Wideband Propagation Measurement Results, Simulation Models, and Processing Techniques For a Sliding Correlator Measurement System

William G. Newhall

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

Master of Science
in
Electrical Engineering

Theodore S. Rappaport, Chair
Warren L. Stutzman
Dennis Sweeney

November 13, 1997
Blacksburg, Virginia

Wideband Propagation Measurement Results, Simulation Models, and Processing Techniques For a Sliding Correlator Measurement System

William G. Newhall

Abstract

Radio wave propagation measurements provide a way to accurately and reliably characterize environments to assist in the development and optimization of wireless communication systems. As digital radio systems occupy wider bandwidths and use multipath signal combining to enhance quality of service, knowledge of time dispersion and the multipath structure of radio channels become increasingly important. The wideband measurement system presented herein provides a practical means to precisely measure the delays and strengths of individual multipath components which arrive at a radio receiver.

Presented in this Thesis are fundamental theory, practical implementation, and simulation models for a sliding correlator measurement system. The sliding correlator technique is explained in detail and large-scale measurement survey is presented. Techniques for statistically quantifying the characteristics of propagation using the sliding correlator measurements are presented and compared. The development of simulations of the sliding correlator system is described, and simulation results are used to test conventional and newly developed post-processing algorithms.

This Thesis presents a practical view of the sliding correlator measurement system, but its foundations are rooted in the theoretical results which are explained and derived herein. Propagation researchers and students in the wireless communication field may find this work and the cited references useful for continued study of wideband propagation measurements or for application of the sliding correlator system as a wideband measurement solution.

To those who think they can, and do.

Acknowledgements

I would like to thank Theodore Rappaport, Warren Stutzman, and Dennis Sweeney for participating on my committee and for their encouragement and technical assistance with this research. I would also like to express my appreciation to many MPRG graduate students for assisting me with this endeavor, especially Joe Liberti, Roger Skidmore, Keith Blankenship, and Kevin Saldanha. The MPRG staff was also extremely helpful, especially Annie Wade, Jenny Frank, Aurelia Scharnhorst, and Hilda Reynolds.

I also owe many thanks to Bob Brickhouse, Greg Bump, Scott Elson, Jeremy Harker, Scott McCulley, Tom Conley, and many others at Grayson Wireless for their encouragement. I especially appreciate the enthusiastic support and excellent management of Ken Talbott.

I have enjoyed the companionship, motivation, and encouragement supplied by many friends and family members. I thank Mike Metzgar, Jen Lesser, Roger Skidmore, Henry Giamella, Bob Newhall, and Barbara Ruebush.

For broadening my experience at Virginia Tech and my lifelong interests with the discipline of flight, I must thank Richard Humphreys, Mike Mills, Tim Pratt, Vince Crewey, Neal Kegley, and other fellow pilots at the Virginia Tech Airport and in the Civil Air Patrol.

Finally, I owe my parents, Robert and Roberta Newhall, the greatest thanks for the continuous encouragement, generous support, and abundant love they have given me throughout my studies at Virginia Tech.

Bill Newhall
December 11, 1997

Wideband Propagation Measurement Results, Simulation Models, and Processing Techniques For a Sliding Correlator Measurement System

William G. Newhall

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Perspective

Radio has had a profound impact on society. Radio in its most popularly accepted sense, the transmission and reception of information using electromagnetic waves, has unleashed the population from the restrictions of distance and has allowed nearly instantaneous dissemination of facts, news, and knowledge. Beginning near the turn of the twentieth century, radio worked its way into lives of all social and economic classes of people. From the early transatlantic crossings of Marconi's Morse code signaling and experimental voice broadcasts of the first enthusiastic amateurs, radio has evolved into reliable and essential communication systems.

A rich history precedes today's high speed, low cost, fixed and portable wireless systems (see [2], [29], [30], [31]). Guglielmo Marconi, an Italian experimenter who built upon the work of Heinrich Hertz, is credited with not only proving to the world the practicality of wireless communications, but also with development of the first commercial application of radio in 1898: ship-to-shore communications. Three years later, Marconi successfully demonstrated the powerful potential of radio by transmitting messages across the Atlantic Ocean between antennas suspended by kites. The evolution of practical radio systems had begun.

In America, radio began its widespread popularity with the first scheduled broadcast on November 2, 1920. Homebuilt receivers allowed a small group of the radio amateurs to be the first to hear the results of presidential elections. Radio finally and unquestionably proved itself to be commercially viable when the first *sponsored* broadcast was transmitted on August 18, 1922. This date marks the moment when radio extended outside the realm of science and into the marketplace of big business.

Mobile radio was born in the 1940s and 1950s when the development of the transistor allowed compact design of receivers. Whereas the audience up until this time was confined to listening to a radio in the home, the new receivers allowed the audience to listen to broadcasts in their automobiles. It is interesting to note that from this point on in radio history, radio usage peaked during commuter rush hours—apparently a prophecy heeded by the later developers of cellular networks.

The history of today's personal communications originated when mobile radiotelephone networks were introduced. The concept of cellular systems, utilizing relatively small coverage regions and frequency reuse to serve a larger portion of the population, was developed by Bell Labs engineers in the 1960s. Although the theory of cellular was becoming well established, hardware technology lagged behind until the late 1970s when practical systems could be tested. In the early 1980s, cellular systems finally became a reality when the FCC allocated spectrum for the Advanced Mobile Phone System (AMPS). As networks were planned, tested, and activated, demand for propagation prediction, propagation measurements, and network optimization grew exponentially. The first versions of today's complex measurement systems and mobile communications simulation tools were born.

As mobile radio, law enforcement radio, medical radio applications, and other specialized wireless applications began a trend toward digital technology, measurement systems needed to transition from simple path loss recorders to systems which could characterize the time dispersion in radio channels caused by multipath. Systems of this sort which had been developed and applied for advanced research were now finding their way into digital mobile and portable radio applications to predict performance in various environments.

Today, wideband measurement systems are needed to support the future of wireless communications. If history is an accurate indication of the rate of growth of radio, the expansion of technology to build future generations of wireless applications will be staggering.

1.2 Motivation

Wideband measurement systems can be implemented using a method called sliding correlation. The sliding correlator technique has been used since the early 1970s to study the path loss and time dispersion characteristics of radio channels [5]. Since that time, numerous researchers have embarked on a wide range of research efforts which employ wideband measurement techniques including sliding correlation (e.g., [7], [15], [18], [19]). Frequency bands between 800 MHz and 2 GHz currently used for digital point-to-point and mobile communications have been studied extensively for the development of personal communication systems (PCS), wireless local loops, and wireless local area networks (LAN). As wireless technology evolves, new frequency bands will be occupied to accommodate emerging applications in the increasingly congested electromagnetic spectrum. Higher bandwidth requirements and band availability are forcing a migration into the upper limits of the usable radio frequency spectrum, and wideband measurements will continue at these high frequencies to augment past research. Hence, sliding correlator research involving measurement, development, modeling, and simulation continues.

Radio wave propagation research, including measurement system development and radio channel characterization, has been driven, directed, and funded by the wireless communications industry. Industry members with approximately 16 million customers in 1994 funded the research effort to establish the technology and infrastructure which is in place today. Predictions made in August 1997 should be an indication that wireless research will be driven even more intensely: By the end of 1997, industry analysis firms are predicting that there will be 56.6 million subscribers of cellular and PCS generating more than $33.8 billion in revenue [32]. The growth trend is forecast to increase to $93.7 billion by 2007 [32]. To support this outstanding customer demand, optimization of networks for increased capacity and quality will precede the eventual need for the next wireless generation which surpasses current promising system standards. Fortunately for many researchers and engineers, wireless measurement systems will be needed every step of the way.

1.3 Overview and Purpose

The general purpose of this research is to study, develop, explore, and disseminate principles of wideband measurements using a sliding correlator measurement system. Specifically, foundational analytic theory, results of a recent propagation study, system simulation models, and data processing techniques are presented for the purpose of supporting current and future wideband measurement research.

The goal of every researcher should be to direct a research effort toward end results which are beneficial to the research community. In doing so, benefits are realized by the society which supports the industry that depends upon the research. The research presented in the following chapters is admittedly far removed by several layers from serving the general population directly; its purpose is to serve those who are one level closer to providing the actual products and services which have become or will become essential in the function of today's technologically oriented society.

The material in Chapter 2 is intended to be an introduction and a review of wideband measurements and the sliding correlation technique. Students of the radio discipline who are entering the wireless communications research field will find an ample amount of descriptions and theory on which to build future studies about or using the sliding correlator system. Seasoned researchers will find interest in the implementation, synchronization, and analysis techniques described in detail.

Chapter 3 reports the approach, analysis, and execution of a wideband propagation study aimed at determining the characteristics of radio channels within a train yard environment. The statistical results of path loss and delay spread derived from power delay profiles measured with a sliding correlator system may be used for design or simulation of digital radio and transponder systems for rail car tracking and yard communications. Calibration and measurement details are explained to support future measurement work.

Simulations of physical systems has played a vital role in design and optimization in all types of engineering problems. Chapter 4 presents two simulation techniques which were developed primarily for the purpose of testing post processing techniques. The first simulation aids in understanding the generation and correlation properties of PN sequences. The second of the two simulation techniques builds upon a rigorous mathematical model to produce signals generated by the sliding correlator system. These simulations not only serve the purpose of producing power delay profiles for given system and channel specifications, but also aid in understanding the mechanisms which influence measured power delay profiles; this is crucial for specification of a sliding correlator system design and prediction of its performance with respect to noise and multipath detection capabilities.

In Chapter 5, post processing techniques are discussed. The analytical methods for computing channel characterization parameters for real power delay profiles are presented. Subjective factors which affect the results of conventional processing are also discussed. These conventional processing topics provide the measurement system operator with theoretical and practical guidance for post processing tasks. The second half of this chapter presents an algorithm for signal component detection within power delay profiles. The development of this algorithm was driven by two needs which surfaced during the course of this research: validation of propagation prediction tools that predict the arrival of individual multipath components at specific sites, and automation of multipath component identification for use by measurement tools which have a low bandwidth data link between the data collection hardware and the storage medium. The algorithm is demonstrated using power delay profiles simulated using the software written for Chapter 4 research. Profiles measured during the study described in Chapter 2 are also used as test samples.

Hardware implementation, measurement theory and practice, system simulation, and processing to produce meaningful results are the topics which have been studied throughout this research.

Support of current research and expansion of a knowledge base for future researchers are the motives. The following chapters represent a part of this author's contributions.

Chapter 2

The Sliding Correlator Measurement System

As digital radio systems employ standards requiring larger bandwidths to handle higher data throughput, understanding the effect of radio channels on wideband signals becomes increasingly important. Wideband measurements provide the propagation data needed to study characteristics of electromagnetic wave propagation for high data rate communication systems. This chapter presents an introduction to radio channel characterization and wideband measurement systems which are used to study radio wave propagation. The sliding correlator technique for determining the propagation characteristics is described in detail, and an implementation of a sliding correlator measurement system is discussed. Lastly, the analysis of sliding correlator measurement data to produce useful quantitative channel information is discussed.

2.1 A Model for Characterizing Propagation Through Radio Channels

Propagation of electromagnetic waves through radio channels can be characterized by the delays and losses caused by the channel. A practical radio channel typically consists of multiple physical propagation paths, each of which induces a propagation loss and delay. The impulse response of a multiple path (multipath) channel can be described mathematically as shown in equation (2.1-1) and graphically as shown in Figure 2.1-1.

$$h(t)=\sum_{i=1}^{N} a_i e^{jq_i} d(t-t_i) \qquad (2.1\text{-}1)$$

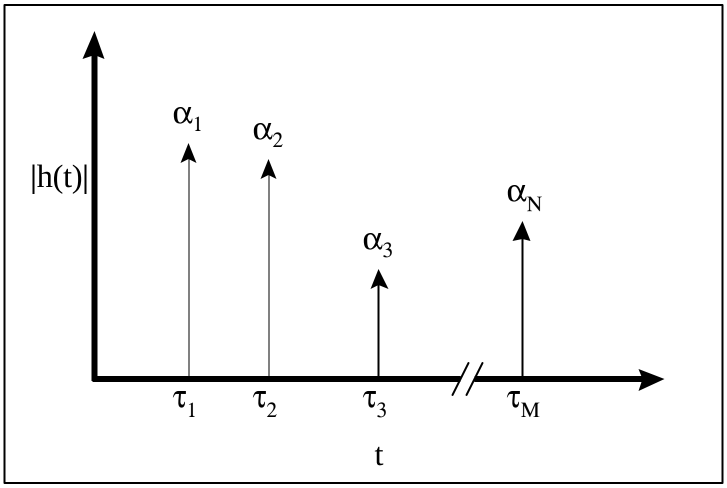


Figure 2.1-1. Pictured is the generalized impulse response of a multipath radio channel. Each delta function represents one physical path, and each path can be characterized by an attenuation and a delay.

Each delta function in (2.1-1) and Figure 2.1-1 represents a single physical path between a transmitter antenna and a receiver antenna. These delta functions are defined as multipath components of the received signal, and the combined set of all multipath components form the multipath structure of the channel. If a line-of-sight (LOS) path exists between the transmitter and receiver antennas, the first arriving component in the multipath structure of the channel will always be the LOS component. Components following the LOS component are caused by additional physical propagation paths through the environment. Components which travel along two separate paths and exhibit the same propagation delay will not be resolvable as individual components. The combination of the components in such a case will produce a resultant component that is the vector sum of the received fields, possibly causing fading of the resultant component.

Reflection, diffraction, and scattering of electromagnetic waves around objects are the fundamental causes of multipath. Reflection of a radio wave occurs for wave fronts which encounter large (relative to wavelength) impeding physical objects. Diffraction occurs when the electromagnetic waves encounter large, sharp irregularities of an object, such as edges or vertices of boundary surfaces [1]. Scattering occurs when electromagnetic fields propagate through a region of space where a high density of small objects exist or where small physical bodies protrude from a large body, such as along a rough wall [2] or through dense clutter.

The distance between a transmitter antenna and a receiver antenna along a specific path determines the absolute propagation time for that path. If multiple paths exist, relative propagation time is defined to be the time difference between a multipath component and a reference component, such as the LOS component or the first arriving component. The distinction between absolute and relative propagation time is made because the quality of a typical digital radio link is more affected by the time dispersion caused by large differences in relative time of arrival of multipath components at the receiver. For high data rate communication systems, relative propagation time (referred to as relative delay or excess delay) can be a significant factor in predicting the performance of the system.

2.2 Overview of Wideband Multipath Measurement Systems

Wideband multipath measurement systems provide an estimate of impulse responses [34] for measured radio channels. Large sets of wideband measurement data are processed to characterize radio channels using parameters which provide a statistical measure of multipath in an environment. Three practical techniques for measuring the impulse response of a radio channel are direct pulse sounding, frequency domain measurement, and sliding correlation [2]. A description and the tradeoffs of each of these types of measurement systems are presented here.

The direct pulse measurement system uses the channel sounding technique of transmitting through the radio channel a sinusoid gated by a very narrow pulse. This system operates similar to a high-resolution bistatic pulse radar system [6]. A block diagram of a direct pulse system is shown in Figure 2.2-1 and Figure 2.2-2.

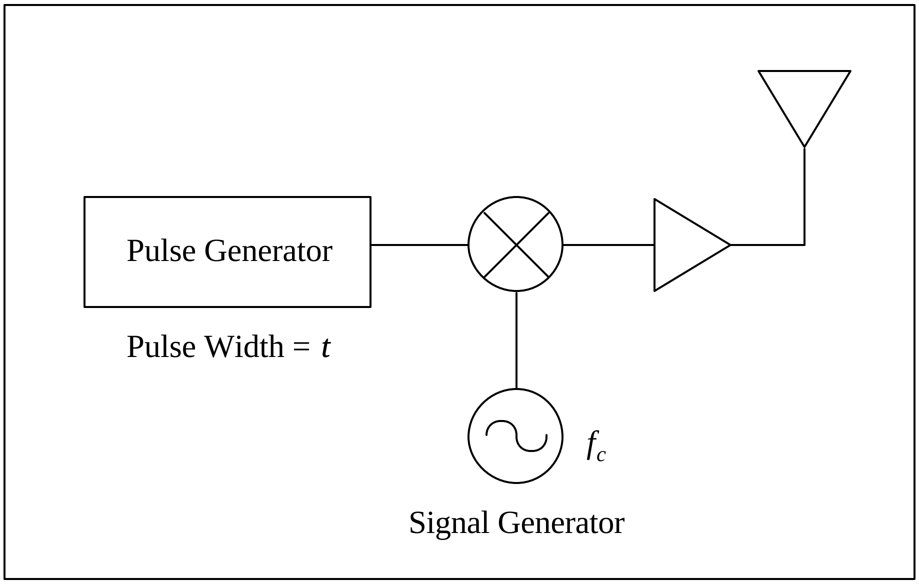


Figure 2.2-1. The direct pulse measurement system transmitter modulates a sinusoidal carrier with a narrow pulse to produce the sounding signal. The sounding signal is amplified and transmitted through the channel using an antenna.

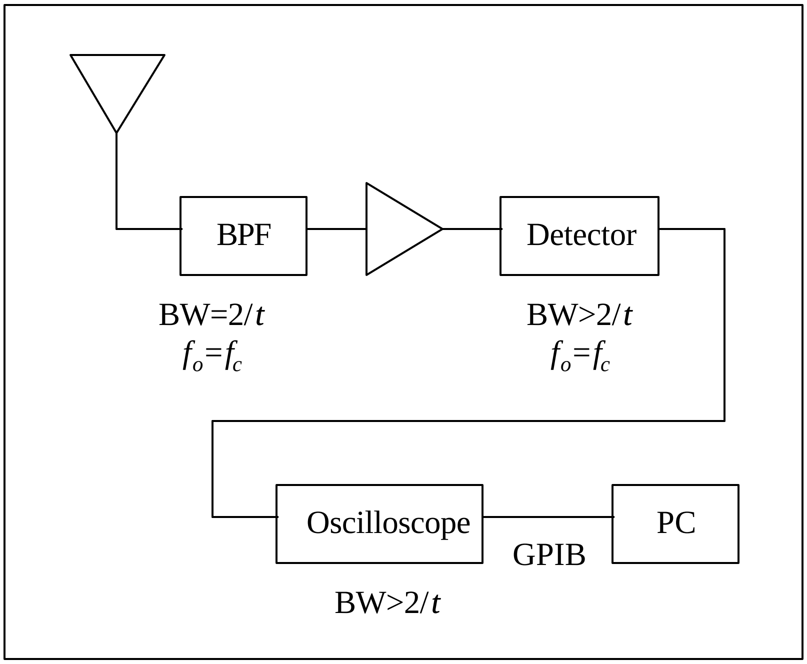


Figure 2.2-2. The direct pulse measurement system receiver requires a wideband filter at the RF front end to attenuate out-of-band interferers. The output of the detector is displayed on the oscilloscope screen.

The pulse measurement system approximates the process of sending a true impulse (delta function) through the channel in order to determine the impulse response. This process is commonly referred to as channel sounding. At the transmitter, the pulse generator modulates a sinusoidal carrier with a train of pulses of width t, and the modulated signal is amplified and transmitted through the channel. Figure 2.2-3 shows the wideband spectrum produced at the transmitter. The transmitted signal, or sounding signal, has a null-to-null bandwidth of $2/t$.

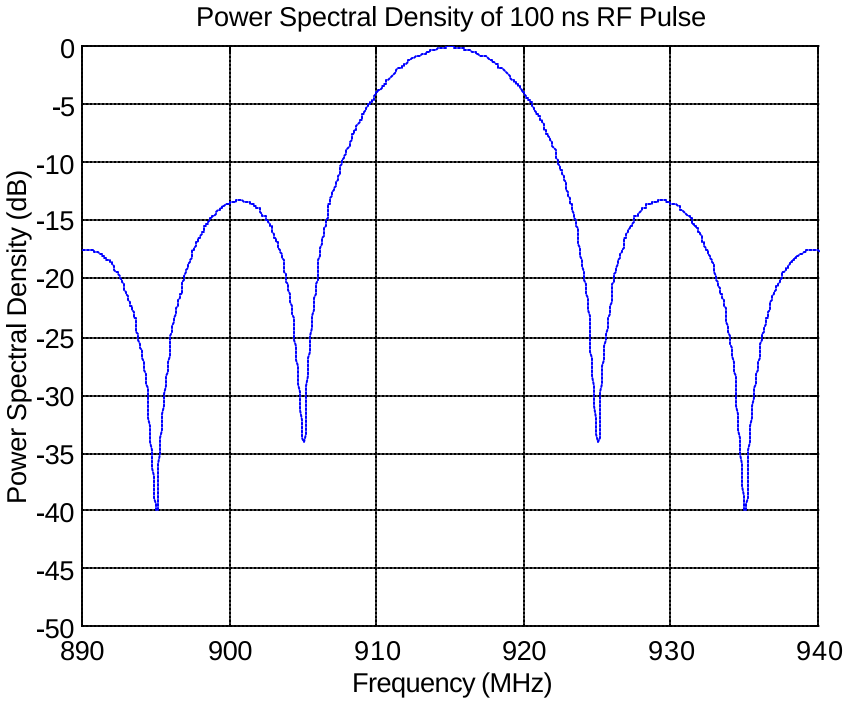


Figure 2.2-3. Shown is the power spectral density of pulse measurement system sounding signal using a 100 ns pulse to modulating 915 MHz continuous wave carrier.

At the receiver, a bandpass filter centered at the carrier frequency with a bandwidth $2/t$ rejects out-of-band signals. The received signal is amplified, power detected, and displayed on an oscilloscope. Shown on the oscilloscope is a power versus time plot of the channel response to the sounding signal. This response is used to approximate the channel impulse response for communication systems using a bandwidth narrower than the bandwidth of the sounding signal used to measure the channel.

One advantage of the pulse system is its relative simplicity. Implementation of the system requires only common RF laboratory equipment and components. A second advantage of the system is its ability to produce true snapshots of the multipath components in the channel. That is, as long as the channel remains invariant over the period beginning when the sounding signal leaves the transmitter and ending when the last multipath component arrives at the receiver, then channel measurements represent the channel at any single instant in time within that period. The primary disadvantage of the pulse system is the requirement of a wideband front end filter at the receiver. The wide bandwidth of the RF front end causes a high noise power at the detector, inhibiting the ability of the receiver to detect low level multipath components. Performing high resolution measurements becomes increasingly difficult when using pulses much shorter than 1 μs because of the large bandwidth required [5]. This type of measurement system was used to measure multipath delay statistics in [3] and [4].

A swept frequency measurement system, illustrated in Figure 2.2-4, measures the impulse response of a radio channel as if the channel were a lossy RF filter. Using an instrument such as a network analyzer, the swept frequency system measures the S_{21} s-parameter over a given frequency range. Performing an inverse discrete Fourier transform on the measured complex response produces the time domain impulse response of the radio channel.

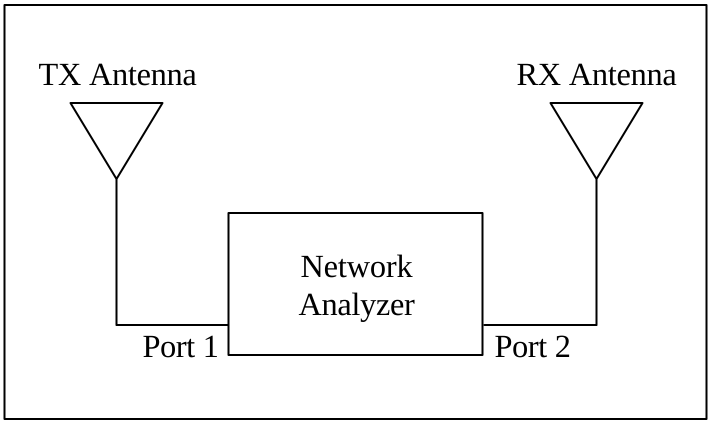


Figure 2.2-4. The basic frequency swept measurement system can be constructed using a network analyzer to measure the channel impulse response. The channel is measured in the same way an RF filter is measured.

This technique produces accurate amplitude and phase information about the radio channel, but suffers from limitations which prohibit use for practical channel measurements. For example, if a network analyzer is used, the transmitter and receiver antennas must both be connected to a single instrument by cables, making long distance or mobile channel measurements impracticable. Also, performing impulse response measurements for time varying channels is difficult if the frequency sweep period of the measurement system is long compared to the changes in the channel, such as variations caused by movement of objects in the propagation environment. If the frequency domain is swept too slowly, true snapshots of the channel will not be obtained. Rather, the frequency components at the end of the sweep will correspond to a channel different from that which existed at the beginning of the sweep.

The sliding correlator measurement system is a channel sounding system which uses direct-sequence spread spectrum (DS-SS) techniques to measure and record the impulse response of a radio channel. Instead of transmitting an RF pulse as with the pulse measurement system, typical sliding correlator transmitters continuously emit a continuous wave (CW) carrier modulated by a pseudorandom sequence. At the receiver, propagation delays of the sounding signal due to multipath are detected by correlating the received signal with a copy of the transmitted signal. The absolute and relative strengths of individual multipath components are also determined using the correlation process.

The sliding correlator inherits many of the advantages of spread spectrum communications systems. Rejection of in-band interferers and processing gain are realized in the correlation process. The sliding correlator system dilates the time scale of the channel responses at the receiver, making acquisition of accurate impulse response data easier. Time scale dilation at the receiver does, however, make it difficult to record snapshots of rapidly time-varying channels. However, as will be shown, the time scaling parameter can be set to a value appropriate for the expected time varying nature of the channel, and in many cases profiles which are equivalent to snapshots of the channel can be obtained.

2.3 Sliding Correlator Theory

The sliding correlator measurement technique, sometimes called the swept time-delay cross-correlation (STDCC) method [11], has been a popular method employed to study multipath characteristics of radio channels. Cox [5] developed one of the earlier systems to study Doppler characteristics of multipath in suburban environments. Devasirvatham used the same technique to study multipath characteristics in both indoor [7] and outdoor [15] radio channels.

Sliding correlator systems used by researchers for multipath measurements varied slightly in implementation, but all systems shared the basic building blocks of a spread spectrum system, used the same parameters for system configuration, and generally produced the same type of channel impulse response data. Figures 2.3-1 and 2.3-2 illustrate the general form of the sliding correlator from which past and current systems were derived.

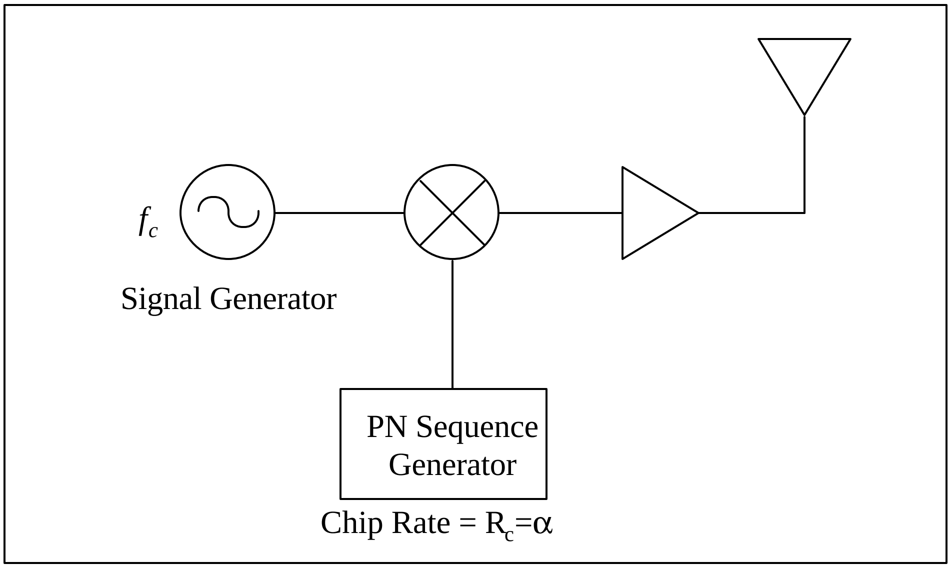


Figure 2.3-1. The general form of the sliding correlator measurement system transmitter consists of a PN sequence generator which is used to modulate a CW signal produced by a signal generator. The modulated signal is amplified and transmitted through the channel.

The signal generator at the transmitter produces a CW carrier signal at a frequency in the center of the band to be measured. The carrier is modulated with a pseudonoise sequence (PN sequence or pseudorandom binary sequence) of length l and chip rate R_c. The chip rate R_c is determined by the transmitter chip clock, a second signal generator in the measurement system. The modulated signal is amplified and transmitted through the radio channel using an antenna appropriate for the center frequency and bandwidth of the sounding signal.

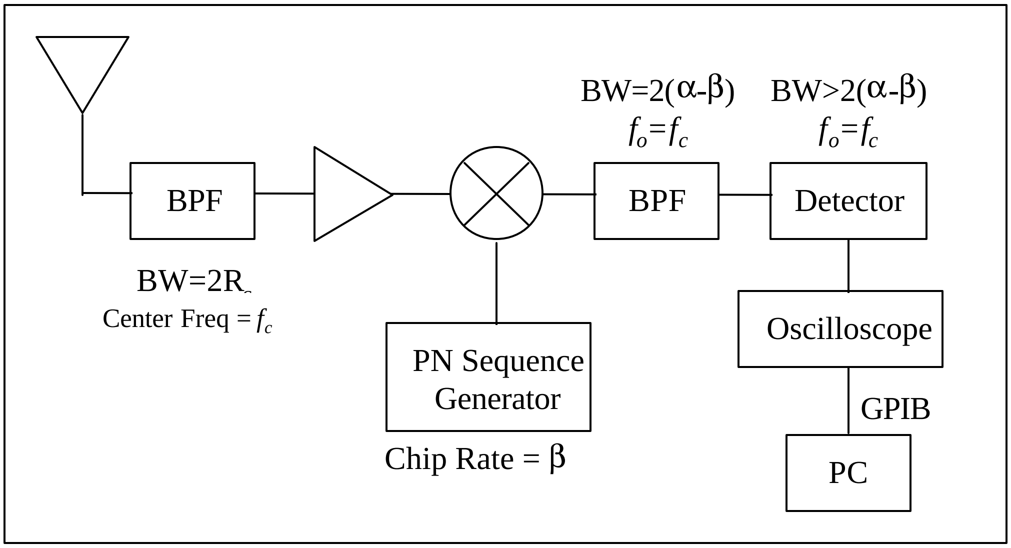


Figure 2.3-2. The general form of the sliding correlator measurement system receiver uses a mixer and a narrowband bandpass filter to correlate the received signal with a copy of the PN sequence produced at the transmitter.

The name "pseudonoise sequence" derives from the fact that many statistical characteristics, most importantly the autocorrelation function of the sequence, resemble those of random binary sequences [8]. True random binary sequences consist of statistically independent bits, each occurring in the sequence with probability one-half. Despite their many random-like characteristics, however, PN sequences are deterministic and periodic [9].

The noise-like characteristics PN sequences are of interest because of the theoretical ability to determine the impulse response of a radio channel using white noise. If $n(t)$ is defined as the input to an unknown channel, and $w(t)$ is the output of the channel, then impulse response of the channel can be determined by cross-correlating the output signal $w(t)$ with a delayed version of the input noise signal $n(t-t)$, where t is the time delay. The autocorrelation function of white noise is an impulse function [12], given by

$$R_n(t) = E\left[n(t)n^*(t-t)\right] = s^2 d(t) \qquad (2.3\text{-}1)$$

where s^2 is the noise power spectral density. If the noise signal $n(t)$ is the input to a channel with impulse response $h(t)$, then the channel output is the convolution of the noise signal and the channel response,

$$w(t) = \int h(x)n(t-x)dx \qquad (2.3\text{-}2)$$

The cross-correlation of the output signal with the input signal, delayed by t, is [11]

$$\begin{aligned} E\left[w(t)n^*(t-t)\right] &= E\left[\int h(x)n(t-x)n^*(t-t)dx\right] \\ &= \int h(x)E\left[n(t-x)n^*(t-t)\right]dx \qquad (2.3\text{-}3) \end{aligned}$$

If x is defined as

$$x = t - x \qquad (2.3\text{-}4)$$

then the expected value in (2.3-3) can be written as

$$\begin{aligned} E\left[n(t-x)n^*(t-t)\right] &= E\left[n(x)n^*(x-(t-x))\right] \\ &= R_n(t-x) \qquad (2.3\text{-}5) \end{aligned}$$

Therefore, the cross-correlation given by (2.3-3) can be expressed as

$$\int h(x)R_n(t-x)dx \qquad (2.3\text{-}6)$$

Finally, using (2.3-1) and (2.3-6) yields

$$\begin{aligned} \int h(x)R_n(t-x)dx &= s^2\int h(x)d(t-x)dx \\ &= s^2h(t) \qquad (2.3\text{-}7) \end{aligned}$$

Dividing by the noise power spectral density constant and performing the cross-correlation over a range of delay t provides the impulse response of the radio channel $h(t)$. Hence, the cross-correlation property of white noise is what allows the impulse response $h(t)$ to be determined.

Generating white noise and using it for channel sounding is impractical; however, the underlying theory of using the correlation property of white noise to measure channels is realizable if a pseudonoise signal with a similar correlation property is used instead of true white noise.

Pseudonoise sequences can be generated in hardware using a feedback shift register. An example of such a register is shown in Figure 2.3-3. Each stage of the shift register contains one binary bit of the register, and each bit except the one in stage *m* shifts to the following stage (*n*+1) with each clock pulse. The bit in stage *m* of the shift register is the output of the PN sequence generator, and corresponds to one chip of the PN sequence. The *c* coefficients assume a value of either zero or one, and determine whether or not a stage is included in the summation. The summation performed is modulo 2 addition, and the result of the summation becomes the stage-1 bit with each clock pulse. The block diagram in Figure 2.3-3 can be expressed as

$$S_n = c_{m-1}S_{n-1} \oplus c_{m-2}S_{n-1} \oplus \cdots \oplus c_0 S_{n-m} \qquad (2.3\text{-}8)$$

where m is the number of stages in the shift register and S_{n} is the output of the sequence generator at clock cycle *n*.

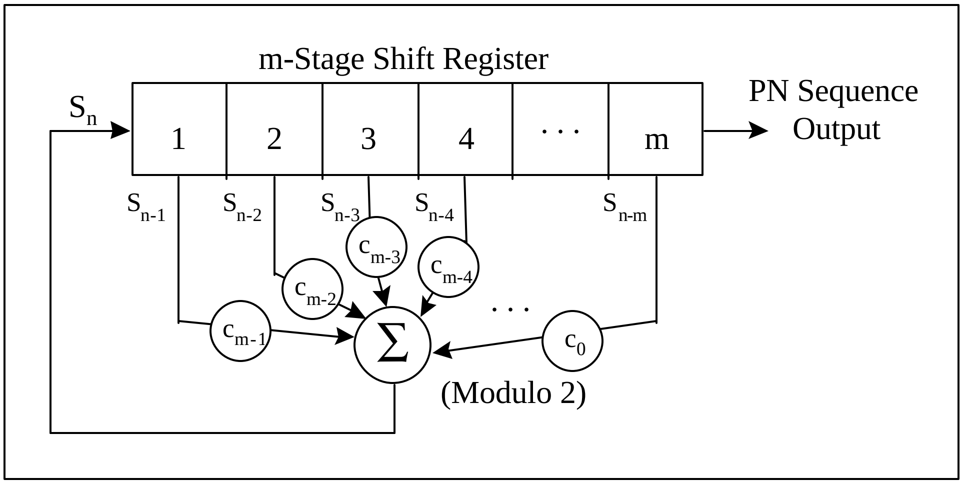


Figure 2.3-3. A linear feedback shift register is used to produce PN sequences for the sliding correlator measurement system.

The sequence produced by the generator modeled by equation (2.3-8) will be a maximal length sequence if

$$P_m(X) = X^m + c_{m-1}X^{m-1} + \cdots + c_1 X + c_0 \qquad (2.3\text{-}9)$$

is a primitive polynomial of degree m, where the c coefficients in equation (2.3-9) correspond to those in equation (2.3-8). If the sequence is of maximal length, then the autocorrelation function of the PN sequence is that shown in Figure 2.3-4, which is produced by evaluating

$$R_p(k) = \frac{1}{N}\sum_{n=1}^{N} Y(n)Y(n-k) \quad , \quad N = 2^m - 1 \qquad (2.3\text{-}10)$$

where $Y(n)$ represents samples of the PN sequence. Note that the autocorrelation function shown in Figure 2.3-4 is similar to the autocorrelation function of a random binary sequence shown in Figure 2.3-5 [9]. Also, note that the autocorrelation function produced by the PN sequence is periodic with period 2^m-1 chips.

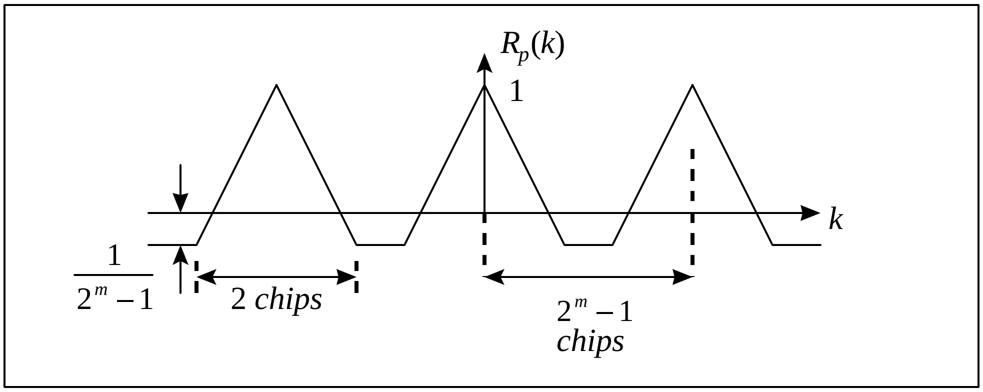


Figure 2.3-4. The autocorrelation function of a PN sequence is the key to using the sliding correlator system for measuring the multipath structure of radio channels.

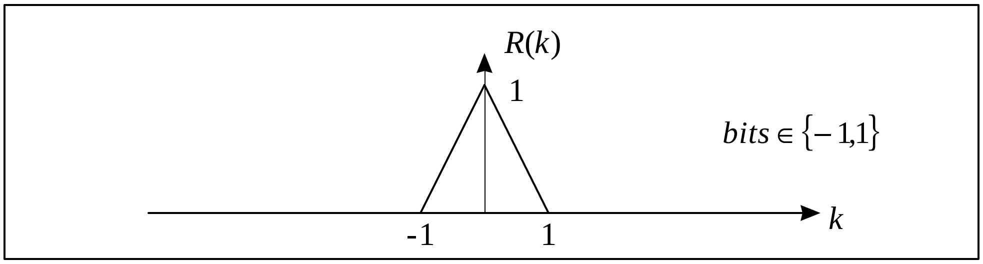


Figure 2.3-5. The autocorrelation of random binary sequence is not periodic like the autocorrelation of the PN sequence. The binary bits are assigned values of either -1 or 1.

A CW carrier modulated by a maximal length PN sequence will produce a transmitter output signal that has a power spectrum envelope given by

$$\left[\frac{\sin(f - f_c)T_c}{(f - f_c)T_c}\right]^2 \qquad (2.3\text{-}11)$$

where f_c is the frequency of the CW carrier and T_c is the period of a transmitter PN chip [10]. This spectrum envelope is recognized as a *sinc* function, and has the form of the Fourier transform of a CW carrier modulated by an RF pulse of duration T_c. The null-to-null bandwidth of the spectrum envelope is twice the chip rate, or

$$B_{RF} = 2R_c \qquad (2.3\text{-}12)$$

The receiver shown in Figure 2.3-2 receives the sounding signal and performs the correlation using a PN sequence generator, a mixer (multiplier), and a narrowband filter (integrator). The output of the narrowband filter is detected and displayed on an oscilloscope.

The sequence produced by the receiver PN sequence generator is clocked at a slightly slower rate compared to the rate of the transmitter PN clock. Thus, the receiver PN sequence as a whole appears to "slide" across the transmitter PN sequence, and hence the name "sliding correlator". Although the PN sequence produced at the receiver is clocked at a slightly slower rate than the transmitted PN sequence, the receiver and transmitter PN sequences themselves are exactly the same. Therefore, a cross-correlation of the transmitter and receiver PN sequences produces the autocorrelation function shown in Figure 2.3-4.

To understand how the sliding correlator functions, first assume that the sounding signal is not corrupted by multipath when received at the receiver. When the transmitter and receiver PN sequences are maximally aligned, the correlation is at a maximum and a voltage peak is produced at the output of the detector. When the sequences are not aligned, the correlation of the two sequences is at a minimum and the output voltage of the detector is very low compared to the peak voltage produced during alignment. In this way, the receiver produces an estimate of the cross-correlation function between the received PN sequence and the receiver PN sequence.

If the received signal is corrupted by multipath, then a delayed and attenuated version of the transmitted signal will exist for each physical propagation path. Then as the receiver PN sequence slides past the received signal, correlation peaks appear at the output of the detector as the receiver sequence aligns with each of the incoming multipath signals. The voltage waveform at the output of the detector represents the *power delay profile* of the channel.

The power delay profile is an estimate of the impulse response of a radio channel. If $R(t)$ is the autocorrelation function of the PN sequence, voltage at the output of the detector (see Figure 2.3-2), which represents the power delay profile, is proportional to

$$\left| R(t) \otimes h(t) \right|^2 \qquad (2.3\text{-}13)$$

where $h(t)$ is the impulse response of the radio channel and $\otimes$ represents the convolution operation. The power delay profile can also be thought of as the response of the radio channel to a triangular pulse (see [5]).

Several parameters affect the capabilities and performance of the sliding correlator measurement system (see [2]). The values to which these parameters are set depend upon the radio channel being tested and the requirements for the channel measurement data. Definitions of these parameters are shown in Table 2.3-1.

Table 2.3-1. Sliding Correlator Parameters.

Parameter	Symbol
Multipath Delay Resolution	Δt
RF Bandwidth	B_{RF}
Sequence length	l
PN Sequence Period	T_{PN}

Slip Rate	f_{slip}
Slide Factor	g
Baseband Bandwidth	B_{bb}

Multipath delay resolution, or simply resolution of the measurement system, is defined to be the minimum relative delay between two multipath components such that the components can be completely resolved in the power delay profile. The resolution of the system is determined by the transmitter chip rate, and is given by

$$\Delta t = \frac{2}{R_c} \tag{2.3-14}$$

Since the base of the correlation peak shown in Figure 2.3-4 is 2 chips ($2T_c=2/R_c$), multipath components separated by Δt are guaranteed to have no overlap in the power delay profile.

The RF bandwidth has also been shown to be related to the chip rate R_c in equation (2.3-12). In order to perform high-resolution measurements with a sliding correlator system, a large bandwidth must be used. PN sequence generator hardware speed limitations, practical RF filter design issues, and legal transmission bandwidth restrictions (FCC regulations) set the upper bound for RF bandwidth, and therefore place a limit on the best multipath resolution capability.

If the PN sequence is generated by a linear feedback shift register as shown in Figure 2.3-3, then the number of stages in the shift register determines the number of chips in one period of the sequence, l. Given a transmitter chip rate R_c, the period of the PN sequence T_{PN} is determined by

$$T_{PN} = \frac{l}{R_c} = lT_c \tag{2.3-15}$$

where T_c is the duration of a chip. The period T_{PN} has a direct relationship to the maximum unambiguous range of the system, which is given by

$$D_{ub} = cT_{PN} = \frac{lc}{R_c} = lT_c c \qquad (2.3\text{-}16)$$

where c is the speed of light, or more accurately, the speed of propagation of the sounding signal through the environment. The maximum unambiguous range is the maximum allowable difference, for satisfactory performance, between the length of the physical path of the first arriving signal component appearing at the receiver and length of the physical path of the last arriving multipath component appearing at the receiver. Physical paths which have larger length differences and which produce significant correlation peaks in the power delay profile will cause ambiguities in the measured impulse response.

The slip rate f_{slip} is the difference between the transmitter and receiver PN chip clocks, and is expressed as

$$f_{slip} = a - b \qquad (2.3\text{-}17)$$

The slip rate leads to the definition of the slide factor g, which provides the horizontal axis time scale for the power delay profiles. The slide factor g, defined as

$$g = \frac{a}{f_{slip}} = \frac{a}{a - b} \qquad (2.3\text{-}18)$$

relates actual propagation time to measurement observation time. The slide factor is the ratio of observed time to actual propagation time, and actual propagation time can be calculated from measurement data by

$$t_{actual} = \frac{t_{observed}}{g} \qquad (2.3\text{-}19)$$

This apparent dilation of time causes, for example, a delay of 100 ns to appear as 1 ms when a slide factor of 10000 is used for the measurement system.

