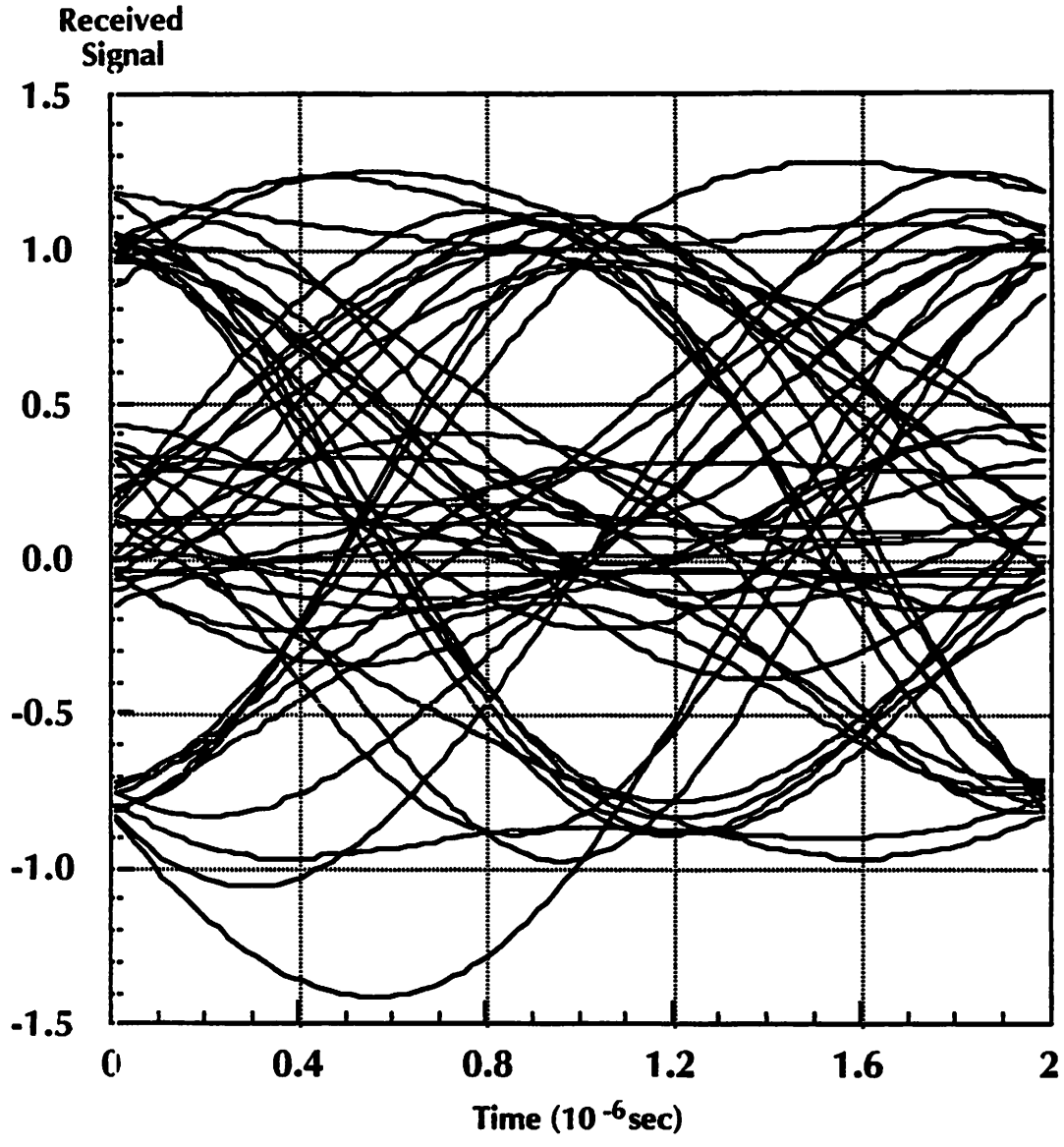

Figure 2.9: Eye Diagram for QPSK, ISI and AWGN included

CHAPTER 3

DIGITAL MODULATION AND MULTIPLE ACCESS

Before any hardware design can be approached, a clear understanding of the desired functionality is required, in the form of the signal modulation that will be employed. Choices such as quadrature amplitude modulation (QAM) versus continuous-phase modulation (CPM) and time-division multiple access (TDMA) versus code-division multiple access (CDMA) need to be made. In this chapter, these system-level design issues will be addressed, and various solutions compared.

When considering the design of the baseband modulation and multiple access schemes, it is important to note that this discussion is essentially restricted to the digital domain, and hence complex filtering and signal generation are not a problem. For example, simple frequency-shift keying, using different frequency tones to encode the data, traditionally utilized a low-frequency analog voltage-controlled oscillator to generate the tones, and then modulated this signal up to the passband. However, settling time, temperature stability, phase noise, and other nonidealities in the baseband oscillator often complicate the system design. Today, a simple direct-digital frequency synthesizer can easily generate the base-

band signals, without any of the difficulties encountered with analog designs. The use of digital baseband processing allows one to limit all of the analog hardware to the passband conversion circuitry, and also opens the possibility of simplifying the analog hardware at the expense of more complicated baseband processing. This will be explored in the next chapter, where the analog passband circuitry will be discussed.

From the analysis of the indoor environment in the previous chapter, whatever multiple access scheme chosen must successfully multiplex 10 users each transmitting 2 Mbps within 40 MHz of total bandwidth, including guard bands. As shown below, given the bandwidth and user multiplexing requirements coupled with a consideration of the severe multipath distortion encountered on the channel, it has been found that a linear QAM scheme using CDMA possesses many desirable advantages.

Modulation Techniques

Quadrature Amplitude Modulation

QAM is simply the digital version of classical analog AM, using modulation on both the in-phase and quadrature signals to achieve the same spectral efficiency as single-sideband modulation [10]. Essentially, a sinusoidal carrier waveform is linearly modulated by band-limited signal of the form

$$m(t) = \sum_{k=0}^{\infty} A_{kT} g(t - kT)$$

where $\{A_{kT}\}$ are the complex-valued transmit symbols corresponding to the user data, and g(t) is a bandlimited baseband pulse. This results in a transmitted signal s(t) equal to:
The $\{A_{kT}\}$ data symbols are chosen from a symbol constellation, as shown in figure 3.1; each group of bits to be transmitted is encoded into the complex-valued symbol shown.

$$s(t) = 2\mathcal{R}e\{e^{j\omega_c t} m(t)\} = \sum_{k=0}^{\infty} 2\mathcal{R}e\{(A_{kT} g(t-kT)) e^{j\omega_c t}\}$$

Clearly, the larger the constellation, the more bits are transmitted per symbol, and hence the required transmission bandwidth is reduced. However, since the distance between constellation points is also reduced as the constellation size increases, the probability that a symbol error occurs is increased. Under additive white Gaussian noise conditions, the

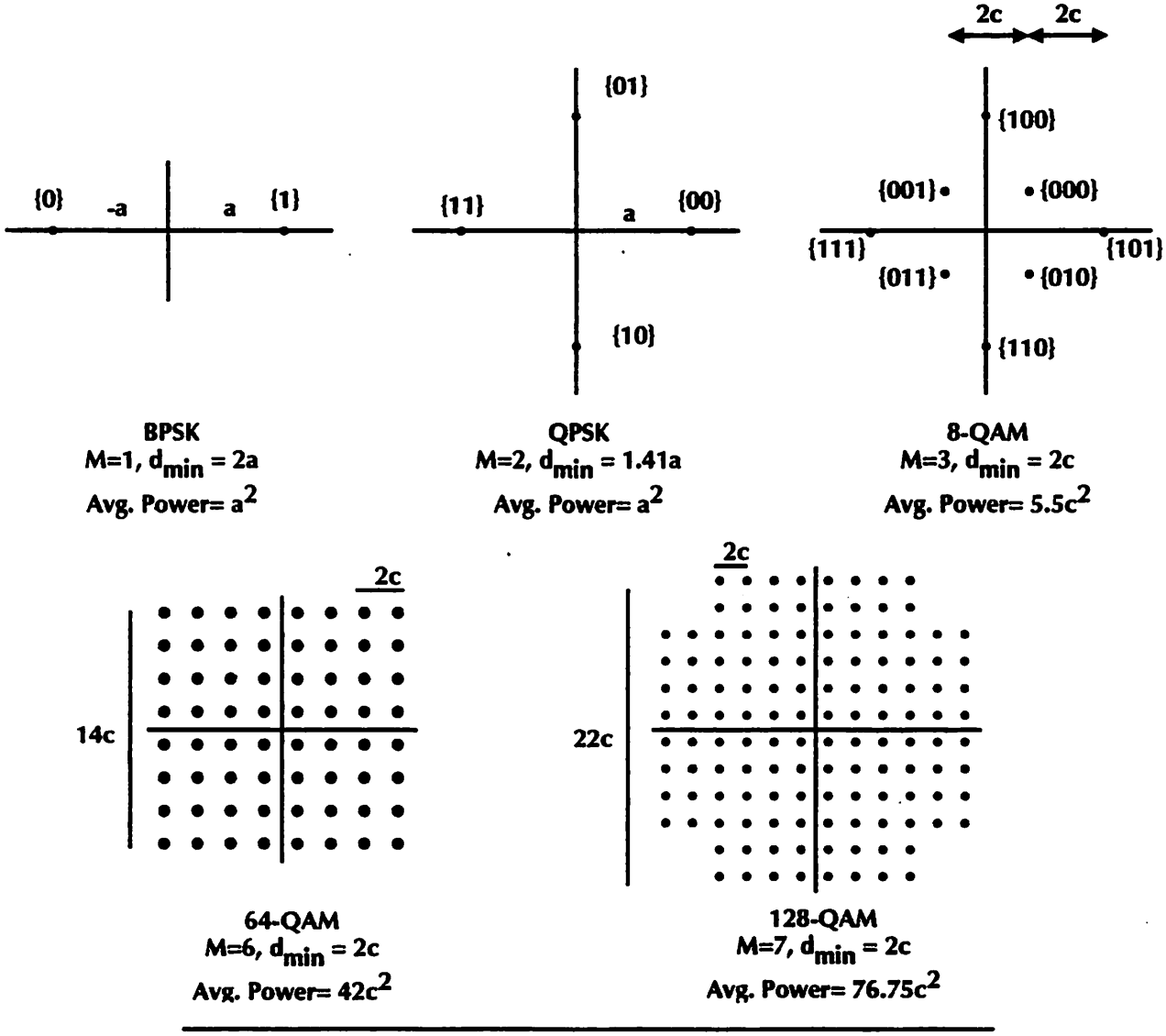

Figure 3.1: QAM Constellations (Cross-configuration)

probability of a bit error for a QAM signal is approximately given by:

$$P_{error} = \left(\frac{\eta}{M\sqrt{2\pi}}\right)\int_{q}^{\infty} e^{-(x/2)^2}\,dx = \left(\frac{\eta}{M}\right)F_{err}(q) \leq \left(\frac{\eta}{M\sqrt{2\pi}}\right)\exp(-q^2/2)$$

where $q=(d_{min}/2\sigma)$, d_{min} is the minimum distance between two nearest-neighbor constellation points, σ is the noise power, M is the number of bits per symbol, and η is a degeneracy factor equal to the number of nearest neighbors at the minimum distance.

The primary advantage of QAM is that it is known to be generally spectral efficient, which is ideal for a personal communications system with its severe bandwidth requirements; also, it is conceptually simple to equalize out the effects of the channel since the use of adaptive linear equalization is possible. However, it also requires significant hardware complexity in the receiver, since coherent demodulation has traditionally been utilized, requiring phase- and carrier-recovery circuitry. For these high data rates at such high carrier frequencies, essential components such as rejection filtering, timing and carrier recovery, automatic gain control, and voltage-controlled local oscillators become complex and expensive to implement. If QAM is to be employed, methods of minimizing the hardware complexity must be developed.

Continuous-Phase Modulation

Continuous-phase techniques present an interesting class of modulation schemes which have the desirable property that they can be incoherently detected with minimal hardware complexity. They are characterized by a constant amplitude envelope, with all of the information carried in the phase of the signal alone. The general form of the transmitted signal is given by [11]:

$$(s(t) = \sqrt{2(EsR)}\cos(\omega_c t + \varphi(t,\alpha))), \varphi(t,\alpha) = 2\pi h\sum_{k=0}^{\infty} A_{kT}q(t-kT)$$

where $q(t)=\int_{-\infty}^{t} g(\tau)d\tau$, E_s is the energy per symbol, and A_{kT} are the transmitted data bits.

Since this is a constant envelope modulation scheme, CPM is known to be power efficient at the transmitter, since a class-C nonlinear power amplifier can be used. Likewise, it is immune to channel amplitude nonlinearities; in fact, for some schemes the receiver can simply hard-limit the signal without any loss of information or signal degradation, obviating the need for high-precision automatic gain control as needed by linear modulation schemes. Another important advantage is that most CPM schemes can be demodulated incoherently, in that the phase of the modulation carrier need not be known for reliable data recovery; for example, to demodulate MSK a simple band-pass filter bank can be used to detect the various frequency tones, thus recovering the data without need for complex phase recovery.

Conceptually, CPM schemes are the digital equivalent to analog FM, with the digital signal first being low-pass filtered before frequency modulation, as shown in figure 3.2. The name CPM comes from the requirement that the phase be continuous at all points, which implies that no "jumps" in phase are allowed, as in simple frequency-shift keying. This continuity requirement improves the spectral efficiency significantly, and it is the bandwidth of the low-pass filter which determines the overall spectrum consumption of the system. For example, one popular and "spectrally efficient" CPM scheme is GMSK, or Gaussian Minimum-Shift Keying[1] [11, 12]. The low-pass filter is specified to have a Gaussian response, and it has been shown that to maximize the spectral efficiency the bandwidth of the Gaus-

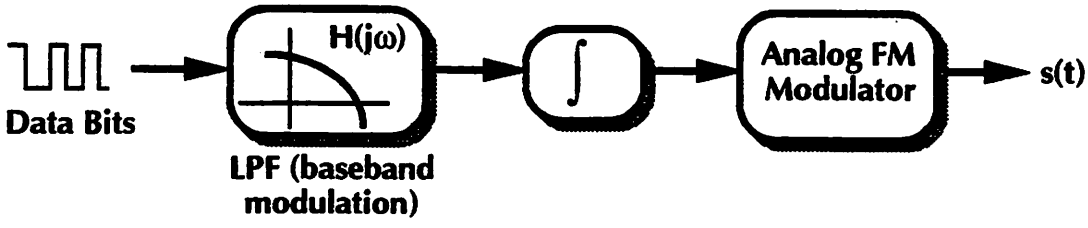

Figure 3.2: General CPM Modulator

sian filter should be set to approximately 0.25R, where R is the bit rate. To compare against simple MSK, the 99% power containment bandwidth for GMSK can be shown to be 0.86R [12], as opposed to 1.2R for MSK.

However, constant-envelope schemes also have several serious disadvantages. First, on the time-varying fading channels that the indoor environment presents, equalization is required to neutralize the intersymbol interference presented by the channel. Such equalization becomes extremely difficult with CPM schemes owing to the fact that the signal is not a linear function of the data, making adaptive techniques nearly impossible. Second, the nonlinear nature of the modulation also makes analysis difficult; for example, there is no simple form for the general bit-error rate for all CPM-type schemes, complicating the design of the baseband modulator. Lastly, the spectral efficiency of CPM is not as good as that of QAM; given the severe bandwidth constraints of our PCS system, the advantages of CPM must outweigh this loss in precious bandwidth to make the use of CPM feasible. A quantitative comparison is made below.

Modulation Schemes: Comparison and Analysis

In figure 3.3, a comparison of the important properties of several modulation schemes is shown; the modulation schemes listed represent the majority of the practical systems in use today. All the schemes are assumed to have the same transmit power $P_{transmit}$ and bit rate R bits/sec, and subjected to additive white Gaussian noise with noise power σ^2. For the class of linear schemes, the baseband pulse shape g(t) is assumed to be a Nyquist raised-cosine pulse with 50% excess bandwidth, which is both practical and realizable [11].

1. GMSK has been of the object of significant interest recently, since it is the modulation scheme that has been specified for use in the European GSM digital cellular system.

From the table, it is immediately evident that linear QAM schemes are more spectrally efficient than nonlinear CPM ones, since Q is assumed to be constant. Even for a small constellation such as QPSK, the spectral efficiency already surpasses that of an optimal GMSK scheme. This argues strongly that for a high-data rate, wideband application like the personal communications system a linear modulation scheme should be utilized. To date, nonlinear modulation has been almost invariably favoured given its transmit power efficiency and incoherent reception capabilities. However, in a wideband microcellular system, where transmit power is already minimal by necessity and bandwidth is critical, continuous-phase modulation is simply not practical.

Considering the linear schemes, another evident point is that as the constellation size increases, the BER also increases dramatically, since the complementary error function is a strong function of its argument. Intuitively, one can understand this as the distance between the constellation points is decreasing since $P_{transmit}$ is held constant. As the noise power also remains constant, the probability that a symbol is received incorrectly will increase. Plotted in figure 3.4 is a graph of the required signal-to-noise ratio $P_{avg}/(RN_o)$, which is the ratio of received energy per bit to the power spectral noise density at the receiver, versus the size of the constellation for constant BER; superimposed is the bandwidth decrease relative to the BPSK case for a fixed data rate. Clearly, the SNR must rapidly increase for the BER to remain constant. Since transmit power is constrained, one of two possibilities must occur in order to exploit the spectral efficiency of larger constellations: either decrease the noise power significantly, or alter the BER curve through the use of error-control coding. Since the noise power is determined primarily by the front-end amplifier in the receiver, an arbitrary reduction in noise power by several dB is difficult to achieve. Hence, the use of coding is necessitated.

Modulation Scheme	Type	Bits/ Symbol	99% Power BW (Hz)	Spectral Efficiency (Bps/Hz)	Probability of bit-error (BER-1)	Demodulation
BPSK	Linear	1	1.5R	0.66	$F_{err}(\sqrt{Q/1.5})$	Coherent
QPSK	Linear	2	0.75R	1.33	$F_{err}(\sqrt{Q/1.5})$	Coherent
DQPSK	Linear	2	0.75R	1.33	$F_{err}(\sqrt{Q/3})$	Incoherent
8-QAM	Linear	3	0.5R	2	$F_{err}(\sqrt{Q/2.75})$	Coherent
16-QAM	Linear	4	0.35R	2.85	$F_{err}(\sqrt{Q/3.5})$	Coherent
128-QAM	Linear	7	0.21R	5	$F_{err}(\sqrt{Q/16.1})$	Coherent
MSK	CPM	1	1.2R	0.833	$F_{err}(\sqrt{Q/2})$	Incoherent
GMSK	CPM	1	0.86R	1.16	$F_{err}(\sqrt{Q/2.38})$	Incoherent

Figure 3.3: Comparison of Modulation Schemes (AWGN only)

$Q = Pavg/(RN_o)$ (E_b/N_o ratio, thermal noise power only)

N_o = Power Spectral Density of additive noise (W/Hz)

R = Bit rate (bps); P_{avg}= Average transmit power (W)

BPSK = Binary Phase-Shift Keying

QPSK = Quadrature Phase-Shift Keying (4-PSK)

DQPSK = Differential Phase-Shift Keying (same constellation as QPSK, data is encoded on phase difference).

MSK = Minimum-Shift Keying.

GMSK = Gaussian Minimum-Shift Keying (BW of Gaussian LPF=0.25R).

Data for MSK, GMSK from [22]; the above neglects the constant multiplier for BER, since it makes little difference at high SNR.

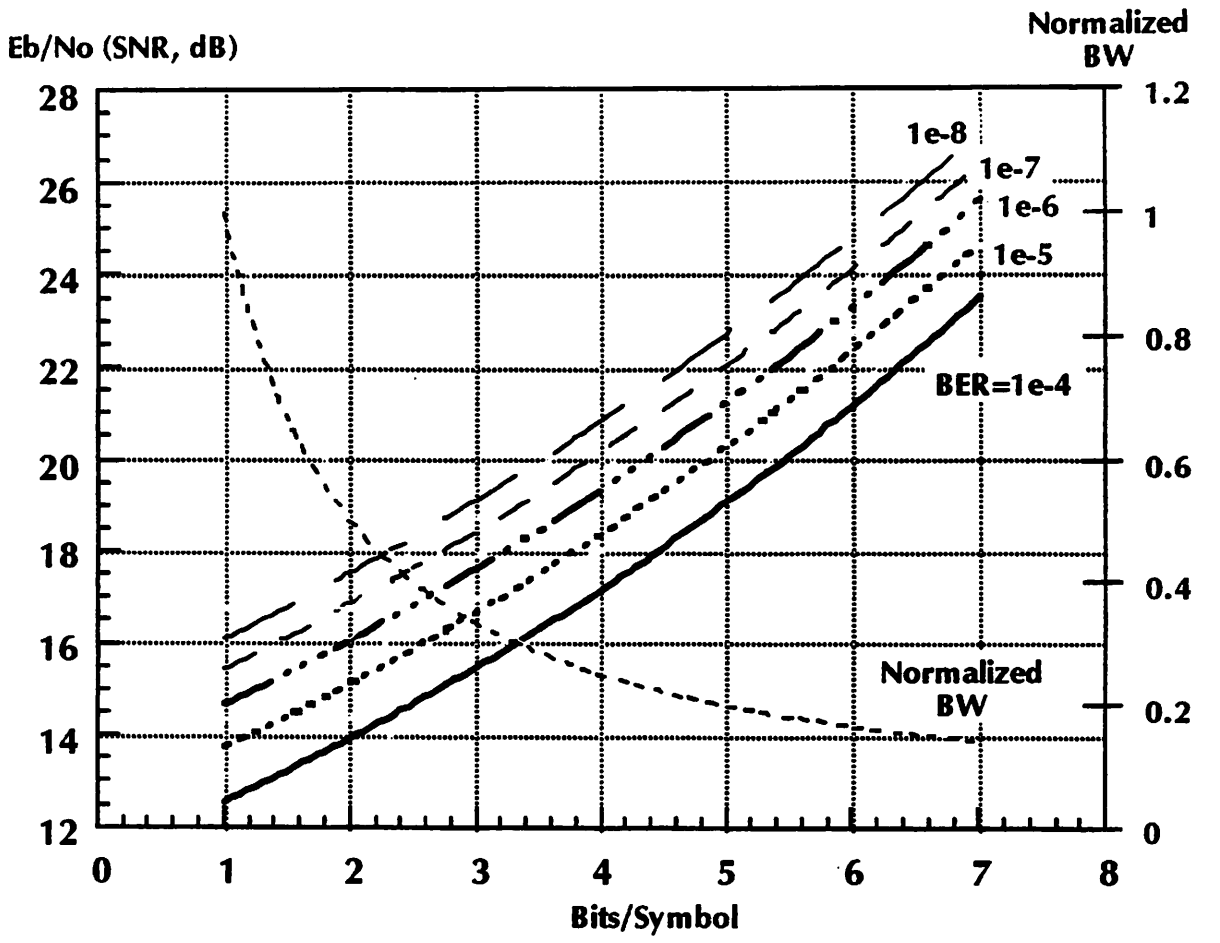

Figure 3.4: Required SNR vs. Constellation Size

Lastly, some mention needs to be made about the effects of the multipath distortion on the BER. To this point, all of the analysis has been done assuming that the only contribution to the error probability has been additive thermal noise from the receiver. This is not unreasonable, assuming that the effects of the channel have been nullified by an adaptive equalizer or the use of spread-spectrum, as described below. It is extremely difficult to analyze the effects of multipath distortion on the BER, since the transfer function of the channel varies wildly even within a local neighborhood. As mentioned in chapter 2, a fading null of 25 dB will completely overwhelm any differences between constellation size or coding, since the received SNR E_b/N_o is penalized tremendously. However, the use of diversity combining is the most effective means of combatting multipath, such as the use of two antennas, or

spatial diversity. In the simplest form of combining, the receiver can simply choose to use the antenna for which the received signal is strongest, since it is unlikely that both antennas will be sitting in extreme fading nulls. As with the multipath itself, an analytical expression for the impact of diversity on the BER is difficult to derive. However, the importance of diversity cannot be sufficiently stressed, and further work is needed to quantify the gains that combining can yield.

