

AN EXPERIMENTAL SYSTEM FOR CHARACTERIZING
WIDEBAND CDMA VECTOR CHANNELS AND
SMART ANTENNAS

by

Noel Tai-Tung Tin

A thesis submitted to the
Department of Electrical and Computer Engineering
in conformity with the requirements
for the degree of Master of Science (Engineering)

Queen's University
Kingston, Ontario, Canada
December, 2000

Copyright © Noel Tai-Tung Tin, 2000

Abstract

Computer simulations of mobile communication systems often use ideal or simple channel propagation models. This assumption is sufficient for relative comparison of algorithms or for testing overall functionality, but insufficient when the actual performance in a physical environment is required. In order to perform physical experiments, a low-cost modular wideband CDMA smart antenna measurement system operating in the 1.9 GHz band has been designed and constructed.

Previous approaches include the studying of DOA (direction-of-arrival) estimation, spatial signature or angle spread at the 1.8-2.0 GHz band using narrowband measurement systems. Other wideband testbeds employ a single transmitter to analyze antenna array receiver performance, beamforming and multipath delay profile.

Our complete system can operate within the frequency range of 1-2 GHz, with multiple portable transmitters and a multiple element antenna array receiver base-station. The transmitters are programmable, allowing varying PN chip-rates and PN code sequences, and the receiver can simultaneously acquire 16k samples per antenna element at 5 samples per PN chip.

Our experimental data allowed us to characterize an outdoor wideband CDMA

channel including multipath peak amplitudes and delays, channel and spatial signature variation. In addition, using signal subspace processing, we compared mobile-to-basestation beamforming performance to that of a single antenna. Results from a 4-user scenario show that a user can get an average SINR gain of 0.176 to 3.335 dB from maximum SINR beamforming.

Acknowledgements

First, I would like to thank my supervisor, Dr. Steven Blostein for his guidance and support throughout this research. I would also like to thank CITR and Dr. Steven Blostein for the generous financial support. Many thanks to the staff at Technical Services, especially Mr. David Lay, for his great help in our hardware implementation.

I would like to thank my parents and my two sisters, who have shown love, patience and confidence in me throughout my academic career at Queen's. Special thanks to Pauline, who has provided me with emotional support and encouragement through good and bad times.

Finally, I would like to thank all the IPCL members for helping in our field measurements, without which our experiments would have been very difficult to complete.

Contents

Abstract	ii
Acknowledgements	iv
Symbols and Abbreviations	xvi
1 Introduction	1
1.1 Motivation	1
1.2 Summary of Contributions	3
1.3 Thesis outline	3
2 Background	5
2.1 Third Generation Mobile Communications	5
2.2 Antenna Array and Beamforming	6
2.3 Mobile Radio Propagation Environment	8
3 Wideband CDMA Experimental System Design	11
3.1 Introduction	11
3.2 Previous System Designs and Specifications	12

3.3	System Description	13
3.3.1	Transmitter Design and Features	14
3.3.2	Receiver Design and Features	17
3.3.3	System Parameters, Setup and Integration	19
3.3.4	Data Collection	23
3.4	Summary	25
4	System Validation and Received Signal Processing	26
4.1	Introduction	26
4.2	Functional Block Testing	27
4.2.1	Transmitter Link Testing	27
4.2.2	Receiver Link Testing	30
4.3	System Analysis and Test	36
4.4	Synchronization Issues of the Measurement System	40
4.4.1	Carrier Synchronization	41
4.4.2	Synchronous Data Sampling	42
4.4.3	Test for Antenna Array Calibration	46
4.4.4	Fractional Chip Sampling Offset	48
4.5	Received Signal Processing	49
4.5.1	Real to Complex Data Conversion	50
4.5.2	Reference PN sequence generation	55
4.5.3	Received Signal Despreading and Code Tracking	56
4.6	Summary	62

5	Wideband CDMA Propagation Characteristics	63
5.1	Introduction	63
5.2	First Outdoor Experimental Setup	64
5.3	Path Loss Exponent Analysis	68
5.4	Multipath Delay Profile Analysis	70
5.4.1	Thresholding for Multipath Detection	71
5.4.2	Multipath Detection and Hypothesis Testing	75
5.4.3	Peak Amplitude and Time Delay Calculations	78
5.5	Single User Multipath Detection	79
5.6	Multiple-User Multipath Detection	85
5.7	Time Variation of Delay Profiles	88
5.7.1	Delay Profile Correlations	89
5.7.2	Multipath Detection of Averaged Profiles	94
5.8	Summary	98
6	Array Signal Processing and Beamforming	100
6.1	Introduction	100
6.2	Fading Correlation between Antenna elements	101
6.2.1	Calculation of Correlations and Results	101
6.3	Second Outdoor Experimental setup	103
6.4	Analysis of Spatial Stability using Beampatterns	104
6.4.1	Beampatterns of Experiment Scenarios	105
6.5	Beamforming Receiver using Signal Subspace Processing	109

6.5.1	Beamforming Receiver	109
6.5.2	Estimating Beamforming Weights	112
6.5.3	Eigenvalues Analysis and Results	114
6.5.4	DOA estimation	121
6.6	SINR Improvement from Beamforming	124
6.7	Summary	131
7	Conclusion	132
7.1	Summary	132
7.2	Future Directions	134
7.3	Conclusion	135
A	Synchronization of Sampled Data	137
A.1	Hardware Setup	137
A.1.1	A/D converter interface	139
A.1.2	FIFO Data Read and Synchronization	139
A.2	Parallel Port Read and Write	142
A.3	Software Control and Data Read	146
A.3.1	Software Synchronization	146
A.3.2	<i>ADport</i> Program Listing	149
	References	160
Vita		166

List of Tables

4.1	Feedback Connections for Linear m-sequences	56
5.1	Measured received signal power from outdoor experiment	69
5.2	Path Loss Exponents for Different Environments [36]	70
5.3	Number of Multipaths Detected: $P_{FA} = 10^{-2}$, resolution = 1/5-chip .	82
5.4	Amplitudes and Delays of Multipaths: $P_{FA} = 10^{-2}$, resolution = 1/5-chip	82
5.5	Number of Multipaths Detected: $P_{FA} = 10^{-3}$, resolution = 1/5-chip .	82
5.6	Amplitudes and Delays of Multipaths: $P_{FA} = 10^{-3}$, resolution = 1/5-chip	82
5.7	Number of Multipaths Detected: $P_{FA} = 10^{-2}$, resolution = 1/2-chip .	83
5.8	Amplitudes and Delays of Multipaths: $P_{FA} = 10^{-2}$, resolution = 1/2-chip	83
5.9	Number of Multipaths Detected: $P_{FA} = 10^{-3}$, resolution = 1/2-chip .	83
5.10	Amplitudes and Delays of Multipaths: $P_{FA} = 10^{-3}$, resolution = 1/2-chip	83
5.11	Number of Multipaths Detected: $P_{FA} = 10^{-2}$, resolution = 1-chip . .	83
5.12	Amplitudes and Delays of Multipaths: $P_{FA} = 10^{-2}$, resolution = 1-chip	84
5.13	Number of Multipaths Detected: $P_{FA} = 10^{-3}$, resolution = 1-chip . .	84
5.14	Amplitudes and Delays of Multipaths: $P_{FA} = 10^{-3}$, resolution = 1-chip	84
5.15	Number of Multipaths Detected: $P_{FA} = 10^{-4}$, resolution = 1/2-chip .	84
5.16	Amplitudes and Delays of Multipaths: $P_{FA} = 10^{-4}$, resolution = 1/2-chip	84

5.17	Number of Multipaths Detected: 3 users	88
5.18	Amplitudes and Delays of Multipaths: 3 users	88
5.19	Correlations between single and averaged profiles	92
5.20	Number of multipaths detected: window = 1 bit	96
5.21	Amplitudes and delays of multipaths: window = 1 bit	96
5.22	Number of multipaths detected: window = 5 bits	97
5.23	Amplitudes and delays of multipaths: window = 5 bits	97
5.24	Number of multipaths detected: window = 10 bits	97
5.25	Amplitudes and delays of multipaths: window = 10 bits	97
5.26	Number of multipaths detected: window = 25 bits	97
5.27	Amplitudes and delays of multipaths: window = 25 bits	98
5.28	Number of multipaths detected: window = 50 bits	98
5.29	Amplitudes and delays of multipaths: window = 50 bits	98
6.1	Fading Correlation of two antennas: all trials	103
6.2	Fading Correlation of two antennas: trials with no multipaths	103
6.3	Eigenvalues of Pre-despreading Covariance Matrices	116
6.4	Eigenvalues of Covariance matrices with despreading of User A	117
6.5	Eigenvalues of Covariance matrices with despreading of User B	118
6.6	Eigenvalues of Covariance matrices with despreading of User C	119
6.7	Eigenvalues of Covariance matrices with despreading of User D	120
6.8	SINR gain from Maximum SINR Beamforming	127
6.9	SINR gain from Maximum SNR Beamforming	128

6.10 SINR gain from Maximum SINR Beamforming: N_o assumed	129
6.11 SINR gain from Maximum SNR Beamforming: N_o assumed	130
A.1 Pin Assignments for Enhanced Parallel Port connector	144

List of Figures

2.1	Smart Antennas vs. Standard Technology [26]	7
2.2	Uniform Linear Array	8
2.3	Example of the Wireless Propagation Environment [43]	9
2.4	Rake Receiver Block Diagram	10
3.1	Direct sequence RF transmitter design	14
3.2	Previous transmitter link design	15
3.3	Digital wideband receiver design	17
3.4	Direct bandpass sampling	19
3.5	Photograph of TX and RX hardware and other testing equipment . .	21
3.6	Photograph of a portable transmitter	22
3.7	Photograph of all four portable transmitters	22
3.8	Photograph of receiving basestation	23
4.1	Direct sequence RF transmitter block diagram	28
4.2	Baseband PN chip waveforms	29
4.3	Baseband PN sequence power spectrum	29
4.4	RF carrier modulated PN sequence power spectrum	30

4.5	RF spectrum at power amplifier output	31
4.6	Receiver link block diagram	32
4.7	Received RF spectrum before low-noise amplifier	32
4.8	Received RF spectrum after low-noise amplification	33
4.9	IF spectrum at IMJ mixer output	33
4.10	IF spectrum after bandpass filtering	34
4.11	Sampled received signal at baseband	35
4.12	Receiver link budget	37
4.13	Single tone receiver sensitivity test: input power -70 dBm	39
4.14	Single tone receiver sensitivity test: input power -50 dBm	40
4.15	Carrier synchronization hardware logic	41
4.16	No carrier synchronization	42
4.17	RF carrier synchronization only	43
4.18	Carrier synchronization achieved using hardware logic	43
4.19	ADport control software	44
4.20	Flowchart for synchronous sampling	45
4.21	Hardware setup for back-to-back test	47
4.22	Back-to-back test: synchronous data collection achieved	47
4.23	RF front-end calibration experiment	48
4.24	Different receiver systems: i) Quadrature (I-Q channel) and ii) One channel digital conversion to complex I-Q channel	51
4.25	I-Q conversion algorithm Step 2 - Truncation of spectrum	54

4.26	Modular multiple-tap sequence generator (MSRG)	55
4.27	Digital early-late tracking loop	59
4.28	Despreading without code tracking loop	60
4.29	Despreading with code tracking loop	61
5.1	Outdoor Experiment #1: scenario 1	65
5.2	Outdoor Experiment #1: scenario 2	66
5.3	Outdoor Experiment #1: scenario 3	67
5.4	Outdoor Experiment #1: scenario 4	67
5.5	Sample Multipath Delay Profile	72
5.6	Delay Profile and Thresholding ($P_{FA} = 10^{-3}$)	75
5.7	Delay Profile and Thresholding (with smoothing)	76
5.8	Detection Performance of CFAR Multipath Detection for Different False Alarm Probabilities	78
5.9	Typical Delay Profile of Multiple Users	86
5.10	Delay Profile of Multiple Users with Fading effects	87
5.11	Collection of delay profiles for Trial #1	90
5.12	Collection of delay profiles for Trial #14	91
5.13	Average delay profiles for Trial #1	93
5.14	Average delay profiles for Trial #14	94
5.15	First 5 Multipath Delay Profiles: Trial #14	95
5.16	First 5 Multipath Delay Profiles: Trial #15	96
6.1	Outdoor experiment #2: four TXs and 4-element array	104

6.2	Beampattern of Outdoor Experiment #2: Scenario 1	106
6.3	Beampatterns of despread sequences: User A	107
6.4	Beampatterns of despread sequences: User B	107
6.5	Beampatterns of despread sequences: User C	108
6.6	Beampatterns of despread sequences: User D	108
6.7	Beamforming Receiver for the Desired User	110
6.8	Histogram of DOA estimates: User A	122
6.9	Histogram of DOA estimates: User B	122
6.10	Histogram of DOA estimates: User C	123
6.11	Histogram of DOA estimates: User D	123
A.1	AD6620 evaluation board system block diagram	138
A.2	A/D converter connector and data latches	140
A.3	FIFO schematic and output registers	141
A.4	FIFO write enable signal "DVN" generation	143
A.5	Parallel port interface, data and command latch	145
A.6	Timing Diagram for FIFO commands and Data Read	147

Symbols and Abbreviations

3G	third generation
\mathbf{a}_i	array response vector for user i
A/D	analog-to-digital
ASIC	application specific integrated chip
BPF	bandpass filter
$c_i(n)$	pseudo noise sequence for user i at time n
CDMA	code division multiple access
CFAR	constant false alarm rate
CW	carrier wave
dBm	decibels related to 1 mW, where 0 dBm = 1 mW
DFT	discrete Fourier transform
DOA	direction of arrival
DSP	digital signal processing
f_c	carrier frequency
FFT	fast Fourier transform
FIFO	first-in-first-out memory register
G	number of chips per symbol
I_M	identity matrix of size $M \times M$
IF	intermediate frequency
IMJ	image rejection
IN_i	interference and noise term for user i
I-Q	in-phase and quadrature

LPF	lowpass filter
LOS	line-of-sight
MAI	multiple access interference
Mcps	Mega chips per second (10^6 chips/sec)
MMSE	minimum mean squared error
PCS	personal communication system
P_D	probability of detection
P_{FA}	probability of false alarm
P_i	total received power from user i by array elements
\overline{PL}	average path loss
PN	pseudo-random noise
RF	radio frequency
\mathbf{R}_{NI}	noise-interference covariance matrix
RX	receiver
\mathbf{R}_{xx}	pre-despreading received signal covariance matrix
$\mathbf{R}_{y_i y_i}$	post-despreading covariance matrix of user i
σ_N^2	noise variance
SINR	signal-to-interference-noise ratio
SNR	signal-to-noise ratio
T_b	bit (or symbol) period
T_c	PN chip period
TX	transmitter
$\mathbf{v}(n)$	additive white Gaussian noise vector
ULA	uniform linear array
y_{B_i}	beamformer's output for user i

Chapter 1

Introduction

1.1 Motivation

Computer simulations of mobile communication systems often use ideal or simple channel propagation models. This assumption is sufficient for relative comparison of algorithms or for testing overall functionality, but insufficient when the actual performance in a physical environment is required.

The performance of wireless communication systems is bounded by the physical limitations imposed by the mobile radio channel [56]. The transmission path between the transmitter and receiver can vary from line-of-sight (LOS) to a path that is blocked by buildings, mountains, etc. Modeling the radio channel has been one of the more difficult aspects of mobile radio system design. It is usually modeled statistically based on experimental measurements made for the proposed communication system and is specific to operating frequency band [56] [60].

Several mobile communication measurement systems were designed and experiments were performed for the 910 *MHz* band in the last two decades including [5] [8] [9] [16]. They were successful in characterizing the channel for the first and early part

of the second generation mobile phone systems. Since propagation characteristics are quite different in the PCS band, in recent years research efforts have switched to the $1.8 - 2.0GHz$ band. Some recent measurement systems were created with the objective to either measure or estimate direction of arrivals (DOA), multipath delays and study the feasibility of beamforming in second and third generation (3G) mobile communication systems.

Reviewing the open literature, several research groups have designed and conducted field testing in the $1.8 - 2.0GHz$ frequency band [7] [41] [46] [58] [69]. As each group's hardware, setup and objectives differ, their systems' capabilities and limitations also vary. In particular, none has employed wideband CDMA with multiple transmitters. Our measurement system presented in this thesis is a prototype wide-band CDMA mobile system which offers a high chip rate and data storage for off-line signal processing, applicable mainly to the uplink (mobile to basestation) of a cellular communication system. It consists of four programmable portable transmitters, and one 4-element receiving antenna array basestation.

In general, there are two main levels of synchronization in a coherent CDMA system; carrier and symbol (or bit) synchronization. For a smart antenna system using coherent processing, synchronous sampling of all antenna elements adds an additional level of synchronization. From the study of previous works, we found that they tend to focus the discussions on results, and only give a brief mention on the design of their testbeds. Therefore, Chapter 3 and 4 are devoted mainly on design, system analysis and performance verification of our measurement system, aiming to provide a background for further improvement and modification of our current design.

With the capability and features of our testbed, we can gather data in a wide-band CDMA multiple-user environment. The transmitters placed at different locations create different scenarios, for example, single or multiple users, equally-spaced

or closely-spaced users, and near-far users. Then beamforming, multipath or spatial-temporal analysis can be performed on the off-line data samples.

1.2 Summary of Contributions

The major contributions of this thesis are:

- the design and construction of a wideband CDMA smart antenna measurement system. The key features of our system include: (1) multiple portable transmitters (four), (2) a basestation receiver with a 4-element antenna array, (3) high chip rate of 7 Mcps, (4) fine sampling resolution of 5 samples per chip, and (5) a method of achieving low-cost carrier synchronization and synchronous sampling by hardware logic/design and custom designed control software.
- an analysis of the characteristics of the wideband CDMA channel, including multipath delay profile, path loss, time variation of delay profiles, and fading correlation between antenna elements at the basestation.
- the feasibility study of a beamforming receiver using the channel parameters calculated from real data.

1.3 Thesis outline

The thesis can be divided into three major topics: design, testing and implementation of our wideband CDMA measurement system (Chapters 3 and 4), wideband CDMA channel propagation characteristics (Chapter 5), and a feasibility study of a smart antenna receiver employing adaptive digital beamforming (Chapter 6).

In Chapter 3, we provide a detailed description of the features and capabilities of our testbed. The following design issues, based on our low-cost modular objective, are discussed: direct conversion using quadrature modulator, link budget analysis, direct bandpass sampling, PN sequence selection, system integration and data collection.

Chapter 4 discuss the results from system validation, received complex signal conversion and demodulation algorithms. System validation is divided into three categories: functional block testing, system level analysis and synchronization of the antenna array. Received signal processing algorithms presented are real-to-complex received signal conversion, initial synchronization with matched filtering, and PN code tracking.

Chapter 5 first describes the outdoor wideband CDMA propagation experiments conducted using a two element antenna array and three mobile transmitters operating at $1.9GHz$. Then, we present an algorithm for the detection of multipath and its delay in a delay profile. Using the multipath signal detection algorithm and measured data, we calculate path loss and multipath delay statistics in an outdoor environment for wideband CDMA.

In Chapter 6, the correlation between antenna elements is first investigated. We then study the array manifold characteristics of an outdoor environment using our 4-element antenna array basestation and four transmitters. The stability of spatial characteristics is investigated by generating beampatterns derived from experimental trials on short blocks of data. Also, a method of beamforming weight estimation from received signal sample covariance matrices is introduced, followed by the calculation of SINR improvement from beamforming over a single antenna receiver.

Chapter 2

Background

2.1 Third Generation Mobile Communications

The demands for cellular mobile communications today are increasing at an alarming rate. The number of subscribers are ever growing and while traditional subscribers use cellular phones for voice communication, *dot com* phones are already in the market which support text-only internet and email services. Since internet services are mostly data communications, the third generation (3G) mobile communication systems will require to support high data rate communications for multimedia applications as well as voice services. Two 3G standardization bodies, 3GPP and 3GPP2 from Development Group (CDG) have proposed specifications and air-interfaces for next generation mobile system defined by ITU (International Telecommunications Union). The 3GPP2 proposed cdma2000 (also known as Wideband cdmaOne) and 3GPP's version is called W-CDMA [21] [45] [67].

W-CDMA is designed to be adopted by GSM (Global System for Mobile Communications) and TDMA (Time-division multiple access) systems and cdma2000 is evolved from IS-95 CDMA systems. There are several differences between the two standards. First, the chip rates of W-CDMA are multiples of 4.096 Mcps, i.e. $n \times 4.096$

Mcps (M chip/s) with $n = 1, 2$ and 4 , whereas cdma2000's chip rates are multiples of IS-95's 1.2288 Mcps at $n \times 1.2288$ Mcps with $n = 1, 3, 6, 9$ and 12 . Secondly, the downlink RF channel for W-CDMA is direct spread, whereas cdma2000 supports multicarrier channels [45].

Both of these proposed 3G systems promise to achieve data rates of 2 Mbps (M bits/s) by 2002 in Europe and 2003 in U.S. according to [26] when the systems are fully implemented. In order to achieve this goal, advanced technology such as smart antennas are proposed [6] [10] [21] to enhance system performance by exploring the spatial diversity gain in addition to time diversity which is currently a topic of active research. Moreover, advancements in hardware design and manufacturing provide linear and efficient power amplifiers [44], high speed digital-to-analog (A/D) converters and all-in-one RF transceiver ASICs (application specific integrated chips). With these new hardware technologies, digital (or software) radio will be an important enabling technology for 3G systems [6] [40]. As standards evolve and services are added or improved, digital radio basestations or mobile units can be upgraded through software re-programming [40]. This will reduce cost and time-to-market for the deployment of new services and features [6]. In Chapters 3 and 4, we will discuss our wideband CDMA smart antenna testbed and its digital receiver signal processing.

2.2 Antenna Array and Beamforming

Smart (or adaptive) antenna technology is based on digital signal processors in a software radio. The signal processors control an antenna array, thereby, directing (or receiving) radio signal precisely at (from) individual users [26]. As the users move around, the smart antennas will track them accordingly. Currently, antenna arrays found in cellular basestations are used to broadcast omnidirectionally [26] as illustrated in Figure 2.1. In this thesis, we focus on the application of adaptive antennas

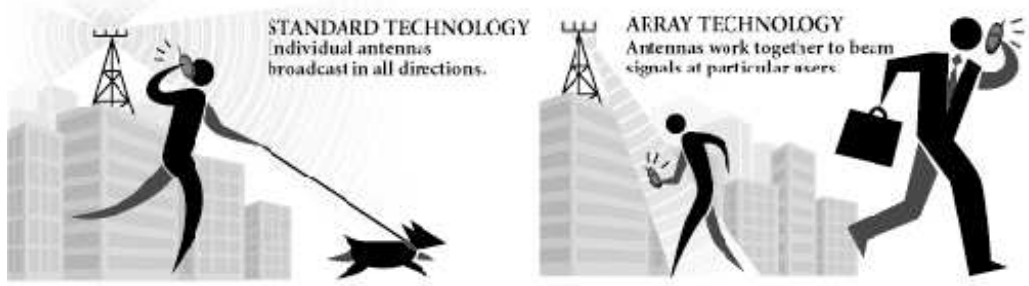


Figure 2.1: Smart Antennas vs. Standard Technology [26]

to the reception of mobile users at the basestation.

In theory, adaptive antenna array of a cellular communication system combines and weights signals received by the multiple antennas such that the signal-to-interference-noise ratio (SINR) is maximized [20] [72]. The process of weighing and combining signals is usually called digital beamforming [68]. System capacity is increased from beamforming as a result of reducing the amount of interference [42].

One crucial required component for digital beamforming is the knowledge of the array response vector which contains the relative phases of received signals at each array element [74]. With this knowledge, the receiver can use the array response vector to isolate the desired signal and suppress interferers. The array response vector depends on the array geometry and for the rest of the discussion relevant to this thesis, we focus on a uniform linear array (ULA). An ULA has identical antennas aligned in the straight line as shown in Figure 2.2. The elements are equally spaced by a distance of d and θ is the angle between the arriving signal and the normal to the array. In this case, the array response vector, $\mathbf{a}(\theta)$, for a M-element ULA is given by [2] [43] [68]:

$$\mathbf{a}(\theta) = \left[1 \quad e^{-j2\pi d \sin \theta / \lambda} \quad \dots \quad e^{-j2\pi(k-1)d \sin \theta / \lambda} \right]^T \quad (2.1)$$

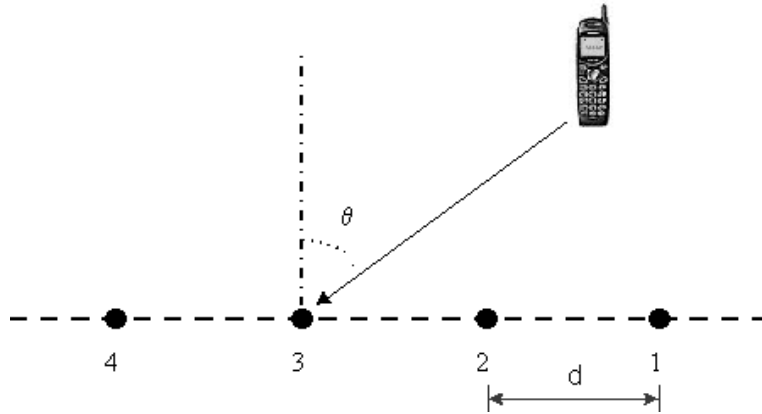


Figure 2.2: Uniform Linear Array

where $\lambda = c/f_c$ is the wavelength of the carrier frequency f_c , c is the speed of light and $k = 1, 2, \dots, M$.

The validity of Equation (2.1) is based on the narrowband data model for array signal processing on received signal vectors [43] [46]. This model assumes that as the signal wavefront propagates across the array, the envelope of the signal remains essentially constant. The term *narrowband* is used here since the assumption of a slowly varying signal envelope is most often satisfied when the signals have a bandwidth that is small relative to the carrier frequency f_c [43] [46]. More importantly, this assumption is also valid for wideband signals, as in the case of CDMA [43], since the carrier frequency of digital cellular system is usually around 1.8 – 2.0 GHz. The topics of antenna arrays and beamforming are presented in more detail in Chapter 6.

2.3 Mobile Radio Propagation Environment

The channel model for typical mobile radio communications in an urban area includes a basestation antenna (or antenna array), a line-of-sight (LOS) propagation path and

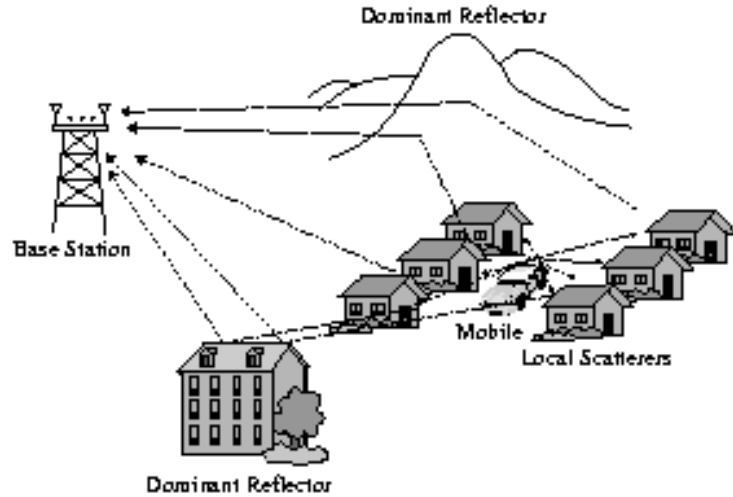


Figure 2.3: Example of the Wireless Propagation Environment [43]

a number of non-LOS reflected propagation paths and the antenna on the mobile [43] [56]. The propagation paths are dependent on natural and man-made structures such as hills, buildings and moving objects located between or surrounding the mobile and the basestation [43] [71]. An example of the environment described above is shown in Figure 2.3. Due to reflections from various objects, the electromagnetic (signal) waves travel along different paths with different lengths and combine at the receiver. These multipath signals can add or subtract vectorially depending on their relative phases and may result in fading [56] [71]. In a multiple access system, each user's signal arrives at the basestation with multiple distinct paths causing fading attenuation [72].

Even the smallest and slowest movement can cause time-variant multipath, resulting in a time-varying signal at the basestation [43] [46]. Thus, a RAKE receiver is usually employed in CDMA systems to combat multipath fading [48] [56]. A RAKE receiver consists of a bank of correlators to combine information obtained from several resolvable multipath components. Each of the correlator is positioned to correlate starting at the delay experienced by a particular multipath component of the desired

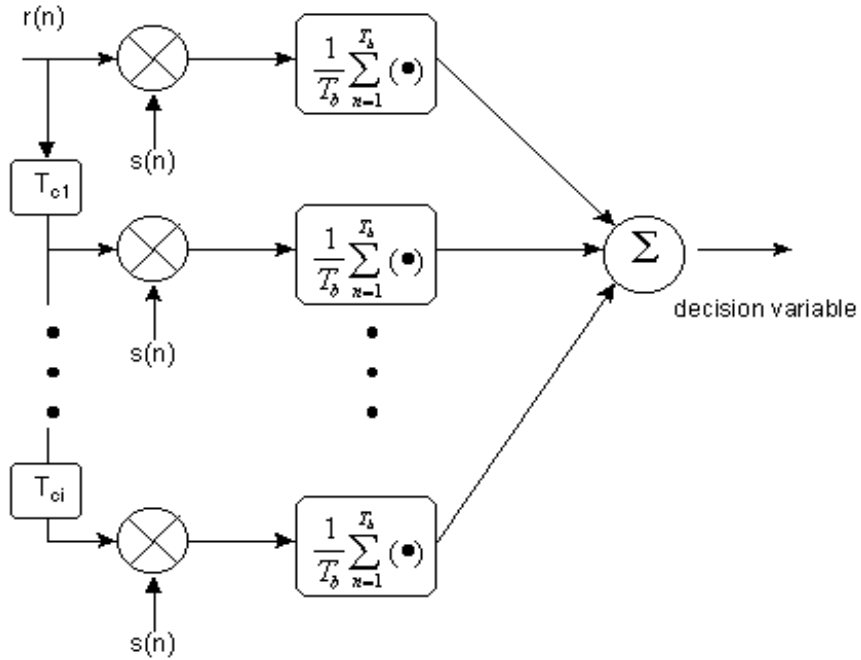


Figure 2.4: RAKE Receiver Block Diagram

signal. A block diagram of a RAKE receiver [48] [53] is shown in Figure 2.4, where T_{c_i} is the RAKE receiver tap in integer multiples of T_c (chip rate), T_b is the bit period and $s(n)$ is the modulating waveform. The value and number of RAKE taps used are dependent on the actual delays of the multipaths relative to the first arrived signal component. The multipath delays can be estimated from received signal covariance matrices as discussed in [43] and from the knowledge of the multipath delay spread of a particular environment from experimental results. In Chapter 5, we examine the wideband radio propagation and multipath characteristics of an outdoor environment using measurement data gathered from our testbed.

Chapter 3

Wideband CDMA Experimental System Design

3.1 Introduction

In the previous chapters, we presented the motivation for developing a measurement system. In this chapter, we provide a detailed description, features, capabilities and design issues of our wideband CDMA experimental system. The initial design and feasibility study began in the summer of 1998. A literature search was taken place to review the current state of art and hardware components availability for wideband radio design. After several revisions in the transmitter and receiver hardware design, we arrived at the current design which is described in the next few sections.

Several mobile communication measurement systems were designed and experiments were performed for the $910MHz$ band in the last two decades including [5] [8] [9] [16]. They were successful in characterizing the channel for the first and early part of the second generation mobile phone systems. Radiation and propagation characteristics are quite different in the PCS band. Thus, in recent years research efforts have switched to the 1.8-2.0 GHz band. With the renewed interests in spatial diversity gain in addition to temporal processing (RAKE receiver), recent measurement

systems were created with the objective to either measure or estimate direction of arrivals (DOA), multipath delays and study the feasibility of beamforming in second and third generation mobile communication systems.

Several research groups have designed and conducted field testing in the 1.8 – 2.0GHz frequency band [7] [41] [46] [58] [69]. As each group’s hardware, setup and objectives differ, their systems’ capabilities and limitations also vary. In particular, none has employed wideband CDMA with multiple transmitters. In order to identify the novel aspects of our system, it is important to review previous work in the open literature relevant to our objectives.

3.2 Previous System Designs and Specifications

Muhammed and Rappaport used a 6-element array for comparing various DOA estimation algorithms. The transmitted signals were CW tones produced from signal generators. The received tones are sampled at $6ksp/s$ (kilo samples per second) baseband by DSP processors [41]. Okamoto and Xu tested spatial signature variation, angle spread, beamforming by sending coded voice data over an RF channel at 1.8 GHz. Their system has either 4 or 8 antenna elements and a sampling rate of $3.072Msp/s$ [46]. The Communications Research Laboratory (CRL) of McMaster University built a smart antenna system equipped with one mobile transmitter and an 8-element circular array for studying DOA and multipath profiles. Their system has a $5MHz$ bandwidth and an A/D converter sampling at $10Msp/s$ [31]

On a larger scale, Savahashi and Adachi of NTT Mobile Communications network Inc. in Japan [58], and Wilson et al. of Institute for Telecommunication Sciences

(ITS) in Colorado [7], performed experiments on wideband CDMA systems. Sava-hashi and Adachi's testbed includes a 4-element antenna, 4-finger Rake combining, a fixed PN chip rate (15Mcps) and spreading code (256-chip orthogonal Gold sequence), and a sampling rate of 35Msps using an 8-bit A/D converter [58]. While NTT studied BER performance and power delay profile in Rayleigh fading, ITS evaluated received signal strength and diversity combining algorithms, e.g. Selection, MRC (maximum ratio combining) and ORC (optimum ratio combining). Maximal length code sequence at 10 Mcps over PCS band was transmitted, and a received signal was digitized at an IF frequency with a sampling rate of 40 Msps [7].

3.3 System Description

Our measurement system is a prototype of wideband CDMA mobile system which offers a high chip rate and data storage for off-line signal processing applicable mainly to the uplink of a cellular communication system. It consists of four programmable portable transmitters (TX) and one 4-element antenna array basestation. The transmitter directly converts baseband PN sequences to RF, which is based on the traditional design of DS (direct sequence) spread spectrum transmitters [14]. Each array element is a one-stage downconversion IF-sampling digital receiver proposed for 3G mobile systems [6] [64]. We employ a low-cost modular approach in hardware system design. Off-the-shelf components such as evaluation boards and components fitted with connectors are chosen whenever applicable and possible to reduce cost as well as development time [66].

The design and capabilities of the transmitter and receiver are discussed separately at first, followed by issues related to smart antenna system parameters and setup. The section concludes with a discussion on data collection.

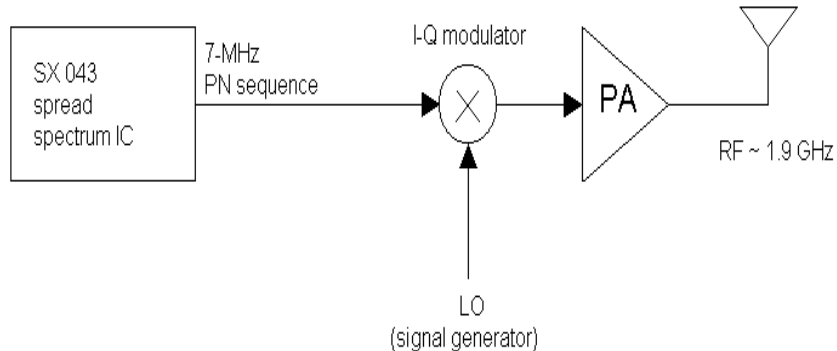


Figure 3.1: Direct sequence RF transmitter design

3.3.1 Transmitter Design and Features

The transmitter is composed of a spread spectrum ASIC known as the SX043 produced by AMI [1], a direct RF modulator, a power amplifier (PA) and an antenna. First, the SX043 generates a PN sequence at a user-specified rate. Then the PN sequence is upconverted directly by an I-Q modulator to RF frequency centered at $1.9GHz$. The resulting RF signal is transmitted after power amplification (see Figure 3.1).

The model for the transmitted signal is defined as follows:

$$y(t) = [\cos 2\pi f_c t + \sin 2\pi f_c t]x(t) \quad (3.1)$$

$$= e^{-j2\pi f_c t}x(t) \quad (3.2)$$

where $x(t) = c(t)d(t)$, $c(t)$ is the baseband PN sequence, $d(t)$ is the data bit sequence and f_c is the carrier frequency (LO). Equation (3.1) implies that $x(t)$, the baseband spreaded signal, is applied to both the I and Q inputs of the quadrature (or I-Q) modulator since I-channel is equal to $x(t)\cos(2\pi f_c t)$ and Q-channel is $x(t)\sin(2\pi f_c t)$. This is equivalent to multiplying the signal by a complex exponential function of frequency f_c as in (3.2). The I-Q modulator is chosen because we are unable to find

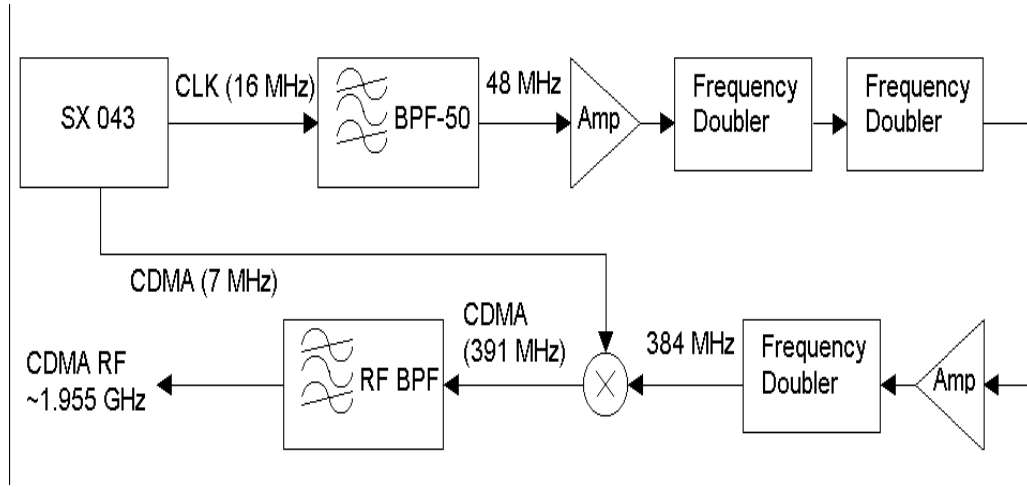


Figure 3.2: Previous transmitter link design

off-the-shelf mixers with connectorized input at our operating frequency range, plus the current I-Q modulator design can be easily modified for QPSK data modulation. In our design $x(t)$ is not data modulated. That is, $d(t)$ is assumed to be '1' for all time. This simplification can be justified since our study of channel propagation characteristics and beamforming in a multiple-user environment does not require actual data bits to be transmitted.

The current transmitter design is arrived after analysis and testing of the previous design based on [4] [65]. The fundamental difference between the previous design and the current version is the generation of LO (local oscillator) signal. The previous design aims to generate its own LO signal from a reference signal originated from the SX043 board (see Figure 3.2). The 3rd harmonic of the 16MHz reference is first extracted by a bandpass filter (BPF), then amplified and frequency doubled three times (a total of 8 times) before it is mixed with the 7Mcps PN sequence. The final step is to isolate the 5th harmonic at mixer's output to obtain a RF bandpass signal centered at 1.955GHz.

In order for the design in Figure 3.2 to function properly, the unwanted or surprising harmonics of the LO signal at $384MHz$ must be kept at minimal. From our experiments, the combination of amplifier, frequency doubler and BPF is unable to produce a clean LO signal with little surprising harmonics. The choice of packaged BPFs is also limited, plus the high input power requirement (+10dBm minimum) of the frequency doubler (with a low power $16MHz$ as input) make it impossible to suppress the LO's harmonics. In addition, if we use a LO which contains harmonics and mix it with the $7MHz$ PN sequence, unwanted signals can be modulated into the desired band. Therefore we decide to modify our design to the direct upconversion method as illustrated in Figure 3.1.

Direct RF modulation is the upconversion of the baseband I-Q channel directly into RF frequency. This technique has been used in wireless systems such as PCS, GSM and CDMA, and will become more popular in 3G systems as hardware advances [13]. The benefits of this method are elimination of the entire IF stage, flexibility in RF frequency and also simplicity of the design. The disadvantage from the standpoint of our transmitter is reduced portability since a signal generator is currently employed for LO generation.

The main features of the transmitter are the programmable chip rate, PN sequence and the operational frequency range. The SX043 spread spectrum IC is capable of chipping rates up to $64Mcps$, and is capable of producing three types of PN sequences; namely Gold, maximal length, and Barker codes [1]. Square pulses are used by the SX043 to maintain our low cost objective since pulse shaping operations at high chip rates require intensive DSP processing power. The I-Q modulator is an integrated circuit (IC) chip allowing direct modulation of frequency carriers from $800MHz$ to $2.5GHz$. In addition, an omnidirectional dipole antenna is used for both transmitting and receiving.