The baseband bandwidth of the channel sounder is equal to the slip rate f_{slip} [2]. If detection is performed at the carrier frequency (or possibly at an intermediate frequency), then the bandwidth of the narrowband filter used for correlation must be twice the slip rate, or

$$B_{NB} = 2 f_{slip} = 2(a - b) \qquad (2.3\text{-}20)$$

Assuming an additive white Gaussian noise (AWGN) noise model at the receiver, the noise power at the detector is equal to the noise power spectral density integrated over the passband characteristic of the narrowband filter. The demodulation of a wideband signal using a narrowband filter provides an improved signal to noise ratio versus systems which use wideband filters throughout the receiver chain (e.g., RF pulse measurement system).

The minimum power required in order to detect a multipath component, relative to the power of the strongest signal component in a power delay profile, is determined by the length of the PN sequence. For a PN sequence with the n^{th} chip of l chips represented by

$$X(n) = (-1)^{S_n} \qquad (2.3\text{-}21)$$

where $S_n \in \{0,1\}$ and S_n defines a valid l-chip PN sequence, then the autocorrelation function $R(k)$ is

$$\begin{aligned} R(k) &= \frac{1}{l}\sum\nolimits_{n=1}^{l} X(n)X(n-k) \\ &= \frac{-1}{l} \quad \text{for} \quad n \neq k\,, \\ &= 1 \quad \text{for} \quad n = k \end{aligned} \qquad (2.3\text{-}22)$$

where l is number of chips in the PN sequence. The non-zero value when $n \neq k$ places a theoretical bound on the minimum detectable multipath component. This bound sets the theoretical dynamic range of the system, defined to be the ratio of the power level of the strongest received multipath component to the smallest discernible multipath component. Using equation (2.3-22), the dynamic range ratio is calculated as $1^2/(-1/l)^2$ which can be expressed as $10\log_{10}(l^2)$ dB.

Each of the described parameters is important to the performance of the sliding correlator measurement system. A discussion on assigning values to these parameters for successful wideband measurements is presented in Chapter 3.

2.4 Implementation of a Sliding Correlator Measurement System

Actual implementations of the basic sliding correlator system shown in Figure 2.3-1 and Figure 2.3-2 require additional hardware for triggering and data acquisition in the baseband section of the receiver, and require filtering and amplification in the RF sections of the transmitter and receiver. Shown in Figure 2.4-1 and Figure 2.4-2 are the block diagrams of the sliding correlator transmitter and receiver used for this research. The diagrams show the system as implemented at Virginia Tech, evolving over several years of research to achieve new frequency bands and easier measurements.

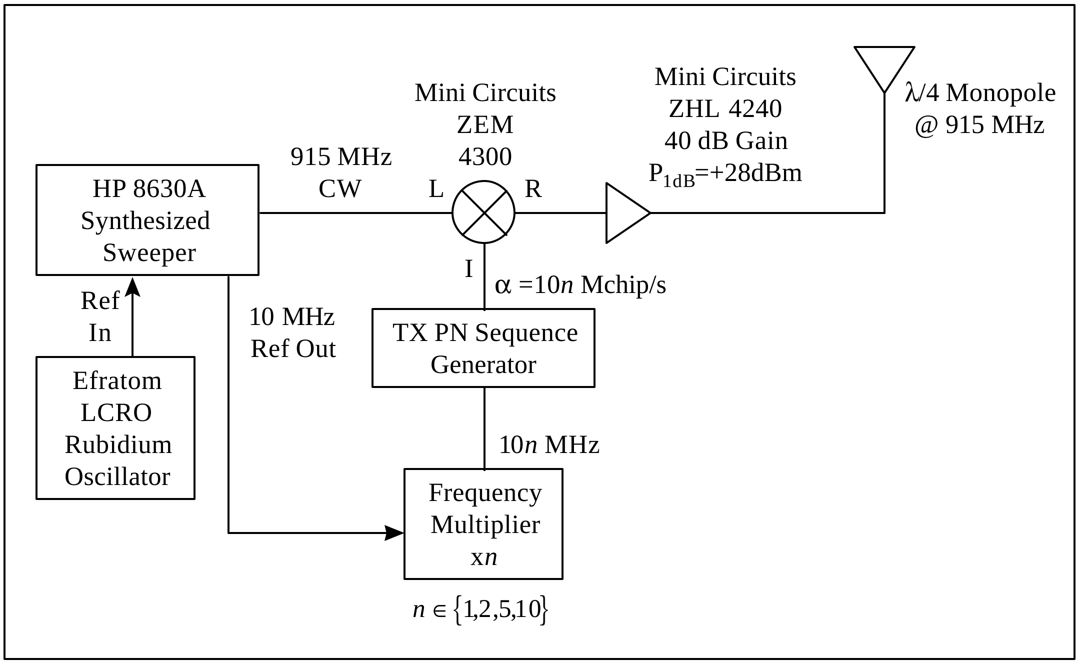


Figure 2.4-1. This implementation of a sliding correlator measurement system transmitter is configured to measure the impulse response of radio channels in the 915 MHz ISM band (see [19]).

The system as shown is configured for multipath measurements in the 915 MHz ISM (Industrial, Scientific, and Medical) band. The transmitter uses a signal generator to produce a sinusoidal carrier at the center frequency f_o of the radio channel band. Using a balanced mixer, the carrier is bi-phase modulated by a PN sequence of chip length l and clocked at transmitter chip rate a. The PN sequence clock rate is adjusted using the frequency multiplier, which multiplies the 10 MHz reference frequency of the signal generator to obtain the desired frequency. The modulated carrier (sounding signal) at the output of the mixer is amplified and transmitted through the channel.

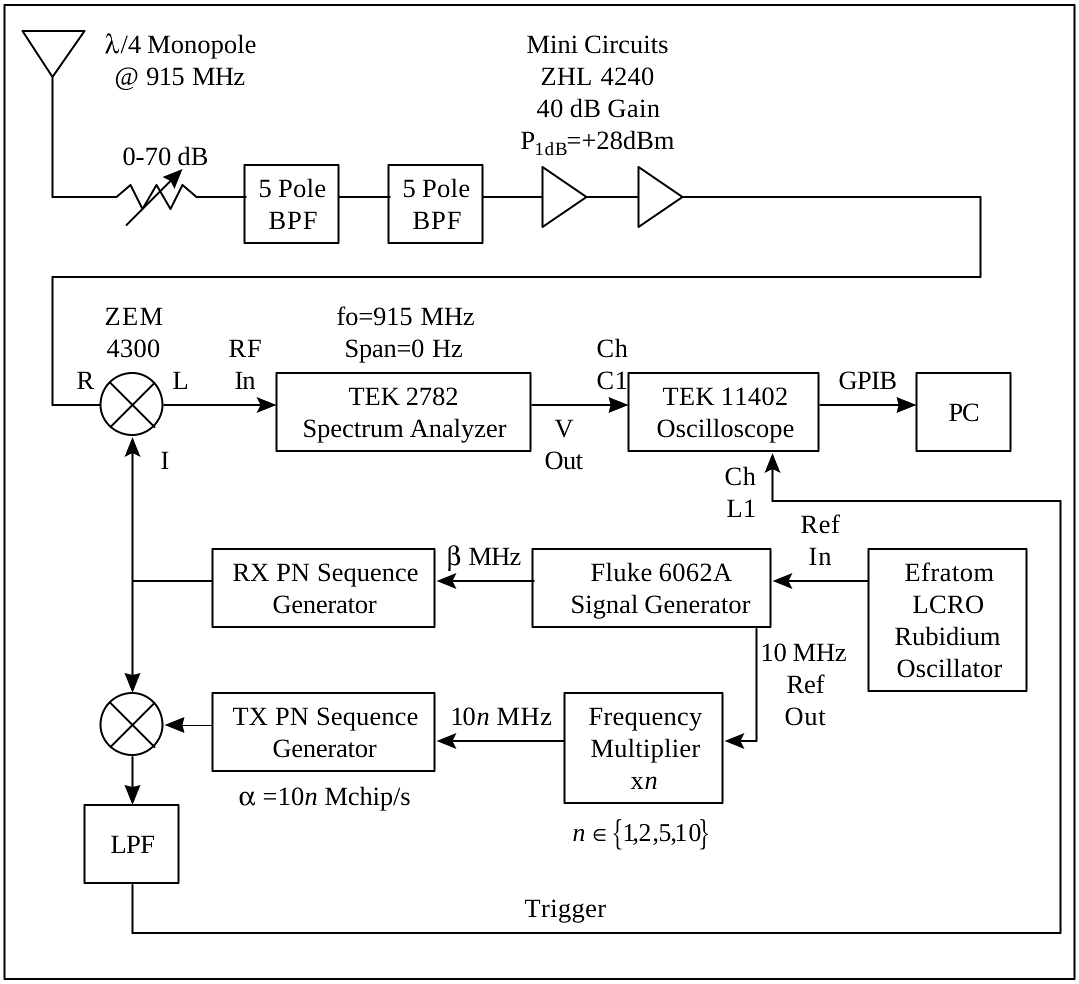


Figure 2.4-2. This implementation of a sliding correlator measurement system receiver is configured to measure the impulse response of radio channels in the 915 MHz ISM band (see [19]).

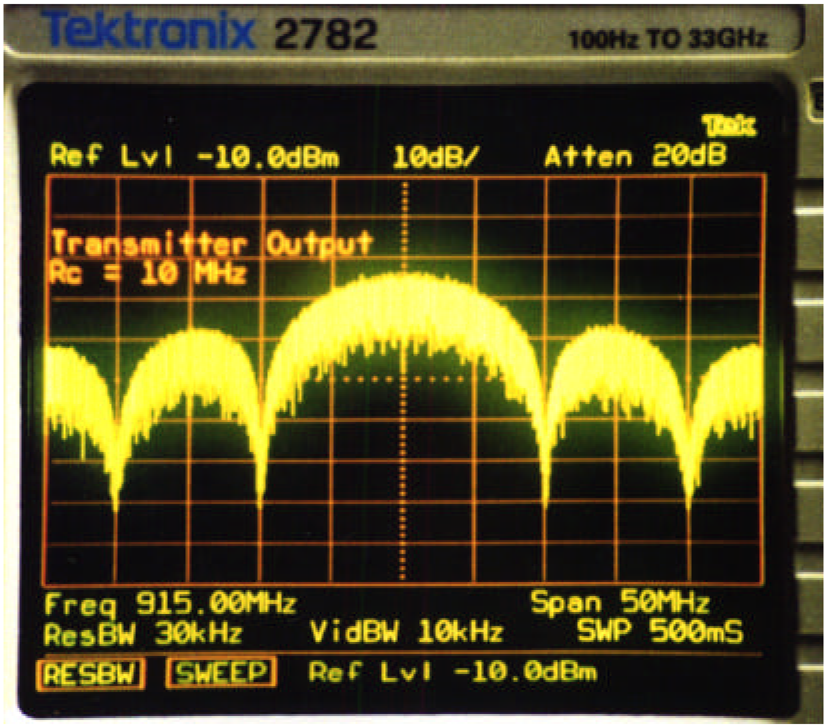
The receiver filters the signal from the antenna using a bandpass filter centered at f_o, and amplifies the signal using amplifiers with a flat passband across the received signal frequency band. The cross-correlation of the received signal, which is typically corrupted by multipath, with a copy of the uncorrupted transmitted sounding signal is performed using the mixer and spectrum analyzer. The cross-correlation is produced by mixing the received signal with a PN sequence identical to that at the transmitter, but clocked at a slightly slower rate. This slower PN sequence is produced by the "RX PN" sequence generator. The cross-correlation process is completed by bandpass filtering the output of the mixer with the spectrum analyzer using a resolution bandwidth equal to the correlation bandwidth and set at zero span. The vertical deflection of the spectrum analyzer display is shown on the oscilloscope and acquired via a GPIB bus using a PC. Table 2.4-1 shows the typical specifications of the system when used to sound radio channels in the 915 MHz band.

Table 2.4-1. Sliding Correlator Parameters (see [19]).

Parameter	Symbol	Value
Transmitter Chip Rate	a	10.00 MHz
Receiver Chip Rate	b	9.990 MHz
Multipath Delay Resolution	Dt	200 ns
RF Bandwidth	B_{RF}	20 MHz
Sequence length	L	2047 chips
PN Sequence Period	T_{PN}	204.7 µs
Slip Rate	f_{slip}	10 kHz
Slide Factor	g	1000
Baseband Bandwidth	B_{bb}	20 kHz
Theoretical Dynamic Range	M	66 dB

The spectrum of the transmitter output is shown in Figure 2.4-3. During acquisition of this output spectrum, the transmitter chip rate R_c was 10 MHz and the CW carrier frequency was 915 MHz. The envelope of the power spectrum shown in Figure 2.4-3 agrees with that expressed by equation (2.3-11). The null-to-null bandwidth of the spectrum is 20 MHz, and nulls occur at integer multiples of 10 MHz on either side of the carrier frequency.

The choice of antennas used for the both the transmitter and receiver depends upon the band being sounded. Generally, wideband omnidirectional antennas such as biconical antennas are used for high resolution, wide bandwidth, general propagation survey measurements. Resonant omnidirectional antennas such as $l/4$ monopoles may be used when the bandwidth of the transmitter output is a very small percentage of the center frequency of the band. Some specialized wideband measurements, such as direction-of-arrival (DOA) measurements where an angular sweep is performed, use a highly directional antenna at the receiver to resolve not only the delay of multipath components, but the angular spread of components in the received signal.



Ref Lvl -10.0dBm 10dB/ Atten 20dB

Freq 915.00MHz Span 50MHz

ResBW 30kHz VidBW 10kHz SWP 500mS

Ref Lvl -10.0dBm

Figure 2.4-3. The spectrum of the signal at the output of the sliding correlator transmitter has a null-to-null bandwidth of 20 MHz when using a chip rate of 10 MHz.

The second PN sequence generator, labeled “TX PN”, is used in this system to produce a time reference for triggering the oscilloscope. This PN sequence is also identical to the other two sequences in the system. Not only is this third PN sequence clocked at the same rate as the PN sequence at the transmitter, but it is also synchronized with the transmitter PN sequence. The transmitter PN sequence clock and the clock for the “TX PN” sequence generator at the receiver are held in sync by the rubidium oscillators. These oscillators, although a secondary frequency standard, are a highly stable source compared to the internal reference oscillators in the signal generators.

Before measurements are performed using this sliding correlator system, the two “TX PN” sequence generators are synchronized by starting both shift registers in the PN sequence generators at an initial state of $S_n = S_{n-1} = \ldots = S_{n-11} = 1$ (shift register stages equal to ‘11111111111’). At the receiver, the “RX PN” sequence (the slow sequence) and the “TX PN” sequence are correlated using a mixer and lowpass filter. The correlation peak produced when the sequences align serves as a time-zero reference, and corresponds to zero propagation time between the transmitter and receiver. Therefore, the absolute propagation delay of multipath components can be determined. Absolute delay information is useful for determining round-trip propagation delay for systems which require handshaking between the transmitter and receiver, and is also useful for estimating transmitter-receiver (T-R) separation when analyzing measurement data.

A sample power delay profile is shown in Figure 2.4-4. For this profile, the vertical axis is a power ratio expressed in decibels (dB) and is used to relate the strength of multipath signal components to the strength of the strongest multipath component. The horizontal axis is absolute propagation time; using propagation time relative to the first arriving signal component is also common. The large correlation peak following the first-arriving signal is caused by multipath. The low-level peaks 20 dB below the peak signal are caused by noise, and are not considered in the analysis of multipath in the profile.

Care must be taken when interpreting the strength of individual multipath components on the power delay profile. The strength of each component shown on the profile is actually only valid when the sequence for that component is maximally aligned with the sliding sequence at the receiver. That is, the strength of a component must be determined from its peak value in the profile. If the measurement system is calibrated such that the absolute power of one of the signal components is known, then the vertical axis of the profile may be expressed in dBm (or any other absolute power unit). The absolute power for a component also must be determined from the peak value of the component.

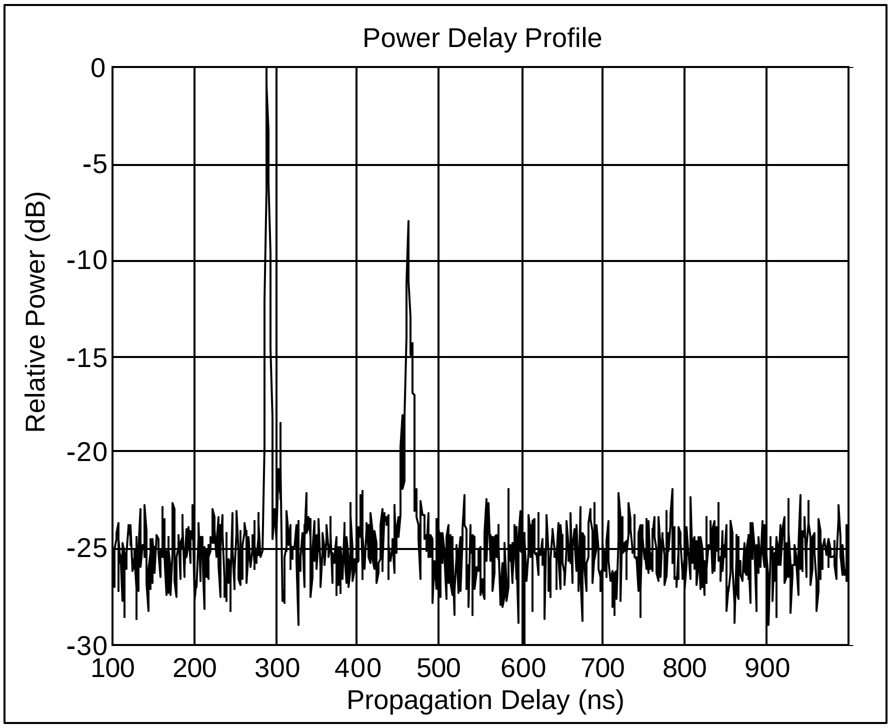


Figure 2.4-4. This power delay profile contains a line-of-sight component at 295 ns and one multipath component at 460 ns. The strength of the multipath component is approximately 8 dB below the strength of the line-of-sight component.

2.5 Analysis of Impulse Response Data

Two major motivations for performing wideband measurements are propagation surveys for the purpose of radio system development and deployment, and model validation for propagation prediction tools. Radio system development and deployment requires statistical analysis of propagation characteristics that affect system performance. Typically, a suite of measurements needs to be performed in the environment in which the radio system is to be deployed, and parameters need to be computed which quantitatively summarize the massive amount of recorded measurement data in order to specify the radio system. One example of such measurement campaigns is described in [15], during which channel impulse response measurements were performed for the initial development of wireless local loop system. Statistical quantities derived from wideband measurements in [3] were computed to assist in the deployment of GSM base stations.

Validation of radio wave propagation prediction tools require measurements for comparison to predicted results. For example, wideband measurements in [13] were used to compare measured power delay profiles to impulse responses predicted by tree-dimensional ray tracing.

Although propagation surveys for system deployment and measurements for model validation are often not mutually exclusive, they may require different forms of measurement data results. For instance, a ray tracing prediction tool may produce as an output an impulse response, showing individual received multipath components. In this case, it may be desirable to compare each predicted multipath component with measured components, and raw power delay profiles can be used for this comparison. Propagation surveys, however, require processing and analysis of the raw profiles to form parameters which statistically quantify propagation over regions, rather than over individual local point-to-point links.

The power of a multipath component can be determined from the peak value of the correlation peak corresponding to that multipath component in the power delay profile. In fact, the total received power, which contains contributions from all multipath components, can be determined for a particular profile. The total received signal of a radio channel in [5] was quantified by defining the average power transmission factor,

$$L = L_o \sum_{k=1}^{M} P(t_k) \qquad (2.5\text{-}1)$$

where L represents the sum of power delay profile samples at all delays t_k. If L_o is determined by calibration of the measurements system, then this is a measure of absolute signal strength.

Analysis of measurement data in [14] demonstrated not only the method of computing absolute wideband power, but also showed in the following way that average wideband power and average CW power are equivalent under certain conditions. If multipath components arrive at the receiver with real amplitude (not power) a_i and phase q_i, then the average received wideband power $E_{a,q}[P_w]$ is defined as the expected value of the sum of the individual multipath powers

$$E_{a,q}\left[P_w\right] = E_{a,q}\left[\sum_i \left|a_i \exp\left(jq_i\right)\right|^2\right] = \sum_i \overline{a_i^2} \qquad (2.5\text{-}2)$$

For CW measurements performed over a local area, the average received power $E_{a,q}[P_{cw}]$ is the expected value of the square of the sum of the signals at the receiver

$$E_{a,q}\left[P_{cw}\right] = E_{a,q}\left[\left|\sum_i a_i \exp\left(jq_i\right)\right|^2\right]$$

$$= \sum_i \overline{a_i^2} + 2\sum_i \sum_{j,j\neq i} r_{ij}\,\overline{\cos\left(q_i - q_j\right)} \qquad (2.5\text{-}3)$$

where r_{ij} is the correlation coefficient of the amplitudes of multipath component i and multipath component j. If the phases are identically and independently uniform distributed (*iid* uniform), then the *cos( )* factor is zero, and the average received power given by equations (2.5-2) and (2.5-3) become equal. The *iid* uniform assumption is reasonable since the differences in distances traversed by each multipath component are usually many wavelengths, and each multipath component is likely to arrive with arbitrary phase [16].

Equation 2.5-2 is used for calculating received power from discrete multipath components. Because multipath components in a power delay profile exist as correlation peaks with finite width, a method must be used which compensates for the shape of the multipath components in the power delay profile, and which does not assume an infinitely narrow impulse for each component.

Using calibration profiles created with the measurement system transmitter and receiver connected end-to-end, each calibration profile can be mapped to the known power at the input to the receiver used during acquisition of the calibration profile. If a calibration profile is produced with receiver input power P_{cal}, then the received power P_r associated with an arbitrary power delay profile (recorded using the same sliding correlator system parameters that were used to record the calibration profile) can be expressed as

$$P_r = P_{cal} \frac{\sum_k p_r(k)\Delta t_r}{\sum_k p_{cal}(k)\Delta t_{cal}} \qquad (2.5\text{-}4)$$

where $p_r(k)$ is the value of the received power delay profile at sample k, $p_{cal}(k)$ is the value of the calibration power delay profile at sample k, and Δt_r and Δt_{cal} are the time differences between samples of the received profile and calibration profile, respectively. The expression in (2.5-4) has the effect of normalizing the received profile to the area of a signal component recorded during calibration, and relating the ratio of areas under profiles to the ratio of received power in the profiles. Using this technique, the received power associated with any individual power delay profiles can be determined as long as an accurate calibration profile is available.

When characterizing path loss in a region for either indoor or outdoor radio channels, the log-distance path loss model is a common method of modeling path loss as a function of distance between a transmitter and receiver. This model is described by

$$\overline{L_P}(d) = \overline{L_P}(d_o)\left(\frac{d}{d_o}\right)^n \qquad (2.5\text{-}5)$$

which may be expressed logarithmically as

$$\overline{L_P}(d) = \overline{L_P}(d_o) + 10n\log_{10}\left(\frac{d}{d_o}\right) \quad \text{[dB]} \qquad (2.5\text{-}6)$$

where $\overline{L_p}(d)$ is average path loss at distance d, $\overline{L_p}(d_o)$ is the path loss at some close-in reference distance d_o, and n is the path loss exponent which describes the trend of path loss over distance. (Path loss units are absolute power in (2.5-5) and dB units in (2.5-6)). For free-space propagation, power falls off with the square of distance, implying n=2. Path loss exponents in real propagation environments increase depending upon obstructions in the environment, such as terrain, buildings, foliage, or walls. The path loss exponent also increases when the transmitter and receiver are separated far enough such that the ground exists in the first Fresnel zone (see [2]); beyond this T-R separation, power theoretically falls with n=4 for CW measurements.

The path loss exponent for a specific environment is experimentally determined using measurements. A large number of received power measurements are recorded throughout the environment at distances greater than the close-in reference distance d_o. Beyond reference distance d_o, a best-fit path loss exponent n is determined from the path loss data. Multiple path loss exponents can also be used to describe path loss by defining several distance "breakpoints" and calculating the best-fit exponent which describes the path loss trend between each pair of breakpoints.

Calculation of a path loss exponent does not require wideband measurements. A CW measurement system may be used to determine n by recording local average received power at many locations in the environment. However, time dispersion parameters (e.g., delay spread) cannot be calculated using CW measurements, and a wideband system must be used. Additionally, wideband systems may be used to determine the path loss exponent equivalent to that calculated using CW measurements. The reason for this is that received power calculated from a power delay profile and local average received power are equivalent if the phases of the received multipath component are *iid*, as previously shown.

The relative propagation delay multipath components causes time dispersion in the radio channel, resulting in a frequency selective fading. Frequency selective fading results in signal attenuation which is not only a function of time, but also varies with frequency and adversely affects wideband radio signals. Time dispersion and frequency selective fading cause intersymbol interference (ISI) in digital communication systems, resulting in an upper bound on symbol rate and an irreducible bit error rate (BER) floor [17].

Time dispersion caused by multipath is commonly quantified by two statistical moments [2], [11]: mean delay and RMS delay spread. Mean delay is the first moment of the power delay profile, given by

$$\bar{t} = \frac{\int tP(t)dt}{\int P(t)dt} \tag{2.5-7}$$

where $P(t)$ is the power delay profile (units of the ordinate are absolute power, not in dB). RMS delay spread is the square root of the second central moment of the power delay profile, given by

$$s_t = \sqrt{\overline{t^2} - (\bar{t})^2} \tag{2.5-8}$$

where

$$\overline{t^2} = \frac{\int t^2 P(t)dt}{\int P(t)dt} \tag{2.5-9}$$

For several modulation types, such as BPSK, QPSK, OQPSK, and MSK, average irreducible bit error rates have been shown to become larger than 10^{-3} when RMS delay spread exceeds 10% of the symbol period and equalization is not used [17]. Therefore, as a general rule, the maximum data rate for acceptable BER performance is limited to $0.1/s_t$ when using a receiver which does not employ an equalizer to combat the effects of frequency selective fading.

Another measure of time dispersion is maximum excess delay, sometimes called the delay interval, defined to be the difference in time delay points where the power delay profile rises above threshold P_I for the first time and falls below threshold P_I for the last time. The threshold P_I is set relative to the maximum value of the power delay profile; for example, the threshold may be set 10 dB below the largest peak to find the duration of time over which the received power shown in the power delay profile rises to within 10 dB of the peak power. If t_1 is the time at which the power rises above the threshold, and t_3 is the time that the power falls below the threshold for the last time (see Figure 2.5-1), then the delay interval is

$$I_P = t_3 - t_1 \tag{2.5-10}$$

The delay window W_q is also a measure of time dispersion and can be calculated directly from power delay profiles [11]. The delay window is the duration of the profile which contains q% of the total energy in the profile. The window is defined to be

$$W_q = (t_4 - t_2) \tag{2.5-11}$$

where t_4 and t_2 are determined by

$$\int_{t_2}^{t_4} P(t)dt = \frac{q}{100}\int_{t_0}^{t_5} P(t)dt \tag{2.5-12}$$

under the condition that the power contained in the region of the profile before the delay window is equal to the power contained in the region of the profile following the delay window.

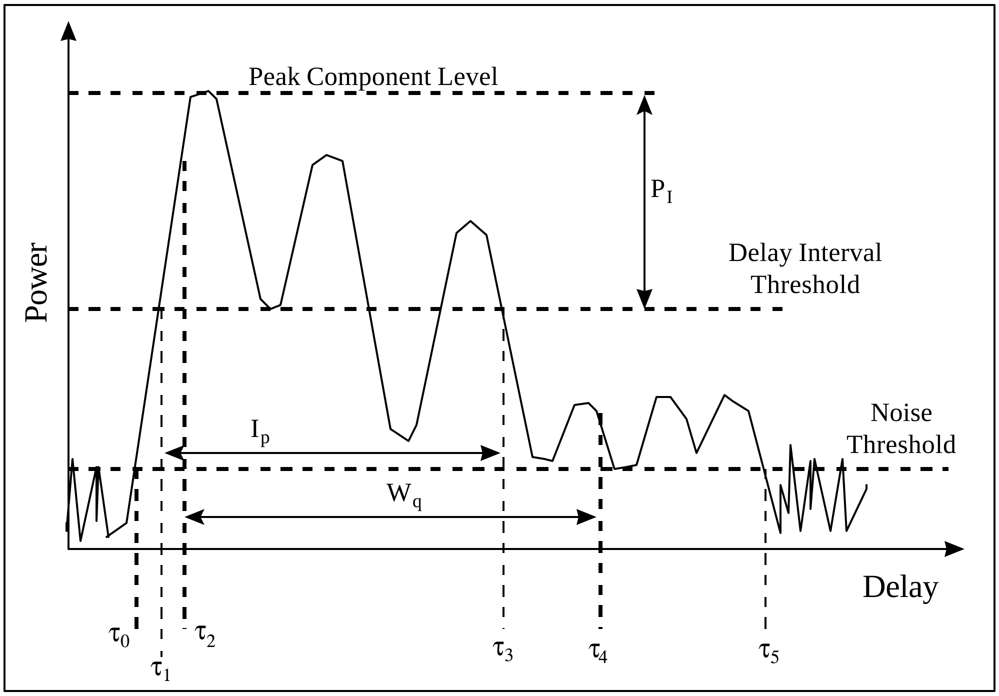


Figure 2.5-1. This drawing of a power delay profile defines the time delay and threshold parameters used to calculate delay interval and delay window.

Each of the time dispersion parameters mentioned in this section provide a measure of time dispersion in a radio channel and are parameters relevant to design, but exactly how to use these parameters to accurately predict system performance is still a topic of research and has been for many years [11], [17].

2.6 Summary

Multipath is caused by reflection, diffraction, and scattering of electromagnetic waves by objects in an environment. High data rates are adversely affected by multipath by inducing frequency-selective fading causing intersymbol interference and reducing the maximum symbol rate for a given error rate.

The sliding correlator measurement system is a spread spectrum system which can be used to measure impulse responses of radio channels. The output of a sliding correlator measurement system is a power delay profile, which shows the delay and power of individual multipath components. Using these profiles, path loss and time dispersion parameters can be calculated which characterize the propagation environment and can be used for wireless system design. Wideband measurement data is also useful for validating propagation models such as ray tracing.

A sliding correlator system has been implemented to sound channels up to 200 MHz wide. The system is configurable to sound many RF bands, and measurement applications such as angle of arrival measurements are also possible.

The following chapters describe data processing and results for a wideband measurement study, sliding correlator simulation for producing realistic power delay profiles, and post-processing techniques for improved accuracy of measurement statistics.

Chapter 3

A Propagation Time Delay and Path Loss Study

Accurate wideband propagation measurements provide the channel data needed to design a digital wireless communication system. Measured channel impulse response data aids the system designer in building simulations of the wireless system, and time delay and path loss statistics derived from wideband propagation studies assist in predicting system performance (e.g., [17]). A high quality measurement survey yields large amounts of well-documented data which can be used for simulation of radio channels and validation of propagation models for present and future propagation research projects.

This chapter describes a wideband propagation study performed to characterize path loss and time dispersion for a digital communication system within a rail yard. Presented first is a description of the measurement sites that were chosen to represent realistic situations for propagation for an intra-yard communication system. A description of the physical layout of the measurement area is given, along with a description of the preliminary site survey performed to measure RF interference in the band of interest. Next, the on-site measurements are presented, including a detailed discussion on the calibration of the sliding correlator measurement system. Finally, analysis of the channel impulse response data is described, and channel characterization results for the rail yard are presented.

3.1 Measurement Plan and Site Survey

The objective of this measurement survey was to characterize a typical rail yard environment with RMS delay spread and path loss parameters. The site which was selected for measurements was the Norfolk Southern yard in Roanoke, Virginia. This rail yard is busy compared to other rail yards and is representative of most large yards in the United States. The rails at this yard spanned as wide as 54 tracks, extending several miles through the city of Roanoke, and the tracks were surrounded by both industrial and rural areas. Over 100 trains passed through each day, an excellent environment for performing measurements during varying yard conditions.

Rail car tracking within this and many other yards is accomplished using passive RF "tags" mounted on the side of the rail cars [18]. The tags are read as the cars pass fixed points along the tracks in the yard. The disadvantage of this system is that rail cars must pass these fixed points in order to be identified and tracked. Tags on stored cars in the yard, resting still on side tracks, can not be queried for identification. An upgraded system using RF transponders attached to the rail cars would allow cars to be queried anywhere within the yard. Current voice communications within this and other rail yards is accomplished with FM transceivers using a very limited number of channels. This system also had a strong potential for upgrade.

The proposed band for a new radio communication system within the yard was the 902 MHz to 928 MHz ISM band. Unlicensed 1W spread spectrum transmitters can operate within this band with certain restrictions. Propagation measurements were warranted because of the unique environment, consisting of large highly reflective obstructions (rail cars and metal storage buildings), some of which remained stationary and some of which moved throughout the site.

The propagation study described here was designed to measure the variant nature of the rail yard. Trains moving throughout the yard cause fluctuations in received power and variations in time dispersive properties. That is, while holding the transmitter and receiver locations constant, the impulse response of the channel was expected to vary greatly due to the motion of the obstructions because of their size and material content (e.g., metal rail car walls). Three measurement sites were used to measure and statistically quantify these changes in the channel.

Three measurements sites were designated within the yard to represent three possible situations: close-in track-to-track communications, parallel-track communications, and cross-track communications. The measurement sites were designated Site One, Site Two, and Site Three. Site One was an area near the perimeter of the tracks near the area where rail cars entered the yard (called the receiving end). At this site, a boxcar, a gondola car, and a caboose rested stationary on the tracks at an area of the yard which spanned 27 tracks wide and was active with incoming trains. This site was used to measure delay spread for relatively short radio links using a transmitter-receiver (T-R) separation of approximately 17 meters. Figure 3.1-1 shows a diagram of the measurement site, and Figure 3.1-2 shows the rail cars at the site.

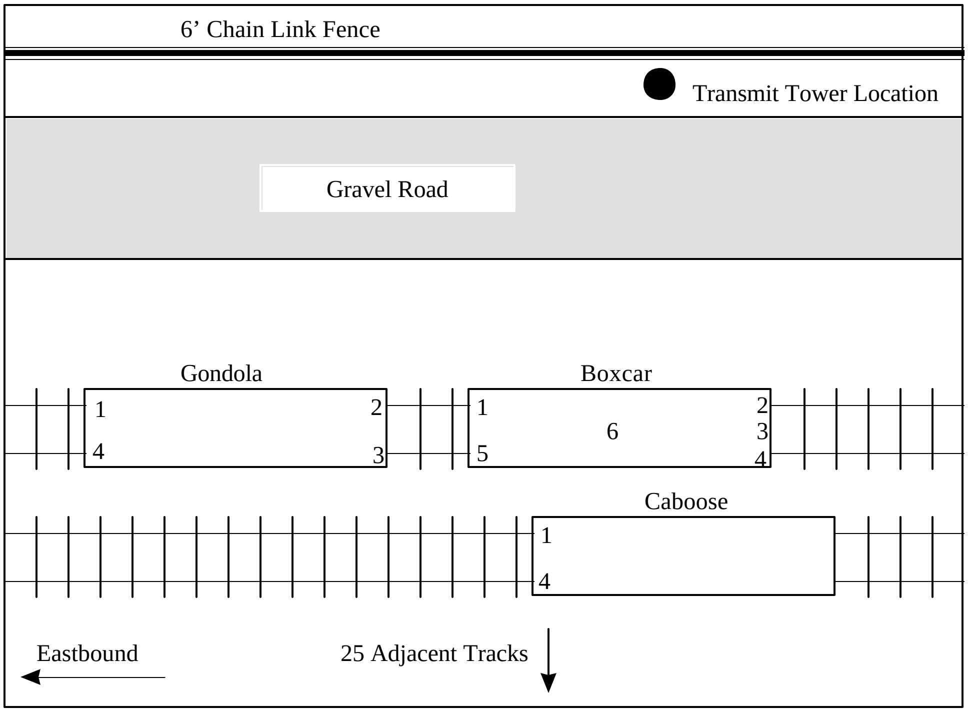


Figure 3.1-1. Measurement Site One was an area near the perimeter of the rail yard where three rail cars remained stationary on the tracks. This site was used to measure RMS delay spread using relatively close-in measurements.

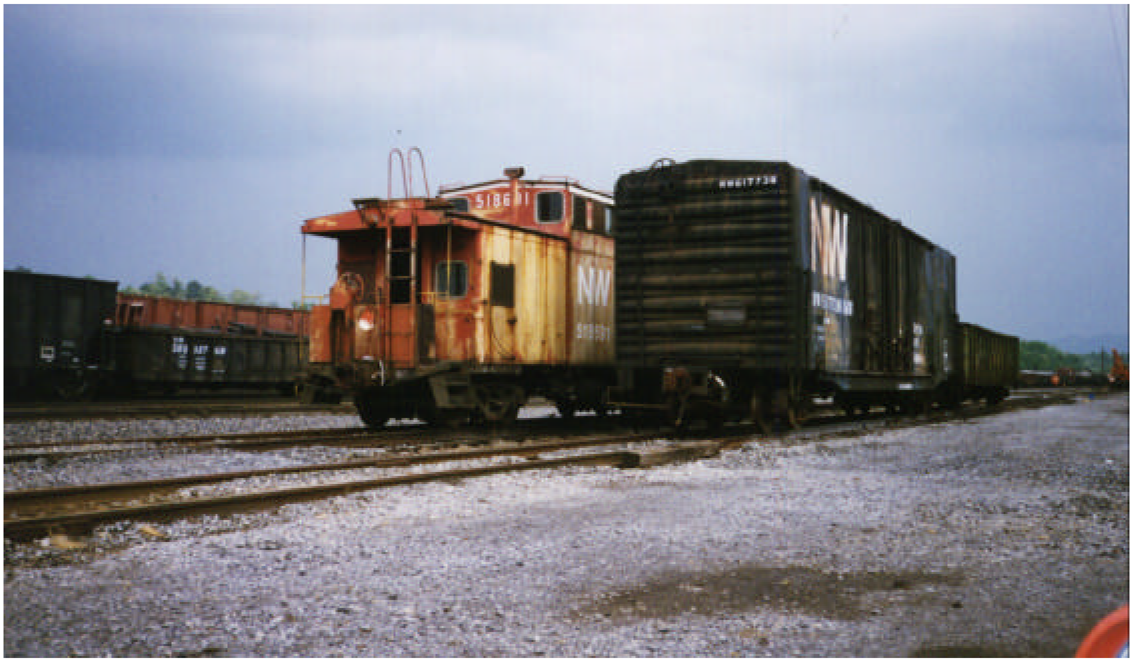


Figure 3.1-2. The receiver antenna was mounted at several locations on these rail cars at Site One. The 25 adjacent tracks were active with trains.

The measurement system transmitter antenna was a λ/4 monopole antenna mounted on a 6.1 meter tower at the location shown in Figure 3.1-1. The numbers shown in the rail car diagrams in Figure 3.1-1 designate receiver antenna locations. At the antenna locations along the perimeter of the rail car diagram, the receiver λ/4 monopole antenna was magnetically mounted at four different positions on the rail car: 1) the *top* position, where the antenna was mounted vertically on the rail car roof within three inches of the roof perimeter; 2) the *top-side* position, where the antenna was mounted vertically on the outside wall of the rail car, just under the roof level, using a right-angle magnetic mount; 3) the *side* position, where the antenna was mounted horizontally on the rail car wall just under the roof level; and 4) the *6-foot* position, where the antenna was mounted vertically on the wall of the rail car six feet above track level.

Site Two was an area extending from the rail cars used at Site One toward the west end of the yard. Using a T-R separation of approximately 190 meters, measurements at this site were performed to record channel impulse response data for a parallel-track site as trains moved throughout the rail yard. The transmitter antenna was mounted on the 6.1 meter tower, and the receiver antenna was positioned at points on the boxcar and caboose. Figure 3.1-3 illustrates the view from the top of the boxcar toward the sliding correlator transmitter at the receiving end of the yard.

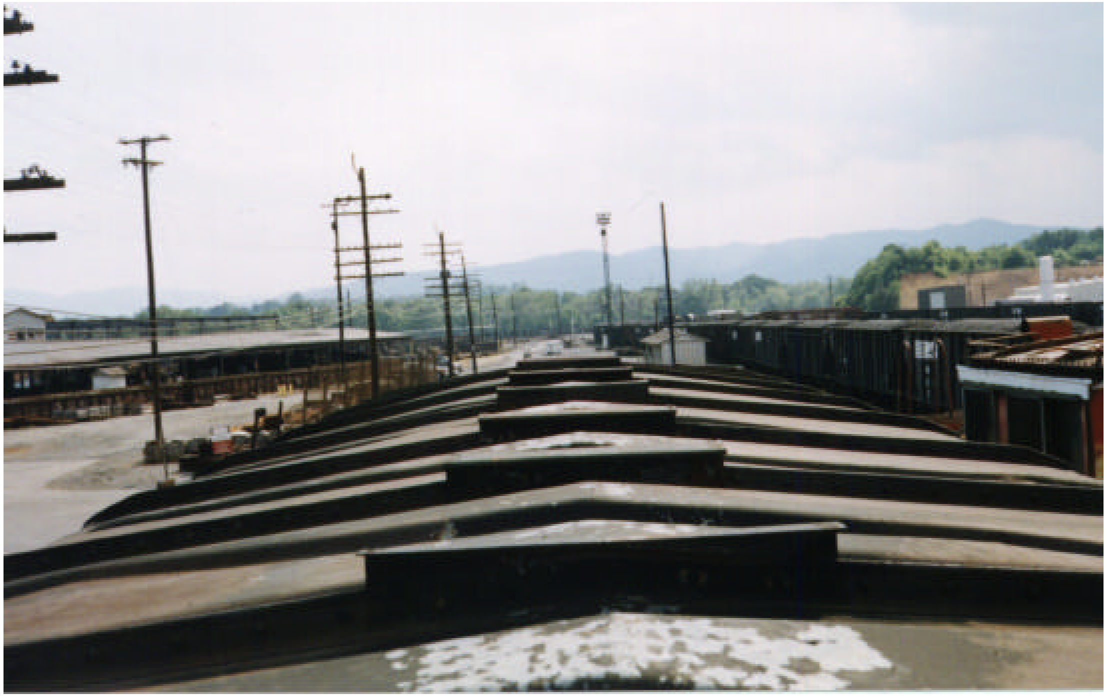


Figure 3.1-3. The receiver antenna was mounted on top of this boxcar and the adjacent caboose for Site Two measurements. The transmit antenna was mounted on the tower 190 meters down-track from the receiver.

Site Three was a region of the yard where tracks were closely spaced and train activity was the busiest. This site was selected to measure power delay profiles for a link perpendicular to the tracks and across the widest part of the train yard. The transmitter antenna was attached to the 6.1 meter tower and mounted on a hill on the north side of the rail yard. Rail cars on the south side of the yard were used for receiver antenna locations, and the T-R separation was approximately 230 meters. Figure 3.1-4 illustrates the view from the transmitter antenna toward the receiver location.

In addition to surveying the physical layout of the rail yard for measurement sites, the RF spectrum at the site was measured. Interfering narrowband signals within the passband of the sliding correlator receiver increase the noise level in the power delay profile and reduce the ability of the receiver to detect low level multipath signal components. A spectrum analyzer was used to measure the interference at the rail yard prior to performing measurements to insure that no continuous narrowband interferers existed in the 902 MHz to 928 MHz band. The ability of the front-end filter of the sliding correlator to attenuate the existing out-of-band signals was also measured. Figure 3.1-5 shows the equipment used to measure the interference at the site.