Multiple-Access Techniques

Multiple-access techniques are methods by which many users can access the system simultaneously, where users are "separated" from one other by allocating a small slice of the available "resources" to each user. For example, when cellular technology is employed, a form of spatial multiple access is achieved, in which two users separated by a sufficient distance can independently use the same frequencies at the same time. Another classical scheme is frequency-division multiple access (FDMA), where users are separated from each other by assigning a distinct frequency band to each. Hence, two sets of independent data can be transmitted at the same time at the same location, without resulting in interference between the two users. Below, a short discussion is presented for three other important schemes: time-division multiple access (TDMA), code-division multiple access (CDMA), and frequency-hopped spread-spectrum multiple access (FH/SSMA), which is a variant of FDMA.

Of importance is the fact that FDMA, TDMA, and CDMA can all be viewed as partitionings of the available signal space into N subsets, where N is the number of users that are to be multiplexed. Optimally, N is equal to the dimensionality of the signal space, which has been shown to be of dimension $2BT$, where B is the total available bandwidth for transmission, and T is the amount of time that is available for the N users to all transmit one symbol. For example, in FDMA each user of the N users transmits in a bandwidth of B/N, over the complete time interval T. As discussed below, for TDMA each user transmits in separate

bursts of time of length T/N, over the complete bandwidth B, and for CDMA each user is assigned an one-dimensional subspace derived from one of $2BT$ orthogonal basis vectors for the signal space. The important realization here is that no one scheme is inherently superior to any other; it is only after considering the transmission environment and the required performance that advantages and disadvantages can be discussed.

Time-Division Multiple Access

In TDMA, the users are separated on a common communications medium by ensuring that they simply transmit at different times. Perhaps the most obvious form of multiple access, it is surprising simple in concept, and variants have long been used in satellite communications and wired local area networks such as Ethernet. Transmitted bursts of "low data rate" user information is multiplexed and interleaved into time-slices on a "high data rate" communications channel, as shown in figure 3.5. Received TDMA data is put through a demultiplexor to reverse the interleaving process, extracting only the time slices that belong to that user. The time slices of user data are typically placed into frames, where each frame contains overhead acquisition information for timing and carrier recovery (framing bits) in addition to the data itself. From this, it is clearly evident that a considerable amount of overhead is expended in reacquiring synchronization and training the adaptive equalizers from frame-to-frame; for the system of figure 3.5, 1 in 3 of the bits transmitted is used only for control purposes.

TDMA multiplexing has been heavily favored for the past several years for use in second- and third- generation microcellular digital radiotelephony systems such as the European GSM project [13], mainly because of its simplicity and ease of implementation. However, problems have already been encountered with the use of TDMA on microcellular channels; to surmount the time-varying channel multipath distortion and fading necessitates the use of an adaptive decision-feedback equalizer (DFE). Several excellent discussions on adaptive DFE design exist [11, 14]; the analysis below only utilizes the results from the theory.

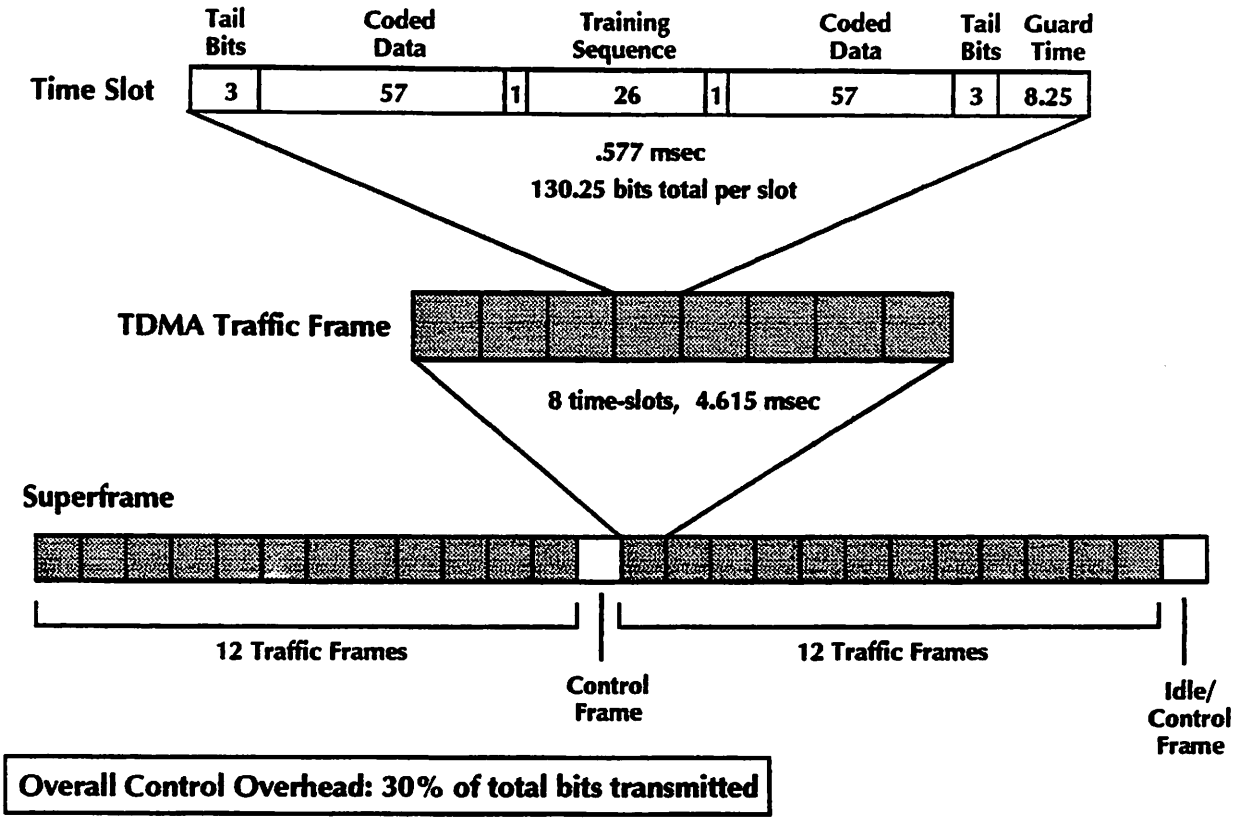

Figure 3.5: TDMA Framing/Control Strategy (GSM)

The usual design rule is that the number of taps in the equalizer structure should provide a time-delay span approximately equal to the length of the channel impulse response that needs to be equalized. For low-data rate (100 kbps TDMA burst rate, 32 kbps average user rate) voice telephony applications, this implies a 3-7 tap adaptive equalizer[2] should be sufficient, where the number taps is the total of the forward filter and feedback filter [15]. Unfortunately, to enable these equalizers to converge sufficiently quickly, use of the so-called "fast" RLS, modified LMS, or the Kalman filter algorithms is needed. These algorithms are complex, and tend to be computationally intensive even at the low data throughput de-

[2] These data values are for the European GSM system.

manded of voice telephony. Under the consideration that the *user* data rates needed for the PCS are already over ten times larger, this implies that the equalizer structure needs to have on the order of 30-50 taps in order to achieve equalization, even assuming no increase in bit-rate for TDMA is incurred. The fact that the channel data rate has increased implies that multipath has an even greater impact on the transmission, since the significant reflections now spans a larger number of symbols; the problem of equalization is thus severely aggravated. Given the complexity and difficulties that have been encountered in forcing even small adaptive DFE's to converge quickly, it will be difficult to design an adaptive DFE that will meet the performance requirements of a PCS transceiver system. The only mitigating factor is that the multipath profile is changing slowly with respect to time, allowing slower convergence rates, as opposed to the typical fast-fading pattern encountered with operation of cellular transceivers in a moving vehicle.

Code-Division Multiple Access

Another multiple-access scheme employs code division, where the users are allowed to transmit simultaneously in the same frequency band, without requiring them to be separated physically by a large distance. Instead, each user is assigned a code waveform from the set $\{p_j(t), j=1..N\}$, where the set represents an orthogonal basis for the signal space and each $\{p_j(t)\}$ occupies B Hz of bandwidth. Since the set $\{p_j(t)\}$ is assumed to be orthogonal, where

$$\int_0^T p_k(t)\, p_j(t)dt = \delta(j, k)$$

δ is a Kronecker-delta function.The assumption is also made that the autocorrelation function of each of the $p_j(t)$ is also a delta function, i.e.:

$$\int_{-\infty}^{\infty} p_j(\tau)p_j(t+\tau)d\tau = \delta(t)$$

Thus, the transmitted waveform for user j is then given by:

$$2\mathcal{R}e\left\{\sum_{k=0}^{\infty} A_{kT}\, p_j(t-kT)\, e^{j\omega_c t}\right\}$$

Since these signals are transmitted at the same time in the same frequency band, the received signal at any receiver is just the linear superposition of these over all *N* users.

Now, after the signals have traversed the channel and have been mixed down to baseband, the receiver is presented with the problem of extracting that particular user's data from the superposition of the data from many users and additive noise, i.e.:

$$\sum_{n=1}^{N}\sum_{k=0}^{\infty} A_{kT}^{n}\, p_n(t-kT) + n(t)$$

However, this is simply accomplished: the receiver knows *a priori* which pulse $p_j(t)$ was used to transmit. Hence, by simply correlating the received signal against the known $p_j(t)$, the correct data can be extracted, since:

$$\int_{kT}^{(k+1)T}\left\{\sum_{n=1}^{N}\sum_{k=0}^{\infty} A_{kT}^{n}\, p_n(t-kT) + n(t)\right\} p_j(t-kT)dt = A_{kT}^{j} + \int_{kT}^{(k+1)T} n(t)\,(p_j(t-kT)dt)$$

As the noise signal is assumed to be nearly white, its power must be evenly spread out among each of the code waveforms, since these represent a basis for the signal space; hence, the power in the noise term after correlation has been reduced by approximately a factor of *N*. This is an elegant scheme for achieving multiple access, since it simply performs a correlation in the receiver, which can be performed digitally.

Essentially, CDMA utilizes pseudo-noise spread-spectrum techniques to achieve its ends, since the original user bandwidth has been expanded to occupy the entire bandwidth available. Alternatively, this can be viewed as multiple high-data rate symbols being sent in place of one user symbol, which effectively is a form of coding redundancy. This increase

in error immunity is called the processing gain P_g, defined as the ratio of the spread spectrum to the original spectrum, and is exemplified by the apparent reduction in the noise power after the correlation despreading [16,17,18]. However, beyond simply providing a multiplexing technique, the coding and processing gains of spread-spectrum also yield two other interesting properties: spread-spectrum signals are low-probability-of-intercept, since knowledge of the precise code used is required, and spread-spectrum signals tend to be naturally immune to multipath distortion [11,19]. Both of these are excellent advantages considering the indoor environment that the transmission is taking place in, and the immunity to multipath is especially important, since it implies that the complex adaptive equalizer is not needed.

The immunity to multipath distortion can be understood by increased time-resolution of a spread-spectrum signal. Since the $p_j(t)$ spreading function is assumed to have a delta-function autocorrelation, the arrival time of each symbol can be determined precisely, by using a "sliding correlator" to detect when the correlation is nonzero. For a single transmitted symbol at t=kT, this sliding correlation is given by:

$$\int_{-\infty}^{\infty} A_{kT} p(\tau - kT)\, p(\tau - t)\, d\tau$$

This is clearly nonzero only when t=kT, as expected. However, suppose the signal has been subject to multipath distortion. Then the sliding correlation yields a nonzero value not only at t=kT, but also at all of the multipath arrival times thereafter as well. Hence, the arrival times of the signal can be resolved precisely; it is only when a multipath arrival time happens to be a precise multiple of T is there interference.

Of course, this is assuming that the $p_j(t)$ are continuous signals that possess an delta-function autocorrelation. In practice, the $p_j(t)$ are digital signals, which take on only values in the set $\{\pm 1\}$, and are similar in nature to Walsh functions. The chip interval T_{chip} is the higher-frequency sampling rate for the $p_j(t)$ spreading function, and T_{chip} is related to the

symbol period T by $T_{chip}=T/P_g$, where P_g is the processing gain given above, since a bandwidth spread of P_g yields the same increase in data rate. Thus, the multipath arrivals can be resolved to a resolution given by T_{chip}; hence, only multipath arrivals that occur within $\pm T_{chip}$ of t=kT will result in distortion and interference. Clearly, the greater the spreading factor, the better the immunity.

An immediate question that arises is how many users can practically be supported in a PN spread-spectrum CDMA system. Assuming that error-control coding is utilized on top of simple spectrum spread and the waveforms are perfectly orthogonal, it has been shown [19] that the decoder error probability P_{error} is bounded by

$$P_{error} \leq \left(\frac{M-1}{2}\right) F_{err}\left(\sqrt{\frac{2P_g R_c d_{min}}{(N-1)+\frac{\sigma}{P_{avg}}}}\right)$$

where M is the size of the code, P_g is the processing gain, R_c is the code rate, d_{min} is the minimum code distance, P_{avg} is the average power of the user signal, and σ represents the additive Gaussian noise power in the receiver. For a specified error probability P_{error} and N users, the design space for the required coding gain, processing gain, and receiver noise in the system is thus constrained.

Unfortunately, spread-spectrum also has its share of problems, the critical one being the synchronization required in the receiver. In the above analysis, it was assumed that the correlation performed in the receiver is perfectly aligned with the incoming signal. If they are not synchronous, the scheme fails since the receiver only sees the partial correlation, which is designed to be small, and thus results in severe information loss. Several methods exist by which synchronization can be acquired, such as a multiple correlation receiver, RAKE receiver, or delay-locked loop. All of these methods are unfortunately of high complexity, and achieving synchronization lock is an extremely difficult issue especially for high data rate communications [19].

The design of the codes used, i.e. the set of orthogonal signaling waveforms, is another difficult issue that has received extensive discussion in recent literature [20]. Perfectly orthogonal codes are difficult to generate; hence, all practical codes always have some nonzero correlation between the {pj(t)} transmit waveforms. This leads to near-far crosstalk problems, where a transmitter close to the receiver can corrupt the data from one further away despite code orthogonality, since the received power from the nearby transmitter is much higher, and hence the undesired correlation may mask out the actual signal. For an indoor microcell environment with short transmission ranges, one does not expect such vast variation in signal power, so the severity of this problem is reduced; likewise, several elegant adaptive power control schemes have been proposed which reduce the transmit power when the distance between the base station and the mobile terminal is small. Likewise, several reasonable choices for PN spreading schemes have already been proposed for CDMA, such as Gold or Kasami sequencing [23, 24], which have small partial correlation effects and are easily generated using maximal-length shift register techniques.

Furthermore, unless all of the transmitters to be multiplexed are synchronized, there exists a problem with cross-correlations. This is similar to the above problem requiring the receiver to be synchronous with the user data; the signals from the other transmitters should correlate to zero, or at least the minimum correlation achievable. If the transmitters are not synchronized, then potentially larger correlations will be seen, which if not designed to be small can also corrupt the received user data. This difficulty leads to spectral inefficiency; usually, more spread, and hence more bandwidth, needs to be utilized to get achieve the same multiplexing as FDMA or an ideal TDMA scheme [17].

Lastly, if CDMA is to be employed, small constellations in the QAM modulation are dictated. The use of larger constellations, while reducing symbol rates, significantly decreases the tolerable signal to noise ratio at the receiver. Using the above expression for the error probability, plotted in figure 3.6 is the number of users that are supportable in a CDMA environment as a function of constellation size assuming that the processing/coding gain and

BER remain constant, and that the penalty in P_{error} for using higher-order constellations is approximately equal to the penalty incurred by the loss in E_b/N_o.To offset this decrease in user capacity, increased coding and processing gain will be required. The situation with CDMA is unique; as transmit power scales, so does the noise, since the vast portion of the noise interference comes from other users. Due to nonzero correlations and other nonideal effects, even greater spreading factors and processing gain will be required for large constellations, thus offsetting the decrease in symbol rate. It has been found that QPSK provides a reasonable constellation configuration for use in CDMA [16, 19].

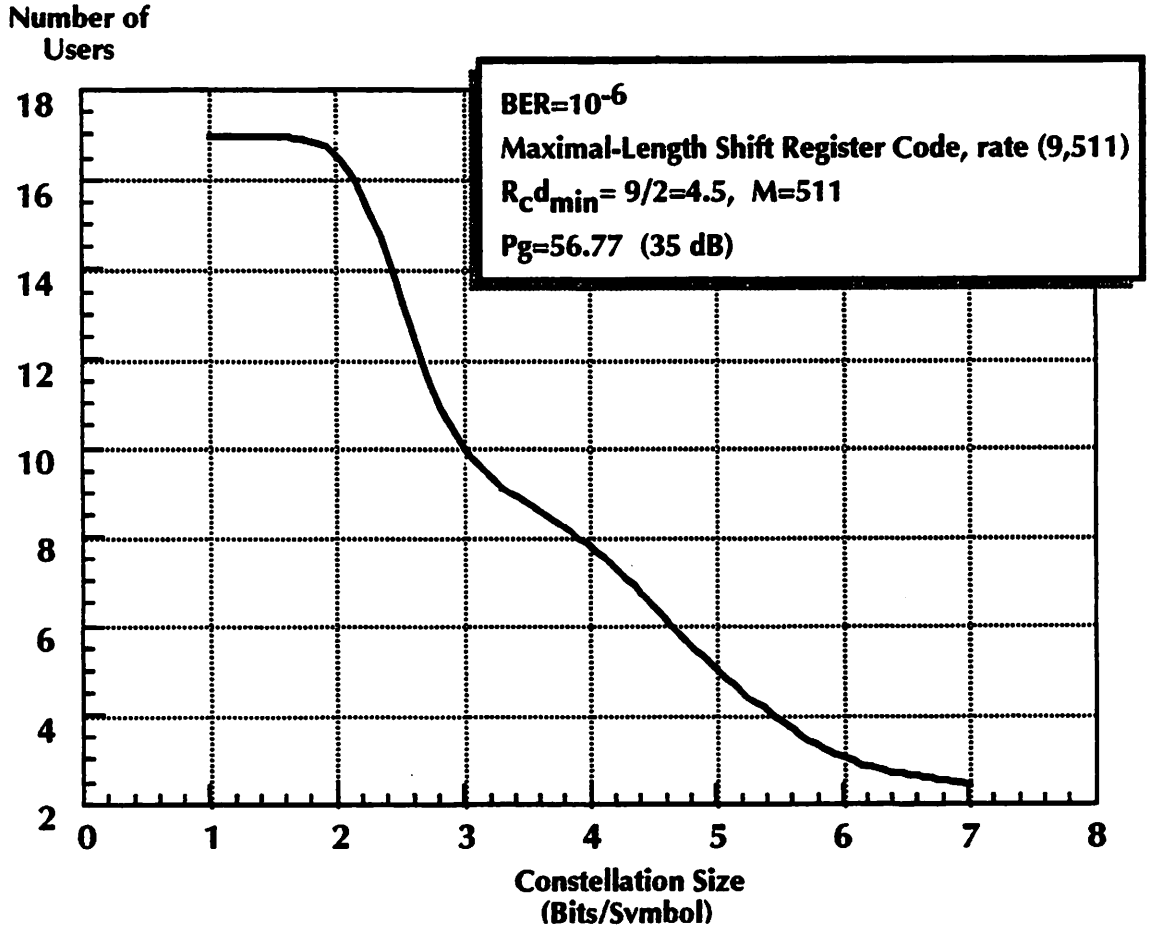

Figure 3.6: Number of supportable users in QAM/CDMA system

Frequency-Hopped Spread-Spectrum Multiple Access

A variant of FDMA, frequency-hopped spread spectrum allocates each user B/N Hz of spectrum, where B is the total bandwidth available to be shared. However, unlike pure FDMA, the frequency band each user is transmitting in is changing with time, hence the name frequency-hopped. The spectrum is "spread" in the sense that the user ranges over the entire B Hz of bandwidth due to the hopping, but only occupies a small amount at any given instant.

Although it would seem that FH/SSMA would be subject to the same disadvantages that TDMA has, the transmitted bit rate for FH/SSMA is the same as the data rate, unlike TDMA. Hence, the fading is narrowband - the channel transfer profile is approximately flat over the bandwidth, and hence much less equalization is needed. In other words, relative to the symbol rate the time-span of the multipath is much shorter than that of TDMA, since we do not need to transmit in high-frequency bursts. In figure 3.7, the simulated transfer function from Chapter 2 is plotted again, except with the bandwidth occupancy of the TDMA and FH/SSMA shown; the narrowbanding effect is evident. Of course, this narrowbanding advantage would also be realized with pure FDMA - no mention of the necessity of frequency hopping has been mentioned yet. Also shown in figure 3.7 is an example hopping pattern for FH/SSMA - on hop number 2, it drops directly into a 25 dB fading null. In all likelihood, with such a deep null, the narrowband signal will be completely engulfed by noise, resulting in link breakdown. It is the fact that it will quickly hop back out of the null that makes FH/SSMA a viable multiple access scheme at all.

The major disadvantage of FH/SSMA arises from the fact that it simply must hop in frequency, necessitating a frequency-agile wideband oscillator. Unfortunately, since the carrier must span the entire bandwidth of the system at the gigahertz passband, direct-digital synthesis becomes impossible, thus requiring an agile analog synthesizer. The frequency settling time for accurate, low-phase noise phase-locked loops tends to be on the order of microseconds [21], or tens of symbol periods, resulting in a resynchronization delay every

time a hop is performed. Likewise, all of the transceivers active within the cell must be synchronized in their hopping scheme, otherwise disastrous cochannel interference will result if two transceivers randomly happen to occupy the same frequency band at the same time.

Toward an Indoor Transmission Scheme

With the above discussions of multiplexing and modulation, the various advantages and disadvantages must be considered in light of the characteristics of the indoor picocellular propagation environment, and the data rates required by the personal communications system. The real advantage of CPM techniques, power efficiency in the transmitter, is negated by the fact that the distances involved with a picocellular scheme are small, with very low

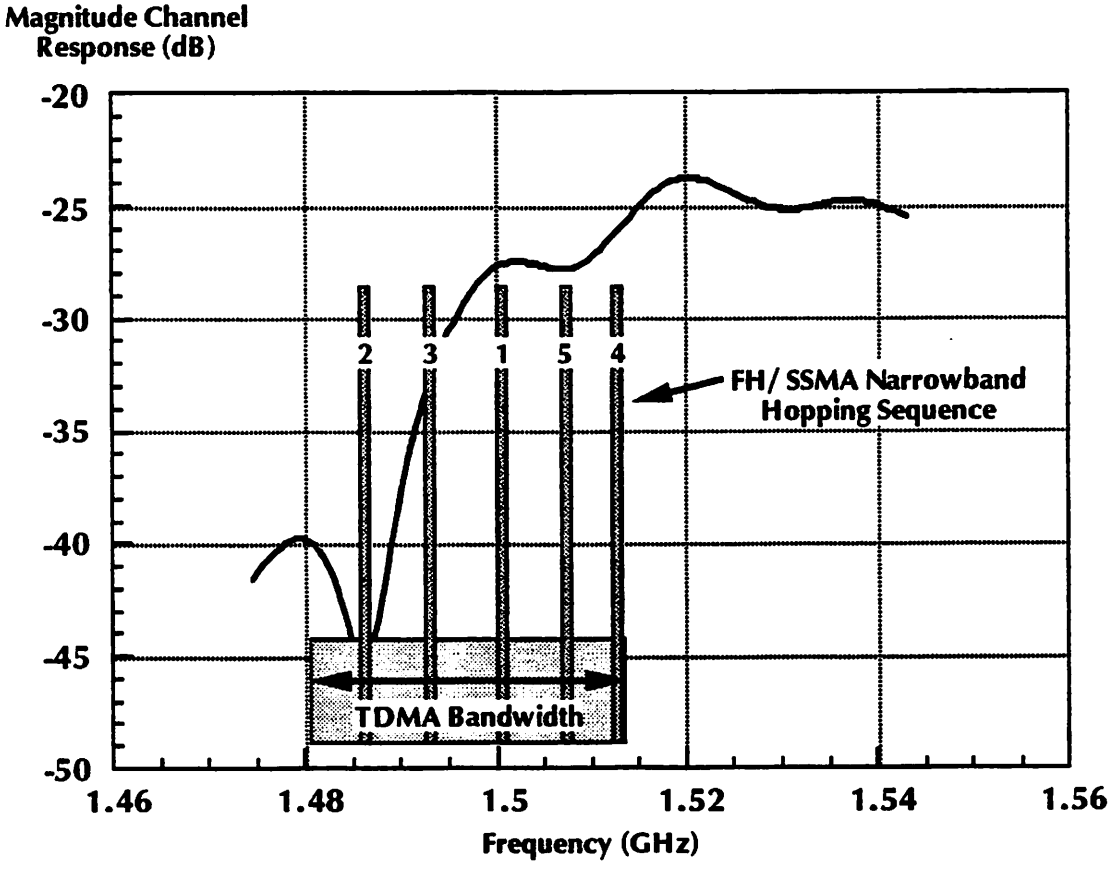

Figure 3.7: Comparison of TDMA and FH/SSMA under multipath

transmit power requirements. Given the poorer spectral efficiency of CPM, QAM is thus the logical choice for modulation. Under the same considerations, the equalization difficulties and overhead costs that have already been encountered with TDMA do not bode well for its use in a system that already requires user data rates of 2 MBps. FH/SSMA is an attractive compromise between TDMA and CDMA, possessing the simplicity of an TDMA scheme with much lower equalization complexity than TDMA and much lower design complexity than CDMA. However, the natural immunity of CDMA to multipath and its provision of low-probability-of-intercept transmission makes it a natural choice for use in such a personal communication system, despite its lower spectral efficiency and difficulties in design; we will see in the next chapter that if CDMA is employed several significant reductions in analog hardware complexity can be achieved essentially for free.