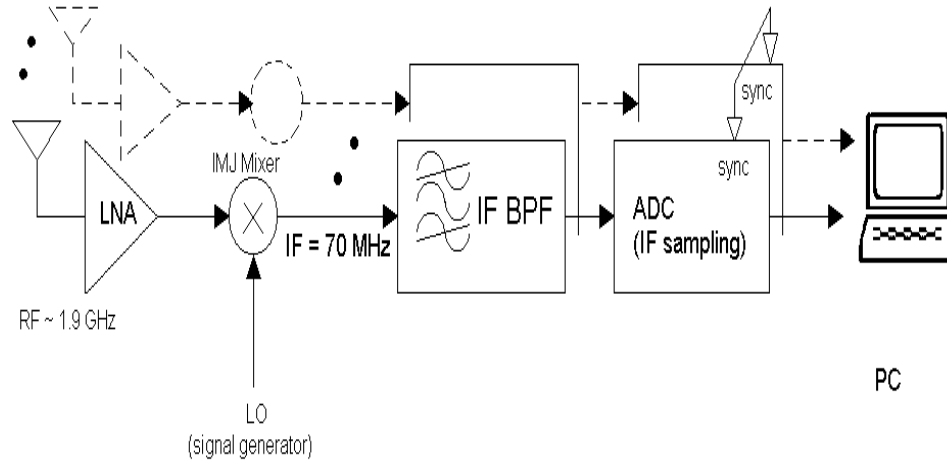


Figure 3.3: Digital wideband receiver design

3.3.2 Receiver Design and Features

The basic structure of the receiver is a one-stage downconversion superheterodyne scheme. Each antenna element of the basestation consists of a RF low noise amplifier (LNA), a mixer, an IF (intermediate frequency) bandpass filter, and a digital receiver with a fast A/D (analog-to-digital) converter. The incoming RF signal, received separately by each antenna, passes through a one-stage downconversion from RF to IF handled by the image-rejection (IMJ) mixer, and is then bandpass filtered before analog-to-digital conversion. For every snapshot of the incoming signal, the raw data are first stored in onboard data FIFO and then downloaded to the hard-disk through a parallel port on a PC (see Figure 3.3). The spacing between each antenna is $\lambda/2 \simeq 7.9cm$ for carrier frequency, f_c , at $1.9GHz$.

In a traditional superheterodyne receiver, the IMJ mixer is normally implemented as a RF image rejection pre-mixer filter followed by a mixer [33]. The centre frequency of the RF filter is f_c and its bandwidth is equal to the channel spacing. As a result, the carrier frequency of this particular receiver design is fixed by the RF filter. Switching to a different carrier frequency usually requires replacing the RF filter

with a different f_c which is expensive due to its high-Q factor. The image rejection mixer in our system is operational in the 1-2 GHz band, and thus provides additional flexibility for the testing of different carrier frequencies.

The received signal of one receiver link at IF before digitization can be modeled as:

$$r(t) = y(t - \tau) \cos 2\pi(f_c - f_{IF})t \quad (3.3)$$

$$= [\cos 2\pi f_c(t - \tau) + \sin 2\pi f_c(t - \tau)]x(t - \tau) \cos 2\pi(f_c - f_{IF})t \quad (3.4)$$

$$\stackrel{BPF}{\simeq} \frac{1}{2}[\cos 2\pi f_{IF}(t - \tau) + \sin 2\pi f_{IF}(t - \tau)]x(t - \tau) \quad (3.5)$$

where f_{IF} is the intermediate frequency, τ is a random time delay and $y(t)$ is given in (3.1). The unwanted upper sideband, $2f_c - f_{IF}$, in (3.4) results from downconversion and is bandpass filtered. This operation is expressed by the notation ‘BPF’ in (3.5). The passband of the filter is 63 to 77 MHz. For simplicity, the background noise term and other channel effects in the above equations are omitted without any loss of generality for the purpose of providing a model for the receiving hardware structure.

The A/D converter is capable of IF wideband digitization at a speed of up to 64 Msps. In all our experiments, the converter is configured to sample at 35 Msps. Since the IF frequency is 70 MHz and by sampling at 35 Msps, the 7 MHz bandpass signal is directly downconverted from IF to baseband. A digital lowpass filter is required to select the baseband image of the original bandpass signal and reject other replicas at 35 and 70 MHz (see Figure 3.4). This process is usually called direct bandpass sampling [70]. By avoiding analog and/or digital demodulation, direct bandpass sampling reduces the complexity of the receiver and off-line data processing. Direct bandpass sampling can be successfully employed since the IF carrier is only one order of magnitude larger than the signal bandwidth.

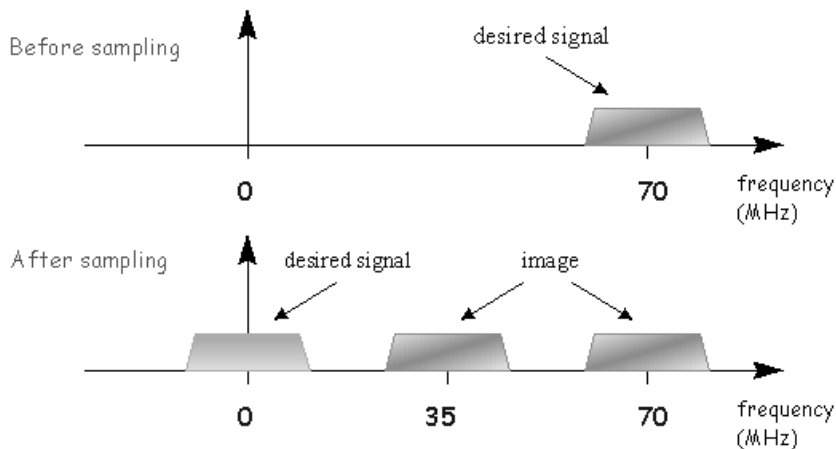


Figure 3.4: Direct bandpass sampling

3.3.3 System Parameters, Setup and Integration

After reviewing the features of the transmitter and receiver separately, we discuss some issues concerning the hardware's setup and integration of individual components, choice of PN chip rate, PN sequences and sampling rate for the whole system.

Third generation (3G) mobile systems offer high data rate multimedia services to accommodate mobile users who want to use wireless access for multimedia and internet applications. In order to meet the above requirements regional standards committees put forward proposals for wideband CDMA parameters for IMT-2000 [45]. The proposed nominal chip rates for wideband CDMA are 4.096, 8.192 and 16.384 *Mchips/s*, Gold sequences are chosen for user separation with spreading factors from 4 to 256 [45].

When the design process of our wideband system commenced two years ago, most of the 3G mobile hardware were not available commercially. After thorough search for appropriate components that fit our system's parameters, cost and availability, our

system's primary limitation is maximum channel bandwidth. The maximum bandwidth of our system is 15 MHz (double-sideband) limited by the A/D converter's wideband sampling bandwidth [12]. As a result the maximum chip rate achievable is 7 *Mcps* ($7 \times 2 = 14\text{MHz}$) with a design margin of 1*MHz*.

In contrast, our programmable transmitter provides various choices of PN sequence and length (5 to 4096 chips). Gold sequences are proposed for the 3G mobile system primarily because they provide a large number of codes for a given length, and their cross-correlations are uniform and guarantee bounded [14]. This property is attractive for code-division-multiplexed signals [14] where the number of users is large. Maximal length sequences are also an appropriate choice as it has been used extensively in communications [14]. Since the maximum number of users in our measurement system is four, it is a simpler task to select four sets of maximal length sequences of the same length which exhibit minimal cross-correlation. In fact, the designer of the SX043 had already performed this task and listed the choices in the user manual [1]. Also, the generation of PN sequences at the receiver for despreading is simple for maximal length sequences (see Section 4.5). Based on the above reasons, maximal length sequences are chosen for our testing purposes.

In our study of multipath propagation characteristics using the sliding correlator method (see Section 5.4), the length of the PN sequence, or symbol period, must be at least the same as the expected delay spread to avoid interference. The duration for a 31 and 127-chip 7*Mcps* sequence are roughly 4.4 μs and 18 μs , respectively. The channel sounding experiment in [16] obtain delay spread of 0.14 μs – 0.35 μs and maximum delay spread of 0.73 μs – 1.86 μs . We conclude that the time window of either the 31 or 127-chip sequence per symbol is sufficient to capture most of the significant multipath components. In addition, it is generally easier to apply multi-user detection (MUD) to a system with short codes because cross-correlation does not vary with every symbol, which is not the case with long spreading codes [45]. Therefore the

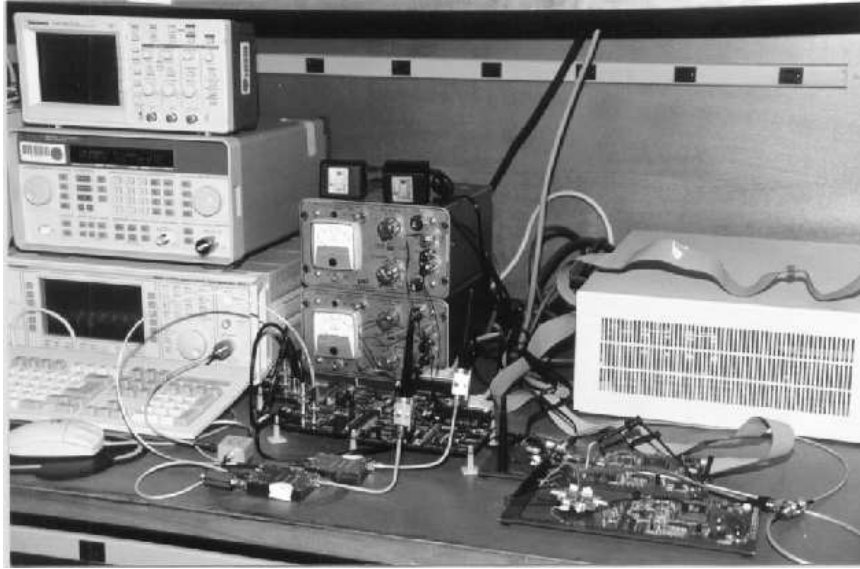


Figure 3.5: Photograph of TX and RX hardware and other testing equipment

data gathered in our experiments is also suitable for studying MUD in a wideband CDMA system.

The experiments performed using this system are conducted outdoors, so the transmitters and basestation must be designed and packaged for easy transport and outdoor use. Photographs of the measurement system are shown in Figures 3.5, 3.6, 3.7 and 3.8.

Once each transmitter is programmed through a PC port, then it can run on battery power without connection to PC. The only physical connection remaining is a 75-foot cable supplying a LO signal for upconversion. Using the long cable the transmitters can be moved around for testing at different locations. The SX043 evaluation board and the I-Q modulator draw power from four AA alkaline cells, and the power amplifier has its own power source provided by a 12V rechargeable lead-acid battery.

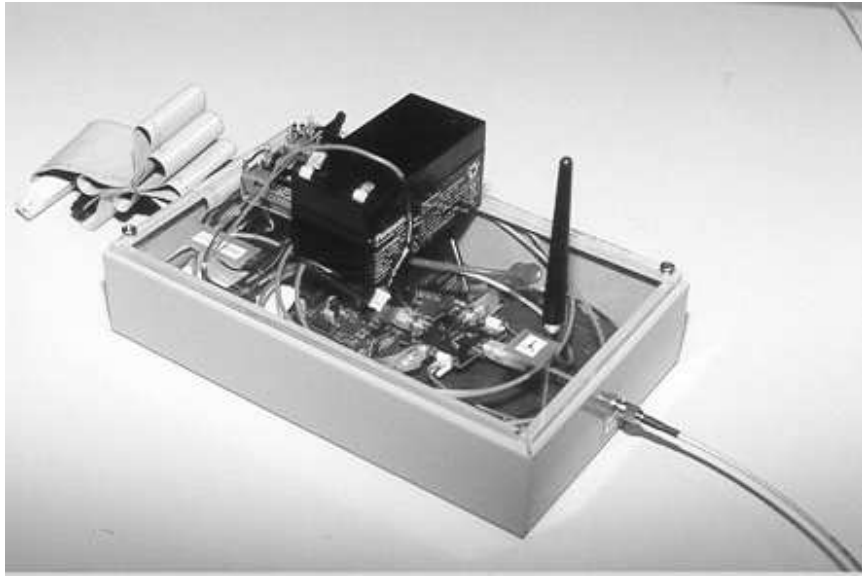


Figure 3.6: Photograph of a portable transmitter

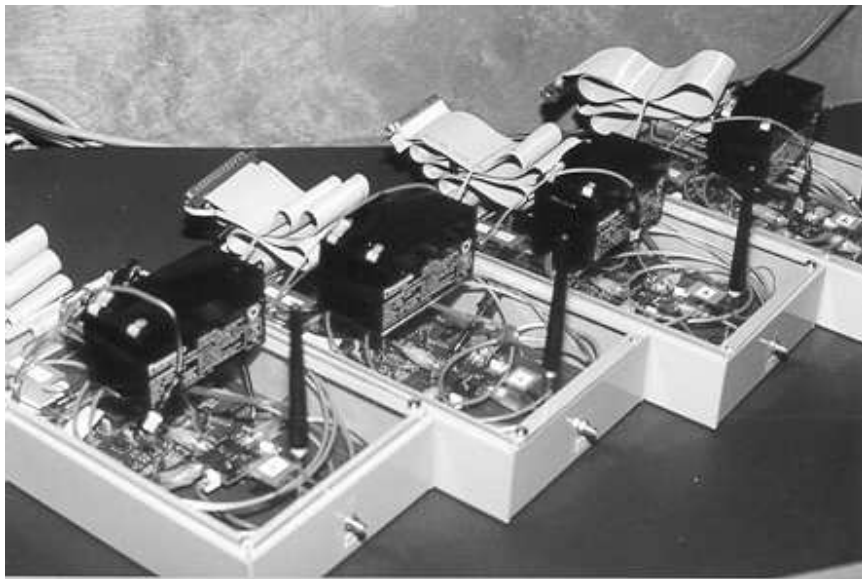


Figure 3.7: Photograph of all four portable transmitters

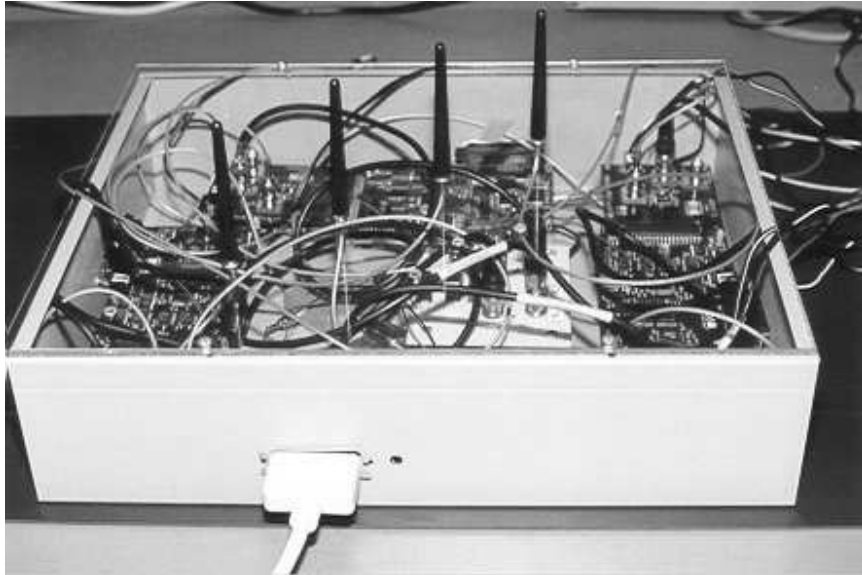


Figure 3.8: Photograph of receiving basestation

The basestation is connected to a PC's parallel port at all times for continuous download of sampled data collected in the onboard FIFO. The details on data collection will be discussed in the next section. There is no need to design portable power for the rest of the receiving hardware. The basestation, signal generators, power supplies and PCs are stationed in a fixed, central location during experiments and transmitters are moved around to create different scenarios.

3.3.4 Data Collection

Utilizing the receiver setup mentioned previously, synchronized snapshots of transmitted signals can be obtained. Synchronous sampling is achieved by clocking the A/D converters externally and loading data in the onboard FIFOs synchronously. The 'sync' line in Figure 3.3 is driven by the master board which controls the timing of the FIFO's data loading. Synchronization issues are discussed in more detail later in Section 4.4.

For data collection, the A/D converter evaluation board module's FIFO size limits the maximum number samples to 16K per snapshot of incoming signal. A custom designed software program, *ADport*, is written to generate the "sync" signal for sampling and downloading data onto the PC. *ADport* has the option to save data in batches of 1K, 2K, 4K, 8K or 16K bytes in ASCII format. All the received data stored on the PC are then readily available for post-processing. See Appendix A for details on our software design.

Since the chip rate and sampling rate chosen are $7Mcps$ and $35Msps$, respectively, the oversampling increases the sampling resolution to 5 samples per chip. With 5 samples per chip and maximum FIFO size of 16k samples, the maximum number of data bits per snapshot is $16 \times 1024 / (31 \times 5) = 105bits$ for 31-chip or $25bits$ for 127-chip sequences, corresponding to a time window of $0.465ms$.

Oversampling provides several benefits in system design. It minimizes the requirements of the anti-aliasing filter by shifting the spectral copies of the sampled signal further apart. In addition, oversampling leads to an additional SNR gain because the distribution of noise is spread wider in bandwidth as the sampling rate increases [12] [70]. In multipath delay analysis, more importantly, the added resolution allow us to determine delay parameters with fractional-chip accuracy.

3.4 Summary

In this chapter, the design and features of our wideband CDMA smart antenna measurement system is presented.

The key features of the system can be summarized in the following:

- multiple portable transmitters (4)
- basestation receiver with 4-element antenna array
- high chip rate of 7 Mcps
- fine sampling resolution of 5 samples per chip

With the capability and features of our system, we can gather data in a wideband CDMA multiple user environment. The transmitters placed at different locations create different scenarios, for example, single or multiple users, equally spaced or closely spaced users, and near-far users. Then beamforming, multipath or spatial-temporal analysis can be performed on the data samples.

The following design issues have been discussed: direct conversion using quadrature modulator, image rejection mixer, direct bandpass sampling, PN sequence selection, system integration and data collection. In short, an effort is made to develop a system which satisfies or resembles third generation wideband CDMA specifications, wideband digital radio design, and flexibility in hardware for future development while maintaining our low-cost modular objective.

Chapter 4

System Validation and Received Signal Processing

4.1 Introduction

In the previous chapter the design, features and capabilities of our testbed are presented. In this chapter, we discuss the results from system validation and received signal processing algorithms. System validation is divided into three categories; functional block testing, system level analysis and synchronization of the antenna array. Digital signal processing at the basestation is performed off-line on Sun workstations. The three main signal processing procedures used are real-to-complex conversion of received signal, reference PN sequence generation and PN sequence synchronization. These procedures are required for further experimental data analysis and are presented in this chapter.

Each block in the transmitter (TX) and receiver (RX) chain is experimentally verified against hardware specifications. Performance of the TX and RX links are then tested and compared with design estimates. The details of block and system performance is described in Sections 5.2 and 5.3, respectively. Synchronization issues such as carrier phase and sampling clock synchronization, and synchronous sampling

are described in Section 5.4. The last section presents algorithms for off-line digital signal processing.

4.2 Functional Block Testing

This section discusses the testing performed on hardware components in the transmitter and receiver link. The aim is to ensure they are functioning properly as described in the datasheets provided by the manufacturers. First we begin with the transmitter link test and then follow with the receiver link. Some of the functional characteristics we have tested are power gain/loss, chip rate, PN sequence spectrum, receiver gain and received signal samples.

4.2.1 Transmitter Link Testing

The transmitter link consists of three main blocks (Figure 4.1): SX043 spread spectrum integrated circuit (IC) for pseudo-noise (PN) sequence generation, in-phase and quadrature (I-Q) modulator and power amplifier (PA).

The SX043 is programmed through the PC interface to produce PN waveforms at 7.11Mcps baseband. The chip rate is not a round number because the SX043 IC's chip rate is set by programming an integer divisor for its 64MHz internal clock. Therefore the closest chip rate to 7Mcps is $64/9 \simeq 7.11$. One of the pre-programmed PN sequences is captured on the oscilloscope screen and shown in Figure 4.2. From the plot the peak-to-peak voltage of the waveforms is roughly 5 V p-p as this is the power supply voltage (V_{DD}) of SX043 with $\pm 5\%$ voltage regulation. The ripples exhibit at both positive and negative peaks are due to hardware limitations and are expected.

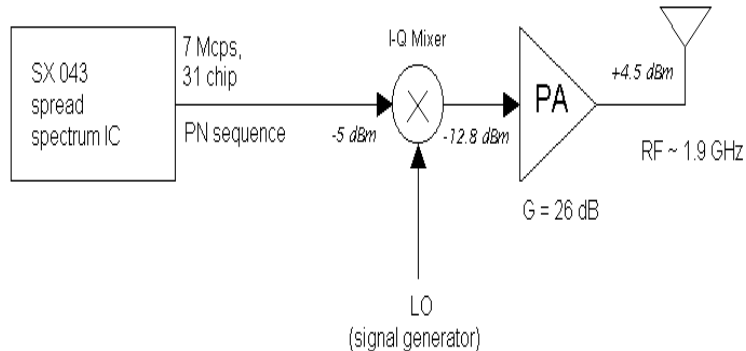


Figure 4.1: Direct sequence RF transmitter block diagram

Similar waveforms are previously reported in [14]. A 31-chip maximal length PN sequence is chosen for our test and the capture in Figure 4.2 shows two distinct periods of the sequence. Since the time scale is $1 \mu s/div$, the period is approximately 4.4 units or $4.4 \mu s$. Thus, it matches expected period of $4.36 \mu s$ ($31 \div 7.11 \times 10^6 s$).

The baseband spectrum of the same maximal length PN sequence [75] is obtained from a spectrum analyzer is shown in Figure 4.3. The spectrum is centered at 0 MHz and the first null which represents the chip rate is approximately at 7.11 Mcps.

After I-Q modulation, the spread spectrum signal is now centered at 1.9 GHz as in Figure 4.4. The power of the main peak is about -12.8 dBm. Compared to the power of -5 dBm at baseband (Figure 4.3), the modulator plus the input voltage matching circuits exhibit an insertion loss of 7.8 dBm. By inspection of Figure 4.3, the power of the side lobes is proportional to a function of $\text{sinc}^2 f$ as expected from [75]. This is a property of maximal length sequence's power spectrum as described in [75].

Since the signal power at the modulator output is not high enough to provide

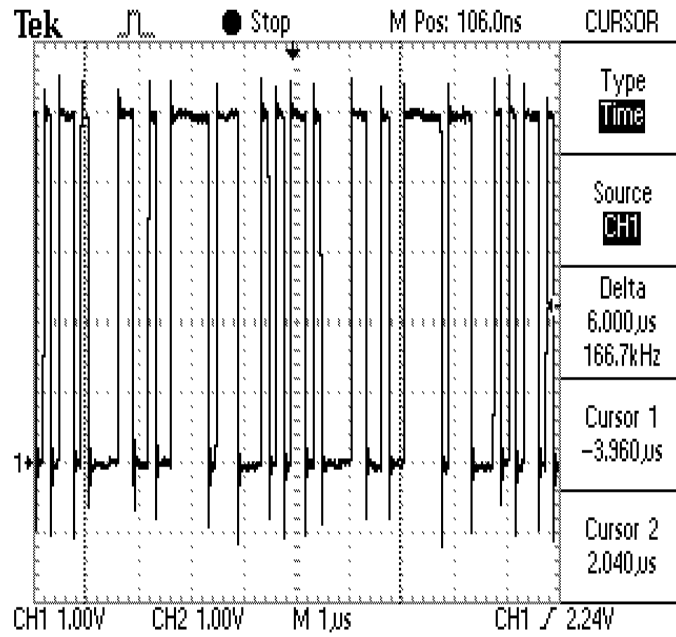


Figure 4.2: Baseband PN chip waveforms

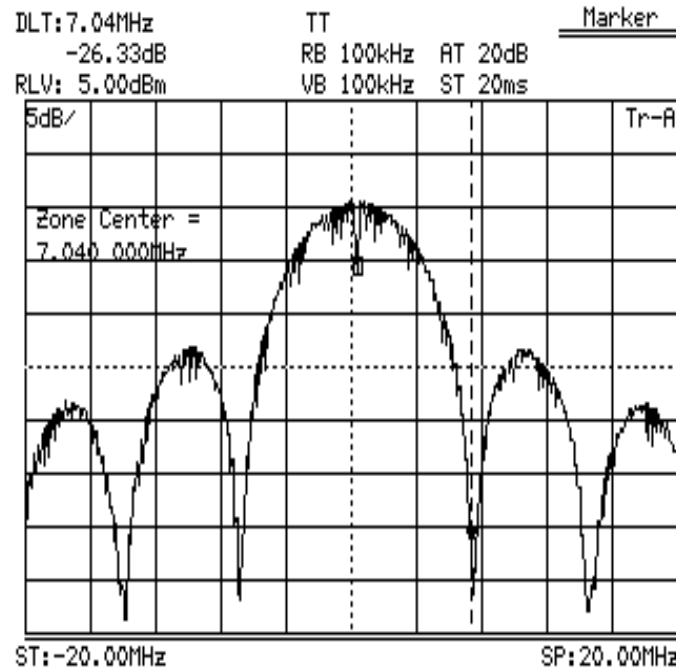


Figure 4.3: Baseband PN sequence power spectrum

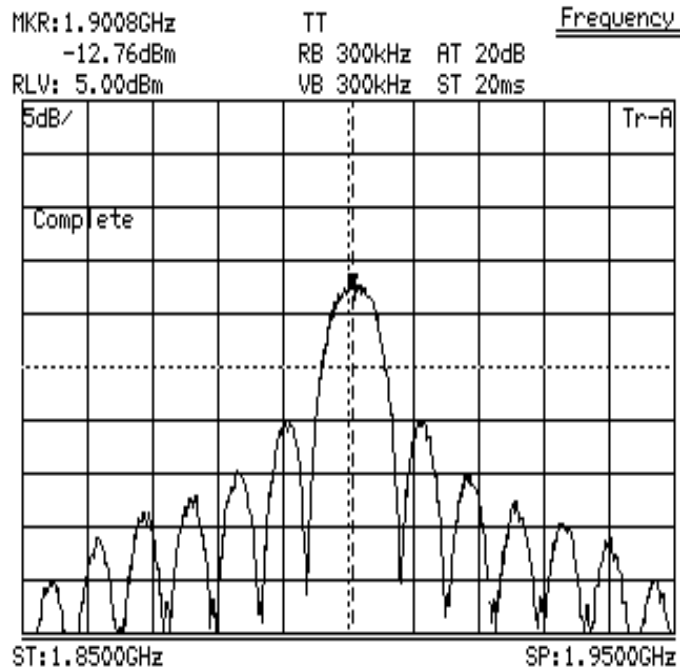


Figure 4.4: RF carrier modulated PN sequence power spectrum

adequate range for the TX-RX system, a power amplifier is inserted before the transmitting antenna. At the output at the power amplifier, the peak power is increased to +4.5 dBm (see Figure 4.5) for a gain of 17.3 dB. The gain listed for the power amplifier in the datasheet is 26 dB [38]. Further testing indicates that the gain is closer to the specified gain of 26 dB under the single tone test. The single tone test refers to gain measurements obtained by using a sine wave as input, as opposed to a wideband signal.

4.2.2 Receiver Link Testing

The receiver link consists of four main blocks: low-noise amplifier (LNA), image rejection (IMJ) mixer, intermediate frequency (IF) bandpass filter and analog-to-digital (A/D) converter evaluation board modules (Figure 4.6).

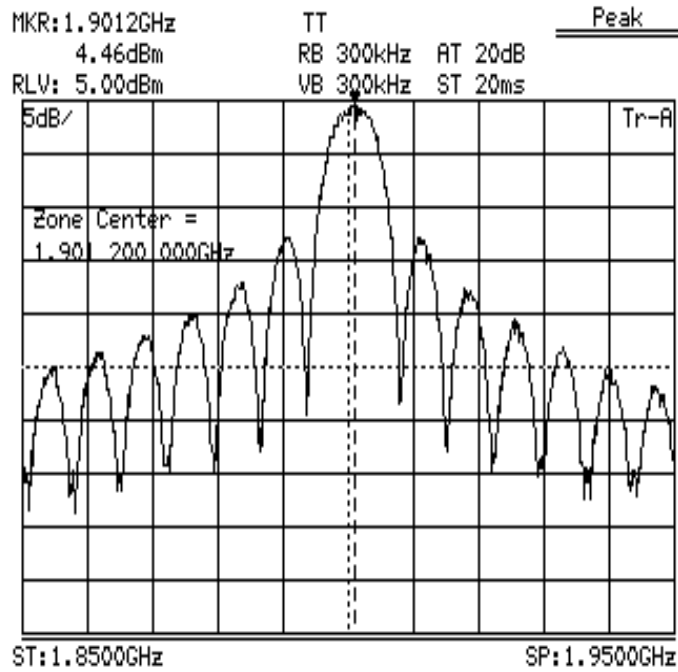


Figure 4.5: RF spectrum at power amplifier output

Signals are received by an omnidirectional dipole antenna at each array element. An RF test signal is placed at the input of the low-noise amplifier (LNA). The input main peak power is -17.2 dBm and is shown in Figure 4.7. At the output of LNA, the PN sequence’s power is amplified to +2.8 dBm (see Figure 4.8). The gain of the LNA is then approximately 20 dB and it matches the specified gain in the datasheet of 20 dB minimum [38].

The next functional block in the receiver chain is the IMJ mixer. One of the mixer’s specifications which is related to the system’s link budget is the conversion (insertion) loss which is listed at 8 dB (loss) [39]. The main peak power at IF output of the mixer in Figure 4.9 is -12.1 dBm, a loss of 14.9 dB (2.8-(-12.1) dBm). However, under the single tone test described previously, a conversion loss of only 9 dB is found.

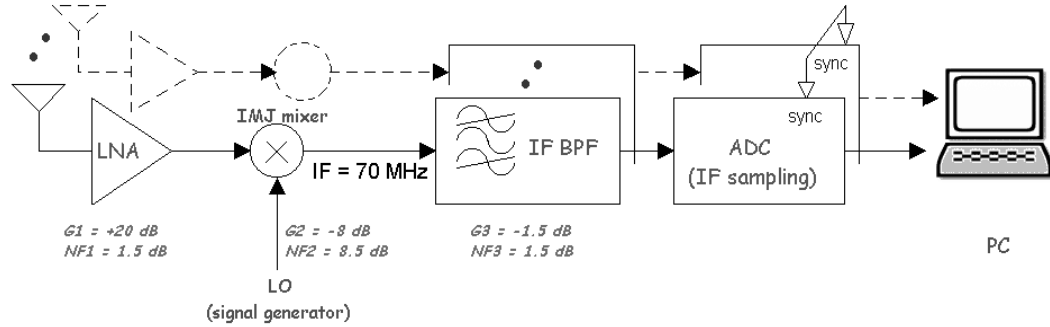


Figure 4.6: Receiver link block diagram

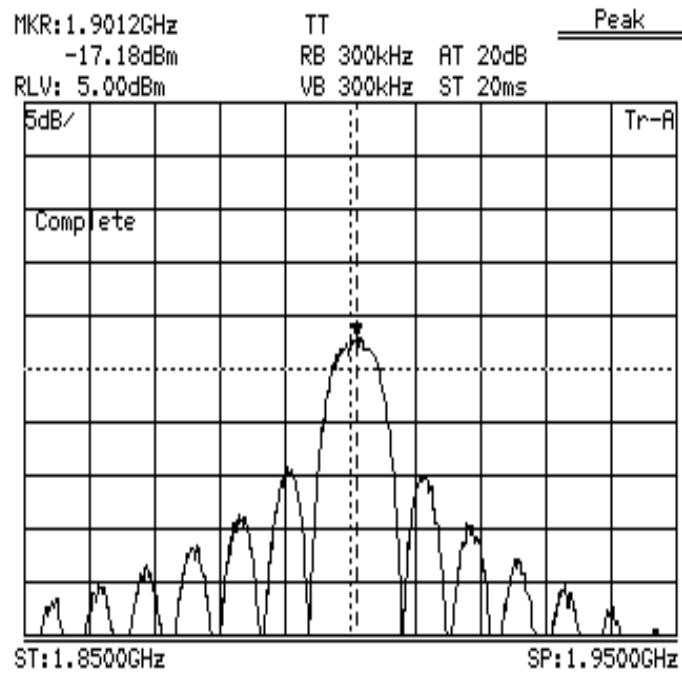


Figure 4.7: Received RF spectrum before low-noise amplifier

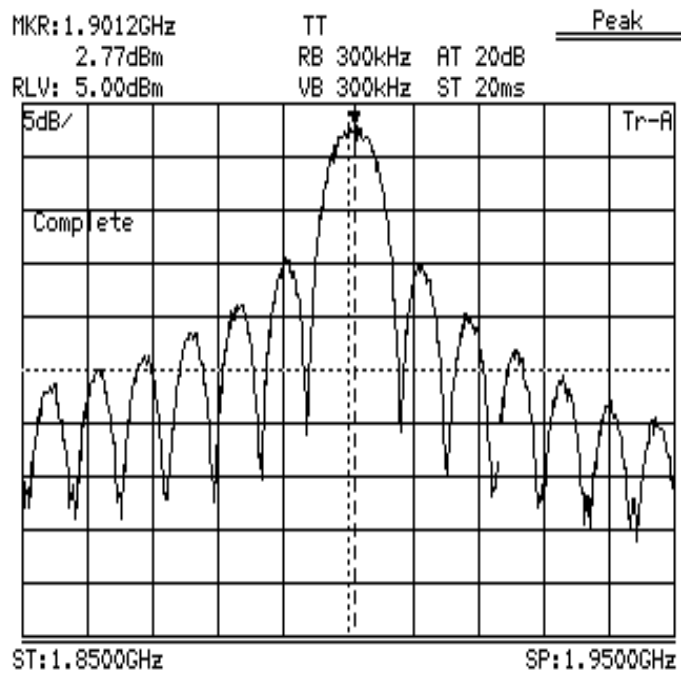


Figure 4.8: Received RF spectrum after low-noise amplification

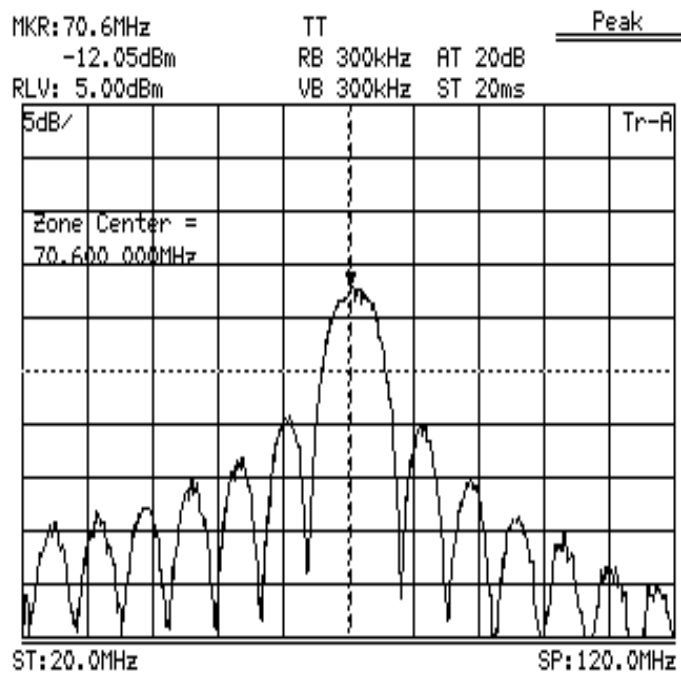


Figure 4.9: IF spectrum at IMJ mixer output

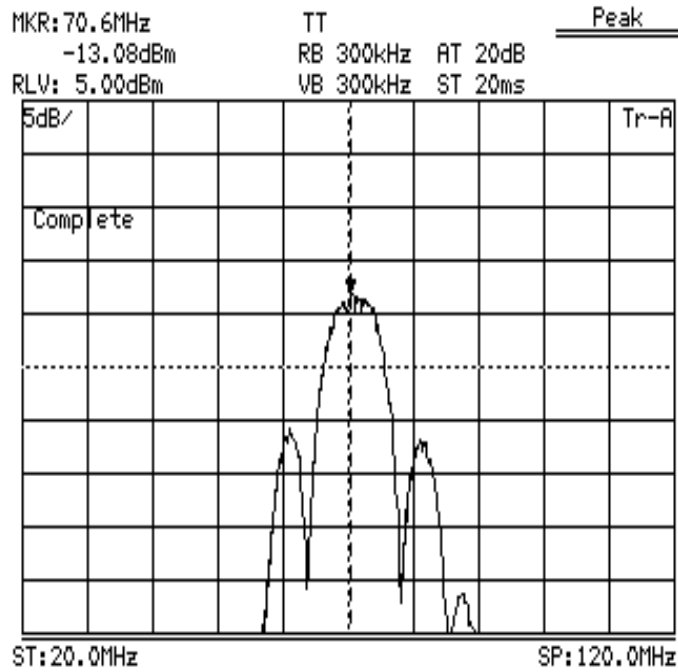


Figure 4.10: IF spectrum after bandpass filtering

The bandpass filter has a specified maximum insertion loss of 1.5 dB in the pass-band from 63.0 to 77.0 MHz [38]. The experimental insertion loss is found to be 1.1 dB by comparing Figure 4.9 and Figure 4.10. Also from Figure 4.10, large attenuation of $> 15\text{dB}$ in the stopband ($< 51\text{MHz}$ and $> 94\text{MHz}$) is shown as most of the sidelobes were attenuated below the display range of the spectrum analyzer.

After bandpass filtering the signal is downconverted to baseband via bandpass sampling at the 70 MHz IF. An example of a received signal is shown in Figure 4.11. The amplitude scale is relative to the maximum output level of the A/D converter. The A/D converter has a 12-bit resolution and all data bits are most-significant-bit (MSB) justified into a 16 bit word. Thus, the maximum output level is ± 32768 .

The functional block testing verified proper operation of each hardware component in the transmitter and receiver chain. Most of the results agreed with the

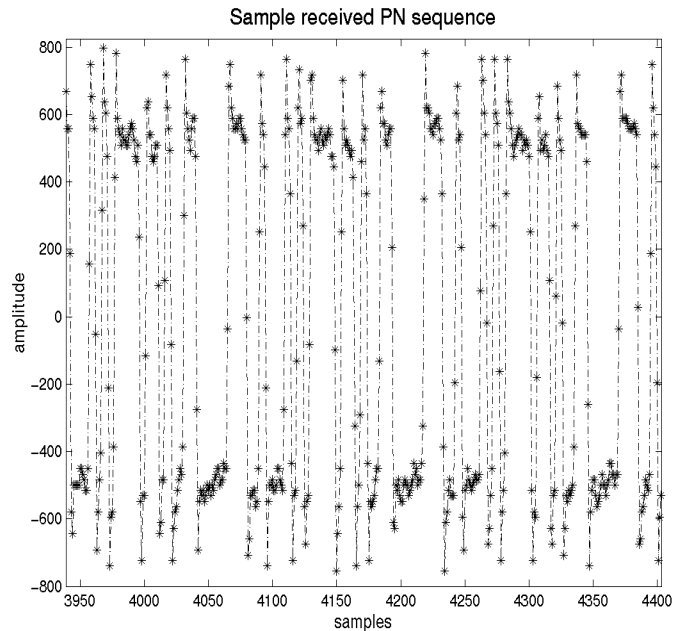


Figure 4.11: Sampled received signal at baseband

specifications provided by the manufacturers, except for the power amplifier and image rejection mixer. Under single tone (narrowband signal) test all components meet specifications, however, their performance with wideband signals is less satisfactory, comparing to the system's link budget discussed in Section 4.3. All the gain/loss measurements do not account for the cable and connector losses. The cables and connectors used in the system test have losses about 0.5-1.5 dB. In general, the loss in a cable is a function of its quality, frequency and cable length. Since we did not include cable loss in the above calculations, the power amplifier and image reject mixer gain/loss with wideband signals will be 0.5-1.5 dB closer to specifications taking cable loss into consideration.

4.3 System Analysis and Test

The system level testing is mainly focused on the receiver's sensitivity, dynamic range, link budget and antenna array phase calibration test. In addition, a simple free space propagation test is performed to provide a reference to estimate path loss in Section 5.3.

The receiver's sensitivity and dynamic range are determined by the operating range of the RF front-end amplifier, mixer and the dynamic range of the A/D converter. Maximum signal-to-noise ratio (SNR) at the A/D converter is achieved when it is operated at full-scale. Since the A/D's evaluation board does not include an automatic gain control (AGC) circuit, we cannot fully utilize the dynamic range of the A/D converter. Therefore the SNR is depended on the input signal level relative to full-scale. In calculation of the useful dynamic range, a minimum SNR must be added to satisfy the receiver's minimum performance requirement with a BER (bit-error rate) of 10^{-3} .

The receiver's link budget is calculated from the gain and noise figure (NF) of each block from the hardware specifications as shown in Figure 4.12. The system gain is $Gain_{sys}(dB) = G1_{dB} + G2_{dB} + G3_{dB} = 10.5dB$. The system noise figure is determined from the cascaded system noise figure [32] as well from the quantities shown in Figure 4.12:

$$NF_{sys} = NF1 + \frac{NF2 - 1}{G1} + \frac{NF3 - 1}{G1G2} \quad (4.1)$$

$$= 1.497 = 1.75dB \quad (4.2)$$

Using the gains obtained from functional block testing in the Section 4.4.2, the experimental gain is 4 dB ($20+(-14.9)+(-1.1)$). The experimental gain (4 dB) is much lower than the link budget (10.5 dB) due to fact that the gain at each block is roughly

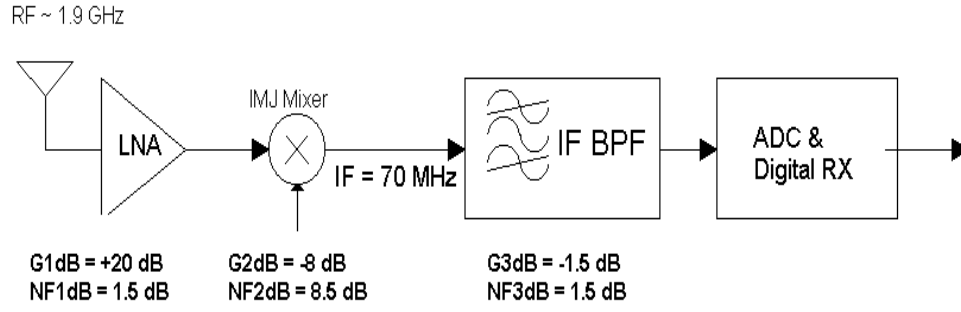


Figure 4.12: Receiver link budget

estimated from the main peak at the spectrum analyzer and also that cable losses are not included in the link budget. For a more accurate power measurement, a RF power meter, which is capable of measuring instantaneous power at RF frequency, should be used for verifying gains in Section 4.2.2.