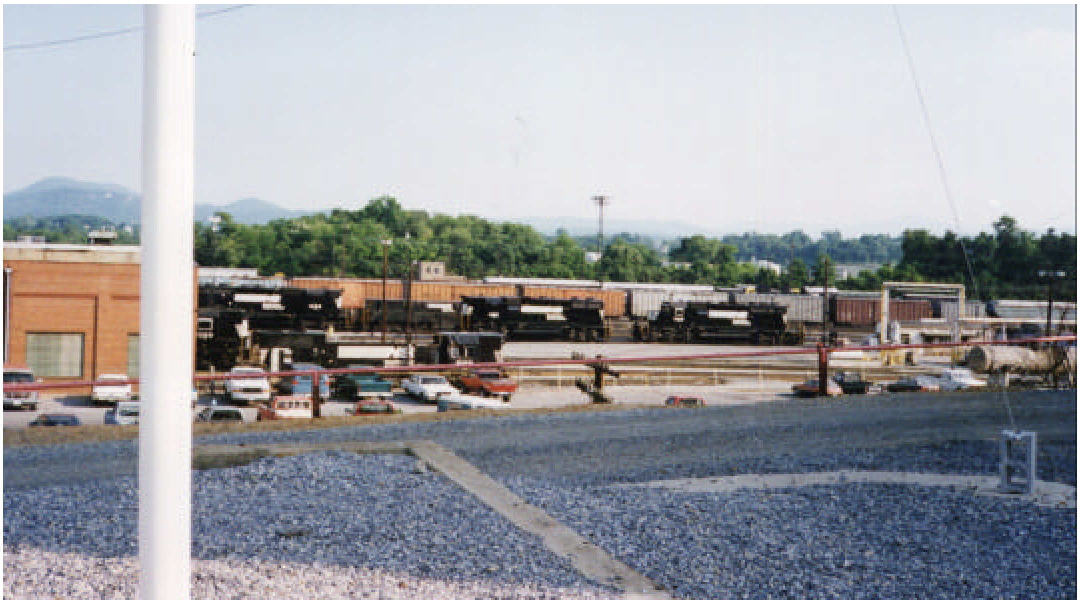


Figure 3.1-4. From the top of the hill, the transmitter antenna had a view of the entire yard. This part of the yard was cluttered with active trains, metal and brick buildings, and other heavy equipment.

A $\lambda/4$ monopole was used as the receiving antenna, and two amplifiers provided a total gain of 65 dB. Figure 3.1-6 shows the received spectrum as recorded by the spectrum analyzer. The spectrum shows that strong interferers are present outside of the ISM band (902 MHz to 928 MHz). The narrowband signals above 928 MHz are likely to be signals from a nearby paging network, and signals below 894 MHz are most likely narrowband cellular telephone signals. Figure 3.1-7 shows the received spectrum with the measurement system front-end filter (see Figure 2.4-2) inserted between the antenna and the first amplifier stage. This figure shows that the interfering signals are attenuated by more than 40 dB. The attenuation of the interferers by the front-end filter in addition to the spreading gain of the sliding correlator resulted in satisfactory noise levels in the power delay profiles.

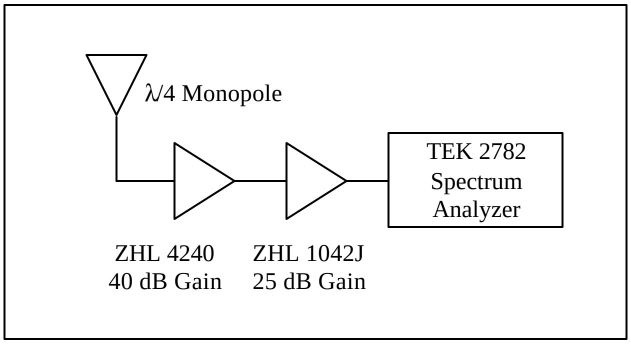
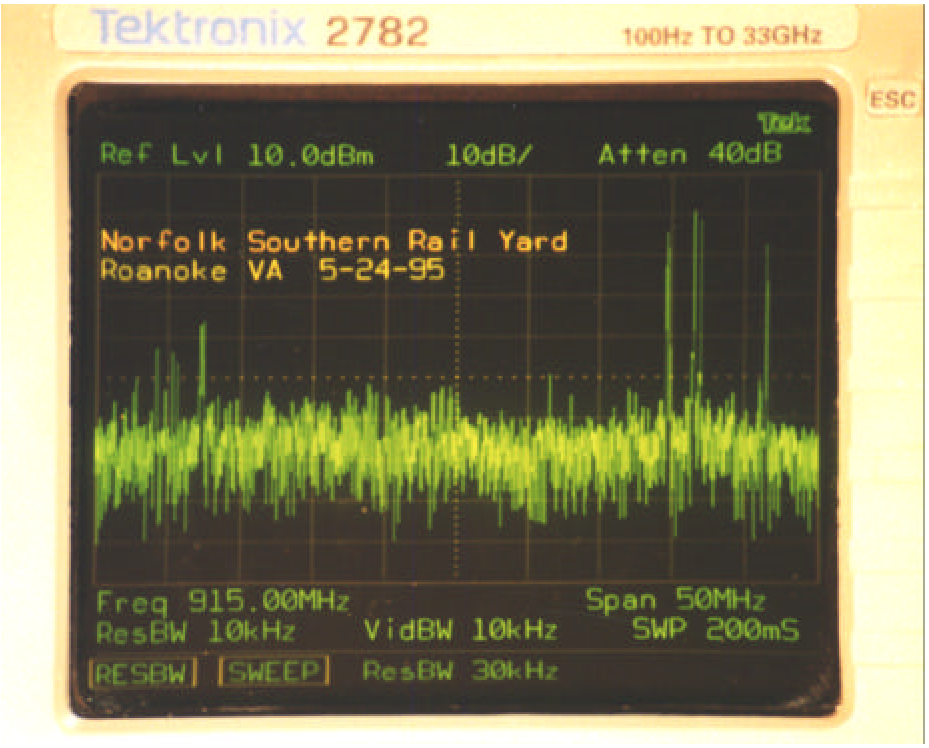


Figure 3.1-5. Two RF amplifiers, a monopole antenna, and a spectrum analyzer were used to measure signals which could interfere with the wideband measurements.



Ref Lvl 10.0dBm 10dB/ Atten 40dB

Norfolk Southern Rail Yard
Roanoke VA 5-24-95

Freq 915.00MHz Span 50MHz
ResBW 10kHz VidBW 10kHz SWP 200mS
[RESBW] [SWEEP] ResBW 30kHz

Figure 3.1-6. The unfiltered received signal shows strong narrowband interferers at the high end of the band. These interferers are likely to be signals transmitted by a paging system.

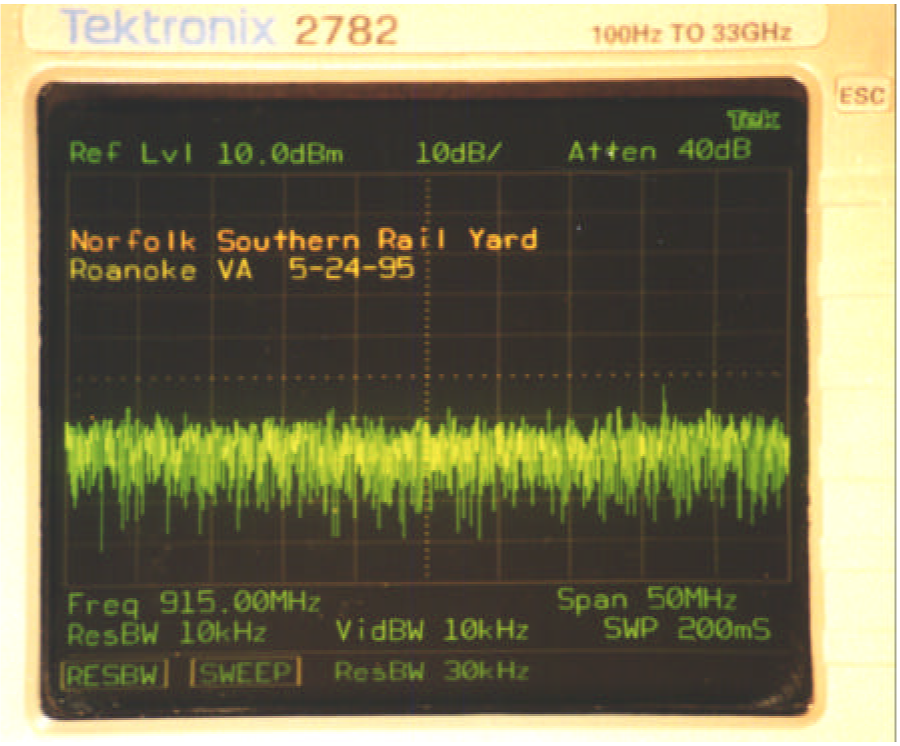


Figure 3.1-7. The filtered RF spectrum shows that the interferers have been reduced to acceptable levels.

3.2 Calibration and Measurement Details

The implementation of the sliding correlator measurement system discussed in Chapter 2 was used for the rail yard measurements. Table 3.2-1 shows the parameter settings for this configuration of the system. A transmitter chip rate of 10 MHz produced a spectrum at the output of the transmitter with a null-to-null bandwidth of 20 MHz, slightly less than the 26 MHz bandwidth of the ISM band. By using a 10 MHz chip rate, multipath components could be completely resolved when separated by more than 200 ns. A receiver chip rate of 9.99 MHz was used to produce a slide factor of 1000, implying that 1 μs of propagation time delay appeared as 1 ms of delay on the oscilloscope screen.

All of the results derived from wideband measurement data are highly dependent upon the calibration of the sliding correlator measurement system. Changes in temperature, the use of different cables, and even slow time variations within the equipment cause deviations which affect the precision and accuracy of the measurement system. Periodic calibration of the system during a measurement survey corrects for this system drift.

Table 3.2-1. These sliding correlator parameters were used to measure the channel impulse response data at the rail yard sites. The parameters are related as described in Chapter 2.

Parameter	Symbol	Value
Transmitter Chip Rate	a	10.00 MHz
Receiver Chip Rate	b	9.990 MHz
Multipath Delay Resolution	Dt	200 ns
RF Bandwidth	B_{RF}	20 MHz
Sequence length	l	2047 chips
PN Sequence Period	T_{PN}	204.7 µs
Slip Rate	f_{slip}	10 kHz
Slide Factor	g	1000
Baseband Bandwidth	B_{bb}	20 kHz

Calibration of the sliding correlator measurement system results in an accurate map of received signal component power to oscilloscope voltage. Figure 3.2-1 illustrates a model of the measurement system receiver used to describe the calibration procedure. The spectrum analyzer is modeled as a narrowband filter followed by a power detector and a display which plots power versus time with the spectrum analyzer set at zero span. The center frequency of the narrowband filter is the center of the measurement band, and the bandwidth of the filter is $2f_{slip}$, set by the resolution bandwidth of the spectrum analyzer. The correlator block consists of the mixer and the narrowband filter.

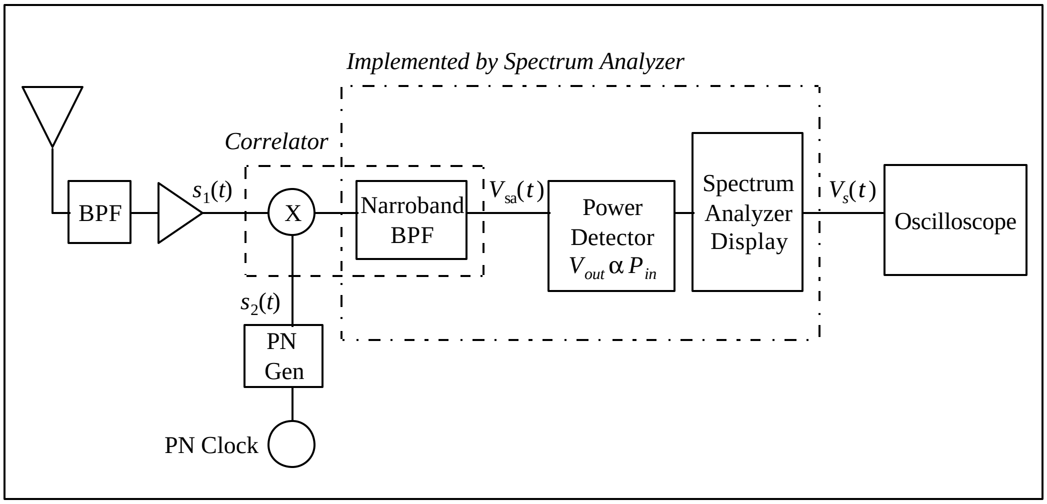

Implemented by Spectrum Analyzer
Correlator
BPF
$s_1(t)$
X
Narroband BPF
$V_{sa}(t)$
Power Detector $V_{out} \alpha P_{in}$
Spectrum Analyzer Display
$V_s(t)$
Oscilloscope
$s_2(t)$
PN Gen
PN Clock

Figure 3.2-1. This model of the measurement system receiver is used to describe the measurement system calibration procedure. The spectrum analyzer is modeled as a narrowband filter, a power detector, and a display. The correlator is modeled as a mixer and narrowband filter.

The power delay profile is viewed on and acquired from an oscilloscope, which displays voltage versus time on a linear scale. Note that the input voltage to the oscilloscope is the vertical output of the spectrum analyzer, which is a voltage proportional to the vertical deflection of the trace shown on the spectrum analyzer display. Because the spectrum analyzer trace is displayed on a logarithmic scale, the voltage shown on the oscilloscope is actually proportional to the log of the power detected by the spectrum analyzer (similar to the relationship between the input and output of a log amplifier). If $V_s(t)$ is the voltage at the oscilloscope input corresponding to a delay t as shown in a power delay profile, and $P_{sa}(t)$ is the power displayed on the spectrum analyzer screen, then

$$V_s(t) \propto 10\log(P_{sa}(t)) \qquad (3.2\text{-}1)$$

A plot of the waveform $V_s(t)$ versus t represents the power delay profile on the oscilloscope screen. Equation (3.2-1) can be expressed as

$$V_s(t) = V_o + C \cdot 10\log(P_{sa}(t)) \qquad (3.2\text{-}2)$$

where V_o is a voltage offset corresponding to an spectrum analyzer input of 0 dBm, and C is a proportionality constant in units of V/dB. The voltage $V_{sa}(t)$ produced by the mixer and narrowband filter at the output of the correlator block is given by

$$V_{sa}(t) = \int s_1(x)s_2(t+x)dx \qquad (3.2\text{-}3)$$

Now suppose that the received signal is increased in strength such that the input to the correlator is $s_1'(t) = as_1(t)$. Then the resulting $V_{sa}'(t)$ is given by

$$V_{sa}'(t) = \int s_1'(x)s_2(t+x)dx = \int as_1(x)s_2(t+x)dx$$

$$= a\int s_1(x)s_2(t+x)dx = aV_{sa}(t) \qquad (3.2\text{-}4)$$

Therefore, multiplying $s_1(t)$ by the constant a results in an increase of the voltage $V_{sa}(t)$ by a, and

$$\frac{s_1'(t)}{s_1(t)} = \frac{V_{sa}'(t)}{V_{sa}(t)} \qquad (3.2\text{-}5)$$

If G is the power gain which relates $s_1(t)$ to $s_1'(t)$ by $s_1'(t) = \sqrt{G}s_1(t)$, then by equation (3.2-5), $V_{sa}(t)$ and $V_{sa}'(t)$ are related by

$$V_{sa}'(t) = \sqrt{G}V_{sa}(t) \qquad (3.2\text{-}6)$$

The powers $P_{sa}(t)$ and $P_{sa}'(t)$ displayed as a function of delay t on the spectrum analyzer, corresponding to the voltages $V_{sa}(t)$ and $V_{sa}'(t)$ respectively, are then related by

$$P_{sa}'(t) = GP_{sa}(t) \qquad (3.2\text{-}7)$$

Using equation (3.2-2), the voltage $V_{sa}'(t)$ displayed on the oscilloscope are given by

$$V_s'(t) = V_o + C\cdot 10\log(P_{sa}'(t))$$

$$= V_o + C\cdot 10\log(GP_{sa}(t))$$

$$= V_o + C \cdot 10\log(P_{sa}(t)) + C \cdot 10\log(G) \qquad (3.2\text{-}8)$$

Combining equations (3.2-2) and (3.2-8) forms a relationship between $V_s(t)$ and $V_s'(t)$, given by

$$V_s'(t) = V_s(t) + C \cdot 10\log(G) \qquad (3.2\text{-}9)$$

Equation (3.2-9) demonstrates that an increase by G of the received power at the antenna port of the measurement system receiver results in a $C \cdot 10\log(G)$ increase in voltage at the oscilloscope. The quantity $10\log(G)$ is the power increase in dB, and the oscilloscope voltage is shown to vary linearly with the received power expressed in dB. This means that an linear increase in received power results in a linear upward shift of a signal component on the oscilloscope, and C is the proportionality constant in units of V/dB. By defining $P_r(t)$ as the power delay profile referenced to the antenna port in units of logarithmic power versus delay t, then the relationship between $P_r(t)$ and the oscilloscope voltage $V_s(t)$ can be expressed as

$$P_r(t) = P_o + DV_s(t) \quad \text{[dBm]} \qquad (3.2\text{-}10)$$

where D=1/C is the proportionality constant in units of dB/V and P_o is the received power of a signal component expressed in dBm which causes a 0 V peak on the oscilloscope. Equation (3.2-10) is the calibration equation that is used to express power delay profiles on a power versus delay plot. The values of P_o and D are determined empirically using calibration power delay profiles.

For a fixed configuration of the sliding correlator measurement system, the coefficient D and the offset P_o can be determined in the field by connecting the transmitter and receiver end-to-end through an attenuator as shown in Figure 3.2-2. The calibrated attenuator provides a known, controllable, single-path channel between the transmitter and receiver. Assuming no reflections exist in the T-R link (i.e., the transmitter, receiver, attenuator, and any connecting cables are impedance-matched), then only a single correlation peak exists in the power delay profile as shown in Figure 3.2-3. Since the transmitter power, attenuator value, and measured cable losses are known, the peak voltage of the signal component can be mapped to the power into the receiver.

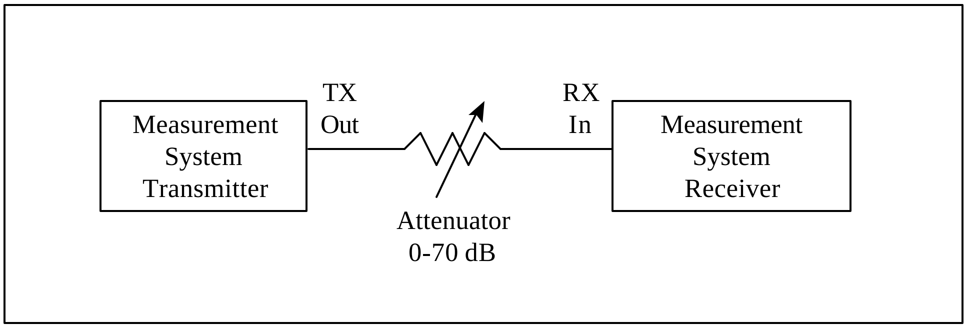


Figure 3.2-2. The transmitter and receiver are connected end-to-end through an attenuator to record calibration profiles. The calibration profiles are used to calculate the D and P_o constants.

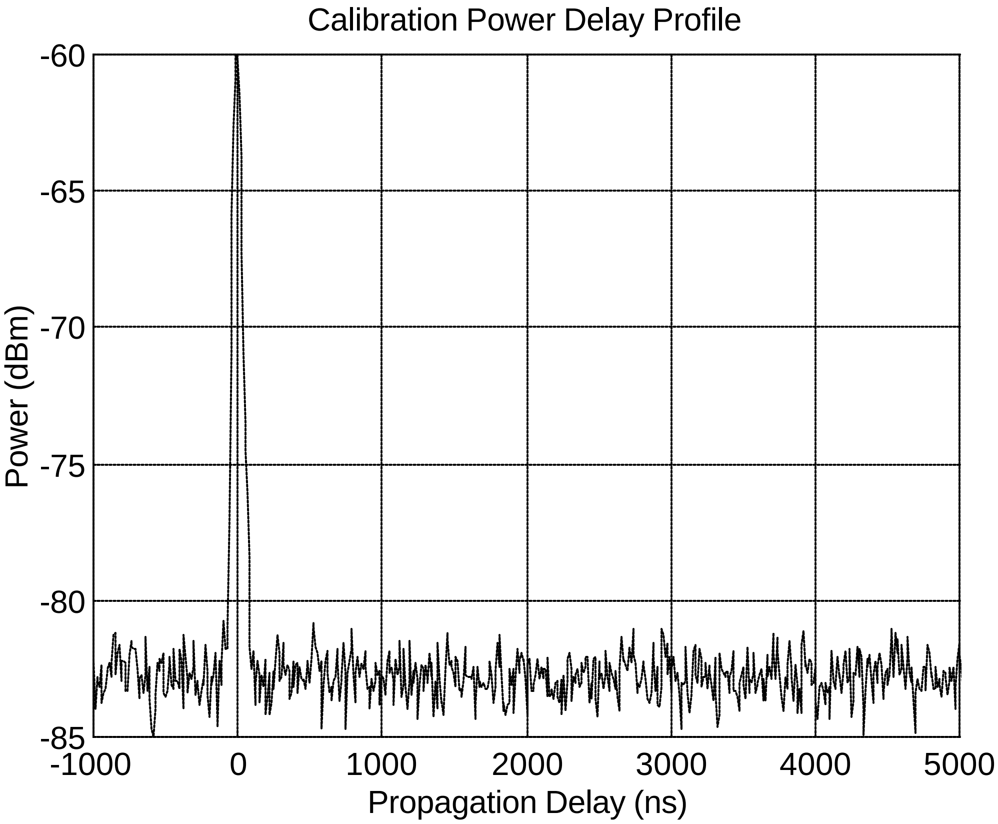


Figure 3.2-3. The calibration profile contains only one signal component. The peak of the signal component is mapped to the power into the receiver to determine the relationship between the voltage measured on the oscilloscope and the power of a signal component in a power delay profile.

The mapping of received power to peak voltage is performed over the entire dynamic range of the receiver. Beginning with a transmitter power which produces a correlation peak which is barely distinguishable from noise, the power is increased until nearly reaching the maximum input power to the receiver, usually determined by the power limitation of the front-end amplifiers or the mixer. From this set of calibration data, a plot is produced which shows the power into the receiver versus peak correlator voltage. Alternatively, a mapping of transmitter output power versus peak correlator voltage is produced as shown in Figure 3.2-4, and the attenuation between the transmitter and receiver during calibration is used at the time of processing to determine the relationship between power into the receiver and peak correlator voltage.

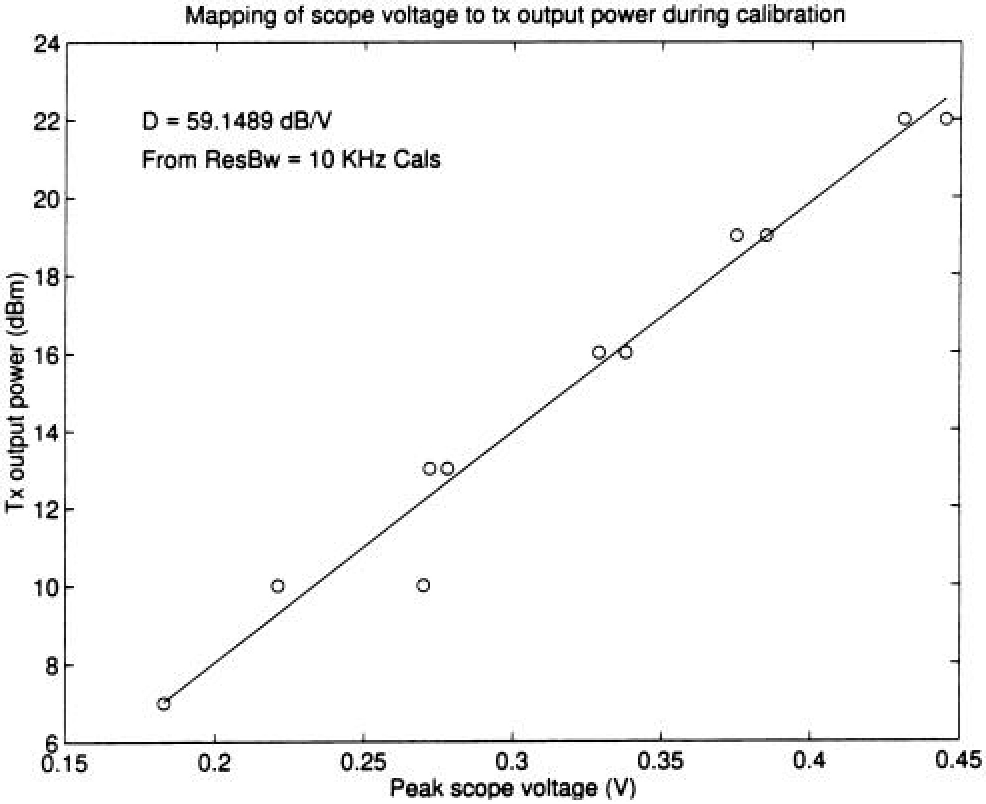


Figure 3.2-4. The linear relationship between transmitter power and oscilloscope voltage is clear from this calibration plot. Note that the power is expressed in dBm.

The linear relationship between received power and correlator voltage is clear from the plot. The values of P_o and D which minimize the mean-square error [20] between the measured data and a line through the measured points are given by

$$D = \frac{N\left(\sum_{k=1}^{N} V_k P_k\right) - \left(\sum_{k=1}^{N} V_k\right)\left(\sum_{k=1}^{N} P_k\right)}{N\left(\sum_{k=1}^{N} V_k^{\,2}\right) - \left(\sum_{k=1}^{N} V_k\right)^2} \qquad (3.2\text{-}12)$$

$$P_o = \frac{\left(\sum_{k=1}^{N} V_k^{\,2}\right)\left(\sum_{k=1}^{N} P_k\right) - \left(\sum_{k=1}^{N} V_k P_k\right)\left(\sum_{k=1}^{N} V_k\right)}{N\left(\sum_{k=1}^{N} V_k^{\,2}\right) - \left(\sum_{k=1}^{N} V_k\right)^2} \qquad (3.2\text{-}13)$$

Where V_k and P_k are the k^{th} of N measured samples of peak voltage and power into the receiver for N calibration profiles. The parameters D and P_o remain constant as long as properties for each component within the measurement system remain constant. Experience has shown that during the course of a measurement survey these parameters vary only slightly but do nonetheless change. Therefore, for accurate wideband measurement data these parameters need to be determined several times throughout a measurement survey.

The calibration procedure was completed six times throughout the train yard measurement survey. The system remained stable throughout all of the measurements as shown in the calibration runs performed at the beginning and end of each of the three days.

Following calibration of the measurement system at Site One, the close-in measurement site, power delay profiles were recorded periodically over several hours. Between each measurement, the receiver antenna was placed on the rail cars at the locations defined by Figure 3.1-1. For measurements that used the receiver antenna mounted on the boxcar and the gondola, a transmit power of -4.5 dBm was used. The transmit power was increased to 10.5 dBm for measurements on the caboose because the receiver antenna was shadowed by the boxcar. At Site One, a total of 44 power delay profiles were recorded at 39 receiver antenna locations.

Site Two measurements recorded power delay profiles for the parallel-track site using a T-R separation of approximately 190 meters. At this site, three receiver antenna locations were used to observe changes in the channel as yard conditions changed. The receiver antenna locations were the roof of the boxcar, the roof of the caboose, and the top of a 6 foot mast mounted between the boxcar and caboose. The transmitter antenna was mounted 6.1 meters above the track level. A total of 165 power delay profiles were recorded over a period of several hours as the channel impulse response changed due to trains moving throughout the rail yard.

Site Three, the cross-track site, was used to measure delay and path loss statistics for a link across a widest part of the rail yard. The 6.1 meter transmitter antenna tower was mounted on the at the north side of the yard, and the transmitter antenna height was approximately 15 meters above track level. Four different receiver antenna locations were used on storage rail cars and passenger rail cars on the south side of the yard. The T-R separation at this site was approximately 230 meters. A total of 187 power delay profiles were collected at Site Three.

3.3 Data Analysis and Measurement Results

A total of 396 power delay profiles were recorded, analyzed, and plotted. The calibration profiles were used to calculate the D and P_o parameters to reconstruct the measurement power delay profiles on power versus time plots. Figure 3.3-1, Figure 3.3-2, and Figure 3.3-3 show samples of the power delay profiles recorded at each of the three measurement sites.

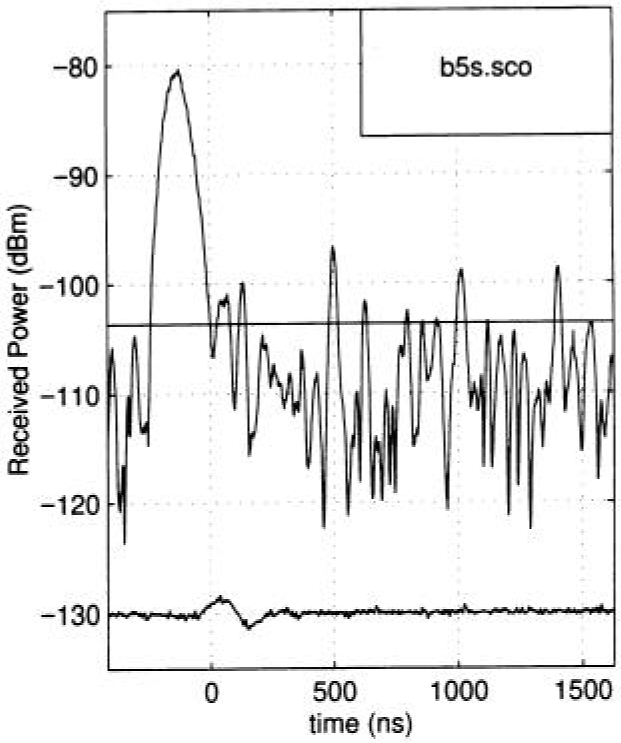


Figure 3.3-1. This power delay profile was recorded at Site One with the receiver antenna mounted on the boxcar. The trace at approximately –130 dBm is the trigger signal used to trigger the oscilloscope.

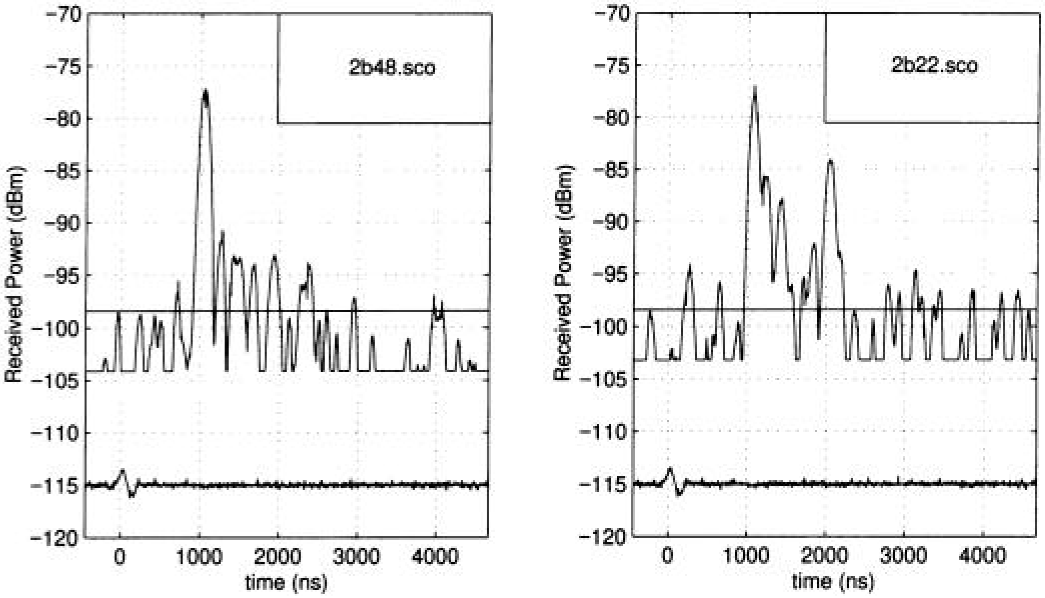


Figure 3.3-2. These power delay profiles represent two channel measurement at Site Two. Several multipath components exist in both profiles.

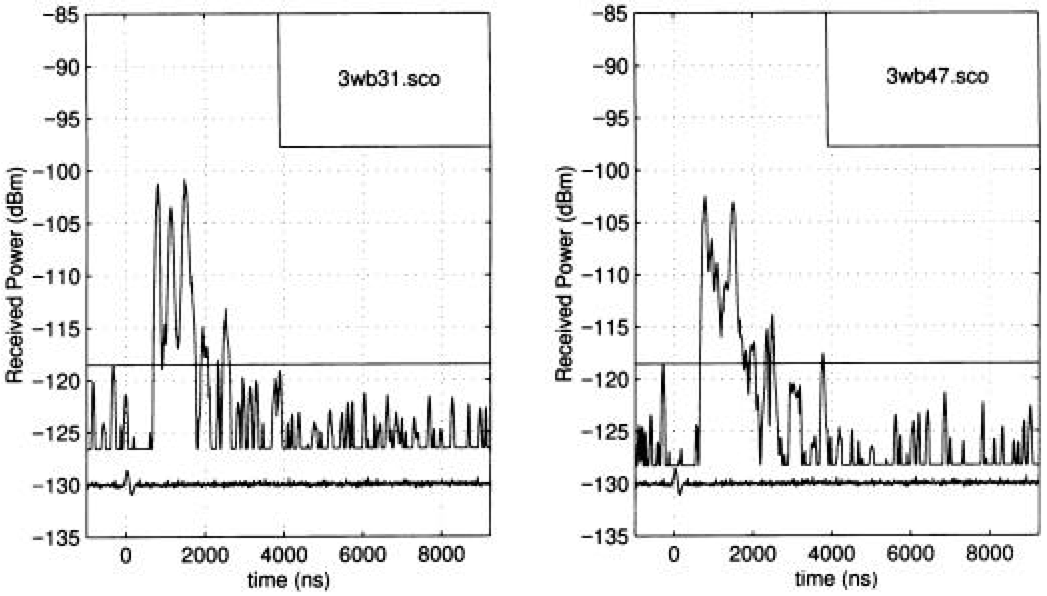


Figure 3.3-3. Strong multipath signal components are shown in these power delay profiles recorded at Site Three.

RMS delay spread and path loss exponent values, described in Chapter 2, were computed from data taken at individual sites. A noise threshold was defined so that the data points of the noise floor shown in the above power delay profiles were not included in the RMS delay spread and path loss calculations. The noise threshold was defined using a portion of the power delay profile where multipath components were observed not to exist. For Site One profiles, the peak power in the last 10% of time delay of the power delay profile was defined to be the noise threshold. For profiles measured at Site Two and Site Three, the first 10% of the profile was a range of delay which preceded the first-arriving component, and power peaks in this region were defined to be the noise threshold. All data points below the noise threshold were discarded in the computation of channel statistics.

Using the RMS delay spread computations from individual profiles, the average RMS delay spread, maximum RMS delay spread, and standard deviation of RMS delay spread values were computed for sets of data. Table 3.3-1 shows a summary of these results.

Table 3.3-1. Summary of RMS delay spread and path loss exponents.

Location	n	σ (dB)	Avg σ_τ (ns)	Max σ_τ (ns)	StDev σ_τ (ns)
Site 1	-	-	94.6	280.7	58.3
Site 2	2.69	3.43	331.0	731.5	145.0
Site 3	3.41	2.49	565.8	1161.2	164.3
Sites 2 & 3	3.08	8.78	454.3	1161.2	194.3
All Sites	-	-	412.9	1161.2	216.8

During measurements at the rail yard, power delay profiles would change over time due to movement of trains throughout the yard. Thus, the RMS delay spread varied from profile to profile even when the transmitter and receiver antennas remained at fixed points. Figures 3.3-4 through 3.3-7 show the cumulative distribution functions (CDF) of RMS delay spread for measurements taken at each site (see [18]). The largest RMS delay spread, as shown in the CDF plots, was measured to be 1161 ns at Site Three. This was expected because of the large number of multipath scatterers and the large T-R separation.

Using well known results [3][17] which state that satisfactory bit error rates can be achieved with a radio system that does not use an equalizer if the RMS delay spread does not exceed 10% of the modulation symbol duration, it was calculated that unequalized data rates on the order of 137 kHz to 356 kHz may be used for close-in rail yard links. Using measurement results from the cross-yard measurements where the T-R separation was larger, a maximum unequalized data rate of 86 kHz can be anticipated.

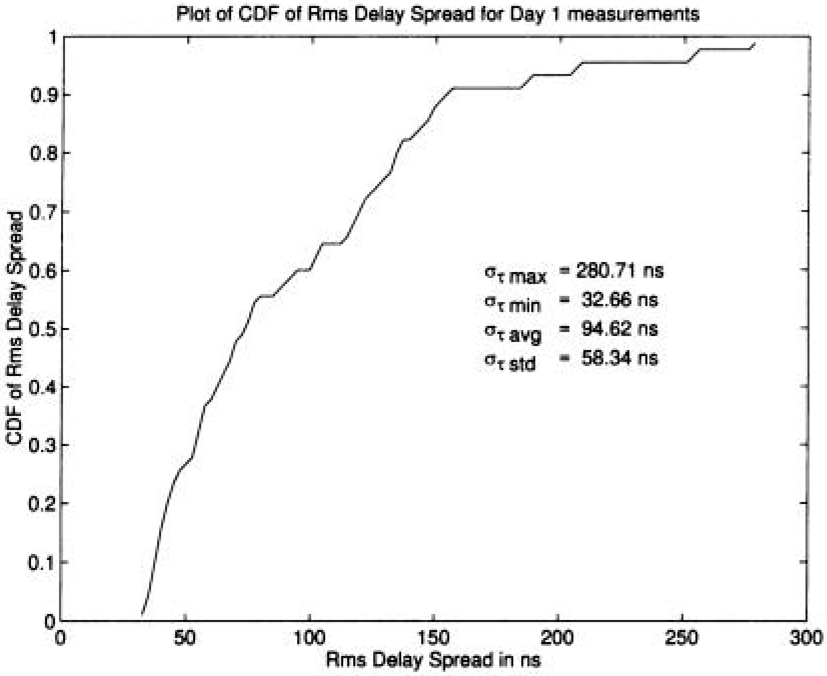


Figure 3.3-4. This CDF of RMS delay spread was calculated using measurement data taken at Site One. The smallest RMS delay spread values were recorded at Site One.

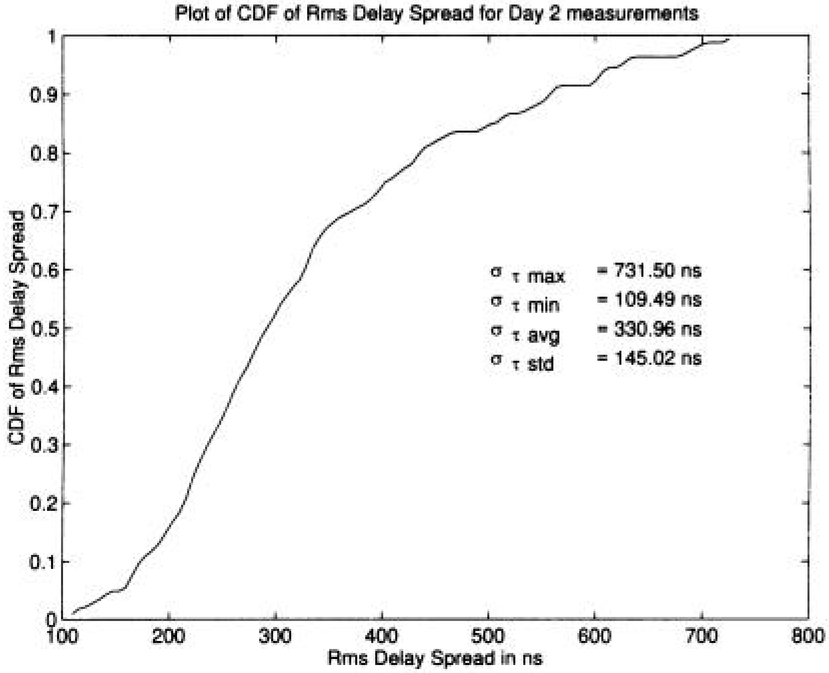


Figure 3.3-5. This CDF of RMS delay spread was calculated using measurement data taken at Site Two.

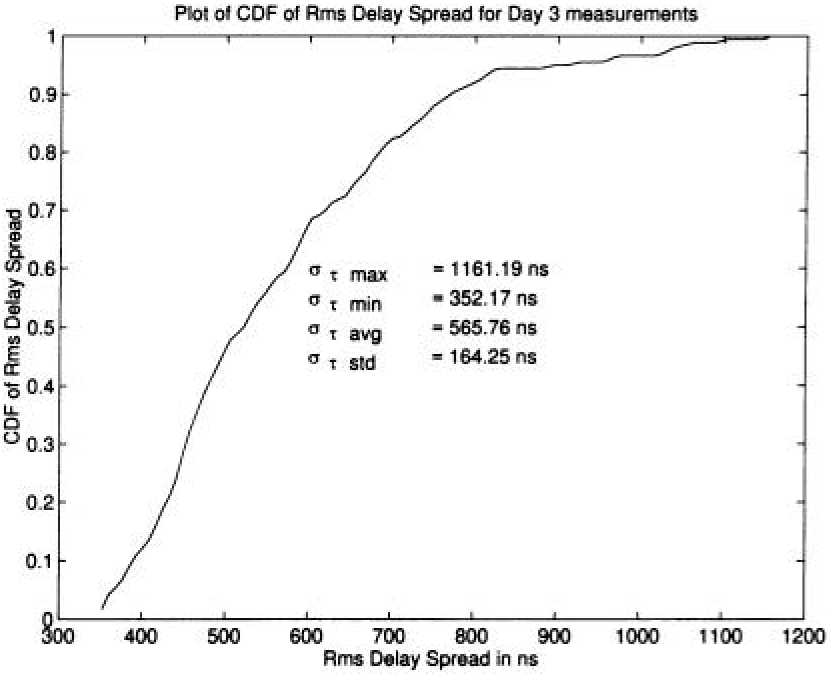


Figure 3.3-6. This CDF illustrates the delay spread distribution for measurements taken at Site Three. The largest RMS delay spread value was recorded at Site Three.

For close-in links, path loss exponents were close to that of free space (*n*=2). At Site One, the calculated best-fit path loss exponent was 2.7, implying 27 dB of loss per decade of distance. For links across the yard, as shown by measurements at Site Three, shadowing due to passing trains resulted in higher path loss. Measurements showed that path loss increased by 34 dB per decade of distance. The best-fit path loss exponent was 3.4, close to path loss exponents calculated from urban microcellular measurements [2].

3.4 Summary

Time dispersion in RF channels caused by multipath limits the maximum data rate for wireless communication systems. RMS delay spread is a quantity which can be determined from sliding correlator measurements and is frequently used to quantify time dispersion and characterize an environment. In this chapter, a propagation study at the Norfolk Southern rail yard was presented. The purpose of the study was to characterize time dispersion in a rail yard environment for train yard communication and tracking systems.

A sliding correlator system was used to measure impulse response data at three sites within the rail yard. Site One was the smallest measurement site which was used to measure channel responses for close-in radio communications. Site Two was located along the perimeter of the rail yard and was used to measure delay and path loss statistics for a parallel-track link. Site Three was the largest of the measurement sites and was used to determine delay and path loss statistics for a link across the widest part of the rail yard.

Each power delay profile recorded at the measurement sites was processed to determine the RMS delay spread and received power. The delay spread and received power results were summarized by combining data for each measurement site. The largest RMS delay spread was recorded at Site Three, as expected since this site had the largest T-R separation. The maximum measured RMS delay spread value for Site Three was 1161 ns, corresponding to a maximum symbol rate of 86 kHz using the assumption that an RMS delay spread of more that 10% of the symbol period results in unacceptable error rates. The maximum RMS delay spread values measured at Site One and Site Two were considerably smaller at 281 ns and 732 ns, respectively.

This chapter has presented a practical application for the sliding correlator measurement system. The results presented here may be used for design and simulation of digital radio systems for rail yard communications.

Chapter 4

Simulation of a Sliding Correlator Measurement System

Software simulations of wireless systems provide an accurate way to test, validate, and understand the operation of physical systems. Simulations give the system designer or researcher another alternative for exploring signals within a system without the expense, difficulty, and potential problems of probing an active system. Simulations also provide a means of testing designs before prototypes or production models are built. In addition, the parameters of simulations can be controlled so that algorithms which use the simulated signals can be tested under known conditions. Such is the case for the simulations presented in this chapter, where a sliding correlator measurement system is simulated for the purpose of producing channel impulse response data for testing post-processing algorithms. This chapter presents two methods for simulating the sliding correlator system. Work presented in Chapter 4 uses the simulations to validate newly developed post processing algorithms.