Hence QAM/QPSK, coupled with CDMA, provides several desirable properties for a PCS transceiver system. Estimated spread factors up to 40 can be achieved given that 40 MHz is available per cell, yielding commensurate processing gains of around 16 dB. With these spreads, there certainly exist a sufficient number of PN codes to support ten users; code reuse, just like frequency reuse, is possible in a cellular system.

The real issue lies in the hardware complexity required for implementing these schemes. Whether or not a low-cost, manufacturable, integrated microwave RF modem that can achieve these performance levels remains to be seen. With this in mind, chapter 4 addresses this issue, analyzing these hardware performance requirements and presenting a new transceiver architecture.

CHAPTER 4

ANALOG MODULATION AND VLSI TRANSCEIVERS

In this chapter, the issues behind the hardware implementation of an analog modulator/ demodulator will be discussed, for a design capable of supporting the required baseband data rates in excess of 2 Mbps using QPSK and CDMA. In addition to meeting the system performance requirements, another major design goal is that the analog hardware be simplified as much as possible; since the carrier frequencies are well above 1 GHz, the complexity and difficulties in implementation imply that simplifying the circuitry or relaxing the required analog performance should be paramount. Given the tremendous levels of digital computation achievable by today's scaled MOS technologies, as much as possible of the required signal processing should be implemented at baseband, in the digital domain. Many analog transceivers used in today's digital communications systems do not make use of the fact that the underlying signal is digital, and simply utilize a conventional analog transceiver after converting the digital bit stream into an analog baseband signal. Use of such techniques as sampling demodulation, and homodyne receiver architectures all present new methods in developing high-performance demodulators, which take advantage of the fact that digital data that is being transmitted.

Concomitant with the goal of simplification is the desire for monolithic integration of as much of the analog circuitry as possible. Traditionally, the realm of gigahertz-band RF front-end circuitry has been dominated by discrete GaAs MESFET's using stripline filters. Such discrete-element designs consume significant amount of area on a circuit board; given that separate packages need to be utilized for each component, trace lines must be placed carefully to minimize cross-coupling, and termination resistors are required to minimize transmission-line effects from the trace. Likewise, discrete designs are not good from a power consumption standpoint, since the power required to drive high-speed analog signals across interconnect lines is directly proportional to the amount of parasitic leakage that must be overcome. Since the parasitics for on-chip interconnects are at least an order-of-magnitude lower than board-level interconnects, the power savings realized by monolithic circuit integration are considerable [25]. To date, most of the applications that utilize gigahertz and higher frequencies have been mainly for fixed, nonportable installations such as ground satellite transceivers, and hence power and size have not been a problem. For portable applications, low power and small size are of paramount importance, thus demanding that highly integrated analog technologies be exploited.

Silicon for High-Speed Analog

Given the advances in silicon processing and technology of the past decade, digital technologies have seen a revolutionary breakthrough in both performance and size through the use of device scaling, especially in the arena of silicon complementary MOSFET's. For micro- and millimeter- wave IC technologies, gallium arsenide has been used almost exclusively to date, despite the extra cost and processing difficulties, simply because silicon devices have not possessed the necessary performance. However, the same benefits derived from scaling are reflected in analog applications as well; increases in digital switching speed commensurately yield increased analog small-signal bandwidths.

In figure 4.1, the achievable unity-gain bandwidth f_t for the three major IC technologies - silicon MOS, silicon bipolar, and GaAs - is indicated [26]. Since the carrier frequencies must be placed in the low microwave bands around 1.5 GHz, an f_t of approximately 20 GHz will be needed to meet the performance requirements. Clearly, both conventional silicon MOS and bipolar devices are below this point, although research efforts in bipolar processing have already produced BiCMOS devices with f_t's as high as 16 GHz [27]. Already, many of the components used in present-day analog cellular telephony at 900 MHz have been integrated to a subsystem level using silicon bipolar processes, such as the Plessey SL6643 wideband amplifier-mixer. However, another important point is illustrated in figure 4.1 - silicon MOS performance is nearing the levels necessary to operate at 1.5 GHz. Complete implementation in silicon MOS is interesting since other digital processing functions can be easily included on the same chip; likewise, the sheer manufacturing costs of MOS are much lower than for bipolar, GaAs, or a hybrid BiCMOS.

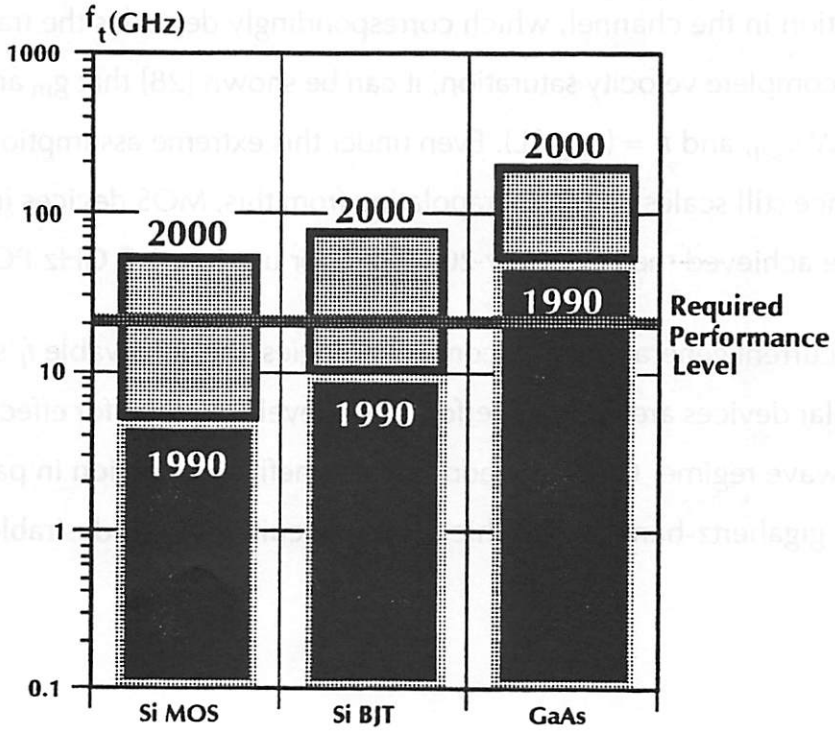

Figure 4.1: Analog Performance Comparison for MOS, BJT, and GaAs

While MOS has been regarded in the past as a "slow" technology, the scaling of device feature sizes has increased MOS performance dramatically over the past decade, and will continue to do so as devices are reduced into the deep-submicron regime. In particular,

$$f_t = \frac{g_m}{C_{gate}} = (\mu_o C_{ox}(\frac{W}{L})(V_{gs} - V_t)) / (W L C_{ox}) = \frac{\mu_o}{L^2}(V_{gs} - V_t)$$

where V_{gs} = gate bias voltage, V_t = device threshold voltage, C_{ox}= oxide capacitance, and μ_o = channel mobility; W and L are the width and length of the device channel, respectively.

As the channel length L is reduced by a factor λ, the f_t thus increases as λ^2. This is borne out well by the performance scaling of MOS devices seen over the past decade. In 1980, a conventional MOS device was fabricated with a minimum drawn length of 10μ, and yielded a corresponding f_t of around 80 MHz; today, a 1.2μ channel length NMOS device can achieve an f_t of 5 GHz, nearly 100 times larger. In figure 4.2, the f_t of MOS devices as a function of channel length is shown. Below 1μ, short-channel nonidealities begin to dominate, such as velocity saturation in the channel, which correspondingly degrades the transconductance g_m. Assuming complete velocity saturation, it can be shown [28] that g_m and f_t are now given by $g_m = C_{ox} W v_{sat}$, and $f_t = (v_{sat} / L)$. Even under this extreme assumption, the analog device performance still scales as 1/L. Extrapolating from this, MOS devices in the 0.1 to 0.2μ range will have achieved the necessary 20 GHz f_t for use in a 1.5 GHz PCS transceiver.

Thus, with the current generation of silicon technologies, the achievable f_t's of conventional MOS and bipolar devices are nearing performance levels required for effective operation in the low-microwave regime. Given the additional benefit of reduction in parasitics afforded by integration, gigahertz-band monolithic analog circuitry is both desirable and feasible.

Transceiver Performance Requirements

Traditionally, the design of linear transceivers have utilized a frequency-coherent demodulation system, as depicted in figure 4.3a. The receiver is very simple in concept: mix down to baseband, sample, and decide on the received symbol. Unfortunately, such a simple structure was found not to work well in practice; homodyne demodulators were considered to be generally impractical, owing to the extreme precision required in the receiver oscillators and band-rejection filters. Due to the coherency assumption of the receiver mixer, we need to perform both carrier and timing recovery; likewise, to counter multipath distortion, equalization must to be performed, with an adaptive decision-feedback equalizer. Although the transmitter structure remains essentially the same, a practical receiver is complicated by

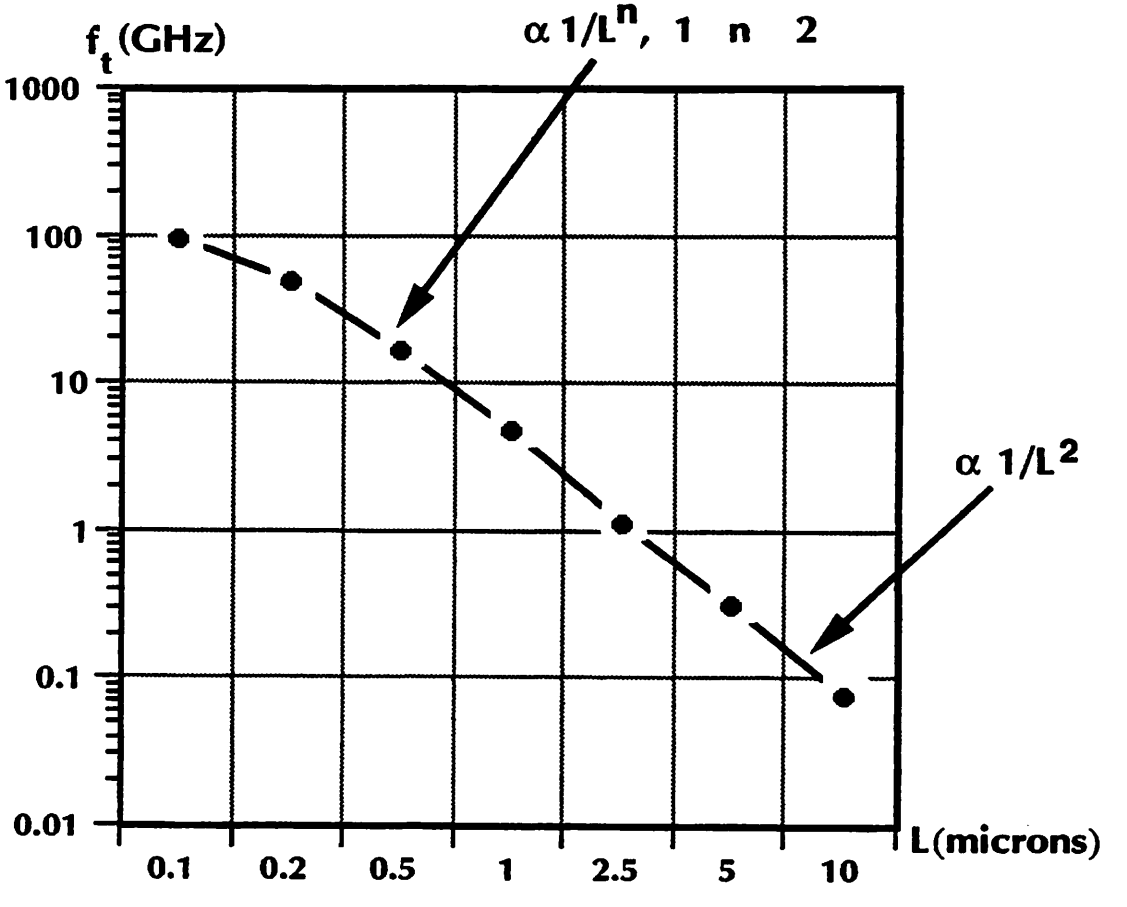

Figure 4.2: f_t vs. Channel Length for MOS devices

several blocks, as shown in figure 4.3b. With the addition of spread-spectrum for CDMA, the resulting transceiver is shown in figure 4.3c. Of course, the equalizer is now absent, to be replaced by the despread correlation and synchronization blocks.

In chapter 3, the difficulty in implementing such a structure was mentioned. The main impediment lies in the complex feedback loop required to maintain zero frequency offset and zero phase offset in the IF-to-baseband mixdown VCO with respect to the modulated signal. It would be ideal if the system could be designed to be "immune" to frequency offset and phase offsets, or if the offsets themselves could be made sufficiently small without requiring a feedback loop.

Since simplifying the circuitry is a key goal, the possibility of a utilizing a true homodyne demodulator will be explored along with the discussion below, where the signal is brought directly down from the passband into baseband. Since no IF is needed, as in a heterodyne scheme, the hardware cost is essentially cut in half, since only one filter-amplifier-mixer stage is needed to achieve the demodulation, at the expense of high-performance compo-

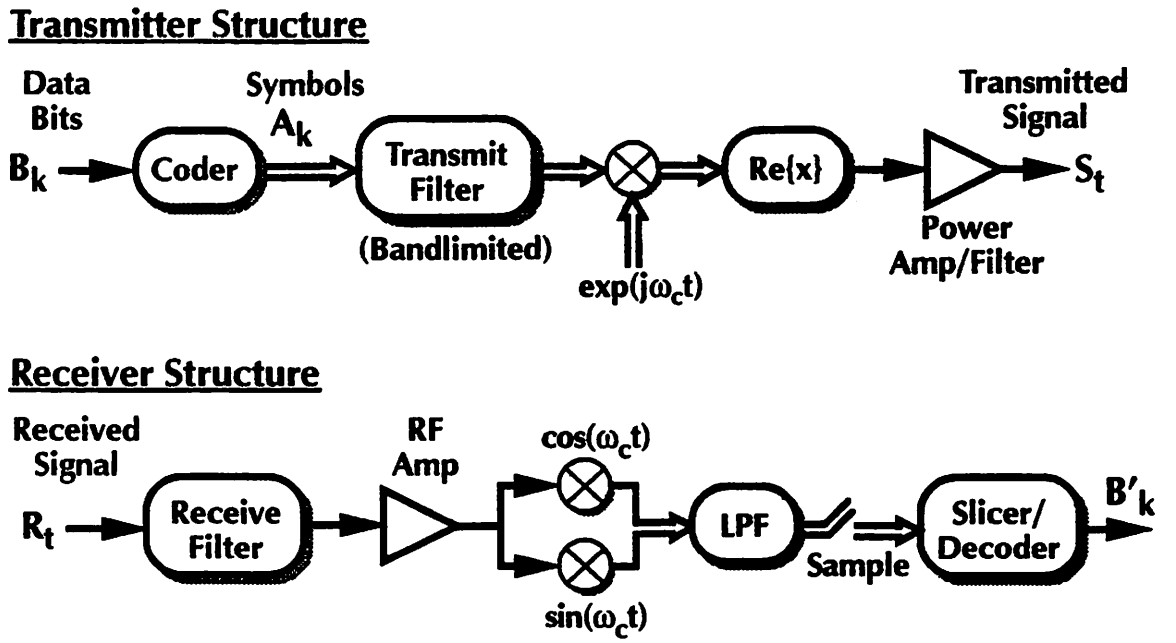

Figure 4.3a: Simple linear transceiver

Practical Coherent Receiver Structure

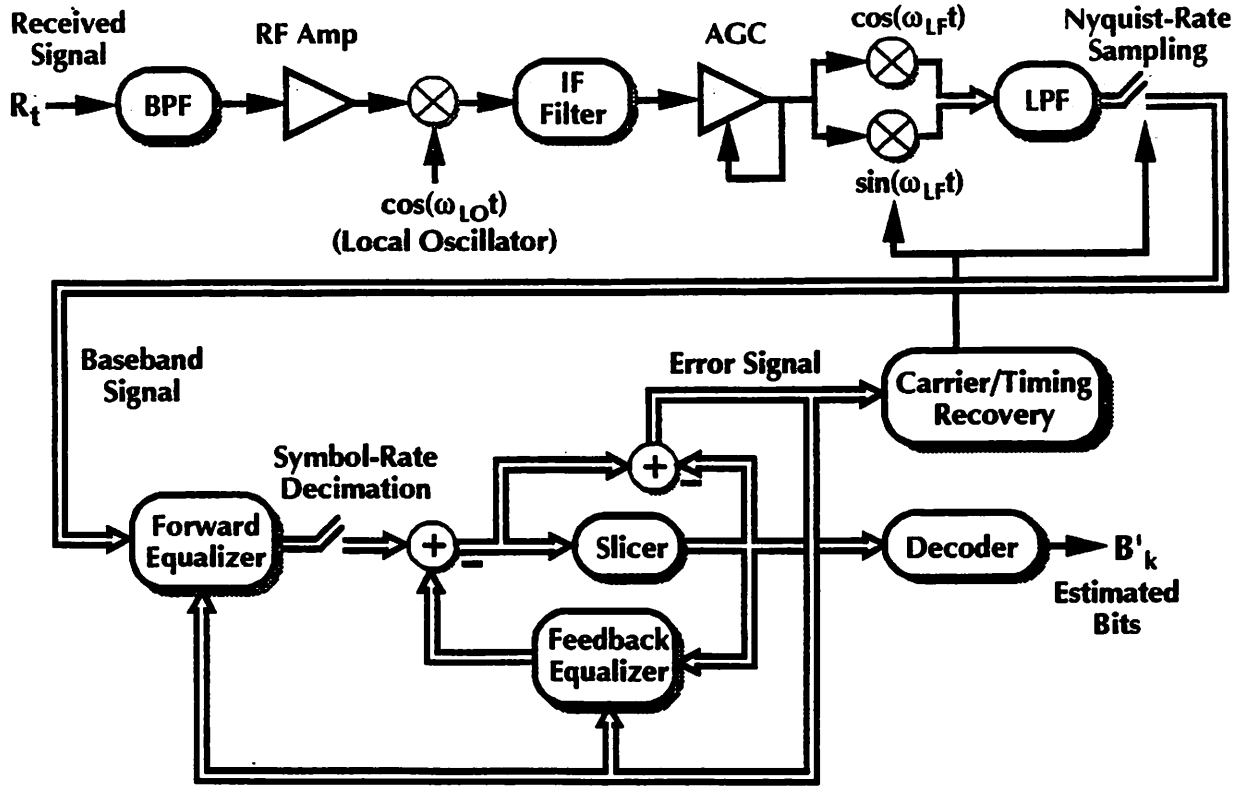

Figure 4.3b: Heterodyne Receiver w/ Equalizer

nents at the RF bands. Already, some new analog transceivers are moving away from conventional heterodyne structures, and taking advantage of the hardware reduction afforded by homodyne architectures [29].

Examining the block diagrams of figure 4.3, there are essentially five critical analog components to the transceiver - amplifiers, filters, oscillators, mixers, and A/D converters. Each will be discussed in turn, in terms of their performance requirements and feasibility for integration. A new structure utilizing sampling demodulation in the receiver will also be proposed, which succeeds in eliminating the explicit mix-down step by careful control of the A/D sampling rates, and which takes advantage of spread-spectrum coding to minimize the filter requirements.

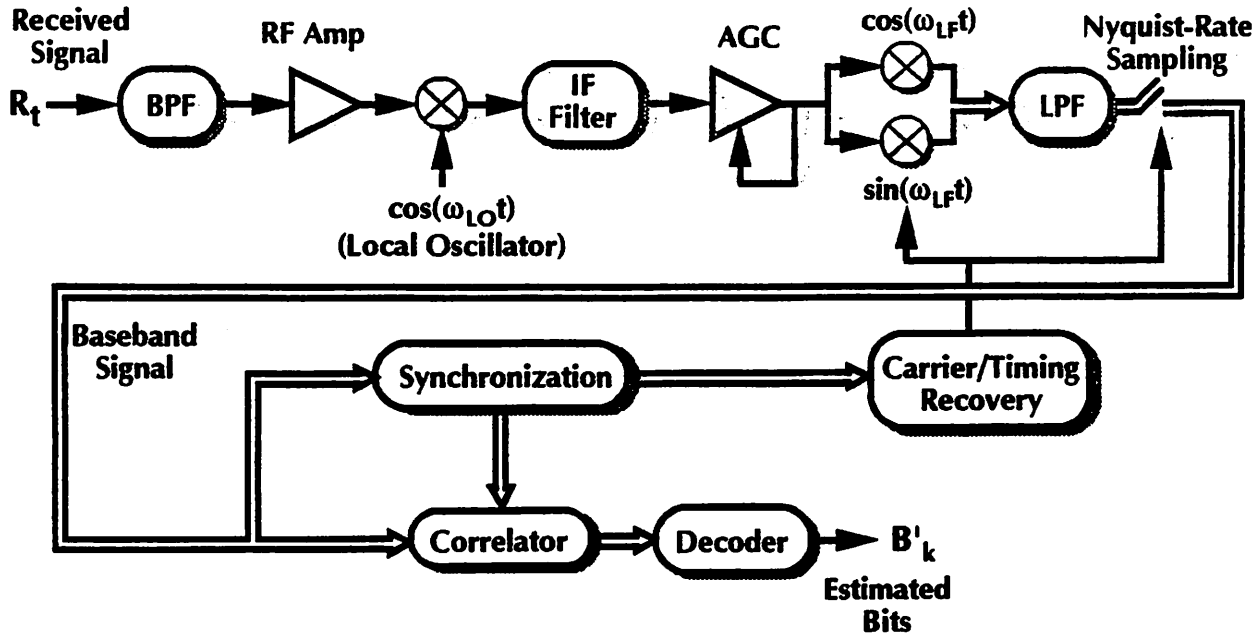

Figure 4.3c: PN/SSMA Receiver Structure

Amplifiers

Of all of the five components, the amplifier is the one whose performance is most directly related to the achievable f_t of the underlying technology, since the maximum -3 dB bandwidth is determined by the value of f_t. In essence, f_t is the frequency at which no gain can be extracted from the device; parasitics have dominated to the point that the active device has become a passive network. Assuming a single dominant-pole frequency response in the amplifier, the relationship between gain and bandwidth is thus given by

$$\text{Gain} \cdot (\text{-3dB bandwidth}) \approx f_t$$

assuming that the device itself is the limiting factor in the performance. Thus, for an f_t of 20 GHz, and a required -3 dB bandwidth of 2 GHz, a gain of approximately 20 dB can be reasonably expected from a single stage amplifier. More realistically, a 5 to 15 dB per-stage gain can be expected, since parasitics will degrade the performance considerably at these frequencies.

This immediately begs the question of how many stages will be needed, or how much total gain, which must be answered by a consideration of the transmission environment. For the transmitter, the signal are relatively large, and do not need tremendous levels of amplification. For the receiver, assuming a microcellular transmit power of -10 dBm and 0 dBi antenna gains, the received power at the boundary of a 4 meter cell is approximately -60 dBm, or 0.07 mV across a 50Ω load. A approximate net gain of 10000 will be needed to amplify this to a usable level between ±1V, or 7 stages of amplification at 10 dB per stage. If all of this amplification is to be done at the RF carrier frequency, as in a homodyne scheme with no baseband amplification, maintaining stability is a considerable problem, whereas with a heterodyne scheme major portion of the amplification is performed in the IF band, at a lower frequency.