The receiver noise floor is calculated as follows [32]:

$$\begin{aligned}
 P_{rn} &= kT + 10 \log(B) + NF_{dB} & (4.3) \\
 &= -174dBm + 71.76dB + 1.75dB \\
 &= -100.49dBm
 \end{aligned}$$

where $kT = -174dBm$ at room temperature of $25^\circ C$ and k is Boltzman's constant, $B = 15MHz$ is the 3dB bandwidth of the receiver, and NF_{dB} is receiver noise figure of $1.75dB$. The receiver sensitivity S_r is then calculated as [32]:

$$\begin{aligned}
 S_r &= P_{rn} + \frac{E_b}{N_o} + \text{implementation loss} & (4.4) \\
 &= -100.49dBm + 6.8dB + 1.0dB \\
 &= -92.69dBm
 \end{aligned}$$

where $E_b/N_o = 6.8\text{ dB}$ is the SNR required for demodulation which we have chosen

to correspond to a BER of 10^{-3} for BPSK demodulation.

Another important specification in digital receiver design is the spurious free dynamic range (SFDR) of the A/D converter. Non-linearities in the A/D converter cause spurious responses in the A/D converter's output spectrum. The SFDR indicates the ability of an A/D converter to simultaneously detect a small signal in the presence of a large interference signal [70]. The SFDR specification of the A/D converter is -71 dBm [12]. If the tolerable carrier-to-interference (C/I) ratio is 10 dB, then the minimum signal level is -61 dBm at the A/D converter or -71.5 dBm at the antenna (assuming a system gain of 10.5 dB). The minimum signal level is then constrained by the A/D converter's SFDR at about -71.5 dBm. With maximum input to the A/D converter at 0 dBm, the dynamic range of the receiver chain is about 71.5 dB.

A simple test of the free-space propagation loss was performed by placing the receiver at a close distance of roughly 1 metre from the transmitter. The signal's main peak power measured at RF is -31.5 dBm. Compared to the transmitted power of +4.5 dBm, the path loss for the 1 metre reference distance is 36dB. This loss can be assumed as free-space propagation loss and is comparable with the result in [16] of 38 dB.

Using the dynamic range and the reference path loss, the range of the measurement system is estimated from the outdoor path loss equation in [16]:

$$\overline{PL} = PL(d_o) + 10n \log\left(\frac{d}{d_o}\right) \quad (4.5)$$

where $n = 2.18$ is the outdoor path loss exponent [16], $d_o = 1$ metre, $PL(d_o) = 36dB$ and $\overline{PL} = 71.5dB$. Solving for d in Equation (4.5) yields the maximum allowable path loss (or range) for the TX-RX link to be 57 metres or 186 feet.

A receiver sensitivity analysis is performed by using single tone input with variable power. A signal generator supplies a sine wave at 1.907 GHz to the input of a receiving antenna. Input power starts at 0 dBm and decreases in steps of 5 dBm down to -70 dBm. The noise power with the -70 dBm input power (see Figure 4.13) is higher than the -50 dBm input power (see Figure 4.14). Therefore the receiver is still functional at -70 dBm input power and this result is comparable to \overline{PL} calculated above.

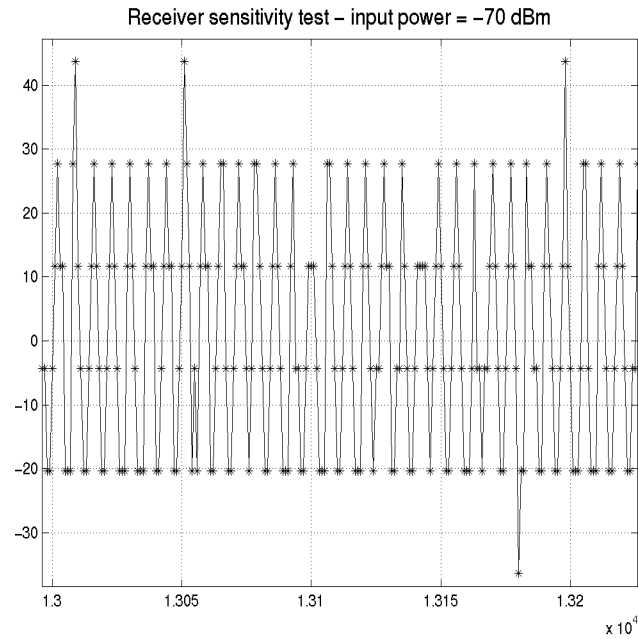


Figure 4.13: Single tone receiver sensitivity test: input power -70 dBm

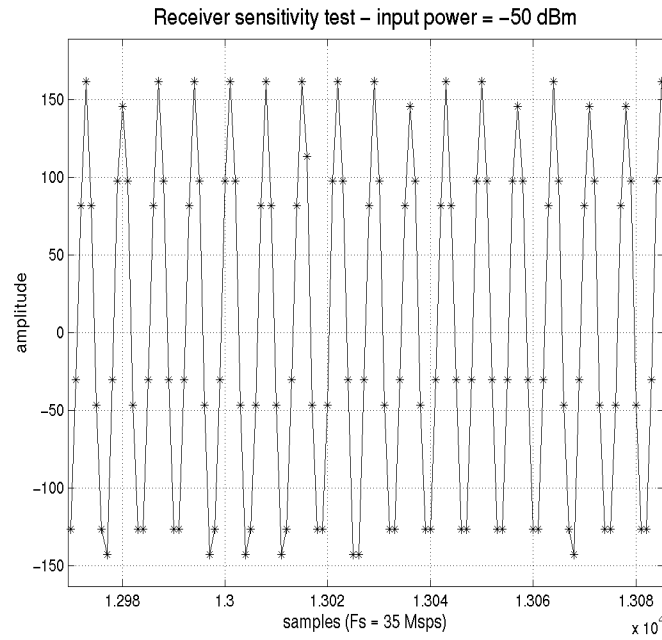


Figure 4.14: Single tone receiver sensitivity test: input power -50 dBm

4.4 Synchronization Issues of the Measurement System

In general there are two main levels of synchronization in a coherent CDMA system; carrier and symbol (or bit) synchronization. For a smart antenna system using coherent processing, synchronous sampling of all antenna elements adds an additional level of synchronization besides the two mentioned above. In our system, synchronous sampling and carrier synchronization are solved by hardware implementation, and symbol synchronization is performed off-line in software.

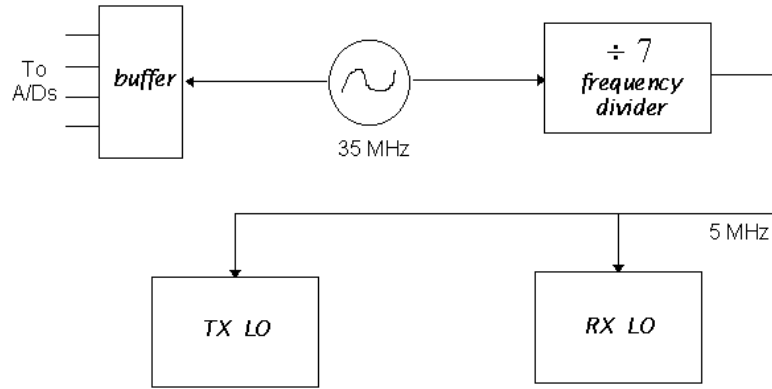


Figure 4.15: Carrier synchronization hardware logic

4.4.1 Carrier Synchronization

Carrier synchronization in our system is achieved by using a common reference signal of 5 MHz. Since bandpass sampling is used to downconvert the IF signal to baseband, the sampling clock must also be kept in constant phase with the receiver's local oscillator (LO). Our solution is to derive the 5 MHz reference from the 35 MHz sampling clock through a frequency divider. The same reference signal is also fed to the transmitter's LO. Thus, carrier synchronization is achieved as illustrated in Figure 4.15. This method eliminates the phase-locked loop (PLL) required for carrier tracking, and therefore reduces receiver hardware complexity as well as a source of uncontrolled distortion.

To illustrate the effects of imperfect synchronization on the received signal and to evaluate the hardware synchronization logic in Figure 4.15, the following three tests are conducted: a) no carrier synchronization, b) only TX and RX LOs synchronized and c) whole system synchronized.

When the carriers' relative phases are not kept constant, the effect on the received signal is similar to multiplying the received waveform by, e.g. $\cos \theta t$, where θ

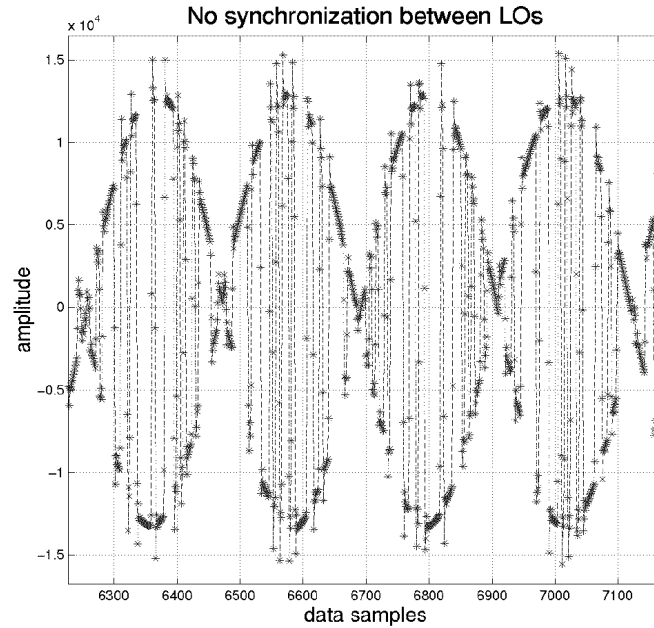


Figure 4.16: No carrier synchronization

is the phase shift. This effect is illustrated in Figure 4.16. If both the transmitter and receiver's LOs (RF carriers) are synchronized, the received signals are almost synchronized except for the sampling clock (IF carrier). In Figure 4.17, the received signal amplitude does not fluctuate as much, i.e. the phase difference θ is smaller compared to the signal in Figure 4.16.

When the synchronization logic described in Figure 4.15 is implemented, the received signal amplitude becomes relatively constant with time (see Figure 4.18).

4.4.2 Synchronous Data Sampling

Four A/D converters are used to sample the data from all four antenna elements individually. Data are then stored temporarily in a data FIFO available on the

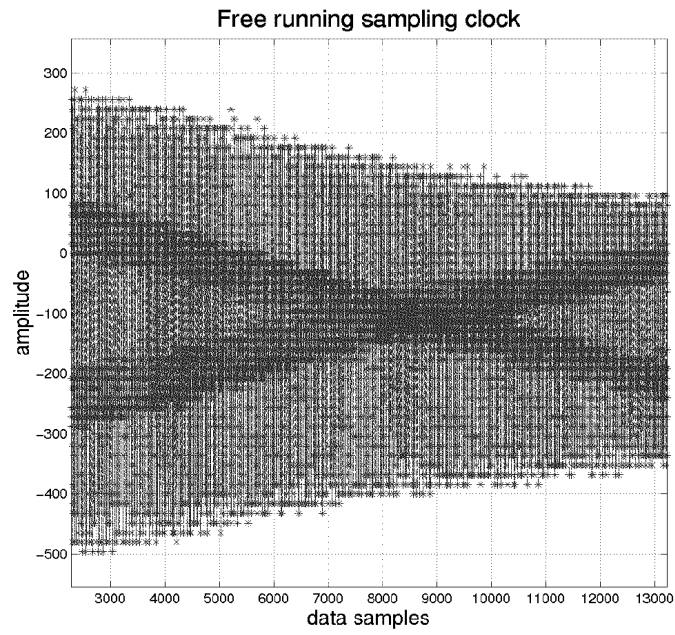


Figure 4.17: RF carrier synchronization only

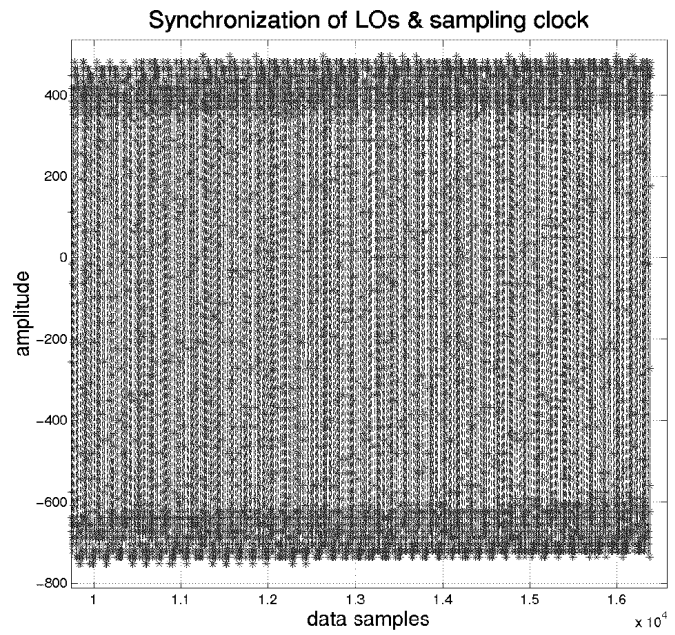


Figure 4.18: Carrier synchronization achieved using hardware logic

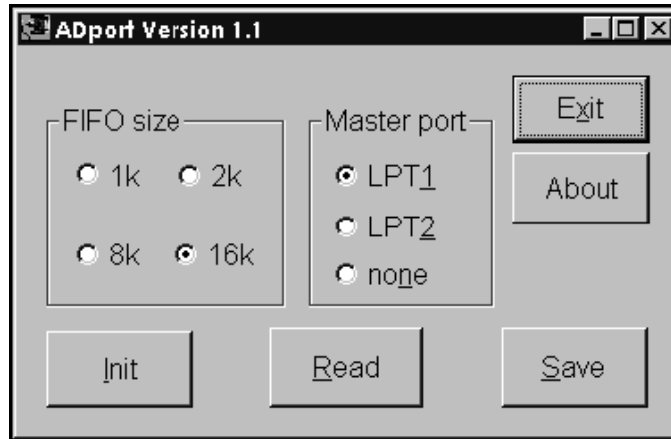


Figure 4.19: ADport control software

evaluation board. The two host PCs control the sampling process and downloading of data. One important task of the control software running on the PCs is to synchronize each array element such that all elements start sampling data at the same time. The second aspect requires that all sampling clocks be tied together or shared one identical clock. Referring to Figure 4.15, the buffer is used to distribute the same 35 MHz sampling clock to all four A/Ds. Failure to synchronize sampling will introduce a frequency dependent phase shift between each element due to the delay in sampling [41]. We have written the control software, *ADport*, in Visual C++ to provide user-friendly environment and allow batch data downloading. The main user interface of *ADport* is captured and displayed in Figure 4.19.

The features of *ADport* consist of radio buttons for the selectable FIFO's batch download size and parallel port selection (LPT1 or LPT2) of where the master board resides. The 'none' option is selected when both of the parallel ports are connected to slave boards. The bottom three buttons, in the order from left to right, represent the steps for data downloading: *initialize*, *read* and *save*. Figure 4.20 illustrates a flowchart detailing the synchronous sampling process. Details of our software design can be found in Appendix A. It is required to run two separate copies of *ADport* on

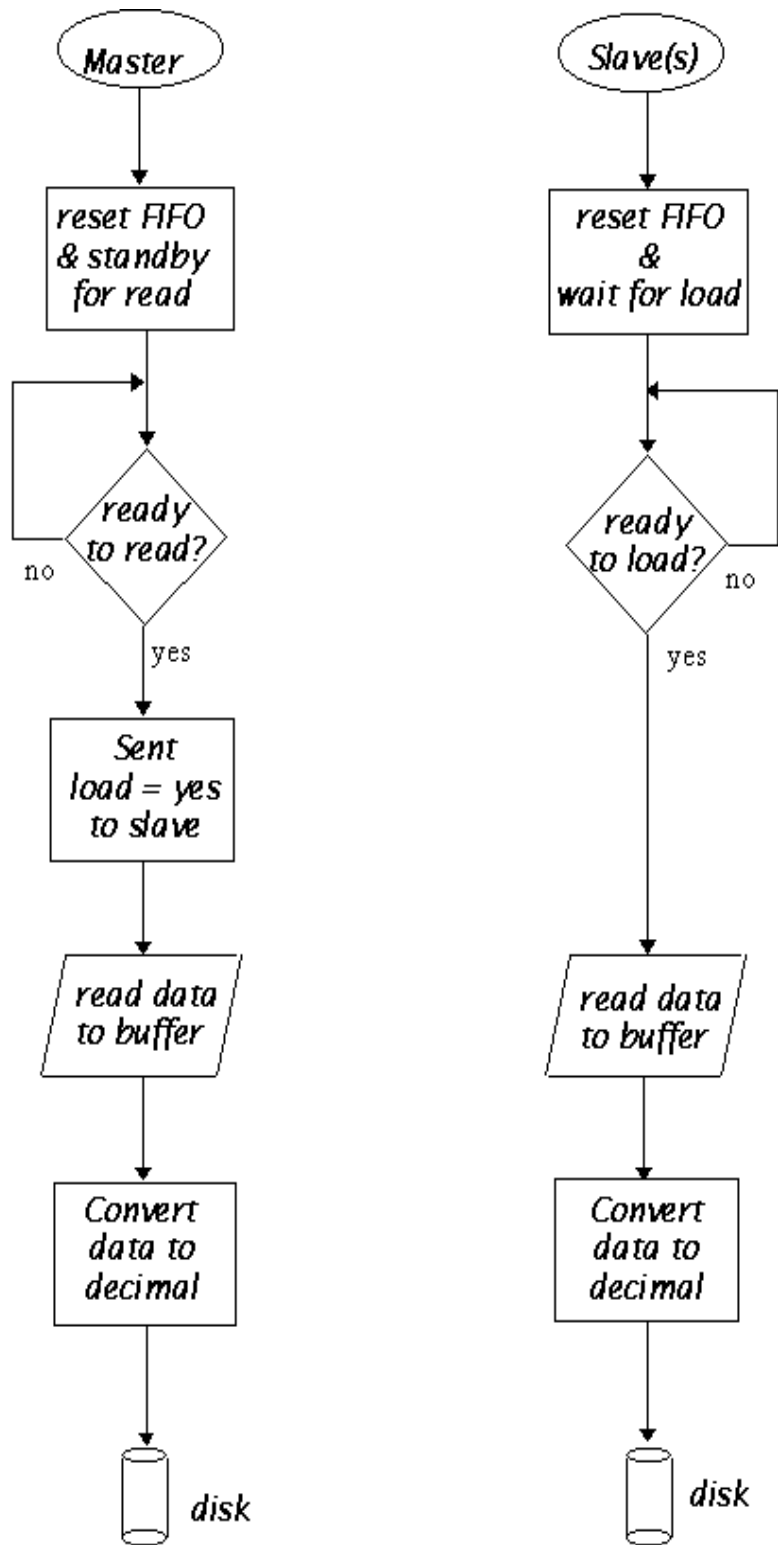


Figure 4.20: Flowchart for synchronous sampling

the two PCs because a PC can only support a maximum of two bi-directional parallel ports, namely LPT1 and LPT2. To achieve synchronous sampling of data from all four A/D boards, the following handshaking procedure is required to execute between the primary (containing the master port) and secondary copies of the software:

1. 'Init' on both primary and secondary copies
2. 'Read' on primary copy
3. 'Save' on primary copy
4. 'Save' on secondary copy
5. repeat 1-4 for next snapshot

where a 'snapshot' is a sequence continuously sampled data collected by the FIFO. Since the maximum size of the FIFO is 16k samples, a snapshot can a maximum of 16k samples. The antenna array basestation is then checked for proper synchronization by feeding a single RF carrier modulated PN sequence directly to all four antenna inputs. A zero-phase four-way shift power splitter is used to divide the signal into four equal-phase sources. This procedure is sometimes referred to as back-to-back test [5]. The back-to-back experiment setup is outlined in Figure 4.21. By inspecting the alignment of peaks and zero crossings of the sampled data in Figure 4.22, we conclude that all the channels are synchronized to within a sample (or $1/35 \times 10^6$ seconds).

4.4.3 Test for Antenna Array Calibration

Beamforming or DOA estimation techniques rely on the phase difference between the incident wavefronts of the incoming signal on each antenna element to accurately beamform towards the desired user. Therefore, it necessary to ensure that there is

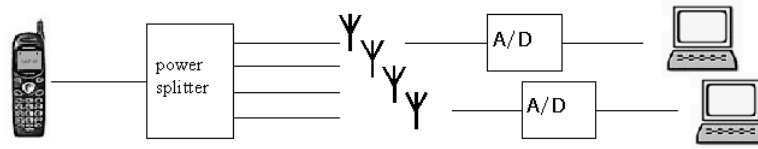


Figure 4.21: Hardware setup for back-to-back test

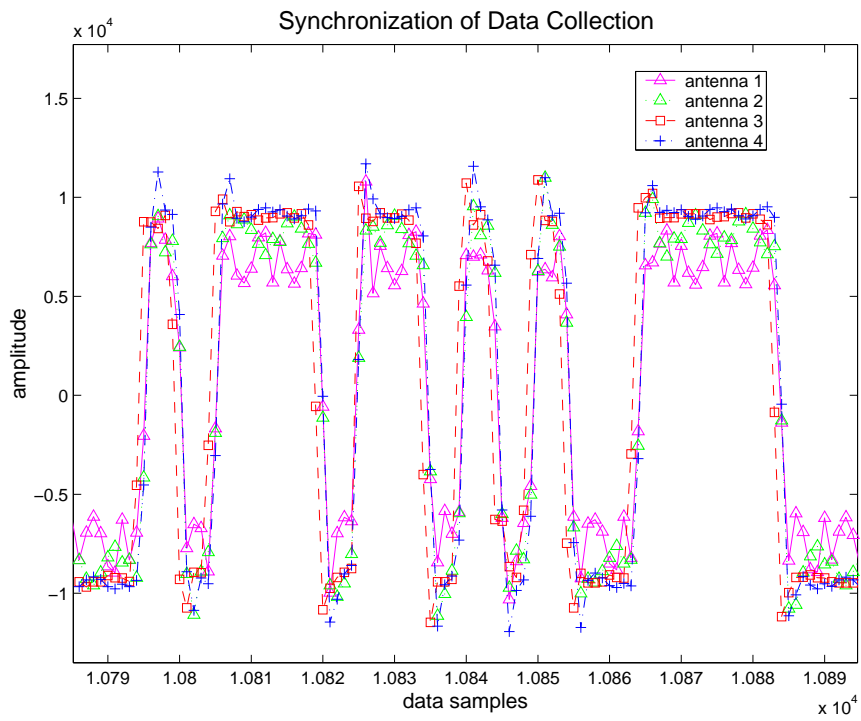


Figure 4.22: Back-to-back test: synchronous data collection achieved

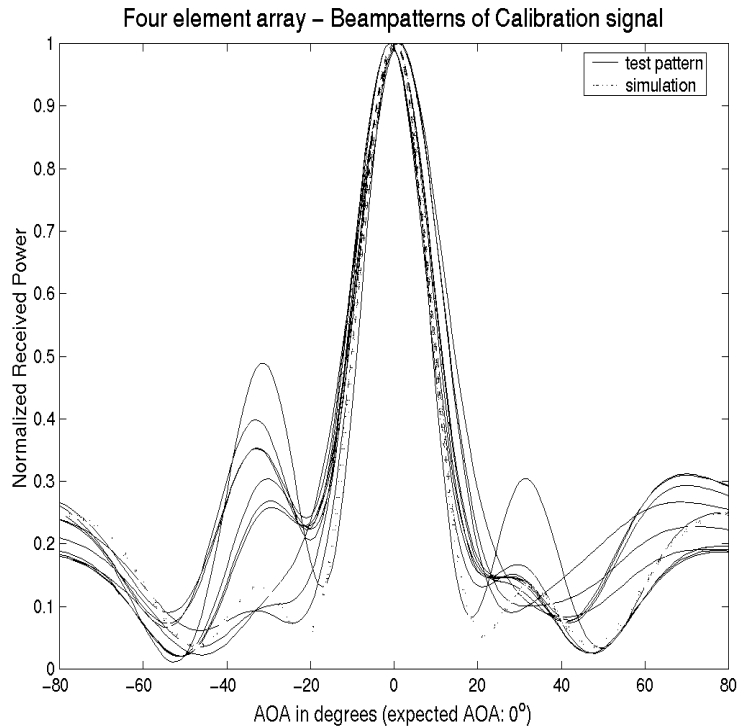


Figure 4.23: RF front-end calibration experiment

minimal phase difference between each element caused by the RF front end. The back-to-back test is also used for this calibration purpose. A series of 10 snapshots are gathered from the setup in Figure 4.21. The signals entering each antenna element are identical equivalent to DOA of zero degrees. The beampatterns in Figure 4.23 indeed have a large peak at 0° DOA, which shows that the array is calibrated.

4.4.4 Fractional Chip Sampling Offset

In a practical system the receiver does not know the chip timing for the desired user *a priori*. This chip asynchronism leads to a fractional chip offset between the receiver's matched filter sampling intervals and the chip timing for the user [37]. For example, if signal is sampled at chip rate, the worst case scenario corresponds to a half-chip

offset, which is an SNR loss of 3 dB. In our measurement system the sampling rate is five times the chip rate, and the SNR loss due to chip asynchronism is less severe. As stated in [37], the SNR loss due to fractional chip effects for a rectangular chip waveform is given by:

$$SNR = 10 \log_{10} \left[\left(\frac{1}{2m} \right)^2 + \left(1 - \frac{1}{2m} \right)^2 \right] \quad dB \quad (4.6)$$

where m/Tc is the sampling rate. For $m = 5$, the SNR loss is 0.86 dB and becomes less significant to signal-to-noise degradation. In addition, since data are processed off-line, we can exploit the fact that the entire data sequence is available for timing recovery.

4.5 Received Signal Processing

This section presents the algorithms used for off-line signal processing. Signal processing at the basestation is performed off-line on the Sun workstations. The advantages of off-line processing are the large amount of computing power available and the knowledge of the whole data sequence. The three main signal processing procedures presented here are real-to-complex data conversion, reference PN sequence generation and PN sequence synchronization (or timing recovery).

All the data gathered in experiments are raw samples of the received signal and require some signal processing before further analysis. Since received signal samples are real, a conversion to in-phase and quadrature (I-Q) format is required. Then, the next step is the generation of a local PN sequence reference for despreading. Before despreading the received signal must be aligned with the local reference. This process is called initial synchronization. If two PN sequences are out of synchronization by one chip period, there will be insufficient signal energy for reliable data demodulation

[75]. When the reference and received signal are synchronized, the despreading can begin. A tracking algorithm is incorporated into the despreading procedure to correct slippage in PN sequence synchronization during despreading. Further discussions on PN sequence tracking is presented in Section 4.5.3.

4.5.1 Real to Complex Data Conversion

In a conventional receiver, often two channels are used in the downconversion from IF to baseband. One channel is multiplied by a cosine and the other channel by a sine. The cosine and sine are generated from the same source, the sine is obtained by shifting the phase of the cosine by 90° . In this way, the zeros of the first cosine align with the peaks of the sine. This allows information of the received signal to be preserved, which would have been lost due to the zeros in the cosine reference. The signal's phase information is vital for beamforming since the received signal represents a propagation wave. The multiplications of the cosine and sine to the received signal can also be represented as one multiplication by $e^{j\theta} = \cos \theta + j \sin \theta$, using Euler's relationship. In other words, this process can be thought of as one multiplication converting data from real to complex numbers [17]. The real axis corresponds to the in-phase (I-) channel, while the imaginary axis corresponds to the quadrature (Q-) channel. The conversion from a real signal channel to a complex representation can be performed digitally using Hilbert or Fourier transforms [17]. In fact, the conversion to I-Q channels do not have to be carried out by analog hardware. This will reduce analog circuitry complexity because of available off-line computing power in our system. Figure 4.24 illustrates the two different mixing techniques.

When a real signal $x(n)$ and its Hilbert transform $\hat{x}(n) = \mathcal{H}_t\{x\}$ are used to form a new complex signal $y(n) = x(n) + j\hat{x}(n)$, the resulted signal $y(n)$ is the *analytic* or complex signal corresponding to the real signal $x(n)$. An ideal Hilbert transformer is

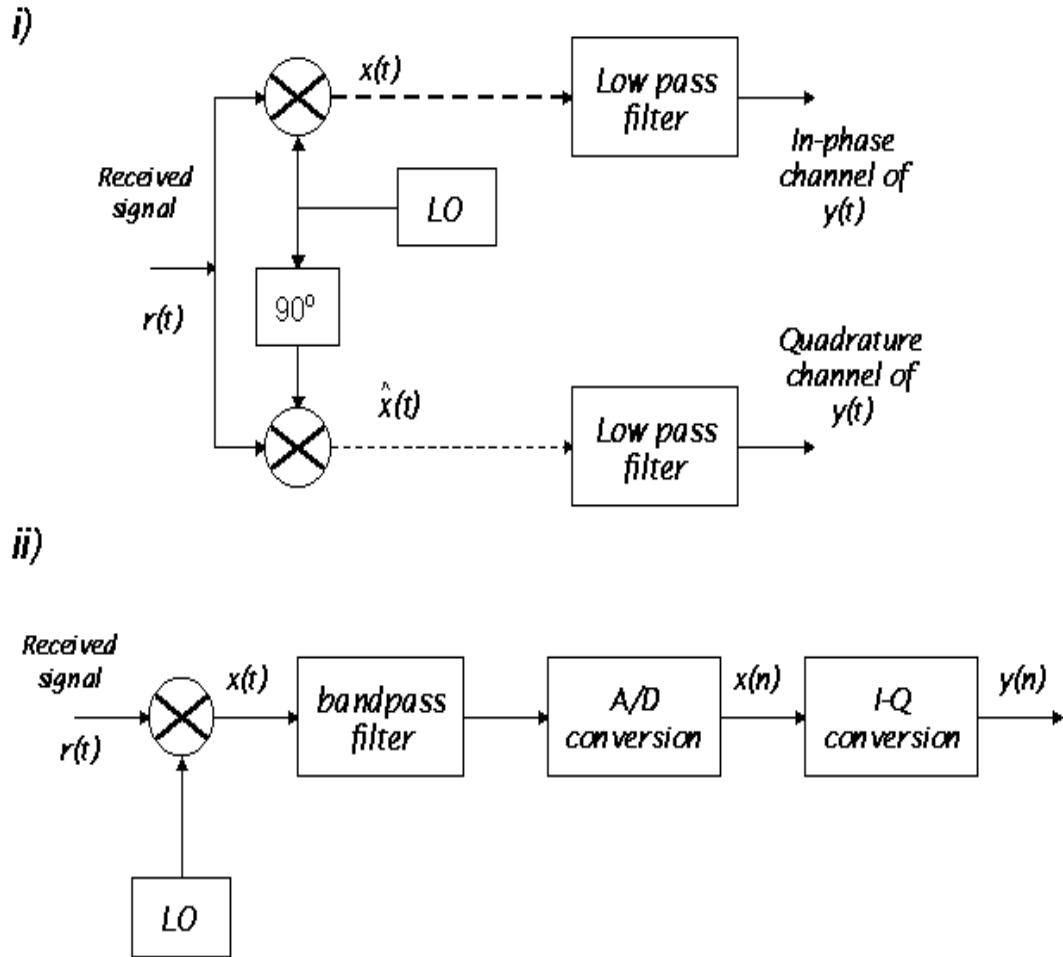


Figure 4.24: Different receiver systems: i) Quadrature (I-Q channel) and ii) One channel digital conversion to complex I-Q channel

an all-pass filter that introduces a 90° phase shift on the input signal. The frequency response of the ideal Hilbert filter is described as [54]:

$$\mathcal{H}_t(\omega) = \begin{cases} -j & 0 < \omega \leq \pi \\ j & -\pi < \omega < 0 \end{cases} \quad (4.7)$$

Analytic signals have the property that all negative frequencies of the real signal $x(n)$ are filtered out [62]. For example let:

$$x(n) = \cos(\omega_o n) = 0.5[e^{j\omega_o n} + e^{-j\omega_o n}] \quad (4.8)$$

Using 4.7, the Hilbert transform of $x(n)$ is:

$$\hat{x}(n) = 0.5[e^{j\omega_o n - j\pi/2} + e^{-j\omega_o n + j\pi/2}] \quad (4.9)$$

$$= 0.5[-je^{j\omega_o n} + je^{-j\omega_o n}] \quad (4.10)$$

Therefore, the analytic signal will be:

$$y(n) = x(n) + j\hat{x}(n) \quad (4.11)$$

$$= 0.5[e^{j\omega_o n} + e^{-j\omega_o n}] + j0.5[-je^{j\omega_o n} + je^{-j\omega_o n}] \quad (4.12)$$

$$= e^{j\omega_o n} \quad (4.13)$$

The example above illustrates that the negative frequency components of $x(n)$ and $j\hat{x}(n)$ cancel out in the sum $x(n) + j\hat{x}(n)$ and only the positive frequency component is left. This occurs for any real signal $x(n)$, not just for sinusoids [62]. It can be shown that this real-to-complex data conversion is reversible, meaning that $x(n)$ can be recovered given $y(n)$.

The real-valued to I-Q conversion at the receiver can either employ Hilbert filter or instead use Fourier transform techniques which utilize the zero negative frequency

property of analytic signals. The Fourier transform based algorithm is introduced in [17] and it is more computationally efficient due to the fast Fourier transform (FFT) algorithms. The real to complex conversion procedure is given as [17]:

1. Take the DFT of the real sequence $x(n)$ to obtain $X(\omega)$
2. Truncate the spectrum $X(\omega)$ to half its original length
3. Take the inverse DFT to obtain a complex sequence $y(n)$

The I-Q conversion algorithm can be explained in the frequency domain. The Fourier transform of the analytic received signal $y(n)$ is $Y(\omega) = X(\omega) + j\hat{X}(\omega)$. Since $\hat{X}(\omega)$ is the Hilbert transform of $X(\omega)$ by the filter in Equation (4.7), $Y(\omega)$ becomes:

$$Y(\omega) = X(\omega) + j[-j\text{sgn}(\omega)X(\omega)] \quad (4.14)$$

$$= [1 + \text{sgn}(\omega)]X(\omega) \quad (4.15)$$

$$= \begin{cases} X(\omega) & \text{for } \omega > 0 \\ 0 & \text{for } \omega < 0 \end{cases} \quad (4.16)$$

The truncation of the spectrum $X(\omega)$ in step 2 of the algorithm is equivalent to eliminating the negative frequency component as stated in Equation 4.16. The procedure is illustrated in Figure 4.25.

There are two details that have been overlooked in this discussion. First the number of frequency bins require to completely specify the DFT for a real sequence of length $2N$ is $N + 1$, not N . However the signal has no information contained at the frequency bin N (or 17.5 MHz), so a length- N inverse DFT can be used. Secondly the resulting complex sequence has a sampling rate of half of its original rate or a decimation by a factor of 2. Since we have an odd number of samples per chip, an

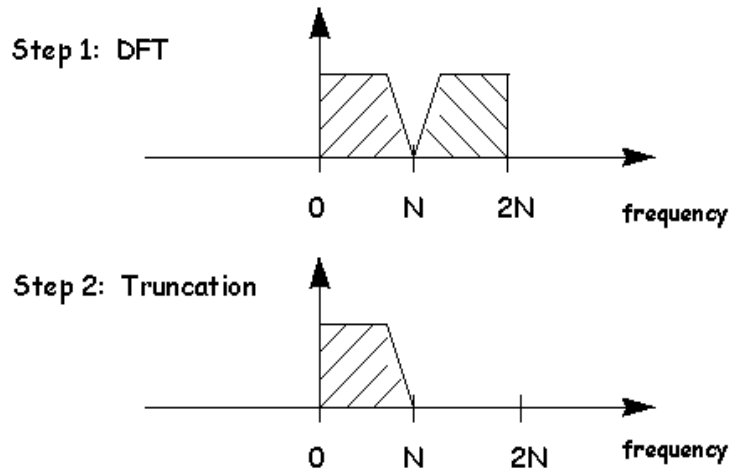


Figure 4.25: I-Q conversion algorithm Step 2 - Truncation of spectrum

interpolation step is inserted before step 1 in order to keep the sampling resolution at an integer number of samples per chip. Thus, the modified real to complex I-Q representation used in our system is:

1. a) Interpolate real sequence by a factor of 2, b) lowpass filtering of image signal resulted from step 1a)
2. Take the FFT of the real sequence $x(n)$ to obtain $X(\omega)$
3. Truncate the spectrum $X(\omega)$ to half its original length
4. Take the inverse FFT to obtain a complex sequence $y(n)$

The despreading process that follows the I-Q complex conversion is similar to conventional despreading methods described in [15] [43] [48], except for the reference PN sequence generation which will be discussed next.

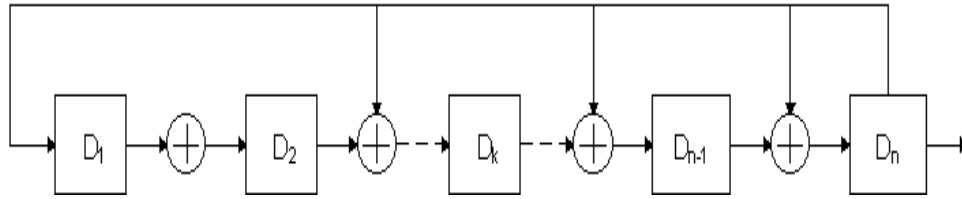


Figure 4.26: Modular multiple-tap sequence generator (MSRG)

4.5.2 Reference PN sequence generation

A linear PN code sequence generator is made up of a linear feedback shift register. The longest code sequence that can be generated is determined by the number of flip-flops or stages of delay in the generator. For n delay stages, the longest sequence that can be generated is $2^n - 1$, and this sequence is called ‘linear maximal’ [75]. There are different designs for generating maximal length sequences, for example, simple shift register generator (SSRG) and modular shift register generator (MSRG) methods described in [14]. The SX043 spread spectrum IC employs the MSRG design for its PN sequence generation [1]. An example of an MSRG sequence generator design is given in Figure 4.26 [14].

By a choosing different combination and location of feedback taps, different code sequences can be generated. One of the problems in code generation is to find the feedback logic that gives the desired code length. Fortunately, in [14] a table of feedback connections is compiled for linear maximal sequences and the SX043 IC uses this table for its sequence generation. A smaller version of this table is also available in the SX043 manual [1]. The first section of the table is shown in Table 4.1 for illustration purposes. To use the table, we first select the code length, say 31 chips, and select a five-stage shift register generator [5,2]. For modular shift register generators, the $[n, r, q, p]_m$ notation signifies an n -stage register with feedback from the n th stage modulo-2 added with the output of the r th, q th and p th stages and fed to the inputs of the $r + 1$, $q + 1$ stages, respectively [14].

<i>Number of Stages</i>	<i>Code Length</i>	<i>Maximal Taps</i>
2	3	[2,1]
3	7	[3,1]
4	15	[4,1]
5	31	[5,2] [5,4,3,2] [5,4,2,1]
6	63	[6,1] [6,5,2,1] [6,5,3,2]
7	127	[7,1] [7,3] [7,3,2,1] [7,4,3,2] [7,6,4,2] [7,6,3,1] [7,6,5,2] [7,6,5,4,2,1] [7,5,4,3,2,1]
:	:	:
:	:	:

Table 4.1: Feedback Connections for Linear m-sequences

The maximal length sequences generated from MSRG registers are real, but the received data after real to complex I-Q conversion contains real and imaginary parts. Therefore, the reference PN sequence is also required to be converted to its complex representation before despreading. The same conversion algorithm described in last section can be applied without any changes.

4.5.3 Received Signal Despreading and Code Tracking

Initial synchronization, known also as timing recovery, is achieved using a matched filter. Matched filtering of the sampled data sequence with the PN reference sequence yields peaks at various points at the output, which can then be used to determine the start of the PN sequence. The reference PN sequence is generated using methods

discussed in the last section. Despreading or code filtering begins at the estimated delay (starting point) determined by initial synchronization using a correlation method. The sequence of the received data corresponding to the length of the PN sequence is correlated with the reference sequence. The output at the correlator will be the despread data bits as the correlator moves along the received data sequence.