Both simulations discussed in this chapter are designed to use a known channel as an input to produce a simulated power delay profile as an output. Other inputs to the simulations include the sliding correlator parameters discussed in previous chapters (see Table 2.4-1 in Chapter 2). The simulated power delay profiles may be used for testing any application that uses raw power delay profile data as an input, such as channel impulse response estimation algorithms. Using the simulated data for testing channel estimation algorithms allows exact knowledge of the channel impulse response so that the algorithms can be easily validated.

4.1 Ideal Correlator Method of Simulation

Simulation of wireless communications systems is often implemented using a low pass equivalent model [9]. Low pass equivalent forms of a system use the complex envelope representation of a signal and eliminate the need for high sample rates required to sample modulated, high-frequency carrier signals. Because lower sample rates are used, simulation time is decreased and requirements on computer performance are reduced. The simulation of the sliding correlator measurement system presented in this section uses the low pass model shown in Figure 4.2-1. All of the simulation components were implemented in MATLAB m-files.

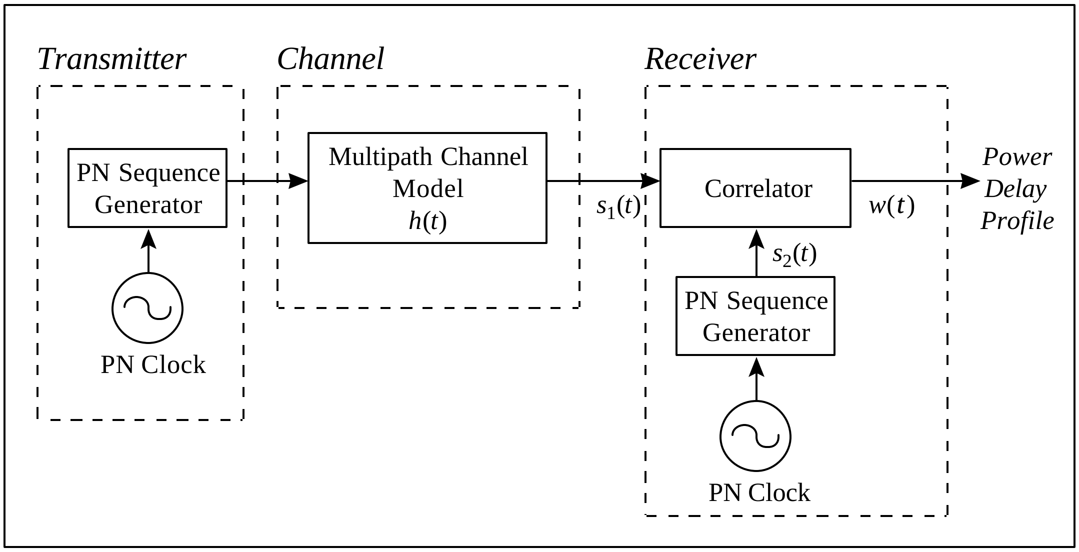


Figure 4.1-1. The low pass model of the sliding correlator measurement system is divided into transmitter, channel, and receiver sections. The transmitter output is modeled as a baseband PN sequence and the receiver is modeled as an ideal correlator.

The transmitter is represented by a PN sequence generator that is clocked at the transmitter PN chip rate. The spectrum at the output of the transmitter model is a *sinc* function centered at 0 Hz with a first-null baseband bandwidth of R_c Hz. The transmitter PN sequence is sampled at n_s samples per chip, and hence the sampling rate in Hz for the transmitter signal is

$$f_s = n_s R_c \qquad (4.1\text{-}1)$$

Execution of the simulation requires the transmitter to produce one period of the transmitted PN sequence. If l is the number of chips in the transmitter PN sequence, then the total number of samples stored N_s is given by

$$N_s = n_s l \qquad (4.1\text{-}2)$$

The transmitted PN sequence is represented in MATLAB by defining a PN chip vector $\bar{s}_{chip}$ which represents n_s samples of a single PN chip,

$$\bar{s}_{chip} = [c_1, c_2, \cdots, c_{n_s}] \qquad (4.1\text{-}3)$$

The elements $c_1, \cdots, c_{n_s}$ define the shape of the PN chip, and $c_1, \cdots, c_{n_s} = 1$ is used to produce a PN sequence with rectangular chips. Given that the value of the k^{th} PN sequence chip for $k = 1 \cdots l$ is given by the scalar quantity $s_{PN}(k) \in \{-1,1\}$, the vector representing the output of the transmitter $\bar{s}_T$ is formed in MATLAB by concatenating $\bar{s}_{chip}$ vectors by

$$\bar{s}_T = [s_{PN}(1)\bar{s}_{chip}, s_{PN}(2)\bar{s}_{chip}, \cdots, s_{PN}(l)\bar{s}_{chip}] \qquad (4.1\text{-}4)$$

Each element of $\bar{s}_T$ represents one sample of the PN sequence, and the length of the vector $\bar{s}_T$ representing one complete cycle of the PN sequence is N_s.

PN sequences can be created using a shift register and a modulo-2 summation of the shift register stage values as discussed in Chapter 2. PN sequence generators can be designed to use only two feedback stages, hence requiring only a single XOR gate as shown in Figure 4-1-2 and Figure 4.1-3. The transmitter model in the simulation allows two feedback taps to be defined for the sequence generator. The taps are located at stage i and stage j of the m-stage shift register, and the tap locations are specified using the square-bracket notation [i,j]. The valid range for setting taps in the simulation is $i, j \in \{1,2,\cdots,m\}$; however, not all tap combinations will produce maximal length PN sequences. Many references (including [9]) provide tables for choosing feedback taps for linear shift register PN generators.

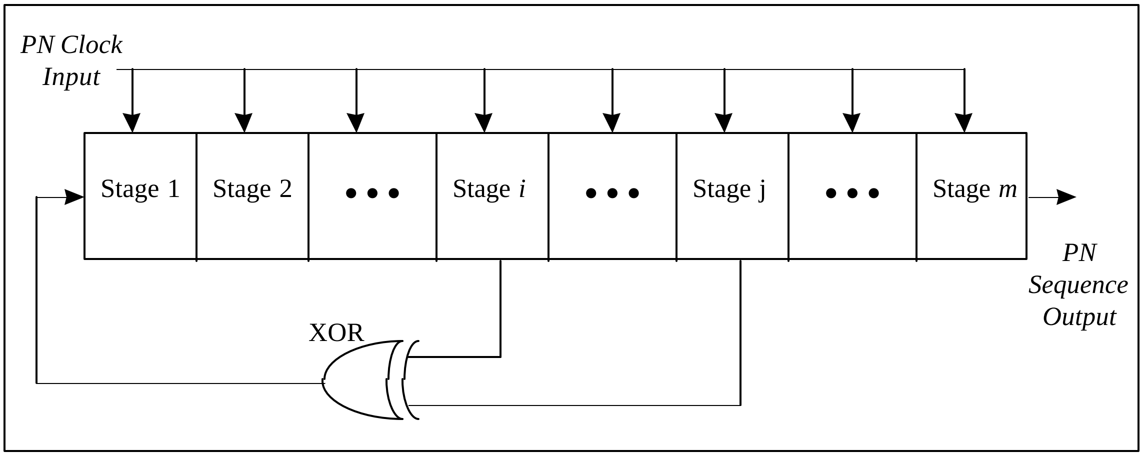


Figure 4.1-2. The PN sequence is generated by simulating an *m*-stage shift register tapped at two of the stages [*i*,*j*]. The values are shifted by a common clock input and stage *m* provides the output.

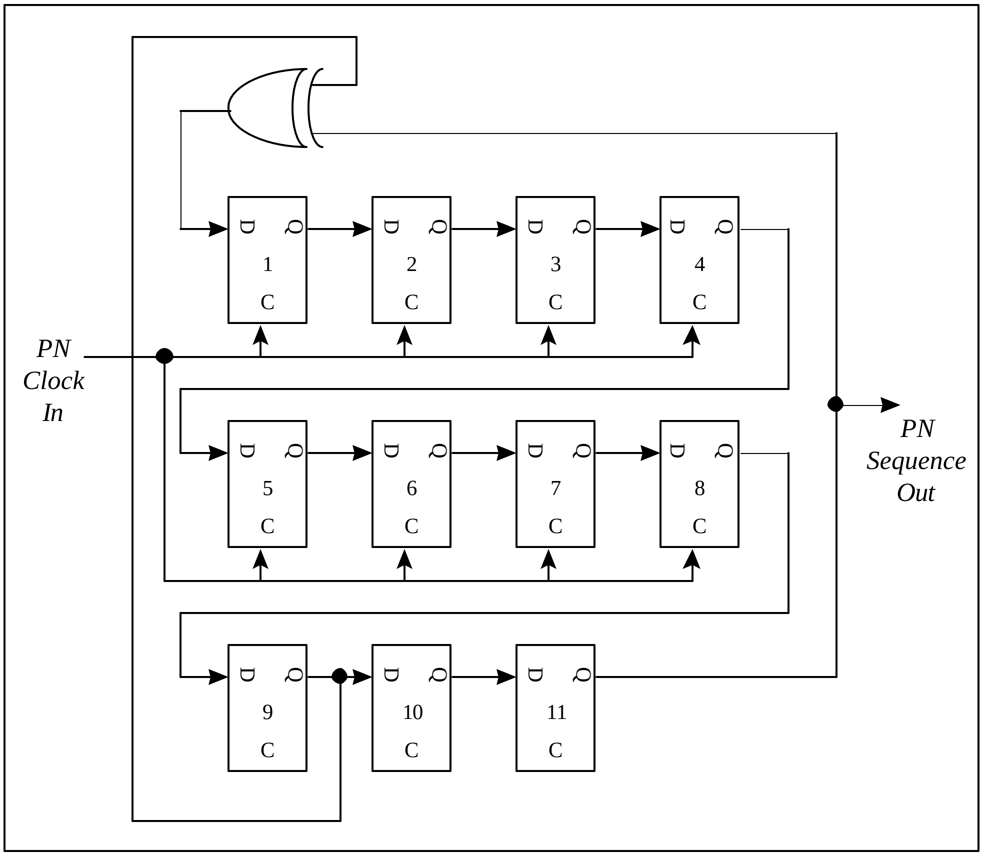


Figure 4.1-3. Emitter Coupled Logic (ECL) D-flip-flops (DFF) are used to implement the shift register. This design requires only one XOR gate because only two taps at [9,11] are used. The transmitter in the simulation uses a model of this circuit to produce the PN sequence.

The model of the radio channel immediately follows the transmitter. The simulated channel is represented by a simple model consisting of a finite set of delay and attenuation pairs. If $x(t)$ is the channel input and $y(t)$ is the channel output, then

$$y(t) = \sum_{i=1}^{N} a_i e^{jq_i} x(t - t_i) \tag{4.1-5}$$

The N pairs of attenuation and delay are implemented as shown in the block diagram in Figure 4.1-4.

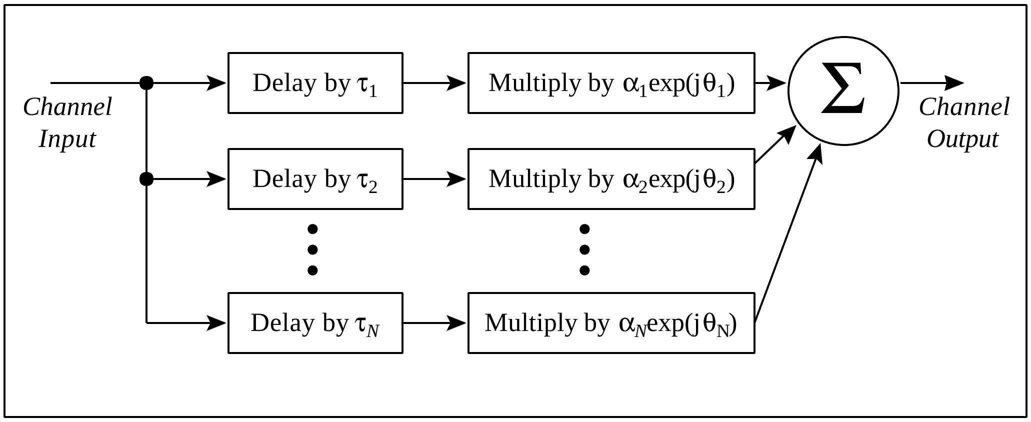

Channel Input
Delay by τ_1
Delay by τ_2
Delay by τ_N
Multiply by $\alpha_1\exp(j\theta_1)$
Multiply by $\alpha_2\exp(j\theta_2)$
Multiply by $\alpha_N\exp(j\theta_N)$
Σ
Channel Output

Figure 4.1-4. The channel is modeled by N pairs of attenuation and delays to represent a finite number of multipath components.

Although the user is allowed to specify any double precision number for the delays, the simulation must round the delay to a number such that the channel input is delayed by an integer number of samples. The delay in units of PN sequence samples d_i corresponding to the time delay t_i for each signal component is given by

$$d_i = n_s l \frac{t_i}{t_{\max}} \tag{4.1-6}$$

where $t_{\max}$ is the maximum measurable delay given by $D_{ub}/c = l/R_c$ (see Chapter 2). Because d_i must be rounded to the nearest integer, d_i has a time resolution of Δt_d given by

$$\Delta t_d = \frac{1}{n_s R_c} \tag{4.1-7}$$

where Δt_d is the time in seconds between samples of the transmitted signal. While the resolution may seem to be a limitation of the simulation, the time resolution of the simulator Δt_d is actually better than the time resolution of the actual sliding correlator Δt system by a factor of $n_s/2$ since

$$\Delta t = \frac{2}{R_c} > \Delta t_d \tag{4.1-8}$$

The channel output is represented by vector $\overline{s}_R$ which is of the same length as the transmitter output vector $\overline{s}_T$. Because of the cyclic nature of the sliding correlator system, samples which are delayed past the last sample of $\overline{s}_R$ are summed with the beginning samples of $\overline{s}_R$. For example, if the transmitted signal is delayed by ten samples (to represent propagation delay), then the last ten samples of $\overline{s}_T$ are combined (summed) with the first ten samples of $\overline{s}_R$ to form the received signal vector. This cyclic adding accounts for delayed signals which shift into the following period of the receiver PN sequence, while still representing the received signal by a vector of length N_s.

The function of the measurement system receiver is to produce a correlation function using as inputs the received signal and a copy of the transmitted signal. In the real sliding correlator system, the correlation is approximated by using a PN sequence that is clocked slightly slower at the receiver. This simulation, however, will assume that the sequences at the transmitter and receiver are clocked at the same rate R_c. Because the difference between the transmitter and receiver chip rates in the real system is extremely small, the equal rate assumption will produce reasonable first-cut approximations to the power delay profiles produced by the real system, but will cause simulated profiles to lack some of the characteristics of measured power delay profiles as will be shown. Using the actual rates of the transmitter and receiver PN sequence generators would produce a more accurate result, but would require an extremely large number of samples to represent the sequences. For example, to simulate a slide factor of 1000, the transmitter chip period must be equal to 0.999 multiplied by the receiver chip period. This implies that at least 1000 samples must be used for each chip, resulting in extremely long execution times to perform simulations and extremely large memory requirements for the simulation software.

Figure 4.1-5 shows the model for the measurement system receiver. The simulated receiver uses as inputs the output of the channel and the transmitted PN sequence. For each delay t, the receiver performs a correlation over one period T_o of the PN sequence. Using the signals $s_1(t)$ and $s_2(t)$ as inputs, the output of the receiver $w(t)$ is given by

$$w(t) = \int_{t=nT_o}^{(n+1)T_o} s_1(t)s_2(t-t)dt \qquad (4.1\text{-}9)$$

where n is an integer. It is typically more efficient to implement a simulation using matrix operations rather than performing calculations on a sample-by-sample basis. The ideal correlator simulation performs the operation described by equation (4.1-9) by multiplying the received signal vector by the transpose of a vector which represents the PN sequence generated at the receiver. For each sample of delay t, the signal w_k is calculated by

$$w_k = \bar{s}_1\bar{s}_2^{\,T} \qquad (4.1\text{-}10)$$

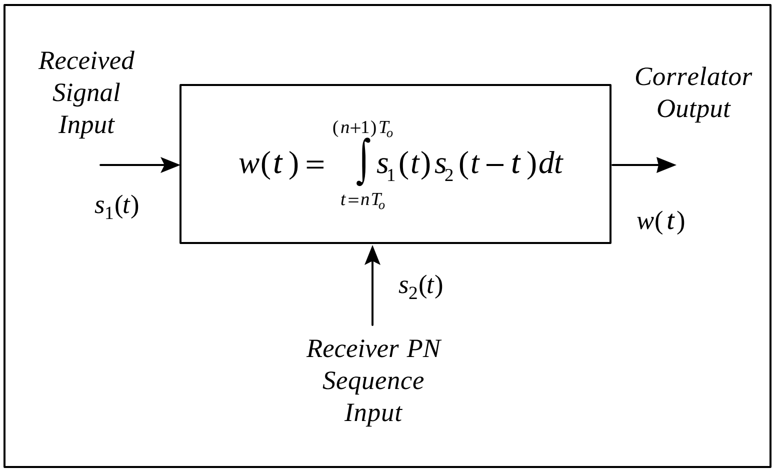

Received Signal Input
$s_1(t)$
$w(t) = \int_{t=nT_o}^{(n+1)T_o} s_1(t)s_2(t-t)dt$
Correlator Output
$w(t)$
$s_2(t)$
Receiver PN Sequence Input

Figure 4.1-5. The measurement system receiver is modeled as a correlator that uses the channel output and the locally generated PN sequence as inputs. The output $w(t)$ is a voltage that represents the simulated power delay profile.

The variable w_k is one element of the vector $\overline{w}$, the vector of samples representing the output of the correlator, where $k = t \cdot N_s / t_{ub} \in \{1,2,\cdots,n_s l\}$ and t_{ub} is the maximum unambiguous measurable delay of the system. The vector $\overline{s}_1$ represents one period ($n_s l = N_s$ samples) of the received signal, and the vector transpose $\overline{s}_2^{\,T}$ represents one period of the PN sequence generated at the receiver delayed by k samples. The signal $w(t)$ which represents the power delay profile is formed by recognizing that the elements of the output vector $\overline{w}$ are samples separated by a period of $1/(n_s R_c)$. The simulated power delay profile can be shown by plotting $10\log(w(t))^2$ versus t. Sample plots are shown later in this section in Figure 4.1-10 and Figure 4.1-11.

An important step in the development of software simulations is validation of the models and signals within the simulation. Each section of the sliding correlator simulation (transmitter, channel, and receiver) was validated individually. Signals at the output of each section were compared to either corresponding signals in the real system or theoretical results. The simulated system was also validated as a whole by comparing simulated impulse response data to the channel impulse response used to generate the output power delay profile.

The transmitter model was validated by comparing the simulated signal output to the output of the PN sequence generator in the real system. Samples of the PN sequence in the real system were acquired using an oscilloscope. Several sample chip runs acquired from the real system were shown to match sample chip runs taken from the simulated transmitter as shown in Figure 4.1-6 and Figure 4.1-7. The comparison also showed that for a significantly long chip run (a run containing more chips than the number of stages in the shift register used to produce the sequence), the chip run acquired from the real system could be uniquely located within the simulated PN sequence. Theory dictates that a chip run of length m or greater appears only once in a maximal length PN sequence generated by an m-stage shift register. See [9] for a comprehensive list of the properties of a maximal length PN sequence.

In addition to comparing the real signal with the simulated signal, the characteristics of the simulated PN sequence were compared against theory. First, the number of bits representing “ones” was shown to exceed the number of bits representing “zeros” by one. For several different numbers of stages assigned to the shift register and for a variety of tapping points, this property was shown to hold true. Second, the autocorrelation function of the simulated PN sequence was shown to be identical to the theoretical correlation function. This is demonstrated in the following samples of simulated power delay profiles.

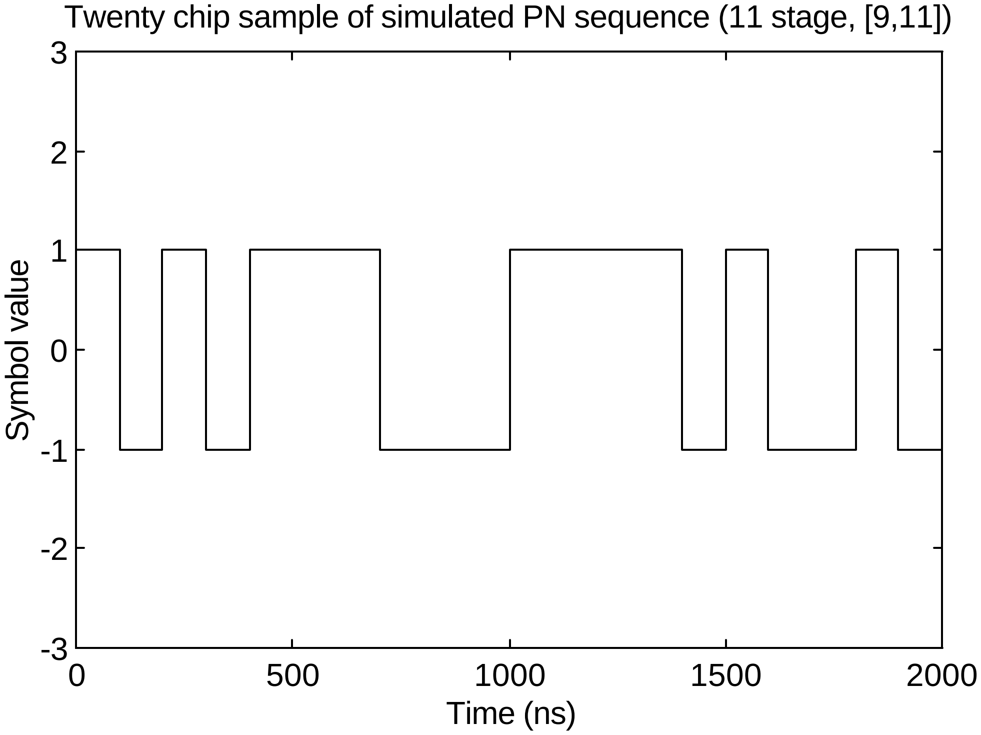


Figure 4.1-6. This plot of a portion of the simulated transmitter signal is identical to the measured signal shown in Figure 4.1-7. A chip rate of 10 MHz was used with an eleven-stage shift register tapped at [9,11].

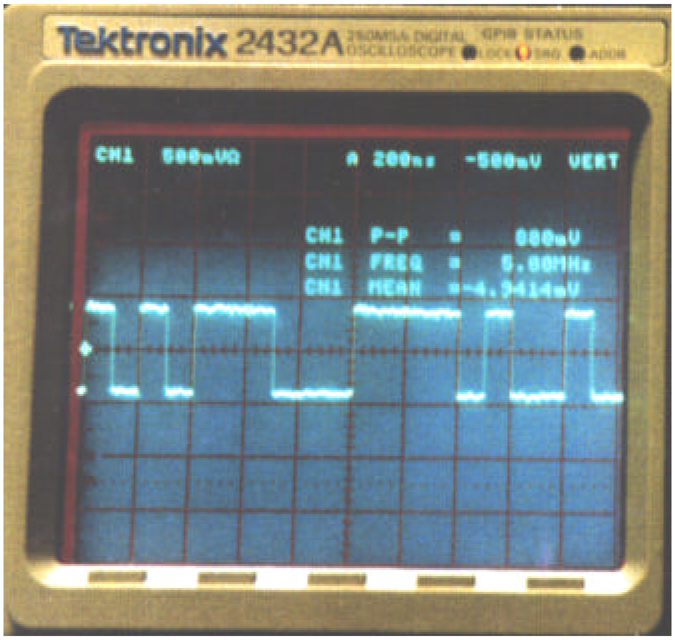


Figure 4.1-7. This signal was measured at the output of the PN sequence generator on the sliding correlator measurement system. The hardware produces the PN sequence with an eleven-stage shift register tapped at [9,11].

The channel model was validated by comparing the channel output to the channel input for an arbitrary input signal. Figure 4.1-8 shows a comparison of the channel input to the channel output for a rectangular pulse. The input to the channel is the faint trace with an amplitude of one and a period of 100 ns. The channel for this example was defined to have an attenuation of 6 dB and a propagation delay of 200 ns. The output of the channel is the dark trace rising at 200 ns and falling at 300 ns. The plot shows that the channel model is functioning properly by delaying the input by 200 ns and attenuating the input signal by one-half of its voltage (6 dB power attenuation).

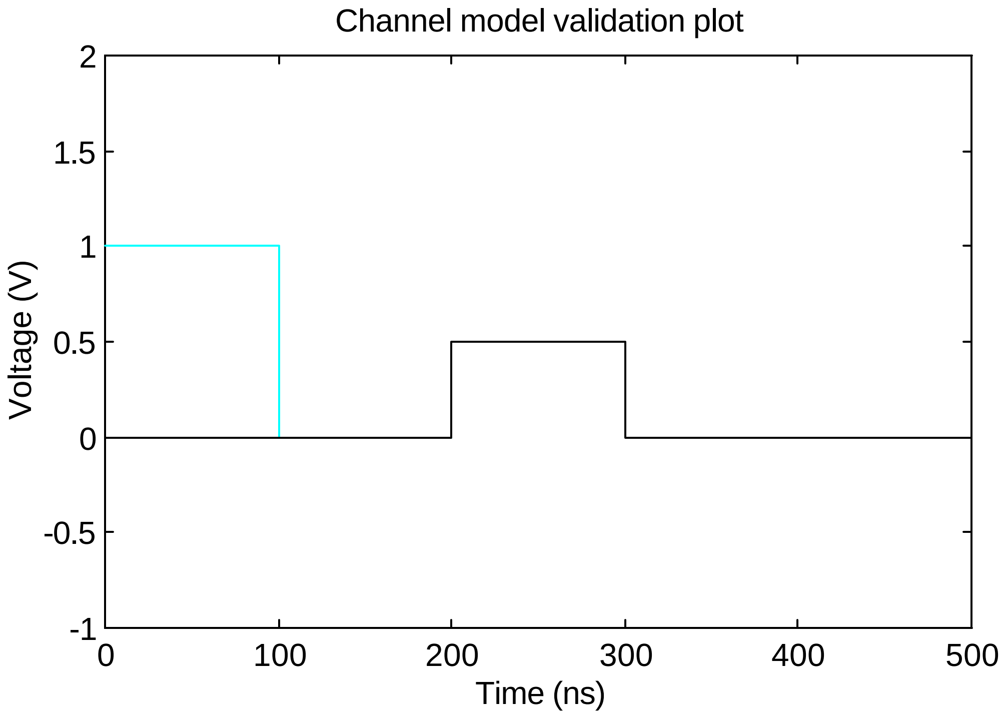


Figure 4.1-8. The faint trace represents the input signal voltage to a channel with an attenuation factor of 6 dB and a propagation delay of 200 ns. The dark trace represents the output of the channel.

Validation of the receiver model was accomplished by comparing the output of the receiver to theoretical results for several channels. One example of a profile used for validation is shown in Figure 4.1-9, where a fictitious channel was used which had 1 µs of delay and 0 dB attenuation. The sliding correlator theory presented in Chapter 2 dictates that the correlation peak produced by the received signal component should lie at the point designated by the small circle on the plot. This is shown to hold true in the figure. Theory also says that the floor of the profile containing a single signal component should exist at a value of 10log(l^2) where l is the number of chips in the PN sequence. For this validation profile, a sequence length of 127 chips was used, and hence a floor of 42.1 dB is expected. The floor of the simulated profile was shown to be 42.1 dB, again matching theoretical results. In addition, the theoretical maximum measurable delay for the 127-chip PN sequence clocked at 10 MHz is lT_c = 127(0.1 µs) = 12.7 µs. The simulated profile trace is shown to end at exactly 12.7 µs. Finally, theory states that the base of the signal component in the profile should be $2/R_c$ = 0.2 µs, and this is demonstrated to be true.

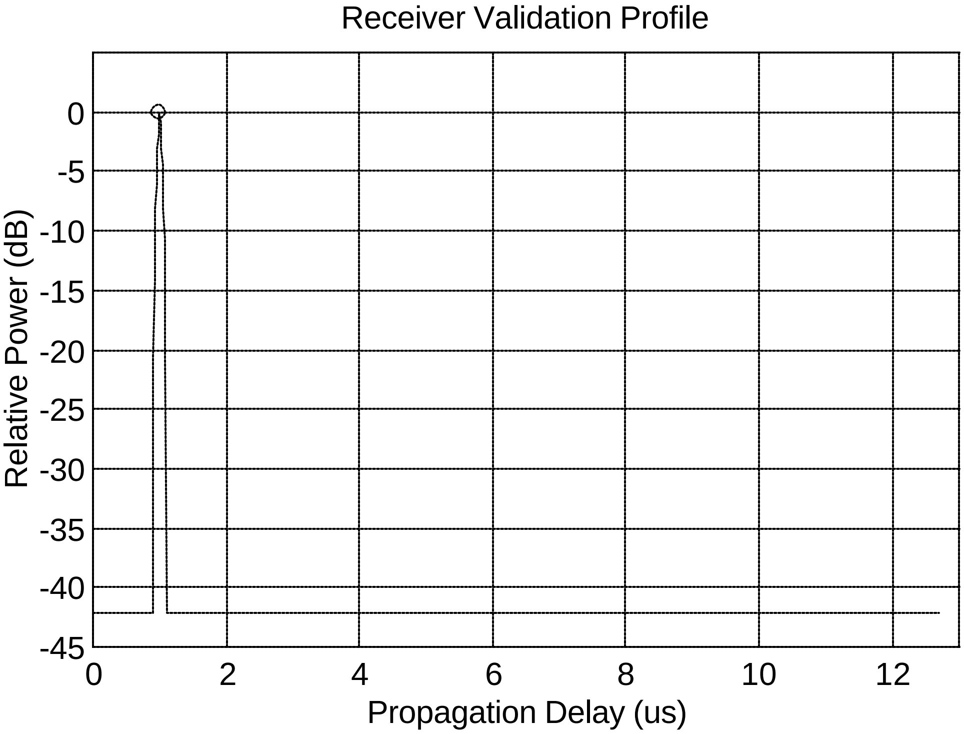


Figure 4.1-9. The receiver model is validated using a channel with 0 dB loss and 1 µs delay. The correlation peak falls within the circle which indicates the theoretical location of the peak.

Figures 4.1-10 and 4.1-11 show sample power delay profiles created using the validated simulation. The profile shown in Figure 4.1-10 was created using a channel defined with four multiple paths. The propagation delays defined for the channel were 200 ns, 300 ns, 600 ns, and 800 ns, and the corresponding attenuation values are 5 dB, 12 dB, 10 dB, and 13 dB, respectively. The simulated system used to measure the channel used a seven-stage PN sequence generator to produce a sequence with 127 chips clocked at 100 MHz. The calculated resolution for the simulated system is 20 ns, and the maximum measurable delay is 1270 ns. The simulated power delay profile is shown to accurately represent the impulse response of the defined channel.

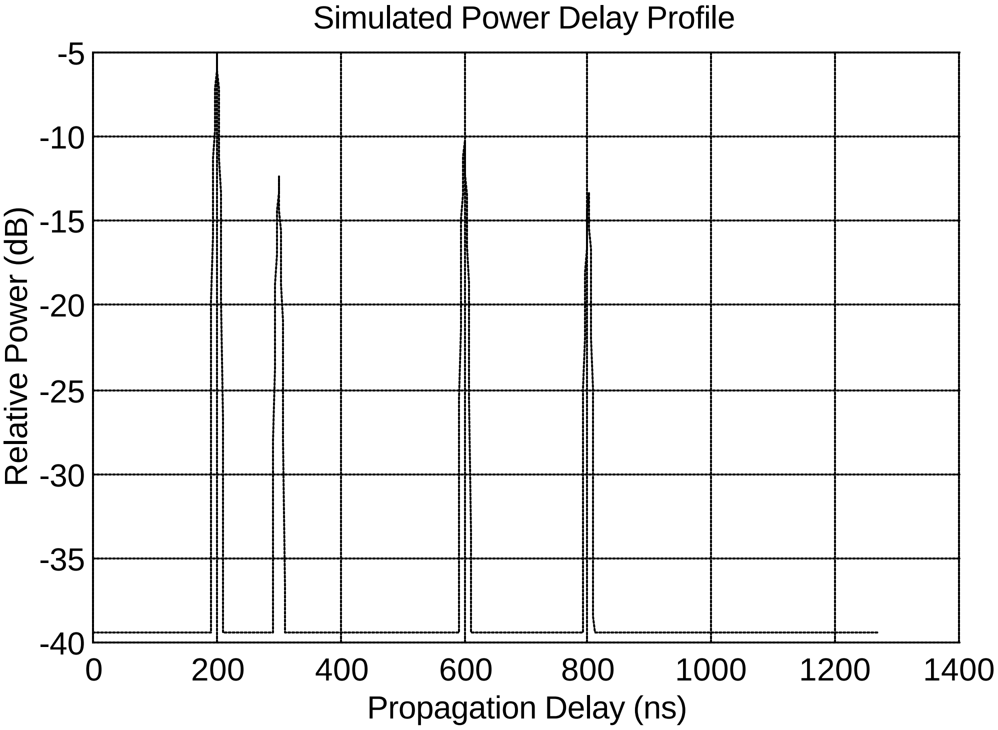


Figure 4.1-10. This power delay profile was simulated for a system using a chip rate of 100 MHz and a seven-stage PN sequence generator tapped at [3,7]. The channel delays are 200 ns, 300 ns, 600 ns, and 800 ns, and the corresponding attenuation values are 5 dB, 12 dB, 10 dB, and 13 dB, respectively.

Figure 4.1-11 was created using some of the same system parameters used for the real measurement system during the rail yard measurement survey discussed in Chapter 3. One notable exception is that the simulated system used a chip rate of 100 MHz to demonstrate a higher resolution, instead of the 10 MHz chip rate used during the rail yard measurement survey. The simulation was performed using an eleven-stage shift register tapped at [9,11]. The propagation delays defined for the channel are 100 ns, 150 ns, 500 ns, 525 ns, and 600 ns and the corresponding attenuation values defined for the channel are 3 dB, 12 dB, 5 dB, 10 dB, and 13 dB, respectively. The maximum measurable delay (to avoid the range ambiguity problem) for the simulated system was 20.47 μs, but the delay range shown in the plot was limited to 1000 ns to provide an expanded view of the signal components.

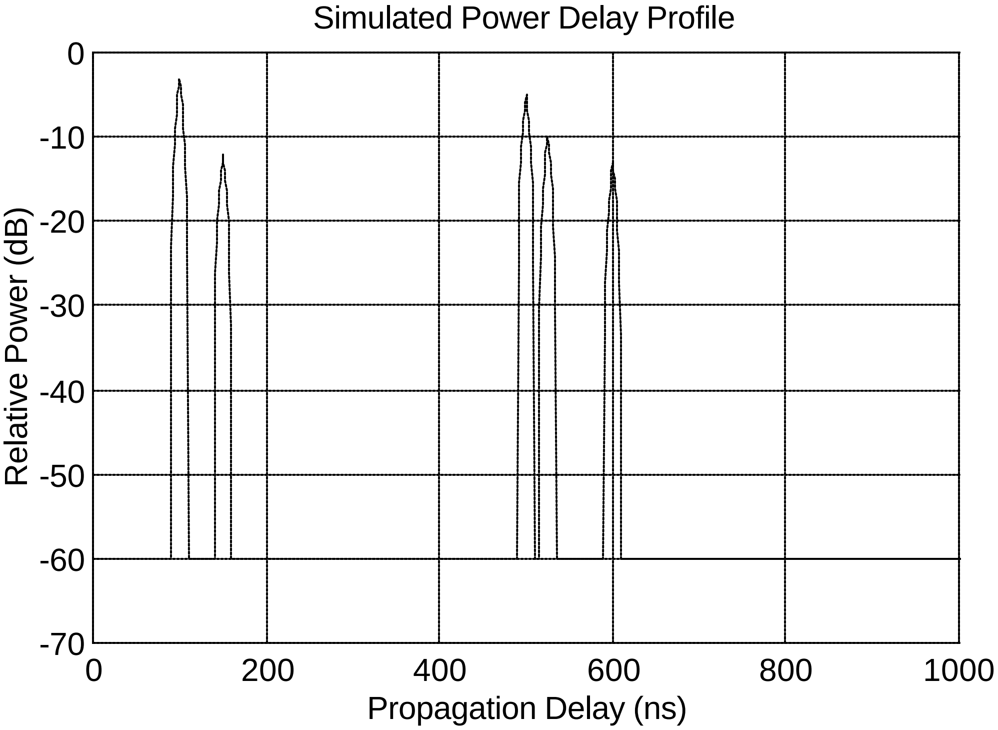


Figure 4.1-11. This power delay profile was simulated using a transmit power of 0 dBm, a chip rate of 100 MHz, and an eleven-stage PN sequence generator tapped at [9,11]. The channel delays are 100 ns, 150 ns, 500 ns, 525 ns, and 600 ns and the corresponding attenuation values are 3 dB, 12 dB, 5 dB, 10 dB, and 13 dB, respectively.

The simulation presented here which uses an ideal correlator at the receiver produces good estimates of the correlation peaks produced by a hardware implementation of the sliding correlator measurement system. Simulation parameters are easy to change and the models of the simulation can run independently to execute partial simulations. The simulation run-time is primarily a factor of the length of the PN sequence, and most of the processing time is used by the correlation of the received signal with the original PN sequence. Simulations which use a PN sequence shorter than 512 chips produce simulated profiles within several minutes using a 200 MHz Pentium-Pro computer. Using a 2047-chip PN sequence for a simulation causes the execution time to increase to approximately 45 minutes.

The simulation falls short of accurately estimating the power level and shape of the power delay profile floor observed in measured profiles. The floor in measured profiles as observed during this research is typically only 25 dB to 30 dB below the peak signal component and appears very noisy in nature. Despite the noise floor inaccuracy, the ideal correlator simulation produces profiles which are useful for testing impulse response data post-processing algorithms. The simulation presented here is used to validate algorithms presented in Chapter 5.

4.2 Analytical Probing Signal Method of Simulation

The ideal correlator measurement system simulation described in the previous section requires a calculation of a new correlation function for each new impulse response defined for the channel model. The correlation process is the largest time sink of the simulation, but this process is required to produce each simulated power delay profile using the ideal correlator model. Presented in this section is another method of simulating the sliding correlator measurement system which quickly generates simulated power delay profiles using an *equivalent probing signal*. Described first is a derivation and definition of a probing signal along with the measurement system model which employs the probing signal. Next, an example of this new measurement system model is used to produce simple power delay profiles similar to the profiles produced by the ideal correlator simulator. Finally, an advanced technique of deriving an accurate probing signal is presented, and this probing signal is used in simulations which not only predict the shape, strength, and delay of signal components but also predict the profile floor between components.

Given that $s_1(t)$ is the input to the measurement system receiver and $s_2(t)$ is the PN sequence generated at the receiver for correlation, the output of the correlator is again expressed as

$$w(t) = \int_{-\infty}^{\infty} s_1(t) s_2(t - t)\, dt \qquad (4.2\text{-}1)$$

The received signal $s_1(t)$ can be related to the transmitted signal $s_T(t)$, using the channel model given by (2.1-1) in Chapter 2, by

$$s_1(t) = \sum_{i=1}^{N} a_i s_T(t - t_i) e^{jq_i} \qquad (4.2\text{-}2)$$

where a_i, t_i, and q_i correspond to the attenuation, delay, and phase induced by the i^{th} of N multipath components induced by the channel. The output of the correlator can now be expressed as

$$w(t) = \int_{-\infty}^{\infty} \left(\sum_{i=1}^{N} a_i s_T(t - t_i) e^{jq_i} \right) s_2(t - t) dt$$

$$= \int_{-\infty}^{\infty} \left(\sum_{i=1}^{N} a_i s_T(t - t_i) e^{jq_i} s_2(t - t) \right) dt$$

$$= \sum_{i=1}^{N} \left(a_i e^{jq_i} \int_{-\infty}^{\infty} s_T(t - t_i) s_2(t - t) dt \right) \qquad (4.2\text{-}3)$$

By applying a change of variable $t=t'+\ t_i$ and then replacing t' with t, the signal $w(t)$ can be written as

$$w(t) = \sum_{i=1}^{N} \left(a_i e^{jq_i} \int_{-\infty}^{\infty} s_T(t) s_2(t - t + t_i) dt \right)$$

$$= \sum_{i=1}^{N} \left(a_i e^{jq_i} \int_{-\infty}^{\infty} s_T(t) s_2(t - (t - t_i)) dt \right) \qquad (4.2\text{-}4)$$

Now the equivalent probing signal can be defined by

$$p(t) = \int_{-\infty}^{\infty} s_T(t) s_2(t - t) dt \qquad (4.2\text{-}5)$$

Using equations (4.2-4) and (4.2-5), the correlator of the output can be expressed as

$$w(t) = \sum_{i=1}^{N} a_i p(t - t_i) e^{jq_i} \qquad (4.2\text{-}6)$$

Equation (4.2-6) demonstrates that the power delay profile represented by $w(t)$ can be simulated using the equivalent probing signal and the attenuation, delay, and phase characteristics of the channel. This method is advantageous because the equivalent probing signal $p(t)$ needs to be computed only once for a fixed configuration of the measurement system. Subsequent simulations of power delay profiles can use (4.2-6) with the same probing signal $p(t)$ to produce simulated power delay profiles for different channels without the need to recalculate a correlation function. This significantly decreases the time required for simulation of power delay profiles using several different channel impulse responses. Figure 4.2-1 shows a block diagram of the simulation that uses a probing signal to produce simulated power delay profiles. The probing signal generator in Figure 4.2-1 may, for example, produce the autocorrelation function of a PN sequence to simulate channel measurements. If the theoretical autocorrelation function of a PN sequence is used as the probing signal, then the simulator produces identical results to the simulator presented in Section 4.1.

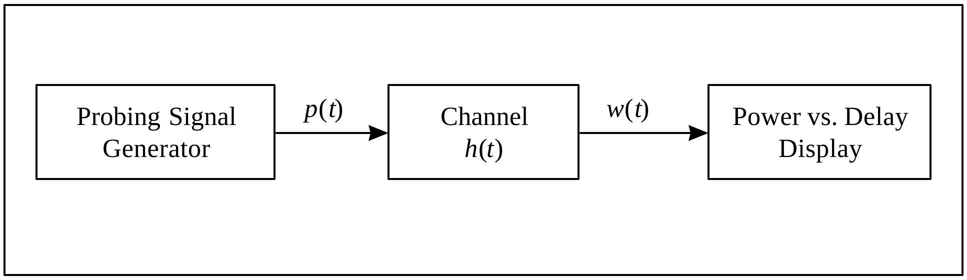


Figure 4.2-1. This simulator uses the probing signal to simulate power delay profiles. The channel definition $h(t)$ may be changed and another power delay profile may be simulated without the need to compute the probing signal again.

Figure 4.2-2 shows a plot of the probing signal used for a simulation of a sliding correlator measurement system employing an ideal correlator at the receiver. This plot was produced for a 127-chip PN sequence clocked at 100 MHz. The base of the correlation peak is 20 ns, equal to the resolution of a 100 MHz sliding correlator system. The simulated power delay profile shown in Figure 4.2-3 was created using the probing signal shown in Figure 4.2-2 and a channel impulse response defined to have four multiple paths. The propagation delays defined for the channel are 200 ns, 300 ns, 600 ns, and 800 ns, and the corresponding attenuation values are 5 dB, 12 dB, 10 dB, and 13 dB, respectively. This simulated profile can be compared to the profile produced by the previous ideal correlator model (see Figure 4.1-10).

The resulting profiles show the same results; however, the run time for multiple simulations using the probing signal are significantly shorter than the time required to produce power delay profiles by correlating the received signal with the receiver PN sequence each time a new simulation is executed.