Lastly, it has already been stated that the noise performance of the transceiver is limited by the front-end amplifier. For a linear QAM scheme, figure 4.4 plots the required overall noise figure (NF) in dB for a BER of 10^{-6} as a function of constellation size, for a -10 dBm transmit power, a transmitter-receiver separation of 4m, a 1m path-loss reference of 30 dB, and a path-loss coefficient of 2.6. These conditions imply an average received signal power of -58 dBm; using the SNR data from chapter 3, the required noise figure can easily be determined, relative to a 50Ω dummy load. It is interesting to note that the NF requirement peaks around a constellation size between 2-3 bits/symbol. Two opposing factors are seen here: the decreasing bandwidth (and hence noise power) as constellation size increases, and the decreasing distance between constellation points; the first serves to lower the BER; the second increases it. Surprisingly, the peak occurs for an extremely small constellation size. Clearly, for constellations larger than this the NF must drop off quickly to maintain the BER; as overall noise figures lower than 10 dB are difficult to achieve, it is impractical to depend heavily on reducing the noise in the analog front-end; instead, dependence on digital coding to gain immunity to noise is a much simpler and more effective solution, a solution not available to classical analog communication systems.

Filters

For the transmitter, very little analog filtering is required, since the baseband transmit filter is more easily implemented in the digital domain; the signal is already bandlimited at the baseband before modulation into the passband. The only analog filter that is desirable is at the output of the power amplifier, where it is used to weakly bandlimit the signal to eliminate spurious frequencies from the oscillator. For the receiver, however, an anti-imaging filter is needed to eliminate all of the undesirable frequency bands before demodulation.

Several options exist for implementing this passband anti-imaging filter. The conventional method is to utilize a high-Q off-chip filter to perform the necessary filtering, such as tuned L-C tank circuits, ceramic resonators, or surface-acoustic wave (SAW) filters. There are several advantages and disadvantages to each of these.

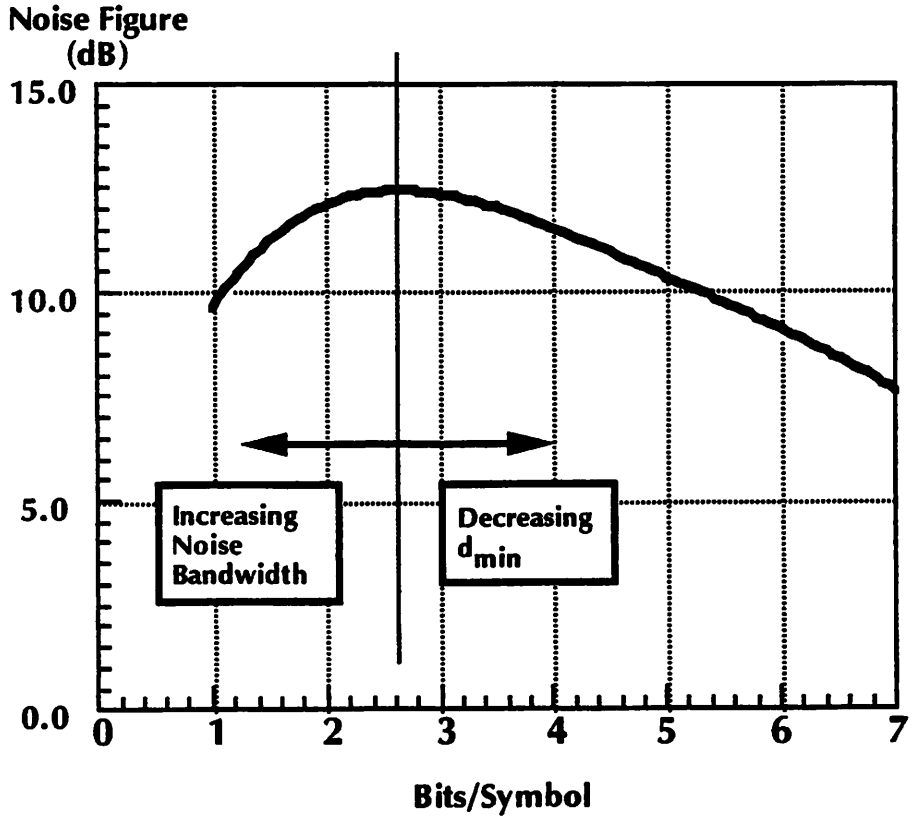

Figure 4.4: Noise Figure vs. Constellation Size

L-C bandpass circuits, or equivalent stripline filters, are relatively simple to design and use, and are effective well into the gigahertz band. Also, by using a varactor diode as a voltage-controlled capacitor, these filters can thus have tunable center frequencies, which is useful if precision narrowband filtering is needed. Unfortunately, the achievable Q of L-C filters is limited by the parasitic resistances, and is thus limited to values between 1 and 100. There exist low-cost, high-Q solutions, however, which use either mechanical or electrical resonance to perform the filtering, and have been used extensively in today's analog cellular telephony systems; unfortunately, these are filters are not voltage-controllable. There are other limitations as well - the mechanical resonance filters, such as ceramic resonators, quartz crystals, or SAW filters, all depend on material vibrational modes, and hence are not easy to fabricate for operation well above 1 GHz. On the other hand, the electromagnetic resonators, such as suspended strip-line filters or combline resonators, utilize waveguide structures to achieve accurate filtering and are effective at frequencies as high as 20 GHz. For example, one commercially available combline filter achieves a fifth-order response over a passband of 2.2 to 2.3 GHz, with the -50 dB rejection points at 1.95 and 2.5 GHz. The major disadvantage of electromagnetic resonance filters is sheer physical size; for example, the filter cited above is10cm x 2.5cm x 1.5cm, which is far too large to be used in a small portable terminal, especially for only a single filter stage.

Since the operation frequencies are in the gigahertz band, another filtering option - the use of on-chip L-C filters - is open to the hardware designer, since long aluminum traces on the chip die begin to exhibit transmission line effects and stripline behaviour. By literally drawing a microstrip L-C circuit out of an aluminum trace, a usable inductor is thus created. Several successful attempts to use such on-chip tank circuits have been made; in the bandpass filter circuit developed by Meyer and Nguyen [30], an effective inductance of 9 nH, resulting in a filter Q of 10, was achieved at 4.0 GHz, with the entire circuit fabricated completely on-chip. At these frequencies, the Q was limited primarily by the 100Ω parasitic resistance of the trace; substrate loss was found to be negligible. This is certainly comparable to L-C filters implemented with off-chip discrete components or striplines, and also possesses the

same advantage of being voltage-tunable by using an on-chip varactor diode. The disadvantage to this method, of course, is the large chip area required by these inductors; by drawing the trace as an aluminum spiral, the area consumed can be minimized.

Of course, the filter performance depends heavily upon the configuration chosen. For example, if a heterodyne scheme were chosen for use as the PCS transceiver, an IF of 200 MHz is quite suitable. Hence, the passband RF filter needs only provide filtering over 200 MHz at 1.5 GHz, for a filter Q of 15, and need not be extremely high-order, since the 200 MHz IF is much larger than the 40 MHz required for the spread-spectrum digital signal. On the other hand, if a homodyne scheme with all of the filtering at the passband, including the anti-aliasing, then the filter must achieve accurately extract the 40 MHz band, and only that band, at the RF. This requires not only a Q of 75, but also sharp band-edge cutoffs, implying a high-order filter response. We will see below that this severe case is indeed is exactly what is implied by sampling demodulation, and there are digital techniques by which this filtering requirement can be minimized.

Oscillators

Almost invariably, the design configuration of choice for the frequency synthesizer is a indirect crystal-based PLL synthesizer, since it uses a negative-feedback loop to lock onto an accurate low-frequency reference. The frequency reference is usually a low-cost crystal oscillator, well known for their temperature stability, low phase noise, and ease of implementation. A crystal-based PLL synthesizer is shown in figure 4.5, using a Pierce oscillator for the low-frequency reference; in such a configuration, the output at frequency $k \cdot f_{xtal}$, where k is the frequency divider ratio. Due to the tightly coupled feedback loop in the PLL, the synthesized frequency is guaranteed to be extremely close to $k \cdot f_{xtal}$, and inherits the frequency stability of the crystal reference. Of importance is the fact that the oscillation frequency is digitally programmable, since the divide-by-k operation is performed using a digital counter [21, 31]. This method of synthesizing accurate multiples of a baseband reference will be taken advantage of below, when sampling demodulation is discussed.

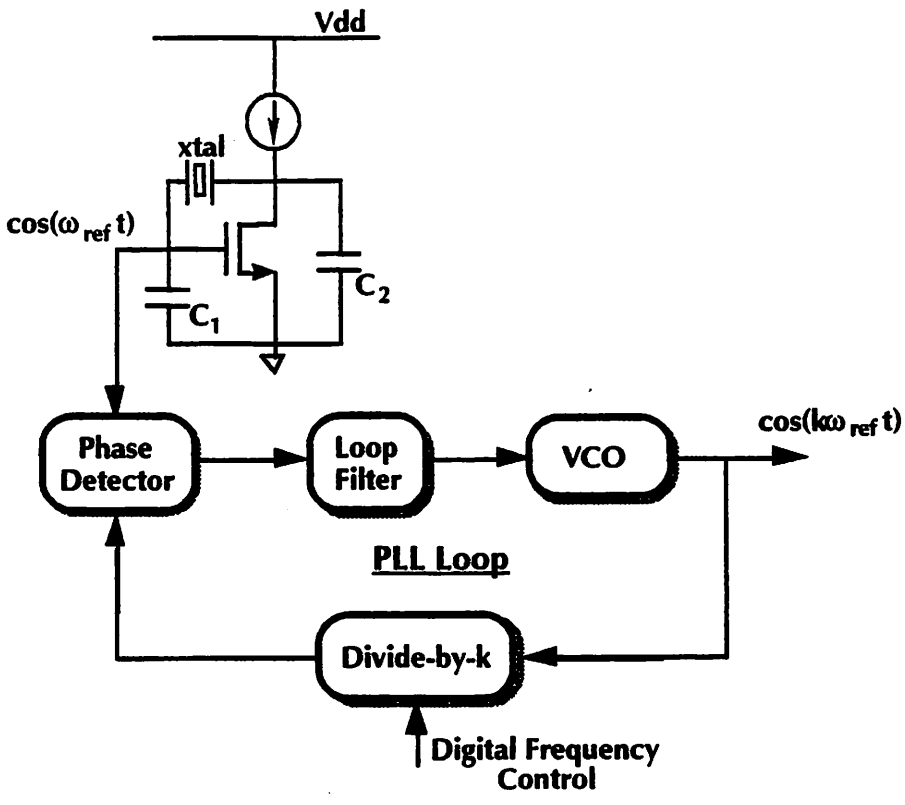

Figure 4.5: Crystal-based PLL Synthesizer

The two performance specifications of an oscillator are the phase noise, and the output power. Since the circuit is assumed to be completely integrated, the oscillator needs only drive the input to the mixer stage, and hence power output levels on the order of 10 dBm are tolerable.[1] The effects of phase noise on the system performance are unfortunately not clear. Since the oscillator output with phase noise is $\cos(\omega_c t+\Delta(t))$, where $\Delta(t)$ is the phase noise, one reasonable model is that the transmitted baseband signal is actually $A_k \exp(j\Delta(t))$, where A_k is the actual symbol. For $\Delta(t)$ small, this behaves as additive noise in the signal, and can be treated as such. Apparently, low noise oscillators can achieve a RMS phase noise as small as 1ppm, defined as the standard deviation of the oscillator period divided by the mean, and for all of the analyses the author has seen is neglected relative to the thermal

1. Calculated as the power for 0.7V across a 50Ω load. This is not a severe performance requirement; 10 dBm is achieved by many circuits available today [32].

noise at the receiver. Although the specification of "as low as possible" seems to suffice, more quantitative work needs to be done.

For the VCO itself, an interesting implementation has been described in [33], which utilizes a coupled pair of L-C oscillators, each tuned to a different frequency, and a Gilbert quad to set the overall output frequency to somewhere between the resonant frequencies of the L-C tanks; the measured performance yielded a tuning range of 200 MHz, with a maximum frequency of 1.8 GHz. However, this VCO utilized only on-chip inductors to achieve the tuning; no external components were needed. Thus, even the synthesizer-oscillator circuit can be almost completely integrated, save the single external crystal reference.

Sampling Demodulation

Before continuing with the discussion of analog performance requirements, the differences between the transmitter and receiver must be considered. Referring back to the block diagrams of figure 4.3, the transmitter only needs to do is to convert a signal to the baseband to the passband; it does not have to contend with degraded signal levels or noise, unlike the receiver. Likewise, the receiver is required to track frequency and phase changes, and attempting to recover the data from a signal corrupted by channel noise and multipath distortion. Thus, the receiver bears most of the difficult design issues; from here on, minimizing the required receiver performance is paramount.

Thus far, three of the five components of the analog transceiver has been discussed. The modulation mixer on the transmitter side is relatively simple; a Gilbert quad or a wideband single-device mixer can be used. However, the remaining two components, the receiver mixer and the sample-and-hold/analog-to-digital converter, will be discussed together, since the two units can be merged, eliminating the explicit downconversion step provided by the mixer and simplifying the circuit considerably. Indeed, the name "sampling demodulation"

is appropriate, since it uses the fact that sampling itself can be viewed as a form of modulation.

When considering demodulation, one very useful method of analysis is to examine the frequency domain representation of the signals. Essentially, mixing in the time domain is equivalent to convolution in the frequency domain; since the carrier signal and the local oscillator are assumed to be synchronized, copies of the original signal spectrum are placed down at baseband and twice the carrier frequency by the mixer. The resulting signal is then low-pass filtered and sampled at Nyquist rates. The important fact is that the final spectrum has aliases at *all* multiples of the sampling frequency ω_{samp}, given by:

$$\sum_{i=-\infty}^{\infty} F(\omega - i\omega_{samp})$$

where $F(\omega)$ is the original baseband spectrum and

$$\frac{1}{2}\left(F(\omega - \omega_{carrier}) + F(\omega + \omega_{carrier})\right)$$

is the received modulated signal. Any demodulator/A-D structure must end up with this as its final spectrum; however, mixing followed by sampling need not be the only solution for achieving this.

Suppose that the received signal is bandpass sampled at the Nyquist rate of the baseband signal, which is well below the Nyquist rate for the incoming RF signal. After sampling, the spectrum of the signal has a transform equal to:

$$\frac{1}{2}\sum_{i=-\infty}^{\infty}\left(F(\omega - \omega_{carrier} - i\omega_{samp}) + F(\omega + \omega_{carrier} - i\omega_{samp})\right)$$

Making the important stipulation that $\omega_{carrier}=k\cdot\omega_{samp}$, where k is an integer, this sum can be simplified to:

$$\sum_{i=-\infty}^{\infty} F(\omega - i\omega_{samp})$$

which is precisely the same result achieved by the original demodulation structure. The entire demodulation step has been reduced to a single sampling operation, sampled at the Nyquist rate for the *baseband* signal. A graphical depiction of this process is shown in figure 4.6; the path indicated by the solid black arrows shows the result of each step of a conventional mixer-sampler structure, and the single path indicated by the large gray arrow is the

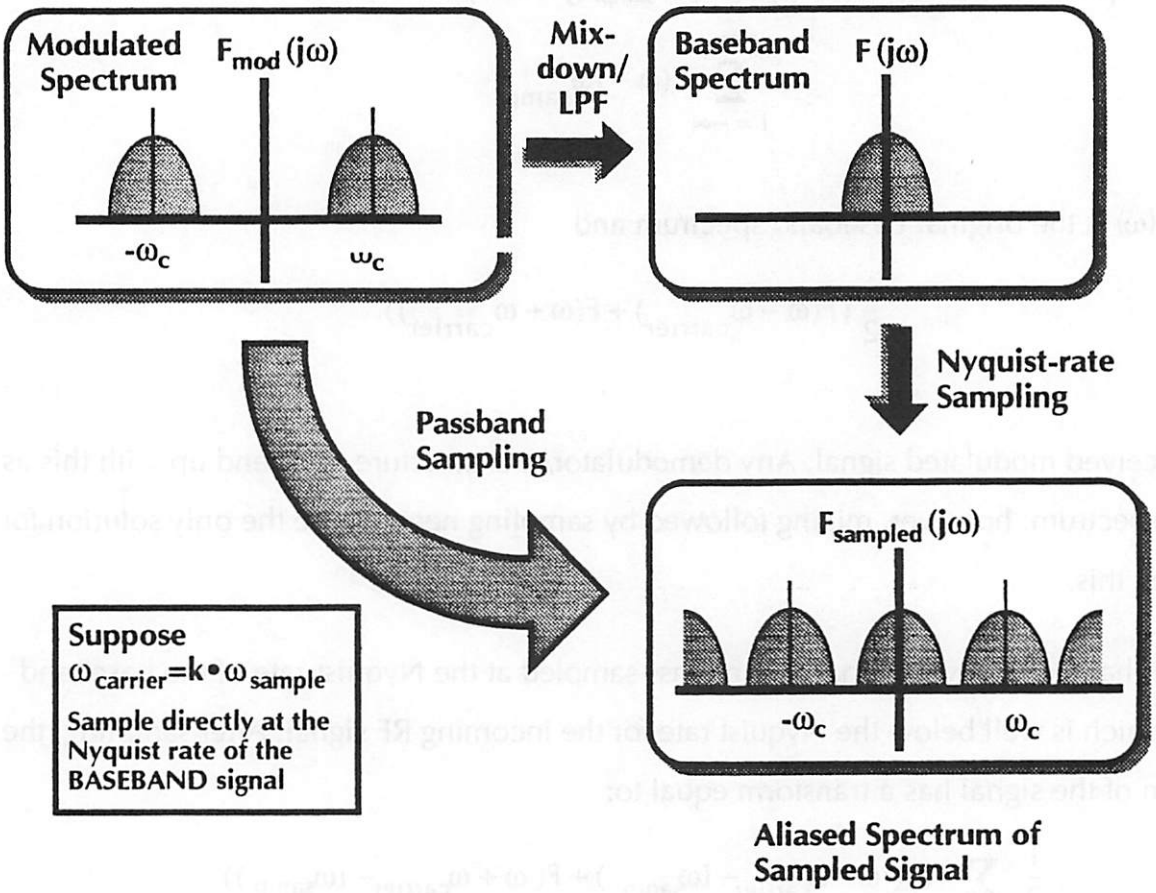

Figure 4.6: Sampling Demodulation

result of bandpass sampling the modulated signal. Clearly, both will yield the same aliased spectrum provided $\omega_{carrier}=k\cdot\omega_{samp}$.

Another interpretation of this result can be seen as follows. The frequency-domain equivalent of sampling is the convolution by a stream of Dirac delta functions, i.e. the effect of sampling the modulated signal with transform $F_{mod}(\omega)$ yields a signal with a Fourier transform equal to:

$$\left(\frac{\omega_{samp}}{2\pi}\right)\int_{-\infty}^{\infty}\left\{F_{mod}(\omega-\Omega)\sum_{i=-\infty}^{\infty}\delta(\Omega-i\omega_{samp})\right\}d\Omega$$

Alternatively, this is the same as multiplication in the time domain of the received signal by a sum of equal-power cosines at all harmonics of the sample rate:

$$f_{mod}(t)\cdot\sum_{i=0}^{\infty}\cos(i\omega_{samp}t)$$

Provided proper anti-alias filtering is done to isolate the modulated signal at $\omega_{carrier} = k\cdot\omega_{samp}$, the required demodulation-and-sampling is thus yielded. Essentially, the mixer has been shown to be redundant, since the sampling operation itself can easily perform the same mixing function.

Of importance is the fact that the sampling rate does *not* change frequencies, even as the carrier changes. The k in the above stipulation was not specified; so long as the carrier frequency is an integer multiple of the baseband sampling rate, this method works. Hence, a voltage-controlled oscillator is not needed here, and a high accuracy crystal oscillator can be used instead. The hardware cost is minimal: an accurate switch, implementable using standard MOS technologies, and a fixed-frequency crystal oscillator, which is the same oscillator as used to reference the transmitter synthesizer PLL. In terms of power, the demodulator is passive, dissipating energy only in the crystal oscillator and the resistive channel of the MOS device, and is certainly much lower than an equivalent full mixer/oscillator configuration.

Theoretically, the sampling demodulator is an excellent implementation for the demodulator in terms of hardware and power, utilizing an active device as a simple passive switch. However, before any judgment of real usefulness is passed, its practicality for use in a transceiver for personal communications must be assessed.

Quadrature Recovery

Since the sampling introduces mixing with equal-power cosines, the in-phase signal is easily recovered, as in the above derivation. However, the quadrature phase requires a sine wave in the mixing operation to successfully recover the information. Clearly, some more work is needed than just the simple sampling step, since the quadrature data will be lost. One immediate method is to utilize an additional sampling operation, shifted in phase by $\pi/2$. By doing so, the time-domain representation of this additional sampling operation is:

$$f_{received}(t) \cdot \left\{ \left\{ \sum_{i=0}^{\infty} (-1)^i \cos(2i\omega_{samp}t) \right\} + \left\{ \sum_{i=0}^{\infty} (-1)^i \sin((2i+1)\omega_{samp}t) \right\} \right\}$$

The proper mixing is yielded if the condition is made that $\omega_{carrier} = k \cdot \omega_{samp}$, and k is odd. Otherwise, the second sampling step simply yields the in-phase signal again. Thus, use of such a system with quadrature modulation requires only a second sampling unit offset by a simple 90° phase shift.

Incoherent Demodulation

To ease the hardware requirements, the necessity of zero phase and frequency offsets must be carefully considered. If possible, carrier recovery should be avoided at all costs, which leads to the requirement that very accurate frequency references are needed in both the transmitter and receiver, and immunity to nonzero phase offsets.

First, the assumption that $\omega_{carrier} = k \cdot \omega_{samp}$ must be examined. Surprisingly, this is not a difficult requirement for the hardware; as discussed above, a digital PLL synthesizer automatically achieves this condition, and does so with extreme accuracy when referenced to a

crystal oscillator. Second, although the requirement of zero frequency offset has been met since, the requirement of zero phase offset still needs to be addressed. The simplest solution is to utilize a QAM scheme that does not require absolute phase data, such as a differential PSK scheme, which encodes the transmitted information on the transition between constellation points, instead of the constellation points themselves. Provided that the received phase is constant or changes slowly with respect to the data rate, absolute phase information is not needed when decoding the incoming symbols. Of course, there is a penalty for doing so; the cost of differential encoding is a 3-dB penalty in BER for the same SNR as compared to the nondifferential scheme, which must be made up in further coding. Several systems, such as the new US digital cellular standard, utilize a DPSK scheme. Again, this is an example of how a digital encoding at baseband allows the elimination of costly analog hardware; removal of the absolute phase dependence requirement is achieved with only a minor expense in digital processing, and no expense in bandwidth.

At this point, it wa fruitful to utilize simulations to verify the theory. The block diagram in figure 4.7 was simulated using CAPSIM, incorporating the concepts of sampling demodulation, $\pi/2$-time-shift recovery of the quadrature phase, and DPSK-assisted incoherent reception. To model the incoherency, a random phase offset was incorporated into the sampling modules (a complete listing of the models used can be found in the appendix), and as in chapter 2, root-raised-cosine filter responses were used for the transmit and receive filter characteristics.

The simulated vector diagram for the baseband modulated signal, at the output of the transmit filter block, is shown in figure 4.8. Similar to the eye diagram, the vector diagram plots the real and imaginary parts of the complex-valued baseband waveform as a parametric function of time. The constellation points are evident, with constant amplitude and phases $\{\pi/4, 3\pi/4, -\pi/4, -3\pi/4\}$; shown in figure 4.9 is the magnitude of the baseband signal alone, and the symbol amplitude threshold is marked. It is against these two figures that the results will be compared.

In figure 4.10, the vector and magnitude diagrams of the signal emerging from the sampling demodulator are shown. The random phase offset introduced is evident, along with the effects of the nonzero phase response of the filter; the constellation points have been rotated by over $\pi/4$, and the originally-square configuration has been elongated. However, the magnitude plot still clearly shows that the only change has been in phase, since the threshold crossover points have remained essentially stationary. For the DQPSK decoder, a simple hard-decode scheme was used - decimation of the signal back to the 31 MHz symbol rate, and then decide the phase transitions. In figure 4.11, the results of decimation are shown; the altered symbol constellation is clear, and more importantly the recovered amplitude is essentially constant, as it should be. From this, the input data was successfully recovered, with no errors in the bit-stream from the random data generator to the output.