First, we consider the signal component from one mobile. The transmitted signal is $s_i(n)$ for user i . The channel response with no multipaths can be written as $h_i = a_i\delta(n - \tau_i)$, where a_i is the attenuation and τ_i is the integer sample delay. Then the digital baseband received signal for the i^{th} mobile can be expressed as:

$$\mathbf{x}(n) = \sum_{i=1}^N \mathbf{a}_i s_i(n - \tau_i) + \mathbf{v}(n) \quad (4.17)$$

where $s_i(n) = c_i(n)$, \mathbf{a}_i is the array response vector and $\mathbf{v}(n)$ is the additive white Gaussian noise (AWGN) vector. We describe the code filtering of received signals that do not involve beamforming and multipath combining. In later sections we consider multipaths and beamforming in our analysis and experiments. The received signal after correlation with the PN sequence of the i^{th} mobile with the estimated delay $\tilde{\tau}_i$ from initial synchronization is denoted by Z_i ,

$$Z_i = \sum_{n=kG}^{(k+1)G} \mathbf{x}(n) c_i(n - \tilde{\tau}_i) \quad (4.18)$$

$$= \sum_{n=1}^G \left[\sum_{j=1}^N \mathbf{a}_j s_j(n - \tau_j) + \mathbf{v}(n) \right] c_i(n - \tilde{\tau}_i) \quad (4.19)$$

$$= a_i d_i G + IN_i \quad (4.20)$$

where Equation (4.17) is used, $k = 0, 1, \dots$ bits and where,

$$IN_i = \sum_{n=1}^G \sum_{\substack{j=1 \\ j \neq i}}^N [\mathbf{a}_j c_j(n - \tau_j) + \mathbf{v}(n)] c_i(n - \tilde{\tau}_i) \quad (4.21)$$

The signal from the i^{th} mobile with time shift τ_i will have a processing gain of G chips per bit, and interference from undesired users can be modeled as additive white Gaussian interference [48] [53]. In this thesis we do not require this assumption.

In some simulated CDMA systems, it is usually assumed that the system is synchronized or the estimated time delay $\tilde{\tau}_i$ is perfectly recovered. However in our system and other practical systems imperfect time delay estimation can cause slippage between the received signal and the local reference PN sequence. As a result, the energy at the correlated output Z_i is reduced because the processing gain G will be lowered. In order to keep the received and the local PN sequences synchronized, a delay-lock loop (DLL) is used to track the PN sequences [34] [75].

In general, code tracking loops or delay-locked loops (DLLs) maintain synchronization of the receiver's replica of the PN code by using two correlators: an *early correlator* and a *late correlator* [34]. The early correlator uses a reference PN code that is advanced in time by some fraction of a chip relative to the current estimated time delay $\tilde{\tau}$. On the other hand, the late correlator uses the same reference PN code that is delayed by the same fraction of a chip. Then the difference signal of the two correlators' output, $e(\tau)$, is used as input to drive a voltage controlled oscillator (VCO). In turn, the VCO will correct the phase of the PN sequence reference generator.

We used a code tracking algorithm similar to the early-late tracking loop described above. DLL is usually implemented in hardware as a real time tracking device. Since the despreading of our system is performed off-line, the delay-lock loop has been adapted for digital implementation as shown in Figure 4.27. The received signal $x(n)$ is separately correlated with two PN code sequences with a time shift difference of 1 sample (or 1/5 chip). Thus, the output Z_E corresponds to the early correlation and

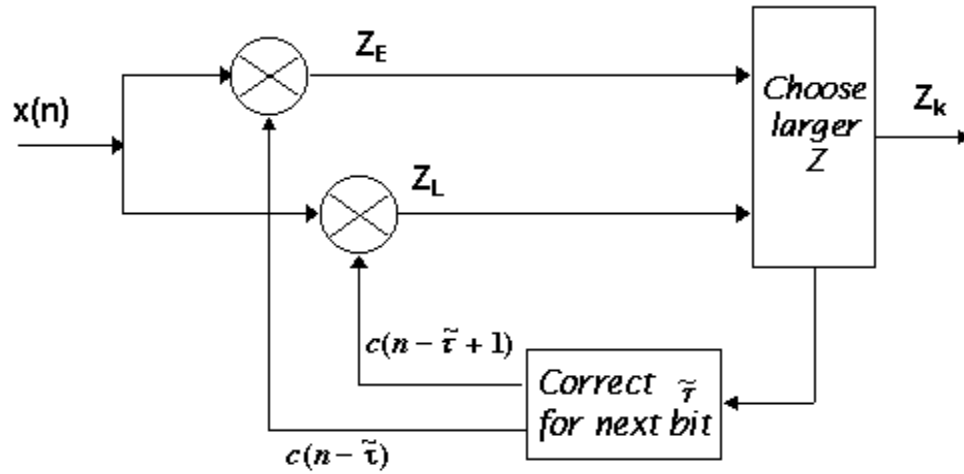


Figure 4.27: Digital early-late tracking loop

Z_L is the late correlation. Then the larger of the Z_E, Z_L is selected as the correct output since it has a higher correlation. This decision is then feedback to the tracking loop so that $\tilde{\tau}_{k+1}$ can be adjusted accordingly for the next bit output Z_{k+1} .

Using the digital early-late tracking loop, the energy at the correlated output is kept to a maximum. The correlator and tracking loop are used to despread for each user using a set of measured data gathered with two transmitters in a near-far location configuration. The bits at the correlator's output are plotted against their respective amplitude. The results are shown for cases without and with code tracking in Figures 4.28 and 4.29, respectively.

The observation made from the plots Figures 4.28 and 4.29 is the difference in the slopes of the curves. Since the transmitted data bits are always '1', we would expect similar amplitudes for each bit, especially for the Near TX where the signal experience less degradation. However, the curve without code tracking (see Figure 4.28) has a negative slope as a function of time. The negative slope suggests that without tracking, the time shift $\tilde{\tau}$ of the PN reference $c(n - \tilde{\tau})$ is moving further

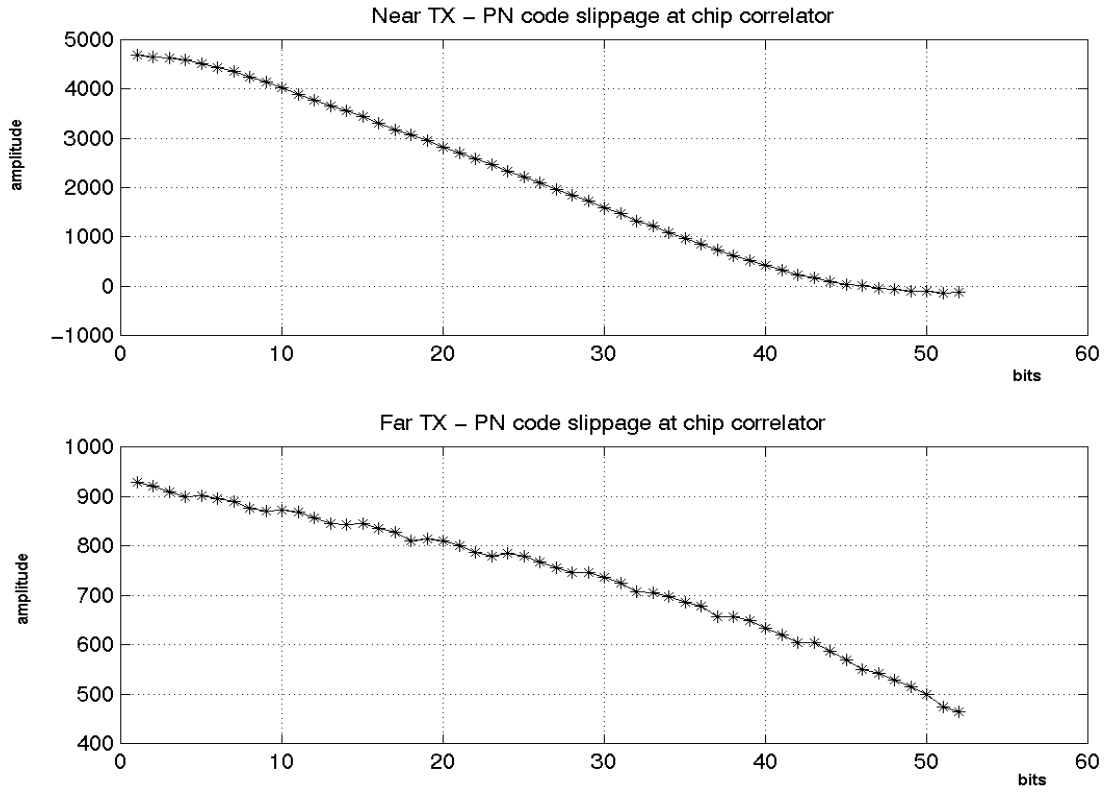


Figure 4.28: Despreading without code tracking loop

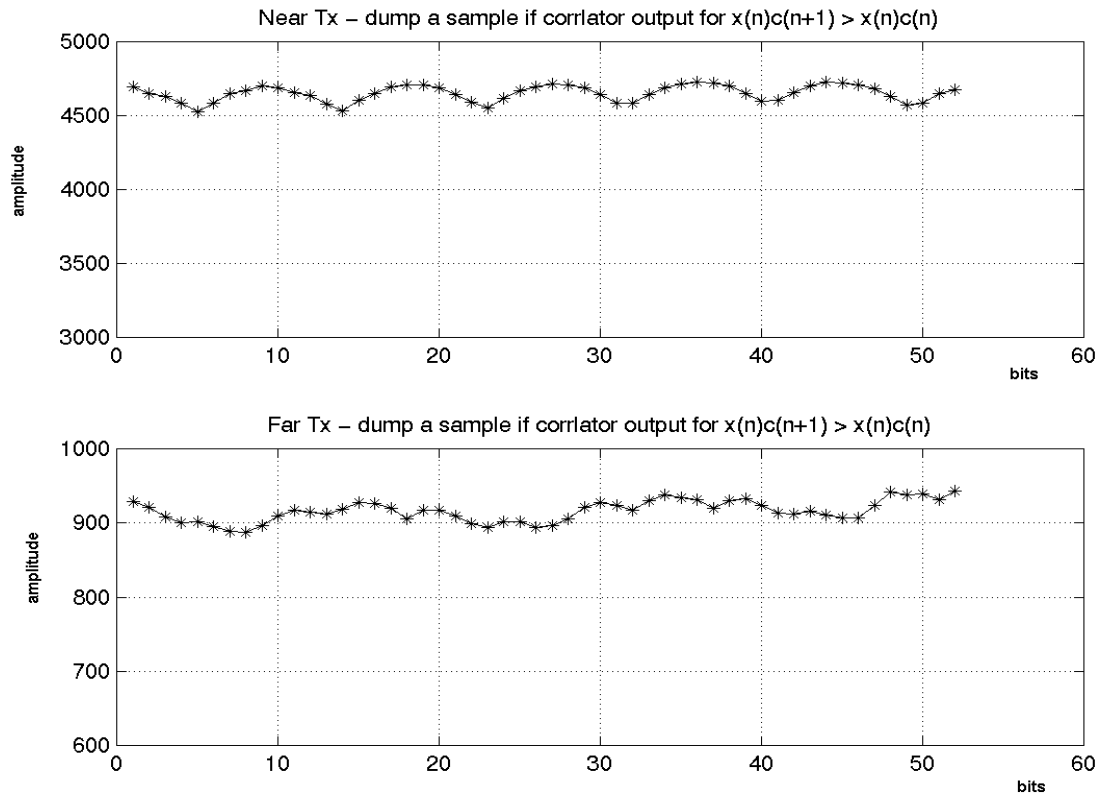


Figure 4.29: Despreading with code tracking loop

away from the received signal $x(n)$, and the value of Z_k at the correlator output is kept decreasing. If the slippage is continued for many more bits, we will expect Z_k to level off at zero and then starts rising when the time shifts between $c(n - \tilde{\tau})$ and $x(n)$ becomes zero again and repeat the same cycle. The only similarity between the two cases (with or without tracking) is that the amplitudes for the first few bits are similar. This is because initial synchronization yields the same starting point for the first bit. Furthermore, the downward slopes in Figure 4.28 indicate that there is an increasing slippage in synchronization between the received and reference PN sequence as the despreading process continues. On the other hand, we obtained a relatively flat line for both near and far TXs using the early-late tracking loop (see Figure 4.29).

4.6 Summary

In this chapter we first presented the results from system validation. In particular, functional block and system level testing are performed to ensure the measurement system is operating properly. Also some calculations are presented to estimate the system's gain, receiver's sensitivity, dynamic range and transmitting range.

Then we discussed the synchronization hardware logic and procedures in order to achieve synchronous sampling for a coherent receiving antenna array basestation. An experimental received signal waveform and beampattern are plotted in Figure 4.22 and in Figure 4.23 to illustrate that synchronous sampling is achieved and the array is well-calibrated. Finally, three received signal processing procedures are presented, namely real to complex I-Q conversion, PN sequence generation and PN code tracking.

Chapter 5

Wideband CDMA Propagation Characteristics

5.1 Introduction

The performance of wireless communication systems is bounded by the physical limitations imposed by the mobile radio channel. The transmission path between the transmitter and receiver can vary from line-of-sight (LOS) to a path that is blocked by buildings, mountains, etc. Even when the transmitter and receiver are stationary, any movements in the surroundings can affect the received signal amplitude [46]. Modeling the radio channel has been one of the more difficult aspects of mobile radio system design. The channel is usually modeled statistically based on experimental measurements made for the proposed communication system and is specific to the operating frequency band [56] [60].

This chapter describes the outdoor wideband CDMA propagation experiments and analysis conducted using a two element antenna array and three mobile transmitters operating at $1.9GHz$. The objectives of the experiments are to test the functioning of the measurement system and study the radio propagation characteristics for wideband CDMA. In particular, we investigate the path loss and multipath delay statistics

in the outdoor environment.

5.2 First Outdoor Experimental Setup

We have performed two sets of different outdoor experiments. The first experiment involves a 2-element receiver antenna array and three transmitters, and the second using a 4-element receiver array and four transmitters. The first experiment was performed during the earlier phase of system development and the second experiment used the full system implementation, i.e. four element array and four portable transmitters. The multipath propagation analysis in this chapter is based on the data from the first experiment. The spatial and beamforming analysis presented in the next chapter utilizes data from both experiments. The second experimental setup is discussed in the next chapter.

The setup of the first experiment is shown in Figure 5.1. Three transmitters A, B and C are separated spatially by 60 degrees and located at equidistant (50 feet) away from the receiving array. Each transmitter is assigned a specific 31-chip PN sequence (or user code). Once the transmitters are programmed, they are left running continuously for the duration of the experiment. When all three transmitters are running simultaneously, we can treat each transmitter as a different user transmitting asynchronously.

The array response vector and beamforming algorithms described in Chapter 2 are based on the assumption that the incident signals are plane waves. Since the incident wave is spherical instead of planar, the direction of arrivals of the wave with respect to each individual antenna element will be different from one another. In [24] Johnson stated that the difference in DOA, due to the plane wave assumption, is

Scenario 1: i) equally spaced DOA
 ii) equidistant TXs

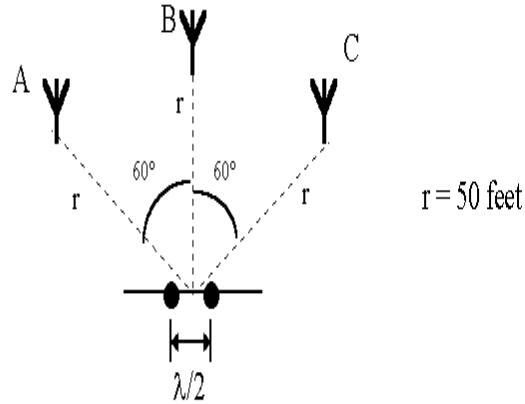


Figure 5.1: Outdoor Experiment #1: scenario 1

proportional to the array's physical dimension and inversely proportional to the T-R (transmitter-to-receiver) separation distance. To achieve a maximum DOA error of 1° in far-field assumption, the rule given by Johnson is to locate the source (TXs) at least 57 times the maximum dimension of the receiver array [24].

The wavelength, λ , for the $1.9GHz$ carrier is approximately $0.1579m$. For one half wavelength array separation, the maximum dimension of the 4-element array is $3/2\lambda = 0.2369m$. Therefore, the T-R for separation far-field assumption should be at least $57 \times 0.2369m = 13.5m$ or 44 feet. Thus the T-R distance of 50 feet used in both outdoor experiments is valid for a plane wave assumption with a maximum DOA error of 1° between the elements [24].

The outdoor experiments were conducted in the Walter Light Hall parking lot. The time of the experiments was chosen such that the parking lot was almost vacated because nearby vehicles can introduce scattering effects and lead to untrackable DOAs

Scenario 2: i) DOA: two closely spaced & one apart
 ii) equidistant TXs

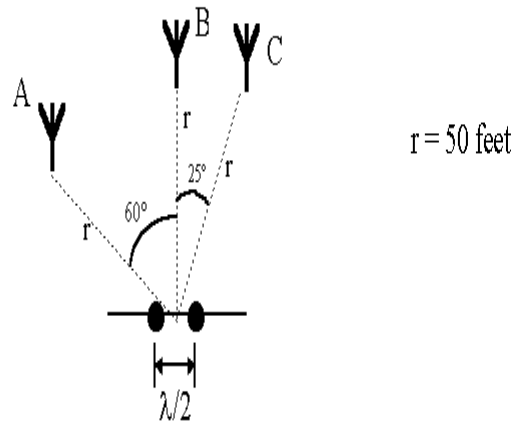


Figure 5.2: Outdoor Experiment #1: scenario 2

of received signals. Two sides of the parking lot are surrounded by buildings, one side by a row of two-storey town houses and the open side is next to a side street.

Using the setup in Figure 5.1, the physical locations of the transmitters are changed to provide other scenarios (Figures 5.2 - 5.4). A total of 20 snapshots are taken for each scenario described below:

1. Equally spaced TXs equidistant from array (Figure 5.1)
2. TX C is moved to 25 degrees apart from TX B, distances of all TXs remains the same (Figure 5.2)
3. Equally spaced TXs (60 degrees), and TX C is moved to 30 feet away from array (Figure 5.3)
4. same as 3) except TX C is now 15 feet away from array (Figure 5.4)

- Scenario 3: i) equally spaced DOA
 ii) two far & one near TXs

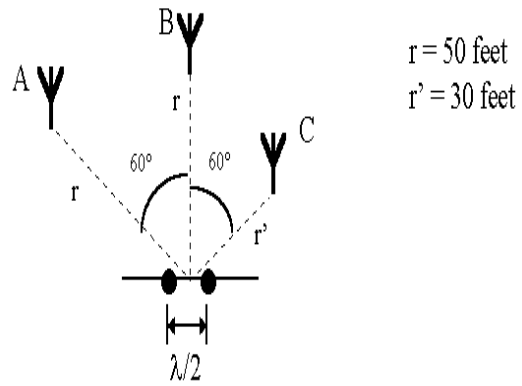


Figure 5.3: Outdoor Experiment #1: scenario 3

- Scenario 4: i) two closely spaced & one apart DOA
 ii) two far & one near TXs

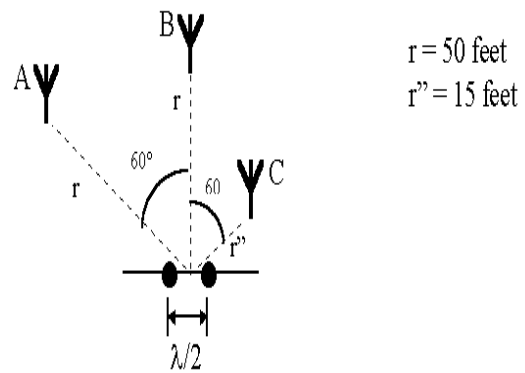


Figure 5.4: Outdoor Experiment #1: scenario 4

5.3 Path Loss Exponent Analysis

Many radio propagation models are derived from combining empirical and analytical methods [16] [18] [36]. These models are discussed extensively in [18] and [56] and procedures for deriving a path loss model from measurement data can be found in [16]. The purpose of our path loss analysis is not to repeat the derivation exercise; rather, we would like to use an existing path loss model to verify that the path loss exponent in the environment of our experiments obeys the existing models.

Both measurement-based and theoretical models state that the average received signal power decreases with distance raised to some exponent [18] [36]. The average path loss for an arbitrary T-R separation is expressed as a ratio of two distances raised to the power of n , the path loss exponent [56]:

$$\overline{PL}(d) \propto \left(\frac{d}{d_o}\right)^n \quad (5.1)$$

or

$$\overline{PL}_{dB}(d) = \overline{PL}_{dB}(d_o) + 10n \log_{10} \left(\frac{d}{d_o}\right) \quad (5.2)$$

where n is the path loss exponent which indicates how fast path loss increases with distance, d_o is a close-in reference determined from measurements close to the transmitter, and d is the T-R separation distance.

For the results presented, a close-in reference of $d_o = 6$ feet is used. To gather data for path loss analysis, TX C is programmed to run by itself in scenarios 1,3, and 4 described before. The power of the received signal averaged over the 20 trials for each scenario is tabulated in Table 5.1.

Using the data in Table 5.1 we can determine the MMSE (minimum mean square error) estimate for the path loss exponent, n . The MMSE estimate can be found by

<i>Distance, d (ft)</i>	<i>Received Power (dB)</i>	<i>Path loss, \overline{PL}_{dB}</i>
6	74.93	0
15	59.00	-15.93
30	55.39	-19.54
50	53.81	-21.12

Table 5.1: Measured received signal power from outdoor experiment

the following method [56]. Let p_i be the received power at distance d_i and let \hat{p}_i be the estimated received power using the path loss model in Equation (5.1). The sum of squared errors between the measured received power and the estimated value is:

$$e(n) = \sum_{i=1}^3 (p_i - \hat{p}_i)^2 \quad (5.3)$$

The value of n which the mean square error is minimized is found by setting the first derivative of $e(n)$ to zero, and then solve for n . Since $\hat{p}_i = p_{dB}(d_o) - \overline{PL}_{dB}(d_i)$, from Equation (5.2) and assume that $p_{dB}(d_o) = 0dB$ and $\hat{p}_i = p_{dB}(d_o) - 10n \log(d_i/6)$. Substituting all the values of p_i and \hat{p}_i into (5.3), the sum of squared errors is then:

$$e(n) = [0 - 0]^2 + [-15.93 - (-3.98n)]^2 + \quad (5.4)$$

$$[-19.54 - (-6.99n)]^2 + [-21.12 - (-9.21n)]^2$$

$$= 149.52n^2 - 789.00n + 1081.62 \quad (5.5)$$

Setting the derivative in (5.5) to zero:

$$\frac{de(n)}{dn} = 299.04n - 789.00 = 0 \quad (5.6)$$

The path loss exponent is estimated to be $n = 2.64$. The sample variance can be obtained as $\sigma^2 = e(n)/4$ for $n = 2.64$. The sample variance is then found to be $10.19dB$. As noted in [56], a greater number of measurements are needed to reduce σ^2 in general. Table (5.2) shows some experimentally determined path loss exponents

<i>Environment</i>	<i>Path loss exponent, n</i>
Free space	2
Urban area cellular radio	2.7 to 3.5
Shadowed urban cellular radio	3 to 5

Table 5.2: Path Loss Exponents for Different Environments [36]

given in [36] for different environments.

The path loss exponent estimated in our measurements is $n = 2.64$ and it fits into lower end of the urban cellular radio environment. Even though the number of data measurements is not large enough to conclude on the actual path loss experienced in the Walter Light Hall parking lot, but our result shows that the path loss behaves in a predictable manner, and it can be estimated using models presented in literature.

5.4 Multipath Delay Profile Analysis

There are three popular methods for multipath propagation measurement (sometimes referred to as channel sounding) at RF frequencies: direct RF pulse, spread spectrum sliding correlator and swept frequency spectrum [56]. Cox [8] first used the sliding correlator method to study multipath delay characteristics in outdoor suburban environments at $910MHz$. Bultitude [5] later used this technique for microcellular channel sounding analysis, as did Sousa [63] and Jorgensen [25]. Since our measurement system transmits PN sequences at a high chip rate, data gathered by a single antenna element can be processed to obtain multipath delay profiles based on the sliding correlator method. The other channel sounding methods cannot be used for our system because a RF pulse generator and vector network analyzer are required

for direct RF pulse method and swept frequency spectrum, respectively.

When the received PN sequence is correlated by sliding the local reference sequence along the received signal, a time autocorrelation function of the transmitted PN sequence is obtained if no multipaths are present. If multipaths are present, they will have different time delays, and will maximally correlate with the reference PN sequence at multiple times. The energy of each individual path passing through the correlator depends on the time delay. Then, the envelope of the channel profile is defined by [8] as:

$$|y(t)| = [x_I^2(t) + x_Q^2(t)]^{1/2} \quad (5.7)$$

where $x_I(t)$ is the I-channel and $x_Q(t)$ is the Q-channel of complex received signal. The plot of the logarithm of $|y(t)|$ in dB versus time is called a multipath delay profile by [8] and [63]. An example of a multipath delay profile is shown in Figure 5.5.

5.4.1 Thresholding for Multipath Detection

Since the delay profile is the magnitude of the received complex impulse response, the effect of noise amplitudes must be accounted before the extraction of multipath (echoes) can begin. This requires an estimate of the background noise variance, σ_N^2 , to set a threshold to distinguish between multipaths and noise peaks. The choice of threshold is important because it affects the multipath statistics calculated from measurement data. If the noise threshold is set too low, then the values of measured delay spreads and peak amplitude variances will be high because noise peaks are treated as multipaths [56].

Thresholding procedures have not been discussed extensively in the past. Papers [8] [9] and [73] do not mention any use of a threshold for multipath detection. On

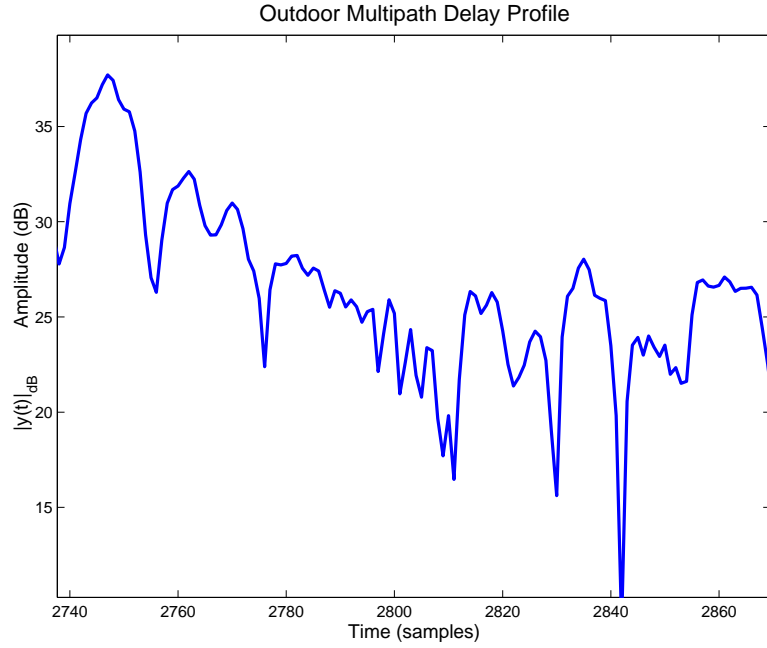


Figure 5.5: Sample Multipath Delay Profile

the other hand, [5], [25] and [55] describe the use of thresholds, determined based on worst case SNR and is set constant for every profile. In [55], profiles are ad hocly discarded which have less than $10dB$ of dynamic range in order to satisfy a minimum SNR.

Our thresholding method is based on the work of [63] where the threshold is set to achieve a constant false alarm rate (CFAR), independent of the SNR. The extraction of multipath echoes from a delay profile can be viewed as a signal detection problem. If no multipath is present, then the delay profile's amplitude is due to noise. On the other hand, multipaths yield stronger peaks with unknown amplitude and delay in the delay profile. Therefore, we define the null and alternative hypothesis as:

$$\mathcal{H}_0 : y[n] = w_{Ray}[n] \quad n = 5, \dots, N - 5 \quad (5.8)$$

$$\mathcal{H}_1 : y[n] = \begin{cases} w_{Ray}[n] & n = 5, \dots, n_i - 1, n_i + 1, \dots, N - 5 \\ A + w_{Ray}[n] & n = n_i \end{cases} \quad (5.9)$$

where $w_{Ray}[n]$ is Rayleigh distributed noise amplitude, A is the unknown multipath peak's amplitude, n_i is the delay time of peak i and $i = 1, 2, \dots, L$ multipaths. The number of multipaths, L , in each profile is unknown and our multipath detection procedure allow us to gather statistics and determine a statistical average of L . In (5.9) the time interval for detection is assumed to be at least greater than a chip period (5 samples) away from the main peak, i.e. n starts from the fifth sample. This is because the smallest delay possible is assumed to be 1 chip away from strongest path (main peak).

For the null hypothesis, if we assumed the noise of the in-phase and quadrature channels is independently distributed zero-mean Gaussian, then the amplitude of the noise envelope is Rayleigh distributed [49]:

$$p(y) = \begin{cases} \frac{y}{\sigma^2} \exp\left(-\frac{y^2}{2\sigma^2}\right) & 0 \leq y \leq \infty \\ 0 & y < 0 \end{cases} \quad (5.10)$$

where σ_N^2 is the time-average power of the received signal (i.e. noise variance). The probably that the signal envelope exceeds a specified value Y is given by the right-tail probability of $p(y)$ [49] [56]:

$$P(Y) = Pr(y > Y) = 1 - \int_0^Y p(y) dy = \exp\left(-\frac{Y^2}{2\sigma_N^2}\right) \quad (5.11)$$

The estimation of sample noise variance, σ_N^2 , is simply the mean square value of all the sample amplitudes if no signal is present. When valid multipaths are present, the estimated σ_N^2 can be greatly distorted because of the large peaks of the multipaths. With the knowledge that very few data points will actually contain peaks (10-15) within a profile, σ_N^2 can be estimated based on the median of the sample's amplitude [63]. The median level of Rayleigh distribution is found by letting $P(Y_{med}) = 0.5$ in

Equation (5.11) and solving for σ_N . Then we will get:

$$\sigma_N = \frac{Y_{med}}{\sqrt{\ln 4}} \simeq 0.85Y_{med} \quad (5.12)$$

Equation (5.12) implies that σ_N is 0.85 times of the median level for a Rayleigh distribution. For the detection of multipaths, we would like to devise a scheme that is independent of the SNR. The threshold is set to obtain a constant false alarm rate (CFAR) and it is widely used in signal detection for radar applications [29] [63]. The CFAR scheme requires to set the threshold to

$$\gamma = \zeta \sigma_N \quad (5.13)$$

where ζ is a constant. The false alarm probability, i.e., the probability that the noise exceeds the threshold in any sample in the absence of an multipath, is equal the probability with $Y = \gamma$ substituted to (5.11) and becomes [63]:

$$P_{FA} = \exp\left(-\frac{\zeta^2}{2}\right) \quad (5.14)$$

The threshold, γ , is calculated by substituting P_{FA} (say 10^{-3}) into Equation (5.14) and solve for ζ . Then ζ and σ_N , estimated from (5.12), are substituted into (5.13) for γ . The procedure for determining the threshold is summarized as follows:

1. find the median, Y_{med} , of the data samples in a delay profile
2. $\sigma_{N_{dB}} = Y_{med_{dB}} - 1.4dB$ (using Equation (5.12))
3. set threshold $\gamma_{dB} = \sigma_{N_{dB}} + \zeta_{dB}$ according to a selected P_{FA} (using Equation (5.13) and (5.14))
4. detect multipath peaks which are above the threshold

A sample delay profile with estimated noise level and the resulting threshold is shown in Figure 5.6. After the threshold is obtained, then the detection process

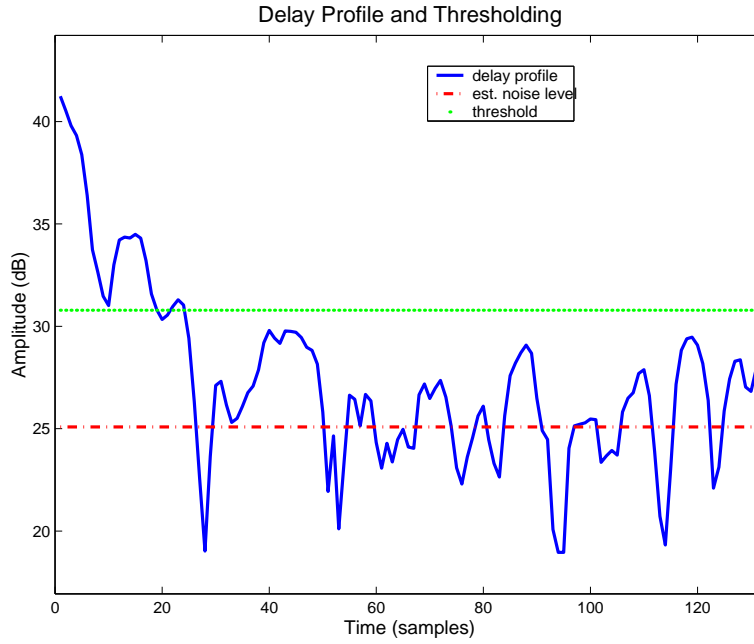


Figure 5.6: Delay Profile and Thresholding ($P_{FA} = 10^{-3}$)

can begin with pre-selected parameters such as P_{FA} and smoothing resolution. The details of this operation and the choice of parameters are presented next.

5.4.2 Multipath Detection and Hypothesis Testing

In the multipath detection process, a given sample in the delay profile is considered a valid peak (or echo) if its amplitude exceeds the threshold and the neighbouring time samples on each side have lower amplitude. In addition the identification of peaks is based on a resolution parameter to smooth out noisy peaks in a profile. A low-pass filter (LPF) for smoothing will have a region of support proportional to the resolution parameter. The same delay profile in Figure 5.6 has been smoothed by a LPF with a resolution of $1/2$ chip (region of support $\pm 1/2$ chip) and shown in Figure 5.7. The main difference between the two profiles is the reduction in the number of noisy bumps on the peaks. In fact, the smoothing operation by the LPF can be treated as an estimator of the true location of a peak taking neighbouring samples

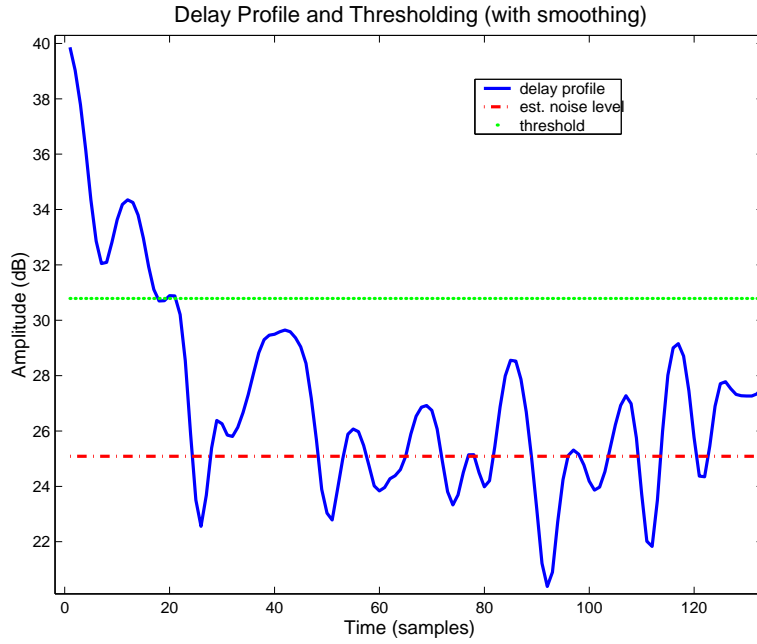


Figure 5.7: Delay Profile and Thresholding (with smoothing)

into account.

In [63] Sousa observed that confidence limits for the median noise level estimation are known to be somewhat larger than predicted by mean level estimation. This is because the mean level estimation is based on the information from all sample points, whereas, median level estimation only uses one sample point. Therefore we try to estimate σ_N^2 recursively and improve its accuracy. After peaks have been identified in our detection procedure, the original data minus the peaks are used to re-estimate the noise variance. The second noise estimate on average, is found to be only $0.6dB$ below the first estimate. For our purposes the first estimate is adequate for multipath detection.

The detection performance of our CFAR detection scheme can be quantified by the probability of detection. The detection probability P_D is the probability at which

\mathcal{H}_1 is chosen in (5.9) when the \mathcal{H}_1 is true [29]. If we assume that the delay profile is Rician distributed if a peak is present, P_D is the right-tail probability of a Rician probability density function (pdf). In [29] Kay shows that P_D can be related to the noncentral Chi-square random variable using (5.7) as follows:

$$P_D = Pr \left\{ y > \sqrt{\hat{\gamma}} \right\} = Pr \left\{ \sqrt{\frac{x_I^2 + x_Q^2}{\sigma_N^2}} > \sqrt{\frac{\hat{\gamma}}{\sigma_N^2}} \right\} \quad (5.15)$$

$$= Pr \left\{ \frac{x_I^2 + x_Q^2}{\sigma_N^2} > \frac{\hat{\gamma}}{\sigma_N^2} \right\} \quad (5.16)$$

$$= Q_{\chi_2^2(\lambda)} \left(\frac{\hat{\gamma}}{\sigma_N^2} \right) \quad (5.17)$$

where $Q_{\chi_2^2(\lambda)}(x)$ is the right tail probability of the non-central chi-squared pdf with two degrees of freedom [29] and where λ is the SNR for the k^{th} sample in the profile [63]. If we let $\gamma = \sqrt{\hat{\gamma}}$, then we obtain:

$$Pr\{y > \gamma\} = Q_{\chi_2^2(\lambda)} \left(\frac{\gamma^2}{\sigma_N^2} \right) \quad (5.18)$$

Using Equations (5.13) and (5.14), then the detection performance can be expressed into:

$$P_D = Q_{\chi_2^2(\lambda)}(-2 \ln P_{FA}) \quad (5.19)$$

Values of P_D in Equation (5.19) are determined numerically as in [29], and the detection performance curves for various false alarm rates are plotted in Figure 5.8. The probability of detection increases slowly with increasing energy-to-noise ratio, λ . This is because for the Rician pdf, as σ_N^2 increases, there is still a high probability that the sample amplitudes will be small [29].

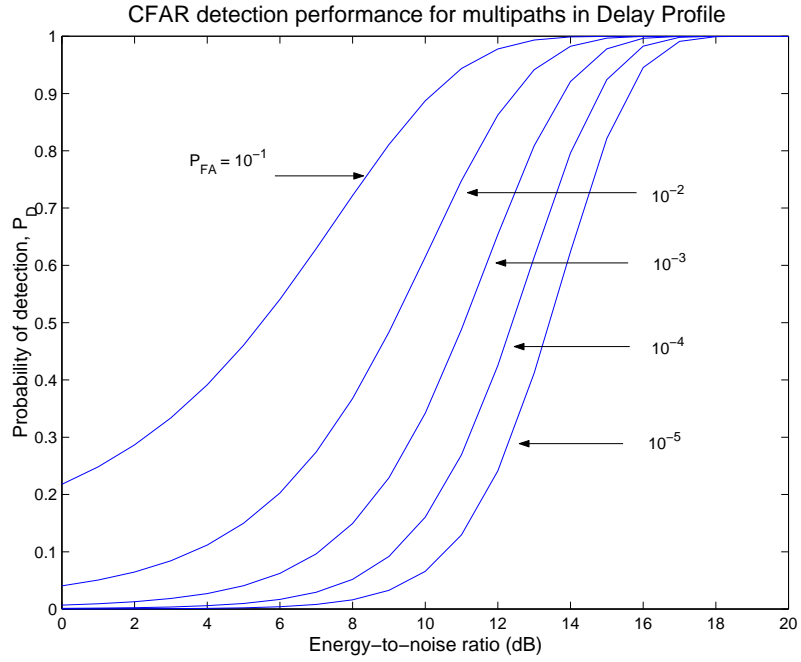


Figure 5.8: Detection Performance of CFAR Multipath Detection for Different False Alarm Probabilities

5.4.3 Peak Amplitude and Time Delay Calculations

During multipath detection, the amplitude and delay of all the valid echos are recorded for each profile. The collection of individual multipath amplitudes and delays from all 20 trials are then used to calculate: mean and variance of peak amplitude, excess delay and excess delay spread.

The sample mean and variance of peak amplitude are calculated respectively, as [49]:

$$\bar{P} = \frac{1}{L} \sum_{i=1}^L P(\tau_i) \quad (5.20)$$

and

$$\bar{v}_P = \frac{1}{L-1} \sum_{i=1}^L (P(\tau_i) - \bar{P})^2 \quad (5.21)$$

where $P(\tau_i)$ is individual peak amplitude and L is the number of detected multipaths. Excess delay time is defined as the first moment of the delay profile relative to the first arrival delay (main peak) in [8] [56] and calculated as:

$$\bar{\tau} = \frac{\sum_{i=1}^L \tau_i P(\tau_i)}{\sum_{i=1}^L P(\tau_i)} - \tau_0 \quad (5.22)$$

where L is the number of multipath peaks detected and $P(\tau_i)$ is the amplitude for delay τ_k . Since these delays are measured relative to the main peak arriving at the receiver, τ_0 can be assumed to be zero. The rms delay spread, σ_τ , is the square root of the second central moment of the delay profile defined as [8] [56]:

$$\sigma_\tau = \sqrt{\overline{\tau^2} - \bar{\tau}^2} \quad (5.23)$$

where

$$\overline{\tau^2} = \frac{\sum_{i=1}^L \tau_i^2 P(\tau_i)}{\sum_{i=1}^L P(\tau_i)} \quad (5.24)$$

The results calculated from Equations (5.20), (5.21), (5.22) and (5.23) are tabulated for comparing different cases of probability of false alarms and smoothing resolutions. The results and analysis are presented in the next section.

5.5 Single User Multipath Detection

For the analysis of multipath propagation effects of a single user in an outdoor environment, transmitter A in scenario 1 (Figure 5.1) is programmed to transmit a 31-chip PN sequence by itself and twenty runs are recorded. Since there are 100

profiles (bits) per snapshot, a total of 2000 profiles are available for analysis.