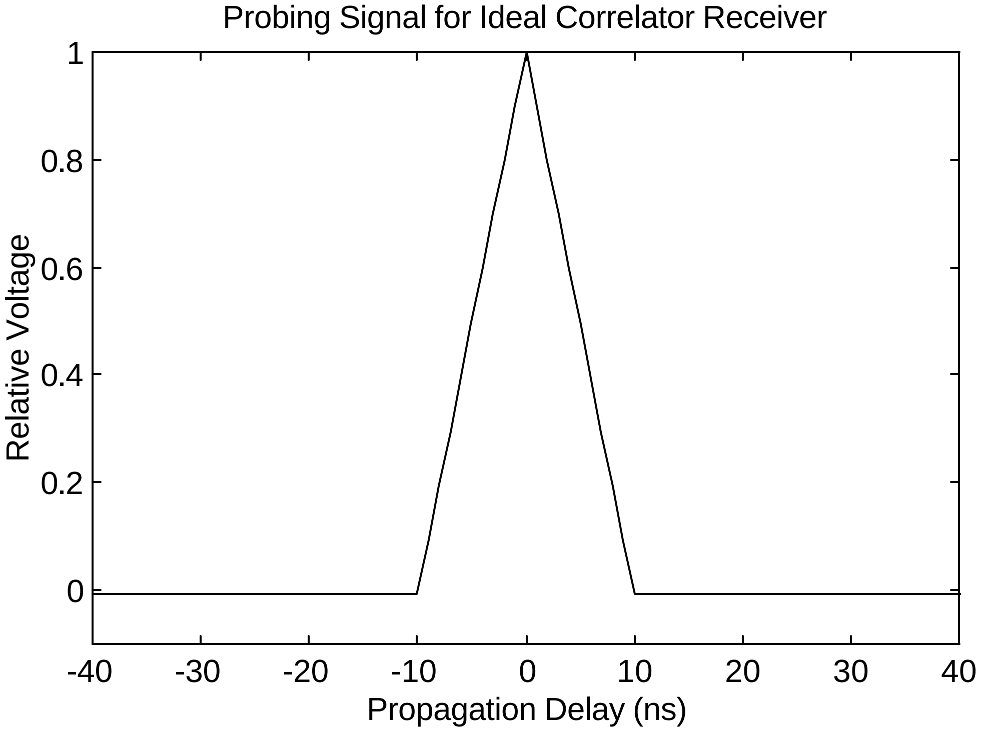


Figure 4.2-2. This plot shows the equivalent probing signal for the sliding correlator simulation using an ideal correlator at the receiver. The probing signal was produced by equation (4.2-5) using a chip rate of 100 MHz and a seven-stage PN sequence generator tapped at [3,7].

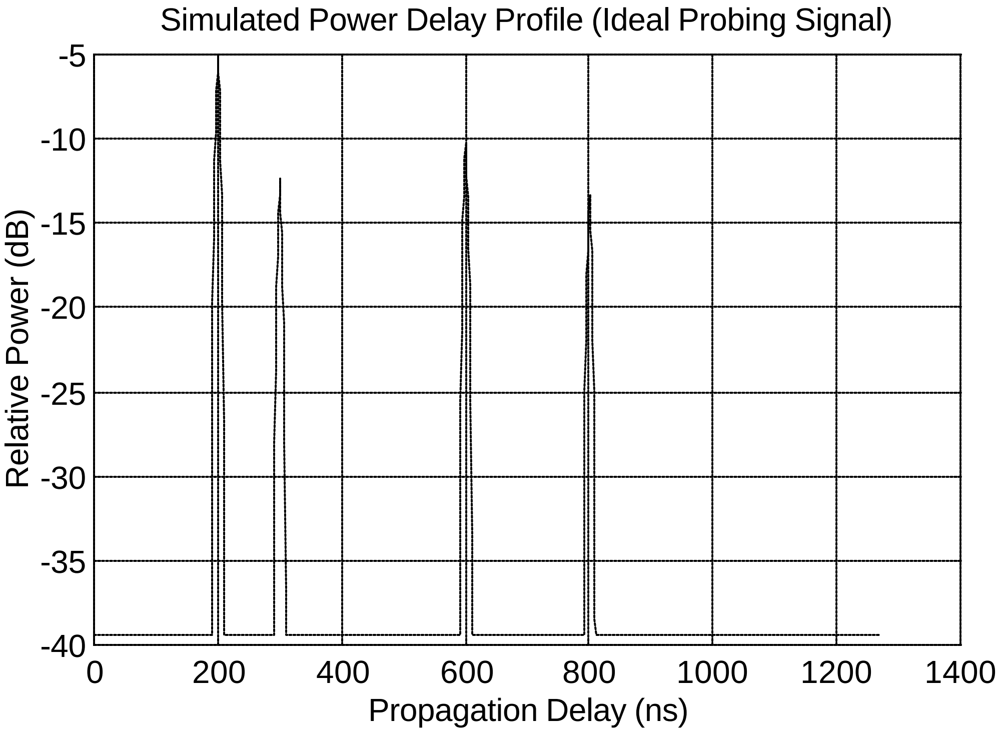


Figure 4.2-3. This power delay profile was simulated using the equivalent probing signal for a receiver employing an ideal correlator. The results of this simulation match the results obtained using a direct implementation of a correlator (see Figure 4.1-10).

Power delay profiles recorded with a real sliding correlator measurement system have distorted low-level signals and a noisy floor not predicted by simulations using the probing signal shown in Figure 4.2-2. Assuming a large signal-to-thermal-noise ratio, the predicted floor of the profile containing a single signal component is approximately $-10\log(l^2)$ relative to the power of the signal component. In practice, the noise floor is typically much higher and can be predicted using a technique that accounts for the noise produced at the receiver due to the minute difference in periods of a transmitter PN chip and a receiver PN chip [24].

It can be observed in the real system that an increase in transmitter power results in an increase in both the multipath components and the noise floor level for a sufficiently high received power level. That is, for a sufficiently high ratio of peak signal component power to the noise floor power, the relative power of the noise floor with respect to the largest multipath component power remains constant. This variable noise floor is called self-noise [24] or correlation noise and can be explained analytically.

The probing signal derived here for a realistic sliding correlator system uses similar techniques described in [24] and [28]. This new probing signal produces a more realistic simulation for the measurement system because it does not assume that the receiver is an ideal correlator using as inputs two PN sequences of equal periods. Figure 4.2-4 shows a model of the sliding correlator measurement system used for the probing signal simulation. The diagram is a baseband representation of the system, and thus the RF carrier is not considered in the model. The simulated chip rates R_{c1} and R_{c2} are the transmitter and receiver chip rates, respectively.

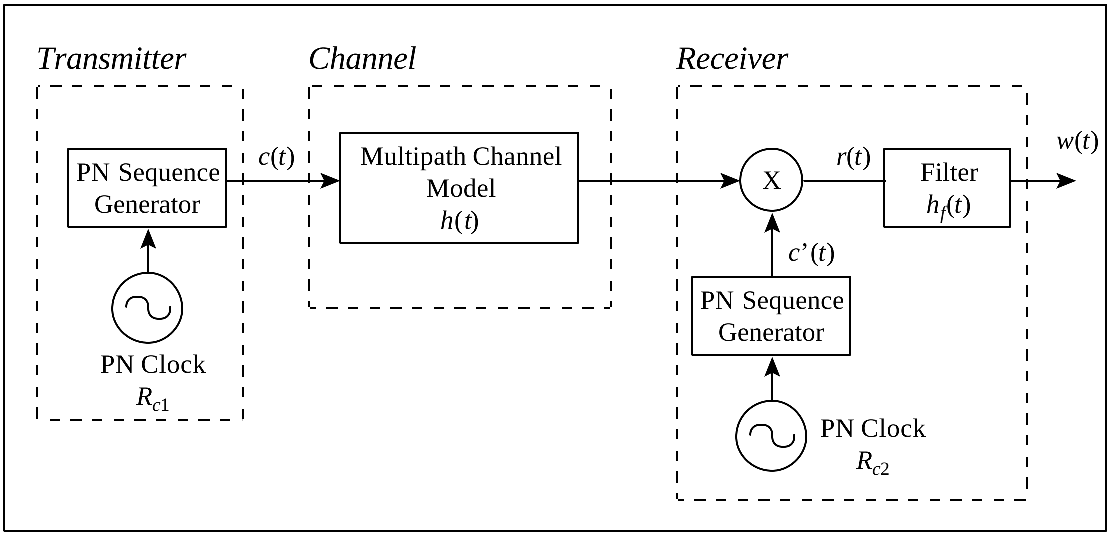


Figure 4.2-4. Baseband equivalent model of the sliding correlator measurement system

The output of the transmitter and receiver PN sequence generator can be expressed as the sum of delayed versions of one period of the PN sequence. That is,

$$c(t) = \sum_{n=-\infty}^{\infty} c_1(t - nT_1) \qquad (4.2\text{-}7)$$

$$c'(t) = \sum_{m=-\infty}^{\infty} c_2(t - mT_2) \qquad (4.2\text{-}8)$$

where expressions for one period for the PN sequence are given by

$$c_1(t) = \sum_{i=0}^{l-1} s_i\, p_1(t - iT_{c1}) \qquad (4.2\text{-}9)$$

$$c_2(t) = \sum_{k=0}^{l-1} s_k\, p_2(t - kT_{c2}) \qquad (4.2\text{-}10)$$

The terms T_{c1} and T_{c2} are the chip periods of the transmitter and receiver PN sequences, given by T_{c1}=1/R_{c1} and T_{c2}=1/R_{c2}. The terms T_1 and T_2 are the periods of the transmitter and receiver PN sequences, given by T_1=lT_{c1} and T_2=lT_{c2} where l is the length of the PN sequence in chips. The coefficients s_i and s_k are the elements (chips) of a valid PN sequence such that

$$s_i, s_k \in \{-1,1\} \quad \text{for} \quad n = 0,1,\cdots,(l-1) \qquad (4.2\text{-}11)$$

The functions $p_1(t)$ and $p_2(t)$ represent the shape of the chips in the PN sequences. These functions are defined by

$$p_1(t) = \begin{cases} 1 & \text{for} \quad 0 \le t \le T_{c1} \\ 0 & \quad\quad\; elsewhere \end{cases} \qquad (4.2\text{-}12)$$

$$p_2(t) = \begin{cases} 1 & \text{for} \quad 0 \le t \le T_{c2} \\ 0 & \quad\quad\; elsewhere \end{cases} \qquad (4.2\text{-}13)$$

The transmitter signal $c(t)$ is convolved with the channel response $h(t)$ to form the received signal. The received signal is multiplied by the receiver PN sequence $c'(t)$ to form the signal $r(t)$ given by

$$r(t) = c'(t)\int_{-\infty}^{\infty} c(t)h(t-t)dt \qquad (4.2\text{-}14)$$

In order to derive the probing signal for this model, the channel response $h(t)$ is set equal to an impulse at t=0, $d(t)$. Then the output of the multiplier can be expressed as

$$r(t) = c(t)c'(t) \tag{4.2-15}$$

using equations (4.2-7) and (4.2-8), this can be rewritten as

$$r(t) = \sum_{n=-\infty}^{\infty} c_1(t - nT_1) \sum_{m=-\infty}^{\infty} c_2(t - mT_2) \tag{4.2-16}$$

Returning to equations (4.2-9) and (4.2-10), the Fourier transforms of the signals $c_1(t)$ and $c_2(t)$ are given by

$$C_1(f) = \sum_{i=0}^{l-1} s_i P_1(f) e^{-j2pfT_{c1}} \tag{4.2-17}$$

$$C_2(f) = \sum_{k=0}^{l-1} s_k P_2(f) e^{-j2pfT_{c2}} \tag{4.2-18}$$

where $P_1(f)$ and $P_2(f)$ are the Fourier transforms of the PN chip definition given in equations (4.2-12) and (4.2-13),

$$P_1(f) = T_{c1}\text{sinc}(fT_{c1}) e^{-jpfT_{c1}} \tag{4.2-19}$$

$$P_2(f) = T_{c2}\text{sinc}(fT_{c2}) e^{-jpfT_{c2}} \tag{4.2-20}$$

Using the relationship between a periodic function in the time domain and its Fourier series coefficients, equation (4.2-16) may be expressed as a complex Fourier series,

$$r(t) = f_1 \sum_{n=-\infty}^{\infty} C_1(nf_1) e^{j2pnf_1 t} \cdot f_2 \sum_{m=-\infty}^{\infty} C_2(mf_2) e^{j2pmf_2 t} \tag{4.2-21}$$

or

$$r(t) = f_1 f_2 \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} C_1(nf_1) C_2(mf_2) e^{j2p(nf_1 + mf_2)t} \tag{4.2-22}$$

The Fourier transform of the signal $r(t)$ can then be expressed as

$$R(f) = f_1 f_2 \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} C_1(nf_1) C_2(mf_2) d(f - (nf_1 + mf_2)) \qquad (4.2\text{-}23)$$

Now that the multiplier output has been derived, an expression for the output of the filter can be calculated using the filter impulse response $h_f(t)$. Given that $H_f(f)$ is the Fourier transform of the filter impulse response $h_f(t)$, the output signal w(t) can be expressed as

$$w(t) = f_1 f_2 \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} C_1(nf_1) H_f(nf_1 + mf_2) C_2(mf_2) e^{j2p(nf_1 + mf_2)t} \qquad (4.2\text{-}24)$$

Note that since the delta functions in equation (4.2-23) occur only at nf_1+$mf_{2,}$ the effect of the filter transfer function only needs to be included in equation (4.2-24) at frequencies nf_1+mf_2. Also note that the output signal is a function of the narrowband filter used to implement $H_f(f)$ in the sliding correlator system. If an RC lowpass filter is used with the transfer function

$$H_f(f) = \frac{1}{1 + j2pfRC} \qquad (4.2\text{-}25)$$

then the output signal becomes

$$w_{RC}(t) = f_1 f_2 \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \frac{C_1(nf_1) C_2(mf_2)}{j2p(nf_1 + mf_2)RC + 1} e^{j2p(nf_1 + mf_2)t} \qquad (4.2\text{-}26)$$

This output signal was derived using $d(t)$ as the channel impulse response. Any constant multiplied by the impulse response to represent an attenuation by the channel will simply carry through the previous calculations and appear as a constant coefficient in (4.2-24) and (4.2-26). A delay t induced by the channel is included by replacing $r(t)$=$c(t)c'(t)$ with $r(t)$=c(t-t)$c'(t)$, which has the effect of delaying the correlation peak in the output signal $w(t)$. Because the two sequences c(t-t) and $c(t)$ maximally align with $c'(t)$ at different chip intervals within $c'(t)$, the correlation peak and noise floor in the output produced by the two input sequences are not exactly shifted and delayed versions with respect to each other. The output will, however, have the same properties with respect to correlation peaks (e.g., width and amplitude) and noise floor level because the correlation characteristics of the PN sequence remain unchanged when the sequence is shifted to simulate a delay. Therefore, instead of simulating similar output signals for each delay, the output signal for the zero-delay case is used as a reasonable estimate for the probing signal for equation (4.2-6). Hence, the probing signal for a system which implements an RC filter at the receiver is given by

$$p_{RC}(t) = f_1 f_2 \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \frac{C_1(nf_1)C_2(mf_2)}{j2p(nf_1+mf_2)RC+1} e^{j2p(nf_1+mf_2)t} \qquad (4.2\text{-}27)$$

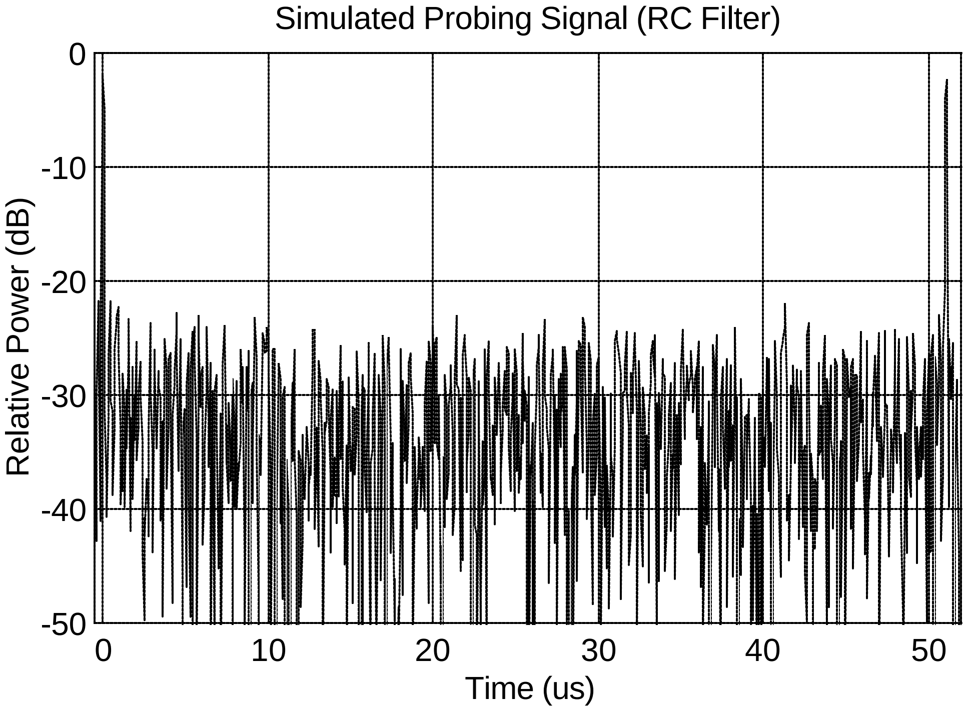


Figure 4.2-5. This probing signal was generated using an RC filter with a 3 dB baseband bandwidth of 10 KHz for correlation in the receiver section. The PN sequence was simulated using a 9-stage shift register tapped at [4,9].

Figure 4.2-5 illustrates slightly more than one full period of the probing signal generated for a sliding correlator using an RC filter with a baseband bandwidth of 10 KHz. A 9-stage shift register tapped at [4,9] was used to produce the PN sequence. The plot clearly shows the existence of self-noise between the two narrow correlation peaks. The plot also correctly shows a 51.1 μs unambiguous measurement range between correlation peaks.

A more realistic filter used for the system simulation is a Gaussian filter with the transfer function given by

$$H(f) = \exp(-a^2 f^2) \qquad (4.2\text{-}28)$$

This type of filter is used by the Tektronix 2782, the spectrum analyzer used for the measurement system implementation presented in Chapter 3. The resolution bandwidth setting of the spectrum analyzer is actually the 6 dB cutoff bandwidth of a Gaussian filter [22]. Figure 4.2-6 shows a comparison between the theoretical response of a Gaussian filter and the measured characteristic of the Tektronix 2782 spectrum analyzer in order to validate the use of (4.2-28) to produce a more realistic probing signal. Using a Gaussian filter as the transfer function in equation (4.2-24) yields the expression

$$p_G(t) = f_1 f_2 \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} C_1(nf_1) e^{-a^2 (nf_1+mf_2)^2} C_2(mf_2) e^{j2p(nf_1+mf_2)t} \qquad (4.2\text{-}29)$$

This expression was used to produce Figure 4.2-7 which shows slightly more than one period of the probing signal produced using a Gaussian filter with a 6 dB bandwidth of 10 KHz. A 511-chip PN sequence was clocked at 10 MHz to produce the transmitter signal, and the same sequence was clocked at 9.990 MHz to produce the input to the multiplier at the receiver. Because of the better filter characteristic approximation, this effective probing signal is a better approximation than the probing signal shown in Figure 4.2-5 for modeling the sliding correlator systems like the one presented in Chapter 2.

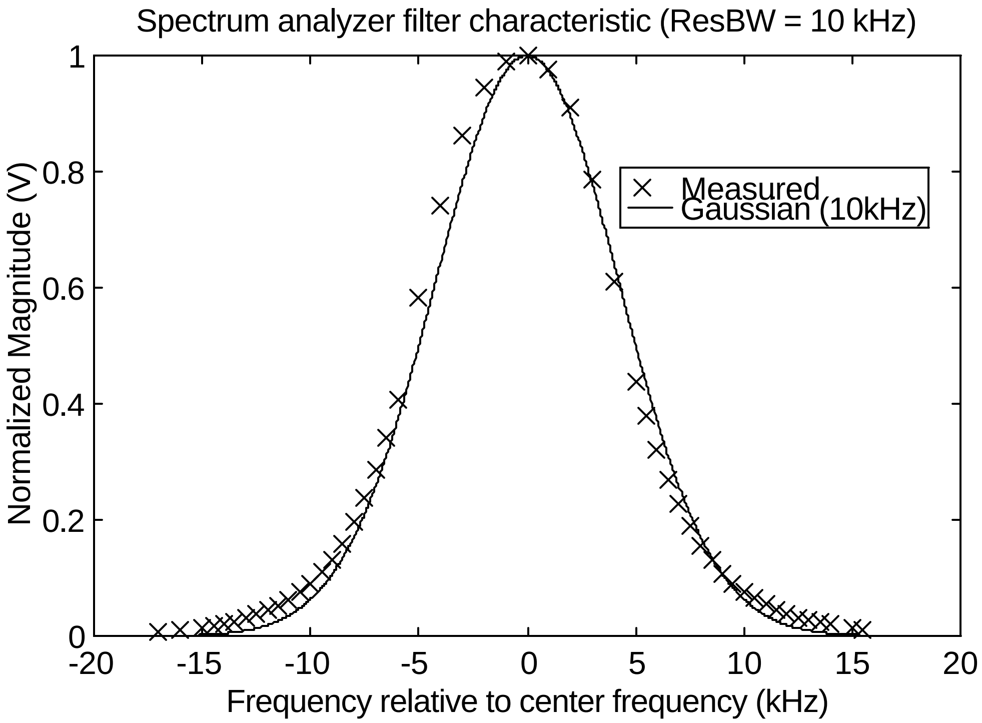


Figure 4.2-6. The solid line represents the theoretical response of a Gaussian filter with a 6 dB bandwidth of 10 KHz. The points marked by 'x' symbols represent the measured response of the spectrum analyzer filter using a resolution bandwidth of 10 KHz.

Figure 4.2-8 shows a power delay profile produced by a simulation of a sliding correlator system employing a 9-stage PN sequence generator and a chip rate of 10 MHz at the transmitter. The Gaussian filter probing signal was used along with an implementation of equation (4.2-6) for this simulation. The 511-chip PN sequence at the receiver was clocked at 9.990 MHz, resulting in a slide factor of 1000. The plot illustrates a power delay profile for a typical channel measured in the rail yard environment described in Chapter 3. The propagation delays defined for the channel were 700ns, 1000 ns, 1400 ns, 2000 ns, 2300 ns, 3700 ns, and the corresponding attenuation values relative to the largest signal component are 0 dB, 10 dB, 2 dB, 15 dB, 10 dB, and 20 dB, respectively. Note that characteristics of this simulated power delay profile match those of measured profiles. For example, the signal components are slightly wider than the expected 200 ns resolution, and the peak noise floor level is in the range of 20 dB to 30 dB below the power of the peak signal component rather than the 42 dB level predicted by $-10\log(l^2)$. Also, the low-level components are slightly distorted by the noise floor produced by the self-noise of the high-level components.

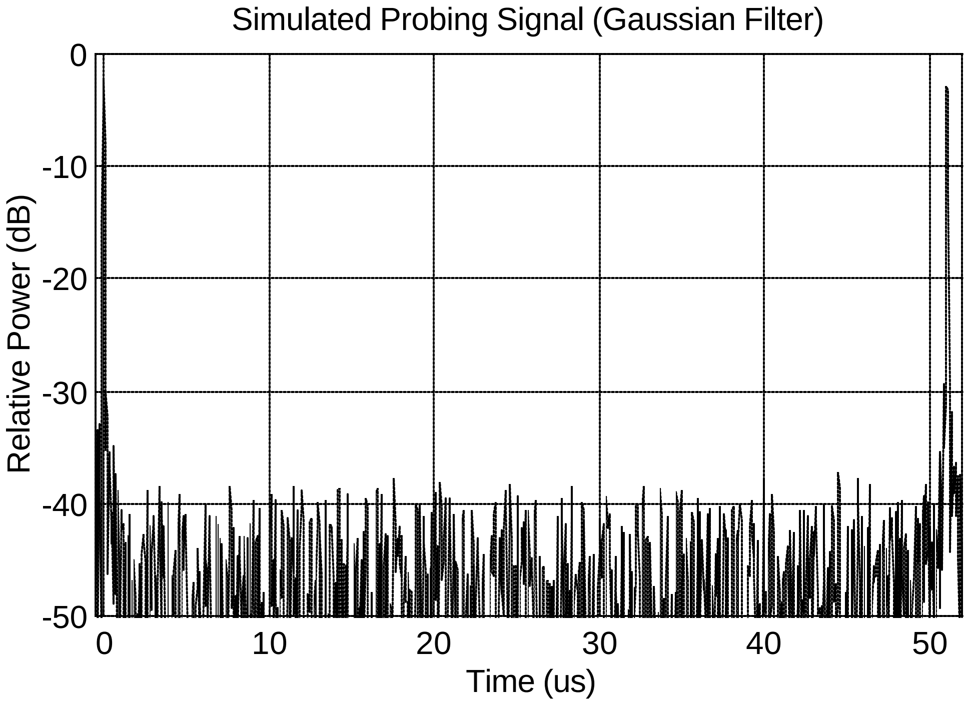


Figure 4.2-7. This probing signal was generated using a Gaussian filter with a 6 dB baseband bandwidth of 10 KHz for correlation in the receiver section. The PN sequence was simulated using a 9-stage shift register tapped at [4,9].

The probing signal used for the simulation of the power delay profile shown in Figure 4.2-8 required approximately five hours of simulation time on a 200 MHz Pentium-Pro computer. MATLAB was used to implement the simulation. The time range for the probing signal extended from 0 μs to 51.1 μs, one full PN sequence period. The probing signal was sampled such that each correlation peak was represented by 40 samples, resulting in 10,220 samples required for the entire probing signal. Although the probing signal requires significant simulation time, power delay profile simulations require much less time once the probing signal samples have been generated. The MATLAB simulation of the power delay profile shown in Figure 4.2-8 executed in under 1 second on the 200 MHz Pentium-Pro computer.

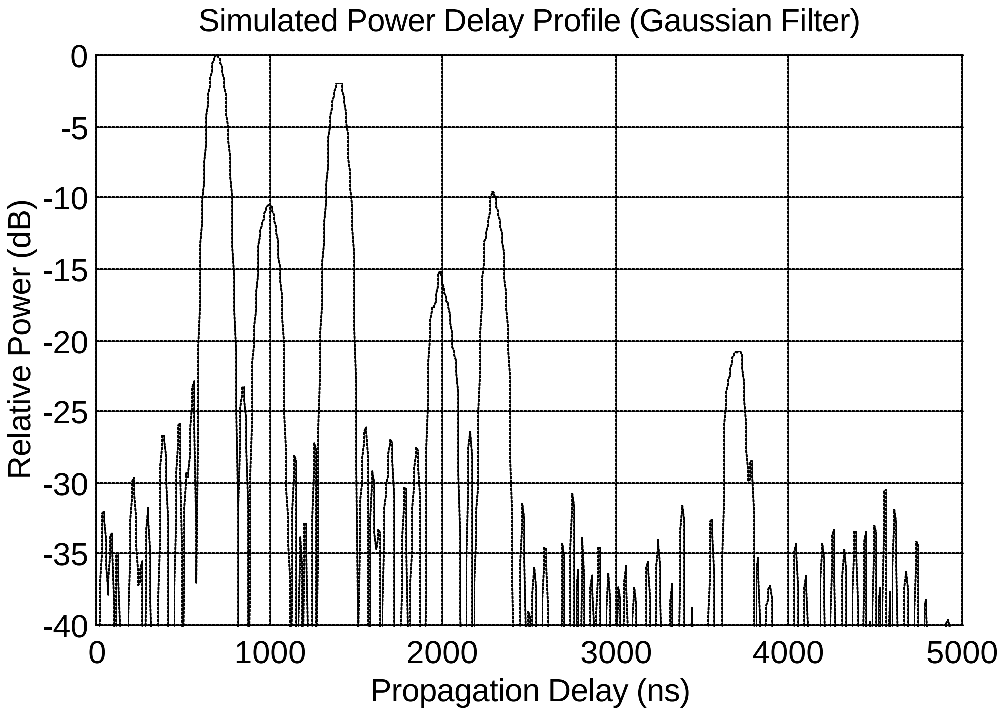


Figure 4.2-8. This power delay profile was simulated using a 511-chip PN sequence clocked at 10 MHz at the transmitter. The PN sequence at the receiver was clocked at 9.990 MHz, resulting in a slide factor of 1000. The simulation realistically represents the components and noise floor observed in measured power delay profiles.

Figure 4.2-9 illustrates a simulated power delay profile for a transmitter and receiver separated by approximately 1 mile in distance. The line of sigh path is significantly obstructed and multipath components are dominant in the profile. The propagation delays used for this simulated channel 5200 ns, 5700 ns, 6500 ns, 6750 ns, 8100 ns, 8500 ns, 9000 ns, and 9500 ns, and the corresponding attenuation values relative to the largest signal component are 10 dB, 0 dB, 20 dB, 5 dB, 10 dB, 13 dB, 20 dB, 15 dB,, respectively. The probing signal used to produce the power delay profile in Figure 4.2-8 was used again here.

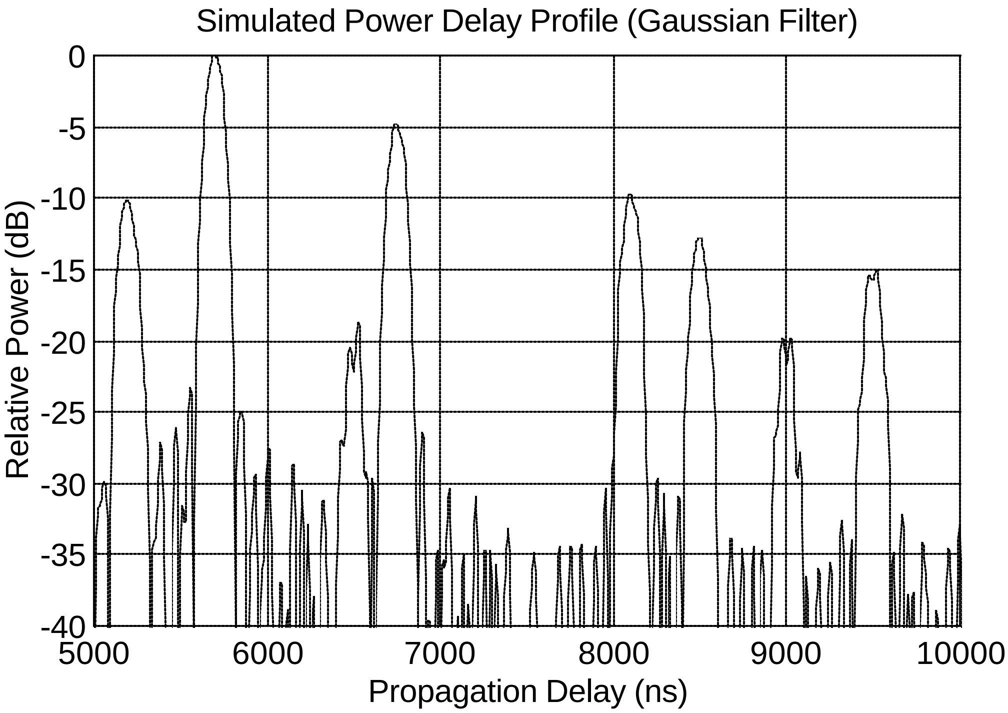


Figure 4.2-9. This power delay profile was simulated using the same probing signal used to produce the profile in Figure 4.2-5. The probing signal did not need to be generated a second time, significantly reducing the simulation time when reevaluating channel responses.

These simulations can be repeated for any number of channel impulse responses and many sliding correlator system configurations. The simulated power delay profiles may be used for testing any application that uses similar profiles to produce channel response estimates or to validate algorithms designed to track the strength and delay of multipath signal components in the received signal. The non-ideal correlator implementation produces output signals which closely resemble the output of the real system and are better approximations than the power delay profiles produced by the ideal correlator simulation. The self-noise characteristic of the sliding correlator profiles illustrated in this section demonstrate why the theoretical limit of $-10\log(l^2)$ is not obtained for high signal-to-thermal-noise ratios when implementing sliding correlation because of the small difference between transmitter and receiver PN sequence periods. In addition to the inclusion of self-noise between signal component peaks in the simulated profiles, the simulation also correctly accounts for distortion of low-level signal components due to the noise.

4.3 Summary

This chapter has presented two methods for simulating the power delay profiles produced by a sliding correlator measurement system. The first method assumes that the receiver acts as an ideal correlator. A baseband representation of the sliding correlator was used which makes simulation results independent of carrier frequency. The simulation was implemented using MATLAB, and a method of computing the output using vector operations to implement the system components was developed to reduce the time required to execute a simulation. Simulated signals were compared to signals within the real system to validate the system model, and simulated power delay profiles demonstrated that paths defined by a known channel impulse response were correctly represented in the simulation results.

The simple ideal-correlator model produces reasonable estimates of the correlation peaks produced by the system, but it fails to predict the noise floor level between signal components and cannot predict the shape distortion of low-level signal components due to the addition of self-noise. Also, a correlation function must be produced for each simulated power delay profile, thus causing long execution times for systems which use long PN sequences. For systems using relatively short sequences, however, the ideal-correlator model produces wideband channel data in a reasonable amount of time.

The second method of simulation presented in this chapter used an equivalent probing signal to generate simulated power delay profiles. Once produced by a measurement system model, the equivalent probing signal can be transmitted through the simulated channel to produce realistic power delay profiles. Two equivalent probing signals were derived: a signal for an ideal correlator receiver and a signal for a receiver which uses a mixer and a filter to implement the correlator functionality. The ideal correlator probing signal was used to compare the output of the equivalent probing signal simulation to the ideal correlator simulation. Power delay profiles produced by the two methods were identical when the same channel was used. The probing signal for the system which uses a mixer and filter instead of an ideal correlator was derived to produce more accurate estimates of the power delay profile measured with the real system.

A baseband model of the sliding correlator was presented which uses a mixer followed by a filter with an arbitrary impulse response. This model was used to produce the probing signals for systems which employ either an RC filter or a Gaussian filter. The Gaussian filter was shown to more accurately represent the filter in the Tektronix 2782 spectrum analyzer, a component used in the hardware implementation of the sliding correlator presented in Chapter 3. Simulated power delay profiles demonstrated the increase in the noise level due to unequal PN sequence periods at the transmitter and receiver. Distortion of low-level signal components due to the noise floor was also illustrated.

In conclusion, the simulations described in this chapter may be used for testing of algorithms which operate on power delay profile data. Also, study of the simulations and simulated results enables a better understanding of the sliding correlator measurement technique for operating existing systems and for developing future measurement system implementations.

Chapter 5

Wideband Channel Measurement Post Processing

Raw power and delay data recorded with a sliding correlator measurement system must be processed to form meaningful results. Presented in this chapter are methods used to process sliding correlator data. The type of processing depends upon how measurement results are used. For measurements which are intended to characterize an environment for system design or optimization, results that statistically quantify propagation characteristics over geographical regions may be required from the processing algorithms. Alternatively, measurements which are intended to test individual, site-specific cases may be processed to produce results which report delay and strength of individual multipath components. For example, propagation models which predict the arrival of individual multipath components at a particular site may be validated using individual measurements rather than statistical summaries of a region. In this chapter, these two different requirements of results are addressed. First, a conventional technique for processing power delay profiles to produce statistical measures of an environment is discussed. Problems associated with this technique are also discussed. Then a method of automating the process of detecting the strength and delay of individual multipath components in a power delay profile is presented. This method is tested using profiles produced by the sliding correlator simulation described in the preceding chapter and also profiles measured during the study described in Chapter 3.

5.1 Conventional Post-Processing of Measured Profiles

Chapter 2 presented several parameters used by propagation researchers to characterize delay and attenuation of radio channels. Of the parameters presented, RMS delay spread and log-distance path loss exponents are the most commonly used to describe the results of measurement surveys in published propagation research. Many radio channel studies have been performed which process power delay profile data to provide RMS delay spread and path loss exponent statistics for wireless system design (e.g., [5], [7], [18], [19]). Presented in this section is a detailed description of the conventional method used to calculate RMS delay spread and path loss exponents from measured power delay profiles. The MPRG sliding correlator system is used as an example of a typical sliding correlator system.

A power delay profile measured by the MPRG sliding correlator system is represented by a set of 1024 power and time data points. The MPRG system, along with other systems which use digital oscilloscopes to perform waveform acquisition, samples the power delay profile waveform in uniformly spaced power and time increments. That is, a uniform quantizer is used to produce samples of the power delay profile, and the time difference between samples Δt_r is constant. If a calibration power delay profile $p_{cal}(t)$ is sampled with the same period so that $\Delta t_{cal} = \Delta t_r$ to form the samples of $p_{cal}(k)$, then the received power equation given in Chapter 2 as

$$P_r = P_{cal} \frac{\sum_k p_r(k) \Delta t_r}{\sum_k p_{cal}(k) \Delta t_{cal}} \qquad (5.1\text{-}1)$$

can be expressed as

$$P_r = P_{cal} \frac{\sum_k p_r(k)}{\sum_k p_{cal}(k)} \qquad (5.1\text{-}2)$$

where $p_r(k)$ is the sampled power delay profile recorded during a channel measurements. From calculations of received power using many profiles, the path loss exponent n (as discussed in Chapter 2) and a variance are calculated to determine the statistical trend of path loss [2]. Recall that RMS delay spread is calculated from continuous waveforms using

$$s_t = \sqrt{\overline{t^2} - (\bar{t})^2} \qquad (5.1\text{-}3)$$

where

$$\bar{t} = \frac{\int t p_r(t)dt}{\int p_r(t)dt} \qquad (5.1\text{-}4)$$

and

$$\overline{t^2} = \frac{\int t^2 p_r(t)dt}{\int p_r(t)dt} \qquad (5.1\text{-}5)$$

For sampled waveforms, the RMS delay spread calculation is performed using

$$\bar{t} = \sum_k \frac{t(k)p_r(k)}{p_r(k)} \qquad (5.1\text{-}6)$$

and

$$\overline{t^2} = \sum_k \frac{t^2(k)p_r(k)}{p_r(k)} \qquad (5.1\text{-}7)$$

Equations (5.1-6) and (5.1-7) require that *each sample point with non-zero power represents a sample of a received signal component*. In reality, nearly every point in a power delay profile has a non-zero power value because of the existence of a noise floor. Including the samples of the noise floor causes errors in calculated channel parameters as discussed in the following section.

One method used to remove the noise floor samples from the power delay profile is to define a power threshold below which all sampled power is assigned a value of 0 Watts. Applying this noise threshold has the effect of removing the noise samples from the calculations. The noise threshold is typically defined using one of two techniques. One technique is to set the noise threshold to a constant level *X* dB below the peak signal component in the profile [33], where *X* dB is an estimate of the power difference between the strongest arriving component and the noise floor. As long as *X* does not exceed this range, the noise samples will be eliminated from the channel parameter calculations. A problem arises when measurements are performed while the measurement system receiver is severely shadowed, and the low strength of all received signal components causes the range between the noise floor and peak signal component to shrink compared to the unobstructed path case. In this case, the value chosen for *X* is likely to overestimate range between the strongest component peak and the noise level, and a large number of noise samples are included in the calculations.

The second technique which works satisfactorily for both self-noise and thermal noise cases involves dynamically defining a noise threshold level for each measured profile. The noise threshold is set equal to the peak power level detected in the portion of a power delay profile where it is highly unlikely that any signal components exist. Delay ranges at the beginning and end of a profile serve well as component-free regions. For example, if measurements are consistently performed with a transmitter-receiver separation of greater than 500 feet, then the range of absolute propagation delay between 0 ns and approximately 500 ns contains only noise samples since the line of sight component has a delay of at least 500 ns (see Figure 5.1-1). Also, if the maximum measurable delay of the system is much larger than the maximum expected delay of signal components, then a range of delay toward the end of a profile is unlikely to contain signal components. Once the peak noise in the "noise-only" region is detected, all profile samples which represent power below the peak noise power are set to 0 Watts and thereby eliminated from the channel parameter calculations. Although some noise peaks may exist above this dynamically defined threshold and are included in the calculations, the contribution of noise samples to the channel parameter calculations is minor compared to the contribution when *all* noise samples are included.

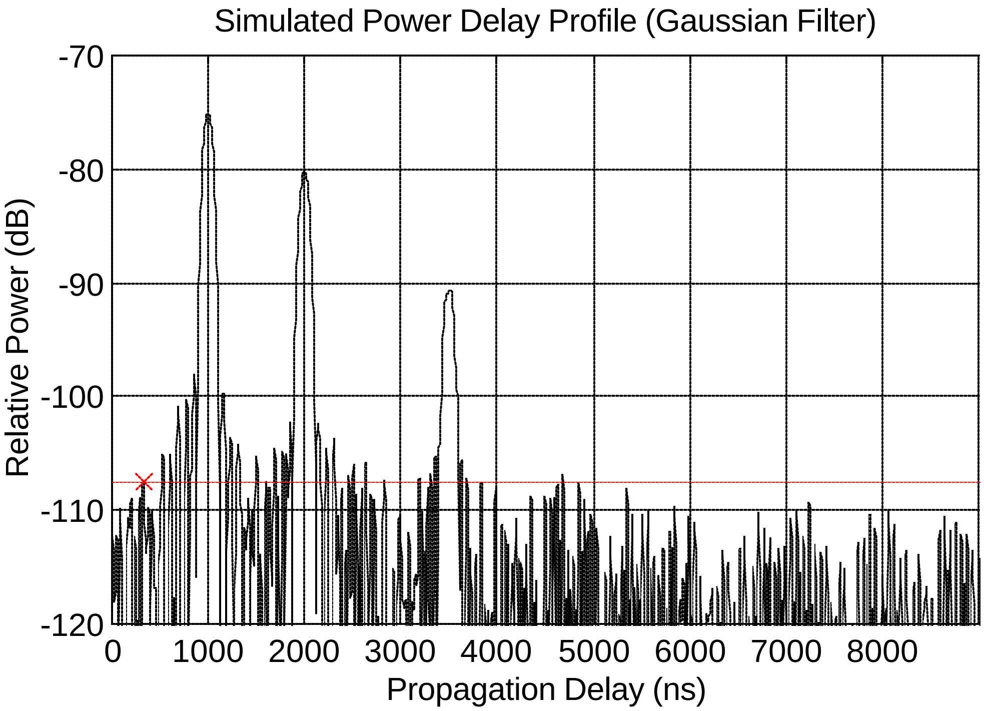


Figure 5.1-1. For wideband measurements performed with the transmitter and receiver consistently separated by more than a distance *d*, the portion of the power delay profile which represents propagation delays across distances less than *d* can be used to set the noise threshold. In this figure, the peak power within a delay range of 0 ns to 500 ns is used to define the noise threshold.

Assume that both thermal and self-noise samples have been removed from the power delay profiles yielding a profile which contains only samples of correlation peaks. Even in this case, minor inaccuracies can exist which are caused by the fact that a power delay profile is only an estimate of the true channel impulse response. In Chapter 2, it was explained that average received wideband power can be expressed as the sum of powers of individual multipath components given by

$$P_w = \sum_{i=1}^{N} P_i \qquad (5.1\text{-}8)$$

where P_i is the power of the i^{th} out of N received multipath components. Equation (5.1-1) performs the operation described by equation (5.1-8) by summing the area under each received signal component and relating that area to the area under a calibration profile associated with a known received power. Hence, the accuracy of the received power calculation is dependent upon accurate calculations of the area beneath the curve of received signal components. This seems to lead to the conclusion that for accurate received power estimates derived from power delay profiles which contain peaks of finite width to represent signal components, a high sample rate should be used to sample $p_r(t)$ so that the summation in equation (5.1-1) more closely approximates an integral to determine the area of the component. That is, it appears that a large number of samples per signal component should be used so that received power calculations derived from $p_r(k)$ are consistent and accurate.

Now consider factors which affect the accuracy of RMS delay spread values derived from power delay profiles using equation (5.1-3). The RMS delay spread of a radio channel increases with the number of widely spread multipath components present at the receiver. For the case where only one signal component exists at a receiver, the RMS delay spread is equal to zero since no time dispersion is induced by multipath in the radio channel. It can be observed that RMS delay spread calculated from a power delay profile using equations (5.1-3) though (5.1-5) will always be nonzero when two or more sample points are used to represent a single signal component in a profile, even when only one component exists in the profile. Although this effect on the overall RMS delay spread result is minor when many signal components exist in the profile, it demonstrates that RMS delay spread is ideally calculated using power delay profiles which use a single sample points to represent the peak of each signal component (see Figure 5.1-2). Using only a single sample per component causes each component to be weighted equally; normally the larger components are erroneously weighted more heavily because larger components contain more samples above the noise threshold compared to low-level components. Note that using a single sample point per component seemingly conflicts with the previous notion that a large number of samples per component should be used in order to obtain accurate received power measurements using equation (5.1-1). However, equation (5.1-8) is an appropriate replacement for (5.1-1) for determining received power if a single point per component is used.