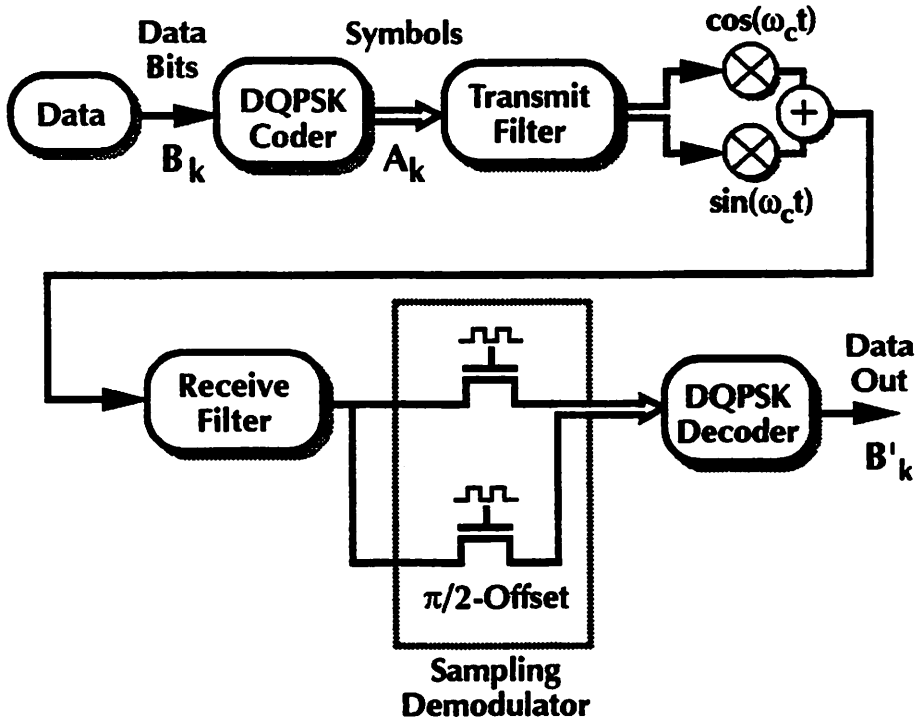

Figure 4.7: Block Diagram for DQPSK Simulation

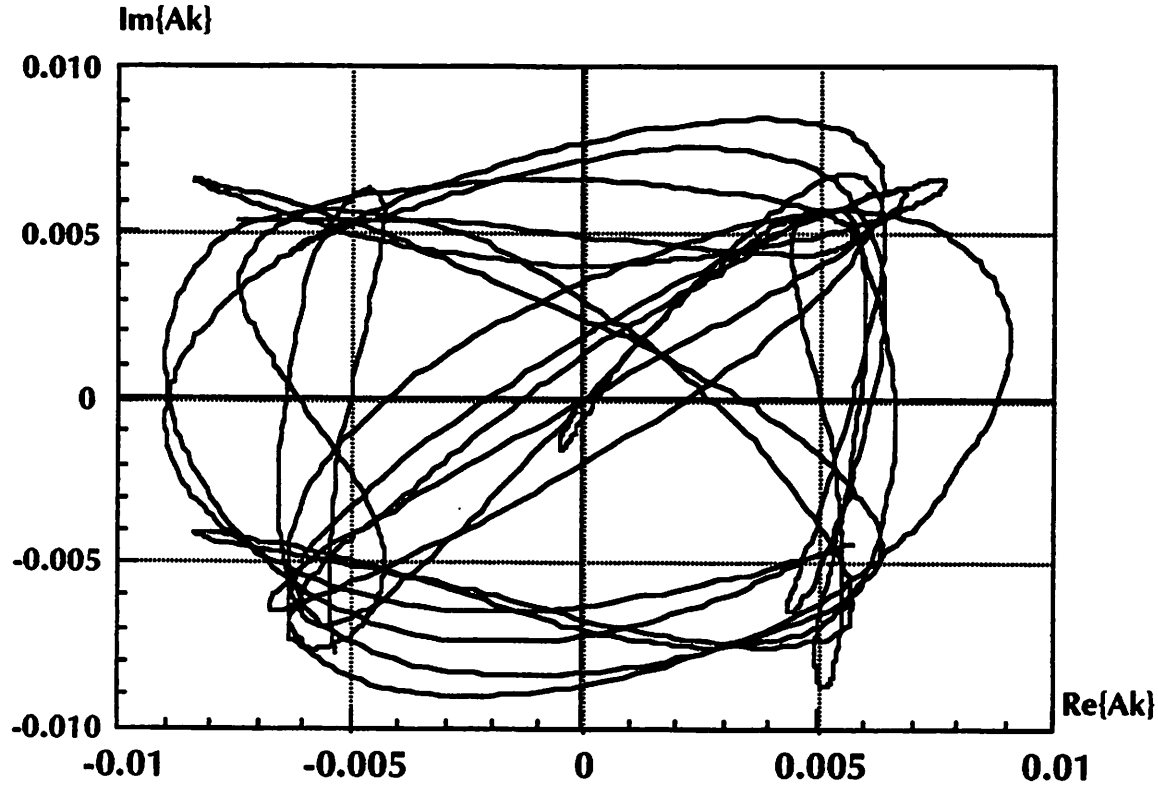

Figure 4.8: Baseband Vector Diagram, DQPSK Signal

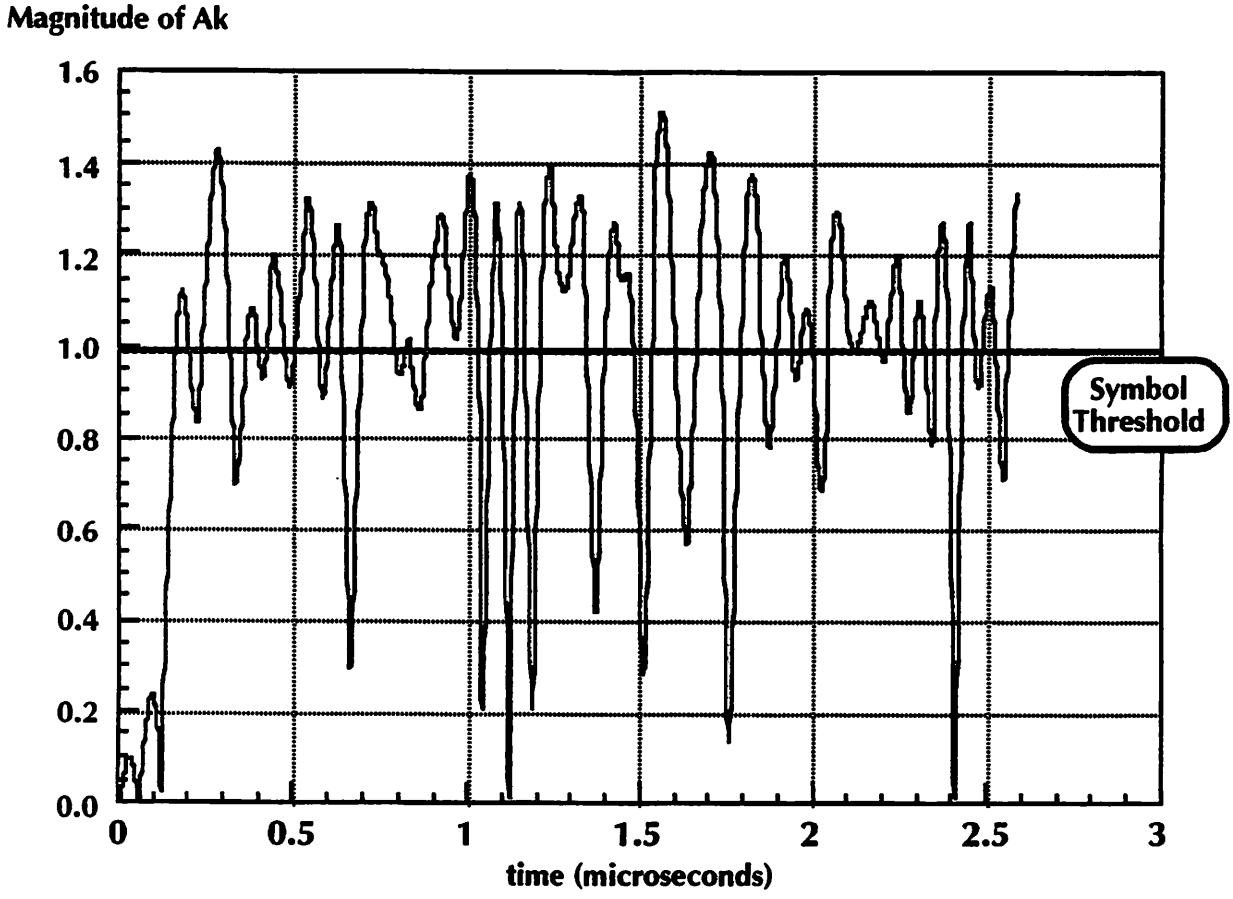

Figure 4.9: Magnitude of Modulated DQPSK Baseband Signal

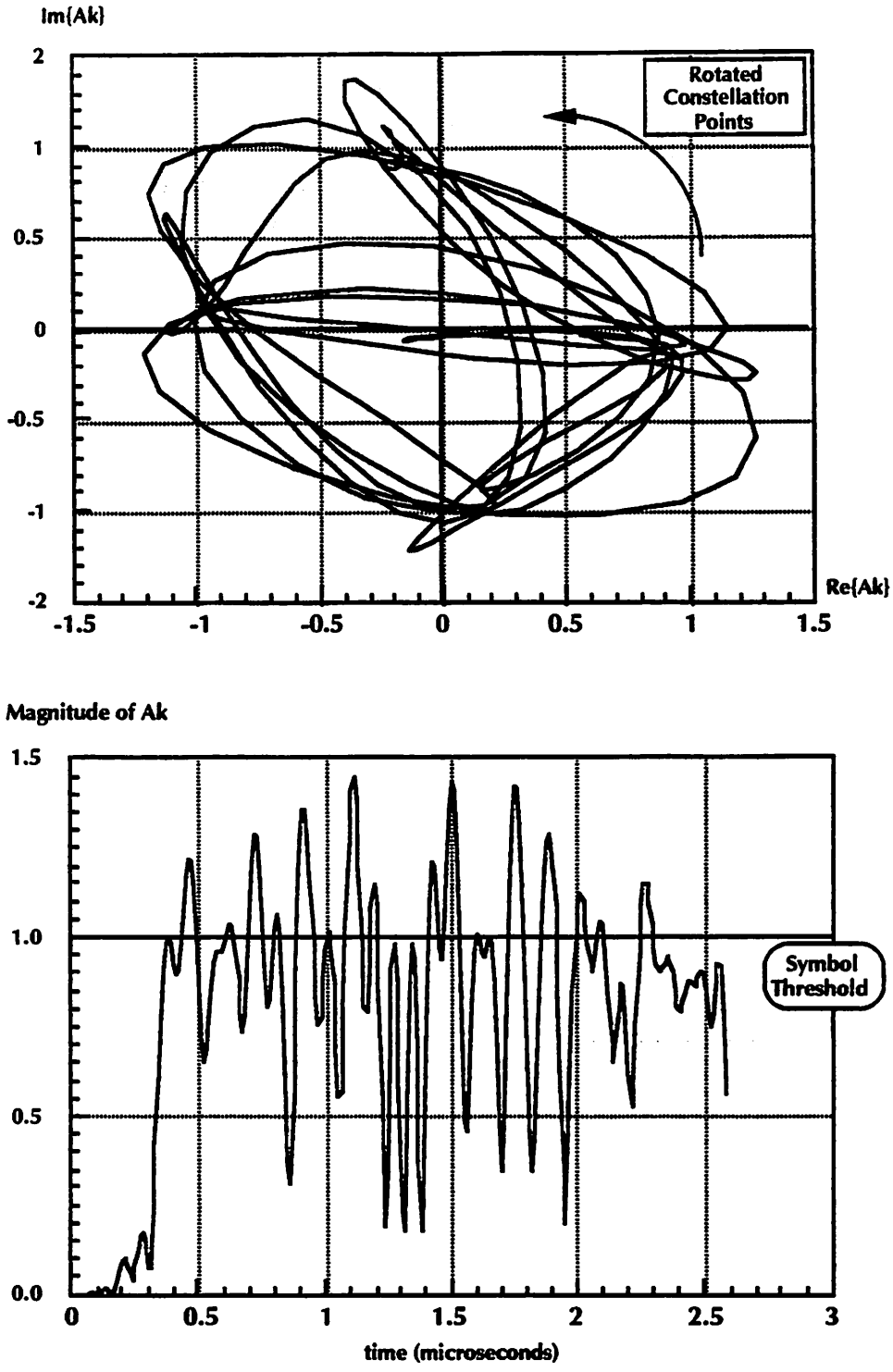

Figure 4.10: DQPSK Vector and Magnitude Plots after demodulation

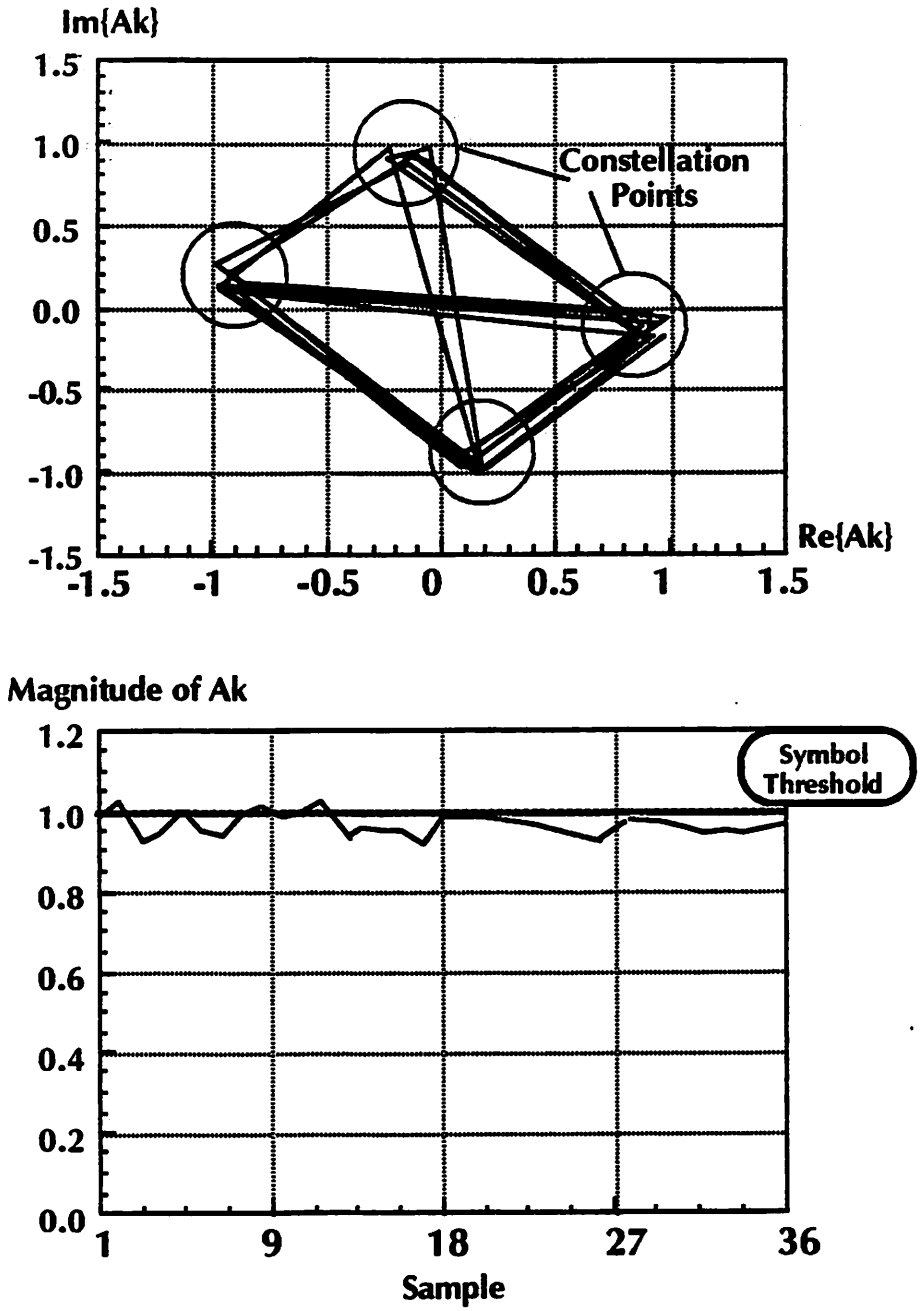

Figure 4.11: DQPSK Vector and Magnitude Plots after decimation

Rejection Filter Requirements

Given that a sampling demodulator is employed, anti-alias filtering must obviously be performed before the sampling step, which places such filtering in the passband instead of the baseband, making the task significantly more difficult. If homodyne demodulation is anticipated, this leads to some extremely high rejection requirements; for a spread bandwidth of 40 MHz, and an RF carrier at 1500 MHz, this requires a filter Q of at least 40, which is difficult to achieve without resorting to large external inductors. Active filtering is also difficult at high frequencies, since these usually employ op-amps in feedback; both pole placement and stability of the feedback loop become critical. Furthermore, since one specific frequency band must be selected, the filters must be implemented as a voltage-controlled component, tracking the changes in the RF carrier frequency as the transceiver moves from cell to cell. If the Q requirements are high, then the tuning accuracy of the bandpass filter is correspondingly high, again complicating the design. Likewise, since the filtering must be extremely tight, the band-edge rolloff, and hence the order of the filter, must be correspondingly high.

The important issue then is to reduce the filter rejection requirements, by permitting the use of lower Q and lower-order response filters. Traditionally, this has been accomplished by simply utilizing larger guard bands between the active frequency bands. However, this is wasteful of precious spectrum, and should be avoided. A more elegant means of reducing the rejection filter requirements presents itself, by simply utilizing the orthogonality of spread-spectrum signals. If the filter Q is lowered to the point that significant aliasing does occur if wide guard bands are not used, then the aliasing can be thought of as cochannel interference, since multiple users now appear to have been transmitting in the same frequency band at the same time. However, this is precisely the situation envisaged by CDMA. If two bandlimited signals are allowed to alias, but are also required to be orthogonal, then signal recovery can be accomplished by a simple correlation.

Using a QPSK signal with an M-sequence spectrum spread, the block diagram in figure 4.12 has been simulated, with the two equal-power signals modulated at carriers of 1500 MHz and 1600 MHz (chosen for simplicity). The user data rate was assumed to be 2 MBps, resulting in a 31 MBaud symbol rate after spreading, with a demodulation sampling rate set at 100 MHz. A raised-cosine transmit filter was used, and the receive filter was assumed to possess a low-Q, low-order passband response; this was modeled as having no receive filter whatsoever, and hence no anti-aliasing between the two frequency bands. Synchronization of the spreader was done manually and the correct correlation point was programmed

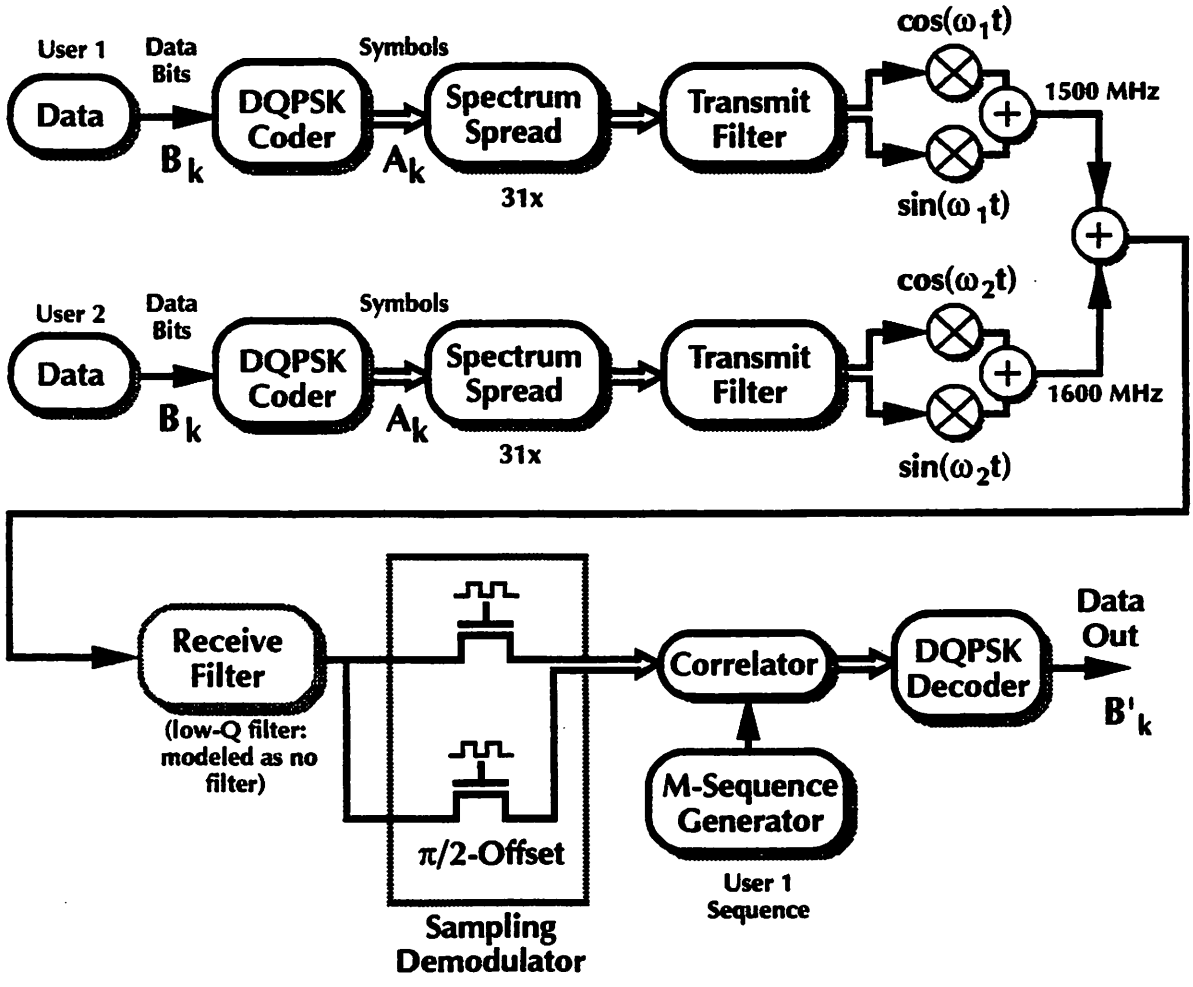

Figure 4.12: Block Diagram for CDMA Simulation

into the correlator, since the number of simulated bits required to achieve satisfactory simulation was prohibitively large.

In figure 4.13 and 4.14, the vector and magnitude plots of the signal after the sampling demodulation block are shown. Indeed, there has been significant aliasing distortion between the two users, in comparison to the single user situation simulated above, especially in the magnitude plot in figure 4.14. Surprisingly, the vector plot degenerated from a square constellation pattern into the eccentric ellipsoids seen in figure 4.13; one partial explanation for this is that the two users had not phase offset and were thus unintentionally synchronized. The most likely explanation lies in the nature of the spreading code. Since there are 31 spread symbols transmitted per user symbol $\{A_k\}$, and the spreading code is an antipodal $\{\pm 1\}$ sequence, the majority of the transitions will lie along $\{\pm A_k\}$, and only a few transitions actually between constellation points, thus resulting in the degeneration.

After correlation with the spreading sequence shown in figure 4.15, the output of the decoder yielded the data sequence {0 0 1 0 0 0 0 1 1 0 1 0 0 0 0 1 1 1}. The original transmitted sequence was {1 0 0 0 0 1 1 0 1 0 0 0 0 1 1 1 0 1}. These two sequences are identical up to the delay through the transmit filter, indicating that the individual user data has been successfully recovered from the aliased signal.

Thus, the filter requirements need not be as high; so long as the codes utilized in adjacent bands are uncorrelated, or only weakly correlated, then aliasing can be allowed with no loss in data to the user. Of course, the penalty here is increased processing gain requirement from the spread-spectrum coder; for a spread factor of 31, in chapter 3 it was found that 21 separate users can be effectively multiplexed. This implies that the receiver can fully alias approximately 3 bands and successfully recover the signals from all users, by simply correlating with the proper spreading code.This relaxes the required filter Q from 75 down to approximately 20, and can tolerate lower-order bandedges.