In previous sections, the method of hypothesis testing for multipaths detection is described. Our method has two variable parameters: the probability of false alarm (P_{FA}) and smoothing resolution. Changing these parameter values can produce different results, thus, three reasonable values are chosen for each parameter to further investigate and decide on the most suitable parameters for our purposes. The results are presented in Tables 5.3-5.16.

The probability of false alarms investigated are 10^{-2} , 10^{-3} and 10^{-4} . These values are chosen based on the performance curve in Figure 5.8. The energy-to-noise ratio of our profiles is normally about $10 - 12dB$, therefore a false alarm probability of 10^{-4} is rather conservative as the detection probability is 0.45 or less and the number of expected false alarms will be one or less with 2000 profiles. On the other hand, a false alarm probability of 10^{-2} has a P_D of 0.7 – 0.9 but the penalty for choosing a higher P_{FA} is that several peaks that will be falsely detected among the 2000 profiles.

As the probability of false alarm increases, the number of detected echoes also increases because higher P_{FA} arises from a lower threshold. Tables 5.7, 5.9 and 5.15 show that for the same resolution parameter of 1/2-chip, the number of 2^{nd} strongest peaks detected for $P_{FA} = 10^{-4}$ is 477 and jumps to 700 and 967 for $P_{FA} = 10^{-3}$ and 10^{-2} , respectively. Also, the mean peak amplitudes are lower and rms delay spreads are higher for 2^{nd} peaks in Table 5.8 comparing to 5.10 and 5.16. The mean amplitude is lower probably because the threshold is set at a lower value (higher P_{FA}). Thus, 2^{nd} peaks with lower amplitudes are being considered as valid and lowers the average amplitude. At the false alarm rate of 10^{-2} , we would expect an average of 20 false alarms with 2000 profiles ($10^{-2} \times 2000$). The false alarms can be a contributing factor to why the 2^{nd} peak delay spread of 1.37chips in Table 5.8 is much higher than

0.38*chips* and 0.26*chips* in Table 5.10 and 5.16, respectively. Since the false alarms are noise peaks, their respective time delays have greater variance than the ones for true peaks. A similar trend is also found at different smoothing resolutions: 1/5-chip resolution are given in Tables 5.4 and 5.6, and 1-chip resolution in Tables 5.12 and 5.14. Since a false alarm probability of 10^{-4} is too conservative and 10^{-2} allows too many false peaks, we conclude that we should choose the false alarm probability parameter for our testing as $P_{FA} = 10^{-3}$.

Three different values of smoothing resolution are tested: 1/5-chip, 1/2-chip and 1-chip. Since one-fifth of a chip is our A/D sampling resolution, the region of support of $\pm 1/5\text{chip}$ provides only coarse averaging. The region of support of $\pm 1\text{chip}$ is two times the minimum resolvable multipath delay of 1 chip period [48] [75] and it provides too much smoothing. As for 1/2-chip resolution, the region of support is equal to 1 chip ($\pm 1/2\text{chip}$). The 1/2-chip resolution is a reasonable value because the region of support for 1-chip resolution is too long and may combine two resolvable peaks by averaging over the interval of 2 chips, whereas 1/5-chip resolution is too short to remove noisy peaks.

In Tables 5.3, 5.7 and 5.11, the number of detected multipaths decreases as resolution changes from 1/5-chip, 1/2-chip and to 1-chip for a constant false alarm probability, mainly because the smoothing operation removes more noisy peaks with longer region of support. Also an interesting effect of smoothing is the effect on the peak amplitudes. The mean peak amplitudes are similar for 1/5- and 1/2-chip resolution in Tables 5.4 and 5.8, but the amplitude range of the delay profile is being compressed when the region of support is longer as in the 1-chip resolution in Table 5.12. The relative peak amplitudes in Table 5.12 are higher, meaning that the multipath peaks are closer to the main peak's amplitude. Since the duration of peaks and valleys in a delay profile is usually less than 2 chips, averaging samples over an interval of 2 chips will lower peak amplitudes while raising the valleys. Since a smoothing resolution of

1/5-chip is too coarse and 1-chip removes too many peaks including those that are valid, so, 1/2-chip smoothing resolution is suitable for our analysis.

<i>Peak</i>	<i>Number of Occurrence</i>
1 st	2000
2 nd	1130
3 rd	429
4 th	148

Table 5.3: Number of Multipaths Detected: $P_{FA} = 10^{-2}$, resolution = 1/5-chip

<i>Peak</i>	<i>Amplitude (dB)</i>	<i>Var</i>	<i>Mean Delay (chips)</i>	<i>Delay Spread</i>
1 st	-	11.40	-	-
2 nd	-6.53	2.04	2.69	0.88
3 rd	-7.45	2.60	6.17	4.41
4 th	-8.14	3.36	10.42	6.35

Table 5.4: Amplitudes and Delays of Multipaths: $P_{FA} = 10^{-2}$, resolution = 1/5-chip

<i>Peak</i>	<i>Number of Occurrence</i>
1 st	2000
2 nd	801
3 rd	175
4 th	29

Table 5.5: Number of Multipaths Detected: $P_{FA} = 10^{-3}$, resolution = 1/5-chip

<i>Peak</i>	<i>Amplitude (dB)</i>	<i>Var</i>	<i>Mean Delay (chips)</i>	<i>Delay Spread</i>
1 st	-	11.40	-	-
2 nd	-6.26	2.19	2.55	0.44
3 rd	-6.94	2.75	6.79	4.70
4 th	-6.19	3.77	9.15	4.48

Table 5.6: Amplitudes and Delays of Multipaths: $P_{FA} = 10^{-3}$, resolution = 1/5-chip

<i>Peak</i>	<i>Number of Occurrence</i>
1 st	2000
2 nd	967
3 rd	327
4 th	57

Table 5.7: Number of Multipaths Detected: $P_{FA} = 10^{-2}$, resolution = 1/2-chip

<i>Peak</i>	<i>Amplitude (dB)</i>	<i>Var</i>	<i>Mean Delay (chips)</i>	<i>Delay Spread</i>
1 st	-	10.84	-	-
2 nd	-5.91	1.91	2.56	1.37
3 rd	-7.08	2.14	7.23	4.97
4 th	-7.54	4.07	9.64	5.95

Table 5.8: Amplitudes and Delays of Multipaths: $P_{FA} = 10^{-2}$, resolution = 1/2-chip

<i>Peak</i>	<i>Number of Occurrence</i>
1 st	2000
2 nd	700
3 rd	124
4 th	7

Table 5.9: Number of Multipaths Detected: $P_{FA} = 10^{-3}$, resolution = 1/2-chip

<i>Peak</i>	<i>Amplitude (dB)</i>	<i>Var</i>	<i>Mean Delay (chips)</i>	<i>Delay Spread</i>
1 st	-	10.84	-	-
2 nd	-5.67	1.90	2.35	0.38
3 rd	-6.50	2.50	8.23	5.10
4 th	-5.45	5.78	5.81	4.48

Table 5.10: Amplitudes and Delays of Multipaths: $P_{FA} = 10^{-3}$, resolution = 1/2-chip

<i>Peak</i>	<i>Number of Occurrence</i>
1 st	2000
2 nd	550
3 rd	91
4 th	0

Table 5.11: Number of Multipaths Detected: $P_{FA} = 10^{-2}$, resolution = 1-chip

<i>Peak</i>	<i>Amplitude (dB)</i>	<i>Var</i>	<i>Mean Delay (chips)</i>	<i>Delay Spread</i>
1 st	-	12.37	-	-
2 nd	-2.45	2.44	2.37	2.94
3 rd	-3.48	2.00	11.92	4.81
4 th	-	-	-	-

Table 5.12: Amplitudes and Delays of Multipaths: $P_{FA} = 10^{-2}$, resolution = 1-chip

<i>Peak</i>	<i>Number of Occurrence</i>
1 st	2000
2 nd	319
3 rd	19
4 th	0

Table 5.13: Number of Multipaths Detected: $P_{FA} = 10^{-3}$, resolution = 1-chip

<i>Peak</i>	<i>Amplitude (dB)</i>	<i>Var</i>	<i>Mean Delay (chips)</i>	<i>Delay Spread</i>
1 st	-	12.37	-	-
2 nd	-1.69	1.71	1.81	0.32
3 rd	-2.11	1.05	11.14	2.83
4 th	-	-	-	-

Table 5.14: Amplitudes and Delays of Multipaths: $P_{FA} = 10^{-3}$, resolution = 1-chip

<i>Peak</i>	<i>Number of Occurrence</i>
1 st	2000
2 nd	477
3 rd	47
4 th	2

Table 5.15: Number of Multipaths Detected: $P_{FA} = 10^{-4}$, resolution = 1/2-chip

<i>Peak</i>	<i>Amplitude (dB)</i>	<i>Var</i>	<i>Mean Delay (chips)</i>	<i>Delay Spread</i>
1 st	-	10.84	-	-
2 nd	-5.26	1.60	2.29	0.26
3 rd	-6.17	2.54	5.92	3.94
4 th	-2.70	4.00	2.46	1.16

Table 5.16: Amplitudes and Delays of Multipaths: $P_{FA} = 10^{-4}$, resolution = 1/2-chip

5.6 Multiple-User Multipath Detection

In a CDMA cellular system the undesired users (interferences), after despreading at the receiver, appear as noise to the desired user [19] [35]. This assumption is based on the cross-correlation characteristics of PN sequences assigned to different users [52]. If the experiment in Section 5.5 is repeated again with multiple users, the noise level of delay profiles is expected to be higher after despreading [35]. The experiment in Section 5.5 is repeated here with multiple transmitters. Twenty trials are recorded for the case where three TXs A, B and C are transmitting. Assuming that TX A is the desired user, the delay profiles are generated by despreading the received signal with TX A's code using the sliding correlator method described in Section 5.4.

Typical profiles from the multiple-user scenario is shown in Figure 5.9. The noise level in these profiles is higher than the single-user case as shown in Figure 5.6. In fact, we calculated that the average noise level of all delay profiles with multiple users is $33.8dB$, which is $4.53dB$ higher than the single-user profiles.

Since the noise level is higher in the multiple-user scenario, the threshold for multipath detection must also set higher in order to keep the same false alarm probability as suggested by Equation (5.13). Given that the transmitting power is the same as in the single-user scenario, the increase in threshold level means that the probability of detection is lower as the energy-to-noise ratio is reduced. The probability of detection, P_D in (5.19) and Figure 5.8 indicates that P_D is lower when the energy-to-noise ratio is reduced. Thus we expect the number of multipaths detected to be fewer with multiple users.

Using the CFAR detection scheme described earlier, the profiles from all twenty trials are analyzed. At first glance, the multipath statistics obtained have high mean and variances for both amplitude and delay statistics. The results from each trial are

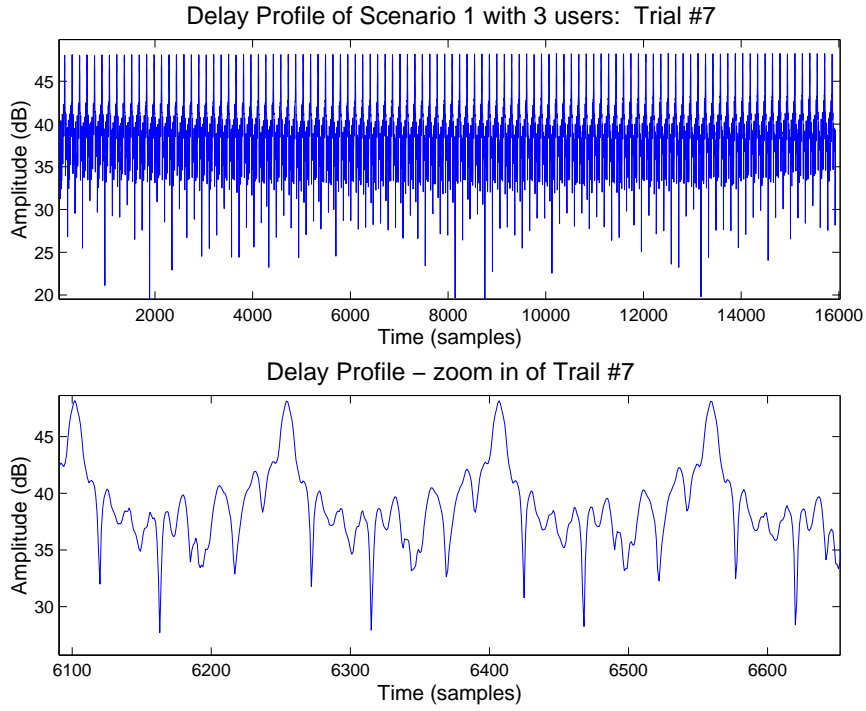


Figure 5.9: Typical Delay Profile of Multiple Users

then analyzed separately and found that Trial #2 is the root of the problem. The 2^{nd} peak amplitudes detected for that particular trial are between 3 to 5dB (about 40 detected peaks), which is not expected because the main peaks should always be higher. Inspecting the delay profiles of Trial #2 in Figure 5.10 yields some interesting insights. The first 5000-6000 thousand samples in the delay profile are very noisy and the main peaks are missing (see bottom plot). The reason for those erroneous results is that the initial PN sequence acquisition algorithm cannot find a main peak to provide proper tracking of the PN sequence. Our acquisition algorithm is designed to search for a valid main peak within the first 1000 samples. Thus, when a main peak is absent at the beginning of the trial, the algorithm falsely tracks the wrong peak. The high 2^{nd} peak amplitudes detected from this trial are actually the main peaks starting from about 8000 samples and beyond.

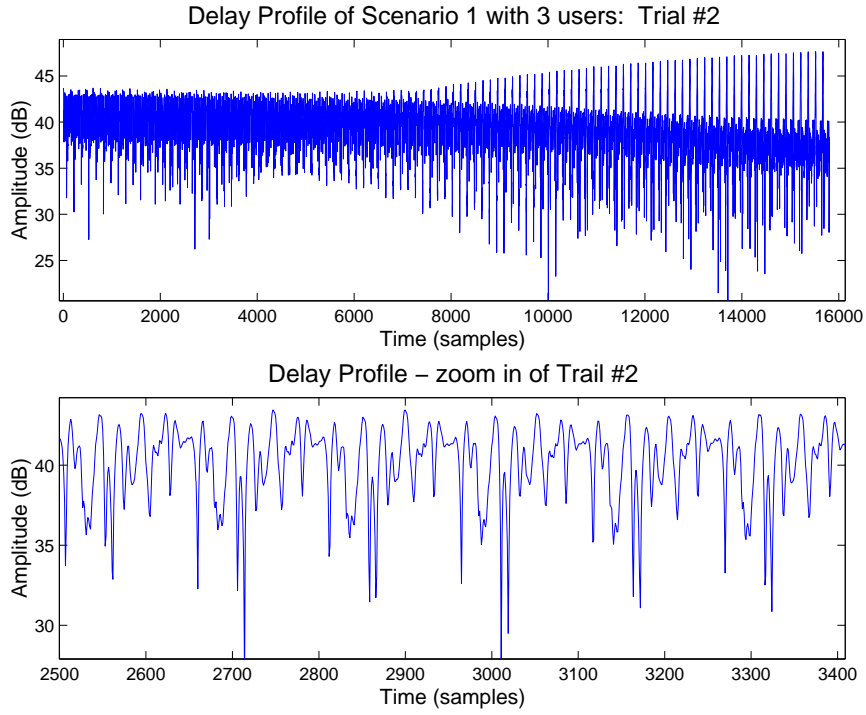


Figure 5.10: Delay Profile of Multiple Users with Fading effects

As mentioned in [22], a short search interval for acquisition in a multipath channel can lead to a total miss of a correct code phase. To correct the tracking problem described above requires to design a more advance algorithm which includes out-of-lock detection and re-acquisition procedures [59] and is out of the scope of this thesis. Since there is only one trial that creates the errors, the multipath detection is performed for the multiple-user scenario excluding Trial #2. The P_{FA} of 10^{-3} and smoothing resolution of $1/2$ -chip are used in the detection, and the results are summarized in Tables 5.17 and 5.18. The number of detected 2^{nd} and 3^{rd} peaks are fewer comparing to Table 5.9 because of the higher thresholds required with multiple users. The mean amplitudes and delays are similar except for delay spread. The delay spread of 2^{nd} peak with three users is 2.94 chips while it is only 0.38 chips in the single user case (Table 5.10). Two out of 50 detected 2^{nd} peaks in Table 5.17 have delays close to 13 chips which are not expected, based on the experimental environment and TX-RX

<i>Peak</i>	<i>Number of Occurrence</i>
1 st	1900
2 nd	50
3 rd	3
4 th	0

Table 5.17: Number of Multipaths Detected: 3 users

<i>Peak</i>	<i>Amplitude (dB)</i>	<i>Var</i>	<i>Mean Delay (chips)</i>	<i>Delay Spread</i>
1 st	-	6.11	-	-
2 nd	-7.60	1.01	2.81	2.05
3 rd	-8.34	1.81	2.66	0.36
4 th	-	-	-	-

Table 5.18: Amplitudes and Delays of Multipaths: 3 users

separation distance. These two peaks with long delay are more likely caused by false alarms than by a tracking problem because a tracking problem tends to create multiple errors consecutively as discussed above. Excluding these two data points the mean delay and delay spread become 2.42 chips and 0.40 chips respectively, which replace the third row of Table 5.18. These results are very similar to those found in the third row of Table 5.10. Thus, the large delay spread of the 2nd peak (2.05 chips) is likely a result of two false alarms.

5.7 Time Variation of Delay Profiles

The multipath detection results presented in the last two sections are statistical averages gathered from twenty experimental trials. Although the variance of peak amplitude and delay spread give some indication of the level of variation about their mean, little information can be summarized on the consistency of consecutive delay profiles recorded in time. The variation of consecutive delay profiles serves as useful information since it is related to the stability of the operating environment and channel.

To study the variation of delay profiles, consecutive profiles within a snapshot are averaged as follows:

$$y_{avg}(n) = \frac{1}{N_w} \sum_{i=1}^{N_w} y_i(n) \quad (5.25)$$

where y_i is individual delay profile, N_w is the window size for averaging. The values of window sizes used are 1, 5, 10, 25, 50 and 100 bits. A window size of 1 bit is same as no averaging, whereas window size of 100 bits is equivalent to an average of all profiles within a snapshot.

Two methods are employed to characterize delay profile variations: (1) the correlation coefficient between profiles is generated using different window sizes, and (2) multipath statistics of averaged profiles are generated. The first method compares averaged profiles from different window sizes and give indications of profile stability within a trial and across separate trials. Applying the multipath detection procedure to averaged profiles allows us to compare the stability of multipath amplitudes and delays to those with no averaging presented earlier.

5.7.1 Delay Profile Correlations

To illustrate the variation of delay profiles between different trials, two samples are shown in Figures 5.11 and 5.12. These two plots show consecutive profiles of the snapshots where each snapshot contains about 100 profiles (or bits). The plot of Trial #1 stays rather constant throughout, but Trial #14 varies, especially within the first 50 profiles.

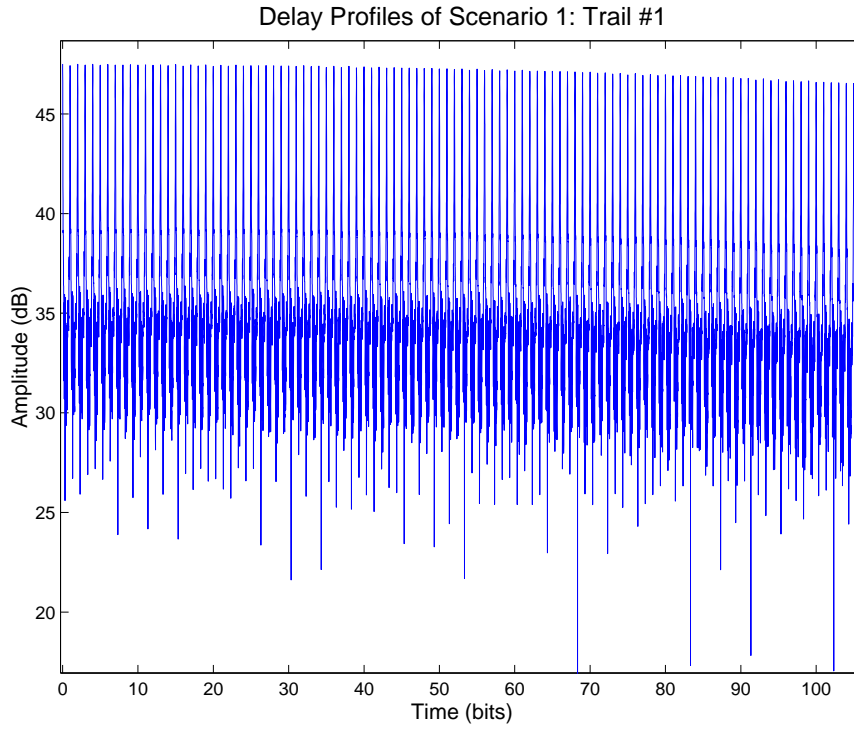


Figure 5.11: Collection of delay profiles for Trial #1

To quantify these variations, the correlation coefficient, r , between the first profile of each snapshot and all other window sizes are calculated as in [49]:

$$r = \frac{C}{\sigma_{y_1} \sigma_{y_a}} \quad (5.26)$$

and

$$C = E(\mathbf{y}_1 - \mu_{y_1})(\mathbf{y}_a - \mu_{y_a}) \quad (5.27)$$

where μ_{y_1}, σ_{y_1} and μ_{y_a}, σ_{y_a} are means and standard deviations of the first and averaged profile, respectively. The results are tabulated in Table 5.19. The values of each column are the correlations between the first profile and the average profile with the corresponding window size for all twenty trials. Most of the correlations are greater than 0.9 because of the similarity of the large main peaks for all delay profiles. Even

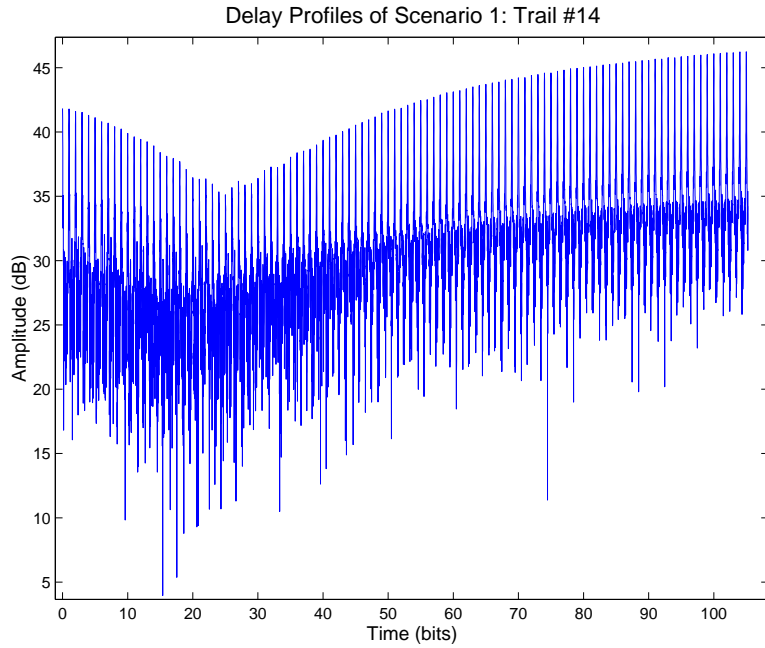


Figure 5.12: Collection of delay profiles for Trial #14

though the lower peaks are varying throughout the snapshot, the correlation values tend to remain high due to the main peaks.

Referring back to the examples of Trial #1 and #14, averaged profiles are plotted on the same axis to illustrate the difference between stable and varying snapshots. Trial #1 is illustrated in Figure 5.13 where the average profile of 100 bits closely resembles the first profile. These two profiles have a correlation value of 0.9943 and other correlations with smaller window sizes in Trial #1 (Table 5.19) have similar values. In contrast, the plot Trial #14 in Figure 5.14 illustrates large variations between averaged profiles with different window sizes. All four averaged profiles are dissimilar except for the time interval near the main peak. Thus their correlation values are lower comparing to those of Trial #1.

<i>1st Profile of Trial #</i>	<i>Average Profile's window size (bits)</i>				
	5	10	25	50	100
1	0.9976	0.9961	0.9952	0.9947	0.9943
2	0.9984	0.9973	0.9973	0.9969	0.9967
3	0.9585	0.9529	0.9522	0.9486	0.9350
4	0.9952	0.9958	0.9958	0.9958	0.9951
5	0.9964	0.9965	0.9962	0.9962	0.9963
6	0.9716	0.9730	0.9729	0.9698	0.9691
7	0.9990	0.9990	0.9998	0.9987	0.9986
8	0.9672	0.9683	0.9675	0.9659	0.9646
9	0.9236	0.9217	0.9244	0.9246	0.9236
10	0.9597	0.9606	0.9614	0.9592	0.9550
11	0.9273	0.9281	0.9172	0.9060	0.9071
12	0.9945	0.9949	0.9956	0.9953	0.9951
13	0.9970	0.9947	0.9937	0.9906	0.9784
14	0.9451	0.9355	0.9053	0.9069	0.9017
15	0.8577	0.8478	0.8351	0.8142	0.7791
16	0.9337	0.9339	0.9303	0.9214	0.9151
17	0.9512	0.9513	0.9418	0.9271	0.9215
18	0.9906	0.9894	0.9895	0.9889	0.9894
19	0.9938	0.9935	0.9940	0.9939	0.9938
20	0.8958	0.9029	0.8710	0.8250	0.7784

Table 5.19: Correlations between single and averaged profiles

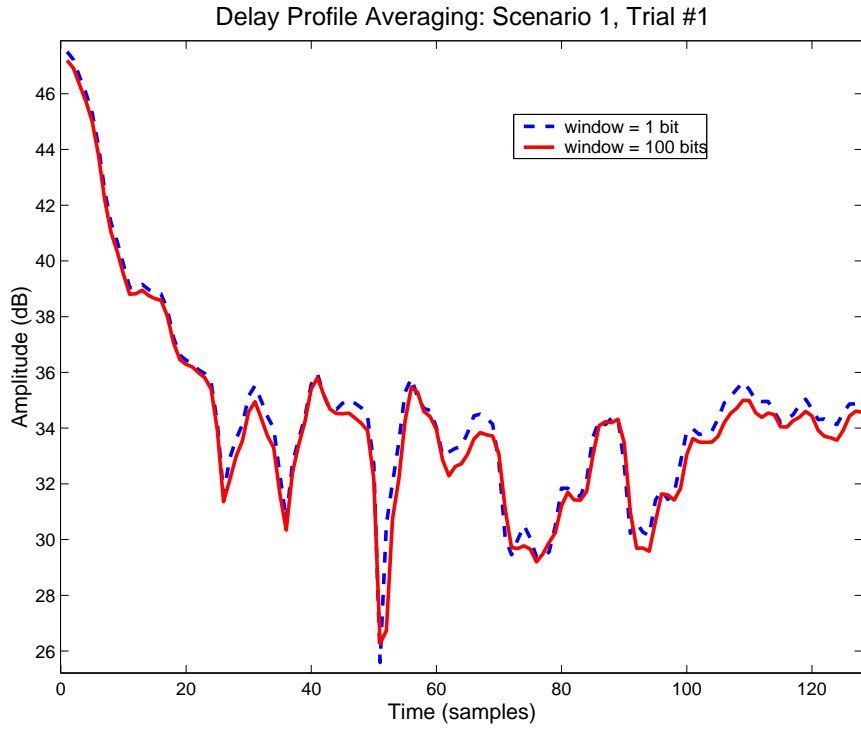


Figure 5.13: Average delay profiles for Trial #1

An interesting comparison to Trial #14 is Trial #15 which has the lowest correlations out of all trials for any window sizes. If we plot the first 5 profiles of Trial #14 (Figure 5.15) and compare it to that of Trial #15 (Figure 5.16) for further examination, we notice that Trial #15 has more irregularities in its profiles. This explains why its correlation value is only at 0.8577 for a window size of 5 bits, whereas Trial #14's correlation is 0.9451. Trial #20 is also similar to Trial #15.

If we consider correlation coefficients of 0.97 or higher as stable, then 50% (ten) of the twenty trials are considered to be stable and the other ten are time varying.

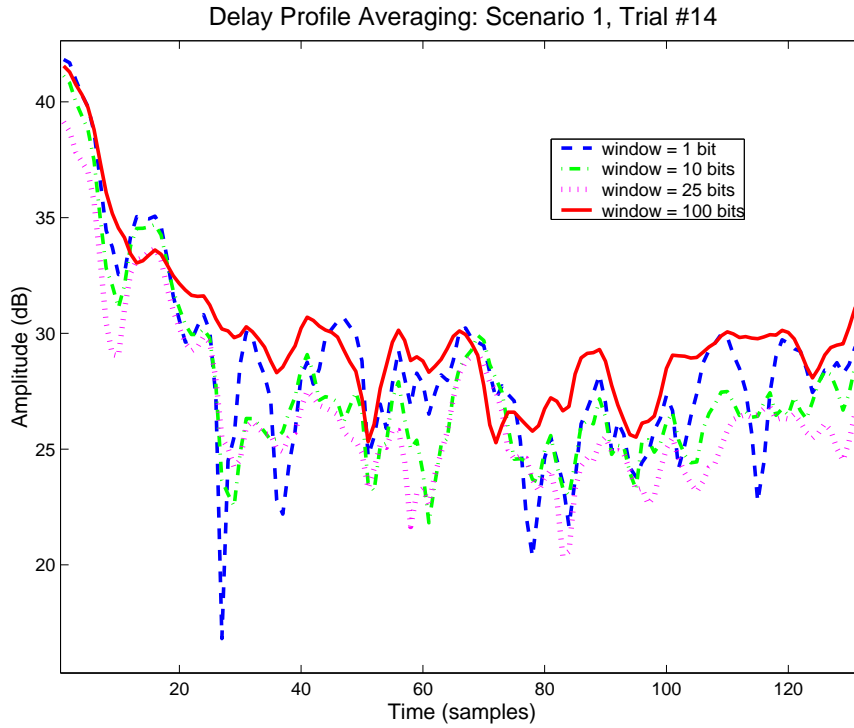


Figure 5.14: Average delay profiles for Trial #14

5.7.2 Multipath Detection of Averaged Profiles

The same multipath detection algorithm introduced in Section 5.4 is applied to averaged profiles. Each snapshot profiles are averaged according to a particular window size. Then, the multipaths are detected under a false alarm probability of 10^{-3} and 1/2-chip smoothing resolution. The results are gathered and presented in Tables 5.20 to 5.29. Tables 5.20 and 5.21 are identical to the case of $P_{FA} = 10^{-3}$ and 1/2-chip resolution with no averaging presented in Section 5.5.

One apparent trend is that the proportion of detected 3rd and 4th multipath peaks decreases as the window size increases from 1 bit to 50 bits. In fact, no 3rd or 4th peak is detected when 50 profiles are averaged together as listed in Table 5.28. On the other hand, the proportion of 2nd peaks detected remains at roughly 35-38% of

the total number of main peaks calculated from Tables 5.20, 5.22, 5.24, 5.26 and 5.28. The stability of 2^{nd} multipath peaks is also demonstrated in the amplitude and delay statistics. The means and variances of the 2^{nd} peak's amplitude and delay remain rather constant for different averaging windows. For example, the mean 2^{nd} *peak* amplitudes fluctuates within 1 dB and mean delays are within 0.3 chips in Tables 5.21, 5.23, 5.25, 5.27 and 5.29.

The investigation of delay profile averaging help us to understand the underlying dynamics of multipath characteristics. If delay profiles are rather stable, then averaging consecutive delay profiles within a trial would hardly change the multipath statistics comparing to those with no averaging. This is true for the case of 2^{nd} multipath peaks. However results also show that the occurrence of 3^{rd} and 4^{th} peaks are greatly reduced, meaning the dynamics of lower and less frequent peaks are removed from averaging. Therefore, we conclude that 3^{rd} and 4^{th} peaks are more dynamic.

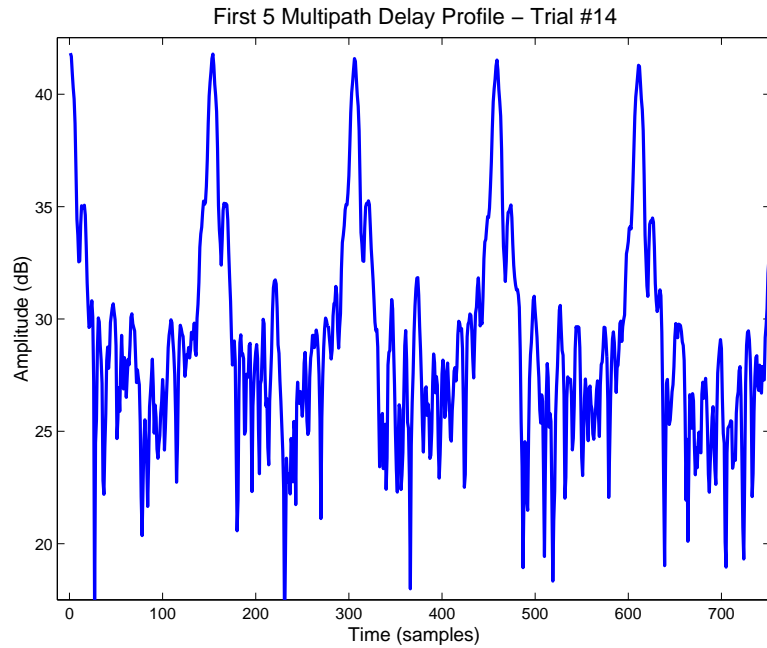


Figure 5.15: First 5 Multipath Delay Profiles: Trial #14

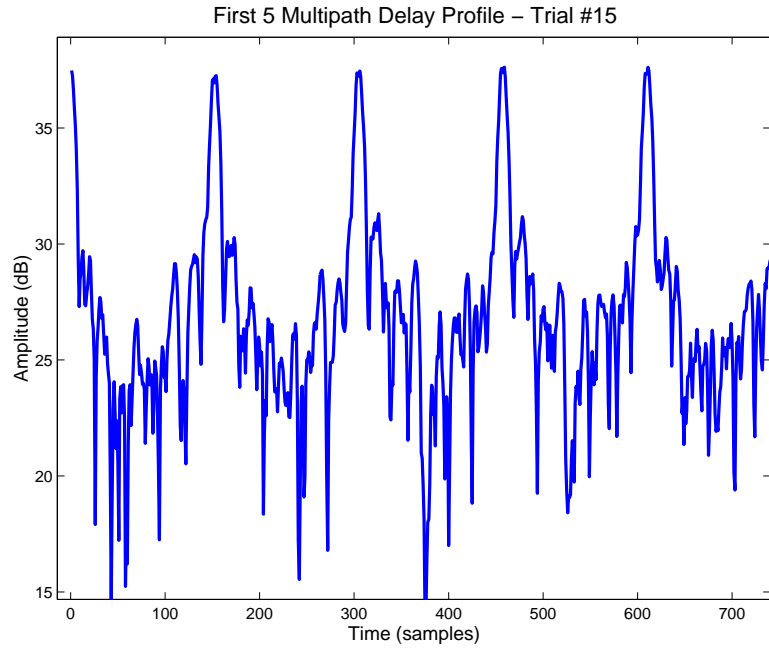


Figure 5.16: First 5 Multipath Delay Profiles: Trial #15

<i>Peak</i>	<i>Number of Occurrence</i>
1 st	2000
2 nd	700
3 rd	124
4 th	7

Table 5.20: Number of multipaths detected: window = 1 bit

<i>Peak</i>	<i>Amplitude (dB)</i>	<i>Var</i>	<i>Mean Delay (chips)</i>	<i>Delay Spread</i>
1 st	-	10.84	-	-
2 nd	-5.67	1.90	2.35	0.38
3 rd	-6.50	2.50	8.23	5.10
4 th	-5.45	5.78	5.81	4.48

Table 5.21: Amplitudes and delays of multipaths: window = 1 bit

<i>Peak</i>	<i>Number of Occurrence</i>
1 st	400
2 nd	149
3 rd	16
4 th	0

Table 5.22: Number of multipaths detected: window = 5 bits

<i>Peak</i>	<i>Amplitude (dB)</i>	<i>Var</i>	<i>Mean Delay (chips)</i>	<i>Delay Spread</i>
1 st	-	11.14	-	-
2 nd	-6.36	2.15	2.51	0.38
3 rd	-7.15	2.44	4.12	0.21
4 th	-	-	-	-

Table 5.23: Amplitudes and delays of multipaths: window = 5 bits

<i>Peak</i>	<i>Number of Occurrence</i>
1 st	200
2 nd	76
3 rd	10
4 th	0

Table 5.24: Number of multipaths detected: window = 10 bits

<i>Peak</i>	<i>Amplitude (dB)</i>	<i>Var</i>	<i>Mean Delay (chips)</i>	<i>Delay Spread</i>
1 st	-	11.17	-	-
2 nd	-6.40	2.08	2.56	0.38
3 rd	-7.42	3.04	3.42	0.99
4 th	-	-	-	-

Table 5.25: Amplitudes and delays of multipaths: window = 10 bits

<i>Peak</i>	<i>Number of Occurrence</i>
1 st	80
2 nd	28
3 rd	2
4 th	0

Table 5.26: Number of multipaths detected: window = 25 bits

<i>Peak</i>	<i>Amplitude (dB)</i>	<i>Var</i>	<i>Mean Delay (chips)</i>	<i>Delay Spread</i>
1 st	-	11.01	-	-
2 nd	-6.50	1.98	2.51	0.34
3 rd	-7.86	0.29	3.43	0.91
4 th	-	-	-	-

Table 5.27: Amplitudes and delays of multipaths: window = 25 bits

<i>Peak</i>	<i>Number of Occurrence</i>
1 st	40
2 nd	15
3 rd	0
4 th	0

Table 5.28: Number of multipaths detected: window = 50 bits

<i>Peak</i>	<i>Amplitude (dB)</i>	<i>Var</i>	<i>Mean Delay (chips)</i>	<i>Delay Spread</i>
1 st	-	10.73	-	-
2 nd	-6.38	1.43	2.48	0.23
3 rd	-	-	-	-
4 th	-	-	-	-

Table 5.29: Amplitudes and delays of multipaths: window = 50 bits

5.8 Summary

In this chapter we studied outdoor multipath characteristics for wideband CDMA. The setup of our first experiment is described. All scenarios are illustrated in Figures 5.1 to 5.4. An investigation on propagation path loss shows that the signals experience a path loss exponent of $n = 2.64$, similar to those in urban environment found in [36].

By formulating a hypothesis test problem for each delay profile, we can detect whether multipath is present. The threshold setting is discussed which involves the estimation of noise variance in a profile. The detection performance of our test is derived and plotted in Figure 5.8. Multipath statistics are gathered for both single and

multiple-user scenarios. The results are presented in Tables 5.9-5.10 for single user and Tables 5.17-5.18 for multiple users. We found that in a multiple-user environment, the simple PN acquisition and tracking does not work if fading and interference are severe. The study of time variation of delay profiles indicates that about 50% of the trials are time varying within the snapshot. The multipath statistics of averaged profiles for 2nd strongest peaks are rather stable in time. However, this stability does not hold for the weaker paths.

Chapter 6

Array Signal Processing and Beamforming

6.1 Introduction

An M -element antenna array can provide a mean power gain of M over white noise, but its performance in mitigating interference depends on the actual received data in a physical environment as well as the design of the receiver's signal processing algorithm [20]. In urban mobile environments, multipath reflections or scattering of the transmitted signal are received by the basestation. The purpose of an antenna array at the basestation is to provide spatial diversity gain, combat multipath fading and suppress interfering signals [2] [72].

In this chapter, we study the spatial characteristics of an outdoor environment using our smart antenna receiver testbed and multiple TXs. The correlation between antenna elements is first investigated. The work by [57] and [74] shows that the amount of scattering in the environment is inversely proportional to fading correlation of the antenna elements. By studying fading correlation, we can determine the amount of scattering in the experimental environment.

The second outdoor experiment setup is introduced in Section 6.3. We then study

the stability of spatial characteristics using beampatterns generated from experimental trials recorded over time in Section 6.4. We introduce a beamforming receiver using signal subspace processing based on the work of [27] in Section 6.5. The method of beamforming weight estimation is first discussed in Section 6.5, then the eigenvalues of the pre-despreading and post-despreading covariance matrices are analyzed. We attempt to estimate the DOA of signals from the i^{th} transmitter using array response vectors \mathbf{a}_i obtained from the pre-despreading and post-despreading covariance matrices. In the last section, we quantify the SINR improvement our beamforming receiver over a single antenna receiver.

6.2 Fading Correlation between Antenna elements

The study of fading correlation provides an estimate of the amount of scattering in the experimental environment. Previous simulation results by [57] and [74] on scattering in mobile environment indicate the amount of angle spread or scattering increases as fading correlation between two antenna elements decreases for a given antenna separation. This is also true with receiving antenna mutual coupling effects considered in [74]. The model used in their calculations assume the basestation antenna has a LOS to the mobile, with local scattering around the mobile producing signals which arrive within a given range of angles [57] [74]. In this section, we examine the fading correlation calculated from our experimental data.

6.2.1 Calculation of Correlations and Results

If we rewrite the received signal as $\mathbf{r}_i = \mathbf{x}_i + j\mathbf{y}_i$, where \mathbf{x}_i is the real part of \mathbf{r}_i and \mathbf{y}_i is the imaginary part of \mathbf{r}_i for the i^{th} antenna. The normalized fading correlation between two antennas is then defined by [57] as:

$$R = \frac{(|R_{x_1x_2}|^2 + |R_{x_1y_2}|^2)^{1/2}}{\sigma_{x_1}\sigma_{x_2}} \quad (6.1)$$

where $R_{x_1x_2}$ is the correlation between the real parts of \mathbf{r}_1 and \mathbf{r}_2 , and $R_{x_1y_2}$ is the correlation between the real part of \mathbf{r}_1 and imaginary part of \mathbf{r}_2 . Using Equation (6.1), the correlations between two antenna elements are calculated. Since the scattering model [57] and [74] used is based on a single signal source, we calculate the fading correlations using single TX scenario data from our first outdoor experiment (see Section 5.2). In order to make a fair comparison to their results, the antenna separation is $1/2 \lambda$. We have three separate sets of data gathered for each user A, B or C transmitting individually in scenario 1 of Figure 5.1.