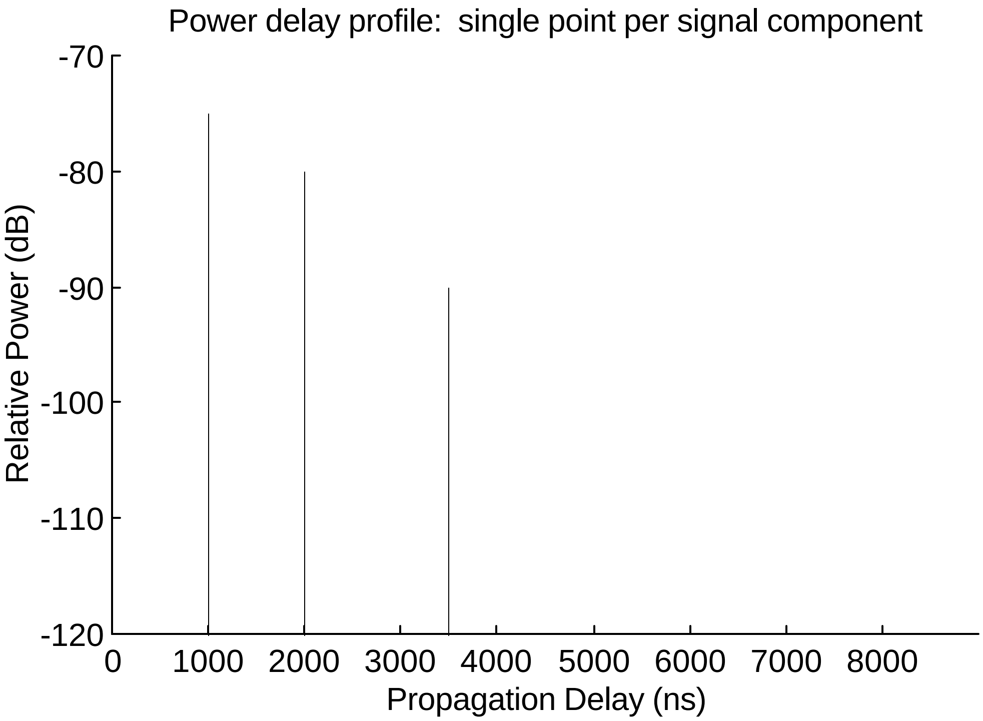


Figure 5.1-2. RMS delay spread and received power calculations are more accurate when the channel is represented by power delay profiles which contain a single sample point to represent the peak of each signal component. These points must be sampled when the PN sequences are maximally aligned.

The preceding discussion suggests that the best format of power delay profiles for deriving channel parameters is one which closely resembles an impulse response with regard to containing an infinitely narrow pulse to represent each signal component arriving at a receiver. Formatting wideband data in such a way eliminates the errors associated with using the area under components to determine received power and solves the problem of weighting more heavily components represented by multiple samples. Inaccuracies and potential practical solutions are studied more in depth in section 5.2. In addition, if signal components can be positively identified in the power delay profile, the noise floor is not an accuracy factor for either RMS delay spread and received power calculations. Elimination of the noise floor and identification of individual signal components is the subject of section 5.3.

5.2 Effects of Noise Threshold on Calculated Channel Parameters

The implementation of a noise threshold reduces the effect of the noise floor on calculated channel parameters by eliminating most of the sample points which are not associated with a received signal component. However, the noise threshold also eliminates data points associated with the portion of component peaks where the PN sequences are not maximally aligned and may eliminate low-level correlation peaks entirely. This results in inaccuracies in the calculated channel parameters. Presented here is an investigation into the effects of the noise threshold on RMS delay spread and received power calculated from power delay profiles.

The sample power delay profile shown in Figure 5.2-1 is used to demonstrate the error in calculated received power caused by the application of a noise threshold. The power delay profile represents the impulse response of a single-path channel with a delay of 1000 ns. The true received power is -80 dBm. Figure 5.2-2 shows the calculated received power versus noise threshold level for the profile shown in Figure 5.2-1. As the noise threshold increases, the calculated received power falls because additional signal component area is discarded from the power calculation. Each downward step of calculated received power as the noise threshold increases represents an additional sample of the power delay profile lost from the received power calculation. It is demonstrated by the plot that notable error occurs only when the noise threshold approaches the peak of the signal component within about 2 dB. This implies that for received power calculations performed using profiles containing many strong signal components, the error in total received power is likely to be negligible as long as the noise threshold is several dB below the strong received signal peaks. In general, as long as sufficient power levels are used at the measurement system transmitter such that at least one of the received components rises several dB above the thermal noise floor, errors in total received power caused by application of a noise threshold can be considered negligible.

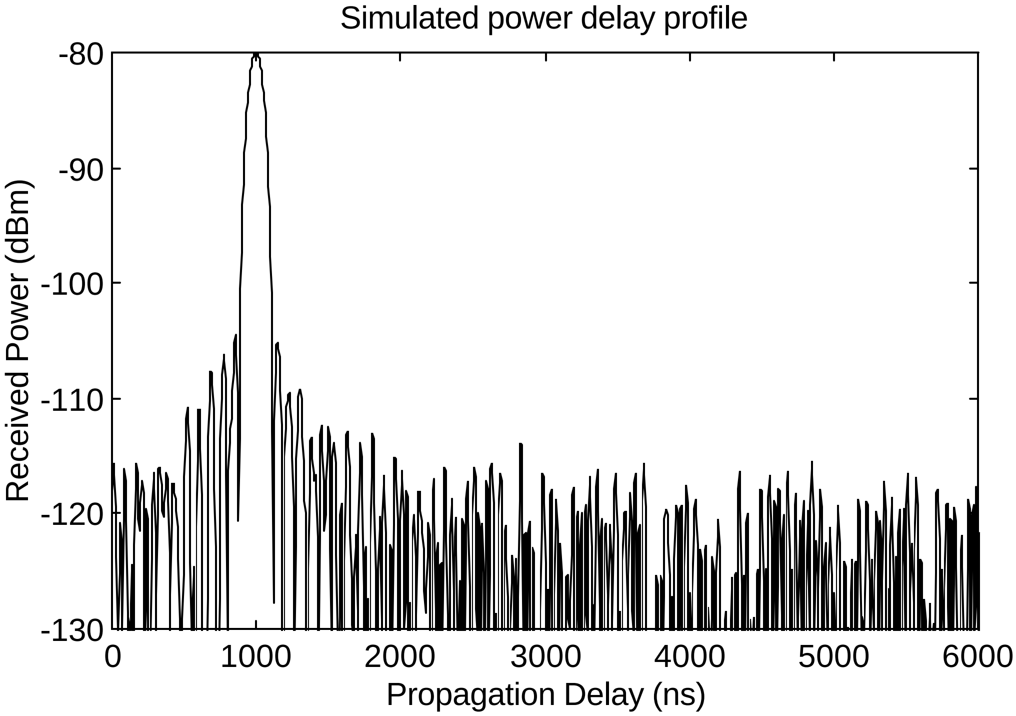


Figure 5.2-1. This power delay profile is used to investigate the effect of variations in noise threshold levels on calculated received power.

Although the inaccuracy in calculated received power as the noise threshold varies is a minor concern in most cases, errors in RMS delay spread due to variations in the noise threshold can be quite significant. Figure 5.2-3 illustrates a power delay profile which is used to demonstrate dependency of RMS delay spread calculations on noise threshold level. This power delay profile was simulated for a 1900 MHz wireless channel with a log-normal model path loss exponent of 3.5. Seven signal components arrive at the receiver with uniformly increasing propagation delay, and the true RMS delay spread for this channel is 517 ns. Using this profile, the RMS delay spread was calculated using noise threshold levels ranging from -170 dB to -125 dB. Figures 5.2-4 and 5.2-5 show the relationship between calculated RMS delay spread and noise threshold level.

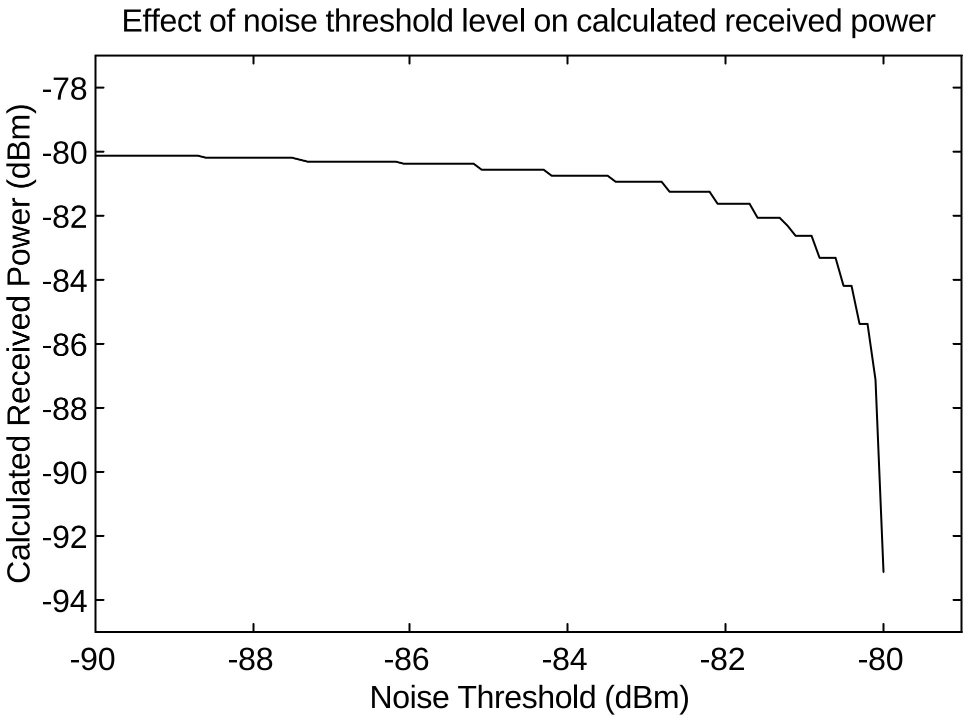


Figure 5.2-2. This plot shows calculated received power versus noise threshold level using the power delay profile in Figure 5.2-1. It is shown that as the noise threshold is increased, the calculated received power underestimates the true received power.

It is evident in Figure 5.2-5 that calculated RMS delay spread is significantly overestimated when the noise threshold is assigned to a level below the peaks of the correlation noise floor. Although the noise floor strength is more than 25 dB below the peak signal component, the large number of sample points causes the calculated RMS delay spread to exceed the true RMS delay spread by nearly a factor of 10. The calculated RMS delay spread rapidly approaches the true value as the noise threshold level is increased above the level of the noise peaks. Figure 5.2-4 shows an expanded plot of RMS delay spread versus noise threshold level, ranging from a power at which the noise threshold level is slightly below some of the largest noise peaks to the power at which the noise threshold level exceeds the largest signal component. The horizontal line indicates the true RMS delay spread for the channel. As expected, the RMS delay spread falls to zero when the noise threshold level rises above the peak of the strongest signal component. As the noise threshold power falls, the RMS delay spread increases as additional samples of the strongest component are included in the calculation. Despite the fact that only one signal component is included in the calculations at this point, the RMS delay spread is erroneously nonzero. As the noise threshold is reduced further, the calculated RMS delay spread increases in steps as the contributions of additional signal components are included the computation. Note that there is only a very narrow range of noise threshold levels where the RMS delay spread is within a few percent of the true value.

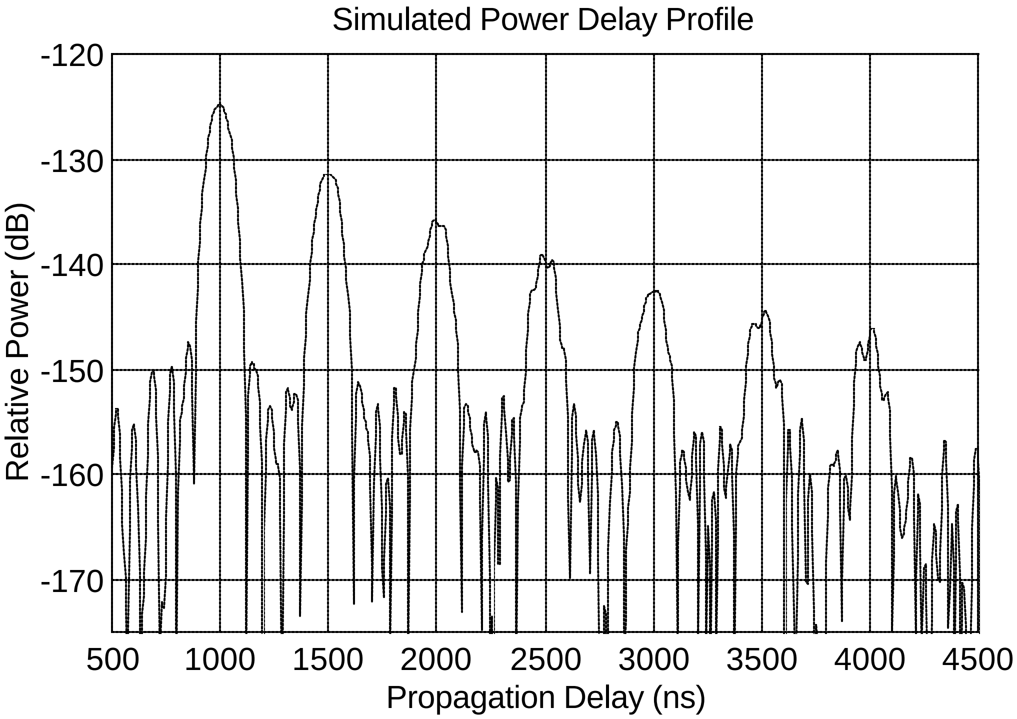


Figure 5.2-3. A 1900 MHz channel with a log-normal path loss exponent of 3.5 was used to produce this power delay profile. This profile is used to investigate the effect of the noise threshold on RMS delay spread calculations.

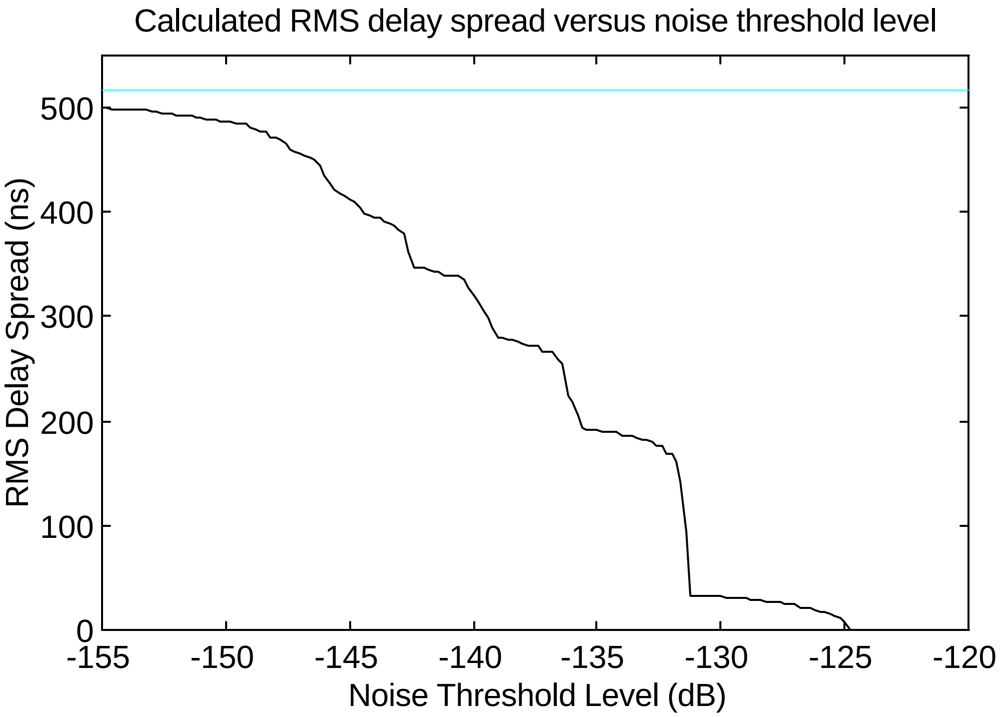


Figure 5.2-4. The dependency of calculated RMS delay spread on noise threshold level is demonstrated here. As the noise threshold is lowered from the peak of the strongest component, the calculated RMS delay spread increases. The horizontal line represents the true RMS delay spread of the channel.

The primary factors which affect the accuracy of RMS delay spread calculations are the existence of a noise floor and the unequal weighting of multipath components with different received power levels. The inclusion of the noise floor causes the calculated RMS delay spread to drastically rise, and the unequal weighting causes RMS delay spread to underestimate the true value when low level received signal components exist. One possible and practical way to eliminate these factors is to perform noise floor suppression and signal component detection as discussed in the following section. These solutions not only attempt to improve the accuracy of calculated channel parameters, but they also represent the channel with discrete amplitude and delays much like an impulse response representation of a wireless channel.

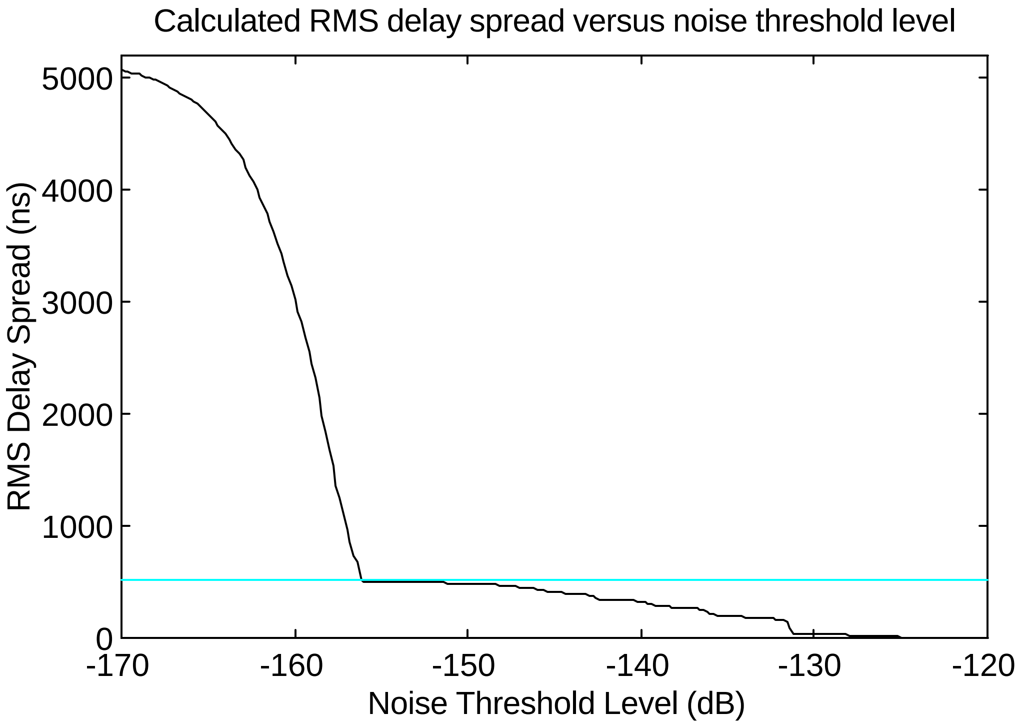


Figure 5.2-5. As the noise threshold is lowered into the noise floor, large errors in calculated RMS delay spread occur. The horizontal line represents the true RMS delay spread of the channel.

5.3 Signal Component Detection Processing for Power Delay Profiles

In this section a method of processing power delay profiles is presented which effectively represents the measured channels as a discrete set of signal strengths and delays much like a theoretical channel impulse response. There are several advantages to representing the measured channel data as a set of discrete strength and delay samples. For example, signal components are positively identified rather than roughly described by finite-width peaks in a typical power delay profile. Also, working with discrete signal components during processing ensures repeatable results, especially when comparing simulated impulse responses (e.g., ray-tracing results) to measured power delay profiles. In addition, representation of a channel measurement with a power delay profile consumes considerably more computer memory for storage and execution time for processing as compared to a discrete profile representation. This is especially important when the wideband channel data must be transmitted over a low bandwidth link (e.g., the serial port link from the measurement hardware to a computer).

Finally, the discrete profile representation conveys a more intuitive picture of the discrete nature of a wireless channel multipath structure. Processing discrete components also assists in eliminating the adverse effects of applying a noise threshold during channel parameter calculations discussed in the previous section.

Figure 5.3-1 shows a high level block diagram of the technique which processes power delay profiles to produce discrete component representations. The technique performs two primary operations on the power delay profile. First, the power delay profile samples are processed by the *noise floor suppression* stage. During this stage, component regions are identified; component regions are the ranges of propagation delay during which received signal components are likely to exist. All sample points outside of the component regions are considered noise and eliminated from the profile. The result is a “noiseless” power delay profile where a majority of the noise samples have been removed. The “noiseless” profile is used as the input to the *signal component detection* stage during which the strength and delay of individual received signal components are identified. The output of the signal component detection stage is a discrete profile which represents all of the signal components identified during processing.

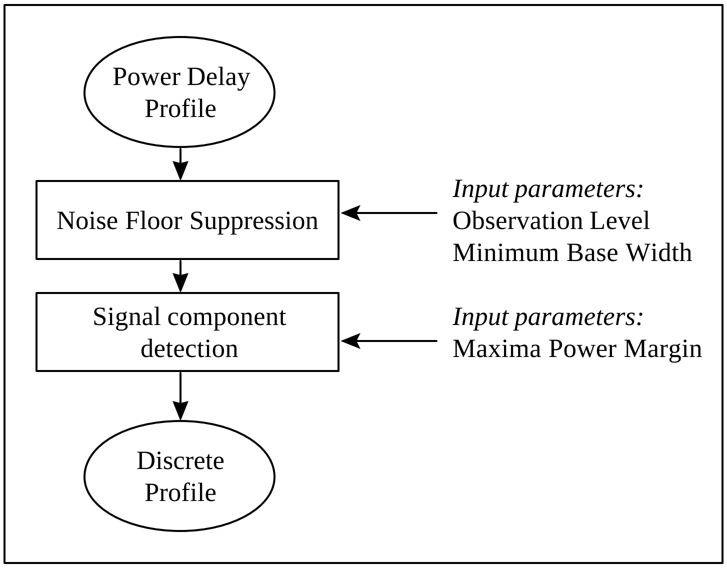


Figure 5.3-1. The power delay profile is used as the input to the noise floor suppression stage where the noise samples are removed from the floor of the profile. The individual signal components are identified during the signal component detection stage to produce the discrete profile.

Figure 5.3-2 illustrates the noise floor suppression algorithm in greater detail. Instead of employing the noise threshold level used for conventional processing, an observation level is defined. Rather than eliminating all sample points below a specified level, the observation level is used to identify where signal components potentially exist in the power delay profile. Figure 5.3-3 illustrates a power delay profile with an observation level defined (the horizontal line through the noise floor is the observation level). This power delay profile is used to demonstrate signal component detection. The delays of the components are 1 μs, 2 μs, 3 μs, 4 μs, 5 μs, and 6 μs, and the strengths of the components are 0 dB, -4 dB, -8 dB, -12 dB, -16, and -20 dB, respectively.

The algorithm first identifies all crossings of the power delay profile trace across the observation level. Next, the delay between the upward crossing (positive derivative) and the following downward crossing (negative derivative) is calculated for all crossings. This array of delays specifies the base width for all peaks rising above the observation level. A second user-defined parameter, minimum base width, is used to search for component regions in the profile. The bases of all peaks in the profile are compared with the minimum base width parameter, and regions where the base width exceeds the minimum base width are identified as component regions. The “x” symbols shown in Figure 5.3-3 indicate the endpoints of the component regions. All samples outside of the component regions are considered noise and are removed from the profile to produce the “noiseless” profile shown in Figure 5.3-4..

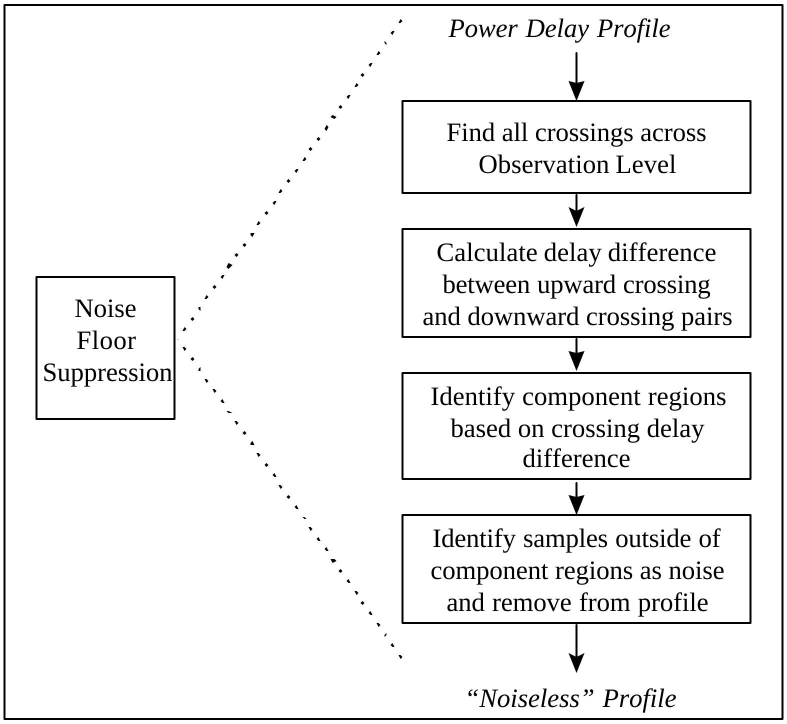

Power Delay Profile
Find all crossings across Observation Level
Calculate delay difference between upward crossing and downward crossing pairs
Identify component regions based on crossing delay difference
Identify samples outside of component regions as noise and remove from profile
"Noiseless" Profile
Noise Floor Suppression

Figure 5.3-2. The first step in noise floor suppression is identification of all crossings across the observation level. The delay difference between upward and downward crossings is compared to the user-specified minimum base width to determine the component regions. All points outside the component regions are eliminated to produce the "noiseless" profile shown in Figure 5.3-4.

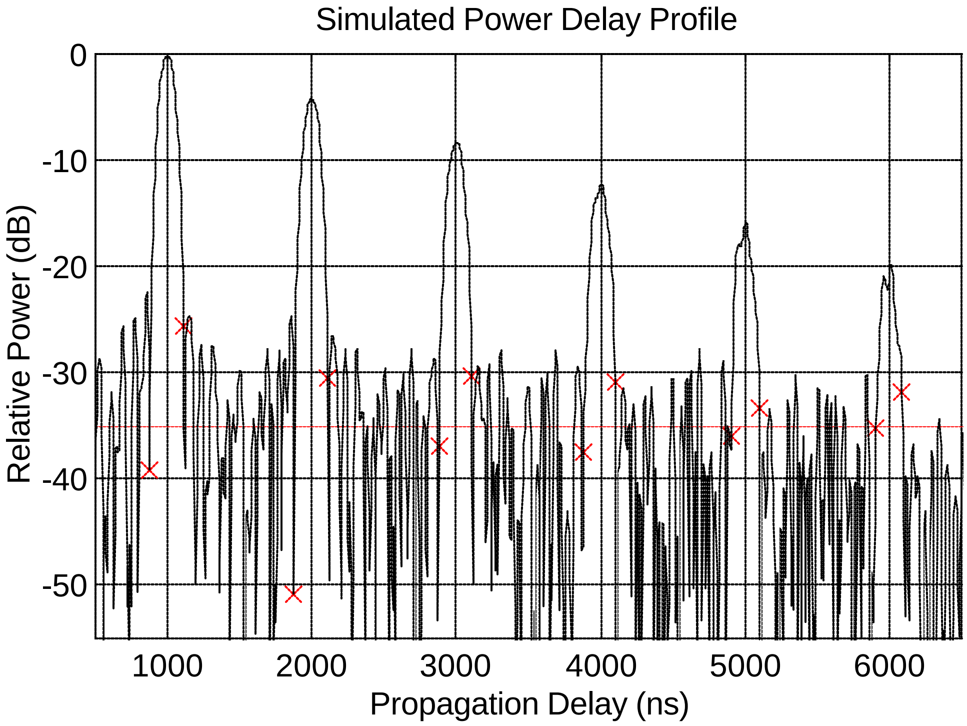


Figure 5.3-3. This power delay profile is used to demonstrate detection of signal components and suppression of the noise floor. The horizontal line is the observation level used during processing, and the "x" symbols identify the endpoints of the detected component regions.

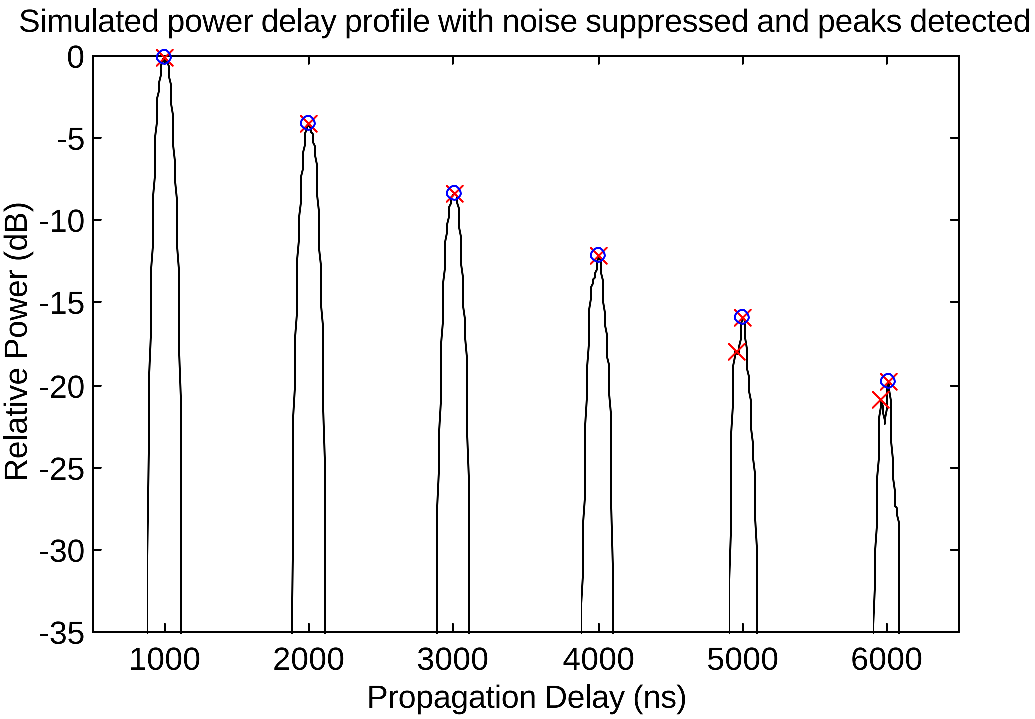


Figure 5.3-4. The noise floor samples have been eliminated from this power delay profile. The “x” symbols in this figure identify the local maxima within the component regions. The circles indicate all maxima which have been identified as received signal components.

Figure 5.3-5 illustrates a block diagram of the component detection algorithm. Once the noise floor samples have been removed from the profile, a search is performed using the remaining samples to find all local maxima. Local maxima are designated by “x” symbols in Figure 5.3-4. These local maxima are the candidates for received signal components. A third user-specified parameter, the maxima power margin, is a quantity which defines the minimum allowable power level difference between the level of a local maximum and the level of the adjacent local minima for the local maximum to be considered a signal component. Any local maximum within a component region which satisfies the maxima power margin criteria is considered a signal component (see Figure 5.3-6). This test handles the case of two or more closely arriving multipath components and reduces the possibility of identifying a noise spike or other anomaly as a signal component. If no local maxima within a component region satisfy the maxima power margin criteria, then the absolute maximum within the component region is assumed to be the signal component peak. The signal components detected in Figure 5.3-4 are identified by the small circles around the local maxima.

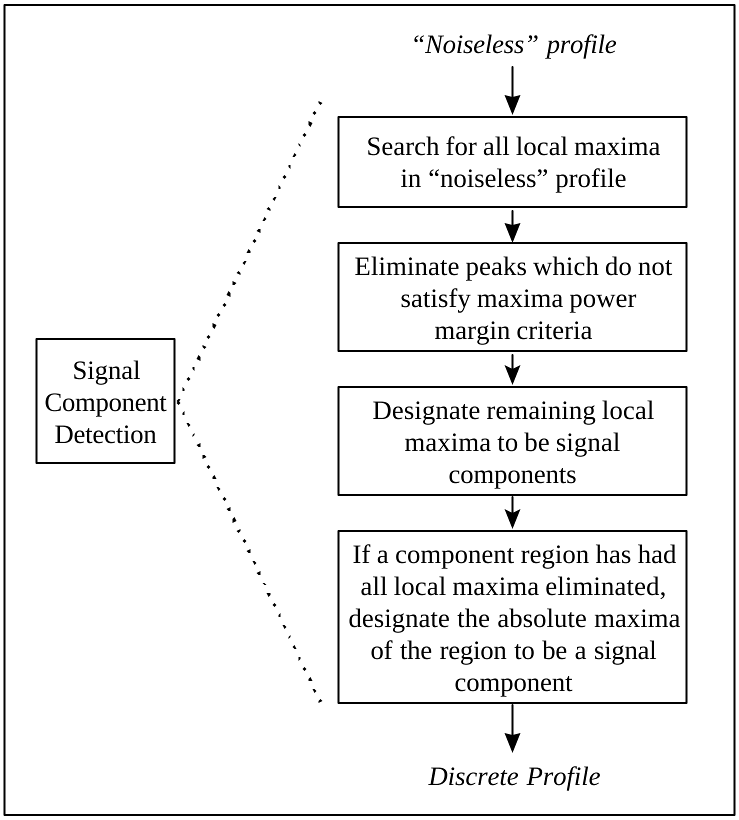

"Noiseless" profile
Search for all local maxima in "noiseless" profile
Eliminate peaks which do not satisfy maxima power margin criteria
Designate remaining local maxima to be signal components
If a component region has had all local maxima eliminated, designate the absolute maxima of the region to be a signal component
Discrete Profile
Signal Component Detection

Figure 5.3-5. The first step in signal component detection is a search for all local maxima within the component regions. A test is performed to remove from the signal component candidate list all local maxima which do not satisfy the maxima power margin criteria. If no maxima satisfy this criteria within a single region, then the absolute maximum within that particular region is considered to be the signal component.

Figure 5.3-7 illustrates the output of a signal component detection software implementation. The entire detection algorithm was implemented using MATLAB. The discrete profile contains only one sample point per signal component, as illustrated by the peak of each vertical line. The computed strengths of the components are -0.1 dB, -4.1 dB, -8.3 dB, -12.2 dB, -15.9 dB, and -19.8 dB. The computed delays of the components are 1.00 μs, 1.99 μs, 3.01 μs, 4.01 μs, 5.01 μs, and 6.01 μs. These results compare very well to the true delays and strengths defined for the channel, deviating in strength by at most 0.3 dB and in delay by at most 10 ns.

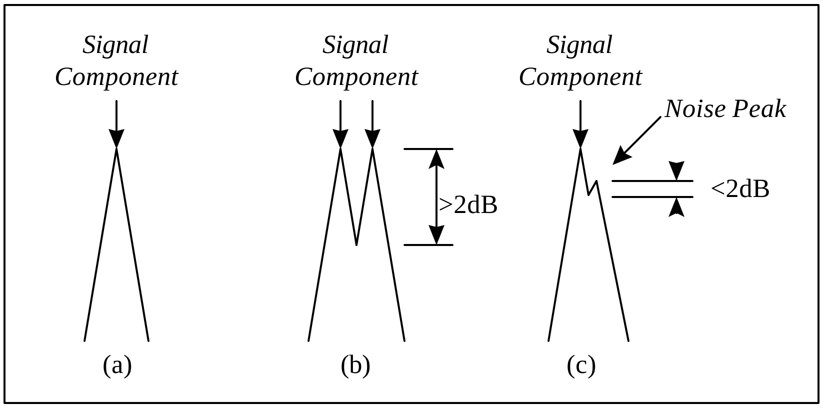


Figure 5.3-6. Single signal components within a component region as shown in case (a) are easy to identify by finding the single local maximum. Case (b) illustrates a case where two components must be detected within a single component region. In case (c), a noise spike affecting the trace within a component region must not be falsely identified as a signal component. The maxima power margin is used to reduce the possibility of false component detection.

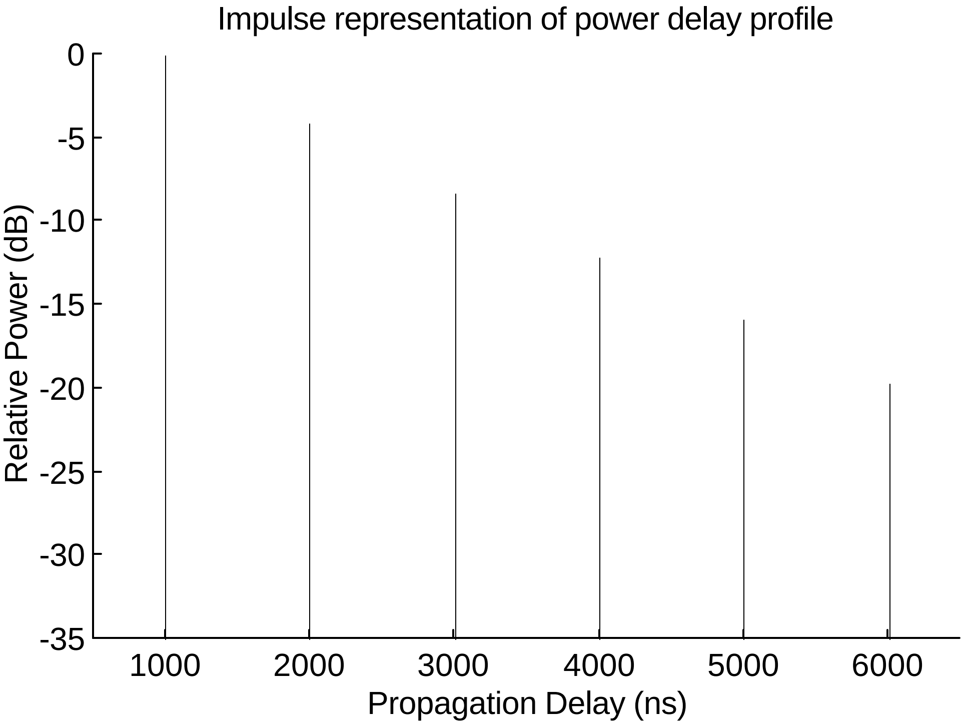


Figure 5.3-7. This discrete profile uses only one sample point per detected signal component. The delays and strengths of the components in this profile are shown to be almost exactly those of the actual channel.

The reliability of signal component detection processing is dependent upon the choice of observation level, minimum base width, and maxima power margin parameters. Detection of signal components relies heavily on correctly setting the observation level parameter. Consider the power delay profile shown in Figure 5.3-8. The observation level is defined to be -17 dB, approximately 10 dB above the noise floor of the profile. As shown by the "x" symbols which represent the endpoints of the component regions, when the observation level is set significantly above the noise floor, signal components can remain undetected during the observation level crossing detection process. The result is the profile shown in Figure 5.3-9, where received signal components were eliminated from the profile because they were identified to be noise. This results in low received power estimates and low RMS delay spread estimates.

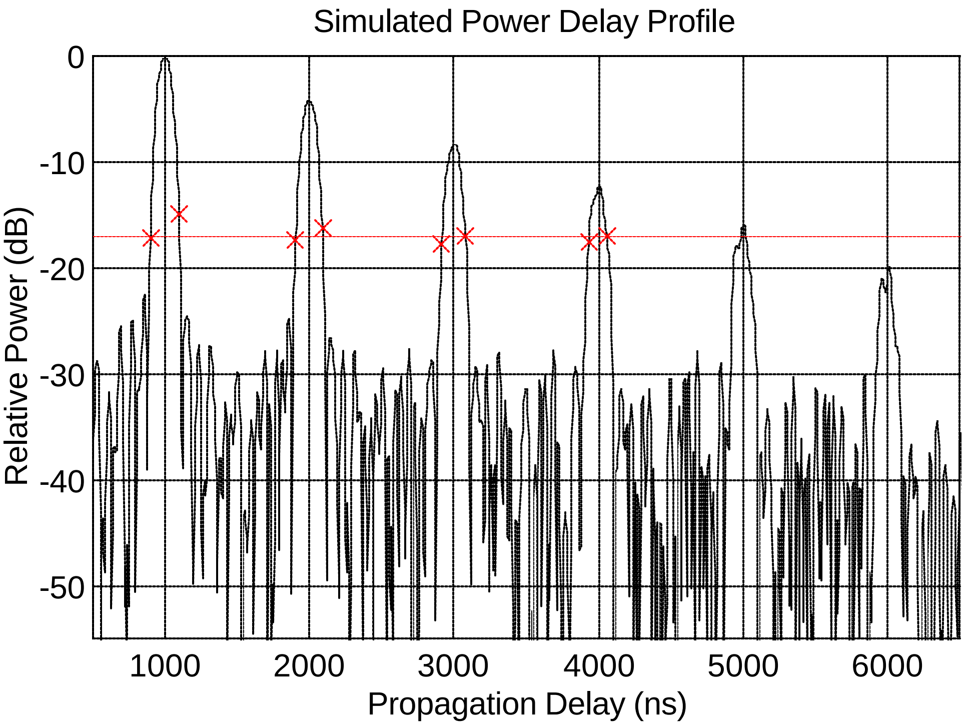


Figure 5.3-8. This observation level in this profile is set too high in order to demonstrate that low level signal components can be missed. The absence of "x" symbols on the low level components indicate that the observation level crossing and minimum base width processes caused the failure in detection.

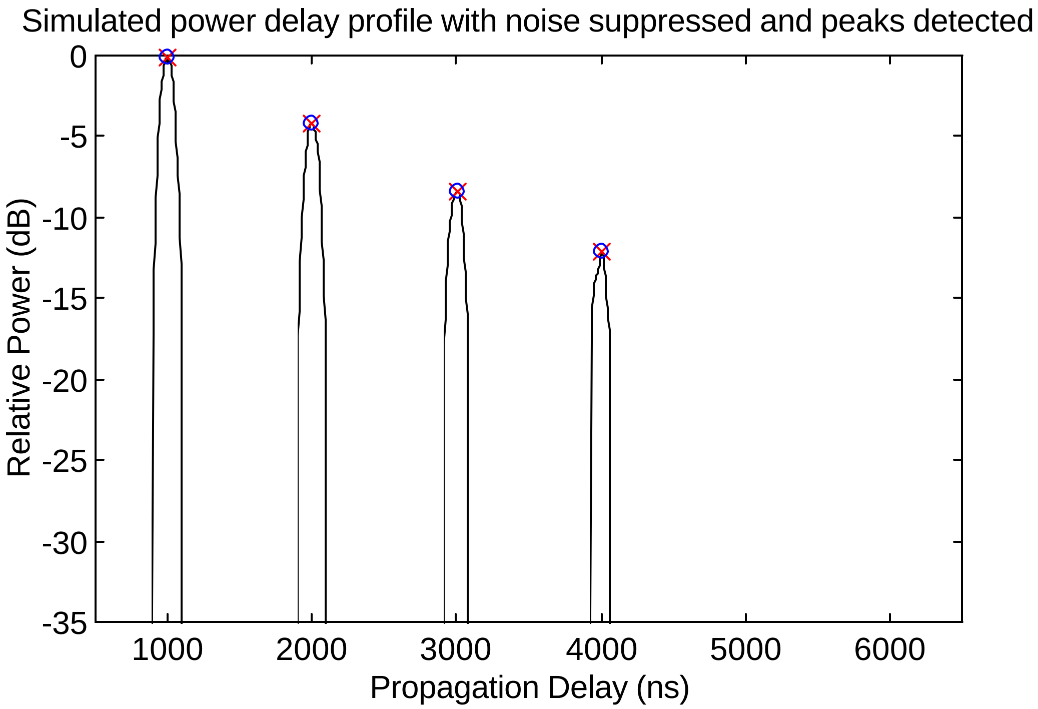


Figure 5.3-9. The two latest arriving signal components are falsely identified as noise and are eliminated from the profile because the observation level is set too high.

Figures 5.3-10 and 5.3-11 illustrate the effects of setting the observation level low compared to the noise floor level. Because large gaps occur between crossings of the observation level by the trace of the noise floor, the component regions around the signal components are expanded and component regions are defined where no components actually exist. The result is identification of several noise peaks as signal components as shown in Figure 5.3-11.