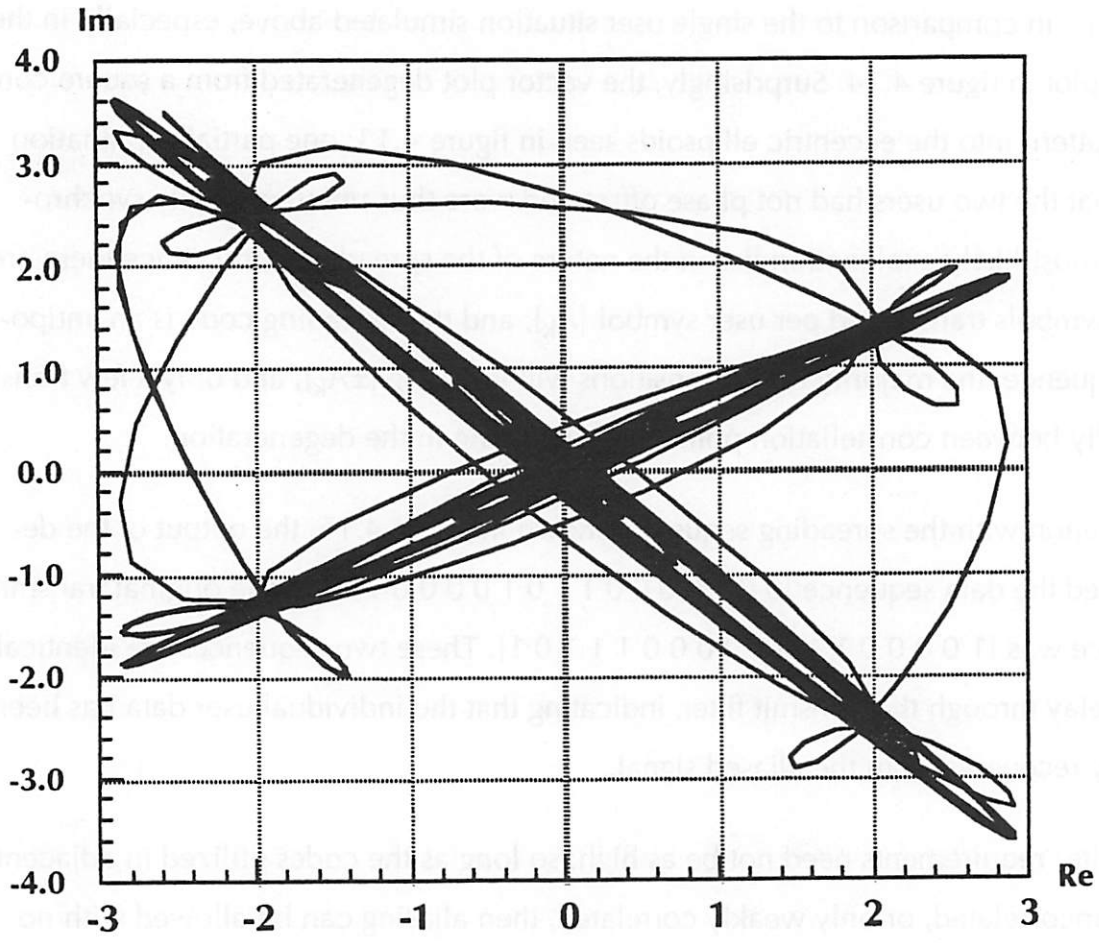

Figure 4.13: Vector Diagram of 2 aliased CDMA signals

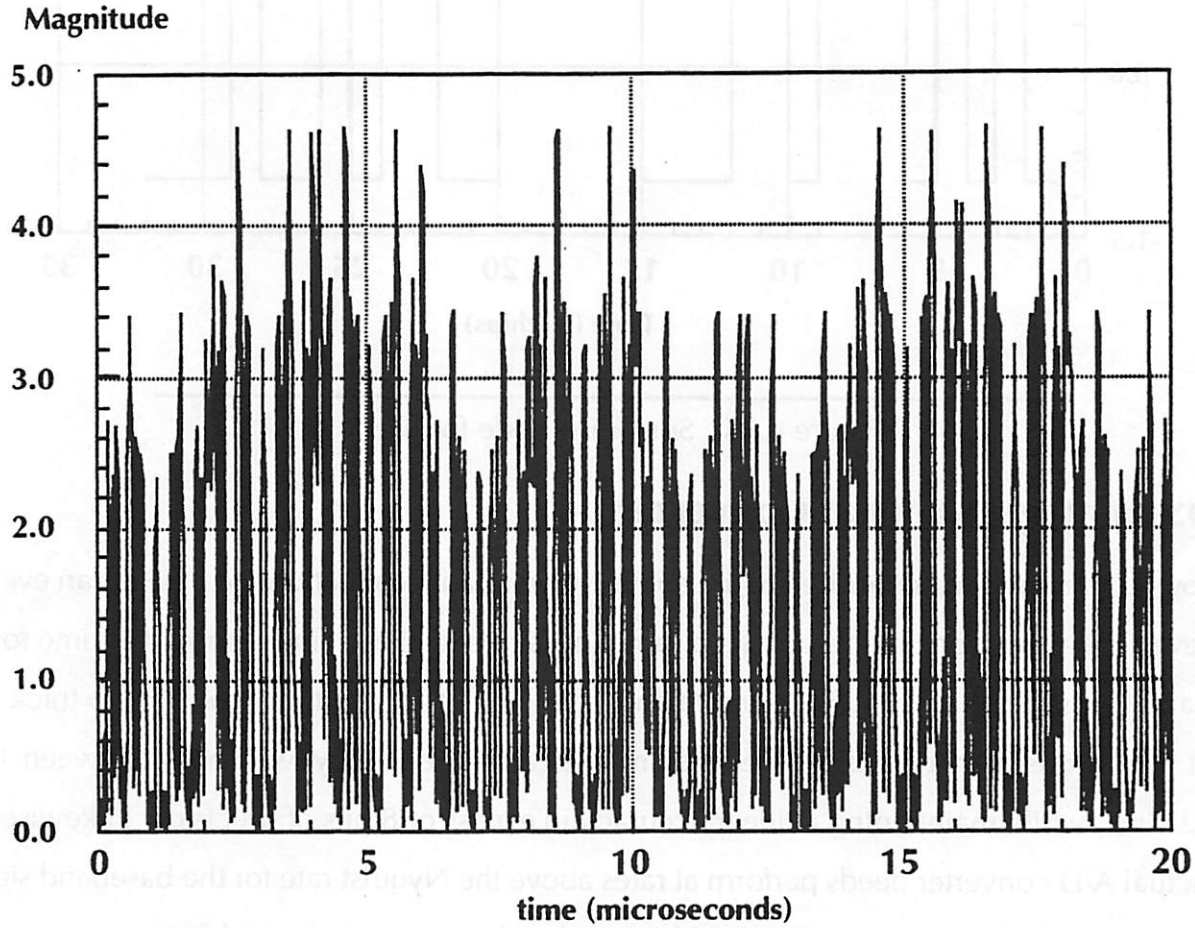

Figure 4.14: Magnitude Plot of 2 aliased CDMA signals

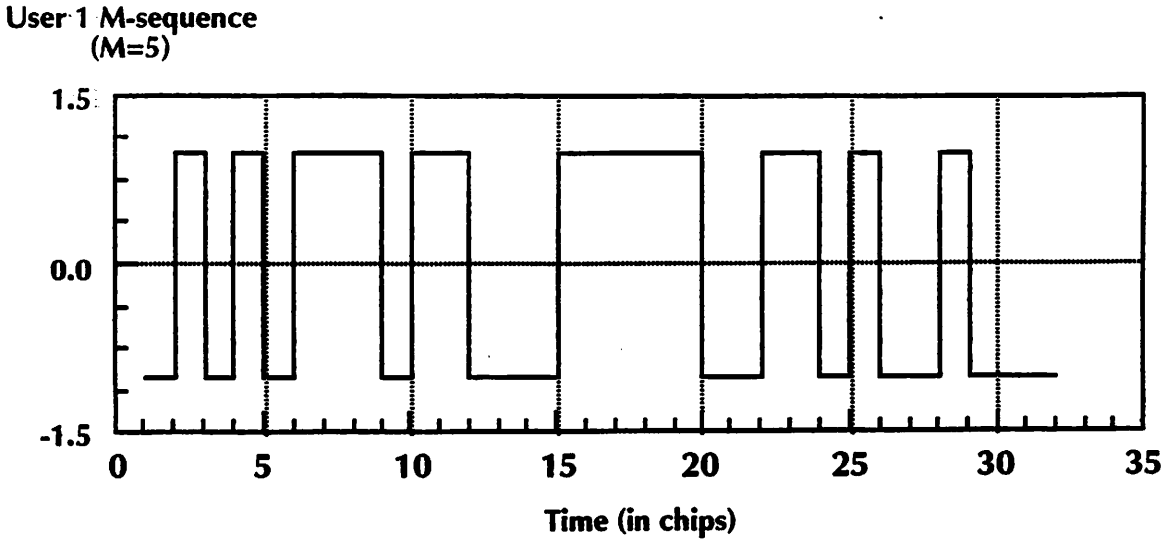

Figure 4.15: Spreading code for user 1

MOS Analog-to-Digital Converters

Of course, the sampling described in the above theory is an idealization; no device can ever achieve "delta-function" precision in the sampling, since it takes a finite amount of time for the sampling switch to close. The sample-and-hold (S/H) itself must have be able to track input waveforms with a bandwidth exceeding the carrier frequency, which lies between 1 and 2 GHz, while maintaining a linear accuracy in excess of 8 bits of precision[2]; likewise, the actual A/D converter needs perform at rates above the Nyquist rate for the baseband signal, 65 Msamples/sec for a 2 MBps DQPSK signal with a spectrum spread factor of 31.

For a silicon MOS sample-and-hold, there are several significant nonidealities which must be considered. In figure 4.16, a basic MOS S/H is displayed, along with a plot of a signal to be sampled and the output of the S/H. Essentially, the MOS device is a simple on-off switch, passing charge to the sampling capacitor when in track mode, and preserving the charge when in hold mode.

[2] Apparently quantization noise becomes a serious issue for lower precision; as far as the author is aware, however, no quantitative work has been done to date.

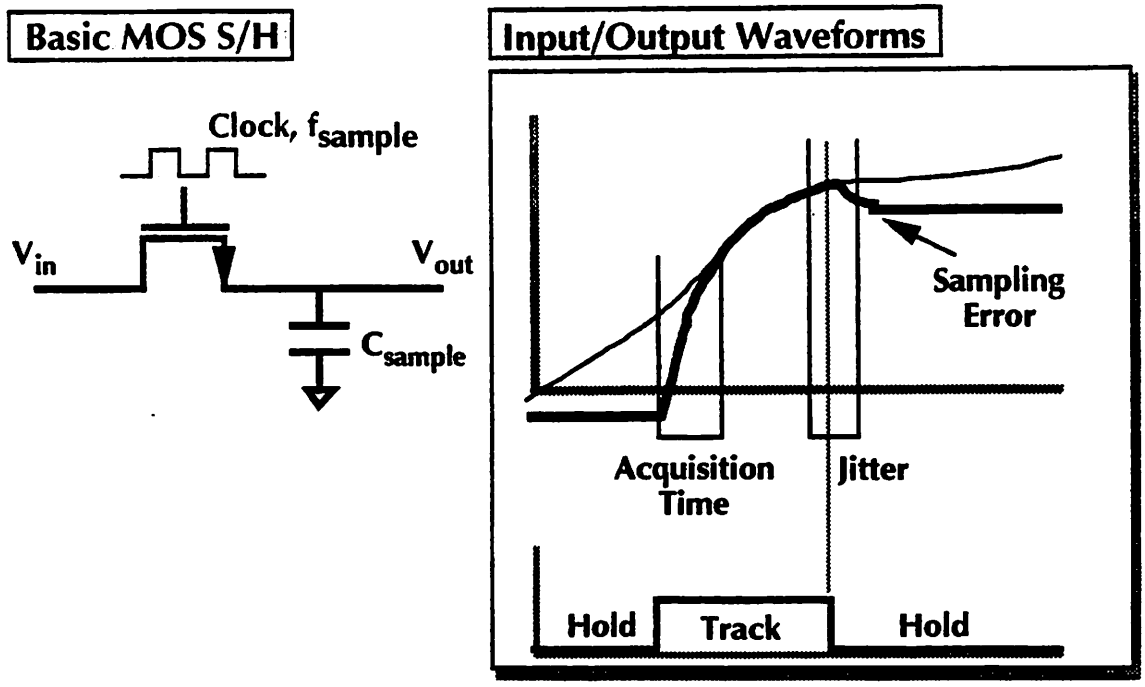

Figure 4.16: MOS Sample-and-Hold Characteristics

First, when the S/H is in track mode, the switch is closed and the voltage across the sampling capacitor should ideally equal the input voltage. However, the drain-source resistance R_{ds} of the MOS device is nonzero, equal to:

$$R_{ds} = \frac{1}{\mu C_{ox} \left(\frac{W}{L}\right) \left(V_{gs} - V_t\right)}$$

where the device parameters are the same as described in the discussion of f_t above. The effect of r_{ds} is to create an R-C lowpass filter with C_{sample}, hence limiting the bandwidth that the system can track accurately. Likewise, the acquisition time t_{acq} needed to switch from hold to track mode is determined by this lowpass filter, and thus t_{acq}=(1/Track BW)= $R_{ds} \bullet C$.

Second, as the switch moves from track mode to hold mode, the charge stored in the channel of the MOS device must be removed, and hence is injected directly onto the sampling capacitor. This results in a sampling error ΔV, which is equal to:

$$\Delta V = \frac{Q_{chan}}{C_{sample}} = \frac{WLC_{ox}(V_{gs} - V_t)}{C_{sample}}$$

The above yields the classical figure-of-merit for a MOS device[34], known as the speed-accuracy product, equal to $\Delta V \bullet t_{acq}$, and can be simplified to L^2/μ. Since minimizing this is the key goal, and like f_t is proportional to the square of the channel length, as devices scale the performance of MOS devices improves dramatically. For 8-bit accuracy on a 1 V swing, a track bandwidth of 3 GHz, and μ =400 V/(cm^2 -sec), a minimum drawn L of 0.2 microns is required.

Lastly, the phase noise of the crystal oscillator must be considered. As in the above discussion of the transmitter VCO, the jitter can be modeled as a phase error in the recovered signal. However, a new consideration arises, since the S/H must accurately sample a 1.5 GHz waveform at 60 MHz. Thus, the jitter window shown in figure 4.6 must be sufficiently small to capture the 1.5 GHz waveform. As a first-order analysis, supposing that the signal to be sampled is given by $\cos(2\pi \bullet 1.5 \bullet 10^9 t)$, sampling at 60 MHz with zero phase difference should yield a constant value of 1. Assuming a SNR due to jitter of 60 dB, this implies that the actual sampling point must be within ±4.5 psec of the ideal sampling point. Setting this equal to 3 standard deviations of the oscillation period, for a 65 MHz oscillator this yields a phase noise of 90 ppm.

In a recent thesis, it was reported that there is apparently no limit to the performance of MOS S/H switches using differential bottom-plate sampling techniques [35], save that of the maximum velocity of electrons in silicon. Whether or not these requirements can be met remains to be seen; however, sampling oscilloscope inputs are today already capable of sensitivity as low as 1 mV, with 8-bits of accuracy at sampling rates of 4 GHz [36]. Although these are currently implemented using GaAs MESFET's, Si MOSFET's are not far behind in performance, with an 8-bit 500 MHz Si flash converter being recently reported [37]. Thus, the required switching speeds and accuracy are evidently possible, and will be realizable as MOS devices continue to scale into the submicron range.

A New Transceiver Structure

Summarizing the results of the previous sections, the performance requirements performance of the new transceiver structure is shown in figure 4.17. The entire transceiver has been simplified back into the block diagram of figure 4.3a, the basic homodyne demodulator, with one additional simplification: the mixer and VCO downconverter has been eliminated, having been merged into the sample-and-hold through the use of sampling demodulation. Certainly, some of these performance requirements, such as the noise figure of the front-end amplifier and the sampling accuracy of the MOS device, are not trivial to meet. However, several key points need to be stressed about this system which justify these performance requirements:

- The transceiver utilizes a homodyne recovery scheme - no intermediate frequency is needed, thus eliminating the need for additional/replicated hardware for processing at the IF bands.

- The transceiver is intended to be fully integrated, except for a single external timing crystal. For portable applications, this is of extreme importance, since this minimizes the power consumed in driving large off-chip parasitics such as striplines and board traces. Likewise, the area consumed by a single chip is negligible compared to a module containing many individually packaged components, thus making the transceiver both small and lightweight.The use of on-chip filtering is of particular importance, since on-chip stripline filters have not been exploited previously.

- The transceiver explicitly does not use gallium arsenide anywhere in its design; the performance of state-of-the-art silicon technologies are comparable to that of GaAs, and hence should be utilized to its fullest extent.

Component	Specification	Comments
Transmitter		
D/A Converter	8-Bits accuracy, > 60 MHz conversion rate.	Power consumption may be severe at these rates.
Oscillator	Phase noise < 50 ppm Output Power: ~+10 dBm@carrier	PLL-Type, output frequency = k•(crystal reference). On-chip LC VCO possible.
Mixer	DC-60 MHz needed for 1 input; 1.5-2 GHz overall bandwidth	Achievable; Gilbert quad or single-device
Power Amplifier	Output Power: -10 dBm@ 2GHz -3dB BW > 2 GHz	Low power output; Some weak filtering also desirable at output to eliminate frequency spurs.
Receiver		
Filter	Sampling Demod: Q>75, No CDMA Q>20, CDMA	Assumes homodyne demod; these numbers are considerably reduced if a heterodyne scheme is employed.
Amplifier	80 dB total gain; ~10 dB total noise figure for front end; 3dB BW of RF amplifiters > 2 GHz	Stability is a problem if all gain is to be done at the RF (i.e. homodyne) before sampling demodulation. Requires ~ 0.2 micron MOS devices.
Sample-and-Hold	8-bit accuracy over 1V; Track BW > 3 GHz $\Delta \bullet t_{acq} < 1.3 \bullet 10^{-12}$ V•sec	Requires 0.2 micron MOS devices.
A/D Converter	8-bit accuracy; > 60 MHZ conversion rate	Power consumption may be severe at these rates.
Crystal Frequency Reference	Phase noise < 50 ppm (±2.5 psec deviation in sample point @ 60 MHz oscillation)	Accuracy needed since a 1.5-2 GHz waveform is to be sampled.

Figure 4.17: Analog Performance Requirements

- The transceiver employs sampling demodulation in the demodulator, merging the mixer directly into the A/D conversion circuitry, simplifying the hardware considerably.

Likewise, this proposed system capitalizes heavily upon increased baseband processing to meet its goals - the use of differential encoding on the signal to eliminate the need for phase recovery, digital phase-locking to an accurate baseband oscillator to obviate the need for frequency recovery, and use of PN spread-spectrum to alleviate the passband anti-alias filtering requirements. It is the amalgamation of all of these techniques which yields such a simple circuit solution to the analog transceiver, as opposed to the conventional coherent multi-IF heterodyne demodulators, which are significantly more complex with less aggressive performance requirements. Using the advances in silicon MOS processing yielded by the efforts for increased digital performance, complete implementation in MOS is also feasible - for a 0.2μ technology, the achievable f_t's will have reached the 20 GHz range, and the speed-accuracy product for the sample-and-hold will be sufficient to accurately sample the 1.5 GHz modulated signal at 100 MHz.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

In this project, the performance specifications for a high-speed, integrated transceiver system have been described. Intended for use to provide a mobile link in a future portable digital personal communications system, supporting digital video and remote data access, such a transceiver must support continuous data rates in excess of 2 MBps per user, while maintaining the requirements of minimal bandwidth consumption, low bit-error rates, portability, and small physical dimensions - criteria which have not been met by any transceiver system to date.

First, the concept of cellular networking is extended to exploit as much gain in spectral efficiency as possible. Especially effective in an indoor environment, cellular networks utilize frequency reuse by separating users in the same frequency band spatially, and lowering transmit power as to make the cells as small as possible within the constraints of complexity of network control. For an indoor environment, cells on the order of 4 meters in radius are viable, with each cell potentially being contained in a single office or room, with the system employing a frequency reuse factor of 6-7. With such a configuration, ten users per

cell are supportable, each transmitting at 2 MBps. Although the system consumes a total of 250 MHz, this could support the entire population of the Department of EECS at this University with 500 persons all transmitting 2 MBps simultaneously. Of course, the unfortunate difficulty with indoor microcell transmission lies in the multipath characteristics of the channel, which results in intersymbol interference and symbol degradation. To evaluate the effects of multipath on the system, a model developed from statistical channel measurements has been built for the CAPSIM simulation engine; the simulated impulse response and channel eye diagrams when transmitting QPSK signals for such a channel are shown in chapter 2. Fading nulls as deep as 25 dB are encountered; as the transceiver moves over a short distance, the SNR can thus vary tremendously, warranting the use of coding and diversity to minimize the impact of multipath.

The next system specification lies in the modulation used to transmit the signals and the multiple access schemes used to separate individual users. Due to the extreme advantages of spectral efficiency afforded by linear schemes as shown in figure 2.3, and the low transmit power of a microcell environment, continuous-phase modulation has few advantages over quadrature amplitude modulation. The major advantage of CPM schemes is the ability to demodulate the received signal incoherently; coherent demodulation is undesirable since carrier and phase recovery requires complex analog feedback loops and hence increased complexity in the analog hardware. However, through the use of differential encoding and accurate frequency references, QAM can also be demodulated in an incoherent fashion, and thus simplifying the analog circuitry at a minimal cost in digital encoding and processing.

As far as multiple-access schemes are concerned, there are two promising candidates: CDMA or FH/SSMA. CDMA uses pseudo-noise spread-spectrum coding to allow all users to transmit in the full frequency band all of the time, but separated within the signal space by orthogonal codes. By employing a correlation receiver, the signals can thus be uniquely recovered. CDMA also has the significant advantage of being naturally immune to multi-

path distortion, due to the increased time-resolution effect of the spectrum spread, thus eliminating the need for a complex adaptive decision feedback equalizer to eliminate the effects of the channel distortion. Unfortunately, CDMA also has its share of problems, difficulty in code design, synchronization of the correlation, and near-far effects being the most significant. These difficulties leads to some inefficiency; it was shown that only 21 as opposed to 31 users could be multiplexed with a spreading factor of 31 with a BER of 10^{-6} and QPSK modulation. Likewise, since the performance of CDMA with linear modulation is strongly dependent on maximizing the minimum distance between constellation symbols, the use of smaller constellations is justified.

On the other hand, FH/SSMA is the modern version of frequency-division multiple access, allocating to each user a small slice of the available spectrum. However, the allocated band moves about, hopping in frequency to minimize the effects of the multipath channel by simply moving out of fading nulls. The advantage of FH/SSMA is that it attempts to transmit each user's data in as small a bandwidth as possible, thus minimizing the size and complexity of the required adaptive equalizer. Unfortunately, the ability to demodulate the signal in a homodyne fashion is compromised, as impracticably-high filtering requirements are required. As was described in chapter 4, homodyne demodulation is one way simplifying the analog receiver, since the entire IF stage is eliminated. Due to its immunity to multipath, and the ability to minimize the filtering requirements when used in a homodyne scheme, CDMA was chosen.

Given that CDMA and QPSK modulation are to be employed, the analog hardware requirements were established in chapter 4, with the key goals of simplification and integration of the analog hardware. Using MOS device scaling, it was shown that an MOS transceiver capable of supporting the 2 MBps data rate at a carrier of 2 GHz is feasible, given a scaled technology with a minimum feature size of 0.2 microns. When the channel lengths have scaled to this point, the achievable f_t of the device, and the speed-accuracy product of the sample-and-holds have reached the necessary performance levels, as listed in figure 4.17.

Taking advantage of this, the use of highly accurate crystal-referenced PLL VCO's, and digital differential encoding, the concept of sampling demodulation has been developed, which completely eliminates the downconversion mixer and VCO in favor of a single fixed-frequency sampling switch. Indeed, the vast majority of the complexity of the system has been moved into the digital domain, such as the need for CDMA synchronization, to allow the return to an extremely simple, homodyne demodulator that does not even require an explicit mixing step. Combined with the use of on-chip inductors and filters, a completely integrated transceiver system, supporting 2 MBps with QPSK and CDMA, has been specified.