The fading correlations are first calculated using all the trials, and then only in a subset of the trials with no multipaths present. The results are tabulated in Tables 6.1 and 6.2. The location of the TXs are -60° , 0° and 60° away from normal perpendicular to the broadside of the array for user A, B and C, respectively (Figure 5.1). The multipath detection procedure discussed in Section 5.4 is used to perform multipath detection on each separate trial of each data set. We found that there are 9 trials for user A, 6 trials for user B and 10 trials for user C's data for which have no significant multipaths. The fading correlations in Table 6.2 are calculated from those trials.

All the fading correlations are 0.746 or above which corresponds to a low scattering environment with angle spread of 20° or less, according to results in [57] and [74]. With the multipaths removed, we would expect for higher correlation as is shown in Table 6.2. The correlations are still statistically significantly below 1.0, especially at DOA of 0° , so there is evidence of scattering. The correlation values with multipaths are unfair comparisons to those in [57] and [74] as their scattering model does not include resolvable multipaths. However, Tables 6.1 and 6.2 show that the presence of the resolvable multipath reduces the fading correlation between antenna elements.

<i>Fading Correlation</i>	<i>Direction of Arrival (DOA)</i>		
	-60°	0°	60°
<i>Mean</i>	0.892	0.746	0.772
<i>Variance</i>	0.019	0.056	0.040

Table 6.1: Fading Correlation of two antennas: all trials

<i>Envelope Correlation</i>	<i>Direction of Arrival (DOA)</i>		
	-60°	0°	60°
<i>Mean</i>	0.976	0.837	0.874
<i>Variance</i>	0.002	0.056	0.042

Table 6.2: Fading Correlation of two antennas: trials with no multipaths

6.3 Second Outdoor Experimental setup

The second outdoor experiment is performed at the same location, Walter Light Hall parking lot, as the first experiment described in Section 5.2. This experiment utilizes the full implementation of our testbed: four element array basestation and four portable transmitters. The objective of this experiment is to gather data from a 4-element antenna array RX with multiple TXs transmitting simultaneously. These data are used for analysis in beampatterns, DOA estimations and SINR improvement from beamforming.

The setup for the 4-element array and 4 TXs experiment is shown in Figure 6.1. Four transmitters A, B, C and D are separated spatially as indicated in the figure, and located at equidistant (50 feet) away from the basestation. Each transmitter is assigned a specific 127-chip PN sequence (or user code). The 127-chip sequence is used instead of 31-chip sequence because the SX043 IC can only support up to 3 user codes for sequences with a length of 31 chips [1].

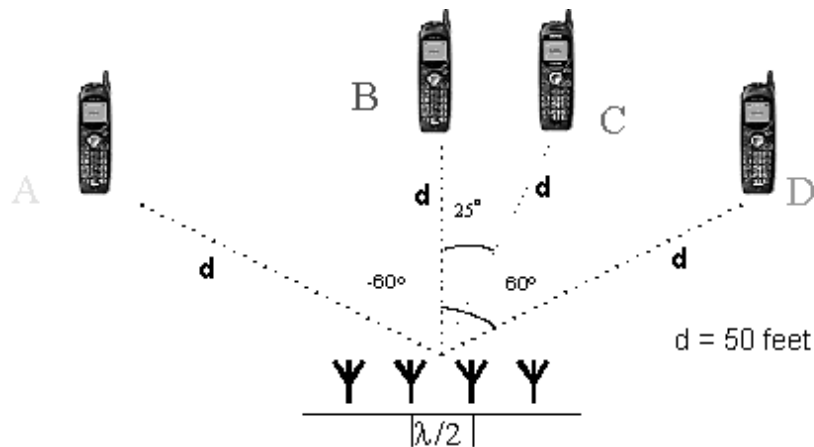


Figure 6.1: Outdoor experiment #2: four TXs and 4-element array

At the time of the experiment, the parking lot is almost vacated except that there is a car located midway in the line-of-sight from TX A to basestation. Otherwise, the testing environment is the same as discussed in Section 5.2. Using the setup in Figure 6.1, a total of 30 snapshots are recorded with four TXs transmitting simultaneously.

6.4 Analysis of Spatial Stability using Beampatterns

Early attempts to localize signal sources using antenna array are through beamsteering techniques [30]. The basic concept is to steer the array in one direction at a time and measure the output power. The DOA estimates are the steering locations which result in maximum output power [30]. The array output at time n , $y_B(n)$, is a linear combination of the complex-valued data (as converted in Chapter 4) at the M elements [68]:

$$y_B(n) = \sum_{i=1}^M w_i^* x_i(n) = \mathbf{w}^H \mathbf{x}(n) \quad (6.2)$$

where $*$ represents complex conjugate, H is Hermitian complex conjugate transpose, $x_i(n)$ is individual element's received data and \mathbf{w} is the $M \times 1$ weight vector. With the samples at the array output $y(1), y(2), \dots, y(N_D)$, the output power is measured by [41]:

$$P_B(\mathbf{w}) = \sum_{n=1}^{N_D} |y_B(n)|^2 = \sum_{n=1}^{N_D} |\mathbf{w}^H \mathbf{x}(n)|^2 \quad (6.3)$$

6.4.1 Beampatterns of Experiment Scenarios

In order to analyze the stability of spatial signatures in the outdoor environment, we let weight vectors as $\mathbf{w}(\theta) = \mathbf{a}(\theta)$, where $\mathbf{a}(\theta)$ is the array response vector corresponds to the ULA of our testbed in Equation (2.1). By varying θ over -80° to $+80^\circ$ range for $\mathbf{w}(\theta)$ in Equation (6.3) and calculating $P_B(\mathbf{w}(\theta))$ for all angles, a beampattern is created [68]. A beampattern is the array output power as a function of steering angles. The location of the peak in $P_B(\mathbf{w}(\theta))$ is the DOA estimate of the incoming signal.

The beampattens of all the trials generated from received data are plotted on the same axis. If the beampatterns of all the trials are similar, i.e. if the plots are aligned, then the spatial signature is stable. The beampatterns for the scenario in Figure 6.1 are plotted in Figure 6.2. Since all four TXs from different directions are transmitting simultaneously, we would expect higher peaks in the beampattern at several DOAs as seen in Figure 6.2.

If the received data is despread with the PN code of user i , then $\mathbf{x}(n)$ in Equation (6.2) is replaced with the despread sequence $\mathbf{y}_{b_i}(n)$ and beampatterns are generated for each user corresponding to the strongest path. The beampatterns of despread

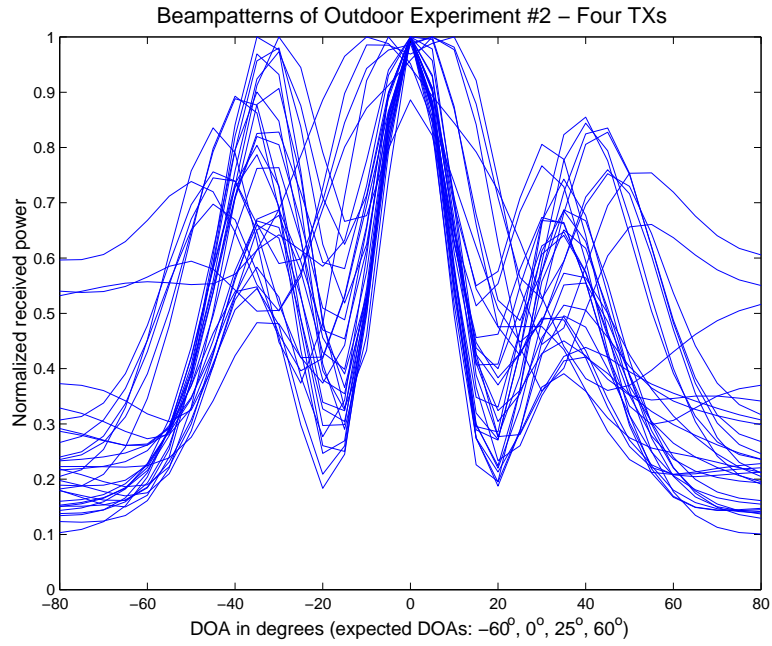


Figure 6.2: Beampattern of Outdoor Experiment #2: Scenario 1

outputs, $y_{b_i}(n)$ for users A, B, C and D are shown in Figures 6.3 to 6.6, respectively. The beampatterns for user B and D have strong peaks around DOA of 0° and the beampatterns vary relatively less from trial to trial. This suggests the DOAs of the strongest path for user B and D are rather stable. On the contrary, Figures 6.3 and 6.5 show varying beampatterns throughout different trials, hence, they suggest that the strongest paths arrive at the antenna array with various angles for user A and C.

The beampatterns in Figures 6.3 to 6.6 demonstrated that even if transmitters are stationary, other factors in the environment such as cars, human bypassers, building and backyard fences can cause variations in the beampattern. This observation provides insights on our beamforming results discussed in the next section.

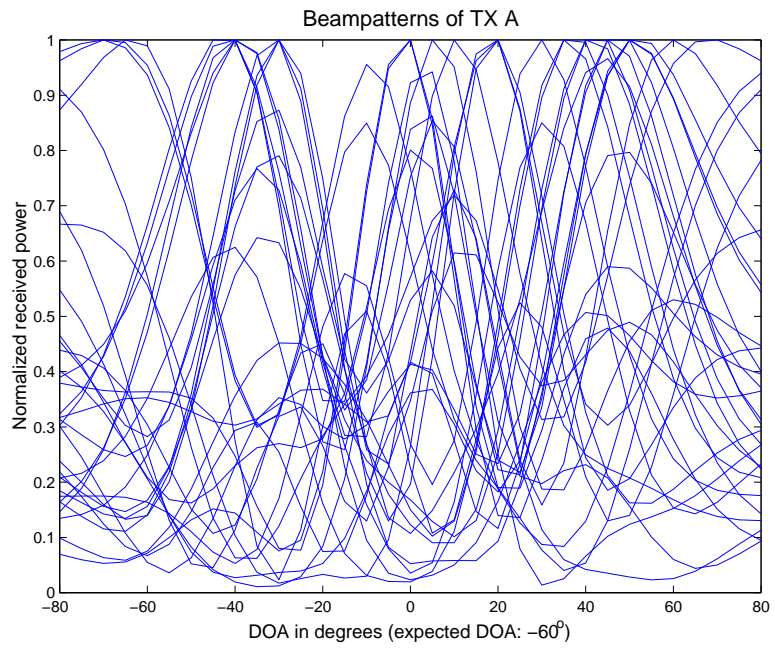


Figure 6.3: Beampatterns of despread sequences: User A

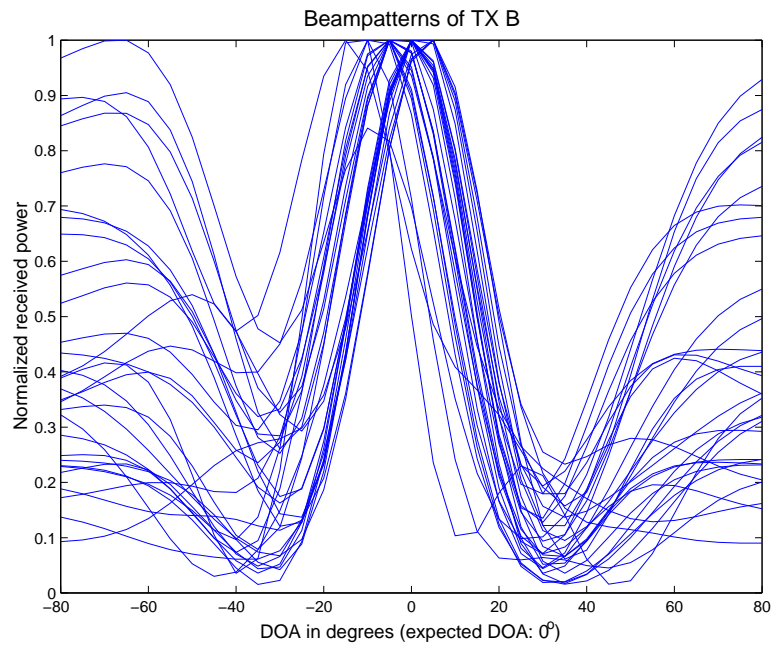


Figure 6.4: Beampatterns of despread sequences: User B

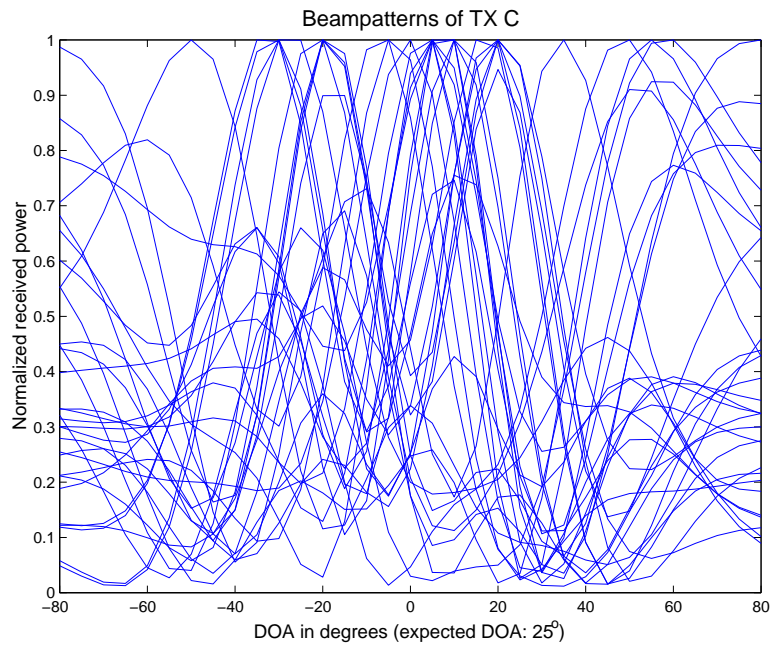


Figure 6.5: Beampatterns of despread sequences: User C

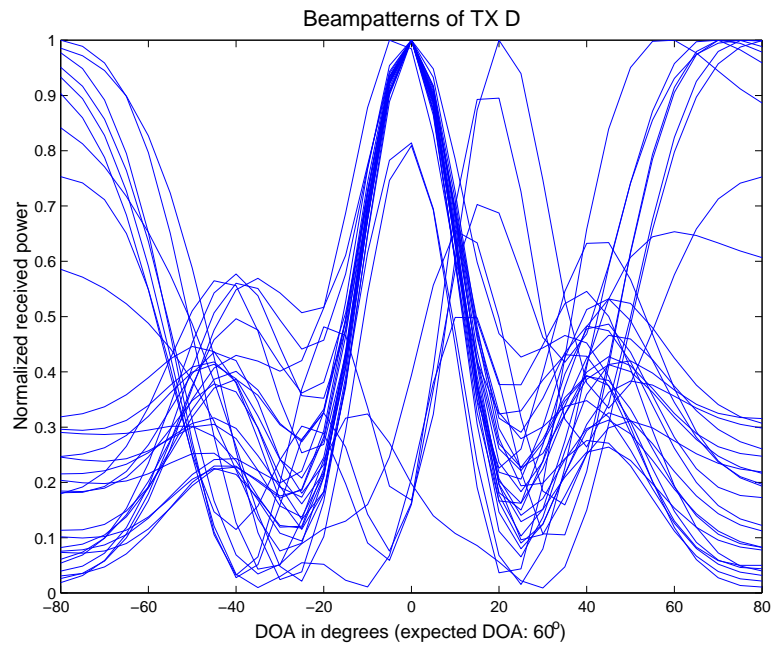


Figure 6.6: Beampatterns of despread sequences: User D

6.5 Beamforming Receiver using Signal Subspace Processing

The received signal model used for beamforming analysis in this section is based on the work of Karimi and Blostein [27] [28]. In fact the model is the same as in [27] except that the PN code, $\tilde{c}(n) = c^I(n) + jc^Q(n)$, in our receiver is complex instead of scalar. The equations for beamforming weight estimation and SINR calculations discussed in this section are also derived following the steps outlined in [27]. The use of complex PN code sequence in our derivations yields equations that are only differ by constant factors. Therefore details of the derivation are omitted with reference to [27] and [28] and any differences are point out as necessary throughout this section.

6.5.1 Beamforming Receiver

In the 4-element basestation receiving array and four TXs described in Section 6.3, the overall received signal using complex bandpass representation can be written as follows [27]:

$$\mathbf{x}(n) = (x_1(n), x_2(n), \dots, x_M(n))^T \quad (6.4)$$

$$= \sum_{i=1}^N \sqrt{2P_i} \mathbf{a}_i s_i(n - \tau_i) e^{j\theta_i} + \mathbf{v}(n) \quad (6.5)$$

where P_i is the total received power from user i of all elements, \mathbf{a}_i represents the array response vector, $s_i(n) = \tilde{c}_i(n)$ is the PN code assigned to user i , M is the number of antenna element and N is the number of users. The time and phase shifts for the i^{th} user are given by τ_i and θ_i respectively. The thermal noise vector is given by $\mathbf{v}(n)$, and can also be written as:

$$\mathbf{v}(n) = (v^{(1)}(n), v^{(2)}(n), \dots, v^{(M)}(n))^T \quad (6.6)$$

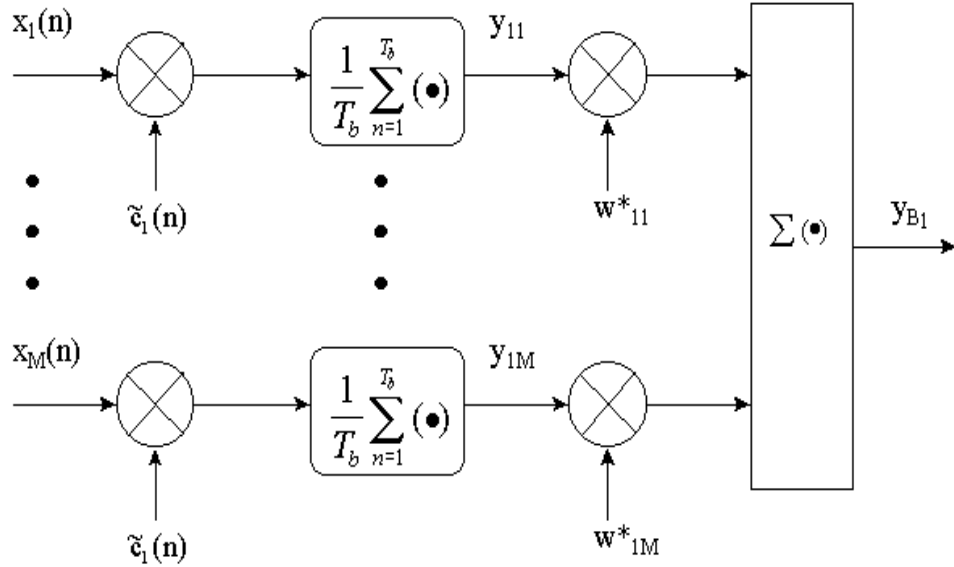


Figure 6.7: Beamforming Receiver for the Desired User

where $v^{(1)}(n)$ to $v^{(M)}(n)$ represent independent white Gaussian noise terms with zero means for all elements. Since $E(\mathbf{v}(n)) = 0$, the covariance matrix of $\mathbf{v}(n)$ is given by

$$E(\mathbf{v}(n)\mathbf{v}^*(n)) = 2N_o I_M \quad (6.7)$$

where N_o is the noise spectral density and I_M is the $M \times M$ identity matrix. The optimum beamforming receiver is shown in Figure 6.7. The received signal $\mathbf{x}(n)$ is first despread with the desired user's sequence $\tilde{c}_i(n)$ and then it is multiplied by the $M \times 1$ complex beamforming weight vector \mathbf{w}_i , where $\mathbf{w}_i = [w_{i1} \ w_{i2} \dots \ w_{iM}]^T$. The PN code for user i has a period of T_b and a chip duration T_c . The cross-correlation of the PN codes is given as

$$E(\tilde{c}_i(n - \tau_i)\tilde{c}_j(n)) = 0 \quad \text{for } i \neq j \quad (6.8)$$

and the autocorrelation is differ by a factor of 2 to the scaler case used in [27], i.e., $E(\tilde{c}_i(n)\tilde{c}_i^*(n)) = 2$. Without loss of generality, we assume the desired user is $i = 1$,

and τ_1 and θ_1 are assumed to be zero. Then from (6.5) can be written as

$$\mathbf{x}(n) = \sqrt{2P_1}\tilde{c}_1(n)\mathbf{a}_1 + \sum_{i=2}^N \sqrt{2P_i}\mathbf{a}_i\tilde{c}_i(n - \tau_i)e^{j\theta_i} + \mathbf{v}(n) \quad (6.9)$$

The input vector to user 1's beamforming network $\mathbf{y}_1 = (y_{11}, y_{12}, \dots, y_{1M})^T$ is given by

$$\mathbf{y}_1 = \frac{1}{T_b} \sum_{n=1}^{T_b} \mathbf{x}(n)\tilde{c}_1(n) \quad (6.10)$$

Combining (6.9) and (6.10), it can be shown that \mathbf{y}_1 can be expressed as [27]:

$$\begin{aligned} \mathbf{y}_1 &= 2\sqrt{2P_1}\mathbf{a}_1 + \sum_{i=2}^N \frac{1}{T_b} \sum_{n=1}^{T_b} \sqrt{2P_i}\tilde{c}_i(n - \tau_i)\tilde{c}_1(n)\mathbf{a}_i e^{j\theta_i} \\ &\quad + \frac{1}{T_b} \sum_{n=1}^{T_b} \mathbf{v}(n)\tilde{c}_1(n) \\ &= 2\sqrt{2P_1}\mathbf{a}_1 + \mathbf{N}\mathbf{I} \end{aligned} \quad (6.11)$$

$$(6.12)$$

where $\mathbf{N}\mathbf{I}$ represents the multiple access interference (MAI) and thermal noise terms. As seen from Figure 6.7, beamformer weights are applied after the signal is despread [27]. The beamformer's output, y_{B_1} , can be written as

$$y_{B_1} = \mathbf{w}_1^H \mathbf{y}_1 \quad (6.13)$$

$$= 2\sqrt{2P_1}\mathbf{w}_1^H \mathbf{a}_1 + \mathbf{w}_1^H \mathbf{N}\mathbf{I} \quad (6.14)$$

using Equation (6.11). The SINR is then determined as

$$SINR = \frac{E|2\sqrt{2P_1}\mathbf{w}_1^H \mathbf{a}_1|^2}{E|\mathbf{w}_1^H \mathbf{N}\mathbf{I}|^2} = \frac{8P_1|\mathbf{w}_1^H \mathbf{a}_1|^2}{\mathbf{w}_1^H \mathbf{R}_{\mathbf{N}\mathbf{I}} \mathbf{w}_1} \quad (6.15)$$

where $\mathbf{R}_{\mathbf{NI}}$ is the covariance matrix of the combined noise-interference term \mathbf{NI} given by

$$\mathbf{R}_{\mathbf{NI}} = E(\mathbf{NIN}^{\mathbf{H}}) \quad (6.16)$$

In order to implement beamforming, the receiver should be able to estimate the array response vector of the desired signal \mathbf{a}_1 , the noise-interference covariance matrix, $\mathbf{R}_{\mathbf{NI}}$, and the received power of the desired user, P_1 [27]. In Section 6.5.2, the algorithm introduced in [27] is used to determine these variables. However in [27] the received power P_1 is assumed known, whereas in our system P_1 is unknown and must be estimated as suggested in [28].

6.5.2 Estimating Beamforming Weights

The pre-despreading and post-despreading signal covariance matrices in the receiver are used to estimate the array response vectors \mathbf{a}_i , noise-interference covariance matrices \mathbf{NI} , and the received powers P_i . From (6.7), (6.8) and (6.9), and since $E[\mathbf{x}] = 0$, the pre-despreading matrix can be shown to be [27]:

$$\mathbf{R}_{\mathbf{xx}} = E[\mathbf{xx}^{\mathbf{H}}] \quad (6.17)$$

$$= 2 \left(2P_1 \mathbf{a}_1 \mathbf{a}_1^{\mathbf{H}} + \sum_{i=2}^N 2P_i \mathbf{a}_i \mathbf{a}_i^{\mathbf{H}} + N_o \mathbf{I}_{\mathbf{M}} \right) \quad (6.18)$$

where PN chips and noise samples are independent. From Equation (6.11) and since $E[\mathbf{y}_1] = 0$, the post-despreading matrix $\mathbf{R}_{\mathbf{y}_1 \mathbf{y}_1}$ for the desired user $i = 1$ can also be shown to be [27]:

$$\mathbf{R}_{\mathbf{y}_1 \mathbf{y}_1} = E[\mathbf{y}_1 \mathbf{y}_1^{\mathbf{H}}] \quad (6.19)$$

$$= 2 \left(4P_1 \mathbf{a}_1 \mathbf{a}_1^{\mathbf{H}} + \frac{4T_c}{3T_b} \sum_{i=2}^N 2P_i \mathbf{a}_i \mathbf{a}_i^{\mathbf{H}} + \frac{2}{T_b} N_o \mathbf{I}_{\mathbf{M}} \right) \quad (6.20)$$

where the variance of multiple access interference (MAI) is derived using method in [27] where the PN chip waveforms are assumed rectangular. It can be shown $E(MAI^2)$ is equal to

$$E(MAI^2) = \frac{4T_c}{3T_b} \sum_{i=2}^N 2P_i |\mathbf{w}_1^H \mathbf{a}_i|^2 \quad (6.21)$$

at the beamformer's output, and the MAI term in (6.20) is derived using (6.21).

Since adequate data is available to obtain estimates for the covariance matrices, $\mathbf{R}_{\mathbf{x}\mathbf{x}}$ and $\mathbf{R}_{\mathbf{y}_1\mathbf{y}_1}$ in (6.18) and (6.20) can be combined in the following to remove the MAI terms:

$$\hat{\mathbf{R}}_{\mathbf{S}\mathbf{N}} = T_b \hat{\mathbf{R}}_{\mathbf{y}_1\mathbf{y}_1} - \frac{4}{3} T_c \hat{\mathbf{R}}_{\mathbf{x}\mathbf{x}} \quad (6.22)$$

$$= 2P_1 \mathbf{a}_1 \mathbf{a}_1^H (2T_b - \frac{4}{3} T_c) + 2N_o (1 - \frac{2}{3} T_c) \mathbf{I}_M \quad (6.23)$$

where $\hat{\mathbf{R}}_{\mathbf{y}_1\mathbf{y}_1}$ and $\hat{\mathbf{R}}_{\mathbf{x}\mathbf{x}}$ are the estimated covariance matrices from measurement data. For simplicity, Equation 6.23 can rewrite as

$$\hat{\mathbf{R}}_{\mathbf{S}\mathbf{N}} = \mathbf{B} + k \mathbf{I}_M \quad (6.24)$$

and $\mathbf{B} = 2P_1 \mathbf{a}_1 \mathbf{a}_1^H (2T_b - \frac{4}{3} T_c)$ and $k = 2N_o (1 - \frac{2}{3} T_c)$. Where \mathbf{B} is a rank 1 matrix with eigenvector $e_1 = \mathbf{a}_1$ and eigenvalue

$$d_1 = 2P_1 (2T_b - \frac{4}{3} T_c). \quad (6.25)$$

It is shown in [27] that the eigenvectors resulted from the eigen-decompositions of $\hat{\mathbf{R}}_{\mathbf{S}\mathbf{N}}$ and \mathbf{B} are the same. Hence by estimating the eigenvector and eigenvalue of $\hat{\mathbf{R}}_{\mathbf{S}\mathbf{N}}$, the eigenvector and eigenvalue of $2P_1 \mathbf{a}_1 \mathbf{a}_1^H (2T_b - \frac{4}{3} T_c)$ can be obtained [27]. Then, Equation (6.23) can be estimated by the following approximation:

$$\hat{\mathbf{R}}_{\mathbf{S}\mathbf{N}} = T_b \hat{\mathbf{R}}_{\mathbf{y}_1\mathbf{y}_1} - \frac{4}{3} T_c \hat{\mathbf{R}}_{\mathbf{x}\mathbf{x}} = \sum_{i=1}^M \lambda_i e_i e_i^H \approx \lambda_1 e_1 e_1^H \quad (6.26)$$

where λ_1 is the largest eigenvalue and $e_1 = \mathbf{a}_1$ is the corresponding eigenvector. Since $\lambda_1 = d_1 + k$ with k from (6.24) and d_1 from (6.25), the received power P_1 of the M-elements can be estimated as suggested by [28]:

$$P_1 = \frac{\lambda_1 - k}{2(2T_b - \frac{4}{3}T_c)} \quad (6.27)$$

$$= \frac{\lambda_1 - 2N_o(1 - \frac{2}{3}T_c)}{2(2T_b - \frac{4}{3}T_c)} \quad (6.28)$$

where k is estimated as the average of the three smallest eigenvalues λ_2 , λ_3 and λ_4 . The post-despreading interference-noise covariance matrix \mathbf{R}_{NI} is calculated from (6.20) as

$$\mathbf{R}_{\text{NI}} = \mathbf{R}_{\mathbf{y}_1\mathbf{y}_1} - 8P_1\mathbf{a}_1\mathbf{a}_1^{\text{H}} \quad (6.29)$$

Since we can estimate \mathbf{a}_1 , P_1 and \mathbf{R}_{NI} from Equations (6.26), (6.28) and (6.29), the beamforming weights \mathbf{w}_1 and SINR of our experimental data can be calculated for analysis in Section 6.6. An alternative for estimation of P_1 is to assume the noise density, N_o as the noise floor of the receiver chain, and then calculate P_1 using substituting N_o into (6.28). This approach is also used in Section 6.6 as a comparison for the estimation of k and P_1 in (6.27).

6.5.3 Eigenvalues Analysis and Results

If we solve for the eigenvalues of the pre-despreading and post-despreading sample covariance matrices described in the previous section and arrange them in ascending order, we would expect that the eigenvalues of $\hat{\mathbf{R}}_{\text{xx}}$ will be closer in magnitude than those of $\hat{\mathbf{R}}_{\text{SN}}$ because the desired signal in $\hat{\mathbf{R}}_{\text{SN}}$ has benefit from a gain through despreading.

The eigenvalues of the pre-despreading covariance matrices are obtained from the eigen-decomposition of $\hat{\mathbf{R}}_{\mathbf{xx}}$ [30]:

$$\hat{\mathbf{R}}_{\mathbf{xx}} = \frac{1}{N_D} \sum_{n=1}^{N_D} \mathbf{x}(n)\mathbf{x}^H(n) = \sum_{i=1}^M \xi_i u_i u_i^H \quad (6.30)$$

where N_D is the number of samples points, M is the number of array elements, ξ_i are the eigenvalues of $\hat{\mathbf{R}}_{\mathbf{xx}}$. The eigenvalues calculated from the pre-despreading sample covariance matrix is tabulated in Table 6.3. We observe that ξ_1 for all trials is about twice to ten times larger than ξ_2 , suggesting there is a direction which the received signal is slightly stronger in each trial. Since the values of ξ_4 , ξ_3 and ξ_2 are not significantly smaller than ξ_1 , we conclude that the combined effect of noise and interference are captured in those three smallest eigenvalues.

The eigenvalues of post-despreading sample covariance matrices, $\hat{\mathbf{R}}_{\mathbf{SN}}$, in Equation (6.26) are calculated for all users by despreading with their corresponding PN code. The results are tabulated, again in ascending order, in Tables 6.4 to 6.7 for users A to D, respectively. The largest eigenvalues λ_1 of user B, C and D are about 2 orders of magnitude stronger than the three smallest ones in most trials. However, the second largest eigenvalues λ_2 of user A are only an order of magnitude smaller than λ_1 in about half of the trials. This can be explained by the fact that TX A has an obstacle in its line-of-sight to the basestation. In addition, λ_2 is relatively higher for user A probably because signals from TX A experience more multipaths created by the obstacle and surrounding environment. In general, the three smallest eigenvalues are much smaller than λ_1 for $\hat{\mathbf{R}}_{\mathbf{SN}}$. This indicates that the subtraction of the two covariance matrices in (6.23) is successful in removing MAI for most cases. This justifies the estimation of noise term $k = 2N_o(1 - \frac{2}{3}T_c)$ in Equation (6.24) by the average value of λ_2 , λ_3 and λ_4 .

<i>Eigenvalues of Trial #</i>	<i>Eigenvalues in ascending order (x 10⁶)</i>			
	ξ_4	ξ_3	ξ_2	ξ_1
1	0.1018	0.6157	1.1440	2.3039
2	0.2466	0.5209	0.6667	2.6667
3	0.2206	0.6314	0.8704	1.8228
4	0.1522	0.4998	1.0095	3.2271
5	0.1546	0.4320	1.0546	3.1839
6	0.0473	0.1798	0.3873	3.6139
7	0.0076	0.1524	0.8311	2.1033
8	0.0473	0.2846	0.5610	3.7250
9	0.0776	0.2711	0.4667	3.2586
10	0.2530	0.4176	0.6060	2.4114
11	0.1636	0.3155	0.4830	3.0139
12	0.2481	0.4522	0.6297	2.8689
13	0.0281	0.1299	1.0833	2.5241
14	0.1309	0.3653	0.7646	2.3727
15	0.0566	0.2387	0.4928	3.5627
16	0.1154	0.2178	0.9987	2.0655
17	0.1462	0.5342	0.9041	2.8212
18	0.1017	0.1537	0.3535	3.5579
19	0.0834	0.2581	0.5392	2.6467
20	0.0375	0.1257	0.4439	3.2156
21	0.0689	0.3345	0.4748	2.3519
22	0.1020	0.3736	0.5401	2.2152
23	0.1104	0.1585	0.3892	3.2608
24	0.0877	0.1379	0.5360	2.3824
25	0.0815	0.2073	0.4680	3.2722
26	0.0277	0.2480	0.3232	3.6172
27	0.1063	0.3258	0.5237	2.3334
28	0.0744	0.1460	0.5177	2.6293
29	0.0725	0.1264	0.3787	3.4682
30	0.0597	0.3342	0.5381	2.2635

Table 6.3: Eigenvalues of Pre-despreading Covariance Matrices

<i>Eigenvalues of Trial #</i>	<i>Eigenvalues in ascending order (x 10⁸)</i>			
	λ_4	λ_3	λ_2	λ_1
1	0.0019	0.0382	0.0983	2.7017
2	0.0013	0.0211	0.1079	2.7899
3	0.0042	0.0287	0.0886	2.2912
4	0.0130	0.0293	0.0975	1.8581
5	0.0032	0.0325	0.1614	2.1376
6	0.0002	0.0157	0.0278	2.9830
7	0.0024	0.0044	0.0993	2.2447
8	0.0017	0.0210	0.0821	2.1778
9	0.0014	0.0147	0.1430	2.3411
10	0.0138	0.0297	0.0802	2.1592
11	0.0120	0.0151	0.1357	2.2184
12	0.0122	0.0150	0.1067	1.9107
13	0.0010	0.0087	0.0951	2.3655
14	0.0175	0.0319	0.0927	2.2270
15	0.0035	0.0140	0.0387	2.8728
16	0.0067	0.0299	0.0582	2.8817
17	0.0046	0.0276	0.0809	2.6663
18	0.0006	0.0101	0.1542	1.9213
19	0.0067	0.0129	0.1460	2.3825
20	0.0011	0.0063	0.0488	1.8072
21	0.0022	0.0146	0.1282	1.7226
22	0.0037	0.0195	0.0384	2.0622
23	0.0033	0.0111	0.1342	2.3596
24	0.0050	0.0101	0.0348	1.7818
25	0.0090	0.0093	0.1062	2.8616
26	0.0040	0.0328	0.0655	2.5892
27	0.0062	0.0168	0.0220	1.4030
28	0.0055	0.0115	0.1370	2.2627
29	0.0057	0.0118	0.1725	2.4924
30	0.0110	0.0234	0.0877	1.8126

Table 6.4: Eigenvalues of Covariance matrices with despreading of User A

<i>Eigenvalues of Trial #</i>	<i>Eigenvalues in ascending order (x 10⁹)</i>			
	λ_4	λ_3	λ_2	λ_1
1	0.0005	0.0041	0.0075	2.0133
2	0.0006	0.0033	0.0046	1.0343
3	0.0013	0.0026	0.0078	0.2196
4	0.0004	0.0035	0.0046	1.6661
5	0.0015	0.0050	0.0142	1.5656
6	0.0003	0.0020	0.0021	2.3503
7	0.0007	0.0035	0.0052	1.4022
8	0.0002	0.0008	0.0191	2.4275
9	0.0027	0.0033	0.0078	1.8853
10	0.0011	0.0026	0.0085	0.6578
11	0.0006	0.0010	0.0034	1.6256
12	0.0018	0.0025	0.0053	1.1366
13	0.0001	0.0034	0.0080	1.9559
14	0.0004	0.0034	0.0049	2.0938
15	0.0006	0.0013	0.0040	1.9731
16	0.0001	0.0005	0.0041	1.7411
17	0.0014	0.0034	0.0056	2.3345
18	0.0006	0.0010	0.0023	2.5991
19	0.0009	0.0020	0.0050	1.1408
20	0.0001	0.0020	0.0027	1.8090
21	0.0010	0.0016	0.0049	0.7527
22	0.0006	0.0018	0.0053	0.6287
23	0.0009	0.0012	0.0023	2.4652
24	0.0004	0.0010	0.0042	1.0707
25	0.0009	0.0020	0.0047	1.9283
26	0.0002	0.0012	0.0014	2.1937
27	0.0011	0.0012	0.0064	0.7623
28	0.0002	0.0018	0.0044	1.2707
29	0.0004	0.0006	0.0032	2.1385
30	0.0014	0.0015	0.0110	0.5301

Table 6.5: Eigenvalues of Covariance matrices with despreading of User B

<i>Eigenvalues of Trial #</i>	<i>Eigenvalues in ascending order (x 10⁹)</i>			
	λ_4	λ_3	λ_2	λ_1
1	0.0020	0.0027	0.0066	1.0512
2	0.0009	0.0021	0.0114	0.3722
3	0.0029	0.0033	0.0068	0.9824
4	0.0003	0.0010	0.0050	0.3353
5	0.0004	0.0006	0.0025	0.4221
6	0.0007	0.0007	0.0085	0.2366
7	0.0003	0.0021	0.0087	0.3086
8	0.0001	0.0027	0.0067	0.2574
9	0.0004	0.0016	0.0051	0.2823
10	0.0008	0.0026	0.0082	0.6629
11	0.0002	0.0019	0.0159	0.3855
12	0.0008	0.0026	0.0134	0.4963
13	0.0001	0.0008	0.0124	0.6659
14	0.0019	0.0022	0.0103	0.5623
15	0.0001	0.0012	0.0116	0.2669
16	0.0003	0.0022	0.0089	0.7319
17	0.0009	0.0020	0.0146	0.5989
18	0.0002	0.0011	0.0160	0.3473
19	0.0004	0.0021	0.0066	0.3016
20	0.0004	0.0037	0.0069	0.2117
21	0.0003	0.0022	0.0072	0.3873
22	0.0001	0.0015	0.0051	0.5600
23	0.0008	0.0009	0.0121	0.3286
24	0.0003	0.0025	0.0060	0.3271
25	0.0000	0.0022	0.0057	0.4014
26	0.0002	0.0009	0.0048	0.3618
27	0.0005	0.0021	0.0126	0.3985
28	0.0007	0.0013	0.0110	0.3274
29	0.0003	0.0018	0.0066	0.3305
30	0.0004	0.0022	0.0102	0.3987

Table 6.6: Eigenvalues of Covariance matrices with despreading of User C

<i>Eigenvalues of Trial #</i>	<i>Eigenvalues in ascending order (x 10⁹)</i>			
	λ_4	λ_3	λ_2	λ_1
1	0.0003	0.0035	0.0079	0.7333
2	0.0018	0.0021	0.0134	1.9519
3	0.0020	0.0029	0.0035	1.4841
4	0.0044	0.0047	0.0144	2.1299
5	0.0025	0.0033	0.0139	2.1320
6	0.0017	0.0023	0.0088	1.4629
7	0.0006	0.0016	0.0021	1.1929
8	0.0017	0.0030	0.0113	2.0691
9	0.0024	0.0039	0.0091	1.8562
10	0.0009	0.0030	0.0151	2.1047
11	0.0001	0.0021	0.0148	2.0468
12	0.0006	0.0006	0.0027	2.1646
13	0.0001	0.0051	0.0064	1.3549
14	0.0024	0.0028	0.0042	0.7557
15	0.0018	0.0033	0.0102	2.2510
16	0.0009	0.0042	0.0061	0.7421
17	0.0005	0.0021	0.0075	1.2626
18	0.0003	0.0010	0.0027	1.1996
19	0.0004	0.0007	0.0165	2.3134
20	0.0005	0.0050	0.0164	1.9913
21	0.0004	0.0025	0.0163	2.1888
22	0.0005	0.0023	0.0027	2.2026
23	0.0001	0.0028	0.0092	0.9848
24	0.0005	0.0017	0.0087	1.7721
25	0.0008	0.0037	0.0169	2.5007
26	0.0013	0.0052	0.0117	1.9089
27	0.0002	0.0020	0.0177	2.1037
28	0.0005	0.0015	0.0104	1.8137
29	0.0006	0.0008	0.0170	2.4808
30	0.0002	0.0014	0.0108	2.2639

Table 6.7: Eigenvalues of Covariance matrices with despreading of User D

6.5.4 DOA estimation

In Section 6.5.2 the array response vector, \mathbf{a}_i , is estimated as the eigenvector corresponds to the largest eigenvalue of signal and noise covariance matrix $\hat{\mathbf{R}}_{\text{SN}}$. In this section, we attempt to estimate the direction-of-arrival (DOA) of the desired signal based on the \mathbf{a}_i estimate. Even though the signal subspace beamforming method introduced in Section 6.5.2 does not require the knowledge of DOA, our motivation in this section is to explore the possibility of DOA estimation from \mathbf{a}_i which may be useful in other applications.