From the examples, it appears that the observation level should be set to a level that is 3 dB to 5 dB *below* the peaks of the noise floor. As shown next, accurate results for channel parameters can be obtained using observation levels which deviate several dB from this value.

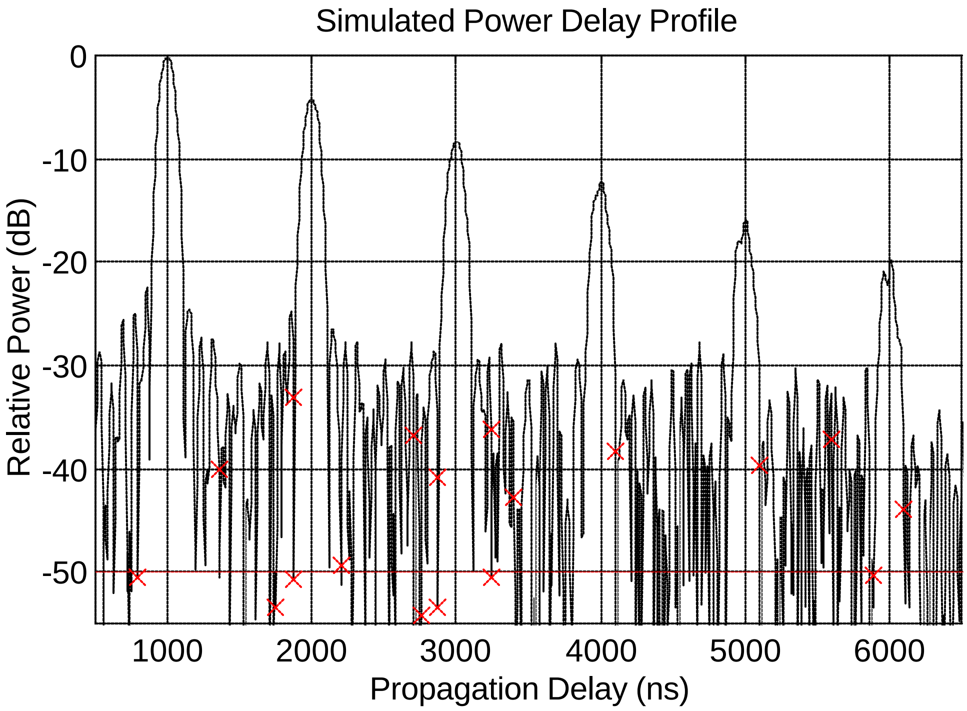


Figure 5.3-10. The observation level for this profile is defined to be -50 dB. This low level causes several noise peaks to be falsely identified as received signal components.

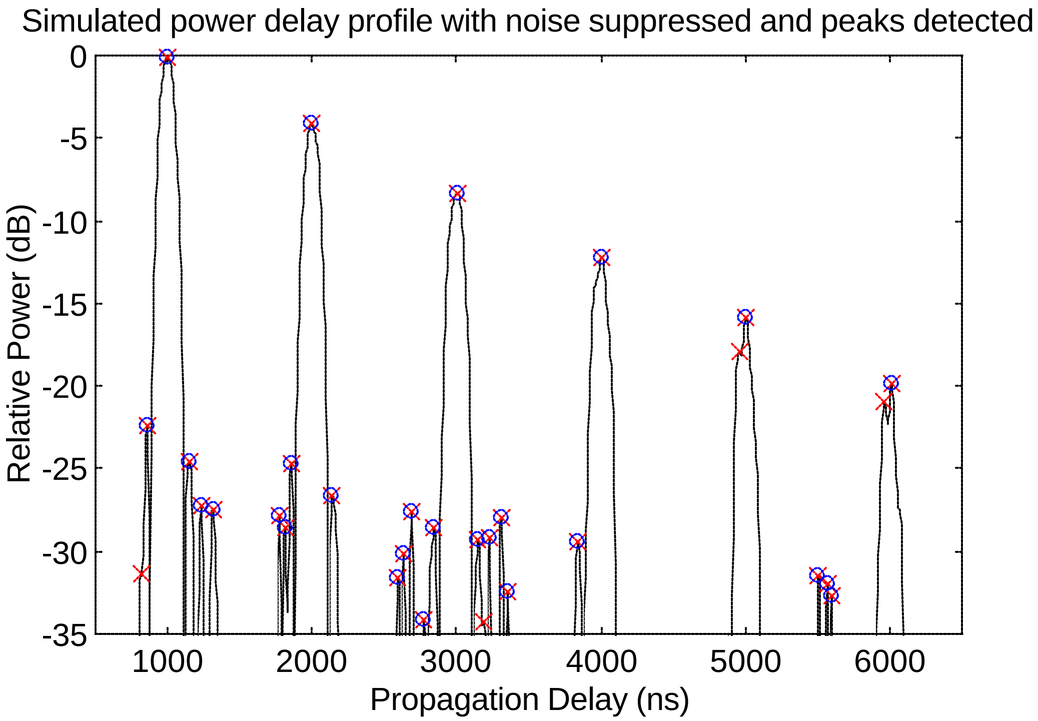


Figure 5.3-11. The low observation level causes noise peaks to be identified as signal components.

The minimum base width and maxima power margin parameters attempt to model the process of a human operator identifying individual signal components within a power delay profile. Thus, setting the parameters is somewhat of a subjective choice. The minimum base width can be set by measuring the base of a signal component which an operator considers to be the smallest discernable signal component. The maxima power margin can be set by using two closely spaced signal components which can be positively identified as separate components and measuring the power difference between the maxima and the minimum between the components. An advantage of modeling the human process rather than relying on component identification by the operator is that the model removes the inconsistency of the human operator and makes the process repeatable. Although the selection of minimum base width and maxima power margin parameters is subjective, holding these parameters and the observation level constant yields consistent and repeatable results. Also, the detection model can be used to automate wideband measurement data analysis rather than requiring an operator to identify all signal components in each of a large number of power delay profiles.

Figure 5.3-12 illustrates how calculated RMS delay spread changes as the observation level changes for the power delay profile shown in 5.3-3. The RMS delay spread shown by the solid line is calculated with equations (5.1-3), (5.1-6), and (5.1-7) using only discrete components. Also shown on the plot is the RMS delay spread versus noise threshold level calculated by the conventional technique using the power delay profile in Figure 5.3-3. Note that the actual RMS delay spread for the channel is 977 ns. The curve which was calculated using component detection has a relatively large region between -45 dB and -25 dB where the RMS delay spread is effectively independent of the observation level. That is, a 20 dB range exists within which the observation level could be set to obtain accurate RMS delay spread results. Using the conventional technique, however, there exists a region along the noise threshold axis of only approximately 5 dB where calculated RMS delay spread is within a few percent of the true value. Selecting a noise threshold level that is too low results in significant errors in RMS delay spread because of inclusion of noise samples in the calculation. Conservatively setting the noise threshold level to a value that is well above the noise floor results in low estimates of RMS delay spread. This plot essentially demonstrates that accurate RMS delay spread values can be obtained more easily using component detection because of the larger margin of error allowed when choosing the observation level versus the small margin when selecting a noise threshold level for processing.

The flat region of the solid line between -45 dB and -20 dB in Figure 5.3-12 indicates a calculated RMS delay spread of 984 ns. This is slightly higher than the true value of 977 ns, most likely caused by distortion of the low level signal components by self-noise. The distortion causes deviations in both component strength and delay from the true value, resulting in small errors in the calculated channel parameters. Below an observation level of -45 dB, an increasing number large noise peaks are detected as signal components causing the RMS delay spread estimate to rise. Above an observation level of -25 dB, low level signal components are detected as noise and eliminated from the channel parameter calculations, resulting in the step-down shape of the solid curve as individual signal components are lost.

The signal component detection algorithm was tested on actual measured profiles in addition to simulated profiles. Power delay profiles recorded during the rail yard propagation survey (presented in Chapter 3) were used as sample data. Figure 5.3-13 illustrates a measured power delay profile. Recall that a chip rate of 10 MHz and a center frequency of 915 MHz was used during the rail yard survey.

The trace in Figure 5.3-14 shows the data points of the profile which are detected as potential signal components. Within several of the component regions, several maxima are detected which clearly do not correspond to the correlation peaks of signal components. Because the maxima power margin is set to 2 dB, the insignificant maxima are not identified as signal components. Within regions of the profile where noise corrupts the signal components near their correlation peaks (e.g., the fourth-arriving signal component), a local maxima could not be positively identified as a signal component as evidenced by the lack of a circle around the peak. In these cases, the fallback assumption that the absolute maximum within the component region is the signal component peak performs well.

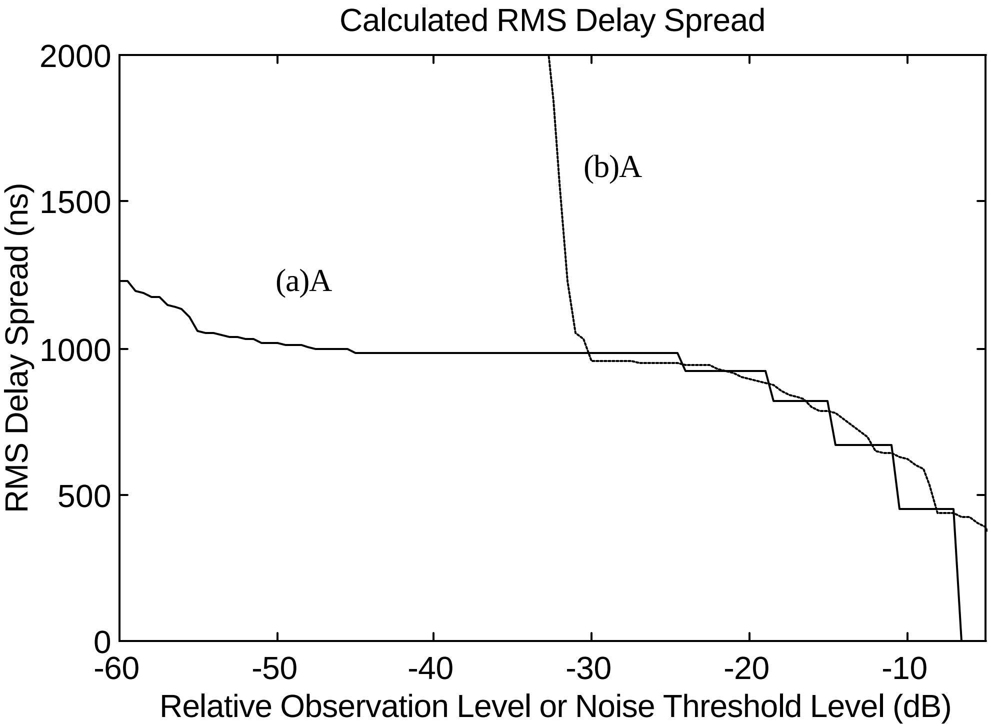


Figure 5.3-12. Accurate estimates of RMS delay spread are easier to obtain using component detection processing because of the large margin of error allowed when setting the observation level. Line (a) was calculated using discrete profiles, and line (b) was calculated using the conventional processing technique. True RMS delay spread for this channel is 977 ns.

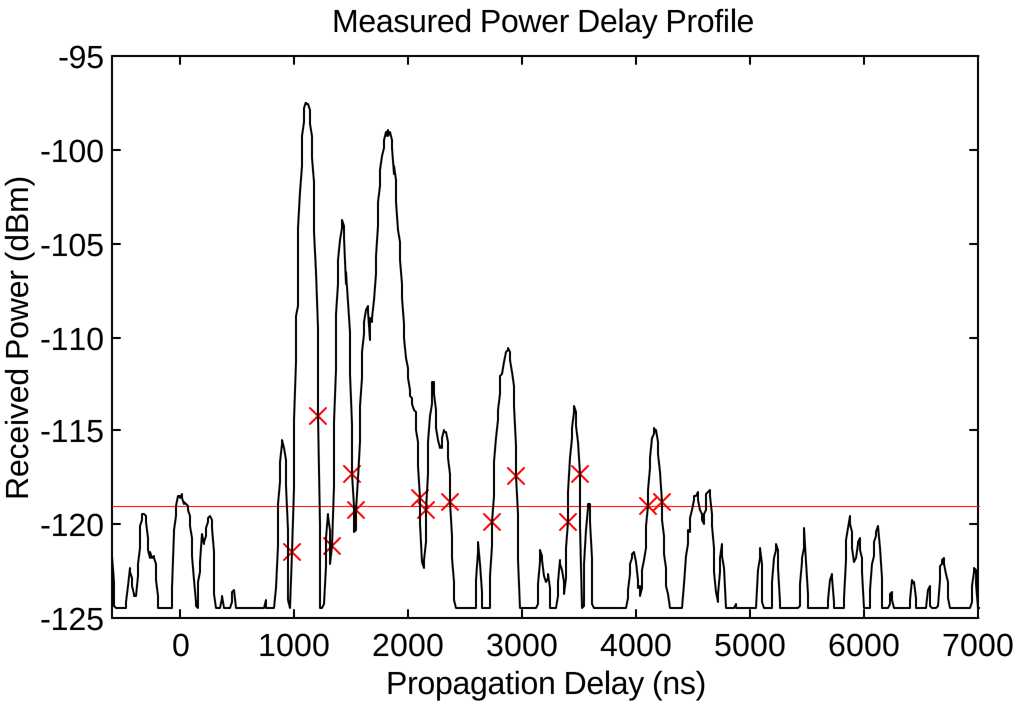


Figure 5.3-13. This power delay profile (3wb13.sco) was measured using the MPRG sliding correlator system during the train yard measurements. The horizontal line is the observation level and the “x” symbols represent the data points closest to the crossings.

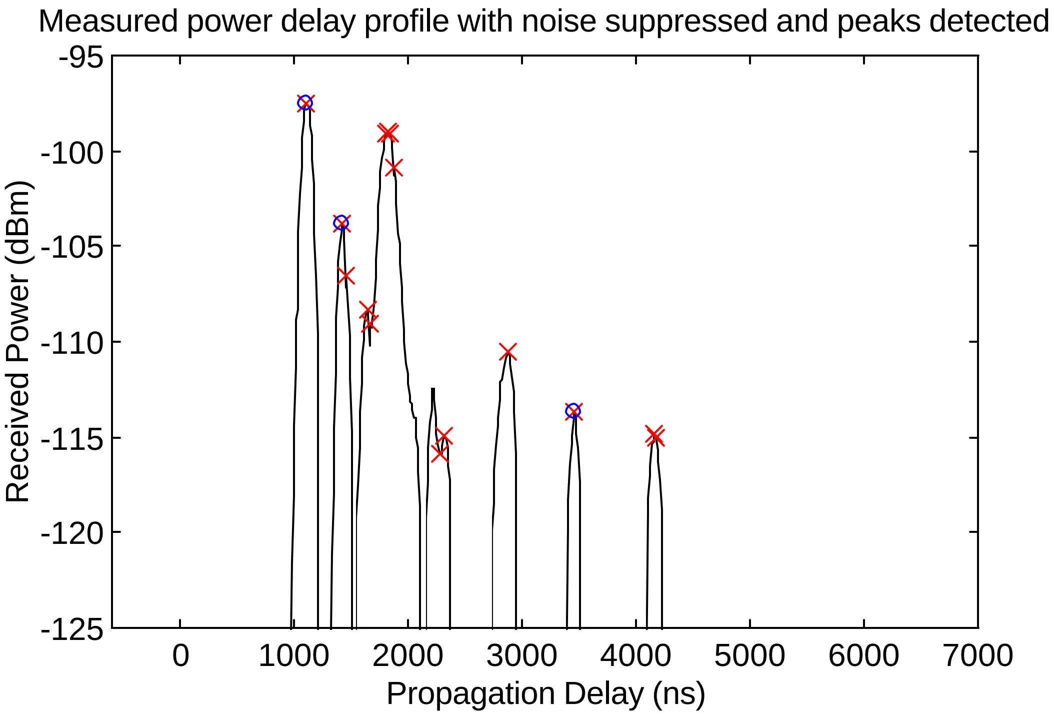


Figure 5.3-14. All of the major correlation peaks in the measured profile have been detected. A minimum base width of 100 ns was used.

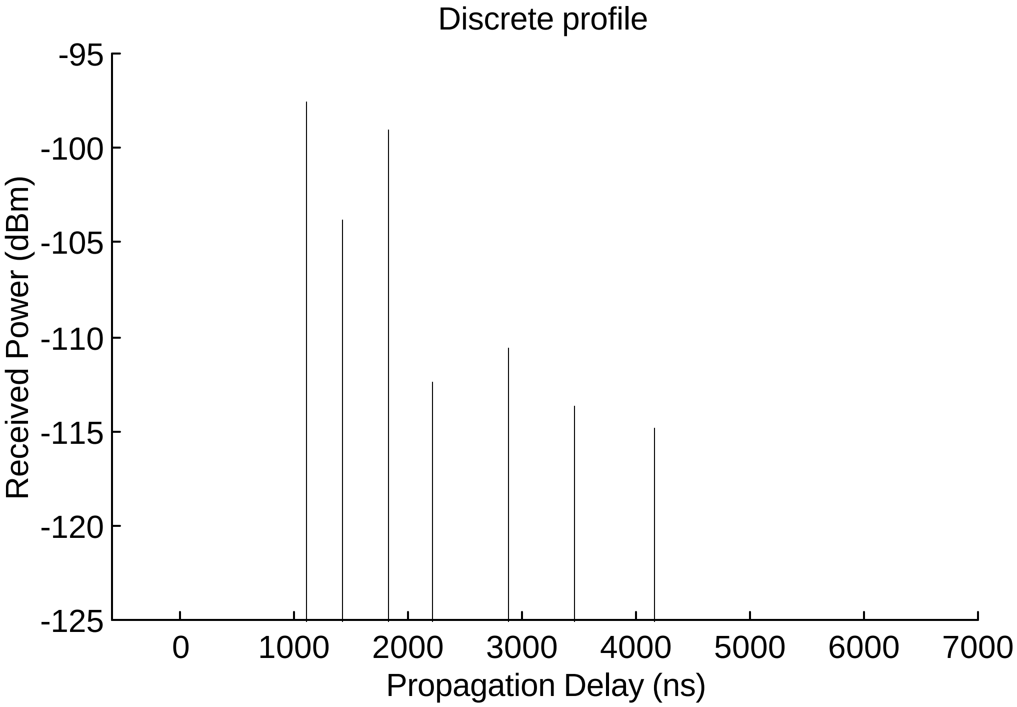


Figure 5.3-15. This discrete profile represents a radio channel measured during the rail yard propagation survey.

All detected components are shown in Figure 5.3-15. The strong components have unequivocally been identified by the algorithm. It is, of course, arguable whether or not all low level signal components have been captured. Identification of low level components in a power delay profile is, however, a subjective decision.

In essence, what has been developed in this section is a method for automating the process of obtaining estimates of impulses responses from power delay profiles. Component detection processing yields more consistent impulse response estimates versus identification of signal components by an operator because signal component selection criteria is held constant, rather then depending upon the consistency of an operator. Automation of the process also yields results more quickly and with less effort. In addition, estimates of RMS delay spread and received power derived from wideband channel profiles are more exact when calculated using discrete profiles rather than profiles that contain signal components of finite width.

5.4 Summary

Collection of received power and delay data with a sliding correlator measurement system must be complemented with accurate processing to produce meaningful results for researchers and wireless system designers. Results that statistically quantify propagation characteristics derived from a large number of power delay profiles are useful for system design applications. Results that specify exact delay and strength of multipath within specific power delay profiles are useful for validation of propagation models and other site-specific research applications. Presented in this section were the details of conventional processing to obtain multipath statistics and some of the problems that may be encountered. Also presented was a technique that can be used to automate the process of identifying individual signal components within a power delay profile.

The accuracy of conventional processing is dependent upon a user-defined noise threshold which removes the contribution of noise from the received power and RMS delays spread calculations. Plots of received power versus noise threshold level demonstrated that calculated received power is not highly sensitive to changes in noise threshold level. Plots of the RMS delay spread versus noise threshold level demonstrated that RMS delay spread is sensitive to changes in noise threshold level when the level falls within the vicinity of the noise floor or rises above components which significantly contribute to RMS delay spread. Accuracy of calculated RMS delay spread and received power is improved if signal strength is sampled only at the peak of each received signal component.

An algorithm was developed which detects the received signal components within a power delay profile. User-defined parameters of the algorithm include an observation level, a minimum base width, and a maxima power margin. The algorithm was tested on both simulated and measured power delay profiles. It was shown using simulated profiles that the sensitivity of RMS delay spread to changes in observation level is much less compared to the sensitivity of RMS delay spread to noise threshold level using the conventional processing technique. The discrete profiles produced by this technique are suited for comparison to the results of propagation prediction tools which predict individual multipath components at specific sites.

Chapter 6

Conclusion

6.1 Accomplishments and Contributions

Measurement, analysis, characterization, and simulation of wireless channels is a complex and necessary task for many communications researchers and engineers. Wideband measurement systems provide the propagation data to analyze and characterize path loss and time dispersion of radio channels. Results of measurements can be used to validate propagation prediction tools and to tune propagation models for wireless network design and optimization. The sliding correlator measurement system has proven to be an effective tool to perform these wideband measurements.

The research presented in the preceding chapters has covered practical and theoretical aspects of the sliding correlator measurement system. The primary accomplishments of this research include:

- Description of the hardware implementation of a sliding correlator system capable of 10 ns signal component resolution including and introduction to a hardware PN sequence synchronization technique

- A comprehensive review of the methods of characterizing path loss and time dispersion induced by radio channels.

- Path loss and time dispersion characterization of a train yard environment for the development of communication and tracking systems

- Wideband channel measurement processing software for statistical characterization of multipath in radio channels

- Development of a sliding correlator system simulator which produces realistic power delay profiles to assist in post processing algorithm development and measurement system specification and design

- Development of an algorithm for signal component detection within power delay profiles to produce accurate channel characterization results and discrete profile representations of radio channels

Chapter 2 introduced a basic channel model and presented an overview of wideband measurement systems which are used to resolve multipath signal components. The theory of sliding correlator measurement systems reviewed in Chapter 2 provided a foundation for the measurement survey, simulation development, and post processing algorithm presented in following chapters. Chapter 2 is a useful reference for future surveys which use the sliding correlator and for future measurement system development.

Chapter 3 described in depth a measurement survey of an environment which had not been well studied by propagation researchers. The results of the measurements characterize the train yard propagation environment for potential digital wireless applications. This chapter not only shows the results for one specific environment, but also demonstrates in general how to practically measure, process, and statistically summarize wideband measurement data.

The simulations developed in Chapter 4 provide a way to produce power delay profiles for user-defined delay and path loss characteristics of a radio channel. In addition to providing a means to validate post processing algorithms, the simulations demonstrate important system details which are helpful in developing an understanding of the signals and functions of the measurement system.

Chapter 5 reviewed the conventional wideband measurement processing technique and demonstrated problems which may surface when applying a noise threshold to reduce the effect of the noise floor on calculated channel parameters. The section which presents sensitivity of calculate channel parameters to noise threshold levels is especially of interest to those who have the task of compiling the results of a wideband measurement survey. A multipath signal component detection algorithm developed in Chapter 5 may be useful for validating propagation prediction tools and for transmitting power delay profile data across low-bandwidth data links. Multipath signal detection processing represents power delay profile in a more compact, more intuitive format.

6.2 Future Work

The sliding correlator will evolve in two ways to serve the needs of future wireless applications: improvement in signal component resolution and operation in higher frequency bands. Better resolution of the system is the more difficult challenge because of the high chip rates involved. The current implementation of the PN sequence generators (developed with emitter-coupled logic integrated circuits) should allow chip rates up to approximately 250 MHz using well designed circuit boards and component layout. Measurements in higher frequency bands requires redesign of the RF front end and modulation sections of the sliding correlator system. The design of a new measurement system will require (and develop) both digital and RF expertise.

In addition to the development and optimization of new personal communications networks, development of medical telemetry, law enforcement communications, video distribution, and wireless internet connectivity systems will drive and direct the course of new measurement systems and radio channel surveys. The increased commercial usage of direct sequence spread spectrum CDMA (code division multiple access) will require the development of systems similar to the sliding correlator to determine usable signal levels and interference characteristics. Spread spectrum PCS systems currently based on the 800 MHz and 1900 MHz CDMA standards (see [26], [27]) are already in need of measurement systems dedicated to tracking and measuring strength and delay of PN sequences offset in delay as observed at a receiver.

6.3 Closing

Much has changed with the world and the demands placed upon technology since the days of Nikola Tesla, whose intentions were to “stagger humanity” with the ability to communicate “with every city in the world … at the speed of 1500 to 2000 words per minute” [29]. However, the desire to build better, faster, and more robust wireless communication systems to serve the needs of business, safety, education, and entertainment has always been a constant goal. It is not the author’s intention to “stagger humanity” with this research; however, it is the author’s hope that the engineers and students involved in wireless communications research will find the information, developments, problems, and solutions addressed by this work useful for engineering design and as equally useful for education.

References

[1] Rohan P., *Introduction to Electromagnetic Wave Propagation*, Artec House, 1991.

[2] Rappaport T.S., *Wireless Communications Principles and Practice*, Prentice Hall, 1996.

[3] Seidel S.Y., *et. al.*, "Path Loss, Scattering, and Multipath Delay Statistics in Four European Cities for Digital Cellular and Microcellular Radiotelephone," *IEEE Trans. on Veh. Tech.*, Vol. 40, No.4, Nov. 1991.

[4] Rappaport T.S., *et. al.*, "900 MHz Multipath Propagation Measurements for U.S. Digital Cellular Radiotelephone," *IEEE Trans. on Veh. Tech.*, Vol. 39, No.2, May 1990.

[5] Cox D.C., "Delay Doppler Characteristics of Multipath Propagation at 910 MHz in Suburban Mobile Radio Environment," *IEEE Trans. on Ant. and Prop.*, Vol. AP-20, No. 5, Sept. 1972.

[6] Skolnik M.I., *Introduction to Radar Systems*, McGraw Hill, 1980.

[7] Devasirvatham D.M.J., "Time Delay Spread and Signal Level Measurements of 850 MHz Radio Waves in Building Environments," *IEEE Trans. on Ant. and Prop.*, Vol AP-34, No. 11, Nov. 1986.

[8] Golomb S., *Shift Register Sequences*, Aegean Press, 1982.

[9] Jeruchim M.C., Balaban P., Shanmugan K.S., *Simulation of Communication Systems*, 1994.

[10] Dixon R.C., *Spread Spectrum Systems*, Wiley, 1984.

[11] Parsons J.D, *Mobile Communication Systems*, Halsted Press, 1989.

[12] Stark H., Woods J.W., *Probability, Random Processes, and Estimation Theory for Engineers*, Prentice Hall, 1994.

[13] Athanasiadou G.E., Nix A.R., McGeehan J.P., "Indoor 3-D Ray Tracing Predictions and Their Comparison with High Resolution Wideband Measurements," *Proc IEEE VTC '96*, Vol. 1, May 1996.

[14] Rappaport T.S., "Characterization of UHF Mulitpath Radio Channels in Factory Buildings," *IEEE Trans. on Ant. and Prop*., Vol. 37, No. 8, Aug. 1989.

[15] Devasirvatham D.M.J., Murray R.R., Wolter D.R., "Time Delay Spread Measurements in a Wireless Local Loop Testbed," *Proc. IEEE VTC '95*, Vol. 1, July 1995.

[16] Kennedy, R.S., *Fading Dispersive Communication Channels*, Wiley, 1969.

[17] Chuang J. C-I, "The Effects of Time Delay Spread on Portable Radio Communications Channels with Digital Modulation," *IEEE J. Select Areas Comm*, Vol. SAC-5, No. 5, June 1987.

[18] Newhall W.G., Saldanha K., Rappaport T.S., "Propagation Time Delay Spread Measurements at 915 MHz in a Large Train Yard," *Proc IEEE VTC '96*, Vol. 2, May 1996.

[19] Newhall W.G., Saldanha K., Rappaport T.S., "Using RF Channel Sounding Measurements to Determine Delay Spread and Path Loss," *RF Design*, Vol. 19, No. 1, Jan. 1996.

[20] Burden R.L., Faires J.D., *Numerical Analysis*, 5th ed., PWS Publishing Company, 1993.

[21] Meinel H., et. Al, "A 40 GHz Railway Communication System," *IEEE J. Select Areas Comm.*, Vol. SAC-1, No. 4, Sept. 1983.

[22] *Tektronix 2782 Spectrum Analyzer Information Manual*, Tektronix, Inc.

[24] Talvitie J., Poutanen T., "Self-noise as Factor Limiting the Dynamic Range in Impulse Response Measurements Using Sliding Corraltion," *IEEE 3rd Int'l Symp. on Spread Spectrum Techniques and Applications*, 1994.

[26] *Personal Station – Base Station Compatibility Requirements for Dual Mode Wideband Spread Spectrum Cellular System*, Electronic Industry Association, Interim Standard IS-95-A.

[27] *Personal Station – Base Station Compatibility Requirements for 1.8-2.0 MHz Code Division Multiple Access (CDMA) Personal Communication Systems*, ANSI Standard J-STD-008.

[28] Benvenuto, N., “Distortion Analysis on Measuring the Impulse Response of a System Using a Crosscorrelation Method,” *AT&T Bell Laboratories Technical J.*, Vol. 63, No. 10, Dec. 1984.

[29] Fahie J.J., *A History of Wireless Telegraphy*, Arno Press, 1971.

[30] “Radio as a Medium of Communication,” *Encyclopedia Americana*, Grolier Inc., Vol. 23, pp. 147-150, 1997.

[31] Cantril H., Allport G.W., “The Psychology of Radio,” Harper & Brothers Publishers, 1935.

[32] “Wireless Growth to Increase 32%,” *Radio Communications Report,* Vol. 16, No. 32, August 1997.

[33] Wepman J.A, Hoffman J.R., Loew L.H., “Analysis of Impulse Response Measurements for PCS Channel Modelling Applications,” *IEEE Trans. Veh. Tech.,* Vol. 44, No. 3, August 1995.

[34] Couch L.W., *Digital and Analog Communication Systems*, 4th ed., Macmillan Publishing Company, 1993.

Appendix

Simulation and Processing Source Code

The following source code was executed using MATLAB® version 4.2.

1.1 Ideal Correlator Simulation

7-Stage PN Sequence Generator Simulated Power Delay Profile

```
% idealex1.m
% William G. Newhall
% Measurement system simulation using an ideal correlator and
% a baseband representation of the transmitter and receiver.
% Produces a simulated power delay profile plot.
%
%

clear;

% ============================================================
% User definable parameters

chipClock   = 100e6;                % TX chip clock in Hz
powerCalFactor    = 0;              % Power offset calibration in dB
chDelSec    = [200 300 600 800]*1e-9;     % Channel delay vector (sec)
chAttendB   = [5 12 10 13];         % Channel attenuation vector (dB)

% ============================================================
% Calculations

PN = pngenfcn( 7,3,7 );
TXPN = txgenfcn( PN, ones( 1,10 ) );

chipPeriod = 1 / chipClock;         % TX chip period (sec)
maxDelay = chipPeriod * length( PN );     % Max measurable delay (sec)
chDel = round( chDelSec * ...
      length( TXPN ) / maxDelay );  % Channel delay vector (# samples)
RXPN = chan( TXPN, chDel, chAttendB );    % Simulate received signal
[PNcorr, PNcorrdB] =  ...
      correlat( RXPN, TXPN ); % Compute correlation function
propDly = (0:maxDelay/(length(PNcorrdB)-1):maxDelay); % Delay axis
vector
plot( propDly*1e9,PNcorrdB+powerCalFactor, 'k' ); % plot PDP
```

```
title('Simulated Power Delay Profile')
xlabel('Propagation Delay (ns)')
ylabel('Relative Power (dB)')
grid on
```

11-Stage PN Sequence Generator Simulated Power Delay Profile

```
% idealex2.m
% William G. Newhall
% Measurement system simulation using an ideal correlator and
% a baseband representation of the transmitter and receiver.
% Produces a simulated power delay profile plot.
%
%

clear;

% ===========================================================
% User definable parameters

chipClock   = 100e6;                % TX chip clock in Hz
powerCalFactor    = 0;              % Power offset calibration in dB
chDelSec    = [100 150 500 525 600]*1e-9; % Channel delay vector (sec)
chAttendB   = [3 12 5 10 13];       % Channel attenuation vector (dB)

% ===========================================================
% Calculations

PN = pngenfcn( 11,9,11 );
TXPN = txgenfcn( PN, ones( 1,10 ) );

chipPeriod = 1 / chipClock;         % TX chip period (sec)
maxDelay = chipPeriod * length( PN );     % Max measurable delay (sec)
chDel = round( chDelSec * ...
      length( TXPN ) / maxDelay );  % Channel delay vector (# samples)
RXPN = chan( TXPN, chDel, chAttendB );    % Simulate received signal
[PNcorr, PNcorrdB] =  ...
      correlat( RXPN, TXPN ); % Compute correlation function
propDly = (0:maxDelay/(length(PNcorrdB)-1):maxDelay); % Delay axis
vector
plot( propDly*1e9,PNcorrdB+powerCalFactor, 'k' ); % plot PDP
title('Simulated Power Delay Profile')
xlabel('Propagation Delay (ns)')
ylabel('Relative Power (dB)')
grid on
axis ([0 1000 -70 0])
```

PN Generator Function (Support Function)

```
% PNgenFcn.m
% William G. Newhall
% Generates PN sequence and returns PN.
% PN = pngenfcn( shiftRegLen, tap1, tap2 )

function PN = pngenfcn( shiftRegLength, tap1, tap2 )
```

```
% ============================================================
% Calculations

PNlength = 2^shiftRegLength - 1;    % Length of sequence
PN = zeros( 1, PNlength );          % Initialize PN to 0's
shiftReg = ones( 1,shiftRegLength );      % Initialize shift reg to 1's

for i = 1:PNlength
  shiftIn = rem( shiftReg(tap1)+shiftReg(tap2), 2 );  % XOR of taps
  PN(i) = shiftReg( shiftRegLength );                 % Shift reg output
  shiftReg = [shiftIn shiftReg( 1:shiftRegLength-1 )];      % Shift
right, shift in XOR
end % end for PNlength

PN = PN * 2 - 1;  % convert 0->-1, 1->1
```

Transmitter Output Function (Support Function)

```
% TXgenfcn.m
% William G. Newhall
% Generates the transmitter PN sequence samples.
% TXPN = TXgenfcn( PN, TXchip );

function TXPN = TXgenfcn( PN, TXchip );

clear TXPN;                  % initialize TXPN
TXPN = [];
PNlength=length( PN );
for chipIndex = 1:PNlength          % form transmitted signal samples
  TXPN = [TXPN PN( chipIndex )*TXchip];   % concatenate TX chips
end % end for chipIndex
```

Radio Channel Function (Support Function)

```
% chan.m
% William G. Newhall
% Produces output of a multipath channel using chIn as
% an input.  Delays and attenuates according to chDel
% and chAttendB vectors.

function chOut = chan( chIn, chDel, chAttendB )

chAtten = 10.^( chAttendB/20 );           % Linear chan atten
chOut = zeros( 1, length( chIn ) ); % Initialize chan output
for compIndex = 1:length( chDel )   % Calculate chan output
  chOut = chOut + seqDelay( chIn, chDel( compIndex ) ) ...
           / chAtten( compIndex );
end % end for compIndex
```

PN Sequence Delay Function (Support Function)

```
% SeqDelay.m
% William G. Newhall
% Output is the input sequence S delayed by n samples

function delS = seqDelay( S, n )

lenS = length( S );
if( n == 0)             % delay = 0 case
  delS = S;
elseif( (n>0) & (n<lenS) )    % 0 < delay < sequence length
  delS = [S( lenS-n+1:lenS ) S( 1:lenS-n )];
else
  error( 'n must be greater than zero and less than sequence length' );
end
```

Correlation Function (Support Function)

```
% Correlat.m
% William G. Newhall
% Produces the correlation function for PN and rxPN.  Output is
% PNcorr (correlation function) and PNcorrdB (correlation function
% expressed in dB normalized to 0 dB)

function [PNcorr, PNcorrdB] = correlat( PN, rxPN )

slidPN = rxPN;                      % The sliding sequence
PNcorr( 1 ) = PN * slidPN';         % first point in corr fcn
lastIndex = length( PN );           % number of chips in PN sequence
printflag = 50;
for i = 2:lastIndex
  if( i==printflag )
    printflag = printflag + 50;
    disp( ['i = ' num2str( i ) ' out of ' num2str( lastIndex )] );
  end
  slidPN = [slidPN( lastIndex ) slidPN( 1:lastIndex-1 )]; % slide
sequence
  PNcorr( i ) = PN * slidPN';       % multiply and integrate
end % end for i

PNcorr = PNcorr / lastIndex;        % normalize to 1
PNcorrdB = 10 * log10( PNcorr.^2 ); % express in dB
```

1.2 Probing Signal Simulation

Ideal Correlator Probing Signal

```
% idprbex1.m
% Run me to produce a sample ideal correlator probing signal.
% William G. Newhall
%

clear;

% ===========================================================
% User definable parameters

chipClock   = 100e6;    % TX chip clock in Hz
PN = pngenfcn( 7,3,7 );       % PN sequence definition


% ===========================================================
% Calculations

[ProbSig, ProbSigTime] = idprbgen( PN, chipClock );

% Show negative half of probing signal as well as positive half
ProbSig = [ fliplr( ProbSig( 2:floor( length( ProbSig ) / 2 ) ) ),
ProbSig( 1:floor( length( ProbSig ) / 2 ) ) ];
ProbSigTime = [ -fliplr( ProbSigTime( 2:floor( length( ProbSigTime ) / 2
) ) ), ProbSigTime( 1:floor( length( ProbSigTime ) / 2 ) ) ];


plot( ProbSigTime*1e9, ProbSig, 'k-' ); % plot PDP
title('Probing Signal for Ideal Correlator Receiver')
xlabel('Propagation Delay (ns)')
ylabel('Relative Voltage')
axis( [-40, 40, -0.1, 1] );
grid off
```

Simulated Power Delay Profile Based on Ideal Probing Signal

```
% idprbex1.m
% Run me to produce a sample ideal correlator probing signal.
% William G. Newhall
%

clear;

% ==========c=================================================
% User definable parameters

chipClock   = 100e6;    % TX chip clock in Hz
PN = pngenfcn( 7,3,7 );       % PN sequence definition
chDelSec    = [200 300 600 800]*1e-9;      % Channel delay vector (sec)
chAttendB   = [5 12 10 13];         % Channel attenuation vector (dB)

% ===========================================================
% Calculations

[ProbSig, ProbSigTime] = idprbgen( PN, chipClock );

chipPeriod = 1 / chipClock;         % TX chip period (sec)
maxDelay = chipPeriod * length( PN );     % Max measurable delay (sec)
chDel = round( chDelSec * ...
      length( ProbSig ) / maxDelay );     % Channel delay vector (#
samples)
```

```
PDP = chan( ProbSig, chDel, chAttendB );% Transmit probing sig thru
channel
%ProbSigTime = ProbSigTime ...            % Show positive time,
%     + max( ProbSigTime );         %   not PN cycle



plot( ProbSigTime*1e9, 10*log10( PDP.^2 ), 'k-' ); % plot PDP
title('Simulated Power Delay Profile (Ideal Probing Signal)')
xlabel('Propagation Delay (ns)')
ylabel('Relative Power (dB)')
grid on
```

Generates 10 KHz (3dB baseband BW) RC, 9-stage [4,9] Probing Signal Example

```
% Psamprc.m
% Probing signal sample used to show full period of probing signal:
% PN: 9 stage [4,9], 10MHz chip rate, K=1000, RC filter (Baseband BW3dB
= 10KHz)
%

% Uses output generated by probsig7.m and stored in probsig7.mat.
load probsig7

figure
plot (t*1000, 10*log10(abs(y).^2),'k');
xlabel ('Time (us)');
ylabel ('Relative Power (dB)');
title ('Simulated Probing Signal (RC Filter)');
grid on;
axis ([min(t*1000) max(t*1e3) -50 0])
```

Produces Probing Signal 10 KHz (3dB baseband BW) RC, 9-stage [4,9]

```
% Probsig7.m
% Desired signal output.

% User input -----------------------------------

Rc1=10e6;         % Chip Rate
K=1000;                 % Slide factor

s = pngen (9,4,9);      % PN sequence definition (shiftRegLen, tap1,
tap2)

nrange=[-300,300];
mrange=[-300,300];

trange=[-0.0005, 0.0520];
numt=1024;
```

```
% End user input ----------------------------------------------

% Calculated parameters

L=length (s);

Rc2=Rc1*(1-1/K);  % Chip rate

Tc1=1/Rc1;        % Chip period
Tc2=1/Rc2;        % Chip period

T1=L*Tc1;         % Sequence period
T2=L*Tc2;         % Sequence period

f1=1/T1;          % Sequence repitition rate
f2=1/T2;          % Sequence repitition rate

tstep=(trange(2)-trange(1))/numt;
t=trange(1):tstep:trange(2);

deltaf = 10000;         % 3-dB cutoff for filter
RC = 1/(2*pi*deltaf);   % filter time constant


% Desired signal
tic;

nmarg=zeros( nrange(2)-nrange(1)+1 , mrange(2)-mrange(1)+1);

n=nrange(1):nrange(2);
m=(nrange(1):mrange(2))';

for mindex=mrange(1):mrange(2);
  nmarg (mindex-mrange(1)+1,:) = n*f1+mindex*f2;
end

disp('past for')

c1vec=zeros(1,nrange(2)-nrange(1)+1);
c2vec=zeros(mrange(2)-mrange(1)+1,1);

for nval=nrange(1):nrange(2)
  c1vec(1,nval-nrange(1)+1)=C1(nval*f1,s,Tc1);
end % nindex

for mval=mrange(1):mrange(2)
  c2vec(mval-mrange(1)+1,1)=C2(mval*f2,s,Tc2);
end % mindex

yconst=f1*f2 * c2vec*c1vec ./ (j*2*pi*nmarg*RC + 1);

toc
tic
y=zeros(1,length(t));
```

```
for tindex=1:length(t)
  disp (['t=' num2str(tindex)]);
  y(tindex) = sum( sum( yconst .* exp(j*2*pi*nmarg*t(tindex)) ) );
end % tindex

toc

figure
plot (t*1000, 10*log10(abs(y).^2));
xlabel ('Time (ms)');
ylabel ('Relative Power (dB)');
title ('Power delay profile');
grid on;
```

PN Sequence Generator for Probing Signal Examples

```
% PNgen.m
% Programmer:  W.G. Newhall
% Generates PN sequence and stores in variable PN
function PN= pngen (shiftRegLength, tap1, tap2);

% Generate PN sequence

% Calculation of PN
PNlength = 2^shiftRegLength - 1;          % define length of PN sequence
PN = zeros (1,PNlength);                  % initialize PN to all zeros
shiftReg = ones(1,shiftRegLength);        % initialize shift reg to ones

for i = 1:PNlength
  shiftIn = rem (shiftReg(tap1)+shiftReg(tap2) , 2);  % XOR of taps
  PN (i) = shiftReg (shiftRegLength);                 % ouput val of
shift reg
  shiftReg = [shiftIn shiftReg (1:shiftRegLength-1)]; % shift right and
set
end  % end PNlength                             %   first val to XOR val

PN = PN*2 - 1;                            % convert: 1->1, 0->-1

% Debug check:
%numPNones = sum ((PN+1)/2)               % checks number of 1's
%numPNnegOnes = sum ((-PN+1)/2)                 % checks number of -1's
```

Produces 10 KHz Gaussian (6dB baseband BW), 9-stage [4,9] Probing Sig Example

```
% Psampg.m
% Probing signal sample used to show full period of probing signal:
% PN: 9 stage [4,9], 10MHz chip rate, K=1000, Gaussian filter (Baseband
BW = 10KHz)
%

% Uses output generated py probsig5.m and stored in probsig5.mat.
load probsig5
```

```
figure
plot (t*1000, 10*log10(abs(y).^2),'k');
xlabel ('Time (us)');
ylabel ('Relative Power (dB)');
title ('Simulated Probing Signal (Gaussian Filter)');
grid on;
axis ([min(t*1000) max(t*1e3) -50 0])
```