Future Work

With this specification as a springboard, there are many possible routes for further design and analysis. First, the obvious one is to perform the low-level analog hardware design and implement blocks which meet the performance levels specified above. Although 0.2 micron technologies are not readily available at the moment, 0.5 micron processes are becoming relatively common, and can be used to explore the design limitations with a fabricated circuit. In this thesis, the issue of low-power has been addressed in generality, by minimizing parasitics and required hardware; however, specific hardware strategies for reducing power consumption can be further explored. For example, the required high-speed 8-bit A/D converter can be implemented as a flash topology, at the expense of extreme power requirements in the form of 256 comparators. A more elegant solution lies in the use of pipelined A/D converters [38], which use a sample-and-hold as a sort of analog "pipeline register" through multiple stages of small flash converters. For example, an 8-bit converter can be implemented as a pipeline of 3-3-2-bit converters, requiring only 20 comparators to perform the same operation. Other power-minimization strategies can be and need to be explored for the other components.

Second, the combined effects of diversity and multipath fading can be quantified, in terms of their impact on BER. Although it is well-known that diversity mitigates the effects of multipath, no exact, quantitative analysis exists, the difficulty of such an analysis arising from the extreme variability of the transmit channel. Likewise, the exact coding scheme to be used has not been specified; in the above analyses, a maximal-length shift-register code has been uniformly assumed, being a well-known coding scheme with good coding gain. However, the possibility of better codes always exists; in particular, the use of Ungerboeck/trellis codes [39] is one candidate for further exploration. Likewise, little effort has been given to development of extremely low-rate codes, for use in P/N spread-spectrum systems. Certainly, the lower the code rate the larger the possible coding gain, and for every dB of coding gain the performance and spectral efficiency of CDMA systems improves significantly, due to its extreme sensitivity to SNR. Likewise, the development of good synchronization strategies for P/N spread-spectrum is needed. Although RAKE receivers and delay-locked loops are well-known solutions for this, they are difficult to implement given the extremely high data rates of the system, and alternative solutions need to be considered.

Bibliography

[1] Coding of Moving Pictures and Associated Audio, ISO-IEC/JTC1/SC2/WG11/MPEG, Draft of Standard ISO 11172, MPEG Video Committee; International Organization for Standardization (ISO), December 18, 1990.

[2] D.J. Hait, D.G. Messerschmitt. *The BLOSIM Simulation Program, Research Project Report*, U.C.Berkeley,1985.

[3] *CAPSIM User's Guide*, XCad Corp, Raleigh, N.C., 1989.

[4] W. C-Y Lee. *Mobile Cellular Telecommunications Systems*, McGraw-Hill Book Co.,USA 1989

[5] S.Y. Seidel, T.S. Rappaport. "914 MHz Path-Loss Prediction Models for Indoor Wireless Communications in Multi-floored Buildings," Submitted to *IEEE Transactions on Antennas and Propagation*, May 17, 1991.

[6] S.Y. Seidel, T.S. Rappaport. "Radio Channel Models in Manufacturing Environments," Wireless Information Networks Workshop, Rutgers University, Rutgers, New Jersey, June 15, 1989.

[7] K.L. Blackard, T.S. Rappaport, B. Tuch. "Radio Frequency Noise Measurements and Models for Indoor Wireless Communications at 918 MHz, 2.44 GHz, and 4.0 GHz," Submitted to *IEEE Transactions on Antennas and Propagation*, June 1991.

[8] S.O. Ohrvick et al. "Microcell Radio Channel: Preliminary Report on Indoor Field-Strength Measurements at 900 and 1700 MHz," *Memorandum of the Department of Applied Electronics*, University of Lund; Lund, Sweden, Jan, 1989.

[9] A.M. Saleh, R. A. Valenzuela. "A Statistical Model for Indoor Multipath Propagation", *IEEE Journal of Selected Areas in Communications*, Vol. SAC-5, No. 2, February, 1987, pp. 128-137.

[10] E.A.Lee, D.G.Messerschmitt. *Digital Communication*, Kluwer Academic Publishers, USA, 1988.

[11] B.Ekelund, M.Torkelson. "Waveform Generation for CPM Transmitters," *Memorandum of the Department of Applied Electronics*, University of Lund; Lund, Sweden, Jan, 1987.

[12] K. Murota, K. Hirade. "Transmission Performance of GMSK Modulation," *Transactions of the IECE of Japan*, Vol. 64-B, October, 1981, p. 1123.

[13] K. Raith, J.Uddenfeldt. "Capacity of Digital Cellular TDMA Systems," *IEEE Transactions on Vehicular Technology*, Vol. VT-40, No. 2, May 1991, pp. 323-332.

[14] L. Svennson. *Implementation Aspects of Decision-Feedback Equalizers for Mobile Telephones, Ph.D Dissertation*, Department of Applied Electronics, University of Lund, Lund, Sweden, May, 1990.

[15] J.G. Proakis. "Adaptive Equalization for TDMA Digital Mobile Radio," *IEEE Transactions on Vehicular Technology*, Vol. VT-40, No. 2, May 1991, pp. 323-332.

[16] M.K. Simon, J.K. Omura, R.A Scholtz, B.K. Levitt. *Spread-Spectrum Communications, Vol. 1-3*, Computer Science Press Inc., USA 1985.

[17] C.E. Cook, H.S. Marsh. "An Introduction to Spread Spectrum," *IEEE Communications Magazine*, March 1983, pp. 8-16.

[18] R.L. Pickholtz, D.L. Schilling. "Theory of Spread-Spectrum Communications - A Tutorial," *IEEE Transactions on Communications*, Vol. COM-30, No. 5, May 1982, pp.855-883.

[19] John G. Proakis, *Digital Communications*, McGraw-Hill Book Co., USA, 1983

[20] D.L. Schilling, R.L. Pickholtz, L.H Milstein, eds. "Spread-Spectrum Communications I and II," Special Issues of the *IEEE Journal of Selected Areas in Communications*, Vol. JSAC-8, No. 4-5, May-June 1990.

[21] R.C.Stirling. *Microwave Frequency Synthesizers*, Prentice-Hall Inc., New Jersey, USA 1987.

[22] Feher, K. *Advanced Digital Communications*, Prentice-Hall Inc., New Jersey, USA 1987.

[23] R. Gold. "Optimal Binary Sequences for Spread-Spectrum Multiplexing," *IEEE Transactions on Information Theory*, Vol. IT-13, May, 1980, pp.619-621.

[24] E.R. Berlekamp. *Algebraic coding theory*, McGraw-Hill Book Co., USA, 1968.

[25] H.B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*, Addison-Wesley Publishing Co., USA, 1990.

[26] P.K. Ko. *Course Notes for EECS 231, Physics of Semiconductor Devices*, Department of EECS, U.C. Berkeley, Berkeley, CA, Spring 1991.

[27] T. Yuzuriha, T. Yamaguchi, J. Lee. "Submicron Bipolar-CMOS Technologies Using 16 GHz f_t Double-Poly Si Bipolar Devices," IEEE 1988 International Electron Devices Meeting, San Francisco, CA, Dec. 11-14, 1988.

[28] R.K. Watts. *Submicron Integrated Circuits*, John Wiley and Sons, USA, 1989.

[29] J. Sevenhans et al. "An Integrated Si-Bipolar RF Transceiver for a Zero-IF 900 MHz GSM Digital Mobile Radio Front-end of a Hand Portable Phone," IEEE 1991 Custom Integrated Circuits Conference, San Diego, CA, May 12-15, 1991.

[30] N.M.Nguyen, R.G.Meyer. "Si IC-Compatible Inductors and LC Passive Filters," *IEEE Journal of Solid-State Circuits*, Vol. 25, No. 4, August 1990, pp.1028-1030.

[31] P.R. Gray, R.G. Meyer. *Analysis and Design of Analog Integrated Circuits, 2nd Edition,* John Wiley and Sons, USA, 1984.

[32] G.D. Vendelin, A.M. Pavio, U.L Rohde. *Microwave Circuit Design Using Linear and Nonlinear Techniques,* John Wiley and Sons, USA 1990.

[33] N. Nguyen. "A 1.8 GHz Monolithic LC Voltage-Controlled Oscillator," Integrated Circuits Seminar, Department of EECS, U.C. Berkeley, Berkeley, CA, Oct. 7, 1991.

[34] P.R. Gray. *Course Notes for EECS 290Y, Analog-Digital Converter Technologies,* Department of EECS, U.C. Berkeley, Berkeley, CA, Spring 1989.

[35] Y-M. Lin. *Performance Limitations of High-Resolution Video-Rate Analog-Digital Interfaces, Ph.D Dissertation,* Memorandum # UCB/ERL M90/55, U.C. Berkeley, 19 June 19,1990.

[36] K. Rush, P. Byrne. "A 4 GHz 8-Bit Data Acquisition System," IEEE 1991 International Solid-State Circuits Conference, San Francisco, CA, Feb. 13-15, 1991.

[37] Y. Gendal, Y. Komatsu et al. "An 8-Bit 500 MHz ADC," IEEE 1991 International Solid-State Circuits Conference, San Francisco, CA, Feb. 13-15, 1991.

[38] S.H. Lewis. *Video-Rate Analog-to-Digital Conversion Using Pipelined Architectures, Ph.D Dissertation,* Memorandum # UCB/ERL M87/90, U.C. Berkeley, Nov. 18, 1987.

[39] G. Ungerboeck. "Trellis-Coded Modulation with Redundant Signal Sets, Parts I and II," *IEEE Communications Magazine,* Vol. 25, No. 2, February 1987.

Appendix

Source Listings for Simulations

1. mpath.h- C-header file for any code requiring the GenerateMPath subroutine.
2. multipath.c- source listing for the GenerateMPath subroutine.
3. randomVar.c- C code to generate random variables with certain distributions, such as Gaussian or Exponential.
4. mpathImpulse.s- CAPSIM star which generates a (wideband) impulse response to the multipath channel. Useful for "true" analog simulations which model the complete modulation process.
5. multipath.s- star which generates the baseband-equivalent for the multipath channel. Useful for testing the baseband modulation/coding only, since the entire analog passband modulation/demodulation has been encapsulated into this model.
6. pifourDPSKdemod.s- demodulator and differential decoder for DPSK baseband modulation.
7. pifourDPSKmod.s- modulation and differential encoder for DPSK.
8. impRespt.t- CAPSIM galaxy listing to test the channel model; generates a simple impulse response.
9. noisi.t- galaxy listing for simulation of QPSK modulated signals transmitted over an ideal channel (no multipath, no noise).
10. isi.t- same as above, with multipath incorporated.
11. isinoise.t- same as above, with both multipath and noise effects introduced.
12. sample.t- galaxy to simulate a DQPSK transmitted signal being demodulated through the use of sampling demodulation.
13. transmit.t- galaxy to generate the modulated user signal for a CDMA system. To simplify simulation, the transmit and receive halves of this simulation were broken apart.
14. receive.t- CDMA receiver model.

```
/* mpath.h-
 *      Defines the data structures to describe a multipath profile.
 *      Used by the multipath stars and subroutines....
 *
 *      Sam Sheng, ssheng@zion.berkeley.edu
*/

#include <sys /types.h>
#include <sys /time.h>

/* Library functions */
time_t time();
double GaussRV(),ExponRV();
void GenerateMPath();



/* Useful macros */
#define min(a,b)        ((a) < (b)) ? (a) : (b)
#define max(a,b)        ((a) > (b)) ? (a) : (b)



/* Basic constants */
#define MAX_NUM_PATHS 100
#define VEL_LIGHT 2.99e8
#define PI 3.141592653589793



/* Structure to hold information about
 * amplitude and arrival times of the multipath.
*/
typedef struct {
     int number_paths;
     double ampl[MAX_NUM_PATHS];
     double arrTimes[MAX_NUM_PATHS];
} MPathType;



/* Holds information about the environment,
 * eg, los vs obs, path loss, number of paths, etc.
*/
typedef struct {
     int los;                          /* LOS or OBS topography? */
     double minDist;                   /* distance corresp. to first arrival */
     double pthLossExp;                /* path loss exponent */
     double refPathLoss;               /* reference path loss at 1m */
     double meanNumPath,sdPaths;       /* mean, SD of number of path components */
     double sdAmplit;                  /* SD of amplitude about power mean, in dB */
     double meanExcessDly;             /* mean delay time, in nanoseconds */
} EnvType;
```

```
/*
 * randomVar.c -
 *
 * The random variable routines used by these models
 *
 * There are two routines:  GaussRV and ExponRV.
 * Each returns a gaussian random variable, and exponential random
 * variable respectively.  They are WHITE processes with respect to
 * time, so don't expect any correlations between successive samples
 *
 * If you need to have correlated samples, with some model, use
 * a filter to achieve the desired power spectral density.
 *
*/

#include <stdio.h>
#include <math.h>


/* big = 2^31 */
#define BIG 2147483648.0


/* GaussRV()
 *
 * A truly naive method of generating random samples (but it passes
 * a blss Q-Q test pretty well), by simply exploiting the central limit
 * theorem.  We take a large number of [0,1] uniform RV's and
 * form the experimental mean.  This should be gaussian,
 * with mean NUM_TO_SUM /2 and variance = NUM_TO_SUM /12 (the 12 comes
 * from the variance of a uniform [0-1] RV)
 *
 * YOU MUST CALL srandom(seed) before using this routine!
 *
*/



#define NUM_TO_SUM          25

double GaussRV(mean,sigma)
double mean;
double sigma;
{
    double gaussianRV;
    double sumOfUniform;

    int i;



    sumOfUniform = 0.0;
    for (i=0; i<NUM_TO_SUM; i++) {
        sumOfUniform  += ((double) random())  / BIG;
    }
    gaussianRV = ((sumOfUniform - (NUM_TO_SUM /2.0))*
                  sqrt(12.0 /NUM_TO_SUM)*
                  sigma) + mean;

    return((double) gaussianRV);
}
```

randomVar.c

```
/* ExponRV:
 *
 * Produces exponentially distributed RV's by the inverse CDF
 * method, by mapping a [0,1] Uniform RV with beta*ln(1-uniform)
 * Passes the Q-Q test, done in blss.
 *
 * Mean /SD of the resulting random variables is beta.
* /


double ExponRV(beta)
double beta;
{
      double exponRV;
      double uniformRV;

      uniformRV = ((double) random()  / BIG);

      exponRV = -beta * log(1 - uniformRV);
      return((double) exponRV);
}
```

```
/* mpath.c
 *
 * This collection of subroutines generate a multipath profile
 * contained in the MpathType structure (see mpath.h).
 *
 * It does not attempt to do any quantization;  it simply generates
 * a profile based on the statistics of the indoor fading environment.
 *
 * Requires randomVar.c to generate spectrally-correct random variables....
 *
*/

#include <stdio.h>
#include <math.h>
#include <sys /types.h>
#include <sys /time.h>

#include "/home/zion3/ssheng/masters/src/subs/mpath.h"

/*
 * Compare two floating point numbers.
 * Needed for quicksort routine....
*/

static int dblcomp(i,j)
double *i,*j;
{
     return((*i < *j) ? -1 : 1);
}


/*
 * Generate a multipath profile....
*/
void GenerateMPath(mpath,envir,printResult)
MPathType *mpath;
EnvType envir;
int printResult;
{
     int i,number_paths;
     double thePathDelay,thePathLoss,pathDist;
     extern int dblcomp();

     /* Initialize random variables */
     srandom((int) time(NULL));


     /*************************************************************
      * Computes the number of paths, using a gaussian
      * random variable (at least 1, at most MAX_NUM_PATHS)
      *************************************************************/

     number_paths =
          (int) min(max(GaussRV(envir.meanNumPath,envir.sdPaths),1.0),MAX_NUM_PATHS);
     mpath->number_paths = number_paths;


     /*************************************************************
      * Compute and sort the possible delays,
      * using an exponentially-distributed model
      *************************************************************/
     for (i=0; i<number_paths; i++) {
```

```
        /* compute path delay, in nanoseconds */

        if (i==0) {                                          /* minimum-distance path */
            thePathDelay = (envir.minDist /VEL_LIGHT);
        } else {
            thePathDelay = ((1.0e-9 * ExponRV(envir.meanExcessDly))
                                    + (envir.minDist /VEL_LIGHT));
        }
        (mpath->arrTimes)[i] = thePathDelay;
    }
    qsort((char *) mpath->arrTimes,number_paths,sizeof(double),dblcomp);


    /*****************************************************************
     * Compute the path loss corresponding to each path
     * (in dB!) We take a normally distributed amplitude
     * variation about a power path-loss law
     *****************************************************************/
    for (i=0; i<number_paths; i++) {

        pathDist = ((mpath->arrTimes)[i])*VEL_LIGHT;          /* path length in m */

        thePathLoss =
            10.0 * (envir.pthLossExp) * log10(pathDist) +
            (envir.refPathLoss) + GaussRV(0.0,envir.sdAmplit);
        (mpath->ampl)[i] = thePathLoss;
    }

    if (printResult) {
        fprintf(stderr,"Multipath Profile Generated:\n");
        fprintf(stderr,"-------------------------------\n");
        fprintf(stderr,"\tnumber of paths: %d\n",number_paths);
        for (i=0; i<number_paths; i++) {
            fprintf(stderr,
                "\tArrival Time: %e\tAmplitude (dB down): %e\n",
                (mpath->arrTimes)[i], (mpath->ampl)[i]
            );
        }
    }
}
```

```
/* mpathImpul.s
 *
 * This star represents the physical channel for an indoor multipath
 * propagation model.
 *
 *      a) the number of paths is normally distributed
 *      b) the delay times for each path is exponentially distributed
 *
 * This star generates the true impulse response of the channel.
 *
 * We calculate the attenuation of the path by using the exponential decay
 * model documented by Ohrvik, with power = 2.5.  Speed of light is assumed
 * to be its vacuum value, and distance is calculated as delay*spd_of_light
 * This is encapsulated into the function GenerateMPath.
 *
 * Essentially, this routine builds an FIR filter from a randomly generated
 * multipath profile, and uses the FIR filter as a digital equivalent for
 * the analog channel.
 *
 * Author- S.Sheng, ssheng@zion.berkeley.edu,
* /

#include "/home/zion3/ssheng/masters/src/subs/mpath.h"

/* Begin model * /

input_buffers
      delay_min = 0;
      delay_max = samples_delay + 1;
      float input;
end


output_buffers
      float output;
end


declarations
      int i;
      float outputSum;
end


states
      EnvType envir;                    /* propagation env. data * /
      MPathType mpath;                  /* multipath data * /
      float* quantArrTimes;             /* array of quantized arrival times * /
      int maxSamples;                   /* length of equivalent FIR filter * /
end


parameters
      param_def = "Enter sampling frequency";
      float sample_rate = 5.0e9;

      param_def = "Enter minimum-path distance";
      float minimumDist = 10.0;

      param_def = "Enter maximum delay allowable in sample periods";
      int samples_delay = 1000;

      param_def = "Print generated multipath profile?";
      int printResult = 1;
```

```
end


/*
 * Begin initialization code
*/
initialization_code

    /*************************************************************
     * Set up propagation environment.
     * These could be made into parameters of the model,
     * but with the exception of minDist there's no real reason
     * to right now, since we are looking specifically at the
     * indoor propagation environment.
     ************************************************************/

    envir.los= (int)                        1;          /* LOS environment? */
    envir.minDist= (double)          minimumDist;       /* minimum 1st-arrival dist */
    envir.pthLossExp= (double)              2.6;        /* Path loss exponent */
    envir.refPathLoss= (double)             25.0;       /* 1m path loss ref */
    envir.meanNumPath= (double)             22.4;       /* Mean number of paths */
    envir.sdPaths= (double)                 7.1;        /* sd of above */
    envir.sdAmplit= (double)                4.0;        /* sd (in dB) of amplitude */
    envir.meanExcessDly= (double)           30.0;       /* mean of excess delay tau */


    /*************************************************************
     *   Generate a multipath profile
     ************************************************************/

    GenerateMPath(&mpath,envir,printResult);


    /*************************************************************
     * Quantize the arrival times into capsim sample points
     * Fortunately, the GenerateMPath routine presorts the arrival
     * times for us, although we need to convert the "amplitude" value
     * back from dB into a true attenuation factor.
     *
     * Caveat: we need to make sure that the delay does not exceed the
     *             maximum possible length of our FIR filter, and set
     *             the maxSamples variable (true length of FIR filter)
     ************************************************************/

    quantArrTimes = (float *) calloc(samples_delay,sizeof(float));
    for (i=0; i< samples_delay; i++) {                                  for
        quantArrTimes[i] = 0.0;
    }

    /* Due to the time quantization, we add together all of the responses
     * that occur in that time step.
    */

    for (i=0; i< mpath.number_paths; i++) {                             for
        int quantizedTime;

        quantizedTime = (int) (mpath.arrTimes[i]*sample_rate);

        if (quantizedTime >= samples_delay) {
            fprintf(stderr,
                "WARNING: multipath profile exceeds delay length in model\n");
        } else {
            quantArrTimes[quantizedTime] += (float) exp10(-mpath.ampl[i] /20.0);
```

...for

```
                maxSamples = quantizedTime;
            }
        }
end


main_code
        while (it_in(0))                {                                   while
            it_out(0);

            outputSum = 0.0;
            for(i=0; i<=maxSamples; i++) {
                outputSum += (input(i)*quantArrTimes[i]);
            }
            output(0) = outputSum;
        }
        return(0);
end
```

```
/* multipath.s
 *
 * See also mpathImpul.s.
 *
 * This star calculates the equivalent BASEBAND multipath response,
 * which is essentially the channel response h(t)*exp(-jwct), where
 * wc is the carrier frequency.
 *
 * We calculate the attenuation of the path by using the exponential decay
 * model documented by Ohrvik, with power = 2.5.  Speed of light is assumed
 * to be its vacuum value, and distance is calculated as delay*spd_of_light
 * This is encapsulated into the function GenerateMPath.
 *
 * A similar FIR filter is built here, except the coefficients are now
 * necessarily complex.
 *
 * The inputs and outputs are COMPLEX valued, with two inputs /outputs
 * (one for real, and one for imaginary).
 *
 * Author- S.Sheng, ssheng@zion.berkeley.edu,
* /

#include "/home/zion3/ssheng/masters/src/subs/mpath.h"

/* Begin model * /

input_buffers
        delay_min = 0;
        delay_max = samples_delay + 1;
        float realX;

        delay_min = 0;
        delay_max = samples_delay + 1;
        float imagX;
end


output_buffers
        delay_max = 0;
        float realY;

        delay_max = 0;
        float imagY;
end

declarations
        int i;
        float amplitude;
        float firSumReal;
        float firSumImag;
end


states
        EnvType envir;                  /* propagation env. data * /
        MPathType mpath;                /* multipath data * /
        float* firReal;                 /* Real and imag components of * /
        float* firImag;                 /*       the channel * /
        int maxSamples;                 /* length of equivalent FIR filter * /
end


parameters
        param_def = "Enter carrier frequency";
```

```
    float carrier = 1.5e9;

    param_def = "Enter sampling frequency";
    float sample_rate = 2.0e9;

    param_def = "Enter minimum-path distance";
    float minimumDist = 10.0;

    param_def = "Enter maximum delay allowable in sample periods";
    int samples_delay = 1000;

    param_def = "Print generated multipath profile?";
    int printResult = 0;
end


/*
 * Begin initialization code
*/
initialization_code

    /*************************************************************
     * Set up propagation environment.
     * These could be made into parameters of the model,
     * but with the exception of minDist there's no real reason
     * to right now, since we are looking specifically at the
     * indoor propagation environment.
     ************************************************************/

    envir.los= (int)                    1;      /* LOS environment? */
    envir.minDist= (double)             minimumDist;  /* minimum 1st-arrival dist */
    envir.pthLossExp= (double)          2.6;    /* Path loss exponent */
    envir.refPathLoss= (double)         25.0;   /* 1m path loss ref */
    envir.meanNumPath= (double)         22.4;   /* Mean number of paths */
    envir.sdPaths= (double)             7;      /* sd of above */
    envir.sdAmplit= (double)            4.0;    /* sd (in dB) of amplitude */
    envir.meanExcessDly= (double)       30.0;   /* mean of excess delay tau */


    /*************************************************************
     *   Generate a multipath profile
     ************************************************************/

    GenerateMPath(&mpath,envir,printResult);


    /*************************************************************
     * Quantize the arrival times into capsim sample points
     * Fortunately, the GenerateMPath routine presorts the arrival
     * times for us, although we need to convert the "amplitude" value
     * back from dB into a true attenuation factor.
     *
     * Caveat: we need to make sure that the delay does not exceed the
     *         maximum possible length of our FIR filter, and set
     *         the maxSamples variable (true length of FIR filter)
     ************************************************************/

    firReal = (float *) calloc(samples_delay,sizeof(float));
    firImag = (float *) calloc(samples_delay,sizeof(float));
    for (i=0; i< samples_delay; i++) {
        firReal[i] = 0.0;
        firImag[i] = 0.0;
    }
```