The array response of a 4-element ULA is given by (2.1) in Chapter 2 as

$$\mathbf{a}_i(\theta) = \begin{bmatrix} 1 & e^{-j2\pi d \sin \theta/\lambda} & e^{-j4\pi d \sin \theta/\lambda} & e^{-j6\pi d \sin \theta/\lambda} \end{bmatrix}^T \quad (6.31)$$

From the \mathbf{a}_i estimate, DOA is calculated by solving for θ in (6.31) for user i since d and λ are also known. As we have seen from Figures 6.3 to 6.6, the beampatterns can vary from trial to trial. Therefore, we estimate the DOAs from \mathbf{a}_i separately for each trial for user i , and then all DOA estimates from thirty trials are tabulated in histograms shown in Figures 6.8 to 6.11 for user A, B, C and D, respectively.

The DOA estimates for all four users are within the range from -40 degrees to 40 degrees. Thus, these DOAs are not the true location of the four TXs described in Figure 6.1, and the use of \mathbf{a}_i for true DOA estimation cannot be achieved by this method. However, the histograms of user B and D are indicate that in the majority of the time, the signal arrives in one or two directions. In fact, this is also visible from their beampatterns in Figures 6.4 and 6.6, which are also more stable as compared to those of users A and C.

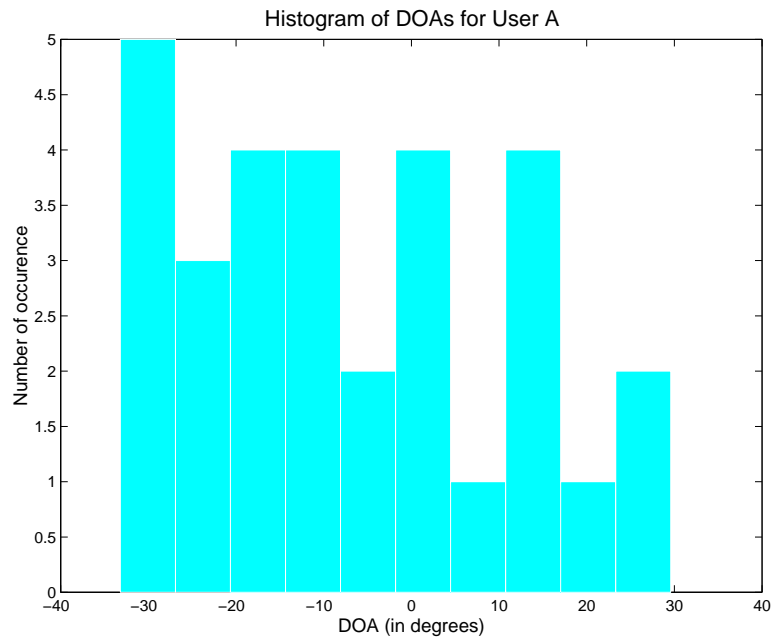


Figure 6.8: Histogram of DOA estimates: User A

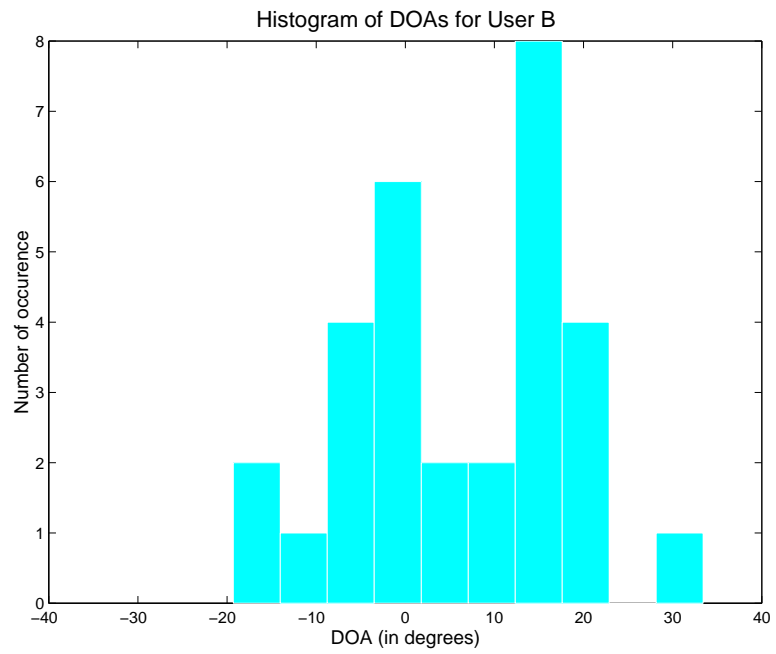


Figure 6.9: Histogram of DOA estimates: User B

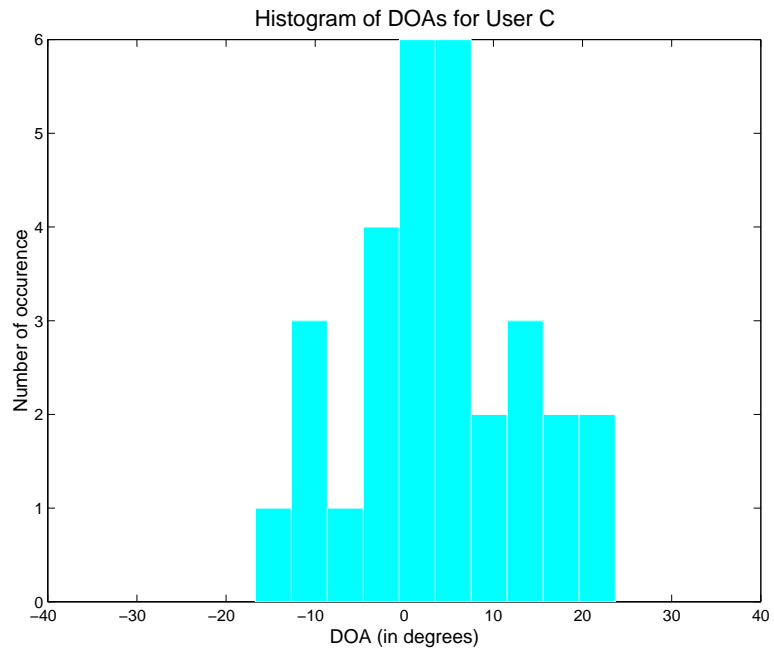


Figure 6.10: Histogram of DOA estimates: User C

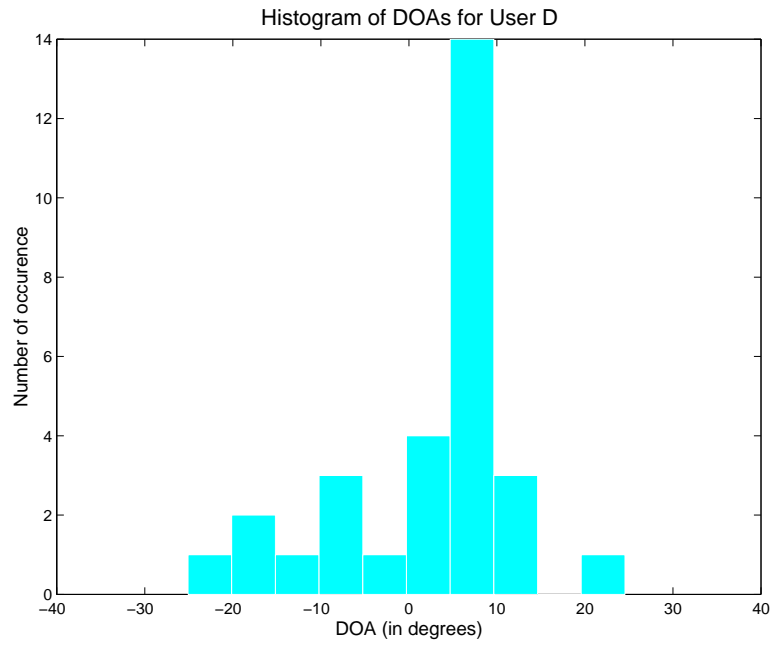


Figure 6.11: Histogram of DOA estimates: User D

6.6 SINR Improvement from Beamforming

The signal-to-interference-noise ratio (SINR) for beamforming is given in (6.15). In this section, the performance of the beamforming receiver introduced in Section 6.5.1 is evaluated by comparing SINR for the case of beamforming and no beamforming. Beamforming weights are selected with two different criteria: maximizing SINR and maximizing SNR at the beamformer's output.

It is shown in [43] that the optimum weight for maximum SINR beamforming is [27] [43]:

$$\mathbf{w}_i = \kappa \mathbf{R}_{\text{NI}}^{-1} \mathbf{a}_i \quad (6.32)$$

where κ is a constant. Since κ does not affect the SINR, it can be set to unity. For maximum SNR beamforming the weight vector is the equal to array response vector as shown in [68], i.e. $\mathbf{w}_i = \mathbf{a}_i$. In order to compare the SINR from beamforming to no beamforming, we need to calculate the average SINR for a single antenna element without beamforming.

Since the estimate of P_i in Equation (6.28) is the total received power for an M -element antenna array [27], the average received power for each element is $P_{avg_i} = \frac{1}{4}P_i$. Also, the average interference and noise power can be calculated as $P_{NI} = \frac{1}{4}\text{trace}(\mathbf{R}_{\text{NI}})$. The beamforming weight becomes a scalar of $w_i = 1$ for the case of no beamforming. The SINR expression for beamforming performance evaluation derived in Section 6.5.1 is:

$$\text{SINR}_B = \frac{8P_i |\mathbf{w}_i^H \mathbf{a}_i|^2}{\mathbf{w}_i^H \mathbf{R}_{\text{NI}} \mathbf{w}_i} \quad (6.33)$$

for user i using P_i , \mathbf{a}_i and \mathbf{R}_{NI} estimated from methods presented in Section 6.5.2. The SINR expression for non-beamforming is then

$$SINR_{NB} = \frac{8P_{avg_i}}{P_{NI}} \quad (6.34)$$

Using Equations (6.33) and (6.34) the SINR gain from beamforming can be calculated as:

$$SINR_{Gain_{dB}} = 10 \times \log_{10}(SINR_B/SINR_{NB}) \quad dB \quad (6.35)$$

From (6.35), the SINR gain are calculated for the cases of maximum SINR beamforming and maximum SNR beamforming. The SINR gain is calculated separately for each trial for every user and tabulated in Tables 6.8 and 6.9 for maximum SINR and SNR beamforming, respectively. Then the SINR calculations are repeated using the alternative method for P_i estimation with N_o assumed as the receiver's noise floor of -71.5 dBm calculated in Section 4.3. The results for the alternative method are shown in Tables 6.10 and 6.11.

The difference between the two different beamformers, maximum SINR and maximum SNR, is clearly shown from our results. The maximum SNR beamformer receives little SINR gain from beamforming. The average gain ranges from 0.049 dB to 0.147 dB for the four users. This suggests that interference cannot be reduced effectively by beamforming towards the desired signal. Whereas the maximum SINR beamformer achieves little gain in some trials and high gain in others. The average gain for maximum SINR beamformer ranges from 0.176 dB to 3.335 dB for the four users. More interesting is that individual trials can have SINR gain up to 9.96 dB in the extreme case (user C, Trial #2). In addition, the SINR gain varies greatly from trial to trial, meaning that the beamforming weights should be re-estimated for each trial (consisting of a 25-bit block) in order to obtain the benefits from beamforming. Last but not least, the gain for user A and user C on average are much higher than that of users B and D. One of the reasons for this observation can be that the spatial signature of users B and D are similar, thus, the maximum SINR beamformer is

unable to null user B when the desired user is D, and vice versa. Once again, the beampatterns from Figures 6.4 and 6.6 support this argument.

The SINR gain from the alternative method match very closely to those when the noise term is estimated, rather than assumed known. The difference in mean SINR gain for the two methods is at most 0.14 dB (user A) for maximum SINR beamforming and 0.0089 dB for maximum SNR beamforming (user A). The advantage of the first method is that it does not require any prior knowledge of the noise density N_o . The agreement of two methods suggests that the estimation of the noise term k using Equation (6.24) is appropriate.

<i>Trial #</i>	<i>SINR gain (dB)</i>			
	<i>User A</i>	<i>User B</i>	<i>User C</i>	<i>User D</i>
1	1.2686	0.2421	0.5478	0.1817
2	0.5657	0.7359	9.9663	0.1242
3	0.8032	0.3068	1.0875	0.1172
4	0.4499	1.0176	7.7437	0.0762
5	2.2636	0.4619	1.8951	0.1316
6	2.2671	0.6009	0.5039	0.1124
7	1.7231	0.0618	3.6887	0.4342
8	0.7300	0.1126	1.3774	0.0741
9	6.4310	0.1076	0.8440	0.1044
10	1.4068	1.3040	3.3895	0.8025
11	2.1046	0.8328	6.2468	0.1304
12	2.7144	0.6690	0.1731	0.2064
13	2.2060	1.2797	1.6273	0.0502
14	3.0421	0.0691	5.6276	0.1126
15	2.5736	0.1752	0.3302	0.3176
16	0.9519	0.0543	3.4692	0.1908
17	1.8434	0.0565	3.9990	0.1774
18	2.8481	0.5418	6.2744	0.2001
19	1.5273	0.8989	7.2039	0.0996
20	3.8369	0.2343	5.8287	0.1421
21	0.5219	0.7031	5.1806	0.1345
22	3.1424	0.9759	0.6126	0.0488
23	1.5227	0.0849	2.5044	0.5099
24	1.7883	1.0521	1.9918	0.0994
25	6.3051	0.6406	3.9968	0.1081
26	7.2112	0.0638	2.3029	0.1336
27	1.2548	0.6272	1.8578	0.1967
28	1.1770	1.3359	1.6314	0.1032
29	6.6389	0.1764	6.3639	0.1026
30	6.2599	0.4423	1.7677	0.0626
mean	2.5793	0.5288	3.3345	0.1762

Table 6.8: SINR gain from Maximum SINR Beamforming

<i>Trial #</i>	<i>SINR gain (dB)</i>			
	<i>User A</i>	<i>User B</i>	<i>User C</i>	<i>User D</i>
1	0.0893	0.0024	0.0634	0.0830
2	0.1385	0.0357	0.0635	0.0569
3	0.2252	0.1191	0.1184	0.1060
4	0.2780	0.2941	0.0692	0.0574
5	0.5128	0.0710	0.2404	0.1081
6	0.2367	0.1174	0.2084	0.1023
7	0.0831	0.0153	0.2026	0.0758
8	0.1019	0.0354	0.0697	0.0586
9	0.1641	0.0911	0.3223	0.0797
10	0.1309	0.0699	0.0547	0.0868
11	0.0810	0.0214	0.0523	0.0829
12	0.1073	0.0208	0.1170	0.0876
13	0.2099	0.0458	0.0699	0.0281
14	0.1168	0.0588	0.0196	0.0818
15	0.1336	0.0315	0.1161	0.1137
16	0.1036	0.0247	0.0294	0.1489
17	0.1321	0.0523	0.1165	0.1511
18	0.2208	0.0006	0.0777	0.0538
19	0.0678	0.0335	0.0633	0.0836
20	0.0749	0.0233	0.3018	0.1166
21	0.1919	0.0508	0.0477	0.0770
22	0.1515	0.0406	0.0595	0.0326
23	0.1032	0.0085	0.1557	0.1218
24	0.0583	0.0164	0.0876	0.0825
25	0.1290	0.0196	0.0529	0.0913
26	0.1989	0.0180	0.0481	0.0924
27	0.1983	0.0479	0.0502	0.0921
28	0.0627	0.0238	0.0894	0.0849
29	0.0372	0.0092	0.0496	0.0760
30	0.0685	0.0718	0.0457	0.0610
mean	0.1469	0.0490	0.1021	0.0858

Table 6.9: SINR gain from Maximum SNR Beamforming

<i>Trial #</i>	<i>SINR gain (dB)</i>			
	<i>User A</i>	<i>User B</i>	<i>User C</i>	<i>User D</i>
1	1.1837	0.2401	0.5426	0.1779
2	0.5278	0.7307	9.4378	0.1223
3	0.7561	0.3065	1.0296	0.1182
4	0.4206	0.9635	8.0073	0.0761
5	2.1315	0.4622	1.8445	0.1327
6	1.9708	0.6049	0.5004	0.1133
7	1.6880	0.0618	3.4595	0.4375
8	0.6877	0.1124	1.3104	0.0741
9	5.7393	0.1087	0.8600	0.1053
10	1.2778	1.3128	3.2435	0.8091
11	1.9293	0.8113	5.9461	0.1313
12	2.4052	0.6667	0.1629	0.2081
13	1.9889	1.2696	1.5450	0.0501
14	2.8262	0.0693	6.1806	0.1128
15	2.3189	0.1750	0.3226	0.3155
16	0.9161	0.0541	3.3692	0.1899
17	1.7332	0.0566	3.8177	0.1791
18	2.4794	0.5406	5.5110	0.2004
19	1.3776	0.8944	6.5865	0.1005
20	3.6426	0.2342	6.0055	0.1441
21	0.4699	0.6975	4.8814	0.1356
22	3.0064	0.9614	0.6026	0.0488
23	1.3908	0.0848	2.3553	0.5143
24	1.7194	1.0452	1.9155	0.1001
25	5.7280	0.6389	3.8448	0.1093
26	6.9270	0.0638	2.5383	0.1350
27	1.2059	0.6200	1.7486	0.1986
28	1.0685	1.3313	1.5463	0.1040
29	5.4133	0.1761	5.9359	0.1035
30	8.5288	0.4301	1.6809	0.0630
mean	2.4486	0.5242	3.2577	0.1770

Table 6.10: SINR gain from Maximum SINR Beamforming: N_o assumed

<i>Trial #</i>	<i>SINR gain (dB)</i>			
	<i>User A</i>	<i>User B</i>	<i>User C</i>	<i>User D</i>
1	0.0841	0.0024	0.0628	0.0813
2	0.1296	0.0355	0.0629	0.0560
3	0.2128	0.1190	0.1127	0.1070
4	0.2602	0.2797	0.0687	0.0573
5	0.4882	0.0710	0.2351	0.1090
6	0.2117	0.1181	0.2070	0.1032
7	0.0817	0.0153	0.1942	0.0764
8	0.0964	0.0353	0.0668	0.0586
9	0.1556	0.0921	0.3280	0.0804
10	0.1204	0.0704	0.0531	0.0875
11	0.0756	0.0209	0.0511	0.0835
12	0.0980	0.0208	0.1102	0.0883
13	0.1933	0.0455	0.0669	0.0280
14	0.1108	0.0589	0.0191	0.0819
15	0.1234	0.0315	0.1135	0.1130
16	0.1001	0.0246	0.0288	0.1482
17	0.1255	0.0524	0.1132	0.1525
18	0.1992	0.0006	0.0731	0.0539
19	0.0621	0.0334	0.0610	0.0844
20	0.0724	0.0233	0.3062	0.1182
21	0.1734	0.0505	0.0462	0.0777
22	0.1468	0.0400	0.0586	0.0327
23	0.0955	0.0084	0.1486	0.1228
24	0.0565	0.0163	0.0849	0.0831
25	0.1233	0.0196	0.0516	0.0923
26	0.1957	0.0180	0.0472	0.0934
27	0.1914	0.0474	0.0478	0.0930
28	0.0576	0.0237	0.0855	0.0856
29	0.0338	0.0092	0.0481	0.0767
30	0.0631	0.0699	0.0438	0.0613
mean	0.1380	0.0485	0.0999	0.0862

Table 6.11: SINR gain from Maximum SNR Beamforming: N_o assumed

6.7 Summary

In this chapter, we studied the spatial characteristics of an outdoor environment using experimental data. The fading correlation between antenna elements suggests that the basestation antenna array experiences an angle spread of about 20° or less. The beampatterns in Section 6.4 indicates that the orientation of the transmitters with respect to the basestation and its surroundings has an effect on the stability of beampatterns. The peaks of the beampatterns and the DOA estimates calculated from array response vectors show that the direction-of-arrival of the received signal does not correspond to the true location of the transmitters.

The beamforming receiver introduced in Section 6.5 is used to evaluate the performance of antenna array in reducing interference or maximizing the signal-to-interference-noise ratio (SINR). Our method of SINR calculation does not require any prior information on received signal power or noise density. Results from Section 6.6 show that a user can get an average of 0.176 to 3.33 dB gain from maximum SINR beamforming. The amount of gain achieved by beamforming is shown to be related to the spatial characteristics of each user. User A in Table 6.8 has an average SINR gain of 2.58 dB while users B and D have gains of 0.529 and 0.176 dB because their spatial characteristics are similar as illustrated by their respective beampatterns.

Chapter 7

Conclusion

7.1 Summary

In this thesis we presented the design and features of our wideband CDMA smart antenna measurement system. The key features of our testbed are:

- multiple portable transmitters (four)
- basestation receiver with 4-element antenna array
- high chip rate of 7 Mcps
- fine sampling resolution of 5 samples per chip

Using our testbed, we gathered data in a outdoor wideband CDMA environment. We conducted two outdoor experiments: outdoor experiment #1 was performed with three transmitters and a 2-element antenna array, and outdoor experiment #2 with four transmitters and a 4-element antenna array. The experimental scenarios are illustrated in Figures 5.1 to 5.4 for experiment #1, and in Figure 6.1 for experiment #2.

The discussion on design issues explained how we designed our system such that it satisfies or resembles 3G wideband CDMA specifications, wideband digital radio

design, and flexibility in hardware for future development while maintaining our low-cost modular objective. Then functional testing was performed to ensure proper operation of our testbed. An experimental beam pattern is plotted in Figure 4.23 to illustrate that the array was well-calibrated.

System synchronization issues regarding synchronous sampling between array elements and carrier synchronization were solved by hardware logic and a custom designed software. Whereas, initial synchronization and tracking of PN code were achieved through matched filtering and digital early-late tracking loop.

We studied outdoor propagation characteristics for wideband CDMA. An investigation on propagation path loss showed that signals experienced a path loss exponent of $n = 2.64$, similar to those in urban environment found in [36]. Using our CFAR multipath detection algorithm, multipath statistics such as multipath peak amplitudes, mean delay and delay spread were gathered for both single and multiple-user scenarios. The results are presented in Tables 5.9-5.10 for single-user and Tables 5.17-5.18 for multiple-user. We found that in multiple-user environment, the simple PN acquisition and tracking does not work well if fading and interference were severe. The study of time variation of multipath delay profiles indicated that about 50% of the trials were time varying within a snapshot. We found that multipath statistics of averaged delay profiles (see Tables 5.20 - 5.29) for 2^{nd} strongest peaks were rather stable in time. However, this stability did not hold for the weaker paths.

We also investigated the spatial characteristics of the same outdoor environment. Fading correlation results summarized in Table 6.2 suggested that the basestation antenna array experienced an angle spread of about 20° or less. The beam patterns generated for each user's despread output (Figures 6.3 to 6.6) indicated that the orientation of the transmitters with respect to the basestation and its surroundings has

an effect on the stability of beampatterns. The peaks of the beampatterns and the DOAs estimated using experimental array response vectors show that the direction of arrival of received signal does not correspond to the true location of the transmitters.

A beamforming receiver was first introduced and then used to evaluate the improvement of SINR with beamforming over a single antenna. Our proposed method of SINR calculation does not require any prior information on received signal power or noise density. Results showed that a user can get an average of 0.176 to 3.33 dB gain from maximum SINR beamforming (see Table 6.8). By comparing SINR gains and the corresponding beampatterns of the users, we concluded that the amount of SINR gain is related to the spatial characteristics of the user relative to the others.

7.2 Future Directions

Future extensions of the research could be,

- modify the existing system to support higher chip rate. The current system has a total bandwidth of 15 MHz. Since SX043 produces PN sequences with double sideband, our chip rate is limited to 7 Mcps (with a 1 MHz as design margin). The transmitter chip rate can be doubled if a baseband analog circuit is designed to remove the one of the sidebands before upconversion.
- design a reference phase lock system for the receiver link. Currently, a common 5 MHz reference signal is used to phase lock the TX and RX local oscillator. However, our solution requires a 75-foot long cable to connect between the TX and its LO to provide portability. With the phase lock system, the TX range will not be limited by the length of the coaxial cable.

- increase the data storage capacity of a snapshot. The current onboard FIFO of the A/D evaluation board module has a maximum size of 16k samples. The length of a snapshot is limited by the FIFO size. By studying the schematics and timing diagram provided in Appendix A, the current FIFO memory register may be modified or replaced to increase the storage capacity.
- further field measurements which may include the following: larger TX-RX distances, using different PN sequences, exploit different array geometry and array element separation distances.
- evaluation of joint detection schemes. We mentioned in Chapter 3 that joint detection schemes such as multi-user detection (MUD) performs better with short PN codes. Therefore the four TXs data gathered in experiment #2 is suitable for evaluation of MUD in a wideband CDMA system.
- MIMO (multi-input multi-output) channel characterization for space-time coding. A channel can be characterized as a multiple-input and multiple-output system. Using our measurement data, the outdoor channel can be characterized by MIMO method and evaluate space-time coding for wideband CDMA.

7.3 Conclusion

In this thesis, a wideband CDMA smart antenna measurement system is presented. The testbed includes a 4-element antenna array and four portable programmable transmitters with off-line data storage. It operates in the 1.9 GHz band with a high chip rate of 7 Mcps.

From our measurement data, we characterized the outdoor channel in single and multiple-transmitter scenarios by: gathering statistics on multipath peak amplitudes and delays, and studying channel variation and dynamics through multipath delay

profile averaging and delay profile correlations. The multipath statistics plus the channel dynamics gathered can provide valuable information for RAKE receiver tap selection and update rate.

The time variation analysis of beampatterns gives an estimate of the stability of spatial characteristics in multiple-user scenarios. Our beamforming algorithm has an average SINR gain (over single antenna) of 2.579 dB for user A, 0.529 dB for user B, 3.335 dB for user C, and 0.176 dB for user D over 30 experimental trials. The advantage of our beamformer is that it does not require any prior knowledge of received signal power and noise density.

Appendix A

Synchronization of Sampled Data

In chapter 4 we have discussed the importance of synchronization in gathering sampled data for an antenna array. As mentioned briefly in that chapter, synchronization is achieved through hardware and software design. In this appendix the hardware design and logic are first presented, and then concludes with a software algorithm and data read timing analysis.

A.1 Hardware Setup

High speed data available at the A/D converter's output enter AD6620 evaluation board (with FIFO storage) through a data latch. The timing for latching data into FIFO is controlled by the sampling clock. Since we gather data directly from A/D converter, the data bypass the AD6620 IC chip and then latch by a second latch latch into the FIFO (see Figure A.1). When the FIFO is filled with data, the software can start reading the data onto the PC through a IEEE 1284 parallel port interface (see Figure A.1) [11].

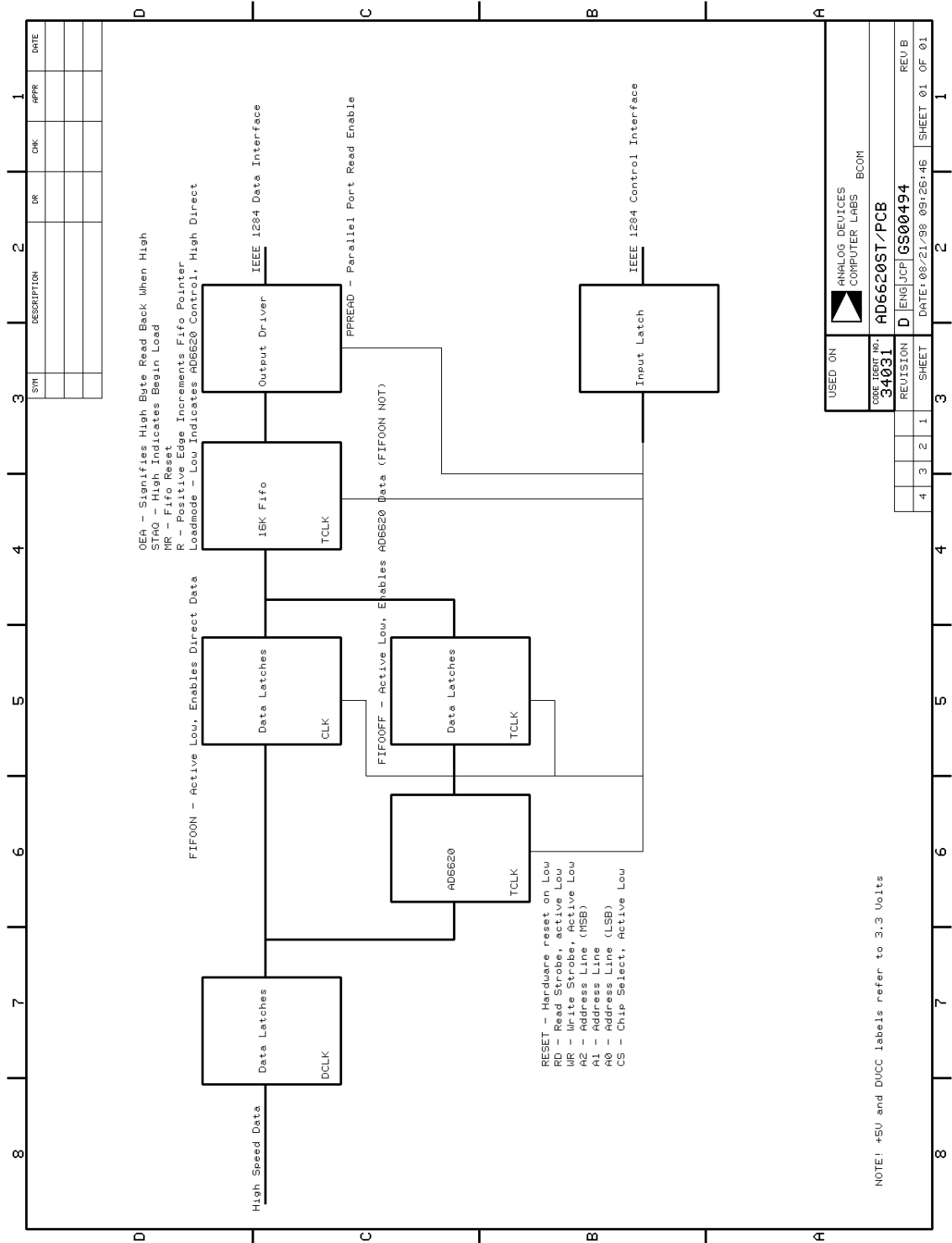


Figure A.1: AD6620 evaluation board system block diagram

A.1.1 A/D converter interface

A 50-pin connector is used to connect the AD6620 board to the A/D converter, the data lines are available on I0-I15 (Pin 2-9 and Pin 12-19) and the sampling clock on Pin 10 (Figure A.2). The data bus is 16-bit wide because the AD6620 evaluation board is also designed to connect other A/D converters with 16-bit resolution. The lines I0-I3 are grounded at the A/D converter's output [12], thus, the data format is MSB (most significant bit) justified into a 16-bit word.

There are two sets of data latches [11] shown in Figure A.2. The first set is for both data bypass or AD6620 mode. The OE (output enable) pins on the latches are grounded which means that data are continuously latch and available to bypass FIFO latch. In the bypass mode the data are available at FIFO's input data pins only when the latch has been selected to turn on by "FIFOON" signal generated by the control software. Software control issues is discussed later in this appendix.

A.1.2 FIFO Data Read and Synchronization

The schematics of the FIFO and its output registers are displayed in Figure A.3. Besides the data pins, there are also control and status pins which can be read or written by control software to perform operations such as reset, enable reading or writing or indicate FIFO is full. Two 8-bit output registers are used at the FIFO's output because the parallel port's data bus is only 8-bit wide. The U204 chip in Figure A.3 latches the high-byte and U205 latches the low-byte of a sampled data.

The functions of the FIFO's control and status pins are summarized as follows:

- FIFO's master reset (pin 62) is controlled by the "MR" line. The pin is active low.

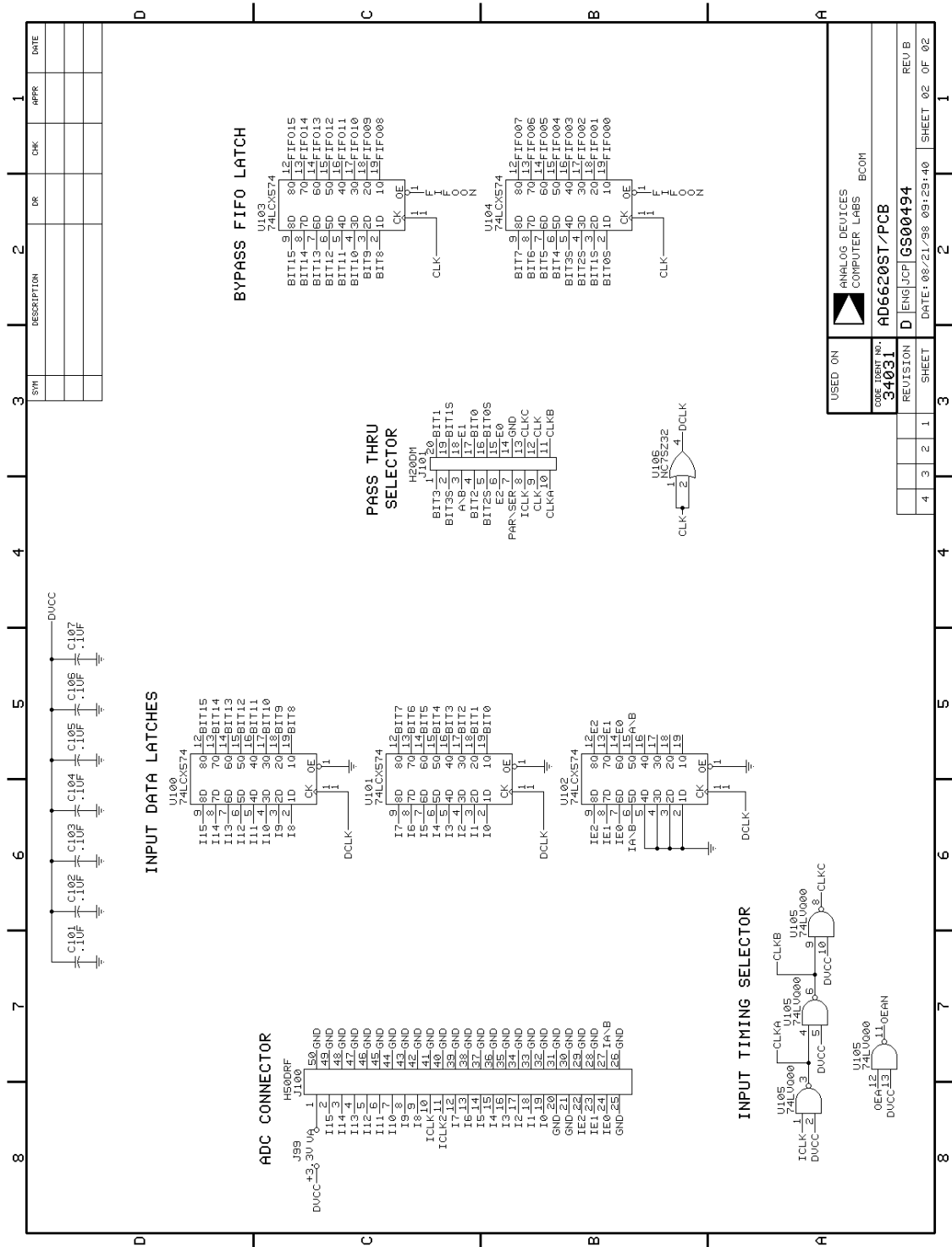


Figure A.2: A/D converter connector and data latches

- data read is enabled at all times (pin 51)
- data output pins are enabled at all times (pin 49)
- data loading to FIFO (pin 1) is controlled by "DVN" line. The pin is active low.
- data read for every sample is clocked by the "R" (pin 52) signal
- "Full" status line is inserted low by FIFO when it becomes full (pin 58)

Since the "DVN" line controls when the FIFO starts to load, the sampled data from all four elements of the array can be synchronized by hard-wiring this "DVN" line from one board (called master) to the others (called slaves). In order to avoid two sources driving the 'DVN' signal on a slave board, we have to disable the "DVN" line on Pin 14 of U303 (see Figure A.4) from the slaves boards before hard-wiring them together with the master "DVN" line.

For each cycle of FIFO's read clock ("R"), a new 16-bit data is loaded to FIFO output registers and split into high and low bytes. The high byte is first read by the software and followed by the low byte. The rest of the data samples in FIFO are read similarly by interleaving high and low bytes, and later combined into a 16-bit word by control software. The "OEA" and "OEAN" control lines (Figure A.3) are responsible for the timing of interleaving data read. These two lines are also controlled by software.

A.2 Parallel Port Read and Write

The parallel port interface on the AD6620 evaluation board serves two purposes: allow FIFO commands write to AD6620 board and reading data from FIFO. Thus

<i>Pin</i>	<i>EPP Signal</i>	<i>In/Out</i>	<i>Function</i>
1	nWrite	Out	A low on this line indicates a Write, high indicates a Read
2-9	Data 0-7	In/Out	Data bus, bi-directional
11	nWait	In	Used for handshaking, a EPP cycle can be started when low, and finished when high
12	nFull	In	FIFO status. A high indicates FIFO is empty, and a low indicates FIFO is Full
14	nData Strobe	Out	when low, indicates Data transfer
36	nAddress Strobe	Out	when low, indicates Address transfer

Table A.1: Pin Assignments for Enhanced Parallel Port connector

the transfer of data requires bi-directional parallel port, in particular, the IEEE-1284 standard parallel port with EPP (enhanced parallel port) support from the PC's system BIOS. In this section we briefly describe the read/write cycles and settings for EPPs, for in-depth discussions please refer to [50] [51].

The parallel port connector J400 in the schematic shown in Figure A.5 is used to interface with printer ports (LPT1 or LPT2) on a PC. The pin assignments of the connector is listed in Table A.1. The table users "n" in front of the signal name to indicate that the signal is active low.

The address and data bus are both connected to the data pins of the parallel port as illustrated in Figure A.5. This implies that the address and data bus are time multiplexed. By allowing either the address or data strobe to be active in read/write cycles, address and data bus multiplexing is achieved. The FIFO command latch in

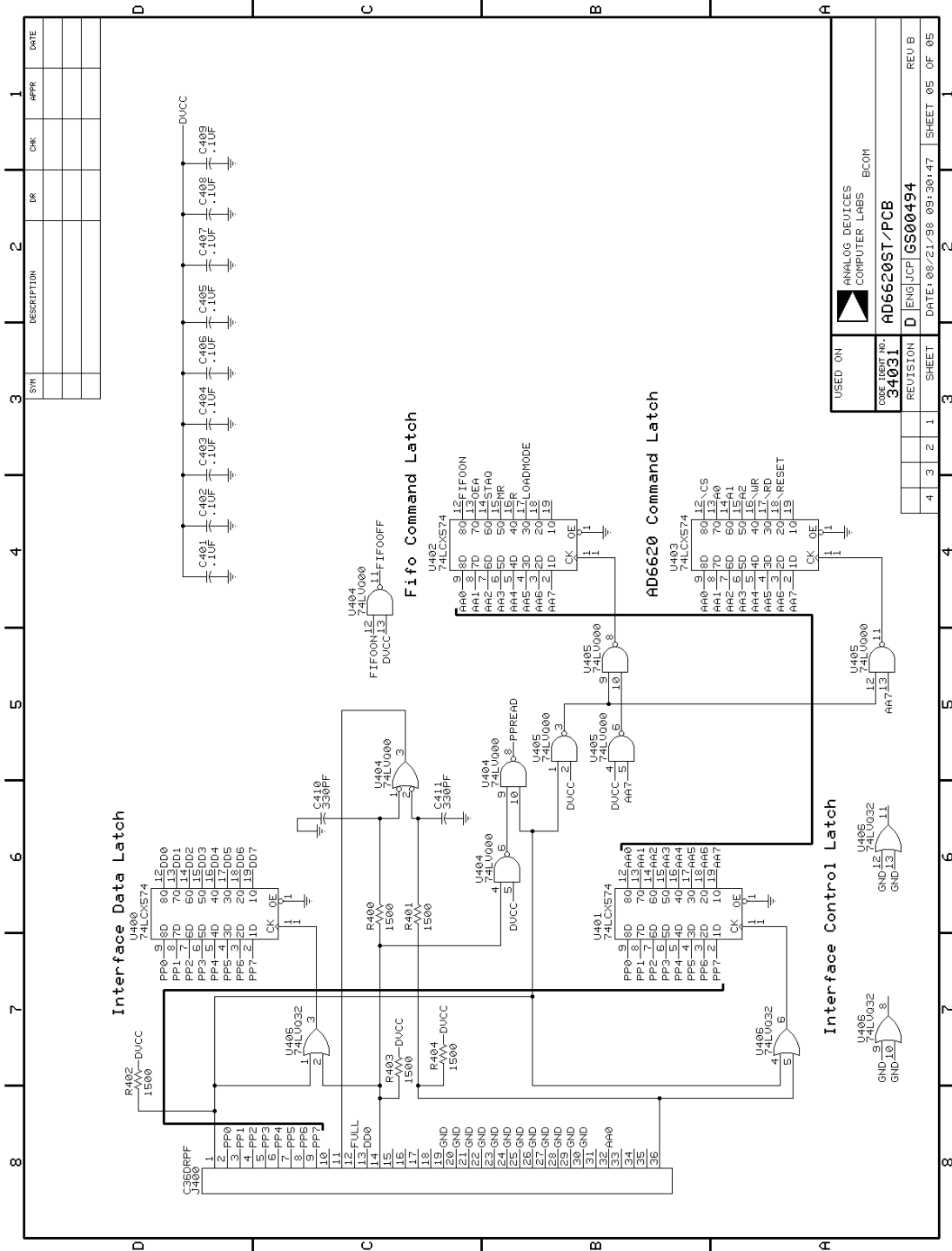


Figure A.5: Parallel port interface, data and command latch

SYN	DESCRIPTION	DR	CHK	APPR	DATE

USED ON	ANALOG DEVICES COMPUTER LABS	B.COM
DATE	08/21/98	09:30:47
DESIGN NO.	34031	AD6620ST/PCB
REVISION	D	JCP/SGS00494
SHEET	1	1
SHEET	05	05
REV/B		

Figure A.5 shows the bit settings of the command register. Different commands can be sent to the FIFO by writing different values to this command register.