Generates 10 KHz Gaussian (6dB baseband BW), 9-stage [4,9] Probing Signal

```
% Probsig5.m
% Desired signal output.

% User input -----------------------------------

Rc1=10e6;         % Chip Rate
K=1000;                 % Slide factor

s = pngen (9,4,9);      % PN sequence definition (shiftRegLen, tap1,
tap2)

nrange=[-300,300];
mrange=[-300,300];

trange=[-0.0005, 0.0520];
numt=1024;

% End user input ----------------------------------------------

% Calculated parameters

L=length (s);

Rc2=Rc1*(1-1/K);  % Chip rate

Tc1=1/Rc1;        % Chip period
Tc2=1/Rc2;        % Chip period

T1=L*Tc1;         % Sequence period
T2=L*Tc2;         % Sequence period

f1=1/T1;          % Sequence repitition rate
f2=1/T2;          % Sequence repitition rate

tstep=(trange(2)-trange(1))/numt;
t=trange(1):tstep:trange(2);

deltaf = Rc1/K;         % 3-dB cutoff for filter
%RC = 1/(2*pi*deltaf);  % filter time constant

% Gaussian Filter Parameters
A = 10^(6/10);          % Cutoff attenuation at cutoff freq fo
fo = deltaf;                  % AdB cutoff of Gaussian filter
a = sqrt(log(sqrt(A)))/fo;    % Gaussian filter parameter alpha
```

```
% Desired signal
tic;

nmarg=zeros( nrange(2)-nrange(1)+1 , mrange(2)-mrange(1)+1);

n=nrange(1):nrange(2);
m=(nrange(1):mrange(2))';

for mindex=mrange(1):mrange(2);
  nmarg (mindex-mrange(1)+1,:) = n*f1+mindex*f2;
end

disp('past for')

c1vec=zeros(1,nrange(2)-nrange(1)+1);
c2vec=zeros(mrange(2)-mrange(1)+1,1);

for nval=nrange(1):nrange(2)
  c1vec(1,nval-nrange(1)+1)=C1(nval*f1,s,Tc1);
end % nindex

for mval=mrange(1):mrange(2)
  c2vec(mval-mrange(1)+1,1)=C2(mval*f2,s,Tc2);
end % mindex

yconst=f1*f2 * c2vec*c1vec .* exp (-a^2 * nmarg.^2); % uses Gaussian
filt
toc
tic
y=zeros(1,length(t));

for tindex=1:length(t)
  disp (['t=' num2str(tindex)]);
  y(tindex) = sum( sum( yconst .* exp(j*2*pi*nmarg*t(tindex)) ) );
end % tindex

toc

figure
plot (t*1000, 10*log10(abs(y).^2));
xlabel ('Time (ms)');
ylabel ('Relative Power (dB)');
title ('Power delay profile');
grid on;
```

Produces sample power delay profiles, 9-stage tapped at [4,9]

```
% noisex1.m
% William G. Newhall
% Plots simulated power delay profile using the self-noise simulation
model.
% Simulation uses a 9-stage shift register [4,9] an a 10 MHz chip rate.
% Probsig14.mat defines the probing signal in varibles y and t.  Run
Probsig14.m
% to produce this file and place in same directory as noisex*.m.  A
Gaussian
% filter is used for this simulation.

clear;

% ==========================================================
% User definable parameters

powerCalFactor    = 0;              % Power offset calibration in dB
chDelSec    = [700 1000 1400 2000 2300 3700]*1e-9;    % Channel delay
vector (sec)
chAttendB   = [0 10 2 15 10 20];          % Channel attenuation vector
(dB)
K = 1000;                     % slide factor used for probing signal

% ==========================================================
% Calculations

% load probing signal
load probsg14;
t = t/K;    % account for slide factor

samplePeriod = ( max(t)-min(t) ) / length( y );
chDel = round( chDelSec / samplePeriod );
PDP = chan( y, chDel, chAttendB );
PDPdB = 10*log10( abs(PDP).^2 );
PDPdB = PDPdB - max(PDPdB);         % normalize to peak component
plot( t*1e9, PDPdB, 'k-');
axis( [0,5000,-40,0] )

title( 'Simulated Power Delay Profile (Gaussian Filter)' )
xlabel( 'Propagation Delay (ns)' )
ylabel( 'Relative Power (dB)' )
grid on
```

Produces sample power delay profiles, 9-stage tapped at [4,9]

```
% noisex2.m
% William G. Newhall
% Plots simulated power delay profile using the self-noise simulation
model.
% Simulation uses a 9-stage shift register [4,9] an a 10 MHz chip rate.
% Probsig14.mat defines the probing signal in varibles y and t.  Run
Probsig14.m
% to produce this file and place in same directory as noisex*.m.  A
Gaussian
% filter is used for this simulation.

clear;

% ==========================================================
% User definable parameters

powerCalFactor    = 0;              % Power offset calibration in dB
chDelSec    = [5200 5700 6500 6750 8100 8500 9000 9500]*1e-9;     %
Channel delay vector (sec)
chAttendB   = [10 0 20 5 10 13 20 15];          % Channel attenuation
vector (dB)
K = 1000;                     % slide factor used for probing signal

% ===========================================================
% Calculations

% load probing signal
load probsg14;
t = t/K;    % account for slide factor

samplePeriod = ( max(t)-min(t) ) / length( y );
chDel = round( chDelSec / samplePeriod );
PDP = chan( y, chDel, chAttendB );
PDPdB = 10*log10( abs(PDP).^2 );
PDPdB = PDPdB - max(PDPdB);         % normalize to peak component
plot( t*1e9, PDPdB, 'k-');
axis( [5000,10000,-40,0] )

title( 'Simulated Power Delay Profile (Gaussian Filter)' )
xlabel( 'Propagation Delay (ns)' )
ylabel( 'Relative Power (dB)' )
grid on
```

Produces Probing Signals for NoiseEx1 and NoiseEx2

```
% Desired signal output.

% User input -----------------------------------

Rc1=10e6;         % Chip Rate
K=1000;                 % Slide factor

s = pngen (9,4,9);      % PN sequence definition (shiftRegLen, tap1,
tap2)

nrange=[-300,300];
mrange=[-300,300];

trange=[0, 0.0511];
numt=10219;

% End user input ----------------------------------------------

% Calculated parameters

L=length (s);

Rc2=Rc1*(1-1/K);  % Chip rate
```

```
Tc1=1/Rc1;        % Chip period
Tc2=1/Rc2;        % Chip period

T1=L*Tc1;         % Sequence period
T2=L*Tc2;         % Sequence period

f1=1/T1;          % Sequence repitition rate
f2=1/T2;          % Sequence repitition rate

tstep=(trange(2)-trange(1))/numt;
t=trange(1):tstep:trange(2);

deltaf = Rc1/K;         % 3-dB cutoff for filter
%RC = 1/(2*pi*deltaf);  % filter time constant

% Gaussian Filter Parameters
A = 10^(6/10);          % Cutoff attenuation at cutoff freq fo
fo = deltaf;                  % AdB cutoff of Gaussian filter
a = sqrt(log(sqrt(A)))/fo;    % Gaussian filter parameter alpha
% Hey Mike, you owe me a floor for a year.

% Desired signal
tic;

nmarg=zeros( nrange(2)-nrange(1)+1 , mrange(2)-mrange(1)+1);

n=nrange(1):nrange(2);
m=(nrange(1):mrange(2))';

for mindex=mrange(1):mrange(2);
  nmarg (mindex-mrange(1)+1,:) = n*f1+mindex*f2;
end

disp('past for')

c1vec=zeros(1,nrange(2)-nrange(1)+1);
c2vec=zeros(mrange(2)-mrange(1)+1,1);

for nval=nrange(1):nrange(2)
  c1vec(1,nval-nrange(1)+1)=C1(nval*f1,s,Tc1);
end % nindex

for mval=mrange(1):mrange(2)
  c2vec(mval-mrange(1)+1,1)=C2(mval*f2,s,Tc2);
end % mindex

yconst=f1*f2 * c2vec*c1vec .* exp (-a^2 * nmarg.^2); % uses Gaussian
filt
toc
tic
y=zeros(1,length(t));

for tindex=1:length(t)
```

```
  disp (['t=' num2str(tindex)]);
  y(tindex) = sum( sum( yconst .* exp(j*2*pi*nmarg*t(tindex)) ) );
end % tindex

toc

figure
plot (t*1000, 10*log10(abs(y).^2));
xlabel ('Time (ms)');
ylabel ('Relative Power (dB)');
title ('Power delay profile');
grid on;

save probsg14 y t
```

Spectrum Analyzer Filter/Gaussian Filter Comparison

```
% Plots filter characteristic of spectrum analyzer
% filter (ResBW)

resBW = 10000;   % resBW in Hz
[f,H] = sachar (resBW); % arg specifies resolution bandwidth

fkHz=f/1000;  % change Hz -> kHz
figure;
axis ([min(fkHz) max(fkHz) -50 max(H)]);

Hlin= 10.^(H/20);
Hlin= Hlin/max(Hlin);

plot (fkHz, Hlin, 'kx');

fo=10000;
aparam=sqrt(0.5*log(2))/fo;
[fgauss,Hgauss] = gfilt (5000,6,f);
hold on;
fgausskHz = fgauss/1000;
plot (fgausskHz,Hgauss,'k-');

title (['Spectrum analyzer filter characteristic (ResBW = '
num2str(resBW/1000) ' kHz)']);
xlabel ('Frequency relative to center frequency (kHz)');
ylabel ('Normalized Magnitude (V)');

legend ('Measured', 'Gaussian (10kHz)');

grid on;

hold off

figure

plot (fkHz, 10*log10(Hlin.^2), 'x');
hold on
```

```
plot (fgausskHz,10*log10(Hgauss.^2));

title (['Spectrum analyzer filter characteristic (ResBW = '
num2str(resBW/1000) ' kHz)']);
xlabel ('Frequency relative to center frequency (kHz)');
ylabel ('Normalized Magnitude');

legend ('Measured', 'Gaussian (10kHz)');
hold off
```

Measured Spectrum Analyzer Characteristic

```
% Returns measured spectrum analyzer filter data

function [f,H]=sachar (resBW)

if (resBW == 10000)
  fpos=1e6*[0 .001 .002 .003 .004 .005 .0055 .006 .0065 .007 .0075 .008
.0085 .009 .0095 .0100 .0105 .0110 .0115 .0120 .0125 .0130 .0135 .0140
.0150 .0155];
  Hpos = [-12.2 -12.4 -13.0 -14.3 -16.5 -19.4 -20.6 -22.1 -23.6 -25.1 -
26.7 -28.4 -29.9 -31.5 -33.0 -34.5 -36.1 -37.5 -39.0 -40.5 -42.0 -43.6 -
44.8 -46.5 -49.8 -50.6];

  fneg=-1e6*(1-[.999 .998 .997 .996 .995 .994 .9935 .993 .9925 .992
.9915 .991 .9905 .9900 .9895 .9890 .9885 .9880 .9875 .9870 .9865 .9860
.9855 .9850 .9840 .9830]);
  Hneg = [-12.3 -12.7 -13.5 -14.8 -16.9 -20.0 -21.5 -23.1 -24.7 -26.4 -
28.1 -29.8 -31.3 -33.0 -34.6 -36.2 -37.8 -39.5 -41.0 -42.7 -44.0 -45.8 -
46.9 -48.6 -51.8 -54.7];

  f= [fliplr(fneg) fpos];
  H= [fliplr(Hneg) Hpos];

elseif (resBW == 30000)
  fpos=1e6*[0 .003 .005 .007 .010 .013 .015 .017 .020 .023 .025 .027
.030 .032 .035 .037 .040 .042 .045 .047 .050 .052 .055];
  Hpos = [-11.2 -11.6 -12.1 -12.9 -14.5 -16.4 -17.9 -19.6 -22.1 -24.6 -
26.4 -28.3 -31.0 -32.9 -35.5 -37.2 -40.0 -41.4 -44.0 -45.8 -48.2 -49.8 -
52.1];

  fneg=-1e6*(1-[.995 .992 .990 .987 .985 .982 .980 .977 .975 .972 .970
.967 .965 .962 .960 .957 .955 .952 .950 .947 .945]);
  Hneg = [-11.9 -13.0 -14.0 -15.9 -17.3 -19.8 -21.5 -23.9 -25.7 -28.6 -
30.4 -33.1 -34.8 -37.5 -39.2 -41.8 -43.6 -46.2 -47.6 -49.8 -51.7];

  f= [fliplr(fneg) fpos];
  H= [fliplr(Hneg) Hpos];

else
  error (['Filter data for ' num2str (resBW) ' not availible.']);

end
```

1.3 Processing Source Code

Calculated Received Power Versus Noise Threshold Level

```
% Ndemo2.m
% William G. Newhall
% Demonstration of noise threshold problem.
% Plots calculated received power versus noise threshold level.
% Uses pdpgen, which in turn uses probsig14.mat to produce the power
delay
% profile.

clear;

% ==================================================
% User-defined parameters
NoiseThreshdBm = -90:0.1:-80; % noise threshold range

tau1 = 1e-6;                  % delay 1 for power delay profile (sec)
P1dBm = -80;                  % strength 1 for power delay profile (dBm)

tauCal1 = 1e-6;               % delay for calibration profile (sec)
Pcal1dBm = -80;               % strength for calibration profile (dBm)

% ==================================================

% Calculate calibration parameters
[PDPcaldBm, tauCal] = pdpgen( tauCal1, Pcal1dBm );
NoiseThreshCaldBm = max( PDPcaldBm( 0.9*length( PDPcaldBm ) : length(
PDPcaldBm ) ) );

% simulate power delay profile
[PDPdBm, tau] = pdpgen( [tau1], [P1dBm], 1); % generate PDP
%axis([0,6000,-105,-80]);

% Calculate total profile power as noise threshold increases
for p1Index = 1:length( NoiseThreshdBm )
  disp( ['Calculating ' num2str( p1Index ) ' out of ' num2str( length(
NoiseThreshdBm ) ) ] );
  PrecdBm( p1Index ) = recpwr( tau, PDPdBm, NoiseThreshdBm( p1Index ),
tauCal, PDPcaldBm, NoiseThreshCaldBm );
end % p1Index

%StoNthresh = P1dBm - NoiseThreshdBm;
%PerrordBm = PrecdBm - P1dBm;
%figure
%plot( StoNthresh, PerrordBm, 'k' );
%axis([0,10,-15,0])
% Mile High Club or Bust…

figure
plot( NoiseThreshdBm, PrecdBm, 'k' )
axis( [-90 -79 -95 -77] )
```

```
title( 'Effect of noise threshold level on calculated received power' )
xlabel( 'Noise Threshold (dBm)' )
ylabel( 'Calculated Received Power (dBm)' )

figure
plot( tau*1e9, PDPdBm, 'k' )
title( 'Simulated power delay profile' )
xlabel( 'Propagation Delay (ns)' )
ylabel( 'Received Power (dBm)' )
axis([0,6000,-130,-80]);
```

Calculated RMS Delay Spread Versus Noise Threshold Level

```
% Ndemo4.m
% Plots rms delay spread versus noise threshold for a
% user-defined channel.  True RMS delay spread is also calculated
% for comparison.
% William G. Newhall
% All variables saved in noisex4.mat

clear;

%=======================================
% User definables

do = 1;           % reference distance (m)
freq = 1.9e9;     % frequency (Hz)
delay = ( [1 1.5 2 2.5 3 3.5 4] )*1e-6;   % delays at which path loss is
calculated (sec)
n = 3.5;    % path loss exponent
dist = delay * 3e8;     % distance traversed
RmsDlyCalcNoiseThresh = -170:0.2:-120;

%=======================================
% Calculations
PLdB = pathloss( freq, do, n, delay );    % calculate path loss

[PDPdBm,tau] = pdpgen( delay, 0-PLdB, 1 ); % generate PDP
title('Simulated Power Delay Profile')
axis( [500,4500,-175,-120])

TrueRmsDly = RmsDly( delay, 0-PLdB ) % actual RMS delay

for threshIndex = 1:length( RmsDlyCalcNoiseThresh )
  disp(['Processing ' num2str( threshIndex ) ' out of ' num2str( length(
RmsDlyCalcNoiseThresh ) ) ]);
  ProcRmsDly( threshIndex ) = RmsDly( tau, PDPdBm,
RmsDlyCalcNoiseThresh( threshIndex ) );
end

figure
plot( RmsDlyCalcNoiseThresh, ProcRmsDly*1e9, 'k' );
hold on
```

```
plot( [min(RmsDlyCalcNoiseThresh) max(RmsDlyCalcNoiseThresh)]
,[TrueRmsDly TrueRmsDly]*1e9, 'c')
axis( [-155 -120 0 550] )
hold off
title( 'Calculated RMS delay spread versus noise threshold level')
xlabel( 'Noise Threshold Level (dB)' )
ylabel( 'RMS Delay Spread (ns)' )

figure
plot( RmsDlyCalcNoiseThresh, ProcRmsDly*1e9, 'k' );
hold on
plot( [min(RmsDlyCalcNoiseThresh) max(RmsDlyCalcNoiseThresh)]
,[TrueRmsDly TrueRmsDly]*1e9, 'c')
axis( [-170 -120 0 5200] )
hold off
title( 'Calculated RMS delay spread versus noise threshold level')
xlabel( 'Noise Threshold Level (dB)' )
ylabel( 'RMS Delay Spread (ns)' )
```

Sample Component Detection Run

```
% ProcEx1.m
% Demonstrates power delay profile processing
% using component detection
% William G. Newhall

clear;

%
========================================================================
======
% User defineables
=============================================================

propDelay = [1,2,3,4,5,6]*1e-6;           % multipath propagation delay
(sec)
componentPwr = [0,-4,-8,-12,-16,-20];     % multipath component power
(dBm)
obsLevel = -35;                     % observation level (dBm)
minimumCrossingSep = 0.1e-6;              % minimum crossing sep to
identify signal component
floorLevel = -200;                  % floor level after noise is removed
(dBm)
peakMargindB = 2;             % peak margin (dB)

%
========================================================================
======
% Produce and plot profile
=====================================================

[PdBm, tau] = pdpgen( propDelay, componentPwr, 1 );
title( 'Simulated Power Delay Profile' )
PdBm = PdBm( 1:2500 );        % limit to first 2500 points
```

```
tau = tau( 1:2500 );
axis([ 500,6500, -55,0 ]);
hold on
plot( [ min(tau) max(tau) ]*1e9 , [ obsLevel obsLevel ] , 'r-' );
hold off

%
========================================================================
======
% Find all crossings
===========================================================

for crIndex = 1:( length( PdBm ) - 1 )
  if( PdBm( crIndex ) < obsLevel & PdBm( crIndex+1 ) > obsLevel )
    crossRecord( crIndex ) = 1;
  elseif( PdBm( crIndex ) > obsLevel & PdBm( crIndex+1 ) < obsLevel )
    crossRecord( crIndex ) = -1;
  else
    crossRecord( crIndex ) = 0;
  end % if
end % for crIndex
if( length( crossRecord ) == 0 )
  error( 'No observation level crossings found.' )
end

%
========================================================================
======
% Identify peaks by distance between + to - crossing pairs
=====================

crossIndexes = find( crossRecord );
firstIndex = 1;
lastIndex = length( crossIndexes );
if( crossRecord( crossIndexes( 1 ) ) == -1 )  % use first upward
crossing
  firstIndex = 2;
end
if( crossRecord( crossIndexes( lastIndex ) ) == 1 )
  lastIndex = lastIndex - 1;  % only use last downward crossing
end

for sepIndex = firstIndex : 2 : lastIndex  % by pairs
  % calculate separation between upward and downward crossings
  tauSep( sepIndex ) = tau( crossIndexes( sepIndex+1 ) ) - tau(
crossIndexes( sepIndex ) );
end % for sepIndex

tauSep = tauSep( find( tauSep ) );  % eliminate zero values

%
========================================================================
======
% Decide where components could exist based on separation of crossings
=========

componentIndexes = [];
for sepIndex = 1 : length( tauSep )
  if( tauSep( sepIndex ) > minimumCrossingSep )
    componentIndexes = [ componentIndexes crossIndexes( firstIndex - 1 +
( 2*sepIndex-1 : 2*sepIndex ) )];
  end % if tauSep
end % for sepIndex
if( length( componentIndexes ) == 0 )
  error( 'No Components Found' )
end
% componentIndexes contains endpoints of peaks.  First element is left
bound of first peak,
% second element is right bound of first peak, third element is left
bound of second peak, etc.

hold on     % plot component bounds as x's on the PDP plot
plot( tau( componentIndexes )*1e9, PdBm( componentIndexes ), 'rx' );
hold off

%
========================================================================
======
% Remove noise (between component bound pairs) from plot
=======================

PDPNoNoisedBm = PdBm;
PDPNoNoisedBm( 1 : componentIndexes( 1 ) - 1 ) = floorLevel * ones( 1,
componentIndexes( 1 ) - 1 );
PDPNoNoisedBm( ( componentIndexes( length( componentIndexes ) ) + 1 ) :
...
               length( PDPNoNoisedBm ) ) = floorLevel * ones( 1, ...
               length( PDPNoNoisedBm ) - componentIndexes( length(
componentIndexes ) ) );

for sampleIndex = 2 : 2 : ( length( componentIndexes ) - 2 )
  PDPNoNoisedBm( ( componentIndexes( sampleIndex ) + 1 ) : (
componentIndexes( sampleIndex + 1 ) - 1 ) ) ...
     = floorLevel * ones( 1, componentIndexes( sampleIndex + 1) -
componentIndexes( sampleIndex ) - 1 );
end % for sampleIndex

% plot PDP with noise removed
figure;
plot( tau*1e9, PDPNoNoisedBm, 'k-' )
axis([ 500,6500, -35,0 ]);
title( 'Simulated power delay profile with noise suppressed' )
xlabel( 'Propagation Delay (ns)' )
ylabel( 'Relative Power (dB)' )
```

```
%
========================================================================
======
% Search for all peaks in the "noiseless" profile

peakRecord = [];
peakRecordIndex = 1;
for peakIndex = 2 : ( length( PDPNoNoisedBm ) - 1 )
  if( ( PDPNoNoisedBm( peakIndex ) > PDPNoNoisedBm( peakIndex - 1 ) ) &
...
      ( PDPNoNoisedBm( peakIndex ) > PDPNoNoisedBm( peakIndex + 1 ) ) )
    peakRecord( peakRecordIndex ) = peakIndex;
    peakRecordIndex = peakRecordIndex + 1;
  end % if peak
end % for peakIndex
hold on
plot( tau( peakRecord )*1e9 , PDPNoNoisedBm( peakRecord ), 'rx' )
hold off
title( 'Simulated power delay profile with noise suppressed and peaks
detected' )

%
========================================================================
======
% Eliminate peaks which do not rise above peakMargindB above adjacent
minimums

% Find endpoints of local maxima (peak endpoints)

for peakIndex = 1:length( peakRecord )  % loop through all peak records
  disp( ['Working on peak ' num2str( peakIndex ) ' of ' num2str( length(
peakRecord) ) '...'] );
  minFound = 0;
  peakTemp = peakRecord( peakIndex );
  while( minFound == 0 )  % Find index of minimum on right side of peak
%    peakTemp
%    length( PDPNoNoisedBm )
%    PDPNoNoisedBm( peakTemp )
%    PDPNoNoisedBm( peakTemp + 1 )
    if( peakTemp == length( PDPNoNoisedBm ) ) % right-most point case
      minFound = 1;
      rightMinIndex = peakTemp;
    elseif( PDPNoNoisedBm( peakTemp ) <= PDPNoNoisedBm( peakTemp + 1 ) )
      minFound = 1; % found local minimum
      rightMinIndex( peakIndex ) = peakTemp;
    else
      peakTemp = peakTemp + 1;
    end % if
%    minFound
  end % while (right min)

  minFound = 0;
  peakTemp = peakRecord( peakIndex );
  while( minFound == 0 )  % Find index of minimum on left side of peak
```

```
    if( peakTemp == 1 ) % left-most point case
      minFound = 1;
      leftMinIndex = peakTemp;
    elseif( PDPNoNoisedBm( peakTemp ) <= PDPNoNoisedBm( peakTemp - 1 ) )
      minFound = 1; % found local minimum
      leftMinIndex( peakIndex ) = peakTemp;
    else
      peakTemp = peakTemp - 1;
    end % if
  end % while ( left min )
end % for peakIndex

% Append components to signalRecord if power margin is satisfied

signalRecord = [];
for peakIndex = 1:length( peakRecord )
  if( ( PDPNoNoisedBm( peakRecord( peakIndex ) ) - PDPNoNoisedBm(
leftMinIndex( peakIndex ) ) ...
      > peakMargindB ) & ...
      ( PDPNoNoisedBm( peakRecord( peakIndex ) ) - PDPNoNoisedBm(
rightMinIndex( peakIndex ) ) ...
      > peakMargindB ) )
    signalRecord = [signalRecord peakRecord( peakIndex )];
  end  % if
end % for peakIndex

hold on
plot( tau( signalRecord )*1e9 , PDPNoNoisedBm( signalRecord ), 'bo' )
hold off

% Check that each peak has at least one signal component assigned to it
assignmentCheck = zeros( 1, ( length( componentIndexes ) / 2 ) );  % one
element for each peak
for signalIndex = 1 : length( signalRecord )
  for checkIndex = 1 : length( assignmentCheck )
    if( ( signalRecord( signalIndex ) > componentIndexes( 2*checkIndex-1
) ) & ...
        ( signalRecord( signalIndex ) < componentIndexes( 2*checkIndex )
) )
      assignmentCheck( checkIndex ) = assignmentCheck( checkIndex ) + 1;
      break;
    end % if
  end % checkIndex
end % for signalIndex

% assignment check is a vector whos elements are the number of
"signalRecord peaks" assigned
% to each peak.  For example, if three peaks exist, and one signal is
detected in each of the
% first two, and two signal components are detected in the last peak,
then assignmentCheck = [1 1 2]

% assign a signal component if no detected signal exists within a peak.
Assign the max value
```

```
% of the peak.

for checkIndex = 1 : length( assignmentCheck )
  if( assignmentCheck( checkIndex ) == 0 )
    [ maxTmp , maxIndexTmp ] = max( PDPNoNoisedBm( componentIndexes(
2*checkIndex-1 ) : componentIndexes( 2*checkIndex ) ) );
    maxIndexTmp = maxIndexTmp + componentIndexes( 2*checkIndex-1 ) - 1;
    signalRecord = [ signalRecord maxIndexTmp ];
  end % if
end % for checkIndex

figure
plotline( tau( signalRecord )*1e9 , PDPNoNoisedBm( signalRecord ), -200,
'k-' )
axis([ 500,6500, -35,0 ]);
title( 'Impulse representation of power delay profile' )
xlabel( 'Propagation Delay (ns)' )
ylabel( 'Relative Power (dB)' )
```

Produces Power Delay Profile for Detection Processing Simulation

```
% PdpGen.m
%
% William G. Newhall
%
% Plots simulated power delay profile using the self-noise simulation
model.
% SigDelay is a vector that defines the delays for the signal component
powers
% given in SigPower.  Pdbm is the power delay profile in dBm and P is
the power
% delay profile in mW.  Use plotflag=1 to plot resulting profile.
%
% Simulation uses a 9-stage shift register [4,9] an a 10 MHz chip rate.
% Probsig14.mat defines the probing signal in varibles y and t.  Run
Probsig14.m
% to produce this file and place in same directory as Pdpgen.m.  A
Gaussian
% filter is used for this simulation.
%
% function [Pdbm,tau] = pdpgen( SigDelay, SigPowerdBm, plotflag );

function [Pdbm,tau] = pdpgen( SigDelay, SigPowerdBm, plotflag );

chDelSec    = SigDelay;       % Channel delay vector (sec)
chAttendB   = 0 - SigPowerdBm;      % Channel attenuation vector (dB)
K = 1000;                     % slide factor used for probing signal

% ============================================================
% Calculations

% load probing signal
load probsg14;    % read y and t from disk
t = t/K;    % account for slide factor
```

```
y=y/max(y); % Normalize probing signal to 0 dB

samplePeriod = ( max(t)-min(t) ) / length( y ); % calculate profile
sample period
chDel = round( chDelSec / samplePeriod ); % calculate channel delay
vector
PDP = ( chan( y, chDel, chAttendB ) ).^2;       % produce channel output
PDPdB = 10*log10( abs(PDP) );       % form PDP in dB

if( nargin == 3 )  % check for 3 input argyments
  if( plotflag ~= 0 )
    figure
    plot( t*1e9, PDPdB, 'k-');
    %axis( [0,5000,-40,0] )
    title( 'Simulated Power Delay Profile (Gaussian Filter)' )
    xlabel( 'Propagation Delay (ns)' )
    ylabel( 'Relative Power (dB)' )
    grid on
  end
end

tau=t;
Pdbm = PDPdB;
P = 10.^(PDPdB/10);
```

Component Detection/Conventional RMS Delay Spread Comparison

```
% plotpr4.m
% Plots rms dly vs obs level/noise thresh using procex14.mat data.

clear;

load procex4

figure
plot( obsLevelArray, rmsRecord*1e9, 'k-' );
title( 'Calculated RMS Delay Spread' )
xlabel( 'Relative Observation Level or Noise Threshold Level (dB)' )
ylabel( 'RMS Delay Spread (ns)' )
axis( [-60 -5 0 2000] )

load ndemo6

hold on
plot( RmsDlyCalcNoiseThresh, ProcRmsDly*1e9, 'k:' );
hold off
```

Data for Detection/Conventional RMS Delay Spread Comparison

```
% ndemo6.m
% Plots rms delay spread versus noise threshold for a
% user-defined channel.  True RMS delay spread is also calculated
% for comparison.
% William G. Newhall
% All variables saved in noisex4.mat
```

```
clear;

%=======================================
% User definables

delay = [1,2,3,4,5,6]*1e-6;   % delays at which path loss is calculated
(sec)
RmsDlyCalcNoiseThresh = -60:0.5:0;
componentPwrdBm = [0,-4,-8,-12,-16,-20];  %

%=======================================
% Calculations

[PDPdBm,tau] = pdpgen( delay, componentPwrdBm , 1 ); % generate PDP
title('Simulated Power Delay Profile')
axis([ 500,6500, -55,0 ]);

TrueRmsDly = RmsDly( delay, componentPwrdBm ) % actual RMS delay

for threshIndex = 1:length( RmsDlyCalcNoiseThresh )
  disp(['Processing ' num2str( threshIndex ) ' out of ' num2str( length(
RmsDlyCalcNoiseThresh ) ) ]);
  ProcRmsDly( threshIndex ) = RmsDly( tau, PDPdBm,
RmsDlyCalcNoiseThresh( threshIndex ) );
end

figure
plot( RmsDlyCalcNoiseThresh, ProcRmsDly*1e9, 'k' );
hold on
plot( [min(RmsDlyCalcNoiseThresh) max(RmsDlyCalcNoiseThresh)]
,[TrueRmsDly TrueRmsDly]*1e9, 'c')
axis( [-50 0 0 1000] )
hold off
title( 'Calculated RMS delay spread versus noise threshold level')
xlabel( 'Noise Threshold Level (dB)' )
ylabel( 'RMS Delay Spread (ns)' )

figure
plot( RmsDlyCalcNoiseThresh, ProcRmsDly*1e9, 'k' );
hold on
plot( [min(RmsDlyCalcNoiseThresh) max(RmsDlyCalcNoiseThresh)]
,[TrueRmsDly TrueRmsDly]*1e9, 'c')
%axis( [-60 0 0 1000] )
hold off
title( 'Calculated RMS delay spread versus noise threshold level')
xlabel( 'Noise Threshold Level (dB)' )
ylabel( 'RMS Delay Spread (ns)' )

save ndemo6
```

Data For Detection/Conventional RMS Delay Spread Comparison

```
% ProcEx4.m
% Demonstrates power delay profile processing
% using component detection
% William G. Newhall

clear;

%
========================================================================
======
% User defineables
=============================================================

propDelay = [1,2,3,4,5,6]*1e-6;           % multipath propagation delay
(sec)
componentPwr = [0,-4,-8,-12,-16,-20];     % multipath component power
(dBm)
% obsLevel = -35;                   % observation level (dBm)
obsLevelArray = -60:0.5:-5;
minimumCrossingSep = 0.1e-6;              % minimum crossing sep to
identify signal component
floorLevel = -200;                  % floor level after noise is removed
(dBm)
peakMargindB = 2;             % peak margin (dB)

%
========================================================================
======
% Produce and plot profile
=====================================================

[PdBm, tau] = pdpgen( propDelay, componentPwr, 1 );
title( 'Simulated Power Delay Profile' )
PdBm = PdBm( 1:2500 );        % limit to first 2500 points
tau = tau( 1:2500 );
axis([ 500,6500, -55,0 ]);
%hold on
%plot( [ min(tau) max(tau) ]*1e9 , [ obsLevel obsLevel ] , 'r-' );
%hold off

% Loop through following code once for each observation level
%
************************************************************************
*************
for obsLoopIndex = 1:length( obsLevelArray )
  disp( ['Process number ' num2str( obsLoopIndex) ' of ' num2str(
length( obsLevelArray ) )] )
  obsLevel = obsLevelArray( obsLoopIndex );
%
========================================================================
======
% Find all crossings
===========================================================

crossRecord = [];
```

```
for crIndex = 1:( length( PdBm ) - 1 )
  if( PdBm( crIndex ) < obsLevel & PdBm( crIndex+1 ) > obsLevel )
    crossRecord( crIndex ) = 1;
  elseif( PdBm( crIndex ) > obsLevel & PdBm( crIndex+1 ) < obsLevel )
    crossRecord( crIndex ) = -1;
  else
    crossRecord( crIndex ) = 0;
  end % if
end % for crIndex
if( length( crossRecord ) == 0 )
  error( 'No observation level crossings found.' )
end

%
========================================================================
======
% Identify peaks by distance between + to - crossing pairs
=====================

crossIndexes = find( crossRecord );
firstIndex = 1;
lastIndex = length( crossIndexes );
if( crossRecord( crossIndexes( 1 ) ) == -1 )  % use first upward
crossing
  firstIndex = 2;
end
if( crossRecord( crossIndexes( lastIndex ) ) == 1 )
  lastIndex = lastIndex - 1;  % only use last downward crossing
end

tauSep = [];
for sepIndex = firstIndex : 2 : lastIndex  % by pairs
  % calculate separation between upward and downward crossings
  tauSep( sepIndex ) = tau( crossIndexes( sepIndex+1 ) ) - tau(
crossIndexes( sepIndex ) );
end % for sepIndex

tauSep = tauSep( find( tauSep ) );  % eliminate zero values

%
========================================================================
======
% Decide where components could exist based on separation of crossings
=========

componentIndexes = [];
for sepIndex = 1 : length( tauSep )
  if( tauSep( sepIndex ) > minimumCrossingSep )
    componentIndexes = [ componentIndexes crossIndexes( firstIndex - 1 +
( 2*sepIndex-1 : 2*sepIndex ) )];
  end % if tauSep
end % for sepIndex
if( length( componentIndexes ) == 0 )
  error( 'No Components Found' )
```

```
end
% componentIndexes contains endpoints of peaks.  First element is left
bound of first peak,
% second element is right bound of first peak, third element is left
bound of second peak, etc.

%hold on    % plot component bounds as x's on the PDP plot
%plot( tau( componentIndexes )*1e9, PdBm( componentIndexes ), 'rx' );
%hold off

%
========================================================================
======
% Remove noise (between component bound pairs) from plot
=======================

PDPNoNoisedBm = PdBm;
PDPNoNoisedBm( 1 : componentIndexes( 1 ) - 1 ) = floorLevel * ones( 1,
componentIndexes( 1 ) - 1 );
PDPNoNoisedBm( ( componentIndexes( length( componentIndexes ) ) + 1 ) :
...
               length( PDPNoNoisedBm ) ) = floorLevel * ones( 1, ...
               length( PDPNoNoisedBm ) - componentIndexes( length(
componentIndexes ) ) );

for sampleIndex = 2 : 2 : ( length( componentIndexes ) - 2 )
  PDPNoNoisedBm( ( componentIndexes( sampleIndex ) + 1 ) : (
componentIndexes( sampleIndex + 1 ) - 1 ) ) ...
     = floorLevel * ones( 1, componentIndexes( sampleIndex + 1) -
componentIndexes( sampleIndex ) - 1 );
end % for sampleIndex

% plot PDP with noise removed
%figure;
%plot( tau*1e9, PDPNoNoisedBm, 'k-' )
%axis([ 500,6500, -35,0 ]);
%title( 'Simulated power delay profile with noise suppressed' )
%xlabel( 'Propagation Delay (ns)' )
%ylabel( 'Relative Power (dB)' )


%
========================================================================
======
% Search for all peaks in the "noiseless" profile

peakRecord = [];
peakRecordIndex = 1;
for peakIndex = 2 : ( length( PDPNoNoisedBm ) - 1 )
  if( ( PDPNoNoisedBm( peakIndex ) > PDPNoNoisedBm( peakIndex - 1 ) ) &
...
      ( PDPNoNoisedBm( peakIndex ) > PDPNoNoisedBm( peakIndex + 1 ) ) )
    peakRecord( peakRecordIndex ) = peakIndex;
    peakRecordIndex = peakRecordIndex + 1;
  end % if peak
end % for peakIndex
hold on
%plot( tau( peakRecord )*1e9 , PDPNoNoisedBm( peakRecord ), 'rx' )
%hold off
%title( 'Simulated power delay profile with noise suppressed and peaks
detected' )

%
========================================================================
======
% Eliminate peaks which do not rise above peakMargindB above adjacent
minimums

% Find endpoints of local maxima (peak endpoints)

rightMinIndex = [];
leftMinIndex = [];

for peakIndex = 1:length( peakRecord )  % loop through all peak records
  disp( ['Working on peak ' num2str( peakIndex ) ' of ' num2str( length(
peakRecord) ) '...'] );
  minFound = 0;
  peakTemp = peakRecord( peakIndex );
  while( minFound == 0 )  % Find index of minimum on right side of peak
%    peakTemp
%    length( PDPNoNoisedBm )
%    PDPNoNoisedBm( peakTemp )
%    PDPNoNoisedBm( peakTemp + 1 )
    if( peakTemp == length( PDPNoNoisedBm ) ) % right-most point case
      minFound = 1;
      rightMinIndex = peakTemp;
    elseif( PDPNoNoisedBm( peakTemp ) <= PDPNoNoisedBm( peakTemp + 1 ) )
      minFound = 1; % found local minimum
      rightMinIndex( peakIndex ) = peakTemp;
    else
      peakTemp = peakTemp + 1;
    end % if
%    minFound
  end % while (right min)

  minFound = 0;
  peakTemp = peakRecord( peakIndex );
  while( minFound == 0 )  % Find index of minimum on left side of peak
    if( peakTemp == 1 ) % left-most point case
      minFound = 1;
      leftMinIndex = peakTemp;
    elseif( PDPNoNoisedBm( peakTemp ) <= PDPNoNoisedBm( peakTemp - 1 ) )
      minFound = 1; % found local minimum
      leftMinIndex( peakIndex ) = peakTemp;
    else
      peakTemp = peakTemp - 1;
    end % if
  end % while ( left min )
end % for peakIndex

% Append components to signalRecord if power margin is satisfied

signalRecord = [];
for peakIndex = 1:length( peakRecord )
  if( ( PDPNoNoisedBm( peakRecord( peakIndex ) ) - PDPNoNoisedBm(
leftMinIndex( peakIndex ) ) ...
      > peakMargindB ) & ...
      ( PDPNoNoisedBm( peakRecord( peakIndex ) ) - PDPNoNoisedBm(
rightMinIndex( peakIndex ) ) ...
      > peakMargindB ) )
    signalRecord = [signalRecord peakRecord( peakIndex )];
  end  % if
end % for peakIndex

%hold on
%plot( tau( signalRecord )*1e9 , PDPNoNoisedBm( signalRecord ), 'bo' )
%hold off

% Check that each peak has at least one signal component assigned to it
assignmentCheck = zeros( 1, ( length( componentIndexes ) / 2 ) );  % one
element for each peak
for signalIndex = 1 : length( signalRecord )
  for checkIndex = 1 : length( assignmentCheck )
    if( ( signalRecord( signalIndex ) > componentIndexes( 2*checkIndex-1
) ) & ...
        ( signalRecord( signalIndex ) < componentIndexes( 2*checkIndex )
) )
      assignmentCheck( checkIndex ) = assignmentCheck( checkIndex ) + 1;
      break;
    end % if
  end % checkIndex
end % for signalIndex

% assignment check is a vector whos elements are the number of
"signalRecord peaks" assigned
% to each peak.  For example, if three peaks exist, and one signal is
detected in each of the
% first two, and two signal components are detected in the last peak,
then assignmentCheck = [1 1 2]
% Dancing on the beach, getting taken “for a ride” in Boston, I don’t
know about you, Roger!

% assign a signal component if no detected signal exists within a peak.
Assign the max value
% of the peak.

for checkIndex = 1 : length( assignmentCheck )
  if( assignmentCheck( checkIndex ) == 0 )
    [ maxTmp , maxIndexTmp ] = max( PDPNoNoisedBm( componentIndexes(
2*checkIndex-1 ) : componentIndexes( 2*checkIndex ) ) );
    maxIndexTmp = maxIndexTmp + componentIndexes( 2*checkIndex-1 ) - 1;
    signalRecord = [ signalRecord maxIndexTmp ];
  end % if
end % for checkIndex

%figure
%plotline( tau( signalRecord )*1e9 , PDPNoNoisedBm( signalRecord ), -
200, 'k-' )
%axis([ 500,6500, -35,0 ]);
%title( 'Impulse representation of power delay profile' )
%xlabel( 'Propagation Delay (ns)' )
%ylabel( 'Relative Power (dB)' )

rmsRecord( obsLoopIndex ) = rmsdly( tau( signalRecord ), PDPNoNoisedBm(
signalRecord ), floorLevel+1 );

end % for obsLoopIndex
%
************************************************************************
*************

figure
plot( obsLevelArray, rmsRecord*1e9, 'k-' );
title( 'Calculated RMS Delay Spread' )
xlabel( 'Relative Observation Level or Noise Threshold Level (dB)' )
ylabel( 'RMS Delay Spread (ns)' )

save ProcEx4
```

Calculates RMS Delay Spread for Profiles

```
% RmsDly.m
%
% William G. Newhall
%
% Calculates RMS delay spread for an input power
% delay profile.
% tau is a vector of delays (any time units).  PdBm is a vector of power
in dBm.
% NoiseThreshdBm is the noise threshold in dBm (below which all points
are
% discarded from the calculation).  If NoiseThreshdBm is left out, no
noise
% threshold is applied.
%
% function RMSdlySpread = RmsDly( tau, PdBm, NoiseThreshdBm )
%

function RMSdlySpread = RmsDly( tau, PdBm, NoiseThreshdBm )

P = 10.^( PdBm/10 );
NumPts = length( tau );

% Handle noise threshold
```

```
if( nargin == 3 ) % noise thresh defined if 3 arguments
  NoiseThresh = 10^( NoiseThreshdBm/10 );
  for i=1:NumPts
    if( P(i) < NoiseThresh )  % set all power below noise thresh to 0 mW
      P(i) = 0;
    end
  end
end

% Calculate RMS Delay spread
if( sum( P ) ~= 0 )
  taubar = ( P*tau' ) / sum( P );
  tausquaredbar = ( P*( tau.^2 )' ) / sum( P );
  RMSdlySpread = sqrt( tausquaredbar - ( taubar^2 ) );
else
  RMSdlySpread = 0;
end
```

Vitae

William George Newhall

Bill Newhall was born on January 15, 1972, in Somerville, New Jersey. He grew up in Bridgewater, New Jersey, and graduated from Bridgewater-Raritan High School-West in 1990. Bill studied at Virginia Polytechnic Institute and State University for his Bachelor of Science degree in Electrical Engineering. He was named General Motors Scholar in 1992 and was the recipient of several academic scholarships, including the Radio Club of America Scholarship and the Golden Key Scholarship. Bill graduated In Honors and *Summa Cum Laude* in 1994.

In 1994, Bill Newhall began his work toward a Master of Science degree in Electrical Engineering at Virginia Tech and began his research with the Mobile and Portable Radio Research Group. Bill worked as a lab instructor, research assistant, and product development engineer while progressing toward his Masters degree. His research involved many aspects of wireless communications engineering, including RF measurements, RF hardware design and development, system simulation, and software product development. His engineering experience while obtaining his academic degrees includes work terms at Grayson Wireless, General Motors, and AT&T.

In 1995, Bill began working with Grayson Wireless to assist in the development of wireless test and measurement equipment. He has authored several published papers and professional presentations while working at Virginia Tech and Grayson Wireless. Bill began pursuing his Ph.D. in Electrical Engineering at Virginia Tech in 1997.

Bill is an active private pilot and a serves the community as a volunteer in the Civil Air Patrol.