for

```
     /* Due to the time quantization, we add together all of the responses
      * that occur in that time step.
     * /

     for (i=0; i< mpath.number_paths; i++) {                    for
          int quantizedTime;

          quantizedTime = (int) (mpath.arrTimes[i]*sample_rate);

          if (quantizedTime >= samples_delay) {
               fprintf(stderr,
                    "WARNING: multipath profile exceeds delay length in model\n");
          } else {

               amplitude = (float) exp10(-mpath.ampl[i] /20.0);

               firReal[quantizedTime] +=
                    (float) (amplitude*(cos(carrier*2.0*PI*mpath.arrTimes[i])));
               firImag[quantizedTime] +=
                    (float) (-amplitude*(sin(carrier*2.0*PI*mpath.arrTimes[i])));
               maxSamples = quantizedTime;
          }
     }
end


main_code
     while (it_in(0) && it_in(1)) {                             while
          it_out(0);
          it_out(1);

          firSumReal = 0.0;
          firSumImag = 0.0;
          for(i=0; i<=maxSamples; i++) {
               firSumReal += (realX(i)*firReal[i] - imagX(i)*firImag[i]);
               firSumImag += (realX(i)*firImag[i] + imagX(i)*firReal[i]);
          }
          realY(0) = firSumReal;
          imagY(0) = firSumImag;
     }
     return(0);
end
```

```
/*
 * pifourDPSKdemod.s
 *
 *    Demodulates a differentially encoded phase signal
 *    See pifourDPSKmod.s for the differential mapping scheme.
 *
 *    This does this the brute force way; just compute the
 *    differential phase and see what the output is.
 *    It takes one parameter; the peak signal that the system
 *    is expecting.
 *
*/

#define REAL 0
#define IMAG 1

input_buffers
      float dataR;
      float dataI;
end

output_buffers
      float dataOut;
end

/* Note that we use a global array lookup for
 * the 4PSK encoding
*/

declarations
    static float fourPSKArray[2][2][2] = {
                                                    { {1.0,1.0},{-1.0,1.0} },
                                                    { {1.0,-1.0},{-1.0,-1.0} }
                                                };
end

parameters
      float peak = 1.0;
end

states
      float theSymbR;
      float theSymbI;

      float prevSymbR;
      float prevSymbI;

      float phaseSymb;
      float phasePrev;

      float diffmag;
      float diffphase;

      float dataOut0;
      float dataOut1;

      float temp;
end

initialization_code
      prevSymbR=0.0;
      prevSymbI=0.0;
      theSymbR = 0.0;
      theSymbI = 0.0;
```

```
end

main_code

    while (it_in(0) && it_in(1)) {                                    while

        theSymbR = dataR(0);
        theSymbI = dataI(0);

        /* Figure the differential mag */

        diffmag = (float) ((theSymbR-prevSymbR)*(theSymbR-prevSymbR) +
                  (theSymbI-prevSymbI)*(theSymbI-prevSymbI));

        if (diffmag < (peak*peak /2.0)) {
            dataOut0=0.0;
            dataOut1=0.0;
        } else {

            /* Figure the differential phase */

            phaseSymb = (float) atan2((double) theSymbI,(double) theSymbR);
            phasePrev = (float) atan2((double) prevSymbI,(double) prevSymbR);

            diffphase = (float) ((phaseSymb - phasePrev)*(360.0 /(2.0*3.14159)));
            if (diffphase <= -180.0) {
                diffphase += 360.0;
            } else if (diffphase > 180.0) {
                diffphase -= 360.0;
            }

            /* Hard decode this */

            if ((diffphase >= 0.0) && (diffphase <= 135.0)) {
                dataOut0 = 0.0;
                dataOut1 = 1.0;
            } else if ((diffphase <= 0.0) && (diffphase >= -135.0)) {
                dataOut0 = 1.0;
                dataOut1 = 0.0;
            } else {
                dataOut0 = 1.0;
                dataOut1 = 1.0;
            }
        }

        /* Output */

        it_out(0);
        dataOut(0) = dataOut0;
        it_out(0);
        dataOut(0) = dataOut1;

        /* Update */

        prevSymbR = theSymbR;
        prevSymbI = theSymbI;
    }
    return(0);
end
```

```
/*
 * pifourDPSKmod.s
 *
 * Modulates a fourDPSK stream from its input, shifted in phase by pi/4.
 * INPUT DATA IS ASSUMED TO BE INTEGER BITS, as float data.
 * It also dumps data out to a file called dataSequence,
 * which is useful for monitoring the signal.
 *
 * It is implicit in the model that the starting symbol is (0,0);
 * hence we do not need to transmit it.
*/

#define REAL 0
#define IMAG 1

input_buffers
     float data;
end

output_buffers
     float inphase;
     float quad;
end

/* Note that we use a global array lookup for
 * the 4PSK encoding
*/

declarations
     static float fourPSKArray[2][2][2] = {
                                                    { {1.0,1.0},{-1.0,1.0} },
                                                    { {1.0,-1.0},{-1.0,-1.0} }
                                                };
     int j;
end

/*
 * sampleRate is the sampling rate of the "analog" output of this
 * signal.
 *
 * dataRate is the incoming BIT rate. dataRate/2 = BAUDrate
*/

parameters
     float sampleRate
     float dataRate;
end


states
     int sampNo;

     int theBit;
     int prevBit;

     int prevSymbR;
     int prevSymbI;

     FILE* dataSequence;

     int temp;
end

initialization_code
```

```
    sampNo = 0;

    prevSymbR=0;
    prevSymbI=0;

    theBit = 0;
    prevBit=0;

    dataSequence = fopen("dataSequence","w");
end

main_code

    while (it_in(0)) {                                                    while

        theBit = (int) data(0);

        /* Need to store previous bit */
        if ((sampNo % 2) == 0) {
            prevBit = theBit;
        } else {

            it_out(0);
            it_out(1);

            /* Note that the modulo operation is equivalent to
             * taking differences! (look at the constellation)
             * The if statement at the beginning takes care of
             * the necessary trading of the imaginary part and the
             * real part if the previous state is imaginary, ie (0,1) or (1,0).
             *
            */

            if ((prevBit != theBit) && (prevSymbR != prevSymbI)) {
                temp = prevSymbI;
                prevSymbI = prevSymbR;
                prevSymbR = temp;
            }

            inphase(0) = fourPSKArray[(prevBit + prevSymbR) % 2]
                                     [(theBit + prevSymbI) % 2][REAL];
            quad(0)    = fourPSKArray[(prevBit + prevSymbR) % 2]
                                     [(theBit + prevSymbI) % 2][IMAG];


            /* Memorize where the output signal was */

            prevSymbR = (prevBit + prevSymbR) % 2;
            prevSymbI = (theBit + prevSymbI) % 2;

            fprintf(dataSequence,"%d %f %f\n",sampNo,inphase(0),quad(0));


            /*
             * Dump out a sequence of zeros corresponding to the
             * the D /A operation.  Note that if we don't want
             * any padding zeros, then the sampleRate is half the
             * dataRate!!!!! (since we encoded 2 bits per symbol)
            */

            for (j=1; j< ((int)(2*sampleRate /dataRate)); j++) {

                it_out(REAL);
                it_out(IMAG);
```

pifourDPSKmod.s

...while

```
                    inphase(0) = 0.0;
                    quad(0) = 0.0;
                }
            }
            sampNo++;
        }
        return(0);
end

wrapup_code
        fclose(dataSequence);
end
```

impResp.t

```
# impResp.t
#
# Generates the wideband impulse response of the
# multipath channel.  This star is useful for testing
# the GenerateMPath routine, as well as examining the
# effects of changing the parameters of the environment
# (such as number of paths, amplitudes, etc).
#

arg -1 (none)

# .
# impulse generator
#
param int 5000
star impulse impulse

#
# Multipath environment
#
param float 40e9
param float 10.0
param int 5000
param int 1
star channel mpathImpul

#
# output data to the file idealData
#
param file impulseData
param int 1
star prfile0 prfile

connect impulse 0 channel 0
connect channel 0 prfile0 0
```

```
# topology file:  noisi.t
#
# Checks the data generator.
#
# The transmit and receive filters are assumed to be root-raised-cosine
# response filters (hence the overall response is Nyquist).
# Also, transmit amplitudes are normalized to 1.0, since power scales
# anyways.
#
# Baseband Sampling rate: 200MHz
# Carrier frequency: 1.5 GHz
# Baud rate: 1 MBaud
#

arg -1 (none)

#
# Transmitter side
#

param int 100
param float 1.0e+06
param float 2.0e+08
star psk dataSymbols

param file xmitFilt
param int 1000
star xmitFiltR convolve

param file xmitFilt
param int 1000
star xmitFiltI convolve

param float 50
star gainR gain

param file xmitFilt
param int 1000
star rxFiltR convolve

param float 50
star gainI gain

param file xmitFilt
param int 1000
star rxFiltI convolve

param file noisiDataR
param int 1
star prfile0 prfile

param file noisiDataI
param int 1
star prfile1 prfile

connect psk 0 xmitFiltR 0
connect psk 1 xmitFiltI 0
connect xmitFiltR 0 gainR 0
connect xmitFiltI 0 gainI 0
connect gainR 0 rxFiltR 0
connect rxFiltR 0 prfile0 0
connect gainI 0 rxFiltI 0
connect rxFiltI 0 prfile1 0
```

isi/isi.t

```
# topology file:  isi.t
#
# This file simulates an eye diagram of a QPSK signal being subject
# to a multipath environment.
# It is useful for demonstrating the extensive signal corruption that
# ISI can result in, as well as generating useful things like eye
# diagrams.
#
# No gaussian noise is added here.
#
# The transmit and receive filters are assumed to be root-raised-cosine
# response filters (hence the overall response is Nyquist).
# Also, transmit amplitudes are normalized to 1.0, since power scales
# anyways.
#
# Baseband Sampling rate: 200MHz
# Carrier frequency: 1.5 GHz
# Baud rate: 1 MBaud
#

arg -1 (none)

#
# Transmitter side
#

param int 100
param float 1.0e+06
param float 2.0e+08
star psk dataSymbols

param file xmitFilt
param int 1000
star xmitFiltR convolve

param file xmitFilt
param int 1000
star xmitFiltI convolve

#
# Multipath profile
#
param float 1.5e+09
param float 2.0e+08
param float 5.0
param int 200
param int 1
star multipth multipath

param float 50
star gainR gain

param file xmitFilt
param int 1000
star rxFiltR convolve

param float 50
star gainI gain

param file xmitFilt
param int 1000
star rxFiltI convolve

param file isiDataR
```

```
# topology file:  isinoisei.t
#
# This file simulates an eye diagram of a QPSK signal being subject
# to a multipath environment.
# It is useful for demonstrating the extensive signal corruption that
# ISI can result in, as well as generating useful things like eye
# diagrams.
#
# Gaussian noise is added in this simulation.
#
# The transmit and receive filters are assumed to be root-raised-cosine
# response filters (hence the overall response is Nyquist).
# Also, transmit amplitudes are normalized to 1.0, since power scales
# anyways.
#

arg -1 (none)

#
# Transmitter side
#

param int 100
param float 1.0e+06
param float 2e+08
star psk dataSymbols

param file xmitFilt
param int 1000
star xmitFiltR convolve

param file xmitFilt
param int 1000
star xmitFiltI convolve

#
# Multipath profile
#

param float 1.5e+09
param float 2.0e+8
param float 5.0
param int 200
param int 1
star multipth multipath


#
# Noise generators
#

param int 30000
param float 69.0e-6
param int 13624
star noiseR gaussNoise

param int 30000
param float 69.0e-6
param int 23251
star noiseI gaussNoise

param float 50
star gainR gain
```

isi/isi.t

```
param int 1
star prfile0 prfile

param file isiDataI
param int 1
star prfile1 prfile

connect psk 0 xmitFiltR 0
connect psk 1 xmitFiltI 0
connect xmitFiltR 0 multipth 0
connect xmitFiltI 0 multipth 1
connect multipth 0 gainR 0
connect multipth 1 gainI 0
connect gainR 0 rxFiltR 0
connect rxFiltR 0 prfile0 0
connect gainI 0 rxFiltI 0
connect rxFiltI 0 prfile1 0
```

```
##########################################################################
# DPSK.t
#     Modulates a 4-DPSK signal and then demodulates using two sampling
#     demodulator units, separated in phase by pi /4.  The PHASE of the
#     sampling operation is now completely unknown!!!!!  The differential
#     nature of the demodulation will still enable the signal to be recovered.
#
#     Also, differential encoding is also good if the phase changes slowly
#     with time, since the constellation "adapts" itself as to account for
#     the phase change.  Time varying phase is not modeled here.
##########################################################################


##########################################################################
# Will create the following files:
#     rawData: the binary data sequence generated by the datastar.
#     dataSequence: the symbol output from the baseband Modulation.
#     timebase: the time for each analog sample
#     anadata: the analog modulated waveform
#     userData: the output of the sampling demod.
#     decodedData: the output of the DPSK decoder.
##########################################################################

arg -l (none)

##########################################################################
# send out 80 bits of data
##########################################################################
param int 80
param int 1132514
star user data

##########################################################################
# encode differentially and with QPSK
#           analog output rate is 15 GHz! 10x oversample above the CARRIER..
#           spread factor of 31 on the data rate (assumes TDMA or SSMA)
##########################################################################
param float 15e9
param float 31e6
star basebandMod pifourDPSKmod

##########################################################################
# baseband filtering: 25% raised cosine filter
##########################################################################

param file xmitFilt
param int 5400
star xmitFiltR convolve

param file xmitFilt
param int 5400
star xmitFiltI convolve

#
# baseband data dump
#
star forkBBR frk
star forkBBI frk

param file basebandDataR
param int 1
star prfile0 prfile

param file basebandDataI
```

isinoize/isinoise.t

```
star addr add

param file xmitFilt
param int 1000
star rxFiltR convolve

param float 50
star gainI gain

star addi add

param file xmitFilt
param int 1000
star rxFiltI convolve

param file isiDataR
param int 1
star prfile0 prfile

param file isiDataI
param int 1
star prfile1 prfile

connect psk 0 xmitFiltR 0
connect psk 1 xmitFiltI 0
connect xmitFiltR 0 multipth 0
connect xmitFiltI 0 multipth 1
connect multipth 0 addr 0
connect multipth 1 addi 0
connect noiseR 0 addr 1
connect noiseI 0 addi 1
connect addr 0 rxFiltR 0
connect addi 0 rxFiltI 0
connect rxFiltR 0 gainR 0
connect gainR 0 prfile0 0
connect rxFiltI 0 gainI 0
connect gainI 0 prfile1 0
```

```
param int 1
star prfile1 prfile

###############################################################################
# transmit modulation
###############################################################################

star modR mixer
star modI mixer

param int 43300
param float 1.0
param float 15.0e9
param float 1.5e9
param float 0
star cosgen sine

param int 43300
param float 1.0
param float 15.0e9
param float 1.5e9
param float -90.0
star singen sine

star sumInQuad add

param file modded
param int 1
star dumpmodR prfile

star forkmodded frk

###############################################################################
# receive filtering
###############################################################################

param file rxfilt
param int 5400
star rxfilt convolve

star forkrxFilt frk

param file rxfiltDump
param int 1
star rxfiltDump prfile

connect user 0 basebandMod 0
connect basebandMod 0 xmitFiltR 0
connect basebandMod 1 xmitFiltI 0
connect xmitFiltR 0 forkBBR 0
connect forkBBR 0 modR 0
connect forkBBR 1 prfile0 0
connect xmitFiltI 0 forkBBI 0
connect forkBBI 0 modI 0
connect forkBBI 1 prfile1 0

connect cosgen 0 modR 1
connect singen 0 modI 1

connect modR 0 sumInQuad 0
connect modI 0 sumInQuad 1

connect sumInQuad 0 forkmodded 0
connect forkmodded 0 rxfilt 0
```

```
connect forkmodded 1 dumpmodR 0
connect rxfilt 0 forkrxFilt 0
connect forkrxFilt 0 rxfiltDump 0

########################################################################
#
# RECEIVER side decoding.
#
########################################################################


##############################################################################
# Sampling demodulation: Inphase and quadphase
#
# Beware: the input frequency for the samplestar is oversample x
# fcarrier owing to the oversampling.  The phase offset was randomly
# chosen to be 0.22 cycles.
########################################################################
param float 15.0e9
param float 100e6
param float 0
param int 0
star sampleR sample

param float 15.0e9
param float 100e6
param float 1.570797
param int 0
star sampleI sample

##############################################################################
# Gain term to offset loss in filters
##############################################################################

param float 50.0
star gainR gain

param float -50.0
star gainI gain

##############################################################################
# Decimate back down to the original bitstream
##############################################################################

param float 0.155
param float 36.0
param int 1
star decimR resmpl

param float 0.155
param float 36.0
param int 1
star decimI resmpl

##############################################################################
# Decode differentially
##############################################################################
param float 1.0
star DPSKdemod pifourDPSKdemod

##############################################################################
# Data dumping stars (forks and prfiles)
##############################################################################
star forksampdumpR frk
```

```
##############################################################################
# transmit.t
#     Modulates a 4-DPSK signal, and then spectrum-spreads it using a
#     constraint-length 5 MLSR sequence generated below.
#     Only a transmit filtering stage is used, since I actually WANT them
#     to alias (hence I assume a very sloppy, wideband receive filter).
#     In other words, no receive filter at all!
#
#     Thus, since I want to preserve the Nyquist zero-forcing criterion,
#     the xmit filter has a raised cosine xtic (not root-raised cosine)
#     This is intended to be a used as a subblock for the full simulation.
#
#     To save processing time, this block will be run by itself to generate
#     the modulated data for each user, the data will be added, and then
#     the receive block will read the generated data.
#
#     THE SPREADING SEQUENCE MUST BE CHANGED FOR EACH USER!!!!!
##############################################################################

arg -1 (none)

##############################################################################
# send out numbits of data per user
##############################################################################
param int 20
param int 1132514
star user data

##############################################################################
# encode differentially and with QPSK
##############################################################################
param float 15.5e6
param float 31e6
star basebandMod pifourDPSKmod

##############################################################################
# Spectrum spread to 31 MBaud
##############################################################################

param float 15.0e9
param float 15.5e6
param int 18
star spreaderR spreader

param float 15.0e9
param float 15.5e6
param int 18
star spreaderI spreader

star forkspreadR frk
star forkspreadI frk


##############################################################################
# baseband filtering: 25% raised cosine filter
##############################################################################

param file xmitFilt
param int 5800
star xmitFiltR convolve

param file xmitFilt
param int 5800
star xmitFiltI convolve
```

```
star forksampdumpI frk

star forkoutdumpR frk
star forkoutdumpI frk

param file sampledR
param int 1
star dumpsampleR prfile

param file sampledI
param int 1
star dumpsampleI prfile

param file outR
param int 1
star dumpoutR prfile

param file outI
param int 1
star dumpoutI prfile

param file decodedOut
param int 1
star decodedDump prfile

##############################################################################
# netlists
##############################################################################

connect forkrxFilt 1 sampleR 0
connect forkrxFilt 2 sampleI 0

connect sampleR 0 gainR 0
connect gainR 0 forksampdumpR 0
connect forksampdumpR 0 decimR 0
connect forksampdumpR 1 dumpsampleR 0

connect sampleI 0 gainI 0
connect gainI 0 forksampdumpI 0
connect forksampdumpI 0 decimI 0
connect forksampdumpI 1 dumpsampleI 0

connect decimR 0 forkoutdumpR 0
connect forkoutdumpR 0 DPSKdemod 0
connect forkoutdumpR 1 dumpoutR 0
connect decimI 0 forkoutdumpI 0
connect forkoutdumpI 0 DPSKdemod 1
connect forkoutdumpI 1 dumpoutI 0
connect DPSKdemod 0 decodedDump 0
```

transmit.t

```
connect forkspreadR 1 dumpspreadR 0
connect forkspreadI 1 dumpspreadI 0

connect spreaderR 1 nullstar 0
connect spreaderI 1 nullstar 1

connect xmitFiltR 0 forkBBR 0
connect forkBBR 0 modR 0
connect forkBBR 1 prfile0 0
connect xmitFiltI 0 forkBBI 0
connect forkBBI 0 modI 0
connect forkBBI 1 prfile1 0

connect cosgen 0 modR 1
connect singen 0 modI 1

connect modR 0 sumInQuad 0
connect modI 0 sumInQuad 1

connect sumInQuad 0 dumpmodR 0
```

```
#
# baseband data dump
#
star forkBBR frk
star forkBBI frk

param file basebandDataR
param int 1
star prfile0 prfile

param file basebandDataI
param int 1
star prfile1 prfile

##############################################################################
# transmit modulation
##############################################################################

star modR mixer
star modI mixer

param int 305000
param float 1.0
param float 15.0e9
param float 1.5e9
param float 0
star cosgen sine

param int 305000
param float 1.0
param float 15.0e9
param float 1.5e9
param float -90.0
star singen sine

star sumInQuad add

param file modded
param int 1
star dumpmodR prfile

param file spreadR
param int 1
star dumpspreadR prfile

param file spreadI
param int 1
star dumpspreadI prfile

star nullstar sink

##############################################################################
# netlist
##############################################################################

connect user 0 basebandMod 0
connect basebandMod 0 spreaderR 0
connect spreaderR 0 forkspreadR 0
connect forkspreadR 0 xmitFiltR 0
connect basebandMod 1 spreaderI 0
connect spreaderI 0 forkspreadI 0
connect forkspreadI 0 xmitFiltI 0
```

```
param int 1
star dumpsampleI prfile

###############################################################################
#
# correlate! Use the correlation block and a spreader code to acquire the
# data from the demodulated output
#
###############################################################################

param float 31e6
param float 31e6
param int 18
star spreadRec spreader

param float 1.0
param int 30
star constgen const

star forkCorrelA frk
star forkCorrelB frk

star correlR correlSimple
star correlI correlSimple

###############################################################################
#
# Output stars and auxiliary control
#
###############################################################################

param file decimoutR
param int 1
star dumpoutR prfile

param file decimoutI
param int 1
star dumpoutI prfile

param file msequence
param int 1
star dumpmseq prfile

param file correlOut
param int 1
star dumpcorrel prfile

connect loader1 addTx 0
connect loader2 addTx 0

connect addTx 0 sampleR 0
connect addTx 1 sampleI 0

connect sampleR 0 gainR 0
connect gainR 0 forksampdumpR 0
connect forksampdumpR 0 decimR 0
connect forksampdumpR 1 dumpsampleR 0

connect sampleI 0 gainI 0
connect gainI 0 forksampdumpI 0
connect forksampdumpI 0 decimI 0
connect forksampdumpI 1 dumpsampleI 0

connect decimR 0 forkoutdumpR 0
```

```
#########################################################################
#
# Receiver modeling for the CDMA version of the system!
#
#########################################################################

#########################################################################
# Get modulated data from the two tx directories
# and sum them up
#########################################################################

param file tx1mod
star loader1 read_file

param file tx2mod
star loader2 read_file

star addTx add

#########################################################################
# Beware: the input frequency for the samplestar is oversample x
# fcarrier owing to the oversampling.  The phase offset was randomly
# chosen to be 0.22 cycles.
#########################################################################
param float 15.0e9
param float 100e6
#param float 0.22
param float 0
param int 0
star sampleR sample

param float 15.0e9
param float 100e6
param float 1.570797
param int 0
star sampleI sample

param float 1.0
star gainR gain

param float -1.0
star gainI gain

param float 0.155
param float 36.0
param int 1
star decimR resmpl

param float 0.155
param float 36.0
param int 1
star decimI resmpl

star forksampdumpR frk
star forksampdumpI frk

star forkoutdumpR frk
star forkoutdumpI frk

param file sampledR
param int 1
star dumpsampleR prfile

param file sampledI
```

--

receive.t

```
connect forkoutdumpR 0 correlR 0
connect forkoutdumpR 1 dumpoutR 0
connect decimI 0 forkoutdumpI 0
connect forkoutdumpI 0 correlI 0
connect forkoutdumpI 1 dumpoutI 0

connect constgen 0 spreadRec 0
connect spreadRec 0 forkCorrelA 0
connect spreadRec 1 forkCorrelB 0
connect forkCorrelA 0 correlR 1
connect forkCorrelA 1 correlI 1
connect forkCorrelA 2 dumpmseq 0
connect forkCorrelB 0 correlR 2
connect forkCorrelB 1 correlI 2

connect correlR 0 dumpcorrel 0
connect correlI 0 dumpcorrel 1
```