A.3 Software Control and Data Read

With the understanding of hardware and parallel port settings, software procedures are written for initializing AD6620 board, reading data (sampling) and saving data on hard disk. There is no timing diagram provided by the AD6620 board's manual, therefore, we have to derive the appropriate read/timing timings from the schematic. The timing diagram is then verified by observing the address and data lines on oscilloscope. The timing diagram is illustrated in Figure A.6.

First the FIFO is reset by inserting "MR" low, then inserting both "STAQ" and "LOADMODE" to high produces a low "DVN" signal. A low on "DVN" instructs the FIFO start to loading data from A/D converter, i.e. starts sampling. When "FULL" is asserted low by FIFO, onboard data is now ready to transfer to PC. The FIFO read is first disabled by setting "DVN" to high. Then "OEA" and "R" provide high and low byte interleaving and FIFO read clock, respectively.

A.3.1 Software Synchronization

The timing diagram shown in Figure A.6 is valid for a single array element. For case of four elements, as explained in section 4.4.2, a handshaking is required between the primary and secondary copies of the *ADport* software. The steps are as follows:

1. "Init" on both primary and secondary copies

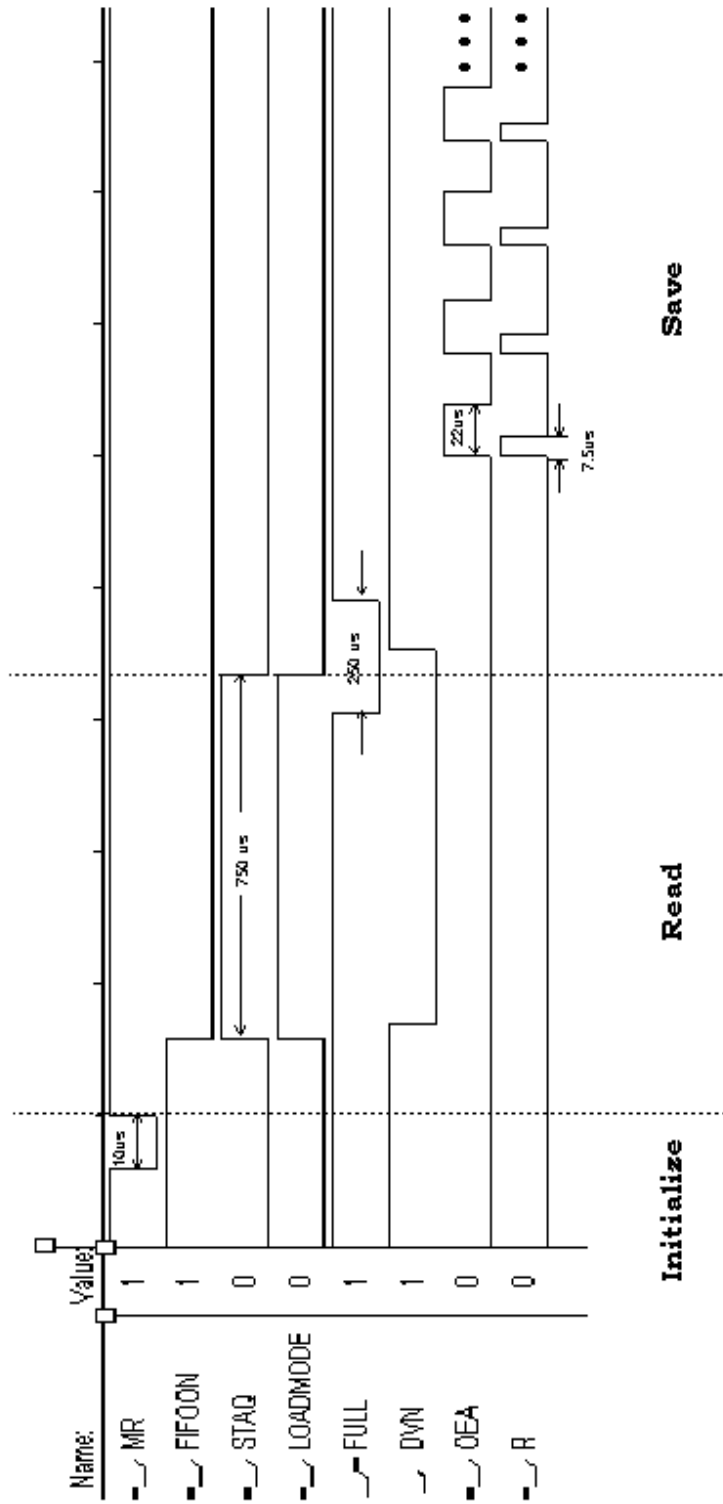


Figure A.6: Timing Diagram for FIFO commands and Data Read

2. "Read" on primary copy
3. "Save" on primary copy
4. "Save" on secondary copy
5. repeat 1-4 for next snapshot

The reason for the synchronization steps can be explained using the timing diagram (Figure A.6). Execute step 1 clears FIFOs on all elements and put them ready to load sampled data. The "Read" function run on the primary copy inserts "DVN" to low (starts loading). Since the "DVN" line is hard-wired from the master to all slave boards, the three slaves also receive the load command. In this way, synchronous sampling is achieved. The "Save" function should execute on the primary copy first because it disables "DVN" for all boards before data transfer to PC can begin.

For specific details in the read/write cycle, please refer to program listings of *AD-port*. Detail comments are given for all the procedures in the program code regarding FIFO control and data downloading. The program listing of is provided in the next section.

A.3.2 ADport Program Listing

```

//*****
// ADportDlg.cpp : ADport Version 1.1 Implementation file
//
// Description:   This is the second version of ADport which allows
//               synchronized data read between two different PCs.
//               N.B. particular steps for data collection must be
//               followed in order to achieve the required
//               synchronization
//
// Written by:   Noel Tin
//
// Date:        May, 2000.
//*****

#include "stdafx.h"
#include "iostream.h"
#include "math.h"
#include "string.h"
#include "ADport.h"
#include "ADportDlg.h"
/** include header file from DLPortIO
#include "DLPortIO.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

//user-define variable types

typedef unsigned char UCHAR;    // BYTE

//-----
// Constants
//-----

// Masks parallel port bits
const UCHAR BIT0 = 0x01;      // because the type of DLPortReadPortUChar is UCHAR
const UCHAR BIT1 = 0x02;      // ...this is used in GetPin()
const UCHAR BIT2 = 0x04;
const UCHAR BIT3 = 0x08;
const UCHAR BIT4 = 0x10;
const UCHAR BIT5 = 0x20;
const UCHAR BIT6 = 0x40;
const UCHAR BIT7 = 0x80;

// Masks for AD6640 data bits
const UCHAR ADBit0 = 16;      // decimal value of bit 4 of LowByte
const UCHAR ADBit1 = 32;      //           5
const UCHAR ADBit2 = 64;      //           6
const UCHAR ADBit3 = 128;     //           7
const UCHAR ADBit4 = 1;       // decimal value of bit 0 of HighByte
const UCHAR ADBit5 = 2;       //           1
const UCHAR ADBit6 = 4;       //           2
const UCHAR ADBit7 = 8;       //           3
const UCHAR ADBit8 = 16;      //           4
const UCHAR ADBit9 = 32;      //           5
const UCHAR ADBit10 = 64;     //           6
const UCHAR ADBit11 = 128;    //           7

```

```

// Printer Port pin numbers
const ACK_PIN      = 10;
const BUSY_PIN     = 11;
const FIFOFULL_PIN = 12;
const SELECTOUT_PIN = 13;
const ERROR_PIN    = 15;
const STROBE_PIN   = 1;
const AUTOFD_PIN   = 14;
const INIT_PIN     = 16;
const SELECTIN_PIN = 17;

const MAX_LPT_PORTS = 2;      // maximum number of ports in a PC

////////////////////////////////////
// variables
////////////////////////////////////

BYTE   LPTNumber;           // Current number of the printer port, default=1
DWORD  LPTMaster;          // The address of the printer port connected to master board
DWORD  LPTSlave;           // " " " " " " " " " slave "
DWORD  Count;              // number of bytes to read from FIFO
DWORD  LPTAddress[MAX_LPT_PORTS+1]; // List of port addresses installed on the system

int     BufferM[16*1024];   // Buffer for storing temp. data read from Master board
int     BufferS[16*1024];   // " " " " " " " " Slave "

////////////////////////////////////
// CAboutDlg dialog used for App About
////////////////////////////////////

class CAboutDlg : public CDialog
{
public:
    CAboutDlg();

    // Dialog Data
   //{{AFX_DATA(CAboutDlg)
    enum { IDD = IDD_ABOUTBOX };
    //}}AFX_DATA

    // ClassWizard generated virtual function overrides
   //{{AFX_VIRTUAL(CAboutDlg)
protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV support
    //}}AFX_VIRTUAL

    // Implementation
protected:
    //{{AFX_MSG(CAboutDlg)
    virtual void OnOK();
    //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};

CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)
{
    //{{AFX_DATA_INIT(CAboutDlg)
    //}}AFX_DATA_INIT
}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{

```

```

        CDialog::DoDataExchange(pDX);
        //{{AFX_DATA_MAP(CAboutDlg)
        //}}AFX_DATA_MAP
    }

BEGIN_MESSAGE_MAP(CAboutDlg, CDialog)
    //{{AFX_MSG_MAP(CAboutDlg)
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()

////////////////////////////////////
// CADportDlg dialog

CADportDlg::CADportDlg(CWnd* pParent /*=NULL*/)
    : CDialog(CADportDlg::IDD, pParent)
{
    //{{AFX_DATA_INIT(CADportDlg)
    m_LPT1 = -1;
    m_LPT2 = -1;
    m_16k = -1;
    m_1k = -1;
    m_2k = -1;
    m_8k = -1;
    m_none = -1;
    //}}AFX_DATA_INIT
    // Note that LoadIcon does not require a subsequent DestroyIcon in Win32
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
    pFileSave = 0;
}

void CADportDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(CADportDlg)
    DDX_Radio(pDX, IDC_LPT1, m_LPT1);
    DDX_Radio(pDX, IDC_LPT2, m_LPT2);
    DDX_Radio(pDX, IDC_16k, m_16k);
    DDX_Radio(pDX, IDC_1k, m_1k);
    DDX_Radio(pDX, IDC_2k, m_2k);
    DDX_Radio(pDX, IDC_8k, m_8k);
    DDX_Radio(pDX, IDC_none, m_none);
    //}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CADportDlg, CDialog)
    //{{AFX_MSG_MAP(CADportDlg)
    ON_WM_SYSCOMMAND()
    ON_WM_PAINT()
    ON_WM_QUERYDRAGICON()
    ON_BN_CLICKED(IDExit, OnExit)
    ON_BN_CLICKED(IDC_About, OnAbout)
    ON_BN_CLICKED(ID_FILE_SAVE_AS, OnFileSaveAs)
    ON_BN_CLICKED(IDC_Read, OnRead)
    ON_BN_CLICKED(IDC_LPT1, OnLpt1)
    ON_BN_CLICKED(IDC_LPT2, OnLpt2)
    ON_BN_CLICKED(IDC_1k, On1k)
    ON_BN_CLICKED(IDC_2k, On2k)
    ON_BN_CLICKED(IDC_8k, On8k)
    ON_BN_CLICKED(IDC_16k, On16k)
    ON_BN_CLICKED(IDC_none, Onnone)
    ON_BN_CLICKED(IDC_Init, OnInit)
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()

////////////////////////////////////

```

```

// CADportDlg message handlers

BOOL CADportDlg::OnInitDialog()
{
    m_LPT1 = 0;           //default port is LPT1
    m_LPT2 = m_none = -1;

    m_16k = 0;          //default 16k read
    m_2k = m_8k = m_1k = -1;

    CDialog::OnInitDialog();

    // Add "About..." menu item to system menu.

    // IDM_ABOUTBOX must be in the system command range.
    ASSERT((IDM_ABOUTBOX & 0xFFFF) == IDM_ABOUTBOX);
    ASSERT(IDM_ABOUTBOX < 0xF000);

    CMenu* pSysMenu = GetSystemMenu(FALSE);
    if (pSysMenu != NULL)
    {
        CString strAboutMenu;
        strAboutMenu.LoadString(IDS_ABOUTBOX);
        if (!strAboutMenu.IsEmpty())
        {
            pSysMenu->AppendMenu(MF_SEPARATOR);
            pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu);
        }
    }

    // Set the icon for this dialog. The framework does this automatically
    // when the application's main window is not a dialog
    SetIcon(m_hIcon, TRUE);           // Set big icon
    SetIcon(m_hIcon, FALSE);        // Set small icon

    // TODO: Add extra initialization here

    LPTAddress[0] = 0x378;           // no master
    LPTAddress[1] = 0x378;           // LPT1 address
    LPTAddress[2] = 0x278;           // LPT2 address

    LPTNumber = 1;                  //default LPT
    Count = 1024*16;                //default FIFO read size

    return TRUE; // return TRUE unless you set the focus to a control
}

void CADportDlg::OnSysCommand(UINT nID, LPARAM lParam)
{
    if ((nID & 0xFFFF) == IDM_ABOUTBOX)
    {
        CAboutDlg dlgAbout;
        dlgAbout.DoModal();
    }
    else
    {
        CDialog::OnSysCommand(nID, lParam);
    }
}

// If you add a minimize button to your dialog, you will need the code below
// to draw the icon. For MFC applications using the document/view model,
// this is automatically done for you by the framework.

void CADportDlg::OnPaint()

```

```

{
    if (IsIconic())
    {
        CPaintDC dc(this); // device context for painting

        SendMessage(WM_ICONERASEBKGND, (WPARAM) dc.GetSafeHdc(), 0);

        // Center icon in client rectangle
        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon + 1) / 2;

        // Draw the icon
        dc.DrawIcon(x, y, m_hIcon);
    }
    else
    {
        CDialog::OnPaint();
    }
}

// The system calls this to obtain the cursor to display while the user drags
// the minimized window.
HCURSOR CADportDlg::OnQueryDragIcon()
{
    return (HCURSOR) m_hIcon;
}

//-----
// SetLPTNumber()
//       Selects the LPT port to use for all LPT operations
//-----
void SetLPTNumber(BYTE Number)
{
    if (Number>=0 && Number<=MAX_LPT_PORTS)
    {
        LPTNumber=Number;
        LPTMaster=LPTAddress[Number];
    }
} // end SetLPTNumber()

//-----
// GetPin()
//       Index valid is in the range 1-25 only (other values return false)
//       Reading the pin returns true when it is 5V, or false when it at 0V.
//-----
bool GetPin(BYTE Pin, DWORD LPTBase)
{
    switch (Pin)
    {
        case 1: return (DlPortReadPortUchar(LPTBase+2)&BIT0)==0; // Inverted
        case 2: return (DlPortReadPortUchar(LPTBase)&BIT0)!=0;
        case 3: return (DlPortReadPortUchar(LPTBase)&BIT1)!=0;
        case 4: return (DlPortReadPortUchar(LPTBase)&BIT2)!=0;
        case 5: return (DlPortReadPortUchar(LPTBase)&BIT3)!=0;
        case 6: return (DlPortReadPortUchar(LPTBase)&BIT4)!=0;
        case 7: return (DlPortReadPortUchar(LPTBase)&BIT5)!=0;
        case 8: return (DlPortReadPortUchar(LPTBase)&BIT6)!=0;
        case 9: return (DlPortReadPortUchar(LPTBase)&BIT7)!=0;
    }
}

```

```

    case 10: return (DlPortReadPortUchar(LPTBase+1)&BIT6)!=0;
    case 11: return (DlPortReadPortUchar(LPTBase+1)&BIT7)==0; // Inverted
    case 12: return (DlPortReadPortUchar(LPTBase+1)&BIT5)!=0;
    case 13: return (DlPortReadPortUchar(LPTBase+1)&BIT4)!=0;
    case 14: return (DlPortReadPortUchar(LPTBase+2)&BIT1)==0; // Inverted
    case 15: return (DlPortReadPortUchar(LPTBase+1)&BIT3)!=0;
    case 16: return (DlPortReadPortUchar(LPTBase+2)&BIT2)!=0;
    case 17: return (DlPortReadPortUchar(LPTBase+2)&BIT3)==0; // Inverted
    default: return false; // pins 18-25 (GND), and other invalid pins
}
} // end GetPin()

//-----
// WaitForResponse()
//      polling loop for acknowledgment or response from port
//      return = 1 if response is detected
//-----
bool WaitForResponse(BYTE Pin, DWORD LPTBase)
{
    DWORD timer = 0;
    CString error = "";

    while (GetPin(Pin, LPTBase)) // if not ready...wait
    {
        timer ++;
        if (timer == 100000) // if timeout
        {
            if (LPTBase == 0x378)
                error = "FIFO is empty, unable to read LPT1.
                Check connection or power";
            else
                error = "FIFO is empty, unable to read LPT2.
                Check connection or power";
            AfxMessageBox(error);
            break;
        }
    }
    if (timer < 100000)
        return true;
    else
        return false;
} // end WaitForResponse()

//-----
// WriteCommand()
//      output FIFO commands stored in the Command variable to Port at address LPTBase
//      return = 1 if WriteCommand() is successful
//-----
void WriteCommand(BYTE Command, DWORD LPTBase)
{
    DlPortWritePortUchar(LPTBase+2, 1); // set Strobe to low
    DlPortWritePortUchar(LPTBase, Command); // write command to data lines

    DlPortWritePortUchar(LPTBase+2, 9); // Saddress and Strobe both low

    DlPortWritePortUchar(LPTBase+2, 1); // clear Saddress
    DlPortWritePortUchar(LPTBase+2, 0); // clear Strobe
} //end WriteCommand()

```

```

//-----
// ReadData()
//   read a byte of data from the Port at address LPTBase. This involves reversing
//   the data direction of the port
//-----
BYTE ReadData(DWORD LPTBase)
{
    BYTE Data = 0;                // a byte can never be < 0

    D1PortWritePortUchar(LPTBase+2, 32); // set bi-directional bit = 1 (input)
    D1PortWritePortUchar(LPTBase+2, 34); // set SData low (data strobe)
    Data = D1PortReadPortUchar(LPTBase); // read data
    D1PortWritePortUchar(LPTBase+2, 0); // clear SData and reset bi-directional pin = 0

    return Data;
} //end ReadData()

//-----
// CombineBytes()
//   convert unsigned AD6640 data into signed data
//   NOTE: the first 4 bits of the LowByte is grounded at 6640 output
//-----
int CombineBytes(BYTE LowByte, BYTE HighByte)
{
    int Data = 0;

    if (LowByte&ADBit0) Data += (int)(pow(2,4));
    if (LowByte&ADBit1) Data += (int)(pow(2,5));
    if (LowByte&ADBit2) Data += (int)(pow(2,6));
    if (LowByte&ADBit3) Data += (int)(pow(2,7));
    if (HighByte&ADBit4) Data += (int)(pow(2,8));
    if (HighByte&ADBit5) Data += (int)(pow(2,9));
    if (HighByte&ADBit6) Data += (int)(pow(2,10));
    if (HighByte&ADBit7) Data += (int)(pow(2,11));
    if (HighByte&ADBit8) Data += (int)(pow(2,12));
    if (HighByte&ADBit9) Data += (int)(pow(2,13));
    if (HighByte&ADBit10) Data += (int)(pow(2,14));
    if (HighByte&ADBit11) Data += (int)(pow(2,15));

    return Data;
} //end CombineBytes()

//-----
// Complement()
//   finding the two's complement of the data and return the signed value
//-----
int Complement(BYTE LowByte, BYTE HighByte)
{
    int Data;

    Data = CombineBytes(LowByte, HighByte);
    Data = Data - (int)(pow(2,16)); // formula for finding 2's complement
    // N = 2^bits - X, bits = # of bits

    return Data;
} // end Complement()

```

```

//-----
// ConvertData()
// converted AD6640 two's complement data into signed intergers
//-----
int ConvertData(BYTE LowByte, BYTE HighByte)
{
    int Buffer;

    if ((LowByte == 0)&&(HighByte == 0))
        Buffer = 0; // for 2's complement N = 0 and !N = 0
    else {
        if (HighByte&ADBit11) // D11 = 1 -> negative number
            Buffer = Complement(LowByte, HighByte);
        else
            Buffer = CombineBytes(LowByte, HighByte);
    } //end else

    return Buffer;
} // end ConvertData()

//-----
// Idle()
// a for loop to create some idle time
//-----
void Idle(DWORD Time)
{
    for (DWORD z=0; z<Time; z++); // wait
}

void CAboutDlg::OnOK()
{
    // TODO: Add extra validation here

    CDialog::OnOK();
    EndDialog(IDOK);
}

void CADportDlg::OnExit()
{
    // TODO: Add your control notification handler code here
    CWnd::DestroyWindow();
}

void CADportDlg::OnAbout()
{
    // TODO: Add your control notification handler code here
    CAboutDlg dlg;
    dlg.DoModal();
}

//-----
// SaveFile()
// procedure to save Buffered data on to disk
//-----
void SaveFile(CString filename, int Buffer[16*1024])
{
    //code to save data to disk
}

```

```

FILE *stream;    /* I/O stream file pointer */

/* Open file in text mode for write */
if((stream = fopen( filename, "w+t" )))
{
    for (DWORD j = 0; j < Count; j++)
        fprintf(stream, "%d\n", Buffer[j]);
        /* write one number w/ <CR> */

    if (fclose( stream )) /* close stream */
        AfxMessageBox("The file was not closed");
}
else
    AfxMessageBox( "Problem opening the file" );

} // end SaveFile()

/////////////////////////////////////////////////////////////////
// OnFileSaveAs()
// a function which handles the press of the SAVE key. First it will prompt
// the user for a file name, and then saves the data in Buffer to disk
/////////////////////////////////////////////////////////////////
void CADportDlg::OnFileSaveAs()
{
    // TODO: Add your control notification handler code here

    if (LPTNumber != 0)
        WriteCommand(8, LPTMaster); // disable write enable pin on FIFO (WEN),
        // Loadmode = STAQ = 0 -> DVN = 1
        // no need to write slave
    else
        Idle(10000); // wait for master to disable write

/////////////////////////////////////////////////////////////////
// Data READ CYCLE
/////////////////////////////////////////////////////////////////

    BYTE LowByteM, LowByteS; // lowbyte data for master and slave boards
    BYTE HighByteM, HighByteS; // highbyte data " " " " "

    for (DWORD k = 0; k<Count; k++) {
        WriteCommand(26, LPTMaster); // set OEA = 1, R = 1 (0001 1010)
        // output enable at latch and RCLK for FIFO
        WriteCommand(26, LPTSlave);

        WriteCommand(10, LPTMaster); // OEA = 1, R = 0 (0000 1010)
        // high byte output enabled & stop RCLK at FIFO)
        WriteCommand(10, LPTSlave);

        HighByteM = ReadData(LPTMaster); // ** Read HIGH byte data
        HighByteS = ReadData(LPTSlave);

        WriteCommand(8, LPTMaster); // OEA = 0, R = 0
        //(low byte output at latch...OEA = 1)
        WriteCommand(8, LPTSlave);

        LowByteM = ReadData(LPTMaster); // ** Read LOW byte data
        LowByteS = ReadData(LPTSlave);

        /* Convert 6640's data and store into Buffer arrays*/
        BufferM[k] = ConvertData(LowByteM, HighByteM);
    }
}

```

```

        BufferS[k] = ConvertData(LowByteS, HighByteS);
} // end for

pFileSave = new CFileDialog(FALSE, NULL, "scen", OFN_OVERWRITEPROMPT |
    OFN_PATHMUSTEXIST, "TMD files (*.tmd)|*.tmd|All Files (*.*)|*.*||");
if(pFileSave) {
    int click = pFileSave->DoModal();

    if (click != IDOK)
        return;

    CString filename = pFileSave->GetPathName();
    CString filenameM, filenameS;

    if (LPTNumber != 0) {
        filenameM = filename + ".M.tmd"; // filename for Master board data
        filenameS = filename + ".S.tmd"; // " " " " " "
    }
    else {
        filenameM = filename + "S1.tmd"; // filename for slave board 1 data
        filenameS = filename + "S2.tmd"; // " " " " " 2 "
    }

    //save buffered data to disk
    SaveFile(filenameM, BufferM);
    SaveFile(filenameS, BufferS);

    AfxMessageBox("files saved to disk");
} //endif

WriteCommand(142, LPTMaster); // initialize FIFOs and put them into standby mode
                               // (this reduces current drawn by 6620)
WriteCommand(142, LPTSlave);
Idle(500);
WriteCommand(113, LPTMaster);
WriteCommand(113, LPTSlave);

} //end OnFileSaveAs

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// OnRead()
// function which handles the pressing of READ button. It reads a block of
// data from FIFO, convert, and then stored to BUFER.
//
// NOTE: AD6640's OUPUTPUT BITS ARE TWO'S COMPLEMENT & D11 IS INVERTED
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
void CADportDlg::OnRead()
{
    // TODO: Add your control notification handler code here

    if (LPTNumber != 0)
        WriteCommand(44, LPTMaster); // Loadmode = 1, STAQ = 1 (load FIFO)
                                     // no need to write slave (DVN is disconnected)

    if (!(WaitForResponse(FIFOFULL_PIN, LPTMaster))) // wait...while FIFO not full
    {
        if (!(WaitForResponse(FIFOFULL_PIN, LPTSlave)))
            return; // return if both FIFO is empty
    }

    AfxMessageBox("Data Read Finished");
    UpdateData(FALSE);
}

```

```

} // end OnRead()

void CADportDlg::OnLpt1()
{
    // TODO: Add your control notification handler code here
    LPTNumber = 1;
    m_LPT1 = 0;
    m_LPT2 = m_none = -1;
    UpdateData(FALSE);        // redisplay new data
}

void CADportDlg::OnLpt2()
{
    // TODO: Add your control notification handler code here
    LPTNumber = 2;
    m_LPT2 = 0;
    m_LPT1 = m_none = -1;
    UpdateData(FALSE);        // redisplay new data
}

void CADportDlg::Onnone()
{
    // TODO: Add your control notification handler code here
    LPTNumber = 0;
    m_none = 0;
    m_LPT1 = m_LPT2 = -1;
    UpdateData(FALSE);        // redisplay new data
}

void CADportDlg::On1k()
{
    // TODO: Add your control notification handler code here
    Count = 1024;
    m_1k = 0;
    m_2k = m_8k = m_16k = -1;
    UpdateData(FALSE);
}

void CADportDlg::On2k()
{
    // TODO: Add your control notification handler code here
    Count = 2048;
    m_2k = 0;
    m_1k = m_8k = m_16k = -1;
    UpdateData(FALSE);
}

void CADportDlg::On8k()
{
    // TODO: Add your control notification handler code here
    Count = 8*1024;
    m_8k = 0;
    m_1k = m_2k = m_16k = -1;
    UpdateData(FALSE);
}

void CADportDlg::On16k()
{
    // TODO: Add your control notification handler code here
    Count = 16*1024;
    m_16k = 0;
}

```

```

        m_1k = m_2k = m_8k = -1;
        UpdateData(FALSE);
    }

void CADportDlg::OnInit()
{
    // TODO: Add your control notification handler code here
    SetLPTNumber(LPTNumber); // set Master Port address into LPTBase

    if (LPTMaster == 0x278) // set Slave port address
        LPTSlave = 0x378;
    else
        LPTSlave = 0x278; // if no master port or master at LPT1

    //////////////////////////////////////
    // Initialize and load data into FIFO
    //////////////////////////////////////

    BYTE ControlBits = 0; // initialize LPT port control register (0000 0100).
                          // set nWrite, nData strobe, nAddress strobe, Reset pins &
                          // enable output (BIT 5)

    DlPortWritePortUchar(LPTMaster+2, ControlBits); // send control bits to register
    DlPortWritePortUchar(LPTSlave+2, ControlBits);

    WriteCommand(142, LPTMaster); // initialize FIFOs and put them into standby mode

    WriteCommand(142, LPTSlave);

    Idle(500);
    // following commands used are obtained from EXPERIMENTATION

    WriteCommand(113, LPTMaster);
    WriteCommand(113, LPTSlave);
    Idle(2000);

    WriteCommand(0, LPTMaster); // reset FIFO -> MR = 0
    WriteCommand(0, LPTSlave);
    Idle(2000);

    WriteCommand(8, LPTMaster); // Loadmode = 0, STAQ = 0, MR = 1 (clear FIFO reset)
    WriteCommand(8, LPTSlave);
    Idle(1000);

    AfxMessageBox("Initialization DONE");
} // end OnInit()

```

Bibliography

- [1] American Microsystems, Inc. Spread spectrum ICs user's manual, April 1996.
- [2] Anderson et al. An Adaptive Array for Mobile Communication Systems. *IEEE Transactions on Vehicular Technology*, pages 230–236, February 1991.
- [3] M. Beach and J. McGeeham. Linearity Considerations in Adaptive Antenna Array applications. *Proceeding of Personal, Indoor & Mobile Radio Communication Conference*, pages 682–686, 1995.
- [4] D. Beal and G. Hill. A low cost CDMA transmitter using the AX602 ASIC, microcontroller and minimal RF circuitary. *RF Design*, pages 26–32, February 1995.
- [5] J. Bultitude and G. Bedal. Propagation Characteristics on Microcellular urban Mobile Radio Channels at 910 mhz. *IEEE Journal on Selected Areas in Communications*, pages 31–39, January 1989.
- [6] Carsello et al. IMT-2000 Standards: Radio Aspects. *IEEE Personal Communications*, pages 30–40, August 1997.
- [7] M. Cotton and P. Wilson. A Test Bed for the Evaluation of Adaptive Antennas. *Proceeding of 1998 International Symposium on Advanced Radio Technology*, 1998.
- [8] D.C. Cox. Delay Doppler Characteristics of Multipath Propagation at 910 mhz in a Suburban Mobile Radio Environment. *IEEE Transactions on Antennas and Propagation*, pages 625–635, September 1972.
- [9] D.C. Cox. 910 mhz Urban Mobile Radio Propagation: Multipath Characteristics in New York City. *IEEE Transactions on Communications*, pages 1188–1194, November 1973.
- [10] Dahlman et al. UMTS/IMT-2000 Based on Wideband CDMA. *IEEE Communications Magazine*, pages 70–80, September 1998.
- [11] Analog Devices. AD6620 Evaluation Board Manual. *Analog Devices data sheets*, 1998.

- [12] Analog Devices. AD6640 - IF sampling A/D converter. *Analog Devices data sheets*, 1998.
- [13] Analog Devices. Analog Devices expands RF product portfolio with new Direct I&Q modulator performing up to 2.5 GHz. Press Releases, May 1999.
- [14] R.C. Dixon. *Spread Spectrum Systems*. John Wiley & Sons, 1984.
- [15] M. Earnshaw. *An Investigation into Improving Performance of Cellular CDMA Communications Systems with Digital Beamforming*. PhD thesis, Queen's University, 1998.
- [16] M. Feuerstein and T.S. Rappaport. Path Loss, Delay Spread, and Outage Models as Functions of Antenna Height for Microcellular System Design. *IEEE Trans. on Vehicular Technology*, pages 487–498, August 1994.
- [17] J.P. Fritch. *Sunthetic Aperature Radar*. Springer-Verlag, 1988.
- [18] J.D. Gibson. *The Mobile Communications Handbook*. IEEE Press, 1996.
- [19] Gilhousen et al. On the Capacity of a Cellular System. *IEEE Transactions on Vehicular Technology*, pages 303–312, May 1991.
- [20] Grant et al. Adaptive arrays for narrowband CDMA base stations. *Electronics and Communication Engineering*, pages 156–166, August 1998.
- [21] CDMA Development Group. 3G Pavilion: Detailed Information. <http://www.cdg.org>, 2000.
- [22] J. Iinatti and M. Latva-aho. Matched Filter Acquisition for CDMA Systems in Multipath Channels. *IEEE GLOBECOM*, pages 3449–3454, 1998.
- [23] B. Jabbari and E. Dinan. Spreading Code for Direct Sequence CDMA Cellular Networks. *IEEE Communications Magazine*, pages 48–54, September 1998.
- [24] D. H. Johnson. *Array Signal Processing: concepts and techniques*. Prentice Hall, 1993.
- [25] Jorgensen et al. Application of Channel Sounding to CDMA PCS Design at 1900 MHz. *IEEE VTC conference*, pages 1937–1941, 1997.
- [26] L. Kahney. The Third-Generation Gap. *Scientific American*, October 2000.
- [27] J. Karimi. Personal Hand-held Communications Via L-band CDMA-based Geostationary Beamforming Satellites. Master's thesis, Queen's University, 1996.
- [28] J. Karimi and S. Blostein. Using Array Signal Processing to Improve Rural Area Coverage in Future Personal Satellite Communications Services. *European Transactions on Telecommunications*, to be appeared in 2001.

- [29] S. M. Kay. *Statistical Signal Processing - Detection Theory Vol. II*. Prentice Hall, 1998.
- [30] H. Krim and M. Viberg. Two Decades of Array Signal Processing Research. *IEEE Signal Processing Magazine*, pages 67–94, July 1996.
- [31] Communications Research Laboratory. PCS-Vector Channel Sounding Smart Antenna. *CITR annual conference*, 1998.
- [32] L.E. Larson. *RF and microwave circuit design for wireless communications*. Artech House, 1997.
- [33] J. Lau. Architectural Considerations of RF Circuits. *The Hong Kong University of Science & Technology*, September 1999.
- [34] J. Lee and L. Miller. *CDMA Systems Engineering Handbook*. Artech House, 1998.
- [35] W.C.Y. Lee. Overview of Cellular CDMA. *IEEE Transactions on Vehicular Technology*, pages 291–302, May 1991.
- [36] J.C. Liberti. *Smart Antennas for Wireless Communications*. Prentice Hall, 1999.
- [37] U. Madhow. Blind Adaptive Interference Suppression for Direct-Sequence CDMA. *IEEE Proceedings*, pages 2049–2069, October 1998.
- [38] Mini-Circuits. *RF/IF Designer's Guide*. Prentice Hall, 1999.
- [39] Miteq - microwave and components. Image rejection mixer datasheet, August 2000.
- [40] K. Morris and P. Kenington. A Broadband Linear Power Amplifier for Software Radio Applications. *IEEE VTC*, pages 2150–2154, 1998.
- [41] R. Muhamed and T.S. Rappaport. Direction of Arrival Estimation using Antenna Arrays. Master's thesis, Virginia Polytechnic Institute and State University, 1996.
- [42] Naguib et al. Capacity Improvement with Base-station Antenna Arrays in Cellular CDMA. *IEEE Transactions on Vehicular Technology*, pages 691–698, August 1994.
- [43] A. Naguib. *Adaptive Antennas for CDMA Wireless Networks*. PhD thesis, Stanford University, 1996.
- [44] S. Novis and L. Pelletier. RF PA requirements for third-generation systems. *RF Design*, page 38, January 1999.

- [45] T. Ojanpera and R. Prasad. An Overview of Air Interface Multiple Access for IMT-2000/UMTS. *IEEE Communications Magazine*, pages 82–95, September 1998.
- [46] G.T. Okamoto. *Smart Antenna Systems and Wireless LANs*. Kluwer Academic Publishers, 1999.
- [47] M. Oliphant. Radio interfacec makes the difference in 3G cellular systems. *IEEE Spectrum Magazine*, pages 53–58, October 2000.
- [48] L.Y. Pan. Beamforming in the Uplink and Downlink Channels of a Cellular CDMA Communication System. Master’s thesis, Queen’s University at Kingston, 1999.
- [49] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 1991.
- [50] C. Peacock. Interfacing the Enhanced Parallel Port. www.senet.com.au/cpeacock, 1997.
- [51] C. Peacock. Interfacing the Standard Parallel Port. www.senet.com.au/cpeacock, 1998.
- [52] Pickholtz et al. Spread Spectrum for Mobile Communications. *IEEE Transactions on Vehicular Technology*, pages 313–322, May 1991.
- [53] J. Proakis. *Digital Communications*. McGraw-Hill, 1995.
- [54] J. Proakis and D. Manolakis. *Digital Signal Processing - Principles, Algorithms and Applications*. Prentice Hall, 1996.
- [55] Rappaport et al. 900-MHz Multipath Propagation Measurements for U.S. Digital Cellular Radiotelephone. *IEEE Transactions on Vehicular Technology*, pages 132–139, May 1990.
- [56] T.S. Rappaport. *Wireless Communications*. Prentice Hall, 1999.
- [57] J. Salz and J. Winters. Effect of Fading Correlation on Adaptive Arrays in Digital Mobile Radio. *IEEE Transactions on Vehicular Technology*, pages 1049–1057, November 1994.
- [58] M. Sawahashi and F. Adachi. 1.92 Mbps data transmission experiments over a coherent W-CDMA mobile radio link. *Proceeding of IEEE VTC '98*, pages 1300–1304, 1998.
- [59] W. Sheen and G. Stuber. Effect of Multipath Fading on Delay-Locked Loops for Spread Spectrum Systems. *IEEE Transactions on Communications*, pages 1947–1956, Feb/Mar/Apr 1994.

- [60] C. Smith. *Practical Cellular & PCS Design*. McGraw-Hill, 1998.
- [61] J. Smith. *Modern communications circuits, 2ed.* McGraw-Hill, 1998.
- [62] J.O. Smith. *Sinusoids and Eponentials*. CCRMA, Stanford University, 2000.
- [63] Sousa et al. Delay Spread Measurements for the Digital Cellular Channel in Toronto. *IEEE transactions on Vehicular Technology*, pages 837–847, November 1994.
- [64] Swarts et al. *CDMA Techniques for Third Generation Mobile Systems*. Kluwer Academic Publishers, 1999.
- [65] C.Y. Tam et al. ELEC 490 Group 32 - Final report, April 1999.
- [66] N.T. Tin. Wideband CDMA Smart Antenna Measurement System. *20th Biennial Symposium on Communications, Queen's University*, pages 186–190, 2000.
- [67] International Telecommunication Union. What is IMT-2000? <http://www.itu.int>, 2000.
- [68] B. Van Veen and K. Buckley. Beamforming: A Versatile Approach to Spatial Filtering. *IEEE ASSP Magazine*, pages 4–24, April 1988.
- [69] Ward et al. Characterising the radio propagation channel for smart antenna systems. *Electronics & Communication Engineering Journal*, pages 191–200, August 1996.
- [70] J.A. Wepman. Analog-to-Digital Converters and Their Applications in Radio Receivers. *IEEE Communications Magazines*, pages 39–45, May 1995.
- [71] G. White. *Mobile Radio Technology*. Newnes, 1994.
- [72] J.H. Winters. Smart Antennas for Wireless Systems. *IEEE Personal Communications*, pages 23–27, February 1998.
- [73] Wu et al. On Channel Model Parameters for Microcellular CDMA Systems. *IEEE Transactions on Vehicular Technology*, pages 706–711, August 1995.
- [74] A. Wyglinski. Performance of CDMA System using Digital Beamforming with Mutual Coupling and Scattering Effects. Master's thesis, Queen's University at Kingston, 2000.
- [75] R. Ziemer and R. Peterson. *Digital Communications and Spread Spectrum Systems*. MacMillan, 1985.
- [76] R. Ziemer and W. Tranter. *Principles of Communications: Systems, Modulation and Noise*. Houghton Mifflin Company, 1995.

Vita

Noel Tai-Tung Tin

EDUCATION

M.Sc.	Electrical and Computer Engineering,	Queen's University,	1998–2000
B.Sc.Eng	Electrical and Computer Engineering,	Queen's University,	1994–1998
B.A.	Department of Economics,	Queen's University,	1995–1998

EXPERIENCE

RF Access Technology Planning Specialist, (2000–). Bell Mobility, Mississauga, ON.

Research Assistant, (1998–2000). Image Processing & Communications Laboratory, Queen's University.

Teaching Assistant, (1998–2000). Electrical and Computer Engineering, Queen's University.

Summer Research Assistant, (1998). Image Processing & Communications Laboratory, Queen's University.

PUBLICATIONS

N. T. Tin and S. D. Blostein. Wideband CDMA Smart Antenna Measurement System. *Proceedings of the Queen's 20th Biennial Symposium on Communications*, pages 186-190, 2000.

N. T. Tin and S.D. Blostein. Experiments on a Wideband CDMA Smart Antenna System. *CITR Annual Research Conference*, 2000.